

# Valuation Engineer Journal

*The engineering of value: measured, disciplined, reproducible.*

Volume 1 · Issue 1 · Summer 2026



## IN THIS ISSUE

The Doctrine–Practice Gap in Real Estate Appraisal  
Residual Constraint Approach (RCA): Framework & Protocol  
earthUI, glmnetUI & mgcvUI: Open Modeling Interfaces

**FOUNDATIONS (F0–F6)** Heterogeneous Good · Characteristics Space ·  
Hedonic Price Function · Implicit Price · Hedonic Regression · Latent Variable

**RECENT COURT DECISIONS** United States · European Union · Latin America

# Valuation Engineer Journal

Volume 1 · Issue 1  
Summer 2026  
ISSN 3143-5033

**Editor-in-Chief** William Bert Craytor, SRA

## Contents

|                                                                      |            |
|----------------------------------------------------------------------|------------|
| <b>Mission</b>                                                       | <b>2</b>   |
| <b>Editorial</b>                                                     | <b>3</b>   |
| Welcome to the Inaugural Issue . . . . .                             | 4          |
| <b>Letters to the Editor</b>                                         | <b>6</b>   |
| An Invitation to Correspond . . . . .                                | 7          |
| <b>Articles</b>                                                      | <b>8</b>   |
| The Doctrine–Practice Gap in Real Estate Appraisal . . . . .         | 9          |
| Residual Constraint Approach (RCA): Framework & Protocol . . .       | 40         |
| earthUI: An Interactive Interface for the earth (MARS) Package .     | 97         |
| glmnetUI: An Interactive Interface for the glmnet (Lasso and Elastic |            |
| Net) Package . . . . .                                               | 142        |
| mgcvUI: An Interactive Interface for the mgcv (Generalized Additive  |            |
| Models) Package . . . . .                                            | 178        |
| <b>Foundations</b>                                                   | <b>215</b> |
| Overview (F0) . . . . .                                              | 216        |
| Heterogeneous Good (F1) . . . . .                                    | 222        |
| Characteristics Space (F2) . . . . .                                 | 229        |
| Hedonic Price Function (F3) . . . . .                                | 236        |
| Implicit Price (F4) . . . . .                                        | 242        |
| Hedonic Regression (F5) . . . . .                                    | 249        |
| Latent Variable (F6) . . . . .                                       | 257        |
| <b>Recent Court Decisions</b>                                        | <b>264</b> |
| Recent Court Cases – US (R1) . . . . .                               | 265        |
| Recent EU Court Cases (R2) . . . . .                                 | 268        |
| Recent Latin America Court Cases (R3) . . . . .                      | 274        |

## Mission

The *Valuation Engineer Journal* publishes rigorous, practitioner-oriented research and case studies on the engineering of value: methods, models, and measurements that connect technical decisions to economic outcomes. We prioritise reproducibility, transparency of assumptions, and clarity of exposition.

# Editorial

DEPARTMENT: EDITORIAL

## Welcome to the Inaugural Issue

William Bert Craytor, SRA — Editor-in-Chief

Vol. 1, Issue 1 · Summer 2026

Published July 1, 2026

DOI: 10.67330/0tymfa32

© 2026 William Bert Craytor. Published by Pacific Vista Net.

This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 License.

**Abstract.** The Editors' Message for Volume 1, Issue 1: why this journal exists, what the five articles and three departments contribute, and an invitation to take part.

**Keywords:** editorial, valuation engineering, inaugural issue

Welcome to the inaugural issue of *The Valuation Engineer Journal*.

This journal exists because of a gap. Appraisal doctrine has long described valuation as the disciplined measurement of market behavior, yet everyday practice still rests largely on manual comparable selection and adjustments estimated by judgment alone. Meanwhile, the statistical tools capable of measuring what markets actually pay for property characteristics — hedonic regression, modern spline models, cross-validated machine learning — have matured for decades with little uptake in working appraisal. The valuation engineer is our name for the practitioner who closes that gap: licensed appraisal competence joined with data engineering, statistics, and reproducible protocol.

The five articles of this issue lay the working foundation. The opening article examines the doctrine–practice gap itself and argues that the profession's own standards already demand more measurement than its everyday methods deliver. Second — and central to everything this journal intends — the Residual Constraint Approach is presented in full: a protocol that lets the regression channel measure what can be measured, disciplines what cannot into an explicit residual breakdown, and constrains the total so that judgment can refine but never invent value. Three software articles follow, documenting earthUI and its companion interfaces for penalized regression and generalized additive models — the open tooling this journal builds on — shortly before press consolidated into a single application, as the publisher's notes in each article describe.

Three departments accompany the articles. *Foundations* opens with seven short entries (F0–F6) tracing the conceptual chain from heterogeneous goods through characteristics space, the hedonic price function, implicit prices, and hedonic regression, to the latent variables that live in residuals.

Each entry cites its scholarly lineage, carries its own DOI, and ends with a machine-readable Prolog formalization — the beginning of a knowledge base we intend to grow, issue by issue, into a precise shared vocabulary for the field. *Recent Court Decisions* surveys rulings from the United States, the European Union, and Latin America that bear on valuation practice, because the legal environment is part of the engineering problem. *Letters to the Editor* opens the correspondence column: methods that are meant to be used get tested in ways their authors never anticipated, and this is where those findings — and the disagreements they provoke — belong.

Every item in this issue is published open access under CC BY 4.0, in both PDF and HTML, with a Crossref DOI, and preserved independently on Zenodo. We invite contributions — articles, Foundations entries, correspondence, and disagreement — from appraisers, economists, and engineers alike.

William Bert Craytor, SRA  
*Editor-in-Chief*

## Letters to the Editor

DEPARTMENT: LETTERS TO THE EDITOR

## An Invitation to Correspond

William Bert Craytor, SRA — Editor-in-Chief

Vol. 1, Issue 1 · Summer 2026

Published July 1, 2026

DOI: 10.67330/r55mw896

© 2026 William Bert Craytor. Published by Pacific Vista Net.

This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 License.

**Abstract.** The editor opens the Letters column: what this journal hopes to receive from its readers, and how to send it.

**Keywords:** letters, correspondence, peer commentary

A journal that only speaks is half a journal. This column exists for the other half.

*The Valuation Engineer Journal* publishes methods that are meant to be used — and methods that are used get tested in ways their authors never anticipated. If you have applied the Residual Constraint Approach and found a step that fails in your market, we want the letter. If a Foundations entry defines a term more narrowly than your practice requires, we want the letter. If a court decision we summarized reads differently to a practitioner in that jurisdiction, we want that letter too. Disagreement, replication, refinement, and plain correction are all in scope; so are short reports from practice — a technique that worked, a data problem solved, a failure worth warning others about.

Ground rules are few. Letters should be signed, concise (a page is ideal), and specific — cite the article, section, or entry you are addressing, and give evidence the way you would in a workfile. We may edit for length and clarity, always with the author's approval. Letters are published open access under CC BY 4.0, like everything else in these pages, and authors whose work is discussed will be offered the chance to reply in the same issue.

Send letters through the journal site ([journal.valuation-engineer.com](http://journal.valuation-engineer.com)) or to the editor directly. The first real letters this journal receives will mark the moment it becomes a conversation rather than a publication — we are looking forward to them.

William Bert Craytor, SRA

*Editor-in-Chief*

## Articles

# The Doctrine–Practice Gap in Real Estate Appraisal: A Structured Account of Functions, Boundaries, and Tensions

William Bert Craytor

Vol. 1, Issue 1 · May 21, 2026

Published July 1, 2026

DOI: [10.67330/x7pbf13](https://doi.org/10.67330/x7pbf13)

© 2026 William Bert Craytor. Published by Pacific Vista Net.

This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

**Abstract.** Real estate appraisal occupies a position of significant public interest, yet its official doctrine—centered on opinions of market value under past and current conditions—sits in growing tension with how appraisal products are actually used in contemporary financial pipelines. This article builds that case in five stages, each paired with an explicit, queryable Prolog knowledge base: the first two establish the social and economic functions appraisal serves; the third identifies its primary end product, an opinion of market value; and the fourth examines appraisal’s temporal and methodological orientation relative to quantitative and financial forecasting, together with the regulatory boundary that separates them. These converge on the central claim of the fifth: in mortgage lending, market value scaled by a loan-to-value ratio functions as an implicit proxy for future foreclosure value—a forward-looking use the present-value doctrine does not acknowledge. Many appraisers, unaware of this collateral-valuation role, do not treat market value with the caution toward future value trends it warrants—an issue this account is meant to expose and to invite further work on.

**Keywords:** real estate appraisal, market value, valuation doctrine, knowledge representation, appraisal standards, foreclosure value

## Contents

|                                                                                            |                    |
|--------------------------------------------------------------------------------------------|--------------------|
| <a href="#">1 Social Importance</a>                                                        | <a href="#">1</a>  |
| <a href="#">2 Economic Importance</a>                                                      | <a href="#">6</a>  |
| <a href="#">3 The Importance of Market Value Estimates as an End Product of Appraisals</a> | <a href="#">10</a> |
| <a href="#">4 The Temporal Orientation of Appraisal Activity</a>                           | <a href="#">14</a> |
| <a href="#">5 Market Value × LTV as a Proxy for Foreclosure Value</a>                      | <a href="#">20</a> |
| <a href="#">6 Conclusion</a>                                                               | <a href="#">27</a> |

## 1 Social Importance

Real estate and other asset appraisals serve an essential social and economic function by helping buyers, sellers, lenders, insurers, investors, courts, taxing authorities, and the general public make informed and equitable decisions regarding the transfer, financing, taxation, and stewardship of property and other forms of wealth. High-quality appraisals contribute to market transparency, public confidence, efficient capital allocation, and the stability of financial institutions by

providing independent and well-supported opinions of value grounded in evidence, analysis, and professional judgment. (LaCour-Little and Malpezzi 2003; K'Akumu and Larsen 2024; Hendershott and Kane 1995; Lee, Peng, and Liao 2020)

Poor-quality appraisals, by contrast, can produce serious consequences at multiple time horizons. In the short term, inaccurate appraisals may cause individual buyers or sellers to suffer immediate financial harm through overpayment, underselling, excessive taxation, or inappropriate lending decisions. In the medium term, systematic appraisal deficiencies can distort market signals, impair credit underwriting, encourage speculative behavior, contribute to inequitable taxation or insurance coverage, and increase litigation and transactional distrust. Over the long term, persistent failures in appraisal quality may undermine confidence in markets and institutions, contribute to financial instability, misallocation of resources, neighborhood decline or overinvestment, and broader social inequities affecting households, businesses, and communities. (LaCour-Little and Malpezzi 2003; Mayer and Nothaft 2022)

Because real estate and other assets often represent a substantial portion of personal, corporate, and public wealth, appraisal quality is not merely a private technical concern, but a matter of significant public interest tied to economic resilience, fairness, and the long-term public good. (Mayer and Nothaft 2022)

```
%% appraisal_social.pl
%%
%% A Prolog knowledge base encoding the social and public-interest role
%% of real estate and other asset appraisals. Companion to
%% appraisal_economics.pl, which covers the strictly economic axis.
%%
%% All domain-specific predicates are prefixed with `social_` so this
%% module can be loaded alongside appraisal_economics.pl (and any other
%% companion KBs) without name collisions, and so cross-KB rules can
%% mention both vocabularies unambiguously.
%%
%% Conceptual model:
%%   - Appraisals support social DECISION TYPES.
%%   - Each decision involves identifiable PARTIES.
%%   - Appraisal QUALITY (high | low) drives EFFECTS across time HORIZONS.
%%   - Emphasis: equity, public confidence, public interest.
%%
%% Author: generated for Bert Craytor, ValEngr / RCA context.

:- module(appraisal_social,
    [ social_horizon/1
    , social_party/1
    , social_decision_type/1
    , social_involves/2           % DecisionType, Party
    , social_effect/4           % Quality, Horizon, Domain, Effect
    , social_contribution/1
    , social_hallmark/1
    , social_equity_principle/1
    , social_public_interest_claim/1
    , social_positive_effect/3
    , social_negative_effect/3
    , social_horizon_summary/2
    , social_consequences_of_quality/2
    , social_parties_affected_by/2
    , social_public_harm/2
    , social_private_harm/2
    , social_appraisal_is_public_interest/0
    ]).
```

```

:- discontinuous social_effect/4.
:- discontinuous social_involves/2.

%% -----
%% 1. Time horizons
%% -----

social_horizon(short_term).
social_horizon(medium_term).
social_horizon(long_term).

%% -----
%% 2. Parties whose decisions depend on appraisals
%%
%% The opening sentence enumerates these explicitly.
%% -----

social_party(buyers).
social_party(sellers).
social_party(lenders).
social_party(insurers).
social_party(investors).
social_party(courts).
social_party(taxing_authorities).
social_party(general_public).

%% -----
%% 3. Types of decisions appraisals inform
%% -----

social_decision_type(property_transfer).
social_decision_type(financing).
social_decision_type(taxation).
social_decision_type(stewardship).
social_decision_type(insurance_coverage).
social_decision_type(adjudication).           % courts: condemnation, divorce, estate
social_decision_type(public_administration).

social_involves(property_transfer,    buyers).
social_involves(property_transfer,    sellers).
social_involves(financing,            lenders).
social_involves(financing,            buyers).
social_involves(taxation,             taxing_authorities).
social_involves(taxation,             general_public).
social_involves(stewardship,          investors).
social_involves(stewardship,          general_public).
social_involves(insurance_coverage,   insurers).
social_involves(insurance_coverage,   buyers).
social_involves(adjudication,         courts).
social_involves(public_administration, taxing_authorities).
social_involves(public_administration, general_public).

%% -----
%% 4. What high-quality appraisals contribute (second sentence)
%% -----

social_contribution(market_transparency).

```

```

social_contribution(public_confidence).
social_contribution(efficient_capital_allocation).
social_contribution(stability_of_financial_institutions).

%% Characteristics of a high-quality appraisal opinion of value.
social_hallmark(independence).
social_hallmark(well_supported).
social_hallmark(grounded_in_evidence).
social_hallmark(grounded_in_analysis).
social_hallmark(grounded_in_professional_judgment).

%% -----
%% 5. Effects of appraisal quality by horizon
%% -----

%% ---- SHORT TERM: high quality ----
social_effect(high, short_term, property_transfer, fair_pricing_for_parties).
social_effect(high, short_term, financing, appropriate_lending_decisions).
social_effect(high, short_term, taxation, accurate_individual_assessment).

%% ---- SHORT TERM: low quality ----
social_effect(low, short_term, property_transfer, buyer_overpayment).
social_effect(low, short_term, property_transfer, seller_underselling).
social_effect(low, short_term, taxation, excessive_individual_taxation).
social_effect(low, short_term, financing,
    ↳ inappropriate_lending_decisions).
social_effect(low, short_term, property_transfer,
    ↳ immediate_financial_harm_to_individuals).

%% ---- MEDIUM TERM: high quality ----
social_effect(high, medium_term, markets, clear_market_signals).
social_effect(high, medium_term, financing, sound_credit_underwriting).
social_effect(high, medium_term, taxation, equitable_tax_assessment).
social_effect(high, medium_term, insurance_coverage,
    ↳ equitable_insurance_coverage).
social_effect(high, medium_term, adjudication, reduced_litigation_pressure).

%% ---- MEDIUM TERM: low quality ----
social_effect(low, medium_term, markets, distorted_market_signals).
social_effect(low, medium_term, financing,
    ↳ impaired_credit_underwriting).
social_effect(low, medium_term, markets,
    ↳ encouragement_of_speculation).
social_effect(low, medium_term, taxation, inequitable_taxation).
social_effect(low, medium_term, insurance_coverage,
    ↳ inequitable_insurance_coverage).
social_effect(low, medium_term, adjudication, increased_litigation).
social_effect(low, medium_term, markets, transactional_distrust).

%% ---- LONG TERM: high quality ----
social_effect(high, long_term, institutions, public_confidence_in_institutions).
social_effect(high, long_term, markets, public_confidence_in_markets).
social_effect(high, long_term, institutions, financial_stability).
social_effect(high, long_term, communities, equitable_community_outcomes).
social_effect(high, long_term, public_interest, economic_resilience).
social_effect(high, long_term, public_interest, social_fairness).
social_effect(high, long_term, public_interest, long_term_public_good).

```

```

%% ---- LONG TERM: low quality ----
social_effect(low, long_term, markets, eroded_market_confidence).
social_effect(low, long_term, institutions, eroded_institutional_confidence).
social_effect(low, long_term, institutions, financial_instability).
social_effect(low, long_term, markets, resource_misallocation).
social_effect(low, long_term, communities, neighborhood_decline).
social_effect(low, long_term, communities, neighborhood_overinvestment).
social_effect(low, long_term, communities, social_inequities_for_households).
social_effect(low, long_term, communities, social_inequities_for_businesses).
social_effect(low, long_term, communities, social_inequities_for_communities).

%% -----
%% 6. Equity principles and public-interest claims
%% -----

social_equity_principle(informed_decisions).
social_equity_principle(equitable_decisions).
social_equity_principle(fair_treatment_of_parties).
social_equity_principle(independence_of_valuer).

social_public_interest_claim(
    'Real estate and other assets are a substantial portion of personal, corporate,
    ↪ and public wealth.').
social_public_interest_claim(
    'Appraisal quality is not merely a private technical concern.').
social_public_interest_claim(
    'Appraisal quality is tied to economic resilience.').
social_public_interest_claim(
    'Appraisal quality is tied to fairness.').
social_public_interest_claim(
    'Appraisal quality is tied to the long-term public good.').

%% Top-level conclusion of the passage.
social_appraisal_is_public_interest :-
    social_public_interest_claim(_).    % at least one such claim exists

%% -----
%% 7. Derived / query predicates
%% -----

social_positive_effect(Horizon, Domain, Effect) :-
    social_effect(high, Horizon, Domain, Effect).

social_negative_effect(Horizon, Domain, Effect) :-
    social_effect(low, Horizon, Domain, Effect).

social_horizon_summary(Horizon, Effects) :-
    findall(Quality-Effect,
        social_effect(Quality, Horizon, _Domain, Effect),
        Effects).

social_consequences_of_quality(Quality, List) :-
    findall(consequence(Horizon, Domain, Effect),
        social_effect(Quality, Horizon, Domain, Effect),
        List).

social_parties_affected_by(DecisionType, Parties) :-
    setof(P, social_involves(DecisionType, P), Parties).

```

```

% Harms that fall on the public rather than on transaction principals.
social_public_harm(Horizon, Effect) :-
    social_effect(low, Horizon, Domain, Effect),
    member(Domain, [markets, institutions, communities, public_interest,
        ↪ taxation]).

% Private (transaction-level) harms.
social_private_harm(Horizon, Effect) :-
    social_effect(low, Horizon, Domain, Effect),
    member(Domain, [property_transfer, financing, insurance_coverage]).

```

## 2 Economic Importance

Real estate and other asset appraisals play a fundamental economic role by supporting the efficient operation of credit markets, investment decisions, taxation systems, financial reporting, insurance underwriting, estate administration, corporate transactions, and public policy planning. Reliable appraisals help establish credible measures of collateral and market value that allow capital to flow more efficiently between borrowers, lenders, investors, governments, and businesses. (LaCour-Little and Malpezzi 2003; Ebrahim and Hussain 2010)

In the short term, accurate appraisals reduce transactional uncertainty, improve lending decisions, facilitate fair negotiations, and help prevent immediate financial losses arising from overvaluation or undervaluation. In the medium term, appraisal quality influences the stability and efficiency of broader markets by affecting mortgage risk, investment allocation, tax equity, insurance adequacy, and the pricing behavior of market participants. Weak or systematically biased appraisals during this period can contribute to inflated asset bubbles, impaired lending portfolios, distorted development incentives, and increased default or litigation risk. (LaCour-Little and Malpezzi 2003; Hendershott and Kane 1992; Griffin and Maturana 2016)

Over the long term, the cumulative economic consequences of appraisal quality become even more significant, influencing patterns of capital formation, infrastructure investment, intergenerational wealth transfer, public revenues, housing affordability, and overall financial system resilience. Sustained deficiencies in valuation practices can contribute to chronic misallocation of capital, banking instability, reduced investor confidence, and inefficient use of land and productive assets, while strong appraisal standards help support stable markets, sound economic planning, and long-term economic growth. (Ghent, Torous, and Valkanov 2019; Hendershott and Kane 1992; Ebrahim and Hussain 2010)

```

% appraisal_economics.pl
%
% A Prolog knowledge base encoding the economic role of real estate
% and other asset appraisals across short-, medium-, and long-term
% horizons, including both the consequences of high-quality appraisal
% practice and the consequences of weak or biased practice.
%
% All domain-specific predicates are prefixed with `economic_` so this
% module can be loaded alongside appraisal_social.pl (and any other
% companion KBs) without name collisions, and so cross-KB rules can
% mention both vocabularies unambiguously.
%
% Conceptual model:
% - Appraisals support a set of economic FUNCTIONS.
% - Each function operates over one or more time HORIZONS.
% - Appraisal QUALITY (high / low) drives EFFECTS on those functions.

```

```

%% - Effects are POSITIVE (benefits) when quality is high and
%%      NEGATIVE (harms) when quality is low/systematically biased.
%%
%% Author: generated for Bert Craytor, ValEngr / RCA context.

:- module(appraisal_economics,
    [ economic_horizon/1
      , economic_function/1
      , economic_supports/2
      , economic_effect/4
      ↪ Effect
      , economic_stakeholder/1
      , economic_benefits/2
      , economic_positive_effect/3
      , economic_negative_effect/3
      , economic_horizon_summary/2
      , economic_consequences_of_quality/2
      , economic_functions_at_horizon/2
      , economic_stakeholders_affected_by/2
      , economic_reliable_appraisals_improve/1
      , economic_appraisal_quality_matters_at/1
    ]).

:- discontiguous economic_effect/4.
:- discontiguous economic_supports/2.

%% -----
%% 1. Time horizons
%% -----

economic_horizon(short_term).
economic_horizon(medium_term).
economic_horizon(long_term).

%% -----
%% 2. Economic functions supported by appraisals
%% -----

economic_function(credit_markets).
economic_function(investment_decisions).
economic_function(taxation_systems).
economic_function(financial_reporting).
economic_function(insurance_underwriting).
economic_function(estate_administration).
economic_function(corporate_transactions).
economic_function(public_policy_planning).

%% -----
%% 3. Stakeholders whose decisions depend on appraisals
%% -----

economic_stakeholder(borrowers).
economic_stakeholder(lenders).
economic_stakeholder(investors).
economic_stakeholder(governments).
economic_stakeholder(businesses).
economic_stakeholder(insurers).
economic_stakeholder(taxpayers).

```

```

economic_stakeholder(homeowners).

%% Capital flows more efficiently when appraisals are reliable.
economic_benefits(borrowers,    credit_markets).
economic_benefits(lenders,      credit_markets).
economic_benefits(investors,    investment_decisions).
economic_benefits(governments,  taxation_systems).
economic_benefits(governments,  public_policy_planning).
economic_benefits(businesses,   corporate_transactions).
economic_benefits(businesses,   financial_reporting).
economic_benefits(insurers,     insurance_underwriting).
economic_benefits(taxpayers,    taxation_systems).
economic_benefits(homeowners,   credit_markets).

%% -----
%% 4. Which functions are most salient at each horizon
%% -----

economic_supports(credit_markets,    short_term).
economic_supports(investment_decisions, short_term).
economic_supports(corporate_transactions, short_term).

economic_supports(credit_markets,    medium_term).
economic_supports(investment_decisions, medium_term).
economic_supports(taxation_systems,   medium_term).
economic_supports(insurance_underwriting, medium_term).

economic_supports(credit_markets,    long_term).
economic_supports(taxation_systems,   long_term).
economic_supports(public_policy_planning, long_term).
economic_supports(financial_reporting, long_term).
economic_supports(estate_administration, long_term).

%% -----
%% 5. Effects of appraisal quality, by horizon and function
%% -----

%% ---- SHORT TERM: high quality ----
economic_effect(high, short_term, credit_markets,
    ↪ reduced_transactional_uncertainty).
economic_effect(high, short_term, credit_markets,
    ↪ improved_lending_decisions).
economic_effect(high, short_term, corporate_transactions, fair_negotiations).
economic_effect(high, short_term, investment_decisions,
    ↪ prevention_of_immediate_losses).

%% ---- SHORT TERM: low quality ----
economic_effect(low, short_term, credit_markets,
    ↪ transactional_uncertainty).
economic_effect(low, short_term, credit_markets,                poor_lending_decisions).
economic_effect(low, short_term, investment_decisions,
    ↪ losses_from_overvaluation).
economic_effect(low, short_term, investment_decisions,
    ↪ losses_from_undervaluation).

%% ---- MEDIUM TERM: high quality ----
economic_effect(high, medium_term, credit_markets,                stable_mortgage_risk).

```

```

economic_effect(high, medium_term, investment_decisions,
↳ efficient_investment_allocation).
economic_effect(high, medium_term, taxation_systems,      tax_equity).
economic_effect(high, medium_term, insurance_underwriting,
↳ adequate_insurance_coverage).
economic_effect(high, medium_term, investment_decisions,
↳ rational_market_pricing).

%% ---- MEDIUM TERM: low quality ----
economic_effect(low, medium_term, investment_decisions,   inflated_asset_bubbles).
economic_effect(low, medium_term, credit_markets,
↳ impaired_lending_portfolios).
economic_effect(low, medium_term, public_policy_planning,
↳ distorted_development_incentives).
economic_effect(low, medium_term, credit_markets,         increased_default_risk).
economic_effect(low, medium_term, corporate_transactions,
↳ increased_litigation_risk).

%% ---- LONG TERM: high quality ----
economic_effect(high, long_term, investment_decisions,    sound_capital_formation).
economic_effect(high, long_term, public_policy_planning,
↳ sound_infrastructure_investment).
economic_effect(high, long_term, estate_administration,
↳ orderly_intergenerational_wealth_transfer).
economic_effect(high, long_term, taxation_systems,        stable_public_revenues).
economic_effect(high, long_term, public_policy_planning,
↳ supported_housing_affordability).
economic_effect(high, long_term, credit_markets,
↳ financial_system_resilience).
economic_effect(high, long_term, investment_decisions,
↳ long_term_economic_growth).

%% ---- LONG TERM: low quality ----
economic_effect(low, long_term, investment_decisions,
↳ chronic_capital_misallocation).
economic_effect(low, long_term, credit_markets,           banking_instability).
economic_effect(low, long_term, investment_decisions,
↳ reduced_investor_confidence).
economic_effect(low, long_term, public_policy_planning,   inefficient_land_use).
economic_effect(low, long_term, public_policy_planning,
↳ inefficient_productive_asset_use).

%% -----
%% 6. Derived / query predicates
%% -----

economic_positive_effect(Horizon, Function, Effect) :-
    economic_effect(high, Horizon, Function, Effect).

economic_negative_effect(Horizon, Function, Effect) :-
    economic_effect(low, Horizon, Function, Effect).

economic_horizon_summary(Horizon, Effects) :-
    findall(Quality-Effect,
        economic_effect(Quality, Horizon, _Function, Effect),
        Effects).

economic_consequences_of_quality(Quality, List) :-

```

```

    findall(consequence(Horizon, Function, Effect),
            economic_effect(Quality, Horizon, Function, Effect),
            List).

economic_functions_at_horizon(Horizon, Functions) :-
    setof(F, economic_supports(F, Horizon), Functions).

economic_stakeholders_affected_by(Function, Stakeholders) :-
    setof(S, economic_benefits(S, Function), Stakeholders).

%% Overall claim: reliable appraisals improve allocative efficiency.
economic_reliable_appraisals_improve(capital_flow_efficiency).
economic_reliable_appraisals_improve(market_stability).
economic_reliable_appraisals_improve(long_term_economic_growth).

%% Top-level qualitative claim: appraisal quality matters at every horizon.
economic_appraisal_quality_matters_at(Horizon) :-
    economic_horizon(Horizon),
    \+ \+ economic_effect(high, Horizon, _, _),
    \+ \+ economic_effect(low, Horizon, _, _).

```

### 3 The Importance of Market Value Estimates as an End Product of Appraisals

Market value appraisals play an important role in collateral risk analysis because they provide a professionally developed estimate of the probable price a property or asset could achieve under normal market conditions, thereby serving as a foundational benchmark for lenders, investors, and collateral analysts evaluating credit exposure and potential foreclosure outcomes. (Agarwal, Ben-David, and Yao 2015) Although foreclosure or liquidation value may differ substantially from market value due to distressed marketing conditions, accelerated sale periods, legal costs, physical deterioration, or adverse market sentiment, collateral analysts frequently rely upon market value appraisals as the starting point for estimating recovery potential under stressed scenarios. (Campbell, Giglio, and Pathak 2011; Pennington-Cross 2006; Clauretie and Daneshvary 2009)

By analyzing the relationship between market value, loan balances, loan-to-value ratios, market trends, property condition, liquidity, and local supply-demand conditions, collateral analysts assess the likelihood and severity of losses in the event of borrower default. (Qi and Yang 2009)

In the short term, accurate market value appraisals help lenders make prudent underwriting decisions and establish appropriate collateral margins. In the medium term, they support portfolio risk management, loan securitization, reserve allocation, and regulatory compliance by helping institutions evaluate changing collateral adequacy across economic cycles. Over the long term, consistently reliable appraisal practices contribute to financial system stability by reducing the risk of excessive leverage, poorly secured lending, and systemic mispricing of collateral assets that can amplify foreclosure losses and broader economic disruptions during periods of market stress. (Eriksen et al. 2019)

```

%% appraisal_collateral.pl
%%
%% A Prolog knowledge base encoding the role of market value (MV)
%% appraisals in collateral risk analysis. Companion to
%% appraisal_economics.pl and appraisal_social.pl.
%%

```

```

%% Naming convention:
%%   - Predicates that have direct counterparts in the companion KBs
%%     (horizon, effect, summary, consequences, etc.) are prefixed
%%     `collateral_` so this module can be loaded alongside the others
%%     without name collisions, and so bridge rules can mention each
%%     vocabulary unambiguously.
%%   - Predicates that are unique to the collateral-analysis domain
%%     (value_basis, divergence_factor, risk_input, etc.) are NOT
%%     prefixed. They have no analogues in the other KBs and no
%%     collision risk, so the prefix would only add noise.

:- module(appraisal_collateral,
    [ collateral_horizon/1
    , collateral_analyst/1
    , collateral_effect/4                                % Quality, Horizon, Domain,
    ↪ Effect
    , collateral_positive_effect/3
    , collateral_negative_effect/3
    , collateral_horizon_summary/2
    , collateral_consequences_of_quality/2
    , value_basis/1
    , uses/2                                              % Analyst, ValueBasis
    , divergence_factor/1
    , stress_scenario/1
    , risk_input/1
    , risk_output/1
    , inputs_for/2
    , mv_is_benchmark_for/1
    ]).

:- discontiguous collateral_effect/4.
:- discontiguous uses/2.

%% -----
%% 1. Time horizons (namespaced -- parallel to economic/social_)
%% -----

collateral_horizon(short_term).
collateral_horizon(medium_term).
collateral_horizon(long_term).

%% -----
%% 2. Who uses MV appraisals in collateral risk analysis
%% -----

collateral_analyst(lenders).
collateral_analyst(investors).
collateral_analyst(collateral_analysts).
collateral_analyst(portfolio_risk_managers).
collateral_analyst(regulators).
collateral_analyst(securitization_arrangers).

%% -----
%% 3. Value bases distinguished in the passage
%% -----

value_basis(market_value).          % "normal market conditions"
value_basis(foreclosure_value).     % distressed

```

```

value_basis(liquidation_value).      % distressed, accelerated

%% Which value bases each analyst type works with.
uses(lenders,                        market_value).
uses(investors,                      market_value).
uses(collateral_analysts,           market_value).
uses(collateral_analysts,           foreclosure_value).
uses(collateral_analysts,           liquidation_value).
uses(portfolio_risk_managers,        market_value).
uses(securitization_arrangers,       market_value).
uses(regulators,                    market_value).

%% Top-level claim of the opening sentence.
mv_is_benchmark_for(credit_exposure).
mv_is_benchmark_for(foreclosure_outcomes).
mv_is_benchmark_for(recovery_potential).

%% -----
%% 4. Why foreclosure / liquidation value diverges from MV
%% -----

divergence_factor(distressed_marketing_conditions).
divergence_factor(accelerated_sale_period).
divergence_factor(legal_costs).
divergence_factor(physical_deterioration).
divergence_factor(adverse_market_sentiment).

%% Stressed scenarios in which divergence factors apply.
stress_scenario(borrower_default).
stress_scenario(foreclosure).
stress_scenario(forced_liquidation).
stress_scenario(market_downturn).

%% -----
%% 5. Inputs the collateral analyst combines with MV, and the
%%     outputs they produce
%% -----

risk_input(market_value).
risk_input(loan_balance).
risk_input(loan_to_value_ratio).
risk_input(market_trends).
risk_input(property_condition).
risk_input(liquidity).
risk_input(local_supply_demand).

risk_output(likelihood_of_loss).
risk_output(severity_of_loss).
risk_output(recovery_estimate).
risk_output(loss_given_default).

%% inputs_for(+Output, -Inputs)
%% All risk inputs are used to produce each risk output.
inputs_for(Output, Inputs) :-
    risk_output(Output),
    setof(I, risk_input(I), Inputs).

%% -----

```

```

%% 6. Effects of appraisal quality by horizon
%% -----

%% ---- SHORT TERM: high quality ----
collateral_effect(high, short_term, underwriting, prudent_underwriting_decisions).
collateral_effect(high, short_term, underwriting, appropriate_collateral_margins).

%% ---- SHORT TERM: low quality ----
collateral_effect(low, short_term, underwriting,
↳ imprudent_underwriting_decisions).
collateral_effect(low, short_term, underwriting, inadequate_collateral_margins).
collateral_effect(low, short_term, underwriting, mispriced_credit_exposure).

%% ---- MEDIUM TERM: high quality ----
collateral_effect(high, medium_term, portfolio,
↳ effective_portfolio_risk_management).
collateral_effect(high, medium_term, portfolio, sound_loan_securitization).
collateral_effect(high, medium_term, portfolio, appropriate_reserve_allocation).
collateral_effect(high, medium_term, regulatory, regulatory_compliance).
collateral_effect(high, medium_term, portfolio,
↳ cycle_aware_collateral_evaluation).

%% ---- MEDIUM TERM: low quality ----
collateral_effect(low, medium_term, portfolio,
↳ weakened_portfolio_risk_management).
collateral_effect(low, medium_term, portfolio, mispriced_securitizations).
collateral_effect(low, medium_term, portfolio, inadequate_loss_reserves).
collateral_effect(low, medium_term, regulatory, regulatory_noncompliance_risk).
collateral_effect(low, medium_term, portfolio, misjudged_collateral_adequacy).

%% ---- LONG TERM: high quality ----
collateral_effect(high, long_term, system, financial_system_stability).
collateral_effect(high, long_term, system, restraint_on_excessive_leverage).
collateral_effect(high, long_term, system, well_secured_lending).
collateral_effect(high, long_term, system, accurate_systemic_collateral_pricing).
collateral_effect(high, long_term, system, dampened_foreclosure_amplification).

%% ---- LONG TERM: low quality ----
collateral_effect(low, long_term, system, financial_system_instability).
collateral_effect(low, long_term, system, excessive_leverage).
collateral_effect(low, long_term, system, poorly_secured_lending).
collateral_effect(low, long_term, system, systemic_collateral_mispricing).
collateral_effect(low, long_term, system, amplified_foreclosure_losses).
collateral_effect(low, long_term, system,
↳ amplified_economic_disruption_under_stress).

%% -----
%% 7. Derived / query predicates (namespaced)
%% -----

collateral_positive_effect(Horizon, Domain, Effect) :-
    collateral_effect(high, Horizon, Domain, Effect).

collateral_negative_effect(Horizon, Domain, Effect) :-
    collateral_effect(low, Horizon, Domain, Effect).

collateral_horizon_summary(Horizon, Effects) :-
    findall(Quality-Effect,

```

```

collateral_effect(Quality, Horizon, _Domain, Effect),
Effects).

collateral_consequences_of_quality(Quality, List) :-
    findall(consequence(Horizon, Domain, Effect),
            collateral_effect(Quality, Horizon, Domain, Effect),
            List).

```

## 4 The Temporal Orientation of Appraisal Activity

Appraisers generally develop point-in-time opinions of market value as of a specific effective date—retrospective (past), current, or, with defined limitations, prospective (future)—rather than probabilistic forecasts over extended time periods. This temporal and methodological orientation represents one of the central distinctions between regulated appraisal practice and the work of quantitative analysts, financial analysts, investment analysts, and economists. (Quan and Quigley 1991)

In most appraisal assignments, particularly those involving market value under standards such as the Uniform Standards of Professional Appraisal Practice (USPAP) or International Valuation Standards (IVS), the appraiser produces a reconciled point estimate (or, in permitted cases, a narrow range) anchored to observable market evidence, prevailing conditions, and recognized methodologies as of one effective date. USPAP explicitly permits value conclusions as a specific amount, a range of numbers, or a relationship to a benchmark, yet the reconciled opinion remains a credible point indication tied to that date. (The Appraisal Foundation 2024; International Valuation Standards Council 2025)

Although appraisers necessarily incorporate market participants' reasonable expectations regarding trends, supply and demand, depreciation, capitalization or discount rates, and absorption, they do so to support a market-supported value at the effective date, not to forecast speculative movements. Prospective value opinions (e.g., stabilized value upon project completion) require explicit extraordinary assumptions or hypothetical conditions and remain constrained to a defined future effective date with market-derived support.

By contrast, quantitative, financial, and investment analysts frequently address uncertainty through estimates covering a range of possible outcomes across time periods, employing probability distributions, Monte Carlo simulations, scenario analysis, sensitivity testing, or value-at-risk metrics. Their focus often centers on expected future performance, risk-adjusted returns, or investment value to a specific party over varying horizons, rather than a single market value point at one date. (Hull 2018)

Such work also presupposes technical competencies that an appraisal license neither requires nor certifies. Fitting a multivariate adaptive regression spline (MARS) model, for instance, calls for the judgment to control model complexity so that strong in-sample fit (a high  $R^2$ ) is not bought at the cost of overfitting but instead carries over to held-out data (a high cross-validated  $R^2$ ), which in turn presumes a working understanding of the underlying algorithms. (Friedman 1991; Steurer, Hill, and Pfeifer 2021) Alongside such quantitative facility, the work rewards the conceptual capacity to handle highly structured concepts, to discern fine distinctions in meaning, and to assimilate unfamiliar methods. These are skills rather than credentials: a licensed appraiser may possess them and an unlicensed analyst may lack them, which is exactly why they do not, by themselves, determine on which side of the regulatory line a given task falls.

Appraisers may also engage in limited future-oriented elements, such as projecting income streams, vacancy, expenses, and reversionary values in discounted cash flow analyses, or estimating prospective values in development or litigation contexts. These components stay within appraisal bounds when grounded in current market evidence and professional standards that emphasize supportability and avoidance of unsupported speculation. (Diaz and Wolverton 1998; Gallimore

1996; Crosby, Lizieri, and McAllister 2010) Modern appraisal practice increasingly integrates statistical tools, scenario considerations, and risk analytics—particularly in mass appraisal, automated valuation models, or portfolio work—yet the core deliverable typically remains a point-in-time market value opinion. (International Association of Assessing Officers 2017; Office of the Comptroller of the Currency and Board of Governors of the Federal Reserve System and Federal Deposit Insurance Corporation and National Credit Union Administration and Consumer Financial Protection Bureau and Federal Housing Finance Agency 2024; Alexandrov, Goodman, and Neal 2023)

One might characterize appraisal practice as primarily evidentiary and oriented toward past-to-present (or limited prospective) point estimates, whereas financial and quantitative analysis is more probabilistic, distributional, and explicitly future-oriented across time horizons. (Geltner 1991) Appraisers ask, “What is the most supportable market value indication as of the specified effective date, given current evidence and defined assumptions?” Financial and quantitative analysts more often inquire, “What range of outcomes is likely over a future period, with what probabilities, and how should resources be allocated?”

The boundary is not absolute and continues to evolve with market complexity. Nevertheless, fundamental differences persist in purpose (market value vs. investment worth), evidentiary requirements, and regulatory frameworks.

The operative line, however, is drawn not by skill but by law: the defining question is simply whether a given task is one for which an appraiser’s license is legally required. Quantitative analysts, financial analysts, economists, investment analysts, portfolio managers, and related professionals may legally estimate, forecast, model, or analyze future asset values—including distributional or range-based projections—without a real estate appraiser’s license, provided the work does not constitute a regulated appraisal. (Murphy 2012) The key distinctions rest on:

1. the type of value (e.g., a market value point estimate vs. investment value or a probabilistic forecast);
2. the temporal focus (a specific effective date vs. a time period or horizon);
3. the form of output (a point or narrow range vs. a full probability distribution);
4. the intended purpose and whether the work is represented as an appraisal; and
5. involvement of regulated lending or legal contexts requiring compliance with standards such as USPAP.

This framework originated largely from banking regulations such as FIRREA (1989) in the United States, which sought to ensure credible, independent market value opinions for federally related transactions. (United States Congress 1989; K’Akumu and Larsen 2024) Many sophisticated valuation activities—equity research price targets, feasibility studies, portfolio risk modeling, option pricing, or predictive analytics—fall outside licensure because they serve investment, forecasting, or advisory purposes rather than regulated appraisal functions.

```
%% appraisal_scope_regulation.pl
%%
%% Knowledge base for the boundary of regulated appraisal practice.
%%
%% TWO TIERS, deliberately kept separate:
%%
%% TIER 1 -- THE DEFINING TEST (regulation cluster). A task is a
%% regulated appraisal IFF it is one for which an appraiser's
%% license is legally required. requires_license/1 is that test;
%% it -- and only it -- decides regulated_appraisal/1 vs.
%% unregulated_valuation/1.
%%
%% TIER 2 -- DESCRIPTIVE TENDENCIES (scope cluster). How the two
%% populations -- `appraisal' and `quant_financial' -- typically
%% differ in approach, temporal scope, output form, and skill set.
```

```

%%      These are CORRELATES, not the boundary: the populations overlap
%%      (a licensed appraiser may do quant work; a quant may be
%%      unlicensed), and nothing here determines licensure. No claim is
%%      made about intelligence -- only about training and methodology.
%%
%%  `temporal scope' and `approach' are DISTINCT axes: an approach has a
%%  typical (defeasible) temporal mode, but the two are not the same
%%  viewpoint -- which is why an appraiser using an AVM (appraisal
%%  approach, distributional methods) is possible.

:- module(appraisal_scope_regulation,
  [ % --- TIER 1: the defining test (regulation) ---
    regulation_framework/1
  , regulation_framework_origin/2
  , regulation_distinguishing_factor/1
  , regulation_triggers_licensure/1
  , regulation_definition_of_appraisal/1
  , regulation_unlicensed_activity/2
  , regulation_licensed_activity/2
  , regulation_legal_exposure_condition/1
  , requires_license/1
  , exempt_from_appraisal_license/1
  , regulated_appraisal/1
  , unregulated_valuation/1
  , regulatory_asymmetry/2

    % --- TIER 2: descriptive tendencies (scope) ---
  , valuation_approach/1
  , scope_profession/1
  , scope_orientation/2
  , temporal_scope/1
  , temporal_mode/1
  , scope_temporal_focus/2
  , scope_output_form/2
  , scope_skill/1
  , scope_skill_demanded_by/2
  , scope_inputs_considered/2
  , scope_future_oriented_context/2
  , scope_professional_constraint/1
  , scope_contrast/4
  , scope_question/2
  , scope_convergence_area/1
  ]).

:- discontiguous regulation_unlicensed_activity/2.
:- discontiguous regulation_licensed_activity/2.
:- discontiguous scope_contrast/4.

%% =====
%% TIER 1 -- THE DEFINING TEST (the boundary)
%% =====

regulation_framework(firrea_1989).
regulation_framework(uspap).
regulation_framework(ivs).
regulation_framework(state_licensing).
regulation_framework(state_certification).

```

```

regulation_framework_origin(firrea_1989,      savings_and_loan_crisis).
regulation_framework_origin(uspap,           firrea_1989).
regulation_framework_origin(state_licensing, firrea_1989).

regulation_distinguishing_factor(kind_of_value).
regulation_distinguishing_factor(purpose_of_valuation).
regulation_distinguishing_factor(temporal_focus).
regulation_distinguishing_factor(output_form).
regulation_distinguishing_factor(represented_as_appraisal).
regulation_distinguishing_factor(federally_related_transaction_involvement).

regulation_triggers_licensure(federally_related_mortgage_transaction).
regulation_triggers_licensure(work_represented_as_appraisal).
regulation_triggers_licensure(work_meeting_state_appraisal_definition).
regulation_triggers_licensure(regulated_lending_transaction).

regulation_definition_of_appraisal(formal_opinion_of_value).
regulation_definition_of_appraisal(concerning_identified_real_property).
regulation_definition_of_appraisal(developed_per_recognized_methods).
regulation_definition_of_appraisal(for_compensation).

regulation_unlicensed_activity(equity_analyst,      stock_price_targets).
regulation_unlicensed_activity(investment_bank,
    ↪ enterprise_value_estimates).
regulation_unlicensed_activity(hedge_fund,
    ↪ commercial_real_estate_appreciation_forecasts).
regulation_unlicensed_activity(economist,
    ↪ housing_market_trend_projections).
regulation_unlicensed_activity(insurance_analyst,
    ↪ replacement_cost_estimates).
regulation_unlicensed_activity(developer,           feasibility_studies).
regulation_unlicensed_activity(internal_corporate_analyst,
    ↪ asset_impairment_estimates).
regulation_unlicensed_activity(internal_corporate_analyst,
    ↪ portfolio_value_estimates).
regulation_unlicensed_activity(quantitative_analyst,
    ↪ predictive_valuation_models).
regulation_unlicensed_activity(avm_developer,
    ↪ algorithmic_residential_price_estimates).

regulation_licensed_activity(licensed_appraiser,
    ↪ market_value_for_federally_related_transaction).
regulation_licensed_activity(licensed_appraiser,
    ↪ formal_opinion_of_value_for_compensation).
regulation_licensed_activity(licensed_appraiser,
    ↪ appraisal_for_regulated_mortgage_lending).
regulation_licensed_activity(licensed_appraiser,
    ↪ appraisal_represented_as_such_under_state_law).

regulation_legal_exposure_condition(holding_self_out_as_licensed_appraiser).
regulation_legal_exposure_condition(preparing_appraisal_as_defined_by_state_law).
regulation_legal_exposure_condition(involved_in_transaction_requiring_licensed_app
    ↪ raisal).

%% --- The test itself: does the TASK require an appraiser's license? ---
requires_license(Task) :-
    regulation_triggers_licensure(Task), !.
requires_license(Task) :-

```

```

    regulation_licensed_activity(_, Task), !.

exempt_from_appraisal_license(Task) :-
    regulation_unlicensed_activity(_, Task),
    \+ requires_license(Task).

:- dynamic activity_satisfies_definition/1.

activity_satisfies_definition(Task) :-
    regulation_licensed_activity(_, Task).

regulated_appraisal(Task) :-
    activity_satisfies_definition(Task),
    requires_license(Task).

unregulated_valuation(Task) :-
    regulation_unlicensed_activity(_, Task),
    \+ regulated_appraisal(Task).

%% Meta-tension: the regulatory asymmetry the passage closes on.
regulatory_asymmetry(current_market_value_for_lending,
    future_oriented_valuation_for_investment).
regulatory_asymmetry(heavily_regulated_appraisal_practice,
    lightly_regulated_quantitative_practice).
regulatory_asymmetry(consumer_and_lending_protection_focus,
    investment_advisory_focus).

regulatory_asymmetry_origin(banking_regulation).
regulatory_asymmetry_origin(consumer_protection).
regulatory_asymmetry_origin(legal_evidentiary_concerns).
regulatory_asymmetry_origin_disclaimer(
    'Not based on any exclusive claim that appraisers alone possess valuation
    ↪ expertise.').

%% =====
%% TIER 2 -- DESCRIPTIVE TENDENCIES (NOT the boundary)
%%
%% Overlapping populations; correlates of who typically lands on each
%% side. None of these determine licensure -- Tier 1 does.
%% =====

%% --- Axis: valuation approach (descriptive populations) ---
valuation_approach(appraisal).
valuation_approach(quant_financial).

scope_profession(appraiser).
scope_profession(quantitative_analyst).
scope_profession(financial_analyst).
scope_profession(investment_analyst).
scope_profession(portfolio_manager).
scope_profession(economist).
scope_profession(investment_strategist).
scope_profession(financial_forecaster).

scope_orientation(appraisal,      evidentiary).
scope_orientation(quant_financial, probabilistic).

%% --- Axis: temporal scope (its OWN viewpoint, distinct from approach) ---

```

```

temporal_scope(retrospective).
temporal_scope(current).
temporal_scope(prospective_point).
temporal_scope(future_range_horizon).

temporal_mode(point_in_time).
temporal_mode(distribution_over_horizon).

%% Typical (defeasible) pairing of approach to temporal mode -- a LINK
%% between two distinct axes, not an identity.
scope_temporal_focus(appraisal,      point_in_time).
scope_temporal_focus(quant_financial, distribution_over_horizon).

%% --- Axis: output form ---
scope_output_form(appraisal,      point_estimate).
scope_output_form(appraisal,      narrow_range).
scope_output_form(quant_financial, probability_distribution).
scope_output_form(quant_financial, range_of_outcomes).

%% --- Axis: skills / methodological toolkit (training, NOT intelligence) ---
scope_skill(mars_without_overfitting).
scope_skill(fit_vs_generalization_balance).    % balancing R2 against CV-R2
scope_skill(algorithmic_understanding).
scope_skill(handling_structured_concepts).
scope_skill(discerning_fine_distinctions_in_meaning).
scope_skill(assimilating_new_methods).

%% Skills typically demanded by advanced quantitative valuation; an
%% appraisal license neither requires nor certifies them.
scope_skill_demanded_by(quant_financial, mars_without_overfitting).
scope_skill_demanded_by(quant_financial, fit_vs_generalization_balance).
scope_skill_demanded_by(quant_financial, algorithmic_understanding).
scope_skill_demanded_by(quant_financial, handling_structured_concepts).
scope_skill_demanded_by(quant_financial, discerning_fine_distinctions_in_meaning).
scope_skill_demanded_by(quant_financial, assimilating_new_methods).

%% --- Inputs the appraiser weighs to support a value at the effective date ---
scope_inputs_considered(appraisal, current_market_evidence).
scope_inputs_considered(appraisal, observable_transactions).
scope_inputs_considered(appraisal, prevailing_market_conditions).
scope_inputs_considered(appraisal, recognized_methodologies).
scope_inputs_considered(appraisal, market_trends).
scope_inputs_considered(appraisal, anticipated_supply_demand).
scope_inputs_considered(appraisal, depreciation).
scope_inputs_considered(appraisal, capitalization_rates).
scope_inputs_considered(appraisal, absorption_patterns).
scope_inputs_considered(appraisal, expectations_already_priced_in).

%% --- Bounded future-oriented contexts that remain within appraisal ---
scope_future_oriented_context(income_property, future_income_streams).
scope_future_oriented_context(income_property, vacancy_rates).
scope_future_oriented_context(income_property, expenses).
scope_future_oriented_context(income_property, reversion_values).
scope_future_oriented_context(income_property, dcf_analysis).
scope_future_oriented_context(development,      prospective_stabilized_value).
scope_future_oriented_context(conservation,      future_damages).
scope_future_oriented_context(environmental,     future_damages).
scope_future_oriented_context(insurance,         future_damages).

```

```

scope_future_oriented_context(litigation,      future_damages).
scope_future_oriented_context(eminent_domain,  future_damages).
scope_future_oriented_context(general,         remaining_economic_life).
scope_future_oriented_context(general,         anticipated_market_reactions).
scope_future_oriented_context(market_analysis, price_trend_direction).

%% --- Professional constraints on any future-oriented component ---
scope_professional_constraint(supportability).
scope_professional_constraint(market_evidence).
scope_professional_constraint(extraordinary_assumptions_or_hypothetical_conditions ↪
↪ _disclosed).
scope_professional_constraint(avoidance_of_unsupported_speculation).

%% --- Contrasts across the descriptive axes:
↪ scope_contrast(A,B,Dimension,Distinction) ---
scope_contrast(appraisal, quant_financial, orientation,
                evidentiary_vs_probabilistic).
scope_contrast(appraisal, quant_financial, temporal_mode,
                point_in_time_vs_distribution_over_horizon).
scope_contrast(appraisal, quant_financial, output_form,
                point_or_narrow_range_vs_probability_distribution).
scope_contrast(appraisal, quant_financial, purpose,
                market_value_vs_investment_worth).
scope_contrast(appraisal, quant_financial, skill_set,
                appraisal_licensing_toolkit_vs_advanced_statistical_toolkit).
scope_contrast(appraisal, quant_financial, typical_methods,
                sales_comparison_cost_income_vs_simulation_option_pricing_var).

%% --- The orienting question each population asks (verbatim from the passage) ---
scope_question(appraisal,
                'What is the most supportable market value indication as of the specified
                ↪ effective date, given current evidence and defined assumptions?').
scope_question(quant_financial,
                'What range of outcomes is likely over a future period, with what
                ↪ probabilities, and how should resources be allocated?').

%% --- Where the two converge (appraisal borrows distributional methods) ---
scope_convergence_area(complex_commercial_valuation).
scope_convergence_area(mass_appraisal).
scope_convergence_area(automated_valuation_models).
scope_convergence_area(institutional_portfolio_analysis).

```

## 5 Market Value × LTV as a Proxy for Foreclosure Value

The preceding sections set out what appraisal is for, what it produces, and where its doctrinal and regulatory boundaries lie; this section reaches the tension they have been building toward.

One of the deeper practical realities underlying mortgage lending and collateral analysis is that, in actual economic function, a market value appraisal used in mortgage underwriting often serves as an implicit proxy for expected collateral recoverability over some future holding period, even though formally the appraisal is framed as a present-market-value opinion as of a specific effective date. (Foster and Van Order 1984; Kau et al. 1995; Deng, Quigley, and Van Order 2000)

An 80% loan-to-value ratio is not arbitrary. Historically, it evolved partly as a risk buffer intended to protect lenders against:

- moderate market declines,
- foreclosure expenses,

- liquidation discounts,
- carrying costs,
- legal expenses,
- accrued interest,
- and uncertainty during the time required to dispose of collateral after default.(Foote, Gerardi, and Willen 2008; Qi and Yang 2009)

The buffer is also why the loan amount itself—the product of market value and the permitted loan-to-value ratio—can be read as the lender’s working estimate of recoverable foreclosure value: capping the loan at a fraction of market value is, in effect, a bet that forced-sale proceeds will not fall below the loan balance. How large the haircut *should* be, however, is far harder to pin down than the round 80% figure suggests. Estimates of the foreclosure discount vary widely across markets, methods, and periods; one frequently cited study of forced sales places it near 27% below comparable arm’s-length prices(Campbell, Giglio, and Pathak 2011), while other analyses—correcting for property condition, location, and selection—report materially different magnitudes.(Pennington-Cross 2006; Clauretie and Daneshvary 2009)

The conventional 80% threshold, by contrast, is essentially a single nationwide parameter, applied across enormously heterogeneous local markets and successive phases of the credit cycle. It cannot track any particular market’s discount, and so functions as a very rough, one-size-fits-all approximation rather than a precise mapping from market value to foreclosure recovery. If anything it is a thin one: a 20% haircut sits *below* the 27% discount that study reports, so the maximum conventional loan can exceed gross forced-sale proceeds even before the disposition costs enumerated above are subtracted.

Where borrower equity is too thin to constitute the buffer, mortgage insurance is substituted for it. This is best understood not as a policy dial but as a risk-allocation tradeoff: a segment of borrowers is asset-light yet income-stable, and the incremental default and loss risk of lending to them above 80% LTV is forecast and borne by an insurer—privately, through private mortgage insurance on a high-LTV conventional loan, or publicly, through the Federal Housing Administration. The market in which that tradeoff is routine, however, is a creation of federal housing policy: the government-sponsored enterprises that dominate conventional lending are barred by charter from purchasing loans above 80% LTV without such credit enhancement(United States Code 1970). The terms on which the insurance ends are likewise institutional choices rather than risk computations, and they differ sharply across programs.<sup>1</sup>

So although the appraiser is officially estimating “current market value,” the lender is economically interpreting that value through a forward-looking risk lens. In effect, the lender is asking something closer to:

“If this borrower defaults sometime in the foreseeable future, is the collateral likely to remain sufficient to cover the outstanding debt and foreclosure-related losses?”

That is inherently a future-oriented question.

This rests on a deliberate conceptual separation between:

- the appraiser’s “current market value” opinion, and
- the lender’s practical use of that opinion as a forecast buffer against future collateral impairment.

Historically and legally, regulators attempted to keep appraisers from becoming speculative forecasters because:

<sup>1</sup>Conventional borrower-paid private mortgage insurance terminates automatically at 78% LTV and may be requested at 80% under the Homeowners Protection Act of 1998.(United States Congress 1998) Federal Housing Administration premiums are stricter: for loans originated on or after June 3, 2013 with original loan-to-value above 90% (e.g., 3.5% down), the annual premium runs for the life of the loan, while loans at or below 90% cancel after eleven years.(U.S. Department of Housing and Urban Development 2013) Many FHA borrowers therefore shed mortgage insurance only by refinancing into a conventional loan near 80% LTV.

- speculative forecasting contributed to past lending crises,
- courts wanted appraisals tied to observable evidence,
- and banking regulators sought standardized, defensible valuation practices.

Thus appraisal doctrine evolved toward:

- present value,
- current market evidence,
- defined effective dates,
- and constrained assumptions.

Meanwhile, lenders, investors, and collateral analysts routinely use those same appraisals in probabilistic future-risk frameworks.

In reality, mortgage underwriting is an integrated system involving:

1. current market valuation,
2. future risk estimation,
3. default probability,
4. foreclosure timing,
5. liquidation severity,
6. macroeconomic expectations,
7. and portfolio stress analysis.

The appraiser formally contributes primarily to item (1), while the lender internally combines that result with the others.

The distinction becomes blurry in practice. If a lender would not make a loan unless it believed the collateral would likely preserve sufficient value into the future, then the appraisal indirectly functions as part of a future-value protection model, even if appraisal standards formally avoid describing it that way.

This tension has become even more pronounced with:

- AVMs,
- institutional portfolio analytics,
- securitization models,
- machine learning valuation systems,
- Basel capital modeling,
- and large-scale mortgage stress testing.

Modern collateral analytics increasingly treat property values probabilistically over time rather than as single-point static estimates. (Basel Committee on Banking Supervision 2017; Steurer, Hill, and Pfeifer 2021; Board of Governors of the Federal Reserve System 2021)

One could argue that traditional appraisal regulation reflects a historical compromise:

- appraisers provide disciplined, evidence-based present value anchors,
- while financial institutions assume responsibility for future-risk interpretation.

Whether that separation remains fully coherent in modern financial systems is a legitimate and increasingly debated question.

A practical implication follows for appraisal practice itself. Most appraisers are not aware of this collateral-valuation role—the lender’s use of market value as a forward-looking buffer is rarely visible from within the appraisal assignment—and so do not treat their market value conclusions with the caution toward future value trends that the role warrants. Recognizing how the present-value opinion is actually consumed is a first step toward closing that gap, and an area this account is meant to open for further work.

```
%% appraisal_proxy.pl
%%
```

```

%% A Prolog knowledge base encoding the thesis that market value
%% appraisals, while formally framed as present-value opinions, function
%% in practice as implicit proxies for future collateral recoverability
%% within mortgage underwriting and lender risk management.

:- module(appraisal_proxy,
  [ % --- central thesis ---
    proxy_thesis/1

    % --- LTV-as-buffer mechanism ---
    , proxy_ltv_buffer_against/1
    , proxy_historical_ltv_threshold/1

    % --- formal vs practical interpretation ---
    , proxy_formal_framing/1
    , proxy_practical_interpretation/1
    , proxy_underlying_question/1

    % --- the artificial separation ---
    , proxy_separation_party/2          % Party, Role
    , proxy_doctrine_practice_gap/2    % FormalSide, PracticalSide

    % --- historical compromise ---
    , proxy_historical_reason/1
    , proxy_doctrine_emphasis/1
    , proxy_routine_practical_use/1

    % --- modern intensifiers ---
    , proxy_modern_driver/1
    , proxy_modern_treatment/1

    % --- responsibility allocation ---
    , proxy_appraiser_responsibility/1
    , proxy_lender_responsibility/1

    % --- meta-claim about coherence ---
    , proxy_open_question/1

    % --- practical implication for appraisal practice ---
    , proxy_practical_implication/1
    , proxy_appraiser_blindspot/1
    , proxy_recommended_practice/1
    , proxy_further_work/1

    % --- underwriting system view ---
    , underwriting_component/2          % Number, Component
    , underwriting_responsibility/2     % Component, Party
  ]).

:- discontiguous proxy_thesis/1.
:- discontiguous underwriting_component/2.

%% =====
%% PART 1: THE CENTRAL THESIS
%% =====

proxy_thesis(

```

```

    'A market value appraisal used in mortgage underwriting often serves as an
    ↪ implicit proxy for expected collateral recoverability over a future holding
    ↪ period.').
proxy_thesis(
    'The appraisal is formally framed as a present-market-value opinion as of a
    ↪ specific effective date.').
proxy_thesis(
    'The lender economically interprets the appraisal through a forward-looking
    ↪ risk lens.').
proxy_thesis(
    'There is a deliberate conceptual separation between the appraiser's
    ↪ current-market-value opinion and the lender's use of it as a forecast
    ↪ buffer.').
proxy_thesis(
    'The appraisal indirectly functions as part of a future-value protection model,
    ↪ even though appraisal standards formally avoid describing it that way.').
proxy_thesis(
    'Whether the separation between appraisal doctrine and lender practice remains
    ↪ fully coherent in modern financial systems is an open and increasingly
    ↪ debated question.').

%% =====
%% PART 2: THE LTV-AS-BUFFER MECHANISM
%% =====

proxy_historical_ltv_threshold(eighty_percent).

proxy_ltv_buffer_against(moderate_market_declines).
proxy_ltv_buffer_against(foreclosure_expenses).
proxy_ltv_buffer_against(liquidation_discounts).
proxy_ltv_buffer_against(carrying_costs).
proxy_ltv_buffer_against(legal_expenses).
proxy_ltv_buffer_against(accrued_interest).
proxy_ltv_buffer_against(uncertainty_during_disposition).

%% =====
%% PART 3: FORMAL FRAMING VS. PRACTICAL INTERPRETATION
%% =====

proxy_formal_framing(present_market_value_opinion).
proxy_formal_framing(opinion_as_of_specific_effective_date).
proxy_formal_framing(grounded_in_current_market_evidence).

proxy_practical_interpretation(forward_looking_risk_lens).
proxy_practical_interpretation(forecast_buffer_against_future_impairment).
proxy_practical_interpretation(input_to_future_collateral_recoverability_estimate).

proxy_underlying_question(
    'If this borrower defaults sometime in the foreseeable future, is the
    ↪ collateral likely to remain sufficient to cover the outstanding debt and
    ↪ foreclosure-related losses?').

%% =====
%% PART 4: THE ARTIFICIAL SEPARATION
%% =====

proxy_separation_party(appraiser, opinion_of_current_market_value).
proxy_separation_party(lender, forecast_buffer_against_future_impairment).

```

```

proxy_separation_party(appraiser, evidence_based_present_value_anchor).
proxy_separation_party(lender, future_risk_interpretation).

proxy_doctrine_practice_gap(present_market_value_opinion,
                             implicit_future_value_proxy).
proxy_doctrine_practice_gap(opinion_as_of_effective_date,
                             forecast_over_foreseeable_holding_period).
proxy_doctrine_practice_gap(evidence_based_present_anchor,
                             future_risk_buffer).
proxy_doctrine_practice_gap(static_single_point_estimate,
                             probabilistic_value_over_time).

%% =====
%% PART 5: HISTORICAL COMPROMISE
%% =====

proxy_historical_reason(speculative_forecasting_contributed_to_past_crises).
proxy_historical_reason(courts_required_observable_evidence).
proxy_historical_reason(regulators_sought_standardized_defensible_practices).

proxy_doctrine_emphasis(present_value).
proxy_doctrine_emphasis(current_market_evidence).
proxy_doctrine_emphasis(defined_effective_dates).
proxy_doctrine_emphasis(constrained_assumptions).

proxy_routine_practical_use(lenders_use_in_probabilistic_future_risk_frameworks).
proxy_routine_practical_use(investors_use_in_probabilistic_future_risk_frameworks).
proxy_routine_practical_use(collateral_analysts_use_in_probabilistic_future_risk_f
↪ rameworks).

%% =====
%% PART 6: MODERN INTENSIFIERS
%% =====

proxy_modern_driver(avms).
proxy_modern_driver(institutional_portfolio_analytics).
proxy_modern_driver(securitization_models).
proxy_modern_driver(machine_learning_valuation_systems).
proxy_modern_driver(basel_capital_modeling).
proxy_modern_driver(large_scale_mortgage_stress_testing).

proxy_modern_treatment(probabilistic_over_time).
proxy_modern_treatment(distributional_rather_than_point_estimate).
proxy_modern_treatment(dynamic_rather_than_static).

%% =====
%% PART 7: ALLOCATION OF RESPONSIBILITY UNDER THE COMPROMISE
%% =====

proxy_appraiser_responsibility(disciplined_present_value_anchor).
proxy_appraiser_responsibility(evidence_based_valuation).
proxy_appraiser_responsibility(current_market_value_opinion).

proxy_lender_responsibility(future_risk_interpretation).
proxy_lender_responsibility(default_probability_estimation).
proxy_lender_responsibility(foreclosure_timing_estimation).
proxy_lender_responsibility(liquidation_severity_estimation).
proxy_lender_responsibility(portfolio_stress_analysis).

```

```

proxy_lender_responsibility(macroeconomic_expectation_integration).

%% =====
%% PART 8: OPEN QUESTION (META-CLAIM)
%% =====

proxy_open_question(
    'Whether the separation between disciplined present-value appraisal and
    ↪ lender-side future-risk interpretation remains fully coherent in modern
    ↪ financial systems.').
proxy_open_question(
    'Whether modern probabilistic-over-time collateral analytics make the
    ↪ formal/practical separation untenable in some sectors.').

%% =====
%% PART 9: THE INTEGRATED MORTGAGE UNDERWRITING SYSTEM
%% =====

underwriting_component(1, current_market_valuation).
underwriting_component(2, future_risk_estimation).
underwriting_component(3, default_probability).
underwriting_component(4, foreclosure_timing).
underwriting_component(5, liquidation_severity).
underwriting_component(6, macroeconomic_expectations).
underwriting_component(7, portfolio_stress_analysis).

underwriting_responsibility(current_market_valuation, appraiser).
underwriting_responsibility(future_risk_estimation, lender).
underwriting_responsibility(default_probability, lender).
underwriting_responsibility(foreclosure_timing, lender).
underwriting_responsibility(liquidation_severity, lender).
underwriting_responsibility(macroeconomic_expectations, lender).
underwriting_responsibility(portfolio_stress_analysis, lender).

%% =====
%% PART 10: PRACTICAL IMPLICATION FOR APPRAISAL PRACTICE
%% =====

proxy_practical_implication(
    'Most appraisers are not aware that market value functions, via the
    ↪ loan-to-value ratio, as a proxy for future foreclosure value.').
proxy_practical_implication(
    'The lender''s forward-looking use of market value is rarely visible from
    ↪ within the appraisal assignment.').
proxy_practical_implication(
    'Appraisers therefore do not treat market value conclusions with the caution
    ↪ toward future value trends that the collateral-valuation role warrants.').

proxy_appraiser_blindspot(collateral_valuation_role).
proxy_appraiser_blindspot(forward_looking_use_of_market_value).
proxy_appraiser_blindspot(market_value_as_ltv_proxy_for_foreclosure_value).

proxy_recommended_practice(recognize_how_present_value_opinion_is_consumed).
proxy_recommended_practice(treat_market_value_with_caution_re_future_trends).

proxy_further_work(closing_the_doctrine_practice_gap).
proxy_further_work(equipping_appraisers_for_the_collateral_valuation_perspective).

```

## 6 Conclusion

The five passages encoded in the preceding knowledge bases describe a single profession from five complementary vantage points. Taken in isolation, each passage reads as a self-contained account: the social and public-interest stakes they carry, the economic functions appraisals support, their operational role in collateral risk analysis, the regulatory boundary separating them from adjacent quantitative work, and the tension between their formal framing and their practical use. Taken together, and with the structure made explicit in Prolog, they describe something more interesting — a profession whose official doctrine and economic function have been deliberately held apart, and whose internal coherence is increasingly tested by the analytic tools now applied to its outputs.

The first three knowledge bases share a common schema: `effect(Quality, Horizon, Domain, Effect)`. Each frames appraisal quality as a causal input whose consequences propagate across short-, medium-, and long-term horizons, differing only in the domain through which those consequences are read. The economic lens reads them through credit markets, investment allocation, taxation, and capital formation; the social lens reads them through equity, public confidence, and community outcomes; the collateral lens reads them through underwriting, portfolio management, and financial system stability. The shared schema is not a coincidence. It reflects a substantive claim that runs through all three passages: the quality of valuation work has consequences, those consequences scale with time, and the same underlying failure modes manifest differently depending on whose interests are foregrounded. Shared atoms on the argument side of the predicates — the quality and horizon vocabularies in particular — make the three vocabularies joinable, so that a single failure such as systematic overvaluation can be traced simultaneously through economic mispricing, social inequity, and collateral mismeasurement.

The fourth knowledge base shifts register. It does not describe consequences of appraisal quality; it describes what appraisal *is*, methodologically and legally. Its predicates are taxonomic rather than causal, and its inferential machinery answers questions about jurisdiction rather than impact. This shift is necessary because the consequential framing of the first three passages presupposes a definition of the activity being evaluated, and that definition is itself contested. The scope predicates record what appraisers do — evidence-based present-value opinions, with narrowly bounded future-oriented components — while the regulation predicates record the conditions under which that activity falls within licensed practice. The forward-chaining rules (`requires_license/1`, `exempt_from_appraisal_license/1`, `regulated_appraisal/1`) provide a mechanism for asking whether a given piece of valuation work falls inside or outside the regulated perimeter, a question that the first three knowledge bases assume has already been answered.

The fifth knowledge base advances a thesis. Where the others describe, classify, or trace consequences, the proxy knowledge base argues that the careful doctrinal separation between present-value appraisal and future-value forecasting — the separation the regulation knowledge base documents — is in tension with the actual economic function of appraisals in mortgage lending. The 80% loan-to-value convention, the seven specific risks it buffers against, the underlying lender question recorded verbatim (“If this borrower defaults sometime in the foreseeable future, is the collateral likely to remain sufficient . . .”), and the integrated seven-component underwriting system all support the same claim: the appraisal is formally a present-value opinion and economically a future-value buffer. The historical compromise that produced this arrangement — appraisers as disciplined present-value anchors, lenders as the locus of future-risk interpretation — was a defensible response to the conditions under which appraisal licensing emerged, but the modern intensifiers enumerated in the passage (AVMs, machine learning valuation systems, Basel capital modeling, securitization analytics, large-scale stress testing) treat property values probabilistically over time. Those tools draw doctrinally separated layers back into a single computational pipeline, and the conceptual separation between layers

becomes harder to sustain as a result.

Whether that reconciliation is achievable, and on what terms, is the question with which the proxy passage closes. The knowledge bases do not answer it. They do, however, provide an explicit, queryable account of the considerations relevant to answering it: the consequences of valuation quality across horizons and domains, the jurisdictional boundary that determines which consequences fall within regulated practice, and the structural reasons that boundary is now under pressure. That account is the foundation on which more specific arguments — about methodology, about standards reform, about the relationship between appraisal practice and the broader system of quantitative valuation — can be built.

## Notes on the knowledge representation

Three structural patterns are worth recording explicitly, since they emerged from the encoding process rather than from any single passage.

*First, the family of knowledge bases is not flat.* Three of the five share a schema; one provides the jurisdictional qualifications that determine when that schema applies; one advances a thesis about the gap between formal framing and economic function. A bridge layer combining all five does not treat them as five columns of the same matrix. It treats the first three as describing consequences, the fourth as constraining where those descriptions are operative, and the fifth as identifying where the constraint itself is under pressure. This asymmetric structure is a feature rather than a defect: it mirrors the actual organization of professional practice, in which descriptive claims, jurisdictional claims, and critical claims occupy genuinely different epistemic roles.

*Second, the prefix discipline that distinguishes the modules (*economic\_*, *social\_*, *collateral\_*, *scope\_/regulation\_*, *proxy\_/underwriting\_*) does real work only where there is collision risk.* Predicates with direct counterparts across modules earn their prefix; predicates unique to a single conceptual domain do not. Shared value vocabulary on the argument side — the quality, horizon, and domain atoms — is precisely what allows the modules to communicate, and prefixing those values would defeat that. The pattern that emerged is general: prefix on the predicates, shared atoms on the arguments, with selective prefixing rather than uniform prefixing where the predicate is unique to one lens.

*Third, the central tension identified in the proxy knowledge base is closely related to, but distinct from, the regulatory asymmetry identified in the scope/regulation knowledge base.* The two are recorded through parallel two-place predicates — **regulatory\_asymmetry/2** and **proxy\_doctrine\_practice\_gap/2** — with deliberately different names because they describe different axes of the same underlying arrangement. The regulatory asymmetry concerns who is regulated and how heavily; the doctrine-practice gap concerns what is formally said about the appraiser's product versus what is economically done with it. The two tensions reinforce each other. Regulation pushes appraisers toward present-value doctrine; economic function pulls their output toward future-value use. The combined picture is one of a profession whose official self-description is increasingly difficult to reconcile with the analytic pipelines its products feed.

## References

- Agarwal, Sumit, Itzhak Ben-David, and Vincent Yao (2015). "Collateral Valuation and Borrower Financial Constraints: Evidence from the Residential Real Estate Market". In: *Management Science* 61.9, pp. 2220–2240. DOI: [10.1287/mnsc.2014.2002](https://doi.org/10.1287/mnsc.2014.2002).
- Alexandrov, Alexei, Laurie Goodman, and Michael Neal (Jan. 2023). *Reengineering the Appraisal Process: Better Leveraging Both Automated Valuation Models and Manual Appraisals*. Tech. rep. Urban Institute. URL: <https://www.urban.org/sites/default/files/2023-01/Reengineering%20the%20Appraisal%20Process.pdf>.

- Basel Committee on Banking Supervision (2017). *Basel III: Finalising Post-Crisis Reforms*. Standards d424. Bank for International Settlements. URL: <https://www.bis.org/bcbs/publ/d424.pdf>.
- Board of Governors of the Federal Reserve System (Mar. 2021). *Dodd-Frank Act Stress Test 2021: Supervisory Stress Test Methodology*. Tech. rep. Board of Governors of the Federal Reserve System. URL: <https://www.federalreserve.gov/publications/files/2021-april-supervisory-stress-test-methodology.pdf>.
- Campbell, John Y., Stefano Giglio, and Parag Pathak (2011). “Forced Sales and House Prices”. In: *American Economic Review* 101.5, pp. 2108–2131. DOI: [10.1257/aer.101.5.2108](https://doi.org/10.1257/aer.101.5.2108).
- Clauretie, Terrence M. and Nasser Daneshvary (2009). “Estimating the House Foreclosure Discount Corrected for Spatial Price Interdependence and Endogeneity of Marketing Time”. In: *Real Estate Economics* 37.1, pp. 43–67. DOI: [10.1111/j.1540-6229.2009.00235.x](https://doi.org/10.1111/j.1540-6229.2009.00235.x).
- Crosby, Neil, Colin M. Lizieri, and Patrick M. McAllister (2010). “Means, Motive and Opportunity? Disentangling Client Influence on Performance Measurement Appraisals”. In: *Journal of Property Research* 27.2, pp. 181–201. DOI: [10.1080/09599916.2010.499014](https://doi.org/10.1080/09599916.2010.499014).
- Deng, Yongheng, John M. Quigley, and Robert Van Order (2000). “Mortgage Terminations, Heterogeneity and the Exercise of Mortgage Options”. In: *Econometrica* 68.2, pp. 275–307. DOI: [10.1111/1468-0262.00110](https://doi.org/10.1111/1468-0262.00110).
- Diaz Julian, III and Marvin L. Wolverton (1998). “A Longitudinal Examination of the Appraisal Smoothing Hypothesis”. In: *Real Estate Economics* 26.2, pp. 349–358. DOI: [10.1111/1540-6229.00749](https://doi.org/10.1111/1540-6229.00749).
- Ebrahim, M. Shahid and Sikandar Hussain (2010). “Financial Development and Asset Valuation: The Special Case of Real Estate”. In: *Journal of Banking & Finance* 34.1, pp. 150–162. DOI: [10.1016/j.jbankfin.2009.07.011](https://doi.org/10.1016/j.jbankfin.2009.07.011).
- Eriksen, Michael D. et al. (2019). “The Influence of Contract Prices and Relationships on Appraisal Bias”. In: *Journal of Urban Economics* 111, pp. 132–143. DOI: [10.1016/j.jue.2019.03.001](https://doi.org/10.1016/j.jue.2019.03.001).
- Foote, Christopher L., Kristopher Gerardi, and Paul S. Willen (2008). “Negative Equity and Foreclosure: Theory and Evidence”. In: *Journal of Urban Economics* 64.2, pp. 234–245. DOI: [10.1016/j.jue.2008.07.006](https://doi.org/10.1016/j.jue.2008.07.006).
- Foster, Chester and Robert Van Order (1984). “An Option-Based Model of Mortgage Default”. In: *Housing Finance Review* 3.4, pp. 351–372.
- Friedman, Jerome H. (1991). “Multivariate Adaptive Regression Splines”. In: *The Annals of Statistics* 19.1, pp. 1–67. DOI: [10.1214/aos/1176347963](https://doi.org/10.1214/aos/1176347963).
- Gallimore, Paul (1996). “Confirmation Bias in the Valuation Process: A Test for Corroborating Evidence”. In: *Journal of Property Research* 13.4, pp. 261–273. DOI: [10.1080/095999196368781](https://doi.org/10.1080/095999196368781).
- Geltner, David M. (1991). “Smoothing in Appraisal-Based Returns”. In: *The Journal of Real Estate Finance and Economics* 4.3, pp. 327–345. DOI: [10.1007/BF00161933](https://doi.org/10.1007/BF00161933).
- Ghent, Andra C., Walter N. Torous, and Rossen I. Valkanov (2019). “Commercial Real Estate as an Asset Class”. In: *Annual Review of Financial Economics* 11, pp. 153–171. DOI: [10.1146/annurev-financial-110118-123121](https://doi.org/10.1146/annurev-financial-110118-123121).
- Griffin, John M. and Gonzalo Maturana (2016). “Who Facilitated Misreporting in Securitized Loans?” In: *The Review of Financial Studies* 29.2. Evidence of appraisal inflation and its consequences, pp. 384–419. DOI: [10.1093/rfs/hhv130](https://doi.org/10.1093/rfs/hhv130).
- Hendershott, Patric H. and Edward J. Kane (July 1992). *Office Market Values During the Past Decade: How Distorted Have Appraisals Been?* NBER Working Paper 4128. Available at SSRN: <https://ssrn.com/abstract=226786>. National Bureau of Economic Research. DOI: [10.3386/w4128](https://doi.org/10.3386/w4128). URL: <https://www.nber.org/papers/w4128>.
- (1995). “U.S. Office Market Values During the Past Decade: How Distorted Have Appraisals Been?” In: *Real Estate Economics* 23.2, pp. 101–116. DOI: [10.1111/1540-6229.00660](https://doi.org/10.1111/1540-6229.00660).

- Hull, John C. (2018). *Options, Futures, and Other Derivatives*. 10th ed. New York: Pearson. ISBN: 978-0-13-447208-9.
- International Association of Assessing Officers (2017). *Standard on Mass Appraisal of Real Property*. International Association of Assessing Officers, Kansas City, MO. URL: [https://www.iaao.org/wp-content/uploads/Standard\\_on\\_Mass\\_Appraisal.pdf](https://www.iaao.org/wp-content/uploads/Standard_on_Mass_Appraisal.pdf).
- International Valuation Standards Council (2025). *International Valuation Standards (IVS)*. International Valuation Standards Council, London. Published 31 January 2024; effective 31 January 2025. URL: <https://www.ivsc.org/>.
- K'Akumu, Owiti A. and James E. Larsen (2024). "A Primer on Regulations and the Practice of Residential Property Appraisal". In: *Journal of Real Estate Practice and Education*. Focus on regulatory and public interest aspects. DOI: [10.1080/15214842.2024.2377869](https://doi.org/10.1080/15214842.2024.2377869).
- Kau, James B. et al. (1995). "The Valuation at Origination of Fixed-Rate Mortgages with Default and Prepayment". In: *The Journal of Real Estate Finance and Economics* 11.1, pp. 5–36. DOI: [10.1007/BF01097934](https://doi.org/10.1007/BF01097934).
- LaCour-Little, Michael and Stephen Malpezzi (2003). "Appraisal Quality and Residential Mortgage Default: Evidence from Alaska". In: *The Journal of Real Estate Finance and Economics* 27.2, pp. 211–233. DOI: [10.1023/A:1024728420837](https://doi.org/10.1023/A:1024728420837).
- Lee, Ming-Hsiang, Chien-Wen Peng, and Hui-Fen Liao (2020). "An Analysis of Objectivity in the Real Estate Appraisal Process". In: *International Real Estate Review* 23.4, pp. 483–504.
- Mayer, Yanling G. and Frank E. Nothaft (2022). "Appraisal Overvaluation: Evidence of Price Adjustment Bias in Sales Comparisons". In: *Real Estate Economics* 50.3, pp. 862–881. DOI: [10.1111/1540-6229.12351](https://doi.org/10.1111/1540-6229.12351).
- Murphy, Edward V. (2012). *Regulation of Real Estate Appraisers*. CRS Report RS22953. Congressional Research Service. URL: <https://sgp.fas.org/crs/misc/RS22953.pdf>.
- Office of the Comptroller of the Currency and Board of Governors of the Federal Reserve System and Federal Deposit Insurance Corporation and National Credit Union Administration and Consumer Financial Protection Bureau and Federal Housing Finance Agency (2024). *Quality Control Standards for Automated Valuation Models*. Final Rule, 89 Fed. Reg. 64538 (Aug. 7, 2024). URL: <https://www.federalregister.gov/documents/2024/08/07/2024-16197/quality-control-standards-for-automated-valuation-models>.
- Pennington-Cross, Anthony (2006). "The Value of Foreclosed Property". In: *Journal of Real Estate Research* 28.2, pp. 193–214. DOI: [10.1080/10835547.2006.12091177](https://doi.org/10.1080/10835547.2006.12091177).
- Qi, Min and Xiaolong Yang (2009). "Loss Given Default of High Loan-to-Value Residential Mortgages". In: *Journal of Banking & Finance* 33.5, pp. 788–799. DOI: [10.1016/j.jbankfin.2008.09.010](https://doi.org/10.1016/j.jbankfin.2008.09.010).
- Quan, Daniel C. and John M. Quigley (1991). "Price Formation and the Appraisal Function in Real Estate Markets". In: *The Journal of Real Estate Finance and Economics* 4.2, pp. 127–146. DOI: [10.1007/BF00173120](https://doi.org/10.1007/BF00173120).
- Steurer, Miriam, Robert J. Hill, and Norbert Pfeifer (2021). "Metrics for Evaluating the Performance of Machine Learning Based Automated Valuation Models". In: *Journal of Property Research* 38.2, pp. 99–129. DOI: [10.1080/09599916.2020.1858937](https://doi.org/10.1080/09599916.2020.1858937).
- The Appraisal Foundation (2024). *Uniform Standards of Professional Appraisal Practice (USPAP), 2024 Edition*. The Appraisal Foundation, Appraisal Standards Board, Washington, DC. Effective January 1, 2024. URL: <https://www.appraisalfoundation.org/>.
- U.S. Department of Housing and Urban Development (2013). *Revision of Federal Housing Administration Policies Concerning Cancellation of the Annual Mortgage Insurance Premium*. Mortgagee Letter 2013-04, Federal Housing Administration. Effective June 3, 2013; the annual MIP continues for the life of loans with original LTV above 90%, and for 11 years at or below 90%. URL: <https://www.hud.gov/sites/documents/13-04ml.pdf>.
- United States Code (1970). *Federal National Mortgage Association and Federal Home Loan Mortgage Corporation Charter Acts*. 12 U.S.C. § 1717(b)(2) (Fannie Mae) and 12 U.S.C. §

1454(a)(2) (Freddie Mac). Prohibit GSE purchase of conventional mortgages exceeding 80% LTV absent credit enhancement (private mortgage insurance, seller recourse/repurchase, or a retained seller participation).

United States Congress (1989). *Financial Institutions Reform, Recovery, and Enforcement Act of 1989*. Pub. L. No. 101-73, 103 Stat. 183; Title XI codified at 12 U.S.C. §§ 3331–3356. URL: <https://www.govinfo.gov/content/pkg/STATUTE-103/pdf/STATUTE-103-Pg183.pdf>.

— (1998). *Homeowners Protection Act of 1998*. Pub. L. No. 105-216, 112 Stat. 897; codified at 12 U.S.C. §§ 4901–4910. Automatic termination of borrower-paid private mortgage insurance at 78% LTV; cancellation may be requested at 80%.

# Residual Constraint Approach (RCA): Framework & Protocol

*An updated version of the paper originally published on Zenodo, March 5, 2025.*

William Bert Craytor, SRA

Vol. 1, Issue 1 · Summer 2026

Published July 1, 2026

DOI: [10.67330/kgebx46](https://doi.org/10.67330/kgebx46)

© 2026 William Bert Craytor. Published by Pacific Vista Net.

This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

**Abstract.** A significant limitation of conventional appraisal methods is their reliance on outdated manual techniques, such as matched pair analyses, which bypass property feature value contributions as if they didn't exist: the traditional sales grid has no column for value contributions, only adjustments. In fact, value contributions provide the necessary basis for mathematical constraints that prevent over and undervaluation by traditional appraisers.

The Residual Constraint Approach (RCA) advances the traditional Sales Comparison Approach (SCA) by integrating a multi-phase valuation process that employs Multivariate Adaptive Regression Splines (MARS) alongside rigorous mathematical constraints on feature value contributions, in particular unmeasured features — latent variables such as aesthetics, design, condition and quality — whose values are typically subject to subjective judgment and bias by the traditional appraiser.

The mathematical constraints in RCA rely on the expert application of MARS regression to estimate the value contributions of measurable property features to the sale price. However, this estimate typically accounts for only about 80% of the actual sale price, excluding subjectively assessed features like design, functional utility, condition, and quality. In the San Francisco Bay Area, the residual—representing these subjective components—averages around 20% of the total property value. The 80/20 figure reflects the author's experience in the San Francisco Bay Area under typical data conditions and should not be read as a market-invariant constant. In markets where the achievable  $R^2$  is lower, the framework's response is to identify and measure unmeasured value drivers — distance to the ocean in coastal communities, elevation in hillside neighborhoods, and similar variables collected from external sources when MLS data does not supply them — rather than to accept the limitation and report wider error bands. While MARS residuals are often treated as estimation errors in other contexts, in real estate valuation they serve as an indirect measure of latent variable value. Though this might initially seem impractical, proof demonstrates that the residual can be meaningfully decomposed into descriptive components, provided their contributions sum to the residual—without affecting the final valuation outcome.

This methodology emerges from two decades of empirical application by the author, developed through iterative MARS implementations in R and Python environments.

**Keywords:** real estate valuation, residential appraisal, Residual Constraint Approach, MARS, multivariate adaptive regression splines, sales comparison approach, residual analysis, latent variables, automated valuation

## Contents

### [1 Introduction](#)

|          |                                                  |           |
|----------|--------------------------------------------------|-----------|
| <b>2</b> | <b>Background</b>                                | <b>6</b>  |
| <b>3</b> | <b>Valuation Engineer</b>                        | <b>8</b>  |
| <b>4</b> | <b>Assumptions</b>                               | <b>9</b>  |
| <b>5</b> | <b>RCA Workflow</b>                              | <b>11</b> |
| 5.1      | Types of MARS Analysis . . . . .                 | 11        |
| 5.2      | Processing Workflow . . . . .                    | 11        |
| 5.3      | Data Workflow . . . . .                          | 12        |
| 5.3.1    | Data Requirements for MARS . . . . .             | 13        |
| 5.3.2    | Protocols: Closed Sales Listings . . . . .       | 14        |
| 5.4      | Project Folder Structure . . . . .               | 16        |
| 5.4.1    | Protocols: Multiple MARS (earth) Runs . . . . .  | 17        |
| 5.5      | The Core Excel Workbook . . . . .                | 19        |
| 5.6      | Backups . . . . .                                | 19        |
| <b>6</b> | <b>Workbook Notation</b>                         | <b>20</b> |
| 6.1      | Descriptive Dimensions . . . . .                 | 20        |
| 6.1.1    | The Project Dimension (“a”) . . . . .            | 21        |
| 6.1.2    | Run Version Dimension (“r”) . . . . .            | 22        |
| 6.1.3    | Workflow Stages (“w”) . . . . .                  | 22        |
| 6.1.4    | Excel Workbook Sheets (“s”) . . . . .            | 22        |
| 6.1.5    | Property Listings (“p”) - MLSData rows . . . . . | 23        |
| 6.1.6    | MLS Variables (“v”) MLSData Columns . . . . .    | 23        |
| 6.1.7    | Final Notation Example . . . . .                 | 24        |
| 6.1.8    | Justification . . . . .                          | 24        |
| 6.1.9    | Matrices? . . . . .                              | 25        |
| 6.1.10   | First Simplification . . . . .                   | 25        |
| 6.1.11   | Second Simplification . . . . .                  | 25        |
| <b>7</b> | <b>The RCA Sales Grid</b>                        | <b>27</b> |
| <b>8</b> | <b>Variables</b>                                 | <b>30</b> |
| 8.1      | Variable Origin Ontology . . . . .               | 30        |
| 8.2      | Variable Application Ontology . . . . .          | 31        |
| 8.3      | Second- and Third-Degree Terms . . . . .         | 32        |
| 8.3.1    | Adjustment Variables . . . . .                   | 33        |
| 8.3.2    | Aggregation Variables . . . . .                  | 33        |

|           |                                                                   |           |
|-----------|-------------------------------------------------------------------|-----------|
| 8.3.3     | Special Variables & Notation . . . . .                            | 34        |
| 8.3.4     | Breaking Down Residuals . . . . .                                 | 34        |
| <b>9</b>  | <b>Core RCA Proofs</b>                                            | <b>36</b> |
| <b>10</b> | <b>MARS Model</b>                                                 | <b>37</b> |
| 10.1      | Model Variable and Interaction “ $g$ ” Functions . . . . .        | 37        |
| 10.2      | Indexing of the $g$ Functions . . . . .                           | 37        |
| 10.3      | Graphing Dimensions . . . . .                                     | 38        |
| 10.4      | The Final Model Equation . . . . .                                | 38        |
| 10.5      | Number of Charts Needed to Graph a Model . . . . .                | 39        |
| <b>11</b> | <b>Data Characteristics</b>                                       | <b>42</b> |
| 11.1      | Purpose of the Residual . . . . .                                 | 42        |
| 11.2      | Condition and Quality Ratings as Approximate Regressors . . . . . | 43        |
| 11.3      | Data Errors . . . . .                                             | 43        |
| 11.4      | What Is Accuracy? . . . . .                                       | 44        |
| 11.5      | The Subject Property As An Anomaly . . . . .                      | 46        |
| 11.6      | The CVR2’s Importance In Dealing With Unexpected Data . . . . .   | 46        |
| <b>12</b> | <b>Residuals</b>                                                  | <b>48</b> |
| 12.1      | Ranking & Scoring . . . . .                                       | 48        |
| 12.2      | CQA-Residual Curve . . . . .                                      | 51        |
| 12.3      | The CQA Curve Characteristics . . . . .                           | 51        |
| 12.3.1    | Analysis of the CQA Curve Components . . . . .                    | 51        |
| 12.3.2    | Market Dynamics and Social Implications . . . . .                 | 52        |
| 12.4      | Subject Residual . . . . .                                        | 52        |
| 12.5      | RCA Value Conclusion . . . . .                                    | 54        |
| 12.5.1    | General Considerations . . . . .                                  | 54        |
| 12.5.2    | Review, Auditing, Complaints . . . . .                            | 54        |
| <b>13</b> | <b>Conclusion</b>                                                 | <b>56</b> |

## 1 Introduction

This paper presents the logical and mathematical foundation for a modern approach to accurately estimating the market value of residential and other real estate properties. The method, developed over 20 years of experience applying Multivariate Adaptive Regression Splines (MARS) software to Northern California homes — particularly in San Francisco Bay Area communities — offers a significant advancement in valuation precision. MARS enhances valuation quality by

adeptly capturing non-linear relationships between property features and value, while efficiently identifying and prioritizing the most impactful variables. This capability allows appraisers to uncover value dynamics in new market areas far more rapidly than traditional methods permit. MARS forms the backbone of the proposed Residual Constraint Approach (RCA). The RCA method earns its name through its rigorous framework:

- It imposes strict constraints on value contributions from both measured and unmeasured property features:
  1. All MARS-derived value contributions, including the residual, must sum to the net sale price of each comparable property.
  2. The residual may be decomposed into sub-components with assigned values, provided their total equals the comparable's residual.
- RCA minimizes bias in estimating subjective features — such as aesthetic appeal, condition, and quality — by tying their total value to the residual, which is accurately quantified when a competent valuation engineer constructs the MARS model.

While primarily designed for residential real estate, where subjective features significantly influence market value, RCA holds potential for broader asset valuation applications. Accurate appraisals are critical given the U.S. residential real estate market's estimated \$45 trillion value. Fannie Mae notes that typical appraisals deviate by about 5% from actual sale prices, though this figure masks bias, as appraisers often face pressure to align with sale prices. Moreover, since 60%–70% of appraisals support refinancing — lacking comparable sale prices — the need for precision, as offered by RCA, becomes evident. The Federal Reserve's 2012 analysis estimated that 15%–20% of pre-2008 crisis appraisals deviated by over  $\pm 10\%$  from sale prices, a range likely still relevant for refinancing appraisals today, varying by property type and market. Why does accuracy matter? For low loan-to-value (LTV) transactions — e.g., a buyer with strong credit putting \$500,000 down on a \$1 million home — lenders may waive appraisals, assuming minimal foreclosure risk. However, buyers who skip appraisals risk overpaying, discovering defects only after purchase. This underscores the value of robust valuation methods like RCA. Quantifying appraisal accuracy remains challenging. Researchers lack direct metrics for “market value” deviations, relying instead on indirect proxies. Traditional appraisal methods, rooted in subjective judgment and variable appraiser competency, complicate analysis. Strategies include:

1. Tzioumis (2016) studied 4,782 appraisers across 259,436 appraisals in 11 states, finding one-third within 1% of contract price, half between 1%–3% above, and a minority averaging 5% above — suggesting upward bias. The study identified two appraiser types: “at market value” (close to contract price) and “above market value” (consistently higher).
2. Krivorotov and LaCour-Little (2020) found 90% of purchase appraisals met or exceeded contract prices, highlighting “anchor bias” — the tendency to align with a desired value — especially in refinancing contexts.

These studies reveal bias in appraisal practices, amplified by the limitations of traditional tools. If appraisers were uniformly competent and unbiased, their valuations would converge; persistent overvaluation by some groups signals systemic issues.

The bias studies cited above, and the systemic pattern they point to, reflect a deeper feature of traditional sales-comparison practice that RCA is designed to address. The traditional methodology operates on two channels with very different epistemic status. On the measurable-feature channel — GLA, room counts, lot size, age, location — the appraiser is expected to derive adjustment values objectively, through matched-pair analysis or local linear regression

on comparable sales. Competent appraisers do this work, and the resulting adjustments are at least roughly defensible against the data. Even here, though, the traditional grid records only adjustments; it does not even have a column for value contributions. Nor are traditional adjustments calculated as differences between value contributions — they are estimated directly, from matched pairs or similar procedures, without the underlying contributions ever being computed. In RCA, by contrast, every adjustment is the difference between the subject's and the comparable's value contributions for the given feature — the structure whose consequences Proof II makes precise. On the subjective-feature channel — condition, quality of construction, design, functional utility, aesthetics — the available support procedures range from thin to threadbare. Subjective-feature adjustments can be supported by matched-pair analysis of properties outside the immediate comparable set, by appeal to recognized peer practice, or by reference to the appraiser's documented experience in the market. All three are formally compliant with professional standards. None of them rises to the statistical rigor that the measurable channel achieves routinely.

The result is a procedural asymmetry between what the standards require and what the methodology can deliver. The standards require support for every adjustment, measurable or subjective. The methodology delivers robust support on one channel and a spectrum of weaker procedures on the other, ranging from an anecdotal matched pair of sample size one to a citation of what other appraisers customarily use. The honest competent appraiser doing careful matched-pair work on a view adjustment produces meaningfully better support than one citing peer practice without basis, but a reviewer comparing the two appraisals on the standard form cannot easily distinguish them, because the form records the adjustment value, not the rigor of its derivation. The methodology tolerates the asymmetry because no stronger procedure for subjective features has been generally available to working appraisers.

RCA improves both channels, but in categorically different ways. On the measurable channel, the MARS regression captures non-linear relationships, variable interactions, and basis-function structure that matched-pair analysis and ordinary linear regression cannot represent. Fit against the full set of qualified sales in the market area — typically several hundred — rather than against the three to nine comparables of the traditional grid, the MARS-derived value contributions are both objective and substantially more accurate than the methodology they replace. On the subjective channel, RCA does something different in kind from any of the traditional support procedures. It does not attempt to estimate adjustment values for individual subjective features at all. The contributions of all subjective features — latent variables — collectively appear as the residual: the difference between each comparable's actual net sale price and its MARS-predicted price, calculable from data without appraiser input. The decomposition of that residual into named components — Design, Functional Utility, View, Condition, Quality, and so on — is constrained to sum to the residual total, and by Proof II (Section 9) any decomposition satisfying that constraint yields the same value conclusion. The judgment that remains in the workflow is the placement of the subject in the ranked residual distribution of the comparables, made against an inspectable ranking and explicitly documented in the report. A weak RCA appraisal is still possible — an inexperienced VE can produce an overfit MARS model, place the subject implausibly, or miss anomalies in the data — but it is weak in ways a reviewer can identify by inspecting the model and the ranking, rather than weak in ways that a standard-form review cannot surface.

Beyond refinancing, appraisers owe owners accurate valuations, especially for high-value purchases where errors can cost tens or hundreds of thousands. Yet, many balk at spending \$600–\$1,000 on appraisals, doubting their added value. Advanced methods like RCA, while initially resource-intensive, promise greater precision. With refinement and experience, their efficiency — and adoption — should increase.

## 2 Background

The author's educational background is in mathematics, computer science, and statistics. By profession, he is a software engineer with secondary experience in appraisal and accounting. He has been doing appraisals, off and on, since 2002 and has applied MARS since about 2004 for determining property adjustments and price trends in residential real estate.

Appraising with MARS is “deep experience” in appraisal. MARS generates models that reflect in greater detail than any other statistical method both the variables that contribute to market value and how changes in those values impact sale price. Thus, in appraising with MARS, the appraiser is constantly investigating causal relationships between variable values and sale prices as generated by MARS models.

In the interest of transparency, the RCA methodology is best applied to residential areas with these characteristics:

1. Complex neighborhoods with properties that are quite different from one another
  - Older homes
  - Varied qualities of updating
  - Many different architectures and designs
  - Variation in property attributes like GLA, lot size, elevation, and view.
2. Complex market conditions
  - Quickly changing market conditions
  - Many different types of buyers
3. Good data sources: The RCA protocol is based on Data Mining and needs plentiful quality data. While it can be used for areas with sparse data, it might not be worth the time and cost required.
4. Accurate estimates of market value are desired. RCA should, in most cases, be able to provide  $\pm 1\text{-}2\%$  accuracy. The valuation engineer report should comment on the conditions for accuracy in the final value conclusion. If data or information is lacking, or if consistent patterns in the market have not been uncovered through data mining, then the accuracy will suffer, and the client should be informed of the situation. Specifically, a high  $R^2$  and CVR2 (cross-validated  $R^2$ , computed by repeatedly partitioning the MLS data set into random training and test partitions), a meaningful MARS model, and a strong pattern in property CQA (condition-quality-appeal) features in the ranked residuals are the principal means for assessing accuracy and reliability in the value conclusion.

The development of these RCA protocols over the past 20+ years is based on the availability of MLSListings Inc. (Sunnyvale, CA) and other MLS, which are currently accessible through MLSListings, which comprises over 95% of the author's data analysis experience. The author's expertise stems from extensive appraisal work across 15 primary Northern California counties, with additional experience in 10 secondary counties.

The RCA methodology is most effective in markets that meet two key criteria:

1. Rich availability of high-quality appraisal data
2. Complex, heterogeneous residential areas characterized by:

- Diverse architectural styles
- Varying degrees of property updates
- Multi-decade development patterns

## California County Experience

Statements made in this paper are based on experience valuing properties in the following North California counties:

|            |               |              |            |           |             |
|------------|---------------|--------------|------------|-----------|-------------|
| San Mateo  | Santa Clara   | Marin        | Napa       | Monterey  | Santa Rosa  |
| Alameda    | San Francisco | Contra Costa | Sacramento | El Dorado | San Joaquin |
| Stanislaus | Placer        | Solano       | Mendocino  | Lake      | Colusa      |
| Merced     | Nevada        | San Benito   | Santa Cruz | Sutter    | Yuba        |

The above areas essentially take in the Silicon Valley and its surrounding counties, extending outward into more rural areas, including the Central Valley and Sierra foothills. It should be mentioned that the author has always had access to plentiful and *relatively* accurate data compared to what many appraisers in other less developed areas in the US have at their disposal.

The MLS data used is RESO Certified, complying with the RCF (RESO Common Format). Despite RCF certification, data of usually secondary importance is sometimes missing or incorrect for some properties, either through original tax assessor or subsequent sales agent MLS input errors. Often these errors follow a pattern, such as simply not making an input if the feature doesn't exist. For example, instead of entering 0 for "carport spaces", when there is no carport, nothing is entered. In areas where carports are rare, the analyst might substitute 0 for no data, in order to get a useful adjustment for carports. While R/earth and other MARS implementations can handle missing data, it is up to the Valuation Engineer (VE) to decide on how to handle it.

MLSListings data is also available from the MLS's of some other states, from Florida to Hawaii. This can be useful for testing how well protocols and code work in different states.

### 3 Valuation Engineer

#### Definition 1: Valuation Engineer

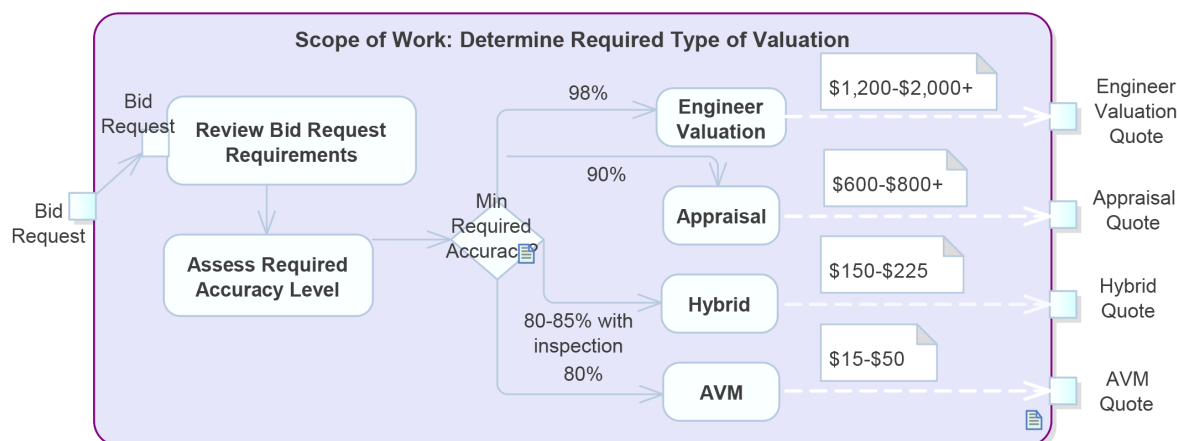
To prevent ambiguity, The term “Valuation Engineer” (or “VE” for short) is used to denote an individual possessing the expertise of a licensed appraiser, coupled with advanced knowledge, skills, and experience in utilizing data mining, statistics, programming, detailed protocols, and artificial intelligence to ascertain the market value of complex properties. It is also important to recognize that valuation is a task that necessitates some form of licensing due to the established potential for fraud.

Lenders typically offer several valuation options:

1. Automated Valuation Models (AVMs) - lowest cost option
2. Hybrid appraisals - utilizing unlicensed inspectors with licensed appraiser oversight
3. Standard appraisals - performed by licensed appraisers

Absent from typical consideration for appraisal work are Engineer-level Appraisals. There simply aren't many appraisers around with the skill set to do these. There should be a need for them. They can provide significantly higher accuracy and nearly guarantee fairly good objectivity with the employment of RCA methodology (MARS and constraints).

For conforming properties in newer subdivisions, standard appraisals generally suffice. However, engineer valuations are recommended for older homes in areas with significant renovation history spanning 30+ years. The choice between these methods should follow a structured decision process based on property characteristics and market conditions.



## 4 Assumptions

### Assumption 1: Geographical Area & Property Type

The scope of this paper primarily focuses on single-family homes and condominiums in California's metropolitan and urban areas. While the Sales Comparison Approach applies to other property types and geographical regions, income and cost approaches typically play a more dominant role when valuing agricultural land, multi-family properties, and commercial real estate. In such cases, the RCA approach may need modification or have limited applicability.

### Assumption 2: Data Storage

CSV and Excel are heavily used by MLS systems and assumed to be the principal way to feed data to RCA programs. Although broker feeds can be used to feed data continually into SQL databases. Backup and intermediate storage often uses SQL databases and two that play big roles are the free and open source SQLite and PostgreSQL. Text or RTF files are also used for report snippets and PNG, JPG, TIFF files are used for graphics. Note that Microsoft's SQL Server is also used by many, but the free edition "SQL Server Express" is very limited (e.g. only 1 CPU Core can be used) and other versions have hefty license fees for commercial use. Real Estate databases in California can be very large and joins can be very time consuming, so a 8+ core computer with 128G of RAM can be useful for large joins. SQL databases are also important when you need to find all of the properties in an arbitrarily bounded area. So, if for example, you used Cluster Analysis to define neighborhoods, you would be inclined to use some SQL database. However, different databases have different built-in functions for handling geographic information and thus are not compatible with one another. For this set of articles on RCA, PostgreSQL will be assumed for complex SQL tasks and SQLite for simple tasks.

### Assumption 3: "R/earth" Is Used For MARS processing.

Until 2017, Salford Systems' MARS software was an affordable solution at \$270 per year. However, after Minitab's acquisition of Salford Systems, MARS was bundled into their SPM package at \$16,000 annually — a price point prohibitive for most small businesses. Fortunately, the R programming language's "earth" package ([Milborrow 2024a](#)), maintained by Stephen Milborrow, has evolved to match or exceed the capabilities of Minitab's SPM MARS. The earth package offers additional advantages: it's freely available, integrates seamlessly with R's comprehensive statistical ecosystem, and can be accessed through Python using Pandas, providing excellent flexibility for different workflows. Discussions related to MARS assume the specific use of R/earth.

**Assumption 4: Notation**

Raw data from external sources like CSV files and Excel spreadsheets is represented using standard Latin characters. This includes Multiple Listing Service (MLS) property data, MARS (Multivariate Adaptive Regression Splines) configuration settings using the R/earth package, and project-specific data. This notation extends to their corresponding data frame representations in R or Python.

When discussing *property features* and their various types and subsets, the notation shifts to Greek letters for enhanced precision and analytical clarity. While this mathematical notation may appear more rigorous, it serves a crucial purpose: distinguishing abstract property characteristics from their raw data representations. As a convention, Greek characters consistently denote property features, while Latin characters indicate spreadsheet data structures. For instance, the symbol  $M$  represents the MLS data matrix, organized with properties as rows and their corresponding features as columns.

**Assumption 5: Programming Languages**

In this field R and Python w/Pandas are the dominant languages, with C or C++ being used when speed is important. R/earth is itself written in R and C.

**Assumption 6: Indexing**

Indexing can vary across programming languages and mathematics, which can get confusing:

- Mathematics and R start indexing at 1.
- C, C#, C++, and Python start at 0.

In this paper, to keep things consistent and simple, all indexing starts at 0 (like Python or C). Here's how it works:

- For lists, matrices, or data frames:
  - Index 0 is reserved for the “subject property” (the main property being studied).
  - Example: In a table where rows are properties, row 0 is the subject property, and column 0 holds unique IDs for all properties.
  - A practical tip: Arrange properties with the subject property first (row 0), then list others from newest to oldest sale date, numbering them sequentially (0, 1, 2, etc.).
- When there's no subject property:
  - If a data structure doesn't include properties (e.g., just variables or functions), index 0 is skipped, and indexing starts at 1.
  - Special note for R users: Since R uses 1-based indexing, when coding this paper's math in R, add 1 to all indexes to adjust from 0-based to 1-based.

## 5 RCA Workflow

Understanding the RCA workflow is essential for effective analysis. The process follows a structured sequence of analytical tasks and decision points

### 5.1 Types of MARS Analysis

The workflow for MARS analysis depends on whether only a market analysis is needed or whether the valuation or appraisal of a given property is required. Note that a specific property appraisal typically requires an analysis of its market area.

- **Market Analysis:** focuses on broad areas such as counties, cities, or defined regions. This approach does not involve a specific subject property and thus requires no value conclusion. The workflow centers on model development, validation, and the generation of supporting analytics. The process concludes once the optimal model is identified and documented.
- **Single Property Analysis** encompasses all elements of market analysis plus the determination of a specific property's value. This differs fundamentally from traditional appraisal methods. In traditional approaches, value is typically derived by adjusting 3-9 comparable sales and computing a weighted average. In contrast, RCA determines value primarily through two components:
  1. The MARS model's predicted value
  2. The estimated property-specific residual (with the associated CQA score)

The Sales Grid in RCA serves primarily to explain value differences between the subject and comparables by illustrating variable contributions. A key advantage of RCA is its stability: while clients may debate comparable selection, the value conclusion remains consistent regardless of which comparables are chosen for presentation, as all sales (whether 15 or 600) are adjusted using the same model. While older sales are automatically adjusted to current market conditions, recent sales are generally preferable. The model's effectiveness in capturing price factors diminishes with temporal distance, as unmeasured market characteristics become more significant over longer periods.

### 5.2 Processing Workflow

The overall processing workflow can be described by these steps:

1. Scope of work.
2. Setup initial project folders and files.
3. Download MLS data. Refer to Protocols 1.1–1.3 on the number of property records needed and setting limitations on date ranges.
4. Complete configurations sheets: Regression variables, Interactions, Aggregations, Calculations.
5. Run MARS (R/earth) regression to create price model.
6. Generate property value contributions & residuals (pure residual and residual/SF) from the MARS model.

7. Rank properties by residual/SF.
8. Create CQA scores for each comparable sale.
9. Assuming a quality ranking of properties has been produced which truly represents the overall appeal of the properties, find the best position for the subject in the ranking and assign it a CQA score, by studying the overall pattern trends in the ranking.
10. Assign the corresponding residual/SF to the subject property as residual/SF · (subject) GLA.
11. **Add the MARS price estimate and assigned residual to arrive at subject value conclusion.**
12. Break down the subject residual into descriptive components and value contributions.
13. Calculate comparable model produced value contributions & adjustments, plus all subtotals and adjusted sale prices.
14. The adjusted sale prices will be equal to the subject final value conclusion.
15. Select the 6-12 most similar properties for the Sales Grid.
16. Break down the total residual amounts for the comparables into residual component value contributions, setting the residual total value to 0 after the breakdown is completed.
17. Recalculate all subtotals and sales grid comparable adjusted sale price. Check values.
18. Upload Sales Grid to Report.

Note that with RCA, the final value conclusion does NOT directly derive from the comparable values in the Sales Grid. With the RCA method, the Sales Grid becomes a tool only for explaining why properties that appear similar to the subject, sell for more or less than the final value conclusion or the subject properties appraised value. This must be viewed a different way: The subject value conclusion depends on the MARS model price estimate, and the residual estimated by the valuation engineer. But these two quantities are very dependent on all of the properties that went into the R/earth regression, which created the ranking of property residuals that determined the CQA score of the subject property thus determined its residual through placement in the ranking by the valuation engineer. The adjustments in the Sales Grid are really determined by the imposed mathematical constraints of RCA, although the residual breakdown is somewhat at the mercy of the VE, the breakdown value contribution totals must always add up to the associated residual. We know from Proof 2 that the while the actual values if the residual components may vary depending on the judgment of the VE, their totals for a given property are completely constrained and therefore will not change the adjusted sale price for the associated property, regardless of how the VE alters the residual breakdown.

### 5.3 Data Workflow

Note that in this paper, only a high-level workflow is presented for the RCA method. Other articles will delve into the various stages and substages of RCA analysis.

### **5.3.1 Data Requirements for MARS**

We will need to specify a beginning date for sales transactions and an ending date based on the effective date of appraisal or valuation. This usually takes into consideration the estimated number of regression variables and interactions that will be needed for each degree of interaction over 1.

### 5.3.2 Protocols: Closed Sales Listings

#### Protocol 1.1: Determining the Number of MLS Closed Sales Listings Required

This protocol describes how to calculate the recommended minimum number of closed sales listings to input into MARS regression. You will need to make a rough estimate, to begin with, run MARS in first, second, and possibly third-degree runs to generate the initial model, and then review the number of rows according to the following rules to determine if more rows are needed. This protocol assumes the independent variables are 100% independent, which in real estate is rarely the case; for example, GLA and room counts are usually highly correlated, and GLA is also likely to show correlation with lot size. This gets into the calculation of “Degrees of Freedom (DOF).” Each truly independent variable gets one degree of freedom. But, when there are dependencies between the so-called independent variables, an accurate calculation of the DOF requires some analysis that is beyond the scope of this paper. Just keep in mind that due to likely co-linearity between variables, the estimates below are probably on the high side.

1. For first-order terms (no interactions), the  $\sim 15$  rows per variable rule works well
2. For each allowed two-way interaction, you should add approximately 15-30 properties
3. For higher-order interactions (3+ variables), each should have 30+ properties.

However, this isn’t a hard rule. The actual number needed can depend on:

- The signal-to-noise ratio in your data
- The complexity of relationships you’re trying to model
- The variability in your response variable
- The quality of your data

Example:

So if you have:

- 10 main variables
- 5 allowed two-way interactions
- 2 allowed three-way interactions

You might calculate your minimum number of sales as:

$$\begin{aligned} N &= (10 \text{ variables} * 15 \text{ rows/variable}) + \\ &\quad (5 \text{ two-way-interactions} * 22 \text{ rows/two-way-interaction}) + \\ &\quad (2 \text{ three-way-interactions} * 30 \text{ rows/three-way-interaction}) \\ &= 320 \text{ rows} \end{aligned}$$

**Alternatively, you may do the following:** After running R/earth on a set of data, you can use the number of basis functions in the model, plus one as an estimate for the degrees of freedom and then take this number times 20-30 to get the number of input records needed.

### Protocol 1.2: Sales Periods Should Encompass Minimal Change in Buyer Tastes

When collecting historical property data, there is no strict time limitation if changes in market preferences can be captured through appropriate variables. Consider these key points for handling architectural evolution in your market: Changes in dominant architectural styles (such as a shift from California Ranch to Mediterranean) can be modeled using:

- Subdivision identifiers
- MLS area codes
- Construction date ranges

However, if these style changes cannot be adequately captured through specific variables, you face an important constraint: while Date of Sale is typically used to adjust for market conditions, using it simultaneously to control for architectural preferences would confound these two distinct effects. In such cases, the preferred approach is to either:

1. Limit your data collection to the period after the introduction of the newer architectural style
2. Restrict your comparable selection to subdivisions sharing the same architectural style

This ensures your analysis maintains consistency in property characteristics while still allowing proper adjustment for market conditions over time. Alternatively, at least in this case, assuming you can extract the architectural style from the data, you could consider the interaction between these two variables. However, such extraction is not always possible.

### Protocol 1.3: If In An Area Of Few Sales Then Do The Best You Can

When working with MARS modeling for sales data, you need approximately 15 transactions for meaningful analysis. If you have fewer sales (for example, 5 properties), you can duplicate each record twice to reach the 15-property threshold. Important considerations for this approach:

1. Use only first-degree regression to reduce over-fitting
2. Ignore or skip cross-validation since the data set is too limited
3. While MARS will likely outperform matched pairs or standard linear regression methods in this scenario, be aware that duplicating data affects the validity of uncertainty measurements and statistical significance

The best long-term solution is to expand your data-set with additional actual sales. However, if you must proceed with limited data, data duplication provides a workable temporary solution while acknowledging its statistical limitations.

## 5.4 Project Folder Structure

It would be a good idea to discuss project folder structure. It is a good idea to have a separate folder for Market Analyses and Single Property Appraisals. Both of these folders have a similar subfolder structure:

```

RCA_Projects/
├── MarketAnalysis/
│   ├── Burlingame_20240501/
│   ├── Pacifica_20240310/
│   ├── RedwoodCity_20240615/
│   └── ...
├── SingleFamily/
│   ├── Pacific_Rosewood_123_20240510/
│   │   ├── Code/
│   │   │   ├── R/
│   │   │   │   └── ...
│   │   │   └── Python/
│   │   │       └── ...
│   │   ├── Data/
│   │   │   ├── MLS_Original.xlsx
│   │   │   ├── MLS/
│   │   │   │   ├── VER202402510153248/
│   │   │   │   │   ├── MLS.xlsx
│   │   │   │   │   ├── MLS_Stage2.xlsx
│   │   │   │   │   ├── MLS_Stage3.xlsx
│   │   │   │   │   ├── MLS_Stage4.xlsx
│   │   │   │   │   ├── MLS_Stage5.xlsx
│   │   │   │   │   └── Documents/
│   │   │   │   │       ├── LinearModels.pdf
│   │   │   │   │       ├── Cum_Dist.pdf
│   │   │   │   │       ├── Model.txt
│   │   │   │   │       ├── PartContrib.pdf
│   │   │   │   │       ├── Pairs.pdf
│   │   │   │   │       ├── Residual1.png
│   │   │   │   │       ├── Residual2.png
│   │   │   │   │       ├── ResidualSF.png
│   │   │   │   │       ├── Rpart.pdf
│   │   │   │   │       ├── VariablesImp.pdf
│   │   │   │   │       ├── VariableVsSP.pdf
│   │   │   │   │       └── ...
│   │   └── ...
│   └── ...
└── ...

```

Stage 1 processing will create the version folder with its sub-folders. It will copy the `MLS_Original.xlsx`, even though this spreadsheet *should* never change, but reflect exactly what was downloaded from the MLS as of a certain date and time; yet there is the possibility on a lengthier project that a new MLS download will be needed. Next, this stage will make a copy of the expanded `MLS.xlsx` workbook with its configuration parameters, into the version folder. There can be many version folders each created for a complete run of MARS.

Then, the following stages will generate column additions, sorting and other changes to the workbook and write an updated version as `MLS_Stage(N)` to the project data folder when the

appraisal is completed. To reiterate, the root MLS.xlsx folder is undergoing constant change, and so the version copy is an ongoing mutation of the original MLS workbook, with variable deletions and additions, plus other changes. The processing R or Python code most likely will run with input from the Data MLS.xlsx workbook. Thus we always have a copy of all the data behind each version of the RCA run.

As many versions are created as needed until it is not possible to make improvements, at which point the best version is selected as the final version. One could change the folder name by adding “\_Final” to the end of it, so it is apparent when looking through the directory. Some older versions with interesting or competing models, should usually be retained. Each version folder contains a number of text, diagram and graph files that can be embedded into a report. The final MLS\_Stage5.xlsx workbook contains as many sheets as necessary for the report. In particular it may have a spreadsheet for transfer to a Fannie Mae Sales Grid. It will also necessarily contain the computations or breakdown behind aggregations. This is potentially the most time-consuming part of the RCA approach - the struggle to find the perfect model in a difficult market area.

Everything needed to recreate the sales grids needs to be copied to the version folder, including the R or Python code used - if the code undergoes frequent changes.

#### 5.4.1 Protocols: Multiple MARS (earth) Runs

##### Protocol 2.1: Overfitting Runs To Surgically Study Market Area Pricing Factors

During initial earth runs, overfitting should be used strategically to discover underlying price factors. For example, a market area represented by 800 sales may contain only 3 properties with workshops, and in those isolated cases the workshop may have a major impact on the final sale price. A forward-pass-only run will produce an overfitted model that shows the approximate value contribution of these workshops. However, since there are only 3 such sales out of 800, the workshop terms will be removed in the backward pass, and certainly under cross-validation. The value that the forward-only pass assigns to workshops can be written down and transferred to the latent variable analysis, to be used for any property with a workshop that appears in the Sales Grid. Such overfitted models also give detailed estimates for many other thin-data features that would be eliminated by backward-pass pruning or cross-validated runs.

Save each of these exploratory runs as its own version folder: the recorded value contributions are evidence the VE will want to cite when breaking down residuals later. Understand, of course, that the VE must ultimately rely on pruning and cross-validation to make the final model robust against properties the regression never saw — such as a subject property that appears later on. Overfitted runs are instruments for study; they are never the final model.

### Protocol 2.2: Multiple Cross-Validated Runs Under Different earth() Parameters

Market areas differ in structure and size, and the VE usually needs to experiment with a series of runs: to develop an understanding of which variables impact price the most; to judge whether the existing set of measured variables does an adequate job of estimating value, or whether new data must be added (distance to water, elevation, distance to shopping centers, and so on); and to discover new kinds of interactions while disallowing others as inefficient or misleading. In the end, the VE wants a good R2 on the training data and the best possible CVR2 against the test data. Getting there may require collecting new data, improving the accuracy of existing data, or tuning the earth() parameters to the size and character of the market area (e.g., for the available degrees of freedom — see Protocol 1.1).

This is precisely why the project structure above supports many run version folders: each candidate model is preserved with its parameters and its R2/CVR2, so the trajectory across runs can be compared. A version whose R2 climbs while its CVR2 stalls or falls is fitting noise rather than market signal, and the version history is what makes that visible.

### Protocol 2.3: Prolog Expansion of Remarks and Descriptive Feature Columns

Although this facility is still in a somewhat experimental phase, earthUI now ships with a built-in Prolog engine (vProlog) and uses Definite Clause Grammars (DCGs) to parse the MLS free-text columns — Public Remarks above all, but also agent remarks and comma-delimited feature lists such as room features or existing structures. The expansion turns phrases like “owned solar,” “3 car garage,” or “40x50 workshop” into ordinary feature columns that can be offered to the regression, and thin-data features recovered this way feed the latent variable analysis directly (see the later subsection on condition and quality ratings as approximate regressors; the feature itself is documented in the earthUI companion article in this issue).

Text expansion multiplies the candidate predictor set, and that necessitates multiple earth/RCA runs for testing: expanded columns must be tried in and out of the model, consolidated into derived counts or booleans where they are collinear, and checked against CVR2 to confirm they carry market signal rather than noise. Treat each such experiment as its own run version folder, exactly as in Protocols 2.1 and 2.2. And since the parsing grammar itself is under development, the VE should review extracted columns against the source text before relying on them in a model.

## 5.5 The Core Excel Workbook

Since most MLS services download property data to Excel CVS worksheets, Excel is probably the most convenient way to store data and various configuration parameters. There is one core Excel Workbook for each project called `MLS.xlsx`, which has one sheet called “`MLSData`”, to store the closed property sales listings that were downloaded from the MLS, as well as several configuration and parameter sheets:

```
Workbook: MLS.xlsx/  
├─ Sheet: Project Data  
├─ Sheet: MLSData  
├─ Sheet: Regression Variables  
├─ Sheet: Calculations  
├─ Sheet: Allowed Interactions  
├─ Sheet: Aggregations  
└─ Sheet: (MARS run parameters)
```

The original data is stored in a workbook named something like `MLS_Original.xlsx`. A copy should be made with a different name, let’s say “`MLS.xlsx`,” which will load into the R or Python RCA program. Both of these workbooks are stored under the Data folder. Now, the `Data/MLS.xlsx` workbook will probably go through many iterations of runs, trying different combinations of variables, MARS parameters and changes to other settings. Each iteration creates a new input/output folder named with the date and time, with the format of `VER_YYYYMMDDHHMMSS`. Each folder will have a copy of the input `MLS.xlsx` workbook and contain the generated Excel Workbooks for each of the four processing stages (Stages 2–5), as well as the output text documents, diagrams and graphs. Note that Stage 5, the final stage, may contain a number of other sheets for the report, depending on the type of client and SOW.

## 5.6 Backups

When writing new or modified Excel spreadsheets to the project version folders, backups can also be made to SQLite or PostgreSQL databases in the project folder. The database files are less likely to become unintentionally corrupted over in subsequent periods, through review or reference use.

## 6 Workbook Notation

### 6.1 Descriptive Dimensions

There are roughly six descriptive dimensions for a workbook “M”:

1. Appraisals “a”
2. Run Versions “r”
3. Workflow Stages “w”
4. Sheets “s”
5. Property Listing “p”
6. Variable Set “v”

So, an individual cell in a particular workbook M can be described as:

$$\begin{matrix} r_* \\ w_* \end{matrix} \begin{matrix} a_* \\ W \\ s_* \end{matrix} \begin{matrix} p_* \\ v_* \end{matrix} \quad (1)$$

(Note: “\*” means some valid index)

or:

$$\begin{matrix} r_k \\ w_m \end{matrix} \begin{matrix} a_g \\ W \\ s_h \end{matrix} \begin{matrix} p_i \\ v_j \end{matrix} \quad \begin{matrix} i = 1, \dots, n_p \\ j = 1, \dots, n_v \\ k = 1, \dots, n_r \\ m = 1, \dots, n_m \\ g = 1, \dots, n_g \\ h = 1, \dots, n_h \end{matrix} \quad (2)$$

where:

- $n_p$  = # of rows (depends on sheet)
- $n_v$  = # of columns (depends on sheet)
- $n_r$  = # of run versions
- $n_m$  = # of workflow stages
- $n_g$  = # of analysis projects
- $n_h$  = # of workbook sheets

Understand that “W” references the entire Excel Workbook of six core data sheets that each have their own table, which on running the associated R or Python program are loaded into separate internal data frames. Both R and Python have internal functions that can take a name and find which row or column of the data frame belongs to. For example, if in the subscript  $v_j$ ,  $v = (\text{“GLA”, “LotSize”, “BathRms”, “BedRms”, } \dots)$  and  $j=0$ , then  $v_j = \text{“GLA”}$ . Either R or Python will internally be able to resolve “GLA” to the column it is stored in, which for example, might be column 15 of the associated data frame. So, in this paper we index into our name arrays, which pass a name to the running program which re-indexes that name into the actual column or row index for the internal data frame. It is a bit complicated. But to be precise, this is necessary and most convenient. It is a “separation of responsibilities” between the analyst role and the developer role; although in reality the Valuation Engineer (VE) could be performing both roles. Note that an R “data frame” is called “dataframe” in Python.

Also note that the MARS run time parameters sheet is not a dimension of W, as W relates to the version of data being input into MARS, separate from the parameters that determine how MARS/earth handles that data.

For the sake of clarity, a complex multidimensional notation for the Workbook is presented which relates more directly to the internal R or Python data frames that the data workbook sheets are loaded into. It is “heavy” notation. But when we get into proofs and so on, we throw a lot of the baggage away by making assumptions about what we are working with. Nonetheless, this notation helps get a better overview of the workflow and data from the outset.

Let’s start with notation for the set of workbooks we will be processing.

### 6.1.1 The Project Dimension (“a”)

Once a Scope of Work (SOW) has been approved, a project folder needs to be setup to store input and output data, including configuration parameters, output graphs, diagrams, spreadsheets and report snippets. The VE should identify all projects with a simple sequential ID, such as an integer, so that if a report should go missing, it would be obvious. If projects are canceled, then there should be a consistent means to indicate the project has been cancelled within the given project folder, but it should never be deleted. We assume project IDs start at 0 and incremented by 1, although there are other possibilities. We further assume all valid project IDs are stored in an ordered set “a” created with the processing program.

The appraisal project data is in the project worksheet of the workbook, but could also be stored in some central SQL database,

Here are the following suggested project fields:

1. Local ID: A short integer ID that is sequentially generated for the company. This would make a great primary key for storage - and could be automatically generated by the database with each successive project.
2. Global GUID: If you want, you can create a GUID (Globally Universal ID) using  
<https://www.uuidgenerator.net/>
3. Property Location (Market Analysis) or Address
4. Effective Date of Appraisal
5. Date of Order
6. Date of Inspection
7. Date of Original Report
8. Date of First Updated Report
9. Date of Second Updated Report
10. Date of Third Updated Report
11. Client Name
12. Client Address
13. Lender Name
14. Lender Address
15. Owner Name
16. Owner Address
17. ....

Example of list of project IDs:

```
a = ("1",
      "2",
      "...",
      "141",
      "142")
```

### 6.1.2 Run Version Dimension (“r”)

Second in the workflow, we execute MARS regression on the input. This will likely involve several to many executions as we alter various parameters to improve the model. Each execution will create a new version folder for the output, with a name based on the date and time of execution. For example,

```
r = ("ver20241210_083124",
      "ver20241209_151130",
      "ver20241208_160944")
```

```
latest_version : r1 = "ver20241210_083124"
```

Promising versions with competitive models and output, should be kept until a final version is decided upon, or perhaps longer if they have some utility. You may want to modify the version folder to indicate it is the final version, by placing “Final” at the end of the name.

### 6.1.3 Workflow Stages (“w”)

The RCA Workflow is separated into usually five separate stages, as listed below. Each stage must successfully complete before starting the next stage. The VE should be able to stop the processing at the end of a stage to review the results before proceeding. This is needed because market areas are fairly complex and often quite unique. Unusual problems can be encountered that take a new approach. The VE must be able to jump into R or Python code to make necessary changes and even try new approaches. And, in such cases, he may want to experiment and try different approaches requiring multiple runs of the same stage he is working on. Although, the VE should also be free to run all stages with one command, - with some caveats (e.g. 3rd Degree Aggregation must have already been determined through a previous run of Stage 2). This means that when each stage completes, it should store all of its internal data in files or databases, so that the next stage can be started as if the previous stage were still in memory, by reloading the data from the previous stage into memory.

Here are the stages:

```
w = ("stage1", -Setup
      "stage2", -MARS processing
      "stage3", -Final Sales Grid
      "stage4", -Report snippets
      "stage5") -Review/Cleanup
```

```
stage2 : w2 = "stage2"
```

### 6.1.4 Excel Workbook Sheets (“s”)

In the RCA Core Excel Workbook, we have the following sheets that are each mapped into a internal data frame:

```
s = ("MLSData",
      "Regression Variables",
      "Calculations",
      "Interactions",
      "Aggregations 1st & 3rd Degree",
      "Aggregations 2nd Degree")
```

The reason that 2nd degree aggregations are put on a separate sheet is that they can be most easily defined with a symmetric table that shows all possible pairs of interactions, allowing the specification of an aggregation variable for each pair's associated value contributions and adjustments to be aggregated (added) to. We couldn't do this with 3rd degree aggregations because that would require a 3 dimensional symmetric table which is not possible in Excel. Furthermore, a 20x20x20 table would be 8,000 cells to deal with. It is easier to just list any third degree term names in the table with the 1st degree aggregations, for example:

| Variable            | Aggregate To |
|---------------------|--------------|
| AboveGradeFinArea   | GLA          |
| BelowGradeFinArea   | BGLA         |
| UnfinArea           | UBA          |
| LotSize             | LotSize      |
| ...                 | ...          |
| GLA__LotSize        | GLA          |
| GLA__DateOfSale     | GLA          |
| BathRms__DateOfSale | GLA          |
| Lat__Long           | Location     |
| Lat__Long__GLA      | Location     |
| ...                 | ...          |

### 6.1.5 Property Listings (“p”) - MLSData rows

Property listings form the rows of the MLSData sheet. They are not found in the other sheets, until Stage 4, where the final report snippets are created with the final selection of comparable properties. Here is a sample list of property IDs:

```
p = ("MLS823211", "MLS823456", "MLS823542", "MLS823721", "...")
```

### 6.1.6 MLS Variables (“v”) MLSData Columns

More will be said about handling variables later. Suffice it to state at this point that MLSData sheet typically has 30-40 variables of various types, but usually only 10-20 will be selected as independent variables for the MARS regression. These variable also appear in other sheets, although typically as a list of all possible variables that can be used for submission to MARS. An example of a list of variables is:

```
v = ("GLA", "SalePrice", "BedRms", "BathRms", "GarageSF", "...")
```

### 6.1.7 Final Notation Example

Lets pull the above notation into a final example. Using the above lists, let

$$\begin{matrix} r_1 \\ w_1 \end{matrix} \begin{matrix} a_{141} \\ W \\ s_0 \end{matrix} \begin{matrix} p_2 \\ v_0 \end{matrix} \quad (3)$$

Note that we are using 0 indexing here, so that the first elements of lists is assumed to have an index of 0. Python also uses 0 indexing, but R uses 1 indexing.

So, the above means that we are talking about the cell value for:

- Variable  $v_0$  or column “GLA”
- For property  $p_2$  or ID “MLS823542”
- Of sheet  $s_0$  or “MLSDData”
- For workflow stage  $w_1$  or “stage2”
- Of version  $r_1$  or  
“ver20241209\_151130”
- Of project  $a_{141}$  or “142”

The above notation can also then be written as:

$$\begin{matrix} \text{“ver20241209_151130”} \\ \text{“stage2”} \end{matrix} \begin{matrix} \text{“142”} \\ W \\ \text{“MLSDData”} \end{matrix} \begin{matrix} \text{“MLS823542”} \\ \text{“GLA”} \end{matrix} \quad (4)$$

### 6.1.8 Justification

“Why don’t we just use indexing or numbers in the notation?” We have two requirements for our notation:

1. Subsets, e.g., the analyst frequently selecting subsets of a list, such as subsets of variables, are used. For instance, there may be 35-40 variables in the MLSDData sheet but only 10-20 of them are used for a given run. From one run to the next, the VE frequently changes the variables submitted to MARS/earth.
2. Iteration, e.g.,  $i = 1, 2, \dots, n_1$   
Our algorithms, protocols, and proofs usually need to be able to iterate through all items in an ordered list, from the first to the last - or pick out some particular item.

So, we need a letter to represent the subset or list of items, and that must have another subscript that we can use to iterate over the items of the list to give is the strings that are used to index into the data frame rows or columns.

### 6.1.9 Matrices?

And what about the matrices of Linear Algebra? Matrices work only with numbers, either integer or real. Matrices as a data structure are supported in both R and Python. However, many of the variables we work with in MLS data are called “strings” or character data. Even if some variables are only numbers, they are often not real numbers but just names. For example, MLS Area Codes for a city may be (“660”, “661”, “662”, “663”, ..., “672”). Earth regression allows variables to be specified as “factor” variables, in which case the numbers are treated simply as names, without any sequence. In such cases, the model produced will merely provide a value contribution for each number, rather than a function that runs over the sequence. Other examples include “C1”, ..., “C6” for Condition and “Q1”, ..., “Q6” for quality.

### 6.1.10 First Simplification

The above notation for the core Excel project workbook is nice to tie things together, but to heavy to carry around for explaining algorithms or mathematical proofs. When we are doing the tedious work, we can assume it is within the given project, version, stage and sheet. We just say what we are working with, if it is not already implicit. All we really need from the above, once we indicate what we are working with is:

$$M_{s_h}^{p_i}_{v_j} \quad \begin{array}{l} i = 1, 2, \dots, n_p \\ j = 1, 2, \dots, n_v \\ h = 1, 2, \dots, n_s \end{array} \quad (5)$$

where:  $n_p$  = # properties  
 $n_v$  = # input variables  
 $n_s$  = # input sheets

### 6.1.11 Second Simplification

It is more convenient to rename the sheets to specific kinds of data frames they support rather than use indexes. For each sheet, the rows and columns have different functionalities. So, the W set of data frames is broken up into M, V, A, I, and C data frames as follows:

$$\diamond \text{ MLS Data} \quad M_{v_j}^{p_i} = W_0^{p_i}_{v_j} \quad \begin{array}{l} i = 0, 1, \dots, n_p \\ j = 0, 1, \dots, n_v \end{array} \quad (6)$$

where:  $n_p$  = # properties  
 $n_v$  = # input variables

#### **Note: Subject Is By Default Always First in Property Lists**

In the “MLSData” Excel spreadsheet and associated data frame, the first row is always reserved for the subject property. It is never sorted with the other MLS properties below it. It is never entered into the MARS regression as independent data, it is entered rather as target data, or as the dependent variable. The same holds for property lists. In all property lists, the first element with index 0, is always assumed to be the subject property ID, unless stated otherwise.

Even if the task is to create a model for a market area without any subject property, the first row is nonetheless loaded with some dummy property data. This makes it far more convenient to write programs, specify algorithms, or do proofs. So, if we are using  $c$  to

represent the list of properties,  $c_0$  is the subject property and  $c_i$ ,  $i = 1, \dots, n_p$  are the comparable listings that will be input to MARS/earth regression.

◇ **Regression Variable Specifications**

$$V_{s_j}^{v_i} = W_1^{v_i}_{s_j} \quad \begin{matrix} i = 1, 2, \dots, n_v \\ j = 1, 2, \dots, n_s \end{matrix} \quad (7)$$

where:  $n_v = \#$  available variables  
 $n_s = \#$  of specifications

Note,  $n_s \leq n_v$ , as we usually select a subset of available variables for input to regression.

◇ **Calculations**

$$C_{c_j}^{v_i} = W_2^{v_i}_{c_j} \quad \begin{matrix} i = 1, 2, \dots, n_v \\ j = 1, 2, \dots, n_c \end{matrix} \quad (8)$$

where:  $n_v = \#$  of variables  
 $n_c = \#$  of calc specs

◇ **Interactions**

$$I_{t_j}^{f_i} = W_3^{f_i}_{t_j} \quad \begin{matrix} i = 1, 2, \dots, n_f \\ j = 1, 2, \dots, n_t \end{matrix} \quad (9)$$

where:  $n_f = \#$  from variables  
 $n_t = \#$  to variables

Note that Data Frame I is a symmetric data frame, with variables alphabetically listed for the “from” rows and “to” columns. The cells are marked with a “1” to indicate an interaction between the associated pair of variables is allowed. Otherwise, “0” indicates that no interaction is permitted. For a second-degree interaction, this is relatively simple, with only one interaction possible. With a third-degree regression, we have three pair interactions between all three variables, where every pair of interactions must be allowed to generate the three variable interaction term (or variable).

◇ **Aggregations**

$$A_{v_j}^{v_i} = W_4^{v_i}_{v_j} \quad \begin{matrix} i = 1, 2, \dots, n_v \\ j = 1, 2, \dots, n_v \end{matrix} \quad (10)$$

where:  $n_v = \#$  of variables

## 7 The RCA Sales Grid

Keep in mind that the RCA protocol requires an R or Python program that can access R/earth and perform all of the detailed steps. It is data intensive and the work could not possibly be done manually. At the same time, assuming the program is bug free, you need a software program to ensure that the required work is done precisely and without human error.

It is also a necessity for VE's who do this work to be able to modify such programs as needed, especially when they are dealing with new unforeseen market areas, neighborhoods, properties or other new requirements or problems.

Within RCA processing we have three types of Sales Grids:

1. The RCA Processing Sales Grid, labeled in this paper as “MLSDData,” which is a very large data frame and associated spreadsheet that may contain many intermediate details that eventually go through an aggregation process, to arrive at the RCA Sales Grid mentioned below. An example would be too large to display here. Most likely valuation engineers will tailor this spreadsheet to their preferences. I intend to write a separate paper on developing such a grid at some future date.
2. The RCA Final Sales Grid which is used in non-GSE reports that looks something like [Table 2](#). This does mimic for example Fannie Mae's form, enough to provide some base of familiarity to the reader. However, notice that it carries more columns per sales comparable than the Fannie Mae grid: each comparable has a “Value Contribution” column alongside its “Measure” and “Adjustment” columns, and the subject has one as well — columns the Fannie Mae form lacks. A simpler form is also the [Table 1](#).
3. The GSE Upload Sales Grid that is used to upload the Sales Grid into software such as Alamode. Note however, that not all appraisal form software packages provide the ability to upload sales grid data from a spreadsheet. Out of necessity, such grids cannot include a separate column for valuation contributions, of which the GSEs like Fannie Mae have no comprehension. The strategy is to give them their required form, but back it up with an RCA Sales Grid, as well as other supporting documentation such as details on any variable aggregations.

The RCA Processing Sales Grids do not look like Fannie Mae Sales Grids but are, roughly, a transpose of the Fannie Mae Sales Grid, with rows and columns switched: Each row contains data for some property, and each column contains data for some property variable. There can easily be hundreds or thousands of property rows and 200 or more columns. The first row is reserved for variable names, and the second for a subject property, even if we are only doing a market analysis. When the spreadsheet is read into a data frame, then the first row of the sheet becomes the names attribute, and the second row for the subject property becomes the first row of the data frame.

The columns in the RCA Sales Grid include identification columns, then primary, external, and synthetic variables combined and fed as the independent variables into MARS. MARS outputs from these variables those that it thinks have a significant impact on the Sale Price, also known as the dependent or target variable. In second-degree regression, we may get pairs of these variables that act as new variables. With third-degree regression, we may also get variable triplets that act as so-called independent variables, although they are, of course, highly correlated with the underlying variables.

Table 1: Sample RCA Spreadsheet: Vertical Layout

| Subject                |                    |                    | Comp 1      |                    |                | Comp 2      |                    |                |
|------------------------|--------------------|--------------------|-------------|--------------------|----------------|-------------|--------------------|----------------|
|                        | Measure            | Value Contribution | Measure     | Value Contribution | Adjustment     | Measure     | Value Contribution | Adjustment     |
| Sale Price             |                    |                    | \$1,150,000 |                    |                | \$1,500,000 |                    |                |
| Base Value             |                    | \$500,000          |             | \$500,000          | \$0.00         |             | \$500,000          | \$0            |
|                        | Measured Features  |                    |             | Measured Features  |                |             | Measured Features  |                |
| Living Area            | 1,530              | \$463,500          | 1,350       | \$382,500          | \$81,000       | 1,700       | \$540,000          | -\$76,500      |
| Lot Size               | 6,000              | \$200,000          | 6,500       | \$220,000          | -\$20,000      | 7,000       | \$240,000          | -\$40,000      |
| Age                    | 53                 | \$63,600           | 57          | \$68,400           | -\$4,800       | 51          | \$61,200           | \$2,400        |
| Bathrooms              | 2                  | \$15,000           | 3           | \$30,000           | -\$15,000      | 3           | \$30,000           | -\$15,000      |
| Bedrooms               | 3                  | \$6,000            | 4           | \$9,000            | -\$3,000       | 4           | \$9,000            | -\$3,000       |
| Total Measured         |                    | \$1,248,100.00     |             | \$1,209,900.00     | \$38,200       |             | \$1,380,200.00     | -\$132,100     |
|                        | Residual Features  |                    |             | Residual Features  |                |             | Residual Features  |                |
| Residual               | CQA 5.8            | \$30,000           | CQA 2.5     | -\$59,900          | \$89,900       | CQA 8.2     | \$119,800          | -\$89,800      |
|                        | Residual Breakdown |                    |             | Residual Breakdown |                |             | Residual Breakdown |                |
| Design                 |                    | 7,000              |             | -\$15,000          | \$22,000       |             | \$35,000           | -\$28,000      |
| Condition              |                    | 5,000              |             | -\$8,000           | \$13,000       |             | \$20,000           | -\$15,000      |
| Quality                |                    | 8,000              |             | -\$10,000          | \$18,000       |             | \$30,000           | -\$22,000      |
| Func Utility           |                    | 5,000              |             | -\$12,000          | \$17,000       |             | \$15,000           | -\$10,000      |
| Amenities              |                    | 5,000              |             | -\$14,900          | \$19,900       |             | \$19,800           | -\$14,800      |
| Sub-Total              |                    | 30,000             |             | -59,900            | \$89,900       |             | 119,800            | -\$89,800      |
| Total                  |                    |                    | Total       |                    |                | Total       |                    |                |
| Tot Value Contribution |                    | \$1,278,100.00     |             | \$1,150,000.00     |                |             | \$1,500,000.00     |                |
| Adjusted Sale Price    |                    |                    |             |                    | \$1,278,100.00 |             |                    | \$1,278,100.00 |



## 8 Variables

On the input side of running MARS, we have only simple variables which are:

1. Property features loaded from the MLS spreadsheet.
2. Derived variables created by the VE to refine the MLS variables or add transformations of existing variables. For example, the VE may prefer to transform Date of Sale into “DaysOffMarket” which is equal to the effective date of the appraisal minus the Date of Sale.
3. The VE may import data from 3rd party sources, such as mortgage interest rates. Or, if latitude, longitude or elevation data are missing from the MLS data, he may import it from other sources. These variables could be called “external variables.”

On the output side of MARS, we work with price model that is a combinations of terms that either:

1. Contain only one of the input variable and provide the equation for the value contribution of that variable. These are called first degree terms or (output) variables. The term itself is treated as a value contribution value and given a name.
2. Contain two of the input variables and are the equation for the value contribution of the interaction of the given two variables. These terms maybe called second degree terms or (output) variables.
3. Contain three of the input variables and are the equation for the value contribution of the interaction between the three variables. These terms may be called 3rd degree terms or (output) variables in this paper.

The output variables of MARS are then value contributions, including the residual value contribution. They are in turn treated as variables in Sales Grid calculations, such as calculating adjustments or creating graphs.

We can have different types of variables that fall into two basic ontologies: (1) Variable Origin and (2) Variable Application, i.e. Non-residual vs Residual

### 8.1 Variable Origin Ontology

1. First Degree Variables
  - 1.1 MLS Regression Variables
  - 1.2 External Regression Variables
  - 1.3 Derived Regression Variables
2. Factor Variables:

To handle factor variables, a function is needed that can identify them. Since they are specified in the Excel project configuration workbook, a function is created that returns true if a variable passed to it has been classified as a factor type.
3. Interaction Variables: These are created by MARS in generating the model, if the degree is set to 2 or higher. However, 3 interacting variables is the most that could be recommended due to the difficulty of interpretation, the probability of overfitting and the impact on Degrees of Freedom and the resultant increase in property records to ensure a robust model.

- 3.1 Second Degree Variables: These are common and a good example would be GLA:BathRms, GLA:LotSize, DateOfSale:BathRms, Latitude \_Longitude.
  - 3.2 Third Degree Variables: Less common interactions that suggest caution and possible over-fitting.
4. Residual Breakdown Variables

## 8.2 Variable Application Ontology

- 1. MARS Generated Value Contributions
  - 1.1 First Degree Variables
    - 1.1.1. MLS Regression Variables
    - 1.1.2. External Regression Variables
    - 1.1.3. Derived Regression Variables
  - 1.2 Interaction Variables
    - 1.2.1. Second Degree Variables
    - 1.2.2. Third Degree Variables
  - 1.3 Residual Breakdown Variables
  - 1.4 Aggregation Variables
- 2. Engineer Generated Variables & Values
  - 2.1 Residual Breakdown Variables
    - 2.1.1. Design/Aesthetics
    - 2.1.2. Landscape
    - 2.1.3. Functional Utility
    - 2.1.4. View
    - 2.1.5. Condition
    - 2.1.6. Quality
    - 2.1.7. ...

These variable groups will be given their notation and described in more detail in the next section

**Protocol 3.1: Factor Variables**

Factor variables, or simply factors, can occur in all terms, including interactions. They are important because they must be specially dealt with for MARS regression and graphing. While variables with character values are necessarily factors, numerical variables must be flagged as factor variables for MARS to recognize them as such. For example, MLS areas are often identified by number. In particular, the neighborhood areas in the San Francisco Bay Area are often identified with unique integers. If these values are passed to MARS as factors, then MARS will attempt to discover if certain areas have an average contributing value to property sales within their boundaries without regard to any relationship with other areas. In contrast, with normal variables, it will assume there is some kind of continuity in value contributions as the variable value changes. This is an important distinction. Furthermore, these factor variables can appear in model terms with non-factor variables, which makes graphing difficult. Specifically, when graphing single variable terms with factors, we lump all such factors together and simply draw a bar graph showing their contributions, positive or negative.

If the factors appear with other variables in terms, they will always appear with one possible value, because factor variables can have only one value for a given property: If an area has the value 662, by definition it can't have any other value for that area. If you find this confusing, to make matters worse, in the MARS-generated model, there is not a single variable name for the factor; MARS will take the root name of the factor variable it was given as input, then create new columns with variable names formed by appending the possible values. So, a single column in the Excel spreadsheet, such as `MlsArea`, will have multiple data frame columns created for each possible value, such as `MlsArea660`, `MlsArea661`, etc. A variable such as `MlsArea661` will have a value of 1 for properties in area 661; otherwise, 0. Then, MARS will regress on those columns as if they represented Boolean variables. These modified names will be used in the model provided by MARS. You will likely want to convert those model names back to their original name to make the graphs more understandable to clients who like it nice and simple (at least as simple as possible). Instead of seeing `MlsArea661=1`, we will see `MlsArea=661`. We then can graph terms by simply stating in the graph header something like "For MLS Area = 662". Thus, N degree terms with M factor variables are graphed as (N-M) degree terms, and the factor dimension disappears into the title.

Note that two or more factor variables could exist in a single term, so you would likely need a graph for each possible pair of values. If variable x with three distinct values and variable y with two distinct values appear in the same term with non-factor variable z, then you would need  $2 \cdot 3 = 6$  graphs. An example would be "MlsArea" + "Architectural Style", e.g. Victorian homes in San Francisco might be worth more or less depending on the area of San Francisco they are in. Fortunately, in most areas, there are usually only 1-3 architectural styles important enough as value contributors to be used in the model (although there may be 6+ different architectural styles).

**8.3 Second- and Third-Degree Terms**

With second-degree or third-degree regression, some of terms in the output model, may include two or respectively three variables. For example:

$$\text{BathRms\_GLA} = -52 * \max(0, 1200 - \text{GLA}) * \max(0, \text{BathRms} - 1)$$

Notice that we name these by alphabetically sorting the variable names and concatenating them,

perhaps with an underscore. We do the same with third degree terms on the variable name triples produced in the model. For example, “BathRms\_\_GLA\_\_LotSz.”

These variables get added to the primary input variables in Stage 3, when the RCA program calculates the contributions of all variables for each property, including the interaction variables.

### **8.3.1 Adjustment Variables**

The RCA program will also calculate the adjustments for all variables and properties in Stage 3, after value contributions are calculated, by subtracting the corresponding comparable property value contribution for a feature from the subject contribution for that feature.

### **8.3.2 Aggregation Variables**

Aggregation variables are defined to replace subsets of the variables, adding all their value contributions and adjustments into a common variable to be used in the report. For example, we may have split both finished and unfinished living area into finer detail variables:

1. Above Grade Legal Finished Area
2. Below Grade Legal Finished Area
3. Above Grade Illegal Finished Area
4. Below Grade Illegal Finished Area
5. Above Grade Unfinished Area
6. Below Grade Unfinished Area

Yes there are areas where this does make sense.

The MARS regression will regress on all of the variables and generate a model that will provide contributions and adjustments for them. But for a form report, these different contributions will probably have to be combined or aggregated into fewer variables. The details of the aggregation should be supplied in an addenda. A more frequent example, would be aggregating interaction adjustments into one of the primary variable adjustments. For example, we might aggregate the adjustment for GLA\_LotSize to GLA or to LotSize, or even split them 50:50 between both, as determined by the Aggregation sheet.

### 8.3.3 Special Variables & Notation

|                  |                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------|
| $\pi_i$          | Actual net sale price for property $i = 1, 2, \dots, n_p$ ; for the subject, $\pi_0$ denotes the value conclusion (see Proof I) |
| $\alpha_i$       | Adjusted net sale price for property $i = 1, 2, \dots, n_p$                                                                     |
| $\rho_i$         | MARS estimated net sale price for property $i = 0, 1, \dots, n_p$                                                               |
| $\beta$          | Base value of the model                                                                                                         |
| $\varphi_{i,j}$  | Value contribution of regression variable $j = 1, 2, \dots, n_r$ for property $i = 0, 1, \dots, n_p$                            |
| $\delta_{i,j}$   | Adjustment for regression variable $j = 1, 2, \dots, n_r$ for property $i = 1, 2, \dots, n_p$                                   |
| $\epsilon_i$     | Total residual for property $i = 0, 1, \dots, n_p$                                                                              |
| $\epsilon_{i,c}$ | Residual component value contribution $c, c = 1, 2, \dots, n_c$ for property $i = 0, 1, \dots, n_p$                             |
| $\xi_{i,c}$      | Adjustment for residual component $c, c = 1, 2, \dots, n_c$ for property $i = 0, 1, \dots, n_p$                                 |

where

$n_p = \#$  of comparable properties

$n_r = \#$  of regression variables

$n_c = \#$  of residual breakdown variables

### 8.3.4 Breaking Down Residuals

Once the MARS regression has created a model, it generates the estimated sale prices from the MARS-selected input variables. It subtracts that value from the corresponding Net Sale Price to get a residual value for each property. Now, the RCA protocol stipulates that once the VE has decided on the final property comparables, usually 6-12 property listings, he needs to go through each property listing and break down the calculated residual into the most likely features (latent variables) he thinks impact value, ignoring the possibility of random data errors. He might, for example, have these breakdown variables for a given property:

1. Design
2. Functional Utility

3. Over/Under Market Sale
4. Condition
5. Quality

Different properties will likely have some different residual breakdown variables, which need to be collected in one ordered set.

**Estimating rare-feature contributions.** Some value-bearing features are too rare for the regression channel: a workshop present in 3 of 780 sales cannot earn a defensible coefficient, and the forward pass will usually pass over such a variable entirely. These are residual features, and their contributions can be estimated *in residual space*: fit the model without the feature, then regress the model residuals on the excluded column(s) by ordinary least squares. For a binary flag the coefficient is a lump-sum contribution (equivalently, the mean residual offset of the flagged sales); for a numeric column it is a per-unit contribution. When several correlated amenities are estimated *jointly* (e.g. outbuilding square footage together with a workshop flag), the regression splits the credit between them rather than double-counting the same dollars in two breakdown rows. Estimates supported by fewer than about ten observations are indicative only and should be corroborated by depreciated cost or a widened sample (e.g. flags recovered from public remarks); the residual constraint itself caps whatever value is finally assigned. earthUI (v0.11+) implements this method directly as the *Residual Contribution* tab, including the joint estimation, per-sale residual evidence, and a documentation note for the workfile.

Notation: We use the notation in the above table to arrive at:

$$\epsilon_i = \epsilon_{i,1} + \epsilon_{i,2} + \dots + \epsilon_{i,n_c} \quad (11)$$

where  $n_c$  is the number of residual breakdown components.

The VE must use his judgment to break down the residual into the given components. One might think this would allow bias to enter the final value conclusion. However, that is not the case because we have mathematical proof that if the breakdown adjustments total the residual, they will not impact the value conclusion. The purpose of the breakdown is not to establish value, but to explain why the residual is what it is in comparison to other property sale residuals. The only spot where the engineer's bias can impact the final value conclusion is the estimate of the residual for the subject since, of course, there is no actual sale price for the subject needed to calculate a residual for it.

Each property has its set of residual value contributions, but these need to be merged into one set C over all properties:

$$\begin{aligned} \text{Letting: } A_i &= \{\{\epsilon_{i,c}\}_{c=1}^{n_c}\} \\ C &= \{x \mid x \in \bigcup_{i=0}^{n_p} A_i\} \end{aligned}$$

where  $[\epsilon_{i,c}]$  is the variable name for the value  $\epsilon_{i,c}$  and  $A_i$  is the set of such variables and  $n_c$  is the number of values in each  $\epsilon_i$ .

So, with the subject and comparable property data in separate columns, then under all the regression model variable value contributions for each property, we should see the set C of  $n_c$  corresponding residual value contributions.

## 9 Core RCA Proofs

Let:

$$\begin{aligned} n_p &= \# \text{ of properties} \\ n_r &= \# \text{ of regression variables} \\ n_c &= \# \text{ of residual component variables} \end{aligned}$$

**Proof I: Each and every comparable adjusted sale price equals the estimated subject sale price.**

For any comparable  $i$ , the adjusted sale price  $\alpha_i$  is the sum of the net sale price  $\pi_i$  plus the adjustments: the differences between the subject and comparable value contributions for the base value, the regression variables  $j = 1, 2, \dots, n_r$ , and the residual. For the subject property, which has no actual sale, the net sale price  $\pi_0$  is defined as the value conclusion  $\beta + \varphi_{0,1} + \varphi_{0,2} + \dots + \varphi_{0,n_r} + \epsilon_0$ . Then:

$$\begin{aligned} \alpha_i &= \pi_i + (\beta - \beta) + (\varphi_{0,1} - \varphi_{i,1}) \\ &\quad + (\varphi_{0,2} - \varphi_{i,2}) + \dots + \\ &\quad (\varphi_{0,n_r} - \varphi_{i,n_r}) \\ &\quad + (\epsilon_0 - \epsilon_i) \\ &= \pi_i + (\beta + \varphi_{0,1} + \varphi_{0,2} + \dots \\ &\quad + \varphi_{0,n_r} + \epsilon_0) \\ &\quad - (\beta + \varphi_{i,1} + \varphi_{i,2} + \dots + \\ &\quad \varphi_{i,n_r} + \epsilon_i) \\ &= \pi_i + \pi_0 - \pi_i \\ &= \pi_0 \end{aligned} \tag{12}$$

**Proof II: Altering the residual breakdown under the constraint that all residual component value contributions sum to the comparable residual, has no impact on the final value conclusion of the RCA protocol.**

*This is almost trivial. Yet without thinking about the math, it is easy to overlook this because traditional appraisers focus on the adjustments rather than the value contributions. It is the value contributions that cancel out here — a decades-long oversight.* For some comparable  $k$ ,  $k > 0$ , the total residual adjustment is equal to the sum of the  $n_c$  breakdown residual components which in turn is equal to the sum of the differences of value contributions, which is equal to the subject total residual minus the comparable total residual. And this is independent of the number of residual components created or what they represent. As long as the value of the residual components equals the total residual for both the subject and chosen comparable,  $\xi_k$  will not change.

$$\begin{aligned} \xi_k &= \xi_{k,1} + \xi_{k,2} + \dots + \xi_{k,n_c} \\ &= (\epsilon_{0,1} - \epsilon_{k,1}) + (\epsilon_{0,2} - \epsilon_{k,2}) + \dots + \\ &\quad (\epsilon_{0,n_c} - \epsilon_{k,n_c}) \\ &= (\epsilon_{0,1} + \epsilon_{0,2} + \dots + \epsilon_{0,n_c}) - \\ &\quad (\epsilon_{k,1} + \epsilon_{k,2} + \dots + \epsilon_{k,n_c}) \\ &= \epsilon_0 - \epsilon_k \end{aligned} \tag{13}$$

## 10 MARS Model

### 10.1 Model Variable and Interaction “ $g$ ” Functions

A core principle of the RCA method, is that “explanatory reasoning” is of utmost importance. Unfortunately providing meaningful graphs for the various value contribution equations, especially when interactions and factor variables are used, can be challenging. It is therefore considered imperative to provide a clear and strict protocol for creating these graphs. This is done herewith, but understand the protocol is aimed at the developer who is writing the R or Python code to generate the graphs, and might be only superficially of interest to those not involved with the coding.

The MARS-generated price model is a sum of a base constant plus additional terms, each involving 1 to 3 variables. To analyze it, we first split the model into individual terms. Then, multiple terms sharing the same set of variables, representing a single interaction, are combined into single interaction expressions. Assume the maximum degree of regression is 3. Consider a third-degree regression model consisting of a base constant,  $n_1$  first-degree expressions,  $n_2$  second-degree interaction expressions, and  $n_3$  third-degree interaction expressions. This gives us four distinct groups of expressions, which are either single variable or multiple variable interactions, each graphed differently based on two factors: (1) whether they include a factor variable (a categorical variable), and (2) the number of unique variables they contain. Specifically:

- First-degree expressions (1 variable) are graphed in 2D.
- Second-degree expressions (2 variables) are graphed in 3D.
- Third-degree expressions (3 variables) are graphed as a *set* of 3D graphs.
- If an expression contains a factor variable, each unique factor reduces the graph’s dimensionality by one (e.g., a 3D graph becomes 2D if one variable is a factor).

The indexing of each function  $g$  provides the information needed to determine how it should be graphed, reflecting the number of unique variables and the presence of factor variables.

### 10.2 Indexing of the $g$ Functions

Each function  $g$  is indexed as  $f_{j,k} g_k^j$ , where:

- $f_{j,k}$  is the first index (a superscript before  $g$ ):
  - $j$  ranges from 0 to 3 and identifies the group:
    - \*  $j = 0$ : the base constant (an exception to the usual indexing).
    - \*  $j = 1$ : first-degree expressions.
    - \*  $j = 2$ : second-degree expressions.
    - \*  $j = 3$ : third-degree expressions.
  - $k$  is the index of a specific function within its group (e.g.,  $k = 1$  for the first function in group  $j$ ).
  - The value of  $f_{j,k}$  indicates the number of unique factor variables in the function:
    - \*  $f_{j,k} = 0$ : no factor variables.
    - \*  $f_{j,k} > 0$ : the number of factor variables (e.g., 1, 2, or 3).
- Trailing indices on  $g$ :

- The superscript  $j$  after  $g$  (e.g.,  $g^j$ ) repeats the group number for clarity.
- The subscript  $k$  after  $g$  (e.g.,  $g_k$ ) matches the  $k$  in  $f_{j,k}$ , identifying the function within its group.

While the indices of  $f$  and  $g$  align numerically, their roles differ:  $f_{j,k}$  counts factor variables, which is critical for graphing decisions, while  $j$  and  $k$  on  $g$  define its group and position.

### 10.3 Graphing Dimensions

The dimension  $d$  of the graph for a function  $f_{j,k}g_k^j$  is calculated as:

$$d = j - f_{j,k} \quad (14)$$

where:

- $j$  is the number of unique variables (or group number, ignoring the constant case).
- $f_{j,k}$  is the number of factor variables.
- $d$  is the number of continuous (non-factor) variables to be graphed, with a maximum of 3 (since the max degree is 3). A function with  $d$  continuous variables is drawn in  $d + 1$  dimensions:  $d = 1$  gives a 2D graph,  $d = 2$  a 3D graph (or a 2D heat map), and  $d = 3$  a set of 3D graphs.

For example:

- If  $j = 3$  (3 variables) and  $f_{j,k} = 0$  (no factor variables), then  $d = 3 - 0 = 3$ , requiring a set of 3D graphs, each fixing one of the variables.
- If  $j = 3$  and  $f_{j,k} = 1$  (one factor variable), then  $d = 3 - 1 = 2$ , requiring a 3D graph (or 2D heat map) for each value of the factor variable appearing in the model.
- If  $j = 0$  (the constant), graphing isn't typically needed, as it's a single value.

When  $d = 3$  (e.g., a third-degree expression with no factor variables), visualizing the full function may require multiple 3D graphs, each fixing one variable at different values to show a subset of the function's behavior.

### 10.4 The Final Model Equation

The MARS model is broken into four groups of expressions (constant, first-, second-, and third-degree). The indexing  $f_{j,k}g_k^j$  tells us:

- $j$ : the group and number of variables.
- $f_{j,k}$ : the number of factor variables.
- $d = j - f_{j,k}$ : the graph dimension.

This structure guides how each  $g$  function is visualized, ensuring the graphing method matches the expression's complexity and variable types.

The sale price estimate  $\hat{P}$  provided by the MARS model can be expressed as:

$$\begin{aligned}
\hat{P} = & f_{0,1}g_1^0 + \sum_{q=1}^{n_1} f_{1,q}g_q^1(x_*) \\
& + \sum_{r=1}^{n_2} f_{2,r}g_r^2(x_*, y_*) \\
& + \sum_{s=1}^{n_3} f_{3,s}g_s^3(x_*, y_*, z_*) \quad (15)
\end{aligned}$$

$n_1 = \#$  of first degree terms  
 $n_2 = \#$  of second degree terms  
 $n_3 = \#$  of third degree terms

## 10.5 Number of Charts Needed to Graph a Model

How many graphs are required to visualize all functions in a model? The answer depends on the variables, their interactions, and whether they are factor or non-factor variables.

The VE can limit variables and interactions in the Excel configuration file for R/earth, influencing which interactions are allowed. A practical approach is to let R/earth evaluate all possible interactions and select those with significant contributions to the sale price estimate, while checking for issues like over-fitting or collinearity. Some interactions may be disallowed, affecting higher-degree interactions (e.g., if a pair is excluded in a third-degree interaction, the entire combination is excluded).

For factor variables (e.g., MLS Area with 12 values interacting with GLA), graphing becomes more complex. This could require 12 graphs—one per area. If another factor like Architectural Style (6 values) interacts with MLS Area and GLA, up to  $12 \cdot 6 = 60$  graphs might be needed. However, R/earth rarely finds all combinations significant; typically, only a few key interactions (e.g., specific areas and styles) warrant graphing, as identified by the model.

Here's a simplified guide to the number of graphs needed:

1. **Single-variable functions (non-factor):** 1 graph.
2. **Single-variable functions (factor):** 1 bar graph combining all factor values.
3. **Base function:** No graph needed.
4. **2nd-degree interactions (no factors):** 1 graph (e.g., 2D heat map with colors for sale price).
5. **2nd-degree interactions (1 factor):** 1 graph per factor value in the model (e.g., 2 values = 2 graphs).
6. **2nd-degree interactions (2 factors):** 1 graph with a matrix of factor combinations (e.g., 3 areas  $\times$  2 styles = 6 cells).
7. **3rd-degree interactions (no factors):** Multiple graphs (e.g., 4 graphs for Bath count = 1, 2, 3, 4).
8. **3rd-degree interactions (1 factor):** 1 graph per factor value in the model.
9. **3rd-degree interactions (2 factors):** 1 graph per combination of factor values in the model.
10. **3rd-degree interactions (3 factors):** 1 graph per value of the factor with the fewest distinct values, using a color matrix for the other two.

More complex scenarios may require advanced graphing techniques.

There are other possibilities with likely more sophisticated graphs.

An example of a complete model generated by MARS is in [Figure 1](#).

Figure 1: MARS Model Components

|                        |     |                                                                                                                                                                                                                                                                  |
|------------------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Basis                  | $=$ | ${}^0g_1^0 = 4957150$                                                                                                                                                                                                                                            |
| Age                    | $=$ | ${}^0g_1^1 = -3163.1809255 \cdot \text{Age}$                                                                                                                                                                                                                     |
| AreaNbr                | $=$ | ${}^1g_2^1 = -623197.75104 \cdot I\{\text{AreaNbr} = 471\}$                                                                                                                                                                                                      |
| Baths                  | $=$ | ${}^0g_3^1 = -111023.51496 \cdot \max(0, 2 - \text{Baths}) + 96735.52418 \cdot \max(0, \text{Baths} - 2)$                                                                                                                                                        |
| Beds                   | $=$ | ${}^0g_4^1 = -233489.66415 \cdot \max(0, \text{Beds} - 5)$                                                                                                                                                                                                       |
| DaysOffMkt             | $=$ | ${}^0g_5^1 = -2577.2188622 \cdot \max(0, 189 - \text{DaysOffMkt}) - 1282.8789534 \cdot \max(0, \text{DaysOffMkt} - 189)$                                                                                                                                         |
| GLA                    | $=$ | ${}^0g_6^1 = -666.73587914 \cdot \max(0, 4027 - \text{GLA}) + 281.36312913 \cdot \max(0, \text{GLA} - 4027)$                                                                                                                                                     |
| Latitude               | $=$ | ${}^0g_7^1 = -18141680.417 \cdot \max(0, \text{Latitude} - 37.573)$                                                                                                                                                                                              |
| LotSize                | $=$ | ${}^0g_8^1 = -85.005683404 \cdot \max(0, 8000 - \text{LotSize}) - 23.562970183 \cdot \max(0, \text{LotSize} - 8000)$                                                                                                                                             |
| Age_Baths              | $=$ | ${}^0g_1^2 = -3916.3594872 \cdot \text{Age} \cdot \max(0, \text{Baths} - 3.5)$                                                                                                                                                                                   |
| Age_Garage             | $=$ | ${}^0g_2^2 = -6503.1897339 \cdot \text{Age} \cdot \max(0, \text{Garage} - 2)$                                                                                                                                                                                    |
| AreaNbr_LotSize        | $=$ | ${}^1g_3^2 = 78.54207825 \cdot I\{\text{AreaNbr} = 463\} \cdot \max(0, 8000 - \text{LotSize})$<br>$+ 87.263298012 \cdot I\{\text{AreaNbr} = 466\} \cdot \max(0, \text{LotSize} - 8000)$                                                                          |
| AreaNbr_Baths          | $=$ | ${}^1g_4^2 = -1070222.5995 \cdot I\{\text{AreaNbr} = 464\} \cdot \max(0, \text{Baths} - 2)$                                                                                                                                                                      |
| AreaNbr_Latitude       | $=$ | ${}^1g_5^2 = 17571072.548 \cdot I\{\text{AreaNbr} = 466\} \cdot \max(0, \text{Latitude} - 37.573)$                                                                                                                                                               |
| AreaNbr_GLA            | $=$ | ${}^1g_6^2 = -328.79173933 \cdot I\{\text{AreaNbr} = 471\} \cdot \max(0, \text{GLA} - 2550)$<br>$+ 525.69503229 \cdot I\{\text{AreaNbr} = 471\} \cdot \max(0, 2550 - \text{GLA})$                                                                                |
| DaysOffMkt_GLA         | $=$ | ${}^0g_7^2 = 0.43839582286 \cdot \max(0, \text{DaysOffMkt} - 661) \cdot \max(0, 4027 - \text{GLA})$                                                                                                                                                              |
| FrPlcNbr_LotSize       | $=$ | ${}^0g_8^2 = 22.059059591 \cdot \max(0, 2 - \text{FrPlcNbr}) \cdot \max(0, \text{LotSize} - 8000)$<br>$+ 39.084508159 \cdot \max(0, \text{FrPlcNbr} - 2) \cdot \max(0, \text{LotSize} - 8000)$                                                                   |
| GLA_LotSize            | $=$ | ${}^0g_9^2 = -0.45274160024 \cdot \max(0, \text{GLA} - 2994) \cdot \max(0, 8000 - \text{LotSize})$                                                                                                                                                               |
| ADU_Age_Garage         | $=$ | ${}^0g_1^3 = -5.4578633919 \cdot \max(0, \text{ADU} - 0) \cdot \text{Age} \cdot \max(0, 2 - \text{Garage})$                                                                                                                                                      |
| AreaNbr_Baths_LotSize  | $=$ | ${}^1g_2^3 = 60.607126605 \cdot I\{\text{AreaNbr} = 460\} \cdot \max(0, \text{Baths} - 2) \cdot \max(0, \text{LotSize} - 2480)$                                                                                                                                  |
| AreaNbr_Baths_FrPlcNbr | $=$ | ${}^1g_3^3 = 593779.41722 \cdot I\{\text{AreaNbr} = 464\} \cdot \max(0, \text{Baths} - 2) \cdot \max(0, \text{FrPlcNbr} - 0)$                                                                                                                                    |
| AreaNbr_GLA_Latitude   | $=$ | ${}^1g_4^3 = 54199.925713 \cdot I\{\text{AreaNbr} = 464\} \cdot \max(0, \text{GLA} - 2670) \cdot \max(0, \text{Latitude} - 37.573)$<br>$+ 58934.565747 \cdot I\{\text{AreaNbr} = 471\} \cdot \max(0, \text{GLA} - 2550) \cdot \max(0, \text{Latitude} - 37.568)$ |
| AreaNbr_DaysOffMkt_GLA | $=$ | ${}^1g_5^3 = 0.37173306643 \cdot I\{\text{AreaNbr} = 470\} \cdot \max(0, 661 - \text{DaysOffMkt}) \cdot \max(0, 4027 - \text{GLA})$                                                                                                                              |
| Beds_GLA_LotSize       | $=$ | ${}^0g_6^3 = -5.1607264026 \cdot \max(0, 4 - \text{Beds}) \cdot \max(0, \text{GLA} - 2994) \cdot \max(0, 8000 - \text{LotSize})$                                                                                                                                 |
| Age_AreaNbr_Baths      | $=$ | ${}^1g_7^3 = -10759.444688 \cdot \text{Age} \cdot I\{\text{AreaNbr} = 460\} \cdot \max(0, \text{Baths} - 3.5)$                                                                                                                                                   |

## 11 Data Characteristics

The following section discusses MARS. MARS is entirely dependent on the data it receives, and most of this data will come from some MLS (Multiple Listing Service) or tax assessors. However, it may be routed through other data providers who maintain and improve data and then sell at some profit. In some areas, it is fairly reliable; in others, it is not so reliable. In critical instances, the property inspector has to make his observations and measurements, and their accuracy can be questioned.

The valuation engineer needs to understand what the data represents and how good of a job it does representing what it is supposed to represent. This invites a whole list of issues:

- How do you classify a variable as being a measurement?
- If a variable is not a measurement, what else can it be?
- How can regression be useful if the data it receives is not accurate?

Most of the data we get from MLS listing services and tax assessor data for regression is numerical data that implies measurements or counts. Some of it is logical, True or False.

### 11.1 Purpose of the Residual

The residual serves as an indirect gauge of the value added by variables not included in the regression model. In the language of the hedonic literature, these unmeasured features are *latent variables*: variables that influence sale price but are not measured directly, in most cases because there is no practical way to measure them objectively. Their values must instead be inferred, and in practice a latent variable value is often estimated through subjective judgment — in RCA, judgment anchored by the residual ranking. By ordering properties based on their residual or residual per square foot, we achieve an objective ranking from properties with the least appeal to those with the most. This ranking method is effective, provided a skilled analyst with a deep understanding of the local market crafts the MARS (Multivariate Adaptive Regression Splines) model. The following are some additional requirements:

- **Quality of Data and Model:** A well-constructed regression model and high-quality data are prerequisites for reliable residuals. In the San Francisco Bay Area, my goal is to develop a model with an R2 of approximately 75-80% and a cross-validation R2 (CV R2) of 55-70%.
- **Avoid Over-fitting, Limit the R2:** The valuation engineer should aim for a maximum R2 of about 80% in most mature markets in higher priced areas like the SF Bay Area because much of a residential property's competitive value lies in its subjective appeal, which isn't quantifiable and thus can't be directly included in regression models. An R2 significantly above 80% might indicate an over-fitted model, prompting further scrutiny. Yet a word of caution: there are market areas with R2 values above 90% without over-fitting and there are difficult to appraise areas, where the best you might do is 50-60% or lower.
- **High R2 in Specific Areas:** However, there are regions, like subdivisions or other uniform residential areas, where higher R2 values are surprisingly common due to the homogeneity of the properties.

## 11.2 Condition and Quality Ratings as Approximate Regressors

Fannie Mae's condition and quality rating scales (C1–C6 and Q1–Q6) deserve special comment. These scales are too crude to be treated as true measured features, and because they are assigned by the appraiser, they are in practice subjective ratings — latent variables in their own right. It is useful to distinguish two levels explicitly: the appraiser's crude rating used as a regressor, *condition-coarse-grain* (condition-1) and *quality-coarse-grain* (quality-1), versus the fine-grain latent variables recovered from the residual, *condition-fine-grain* (condition-2) and *quality-fine-grain* (quality-2). The VE may accept the appraiser's coarse-grain ratings as approximations and regress on them, obtaining approximate value contributions that are useful as initial value assignments for the fine-grain latent variables. Where this is done, the protocol is to subtract the regression-estimated value contributions of these two coarse-grain variables from the model estimate — moving them out of the measured attributes under the regression section of the sales grid — and add them back onto the residual. From there, they are subsequently removed from the residual and assigned to the corresponding fine-grain latent variables, `res_condition1` and `res_quality1`. Additional information extracted from the MLS public remarks (`public_remarks`) and related string lists is then used to make micro refinements to these latent variable values. The end result is that the fine-grain value contributions replace the coarse-grain value contributions: the crude ratings serve only as scaffolding for the approximate model, and the values ultimately reported for condition and quality are the fine-grain, residual-derived ones.

## 11.3 Data Errors

Based on social media comments, the California Multiple Listing Service (MLS) data is generally of higher quality than many other United States regions. Typically, MLS data should be expected to be of good quality in most mature metropolitan areas but poorer in rural areas.

While there are occasional errors in the data for various reasons, these errors should generally cancel each other out and have minimal impact on the averages used in price models if sufficient data exists.

However, systematic biases can occur in specific areas. For instance, consider a neighborhood where many homes were initially constructed with full second stories, but some buyers were given the option to remove second-floor rooms to create vaulted ceilings on one or more first-floor rooms, significantly reducing the gross living area (GLA). The changes were not subsequently updated in the assessor's records for one reason or another. Consequently, MLS might show 2,400 square feet when the measured value is only 2,000 square feet. In such scenarios, the MLS data should be corrected before performing MARS (Multivariate Adaptive Regression Splines) regression. It is important to note that obtaining this information for comparables may require some investigative effort. Discrepancies should be noted for future reference.

Given the foregoing, it is reasonable to assume that in high-quality MLS areas, the errors that spill over into the residual are minimal compared to the impact of subjective and unmeasured features such as design, condition, and quality. This is an important consideration when estimating the accuracy of the RCA method. For example, if the  $R^2$  value for a given market area is 80%, then approximately 20% of the price is attributed to the CQA residual. A  $\pm 10\%$  error in placing the subject within the residual ranking therefore propagates to a  $\pm 10\% \cdot 20\% = \pm 2\%$  error in the final value conclusion. This is how you improve accuracy! In many cases, accurate value estimates well below  $\pm 1\%$  are possible.

Therefore, treating residuals as indirect indicators of the collective value of unmodeled variables in MARS regression is reasonable, provided that the valuation engineer has sufficient local market experience to identify where and why systemic measurement biases exist. This approach ensures

that the analysis accurately reflects actual property values, acknowledging the limitations and nuances of real estate data.

### Worked Example: Measuring Unmeasured Drivers at Sea Ranch

When the achievable  $R^2$  in a market area is lower than the market's apparent homogeneity would lead one to expect, the framework's response is to identify and measure the missing variable rather than to accept the limitation. The Sea Ranch community in Sonoma County illustrates this directly.

A concrete instance of the measure-don't-accept response. The Sea Ranch community in Sonoma County is a coastal development where distance to the ocean is a primary value driver: a home several hundred feet from the bluff commands a substantially different price from one a quarter-mile inland, holding other features constant. The MLS data for the community does not record ocean distance as a variable. An initial MARS model fit on the available MLS variables produced lower explanatory power than the market's apparent homogeneity should have supported, with a residual structure that did not sufficiently order properties by any feature visible in the data — a signal that an important driver was missing from the model rather than that the market was genuinely noisy.

The remedy was direct measurement. The author obtained a community map showing the property layout relative to the coastline and measured the ocean distance for several hundred homes. The measured distances were added as a variable to the MLS dataset and the MARS regression was rerun. The new variable was selected by the model as significant, the explanatory power increased substantially, and the residual ranking subsequently ordered properties by patterns consistent with the remaining subjective features — view, condition, quality — rather than by the latent ocean-distance signal that had previously contaminated it. Elevation has been addressed similarly in hillside neighborhoods where the MLS variable is missing or unreliable.

The methodological principle is general. When the achievable  $R^2$  falls below what the market's apparent homogeneity should support, the first hypothesis is that an important variable is missing from the model, and the appropriate response is to identify what it is and to collect the measurements directly. Sources include community maps, GIS data, tax assessor records, building department records, satellite imagery, and field inspection. The cost of such measurement work is real but bounded; once the variable is collected for a given market area, it can be carried forward into future assignments in that area. RCA absorbs new variables cleanly because the framework is indifferent to which variables enter the MARS regression — the residual structure tells the VE whether the new variable captures real signal, and the value contributions adjust accordingly.

The alternative response — accepting a low  $R^2$  and reporting wider error bands — is defensible when the missing drivers are genuinely unmeasurable (buyer-specific preferences, transient market sentiment, idiosyncratic transaction circumstances) but not when they are merely uncollected. The VE's first move on a weak model should be to ask whether the missing signal is the latter.

## 11.4 What Is Accuracy?

Perfect accuracy for a valuation requires:

1. **Conformity to Market Value:** The hypothetical sales transaction of the subject property adheres 100% to the given definition of Market Value
2. **Accurate and Robust Model Development:** A high-quality MARS model is developed with an  $R^2$  of about 75-80% and a  $CVR^2$  of roughly 55-70% depending on location.

3. **Good Data Error Analysis:** All likely data errors for measured variables are understood and contained, with little spill over into residuals.
4. **Systemic Bias Accounting:** Any systemic bias in the data should be accounted and adjusted for.
5. **Residual Rank Analysis:** Ensure that residual rank properties follow a reasonable pattern of features from lowest to highest rank. Anomalies such as probate sales, shorts, and auction sales should be investigated, explained, and potentially removed.
6. **Subject Placement:** With a thorough understanding of residual ranking, the subject property should be accurately positioned. A property with a significantly higher appeal and another with a significantly lower appeal should be identified, and the subject placed between them and scored. Then attempt to narrow the ranking between the higher and lower properties as much as possible.
7. **Explanatory Reasoning:** Never sacrifice explanatory reasoning for accuracy. The appraiser must be able to explain the differences in value between the subject and comparable properties, at least to someone with reasonable intelligence. For example, a good MARS model with an optimal R<sup>2</sup> and CVR<sup>2</sup> can be used to explain value contributions for measured features, while the breakdown of the residual by the valuation engineer can be used to explain the value contributions of other features.
8. **Over-fitting:** Avoid over-fitting models, by not using property features that can identify either individually or in combination one or several properties. In particular avoid using factor variables whose values can specify a small number of properties. A factor variable that identifies neighborhoods in sizable data sets is appropriate, but one that identifies properties with swimming pools in an area where very few swimming pools exist, must be used with caution. A feature that identifies unique properties such as exact GIS coordinates, parcel numbers or addresses, is absolutely not allowed.

If the regression R<sup>2</sup> is 80%, we expect about 20% of the subject value to be due to the residual. Now, this is only approximate. We know from the CQA-Residual curve that most of the distribution of residuals is at the lower and higher extremes of CQA. Based on the CQA-Residual curve and the extreme slope for the low and high CQA values, we might expect a less accurate estimate of the residual in these areas. However, that is counterbalanced by the relative scarcity of properties in these areas and the greater difference in appeal we see between adjacent properties. So, while we might talk of the impact of a range of  $\pm 5\%$  in the mid-range, at the extremes we would be more concerned with a somewhat narrower range of  $\pm 2 - \pm 3\%$

Calculating the accuracy of the RCA is, in general, somewhat difficult because of the non-linearity of the CQA-Residual curve. But it is possible on a case by case basis, given you have a reasonable residual ranking. The range of CQA scores is 0.0 to 10.0. With a good MARS model, your residual ranking should show a clear pattern of least to highest appeal and you should not find it difficult to rate within  $\pm 5\%$  of the total range or  $\pm 0.5$  for the CQA score. If you have 100 properties in the regression, then that would be ranking the subject property accurately within a group of 10 properties. So, referring to the Burlingame sales data in Table 3, let's calculate the approximate accuracy (or error) for a VE estimated CQA of 7.5, 5.1 and 1.9, where his value conclusion for the subject is \$3,000,000. Note that since Table 3 only has a sample of the several hundred properties used, you will need to interpolate the residual where necessary (assuming linearity). To be clear, we are assuming the VE is very sure in each case that his ranking of the subject is within a range of  $\pm 5$  properties in the residual ranking of only 100 properties.

|               |           |           |            |                 |           |
|---------------|-----------|-----------|------------|-----------------|-----------|
| CQA= 7.5 ±0.5 |           |           |            |                 |           |
| CQA           | Residual  | Abs. Diff | Exp. Error | Hyp. Sale Price | Pct Error |
| 8.00          | \$257,112 |           |            |                 |           |
| 7.50          | \$197,113 | \$59,999  |            |                 |           |
| 7.00          | \$154,615 | \$42,498  | ±\$51,249  | \$3,000,000     | ±1.71%    |

|              |           |           |            |                 |           |
|--------------|-----------|-----------|------------|-----------------|-----------|
| CQA=5.1 ±0.5 |           |           |            |                 |           |
| CQA          | Residual  | Abs. Diff | Exp. Error | Hyp. Sale Price | Pct Error |
| 5.60         | \$50,911  |           |            |                 |           |
| 5.10         | \$19,559  | \$31,352  |            |                 |           |
| 4.60         | -\$28,207 | \$47,766  | ±\$39,559  | \$3,000,000     | ±1.32%    |

|               |            |           |            |                 |           |
|---------------|------------|-----------|------------|-----------------|-----------|
| CQA= 1.9 ±0.5 |            |           |            |                 |           |
| CQA           | Residual   | Abs. Diff | Exp. Error | Hyp. Sale Price | Pct Error |
| 2.40          | -\$228,411 |           |            |                 |           |
| 1.90          | -\$276,381 | \$47,970  |            |                 |           |
| 1.40          | -\$326,938 | \$50,557  | ±\$49,264  | \$3,000,000     | ±1.64%    |

Finally, the only objective way to verify accuracy is to establish a strict protocol (which this paper only partially hints at in broad strokes) and verify its adherence. This is necessary because actual sale prices can deviate from market value due to imperfections in the market. Comparing appraisals and subsequent sales can provide some indication of accuracy. However, the complexities and instability of the real estate market necessitate the expertise of appraisers or highly qualified valuation engineers to provide value estimates based on established standards and protocols. These should be stringent enough to ensure that competent valuation engineers will independently arrive at nearly the same value conclusion. This should be an achievable goal, in most cases where good data is available.

### 11.5 The Subject Property As An Anomaly

Since estimating the Market Value of a property requires analysis of recent sales transactions, it is fair to ask if recent sales transactions do a good job of representing the various features of an older subject property that lacks modernization and repairs. This is particularly true in the San Francisco Bay Area, where recent sales transactions typically are homes that have gone through updating to improve their price for sale. So, the average quality and condition of sales comparables are generally superior to older homes that have not been on the market for quite some time. This difference in value should be picked up by the estimate of the residual for the subject property, or in other words, it's CQA score. This is an important consideration with respect to data quality and accuracy in valuing the subject property.

### 11.6 The CVR2's Importance In Dealing With Unexpected Data

MARS regression is based on multiple iterations of R/earth applied to a randomly selected subset of properties, which typically comprises 80% to 90% of the MLS data set referred to as the training set. The remaining 10% - 20% of properties serve as the test set. In this phase, the MARS model is created from the training data set and then is used to predict the sale prices of properties in the test data set, which are then compared to the actual sale prices. This methodology has been recognized by data miners as the most effective strategy to develop a robust model capable of predicting the sale prices of properties not yet observed.

However, the primary utility of the CVR2 metric lies not in its reflection of the regression model's factual accuracy concerning actual market area sales, but rather in its optimization of the model's predictive capabilities regarding hypothetical property sales as of a specified effective date. These properties are not included in the training data set and may exhibit new or unexpected combinations of characteristics missing in both the training and test data set property data.

Given this context, how can we estimate the sale price of a property that hasn't been updated in the past 15 years, assuming it was listed, for example, one month prior to the effective appraisal date? A model that demonstrates a strong CVR2 is more likely to provide the most accurate estimate, rather than a model that has not gone through this repetitive cross validation. It's also vital that the data set includes a sufficient number of comparable properties with varying degrees of updates, repairs and modernization.

Ultimately, if we possess a robust model that accurately predicts the value contributions of various properties features in our data set of comparable properties, we can effectively place the new subject property within the model's ranking of these properties by residual. This placement allows us to assign an appropriate CQA score and residual to the property.

## 12 Residuals

As indicated by the name “Residual Constraint Approach,” residuals are central to this methodology. Here, residual refers to the difference between a property’s net sale price and the sale price predicted by regression analysis.

A positive residual suggests that the property sold for more than its estimated price, based on typical measurable attributes like gross living area (GLA) and lot size, implying it might have more appeal than average. A negative residual indicates the property sold for less, potentially suggesting less appeal or other uncaptured negative attributes.

Given that larger homes likely have larger residuals, we might use residual per square foot based on GLA for a more normalized comparison. A more in-depth discussion of this issue is reserved for another paper.

### 12.1 Ranking & Scoring

After creating the MARS (Multivariate Adaptive Regression Splines) model for measured variables, residuals are computed for each property. These residuals represent the difference between the observed net sale prices and the sale prices predicted by the model. An additional column should be created for the Residual/SF, that is, the residual divided by the living area of the residence, with the understanding that generally, the residual for specific properties is more significant with the size of the living area. Other possible statistics may work better in some neighborhoods, but they are a subject for another time.

#### **Assumption 7: “Ranking by Residual”**

In this paper, references to “ranking by residual” should be taken to mean ranking by residual, by residual/SF, or ranking by some other function of residuals, as the valuation engineer deems most appropriate for the market area. The author’s experience in SF Bay Area indicates residual/SF is generally the most reliable, especially when dealing with a subject property that has an unusually small or large GLA (Gross Living Area).

Next, the properties are sorted by their residuals from the most negative to the most positive. A Condition-Quality-Appeal (CQA) score is then generated for each property, ranging from 0.0 to 10.0. This score is calculated as follows:

The CQA score corresponds to the percentile rank of a property’s residual among all properties. Specifically, the score is computed by determining the percentage of properties that have residuals lower than that of the given property and dividing that percentage by 10. For instance, if a property’s residual is higher than the residuals of 50% of properties, then its score would be 5.0.

Due to rounding, one property can receive a score of 10.0, representing the property with the highest residual. However, a perfect score of 10.0 is theoretically impossible because, logically, no property can have a residual higher than that of 100% of the properties, itself included. On the other hand, the lowest possible score of 0.0 is possible since at least one property has no other properties with lower residuals. Note that more than one property might have a score of 0.0 due to rounding if many properties are being compared.

Therefore, the CQA score provides an intuitive measure of how well a property’s attributes align with the model’s expectations, with higher scores indicating better alignment with the model’s predictions.

It is essential to acknowledge that the CQA score can only be considered an indirect indicator of a comparable property’s appeal when the sale conditions are commensurate with its market

value. The Valuation Engineer is responsible for ascertaining whether the residual value incorporates over- or under-market elements by providing a comprehensive explanation and thorough investigation of the sales transaction. While significant over- and under-market sales are not frequent, they can occur in circumstances such as forced sales, probate, incompetence, collusion, etc.

Table 3 presents a semi-realistic extract of MLS data, along with residuals and CQA scores, both in total and per square foot. The total number of properties analyzed in this study was approximately 600. It is noteworthy that several variables that were regressed are not displayed. In other words, the residuals presented are after the impact of other variables, such as the date of sale, has been considered by MARS regression.

Table 3: Sales Comparables Sorted By Residual

| SalePrice   | Estimated SP | Residual     | CQA | Residual/SF | CQA SF | GLA   | SaleDate |
|-------------|--------------|--------------|-----|-------------|--------|-------|----------|
| \$3,450,000 | \$2,318,865  | \$1,131,135  | 10  | \$420.50    | 9.98   | 2,690 | 2021-04  |
| \$4,360,757 | \$3,399,922  | \$960,835    | 9.9 | \$290.28    | 9.82   | 3,310 | 2021-09  |
| \$4,000,000 | \$3,134,444  | \$865,556    | 9.8 | \$234.57    | 9.60   | 3,690 | 2020-11  |
| \$5,850,000 | \$5,150,966  | \$699,034    | 9.7 | \$109.91    | 8.35   | 6,360 | 2021-12  |
| \$3,600,000 | \$3,023,366  | \$576,634    | 9.5 | \$269.46    | 9.79   | 2,140 | 2022-06  |
| \$4,425,000 | \$3,987,873  | \$437,127    | 9.2 | \$127.44    | 8.68   | 3,430 | 2021-08  |
| \$2,530,000 | \$2,164,564  | \$365,436    | 8.9 | \$180.02    | 9.28   | 2,030 | 2019-12  |
| \$4,250,000 | \$3,884,719  | \$365,281    | 8.8 | \$87.18     | 7.91   | 4,190 | 2021-08  |
| \$3,350,000 | \$3,034,613  | \$315,387    | 8.5 | \$130.87    | 8.75   | 2,410 | 2021-09  |
| \$3,800,000 | \$3,522,028  | \$277,972    | 8.2 | \$77.43     | 7.72   | 3,590 | 2020-09  |
| \$4,200,000 | \$3,963,748  | \$236,252    | 7.8 | \$56.12     | 6.89   | 4,210 | 2021-01  |
| \$2,900,000 | \$2,702,887  | \$197,113    | 7.5 | \$89.19     | 7.92   | 2,210 | 2021-07  |
| \$4,250,000 | \$4,103,885  | \$146,115    | 6.9 | \$39.38     | 6.29   | 3,710 | 2022-08  |
| \$5,000,000 | \$4,864,958  | \$135,042    | 6.8 | \$28.25     | 5.85   | 4,780 | 2021-07  |
| \$4,320,000 | \$4,199,672  | \$120,328    | 6.5 | \$33.90     | 6.04   | 3,550 | 2022-07  |
| \$2,600,000 | \$2,479,904  | \$120,096    | 6.5 | \$60.96     | 7.00   | 1,970 | 2021-10  |
| \$3,400,000 | \$3,301,146  | \$98,854     | 6.2 | \$30.32     | 5.90   | 3,260 | 2021-06  |
| \$2,000,000 | \$1,901,987  | \$98,013     | 6.2 | \$73.14     | 7.54   | 1,340 | 2019-11  |
| \$3,000,000 | \$2,919,718  | \$80,282     | 5.9 | \$26.50     | 5.79   | 3,030 | 2020-05  |
| \$6,750,000 | \$6,699,089  | \$50,911     | 5.6 | \$7.75      | 5.20   | 6,570 | 2020-08  |
| \$5,652,000 | \$5,601,570  | \$50,430     | 5.5 | \$12.83     | 5.37   | 3,930 | 2022-05  |
| \$2,600,000 | \$2,579,082  | \$20,918     | 5.2 | \$10.51     | 5.33   | 1,990 | 2022-03  |
| \$1,925,000 | \$1,905,441  | \$19,559     | 5.1 | \$11.51     | 5.36   | 1,700 | 2021-01  |
| \$3,200,000 | \$3,209,500  | -\$9,500     | 4.8 | -\$3.21     | 4.88   | 2,960 | 2020-11  |
| \$4,150,000 | \$4,187,560  | -\$37,560    | 4.5 | -\$10.49    | 4.68   | 3,580 | 2021-04  |
| \$2,700,000 | \$2,776,510  | -\$76,510    | 3.9 | -\$23.61    | 4.14   | 3,240 | 2021-07  |
| \$3,450,000 | \$3,567,668  | -\$117,668   | 3.4 | -\$31.05    | 3.90   | 3,790 | 2020-09  |
| \$1,975,000 | \$2,109,255  | -\$134,255   | 3.3 | -\$55.48    | 3.24   | 2,420 | 2020-02  |
| \$1,950,000 | \$2,085,431  | -\$135,431   | 3.2 | -\$68.40    | 2.86   | 1,980 | 2021-02  |
| \$3,695,000 | \$3,859,154  | -\$164,154   | 3.1 | -\$47.04    | 3.56   | 3,490 | 2021-05  |
| \$2,288,000 | \$2,455,747  | -\$167,747   | 2.9 | -\$56.86    | 3.19   | 2,950 | 2020-06  |
| \$2,635,000 | \$2,802,781  | -\$167,781   | 2.9 | -\$56.87    | 3.18   | 2,950 | 2020-09  |
| \$2,050,000 | \$2,217,794  | -\$167,794   | 2.9 | -\$82.66    | 2.46   | 2,030 | 2021-07  |
| \$1,550,000 | \$1,731,804  | -\$181,804   | 2.7 | -\$130.79   | 1.32   | 1,390 | 2021-05  |
| \$2,800,000 | \$3,028,411  | -\$228,411   | 2.4 | -\$81.58    | 2.51   | 2,800 | 2021-02  |
| \$3,700,000 | \$3,976,381  | -\$276,381   | 1.9 | -\$76.99    | 2.62   | 3,590 | 2021-08  |
| \$2,517,500 | \$2,834,974  | -\$317,474   | 1.5 | -\$120.71   | 1.48   | 2,630 | 2019-11  |
| \$2,550,000 | \$2,905,330  | -\$355,330   | 1.1 | -\$115.74   | 1.64   | 3,070 | 2019-12  |
| \$1,815,000 | \$2,263,944  | -\$448,944   | 0.7 | -\$273.75   | 0.15   | 1,640 | 2021-11  |
| \$3,711,000 | \$4,683,454  | -\$972,454   | 0   | -\$226.68   | 0.37   | 4,290 | 2021-12  |
| \$2,225,000 | \$3,332,960  | -\$1,107,960 | 0   | -\$370.56   | 0.03   | 2,990 | 2021-08  |
| \$3,550,000 | \$4,746,489  | -\$1,196,489 | 0   | -\$274.42   | 0.14   | 4,360 | 2022-06  |

## 12.2 CQA-Residual Curve

A plot of the CQA vs. the Residual/SF for a neighborhood will invariably result in the CQA-Residual Characteristic Curve shown in [Figure 2](#). This particular plot represents data from Burlingame, CA, circa 2022. Similar curves are generally observed when plotting CQA scores against Residual values; however, I prefer using Residual/SF for its relevance in contextualizing the CQA with respect to the Gross Living Area (GLA) of the subject property. By multiplying the Residual/SF by the GLA, one can convert the CQA of a property into a corresponding residual value.

The valuation engineer must carefully evaluate whether the size of the property, specifically the Gross Living Area (GLA), significantly influences the magnitude of residuals. This assessment is critical because it can determine the most accurate method for calculating property values. For instance, high-end kitchen appliances and luxurious bathroom fixtures can substantially enhance a home's residual value. However, in neighborhoods where the size and layout of kitchens and bathrooms are generally uniform, these upgrades might not correlate strongly with the GLA. Their impact on the residual value may be less about the space they occupy and more about the quality they represent.

Conversely, expensive exterior features and materials, which notably improve a property's curb appeal, may have a more direct relationship with the property's size. Larger homes can accommodate more elaborate exteriors, such as extensive landscaping, superior roofing materials, or expansive patios, which naturally enhance the property's overall appeal and, consequently, its residual value.

Therefore, it is essential for valuation engineers to consider these dynamics when analyzing residuals. They should not only assess the intrinsic value added by high-quality fixtures and features but also understand how these enhancements interact with the property's physical dimensions. This comprehensive approach ensures that valuations accurately reflect both the qualitative and quantitative attributes of a property.

Consequently, it is recommended to plot both the CQA-Residual and CQA-Residual/SF foot curves and study the differences, which can indicate patterns in a market area. The valuation engineer can have his program compute both residuals and take some weighted average to calculate the residual to be assigned to the subject. It may or may not make much of a difference.

## 12.3 The CQA Curve Characteristics

The Condition-Quality-Appeal (CQA) curve offers a sophisticated model for understanding how property values fluctuate based on appeal, condition, and market dynamics within a given area. This model captures the interplay of economic and physical factors that influence the real estate market, reflecting nuanced realities of property valuation across different market segments.

### 12.3.1 Analysis of the CQA Curve Components

1. Exponential Decay on the Left Side of the Curve:

As the CQA score decreases from around 2 to 0, properties exhibit characteristics of rapid deterioration. This part of the curve is well-represented by an exponential decay function, emphasizing how quickly a property can lose its appeal and value without proper maintenance and updates. Physical degradation such as rotting wood, fading colors, and rusting metal accelerates this decline, particularly in homes that are not regularly maintained.

2. Linear Increase in the Middle Range:

The bulk of the market typically fits within this category, where the appeal increases almost linearly from scores of around 2 to 8. This section reflects properties that are generally maintained to standard but don't necessarily feature exceptional qualities or locations. It represents the broad middle class of housing, encompassing the majority of the housing market where homes are competitively priced based on their standard features and overall condition.

### 3. Exponential Growth on the Right Side of the Curve:

Here, the curve rises exponentially from scores 8 to 10, mirroring the higher appeal end of the market. Properties in this segment often belong to wealthier individuals who not only invest in superior maintenance but also in enhancements that significantly boost property appeal and value. The exponential growth reflects the scarcity of such high-quality properties and the high demand among affluent buyers. And understand, a lower priced home may rate very high in terms of appeal, while a large and expensive home may rate low. These dynamics are uncovered by expert use of MARS and usually go unnoticed by most appraisers.

## 12.3.2 Market Dynamics and Social Implications

The CQA curve implicitly narrates the story of relative wealth and economic disparity within different regions. In places like San Mateo County, the variation in property values between areas like Burlingame and wealthier neighborhoods such as Hillsborough, Woodside, or Atherton illustrates significant socioeconomic divides. This disparity affects everything from property maintenance to market valuations and ultimately influences buyer demographics and real estate market trends.

Moreover, the anecdote about a relatively wealthy individual investing in a high-quality home within a mid-tier neighborhood they are comfortable with and where for various reasons they would prefer to live, underscores another critical aspect of real estate economics — the concept of “overbuilt” properties. Such properties, while very appealing, may not find local buyers able to pay the price they might have fetched in a higher priced market area.

The CQA-Residual curve is characteristic to each market area, and should not be overlooked when doing your analysis.

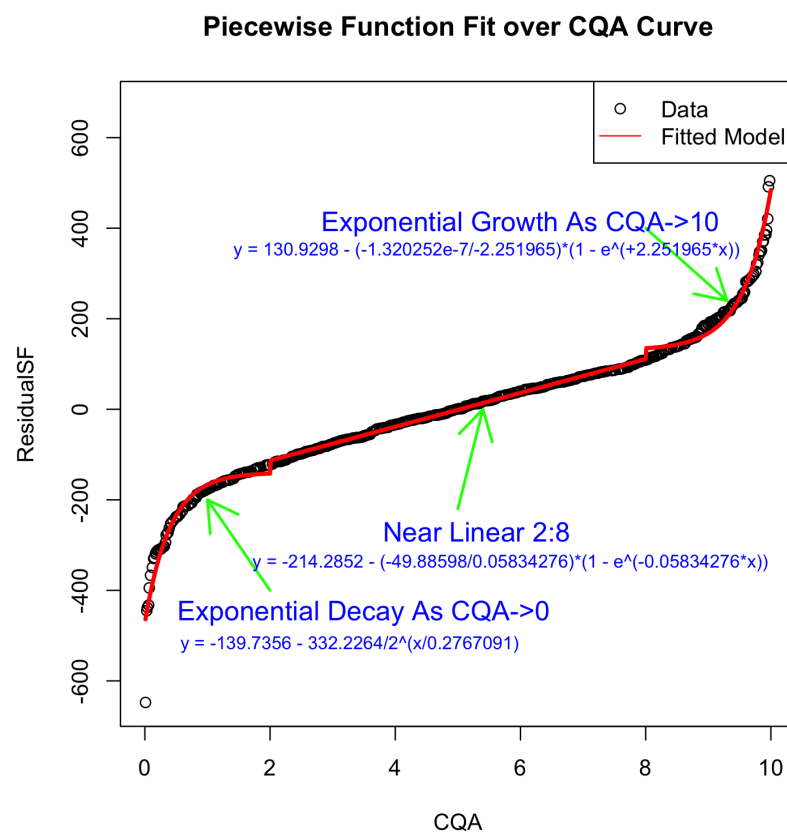
## 12.4 Subject Residual

Unlike the comparables, the subject property does not possess an objectively determined residual; it must be estimated by the Valuation Engineer (VE). This estimation introduces a degree of subjectivity into the Residual Constraint Approach (RCA) protocol, marking a critical juncture where personal judgment plays a pivotal role.

To estimate the Condition-Quality-Appeal (CQA) score, the VE meticulously analyzes the ranked properties, searching for discernible patterns within the rankings. This analysis often involves a detailed review of MLS photos associated with each property. An exhaustive review is impractical for a large dataset, such as 600 properties. Instead, the VE applies practical reasoning to approximate the subject's likely position within the overall ranking. Properties requiring extensive repairs, or “fixers,” are generally positioned near the bottom, whereas those with superior updates and appeal tend to rank near the top.

The VE then adopts a more focused approach, ‘jumping’ within the ranking to identify comparables that closely match the subject's appeal. As the range narrows, the VE pays closer attention to similarities among the properties, particularly in aspects like quality, condition,

Figure 2: CQA-Residual Characteristic Function



design aesthetics (including elements like paint, woodwork, stonework, and windows), functional utility, and other relevant characteristics

It is crucial to understand that the VE's judgments are fundamentally guided by the market-established rankings. This ensures that the subjective elements of the process are anchored by market realities, striving for a fair and accurate valuation of the subject property.

## **12.5 RCA Value Conclusion**

### **12.5.1 General Considerations**

Determining a CQA score is pivotal for establishing the estimated residual value of a property, as these elements are intrinsically linked. While it is possible to assess a property without a CQA, this metric provides valuable context by illustrating where the property stands compared to others in the market area. For instance, a CQA score of 3.2 indicates that the property has greater appeal than 32% of comparable properties in the vicinity. If objections exist to the assigned score, disputants must consult the rankings and justify any proposed adjustments based on photographic evidence or other substantial data. If the ranking system is robust and the Valuation Engineer (VE) has accurately positioned the property within this framework, it is unlikely that significant discrepancies will arise.

In the reporting phase, typically, 6 to 12 recent and relevant sales comparables are selected to support the valuation conclusion. The differences between these comparables and the subject property are analyzed, with each variable being adjusted accordingly. These adjustments are then applied to the net sale prices of the comparables to derive adjusted sale prices. As demonstrated mathematically, these adjusted prices should converge with the estimated sale price of the subject property.

### **12.5.2 Review, Auditing, Complaints**

Once the estimated residual for the subject property is determined, residual adjustments can be calculated for all comparables. These adjustments serve as constraints for further manual breakdowns by the VE, tailored to specific CQA variables deemed pertinent. It is important to emphasize that these detailed breakdowns are explanatory and do not influence the final valuation, as affirmed in Proof II.

Concerning complaints submitted by users not satisfied with the appraisal value conclusion, a traditional appraiser typically has to waste time explaining why this or that property with a higher or lower price does not have the user-expected impact on value. With RCA, all possible comparable sales (usually 100-600) are typically evaluated to precisely the same value as the price conclusion, so it takes little effort to show the adjustments for some arbitrary property. The entire RCA process is automated and objective, except for the scoring of the subject residual. And so another advantage of the RCA is that criticism for imperfections in the subjective judgment of the VE is mainly restricted to that one area - and it can be effectively and efficiently managed. Also, assuming an  $R^2$  of about 80%, the impact of a  $\pm 10\%$  error in estimating the subject residual amounts to a  $\pm 2\%$  error in the final value estimate.

In many cases, the subject fits so perfectly in the residual ranking of properties, that the accuracy can be expected to be within  $\pm 1\%$ .

The sum of the estimated residual and the subject's MARS estimated sale price yields the final value conclusion known as the Residual Constraint Approach (RCA) Value Conclusion. This process ensures a comprehensive and transparent approach to property valuation, fostering

confidence in the accuracy and fairness of the assessment.

## 13 Conclusion

The Residual Constraint Approach is a framework with three substantive contributions: a structural anti-anchoring property derived from fitting the model on the whole market rather than on a small comparable set (Proof I); an invariance result that allows residual breakdown to serve as an explanatory layer without disturbing the value conclusion (Proof II and the remark following it); and an empirical regularity — the CQA-Residual characteristic curve — that appears to recur across market areas in characteristic exponential-linear-exponential form. The framework’s limitations are equally substantive: the residual decomposition is invariant but not identifying, the CQA placement step relies on VE judgment that has not yet been validated through formal inter-rater study, and the accuracy claims await benchmarking against alternatives.

What this paper presents is a framework and a protocol. What it does not yet present, but the larger research program now in progress does, is an executable, audit-traceable pipeline that operationalizes the protocol in software with defensible defaults. That pipeline — the earthUI package on CRAN for the MARS basis-function stage, the glmnetUI package for the regularized structural model on those bases, and the mgcvUI package for spatial smoothing — is the subject of companion articles in this issue and is the bridge from RCA-as-method to RCA-as-deliverable. The doctrine-practice gap in appraisal regulation (Craytor, 2026, this issue) supplies the reason that bridge matters: appraisal outputs are economically consumed as forward-looking risk inputs, and a framework that makes the present-value opinion auditable and reproducible is a precondition for any honest engagement with that forward-looking use. RCA is one move in that larger project. It is not the whole project, and this paper does not claim it is.

## References

- Ambrose, Brent W. et al. (2025). “Do Appraiser and Borrower Race Affect Mortgage Collateral Valuation?” In: *Review of Finance*. DOI: [10.1093/rof/rfaf046](https://doi.org/10.1093/rof/rfaf046).
- Craytor, William Bert (2026). “The Doctrine–Practice Gap in Real Estate Appraisal: A Structured Account of Functions, Boundaries, and Tensions”. In: *Valuation Engineer Journal* 1.1.
- Friedman, Jerome H. (1991). “Multivariate Adaptive Regression Splines”. In: *The Annals of Statistics* 19.1, pp. 1–141.
- Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman (2017). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second. Springer Series in Statistics. New York, NY: Springer. ISBN: 978-0-387-84857-0.
- James, Gareth et al. (2013). *An Introduction to Statistical Learning: With Applications in R*. Springer Texts in Statistics 103. New York: Springer. ISBN: 978-1-4614-7137-0.
- Krivorotov, George and Michael LaCour-Little (2020). “AVM versus Appraisal-Based Underwriting in Refinance Mortgages: The Trade-off Between Noise and Bias”. In: *SSRN Electronic Journal*. DOI: [10.2139/ssrn.3603327](https://doi.org/10.2139/ssrn.3603327).
- Kuhn, Max and Kjell Johnson (2016). *Applied Predictive Modeling*. Corrected at 5th printing. New York: Springer. ISBN: 978-1-4614-6848-6.
- Milborrow, Stephen (2024a). *Earth: Multivariate Adaptive Regression Splines*. R Package. URL: <https://CRAN.R-project.org/package=earth>.
- (2024b). *Notes on the Earth Package*. Tech. rep. URL: <http://www.milbo.org/doc/earth-notes.pdf>.
- (2024c). *Variance Models in Earth*. Tech. rep. URL: <http://www.milbo.org/doc/earth-varmod.pdf>.

- Perry, Andre M., Jonathan Rothwell, and David Harshbarger (2018). *The Devaluation of Assets in Black Neighborhoods*. Tech. rep. Brookings Institution.
- Perry, Andre M., Jonathan Rothwell, J. Williamson, et al. (2024). *How Racial Bias in Appraisals Affects the Devaluation of Homes in Majority-Black Neighborhoods*. Tech. rep. Brookings Institution.
- Property Appraisal and Valuation Equity Task Force (2022). *Action Plan to Advance Property Appraisal and Valuation Equity*. Tech. rep. U.S. Department of Housing and Urban Development.
- Tzioumis, Konstantinos (2016). “Appraisers and Valuation Bias: An Empirical Analysis”. In: *Real Estate Economics* 45.3, pp. 679–712. DOI: [10.1111/1540-6229.12133](https://doi.org/10.1111/1540-6229.12133).
- Williamson, Jake and Mark Palim (2022). *Appraising the Appraisal: A Closer Look at Divergent Appraisal Values for Black and White Borrowers Refinancing Their Home*. Tech. rep. Fannie Mae.
- Zhang, Wengang (2020). *MARS Applications in Geotechnical Engineering Systems: Multi-Dimension with Big Data*. Singapore: Springer. ISBN: 978-981-13-7421-0.

# earthUI: An Interactive Interface for the earth (MARS) Package

William Bert Craytor

Vol. 1, Issue 1 · Summer 2026

Published July 1, 2026

DOI: [10.67330/7fj41b65](https://doi.org/10.67330/7fj41b65)

© 2026 William Bert Craytor. Published by Pacific Vista Net.

This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

**Abstract.** earthUI is a graphical user interface for the R `earth` package, which implements Multivariate Adaptive Regression Splines (MARS). It offers three purpose modes—general predictive modeling, real-estate appraisal, and market-area analysis—and guides the user through data import, model configuration, fitting, and the interpretation of earth’s diagnostics and graphical output. This article documents earthUI’s data-format requirements, modeling workflow, output displays, and complete feature reference.

**Keywords:** MARS, multivariate adaptive regression splines, earth, R, graphical user interface, real estate valuation

**Publisher’s Note (July 2026):** This article describes earthUI 0.10.0. Shortly before press, earthUI 0.11.0 absorbed the companion applications `glmnetUI` and `mgcvUI` (described in the following two articles), consolidating the MARS, elastic-net, and GAM workflows in a single application. The interface and workflow described here are unchanged by the consolidation.

## Contents

|          |                                     |           |
|----------|-------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                 | <b>3</b>  |
| 1.1      | What Is earthUI?                    | 3         |
| 1.2      | One Package, Three Routines         | 4         |
| 1.3      | Three Purpose Modes                 | 4         |
| 1.4      | Real Estate–Specific Features       | 5         |
| 1.5      | Installing earthUI and vProlog      | 5         |
| 1.6      | Getting Started                     | 6         |
| <b>2</b> | <b>MLS Input Data Requirements</b>  | <b>7</b>  |
| 2.1      | File Format & Structure             | 7         |
| 2.2      | Required Columns for Appraisal Mode | 7         |
| 2.3      | Special Column Naming Conventions   | 8         |
| 2.4      | Data Quality & Completeness         | 8         |
| 2.5      | Subject Row Placement               | 8         |
| <b>3</b> | <b>General Purpose Mode</b>         | <b>10</b> |
| 3.1      | Overview                            | 10        |
| 3.2      | The Sidebar Workflow                | 10        |
| 3.3      | Main Panel Tabs                     | 10        |
| 3.4      | Settings Persistence                | 11        |
| 3.5      | Dark Mode                           | 11        |
| 3.6      | Locale & Regional Settings          | 11        |
| 3.6.1    | Supported Countries                 | 12        |
| 3.6.2    | Paper Override                      | 12        |
| 3.6.3    | Saving Defaults                     | 12        |

|          |                                                 |           |
|----------|-------------------------------------------------|-----------|
| <b>4</b> | <b>Appraisal Mode</b>                           | <b>13</b> |
| 4.1      | Subject Row Handling                            | 13        |
| 4.2      | Effective Date & Sale Age                       | 13        |
| 4.3      | Special Column Designations                     | 13        |
| 4.4      | Residual Contribution (Rare-Feature Estimation) | 14        |
| 4.5      | RCA Adjustments Overview                        | 14        |
| <b>5</b> | <b>Market Area Analysis Mode</b>                | <b>16</b> |
| 5.1      | Differences from Appraisal Mode                 | 16        |
| 5.2      | When to Use Market Mode                         | 16        |
| <b>6</b> | <b>Variable Selection</b>                       | <b>17</b> |
| 6.1      | Target Variable(s)                              | 17        |
| 6.2      | The Predictor Table                             | 17        |
| 6.3      | Data Type Detection & Overrides                 | 17        |
| 6.4      | Factor and Linear Flags                         | 18        |
| 6.5      | Special Column Types Reference                  | 18        |
| 6.6      | Multiple Targets                                | 19        |
| <b>7</b> | <b>Parameter Selection</b>                      | <b>20</b> |
| 7.1      | Basic Parameters                                | 20        |
| 7.2      | Forward Pass                                    | 20        |
| 7.3      | Allowed Interactions                            | 21        |
| 7.4      | Pruning                                         | 21        |
| 7.5      | Cross-Validation                                | 21        |
| 7.6      | Subset Filtering                                | 22        |
| 7.7      | Recommended Parameter Values                    | 22        |
| 7.7.1    | Forward Pass Parameters                         | 22        |
| 7.7.2    | Pruning Parameters                              | 23        |
| 7.7.3    | Cross-Validation Parameters                     | 23        |
| 7.7.4    | Variance Model                                  | 24        |
| 7.7.5    | Earth Output File                               | 24        |
| 7.8      | Settings Defaults                               | 24        |
| <b>8</b> | <b>Fitting the Earth Model</b>                  | <b>26</b> |
| 8.1      | The Fit Button                                  | 26        |
| 8.2      | Fitting Modal & Trace Output                    | 26        |
| 8.3      | Fit Log                                         | 26        |
| <b>9</b> | <b>Downloading Data</b>                         | <b>27</b> |
| 9.1      | Output Columns                                  | 27        |
| 9.2      | Column Ordering & Excel Formatting              | 27        |
| 9.3      | CQA Scores                                      | 28        |
| 9.4      | Sorting & Subject Row                           | 28        |
| 9.5      | mgcvUI Auto-Export                              | 28        |
| 9.6      | glmnetUI Import                                 | 28        |
| 9.6.1    | Key Benefits                                    | 29        |
| 9.6.2    | Important: Weight Column Must Match             | 29        |
| 9.6.3    | Workflow                                        | 29        |

|                                                                                |           |
|--------------------------------------------------------------------------------|-----------|
| <b>10 RCA Calculations &amp; Downloading</b>                                   | <b>30</b> |
| 10.1 Opening the RCA Dialog . . . . .                                          | 30        |
| 10.2 CQA Score Interpolation . . . . .                                         | 30        |
| 10.3 Output Columns . . . . .                                                  | 30        |
| 10.4 The RCA Adjustments Tab . . . . .                                         | 31        |
| <b>11 Sales Comparison Grid</b>                                                | <b>32</b> |
| 11.1 Overview & Purpose . . . . .                                              | 32        |
| 11.2 Comp Selection Modal . . . . .                                            | 32        |
| 11.3 Grid Layout . . . . .                                                     | 32        |
| 11.4 Sale Price, Concessions & Net SP . . . . .                                | 33        |
| 11.5 Grouped Rows (Location, Site, Age) . . . . .                              | 33        |
| 11.6 Model Variable Rows . . . . .                                             | 33        |
| 11.7 CQA Residual & Residual Feature Rows . . . . .                            | 33        |
| 11.8 Adjusted Sale Price Formula . . . . .                                     | 34        |
| 11.9 Sheet Protection & Editable Cells . . . . .                               | 34        |
| 11.10 Working with the Grid in Excel . . . . .                                 | 34        |
| <b>12 Downloading Reports</b>                                                  | <b>35</b> |
| 12.1 Generate Quarto Report . . . . .                                          | 35        |
| 12.2 Convert Quarto Report . . . . .                                           | 35        |
| 12.3 Report Contents . . . . .                                                 | 35        |
| <b>13 The Post Processing Modules: Elastic Net and GAM</b>                     | <b>37</b> |
| 13.1 Why Support Fits? . . . . .                                               | 37        |
| 13.2 Launching and Shared State . . . . .                                      | 37        |
| 13.3 The Elastic Net Routine . . . . .                                         | 37        |
| 13.4 The GAM Routine . . . . .                                                 | 37        |
| 13.5 A Practical Workflow . . . . .                                            | 37        |
| <b>14 Demo Dataset: Appraisal_1.csv</b>                                        | <b>39</b> |
| 14.1 Description . . . . .                                                     | 39        |
| 14.2 Columns . . . . .                                                         | 40        |
| 14.3 Suggested Quick Start . . . . .                                           | 40        |
| <b>15 Experimental: Prolog Text Extraction &amp; Derivation Rules</b>          | <b>42</b> |
| 15.1 What It Does . . . . .                                                    | 42        |
| 15.2 Requirements: vProlog (Trealla), Not SWI-Prolog . . . . .                 | 42        |
| 15.3 Enabling and Running It . . . . .                                         | 43        |
| 15.4 Derivation Rules . . . . .                                                | 43        |
| 15.5 Versioned Active Data File . . . . .                                      | 43        |
| 15.6 Programmatic Interface & Bundled Files . . . . .                          | 44        |
| 15.7 Related Experiment: Climate-Region Priors . . . . .                       | 44        |
| 15.7.1 Beyond California: Jurisdiction Packs and Market-Area Modules . . . . . | 44        |

## 1 Introduction

### 1.1 What Is earthUI?

earthUI is a graphical user interface for the R **earth** package, which implements Multivariate Adaptive Regression Splines (MARS) (Friedman 1991). The modelling engine is Stephen

Milborrow’s **earth** package (Milborrow 2024), documented in detail by Milborrow (n.d.[a]) and Milborrow (n.d.[b]). It runs as a local Shiny application — there is no login, no server, and no accounts. You launch it from R, import a dataset (CSV or Excel), configure your model, and fit it interactively.

The application provides a complete workflow: data import, variable configuration, model fitting, diagnostic plots, variable importance, model equations, and downloadable reports in HTML, Word, or PDF format.

## 1.2 One Package, Three Routines

As of the current release, the **earthUI** package also *contains* its two Post Processing Modules, which in earlier releases were the separate packages **glmnetUI** and **mgcvUI**. Installing **earthUI** installs all three routines; each runs as its own local Shiny application:

- **earthUI::launch()** — the **earth** (MARS) application described in this article, on port 7878. **earth()** is the central valuation engine.
- **earthUI::launch\_glmnet()** — the elastic net Post Processing Module (port 7879), which imports the **earth** model’s basis functions and re-estimates them under lasso/elastic-net regularization (see “**glmnetUI** Import” below).
- **earthUI::launch\_mgcv()** — the GAM Post Processing Module (port 7880), which uses the **earth** model’s knot locations as starting points for **mgcv** smooth terms (see “**mgcvUI** Auto-Export” below).

The three routines share one project tree, one settings database, and one copy of the supporting infrastructure, so a project opened in one application is immediately available to the others. The **glmnet** and **mgcv** routines are supporting evidence for the **earth** model’s value conclusion, not competing workflows: the intended pattern is to fit and refine the **earth** model first, then run the modules against the same project when corroboration is wanted.

## 1.3 Three Purpose Modes

**earthUI** organizes work into **projects** — named entities that hold your input files, outputs, and settings together at a fixed location. Every project carries a **Purpose**, chosen in the New Project wizard when the project is created. There are three modes:

- **General** — Earth regression for any type of population or dataset. This is the default mode. It provides the full MARS modeling workflow without any domain-specific additions.
- **For Appraisal** — Earth regression tailored for real estate appraisal. Adds features specific to single-property valuation, including subject property handling, special column designations, Residual Constraint Approach (RCA), and Sales Comparison Grid generation.
- **Market Area Analysis** — Earth regression tailored for market area studies. Adds features for analyzing groups of properties in a defined market rather than valuing a single subject.

In all three modes, the core modeling engine is identical — you are always fitting an Earth (MARS) model. The purpose setting controls which additional tools and interface elements are available.

**Purpose is fixed per project.** A project’s purpose — along with its country, state, county, and city — is locked when the project is created. Switching to a different project resets **earthUI** to a clean default state (imported data, model results, tabs, variable configuration, and **earth** parameters are all cleared) and then restores that project’s saved configuration for its purpose.

## 1.4 Real Estate–Specific Features

When either **For Appraisal** or **Market Area Analysis** is selected, earthUI activates several features designed for real estate analysis:

- **Special column designations** — Each predictor can be tagged with a special role such as `contract_date`, `dom`, `concessions`, `latitude`, `longitude`, `living_area`, `lot_size`, `actual_age`, `effective_age`, `area`, `site_dimensions`, or `display_only`. These designations control how the column is handled during fitting, output, and Sales Grid generation. See Chapter 6 for the complete reference.
- **Rounding of latitude and longitude** — Columns designated as latitude or longitude are automatically rounded to 3 decimal places to prevent overfitting.
- **Sale Age column** — When a column is designated as `contract_date` and an Effective Date is provided, earthUI computes a `sale_age` column (days between sale date and effective date) and substitutes it as a predictor.
- **RCA computations (appraisal only)** — In appraisal mode, after fitting the model earthUI can compute Residual Constraint Approach (RCA) output, which produces per-comparable adjustments, net/gross adjustment summaries, and an adjusted sale price for the subject property.
- **Sales Comparison Grid (appraisal only)** — Generates a multi-sheet Excel workbook with formulas for comparable adjustments, residual feature breakdowns, and adjusted sale prices. See Chapter 11.

***Tip:** The General purpose mode works for any dataset — financial, scientific, engineering, or real estate. The appraisal and market modes simply add convenience features for real estate professionals.*

## 1.5 Installing earthUI and vProlog

earthUI is published on CRAN, so the standard installation is one line in R:

```
install.packages("earthUI")
```

This article describes the current release. If CRAN is still carrying an older version when you read this, the development version installs from GitHub (note the `subdir` — the package lives in the repository’s `pkg/` folder):

```
remotes::install_github("wcraytor/earthUI", subdir = "pkg")
```

**vProlog (optional).** The experimental remarks-processing facility (see “Prolog Text Extraction” near the end of this article) uses the vProlog package — a self-contained embedded Prolog engine. It is deliberately *not* a hard dependency: earthUI installs and runs without it, and the Prolog features simply stay disabled until it is present. vProlog is distributed through the author’s r-universe repository, which serves prebuilt binaries for Windows and macOS (no compiler toolchain needed):

```
install.packages("vProlog",
  repos = c("https://wcraytor.r-universe.dev",
    getOption("repos")))
```

No system Prolog installation (SWI-Prolog or otherwise) is required — the Trealla engine is compiled into the package itself. Everything else in this article works without vProlog.

## 1.6 Getting Started

To use earthUI:

1. **Install the package** from CRAN as described above.
2. **Launch the application:** run `earthUI::launch()` in R, or from the command line run `Rscript -e 'earthUI::launch()'`. The app opens in your web browser on port 7878. You can also access the app directly by navigating to `http://localhost:7878` in your browser. (The Post Processing Modules start the same way: `launch_glmnet()` on port 7879, `launch_mgcv()` on port 7880.)
3. **Create or open a project** in Section 1 of the sidebar. The New Project wizard captures the project name, Purpose (General, For Appraisal, or Market Area Analysis), location (country, state, county, city), and optionally an initial data file, which is copied into the project's `in/` folder.
4. **Import your data** in Section 2, which lists the CSV and Excel files in the active project's `in/` folder. Files you add manually via Finder or Explorer appear after clicking Refresh.
5. **Configure variables** — choose your target(s) and predictors, set data types, and assign any special column roles.
6. **Set Earth parameters** — degree, nprune, subset filters, and other `earth()` arguments.
7. **Fit the model** — click “Fit Earth Model” and review the results in the main panel.
8. **Export** — download predictions as Excel, generate Quarto report bundles, or (in appraisal mode) compute RCA adjustments and Sales Comparison Grids.

Configuration is saved server-side in a project database, keyed by project and purpose, via the “Save current as default” buttons in Sections 3 and 4 (see Chapter 3, “Settings Persistence”).

## 2 MLS Input Data Requirements

For real estate appraisal and market analysis workflows, your input data typically comes from a Multiple Listing Service (MLS) export. This chapter describes the expected file structure and the columns that earthUI can use.

### 2.1 File Format & Structure

earthUI accepts **CSV** and **Excel** (.xlsx, .xls) files. On import, column names are automatically converted to **snake\_case** — for example, “Living SqFt” becomes **living\_sqft**, “Contract Date” becomes **contract\_date**, and “Sale Price” becomes **sale\_price**. This normalization ensures consistent column references throughout the workflow. The CSV separator and decimal mark used during import are determined by the locale settings (see Chapter 3, “Locale & Regional Settings”).

Your data file should be a flat table with one row per property and one column per attribute. The first row of the file must contain column headers.

### 2.2 Required Columns for Appraisal Mode

While earthUI works with any set of columns, the full appraisal workflow (RCA adjustments + Sales Comparison Grid) benefits from having the following columns in your MLS export:

| Column           | Special Type           | Purpose                                                            |
|------------------|------------------------|--------------------------------------------------------------------|
| Sale Price       | (target)               | Response variable for the model                                    |
| Contract Date    | <b>contract_date</b>   | Used to compute <b>sale_age</b> (days from effective date)         |
| Listing Date     | <b>listing_date</b>    | Used with contract date to compute DOM if no DOM column exists     |
| Days on Market   | <b>dom</b>             | Days on market; displayed in Sales Grid                            |
| Concessions      | <b>concessions</b>     | Sale concessions; Net SP = Sale Price – Concessions in Sales Grid  |
| Living Area (SF) | <b>living_area</b>     | Enables per-SF residuals ( <b>residual_sf</b> , <b>cqa_sf</b> )    |
| Lot Size         | <b>lot_size</b>        | Grouped in "Site Size   Dimensions" row in Sales Grid              |
| Site Dimensions  | <b>site_dimensions</b> | Grouped with lot size in Sales Grid                                |
| Latitude         | <b>latitude</b>        | Rounded to 3 dp; used for proximity calculation and Location group |
| Longitude        | <b>longitude</b>       | Rounded to 3 dp; used for proximity calculation and Location group |
| Area ID          | <b>area</b>            | Grouped in "Loc: Long   Lat   Area" row in Sales Grid              |
| Actual Age       | <b>actual_age</b>      | Grouped in "Actual Age   Effective Age" row in Sales Grid          |
| Effective Age    | <b>effective_age</b>   | Grouped with actual age in Sales Grid                              |
| Parcel Number    | (none)                 | Displayed in APN row of Sales Grid                                 |
| Listing ID       | (none)                 | Displayed in APN row of Sales Grid                                 |
| Address          | <b>display_only</b>    | Displayed in Address row of Sales Grid; excluded from model        |

Spreadsheet column names can be in a foreign language — the “special” names are in English so that the R program can give them special treatment. Otherwise, the given column names show up in the regression models, graphs, and (if doing appraisals) the Intermediate Sales Grid.

Not all columns are required. earthUI adapts — if a column is missing, the corresponding feature is simply omitted. For example, if no **concessions** column is designated, the Net SP

row in the Sales Grid shows Sale Price without a concessions deduction. However, for real estate pricing models certain columns are highly recommended to achieve acceptable fit:

1. **Sale Age** — the number of days between the contract sale date and the effective date of the appraisal or analysis. If multi-year sales history is being used, especially for periods over 5 years, `sale_age` often plays a central role in estimating the sale price. In fact it is often so important that without it, earthUI fails to provide any model at all.
2. **Living Area** — also goes by names such as “Living Sqft,” “GLA” (gross living area) and so on. This is another leading determinant of sale price.
3. **Total Bath Count** — the total number of full, quarter, half, and 3/4 bathrooms. For example, two full baths and one half-bath would be a value of 2.5.
4. **Garage Bays** or **Garage Area** — the number of garage spaces or the garage square footage.
5. **Lot Size** — the land area of the property, typically in square feet or acres.
6. **Longitude, Latitude**, and if available **Area ID**. Adjustments for these will be combined under a single Location adjustment in the Sales Grid.

## 2.3 Special Column Naming Conventions

earthUI identifies columns by their **special type designation**, not by their column name. You can name your columns anything you like in the MLS export — what matters is that you assign the correct special type in the Variable Configuration table (Chapter 6).

For example, your MLS might export living area as “GLA”, “Living SqFt”, “liv\_area”, or “gross\_living\_area”. After import (where it becomes snake\_case), you simply designate it as `living_area` in the Special dropdown. earthUI will then use it for per-SF residual calculations and Sales Grid grouping regardless of its original name.

## 2.4 Data Quality & Completeness

- **Missing values (NA):** Rows with NA values in any predictor or target column are automatically removed before fitting. The `na.action` is always set to `na.fail` internally, with NA removal handled before the call to `earth()`.
- **Date columns:** Dates should be in a format R can parse (e.g., 2025-06-15, 06/15/2025, June 15, 2025). earthUI auto-detects date columns when at least 80% of values parse successfully. Two-digit years (e.g., 06/15/25) are read as the nearest matching year, so recent MLS exports parse correctly; four-digit years are always taken literally.
- **Numeric columns:** Sale price, living area, lot size, concessions, and similar fields must be numeric. If your MLS exports prices with currency symbols or thousands separators (e.g., “\$350,000” or “350.000,00”), you may need to clean these before import.
- **Factor columns:** Categorical variables like area ID, style, or condition should contain a manageable number of unique values. earthUI auto-detects factors but you can override the detection.

**Tip:** Review the T/F/NA column in the predictor table after import — it shows true/false/NA counts for each column. Columns with more than 30% missing values are flagged in red. Consider excluding high-NA columns or cleaning the data before import.

## 2.5 Subject Row Placement

In **Appraisal** mode, **row 1 must be the subject property**. All remaining rows are comparable sales. The subject’s sale price can be left blank (NA) or set to any value — earthUI treats it as NA during fitting regardless.

In **Market Area Analysis** mode, placing a subject in row 1 is optional, and no row is excluded automatically. To hold a subject row (or any other row) out of the fit, give it a zero value in a designated weights column or use a subset filter.

In **General** mode, there is no special row handling — all rows are treated equally.

## 3 General Purpose Mode

### 3.1 Overview

General Purpose mode is the default when you launch earthUI. It provides the complete MARS modeling workflow for any dataset — not just real estate. You can use earthUI for scientific data, financial analysis, engineering studies, or any regression problem where you want to explore non-linear relationships and interactions between variables.

In General mode, the interface omits the real estate-specific features (special columns, sale age, coordinate rounding, RCA). The sidebar is streamlined to focus on variable selection, parameter configuration, model fitting, and export.

### 3.2 The Sidebar Workflow

The sidebar is organized into numbered, collapsible sections that guide you from project selection through export:

**1. Project** — Pick an existing project or create a new one via the New Project wizard, which captures the purpose, country, and administrative location (all locked once created). The active project determines where input files are read from and where all outputs are written — there is no separate output-folder field.

**2. Import Data** — Choose a CSV or Excel file from the active project’s `in/` folder; a Refresh link rescans the folder for files added manually. For Excel files with multiple sheets, a sheet selector appears. Column names are automatically converted to `snake_case`.

**3. Variable Configuration** — Target variable selector (supports multiple targets), Effective Date, and the predictor table with checkboxes for Include, Factor, Linear, and Latent. See Chapter 6 for full details.

**4. Earth Call Parameters** — All arguments to the `earth()` function: degree, penalty, nk, pruning method, cross-validation, and more. See Chapter 7 for the complete parameter reference.

**Execute Prolog Processing** — An optional, unnumbered section between Sections 4 and 5, disabled unless the experimental Prolog feature is enabled in Settings. See Chapter 14.

**5. Fit Earth Model** — A random-seed field and the button that runs the model asynchronously. See Chapter 8 for fitting details.

**6. Download Estimated Target Variable(s) & Residuals** — (titled “Download Estimated Sale Prices & Residuals” in appraisal and market modes) Exports predictions, residuals, CQA scores, and per-g-function contributions as an Excel file. See Chapter 9.

**7–8. RCA Adjustments and Sales Grid** — Appraisal mode only. See Chapters 10 and 11.

**9. Generate Quarto Report** — Writes a self-contained Quarto report bundle (`.qmd` source plus assets) to the project’s output folder. Numbered 7 outside appraisal mode. See Chapter 12.

**10. Convert Quarto Report** — Renders any `.qmd` file to HTML, Word, or PDF. See Chapter 12.

### 3.3 Main Panel Tabs

After fitting, the main panel provides ten tabs:

| Tab                   | Contents                                                                                                                                                                                      |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data                  | Preview of imported data. In appraisal mode, split into Subject Property and Comparable Sales tables.                                                                                         |
| Equation              | The fitted model equation displayed in LaTeX/MathJax, showing each basis function and its coefficient.                                                                                        |
| Summary               | Key metrics ( $R^2$ , GRSq, GCV, RSS, CV $R^2$ ) and a coefficients table.                                                                                                                    |
| Variable Importance   | Bar chart and ranked table of predictor importance scores.                                                                                                                                    |
| Contribution          | g-function contribution plots. Select a g-function from the dropdown; univariate predictors show line plots, bivariate interactions (degree $\geq 2$ ) show 3D perspective and contour plots. |
| Correlation           | Heatmap of predictor correlations.                                                                                                                                                            |
| Diagnostics           | Three plots: Residuals vs Fitted, Normal Q-Q, and Actual vs Predicted.                                                                                                                        |
| RCA Adjustments       | (appraisal only) Histograms of Residual Adj %, Net Adj %, and Gross Adj % across the comparables, populated after Step 7 (Calculate RCA Adjustments) completes.                               |
| Residual Contribution | Estimates market contributions of rare or excluded features by jointly regressing model residuals on columns the model never used (see <a href="#">subsection 4.4</a> ).                      |
| ANOVA                 | ANOVA decomposition table showing each basis function's contribution to RSS.                                                                                                                  |
| Earth Output          | Raw text output from 'earth::summary()', including the pruning pass details.                                                                                                                  |

### 3.4 Settings Persistence

earthUI saves your configuration server-side in a project database (`projects.sqlite` in the `regProj` root), keyed by project and purpose. Saving is **button-driven**: the “Save current as default” button in Section 3 saves the target, effective date, and predictor configuration, while the button in Section 4 saves the earth call parameters and the Allowed Interactions matrix. Each button updates only its own half, so they never clobber each other, and **fitting does not auto-save** — you can experiment freely without overwriting your saved defaults. A project's several input files (a small test extract, the full dataset) share one configuration, and because settings live in the project tree, they travel with it when you sync, back up, or share the `regProj` folder.

### 3.5 Dark Mode

Click the moon/sun icon in the upper-right corner to toggle between light and dark themes. The theme preference is saved in local storage and persists across sessions.

### 3.6 Locale & Regional Settings

earthUI supports international number, date, and CSV formatting conventions through a country-based locale system. The **Country** dropdown in the **Settings** (gear) menu of the top menu bar selects a preset for 31 supported countries. Each preset configures:

- **CSV separator** — comma (,) for US/UK/Japan or semicolon (;) for most of Europe, where the comma is used as a decimal mark.
- **Decimal mark** — period (.) or comma (,).
- **Thousands separator** — comma (US/UK/Japan), period (Germany/Italy/Spain), space (Finland/France/Poland/Baltics/Ukraine/Russia), or apostrophe (Switzerland).

- **Date format** — MM/DD/YYYY (US), DD/MM/YYYY (most of Europe), or YYYY-MM-DD (Sweden/Lithuania/Japan/Canada).
- **Paper size** — Letter (US/Canada/Mexico) or A4 (everywhere else).

### 3.6.1 Supported Countries

| Country                                                                         | CSV Sep | Decimal | Thousands | Date    | Paper  |  |
|---------------------------------------------------------------------------------|---------|---------|-----------|---------|--------|--|
| US, Canada (csv)                                                                | ,       | .       | ,         | MDY/YMD | Letter |  |
| UK, Ireland, AU, NZ                                                             | ,       | .       | ,         | DMY     | A4     |  |
| Germany, Austria, Belgium, Netherlands, Italy, Spain, Portugal, Denmark, Brazil | ;       | ,       | .         | DMY     | A4     |  |
| France, Poland, Czech Rep., Norway, Latvia, Estonia                             | ;       | ,       | (space)   | DMY     | A4     |  |
| Finland, Sweden                                                                 | ;       | ,       | (space)   | DMY/YMD | A4     |  |
| Switzerland                                                                     | ;       | ,       | '         | DMY     | A4     |  |
| Ukraine, Russia                                                                 | ;       | ,       | (space)   | DMY     | A4     |  |
| Turkey                                                                          | ;       | ,       | .         | DMY     | A4     |  |
| Japan, South Korea                                                              | ,       | .       | ,         | YMD     | A4     |  |
| Mexico                                                                          | ,       | .       | ,         | DMY     | Letter |  |
| Lithuania                                                                       | ;       | ,       | (space)   | YMD     | A4     |  |

### 3.6.2 Paper Override

Below the country selector, a **Paper** dropdown (Letter or A4) lets you override the report page size without switching countries. The CSV separator, decimal mark, and date-format order follow the selected country's preset; when you change the country, the paper override resets to that country's default.

### 3.6.3 Saving Defaults

Click **Save** in the Settings menu to store your locale preferences globally. These defaults apply to all future sessions regardless of which project you open. Per-project settings (target, predictors, parameters) are saved separately in the project database; locale defaults persist across all projects via a user-level SQLite database. This two-level approach is designed for organizations like audit firms that work with data from multiple countries — set your most common country as the default, then adjust when needed.

***Tip:** Number formatting on plot axes and slope labels automatically adapts to your locale. German locale uses periods for thousands (200.000), Finnish uses spaces (200 000), Swiss uses apostrophes (200'000). No currency symbols are displayed — earthUI is currency-agnostic.*

## 4 Appraisal Mode

When you select **For Appraisal** as the Purpose, earthUI configures itself for single-property valuation. All features described in Chapter 3 remain available; this chapter covers only the appraisal-specific additions.

### 4.1 Subject Row Handling

In appraisal mode, **row 1 of your dataset is the subject property** and all remaining rows are comparable sales. Your input file must be organized accordingly (see Chapter 2). The subject's sale price can be left blank or set to any value — earthUI automatically treats it as NA during fitting.

After importing, the Data tab splits into two sections: **Subject Property** (row 1) and **Comparable Sales** (rows 2+). Row 1 is always excluded from model fitting — the notification “Skipping row 1 (subject). Fitting on N rows.” confirms this. After fitting, the model still generates predictions for the subject row, shown as `est_<target>` in the output.

### 4.2 Effective Date & Sale Age

An **Effective Date** field appears in the Variable Configuration section in every purpose mode (defaulting to today's date) — general and market analyses have an effective date too, not just appraisals — but it is only *used* in appraisal and market modes. There, if you designate a column as `contract_date` in the Special column dropdown, earthUI computes a `sale_age` column — the number of integer days between each sale's contract date and the effective date. This column replaces the original date column as a predictor.

The first time you click Fit after designating a contract date, earthUI creates the `sale_age` column and notifies you to click Fit again to include it.

### 4.3 Special Column Designations

In appraisal and market modes, a **Special** dropdown appears for each predictor in the Variable Configuration table. The complete list of special types and their effects:

| Special Type    | Effect                                                                                                                                     |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| no              | Default — no special handling                                                                                                              |
| contract_date   | Triggers automatic <code>sale_age</code> computation from the Effective Date                                                               |
| listing_date    | Used with contract date to compute DOM when no <code>dom</code> column exists                                                              |
| dom             | Identifies the Days on Market column; displayed in Sales Grid                                                                              |
| concessions     | Identifies sale concessions; used in Net SP formula (Sale Price – Concessions) in Sales Grid                                               |
| latitude        | Values rounded to 3 dp; used for Haversine proximity and Location group in Sales Grid                                                      |
| longitude       | Values rounded to 3 dp; used for Haversine proximity and Location group in Sales Grid                                                      |
| area            | Grouped in "Loc: Long   Lat   Area" row in Sales Grid                                                                                      |
| living_area     | Enables per-SF residuals ( <code>residual_sf</code> , <code>cqa_sf</code> ) in download output                                             |
| lot_size        | Grouped in "Site Size   Dimensions" row in Sales Grid                                                                                      |
| site_dimensions | Grouped with lot size in Sales Grid                                                                                                        |
| actual_age      | Grouped in "Actual Age   Effective Age" row in Sales Grid                                                                                  |
| effective_age   | Grouped with actual age in Sales Grid                                                                                                      |
| display_only    | Column included in exports but excluded from model fitting (multiple allowed)                                                              |
| remarks         | (Experimental) Free-text remarks column parsed into feature columns by the optional Prolog step (multiple allowed); see Chapter 14         |
| list            | (Experimental) Comma-delimited column split into one-hot TRUE/FALSE columns by the optional Prolog step (multiple allowed); see Chapter 14 |

Only one column per special type is allowed (except `display_only`, `remarks`, and `list`). Assigning a special to a second column automatically clears it from the first. A small blue badge appears next to the variable name showing its assigned special type.

#### 4.4 Residual Contribution (Rare-Feature Estimation)

Features too rare for the regression channel — a workshop present in a handful of sales — belong to the RCA residual layer. The **Residual Contribution** tab estimates their market contribution by regressing the fitted model's residuals on columns the model never used, via ordinary least squares. Selecting several columns estimates them *jointly*, so correlated amenities (e.g. `outbuilding_sqft` plus `pr_has_workshop`) split the credit instead of double-counting. Binary flags yield lump-sum estimates and list each flagged sale's residual as evidence; numeric columns yield per-unit contributions; the reported standard error is one SE ( $t \geq 2$  as the usual significance yardstick). Terms supported by fewer than ten observations receive an automatic CAUTION in the generated workfile note, which documents method, estimates, and caveats in one paragraph for the report. Only predictors the model actually *used* are excluded from the candidate list — a rare flag offered to the fit but never selected by the forward pass remains eligible, since the model absorbed none of its value.

#### 4.5 RCA Adjustments Overview

The **Calculate RCA Adjustments & Download** button (sidebar section 7, visible only in appraisal mode after fitting) computes market-derived adjustments for each comparable relative to the subject. The full RCA workflow is described in Chapter 10.

After computing RCA adjustments, the **Generate Sales Grid & Download** button (sidebar section 8) becomes available. The Sales Grid workflow is described in Chapter 11.

**Tip:** The CQA score you assign to the subject controls how much of the residual distribution is attributed to the subject. A score of 5.00 places the subject at the median of the comparables.

## 5 Market Area Analysis Mode

When you select **Market Area Analysis** as the Purpose, earthUI provides the same real estate-specific features as appraisal mode (special columns, sale age, coordinate rounding) but is oriented toward analyzing a group of properties rather than valuing a single subject.

### 5.1 Differences from Appraisal Mode

- **No automatic subject exclusion** — unlike appraisal mode, market mode does not automatically exclude row 1 from fitting. All rows are included; to hold a subject row out of the fit, give it a zero value in a designated weights column (as in the demo dataset) or use a subset filter.
- **No RCA or Sales Grid sections** — the “Calculate RCA Adjustments & Download” and “Generate Sales Grid & Download” steps are not available. Market mode focuses on model fitting and output, not per-comparable adjustments.
- **Sidebar numbering** — without the RCA and Sales Grid steps, the Generate Quarto Report section is numbered 7 instead of 9 (the Convert Quarto Report section remains 10).

### 5.2 When to Use Market Mode

Market Area Analysis mode is appropriate when you are:

- Building a regression model for a neighborhood or market area to understand value drivers
- Analyzing how variables like square footage, age, lot size, and location affect sale prices across a group of properties
- Preparing support for a market conditions analysis or neighborhood delineation
- Working with a dataset that includes a subject property in row 1 but you want the option of including or excluding it from the fit

***Tip:** Market mode is also useful for general real estate regression where you want special column features (sale age, coordinate rounding) but do not need the RCA adjustment workflow.*

## 6 Variable Selection

Section 3 of the sidebar — **Variable Configuration** — is where you choose which columns participate in the model and how they are treated.

### 6.1 Target Variable(s)

The **Target (response) variable(s)** dropdown at the top of Section 3 lists every column in your dataset. Select one column for a standard single-response model, or multiple columns for a multi-response model. When multiple targets are selected, the model fits all responses simultaneously using `earth(cbind(y1, y2, ...) ~ .)`.

Columns selected as targets are automatically excluded from the predictor list.

### 6.2 The Predictor Table

Below the target selector, a table lists every remaining column with the following fields:

| Column          | Description                                                                                                                                                                                                |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Variable</b> | Column name (full name shown in tooltip if truncated), preceded by a $\times$ / $+$ <b>toggle</b> (see below). In appraisal/market modes, a blue badge shows the assigned special type.                    |
| <b>Type</b>     | Data type dropdown: <b>numeric</b> , <b>integer</b> , <b>character</b> , <b>logical</b> , <b>factor</b> , <b>Date</b> , <b>POSIXct</b>                                                                     |
| <b>Include</b>  | Checkbox — include this column as a predictor in the model                                                                                                                                                 |
| <b>Factor</b>   | Checkbox — treat this column as a categorical variable                                                                                                                                                     |
| <b>Linear</b>   | Checkbox — force linear entry only (no hinge functions)                                                                                                                                                    |
| <b>Latent</b>   | Checkbox — hold this column out of the model entirely (excluded from <code>earth</code> and from the data exported to <code>glmnetUI</code> / <code>mgcvUI</code> ), reserving it for latent-variable work |
| <b>Special</b>  | Dropdown (appraisal/market only) — see Special Column Types Reference below                                                                                                                                |
| <b>T/F/NA</b>   | Three counts per column: for logicals, TRUE/FALSE/NA; for numerics, $\neq 0$ / $= 0$ /NA. Shown in red when more than 30% of values are missing                                                            |

A hint line above the table explains the abbreviations: “Type = column data type, Inc = include as predictor, Factor = treat as categorical, Linear = linear-only (no hinges), Special = column role (e.g. `contract_date`).”

**Soft-disable instead of delete.** The  $\times$  toggle beside each variable name greys out the row and holds the column out of the model without removing it from the data; it flips to a green  $+$  that re-enables the column. Columns are never destroyed.

### 6.3 Data Type Detection & Overrides

`earthUI` automatically detects data types on import. Numeric, integer, logical, factor, and date columns are recognized. Character columns that look like dates (at least 80% of values parse against common date formats) are classified as **Date**.

You can override any detection by changing the **Type** dropdown. When you change a column to **character** or **factor**, the Factor checkbox is automatically checked. Changing types affects how the column is passed to the `earth()` function.

## 6.4 Factor and Linear Flags

- **Factor** — When checked, the column is treated as a categorical variable. This is appropriate for columns like **style**, **area\_id**, or **grade** that represent discrete groups rather than continuous measurements. Factor columns enter the model as indicator (dummy) variables.
- **Linear** — When checked, the column is forced to enter the model linearly — no hinge (piecewise-linear) functions are created for it. This is useful when you know a variable has a strictly linear relationship with the target.

## 6.5 Special Column Types Reference

In appraisal and market modes, the **Special** dropdown provides the following options. Each type can be assigned to at most one column (except **display\_only**, **remarks**, and **list**, which allow multiple):

### Date & Time Types:

- **contract\_date** — Triggers automatic **sale\_age** computation. The original date column is replaced by an integer column measuring days between the sale date and the Effective Date.
- **listing\_date** — Used as a fallback for computing Days on Market (DOM = contract date – listing date) when no explicit **dom** column is designated.
- **dom** — Identifies the Days on Market column. Displayed in the Sales Grid’s APN row and Date of Sale row.

### Monetary Types:

- **concessions** — Identifies sale concessions (seller credits, buyer incentives, etc.). Used in the Sales Grid to compute Net Sale Price:  $\text{Net SP} = \text{Sale Price} - \text{Concessions}$ .

### Size & Location Types:

- **latitude** — Values are automatically rounded to 3 decimal places to prevent overfitting. Used for Haversine proximity calculations (distance from subject to each comp) and grouped in the Location row of the Sales Grid.
- **longitude** — Same rounding treatment as latitude. Used for proximity and the Location group.
- **area** — Typically a neighborhood or area identifier. Grouped with latitude and longitude in the “Loc: Long | Lat | Area” row of the Sales Grid.
- **living\_area** — Enables per-square-foot residual calculations (**residual\_sf** and **cqa\_sf**) in the download output.
- **lot\_size** — Grouped in the “Site Size | Dimensions” row of the Sales Grid.
- **site\_dimensions** — Grouped with lot size in the Sales Grid (e.g., “75x120”).

### Age Types:

- **actual\_age** — Grouped in the “Actual Age | Effective Age” row of the Sales Grid.
- **effective\_age** — Grouped with actual age in the Sales Grid.

### Display Types:

- **display\_only** — The column is included in Excel exports but excluded from model fitting entirely. Use this for address fields, MLS numbers, parcel IDs, or other reference data that should not be a predictor. Multiple columns can have this designation.

### Text Extraction Types (experimental):

- **remarks** — A free-text column (e.g. **public\_remarks**) to be parsed into feature columns by the optional Prolog step. Does nothing until that feature is enabled in Settings. Multiple columns allowed. See Chapter 14.

- **list** — A comma-delimited column (e.g. **existing\_structures**) to be split into one-hot `<col>_<value>` TRUE/FALSE columns by the same step. Multiple columns allowed. See Chapter 14.

## 6.6 Multiple Targets

Selecting more than one target variable fits a multi-response Earth model. When multiple targets are selected:

- The model predicts all responses simultaneously, sharing basis functions across targets
- The **wp (response weights)** button becomes active, allowing you to assign a numeric weight to each target
- Variance models (**varmod.method**) are disabled (not supported for multi-response)
- Results tabs display per-response metrics, equations, and diagnostic plots

***Tip:** Columns designated as **display\_only** remain in your dataset and appear in the Excel export, but are excluded from the predictor table entirely. Use this for ID columns, addresses, or other reference fields.*

## 7 Parameter Selection

Section 4 of the sidebar — **Earth Call Parameters** — provides access to all arguments accepted by the `earth()` function. Each parameter has a blue help icon (?) with a tooltip explanation. Parameters are organized into collapsible subsections.

### 7.1 Basic Parameters

| Parameter      | Default | Description                                                                                                                   |
|----------------|---------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>subset</b>  | (empty) | Row filter expression. See “Subset Filtering” below.                                                                          |
| <b>weights</b> | NULL    | Column selector for case (row) weights. Only numeric columns are listed.                                                      |
| <b>wp</b>      | NULL    | Response weights for multi-target models. Button opens a dialog with one numeric input per target (default 1.0 each).         |
| <b>keepxy</b>  | off     | Retain x, y, subset, and weights in the model object.                                                                         |
| <b>trace</b>   | 0       | Trace level for fitting output (0 through 5).                                                                                 |
| <b>glm</b>     | none    | Optional GLM family: <b>gaussian</b> , <b>binomial</b> , or <b>poisson</b> .                                                  |
| <b>degree</b>  | 1       | Maximum interaction order. Setting degree $\geq 2$ auto-enables cross-validation and reveals the Allowed Interactions matrix. |
| <b>penalty</b> | 2       | GCV penalty per knot. Higher values produce simpler models.                                                                   |

### 7.2 Forward Pass

| Parameter      | Default | Description                                                                                                                          |
|----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>nk</b>      | auto    | Maximum terms before pruning. Auto-filled with the recommended value for the loaded data (see “Recommended Parameter Values” below). |
| <b>thresh</b>  | 0.001   | Forward-step threshold. Smaller values allow more terms.                                                                             |
| <b>minspan</b> | auto    | Minimum span between knots. Auto-filled with the recommended value. Negative values set the maximum knots per predictor.             |

| Parameter             | Default | Description                                                                                              |
|-----------------------|---------|----------------------------------------------------------------------------------------------------------|
| <b>endspan</b>        | auto    | End span — minimum distance from a knot to the edge of the data. Auto-filled with the recommended value. |
| <b>newvar.penalty</b> | 0       | Penalty for introducing a new variable (encourages reuse of existing predictors).                        |
| <b>fast.k</b>         | 20      | Number of parent terms to consider in the fast MARS algorithm.                                           |
| <b>fast.beta</b>      | 1       | Controls the fast MARS aging factor.                                                                     |

### 7.3 Allowed Interactions

When **degree** is set to 2 or higher, an interaction matrix appears below the basic parameters. This is an upper-triangular grid of checkboxes, one for each predictor pair. A checked box means the two predictors are allowed to interact; unchecking it forbids that specific interaction.

Clicking a predictor name (row or column label) toggles all interactions for that variable. **Allow All** and **Clear All** checkboxes at the top provide bulk control. The matrix uses sticky headers so column and row labels remain visible when scrolling.

An info alert reminds you: “Interaction terms increase the risk of overfitting. Cross-validation has been enabled (10-fold).”

### 7.4 Pruning

| Parameter      | Default  | Description                                                                                                         |
|----------------|----------|---------------------------------------------------------------------------------------------------------------------|
| <b>pmethod</b> | backward | Pruning method: <b>backward</b> , <b>none</b> , <b>exhaustive</b> , <b>forward</b> , <b>seqrep</b> , or <b>cv</b> . |
| <b>nprune</b>  | NULL     | Maximum terms after pruning. Leave empty for automatic selection.                                                   |

### 7.5 Cross-Validation

| Parameter       | Default | Description                                                           |
|-----------------|---------|-----------------------------------------------------------------------|
| <b>nfold</b>    | 10      | Number of cross-validation folds. Set to 0 to disable CV.             |
| <b>ncross</b>   | 20      | Number of cross-validation repetitions.                               |
| <b>stratify</b> | on      | Stratify CV samples so each fold has a similar response distribution. |

When  $\text{degree} \geq 2$ , cross-validation is automatically enabled (nfold set to 10) to help guard against overfitting from interactions.

## 7.6 Subset Filtering

The **subset** text input accepts an R expression that filters which rows are used for model fitting. You can type an expression directly (e.g., `sale_age < 365 & area_id == 460`) or use the **Build filter...** button to construct one visually.

**Build Filter Dialog** — Click “Build filter...” to open a guided dialog. Each condition row has a column dropdown, operator (`<`, `>`, `<=`, `>=`, `==`, `!=`), and a value input that adapts to the column type: numeric input for numbers, date picker for dates, and dropdown of unique values for character/factor columns. Conditions are joined with AND (`&`) or OR (`|`) connectors. A preview at the bottom shows the expression and how many rows match. Click **Apply** to insert the expression into the text input.

**Date columns** must use `as.Date("...")` or `as.POSIXct("...")` wrappers in manual expressions. The Build Filter dialog handles this automatically.

**Subset filtering is non-destructive** — excluded rows remain in the dataset and receive predictions in the Excel export.

**Tip:** Use parentheses to group OR conditions when combining with AND. Without parentheses, `A & B | C` is evaluated as `(A & B) | C`, which may not be what you intend.

## 7.7 Recommended Parameter Values

earthUI displays a recommended value below key parameters in Section 4. These recommendations update reactively based on the number of fitting rows ( $n$ ) and selected predictors ( $p$ ). The formulas are derived from Friedman’s MARS paper, earth’s internal algorithms, and empirical testing. The **nk**, **minspan**, and **endspan** fields are pre-filled with their recommended values as the data loads; a manual edit is preserved.

### 7.7.1 Forward Pass Parameters

**nk** (max terms before pruning):

$$nk = \min(100, \max(21, 2p + 1, \lfloor n/10 \rfloor))$$

| $n$  | $p=5$ | $p=10$ |
|------|-------|--------|
| 30   | 21    | 21     |
| 50   | 21    | 21     |
| 100  | 21    | 21     |
| 200  | 21    | 21     |
| 500  | 50    | 50     |
| 1000 | 100   | 100    |
| 1500 | 100   | 100    |

Earth’s default,  $\min(200, \max(20, 2p)) + 1$ , does not account for dataset size and can be too small for large datasets, constraining the forward pass before it explores all predictors adequately. The  $\lfloor n/10 \rfloor$  term allows approximately one term per 10 observations — a standard rule of thumb for avoiding overparameterization. The cap of 100 prevents diminishing returns.

**minspan** (minimum observations between knots):

$$\text{minspan} = \min(16, \lfloor 5 + n/50 \rfloor)$$

**endspan** (minimum observations from data boundaries):

$$\text{endspan} = \min(16, \lfloor 5 + n/28 \rfloor)$$

| $n$  | minspan | endspan |
|------|---------|---------|
| 30   | 5       | 6       |
| 50   | 6       | 6       |
| 100  | 7       | 8       |
| 200  | 9       | 12      |
| 300  | 11      | 15      |
| 500  | 15      | 16      |
| 1500 | 16      | 16      |

Earth’s auto-calculated minspan uses Friedman’s equation 43:  $\lfloor (-\ln(-\ln 0.95) + \ln(p \cdot n)) / (2.5 \ln 2) \rfloor$ , which scales as  $\ln(np)$  and grows too slowly for large datasets. At  $n=1500$  it yields only  $\approx 7$ , giving roughly 245 candidate knot locations per continuous predictor — too many, allowing the forward pass to fit noise. The recommended formula targets approximately 100 candidates per predictor for large  $n$ , while deferring to earth’s auto-calculation for small  $n$  (where it is well-tested). Endspan is set slightly larger than minspan to provide additional boundary protection, which helps when data has thin tails (common in real estate).

**penalty** (GCV penalty per knot):

$$\text{penalty} = \begin{cases} 3 & \text{if degree} > 1 \\ 2 & \text{if degree} = 1 \end{cases}$$

This is earth’s own default — 2 for additive models, 3 for interaction models (the higher penalty compensates for the larger search space).

**newvar.penalty** (penalty for introducing a new predictor):

Recommended: **0.1** when predictors are correlated (e.g., living area with bedroom count, bathroom count, lot size). This biases the forward pass toward reusing predictors already in the model rather than introducing correlated alternatives. The mechanism: each new predictor’s RSS improvement is multiplied by  $1/(1 + \text{penalty})$ . With `newvar.penalty = 0.1`, a new variable must improve RSS by at least 10% more than another knot on an existing variable. This produces simpler models with fewer predictors without affecting final coefficient estimates (the penalty is removed after selection). Leave at 0 if predictors are not correlated. Note that earth does not support a non-zero `newvar.penalty` together with case weights — earthUI disables the input (forcing 0) whenever a weight column is designated.

### 7.7.2 Pruning Parameters

**pmethod**: Recommended: **backward**. The default GCV-based backward pruning is deterministic — the same data always produces the same model, with no dependence on random seeds. This is important for reproducibility, especially in appraisal work. The `cv` method uses cross-validation for pruning which introduces seed dependence.

**nprune**: Recommended: **leave empty** (NULL). Let GCV select the optimal number of terms. Setting `nprune` imposes a hard cap that overrides GCV’s judgment.

### 7.7.3 Cross-Validation Parameters

**nfold** (CV folds): recommended by sample-size band:

| $n$ (fitting rows) | nfold |
|--------------------|-------|
| 0–299              | 5     |
| 300–399            | 6     |
| 400–499            | 7     |
| 500–599            | 8     |
| 600–699            | 9     |
| 700+               | 10    |

More folds train each fold on more data (a less biased estimate) but shrink the held-out test fold. The table graduates up to the standard 10-fold as the sample grows while keeping each test fold large enough (roughly 50+ rows) for a stable per-fold metric, and falls back to 5-fold for small samples — where the  $\text{CVR}^2$  estimate is still noisy, so raise `ncross` (e.g. 15–20) to average out the variance. The cap of 10 reflects that each additional fold is a full earth fit for negligible diagnostic gain. With `pmethod = "backward"`, cross-validation does not affect the model — it only computes the diagnostic  $\text{CVR}^2$  and provides residuals for the variance model.

**ncross** (CV repetitions): recommended by sample-size band:

| $n$ (fitting rows) | ncross |
|--------------------|--------|
| 0–199              | 20     |
| 200–399            | 16     |
| 400–699            | 12     |
| 700–1199           | 10     |
| 1200–2999          | 7      |
| 3000+              | 5      |

Repeats average out fold-partition luck to stabilize  $\text{CVR}^2$ ; larger samples need fewer repeats because a single cross-validation is already more stable. The repeated CV residuals also feed the variance model.

#### 7.7.4 Variance Model

**varmod.method**: Recommended: **lm** — Milborrow’s default: a robust linear variance model for prediction intervals (spread is estimated from noisy residuals, so a simple model generalizes better). If **lm** fails to converge on your data, try **rlm** or **const** (homoscedastic, always converges), or fall back to **earth** (also handles a nonlinear spread). It requires `nfold > 1` and `ncross > 1`.

#### 7.7.5 Earth Output File

After every successful fit, earthUI writes `<filename>_earth_output_<timestamp>.txt` to the output folder. This file contains the model terms, summary statistics ( $R^2$ ,  $\text{GRSq}$ ,  $\text{CVRsq}$ ), variance model details, and trace log. One file is created per fit, providing a cumulative record for comparing parameter configurations.

### 7.8 Settings Defaults

A radio button at the top of Section 4 controls which defaults are loaded:

- **Use last settings for input file** — restores the settings you used last time with this particular file
- **Use default settings** — applies your saved custom defaults
- **Earth defaults** — resets all parameters to factory values

Click **Save current as default** at the bottom of Section 4 to store the current earth parameters and Allowed Interactions matrix; the matching button in Section 3 stores the target, effective date, and predictor configuration. Both are saved per project and purpose in the project database.

## 8 Fitting the Earth Model

### 8.1 The Fit Button

Section 5 of the sidebar contains the **Fit Earth Model** button together with a **Random seed** field. The seed makes cross-validation fold assignments reproducible — a dropdown recalls the last 10 seeds used, and the seed used is recorded in the Earth Output tab after fitting. Before fitting, earthUI validates your configuration — a target must be selected and at least one predictor must be included. In appraisal/market modes, latitude and longitude columns are rounded, and sale age is computed from the effective date and contract date (if designated).

### 8.2 Fitting Modal & Trace Output

When you click Fit, a dark modal overlay appears with:

- **Header** — “Fitting Earth Model” with an elapsed timer counting seconds and an **Abort** button that cancels the background fit
- **Trace log** — a scrollable, monospace text area showing real-time output from the `earth()` function: dataset info, forward pass progress, cross-validation folds, and completion time
- **Color coding** — default lines in green, cross-validation lines in yellow, errors in red

earthUI fits models asynchronously using `callr::r_bg()`, which runs the computation in a separate R process. This keeps the application responsive during long-running fits. A 300ms polling observer reads output from the background process and streams it to the trace log.

When fitting completes, a “Done in X.Xs” message appears, a close button (X) is added to the modal, and the backdrop is removed so the result tabs can render on demand. The modal then auto-dismisses once the visible outputs finish rendering — or you can close it earlier.

On success, a green checkmark is appended to the Fit button. If an error occurs, the error message is displayed in the trace log and a notification appears.

### 8.3 Fit Log

Every fit (success or failure) writes a log file to your output folder:

- **Filename:** `<datafile>_earth_log_<YYYYMMDD_HHMMSS>.txt`
- **Contents:** timestamp, all trace output from the fitting process, and any error messages
- **Location:** the active project’s output folder

***Tip:** If `callr` is not installed, earthUI falls back to synchronous fitting with a simple progress bar. Install `callr` for the full trace-output experience.*

## 9 Downloading Data

After fitting, sidebar section 6 — titled “Download Estimated Target Variable(s) & Residuals” in general mode and “Download Estimated Sale Prices & Residuals” in appraisal and market modes — downloads an Excel file with predictions and diagnostics. This output is used in Step 7 (RCA) to assign a CQA (Condition-Quality-Appeal) rating to the subject property. The output is sorted by `residual_sf` and `cqa_sf` to help you assess where the subject falls in the ranking. If the model is good quality, then the properties should be ranked from lowest appealing to most appealing based on residual features that did not go into the regression. The middle value should be approximately 0, the lower half negative values and the upper half positive values. You should find the worst quality homes, or “fixers” near the bottom of the ranking and the nicest homes at the top. There will usually be exceptions for anomalies such as foreclosures, short sales, probate (inheritance related) sales, and quick sales needed for job change or other reasons. Investigation of anomalies usually turns up a pertinent reason for the price anomaly.

The button label adapts to the purpose mode:

- General/Market: **Download Output (Excel)**
- Appraisal: **Download Intermediate Output (Excel)**

The filename format is `<datafile>_modified_<YYYYMMDD_HHMMSS>.xlsx`.

### 9.1 Output Columns

For each target variable, the following columns are appended:

| Column                                     | Description                                                           |
|--------------------------------------------|-----------------------------------------------------------------------|
| <code>est_&lt;target&gt;</code>            | Model prediction, e.g. <code>est_sale_price</code> (1 decimal place)  |
| <code>residual</code>                      | Actual minus predicted (1 dp)                                         |
| <code>cqa</code>                           | Condition-Quality-Appeal score (2 dp, range 0–10)                     |
| <code>residual_sf</code>                   | Residual divided by living area (if designated, 1 dp)                 |
| <code>cqa_sf</code>                        | CQA calculated from ranking via <code>residual_sf</code> (2 dp)       |
| <code>&lt;variable&gt;_contribution</code> | Per-g-function contribution value (1 dp, one column per g-function)   |
| <code>basis</code>                         | Intercept value contribution, same for all properties (1 dp)          |
| <code>calc_residual</code>                 | Actual minus (basis + all contributions) — verification column (1 dp) |

For multi-target models, a `_<i>` suffix is added to distinguish columns for each target.

### 9.2 Column Ordering & Excel Formatting

The ranking columns are placed at the **leftmost position** in the output file, in this order: `residual_sf`, `cqa_sf`, `residual`, `cqa`. This makes it easy to scan the sorted list and evaluate where the subject falls in the CQA distribution.

These columns have Excel number formatting applied:

| Column                   | Format                    | Example |
|--------------------------|---------------------------|---------|
| <code>residual_sf</code> | Numeric, 2 decimal places | 12.34   |
| <code>cqa_sf</code>      | Numeric, 2 decimal places | 7.25    |
| <code>residual</code>    | Numeric, 0 decimal places | 15,200  |
| <code>cqa</code>         | Numeric, 2 decimal places | 6.80    |

### 9.3 CQA Scores

The CQA (Condition-Quality-Appeal) score ranks each row's residual against all other residuals. For a given row, the CQA is the percentage of rows with a smaller signed residual, multiplied by 10. This produces a 0–10 scale where:

- **High CQA** ( $\approx 9$ – $10$ ) — the property sold for much more than the model predicted (large positive residual)
- **Low CQA** ( $\approx 0$ – $1$ ) — the property sold for much less than predicted (large negative residual)
- **CQA  $\approx 5$**  — the residual is near the median of all residuals

When a `living_area` column is designated, `cqa_sf` provides the same ranking based on per-square-foot residuals.

### 9.4 Sorting & Subject Row

In appraisal and market modes, comparable rows are sorted by `residual_sf` descending (or `residual` if no living area is designated). The subject row (row 1) remains in position 1 and is not sorted. In general mode, rows are exported in their original order.

This sorting, combined with the leftmost ranking columns, allows the appraiser to quickly scan the comparables from most over-predicted to most under-predicted, and assess where the subject's assigned CQA score falls in that distribution.

Factor levels in the prediction data are aligned with the training data before calling `predict()`. Rows with unseen factor levels will produce NA predictions.

***Tip:** The `calc_residual` column should always equal `residual` (within rounding). If they differ significantly, it indicates a computation issue worth investigating.*

### 9.5 mgcvUI Auto-Export

On every successful fit with `degree`  $\leq 2$ , earthUI automatically saves the full result object as an `.rds` file to the project's output folder. The filename follows the pattern `<datafile>_earthUI_result_<YYYYMMDD_HHMMSS>.rds`.

This file can be loaded by the mgcv Post Processing Module — included in the earthUI package and started with `earthUI::launch_mgcv()` — using `readRDS()`. The GAM routine uses the earth model's knot locations and basis functions as starting points for GAM smooth terms, enabling a seamless transition from MARS to GAM modeling.

Models with `degree`  $> 2$  are skipped because mgcvUI only supports pairwise interactions. When earthUI is launched in Trilogy mode (from the Trilogy UI), an additional **Lock Model Output** section lets you lock a particular fit's `.rds` — together with its predictor basis and shared CQA settings — as the canonical earth model that glmnetUI and mgcvUI import.

### 9.6 glmnetUI Import

The same `.rds` file saved for the GAM routine can also be imported into the **elastic net Post Processing Module** — likewise included in the earthUI package and started with `earthUI::launch_glmnet()`. In the glmnet application, navigate to **Section 2: Import from earthUI** and use the Browse button to select the `.rds` file.

glmnetUI uses the standard `earth::model.matrix()` approach: the earth model's basis functions (hinges, interactions, and linear terms) become the columns of glmnet's design matrix. glmnet then performs regularized regression on these basis columns, selecting and shrinking them via lasso/elastic net. This combines earth's adaptive basis construction with glmnet's regularization.

### 9.6.1 Key Benefits

- **Interpretable interactions:** earth's hinge-based interactions (including 3-way terms) have clear geometric meaning, unlike glmnet's native cross-product interactions which are difficult to explain in court or audit settings.
- **Automatic nonlinearity:** earth's hinge functions capture nonlinear relationships that a standard glmnet model with raw predictors would miss.
- **Variable selection:** glmnet can further prune the earth basis, dropping hinge terms that don't contribute.

### 9.6.2 Important: Weight Column Must Match

#### Warning

If earthUI uses a weight column (e.g., to exclude the subject row with weight=0), the same weight column must be designated in glmnetUI's variable configuration. If the weight settings differ, the row counts will not align and the earth basis cannot be appended to glmnet's model matrix.

### 9.6.3 Workflow

1. Fit an earth model in earthUI (any degree).
2. The **.rds** result file is saved automatically to the project's output folder.
3. In glmnetUI, open **Section 2: Import from earthUI** and browse to the **.rds** file.
4. Verify that the **same data file** and **same weight column** are used in both apps.
5. Configure glmnet parameters in Section 5 (the Interaction Matrix is automatically disabled when an earthUI import is active).
6. Click **Fit Model**. The earth basis columns appear in the Equation, Coefficients, and Contributions tabs.

## 10 RCA Calculations & Downloading

The **Calculate RCA Adjustments & Download** button appears in sidebar section 7, visible only in appraisal mode after a model has been fitted. RCA (Residual Constraint Approach) produces market-derived, per-comparable adjustments relative to the subject property.

### 10.1 Opening the RCA Dialog

Clicking the button opens a small modal dialog with:

- **Score type** — radio buttons to choose **CQA** or **CQA per SF** (the SF option is available only when a `living_area` column is designated). If you choose “CQA per SF,” then based on the CQA score you assign the subject, its residual score will be its living area times the `residual_sf` that matches the given `CQA_SF` score.
- **CQA value** — input for the subject’s CQA score (0.00–10.00). This score represents where you believe the subject falls in the quality distribution of the comparables.
- **Generate** button — computes the RCA output and downloads it as Excel. The button stays disabled until a valid score is entered.

### 10.2 CQA Score Interpolation

When you click Generate, earthUI interpolates the subject’s residual from the comparable CQA/residual pairs:

1. The comparables’ CQA scores and residuals are sorted by CQA ascending
2. Linear interpolation (`stats::approx()`) maps your entered CQA value to a residual
3. If using CQA per SF, the per-SF residual is converted back to a total residual by multiplying by the subject’s living area
4. The subject’s estimated value is computed as: model prediction + interpolated residual

### 10.3 Output Columns

The RCA Excel file (`<datafile>_adjusted_<YYYYMMDD_HHMMSS>.xlsx`) includes all intermediate output columns plus:

| Column                                   | Description                                                   |
|------------------------------------------|---------------------------------------------------------------|
| <code>subject_value</code>               | Model prediction + interpolated residual (row 1 only)         |
| <code>subject_cqa</code>                 | The CQA score you entered (row 1 only)                        |
| <code>&lt;variable&gt;_adjustment</code> | Subject contribution minus comp contribution (per g-function) |
| <code>residual_adjustment</code>         | Subject residual minus comp residual                          |
| <code>net_adjustments</code>             | Sum of all adjustments (contribution + residual)              |
| <code>gross_adjustments</code>           | Sum of absolute values of all adjustments                     |
| <code>residual_pct</code>                | Residual adjustment as a fraction of the comp sale price      |
| <code>net_adj_pct</code>                 | Net adjustments as a fraction of the comp sale price          |
| <code>gross_adj_pct</code>               | Gross adjustments as a fraction of the comp sale price        |
| <code>adjusted_sale_price</code>         | Comp sale price + net adjustments                             |

The adjustment columns tell you, for each comparable, how much the model attributes the difference to each variable. `adjusted_sale_price` is the comparable’s sale price after applying all model-derived adjustments — a set of adjusted sale prices that should cluster around the subject’s estimated value.

#### 10.4 The RCA Adjustments Tab

Once the RCA output has been computed, the **RCA Adjustments** tab in the main panel displays histograms of the Residual Adj %, Net Adj %, and Gross Adj % distributions across the comparables (one set per target for multi-target models), giving a quick visual check on how tightly the adjustments cluster.

**Tip:** A CQA score of 5.00 places the subject at the median of the comparables. Scores above 5 indicate the subject is above average quality; below 5 indicates below average.

## 11 Sales Comparison Grid

### 11.1 Overview & Purpose

The **Sales Comparison Grid** is an Excel workbook generated in appraisal mode (sidebar section 8) after computing RCA adjustments. It presents the subject property alongside selected comparables in a structured grid format, with Excel formulas that automatically compute adjustments and an adjusted sale price for each comparable.

The grid is designed for the appraiser's workfile — it combines the regression-derived adjustments from the Earth model with editable cells where the appraiser can allocate the CQA residual to specific property features. The output filename is **SalesGrid\_<YYYYMMDD\_HHMMSS>.xlsx**.

### 11.2 Comp Selection Modal

Clicking “Generate Sales Grid & Download” opens a modal dialog where you select which comparables to include:

- **Recommended comps** are pre-checked. These are comparables with a gross adjustment percentage below 25% of sale price, sorted by sale age ascending (most recent sales first) and capped at 30.
- **Additional comps** are listed below, unchecked by default. These have gross adjustments of 25% or more but are still available for selection (up to 50 shown, sorted by gross adjustment percentage ascending).
- A maximum of **30 comps** can be selected, producing up to **10 sheets** (3 comps per sheet).

After confirming your selection, the selected comps are sorted by gross adjustment percentage ascending, so the most similar comparables appear on Sheet 1.

### 11.3 Grid Layout

Each sheet has a 20-column layout accommodating the subject property plus 3 comparables. The columns for each entity are:

| Entity    | Columns (5 per entity)                                                                 |
|-----------|----------------------------------------------------------------------------------------|
| Subject   | Label, Factual Value 1, Factual Value 2, Factual Value 3, Value Contribution (VC)      |
| Each Comp | Factual Value 1, Factual Value 2, Factual Value 3, Value Contribution (VC), Adjustment |

The rows from top to bottom are:

1. **Title** — sheet name (e.g., “Intermediate Sales Comparable Grid — Sheet 1 of 3”)
2. **Headers** — “Subject” and comp column headers with address
3. **Address** — full property address
4. **APN | MLS# | DOM | Subj.Prox** — parcel number, listing ID, days on market, and Haversine distance (miles) from subject
5. **Sales Price | Concess. | Net SP** — sale price, concessions, and Net Sale Price formula
6. **Regression Features** header row
7. **BASE VALUE** — the model intercept
8. **Date of Sale | OffMkt | OnMkt** — contract date, sale age, and DOM
9. **Grouped rows** (conditional) — Location, Site Size, and/or Age groups
10. **Model variable rows** — one row per predictor (excluding grouped variables)

11. **Blank separator row**
12. **Residual Features** header row
13. **CQA|Residual** — CQA score + remaining residual formula
14. **Residual feature rows** — named + blank rows for appraiser entry
15. **Total VC / Net Adjustment**
16. **Net Adjustment %**
17. **Gross Adjustment %**
18. **Adjusted Sale Price** — formula row
19. **Copyright** footer

#### 11.4 Sale Price, Concessions & Net SP

Row 5 shows three values for each comparable:

- **Sale Price** — the comparable’s sale price from the data
- **Concessions** — the value from the column designated as **concessions** (0 if not designated)
- **Net SP** — an Excel formula: Sale Price – Concessions

The subject column shows “N/A” for sale price (since it is unknown) and any concessions value if available.

#### 11.5 Grouped Rows (Location, Site, Age)

When certain special types are designated and those variables appear in the model, earthUI creates grouped rows that combine related variables:

**Loc: Long | Lat | Area** — Appears when any of **longitude**, **latitude**, or **area** are in the model. Shows factual values for each constituent variable, a combined Value Contribution (sum of the individual VCs), and for comps, an adjustment (subject combined VC – comp combined VC). Styled with light blue background on VC cells.

**Site Size | Dimensions** — Appears when **lot\_size** or **site\_dimensions** are in the model. Same structure as the Location group.

**Actual Age | Effective Age** — Appears when **actual\_age** or **effective\_age** are in the model. Same structure.

Variables consumed by grouped rows are **excluded from the individual model variable rows** below, preventing double-counting in the adjustment totals.

#### 11.6 Model Variable Rows

Below the grouped rows (or directly below BASE VALUE if no grouped rows), one row per remaining model predictor shows:

- **Factual values** for subject and each comp (the actual data values)
- **Value Contribution (VC)** — the g-function’s contribution to the predicted value for that row
- **Adjustment** (comps only) — subject VC minus comp VC

#### 11.7 CQA|Residual & Residual Feature Rows

The **CQA|Residual** row shows each property’s CQA score and contains a formula for the **remaining residual** — the portion of the residual not yet allocated to specific features. The formula is:

$$\text{Remaining Residual} = \text{Total Residual} - \sum (\text{Residual Feature VCs below})$$

Below this are **residual feature rows** — 5 named rows (Location, View/Appeal, Condition, Quality, Other) plus 6 blank rows. These are input cells where the appraiser enters value contributions for features not captured by the model. As the appraiser fills in values, the Remaining Residual formula automatically decreases.

For each comp, the adjustment column contains a formula: subject feature VC – comp feature VC.

### 11.8 Adjusted Sale Price Formula

The **Adjusted Sale Price** row contains Excel formulas:

- **Subject:** Sum of all Value Contribution cells from BASE VALUE through the last residual feature row. This represents the model's total prediction for the subject plus any appraiser-allocated residual features.
- **Comps:** Net SP + sum of all Adjustment cells from the first grouped or variable row through the last residual feature row. This gives the comparable's sale price after all regression-derived and appraiser-entered adjustments.

### 11.9 Sheet Protection & Editable Cells

Each sheet is **protected** to prevent accidental modification of formulas and data. The protection locks:

- All formula cells (Net SP, remaining residual, adjustments, adjusted sale price, etc.)
- All data cells (addresses, sale prices, factual values, value contributions from the model)
- Labels and headers

The only **unlocked** (editable) cells are the residual feature Value Contribution inputs — the cells under the CQA|Residual row where the appraiser enters breakdowns. These are styled with a light yellow background to indicate they are editable.

### 11.10 Working with the Grid in Excel

1. **Open the file** in Excel or a compatible spreadsheet application.
2. **Review the regression adjustments** — the model-derived adjustments are pre-populated and locked.
3. **Allocate the residual** — in the residual feature rows (yellow cells), enter your assessment of how much of the remaining residual is attributable to each feature (Location, View, Condition, Quality, etc.). The Remaining Residual formula will decrease as you allocate.
4. **Check the Adjusted Sale Price** — this formula automatically updates as you enter residual feature values.
5. **Multiple sheets** — if you selected more than 3 comps, navigate between sheets. Each sheet has the same subject in the left column with 3 different comps.

**Tip:** The goal is to allocate the CQA residual across the feature rows until the Remaining Residual is close to zero. This demonstrates that you can account for the difference between each comparable's regression estimate and its actual sale price.

## 12 Downloading Reports

After fitting a model, report generation is a two-step Quarto workflow: **Generate Quarto Report** (sidebar section 9 in appraisal mode, 7 otherwise) writes a self-contained report bundle, and **Convert Quarto Report** (section 10) renders it — or any other Quarto file — to the final formats.

### 12.1 Generate Quarto Report

Clicking **Generate Quarto Report** writes a self-contained Quarto bundle under the project's output folder at `<base>_qmd/`, containing the populated `.qmd` source, all pre-generated plot assets (PNG and PDF), the report data, and a Word reference template. No rendering happens at this step — the bundle can be edited by hand, combined with other Quarto sources via Quarto includes, or previewed standalone with `quarto preview`.

### 12.2 Convert Quarto Report

Section 10 provides a file picker (defaulting to the current project's most recently generated bundle — but you can browse to any `.qmd` file) and format checkboxes:

- **HTML** — an `.html` file with KaTeX math rendering, embedded images, and a table of contents. Uses the Flatly Bootstrap theme and Roboto Condensed font.
- **Word** — a `.docx` file suitable for editing and distribution. Includes a table of contents, page numbers, and uses a custom reference template for consistent styling.
- **PDF** — typeset with LuaLaTeX for professional-quality output. Paper size follows the locale setting (Letter or A4). Includes landscape pages for large interaction matrices and uses Roboto Condensed with Latin Modern Math for equations. The PDF option is offered only when a LaTeX distribution is detected.

Click **Convert** to render the selected formats. Rendering runs in the background — a modal dialog shows elapsed time and Quarto progress while the app remains responsive. A notification confirms the file location on completion.

All operations (model fitting, data downloads, RCA calculations, sales grid generation, and report rendering) are logged with start/end timestamps and elapsed times to `<datafile>_earthui_log.txt` in the output folder, for troubleshooting and performance monitoring.

### 12.3 Report Contents

Every generated report includes the following sections:

1. **Dataset Description** — number of observations, target variable(s), predictors used, categorical predictors
2. **Model Specification** — degree, cross-validation status, number of terms, predictors in model
3. **Allowed Interactions** (degree  $\geq 2$  only) — the interaction matrix with checkmarks, formatted for the output type
4. **Results Summary** —  $R^2$ , GRSq, GCV, RSS, and CV  $R^2$  (if cross-validation was enabled)
5. **Model Equation** — the complete Earth model equation in LaTeX notation
6. **Coefficients & Basis Functions** — table of all terms, their coefficients, and the basis functions that define them
7. **Variable Importance** — bar chart and ranked table of predictor importance scores
8. **g-Function Contributions** — plots for each g-function: line plots for univariate terms, 3D perspective and contour plots for bivariate interactions

9. **Correlation Matrix** — heatmap of predictor correlations
10. **Diagnostics** — Residuals vs Fitted, Normal Q-Q, and Actual vs Predicted plots
11. **ANOVA Decomposition** — table showing each basis function's contribution to RSS
12. **Earth Output** — raw text output from `earth::summary()`, including pruning pass details

For multi-response models, sections 4–8 and 10 are repeated for each target variable.

**Tip:** Reports require the *quarto* R package and a Quarto installation. For PDF output, a LaTeX distribution must also be installed (e.g., via `tinytex::install_tinytex()`) — otherwise the PDF checkbox is hidden.

## 13 The Post Processing Modules: Elastic Net and GAM

The earthUI package ships two Post Processing Modules — an elastic net routine built on `glmnet` and a GAM routine built on `mgcv`. Each is treated in depth by its own article in this issue; this section explains what they are *for* in the earthUI workflow and how the three routines cooperate.

### 13.1 Why Support Fits?

The earth model produces the value conclusion. The Post Processing Modules exist to **corroborate** it: two additional estimates of the same subject, produced by methodologically different engines, fitted to the same project data. When an elastic net re-estimation of the earth basis and an independent GAM smooth arrive at materially the same value, the appraiser’s conclusion no longer rests on a single model’s assumptions — a meaningful advantage in review and litigation contexts. When they *disagree*, that disagreement is itself diagnostic, usually pointing at thin data, an over-flexible term, or an outlier comp worth revisiting. Neither module outranks the earth model; they are supporting evidence, deliberately presented without weighting.

### 13.2 Launching and Shared State

Each routine is a separate local Shiny application:

- `earthUI::launch_glmnet()` — the elastic net application, on port 7879.
- `earthUI::launch_mgcv()` — the GAM application, on port 7880.

All three applications read the same project tree and the same settings database, so the project you have open in earthUI is immediately available in the modules — along with the carry-forward fields earthUI saves for it: the Response (target), the Effective Date, the RCA subject-CQA type and value, and the living-area designation. You do not re-enter these; the modules read them so the same values flow through all three fits.

### 13.3 The Elastic Net Routine

The `glmnet` routine imports the earth model’s basis functions — its hinges, interactions, and linear terms — as the columns of a design matrix and re-estimates them under lasso/elastic-net regularization (see “`glmnetUI` Import” above for the mechanics and the weight-column caveat). The result keeps earth’s interpretable, geometrically meaningful terms while letting the regularizer shrink or drop the ones that do not earn their keep. See the `glmnetUI` article in this issue for the full treatment.

### 13.4 The GAM Routine

The `mgcv` routine goes the other direction: rather than reusing earth’s basis, it fits fresh penalized smooth terms, using the earth model’s knot locations as starting points (see “`mgcvUI` Auto-Export” above). Because the GAM re-derives the functional forms under a different smoothing paradigm, its agreement with the earth model is genuinely independent corroboration rather than an echo. Earth models with  $\text{degree} \leq 2$  are supported. See the `mgcvUI` article in this issue for the full treatment.

### 13.5 A Practical Workflow

1. Fit and refine the **earth model first**, in earthUI, until you are satisfied with it. Every successful fit ( $\text{degree} \leq 2$ ) auto-exports the `.rds` the modules consume.

2. Open the **glmnet** and **mgcv** routines against the same project, import the earth result, and fit. Work in one routine at a time — fits are long-running and the routines coordinate through the shared project, so the sensible rhythm is sequential, not parallel.
3. Iterate freely. Each application keeps its results while you switch between them, and nothing forces you to generate reports along the way.
4. Only when all three results satisfy you, return to each routine and generate its Quarto report. Each method registers its fit and value conclusion with the project's trilogy record, keeping the three reports traceable to one comparative run.

**Tip:** *The modules are optional at every level: earthUI's own workflow — fitting, RCA, Sales Grids, reports — is complete without them. Reach for them when a conclusion warrants corroboration: a difficult subject, a thin market segment, or an assignment likely to face review.*

## 14 Demo Dataset: Appraisal\_1.csv

earthUI includes a demo MLS dataset for exploring the appraisal workflow. Load it programmatically with:

```
demo_file <- system.file("extdata", "Appraisal_1.csv", package = "earthUI")
df <- import_data(demo_file)
```

Or, in the Shiny app, copy it into a project's `in/` folder (or attach it as the initial data file when creating the project) and select it in Section 2 (Import Data).

### 14.1 Description

The file contains 1,501 residential sales plus 1 subject property in row 1 (1,502 rows in all) from a simulated MLS export. The data represents single-family home sales in a multi-area market with a range of property sizes, ages, and locations.

This is not real data, but is based on a realistic neighborhood in Northern California. All identification information has been altered or removed.

## 14.2 Columns

| Column              | Type      | Special Type  | Description                                              |
|---------------------|-----------|---------------|----------------------------------------------------------|
| weight              | numeric   | —             | Observation weight (0 = exclude from fitting)            |
| id                  | numeric   | display_only  | Internal record ID                                       |
| property_id         | numeric   | display_only  | MLS property identifier                                  |
| listing_id          | character | display_only  | MLS listing number                                       |
| parcel_number       | character | display_only  | County assessor parcel number (APN)                      |
| street_address      | character | display_only  | Property address                                         |
| city_name           | character | display_only  | City                                                     |
| postal_code         | character | display_only  | ZIP code                                                 |
| county_name         | character | display_only  | County                                                   |
| contract_date       | Date      | contract_date | Sale contract date (computes sale_age)                   |
| sale_age            | numeric   | —             | Days from contract date to effective date (pre-computed) |
| coe_date            | Date      | display_only  | Close of escrow date                                     |
| listing_status      | character | display_only  | Listing status (e.g., "Sold")                            |
| sale_price          | numeric   | (target)      | Sale price — response variable                           |
| rent                | numeric   | —             | Monthly rent (for multi-target models)                   |
| list_price          | numeric   | display_only  | Listing price                                            |
| original_list_price | numeric   | display_only  | Original listing price                                   |
| living_sqft         | numeric   | living_area   | Gross living area in square feet                         |
| beds_total          | integer   | —             | Number of bedrooms                                       |
| baths_total         | numeric   | —             | Total bath count (e.g., 2.5 = 2 full + 1 half)           |
| lot_size            | numeric   | lot_size      | Lot size in square feet                                  |
| area_id             | integer   | area          | MLS area identifier                                      |
| area_text           | character | display_only  | Area name                                                |
| age                 | numeric   | actual_age    | Property age in years                                    |
| year_built          | integer   | display_only  | Year of construction                                     |
| latitude            | numeric   | latitude      | Latitude (rounded to 3 dp for model)                     |
| longitude           | numeric   | longitude     | Longitude (rounded to 3 dp for model)                    |
| latitude6           | numeric   | display_only  | Full-precision latitude                                  |
| longitude6          | numeric   | display_only  | Full-precision longitude                                 |
| garage_spaces       | integer   | —             | Number of garage bays                                    |
| fp_count            | integer   | —             | Number of fireplaces                                     |
| no_of_stories       | numeric   | —             | Number of stories                                        |
| style               | character | —             | Architectural style                                      |
| view                | character | —             | View type (e.g., "Neighborhood", "Hills")                |
| days_on_market      | integer   | dom           | Days on market                                           |
| listing_date        | Date      | listing_date  | Listing date                                             |
| sale_concessions    | numeric   | concessions   | Seller concessions                                       |

## 14.3 Suggested Quick Start

1. Launch earthUI: `earthUI::launch()`
2. Create a project with Purpose **For Appraisal** in the New Project wizard, attaching `Appraisal_1.csv` as the initial data file
3. Select `Appraisal_1.csv` in Section 2 (Import Data)
4. Select `sale_price` as the target
5. Assign special types as shown in the table above
6. Include predictors: `sale_age`, `living_sqft`, `baths_total`, `lot_size`, `area_id` (as factor), `age`, `latitude`, `longitude`, `garage_spaces`
7. Set degree to 1, click **Fit Earth Model**
8. Download intermediate output (Step 6), review the CQA ranking
9. Compute RCA adjustments (Step 7) with a CQA score of  $\sim 5.00$
10. Generate the Sales Comparison Grid (Step 8)

**Tip:** *The demo dataset is a good way to explore all of earthUI's features before importing your own MLS data. The suggested predictors above typically produce a model with  $R^2$  above 0.85.*

## 15 Experimental: Prolog Text Extraction & Derivation Rules

Version 0.10.0 introduces an **optional and genuinely experimental** facility for extracting model-ready features from the descriptive text columns of an MLS export — the “Public Remarks” column above all, but also agent remarks and comma-delimited feature lists such as room features or existing structures. The extraction engine is Prolog, and in particular **Definite Clause Grammars (DCGs)**: the rules that recognize phrases like “move in ready,” “3 car garage,” or “owned solar” in remark text are written as Prolog grammar rules.

This feature is still in development. It can be used *partially* for extraction today — the bundled grammar already recognizes a useful set of features — but adapting it to your own MLS’s vocabulary requires you to read and write Prolog, and DCGs in particular. If that is not you, skip this chapter entirely: the feature is off by default, and the rest of earthUI works exactly the same without it.

### Warning

This is an experimental feature under active development. Its extraction grammar is neither complete nor validated for production appraisal work — extracted columns must be reviewed against the source text before being relied upon in a model. Using the feature effectively requires working knowledge of Prolog and DCGs. Everything it does is additive and reversible; if anything misbehaves, turn the Settings toggle off and continue normally.

### 15.1 What It Does

MLS listings carry a great deal of value-relevant information in free text and in delimited list fields. This optional step turns that text into ordinary columns your regression can use, via two special column roles (Chapter 6):

- **remarks columns** are parsed by the Prolog DCG grammar into one feature column per extracted attribute. The column prefix is built from the first letters of the source column’s name: `public_remarks` yields `pr_*` columns, `agent_remark` yields `ar_*`, and so on.
- **list columns** (e.g. `existing_structures = "Shed, Workshop"`) are split into one-hot `<col>_<value>` TRUE/FALSE columns. This part is pure R and needs no Prolog at all.

The bundled grammar (`inst/prolog/mls_remarks.pl`) currently extracts, among other things: a graded condition tier with a 1–5 `condition_score`; view flags and view types; kitchen and bath remodel flags; hardwood and other flooring; an upgrade list and count; garage spaces and garage area; pool, spa, and other outdoor features; fireplace count; outbuildings (shed, barn, workshop, casita, guest house, kennel, etc.) with sizes parsed from dimensions like “40x50”; RV parking and storage; solar ownership versus lease, battery storage; lot size (including acre-to-square-foot conversion); sale-context price amounts tied to a subject (price reduction, HOA dues, solar lease or payoff, Mello-Roos); location cues; and a highlights/concerns sentiment score in  $[-1, 1]$ .

### 15.2 Requirements: vProlog (Trealla), Not SWI-Prolog

Remarks parsing runs on the **vProlog** R package — a self-contained Trealla Prolog engine. It is not on CRAN; install it from the author’s r-universe repository as shown in “Installing earthUI and vProlog” at the start of this article (prebuilt binaries for Windows and macOS, source at [github.com/wcraytor/vProlog](https://github.com/wcraytor/vProlog)). No system Prolog installation (SWI-Prolog or otherwise) is required or used. If vProlog is absent, remarks parsing is skipped with a note while `list` splitting still runs. The helper `vProlog_available()` reports whether the engine and the bundled grammar are both present.

### 15.3 Enabling and Running It

1. Open **Settings** (gear) and tick **Enable Prolog / list column processing**. The toggle is off by default, carries an “advanced feature — for users with Prolog expertise” note, is remembered per project, and is shown in the Project Information dialog.
2. In the predictor table, set the **Special** role of the relevant columns to **remarks** or **list** (multiple columns may carry each role).
3. Use the **Execute Prolog Processing** section that sits between Sections 4 and 5 of the sidebar. It offers two steps:

**1. Expand lists & remarks** — runs the DCG grammar over each **remarks** column and one-hot-splits each **list** column. Expansion is *incremental*: only columns designated since the last run are processed, and the button is disabled once everything designated has already been expanded. A progress bar tracks the parse.

**2. Execute derivation rules** — applies your own Prolog `derive/2` rules over the expanded columns (see below). Rules are syntax-checked and dry-run validated before they run.

### 15.4 Derivation Rules

For each data row, earthUI asserts the row’s columns as `cell(Name, Value)` facts and collects every solution of your `derive(Col, Val)` rules — each distinct `Col` becomes a new column, and the *first* value found per column wins, so a fallback clause may follow. Helper predicates (`count_true/2`, `count_false/2`, `count_true_prefix/2`, `cell_or/3`, `is_na/1`) support aggregation and NA handling, and `drop/1` facts discard columns after the rules run. For example:

```
% Merge a collinear cluster of solar flags into one boolean.
derive(solar_any, B) :-
  ( cell(pr_has_solar, true)
  ; cell(pr_solar_owned, true)
  ; cell(pr_has_battery, true) ) -> B = true ; B = false.

% Count amenities instead of OR-ing them (numeric column).
derive(amenity_count, N) :- count_true_prefix(pr_has_, N).

% Threshold -> boolean, with a fallback clause.
derive(big_outbuilding, true) :- cell(pr_outbuilding_sqft, S), S >= 1000.
derive(big_outbuilding, false).

% Discard a raw granular column once consolidated.
drop(pr_price_solar_lease).
```

Rules live in a per-project `<project>_rules.pl` file, seeded from a fully commented template, and are edited via **Edit derivation rules...** in a floating, draggable code editor that can also be docked into the sidebar. A **Check** button validates the rules (syntax scan, load check, and a dry run against row 1 of your data) without executing them.

### 15.5 Versioned Active Data File

After either step, the augmented data frame is written to the project’s input folder as `<original_name>_<YYMMDD_HHMMSS>.xlsx` and becomes the project’s **active** data file — reopening the project loads the augmented data, so expansions are never recomputed, and the same file is what glmnetUI and mgcvUI consume. The raw remarks source columns are automatically held out of the model; the generated columns are *not* auto-included as predictors. Text parsing can produce many columns, so flag the ones you do not (yet) want in the regression as **Latent**, or ×-disable them (Chapter 6).

## 15.6 Programmatic Interface & Bundled Files

The whole pipeline is exported for batch use: `execute_prolog_processing(df, remarks_cols, list_cols, rules_text)` orchestrates the passes, built on `remarks_to_columns()`, `split_list_column()`, `apply_user_rules()`, and `validate_rules()`. The bundled Prolog sources under `inst/prolog/` are `mls_remarks.pl` (the DCG grammar and feature aggregation), `mls_analyze.pl` (a file-IO driver for batch parsing), `mls_rules.pl` (the helper predicates available to derivation rules), and `climate_regions_ca.pl` (below).

## 15.7 Related Experiment: Climate-Region Priors

A companion experimental facility addresses *thin-data* features — ones the regression cannot estimate because they appear in only a handful of comps (a swamp cooler, say). Three R helpers — `climate_region_for()`, `market_area_profile()`, and `climate_feature_priors()` — classify a project's location (county + city) into one of seven buyer-recognized California climate regions (a practical reduction of the California Energy Commission's 16 building-climate zones) and return ordinal contributory-value priors for HVAC and cooling features (heating, A/C, swamp cooler, whole-house fan, filtration) by region. The Prolog module `climate_regions_ca.pl` mirrors the same classification for use inside derivation rules; both sides are generated from one canonical R source so they cannot drift. The priors are intended to *inform* feature valuation and to serve as a plausibility check on estimated coefficients — region assignments (particularly for counties spanning several regions) should be reviewed by the appraiser.

### 15.7.1 Beyond California: Jurisdiction Packs and Market-Area Modules

California is the first jurisdiction, not the boundary. earthUI is used internationally, and the climate-prior knowledge is being restructured as **jurisdiction packs**: self-contained, data-shaped fact tables (zone definitions, admin-unit assignments, feature priors) keyed to the project system's country and admin-level codes, so a pack can describe a U.S. state, a German Land, or a small country outright. A companion layer of **market-area modules** captures local vocabulary and refined priors for a named market area.

Both are designed to be authored *in the field* by Valuation Engineers and appraisers, in their own language — and, realistically, with LLM assistance: the world does not have many Prolog programmers, so field artifacts are deliberately tables rather than programs. The pack format is a documented schema with a complete worked example (California itself), expressly so that an assistant such as Claude can draft a valid pack from a plain-language description of a market; earthUI generates the underlying Prolog from the pack, and a blocking validator (schema, coverage, referential-integrity, and dry-run checks) must pass before a pack can be enabled. Every field-authored pack carries mandatory provenance — the author of record, date, and the authority cited — so the responsible professional, not the drafting tool, stands behind the knowledge, and reports can cite the exact pack and version used.

**Tip:** A practical pattern: expand the remarks, consolidate the granular *pr\_\** flags into a handful of derived counts or booleans with *derive/2* rules, *drop/1* the raw flags you consolidated, and keep only the derived columns as candidate predictors. This keeps the predictor list short and the collinearity manageable.

## References

Friedman, Jerome H. (1991). "Multivariate Adaptive Regression Splines". In: *The Annals of Statistics* 19.1, pp. 1–141. DOI: [10.1214/aos/1176347963](https://doi.org/10.1214/aos/1176347963).

- Milborrow, Stephen (Oct. 1, 2024). *earth: Multivariate Adaptive Regression Splines*. R package. URL: <https://CRAN.R-project.org/package=earth>.
- (n.d.[a]). *Notes on the earth Package*. URL: <http://www.milbo.org/doc/earth-notes.pdf>.
- (n.d.[b]). *Variance Models in earth*. URL: <http://www.milbo.org/doc/earth-varmod.pdf>.

# glmnetUI: An Interactive Interface for the glmnet (Lasso and Elastic Net) Package

William Bert Craytor

Vol. 1, Issue 1 · Summer 2026

Published July 1, 2026

DOI: [10.67330/ja8s2k39](https://doi.org/10.67330/ja8s2k39)

© 2026 William Bert Craytor. Published by Pacific Vista Net.

This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

**Abstract.** glmnetUI is a graphical user interface for the R `glmnet` package, which fits regularized generalized linear models via the Lasso, ridge, and elastic-net penalties. It offers three purpose modes—general predictive modeling, real-estate appraisal, and market-area analysis—and guides the user through data import, model configuration, cross-validated fitting, coefficient paths, and downloadable reports. This article documents glmnetUI’s data-format requirements, modeling workflow, output displays, and complete feature reference.

**Keywords:** Lasso, ridge regression, elastic net, regularized regression, glmnet, R, graphical user interface, real estate valuation

**Publisher’s Note (July 2026):** Shortly before press, glmnetUI was consolidated into earthUI (version 0.11.0 and later), which now provides the elastic-net workflow described here alongside the MARS and GAM workflows. glmnetUI remains available in its final released form (0.5.0) at <https://github.com/wcraytor/glmnetUI>; future development continues in earthUI. This article applies to glmnetUI as released.

## Contents

|          |                                             |           |
|----------|---------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                         | <b>3</b>  |
| 1.1      | History and Background                      | 3         |
| 1.1.1    | The Lasso                                   | 3         |
| 1.1.2    | Ridge Regression and Elastic Net            | 4         |
| 1.1.3    | The glmnet Package                          | 4         |
| 1.1.4    | Comparison with MARS (earth) and GAM (mgcv) | 4         |
| 1.1.5    | Recommended Workflow                        | 5         |
| 1.2      | What Is glmnetUI?                           | 6         |
| 1.3      | Three Purpose Modes                         | 6         |
| 1.4      | Real Estate-Specific Features               | 6         |
| 1.5      | Getting Started                             | 7         |
| <b>2</b> | <b>MLS Input Data Requirements</b>          | <b>8</b>  |
| 2.1      | File Format & Structure                     | 8         |
| 2.2      | Required Columns for Appraisal Mode         | 8         |
| 2.3      | Special Column Naming Conventions           | 9         |
| 2.4      | Data Quality & Completeness                 | 9         |
| 2.5      | Subject Row Placement                       | 9         |
| <b>3</b> | <b>General Purpose Mode</b>                 | <b>10</b> |
| 3.1      | Overview                                    | 10        |
| 3.1.1    | Skip First Row                              | 10        |
| 3.1.2    | Settings Per Purpose                        | 10        |
| 3.2      | The Sidebar Workflow                        | 10        |

|          |                                           |           |
|----------|-------------------------------------------|-----------|
| 3.3      | Projects                                  | 11        |
| 3.4      | Import from earthUI (Optional)            | 11        |
| 3.5      | Main Panel Tabs                           | 12        |
| 3.6      | Settings Persistence                      | 12        |
| 3.7      | Dark Mode                                 | 12        |
| 3.8      | Locale & Regional Settings                | 13        |
| 3.8.1    | Saving Defaults                           | 13        |
| <b>4</b> | <b>Appraisal Mode</b>                     | <b>14</b> |
| 4.1      | Subject Row Handling                      | 14        |
| 4.2      | Effective Date & Sale Age                 | 14        |
| 4.3      | Special Column Designations               | 14        |
| 4.4      | RCA Adjustments Overview                  | 14        |
| <b>5</b> | <b>Market Area Analysis Mode</b>          | <b>15</b> |
| 5.1      | Differences from Appraisal Mode           | 15        |
| 5.2      | When to Use Market Mode                   | 15        |
| <b>6</b> | <b>Variable Selection</b>                 | <b>16</b> |
| 6.1      | Target Variable                           | 16        |
| 6.2      | The Predictor Table                       | 16        |
| 6.3      | Data Type Detection & Overrides           | 16        |
| 6.4      | Expected Signs & Enforcement              | 16        |
| 6.5      | Special Column Types Reference            | 17        |
| <b>7</b> | <b>Parameter Selection</b>                | <b>18</b> |
| 7.1      | Alpha (Mixing Parameter)                  | 18        |
| 7.2      | Lambda Selection                          | 18        |
| 7.3      | Family                                    | 18        |
| 7.4      | Standardize                               | 19        |
| 7.5      | Sign Constraints                          | 19        |
| 7.6      | Relaxed Lasso                             | 19        |
| 7.7      | Random Seed                               | 19        |
| 7.8      | Interaction Matrix                        | 19        |
| 7.9      | Block from Main Effect (Interaction Only) | 20        |
| 7.10     | Advanced Parameters                       | 20        |
| 7.11     | Settings Defaults                         | 20        |
| <b>8</b> | <b>Fitting the Model</b>                  | <b>22</b> |
| 8.1      | The Fit Button                            | 22        |
| 8.2      | What Happens During Fitting               | 22        |
| <b>9</b> | <b>Result Tabs</b>                        | <b>23</b> |
| 9.1      | Data Preview                              | 23        |
| 9.2      | Equation                                  | 23        |
| 9.3      | Correlation                               | 23        |
| 9.4      | Summary                                   | 23        |
| 9.5      | Coefficients                              | 23        |
| 9.6      | Variable Importance                       | 24        |
| 9.7      | Contributions                             | 24        |
| 9.8      | ANOVA                                     | 24        |
| 9.9      | Diagnostics                               | 24        |

|           |                                                  |           |
|-----------|--------------------------------------------------|-----------|
| 9.10      | Glmnet Output . . . . .                          | 25        |
| 9.11      | Report . . . . .                                 | 25        |
| <b>10</b> | <b>Downloading Data</b>                          | <b>26</b> |
| 10.1      | Output Columns . . . . .                         | 26        |
| 10.2      | CQA Scores . . . . .                             | 26        |
| <b>11</b> | <b>RCA Calculations &amp; Downloading</b>        | <b>27</b> |
| 11.1      | Opening the RCA Dialog . . . . .                 | 27        |
| 11.2      | CQA Score Interpolation . . . . .                | 27        |
| 11.3      | Output Columns . . . . .                         | 27        |
| <b>12</b> | <b>Sales Comparison Grid</b>                     | <b>28</b> |
| 12.1      | Comp Selection Dialog . . . . .                  | 28        |
| 12.2      | Workbook Layout . . . . .                        | 28        |
| 12.3      | Sheet Protection & Residual Cells . . . . .      | 28        |
| <b>13</b> | <b>Downloading Reports</b>                       | <b>29</b> |
| 13.1      | The Two-Step Quarto Workflow . . . . .           | 29        |
| 13.2      | Report Formats . . . . .                         | 29        |
| 13.3      | Report Contents . . . . .                        | 29        |
| <b>14</b> | <b>Comparison with earthUI</b>                   | <b>31</b> |
| 14.1      | Key Differences . . . . .                        | 31        |
| 14.2      | Shared Features . . . . .                        | 31        |
| 14.3      | When to Use Which . . . . .                      | 31        |
| <b>15</b> | <b>Demo Dataset: Appraisal_1.csv</b>             | <b>33</b> |
| 15.1      | Description . . . . .                            | 33        |
| 15.2      | Columns . . . . .                                | 33        |
| 15.3      | Suggested Quick Start . . . . .                  | 33        |
| <b>16</b> | <b>System Requirements &amp; Troubleshooting</b> | <b>35</b> |
| 16.1      | Supported Platforms . . . . .                    | 35        |
| 16.2      | Optional Dependencies . . . . .                  | 35        |
| 16.3      | Platform Notes . . . . .                         | 35        |
| 16.4      | Graceful Degradation . . . . .                   | 35        |
| 16.5      | Troubleshooting . . . . .                        | 35        |

## 1 Introduction

### 1.1 History and Background

#### 1.1.1 The Lasso

The **Lasso** (Least Absolute Shrinkage and Selection Operator) was introduced by Robert Tibshirani in 1996. The name is a deliberate play on the cowboy’s lasso — a rope that constrains and selects. Tibshirani chose it because the method literally “lassos” the coefficients, constraining their absolute values and pulling some to exactly zero, thereby selecting which variables remain in the model.

Tibshirani’s insight was to add an  $L_1$  penalty (sum of absolute values of coefficients) to the least squares objective. Unlike the  $L_2$  penalty of ridge regression (Hoerl & Kennard, 1970), which

shrinks coefficients toward zero but never reaches it, the  $L_1$  penalty produces *sparse* solutions where many coefficients are exactly zero. This makes the Lasso both a regularization method and a variable selection method.

### 1.1.2 Ridge Regression and Elastic Net

**Ridge regression** was proposed by Arthur Hoerl and Robert Kennard in 1970. It adds an  $L_2$  penalty (sum of squared coefficients) which shrinks all coefficients proportionally but keeps all variables in the model. Ridge is effective when predictors are correlated (multicollinearity) but does not simplify the model.

The **elastic net**, introduced by Hui Zou and Trevor Hastie in 2005, combines both penalties. The mixing parameter  $\alpha$  controls the blend:  $\alpha = 1$  is pure Lasso,  $\alpha = 0$  is pure ridge, and values in between give a compromise. The elastic net overcomes a limitation of the Lasso: when predictors are highly correlated, the Lasso tends to select one and ignore the rest, whereas the elastic net groups correlated predictors together.

### 1.1.3 The glmnet Package

The R package **glmnet** was developed by Jerome Friedman, Trevor Hastie, and Robert Tibshirani at Stanford University. It implements elastic net regularization for generalized linear models using an extremely efficient coordinate descent algorithm (Friedman, Hastie, and Tibshirani 2010), since extended to Cox proportional-hazards models (Simon et al. 2011) and to all generalized linear model families (Tay, Narasimhan, and Hastie 2023). The package handles gaussian (linear), binomial (logistic), poisson, and other GLM families.

Key features of **glmnet** include:

- **Regularization path:** Fits the model across a grid of lambda values in a single pass, much faster than fitting separately at each lambda.
- **Cross-validation:** `cv.glmnet()` automatically selects the optimal regularization strength via  $k$ -fold cross-validation.
- **Coefficient bounds:** Supports upper and lower limits on individual coefficients (sign constraints).
- **Relaxed lasso:** Refits the model on the selected variables with reduced or no penalization, reducing bias.

### 1.1.4 Comparison with MARS (earth) and GAM (mgcv)

The **earth** application and its two Post Processing Modules (**glmnet** and **mgcv**) use different modeling engines. The following table summarizes the key differences:

| Feature             | glmnet (Elastic Net)                                          | earth (MARS)                                                   | mgcv (GAM)                                                         |
|---------------------|---------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------------------|
| Model type          | Linear (with regularization)                                  | Piecewise linear (adaptive splines)                            | Smooth nonlinear (penalized splines)                               |
| Nonlinearity        | Only through interactions or basis expansion                  | Automatic via hinge functions with data-driven knots           | Automatic via smooth functions (thin plate, cubic, etc.)           |
| Variable selection  | Built-in via $L_1$ penalty (Lasso)                            | Built-in via forward/backward stepwise with GCV pruning        | Optional via double-penalty shrinkage ( <code>select=TRUE</code> ) |
| Interactions        | Pairwise cross-products (hard to interpret)                   | Hinge-based products (interpretable, up to degree 3)           | Tensor product smooths ( <code>te()</code> , <code>ti()</code> )   |
| Interpretability    | Coefficients are linear weights; easy to explain individually | g-functions show piecewise linear partial effects per variable | Smooth partial effect curves; very intuitive                       |
| Best for            | High-dimensional data, variable selection, small samples      | Automatic nonlinearity and interaction detection               | Smooth nonlinear relationships, flexible modeling                  |
| RCA adjustments     | Linear per-variable adjustments                               | Piecewise linear adjustments via g-functions                   | Smooth adjustments via partial effects                             |
| Overfitting control | Cross-validated lambda                                        | GCV-based pruning                                              | REML/GCV smoothing parameter estimation                            |

**MARS** (Multivariate Adaptive Regression Splines) was introduced by Jerome Friedman in 1991. It builds piecewise linear models by adaptively selecting hinge functions  $\max(0, x - k)$  and their knot positions from the data. The R implementation is the **earth** package by Stephen Milborrow. MARS excels at automatically discovering nonlinear relationships and interactions without requiring the user to specify them. Its g-functions (grouped basis terms per variable) provide highly interpretable partial effect curves.

**GAMs** (Generalized Additive Models) were formalized by Hastie and Tibshirani (1990). The R implementation **mgcv** by Simon Wood provides penalized regression splines with automatic smoothness selection via REML or GCV. GAMs produce smooth partial effect curves that are intuitive to visualize and explain. When **earthUI** exports knot locations to **mgcvUI**, the **earth**-derived knots serve as starting points for GAM smooth terms, combining MARS's adaptive knot placement with GAM's smooth estimation.

### 1.1.5 Recommended Workflow

For real estate appraisal and similar applications requiring interpretable, defensible models:

1. **Start with earthUI** to discover the important variables, nonlinear relationships, and interactions in your data. Earth's automatic basis selection and g-functions provide excellent initial insights.
2. **Refine with mgcvUI** if you want smoother partial effects. Import earthUI's knots into mgcvUI for a seamless transition from piecewise linear to smooth models.
3. **Use glmnetUI** when you need aggressive variable selection (Lasso), when the number of predictors approaches or exceeds the number of observations, or when you want a purely linear model with regularization for defensibility.

## 1.2 What Is glmnetUI?

glmnetUI is a graphical user interface for the R `glmnet` package. It runs as a local Shiny application — there is no login, no server, and no accounts. You launch it from R, import a dataset (CSV or Excel), configure your model, and fit it interactively.

As of the current release, glmnetUI is no longer a separately installed package: it ships *inside* the earthUI package as its elastic net Post Processing Module, started with `earthUI::launch_glmnet()`. The application itself — and everything described in this article — is unchanged; only the installation and launch command differ. The three routines (`earth`, `glmnet`, `mgcv`) share one project tree, one settings database, and one copy of the supporting infrastructure.

The application provides a complete workflow: data import, variable configuration, model fitting, diagnostic plots, variable importance, model equations, and downloadable reports in Word, PDF, or HTML format.

**Elastic net regression** combines two regularization techniques:

- **Ridge regression** ( $\alpha = 0$ ): Shrinks all coefficients toward zero proportionally. Good when you believe all variables are relevant.
- **Lasso regression** ( $\alpha = 1$ ): Can shrink coefficients exactly to zero, performing automatic variable selection. Produces simpler, more interpretable models.
- **Elastic net** ( $0 < \alpha < 1$ ): A blend of both approaches. Useful when predictors are correlated.

## 1.3 Three Purpose Modes

Every glmnetUI project has a **purpose**, chosen when you create it (Section 1, *Project*). The active project's purpose determines which tools and interface elements appear; to work in a different mode, create or open a project with that purpose. The three modes are:

- **General** — Elastic net regression for any type of population or dataset. This is the default mode. It provides the full regularized regression workflow without any domain-specific additions.
- **For Appraisal** — Elastic net regression tailored for real estate appraisal. Adds features specific to single-property valuation, including subject property handling, special column designations, and Residual Constraint Approach (RCA).
- **Market Area Analysis** — Elastic net regression tailored for market area studies. Adds features for analyzing groups of properties in a defined market, with optional subject row exclusion.

In all three modes, the core modeling engine is identical — you are always fitting an elastic net (`glmnet`) model. The purpose setting controls which additional tools and interface elements are available.

## 1.4 Real Estate–Specific Features

When either **For Appraisal** or **Market Area Analysis** is selected, glmnetUI activates several features designed for real estate analysis:

- **Special column designations** — Each predictor can be tagged with a special role such as `contract_date`, `dom`, `concessions`, `latitude`, `longitude`, `living_area`, `lot_size`, `actual_age`, `effective_age`, `area`, `site_dimensions`, or `display_only`. These designations control how the column is handled during fitting and output.
- **Rounding of latitude and longitude** — Columns designated as latitude or longitude are automatically rounded to 3 decimal places to prevent overfitting.

- **Sale Age column** — When a column is designated as `contract_date` and an Effective Date is provided, glmnetUI computes a `sale_age` column (days between sale date and effective date) and substitutes it as a predictor.
- **RCA computations (appraisal only)** — In appraisal mode, after fitting the model glmnetUI can compute Residual Constraint Approach (RCA) output, which produces per-comparable adjustments, net/gross adjustment summaries, and an adjusted sale price for the subject property.

***Tip:** The General purpose mode works for any dataset — financial, scientific, engineering, or real estate. The appraisal and market modes simply add convenience features for real estate professionals.*

## 1.5 Getting Started

To use glmnetUI:

1. **Install earthUI** in R: `install.packages("earthUI")` — the glmnet routine is included. (See “Installing earthUI and vProlog” in the earthUI article in this issue for the development version and optional components.)
2. **Launch the application:** run `earthUI::launch_glmnet()` in R. The app opens in your web browser on port 7879. You can also access the app directly by navigating to `http://localhost:7879` in your browser. The app remembers your last-used purpose mode and restores it automatically.
3. **Create or open a project** in Section 1 of the sidebar. A project sets the purpose and work location (country, state, county, city) and its input/output folders. Projects are stored under a shared `regProj` root and are shared with the sibling apps earthUI and mgcvUI. Opening a project with a different purpose is how you switch modes.
4. **Import your data** in Section 2 by selecting a CSV or Excel file from the active project’s input folder. Add files to that folder, then click **Refresh** to list them.
5. **Import from earthUI (optional)** — Section 3 lets you import an earthUI result `.rds` file to use earth’s hinge basis functions with glmnet’s regularization. Skip this step if you do not have an earthUI result.
6. **Configure variables** — choose your target and predictors, set data types, expected signs, and assign any special column roles.
7. **Set glmnet parameters** — alpha, lambda, family, sign constraints, interactions, and other options.
8. **Fit the model** — click “Fit Glmnet Model” and review the results in the main panel.
9. **Export** — download predictions as Excel, generate and convert a Quarto report, or (in appraisal mode) compute RCA adjustments and a Sales Comparison Grid.

Settings are automatically persisted in your browser’s local storage and restored when you reload the same input file.

## 2 MLS Input Data Requirements

For real estate appraisal and market analysis workflows, your input data typically comes from a Multiple Listing Service (MLS) export. This chapter describes the expected file structure and the columns that glmnetUI can use.

### 2.1 File Format & Structure

glmnetUI accepts **CSV** and **Excel** (.xlsx, .xls) files. On import, column names are automatically converted to **snake\_case** — for example, “Living SqFt” becomes **living\_sqft**, “Contract Date” becomes **contract\_date**, and “Sale Price” becomes **sale\_price**. This normalization ensures consistent column references throughout the workflow. The CSV separator and decimal mark used during import are determined by the locale settings (see Chapter 3, “Locale & Regional Settings”).

Your data file should be a flat table with one row per property and one column per attribute. The first row of the file must contain column headers.

### 2.2 Required Columns for Appraisal Mode

While glmnetUI works with any set of columns, the full appraisal workflow benefits from having the following columns:

| Column           | Special Type           | Purpose                                                         |
|------------------|------------------------|-----------------------------------------------------------------|
| Sale Price       | (target)               | Response variable for the model                                 |
| Contract Date    | <b>contract_date</b>   | Used to compute <b>sale_age</b> (days from effective date)      |
| Listing Date     | <b>listing_date</b>    | Used with contract date to compute DOM if no DOM column exists  |
| Days on Market   | <b>dom</b>             | Days on market; displayed in exports                            |
| Concessions      | <b>concessions</b>     | Sale concessions; Net SP = Sale Price – Concessions             |
| Living Area (SF) | <b>living_area</b>     | Enables per-SF residuals ( <b>residual_sf</b> , <b>cqa_sf</b> ) |
| Lot Size         | <b>lot_size</b>        | Site size column                                                |
| Site Dimensions  | <b>site_dimensions</b> | Grouped with lot size                                           |
| Latitude         | <b>latitude</b>        | Rounded to 3 dp                                                 |
| Longitude        | <b>longitude</b>       | Rounded to 3 dp                                                 |
| Area ID          | <b>area</b>            | Market area / neighborhood identifier                           |
| Actual Age       | <b>actual_age</b>      | Property age                                                    |
| Effective Age    | <b>effective_age</b>   | Effective property age                                          |
| Address          | <b>display_only</b>    | Shown in exports; excluded from model                           |

Spreadsheet column names can be in a foreign language — the “special” names are in English so that the R program can give them special treatment. Otherwise, the given column names show up in the regression models, graphs and, if doing appraisals, the output reports.

Not all columns are required. glmnetUI adapts — if a column is missing, the corresponding feature is simply omitted. However, for real estate pricing models certain columns are highly recommended to achieve acceptable fit:

1. **Sale Age** — the number of days between the contract sale date and the effective date of the appraisal or analysis. If multi-year sales history is being used, especially for periods over 5 years, **sale\_age** often plays a central role in estimating the sale price.
2. **Living Area** — also goes by names such as “Living Sqft,” “GLA” (gross living area) and so on. This is another leading determinant of sale price.

3. **Total Bath Count** — the total number of full, quarter, half, and 3/4 bathrooms. For example, two full baths and one half-bath would be a value of 2.5.
4. **Garage Bays** or **Garage Area** — the number of garage spaces or the garage square footage.
5. **Lot Size** — the land area of the property, typically in square feet or acres.
6. **Longitude**, **Latitude**, and if available **Area ID**. These location variables help the model account for geographic price variation.

## 2.3 Special Column Naming Conventions

glmnetUI identifies columns by their **special type designation**, not by their column name. You can name your columns anything you like in the MLS export — what matters is that you assign the correct special type in the Variable Configuration table (Chapter 6).

For example, your MLS might export living area as “GLA”, “Living SqFt”, “liv\_area”, or “gross\_living\_area”. After import (where it becomes snake\_case), you simply designate it as **living\_area** in the Special dropdown. glmnetUI will then use it for per-SF residual calculations regardless of its original name.

## 2.4 Data Quality & Completeness

- **Missing values (NA)**: Rows with NA values in any predictor or target column are automatically removed before fitting.
- **Date columns**: Character columns matching common date formats are auto-parsed on import. 2-digit year formats are prioritized when no 4-digit year is detected.
- **Numeric columns**: Sale price, living area, lot size, concessions, and similar fields must be numeric. If your MLS exports prices with currency symbols or thousands separators (e.g., “\$350,000” or “350.000,00”), you may need to clean these before import.
- **Factor columns**: Text, TRUE/FALSE, and R-factor columns are treated as categorical automatically — a regression can only handle them as factors. A **numeric** column (e.g., a numeric area or style code) becomes a factor only when you tick its **Factor** checkbox; glmnetUI never infers categorical from a numeric’s value range, so discrete numeric predictors like **bath\_count** stay continuous. The Special role is independent of the factor decision.

***Tip:** Review the NA column in the predictor table after import. Columns with many missing values may cause rows to be dropped. Consider excluding high-NA columns or cleaning the data before import.*

## 2.5 Subject Row Placement

In **Appraisal** mode, **row 1 must be the subject property**. All remaining rows are comparable sales. The subject row is excluded from model fitting. After fitting, the model still generates predictions for the subject row.

In **Market Area Analysis** mode, placing the subject in row 1 is optional.

In **General** mode, there is no special row handling — all rows are treated equally.

## 3 General Purpose Mode

### 3.1 Overview

General Purpose mode is the default when you launch glmnetUI. It provides the complete elastic net regression workflow for any dataset — not just real estate. You can use glmnetUI for scientific data, financial analysis, engineering studies, or any regression problem.

In General mode, the interface omits the real estate-specific features (special columns, sale age, coordinate rounding, RCA). The sidebar is streamlined to focus on variable selection, parameter configuration, model fitting, and export.

#### 3.1.1 Skip First Row

A **Skip first row** checkbox appears at the top of the sidebar once a General or Market project is open. When checked, row 1 is excluded from model fitting. This is useful when row 1 contains a target or reference observation that should not influence the model. In Appraisal mode, row 1 (the subject property) is always excluded automatically.

#### 3.1.2 Settings Per Purpose

Settings (predictor selections, model parameters, interactions) are saved separately for each combination of input file and purpose mode. Switching between General, Appraisal, and Market modes preserves each mode's settings independently.

### 3.2 The Sidebar Workflow

The sidebar is organized into numbered, collapsible sections that guide you from project selection through report export. Most sections appear only once an active project is open, and several are specific to the project's purpose. The output folder is derived from the active project — there is no separate output-folder field.

**1. Project** — Create or open a project. A project sets the purpose and work location (country, state, county, city) and its input (`<os>_in/`) and output folders. Projects are stored under a shared `regProj` root (default `~/regProj`, overridable with the `REGPROJ_ROOT` environment variable) and are shared with the sibling apps earthUI and mgcvUI, so models from all three live side by side. Click **Close Project** to switch.

**2. Import Data** — Select a CSV or Excel file from the active project's input folder. Add files to the folder via Finder/Explorer, then click **Refresh** to list them. For Excel files with multiple sheets, a sheet selector appears. Column names are automatically converted to `snake_case`.

**3. Import from earthUI (optional)** — Import an earthUI result `.rds` file to use earth's hinge basis functions with glmnet's elastic net regularization. The browse button is styled to match the Section 2 file picker. This step is optional — skip it if you do not have an earthUI result to import.

**4. Variable Configuration** — Target variable selector, predictor table with checkboxes for Include, Factor, and Force, plus Sign dropdowns. The Special column (for designating contract dates, coordinates, etc.) appears only in Appraisal and Market modes. See Chapter 6 for full details.

**5. glmnet Call Parameters** — All model configuration: alpha, lambda, family, standardize, sign constraints, relaxed lasso, random seed, interaction matrix, and advanced parameters (lambda.min.ratio, nlambdas, CV loss metric, convergence threshold, max iterations, intercept). See Chapter 7 for the complete parameter reference.

**6. Fit Glmnet Model** — The button that runs the model. Results appear in the tabs on the right.

**7. Download Output** — Exports predictions, residuals, CQA scores, and per-variable contributions as an Excel file to the project’s output folder. Available in all purpose modes once the model is fit.

**8. Calculate RCA Adjustments & Download** — Computes per-variable RCA adjustments for each comparable and interpolates the subject residual via CQA. Appears only in Appraisal and Market modes, after fitting. See Chapter 11.

**9. Generate Sales Grid & Download** — Generates the intermediate Sales Comparison Grid, ranking comparables by gross adjustment percentage. Appears only in Appraisal mode, after RCA adjustments are computed. See Chapter 12.

**10. Generate Quarto Report** — Writes a self-contained Quarto (.qmd) report bundle — with all plots and tables — into the project’s output folder. Available once the model is fit.

**11. Convert Quarto Report** — Renders the generated .qmd bundle to HTML, Word, or PDF. PDF requires a LaTeX installation; if none is detected, the PDF option is hidden. See Chapter 13.

In Market and General modes, the hidden appraisal-only steps are skipped and the two report sections renumber accordingly.

### 3.3 Projects

glmnetUI organizes work into **projects** under a shared **regProj** root folder (the same tree used by earthUI and mgcvUI). Each project owns an input folder (`<os>_in`) and per-engine output folders (e.g., `<os>_out_glmnet`).

To create a project, click + **New...** in Section 1 and provide:

- **Purpose** — General, Appraisal, or Market Area.
- **Location** — the project’s political/administrative hierarchy. The set of levels you fill in depends on the country you choose, so foreign naming conventions are supported automatically — a U.S. project uses state/county/city, a French one uses region/département/commune, a Japanese one uses prefecture/city-or-ward/district, and so on.
- **Project Name** — a short label, limited to 8 characters (letters, digits, \_ and -).

The app does not use your typed name verbatim: it builds a unique, time-stamped folder name by prepending the location codes and the creation date-time, in the form `<country>_<admin level codes>_<YYYYMMDD-HHMMSS>_<your name>` — for example, a U.S. project named `burl` becomes `us_ca_081_burlin_20260604-131500_burl`. The 8-character limit applies only to the part you type. Only newly created projects get the timestamp prefix; projects already on disk keep their original names and remain fully accessible.

### 3.4 Import from earthUI (Optional)

Section 3 of the sidebar — **Import from earthUI** — lets you import an earthUI result .rds file. This allows glmnetUI to use earth’s hinge basis functions (piecewise linear terms) with glmnet’s elastic net regularization, combining earth’s automatic nonlinearity detection with glmnet’s variable selection and coefficient constraints.

When an earthUI result is imported, the earth model’s predictors — not the Include checkboxes — drive the glmnet fit, because glmnet fits on earth’s basis expansion. The Variable Configuration table makes this explicit: **Include** and **Type** are locked, rows for predictors the earth model did not use are disabled, and a note explains that the predictor set comes from the earth model. **Force** and **Sign** stay editable for earth’s predictors, since those still apply to the earth basis at fit time.

The browse button is styled to match the Section 2 file picker. This step is entirely optional — if you do not have an earthUI result file, simply skip Section 3 and proceed to Section 4.

### 3.5 Main Panel Tabs

After fitting, the main panel provides the following tabs:

| Tab                 | Contents                                                                                                                                                            |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Preview        | Preview of imported data. In appraisal mode, split into Subject Property and Comparable Sales tables.                                                               |
| Equation            | The fitted model equation displayed in LaTeX/MathJax, showing each non-zero coefficient and interaction terms.                                                      |
| Correlation         | Heatmap of numeric predictor correlations (available before fitting).                                                                                               |
| Summary             | Key metrics ( $R^2$ , Adj $R^2$ , GR $^2$ , CV $R^2$ , RMSE, MAE) and appraisal metrics (COD, PRD, Median Ratio).                                                   |
| Coefficients        | Non-zero coefficients table with sign violation warnings.                                                                                                           |
| Variable Importance | Bar chart and ranked table of predictor importance ( $ \beta  \times \text{sd}(x)$ ). Interactive plotly hover when available.                                      |
| Contributions       | Per-variable partial effect plots with slope labels. Scatter + fit line for numeric predictors, box plots for factors, 3D surface/scatter/heatmap for interactions. |
| ANOVA               | Variance decomposition table: per-variable SS, % of model SS, coefficient.                                                                                          |
| Diagnostics         | Four plots: Coefficient Path, CV Error, Actual vs Predicted, Residuals vs Fitted.                                                                                   |
| Glmnet Output       | Raw model output: random seed used, model print, selected lambda, lambda.min/1se, gamma (if relaxed), and full coefficient vector.                                  |
| Report              | Report configuration and export fields.                                                                                                                             |
| RCA Adjustments     | RCA analysis results (appraisal mode, after computation).                                                                                                           |

### 3.6 Settings Persistence

glmnetUI automatically saves your configuration to the browser's local storage, keyed by the input filename. When you reload the same file, all settings are restored: target selection, predictor checkboxes, data types, expected signs, glmnet parameters, and interaction matrix. The last-used purpose mode is also persisted globally and restored when the app is relaunched. Three options are available:

- **Use last settings for input file** — restore saved settings for this specific file
- **Use default settings** — apply your saved global defaults
- **glmnet defaults** — reset to factory defaults (alpha = 1, CV lambda.1se, 10 folds, gaussian, standardize on)

A **Save current as default** button saves all current settings as the global default.

### 3.7 Dark Mode

Click the moon/sun icon in the upper-right corner to toggle between light and dark themes. The theme preference is saved in local storage and persists across sessions. All UI elements (tables, plots, cards, buttons) adapt to the selected theme via CSS variables.

### 3.8 Locale & Regional Settings

glmnetUI supports international number and CSV formatting conventions through a country-based locale system. The **Settings** dropdown in the title bar provides **Country** and **Paper** selectors for 31 supported countries. Each preset configures:

- **CSV separator** — comma (,) for US/UK/Japan or semicolon (;) for most of Europe, where the comma is used as a decimal mark.
- **Decimal mark** — period (.) or comma (,).
- **Thousands separator** — comma (US/UK/Japan), period (Germany/Italy/Spain), space (Finland/France/Poland/Baltics/Ukraine/Russia), or apostrophe (Switzerland).
- **Paper size** — Letter (US/Canada/Mexico) or A4 (everywhere else).

#### 3.8.1 Saving Defaults

Click **Save as my default** to store your locale preferences globally. These defaults apply to all future sessions regardless of which data file you load.

***Tip:** Number formatting on plot axes and slope labels automatically adapts to your locale. German locale uses periods for thousands (200.000), Finnish uses spaces (200 000), Swiss uses apostrophes (200'000). No currency symbols are displayed — glmnetUI is currency-agnostic.*

## 4 Appraisal Mode

When you select **For Appraisal** as the Purpose, glmnetUI configures itself for single-property valuation. All features described in Chapter 3 remain available; this chapter covers only the appraisal-specific additions.

### 4.1 Subject Row Handling

In appraisal mode, **row 1 of your dataset is the subject property** and all remaining rows are comparable sales. Your input file must be organized accordingly (see Chapter 2). The subject's sale price can be left blank or set to any value — glmnetUI automatically treats it as NA during fitting.

After importing, the Data Preview tab splits into two sections: **Subject Property** (row 1) and **Comparable Sales** (rows 2+). Row 1 is always excluded from model fitting. After fitting, the model still generates predictions for the subject row.

### 4.2 Effective Date & Sale Age

In appraisal and market modes, an **Effective Date** field appears in the Variable Configuration section (defaulting to today's date). If you designate a column as **contract\_date** in the Special column dropdown, glmnetUI computes a **sale\_age** column — the number of integer days between each sale's contract date and the effective date. This column is added as a predictor.

When the Effective Date changes, **sale\_age** is automatically recomputed.

### 4.3 Special Column Designations

In appraisal and market modes, a **Special** dropdown appears for each predictor in the Variable Configuration table. See Chapter 6 for the complete reference of special types and their effects.

### 4.4 RCA Adjustments Overview

The **Calculate RCA Adjustments & Download** button (sidebar section 8, visible in Appraisal and Market modes after fitting) computes market-derived adjustments for each comparable relative to the subject. The full RCA workflow is described in Chapter 11.

***Tip:** The CQA score you assign to the subject controls how much of the residual distribution is attributed to the subject. A score of 5.00 places the subject at the median of the comparables.*

## 5 Market Area Analysis Mode

When you select **Market Area Analysis** as the Purpose, glmnetUI provides the same real estate-specific features as appraisal mode (special columns, sale age, coordinate rounding) but is oriented toward analyzing a group of properties rather than valuing a single subject.

### 5.1 Differences from Appraisal Mode

- **Subject row handling is flexible** — in market mode, row 1 is not automatically treated as a subject property.
- **No Sales Comparison Grid** — the “Calculate RCA Adjustments & Download” step (Section 8) is available in market mode, but the Sales Grid step (Section 9) is appraisal-only.

### 5.2 When to Use Market Mode

Market Area Analysis mode is appropriate when you are:

- Building a regression model for a neighborhood or market area to understand value drivers
- Analyzing how variables like square footage, age, lot size, and location affect sale prices across a group of properties
- Preparing support for a market conditions analysis or neighborhood delineation
- Working with a dataset where you want special column features (sale age, coordinate rounding) but do not need the RCA adjustment workflow

**Tip:** Market mode is also useful for general real estate regression where you want special column features but do not need the RCA adjustment workflow.

## 6 Variable Selection

Section 4 of the sidebar — **Variable Configuration** — is where you choose which columns participate in the model and how they are treated.

### 6.1 Target Variable

The **Target (response) variable** dropdown at the top of Section 4 lists every column in your dataset. Select one column as the response variable (e.g., sale price). The target column is automatically excluded from the predictor list.

### 6.2 The Predictor Table

Below the target selector, a table lists every remaining column with the following fields:

| Column          | Description                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------|
| <b>Variable</b> | Column name                                                                                           |
| <b>Type</b>     | Data type dropdown: <b>numeric</b> , <b>integer</b> , <b>character</b> , <b>Date</b> , <b>POSIXct</b> |
| <b>Include</b>  | Checkbox — include this column as a predictor in the model                                            |
| <b>Factor</b>   | Checkbox — treat the variable as a categorical factor (creates dummy columns in the model matrix)     |
| <b>Force</b>    | Checkbox — force into model (penalty factor = 0, never dropped by lasso)                              |
| <b>Special</b>  | Dropdown (appraisal/market only) — see Special Column Types Reference below                           |
| <b>Sign</b>     | Expected coefficient sign: positive, negative, or either                                              |
| <b>NAs</b>      | Count of missing values                                                                               |

### 6.3 Data Type Detection & Overrides

glmnetUI automatically detects data types on import. Numeric, integer, and date columns are recognized. Character columns that look like dates (common date format patterns) are classified as **Date**.

You can override any detection by changing the **Type** dropdown. Changing types affects how the column is encoded in the model matrix.

Text, TRUE/FALSE, and R-factor columns are treated as categorical automatically. A numeric column becomes a factor only when you tick its **Factor** checkbox — glmnetUI never infers categorical from a numeric's value range, so discrete numeric predictors like a bath count stay continuous unless you say otherwise. The Factor setting persists via localStorage.

### 6.4 Expected Signs & Enforcement

For each predictor, the **Sign** dropdown specifies whether you expect its coefficient to be positive, negative, or either direction. After fitting:

- Coefficients with unexpected signs are **flagged in the Coefficients table**
- Sign warnings appear in the Summary tab

When the **Enforce Sign Constraints** checkbox is enabled in the glmnet parameters (Chapter 7), the model is forced to produce coefficients matching the expected signs. This is implemented via `upper.limits` and `lower.limits` in `glmnet`.

## 6.5 Special Column Types Reference

In appraisal and market modes, the **Special** dropdown provides the following options:

### Weighting:

- **weight** — Observation weight column (available in all modes; only one allowed; rows with `weight = 0` are excluded from fitting)

### Date & Time Types:

- **contract\_date** — Triggers automatic **sale\_age** computation from the Effective Date
- **listing\_date** — Used as a fallback for computing Days on Market
- **dom** — Identifies the Days on Market column
- **sale\_age** — Designates an existing sale-age column, so any column can serve as the sale age rather than having glmnetUI compute one from **contract\_date**

### Monetary Types:

- **concessions** — Identifies sale concessions (seller credits, buyer incentives, etc.)

### Transaction Types:

- **sale\_type** — Identifies the sale/transaction type column

### Size & Location Types:

- **latitude** — Values automatically rounded to 3 decimal places
- **longitude** — Same rounding treatment as latitude
- **area** — Market area or neighborhood identifier (e.g., `area_id`, `neighborhood`, `market_area`)
- **living\_area** — Enables per-square-foot residual calculations (**residual\_sf** and **cqa\_sf**)
- **lot\_size** — Site size column
- **site\_dimensions** — Grouped with lot size (e.g., “75x120”)

### Age Types:

- **actual\_age** — Property age column
- **effective\_age** — Effective property age

### Display Types:

- **display\_only** — Column is included in Excel exports but excluded from model fitting entirely. Use this for address fields, MLS numbers, parcel IDs, or other reference data. Multiple columns can have this designation.

**Tip:** Columns designated as **display\_only** remain in your dataset and appear in the Excel export, but are excluded from the predictor table entirely. Use this for ID columns, addresses, or other reference fields.

## 7 Parameter Selection

Section 5 of the sidebar — **glmnet Call Parameters** — provides access to all configuration options for the elastic net model. Each parameter has a blue help icon (?) with a tooltip explanation.

### 7.1 Alpha (Mixing Parameter)

The alpha parameter controls the type of regularization:

| Alpha Value | Type        | Behavior                                                        |
|-------------|-------------|-----------------------------------------------------------------|
| 0           | Ridge       | Shrinks all coefficients proportionally; never sets any to zero |
| 1           | Lasso       | Can set coefficients exactly to zero (variable selection)       |
| 0 to 1      | Elastic Net | Blend of ridge and lasso                                        |

Two alpha selection methods are available:

- **Fixed** — Set a single alpha value via the slider (default: 1.0)
- **Grid Search** — Test multiple alpha values and select the one with the lowest CV error. Configure the range and number of values.

### 7.2 Lambda Selection

Lambda controls the strength of regularization:

| Method                  | Description                                            |
|-------------------------|--------------------------------------------------------|
| <b>Cross-validation</b> | Recommended. Tests many lambda values using k-fold CV. |
| <b>Manual</b>           | Enter a specific lambda value if you have a reason to. |

When using cross-validation, two lambda choices are available:

- **lambda.1se** (default) — The largest lambda within 1 standard error of the minimum. Produces simpler, more regularized models. Recommended for appraisal work.
- **lambda.min** — The lambda that minimizes cross-validation error. Produces the best-fitting but potentially more complex model.

The **Number of CV Folds** parameter controls how the data is split for cross-validation (default: 10 folds).

### 7.3 Family

Choose the distribution family for your response variable:

| Family          | Use Case                                              |
|-----------------|-------------------------------------------------------|
| <b>gaussian</b> | Continuous responses (e.g., sale price). Most common. |
| <b>binomial</b> | Binary outcomes (e.g., sold/not sold).                |
| <b>poisson</b>  | Count data (e.g., number of sales).                   |

## 7.4 Standardize

When checked (default), all predictors are scaled to have mean 0 and standard deviation 1 before fitting. This ensures the penalty treats all predictors equally regardless of their original scale. Coefficients are returned on the original scale. Usually should be left on.

## 7.5 Sign Constraints

When **Enforce Sign Constraints** is enabled, coefficients are constrained to match the expected signs set in the variable table (Chapter 6):

- A “positive” sign forces the coefficient  $\geq 0$
- A “negative” sign forces the coefficient  $\leq 0$
- Variables set to “either” are unconstrained

This is implemented via the `upper.limits` and `lower.limits` parameters in `glmnet()`.

## 7.6 Relaxed Lasso

Regular lasso uses the same penalty for variable selection AND coefficient estimation, which can over-shrink important coefficients toward zero. **Relaxed lasso** separates these steps:

1. First, select variables via lasso (the penalty determines which coefficients survive)
2. Then, refit the surviving variables with less or no penalty for less biased estimates

The **Gamma** slider controls the degree of relaxation:

| Gamma   | Effect                                         |
|---------|------------------------------------------------|
| 0       | Fully relaxed (OLS refit, no shrinkage at all) |
| 1       | No relaxation (same as regular glmnet)         |
| Between | Blends relaxed and penalized fits              |

With cross-validation, gamma is chosen automatically for optimal performance.

## 7.7 Random Seed

A **Random seed** text input, pre-filled with a random integer, makes cross-validation fold assignments reproducible — `set.seed()` is called before fitting, and the seed used is shown in the Glmnet Output tab and in the fit status message. After each fit, a new seed is auto-generated (using system time, not R’s random number generator). The last 5 seeds used for the current file are shown as clickable links for easy recall; the seed history resets when a new file is loaded.

## 7.8 Interaction Matrix

An interactive upper-triangular matrix lets you control which variable pairs are allowed to interact. Each cell contains a checkbox — checked means the interaction term `x1:x2` is included in the model matrix.

- **Allow All** — Check all interaction pairs
- **Clear All** — Uncheck all pairs (the default for a new file is all unchecked)
- **Click a variable name** — Toggle all interactions for that variable

Interaction selections are saved to browser `localStorage` per input file and restored when you reload the same file.

When any interaction is enabled, the exported report includes an **Enabled Interactions** matrix (predictors on both axes, a check mark where the `x1:x2` product term was included),

rendered in landscape and mirroring earthUI's Allowed Interactions matrix. Note that the lasso may still have shrunk an enabled term's coefficient to zero.

**Tip:** *With many predictors, lasso will automatically zero out unneeded interaction terms. Starting with "Allow All" and letting the model select is a reasonable approach.*

#### Warning

Enabling glmnet interactions will produce adjustments that can be difficult to interpret and explain in court or audit settings. Use interactions cautiously and verify that the resulting adjustments are reasonable.

### 7.9 Block from Main Effect (Interaction Only)

**Right-click** a variable name in the Interaction Matrix to block it from entering the model as a main effect. A bold “1” indicator appears after the variable name. When blocked:

- The variable is **removed** from the main formula terms
- The variable **can still** appear in interaction terms (**x1:x2**)
- Right-click again to unblock
- The state persists via localStorage per input file

A typical use case: when modeling time adjustments that should scale with property size, block **sale\_age** from the main effect and allow it only in an interaction with **living\_area**. This creates a **sale\_age:living\_area** interaction term instead of a flat **sale\_age** coefficient — the time appreciation rate varies by house size.

### 7.10 Advanced Parameters

A collapsible “Advanced” section at the bottom of glmnet Call Parameters exposes additional settings. These use sensible defaults but are visible so all model settings can be documented for court or audit.

- **Lambda Min Ratio** — Ratio of smallest to largest lambda in the regularization path. Default: 0.00001. For small datasets (<50 observations), a larger value (e.g. 0.01) may improve performance.
- **Number of Lambda Values** — Number of lambda values in the path. Default: 100.
- **CV Loss Metric** — Loss function for cross-validation: MSE (default for gaussian), MAE (robust to outliers), or Deviance.
- **Convergence Threshold** — Coordinate descent convergence threshold. Default: 1e-07.
- **Max Iterations** — Maximum coordinate descent iterations. Default: 100,000. Increase if convergence warnings appear.
- **Fit Intercept** — Include an intercept (constant) term. Default: on. The intercept is the “basis value” in appraisal RCA.

### 7.11 Settings Defaults

Three options control how parameters are initialized when a file is loaded:

- **Use last settings for input file** — restore the exact settings used last time with this file
- **Use default settings** — apply your saved global defaults
- **glmnet defaults** — reset to factory settings (alpha = 1, CV lambda.1se, 10 folds, gaussian, standardize on)

Click **Save current as default** to store all current settings as the global default for future files.

## 8 Fitting the Model

### 8.1 The Fit Button

Section 6 of the sidebar contains the **Fit Glmnet Model** button. Clicking it runs the model with your current configuration.

### 8.2 What Happens During Fitting

When you click Fit, glmnetUI:

1. **Prepares the data** — removes rows with NAs, applies weights (if designated), excludes the subject row (in appraisal mode), and encodes factor variables as dummy columns.
2. **Builds the model matrix** — creates the design matrix including any allowed interaction terms from the interaction matrix.
3. **Applies constraints** — if sign enforcement is enabled, sets `upper.limits` and `lower.limits` on coefficients.
4. **Runs cross-validation** — if CV is selected, calls `cv.glmnet()` to find the optimal lambda. If relaxed lasso is enabled, calls `cv.glmnet(..., relax = TRUE)`.
5. **Fits the final model** — stores the fitted model, coefficients, predictions, and residuals.
6. **Updates all result tabs** — Summary, Coefficients, Equation, Variable Importance, Contributions, ANOVA, and Diagnostics are populated.

A status message below the Fit button shows the number of observations, excluded rows, and selected lambda.

## 9 Result Tabs

### 9.1 Data Preview

Shows the imported data as an interactive DataTable. In appraisal/market modes, the preview is split into two tables: **Subject Property** (row 1) and **Comparable Sales** (rows 2+).

Each cell is kept to a single line and truncated to the column width, so wide free-text fields (such as property remarks) do not stretch a row down the page. **Double-click any cell** to open a pop-up showing its full, untruncated contents.

### 9.2 Equation

Displays the fitted model equation rendered in LaTeX via MathJax. Shows all non-zero coefficients with proper mathematical formatting:

- Positive coefficients shown with + sign
- Negative coefficients shown with − sign
- Interaction terms displayed as  $x_1 \times x_2$
- Numbers formatted with commas and appropriate decimal places

### 9.3 Correlation

A heatmap matrix showing Pearson correlations among all numeric predictors and the response variable. Available immediately after data import — no model fitting required.

- Color gradient: blue (−1) → white (0) → red (+1)
- Correlation values printed in each cell
- Text and axis sizing adapts based on the number of variables
- Useful for identifying multicollinearity before fitting

### 9.4 Summary

Model fit statistics displayed as cards:

- **$R^2$**  — proportion of variance explained
- **Adj  $R^2$**  — adjusted for number of predictors
- **GR<sup>2</sup>** — generalized  $R^2$  from glmnet's deviance ratio at the selected lambda
- **CV  $R^2$**  — cross-validation  $R^2$  (when CV is used)
- **RMSE** — root mean squared error
- **MAE** — mean absolute error

In appraisal/market modes, additional cards show:

- **Median Ratio** — median of (predicted / actual)
- **COD** — Coefficient of Dispersion (lower is better)
- **PRD** — Price-Related Differential

An **overfitting warning** appears when training  $R^2$  exceeds CV  $R^2$  by more than 0.1.

Below the cards, a coefficient table with sign warnings duplicates the Coefficients tab for convenience.

### 9.5 Coefficients

A table of all non-zero coefficients showing:

- Variable name
- Coefficient value

- Expected sign (from variable configuration)
- Sign warning flag if the coefficient direction doesn't match expectations

## 9.6 Variable Importance

Standardized coefficient magnitude:  $|\beta| \times \text{sd}(x)$  for each predictor, aggregated across dummy columns for factor variables. Two display modes:

- **Interactive plotly** (when the `plotly` package is installed): Horizontal bar chart with hover tooltips showing exact importance, coefficient, and relative percentage.
- **Static ggplot2** (fallback): Horizontal bar chart with comma-formatted axis labels.

Below the chart, a `DataTable` shows Variable, Importance, Coefficient, and Relative %.

## 9.7 Contributions

Per-variable partial effect plots showing the contribution of each predictor to the model's predictions. Select a variable from the dropdown.

**For numeric predictors:** A scatter plot of the variable's x-values vs. its contribution ( $\beta \times x$ ), with a red line segment and a slope label. The slope label shows the marginal effect per unit (e.g., `+1,234.56/unit`) with adaptive units for small-range variables like latitude/longitude (e.g., `+0.12/0.001`). The subtitle shows the intercept (basis) value.

**For factor variables:** A box plot showing the distribution of contribution values per factor level.

**For two-variable interactions:** Three stacked visualizations — a 3D surface plot (interactive `plotly` when available, static `persp()` otherwise) of the joint contribution, a scatter plot with points colored by contribution, and a heatmap of mean contribution across both variable ranges.

## 9.8 ANOVA

Variance decomposition table showing for each predictor:

- **SS** — Sum of Squares:  $\sum(\text{contribution}_j - \overline{\text{contribution}_j})^2$
- **% of Model SS** — Percentage of total model sum of squares
- **Coefficient** — The dominant coefficient for that predictor

Includes an intercept row and a TOTAL MODEL row. Interaction terms (containing “:”) are shown as separate groups.

## 9.9 Diagnostics

Four diagnostic plots with large fonts (base size 16pt), 15 axis tick marks, and comma-formatted labels (no scientific notation):

1. **Coefficient Path** — Shows how all coefficients change as  $\log(\lambda)$  increases. Lines are colored by variable. Helps visualize the regularization path and variable selection.
2. **CV Error** — Cross-validation error as a function of  $\log(\lambda)$ . Points show mean CV error; error bars show  $\pm 1$  standard error. Dashed vertical lines mark  $\lambda_{\min}$  (blue) and  $\lambda_{1se}$  (green).
3. **Actual vs Predicted** — Scatter plot of observed vs. predicted values. Points should cluster around the 45-degree dashed reference line. Wider scatter indicates more prediction error.
4. **Residuals vs Fitted** — Scatter plot of residuals vs. fitted values. Should show random scatter around the zero line (dashed). Patterns suggest model misspecification (e.g., non-linearity, heteroscedasticity).

Each plot has a **Download PNG** button for saving at 150 DPI.

### 9.10 Glmnet Output

Raw model output showing: the random seed used, the model print, the selected lambda, lambda.min/lambda.1se, gamma (if relaxed lasso is enabled), and the full coefficient vector.

### 9.11 Report

Configure report fields (appraiser name, property address, report date, file number) and export to Word (.docx) or PDF directly from the tab. For the full Quarto-based report workflow (HTML, Word, or PDF), see Chapter 13.

## 10 Downloading Data

After fitting, the **Download Output (Excel)** button (sidebar section 7, available in all purpose modes) exports an Excel file with predictions and diagnostics to the project’s output folder. The section heading reads “Download Estimated Sale Prices & Residuals” in appraisal and market modes and “Download Estimated Target Variable(s) & Residuals” in General mode.

### 10.1 Output Columns

| Column                                | Description                                                     |
|---------------------------------------|-----------------------------------------------------------------|
| <code>est_&lt;target&gt;</code>       | Model prediction (e.g., <code>est_sale_price</code> )           |
| <code>residual</code>                 | Actual – predicted                                              |
| <code>cqa</code>                      | Condition-Quality-Appeal score (0–10 scale)                     |
| <code>residual_sf</code>              | Residual / living area (if <code>living_area</code> designated) |
| <code>cqa_sf</code>                   | CQA calculated from ranking via <code>residual_sf</code>        |
| <code>&lt;var&gt;_contribution</code> | Per-variable contribution to prediction                         |
| <code>basis</code>                    | Intercept value (same for all rows)                             |

A white checkmark appears on the download button after successful completion.

### 10.2 CQA Scores

CQA ranks each row’s residual against all others on a 0–10 scale:

- **High CQA (~9–10):** sold for much more than predicted — likely superior condition/quality/appeal
- **Low CQA (~0–1):** sold for much less than predicted — likely inferior condition or distressed sale
- **CQA ~5:** near the median

In appraisal/market modes, rows are sorted by `residual_sf` descending when a `living_area` column is designated.

## 11 RCA Calculations & Downloading

The RCA (Residual Constraint Approach) workflow is available in **Appraisal** and **Market Area Analysis** modes, after fitting the model. The follow-on Sales Comparison Grid (Chapter 12) is appraisal-only.

### 11.1 Opening the RCA Dialog

Click the **Calculate RCA Adjustments & Download** button in sidebar section 8. Choose:

- **CQA** or **CQA per SF** score type. If “CQA per SF” is selected, the subject’s residual is scaled by living area.
- Enter the subject’s CQA score (0.00–9.99, default 5.00)

### 11.2 CQA Score Interpolation

1. Comparables’ CQA scores and residuals are sorted
2. Linear interpolation maps your CQA value to a residual
3. Subject value = model prediction + interpolated residual

### 11.3 Output Columns

| Column              | Description                              |
|---------------------|------------------------------------------|
| subject_value       | Model prediction + interpolated residual |
| <var>_adjustment    | Subject contribution – comp contribution |
| residual_adjustment | Subject residual – comp residual         |
| net_adjustments     | Sum of all adjustments                   |
| gross_adjustments   | Sum of absolute adjustments              |
| adjusted_sale_price | Comp sale price + net adjustments        |

A white checkmark appears on the button after successful computation.

## 12 Sales Comparison Grid

The Sales Comparison Grid is available in **Appraisal** mode only, after computing RCA adjustments (Step 8). It generates a formatted Excel workbook suitable for inclusion in appraisal reports.

### 12.1 Comp Selection Dialog

Clicking the **Generate Sales Grid & Download** button in sidebar section 9 opens a modal dialog listing all comparable sales from the RCA output. Comps are split into two groups:

- **Recommended comps** (pre-checked): Gross adjustment < 25% of sale price, sorted by sale age (most recent first).
- **Other comps** (unchecked by default): Remaining comps, sorted by sale age.

Select up to 30 comps, then click **Generate Sales Grid** to create the workbook.

### 12.2 Workbook Layout

The generated workbook contains up to 10 sheets, each holding 3 comps side by side:

- **Header rows:** Subject and comp addresses, sale prices, sale dates, and Days on Market
- **Grouped rows:** Location (latitude, longitude, area), Site Size (lot size, site dimensions), and Age (actual age, effective age) with group headers and sub-items
- **Model variable rows:** One row per non-zero model coefficient showing the subject contribution, comp contributions, and adjustment percentages
- **Residual feature rows:** Empty, unlocked cells for appraiser input on features not captured by the model (e.g., condition, quality, view)
- **Summary rows:** Net Adjustment %, Gross Adjustment %, and Adjusted Sale Price with live Excel formulas

### 12.3 Sheet Protection & Residual Cells

Each sheet is protected with a password to prevent accidental edits, but the residual feature value cells are explicitly unlocked. This allows appraisers to enter values for features not in the model while preserving the formula-driven adjustment calculations.

## 13 Downloading Reports

### 13.1 The Two-Step Quarto Workflow

Report export from the sidebar is a two-step Quarto workflow:

1. **Generate Quarto Report** (sidebar section 10) — writes a self-contained Quarto bundle (the `.qmd` source plus plots, `report_data.rds`, and `reference.docx`) into the project's glmnet output folder.
2. **Convert Quarto Report** (sidebar section 11) — renders a `.qmd` file to your chosen output formats. The path defaults to the most recently generated bundle, but you can browse to any `.qmd`.

### 13.2 Report Formats

Three formats are available in the Convert step:

- **HTML** — Self-contained HTML document. Works on all platforms with no extra software beyond pandoc. Recommended default.
- **Word (.docx)** — Formatted document. Suitable for editing and distribution. Works on all platforms.
- **PDF** — Generated via Quarto with LaTeX. Paper size follows the locale setting (Letter or A4). **Requires a LaTeX installation** (see System Requirements appendix). If no LaTeX is detected, the PDF option is automatically hidden from the format checkboxes.

Reports are generated using Quarto when available, with a fallback to rmarkdown. The Report tab additionally offers direct Export to Word and Export to PDF buttons.

### 13.3 Report Contents

Reports include all tab content:

- Dataset description (file name, observations, response, predictors, purpose)
- Model specification (alpha, lambda, family, standardize, relaxed, non-zero coefficients)
- Results summary ( $R^2$ , Adjusted  $R^2$ , Generalized  $R^2$ , CV  $R^2$ , RMSE, MAE)
- Model equation (LaTeX-rendered)
- Coefficient table
- Variable importance (bar chart and table)
- Per-variable contribution plots matching the Contributions tab:
  - **Numeric predictors:** scatter plot with linear fit line
  - **Factor variables:** box plot by level
  - **Interactions:** scatter colored by contribution, heatmap of mean contribution, and static 3D surface snapshot (persp)
- Enabled Interactions matrix (when any interaction is enabled) — predictors on both axes, with a check mark where the `x1:x2` product term was included, rendered in landscape
- Correlation matrix heatmap
- ANOVA decomposition table
- Diagnostic plots:
  - Coefficient Path
  - Cross-Validation Error
  - Actual vs Predicted
  - Residuals vs Fitted
  - Normal Q-Q Plot

- Model output

All axis labels and color legends use comma-formatted numbers (no scientific notation).

A white checkmark appears on the report button after successful generation, and an elapsed timer is shown while the report is being rendered.

## 14 Comparison with earthUI

The earth application and its glmnet Post Processing Module are one toolset for regression modeling. They share the same data format, special column types, RCA workflow, and demo datasets, but use different modeling engines.

### 14.1 Key Differences

| Feature                   | glmnetUI                        | earthUI                                |
|---------------------------|---------------------------------|----------------------------------------|
| Method                    | Elastic net (glmnet)            | MARS/Earth (earth)                     |
| Variable selection        | Automatic via lasso penalty     | Automatic via forward/backward pruning |
| Coefficient relationships | Linear effects only             | Piecewise linear (hinge functions)     |
| Interactions              | Pairwise via interaction matrix | Up to degree 3                         |
| Coefficient signs         | Enforceable via constraints     | Not directly enforceable               |
| Relaxed fitting           | Relaxed lasso available         | Not applicable                         |
| Alpha tuning              | Grid search available           | Not applicable (no mixing parameter)   |
| Sales Comparison Grid     | Available (Step 9)              | Available (Step 8)                     |
| g-Function plots          | Not applicable                  | Grouped term visualization             |
| Report formats            | HTML, Word, PDF                 | HTML, Word, PDF                        |

### 14.2 Shared Features

Both tools provide:

- CSV and Excel import with snake\_case column conversion
- 31-country locale support with CSV separator, decimal mark, and paper size
- Special column designations (contract\_date, living\_area, area, latitude, longitude, etc.)
- Automatic sale\_age computation from effective date
- Dark/light mode toggle
- Settings persistence via localStorage
- Correlation heatmap, variable importance, contributions, ANOVA, diagnostics
- CQA scoring and RCA adjustment workflow
- Sales Comparison Grid with comp selection and formatted Excel output
- Compatible with the same demo datasets (Appraisal\_1.csv)

### 14.3 When to Use Which

- **Use glmnetUI** when you want a simple, interpretable linear model with automatic variable selection. Elastic net is particularly well-suited when you have many predictors and want the model to select the most important ones.
- **Use earthUI** when you expect non-linear relationships (e.g., the effect of living area on price changes at different sizes) or need hinge functions to capture threshold effects. Earth models can also capture higher-order interactions (degree 2 and 3). earthUI additionally offers an experimental Prolog/DCG-based remarks-extraction feature that has no counterpart in glmnetUI.

- **Use both** to compare linear vs. non-linear models on the same data. If both produce similar  $R^2$  values, the simpler glmnet model may be preferred for interpretability.

## 15 Demo Dataset: Appraisal\_1.csv

### 15.1 Description

The demo dataset is bundled with glmnetUI (and shared with earthUI). Locate it with:

```
demo_file <- system.file("extdata", "Appraisal_1.csv", package = "glmnetUI")
```

The file contains 1,502 residential sales (plus 1 subject property in row 1) from a simulated MLS export. The data represents single-family home sales in a multi-area market with a range of property sizes, ages, and locations.

This is not real data, but is based on a realistic neighborhood in Northern California. All identification information has been altered or removed.

### 15.2 Columns

| Column           | Type      | Special Type  | Description                                    |
|------------------|-----------|---------------|------------------------------------------------|
| weight           | numeric   | weight        | Observation weight (0 = exclude from fitting)  |
| id               | numeric   | display_only  | Internal record ID                             |
| property_id      | numeric   | display_only  | MLS property identifier                        |
| listing_id       | character | display_only  | MLS listing number                             |
| parcel_number    | character | display_only  | County assessor parcel number (APN)            |
| street_address   | character | display_only  | Property address                               |
| city_name        | character | display_only  | City                                           |
| postal_code      | character | display_only  | ZIP code                                       |
| county_name      | character | display_only  | County                                         |
| contract_date    | Date      | contract_date | Sale contract date (computes sale_age)         |
| sale_age         | numeric   | —             | Days from contract date to effective date      |
| sale_price       | numeric   | (target)      | Sale price — response variable                 |
| living_sqft      | numeric   | living_area   | Gross living area in square feet               |
| beds_total       | integer   | —             | Number of bedrooms                             |
| baths_total      | numeric   | —             | Total bath count (e.g., 2.5 = 2 full + 1 half) |
| lot_size         | numeric   | lot_size      | Lot size in square feet                        |
| area_id          | integer   | area          | MLS area identifier                            |
| age              | numeric   | actual_age    | Property age in years                          |
| latitude         | numeric   | latitude      | Latitude (rounded to 3 dp for model)           |
| longitude        | numeric   | longitude     | Longitude (rounded to 3 dp for model)          |
| garage_spaces    | integer   | —             | Number of garage bays                          |
| days_on_market   | integer   | dom           | Days on market                                 |
| listing_date     | Date      | listing_date  | Listing date                                   |
| sale_concessions | numeric   | concessions   | Seller concessions                             |

### 15.3 Suggested Quick Start

1. Launch the glmnet routine: `earthUI::launch_glmnet()`
2. Create a new project (Section 1) with purpose **For Appraisal**
3. Place `Appraisal_1.csv` in the project's input folder and select it in Section 2 (click **Refresh** if needed)
4. Select `sale_price` as the target
5. Assign special types as shown in the table above

6. Include predictors: `sale_age`, `living_sqft`, `baths_total`, `lot_size`, `area_id` (as factor), `age`, `latitude`, `longitude`, `garage_spaces`
7. Keep  $\alpha = 1$  (Lasso), CV with `lambda.1se`
8. Click **Fit Glmnet Model**
9. Review Summary, Coefficients, and Diagnostics tabs
10. Download output (Step 7), review the CQA ranking
11. Compute RCA adjustments (Step 8) with a CQA score of  $\sim 5.00$
12. Generate a Sales Comparison Grid (Step 9) with recommended comps

## 16 System Requirements & Troubleshooting

### 16.1 Supported Platforms

glmnetUI runs on **macOS**, **Windows**, and **Linux**. RStudio Desktop (2023.06+) is strongly recommended — it bundles **pandoc** needed for reports.

### 16.2 Optional Dependencies

- **LaTeX** (TinyTeX, MiKTeX, or MacTeX): PDF report generation only. If not detected, the PDF option is automatically hidden. Install with: `tinytex::install_tinytex()`
- **sysfonts + showtext**: Roboto Condensed font for plots. Falls back to system sans-serif automatically if unavailable or offline.
- **earth** ( $\geq 5.3.0$ ): Only needed for the earthUI import pipeline.

### 16.3 Platform Notes

**macOS**: Fewest issues. Install TinyTeX for PDF: `tinytex::install_tinytex()`

**Windows**: Works well with RStudio. Corporate/locked-down machines may have temp directory restrictions — the app will warn in the console but continue without settings persistence.

**Linux**: May need system libraries: `sudo apt install libcurl4-openssl-dev libssl-dev libxml2-dev libsqlite3-dev libfontconfig1-dev`

### 16.4 Graceful Degradation

| Missing Component                 | Behavior                                                                                  |
|-----------------------------------|-------------------------------------------------------------------------------------------|
| No LaTeX                          | PDF option hidden. HTML and Word still available.                                         |
| No internet / fonts fail          | System sans-serif used. Console message logged.                                           |
| No RSQLite / read-only filesystem | Settings don't persist. App runs normally.                                                |
| Temp directory not writable       | Report generation fails with clear error. Set <code>TMPDIR</code> to a writable location. |

### 16.5 Troubleshooting

**“PDF option not available”**: Run `tinytex::install_tinytex()` in R, restart app.

**“Settings will not persist”**: Check permissions on the user data directory (macOS: `~/Library/Application Support/R/glmnetUI/`, Linux: `~/.local/share/R/glmnetUI/`, Windows: `%APPDATA%/R/data/glmnetUI/`).

**Port 7879 already in use**: Run `lsof -ti:7879 | xargs kill` (macOS/Linux) or `Stop-Process -Id (Get-NetTCPConnection -LocalPort 7879).OwningProcess` (Windows PowerShell).

## References

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2010). “Regularization Paths for Generalized Linear Models via Coordinate Descent”. In: *Journal of Statistical Software* 33.1, pp. 1–22. DOI: [10.18637/jss.v033.i01](https://doi.org/10.18637/jss.v033.i01).

- Simon, Noah et al. (2011). “Regularization Paths for Cox’s Proportional Hazards Model via Coordinate Descent”. In: *Journal of Statistical Software* 39.5, pp. 1–13. DOI: [10.18637/jss.v039.i05](https://doi.org/10.18637/jss.v039.i05).
- Tay, J. Kenneth, Balasubramanian Narasimhan, and Trevor Hastie (2023). “Elastic Net Regularization Paths for All Generalized Linear Models”. In: *Journal of Statistical Software* 106.1, pp. 1–31. DOI: [10.18637/jss.v106.i01](https://doi.org/10.18637/jss.v106.i01).

# mgcvUI: An Interactive Interface for the mgcv (Generalized Additive Models) Package

William Bert Craytor

Vol. 1, Issue 1 · Summer 2026

Published July 1, 2026

DOI: [10.67330/xtv1pa25](https://doi.org/10.67330/xtv1pa25)

© 2026 William Bert Craytor. Published by Pacific Vista Net.

This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

**Abstract.** mgcvUI is a graphical user interface for the R mgcv package, which fits Generalized Additive Models (GAMs) using penalized regression splines with automatic smoothness selection. It offers three purpose modes—general predictive modeling, real-estate appraisal, and market-area analysis—and guides the user through data import, smooth-term specification, model fitting, diagnostic and effect plots, and downloadable reports. This article documents mgcvUI’s data-format requirements, modeling workflow, output displays, and complete feature reference.

**Keywords:** generalized additive models, GAM, penalized splines, smoothing, mgcv, R, graphical user interface, real estate valuation

**Publisher’s Note (July 2026):** Shortly before press, mgcvUI was consolidated into earthUI (version 0.11.0 and later), which now provides the GAM workflow described here alongside the MARS and elastic-net workflows. mgcvUI remains available in its final released form (0.3.0) at <https://github.com/wcraytor/mgcvUI>; future development continues in earthUI. This article applies to mgcvUI as released.

## Contents

|          |                                                       |           |
|----------|-------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                   | <b>4</b>  |
| 1.1      | History and Background                                | 4         |
| 1.1.1    | Generalized Additive Models                           | 4         |
| 1.1.2    | The mgcv Package                                      | 4         |
| 1.1.3    | Comparison with MARS (earth) and Elastic Net (glmnet) | 4         |
| 1.1.4    | The earthUI–mgcvUI Pipeline                           | 5         |
| 1.1.5    | Recommended Workflow                                  | 5         |
| 1.2      | What Is mgcvUI?                                       | 6         |
| 1.3      | Three Purpose Modes                                   | 6         |
| 1.4      | Real Estate–Specific Features                         | 7         |
| 1.5      | Getting Started                                       | 7         |
| <b>2</b> | <b>MLS Input Data Requirements</b>                    | <b>8</b>  |
| 2.1      | File Format & Structure                               | 8         |
| 2.2      | Required Columns for Appraisal Mode                   | 8         |
| 2.3      | Special Column Naming Conventions                     | 9         |
| 2.4      | Data Quality & Completeness                           | 9         |
| 2.5      | Subject Row Placement                                 | 9         |
| <b>3</b> | <b>General Purpose Mode</b>                           | <b>10</b> |
| 3.1      | Overview                                              | 10        |
| 3.2      | The Sidebar Workflow                                  | 10        |
| 3.3      | Import from earthUI (Optional)                        | 11        |
| 3.4      | Main Panel Tabs                                       | 11        |

|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
| 3.5      | Settings Persistence                                       | 12        |
| 3.6      | Dark Mode                                                  | 12        |
| 3.7      | Locale & Regional Settings                                 | 12        |
| <b>4</b> | <b>Appraisal Mode</b>                                      | <b>14</b> |
| 4.1      | Subject Row Handling                                       | 14        |
| 4.2      | Effective Date & Sale Age                                  | 14        |
| 4.3      | Special Column Designations                                | 14        |
| 4.4      | RCA Adjustments Overview                                   | 14        |
| <b>5</b> | <b>Market Area Analysis Mode</b>                           | <b>15</b> |
| 5.1      | Differences from Appraisal Mode                            | 15        |
| 5.2      | When to Use Market Mode                                    | 15        |
| <b>6</b> | <b>Variable Selection</b>                                  | <b>16</b> |
| 6.1      | Target Variable                                            | 16        |
| 6.2      | Response Transform                                         | 16        |
| 6.3      | The Predictor Table                                        | 16        |
| 6.4      | Data Type Detection & Overrides                            | 17        |
| 6.5      | Special Column Types Reference                             | 17        |
| <b>7</b> | <b>Parameter Selection</b>                                 | <b>18</b> |
| 7.1      | Parameter Presets                                          | 18        |
| 7.2      | Family                                                     | 18        |
| 7.3      | Method                                                     | 18        |
| 7.4      | Gamma (Smoothing Multiplier)                               | 19        |
| 7.5      | Cross-Validation                                           | 19        |
| 7.6      | Select (Shrinkage)                                         | 19        |
| 7.7      | Default Basis                                              | 19        |
| 7.8      | Default k (Basis Dimension)                                | 19        |
| 7.9      | Tensor Type                                                | 20        |
| 7.10     | Allowed Interactions                                       | 20        |
| 7.11     | Factor-by-Smooth Interactions                              | 20        |
| 7.12     | Advanced Parameters                                        | 20        |
| <b>8</b> | <b>Fitting the Model</b>                                   | <b>21</b> |
| 8.1      | The Fit Button                                             | 21        |
| 8.2      | What Happens During Fitting                                | 21        |
| 8.3      | Background Fitting & Progress                              | 21        |
| <b>9</b> | <b>Result Tabs</b>                                         | <b>22</b> |
| 9.1      | Data                                                       | 22        |
| 9.2      | Equation                                                   | 22        |
| 9.3      | Summary                                                    | 22        |
| 9.4      | Variable Importance                                        | 22        |
| 9.5      | Contribution Plots                                         | 23        |
| 9.5.1    | Univariate Smooth Terms — $s(x)$                           | 23        |
| 9.5.2    | Factor-by-Smooth Terms — $s(x, \text{by} = \text{factor})$ | 23        |
| 9.5.3    | Interaction Terms — $ti(x_1, x_2)$ and $te(x_1, x_2)$      | 23        |
| 9.5.4    | Parametric Terms — Linear and Factor                       | 23        |
| 9.6      | Correlation                                                | 23        |
| 9.7      | Diagnostics                                                | 23        |

|           |                                                  |           |
|-----------|--------------------------------------------------|-----------|
| 9.8       | RCA Adjustments                                  | 23        |
| 9.9       | Anova                                            | 24        |
| 9.10      | Mgcv Output                                      | 24        |
| 9.11      | Sign Check                                       | 24        |
| 9.12      | Concurvity                                       | 24        |
| 9.13      | Basis Check                                      | 24        |
| <b>10</b> | <b>Downloading Data</b>                          | <b>25</b> |
| 10.1      | Output Columns                                   | 25        |
| 10.2      | CQA Scores                                       | 25        |
| <b>11</b> | <b>RCA Calculations &amp; Downloading</b>        | <b>26</b> |
| 11.1      | Opening the RCA Dialog                           | 26        |
| 11.2      | CQA Score Interpolation                          | 26        |
| 11.3      | Output Columns                                   | 26        |
| <b>12</b> | <b>Sales Comparison Grid</b>                     | <b>27</b> |
| 12.1      | Comp Selection Dialog                            | 27        |
| 12.2      | Workbook Layout                                  | 27        |
| 12.3      | Sheet Protection & Residual Cells                | 27        |
| <b>13</b> | <b>Downloading Reports</b>                       | <b>28</b> |
| 13.1      | Step 1: Generate Quarto Report                   | 28        |
| 13.2      | Step 2: Convert Quarto Report                    | 28        |
| 13.3      | Report Contents                                  | 28        |
| <b>14</b> | <b>Exporting Model Functions</b>                 | <b>30</b> |
| 14.1      | R Functions                                      | 30        |
| 14.2      | Python Functions                                 | 30        |
| 14.3      | C++ Header                                       | 30        |
| 14.4      | SQLite Database                                  | 30        |
| <b>15</b> | <b>Comparison with earthUI and glmnetUI</b>      | <b>31</b> |
| 15.1      | Key Differences                                  | 31        |
| 15.2      | Shared Features                                  | 31        |
| 15.3      | Recommended Workflow                             | 31        |
| <b>16</b> | <b>Demo Dataset: Appraisal_1.csv</b>             | <b>32</b> |
| 16.1      | Description                                      | 32        |
| 16.2      | Columns                                          | 32        |
| 16.3      | Suggested Quick Start                            | 33        |
| <b>17</b> | <b>System Requirements &amp; Troubleshooting</b> | <b>34</b> |
| 17.1      | Supported Platforms                              | 34        |
| 17.2      | R Version                                        | 34        |
| 17.3      | Required R Packages                              | 34        |
| 17.4      | Optional Dependencies                            | 34        |
| 17.5      | Platform-Specific Notes                          | 35        |
| 17.5.1    | macOS                                            | 35        |
| 17.5.2    | Windows                                          | 35        |
| 17.5.3    | Linux                                            | 35        |
| 17.6      | Graceful Degradation                             | 35        |
| 17.7      | Troubleshooting                                  | 36        |

|                                                                |    |
|----------------------------------------------------------------|----|
| 17.7.1 “PDF option is not available” . . . . .                 | 36 |
| 17.7.2 “Settings will not persist” . . . . .                   | 36 |
| 17.7.3 “Diagnostics plot unavailable” . . . . .                | 36 |
| 17.7.4 “Report error: number of items to replace...” . . . . . | 36 |
| 17.7.5 Port 7880 already in use . . . . .                      | 36 |
| 17.7.6 Package installation fails on Linux . . . . .           | 36 |

## 1 Introduction

### 1.1 History and Background

#### 1.1.1 Generalized Additive Models

**Generalized Additive Models** (GAMs) were formalized by Trevor Hastie and Robert Tibshirani in their 1990 monograph. GAMs extend the linear model by replacing each linear term  $\beta_j x_j$  with a smooth function  $f_j(x_j)$ , so the model becomes:

$$y = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p) + \varepsilon$$

Each  $f_j$  is estimated from the data using a penalized regression spline. The smooth functions allow the model to capture nonlinear relationships without the user specifying a functional form in advance. The penalty controls the wiggleness of each smooth, preventing overfitting while allowing enough flexibility to track real patterns.

#### 1.1.2 The mgcv Package

The R package **mgcv** (Mixed GAM Computation Vehicle) was developed by Simon Wood at the University of Bath (Wood 2017). First released in 2001, it is now the standard R implementation of GAMs and ships with every R installation. Its underlying methods are described in a series of papers covering stable multiple smoothing parameter estimation (Wood 2004), thin-plate regression splines (Wood 2003), fast stable REML estimation (Wood 2011), and general smooth model selection (Wood, Pya, and Säfken 2016). Key features include:

- **Automatic smoothness selection:** The smoothing parameter for each term is estimated as part of the model fitting, using REML (Restricted Maximum Likelihood), GCV (Generalized Cross-Validation), or ML (Maximum Likelihood).
- **Multiple basis types:** Thin plate regression splines (tp), cubic regression splines (cr), P-splines (ps), and B-splines (bs), each with different properties.
- **Tensor product smooths:** `te()` and `ti()` allow smooth interactions between variables without the curse of dimensionality.
- **Variable selection:** The `select = TRUE` option adds an extra penalty that can shrink smooth terms to zero, effectively performing variable selection.
- **Concurvity diagnostics:** Measures the analogue of multicollinearity for smooth terms.

#### 1.1.3 Comparison with MARS (earth) and Elastic Net (glmnet)

The **earth** application and its two Post Processing Modules (**glmnet** and **mgcv**) use different modeling engines. The following table summarizes the key differences:

| Feature                    | mgcv (GAM)                                                         | earth (MARS)                                                   | glmnet (Elastic Net)                                          |
|----------------------------|--------------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------------------------|
| <b>Model type</b>          | Smooth nonlinear (penalized splines)                               | Piecewise linear (adaptive splines)                            | Linear (with regularization)                                  |
| <b>Nonlinearity</b>        | Automatic via smooth functions (thin plate, cubic, etc.)           | Automatic via hinge functions with data-driven knots           | Only through interactions or basis expansion                  |
| <b>Variable selection</b>  | Optional via double-penalty shrinkage ( <code>select=TRUE</code> ) | Built-in via forward/backward stepwise with GCV pruning        | Built-in via $L_1$ penalty (Lasso)                            |
| <b>Interactions</b>        | Tensor product smooths ( <code>te()</code> , <code>ti()</code> )   | Hinge-based products (interpretable, up to degree 3)           | Pairwise cross-products (hard to interpret)                   |
| <b>Interpretability</b>    | Smooth partial effect curves; very intuitive                       | g-functions show piecewise linear partial effects per variable | Coefficients are linear weights; easy to explain individually |
| <b>Best for</b>            | Smooth nonlinear relationships, flexible modeling                  | Automatic nonlinearity and interaction detection               | High-dimensional data, variable selection, small samples      |
| <b>RCA adjustments</b>     | Smooth adjustments via partial effects                             | Piecewise linear adjustments via g-functions                   | Linear per-variable adjustments                               |
| <b>Overfitting control</b> | REML/GCV smoothing parameter estimation                            | GCV-based pruning                                              | Cross-validated lambda                                        |

**MARS** (Multivariate Adaptive Regression Splines) was introduced by Jerome Friedman in 1991. It builds piecewise linear models by adaptively selecting hinge functions and their knot positions from the data. The R implementation is the **earth** package by Stephen Milborrow.

**Elastic net** regression, implemented in the **glmnet** package by Friedman, Hastie, and Tibshirani (2010), combines Lasso ( $L_1$ ) and ridge ( $L_2$ ) penalties for simultaneous variable selection and coefficient shrinkage.

#### 1.1.4 The earthUI–mgcvUI Pipeline

A powerful workflow combines earthUI’s automatic knot discovery with mgcvUI’s smooth estimation. When earthUI exports an `.rds` result file and mgcvUI imports it:

1. **Earth discovers the knots** — MARS finds data-driven hinge positions (change points) for each variable.
2. **mgcvUI uses them as anchors** — The earth knots become the basis for cubic regression splines (`cr` basis) in the GAM, giving the smooth terms a head start with knot positions that reflect real structure in the data.
3. **mgcv refines the fit** — The penalized spline estimation smooths out the piecewise linear earth fit, producing smooth partial effect curves while preserving the general shape discovered by earth.

This pipeline bridges exploratory modelling (earth) with confirmatory modelling (GAM), combining the best of both approaches.

#### 1.1.5 Recommended Workflow

For real estate appraisal and similar applications requiring interpretable, defensible models:

1. **Start with earthUI** to discover the important variables, nonlinear relationships, and interactions in your data. Earth’s automatic basis selection and g-functions provide excellent initial insights.
2. **Refine with mgcvUI** if you want smoother partial effects. Import earthUI’s knots into mgcvUI for a seamless transition from piecewise linear to smooth models.
3. **Use glmnetUI** when you need aggressive variable selection (Lasso), when the number of predictors approaches or exceeds the number of observations, or when you want a purely linear model with regularization for defensibility.

## 1.2 What Is mgcvUI?

mgcvUI is a graphical user interface for the R `mgcv` package. It runs as a local Shiny application — there is no login, no server, and no accounts. You launch it from R, import a dataset (CSV or Excel), configure your model, and fit a GAM interactively.

As of the current release, mgcvUI is no longer a separately installed package: it ships *inside* the earthUI package as its GAM Post Processing Module, started with `earthUI::launch_mgcv()`. The application itself — and everything described in this article — is unchanged; only the installation and launch command differ. The three routines (`earth`, `glmnet`, `mgcv`) share one project tree, one settings database, and one copy of the supporting infrastructure.

The application provides a complete workflow: data import, variable configuration with smooth term specification, model fitting with background processing, diagnostic plots, smooth partial effect curves, and downloadable reports in Word, PDF, or HTML format.

**Generalized Additive Models** replace the linear relationship  $\beta x$  with a smooth function  $f(x)$  for each predictor. This means:

- **No need to specify functional forms** — the data determines the shape of each relationship.
- **Smooth partial effects** — each predictor’s contribution is a smooth curve, not a straight line, making it easy to see how a variable affects the response at different values.
- **Automatic smoothness selection** — mgcv determines how wiggly each curve should be via REML or GCV, balancing fit against overfitting.

## 1.3 Three Purpose Modes

Every project has a **purpose**, chosen when you create it in Section 1 of the sidebar (**Project**) and locked thereafter. The active project’s purpose determines which tools and interface elements appear; to work in a different mode, create or open a project with that purpose. The three modes are:

- **General** — GAM regression for any type of population or dataset. This is the default mode. It provides the full GAM workflow without any domain-specific additions.
- **For Appraisal** — GAM regression tailored for real estate appraisal. Adds features specific to single-property valuation, including subject property handling, special column designations, and Residual Constraint Approach (RCA).
- **Market Area Analysis** — GAM regression tailored for market area studies. Adds features for analyzing groups of properties in a defined market, with optional subject row handling.

In all three modes, the core modeling engine is identical — you are always fitting a GAM via `mgcv::gam()`. The purpose setting controls which additional tools and interface elements are available.

## 1.4 Real Estate–Specific Features

When either **For Appraisal** or **Market Area Analysis** is selected, mgcvUI activates several features designed for real estate analysis:

- **Special column designations** — Each predictor can be tagged with a special role such as `contract_date`, `dom`, `concessions`, `latitude`, `longitude`, `living_area`, `lot_size`, `actual_age`, `effective_age`, `area`, `site_dimensions`, or `display_only`. These designations control how the column is handled during fitting and output.
- **Sale Age column** — When a column is designated as `contract_date` and an Effective Date is provided, mgcvUI computes a `sale_age` column (days between sale date and effective date) and substitutes it as a predictor.
- **RCA computations (appraisal only)** — In appraisal mode, after fitting the model mgcvUI can compute Residual Constraint Approach (RCA) output, which produces per-comparable adjustments, net/gross adjustment summaries, and an adjusted sale price for the subject property.

***Tip:** The General purpose mode works for any dataset — financial, scientific, engineering, or real estate. The appraisal and market modes simply add convenience features for real estate professionals.*

## 1.5 Getting Started

To use mgcvUI:

1. **Install earthUI** in R: `install.packages("earthUI")` — the mgcv routine is included. (See “Installing earthUI and vProlog” in the earthUI article in this issue for the development version and optional components.)
2. **Launch the application:** run `earthUI::launch_mgcv()` in R. The app opens in your web browser on port 7880.
3. **Create or open a project** in Section 1 of the sidebar. A project sets the purpose (General, For Appraisal, or Market Area Analysis) and work location (country, state, county, city) and its input/output folders. Projects are stored under a shared `regProj` root and are shared with the sibling apps earthUI and glmnetUI. Opening a project with a different purpose is how you switch modes.
4. **Import your data** in Section 2 by selecting a CSV or Excel file from the active project’s input folder. Add files to that folder, then click **Refresh** to list them.
5. **Import from earthUI (optional)** — Section 3 lets you import an earthUI result `.rds` file to seed the GAM with earth-discovered knots. Skip this step if you do not have an earthUI result.
6. **Configure variables** — choose your target and predictors, set data types, assign any special column roles, and mark variables as factor or linear.
7. **Set mgcv parameters** — family, method, gamma, basis type, k, tensor type, interactions, and other options.
8. **Fit the model** — click “Fit Mgc v GAM Model” and review the results in the main panel.
9. **Export** — download predictions as Excel, generate and convert a Quarto report, or (in appraisal mode) compute RCA adjustments and a sales grid.

Settings are automatically persisted in a SQLite database and restored when you reload the same input file.

## 2 MLS Input Data Requirements

For real estate appraisal and market analysis workflows, your input data typically comes from a Multiple Listing Service (MLS) export. This chapter describes the expected file structure and the columns that mgcvUI can use.

### 2.1 File Format & Structure

mgcvUI accepts **CSV** and **Excel** (.xlsx, .xls) files. On import, column names are automatically converted to **snake\_case** — for example, “Living SqFt” becomes **living\_sqft**, “Contract Date” becomes **contract\_date**, and “Sale Price” becomes **sale\_price**. This normalization ensures consistent column references throughout the workflow. The CSV separator and decimal mark used during import are determined by the locale settings (see Chapter 3, “Locale & Regional Settings”).

Your data file should be a flat table with one row per property and one column per attribute. The first row of the file must contain column headers.

### 2.2 Required Columns for Appraisal Mode

While mgcvUI works with any set of columns, the full appraisal workflow benefits from having the following columns:

| Column           | Special Type           | Purpose                                                         |
|------------------|------------------------|-----------------------------------------------------------------|
| Sale Price       | (target)               | Response variable for the model                                 |
| Contract Date    | <b>contract_date</b>   | Used to compute <b>sale_age</b> (days from effective date)      |
| Listing Date     | <b>listing_date</b>    | Used with contract date to compute DOM if no DOM column exists  |
| Days on Market   | <b>dom</b>             | Days on market; displayed in exports                            |
| Concessions      | <b>concessions</b>     | Sale concessions; Net SP = Sale Price – Concessions             |
| Living Area (SF) | <b>living_area</b>     | Enables per-SF residuals ( <b>residual_sf</b> , <b>cqa_sf</b> ) |
| Lot Size         | <b>lot_size</b>        | Site size column                                                |
| Site Dimensions  | <b>site_dimensions</b> | Grouped with lot size                                           |
| Latitude         | <b>latitude</b>        | Rounded to 3 dp                                                 |
| Longitude        | <b>longitude</b>       | Rounded to 3 dp                                                 |
| Area ID          | <b>area</b>            | Market area / neighborhood identifier                           |
| Actual Age       | <b>actual_age</b>      | Property age                                                    |
| Effective Age    | <b>effective_age</b>   | Effective property age                                          |
| Address          | <b>display_only</b>    | Shown in exports; excluded from model                           |

Spreadsheet column names can be in a foreign language — the “special” names are in English so that the R program can give them special treatment. Otherwise, the given column names show up in the regression models, graphs and, if doing appraisals, the output reports.

Not all columns are required. mgcvUI adapts — if a column is missing, the corresponding feature is simply omitted. However, for real estate pricing models certain columns are highly recommended:

1. **Sale Age** — the number of days between the contract sale date and the effective date. If multi-year sales history is being used, **sale\_age** often plays a central role.
2. **Living Area** — also goes by names such as “Living Sqft,” “GLA” (gross living area).
3. **Total Bath Count** — the total number of full, quarter, half, and 3/4 bathrooms.
4. **Garage Bays** or **Garage Area** — garage spaces or square footage.

5. **Lot Size** — the land area of the property.
6. **Longitude, Latitude**, and if available **Area ID** for geographic price variation.

## 2.3 Special Column Naming Conventions

mgcvUI identifies columns by their **special type designation**, not by their column name. You can name your columns anything you like in the MLS export — what matters is that you assign the correct special type in the Variable Configuration table (Chapter 6).

## 2.4 Data Quality & Completeness

- **Missing values (NA)**: Rows with NA values in any predictor or target column are automatically removed before fitting. mgcvUI subsets the data to only the columns used in the model, so NAs in unused columns do not cause row drops.
- **Numeric columns**: Sale price, living area, lot size, and similar fields must be numeric.
- **Factor columns**: Text, TRUE/FALSE, and R-factor columns are treated as categorical automatically — they can only enter the model as factors. A **numeric** column (e.g., a numeric area or style code) becomes a factor only when you tick its **Factor** checkbox; mgcvUI never infers categorical from a numeric's value range, so discrete numeric predictors like `bath_count` stay continuous (smoothed). The Special role is independent of the factor decision.

***Tip:** Review the NA column in the predictor table after import. Columns with many missing values may cause rows to be dropped. Columns with  $\geq 50\%$  NAs are flagged red and automatically excluded.*

## 2.5 Subject Row Placement

In **Appraisal** mode, **row 1 must be the subject property**. All remaining rows are comparable sales. The subject row is excluded from model fitting. After fitting, the model still generates predictions for the subject row.

In **Market Area Analysis** mode, placing the subject in row 1 is optional.

In **General** mode, there is no special row handling — all rows are treated equally.

## 3 General Purpose Mode

### 3.1 Overview

General Purpose mode is the default when you launch mgcvUI. It provides the complete GAM workflow for any dataset — not just real estate. You can use mgcvUI for scientific data, financial analysis, engineering studies, or any regression problem where smooth nonlinear relationships are expected.

In General mode, the interface omits the real estate-specific features (special columns, sale age, RCA). The sidebar is streamlined to focus on variable selection, parameter configuration, model fitting, and export.

### 3.2 The Sidebar Workflow

The sidebar is organized into numbered, collapsible sections that guide you from project selection through report export. Most sections appear only once an active project is open, and several are specific to the project's purpose. The output folder is derived from the active project — there is no separate output-folder field.

**1. Project** — Create or open a project. A project sets the purpose and work location (country, state, county, city) and its input (`<os>_in/`) and output folders. Projects are stored under a shared `regProj` root (default `~/regProj`, overridable with the `REGPROJ_ROOT` environment variable or the Settings dropdown) and are shared with the sibling apps earthUI and glmnetUI, so models from all three live side by side. Click **Close Project** to switch.

**2. Import Data** — Select a CSV or Excel file from the active project's input folder. Add files to the folder via Finder/Explorer, then click **Refresh** to list them. For Excel files with multiple sheets, a sheet selector appears. Column names are automatically converted to snake\_case.

**3. Import from earthUI (optional)** — Import an earthUI result `.rds` file to seed the GAM with earth-discovered knot positions (and to seed variable selections). This step is optional — skip it if you do not have an earthUI result to import.

**4. Variable Configuration** — Target variable selector, response transform (none, log, log10), predictor table with checkboxes for Include, Factor, and Linear. The Special column appears only in Appraisal and Market modes. See Chapter 6 for full details.

**5. MgcV Call Parameters** — All model configuration: parameter presets, family, method, gamma, cross-validation, select, basis type, k, tensor type, interaction matrix, and advanced parameters. See Chapter 7 for the complete parameter reference.

**6. Fit MgcV GAM Model** — The button that runs the model. Results appear in the tabs on the right.

**7. Download Output** — Exports predictions, residuals, CQA scores, and per-variable contributions as an Excel file to the project's output folder. Available in all purpose modes once the model is fit.

**8. Calculate RCA Adjustments & Download** — Computes per-variable RCA adjustments for each comparable and interpolates the subject residual via CQA. Appears only in Appraisal and Market modes, after fitting.

**9. Generate Sales Grid & Download** — Generates the Sales Comparison Grid, recommending comparables by gross adjustment percentage. Appears only in Appraisal mode.

**10. Generate Quarto Report** — Writes a self-contained Quarto (`.qmd`) report bundle — with all plots and tables — into the project's output folder. Available once the model is fit.

**11. Convert Quarto Report** — Renders the generated `.qmd` bundle to HTML, Word, or PDF. PDF requires a LaTeX installation; if none is detected, the PDF option is hidden. See Chapter 13.

The purpose-specific steps are omitted (and the remaining steps renumbered) in the other modes: in Market mode the Sales Grid step is absent, and in General mode both the RCA and Sales Grid steps are absent.

### 3.3 Import from earthUI (Optional)

Section 3 of the sidebar — **Import from earthUI** — lets you import an earthUI result `.rds` file. This enables the earth-mgcv pipeline: earth’s data-driven knot positions become anchor points for GAM smooth terms.

When an earthUI result is imported, mgcvUI:

- Displays a summary showing the target variable, R-squared, number of terms, and knot positions per variable.
- Auto-selects the “Earth Pipeline” parameter preset (cubic regression spline basis,  $\gamma = 1.4$ ).
- Sets the target variable to match earth’s target.
- Pre-checks Include for earth’s predictor variables.
- Pre-checks Factor for earth’s categorical variables and Linear for earth’s linear predictors.
- Pre-populates interactions detected by earth (highlighted yellow in the interaction matrix, locked from manual changes).
- Provides an **Export Knots CSV** button to download earth’s knots as a CSV file.

The Variable Configuration table then shows a note explaining that the variable selections (Include, Linear, Factor) and spline knots were *seeded* from the earth model. Unlike glmnetUI, where the glmnet fit is bound to earth’s basis, mgcvUI fits the GAM from your current selections — earth only provides a head start, and every control remains editable.

A **Clear** button removes the imported earth data and resets to standalone mode.

### 3.4 Main Panel Tabs

After data import, the main panel provides the following tabs (model-dependent tabs populate after fitting):

| Tab                 | Contents                                                                                                                        |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Data                | Interactive DataTable of imported data. In appraisal and market modes, split into Subject Property and Comparable Sales tables. |
| Equation            | Full GAM formula rendered in MathJax, plus a table of smooth function definitions.                                              |
| Summary             | Key metrics ( $R^2$ , CV $R^2$ , Dev. Explained, AIC, n, Smooth count) and smooth/parametric terms tables.                      |
| Variable Importance | Horizontal bar chart of F-statistics (smooth) and $ t $ -values (parametric), plus ranked table.                                |
| Contribution        | Interactive plotly smooth partial effect curves with confidence ribbons and earth knot markers.                                 |
| Correlation         | Heatmap of numeric predictor correlations (available before fitting).                                                           |
| Diagnostics         | Residuals vs fitted, Q-Q plot, histogram, response vs fitted (via <b>gratia</b> ). Actual vs Predicted scatter.                 |
| RCA Adjustments     | RCA analysis histograms (appraisal/market mode, after computation).                                                             |
| Anova               | Combined parametric and smooth terms ANOVA table.                                                                               |
| Mgcv Output         | Raw text: timing, formula, model summary, family, concurvity.                                                                   |
| Sign Check          | Earth vs GAM direction consistency check (when earth imported).                                                                 |
| Concurvity          | Overall and pairwise concurvity matrices with color coding.                                                                     |
| Basis Check         | <code>gam.check()</code> basis dimension adequacy tests.                                                                        |

### 3.5 Settings Persistence

mgcvUI automatically saves your configuration to an SQLite database, keyed by the input filename. When you reload the same file, all settings are restored: target selection, predictor checkboxes, data types, mgcv parameters, and interaction matrix. The purpose is a property of the active project rather than a global setting, and the last-used input file is remembered per project and auto-loaded when the project is reopened.

### 3.6 Dark Mode

Click the moon/sun icon in the upper-right corner to toggle between Nord Light and Nord Dark themes. The theme preference is saved in localStorage and persists across sessions. All UI elements adapt to the selected theme.

### 3.7 Locale & Regional Settings

mgcvUI supports international number and CSV formatting conventions through a country-based locale system. The **Settings** dropdown in the title bar provides **Country** and **Paper** selectors for 30+ supported countries. Each preset configures:

- **CSV separator** — comma (,) for US/UK/Japan or semicolon (;) for most of Europe.
- **Decimal mark** — period (.) or comma (,).
- **Thousands separator** — comma (US/UK/Japan), period (Germany/Italy/Spain), space (Finland/France/Poland/Baltics), or apostrophe (Switzerland).
- **Paper size** — Letter (US/Canada/Mexico) or A4 (everywhere else).

Click **Save** to store your locale preferences globally. The Import Data section also has a per-file **Locale** selector for one-off imports that use different conventions. The same Settings dropdown holds the **regProj root folder** field, which relocates the shared project tree used by mgcvUI, earthUI, and glmnetUI.

**Tip:** Number formatting on plot axes and slope labels automatically adapts to your locale. German locale uses periods for thousands (200.000), Finnish uses spaces (200 000), Swiss uses apostrophes (200'000). No currency symbols are displayed — mgcvUI is currency-agnostic.

## 4 Appraisal Mode

When the active project's purpose is **For Appraisal**, mgcvUI configures itself for single-property valuation. All features described in Chapter 3 remain available; this chapter covers only the appraisal-specific additions.

### 4.1 Subject Row Handling

In appraisal mode, **row 1 of your dataset is the subject property** and all remaining rows are comparable sales. Your input file must be organized accordingly (see Chapter 2). The subject's sale price can be left blank or set to any value — mgcvUI automatically treats it as NA during fitting.

After fitting, the model generates predictions for the subject row, which is the basis for the RCA adjustment workflow.

### 4.2 Effective Date & Sale Age

In appraisal and market modes, an **Effective Date** field appears in the Variable Configuration section (defaulting to today's date). If you designate a column as **contract\_date** in the Special dropdown, mgcvUI computes a **sale\_age** column — the number of integer days between each sale's contract date and the effective date. This column is added as a predictor.

When the Effective Date changes, **sale\_age** is automatically recomputed.

### 4.3 Special Column Designations

In appraisal and market modes, a **Special** dropdown appears for each predictor in the Variable Configuration table. See Chapter 6 for the complete reference of special types and their effects.

### 4.4 RCA Adjustments Overview

The **Calculate RCA Adjustments & Download** button (visible in appraisal and market modes after fitting) computes market-derived adjustments for each comparable relative to the subject. The full RCA workflow is described in Chapter 11.

**Tip:** The CQA score you assign to the subject controls how much of the residual distribution is attributed to the subject. A score of 5.00 places the subject at the median of the comparables.

## 5 Market Area Analysis Mode

When the active project's purpose is **Market Area Analysis**, mgcvUI provides the same real estate-specific features as appraisal mode (special columns, sale age) but is oriented toward analyzing a group of properties rather than valuing a single subject.

### 5.1 Differences from Appraisal Mode

- **Subject row handling is flexible** — in market mode, row 1 is not automatically treated as a subject property.
- **No Sales Grid** — the “Generate Sales Grid & Download” step is not available.

### 5.2 When to Use Market Mode

Market Area Analysis mode is appropriate when you are:

- Building a GAM model for a neighborhood or market area to understand nonlinear value drivers
- Analyzing how variables like square footage, age, lot size, and location affect sale prices across a group of properties
- Preparing support for a market conditions analysis or neighborhood delineation
- Working with a dataset where you want special column features (sale age) but do not need the sales grid workflow

**Tip:** Market mode is also useful for general real estate regression where you want special column features but do not need the single-property valuation workflow.

## 6 Variable Selection

Section 4 of the sidebar — **Variable Configuration** — is where you choose which columns participate in the model and how they are treated.

### 6.1 Target Variable

The **Target (response) variable** dropdown at the top of Section 4 lists every numeric column in your dataset. Select one column as the response variable (e.g., sale price). The target column is automatically excluded from the predictor list.

### 6.2 Response Transform

Below the target selector, a **Response Transform** dropdown offers three options:

- **None (raw)** — Use the response as-is. Appropriate when the relationship between predictors and response is approximately additive on the original scale.
- **Log (natural)** — Apply natural log transform before fitting. When back-transformed, this makes adjustments proportional (%) rather than absolute (\$). Particularly useful for sale prices, where a \$10,000 adjustment means very different things at different price levels.
- **Log10** — Apply base-10 log transform.

***Tip:** Log transform makes time adjustments proportional (%) instead of absolute (\$). For real estate models spanning a wide price range, log transforms often produce better-behaved residuals.*

When a log transform is selected, values  $\leq 0$  in the response are automatically filtered out. All output (predictions, contributions, residuals) is automatically back-transformed to the original scale.

### 6.3 The Predictor Table

Below the target selector, a scrollable table lists every remaining column. Checkbox columns use rotated vertical headers for compactness. The column order is:

| Column   | Description                                                                                                                      |
|----------|----------------------------------------------------------------------------------------------------------------------------------|
| Variable | Column name (monospace font, truncated with ellipsis). Bordered cell.                                                            |
| Type     | Data type dropdown: numeric, integer, character, factor, logical, Date, POSIXct. Bordered dropdown.                              |
| Include  | Checkbox — include this column as a predictor in the model. Vertical header.                                                     |
| Factor   | Checkbox — treat as a categorical factor (creates a separate coefficient per level). Vertical header.                            |
| Linear   | Checkbox — force a linear relationship (no smooth, just $\beta x$ ). Vertical header.                                            |
| Special  | Dropdown (appraisal/market only) — see Special Column Types Reference below. Bordered dropdown.                                  |
| NAs      | Count and percentage of missing values, color-coded: red ( $\geq 50\%$ ), orange ( $\geq 20\%$ ), grey ( $> 0$ ). Bordered cell. |

**Factor vs. Smooth vs. Linear:** By default, included numeric variables get a smooth term  $f(x)$ . Checking **Factor** creates a categorical term (one coefficient per level). Checking **Linear** creates a simple linear term  $\beta x$  instead of a smooth. Variables marked as both Factor and Linear are treated as Factor.

## 6.4 Data Type Detection & Overrides

mgcvUI automatically detects data types on import. Numeric, integer, logical, factor, and date columns are recognized. You can override any detection by changing the **Type** dropdown. Changing types affects how the column is treated in the model.

## 6.5 Special Column Types Reference

In appraisal and market modes, the **Special** dropdown provides the following options:

### Weighting:

- **weight** — Observation weight column (only one allowed; rows with `weight = 0` are excluded from fitting)

### Date & Time Types:

- **contract\_date** — Triggers automatic **sale\_age** computation from the Effective Date
- **sale\_age** — Identifies a pre-computed sale-age column (days from the effective date), used as-is
- **listing\_date** — Used as a fallback for computing Days on Market
- **dom** — Identifies the Days on Market column

### Record Type:

- **sale\_type** — Identifies the record-type column that distinguishes the subject from comparable sales

### Monetary Types:

- **concessions** — Identifies sale concessions (seller credits, buyer incentives, etc.)

### Size & Location Types:

- **latitude** — Values automatically rounded to 3 decimal places
- **longitude** — Same rounding treatment as latitude
- **area** — Market area or neighborhood identifier
- **living\_area** — Enables per-square-foot residual calculations (**residual\_sf** and **cqa\_sf**)
- **lot\_size** — Site size column
- **site\_dimensions** — Grouped with lot size

### Age Types:

- **actual\_age** — Property age column
- **effective\_age** — Effective property age

### Display Types:

- **display\_only** — Column is included in Excel exports but excluded from model fitting entirely. Use this for address fields, MLS numbers, parcel IDs, or other reference data.

**Tip:** Columns designated as *display\_only* remain in your dataset and appear in the Excel export, but are excluded from the predictor table entirely. Use this for ID columns, addresses, or other reference fields.

## 7 Parameter Selection

Section 5 of the sidebar — **MgcV Call Parameters** — provides access to all configuration options for the GAM. Each parameter has a blue help icon (?) with a tooltip explanation.

### 7.1 Parameter Presets

A dropdown at the top offers two presets that configure sensible defaults for common workflows:

| Preset                         | Basis                 | Settings                                  |
|--------------------------------|-----------------------|-------------------------------------------|
| Standalone<br>(discovery)      | tp (thin plate)       | k = auto (10), gamma = 1.2, select = TRUE |
| Earth Pipeline<br>(refinement) | cr (cubic regression) | k = auto (10), gamma = 1.4, select = TRUE |

The **Earth Pipeline** preset is automatically selected when earthUI knots are imported. The cubic regression spline (**cr**) basis is required for earth knot integration because it allows specifying exact knot positions.

### 7.2 Family

Choose the distribution family for your response variable:

| Family                  | Use Case                                                              |
|-------------------------|-----------------------------------------------------------------------|
| <b>gaussian</b>         | Continuous responses (e.g., sale price). Most common.                 |
| <b>Gamma</b>            | Positive continuous responses with variance proportional to the mean. |
| <b>poisson</b>          | Count data (e.g., number of sales).                                   |
| <b>binomial</b>         | Binary outcomes.                                                      |
| <b>inverse.gaussian</b> | Highly right-skewed positive data.                                    |

### 7.3 Method

The smoothing parameter estimation method:

| Method         | Description                                                                            |
|----------------|----------------------------------------------------------------------------------------|
| <b>REML</b>    | Restricted Maximum Likelihood (default, recommended). Most robust against overfitting. |
| <b>GCV.Cp</b>  | Generalized Cross-Validation. Tends to select slightly more complex models.            |
| <b>ML</b>      | Maximum Likelihood. Similar to REML but can undersmooth.                               |
| <b>P-REML</b>  | Pearson REML variant.                                                                  |
| <b>P-ML</b>    | Pearson ML variant.                                                                    |
| <b>GACV.Cp</b> | Generalized Approximate Cross-Validation.                                              |

## 7.4 Gamma (Smoothing Multiplier)

The gamma parameter multiplies the effective degrees of freedom in the smoothing criterion, encouraging smoother (less wiggly) fits. Default is 1.2.

| Value   | Effect                                               |
|---------|------------------------------------------------------|
| 1.0     | Standard smoothing                                   |
| 1.2–1.4 | Slightly smoother than default (recommended)         |
| 2.0+    | Much smoother — good for noisy data or small samples |

Higher gamma values guard against overfitting by penalizing complexity more heavily. The Earth Pipeline preset uses 1.4 for additional smoothing when refining earth’s knots.

## 7.5 Cross-Validation

When checked (default), mgcvUI runs 10-fold cross-validation after fitting to compute a CV R-squared. This provides an honest estimate of out-of-sample predictive power.

## 7.6 Select (Shrinkage)

When checked (default), adds an extra penalty to each smooth term that can shrink it to zero. This enables automatic variable selection — unimportant smooth terms are effectively removed from the model. Implemented via `select = TRUE` in `gam()`.

## 7.7 Default Basis

The spline basis function type for smooth terms:

| Basis     | Full Name                    | Description                                                                      |
|-----------|------------------------------|----------------------------------------------------------------------------------|
| <b>tp</b> | Thin plate regression spline | Default. Isotropic (no knot placement needed). Good general-purpose choice.      |
| <b>cr</b> | Cubic regression spline      | Allows explicit knot placement. Required for earth knot integration.             |
| <b>ps</b> | P-spline                     | Evenly-spaced B-spline basis with difference penalty. Computationally efficient. |
| <b>bs</b> | B-spline                     | Flexible B-spline basis.                                                         |

## 7.8 Default k (Basis Dimension)

The basis dimension  $k$  controls the maximum wiggleness of each smooth. A value of 0 (the default) means “automatic” — mgcvUI uses  $k = 10$  or the number of earth knots, whichever is appropriate.

- **Small k (3–5):** Very smooth curves, nearly linear.
- **Medium k (8–15):** Moderate flexibility (usually sufficient).
- **Large k (20+):** Very flexible — risk of overfitting without strong regularization.

**Tip:** The effective degrees of freedom (EDF) in the Summary tab tells you how many degrees of freedom each smooth actually uses. If EDF is close to  $k - 1$ , the basis may be too small — increase  $k$ . If EDF is much less than  $k$ , the smooth is well-penalized.

## 7.9 Tensor Type

For interactions between continuous variables, two tensor product types are available:

| Type      | Function                | Description                                                                                                                                                         |
|-----------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ti</b> | <code>ti(x1, x2)</code> | Tensor interaction — models only the interaction effect, with main effects handled separately by univariate <code>s()</code> terms. Preferred for interpretability. |
| <b>te</b> | <code>te(x1, x2)</code> | Tensor product — models the entire joint effect including main effects. Harder to decompose for RCA adjustments.                                                    |

## 7.10 Allowed Interactions

A collapsible **Allowed Interactions** section displays an upper-triangular checkbox matrix for all included smooth (non-linear, non-factor) predictors. Each checkbox enables a tensor product smooth between the corresponding variable pair.

- **Allow All / Clear All** — Bulk toggle buttons.
- **Click a variable name** — Toggle all interactions for that variable.
- **Right-click a variable name** — “Block from main effect” — marks the variable for interaction-only use (no univariate smooth).

When earthUI knots are imported, earth-detected interactions are pre-checked and locked (highlighted with a yellow background). If the earth model used degree = 1 (no interactions), an informational message is shown.

### Warning

Enabling tensor interactions increases model complexity substantially. Use interactions when you have domain knowledge or earth-detected evidence supporting them. Verify that the resulting adjustments are reasonable and explainable.

## 7.11 Factor-by-Smooth Interactions

A separate **Factor-by-Smooth Interactions** matrix appears when both factor and smooth variables are included. Each checkbox creates a `s(x, by = factor)` term — a separate smooth curve for each factor level.

## 7.12 Advanced Parameters

A collapsible “Advanced” section exposes additional settings:

- **Optimizer** — Algorithm for smoothing parameter estimation: `outer/newton` (default), `outer/bfgs`, or `efs` (extended Fellner-Schall).
- **Scale** — Scale parameter. 0 (default) = estimate from data. Set to a known value for fixed-scale models.
- **Discrete covariate method** — Fast discretized fitting for large datasets (`discrete = TRUE`).
- **Threads** — Number of parallel threads (1–32). Only effective when `discrete = TRUE`.

## 8 Fitting the Model

### 8.1 The Fit Button

Section 6 of the sidebar contains the **Fit MgcV GAM Model** button. Clicking it runs the model with your current configuration.

### 8.2 What Happens During Fitting

When you click Fit, mgcvUI:

1. **Validates the configuration** — checks for at least one predictor and at least 10 complete rows. Warns if fewer than 50% of rows are complete (listing the columns responsible for most NAs).
2. **Prepares the data** — subsets to model-relevant columns only, removes rows with NAs, applies weights (if designated), excludes the subject row (in appraisal mode), applies response transforms.
3. **Builds the GAM formula** — constructs smooth terms `s(x, bs="cr", k=10)` for numeric predictors, factor terms for categoricals, linear terms for variables marked Linear, and tensor products for selected interactions. Earth knots are embedded as explicit knot positions in `cr` basis terms.
4. **Handles edge cases** — drops constant variables, converts binary predictors to factors, caps  $k$  to the number of unique values minus 1, augments earth knots with data quantiles when more knots are needed.
5. **Fits the GAM** — calls `mgcv::gam()` with the constructed formula, family, method, gamma, select, and other parameters.
6. **Runs cross-validation** (if enabled) — 10-fold CV computing out-of-sample R-squared.
7. **Updates all result tabs** — Summary, Equation, Contribution plots, Diagnostics, and all other tabs are populated.

### 8.3 Background Fitting & Progress

When the `callr` package is available, model fitting runs in a background process so the app remains responsive. A dark-themed terminal-style progress overlay shows:

- **Timer** displaying elapsed seconds/minutes.
- **Color-coded trace log:** blue for status, yellow for CV fold progress, red for errors, green for results.
- An **Abort** button to kill the fitting process if needed.
- A **Close** button that appears after completion.

If `callr` is not available, fitting runs synchronously (the app freezes until fitting completes).

After successful fitting, a white checkmark appears on the Fit button and a status line shows: “R-sq = X, CV R-sq = X, Dev = X%, AIC = X, n = X.”

## 9 Result Tabs

### 9.1 Data

Shows the imported data as an interactive DataTable with horizontal scrolling and 15 rows per page. In appraisal and market modes the tab is split into a **Subject Property** table (row 1) and a **Comparable Sales** table (the remaining rows); in General mode a single table is shown.

Each cell is kept to a single line and truncated to the column width, so wide free-text fields (such as property remarks) do not stretch a row down the page. Single-click selects a row; **double-click any cell** to open a pop-up showing its full, untruncated contents.

### 9.2 Equation

Displays the fitted GAM model in two parts:

**Model equation** (MathJax-rendered):

- Left-hand side shows the response with appropriate transform (log, log10) and link function.
- Right-hand side shows the intercept value, smooth terms as  $f_i(\text{variable})$ , factor terms grouped by variable with level counts, and linear terms with coefficient values.
- Family and method are displayed above the equation.

**Smooth Function Definitions** table:

- Term formula string (e.g., `s(living_sqft, bs="cr", k=8)`)
- Variable name
- Type (s, te, ti, or linear)
- Basis function type
- $k$  value
- Earth knot positions (if imported)

### 9.3 Summary

Six metric cards displayed across the top:

- **R-squared** — proportion of deviance explained
- **CV R-squared** — cross-validation R-squared (if CV enabled)
- **Dev. Explained %** — percentage of deviance explained
- **AIC** — Akaike Information Criterion
- **n** — number of observations used
- **Smooths** — number of smooth terms

Below the cards:

- **Smooth Terms table** — Term, EDF (effective degrees of freedom), Ref.df, F statistic, p-value
- **Parametric Terms table** — Term, Estimate, Std Error, t-value, p-value

### 9.4 Variable Importance

A horizontal bar chart showing the relative importance of each model term:

- **Smooth terms:** Ranked by F-statistic (blue bars)
- **Parametric terms:** Ranked by  $|t|$ -value (green bars)

Below the chart, a DataTable shows Term, Type, EDF, Statistic, and p-value, sorted by statistic descending.

## 9.5 Contribution Plots

Every model term that contributes to the predicted value has an interactive **plotly** contribution plot, displayed in a bordered card. The tab shows plots for all term types:

### 9.5.1 Univariate Smooth Terms — $s(x)$

- **Blue line** with 95% confidence ribbon
- **Grey scatter points** showing partial residuals (for non-log-transformed models)
- **Red dashed vertical lines** at earth knot positions (if earth imported)
- **Hover tooltip**: Variable value, Contribution (\$), 95% CI, Rate (slope per unit)

For **log-transformed models**, the y-axis is back-transformed to dollar contributions using the formula  $\bar{y} \times (e^{f(x)} - 1)$ , making the curves directly interpretable in dollar terms.

For **latitude/longitude** variables, the slope is shown per 0.001 degrees rather than per degree, since that is a more meaningful geographic increment.

### 9.5.2 Factor-by-Smooth Terms — $s(x, \text{by} = \text{factor})$

One colored line per factor level, overlaid on a single plot. Useful for seeing how a predictor's effect varies across groups (e.g., different neighborhoods).

### 9.5.3 Interaction Terms — $ti(x_1, x_2)$ and $te(x_1, x_2)$

**Heatmap** showing the 2D contribution surface. Color scale: red (negative) → white (zero) → blue (positive). Hover shows both variable values and the contribution amount.

### 9.5.4 Parametric Terms — Linear and Factor

- **Numeric linear terms**: scatter plot of data points with the fitted line overlaid
- **Factor terms**: bar chart showing mean contribution per factor level

## 9.6 Correlation

A heatmap matrix showing Pearson correlations among all numeric predictors. Available immediately after data import — no model fitting required.

- Color gradient: blue (−1) → white (0) → red (+1)
- Correlation values printed in each cell
- Useful for identifying concurvity (the smooth analogue of multicollinearity) before fitting

## 9.7 Diagnostics

Two sets of diagnostic visualizations:

**Diagnostic panel** (via `gratia::appraise`): Four plots — residuals vs fitted, Q-Q plot, histogram of residuals, and response vs fitted values.

**Actual vs Predicted** scatter plot: Observed vs predicted values with a 45-degree dashed reference line. For log-transformed models, values are back-transformed to the original scale.

## 9.8 RCA Adjustments

Three histogram plots displayed after RCA computation (appraisal/market mode only):

- **Residual Adjustment %** — Distribution of residual adjustments as a percentage of sale price
- **Net Adjustment %** — Distribution of net adjustments

- **Gross Adjustment %** — Distribution of gross adjustments

Each histogram shows mean, median, and standard deviation in the subtitle, with dashed reference lines for mean and median.

## 9.9 Anova

A combined ANOVA table merging parametric and smooth terms from the GAM summary. Shows Term, Type (parametric or smooth), and all associated statistics.

## 9.10 Mgcvc Output

Raw text output showing:

- Fitting timing
- The GAM formula
- Model print output
- Full `summary(model)` output
- Family and link function
- Overall concurvity measures

## 9.11 Sign Check

When earthUI knots are imported, this tab compares the direction of each variable in the earth model with its direction in the GAM:

- **earth\_direction**: increasing, decreasing, or mixed (based on hinge function signs)
- **gam\_direction**: increasing, decreasing, or mixed (estimated from smooth derivative on a 100-point grid)
- **consistent**: TRUE/FALSE
- **warning**: Description of any mismatch

A status line shows “All directions consistent” or the number of variables with inconsistencies.

## 9.12 Concurvity

Concurvity is the smooth analogue of multicollinearity. This tab shows:

- **Overall Concurvity table**: Rows for worst/observed/estimate measures, columns for each smooth term. Color-coded: white ( $< 0.5$ ), yellow ( $0.5-0.8$ ), red ( $> 0.8$ ).
- **Pairwise Concurvity table**: Worst-case pairwise matrix with the same color coding.

Values above  $\sim 0.8$  in the “worst” row indicate that a smooth can be well-approximated by the other smooths, suggesting redundancy.

## 9.13 Basis Check

Output from `mgcv::gam.check()`: basis dimension adequacy tests for each smooth term. If a smooth’s  $k$  is too small, this test will flag it with a low p-value, suggesting you should increase  $k$ .

## 10 Downloading Data

After fitting, the **Download Output (Excel)** button (sidebar section 7) exports an Excel file with predictions and diagnostics to the active project's mgcv output folder.

### 10.1 Output Columns

| Column             | Description                                                                                            |
|--------------------|--------------------------------------------------------------------------------------------------------|
| basis              | Back-transformed intercept value (same for all rows)                                                   |
| <var>_contribution | Per-smooth-term contribution to prediction. For log models, proportionally allocated to dollar values. |
| est_<target>       | Model prediction (e.g., <code>est_sale_price</code> ), back-transformed from log scale if applicable   |
| residual           | Actual – predicted                                                                                     |
| calc_residual      | Verification: actual – sum-of-contributions (back-transformed)                                         |
| cqa                | Condition-Quality-Appeal (CQA) score (0–10 scale)                                                      |
| residual_sf        | Residual / living area (if <code>living_area</code> designated)                                        |
| cqa_sf             | CQA calculated from ranking via <code>residual_sf</code>                                               |

In appraisal mode, row 1 (subject) has residual/cqa/cqa\_sf set to NA. Rows are sorted by `residual_sf` descending when a `living_area` column is designated.

A white checkmark appears on the download button after successful completion.

### 10.2 CQA Scores

CQA ranks each row's residual against all others on a 0–10 scale:

- **High CQA (~9–10):** sold for much more than predicted — likely superior condition/quality/appeal
- **Low CQA (~0–1):** sold for much less than predicted — likely inferior condition or distressed sale
- **CQA ~5:** near the median

## 11 RCA Calculations & Downloading

The RCA (Residual Constraint Approach) workflow is available in **Appraisal** and **Market Area Analysis** modes, after fitting the model. The subject property must be in row 1.

### 11.1 Opening the RCA Dialog

Click the **Calculate RCA Adjustments & Download** button in sidebar section 8. A modal dialog appears with:

- **Score type:** CQA or CQA per SF (if `living_area` designated). The per-SF option scales the subject's residual by living area.
- **Subject CQA Score:** Accepts values from 0.00 to 10.00. The field starts blank; when an earthUI model has been imported, the score type and value you used in earthUI are carried forward from the shared project settings.

Click **Generate** to compute and download. The Excel file is written to the active project's `mgcv` output folder (`<os>_out_mgcgv`).

### 11.2 CQA Score Interpolation

1. Comparables' CQA scores and residuals are sorted
2. Linear interpolation (`approx()`) maps your CQA value to a residual
3. Subject value = model prediction + interpolated residual
4. Weight-0 rows are treated as if they had the same residual as the subject

### 11.3 Output Columns

| Column                              | Description                                           |
|-------------------------------------|-------------------------------------------------------|
| <code>subject_value</code>          | Model prediction + interpolated residual              |
| <code>subject_cqa</code>            | User-entered CQA score                                |
| <code>&lt;var&gt;_adjustment</code> | Subject contribution – comp contribution (in dollars) |
| <code>residual_adjustment</code>    | Subject residual – comp residual                      |
| <code>net_adjustments</code>        | Sum of all adjustments                                |
| <code>gross_adjustments</code>      | Sum of absolute adjustments                           |
| <code>residual_adj_pct</code>       | Residual adjustment as % of sale price                |
| <code>net_adj_pct</code>            | Net adjustments as % of sale price                    |
| <code>gross_adj_pct</code>          | Gross adjustments as % of sale price                  |
| <code>adjusted_sale_price</code>    | Actual sale price + net adjustments                   |

A white checkmark appears on the button after successful computation.

## 12 Sales Comparison Grid

The Sales Comparison Grid is available in **Appraisal** mode only, after computing RCA adjustments. It generates a formatted Excel workbook suitable for inclusion in appraisal reports.

### 12.1 Comp Selection Dialog

Clicking the **Generate Sales Grid & Download** button opens a modal dialog listing all comparable sales. Comps are split into two groups:

- **Recommended comps** (pre-checked): Gross adjustment < 25% of sale price, sorted by sale age (most recent first). Maximum 30 comps.
- **Other comps** (unchecked by default): Remaining comps, sorted by gross adjustment percentage.

Select your comps, then click **Generate Sales Grid** to create the workbook.

**Required special types:** `contract_date` and `living_area` must be designated. **Recommended special types:** `latitude`, `longitude`, and `lot_size` — the grid will work without them but some fields will be blank.

### 12.2 Workbook Layout

The generated workbook contains multiple sheets, each holding 3 comps side by side:

- **Header rows:** Subject and comp addresses, sale prices, sale dates, and Days on Market
- **Grouped rows:** Location (latitude, longitude, area), Site Size (lot size, site dimensions), and Age (actual age, effective age) with group headers and sub-items
- **Model variable rows:** One row per smooth term showing the subject contribution, comp contributions, and adjustment amounts
- **Residual feature rows:** Empty, unlocked cells for appraiser input on features not captured by the model (e.g., condition, quality, view)
- **Summary rows:** Net Adjustment %, Gross Adjustment %, and Adjusted Sale Price with live Excel formulas

### 12.3 Sheet Protection & Residual Cells

Each sheet is protected to prevent accidental edits, but the residual feature value cells are explicitly unlocked. This allows appraisers to enter values for features not in the model while preserving the formula-driven adjustment calculations.

## 13 Downloading Reports

Report generation is a two-step Quarto workflow, replacing the single-step report of earlier releases.

### 13.1 Step 1: Generate Quarto Report

The **Generate Quarto Report** button (available once the model is fit) writes a self-contained `.qmd` bundle to the active project's `mgcv` output folder. The bundle holds the Quarto source, pre-rendered plot files (PNG and PDF), a `report_data.rds` payload with all tables and model metadata, and a `reference.docx` style template for Word output. Because the bundle is self-contained, you can edit the `.qmd` source or combine it with other Quarto documents before rendering.

Plots are pre-rendered when the bundle is generated; if an individual plot fails, a placeholder image is written and the rest of the bundle generates normally.

### 13.2 Step 2: Convert Quarto Report

The **Convert Quarto Report** section renders a `.qmd` file to one or more output formats via `quarto::quarto_render()`. It defaults to the most recently generated bundle, but the **Browse** button lets you select any `.qmd` file. Three formats are available:

- **HTML** — Self-contained HTML document. Works on all platforms with no extra software. Recommended default.
- **Word (.docx)** — Rendered with the bundled `reference.docx` style template. Suitable for editing and distribution.
- **PDF** — **Requires a LaTeX installation** (see System Requirements below). If no LaTeX is detected, the PDF option is automatically hidden from the format checkboxes.

The legacy single-step renderers, `render_gam_report()` and `export_gam_docx()`, remain available for programmatic use.

### 13.3 Report Contents

The generated report includes:

- **Dataset & Model:** Data file, sample size, family, method, and fitting information
- **Results Summary:**  $R^2$ , CV  $R^2$ , Deviance Explained, AIC, and related fit metrics
- **Model Equation:** Model equation with  $f_i$  notation for smooth terms and explicit coefficients for linear terms, plus the smooth function definitions. Long equations wrap across multiple lines in PDF.
- **Summary tables:** Smooth terms (EDF, F, p-value) and parametric terms (Estimate, Std Error, t, p-value)
- **Enabled interactions:** A matrix of the tensor and factor-by-smooth interaction terms included in the model (shown only when interactions are present)
- **Variable Importance:** Bar chart ranked by F-statistic (smooths) or  $|t|$  (parametric)
- **Smooth Effects** (all term types):
  - Univariate smooth plots (static ggplot2 versions of the interactive plots)
  - Factor-by-smooth multi-line plots
  - Interaction heatmaps ( $ti$  and  $te$  terms)
  - Parametric term plots (scatter for numeric, bar chart for factor)
- **Predictor Correlations** heatmap
- **Diagnostics:** Residual diagnostics and actual vs predicted scatter

- **mgcv Summary:** The full text output of the fitted model
- **Notation appendix:** An unnumbered closing section explaining the value-contribution function notation used in the report

A white checkmark appears on the Generate Quarto Report button after successful generation.

## 14 Exporting Model Functions

mgcvUI can export the fitted GAM's smooth functions as standalone code in multiple languages. Each smooth is evaluated on a 200-point grid and exported as a lookup table with linear interpolation, allowing the model to be used outside of R without any dependency on `mgcv`.

Function export is programmatic in the current release. The exported helpers take a fitted result object (produced by `fit_gam()`) and write the files described below:

- `generate_r_code()`, `generate_python_code()`, and `generate_cpp_code()` — per-language source files
- `export_functions_sqlite()` — SQLite lookup-table database
- `export_functions_zip()` — bundles any combination of the four formats into a zip archive

### 14.1 R Functions

File: `gam_functions.R`

- Header comments with response variable,  $R^2$ ,  $CV R^2$ , and the formula.
- An `INTERCEPT` constant.
- One `approxfun()` per smooth variable with pre-computed x/y vectors.
- Linear term coefficients as named constants.
- Usage: `sum the intercept, each g_var(value), and each coef_var * value.`

### 14.2 Python Functions

File: `gam_functions.py`

- `numpy` import.
- `INTERCEPT` constant.
- Per-variable: `_x_var` and `_fx_var` arrays, `g_var(x)` function using `np.interp()`.
- Linear term coefficients as `COEF_*` constants.

### 14.3 C++ Header

File: `gam_functions.hpp`

- Header-only with `#pragma once`.
- `gam_functions` namespace.
- `interp()` helper using `std::lower_bound`.
- Per-variable: inline function with static const arrays.
- Linear term coefficients as `constexpr double`.

### 14.4 SQLite Database

File: `gam_functions.sqlite`

- Table `smooth_grids`: columns (variable, x, fx) — the 200-point lookup tables.
- Table `model_info`: response, `r_squared`, `cv_rsqr`, `dev_explained`, `aic`, `n_obs`, family, method, formula.
- Table `linear_terms`: variable, coefficient.
- Table `intercept`: intercept value.

## 15 Comparison with earthUI and glmnetUI

The earth application and its Post Processing Modules are one toolset for regression modeling. They share the same data format, special column types, RCA workflow, and demo datasets, but use different modeling engines.

### 15.1 Key Differences

| Feature            | mgcvUI                                | earthUI                        | glmnetUI             |
|--------------------|---------------------------------------|--------------------------------|----------------------|
| Method             | GAM (mgcv)                            | MARS/Earth (earth)             | Elastic net (glmnet) |
| Relationships      | Smooth nonlinear                      | Piecewise linear               | Linear               |
| Variable selection | <code>select=TRUE</code><br>shrinkage | Stepwise pruning               | Lasso penalty        |
| Interactions       | Tensor products<br>(te/ti)            | Hinge products<br>(degree 2–3) | Pairwise products    |
| Partial effects    | Smooth curves with CI                 | g-functions (piecewise)        | Linear slopes        |
| Earth pipeline     | Imports knots from earth              | Source of knots                | Imports hinge basis  |
| Code export        | R, Python, C++,<br>SQLite             | R, Python, C++,<br>SQLite      | —                    |
| Report formats     | Word, PDF, HTML                       | Word, PDF, HTML                | Word, PDF, HTML      |

One earthUI-only differentiator worth noting: earthUI includes an experimental Prolog/DCG feature that extracts structured property attributes from free-text MLS remarks.

### 15.2 Shared Features

All three tools provide:

- A shared `regProj` project tree (default `~/regProj`) — projects, input files, and per-method outputs are shared across the sibling apps
- CSV and Excel import with `snake_case` column conversion
- 30+ country locale support with CSV separator, decimal mark, and paper size
- Special column designations (`contract_date`, `living_area`, `area`, `latitude`, `longitude`, etc.)
- Automatic `sale_age` computation from effective date
- Dark/light mode toggle (Nord theme)
- Settings persistence (SQLite and/or `localStorage`)
- Correlation heatmap, variable importance, diagnostics
- CQA scoring and RCA adjustment workflow
- Sales Comparison Grid with comp selection and formatted Excel output
- Compatible with the same demo datasets (`Appraisal_1.csv`)

### 15.3 Recommended Workflow

- **Start with earthUI** to discover the important variables, nonlinear relationships, and interactions. Earth’s automatic model building provides the best starting point.
- **Refine with mgcvUI** if you want smoother partial effects. Import earthUI’s knots for a seamless transition.
- **Use glmnetUI** when you need a purely linear model with aggressive variable selection or coefficient sign constraints.
- **Compare all three** on the same data to assess whether the additional complexity of nonlinear models is justified.

## 16 Demo Dataset: Appraisal\_1.csv

### 16.1 Description

The demo dataset is shared with earthUI and glmnetUI. It is located in the `demo_mls/` folder of the mgcvUI source, or if earthUI is installed:

```
demo_file <- system.file("extdata", "Appraisal_1.csv", package = "earthUI")
```

The file contains roughly 1,500 residential sale records from a simulated MLS export, with the subject property in row 1 and the comparable sales in the remaining rows. The data represents single-family home sales in a multi-area market with a range of property sizes, ages, and locations.

This is not real data, but is based on a realistic neighborhood in Northern California. All identification information has been altered or removed.

### 16.2 Columns

| Column              | Type      | Special Type  | Description                                                       |
|---------------------|-----------|---------------|-------------------------------------------------------------------|
| sale_type           | character | sale_type     | Record-type code distinguishing the subject from comparable sales |
| weight              | numeric   | weight        | Observation weight (0 = exclude from fitting)                     |
| id                  | numeric   | display_only  | Internal record ID                                                |
| property_id         | numeric   | display_only  | MLS property identifier                                           |
| listing_id          | character | display_only  | MLS listing number                                                |
| parcel_number       | character | display_only  | County assessor parcel number (APN)                               |
| street_address      | character | display_only  | Property address                                                  |
| city_name           | character | display_only  | City                                                              |
| postal_code         | character | display_only  | ZIP code                                                          |
| county_name         | character | display_only  | County                                                            |
| contract_date       | Date      | contract_date | Sale contract date (computes sale_age)                            |
| sale_age            | numeric   | —             | Days from contract date to effective date (pre-computed)          |
| coe_date            | Date      | display_only  | Close of escrow date                                              |
| listing_status      | character | display_only  | Listing status (e.g., “Sold”)                                     |
| sale_price          | numeric   | (target)      | Sale price — response variable                                    |
| rent                | numeric   | —             | Monthly rent (for multi-target models)                            |
| list_price          | numeric   | display_only  | Listing price                                                     |
| original_list_price | numeric   | display_only  | Original listing price                                            |
| living_sqft         | numeric   | living_area   | Gross living area in square feet                                  |
| ppsf                | numeric   | display_only  | Price per square foot                                             |
| beds_total          | integer   | —             | Number of bedrooms                                                |
| baths_total         | numeric   | —             | Total bath count (e.g., 2.5 = 2 full + 1 half)                    |
| lot_size            | numeric   | lot_size      | Lot size in square feet                                           |
| area_id             | integer   | area          | MLS area identifier                                               |
| area_text           | character | display_only  | Area name                                                         |
| age                 | numeric   | actual_age    | Property age in years                                             |
| year_built          | integer   | display_only  | Year of construction                                              |
| latitude            | numeric   | latitude      | Latitude (rounded to 3 dp for model)                              |
| longitude           | numeric   | longitude     | Longitude (rounded to 3 dp for model)                             |
| latitude6           | numeric   | display_only  | Full-precision latitude                                           |
| longitude6          | numeric   | display_only  | Full-precision longitude                                          |
| garage_spaces       | integer   | —             | Number of garage bays                                             |
| fp_count            | integer   | —             | Number of fireplaces                                              |
| no_of_stories       | numeric   | —             | Number of stories                                                 |
| style               | character | —             | Architectural style                                               |

| Column                        | Type      | Special Type              | Description                               |
|-------------------------------|-----------|---------------------------|-------------------------------------------|
| <code>view</code>             | character | —                         | View type (e.g., “Neighborhood”, “Hills”) |
| <code>days_on_market</code>   | integer   | <code>dom</code>          | Days on market                            |
| <code>listing_date</code>     | Date      | <code>listing_date</code> | Listing date                              |
| <code>sale_concessions</code> | numeric   | <code>concessions</code>  | Seller concessions                        |

### 16.3 Suggested Quick Start

1. Launch the mgcv routine: `earthUI::launch_mgcv()`
2. Create a new project (Section 1) with purpose **For Appraisal**
3. Place `Appraisal_1.csv` in the project’s input folder and select it in Section 2 (click **Refresh** if needed)
4. Select `sale_price` as the target
5. Optionally set Response Transform to **Log (natural)** for proportional adjustments
6. Assign special types as shown in the table above
7. Include predictors: `sale_age`, `living_sqft`, `baths_total`, `lot_size`, `area_id` (as factor), `age`, `latitude`, `longitude`, `garage_spaces`
8. Keep defaults: REML,  $\gamma = 1.2$ , tp basis,  $k = \text{auto}$ , `select = TRUE`
9. Click **Fit Mgcv GAM Model**
10. Review Summary, Contribution (smooth plots), and Diagnostics tabs
11. Download output (Step 7), review the CQA ranking
12. Compute RCA adjustments (Step 8) with a CQA score of ~5.00
13. Generate a Sales Comparison Grid (Step 9) with recommended comps

## 17 System Requirements & Troubleshooting

### 17.1 Supported Platforms

mgcvUI runs on **macOS**, **Windows**, and **Linux**. The application is developed and primarily tested on macOS with RStudio. Platform-specific notes are provided below.

### 17.2 R Version

$R \geq 4.1.0$  is required. RStudio Desktop (2023.06 or later) is strongly recommended — it bundles **Quarto** and **pandoc** (needed for report rendering) and provides a convenient environment for launching the app.

### 17.3 Required R Packages

The following packages are installed automatically as dependencies:

| Package           | Purpose                                                      |
|-------------------|--------------------------------------------------------------|
| shiny             | Web application framework                                    |
| mgcv              | GAM model fitting                                            |
| valengrCore       | Shared appraisal core (special roles, earthUI carry-forward) |
| ggplot2, plotly   | Static and interactive contribution plots                    |
| gratia            | Smooth term extraction and diagnostics                       |
| DT                | Interactive data tables                                      |
| quarto, rmarkdown | Quarto report generation and rendering                       |
| readxl            | Excel file import                                            |
| shinyFiles        | File and folder browsers                                     |
| RSQLite, DBI      | Settings and project databases (SQLite)                      |
| jsonlite          | Variable state serialization                                 |

### 17.4 Optional Dependencies

| Component                                 | When Needed                                                                                                                                                                              |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LaTeX</b> (TinyTeX, MiKTeX, or MacTeX) | PDF report generation only. If no LaTeX installation is detected, the PDF option is automatically hidden from the format dropdown. Install with: <code>tinytex::install_tinytex()</code> |
| <b>sysfonts + showtext</b>                | Roboto Condensed font for ggplot2 plots. If unavailable or offline, the app falls back to the system sans-serif font automatically.                                                      |
| <b>earth</b> ( $\geq 5.3.0$ )             | Only needed for the earthUI import pipeline. Not required for standalone use.                                                                                                            |
| <b>callr</b>                              | Background model fitting (keeps the app responsive). Without it, fitting runs synchronously.                                                                                             |
| <b>writexl, openxlsx</b>                  | Excel downloads (model output, RCA, sales grid). The app prompts if missing.                                                                                                             |
| <b>bslib</b>                              | Bootstrap 5 theming (Nord Light/Dark).                                                                                                                                                   |
| <b>officer</b>                            | Legacy programmatic Word export ( <code>export_gam_docx()</code> ).                                                                                                                      |

## 17.5 Platform-Specific Notes

### 17.5.1 macOS

Fewest issues. Homebrew is recommended for system libraries. If PDF reports are needed:

```
# In R:
install.packages("tinytex")
tinytex::install_tinytex()
```

### 17.5.2 Windows

Works well with RStudio Desktop. Key notes:

- **Corporate/locked-down machines:** If the temp directory or user data directory is restricted, settings persistence and report generation may fail. The app will display a warning in the R console but continue operating without persistence.
- **LaTeX:** Install TinyTeX from R (same command as macOS above) or install MiKTeX from [miktex.org](http://miktex.org).
- **Long file paths:** Windows has a 260-character path limit. Keep your working directory path short.

### 17.5.3 Linux

Most variable across distributions. Ubuntu/Debian users may need system libraries before R packages will compile:

```
sudo apt install libcurl4-openssl-dev libssl-dev \
  libxml2-dev libsqlite3-dev libfontconfig1-dev
```

For PDF reports: `tinytex::install_tinytex()` or install `texlive-full` from the system package manager.

For headless servers (no display), plotly interactive plots render as static fallbacks in reports. The Shiny app itself requires a web browser connection.

## 17.6 Graceful Degradation

mgcvUI is designed to work even when optional components are missing:

| Missing Component                 | Behavior                                                                                                                                |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| No LaTeX                          | PDF option hidden from report format dropdown. HTML and Word still available.                                                           |
| No internet / fonts fail          | Roboto Condensed font replaced by system sans-serif. Console message logged.                                                            |
| No RSQLite / read-only filesystem | Settings do not persist between sessions. Console warning displayed. App runs normally otherwise.                                       |
| gratia::appraise fails            | Diagnostics tab and reports show a placeholder message. Use <code>mgcv::gam.check()</code> in the console as an alternative.            |
| Temp directory not writable       | Report generation fails with a clear error message. Resolve by setting <code>TMPDIR</code> environment variable to a writable location. |

## 17.7 Troubleshooting

### 17.7.1 “PDF option is not available”

No LaTeX installation detected. Run in R:

```
install.packages("tinytex")
tinytex::install_tinytex()
```

Then restart the app. The PDF option will appear.

### 17.7.2 “Settings will not persist”

The SQLite database directory cannot be created or written to. Check permissions on:

- **macOS/Linux:** ~/Library/Application Support/R/mgcVUI/ or ~/.local/share/R/mgcVUI/
- **Windows:** %APPDATA%/R/data/mgcVUI/

### 17.7.3 “Diagnostics plot unavailable”

The `gratia::appraise()` function occasionally fails on complex models (many tensor interactions, non-standard families). This is a known upstream issue. Workaround: run `mgcv::gam.check(model)` in the R console to see the diagnostic plots directly.

### 17.7.4 “Report error: number of items to replace...”

This was a known issue in earlier versions related to the `gratia` package. Ensure you have the latest version of `mgcvUI` installed. The current version catches this error and produces the report with a placeholder for the diagnostics section.

### 17.7.5 Port 7880 already in use

If the app fails to start because port 7880 is occupied (from a previous session), run:

```
# macOS/Linux:
lsof -ti:7880 | xargs kill

# Windows (PowerShell):
Stop-Process -Id (Get-NetTCPConnection -LocalPort 7880).OwningProcess
```

### 17.7.6 Package installation fails on Linux

Missing system libraries. Install the development headers listed in the Linux section above, then retry `install.packages()`.

## References

- Wood, S. N. (2003). “Thin-plate regression splines”. In: *Journal of the Royal Statistical Society (B)* 65.1, pp. 95–114. DOI: [10.1111/1467-9868.00374](https://doi.org/10.1111/1467-9868.00374).
- (2004). “Stable and efficient multiple smoothing parameter estimation for generalized additive models”. In: *Journal of the American Statistical Association* 99.467, pp. 673–686. DOI: [10.1198/016214504000000980](https://doi.org/10.1198/016214504000000980).
- (2011). “Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models”. In: *Journal of the Royal Statistical Society (B)* 73.1, pp. 3–36. DOI: [10.1111/j.1467-9868.2010.00749.x](https://doi.org/10.1111/j.1467-9868.2010.00749.x).

- Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R*. 2nd ed. Chapman and Hall/CRC.
- Wood, S. N., N. Pya, and B. Säfken (2016). “Smoothing parameter and model selection for general smooth models (with discussion)”. In: *Journal of the American Statistical Association* 111, pp. 1548–1575. DOI: [10.1080/01621459.2016.1180986](https://doi.org/10.1080/01621459.2016.1180986).

# Foundations

## Overview

*Foundations • The Valuation Engineer*

Bert Craytor, Editor • July 9, 2026

Version 1.0.0

DOI: [10.67330/82vssv81](https://doi.org/10.67330/82vssv81)

*Foundations* publishes precise, citable treatments of concepts that underlie modern valuation methodology. Entries take terms working appraisers encounter — sometimes by other names, sometimes without recognizing them — and place them in their formal context: where they come from, what they mean technically, where they appear in practice, and why they matter for defensibility.

The inaugural issue establishes the six core concepts of the hedonic framework, arranged so that each entry builds on the vocabulary of the previous ones. The sequence runs from *Heterogeneous Good* (the property of real estate that makes valuation methodology necessary at all), through *Characteristics Space* and the *Hedonic Price Function* (the formal apparatus that organizes the bundle of attributes a property represents), through *Implicit Price* and *Hedonic Regression* (how the marginal contribution of each attribute is recovered from market evidence), and finally to *Latent Variable* (what the unrecovered residual signal in any hedonic model actually represents).

A single running example — eight comparables from a Pacifica, California neighborhood — runs through all six entries and is the issue’s pedagogical thread.

## Why these six, in this order

The choice of the six entries is not arbitrary. Each concept is load-bearing for modern hedonic methodology, each is genuinely encountered in everyday appraisal practice, and each is misunderstood or unnamed often enough that putting it in a citable form has practical value.

The ordering is dictated by conceptual dependency. *Heterogeneous good* is the root: it names the property of real estate (and used cars, and labor markets, and a few other goods) that makes the entire hedonic framework necessary. *Characteristics space* is the formal apparatus for representing heterogeneous goods as vectors; nothing further can be said precisely without it. The *hedonic price function* is the equilibrium object that maps points in characteristics space to prices. *Implicit prices* are its partial derivatives — the marginal contributions that every adjustment on a sales-comparison grid can be read as asserting, whether or not the appraiser names them as such. *Hedonic regression* is the empirical estimator that recovers implicit prices from observed transactions; this is the entry where theory becomes data analysis. *Latent variables* pick up where the regression’s residuals begin: the unobserved attributes whose effects survive in the residual as structured signal rather than discardable noise.

A reader who proceeds in this order should come away with a continuous thread from “why does valuation methodology exist at all?” to “what should we make of the residuals in a fitted model?” — a journey that, in the author’s experience, most appraisers traverse only piecemeal over many years of practice.

## The running example

A single dataset of eight Pacifica, California residential comparables runs through all six entries:

| Comp | GLA (sf) | Lot (sf) | View | Cond.     | Sale price  |
|------|----------|----------|------|-----------|-------------|
| A    | 1,650    | 9,000    | yes  | good      | \$1,425,000 |
| B    | 1,650    | 7,200    | no   | good      | \$1,280,000 |
| C    | 1,850    | 7,500    | no   | good      | \$1,407,500 |
| D    | 1,450    | 7,000    | no   | good      | \$1,155,000 |
| E    | 1,700    | 8,000    | yes  | excellent | \$1,470,000 |
| F    | 1,550    | 7,100    | no   | average   | \$1,177,500 |
| G    | 1,700    | 10,500   | no   | good      | \$1,392,500 |
| H    | 1,600    | 7,300    | no   | good      | \$1,312,500 |

The example is deliberately small enough to be reasoned about by hand and rich enough to exercise each concept. Entry 001 examines the \$145,000 difference between Comps A and B as the joint contribution of two characteristics (view and lot) entangled in a single paired-sales observation. Entry 002 places all eight comps as points in a four-dimensional characteristics space and shows that naive distance metrics misrank them. Entry 003 proposes a functional form for the hedonic price function on this neighborhood. Entry 004 recovers the implicit prices analytically and decomposes the original A-vs-B difference into a \$127,039 contribution from view and lot, with \$18,000 unaccounted for. Entry 005 fits the model formally in R and reports the full `lm()` output, including standard errors and residuals. Entry 006 examines the one residual that the model cannot explain — a +\$46,455 deviation on Comp H — and connects it to the latent-variable framing that underlies the Residual Constraint Approach methodology developed elsewhere by the author.

The dataset is available in the issue’s repository at `shared/data/pacifica_comps.csv` for readers who wish to reproduce any of the calculations.

## How to read this issue

A reader new to hedonic methodology should read the six entries in order. The dependencies are real: each entry assumes the vocabulary established in the previous ones, and skipping ahead invites confusion that the entries themselves cannot anticipate.

A reader with prior exposure to hedonic regression — which includes most appraisers who have taken an advanced statistics course, worked with an AVM, or read the mass-appraisal literature — can use the issue as a refresher and a reference. The entries are written to be citable atomically; an article published elsewhere in the journal that references *implicit price* can link to entry 004 by DOI rather than re-deriving the concept.

A reader interested specifically in the Residual Constraint Approach methodology should read entry 006 first to understand the latent-variable framing, then circle back to entries 003-005 for the hedonic apparatus that RCA extends, then to entries 001-002 for the foundational vocabulary on which everything rests.

A reader interested in the question of *what makes an appraisal adjustment defensible* will find that the “Why it matters for defensibility” section of each entry, read in sequence, amounts to a short essay on the methodological standing of adjustment-based valuation. That essay is not assembled explicitly anywhere in the issue, but it is the implicit subject of every entry.

## Editorial posture

*Foundations* is not a textbook department. Each entry is short enough to read in one sitting, written for an audience already fluent in appraisal practice, and designed to give a precise formulation of a

concept rather than a comprehensive exposition. The goal is to provide a stable referent that other articles in the journal can cite.

Entries are reviewed editorially and may carry invited commentary from a subject-matter expert. They are published under a Creative Commons Attribution license and assigned DOIs through both Crossref (as the canonical citation) and Zenodo (as the versioned archive). Each entry may be sharpened over time — minor revisions update the Zenodo version, major revisions update both DOIs with an editorial note describing the change. The Crossref DOI always resolves to the most recent version; specific historical versions are addressable through Zenodo.

Future issues will extend the *Foundations* vocabulary into adjacent areas: spatial dependence, conditional expectation, endogeneity, omitted-variable bias, identification, and the formal apparatus that distinguishes proxy variables from latent variables. Suggestions for concepts that deserve treatment are welcome and may be sent to the editor.

---

*The six entries in this issue:*

1. **Heterogeneous Good** — why valuation methodology must exist at all.
2. **Characteristics Space** — the vector representation of bundle attributes.
3. **Hedonic Price Function** — the equilibrium mapping from characteristics to prices.
4. **Implicit Price** — the marginal market price of each characteristic.
5. **Hedonic Regression** — estimating the function from observed transactions, with a full R worked example.
6. **Latent Variable** — recovering structured signal from residuals.

#### *References.*

- Court, A. T. (1939). Hedonic Price Indexes with Automotive Examples. In *The Dynamics of Automobile Demand*. General Motors Corporation.
- Lancaster, K. J. (1966). A New Approach to Consumer Theory. *Journal of Political Economy*, 74(2), 132–157.
- Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, 82(1), 34–55.
- Craytor, B. (2025). Residual Constraint Approach: Framework & Protocol. Zenodo. [doi:10.5281/zenodo.14787917](https://doi.org/10.5281/zenodo.14787917)

---

## Addendum: Prolog Formalization

The following machine-readable companion (000-overview.pl) formalizes this entry in the journal's Prolog knowledge base. It is also published as a browsable galley alongside this article.

```
%% 000-overview.pl
%%
%% Prolog companion to Foundations Vol. 1, Issue 1, Overview
%% Title: Overview
%% Author: Bert Craytor, Editor
%% Piece ID: foundations_overview
%% Kind: editorial_piece (not a concept; sits outside the dep graph)
%%
%% The Overview is an editorial piece, not a defined concept. Its .pl
%% companion records the issue's structure (which entries appear in
```

```

%% what reading order), the running example, and the reader pathways
%% the editor proposes for readers with different backgrounds.

% =====
% DICTIONARY IMPORT
% =====

:- use_module('.././.././../tools/dictionary/dictionary').
:- dictionary_release('dictionary-2026.1').

% --- From the dictionary (loaded via :- use_module above) -----
% term(foundations_overview, "Overview", foundations_editorial,
%      "Editorial overview of a Foundations issue. Frames the concepts
%      covered, situates them within appraisal practice, and is not
%      itself a defined concept in the dependency graph.",
%      published, 'dictionary-2026.1').
%
% Plus the six concept terms covered by this issue (heterogeneous_good,
% characteristics_space, hedonic_price_function, implicit_price,
% hedonic_regression, latent_variable).
% -----

:- use_terms([ foundations_overview,
                heterogeneous_good, characteristics_space,
                hedonic_price_function, implicit_price,
                hedonic_regression, latent_variable,
                residual_constraint_approach ]).
:- use_relations([depends_on/2]).

% =====
% PIECE METADATA
% =====

article_piece(foundations_overview).
article_kind(overview).
article_issue(volume(1), number(1), year(2026)).
article_id('foundations.2026.overview').

% =====
% ISSUE STRUCTURE: the six concept entries in dependency-ordered sequence
% =====
%
% The published reading order. Each entry assumes vocabulary established
% in the previous ones. This is the same as queries:reading_order/1
% derived from the depends_on graph; recording it here lets readers and
% downstream tools query the issue's structure without invoking the
% topological-sort machinery.

issue_entry(1, heterogeneous_good).
issue_entry(2, characteristics_space).
issue_entry(3, hedonic_price_function).
issue_entry(4, implicit_price).
issue_entry(5, hedonic_regression).
issue_entry(6, latent_variable).

% Each entry's one-line theme as stated in the Overview's closing list.
entry_theme(heterogeneous_good,
            "Why valuation methodology must exist at all.").
entry_theme(characteristics_space,

```

```

    "The vector representation of bundle attributes.").
entry_theme(hedonic_price_function,
    "The equilibrium mapping from characteristics to prices.").
entry_theme(implicit_price,
    "The marginal market price of each characteristic.").
entry_theme(hedonic_regression,
    "Estimating the function from observed transactions, with a full R worked example.").
entry_theme(latent_variable,
    "Recovering structured signal from residuals.").

% =====
% THE RUNNING EXAMPLE
% =====
%
% A single dataset of 8 Pacifica residential comparables runs through
% all six entries. The data lives in kb/pacifica_comps.pl and is
% consulted by each entry that uses it. The Overview's role here is
% only to point at it.

running_example_dataset(pacifica_comps).
running_example_n_comps(8).
running_example_location(pacifica_california).

% Which entries exercise the dataset.
:- consult(' ../kb/pacifica_comps').

% =====
% READER PATHWAYS
% =====
%
% The editor proposes four pathways through the issue, calibrated to
% the reader's background. These are advisory orderings; the canonical
% reading order is issue_entry/2 above.

reader_pathway(new_to_hedonic_methodology,
    "Read entries 1-6 in order. Dependencies are real and skipping invites confusion.",
    [heterogeneous_good, characteristics_space, hedonic_price_function,
     implicit_price, hedonic_regression, latent_variable]).

reader_pathway(prior_exposure_to_hedonic_regression,
    "Use as a refresher and atomic reference. Read in any order.",
    [hedonic_regression, hedonic_price_function, implicit_price,
     characteristics_space, heterogeneous_good, latent_variable]).

reader_pathway(interested_in_rca_methodology,
    "Read 006 first for the framing, then 003-005, then 001-002.",
    [latent_variable, hedonic_price_function, implicit_price,
     hedonic_regression, heterogeneous_good, characteristics_space]).

reader_pathway(interested_in_adjustment_defensibility,
    "Read the 'Why it matters for defensibility' sections in entry order. Forms an implicit short
    ↪ essay.",
    [heterogeneous_good, characteristics_space, hedonic_price_function,
     implicit_price, hedonic_regression, latent_variable]).

% =====
% EDITORIAL POSTURE
% =====

```

```

editorial_posture(
  "Foundations is not a textbook department. Each entry is short, written for an audience already
  ↪ fluent in appraisal practice, and designed to give a precise formulation rather than a
  ↪ comprehensive exposition.").

publication_model(
  [ license(cc_by),
    crossref_doi(canonical_citation),
    zenodo_doi(versioned_archive),
    revision_policy("Minor revisions bump Zenodo version; major revisions bump both DOIs with
    ↪ editorial note.") ] ).

% =====
% FUTURE DIRECTIONS
% =====
%
% Concepts the editor flags as deserving treatment in future Foundations
% issues. Recorded here so the journal's planning is machine-queryable.

future_topic(spatial_dependence).
future_topic(conditional_expectation).
future_topic(endogeneity).
future_topic(omitted_variable_bias_extended).
future_topic(identification).
future_topic(proxy_vs_latent_variable).

```

## Heterogeneous Good

*Foundations • The Valuation Engineer*

Bert Craytor • July 9, 2026

Version 1.0.0

DOI: [10.67330/slscsv65](https://doi.org/10.67330/slscsv65)

**Formal definition.** Heterogeneous goods are goods that are not interchangeable because each is a distinct bundle of characteristics that buyers value separately, and whose characteristics are not separately traded.

**Intuitive framing.** A homogeneous good is one where any item will do: a barrel of WTI crude, a bushel of No. 2 yellow corn, a kilowatt-hour from the grid, a share of a given stock. The grading standards or specifications behind these goods exist precisely to make individual items interchangeable, so that prices apply to the category rather than to the particular item. A heterogeneous good is one where the specific item matters — *this* house, not *that* house; *this* used car, not the one next to it.

Two features make a good heterogeneous in the technical sense. First, buyers care about multiple distinguishable attributes that vary across items in the market. Second, those attributes are bundled by the production technology of the good; they are not separately purchasable. You cannot buy 200 square feet of gross living area on its own. You cannot acquire a school district independently of a parcel of land. The characteristics arrive bundled, and the market sets a price on whole bundles rather than on the characteristics themselves.

Real estate is the canonical heterogeneous good and exhibits the property unusually strongly. Locational characteristics are perfectly non-replicable: no two parcels can occupy the same location, so every property has at least one attribute (its coordinates) that distinguishes it from every other property in the market. Physical, legal, and environmental attributes layer additional dimensions of variation onto a vector that, taken as a whole, is unique to each property.

**Where appraisers encounter it.** Every adjustment in the sales comparison approach is an implicit acknowledgment that the subject and the comparables are heterogeneous goods. If properties were homogeneous, sale prices would already be directly comparable; no grid would be needed. The grid is an attempt to reconcile the prices of distinct bundles by adjusting for differences in the bundle composition. It is worth noting what the grid does *not* do: it never decomposes a sale price into feature value contributions. The traditional grid has no column for value contributions — only adjustments, estimated directly, feature by feature.

The cost approach handles heterogeneity by reconstructing the bundle from its components — foundation, framing, finishes, depreciation — and pricing each separately. The income approach handles it by abstracting the bundle into a single cash-flow profile, with capitalization and yield rates absorbing bundle-level heterogeneity into a market-derived multiplier. All three valuation approaches are methodological responses to the underlying fact that real estate is a heterogeneous good. They differ in how they navigate that fact, not in whether they confront it.

Appraisers also encounter heterogeneity informally whenever they reason about comp selection, paired-sales analysis, or “apples-to-apples” comparisons. The instinct to find comparables that are similar in as many attributes as possible is a practitioner’s response to heterogeneity: minimize the bundle distance between subject and comp, and the unobserved component of the bundle contributes less to the difference in sale prices.

**Why it matters for defensibility.** The heterogeneity of real estate is the reason valuation methodology must exist as a discipline. If properties were interchangeable, market value would be directly observable from posted prices; valuation would reduce to lookup. Because they are not, value must be inferred from the prices of related-but-not-identical properties, and every inference step is a methodological choice that has to be justified.

Acknowledging heterogeneity formally clarifies several practice questions that USPAP and Standards Rule defensibility hinge on. Comp selection is the choice of items in characteristics space close enough to the subject that the implicit prices recovered from them generalize. The reasonableness of an adjustment depends on whether the underlying implicit price is supportable from market evidence. The reconciliation between approaches reflects the appraiser’s judgment about which approach navigates the heterogeneity of *this* property most credibly. Each of these defensibility questions is sharper once “heterogeneous good” is named as the underlying condition.

**Worked appraisal example.** Two recent sales in the same Pacifica, California neighborhood:

| Attribute        | Comp A      | Comp B      |
|------------------|-------------|-------------|
| GLA (sf)         | 1,650       | 1,650       |
| Bedrooms / Baths | 3 / 2       | 3 / 2       |
| Lot (sf)         | 9,000       | 7,200       |
| View             | Yes         | No          |
| Sale price       | \$1,425,000 | \$1,280,000 |

Under a homogeneous-goods framing, these are the same property: identical GLA, identical room count, same neighborhood. Under the heterogeneous-goods framing, they are distinct bundles, differing in two characteristics (lot and view) that buyers value separately. The \$145,000 price difference is the *joint* contribution of those two attributes — two implicit prices entangled in a single pair.

Recovering the implicit prices individually requires more pairs, ideally varying one attribute at a time. Three or four additional sales spanning the view/no-view and large-lot/small-lot combinations begin to identify the implicit prices separately. With enough variation across enough comps, hedonic regression performs the disentanglement formally; with limited variation, the appraiser is constrained to bracketed paired-sales reasoning whose reliability depends explicitly on how close the bundles are to one another in characteristics space.

The example illustrates why heterogeneity is the root condition: even two “identical” houses are not identical, and the methodological apparatus of appraisal exists precisely to handle the distance between them.

### References.

- Lancaster, K. J. (1966). A New Approach to Consumer Theory. *Journal of Political Economy*, 74(2), 132–157.
- Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, 82(1), 34–55.

---

*Cross-references: characteristics space; hedonic price function; implicit price.*

---

### Addendum: Prolog Formalization

The following machine-readable companion (001-heterogeneous-good.pl) formalizes this entry in the journal's Prolog knowledge base. It is also published as a browsable galley alongside this article.

```
%% 001-heterogeneous-good.pl
%%
%% Prolog companion to Foundations Vol. 1, Issue 1, Article 001
%% Title: Heterogeneous Good
%% Author: Bert Craytor
%% Concept: heterogeneous_good
%% Version: 0.1.0-draft
%%
%% This file is the Prolog galley for the article: it encodes the
%% article's main claims, worked example, and inference rules in a form
%% that other Prolog code in the journal can consult, query, and verify
%% against. Read alongside the entry.tex / entry.pdf prose.

% =====
% DICTIONARY IMPORT
% =====
%
% The journal-wide dictionary is the source of truth for term definitions
% and relation schemas. The use_module below pulls those facts in at
% runtime; the comments that follow reproduce the relevant entries so a
% reader skimming this file in isolation can see what's been imported.

:- use_module('../tools/dictionary/dictionary').
:- dictionary_release('dictionary-2026.1').

% --- From the dictionary (loaded via :- use_module above) -----
% term(heterogeneous_good, "Heterogeneous Good", economic_theory,
%      "Heterogeneous goods are goods that are not interchangeable
%      because each is a distinct bundle of characteristics that buyers
%      value separately, and whose characteristics are not separately
%      traded.", published, 'dictionary-2026.1').
%
% term(characteristics_space, "Characteristics Space", economic_theory,
%      "The characteristics space of a class of heterogeneous goods is
%      the vector space whose coordinates index the attributes that
%      buyers value, and in which each good is represented by the
%      vector of its attribute levels.", published, 'dictionary-2026.1').
%
% term(implicit_price, "Implicit Price", economic_theory,
%      "The implicit price of a characteristic is the partial derivative
%      of the hedonic price function with respect to that characteristic,
%      evaluated at a point in the characteristics space; it represents
%      the marginal market price of an incremental unit of the
%      characteristic, holding all other characteristics fixed.",
%      published, 'dictionary-2026.1').
%
% term(hedonic_price_function, ...).
%
% relation(depends_on, 2, "depends on", [foundations_concept, foundations_concept],
%          foundations_methodology, "...", accepted, 'dictionary-2026.1').
% relation(related, 2, ...).
%
% These facts are NOT declared in this file; the use_module above is
% what makes them available at runtime.
% -----
```

```

:- use_terms([ % Foundations concepts (terms/foundations.pl)
               heterogeneous_good, characteristics_space, implicit_price,
               hedonic_price_function,
               % Practice terms (terms/valuation-practice.pl)
               real_estate, homogeneous_good,
               sales_comparison_approach, cost_approach, income_approach,
               comp_selection, paired_sales_analysis, adjustment,
               reconciliation, uspap, standards_rule ]).

:- use_relations([related/2, depends_on/2]).

% =====
% ARTICLE SUBJECT
% =====

article_concept(heterogeneous_good).
article_issue(volume(1), number(1), year(2026)).
article_id('foundations.2026.001').

% =====
% ARTICLE-LOCAL VOCABULARY
% =====
%
% Terms the article needs that are NOT yet in the dictionary, and that
% the editor decided are too narrow to promote. These are field names
% for the worked-example property records | not concepts in the
% journal's ontology.

local_term(gla_sqft).
local_term(bedrooms).
local_term(baths).
local_term(lot_sqft).
local_term(view).

% =====
% FORMAL DEFINITION (restated as Prolog rule)
% =====
%
% From the formal definition: a good is heterogeneous IF buyers care
% about multiple distinguishable attributes that vary across items in
% the market, AND those attributes are bundled by production rather
% than separately traded.

%! heterogeneous(?Good) is nondet.
%
% True when Good satisfies both bundling criteria. The two predicates
% buyers_value_multiple_attributes/1 and attributes_bundled_by_production/1
% must be asserted for specific goods elsewhere; here they characterize
% the formal definition itself.

heterogeneous(Good) :-
    buyers_value_multiple_attributes(Good),
    attributes_bundled_by_production(Good).

% =====
% INTUITIVE FRAMING | CONTRASTS WITH HOMOGENEOUS GOODS
% =====

% Canonical examples of homogeneous goods cited in the article.
canonical_homogeneous_example(wti_crude).

```

```

canonical_homogeneous_example(no_2_yellow_corn).
canonical_homogeneous_example(kilowatt_hour).
canonical_homogeneous_example(common_stock_share).

% Their distinguishing feature: grading or specification standards make
% individual items interchangeable.
interchangeable_via_grading(wti_crude).
interchangeable_via_grading(no_2_yellow_corn).
interchangeable_via_specification(kilowatt_hour).
interchangeable_via_specification(common_stock_share).

% Canonical heterogeneous goods.
canonical_heterogeneous_example(real_estate).
canonical_heterogeneous_example(used_car).
canonical_heterogeneous_example(unique_machinery).

% Why real estate exhibits heterogeneity unusually strongly:
% locational characteristics are perfectly non-replicable.
buyers_value_multiple_attributes(real_estate).
attributes_bundled_by_production(real_estate).
non_replicable_attribute(real_estate, location).

% =====
% APPRAISER ENCOUNTERS
% =====
%
% The three classical valuation approaches are all methodological
% responses to heterogeneity. They differ in HOW they navigate the
% bundle, not in WHETHER they confront it.

valuation_approach(sales_comparison_approach).
valuation_approach(cost_approach).
valuation_approach(income_approach).

responds_to_heterogeneity(sales_comparison_approach,
    "Reconciles the prices of distinct bundles by adjusting for composition differences; the grid
    ↪ records only adjustments, estimated directly, and never decomposes a price into feature value
    ↪ contributions.").
responds_to_heterogeneity(cost_approach,
    "Reconstructs the bundle from priced components: foundation, framing, finishes, depreciation.").
responds_to_heterogeneity(income_approach,
    "Abstracts the bundle into a single cash-flow profile; cap/yield rates absorb heterogeneity.").

% Practitioner-level encounters with heterogeneity.
practitioner_response_to_heterogeneity(comp_selection,
    "Minimize bundle distance between subject and comp in characteristics space.").
practitioner_response_to_heterogeneity(paired_sales_analysis,
    "Isolate an attribute's contribution by holding other attributes constant.").
practitioner_response_to_heterogeneity(adjustment,
    "Quantify the price impact of one differing attribute at a time.").

% =====
% DEFENSIBILITY
% =====
%
% Why naming "heterogeneous good" sharpens USPAP defensibility questions.

defensibility_consequence(comp_selection,

```

```

    "Comps must be close enough in characteristics space that recovered implicit prices generalize
    ↪.").
defensibility_consequence(adjustment,
    "Each adjustment must be supportable from market evidence of the implicit price.").
defensibility_consequence(reconciliation,
    "Choice among approaches reflects which best navigates THIS property's heterogeneity.").

% Root claim: valuation methodology exists as a discipline BECAUSE
% real estate is heterogeneous. If it were homogeneous, value would be
% directly observable from posted prices.
root_claim(
    "Valuation methodology exists as a discipline because real estate is heterogeneous; in the
    ↪homogeneous case, value would be directly observable.").

% =====
% WORKED EXAMPLE | PACIFICA COMPS
% =====
%
% Two recent sales in the same Pacifica, California neighborhood. The
% example demonstrates that even "identical" houses are not identical:
% the $145,000 price spread is the joint contribution of differences in
% lot size and view, two implicit prices entangled in a single pair.

% The 8-comp Pacifica dataset is shared across entries 001-006 and
% lives in kb/pacifica_comps.pl. Consulting it here brings property/1,
% in_market/2, attribute_value/3, and sale_price/2 into this article's
% namespace.
:- consult('../..kb/pacifica_comps').

% --- Derived facts from the example -----

% The attributes that differ between the two comps.
differing_attribute(comp_a, comp_b, lot_sqft).
differing_attribute(comp_a, comp_b, view).

% The price differential and what it's attributable to.
price_differential(comp_a, comp_b, 145000).
attributed_to(price_differential(comp_a, comp_b, 145000),
    [implicit_price_of(lot_sqft), implicit_price_of(view)]).

% --- Article's argument about identification -----

% A single pair cannot separately identify the two implicit prices |
% they are entangled in the single $145K observation.
joint_identification_only(price_differential(comp_a, comp_b, 145000),
    [lot_sqft, view]).

% Separate identification requires more pairs varying one attribute at
% a time. With sufficient variation, hedonic regression performs the
% disentanglement formally; with limited variation, the appraiser uses
% bracketed paired-sales reasoning, whose reliability depends on how
% close the bundles are in characteristics space.
identification_strategy(few_pairs,
    "Bracketed paired-sales reasoning. Reliability depends on bundle proximity in characteristics
    ↪space.").
identification_strategy(many_pairs_with_variation,
    "Hedonic regression (see hedonic_regression).").

% =====

```

```

% INFERENCE RULES
% =====

%! comparable(?A, ?B) is nondet.
%
% A and B are comparable when they are distinct properties in the same
% market. This is the operational definition of "comp" used implicitly
% in the article.

comparable(A, B) :-
    property(A),
    property(B),
    A \== B,
    in_market(A, M),
    in_market(B, M).

%! attribute_difference(?A, ?B, ?Attr, ?DA, ?DB) is nondet.
%
% A and B differ on Attr, with DA the value at A and DB the value at B.
% Use this to enumerate the dimensions on which two comps actually vary.

attribute_difference(A, B, Attr, DA, DB) :-
    comparable(A, B),
    attribute_value(A, Attr, DA),
    attribute_value(B, Attr, DB),
    DA \== DB.

%! identifiable_implicit_price(?Attr) is nondet.
%
% An attribute's implicit price is identifiable from the observed comps
% if there exists at least one pair of comparables differing ONLY in
% that attribute. With our two-comp example this fails for lot_sqft AND
% view (both differ in the same pair), which is exactly the article's
% point.

identifiable_implicit_price(Attr) :-
    comparable(A, B),
    findall(X, attribute_difference(A, B, X, _, _), Diffs),
    Diffs == [Attr].

% =====
% CROSS-REFERENCES
% =====
%
% Concepts the article cross-references (corresponding to the
% \foundationsref{} macros in entry.tex).

cross_reference(characteristics_space).
cross_reference(hedonic_price_function).
cross_reference(implicit_price).

```

## Characteristics Space

*Foundations • The Valuation Engineer*

Bert Craytor • July 9, 2026

Version 1.0.0

DOI: [10.67330/3yymh010](https://doi.org/10.67330/3yymh010)

**Formal definition.** The characteristics space of a class of heterogeneous goods is the vector space whose coordinates index the attributes that buyers value, and in which each good is represented by the vector of its attribute levels.

**Intuitive framing.** The previous entry established that a heterogeneous good is a distinct bundle of characteristics. The natural next question is: how should we describe and compare such bundles formally? The answer, due to Lancaster (1966), is to treat each good as a point in a vector space whose axes are the characteristics buyers care about.

In this representation, a Pacifica residential property is no longer “a house in Park Pacifica” but a vector with coordinates for gross living area, lot size, view, condition, school district, distance to the ocean, and so on. Two properties that share all of these attribute values would occupy the same point in characteristics space; properties that differ in any attribute occupy different points. The geometry of the space encodes how close or how different two properties are, in a way that captures all the dimensions on which buyers make distinctions.

The dimensionality of the space depends on what attributes are relevant to the market under study. For a typical residential appraisal, the relevant space might have a dozen dimensions: physical (GLA, lot, beds, baths, age, quality, condition), locational (school district, view, proximity to amenities or disamenities), and legal/economic (zoning, easements). For a commercial appraisal, the relevant dimensions shift toward cash-flow attributes (lease structure, tenant quality, occupancy), but the underlying geometric idea is the same.

What makes the characteristics space useful is that it converts a qualitative practitioner’s intuition (“these comps are similar to the subject”) into a quantitative claim (“these comps are close to the subject in characteristics space”). Once distance is defined, every sales comparison decision can be examined as a statement about location in the space.

**Where appraisers encounter it.** Appraisers work in characteristics space constantly, usually without naming it. Every line of a sales comparison grid is a coordinate axis; every comp is a row vector of coordinate values; every adjustment is an attempt to move along one axis while holding the others fixed.

The eight Pacifica comparables we will use throughout this issue illustrate the construction directly. Each comp is a point in a four-dimensional characteristics space spanned by GLA, lot size, view, and condition:

| Comp | GLA (sf) | Lot (sf) | View | Cond.     | Sale price  |
|------|----------|----------|------|-----------|-------------|
| A    | 1,650    | 9,000    | yes  | good      | \$1,425,000 |
| B    | 1,650    | 7,200    | no   | good      | \$1,280,000 |
| C    | 1,850    | 7,500    | no   | good      | \$1,407,500 |
| D    | 1,450    | 7,000    | no   | good      | \$1,155,000 |
| E    | 1,700    | 8,000    | yes  | excellent | \$1,470,000 |
| F    | 1,550    | 7,100    | no   | average   | \$1,177,500 |
| G    | 1,700    | 10,500   | no   | good      | \$1,392,500 |
| H    | 1,600    | 7,300    | no   | good      | \$1,312,500 |

Comps A and B from the previous entry are the first two rows; the remaining six expand the example in ways we will exploit through the rest of the issue.

Each comp is a point in  $\mathbb{R}^4$ :

- Comp A is at (1650, 9000, 1, 2).
- Comp B is at (1650, 7200, 0, 2).
- Comp G is at (1700, 10500, 0, 2).

The view and condition coordinates are encoded numerically (view as  $\{0, 1\}$ , condition as  $\{1, 2, 3\}$ ); the price is *not* a coordinate of the characteristics space but a function of position within it.

**Why it matters for defensibility.** The characteristics space framework sharpens what comp selection actually is. A comparable is not just “a similar property” but a point in the relevant region of characteristics space close to the subject. The closer a comp is to the subject in the space, the smaller the adjustments required and the more reliable the indicated value becomes.

This reframing exposes several practice questions that the informal grid obscures:

- *Which dimensions matter?* The choice of axes is a modeling decision. Including “view” as an axis only makes sense if the market prices view distinctly; including “Feng Shui orientation” only makes sense if a market segment prices it. Defensible appraisal requires defending the axes, not just the values along them.
- *What distance metric applies?* Two comps may differ by \$200,000 in price because they are far apart in characteristics space, but the distance computation requires a scaling for each axis. Untransformed Euclidean distance treats one square foot of GLA as equivalent to one square foot of lot, which the market does not.
- *Are the comps in a region where the price function is well-behaved?* If the subject sits in a sparsely-sampled corner of characteristics space, no comp lies close to it and the value indication is necessarily extrapolative.

**Worked appraisal example.** The eight Pacifica comps occupy a relatively compact region of the four-dimensional characteristics space. Consider the question of which comp is “closest” to a hypothetical subject property at (1,700, 7,500, 1, 2) — 1,700 sf, 7,500 sf lot, view, good condition.

If we naively use Euclidean distance with unscaled axes, the lot dimension dominates because its values (7,000–10,500) are numerically much larger than view (0–1) or condition (1–3). Distances to the subject:

| Comp | Distance (unscaled Euclidean) | Notes                                 |
|------|-------------------------------|---------------------------------------|
| A    | 1,501                         | view matches; lot dominates           |
| B    | 308                           | view differs; lot close               |
| C    | 230                           | view differs; closest by raw distance |
| D    | 559                           | view differs; GLA off                 |
| E    | 502                           | view matches; condition differs       |
| F    | 422                           | view and condition differ             |
| G    | 3,000                         | view differs; lot far away            |
| H    | 224                           | view differs; closest of all          |

By raw distance, Comp H is the nearest neighbor. But Comp H has no view and the subject does. The naive metric has misled us: the lot dimension's numerical scale crowded out the categorical view dimension, which actually matters more to the price.

A defensible distance metric would scale each axis by its implicit price — roughly, by how many dollars one unit of variation in that axis is worth. Once the axes are scaled by implicit price, view (worth ~\$100,000) is no longer dominated by lot (~\$25 per square foot). Comp A, which matches the subject on view, then emerges as the appropriate primary comparable.

This is precisely what the implicit-price machinery developed in the following entries provides: a principled scaling of the characteristics space that turns “closeness” from a geometric guess into a market-derived computation.

### References.

- Lancaster, K. J. (1966). A New Approach to Consumer Theory. *Journal of Political Economy*, 74(2), 132–157.
- Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, 82(1), 34–55.

---

*Cross-references: heterogeneous good; hedonic price function; implicit price.*

---

## Addendum: Prolog Formalization

The following machine-readable companion (002-characteristics-space.pl) formalizes this entry in the journal's Prolog knowledge base. It is also published as a browsable galley alongside this article.

```
%% 002-characteristics-space.pl
%%
%% Prolog companion to Foundations Vol. 1, Issue 1, Article 002
%% Title: Characteristics Space
%% Author: Bert Craytor
%% Concept: characteristics_space
%% Version: 0.1.0-draft

% =====
% DICTIONARY IMPORT
% =====

:- use_module('../tools/dictionary/dictionary').
:- dictionary_release('dictionary-2026.1').

% --- From the dictionary (loaded via :- use_module above) -----
```

```

% term(characteristics_space, "Characteristics Space", economic_theory,
%      "The characteristics space of a class of heterogeneous goods is
%      the vector space whose coordinates index the attributes that
%      buyers value, and in which each good is represented by the
%      vector of its attribute levels.", published, 'dictionary-2026.1').
%
% term(heterogeneous_good, ...).
% term(hedonic_price_function, ...).
% term(implicit_price, ...).
%
% generalized_by(characteristics_space, lancaster_1966).
% depends_on(characteristics_space, heterogeneous_good).
% -----

:- use_terms([ characteristics_space, heterogeneous_good,
                hedonic_price_function, implicit_price,
                comp_selection, sales_comparison_approach,
                real_estate ]).
:- use_relations([related/2, depends_on/2, generalized_by/2]).

% =====
% ARTICLE SUBJECT
% =====

article_concept(characteristics_space).
article_issue(volume(1), number(1), year(2026)).
article_id('foundations.2026.002').

% =====
% LOCAL VOCABULARY
% =====
%
% Narrow article-local field names + the dimensions of the worked
% example's characteristics space.

local_term(dimension).
local_term(coordinate).
local_term(euclidean_distance).
local_term(distance_metric).
local_term(nearest_neighbor).

% =====
% FORMAL ATTRIBUTION
% =====

% The construction is due to Lancaster (1966), as noted in citations.
attributed_construction(characteristics_space, lancaster_1966).

% =====
% THE ARTICLE'S CLAIM ABOUT REPRESENTATION
% =====
%
% A heterogeneous good becomes a point in a vector space. Each axis is
% a characteristic buyers value; each good is the vector of its
% attribute levels.

representation_of(heterogeneous_good, point_in_vector_space).

% Examples of dimensions relevant to residential appraisal.

```

```

residential_dimension(gla_sqft).
residential_dimension(lot_sqft).
residential_dimension(view).
residential_dimension(cond).
residential_dimension(bedrooms).
residential_dimension(baths).
residential_dimension(age).
residential_dimension(quality).
residential_dimension(school_district).
residential_dimension(distance_to_amenity).
residential_dimension(zoning).

% Examples of dimensions relevant to commercial appraisal | included
% to make the article's point that the dimensions shift with market.
commercial_dimension(lease_structure).
commercial_dimension(tenant_quality).
commercial_dimension(occupancy).
commercial_dimension(cash_flow_profile).

% =====
% APPRAISER ENCOUNTERS
% =====
%
% Mapping appraisal practice onto the characteristics-space framework.

practice_to_space_correspondence(
    sales_comparison_grid_line, coordinate_axis,
    "A grid line is an axis.").
practice_to_space_correspondence(
    comp_row, row_vector,
    "A comp is a vector of coordinate values.").
practice_to_space_correspondence(
    adjustment, movement_along_axis,
    "An adjustment moves along one axis holding others fixed.").

% =====
% WORKED EXAMPLE
% =====
%
% The 8 Pacifica comps are points in a 4D characteristics space spanned
% by GLA, lot_sqft, view, cond_numeric. Bring in the shared dataset.

:- consult('../..kb/pacifica_comps').

% Axes of the chosen space for this article's analysis.
axis(gla_sqft).
axis(lot_sqft).
axis(view).
axis(cond_numeric).

% Numeric encoding for non-numeric axes.
encode(view, no, 0).
encode(view, yes, 1).
encode(cond_numeric, 1, 1).
encode(cond_numeric, 2, 2).
encode(cond_numeric, 3, 3).

%! coord(?Property, ?Axis, ?Numeric) is nondet.
%
```

```

% Numeric coordinate of Property along Axis. Handles symbolic attribute
% values by encoding them.

coord(P, A, N) :-
    attribute_value(P, A, V),
    ( number(V) -> N = V ; encode(A, V, N) ).

%! point(?Property, -Vector) is det.
%
% Vector representation of Property in the 4D characteristics space.

point(P, [GLA, Lot, View, Cond]) :-
    coord(P, gla_sqft, GLA),
    coord(P, lot_sqft, Lot),
    coord(P, view, View),
    coord(P, cond_numeric, Cond).

% =====
% THE HYPOTHETICAL SUBJECT
% =====
%
% The article tests distance metrics against this hypothetical property.

hypothetical_subject(subject).
attribute_value(subject, gla_sqft, 1700).
attribute_value(subject, lot_sqft, 7500).
attribute_value(subject, view, yes).
attribute_value(subject, cond_numeric, 2).

% =====
% NAIVE EUCLIDEAN DISTANCE (the article's straw distance metric)
% =====
%
% Computes the unscaled Euclidean distance between Property and the
% hypothetical subject. Used to demonstrate that unscaled distances
% mislead: the lot dimension dominates because its numeric range is
% much larger than view (0..1) or cond (1..3).

naive_distance_to_subject(P, D) :-
    point(P, V_P),
    point(subject, V_S),
    sum_of_squares(V_P, V_S, S),
    D is sqrt(S).

sum_of_squares([], [], 0).
sum_of_squares([X|Xs], [Y|Ys], S) :-
    sum_of_squares(Xs, Ys, S0),
    D is X - Y,
    S is S0 + D * D.

% =====
% THE ARTICLE'S CENTRAL OBSERVATION
% =====
%
% Under naive Euclidean distance, Comp H is the nearest neighbor. But
% Comp H has no view and the subject does; view matters more to price
% than the naive metric admits. The fix (developed in entries 003-005)
% is to scale each axis by its implicit price.

```

```

claim(naive_metric_misleads,
    "Unscaled Euclidean distance crowds out small-range axes (like view) under large-range axes (
    ↪ like lot_sqft). A market-defensible distance scales each axis by its implicit price. ").

% =====
% DEFENSIBILITY QUESTIONS RAISED
% =====
%
% Three questions the characteristics-space framing makes explicit.

defensibility_question(which_dimensions,
    "The choice of axes is a modeling decision. Defending an appraisal requires defending the axes,
    ↪ not just the values on them. ").
defensibility_question(distance_metric,
    "Untransformed Euclidean distance treats one sqft of GLA as equivalent to one sqft of lot,
    ↪ which the market does not. Distance requires a scaling per axis. ").
defensibility_question(extrapolation_region,
    "If the subject sits in a sparsely-sampled corner of the space, no comp is truly close and the
    ↪ indication is extrapolative. ").

% =====
% INFERENCE RULES
% =====

%! same_point(?A, ?B) is nondet.
%
% Two properties occupy the same point in the characteristics space.
% (Doesn't occur in the running example, but the article notes that it
% would if all coordinates matched.)

same_point(A, B) :-
    point(A, V),
    point(B, V),
    A \== B.

%! nearest_neighbor_by_naive_distance(-P) is semidet.
%
% The comp closest to the subject under the naive Euclidean metric.
% Used to illustrate the article's point that the naive metric chooses
% Comp H, not Comp A.

nearest_neighbor_by_naive_distance(P) :-
    findall(D-Q,
        ( property(Q), Q \== subject, naive_distance_to_subject(Q, D) ),
        Pairs),
    keysort(Pairs, [_-P|_]).

% =====
% CROSS-REFERENCES
% =====

cross_reference(heterogeneous_good).
cross_reference(hedonic_price_function).
cross_reference(implicit_price).

```

## Hedonic Price Function

Foundations • The Valuation Engineer

Bert Craytor • July 9, 2026

Version 1.0.0

DOI: [10.67330/9x1a0489](https://doi.org/10.67330/9x1a0489)

**Formal definition.** The hedonic price function is the equilibrium mapping from points in the characteristics space of a heterogeneous good to transaction prices, formed as the envelope of buyer bid functions and seller offer functions in a competitive market.

**Intuitive framing.** We now have two ingredients: a heterogeneous good (a bundle of characteristics) and a characteristics space (a vector space in which each bundle is a point). The hedonic price function combines them. It is the function  $p(\mathbf{z})$  that assigns a price to each point  $\mathbf{z}$  in the characteristics space, telling us what the market pays for a bundle with that particular combination of attributes.

The key insight from Rosen (1974) is that  $p(\mathbf{z})$  is not chosen by any single buyer or seller. It emerges from market equilibrium. Each buyer has a *bid function* expressing the maximum they would pay for any bundle, given their preferences and income. Each seller has an *offer function* expressing the minimum they would accept, given their costs and alternatives. The hedonic price function is the envelope of these bids and offers — the boundary along which buyers and sellers actually transact.

For real estate,  $p(\mathbf{z})$  is the answer to the question: if you specified a complete bundle of characteristics, what price would the market produce for it? The answer is not arbitrary. It reflects the aggregate preferences of all buyers in the market and the aggregate costs and alternatives of all sellers, mediated by the equilibrium matching of buyers to bundles.

The hedonic price function is the central theoretical object of hedonic analysis. Implicit prices are partial derivatives of it. Hedonic regression is an empirical estimate of it. The defensibility of any sales-comparison adjustment ultimately reduces to a claim about this function.

**Where appraisers encounter it.** Appraisers reason about  $p(\mathbf{z})$  continually, again usually without naming it. Several practice activities are implicit manipulations of the hedonic price function:

**Adjustments in the sales comparison approach.** When an appraiser adjusts Comp B upward by \$100,000 to account for the fact that the subject has a view and Comp B does not, the appraiser is asserting a claim about  $p(\mathbf{z})$ : namely, that holding all other characteristics fixed, the function increases by \$100,000 as the view characteristic moves from 0 to 1. This is a statement about the local slope of  $p$  in the view dimension.

**Reconciliation between approaches.** The cost approach constructs a value indication from below, summing the component costs of a replicated bundle. The income approach derives value from cash flows, which in equilibrium reflect the price of bundles that produce similar cash flows. The sales comparison approach reads price directly from observed transactions of similar bundles. In a well-functioning market, all three approaches should converge on the same point on  $p(\mathbf{z})$ . When they diverge, the appraiser must judge which approach has measured the function most reliably for the subject's location in characteristics space.

**Market analysis and time adjustments.** The hedonic price function shifts over time as supply, demand, financing, and expectations change. A 6-percent annual appreciation rate is a claim about how  $p(\mathbf{z})$  has shifted as a whole; a finding that view-premium properties have

appreciated faster than no-view properties is a claim that the function has shifted unevenly across the characteristics space.

**Why it matters for defensibility.** Naming the hedonic price function explicitly clarifies several foundational defensibility questions:

**Does the function exist?** The Rosen framework assumes a competitive market with enough transactions that an equilibrium function is meaningful. In thin markets, in rapidly changing markets, or in submarkets with idiosyncratic buyer pools, the function may be poorly defined or unstable. An appraiser working in such a market should acknowledge that the inferential machinery is operating on a shaky theoretical foundation.

**Is the function the same everywhere?** Rosen’s framework yields one function per market segment, not one function globally. Pacifica and San Francisco are distinct markets with distinct hedonic functions; single-family homes and condominiums are distinct submarkets even within a single city. An appraiser conflating segments is implicitly assuming a single function spans both, which is usually wrong.

**What functional form does  $p$  take?** The function need not be linear, log-linear, or any other specific form. Rosen showed only that  $p$  is the envelope of bids and offers; he did not specify its shape. Empirically,  $p$  exhibits nonlinearities (the marginal value of an additional bedroom is often nonconstant in GLA), interactions (view premium depends on lot orientation), and thresholds (a kitchen below some quality bar may attract a discount disproportionate to its condition score). Defensible estimation requires either a functional form supportable from the market or a flexible estimator capable of recovering the shape from data.

**Worked appraisal example.** For the Pacifica neighborhood under study, we will work with a simple additive functional form that the next entry will exercise more fully:

$$p(\mathbf{z}) = \beta_0 + \beta_{\text{GLA}} z_{\text{GLA}} + \beta_{\text{lot}} z_{\text{lot}} + \beta_{\text{view}} z_{\text{view}} + \beta_{\text{cond}} z_{\text{cond}}.$$

This form embeds two strong assumptions: that the contribution of each characteristic is linear in its level, and that the characteristics do not interact. Both assumptions are testable and frequently violated; we adopt them here as a starting point.

For the eight comps in this issue’s running example, our analysis in the next entries will estimate values of  $\beta$  that yield a function something like:

| Parameter                        | Estimated value |
|----------------------------------|-----------------|
| $\beta_0$ (intercept)            | \$80,777        |
| $\beta_{\text{GLA}}$ (per sf)    | \$575           |
| $\beta_{\text{lot}}$ (per sf)    | \$22.70         |
| $\beta_{\text{view}}$ (yes/no)   | \$86,179        |
| $\beta_{\text{cond}}$ (per step) | \$49,824        |

Plugged into the additive form, this function predicts a price for any point in the four-dimensional characteristics space. Substituting Comp A’s coordinates yields a predicted price near its observed \$1,425,000; substituting the hypothetical subject at (1700, 7500, 1, 2) yields a value indication for that property.

The point of the entry is not these specific numbers but the conceptual move: *the function exists, in principle, before we estimate it*. The function is a property of the market. Our estimation is an attempt to recover it. The recovery may be partial, biased, or noisy, and the next three entries are about how to do that recovery responsibly.

*References.*

- Court, A. T. (1939). Hedonic Price Indexes with Automotive Examples. In *The Dynamics of Automobile Demand*. General Motors Corporation.
- Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, 82(1), 34–55.
- Triplet, J. E. (2006). *Handbook on Hedonic Indexes and Quality Adjustments in Price Indexes*. OECD Publishing.

---

*Cross-references: heterogeneous good; characteristics space; implicit price; hedonic regression.*

---

**Addendum: Prolog Formalization**

The following machine-readable companion (`003-hedonic-price-function.pl`) formalizes this entry in the journal's Prolog knowledge base. It is also published as a browsable galley alongside this article.

```
%% 003-hedonic-price-function.pl
%%
%% Prolog companion to Foundations Vol. 1, Issue 1, Article 003
%% Title: Hedonic Price Function
%% Author: Bert Craytor
%% Concept: hedonic_price_function
%% Version: 0.1.0-draft

% =====
% DICTIONARY IMPORT
% =====

:- use_module('../tools/dictionary/dictionary').
:- dictionary_release('dictionary-2026.1').

% --- From the dictionary (loaded via :- use_module above) -----
% term(hedonic_price_function, "Hedonic Price Function", economic_theory,
%      "The hedonic price function is the equilibrium mapping from
%      points in the characteristics space of a heterogeneous good to
%      transaction prices, formed as the envelope of buyer bid
%      functions and seller offer functions in a competitive market.",
%      published, 'dictionary-2026.1').
%
% term(bid_function, "Bid Function", economic_theory, ...).
% term(offer_function, "Offer Function", economic_theory, ...).
% term(sales_comparison_approach, ...).
% term(cost_approach, ...).
% term(income_approach, ...).
%
% formalized_by(hedonic_price_function, rosen_1974).
% depends_on(hedonic_price_function, heterogeneous_good).
% depends_on(hedonic_price_function, characteristics_space).
% -----

:- use_terms([ hedonic_price_function, heterogeneous_good,
               characteristics_space, implicit_price, hedonic_regression,
               bid_function, offer_function,
               sales_comparison_approach, cost_approach, income_approach,
               adjustment, reconciliation ]).
:- use_relations([related/2, depends_on/2, formalized_by/2]).
```

```

% =====
% ARTICLE SUBJECT
% =====

article_concept(hedonic_price_function).
article_issue(volume(1), number(1), year(2026)).
article_id('foundations.2026.003').

% =====
% LOCAL VOCABULARY
% =====

local_term(equilibrium).
local_term(envelope).
local_term(functional_form).
local_term(market_segment).
local_term(linearity).
local_term(separability).
local_term(time_adjustment).
local_term(coefficient).
local_term(intercept).

% =====
% FORMAL ATTRIBUTION
% =====

attributed_formalization(hedonic_price_function, rosen_1974).

% =====
% CONSTRUCTION: ENVELOPE OF BID AND OFFER FUNCTIONS
% =====
%
% The Rosen (1974) construction:  $p(z)$  is the envelope of buyer bid
% functions and seller offer functions.

primitive_of_construction(bid_function, buyer).
primitive_of_construction(offer_function, seller).

constructed_as_envelope_of(hedonic_price_function,
                           [bid_function, offer_function]).

emerges_from(hedonic_price_function, market_equilibrium).

% Property that flows from the construction: the function is not
% chosen by any single market participant.
property(hedonic_price_function, not_set_by_individual_participant).
property(hedonic_price_function, equilibrium_object).
property(hedonic_price_function, defined_per_market_segment).

% =====
% APPRAISER ENCOUNTERS | IMPLICIT MANIPULATIONS OF  $p(z)$ 
% =====

implicit_manipulation_of_hedonic(
    adjustment,
    sales_comparison_approach,
    "Asserts a local slope of  $p$  in the relevant characteristic dimension.").
implicit_manipulation_of_hedonic(

```

```

    reconciliation,
    multi_approach_valuation,
    "Three approaches should converge on the same point of p; divergence requires editorial
    ↪ judgment. ").
implicit_manipulation_of_hedonic(
    time_adjustment,
    market_analysis,
    "p shifts over time; appreciation rates are statements about that shift. ").

% =====
% DEFENSIBILITY QUESTIONS
% =====
%
% The article names three questions the hedonic-price-function framing
% sharpens.

defensibility_question(function_existence,
    "In thin or rapidly-changing markets, the equilibrium function may be poorly defined; the
    ↪ inferential machinery operates on a shaky foundation. ").
defensibility_question(segment_specificity,
    "One function per market segment, not one function globally. Conflating segments is usually
    ↪ wrong. ").
defensibility_question(functional_form,
    "Rosen showed only that p is an envelope, not its shape. Empirically nonlinear with
    ↪ interactions and thresholds; estimation requires defending the form. ").

% =====
% WORKED EXAMPLE: ADDITIVE LINEAR SPECIFICATION
% =====
%
% For the Pacifica neighborhood, the article adopts the additive linear
% specification:
%  $p(z) = \beta_0 + \beta_{\text{gla}} * z_{\text{gla}} + \beta_{\text{lot}} * z_{\text{lot}} + \beta_{\text{view}} * z_{\text{view}} + \beta_{\text{cond}} * z_{\text{cond}}$ 
%
% The next entry (004) takes the coefficients as implicit prices; entry
% 005 estimates them by hedonic regression.

:- consult('../..//kb/pacifica_comps').

specification_form(additive_linear).
specification_assumption(linearity_in_levels,
    "Contribution of each characteristic is linear in its level. ").
specification_assumption(no_interactions,
    "Characteristics do not interact in the price function. ").
specification_caveat(
    "Both assumptions are testable and frequently violated; adopted here as a starting point. ").

% Coefficient values reported in the article (estimated in entry 005,
% asserted here so this entry's worked example can be queried).
coefficient(intercept,      80777.07).
coefficient(beta_gla,       574.93).
coefficient(beta_lot,       22.70).
coefficient(beta_view,      86179.03).
coefficient(beta_cond,      49824.33).

%! predicted_price(?Property, -Predicted) is det.
%
% The fitted additive-linear hedonic price function applied to a
% property's characteristics. Symbolic view/cond are mapped to {0,1}

```

% and {1,2,3} respectively.

```
predicted_price(P, Predicted) :-
    coord(P, gla_sqft, GLA),
    coord(P, lot_sqft, Lot),
    coord(P, view, View),
    coord(P, cond_numeric, Cond),
    coefficient(intercept, B0),
    coefficient(beta_gla, Bg),
    coefficient(beta_lot, B1),
    coefficient(beta_view, Bv),
    coefficient(beta_cond, Bc),
    Predicted is B0 + Bg * GLA + B1 * Lot + Bv * View + Bc * Cond.
```

```
coord(P, A, N) :-
    attribute_value(P, A, V),
    ( number(V) -> N = V
    ; A == view, V == yes -> N = 1
    ; A == view, V == no  -> N = 0
    ; V = N
    ).
```

```
% =====
% THE CENTRAL CONCEPTUAL MOVE
% =====
%
% The function exists, in principle, before estimation. The function is
% a property of the market; estimation is recovery.
```

```
claim(function_precedes_estimation,
    "The hedonic price function is a property of the market and exists conceptually prior to any
    ↪ specific fitted model. Estimation is recovery, and recovery may be partial, biased, or noisy
    ↪ .").
```

```
% =====
% CROSS-REFERENCES
% =====
```

```
cross_reference(heterogeneous_good).
cross_reference(characteristics_space).
cross_reference(implicit_price).
cross_reference(hedonic_regression).
```

## Implicit Price

*Foundations • The Valuation Engineer*

Bert Craytor • July 9, 2026

Version 1.0.0

DOI: [10.67330/md6wke17](https://doi.org/10.67330/md6wke17)

**Formal definition.** The implicit price of a characteristic is the partial derivative of the hedonic price function with respect to that characteristic, evaluated at a point in the characteristics space; it represents the marginal market price of an incremental unit of the characteristic, holding all other characteristics fixed.

**Intuitive framing.** The hedonic price function  $p(\mathbf{z})$  assigns a price to each bundle. Most practical questions in appraisal, however, are not about the price of a complete bundle but about the price of a specific *component* of a bundle. How much is the view worth? How much does an extra hundred square feet add? How much does the difference between “good” and “excellent” condition contribute?

The implicit price answers these questions. The term originates with Rosen (1974), who showed that observed prices of differentiated products together with their characteristics “define a set of implicit or ‘hedonic’ prices”; Lancaster (1966) supplied the underlying view of goods as bundles of characteristics. For each characteristic  $z_i$ , the implicit price is the rate at which  $p(\mathbf{z})$  changes as  $z_i$  varies while all other characteristics are held fixed:

$$\pi_i(\mathbf{z}) = \frac{\partial p(\mathbf{z})}{\partial z_i}.$$

The implicit price has units of dollars per unit of the characteristic — dollars per square foot of GLA, dollars per square foot of lot, dollars per unit step of condition. For a binary characteristic like view, the implicit price is the full premium paid for moving from no-view (0) to view (1).

The word “implicit” captures the central fact: these prices are *not* posted. No one offers to sell “one square foot of GLA” for \$575. The price is implicit in the equilibrium bundle prices that *are* posted, and the hedonic apparatus exists to extract it.

Implicit prices are the appraisal-relevant output of the hedonic framework. The adjustment in a sales comparison grid is, in principle, an estimate of an implicit price multiplied by a characteristic difference. Every adjustment is a wager about the value of  $\pi_i$  at the relevant point in characteristics space.

**Where appraisers encounter it.** The sales-comparison adjustment grid is a table of implicit-price applications, even when appraisers do not name them that way. A grid entry that adjusts a comp by “+\$10,000 for 200 sf less GLA than subject” is asserting that the implicit price of GLA, at the relevant location in characteristics space, is \$50 per square foot. A grid entry that adjusts “\$100,000 for inferior view” is asserting an implicit price of \$100,000 for the binary view characteristic. The entire defensibility of the sales comparison approach rests on whether these implicit prices can be supported from the market. In practice, however, appraisers estimate adjustments directly — from matched pairs, from experience, or from peer practice — without ever computing the underlying value contributions; the traditional grid records only the adjustments themselves. The implicit-price reading is an interpretation of what those adjustments assert, not a description of how they are derived.

This naming also clarifies why paired-sales analysis is the foundational technique for adjustment derivation. Two comps that differ in exactly one characteristic give a direct estimate of that characteristic's implicit price: the price difference divided by the characteristic difference. The reliability of paired-sales analysis depends on whether the two comps are truly close enough in characteristics space that the implicit price recovered from them generalizes to the subject.

Three properties of implicit prices that matter for practice:

**They are local.** The implicit price of GLA at the 1,650-sf point is not necessarily the same as at the 2,500-sf point. The hedonic price function is generally nonlinear, so its partial derivatives vary with location in characteristics space. An adjustment derived from a 1,500-sf paired-sales analysis does not necessarily apply to a 3,000-sf subject.

**They are conditional.** The implicit price of view in a beachfront submarket is not the same as in an inland submarket. The partial derivative is taken holding other characteristics fixed, but the value of that derivative depends on *where* the other characteristics are held fixed.

**They are market-dependent.** Implicit prices are equilibrium prices, so they reflect the preferences of the buyer pool at the time of the transactions. A market shift can change implicit prices even if the characteristics themselves are unchanged. Time adjustments are implicitly admissions that yesterday's implicit prices may not be today's.

**Why it matters for defensibility.** The implicit-price framing exposes adjustment defensibility questions that the informal grid can obscure:

**Where in characteristics space are you?** An implicit price is defensible only if estimated near the subject's location in characteristics space. A view premium derived from inland comps and applied to a beachfront subject is extrapolating across a region of the characteristics space where the function's behavior is unknown.

**What is the local geometry?** If two characteristics interact — for instance, if view is worth more on a larger lot — then treating them as additive separately is a misspecification. The implicit price of view depends on lot size, and an adjustment that ignores the interaction underestimates the function's actual structure.

**How was the implicit price recovered?** A paired-sales analysis, a regression coefficient, an industry rule-of-thumb, and a “professional judgment” all point to the same quantity — the implicit price — but with very different epistemic standing. The appraiser's defensibility depends on saying which source produced the implicit price used in each adjustment.

**Worked appraisal example.** For the Pacifica running example with the additive specification adopted in the previous entry, the implicit prices are simply the coefficients of the linear function:

$$\pi_{\text{GLA}} = \beta_{\text{GLA}}, \quad \pi_{\text{lot}} = \beta_{\text{lot}}, \quad \pi_{\text{view}} = \beta_{\text{view}}, \quad \pi_{\text{cond}} = \beta_{\text{cond}}.$$

With the estimated coefficients from entry 003:

| Characteristic | Implicit price | Units      |
|----------------|----------------|------------|
| GLA            | \$575          | per sf     |
| Lot            | \$22.70        | per sf     |
| View           | \$86,179       | per yes/no |
| Condition      | \$49,824       | per step   |

These implicit prices imply a paired-sales analysis. Returning to the Comp A vs. Comp B difference from entry 001:

- Comp A has view ( $\Delta z_{\text{view}} = +1$ ); contribution +\$86,179.

- Comp A has a 1,800-sf larger lot ( $\Delta z_{\text{lot}} = +1,800$ ); contribution  $1,800 \times \$22.70 = +\$40,860$ .
- All other characteristics identical.
- Predicted price difference:  $\$86,179 + \$40,860 = \$127,039$ .

The observed price difference was \$145,000. The implicit-price decomposition accounts for about \$127,000 of it; the remaining \$18,000 reflects sampling variability in the small dataset (and, more generally, the kinds of unobserved factors that the latent variable entry will take up).

Two operational points are worth noting. First, the linearity assumption is doing real work: under a nonlinear  $p(\mathbf{z})$ , the same physical lot difference might be worth more on a small lot than on a large one, and the additive decomposition above would be biased. Second, the implicit price of \$22.70 per square foot of lot is the marginal value of an incremental square foot, not the total value of the lot. The total contribution of a 9,000-sf lot to the bundle price is  $9,000 \times \$22.70 = \$204,300$ , but adding one more square foot raises the bundle price by only \$22.70.

This distinction — marginal vs. total — is the source of considerable confusion in everyday appraisal language and is one of the most important conceptual gains from naming the implicit price explicitly.

### References.

- Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, 82(1), 34–55.
- Lancaster, K. J. (1966). A New Approach to Consumer Theory. *Journal of Political Economy*, 74(2), 132–157.
- Freeman, A. M., III (1979). Hedonic Prices, Property Values and Measuring Environmental Benefits: A Survey of the Issues. *Scandinavian Journal of Economics*, 81(2), 154–173.
- Taylor, L. O. (2003). The Hedonic Method. In *A Primer on Nonmarket Valuation* (pp. 331–393). Kluwer Academic Publishers.
- Palmquist, R. B. (2005). Property Value Models. In *Handbook of Environmental Economics* (Vol. 2, pp. 763–819). North-Holland.
- Triplett, J. E. (2006). *Handbook on Hedonic Indexes and Quality Adjustments in Price Indexes*. OECD Publishing.

---

*Cross-references: heterogeneous good; characteristics space; hedonic price function; hedonic regression.*

---

## Addendum: Prolog Formalization

The following machine-readable companion (004-implicit-price.pl) formalizes this entry in the journal's Prolog knowledge base. It is also published as a browsable galley alongside this article.

```
%% 004-implicit-price.pl
%%
%% Prolog companion to Foundations Vol. 1, Issue 1, Article 004
%% Title:    Implicit Price
%% Author:   Bert Craytor
%% Concept:  implicit_price
%% Version:  0.1.0-draft

% =====
% DICTIONARY IMPORT
% =====
```

```

:- use_module('.././.././../tools/dictionary/dictionary').
:- dictionary_release('dictionary-2026.1').

% --- From the dictionary (loaded via :- use_module above) -----
% term(implicit_price, "Implicit Price", economic_theory,
%      "The implicit price of a characteristic is the partial
%      derivative of the hedonic price function with respect to that
%      characteristic, evaluated at a point in the characteristics
%      space; it represents the marginal market price of an
%      incremental unit of the characteristic, holding all other
%      characteristics fixed.", published, 'dictionary-2026.1').
%
% term(hedonic_price_function, ...).
% term(paired_sales_analysis, ...).
% term(adjustment, ...).
%
% formalized_by(implicit_price, rosen_1974).
% depends_on(implicit_price, hedonic_price_function).
% -----

:- use_terms([ implicit_price, hedonic_price_function, characteristics_space,
               heterogeneous_good, hedonic_regression,
               sales_comparison_approach, paired_sales_analysis, adjustment,
               comp_selection ]).
:- use_relations([related/2, depends_on/2, formalized_by/2]).

% Provenance: the term originates with Rosen (1974).
formalized_by(implicit_price, rosen_1974).

% =====
% ARTICLE SUBJECT
% =====

article_concept(implicit_price).
article_issue(volume(1), number(1), year(2026)).
article_id('foundations.2026.004').

% =====
% LOCAL VOCABULARY
% =====

local_term(partial_derivative).
local_term(marginal).
local_term(total).
local_term(locality).
local_term(conditionality).
local_term(market_dependence).

% =====
% IMPLICIT PRICE AS PARTIAL DERIVATIVE
% =====
%
%  $\pi_i(z) = \partial p(z) / \partial z_i$ 

implicit_price_definition(
    "Implicit price of characteristic i at point z = partial derivative of p with respect to z_i,
    ↪ evaluated at z.").

```

```

unit_of_implicit_price(continuous, "dollars per unit of the characteristic").
unit_of_implicit_price(binary,      "the full premium between the two states").

% =====
% THREE PROPERTIES OF IMPLICIT PRICES (per the article)
% =====

property_of_implicit_prices(local,
    "The implicit price of GLA at 1,650 sf is not necessarily the same as at 2,500 sf. p is
    ↪ generally nonlinear; partial derivatives vary with location.").
property_of_implicit_prices(conditional,
    "Implicit price depends on where the OTHER characteristics are held fixed. View premium on a
    ↪ beachfront submarket differs from inland.").
property_of_implicit_prices(market_dependent,
    "Implicit prices are equilibrium prices reflecting the current buyer pool. Market shifts change
    ↪ implicit prices even when characteristics do not.").

% =====
% IMPLICIT-PRICE / ADJUSTMENT EQUIVALENCE
% =====
%
% Every adjustment in a sales-comparison grid can be READ as asserting an
% implicit price applied to a characteristic difference. This is an
% interpretation of what the adjustment asserts, not a description of how
% traditional adjustments are derived in practice.

adjustment_as_implicit_price_application(
    "Interpretation: adjustment magnitude = (implicit price of attribute at relevant point) * (
    ↪ characteristic difference between comp and subject) --- what the adjustment asserts, not how
    ↪ it is derived." ).

adjustment_derivation_in_practice(
    "Traditional adjustments are estimated directly --- from matched pairs, experience, or peer
    ↪ practice --- without computing the underlying value contributions; the traditional grid
    ↪ records only the adjustments themselves." ).

% =====
% LINEAR-FORM IMPLICIT PRICES (from entry 003's adopted specification)
% =====
%
% Under the additive linear specification, implicit prices ARE the
% regression coefficients. The article reproduces them:

:- consult('../..//kb/pacifica_comps').

% Implicit price per characteristic, as inherited from the additive
% form of p adopted in entry 003.
implicit_price_of(gla_sqft, 575.00, "per square foot").
implicit_price_of(lot_sqft, 22.70, "per square foot").
implicit_price_of(view, 86179.03, "per yes/no").
implicit_price_of(cond, 49824.33, "per step on the ordinal scale").

% =====
% PAIRED-SALES DECOMPOSITION: COMP A vs COMP B
% =====
%
% Returns to the article-001 paired comparison and decomposes it using
% implicit prices.
%

```

```

% Comp A and Comp B share GLA, bedrooms, baths, condition.
% They differ in: view (+1) and lot_sqft (+1800).
% Predicted differential = 86179 + 40860 = 127039.
% Observed differential = 145000.
% Unexplained          = 17961 (~$18,000).

paired_sales_decomposition(comp_a, comp_b,
    [ contribution(view,      86179),
      contribution(lot_sqft,  40860) ],
    127039,
    145000,
    17961).

% Derived from the data + coefficients (for the linter to type-check
% against the dataset rather than the hand-computed values).
contribution_for_pair(A, B, Attr, Contribution) :-
    coord(A, Attr, VA),
    coord(B, Attr, VB),
    Diff is VA - VB,
    Diff \= 0,
    implicit_price_of(Attr, Pi, _),
    Contribution is Diff * Pi.

% Tiny coord helper for symbolic view/cond values.
coord(P, A, N) :-
    attribute_value(P, A, V),
    ( number(V) -> N = V
    ; A == view, V == yes -> N = 1
    ; A == view, V == no  -> N = 0
    ; V = N
    ).

% =====
% MARGINAL vs TOTAL CONTRIBUTION
% =====
%
% The article emphasizes the distinction. The implicit price of lot is
% $22.70 per square foot (marginal); the TOTAL contribution of Comp A's
% 9,000-sf lot is 9,000 * 22.70 = $204,300.

distinction(marginal_vs_total,
    "Implicit price is MARGINAL ($22.70 per additional sqft of lot). Total contribution of a 9,000-
    ↪sf lot is 9,000 * $22.70 = $204,300. Confusing them is a common source of error in everyday
    ↪appraisal language.").

%! total_contribution(?Property, ?Attribute, -Total) is det.
%
% Total dollar contribution of a characteristic to a property's bundle
% price under the linear specification.

total_contribution(P, A, Total) :-
    coord(P, A, V),
    implicit_price_of(A, Pi, _),
    Total is V * Pi.

% =====
% DEFENSIBILITY QUESTIONS RAISED
% =====

```

```

defensibility_question(location_in_space,
    "An implicit price is defensible only when estimated near the subject's location in
    ↪ characteristics space; a view premium from inland comps applied to a beachfront subject
    ↪ extrapolates into unknown territory.").
defensibility_question(local_geometry,
    "If view interacts with lot size, treating them as additively separable misspecifies the
    ↪ function. The implicit price of view depends on lot.").
defensibility_question(epistemic_provenance,
    "Paired-sales analysis, regression coefficient, industry rule, and professional judgment all
    ↪ point to the same quantity but have different epistemic standing. Defensibility requires
    ↪ declaring the source.").

% =====
% CROSS-REFERENCES
% =====

cross_reference(heterogeneous_good).
cross_reference(characteristics_space).
cross_reference(hedonic_price_function).
cross_reference(hedonic_regression).

```

## Hedonic Regression

*Foundations • The Valuation Engineer*

Bert Craytor • July 9, 2026

Version 1.0.0

DOI: [10.67330/ams1t139](https://doi.org/10.67330/ams1t139)

**Formal definition.** Hedonic regression is the estimation of the hedonic price function from observed transactions in a heterogeneous-goods market, yielding empirical estimates of the implicit prices of the characteristics that enter the model.

**Intuitive framing.** The previous entries described the hedonic price function and its implicit prices as theoretical objects. The function exists in principle, but in practice we observe only a finite set of transactions: bundles that traded, and the prices at which they traded. Hedonic regression is the empirical bridge between the two.

The setup is straightforward. Each transaction supplies a  $(\mathbf{z}_i, p_i)$  pair: the characteristics vector of the transacted bundle and its sale price. The collection of these pairs is a sample from the (unobserved) hedonic price function. Regression fits a functional form to that sample, estimating parameters that index the function. Once the parameters are estimated, partial derivatives of the fitted function yield implicit-price estimates.

Three modeling choices have to be made before regression can run:

1. *Which characteristics enter the model?* The choice determines the dimensions of the estimated function. Omitting a relevant characteristic causes its effect to leak into the coefficients of correlated included characteristics (omitted-variable bias).
2. *What functional form?* Linear, log-linear, log-log, polynomial, basis-expanded, semiparametric, machine-learning. The choice constrains what shape the estimated function can take and which implicit-price patterns it can recover.
3. *What estimator?* OLS, generalized least squares, robust M-estimators, quantile regression, spatial econometric estimators, penalized regression. The choice determines what the estimator is optimizing and what assumptions support its inference.

Each choice is methodologically substantive and each carries defensibility consequences.

**Where appraisers encounter it.** Hedonic regression is the foundation of mass appraisal and the engine of most automated valuation models (AVMs). When a county assessor revalues 100,000 parcels annually, a hedonic regression on recent sales is doing the work. When Zillow or Redfin produces an automated estimate, a hedonic-style model (often a tree ensemble rather than OLS, but conceptually still a hedonic estimator) is producing it. When a lender's secondary-market AVM passes or fails a property for a refi, the same machinery is involved.

Single-property appraisal increasingly draws on hedonic regression as well, even when the final value indication comes from sales comparison. Implicit prices estimated from a regression on a neighborhood sample can support adjustment magnitudes far more defensibly than appraiser intuition or rule-of-thumb percentages. The regression coefficients become evidence, not opinion, for the adjustments applied in the grid.

**Why it matters for defensibility.** Regression-based estimates carry a defensibility profile distinct from paired-sales analysis. The advantages: a regression uses all available sales rather than

two at a time; it can isolate the effect of one characteristic while controlling for many others simultaneously; and it produces standard errors that quantify the uncertainty in each implicit price. The trade-offs: a regression imposes a functional form that may not match the true price function; it requires more data than a paired-sales analysis; and its results can be sensitive to the choice of sample and specification.

Several defensibility questions recur in regression-based appraisal:

**Is the sample appropriate?** The regression estimates the hedonic function as it operates in the sample's market segment and time period. A regression on inland sales applied to a beachfront subject is a defensibility failure regardless of how clean the statistics look.

**Is the specification appropriate?** A linear specification is a strong assumption. If the true function exhibits diminishing returns to GLA above some threshold, the linear fit will misestimate implicit prices in that range. Diagnostics, residual plots, and alternative specifications are part of the defensible workflow.

**What does the standard error mean?** A standard error captures sampling variability under the assumed specification. It does not capture model misspecification, omitted-variable bias, or sample selection. A small standard error on a wrong specification is more dangerous than a large standard error on a correct one.

**Does the result triangulate with other approaches?** A defensible regression-based implicit price should be consistent with what paired-sales analysis suggests, with what cost-approach component costs imply, and with what appraisers familiar with the market believe. A regression coefficient at odds with all three is an invitation to investigate, not to publish.

**Worked appraisal example.** We can now exercise the full hedonic regression machinery on the eight Pacifica comps. The dataset is available at `shared/data/pacifica_comps.csv`.

**R code:**

```
comps <- read.csv("shared/data/pacifica_comps.csv")
fit <- lm(price ~ gla + lot + view + cond, data = comps)
summary(fit)
```

**Output:**

```
Call:
lm(formula = price ~ gla + lot + view + cond, data = comps)

Residuals:
    Min       1Q   Median       3Q      Max
-17995  -8243  -5433   -1409   46455

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  80777.07   164509.35    0.491   0.6571
gla           574.93     109.76    5.238   0.0135 *
lot           22.70      10.38    2.187   0.1165
view        86179.03    31342.97    2.750   0.0708 .
cond        49824.33    27761.27    1.795   0.1706
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 30481 on 3 degrees of freedom
Multiple R-squared:  0.9707, Adjusted R-squared:  0.9317
F-statistic: 24.86 on 4 and 3 DF, p-value: 0.01231
```

**Interpretation.** The fitted model estimates the hedonic price function as:

$$\hat{p}(\mathbf{z}) = 80,777 + 575 z_{\text{GLA}} + 22.70 z_{\text{lot}} + 86,179 z_{\text{view}} + 49,824 z_{\text{cond}}.$$

Each coefficient is the estimated implicit price of the corresponding characteristic. GLA contributes \$575 per square foot (with the narrowest standard error and the strongest significance); view contributes about \$86,000 to a Pacifica property; an additional unit of condition (e.g., moving from “good” to “excellent”) is worth roughly \$50,000.

**Several caveats are immediately visible:**

*The sample is tiny.* Eight observations and four predictors leaves three residual degrees of freedom. The standard errors are correspondingly wide, and the  $p$ -values for lot, view, and condition fail to reach the conventional 0.05 threshold despite their economic plausibility. This is a small-sample artifact, not evidence that these characteristics do not matter. With 80 comps rather than 8, the  $p$ -values would shrink dramatically.

*The overall fit is strong.* Adjusted  $R^2 = 0.93$  and a model  $p$ -value of 0.012 indicate that the included characteristics jointly explain most of the price variation in the sample. The combination of strong joint fit with weak individual significance is a textbook signal of multicollinearity, which here is structural: GLA, lot, and condition covary in the sample.

*One residual stands out.* The maximum residual is +\$46,455, corresponding to Comp H. The other seven residuals all fall within a  $\pm\$18,000$  band. Something about Comp H is not captured by the fitted function — the model is systematically under-predicting its price by an amount nearly three times the next largest residual.

This outlier residual is not noise. It is the signal that an unobserved characteristic of Comp H is contributing to its price in a way the included variables cannot recover. The latent variable entry takes up exactly this question.

**Validation against paired-sales reasoning.** The paired-sales analysis from entry 004 predicted a \$127,039 difference between Comps A and B from view and lot alone. The regression’s implicit prices are identical (since they are the source of that calculation). The regression generalizes the paired-sales logic to all eight comps simultaneously, but it produces nothing fundamentally new at this sample size; it confirms the paired analysis while reporting standard errors that paired-sales reasoning cannot.

*References.*

- Court, A. T. (1939). Hedonic Price Indexes with Automotive Examples. In *The Dynamics of Automobile Demand*. General Motors Corporation.
- Griliches, Z. (1961). Hedonic Price Indexes for Automobiles: An Econometric Analysis of Quality Change. In *The Price Statistics of the Federal Government*. National Bureau of Economic Research.
- Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, 82(1), 34–55.
- Triplett, J. E. (2006). *Handbook on Hedonic Indexes and Quality Adjustments in Price Indexes*. OECD Publishing.

---

*Cross-references: heterogeneous good; characteristics space; hedonic price function; implicit price; latent variable.*

---

## Addendum: Prolog Formalization

The following machine-readable companion (005-hedonic-regression.pl) formalizes this entry in the journal’s Prolog knowledge base. It is also published as a browsable galley alongside this article.

```

%% 005-hedonic-regression.pl
%%
%% Prolog companion to Foundations Vol. 1, Issue 1, Article 005
%% Title: Hedonic Regression
%% Author: Bert Craytor
%% Concept: hedonic_regression
%% Version: 0.1.0-draft

% =====
% DICTIONARY IMPORT
% =====

:- use_module('.././.././../tools/dictionary/dictionary').
:- dictionary_release('dictionary-2026.1').

% --- From the dictionary (loaded via :- use_module above) -----
% term(hedonic_regression, "Hedonic Regression", econometric_method,
%      "Hedonic regression is the estimation of the hedonic price
%      function from observed transactions in a heterogeneous-goods
%      market, yielding empirical estimates of the implicit prices of
%      the characteristics that enter the model.",
%      published, 'dictionary-2026.1').
%
% term(mass_appraisal, ...).
% term(automated_valuation_model, ...). % synonym: avm
% term(omitted_variable_bias, ...).
% term(multicollinearity, ...).
% term(residual, ...).
% term(ordinary_least_squares, ...).
%
% introduced_by(hedonic_regression, court_1939).
% introduced_by(hedonic_regression, griliches_1961).
% formalized_by(hedonic_regression, rosen_1974).
% depends_on(hedonic_regression, hedonic_price_function).
% depends_on(hedonic_regression, implicit_price).
% -----

:- use_terms([ hedonic_regression, hedonic_price_function, implicit_price,
               characteristics_space, heterogeneous_good, latent_variable,
               mass_appraisal, automated_valuation_model,
               omitted_variable_bias, multicollinearity, residual,
               ordinary_least_squares,
               sales_comparison_approach, paired_sales_analysis, adjustment ]).
:- use_relations([related/2, depends_on/2,
                 introduced_by/2, formalized_by/2]).

% =====
% ARTICLE SUBJECT
% =====

article_concept(hedonic_regression).
article_issue(volume(1), number(1), year(2026)).
article_id('foundations.2026.005').

% =====
% LOCAL VOCABULARY
% =====

```

```

local_term(specification).
local_term(coefficient).
local_term(standard_error).
local_term(p_value).
local_term(r_squared).
local_term(adjusted_r_squared).
local_term(residual_standard_error).
local_term(degrees_of_freedom).
local_term(f_statistic).
local_term(significance_level).
local_term(diagnostic).

% =====
% THE EMPIRICAL BRIDGE
% =====
%
% Hedonic regression is the bridge from theoretical  $p(z)$  to a sample of
% observed  $(z_i, p_i)$  pairs.

setup_of_regression(
    each_transaction_supplies_pair("(z_i, p_i): characteristics vector and sale price"),
    sample_from("the unobserved hedonic price function"),
    fits("a functional form to that sample, estimating parameters")).

% =====
% THREE MODELING CHOICES (per the article)
% =====

modeling_choice(which_characteristics,
    "Determines the dimensions of the estimated function. Omission causes omitted_variable_bias.").
modeling_choice(functional_form,
    "Constrains what shape the estimated function can take. Linear, log-linear, polynomial, basis-
    ↪ expanded, semiparametric, ML.").
modeling_choice(estimator,
    "OLS, GLS, robust, quantile, spatial, penalized. Determines optimization criterion and
    ↪ inference assumptions.").

% =====
% PRACTICE CONTEXTS WHERE REGRESSION IS THE ENGINE
% =====

regression_engine_context(mass_appraisal,
    "Assessor revaluation of 100,000 parcels annually.").
regression_engine_context(automated_valuation_model,
    "Zillow/Redfin/lender AVMs; tree ensembles are conceptually hedonic estimators.").
regression_engine_context(single_property_appraisal,
    "Regression coefficients support adjustments more defensibly than appraiser intuition.").

% =====
% DEFENSIBILITY PROFILE: REGRESSION vs PAIRED-SALES
% =====

advantage_over_paired_sales(uses_all_data,
    "Regression uses all available sales rather than two at a time.").
advantage_over_paired_sales(simultaneous_control,
    "Isolates one characteristic while controlling for many others.").
advantage_over_paired_sales(quantified_uncertainty,
    "Produces standard errors quantifying uncertainty in each implicit price.").

```

```

tradeoff(functional_form_constraint,
  "Regression imposes a form that may not match the true price function.").
tradeoff(data_requirement,
  "Requires more observations than paired-sales analysis.").
tradeoff(sample_specification_sensitivity,
  "Results can be sensitive to the choice of sample and specification.").

% =====
% RECURRING DEFENSIBILITY QUESTIONS
% =====

defensibility_question(sample_appropriateness,
  "The regression estimates the function for the sample's market segment and time period;
  ↪ mismatch is a defensibility failure regardless of statistical fit.").
defensibility_question(specification_appropriateness,
  "Linear is a strong assumption; diagnostics, residual plots, and alternative specifications
  ↪ belong in the defensible workflow.").
defensibility_question(standard_error_interpretation,
  "Standard errors capture sampling variability under the assumed specification, NOT
  ↪ misspecification, omitted-variable bias, or sample selection.").
defensibility_question(triangulation,
  "Regression coefficients should be consistent with paired-sales, cost components, and local
  ↪ appraiser opinion; isolated coefficients are an invitation to investigate.").

% =====
% WORKED EXAMPLE: lm(price ~ gla + lot + view + cond)
% =====

:- consult('../..kb/pacifica_comps').

% The fitted model.
fit_call("lm(formula = price ~ gla + lot + view + cond, data = comps)").
fit_estimator(ordinary_least_squares).
fit_software(r_lm).

% Fitted coefficients, with standard errors and significance from
% the R output reproduced in the article.
fitted_coefficient(intercept, 80777.07, 164509.35, 0.491, 0.6571, ' ').
fitted_coefficient(gla, 574.93, 109.76, 5.238, 0.0135, '*').
fitted_coefficient(lot, 22.70, 10.38, 2.187, 0.1165, ' ').
fitted_coefficient(view, 86179.03, 31342.97, 2.750, 0.0708, '.').
fitted_coefficient(cond, 49824.33, 27761.27, 1.795, 0.1706, ' ').

% Diagnostics from the R output.
residual_summary(min, -17995).
residual_summary(q1, -8243).
residual_summary(median, -5433).
residual_summary(q3, -1409).
residual_summary(max, 46455).

model_summary(residual_standard_error, 30481).
model_summary(degrees_of_freedom, 3).
model_summary(r_squared, 0.9707).
model_summary(adjusted_r_squared, 0.9317).
model_summary(f_statistic, 24.86).
model_summary(model_p_value, 0.01231).

% =====
% INTERPRETIVE CAVEATS FROM THE ARTICLE

```

```
% =====

caveat(tiny_sample,
    "n=8, four predictors, three residual df. Standard errors are correspondingly wide; lot/view/
    ↪ cond fail to reach the 0.05 threshold despite economic plausibility. Small-sample artifact.").
    ↪

caveat(structural_multicollinearity,
    "Adj R^2 = 0.93 with model p = 0.012 indicates strong joint fit; weak individual significance
    ↪ is a textbook multicollinearity signature. Here GLA, lot, and condition covary in the sample
    ↪ .").

caveat(comp_h_residual,
    "Comp H's residual is +$46,455; the other seven fall within +/- $18,000. The model
    ↪ systematically under-predicts Comp H by ~3x the next-largest residual. NOT noise. Signal of
    ↪ an unobserved characteristic.").

% =====
% RELATING TO PAIRED-SALES (entry 004)
% =====
%
% The regression's implicit prices ARE the values used in entry 004's
% paired-sales decomposition of A vs B. The regression generalizes that
% logic to all 8 comps simultaneously while reporting standard errors
% that paired-sales reasoning cannot produce.

confirms_paired_sales_from(entry_004,
    "A vs B differential of $127,039 from view and lot alone | coefficients are the same;
    ↪ regression confirms the paired analysis while adding standard errors.").

% =====
% RESIDUAL OF COMP H POINTS FORWARD TO ENTRY 006
% =====

points_forward_to(latent_variable,
    "Comp H's anomalous +$46,455 residual is the signal that an unobserved characteristic is
    ↪ contributing to price. Entry 006 takes up the question.").

% =====
% PREDICTED PRICES FROM THE FITTED MODEL
% =====
%
% Computes predicted price from the fitted coefficients. The residual
% for each comp is sale_price - predicted_price.

predicted_price(P, Predicted) :-
    coord(P, gla_sqft, GLA),
    coord(P, lot_sqft, Lot),
    coord(P, view, View),
    coord(P, cond_numeric, Cond),
    fitted_coefficient(intercept, B0, _, _, _, _),
    fitted_coefficient(gla, Bg, _, _, _, _),
    fitted_coefficient(lot, Bl, _, _, _, _),
    fitted_coefficient(view, Bv, _, _, _, _),
    fitted_coefficient(cond, Bc, _, _, _, _),
    Predicted is B0 + Bg * GLA + Bl * Lot + Bv * View + Bc * Cond.

residual_of(P, R) :-
    sale_price(P, Observed),
```

```
predicted_price(P, Predicted),
R is Observed - Predicted.

coord(P, A, N) :-
  attribute_value(P, A, V),
  ( number(V) -> N = V
  ; A == view, V == yes -> N = 1
  ; A == view, V == no  -> N = 0
  ; V = N
  ).

% =====
% CROSS-REFERENCES
% =====

cross_reference(heterogeneous_good).
cross_reference(characteristics_space).
cross_reference(hedonic_price_function).
cross_reference(implicit_price).
cross_reference(latent_variable).
```

## Latent Variable

Foundations • The Valuation Engineer

Bert Craytor • July 9, 2026

Version 1.0.0

DOI: [10.67330/11n3g168](https://doi.org/10.67330/11n3g168)

**Formal definition.** A latent variable is an unobserved attribute that influences observed outcomes, whose value must be inferred from its effects on observable variables rather than measured directly.

**Intuitive framing.** The first five entries treated the characteristics vector  $\mathbf{z}$  as if it could be fully measured: GLA, lot, view, condition, and so on, all available as columns of data. In practice, no characteristics vector is ever complete. Some attributes are not in the data because they are difficult to measure (interior finish quality, layout flow, micro-locational micro-orientation), because they are absent from the standard data feeds (MLS records do not capture lighting quality, soundproofing, or recent unpermitted improvements), or because they have not yet been recognized as relevant (neighborhood-level features that turn out to predict price only after analysis reveals them).

These unobserved-but-relevant attributes are *latent variables*. They exist, they influence price, but they are not in the columns of our regression. Their effects do not vanish because they are unmeasured; they accumulate in the model’s residuals, where they appear as “noise” from the perspective of the included variables but are in fact structured signal from the perspective of the underlying data-generating process.

Latent variables are the bridge between the idealized hedonic framework and the messy reality of appraisal data. Recognizing them explicitly is a precondition for honest reasoning about model uncertainty, residual structure, and the limits of what observable characteristics can explain.

**Where appraisers encounter it.** Appraisers encounter latent variables in three recurring contexts.

**The quality and condition adjustments.** “Quality” and “condition” as appraisal categories are themselves attempts to collapse a multi-dimensional latent construct into a single ordinal scale. A C3 condition rating in UAD 3.6 is meant to summarize a property’s deferred maintenance, finish wear, mechanical-systems age, and presentation — four to ten underlying attributes condensed into one number. The single-number summary is a proxy for a latent construct; the appraiser’s craft includes judgment about how well the proxy captures the underlying truth in each case.

**The unexplained residual in the grid.** When three adjusted comps indicate value but disagree by more than expected, the disagreement is informative. It is rarely all sampling noise; more often it reflects an attribute the appraiser has not yet identified or adjusted for. Investigating the residual disagreement — “why does this comp indicate \$50,000 above the others?” — is the appraiser’s informal version of latent-variable inference.

**The recent improvement that nobody can see.** A property may have undergone a kitchen remodel, a roof replacement, or a permitted ADU buildout that is not yet reflected in MLS or assessor data. From the appraiser’s perspective, the improvement is observable in person but not in the data. From the regression’s perspective, the improvement is latent. The appraiser must reconcile the in-person observation with the data-driven indication, and both should converge on the latent attribute’s true contribution.

**Why it matters for defensibility.** Latent variables introduce several defensibility considerations that the additive linear model in entry 005 does not address directly.

**Omitted-variable bias.** If a latent variable is correlated with an observed variable, its effect leaks into the observed variable’s coefficient. If recently sold Pacifica homes tend to have remodeled kitchens (a latent attribute) and the contract date of the home sale is in the data, then the age of the sale (aka `sale_age` or “days since sold”) coefficient will absorb some of the kitchen-remodel effect. The estimated implicit price contribution of `sale_age` becomes biased — it reflects `sale_age` *plus* a portion of the unmeasured kitchen-remodel status. Defensible appraisal acknowledges that observed-variable coefficients are conditional on what is missing from the model.

**Residual interpretation.** Standard regression theory treats residuals as Gaussian noise with zero mean. The latent-variable view treats residuals as a mixture of noise and signal: part is sampling variation, part is the cumulative effect of unobserved attributes. Identifying which is which is methodologically substantive. A residual that correlates with location, with sale date, with appraiser identity, or with any other observable suggests that the corresponding latent variable can be partially recovered.

**Model selection.** Adding more observed variables is the obvious remedy for omitted-variable bias, but the latent variables that matter most are often the ones that are hardest to add to the model (condition fine-grain, quality of recent improvements, neighborhood micro-effects). The defensible response is sometimes to recognize that the model will never capture them and to allocate the unexplained variance to the residual rather than to force-fit additional variables.

**Worked appraisal example.** The regression in entry 005 produced an unusual residual for Comp H: the model under-predicted Comp H’s price by \$46,455 while the other seven residuals fell within a  $\pm \$18,000$  band. The observed characteristics of Comp H — GLA 1,600, lot 7,300, no view, condition “good” — place it firmly in the middle of the sample; nothing about its observable profile predicts an unusually high price. And yet it sold for \$46,455 more than the model says it should have.

The natural hypothesis is that Comp H has a latent attribute that contributes to its price but does not appear in the data. In the construction of the dataset for this issue, this is exactly the case: Comp H underwent a recent kitchen remodel that adds approximately \$60,000 to its bundle value. The remodel was not in the regression’s variable list, so the contribution accumulated in the residual rather than in a coefficient. The estimated residual (\$46,455) is slightly smaller than the true latent contribution (\$60,000) because regression spreads small portions of the latent effect across the other coefficients, biasing them upward in correlated dimensions.

Three operational consequences follow:

**Treat the outlier residual as evidence, not error.** The first appropriate response to Comp H’s residual is not to drop it as an outlier but to investigate what about the property explains the under-prediction. In this case, an in-person inspection or a permit records search would reveal the remodel. The model’s failure has diagnostic value.

**Recognize that the other coefficients are slightly biased.** The view, condition, and lot coefficients have absorbed small fractions of the remodel effect via correlations with Comp H’s observed characteristics. The bias is small at  $n = 8$  and would shrink further with more comps and more covariates, but it is not zero.

**The residuals as a whole carry recoverable structure.** This is the substantive claim of the Residual Constraint Approach (RCA) methodology developed by the author (Craytor, 2025): in residential appraisal data, residuals are systematically not iid Gaussian noise. They reflect the structured contribution of unobserved attributes — condition fine-grain, quality fine-grain, functional utility, and other latent constructs — that can be partially recovered by treating the residual itself as a latent-variable signal to be modeled rather than discarded.

The RCA treatment of condition and quality makes this two-level structure operational. The ap-

praiser’s ordinal ratings — the coarse-grain proxies discussed above, which may be labeled condition-1 (condition-coarse-grain) and quality-1 (quality-coarse-grain) — are crude enough that they might as well be treated as latent variables themselves, but they can still be entered as regressors to obtain approximate implicit-price contributions. The protocol then moves those estimated contributions out of the measured attributes in the regression section of the sales grid and back into the residual, from which they are reassigned to the fine-grain latent variables condition-2 (condition-fine-grain) and quality-2 (quality-fine-grain) and refined against additional evidence such as MLS public remarks. The coarse-grain rating supplies an initial value assignment; the fine-grain latent variable carries the recoverable structure. In the end, the fine-grain value contribution replaces the coarse-grain value contribution.

The single-comp illustration here is the simplest case of the RCA intuition. With more data and richer modeling, the structure in the residual can be partitioned into recognizable latent factors, each contributing to the bundle’s market price in ways that the standard hedonic specification cannot reach. Latent variables are not a nuisance to be controlled away; they are the next frontier of defensible valuation modeling.

### References.

- Spearman, C. (1904). “General Intelligence,” Objectively Determined and Measured. *American Journal of Psychology*, 15(2), 201–292.
- Bollen, K. A. (2002). Latent Variables in Psychology and the Social Sciences. *Annual Review of Psychology*, 53, 605–634.
- Craytor, B. (2025). Residual Constraint Approach: Framework & Protocol. Zenodo. [doi:10.5281/zenodo.14787917](https://doi.org/10.5281/zenodo.14787917)

---

*Cross-references: characteristics space; hedonic price function; implicit price; hedonic regression.*

---

## Addendum: Prolog Formalization

The following machine-readable companion (006-latent-variable.pl) formalizes this entry in the journal’s Prolog knowledge base. It is also published as a browsable galley alongside this article.

```
%% 006-latent-variable.pl
%%
%% Prolog companion to Foundations Vol. 1, Issue 1, Article 006
%% Title: Latent Variable
%% Author: Bert Craytor
%% Concept: latent_variable
%% Version: 0.1.0-draft

% =====
% DICTIONARY IMPORT
% =====

:- use_module('../tools/dictionary/dictionary').
:- dictionary_release('dictionary-2026.1').

% --- From the dictionary (loaded via :- use_module above) -----
% term(latent_variable, "Latent Variable", measurement_theory,
%      "A latent variable is an unobserved attribute that influences
%      observed outcomes, whose value must be inferred from its
%      effects on observable variables rather than measured directly.",
%      published, 'dictionary-2026.1').
```

```

%
% term(residual, ...).
% term(omitted_variable_bias, ...).
% term(proxy_variable, ...).
% term(residual_constraint_approach, ...). % synonym: rca
%
% introduced_by(latent_variable, spearman_1904).
% formalized_by(latent_variable, bollen_2002).
% introduced_by(residual_constraint_approach, craytor_2025).
% depends_on(latent_variable, characteristics_space).
% -----

:- use_terms([ latent_variable, residual, omitted_variable_bias,
               proxy_variable, residual_constraint_approach,
               characteristics_space, hedonic_price_function,
               implicit_price, hedonic_regression ]).
:- use_relations([related/2, depends_on/2,
                  introduced_by/2, formalized_by/2]).

% =====
% ARTICLE SUBJECT
% =====

article_concept(latent_variable).
article_issue(volume(1), number(1), year(2026)).
article_id('foundations.2026.006').

% =====
% LOCAL VOCABULARY
% =====

local_term(unobserved).
local_term(noise).
local_term(signal).
local_term(structured_signal).
local_term(uad_3_6).
local_term(c3_condition_rating).
local_term(mls).
local_term(permit_records).
local_term(neighborhood_micro_effect).

% =====
% THE CENTRAL CLAIM
% =====

claim(latent_variables_accumulate_in_residuals,
      "Unobserved-but-relevant attributes do not vanish when omitted from the model; their effects
      ↪ accumulate in the residual, appearing as noise from the model's perspective but as structured
      ↪ signal from the data-generating process's perspective.").

claim(residuals_are_mixture,
      "Standard regression theory treats residuals as iid Gaussian noise; the latent-variable view
      ↪ treats them as a mixture of sampling noise AND signal from unobserved attributes.").

% =====
% THREE CONTEXTS WHERE APPRAISERS ENCOUNTER LATENT VARIABLES
% =====

appraiser_encounter(quality_and_condition_proxies,

```

```

    "UAD 3.6 'C3 condition' is a single ordinal scale collapsing a multi-dimensional construct (
    ↪ deferred maintenance, finish wear, mechanical-systems age, presentation). The rating is a
    ↪ proxy_variable for a latent construct.>").

appraiser_encounter(unexplained_grid_disagreement,
    "When three adjusted comps disagree by more than expected, the disagreement is rarely all
    ↪ sampling noise; usually reflects an attribute not yet identified or adjusted for.>").

appraiser_encounter(unobserved_recent_improvement,
    "A recent remodel, roof replacement, or permitted ADU buildout may be observable in person but
    ↪ not in MLS/assessor data | observable to the appraiser but latent to the regression.>").

% =====
% DEFENSIBILITY ISSUES
% =====

defensibility_issue(omitted_variable_bias,
    "If a latent variable is correlated with an observed variable, its effect leaks into the
    ↪ observed variable's coefficient. Observed-variable coefficients are conditional on what is
    ↪ missing from the model.>").

defensibility_issue(residual_interpretation,
    "Treat residuals as a mixture, not pure noise. A residual that correlates with location, sale
    ↪ date, appraiser identity, or any other observable suggests the latent variable can be
    ↪ partially recovered.>").

defensibility_issue(model_selection_limits,
    "The latent variables that matter most are often hardest to add (condition fine-grain, quality,
    ↪ micro-locational effects). Sometimes the defensible response is to allocate the unexplained
    ↪ variance to the residual rather than force-fit additional variables.>").

% =====
% WORKED EXAMPLE: COMP H'S ANOMALOUS RESIDUAL
% =====

:- consult('../..//kb/pacifica_comps').

% Comp H's regression residual from entry 005.
comp_h_residual_dollars(46455).

% The latent attribute revealed by investigation.
comp_h_latent_attribute(kitchen_remodel).
comp_h_latent_attribute_value_estimate(60000).

% Why the estimated residual is smaller than the latent attribute's
% true contribution: regression spreads small portions of the latent
% effect across correlated coefficients, biasing them upward.
residual_smaller_than_truth_explanation(
    "Regression spreads small portions of the latent effect across the other coefficients via
    ↪ correlations, biasing them slightly upward in correlated dimensions.>").

% =====
% THREE OPERATIONAL CONSEQUENCES
% =====

operational_consequence(outlier_residual_is_evidence,
    "The first response to Comp H's residual is to investigate, not to drop. An in-person
    ↪ inspection or permit search reveals the remodel; the model's failure has diagnostic value.>").

```

```

operational_consequence(other_coefficients_slightly_biased,
    "View, condition, and lot coefficients absorbed small fractions of the remodel effect via
    ↪ correlations with Comp H's observed characteristics. Bias is small at n=8 but not zero.").

operational_consequence(residuals_carry_recoverable_structure,
    "Residuals in residential appraisal data are systematically not iid Gaussian. They reflect
    ↪ structured contributions of unobserved attributes that can be partially recovered: this is
    ↪ the Residual Constraint Approach intuition.").

% =====
% CONNECTION TO RESIDUAL CONSTRAINT APPROACH (RCA)
% =====
%
% The single-comp illustration here is the simplest case of the RCA
% intuition. With richer modeling, residual structure is partitioned
% into recognizable latent factors.

framework_reference(residual_constraint_approach, craytor_2025).

rca_intuition(
    "Treat the residual itself as a latent-variable signal to be modeled rather than discarded.
    ↪ Latent variables are not a nuisance to be controlled away; they are the next frontier of
    ↪ defensible valuation modeling.").

rca_factor_examples(
    [condition_fine_grain, quality_fine_grain, functional_utility]).

% Two-level condition/quality structure (RCA protocol; mirrors the prose).
rating_levels(condition, condition_coarse_grain, condition_fine_grain).
rating_levels(quality, quality_coarse_grain, quality_fine_grain).

coarse_to_fine_grain_protocol(
    "Regress on the appraiser's coarse-grain C1-C6/Q1-Q6 ratings to obtain approximate value
    ↪ contributions; move those contributions out of the measured attributes back into the residual;
    ↪ reassign them to the fine-grain latent variables and refine against MLS public remarks. In
    ↪ the end, the fine-grain value contribution replaces the coarse-grain value contribution.").

% =====
% INFERENCE RULE: anomalous-residual detection
% =====
%
% A property's residual is "anomalous" if its magnitude exceeds the
% typical residual band reported in entry 005 (+/- $18,000 for the other
% seven comps). Useful for flagging properties whose hedonic prediction
% suggests a latent attribute warrants investigation.

typical_residual_band(18000).

anomalous_residual(P, R) :-
    comp_h_residual_dollars(R0), % stand-in: this entry references the
                                % residual computed in entry 005; future
                                % integration would pull from there.

    P == comp_h,
    R = R0.

% =====
% CROSS-REFERENCES
% =====

```

```
cross_reference(characteristics_space).  
cross_reference(hedonic_price_function).  
cross_reference(implicit_price).  
cross_reference(hedonic_regression).
```

## Recent Court Decisions

## Recent Court Cases — United States

*Recent Court Decisions • The Valuation Engineer*

William Bert Craytor • Vol. 1, Issue 1 • Summer 2026

DOI: 10.67330/fz34f711

---

### Case Summaries by Jurisdiction

#### California

**The Retail Property Trust v. Orange County Assessment Appeals Board No. 1.** California Court of Appeal, decided April 2026.

The Court of Appeal addressed the availability of property tax assessment relief under California Revenue and Taxation Code section 170 for access restrictions imposed by governmental orders issued during the COVID-19 pandemic. The plaintiff shopping center argued that restricted access during the pandemic constituted “damage” to the property within the meaning of section 170, entitling it to assessment relief.

*Implication for appraisers.* The opinion’s interpretation of “damage” under section 170 will shape future relief claims based on externally-caused diminutions in property utility, whether arising from public-health orders, infrastructure projects, or other governmental actions affecting access. Appraisers asked to opine on the impact of access restrictions on commercial property value should review the court’s reasoning on whether restricted utility—as distinct from physical injury—constitutes compensable damage for assessment-relief purposes.

#### Colorado

**Cheroutes v. Rocket Mortgage & Solidifi.** U.S. District Court, D. Colorado; pending, status updates in 2026.

The underlying complaint alleges appraisal bias under the Fair Housing Act in connection with a 2021 appraisal of a Denver duplex valued at approximately \$640,000—more than \$220,000 below the prior year’s appraised value. A federal judge previously denied the defendants’ motion to dismiss, holding that Rocket Mortgage and the appraisal management company Solidifi must defend the claims rather than disclaim responsibility on the theory that lenders and AMCs cannot be liable for the conduct of independent fee appraisers. The case is in discovery in 2026.

*Implication for appraisers.* The case is the leading active vehicle for testing AMC and lender liability under the Fair Housing Act for the conduct of fee appraisers. If the court ultimately holds the AMC and lender liable on the merits, it would substantially expand the universe of parties exposed to fair housing claims arising from appraisals, and would likely propagate into AMC–appraiser engagement contracts.

#### Florida

**Florida class action challenging AMC fees charged to residential borrowers.** Filed January 2, 2026.

The complaint challenges appraisal management company fees charged to residential mortgage borrowers, alleging that the markup between the fee paid to the fee appraiser and the fee charged to the borrower at closing exceeds the limits established under federal law and is inadequately disclosed. The case sits at the intersection of the Dodd-Frank “customary and reasonable” fee requirement (12 U.S.C. § 1639e) and the consumer-disclosure regimes under the Truth in Lending Act and the Real Estate Settlement Procedures Act.

*Implication for appraisers.* The litigation tests the structure of AMC compensation and, if certified as a class and decided on the merits, could reshape disclosure practices industry-wide. The outcome may indirectly affect the fee-setting environment for fee appraisers working through AMCs by exposing the spread between AMC-paid and borrower-charged fees to judicial scrutiny.

## Georgia

**Green Rock conservation easement cases.** United States Tax Court, opinions discussed in *Tax Notes* on March 2, 2026.

The Tax Court worked through each prong of the highest-and-best-use test in evaluating conservation easements granted on adjacent parcels (the Green Rock Property and Green Rock South) in the Piedmont province of Georgia. Petitioners’ experts asserted that the highest and best use of the properties before the easement grant was operation of a granite quarry, and used discounted cash flow analysis to estimate a before-easement fair market value of approximately \$13.75 million against an after-easement value of \$180,000.

The court expressed reservations on each prong of the test. On legal permissibility, the court was not persuaded that rezoning to permit quarry operations was reasonably probable, noting in particular that a zoning classification change for the Green Rock Property had been obtained shortly after a new conservation easement appraisal was requested—a sequence the court found difficult to credit. On physical possibility, the court found the petitioners’ geological evidence insufficient to establish that a viable quarry could be developed at the Harman Road parcel, notwithstanding the general known presence of mineable granite in the Piedmont province. The court further held that even if legal permissibility and physical possibility were assumed, this would not be controlling without satisfying the remaining prongs of the analysis.

*Implication for appraisers.* The opinion is a worked example of disciplined sequential application of the four-prong highest-and-best-use test. It reinforces the line established in *Ranch Springs, LLC v. Commissioner*, 164 T.C. No. 6 (2025), and *Beaverdam Creek Holdings, LLC v. Commissioner*, T.C. Memo. 2025-53, that the Tax Court will not accept speculative highest-and-best-use claims even where the general feasibility of the proposed use is documented in the surrounding region. Appraisers preparing easement valuations should develop parcel-specific evidence for each prong rather than rely on regional or analogically-supported feasibility, and should be especially cautious of timing patterns—such as zoning changes obtained contemporaneously with appraisal commissioning—that may invite judicial skepticism.

## Maryland

**Baltimore acquisition of Pimlico Race Course.** Eminent domain proceedings, ongoing 2025–2026.

The City of Baltimore’s acquisition of the Pimlico Race Course through the eminent domain power raises just-compensation questions concerning the relationship between going-concern value and

real property value for specialty-use facilities. Because Pimlico’s operating business is functionally inseparable from its physical plant, conventional real-property comparable sales analysis is of limited utility, and income-approach valuations depend heavily on the assumed continuation of the racing operation.

*Implication for appraisers.* The proceeding is worth monitoring for any appraiser asked to value a single-purpose facility where the operating business and the real estate are functionally inseparable, such as racetracks, motion-picture studios, theme parks, marinas, and certain healthcare or hospitality assets. The methodology that emerges from litigated proceedings will likely influence how just-compensation calculations treat going-concern increments in future condemnation matters.

## **Federal Fair Housing Act Claims Against Appraisers**

**Braxton v. Bank of America.** Filed April 2026.

A homeowner has sued Bank of America and a Virginia-based appraiser, alleging racial discrimination in violation of the Fair Housing Act. The plaintiff asserts that she and her tenants were present at the appraisal and that the appraiser’s conduct evidenced racial animus. The case continues the pattern established in earlier appraisal bias litigation, including the *Connolly/Mott v. loanDepot* matter in the District of Maryland and the *Austin v. Miller* matter in Marin County, California, of individual-plaintiff Fair Housing Act claims against appraisers, AMCs, and lenders.

*Implication for appraisers.* The continued filing pace of appraisal bias cases against individual appraisers, combined with the federal retrenchment on disparate-impact enforcement, indicates that the field of liability exposure has shifted: private plaintiffs and state-level fair-housing enforcers are now the principal sources of appraisal bias liability, rather than federal agencies. Appraisers should review documentation practices to ensure that comparable selection, adjustment derivation, and final-value reconciliation are independently defensible against post-hoc scrutiny, and should be aware that statements or conduct during the inspection itself are frequently the factual basis for plaintiffs’ claims.

# Recent EU Court Cases

*Recent Court Decisions • The Valuation Engineer*

William Bert Craytor • Vol. 1, Issue 1 • Summer 2026

DOI: 10.67330/r1yn1y02

---

## Scope and Sources

This survey is drawn from publicly available primary sources where accessible—most notably the Court of Justice of the European Union’s official case-law database ([curia.europa.eu](https://curia.europa.eu)), HUDOC for the European Court of Human Rights, the Bundesfinanzhof’s published decisions ([bundesfinanzhof.de](https://bundesfinanzhof.de)), the German Federal Constitutional Court ([bverfg.de](https://bverfg.de)), Légifrance for French jurisprudence, and the Upper Tribunal (Lands Chamber) for the United Kingdom—and from secondary commentary in professional and academic publications. Decisions in national languages have been read in the original and are described in English; case captions retain their original spelling and diacritics.

Subscription-only legal-research databases ([juris.de](https://juris.de), [beck-online](https://beck-online.com), Westlaw UK, Lamy, Aranzadi) are not used. National-court decisions below the supreme-court level are surveyed selectively, with priority given to decisions of clear precedential or methodological significance. Case citations herein should be verified against the court’s official report before any reliance for practice or for further publication.

This baseline installment treats Germany, France, and the United Kingdom as the principal national jurisdictions. Italy, Spain, the Netherlands, and the Nordic countries are not separately surveyed in this installment but will be added in subsequent quarterly installments as material warrants. Switzerland, although outside the European Union, will be surveyed in future installments because of the strong methodological influence of Swiss valuation standards on European practice generally.

## European Regulatory Context

Two regulatory developments at the European Union level materially affect the operating environment for appraisers, although neither is itself a court decision:

**EU AI Act and automated valuation models.** Regulation (EU) 2024/1689 (the EU AI Act) entered into force in 2024, with provisions applicable to high-risk AI systems taking effect on a staged basis through August 2026. On 19 May 2026 the European Commission published draft guidelines on the classification of high-risk AI systems for stakeholder consultation. Annex III of the AI Act lists creditworthiness assessment in connection with natural persons among the eight categories of presumptively high-risk use cases, and the Commission’s draft guidelines develop this category in detail. Because automated valuation models (AVMs) used in mortgage origination, mortgage portfolio monitoring, and insurance underwriting are functionally inseparable from creditworthiness assessment for the underlying loans, AVMs operated by credit institutions or insurers within the European Union will, in most configurations, be classified as high-risk AI systems. The compliance consequences include mandatory risk management systems, data governance controls, automatic

logging with at least six months of log retention, human oversight mechanisms allowing operators to understand and override system decisions, and post-market monitoring obligations.

*Implication for appraisers.* European appraisers should expect that AVM-supported workflows operated by their lender clients will be subject to substantially increased documentation and explainability requirements. Where appraisers provide model inputs, calibration data, or human-oversight review of AVM outputs, those activities themselves become elements of the lender’s high-risk-AI compliance posture.

**Energy Performance of Buildings Directive (recast).** Directive (EU) 2024/1275, the recast Energy Performance of Buildings Directive (EPBD), was published in the Official Journal on 8 May 2024 and entered into force on 28 May 2024. The transposition deadline for Member States is 29 May 2026. Among other provisions, the recast EPBD requires Member States to prioritize renovation of the worst-performing sixteen percent of the building stock (non-residential) and the worst-performing forty-three percent of the residential stock; sets minimum energy performance standards (MEPS); requires zero-emission buildings for new public construction from 2028 and all new construction from 2030; and mandates whole-life-carbon assessment for new buildings from 2030.

*Implication for appraisers.* The recast EPBD will affect property valuations through several channels: minimum-energy-performance thresholds create a new category of compliance-driven obsolescence; energy performance certificate (EPC) ratings become a material price determinant for both residential and commercial stock; and renovation costs to achieve compliance must be reflected in cost approaches and in the deduction of remaining economic life. Appraisers should also anticipate that lenders subject to the Mortgage Credit Directive will increasingly require explicit treatment of EPC ratings and MEPS-related capital expenditure requirements in collateral valuations.

## Notes for Subsequent Installments

Quarterly installments following this baseline will focus on changes: new decisions, dispositions of pending matters identified here (notably the German Federal Constitutional Court’s treatment of *1 BvR 472/26*), and Member State implementing legislation transposing the recast EPBD into national law. The Swiss legal framework, treatment of Italian and Spanish supreme-court decisions, and developments under the French *Conseil constitutionnel* will be added as material warrants.

## Case Summaries by Jurisdiction

### Court of Justice of the European Union

**GR Real (Case C-351/23).** Court of Justice of the European Union, judgment of 24 June 2025.

The Court of Justice ruled on a preliminary reference from a Slovak court arising from an extrajudicial mortgage enforcement procedure in which consumers lost their family home through auction while legal proceedings challenging potentially unfair contractual terms were still pending. The Court held that the consumer-protection guarantees under Directive 93/13/EEC require that national procedural rules permit effective judicial review of contractual fairness before a mortgaged residence may be sold through extrajudicial enforcement, and that procedures permitting the sale to proceed irreversibly before such review is completed are incompatible with the Directive.

*Implication for appraisers.* The decision is not directly a valuation case, but it reshapes the procedural environment in which mortgage-collateral valuations are commissioned and used. In several Member States with extrajudicial enforcement procedures (notably Slovakia, the Czech Republic, Hungary, and the Baltic states), the practical effect is to slow the enforcement-auction process and to extend the period during which the lender's collateral valuation is relied upon. Appraisers preparing collateral valuations for enforcement contexts in those jurisdictions should anticipate that valuation reports may be in use for longer periods than previously, and may be tested against market movements between the original valuation date and the eventual sale.

## European Court of Human Rights

**Baku building demolition (Azerbaijan), judgment of July 2025.** European Court of Human Rights, Article 1 of Protocol No. 1.

The European Court of Human Rights found a violation of the right to peaceful enjoyment of possessions in connection with the demolition and expropriation of a building in Baku that had been ordered following authorities' citation of earthquake risk. The Strasbourg court found that the domestic authorities, in determining compensation, had failed to provide sufficient reasons for rejecting the valuation report submitted by the applicants and had unjustifiably adopted only the amount of the expert opinion commissioned by the court itself. The Court awarded a total of €75,000 for pecuniary damage, €3,000 for non-pecuniary damage, and €1,055 for legal costs.

*Implication for appraisers.* The decision reinforces a line of Article 1 Protocol 1 jurisprudence requiring that domestic courts give reasoned consideration to competing valuation evidence in expropriation proceedings, and that the rejection of an applicant's valuation report in favor of a court-commissioned expert opinion must itself be reasoned. For European appraisers practicing in expropriation and just-compensation contexts, the case underscores that the documentary defensibility of a valuation report—the explicit reasoning supporting comparable selection, adjustment derivation, and final-value reconciliation—is itself a fundamental-rights matter when the report is submitted in support of a claim against the State.

## France

**Cour de cassation, Chambre commerciale, 19 June 2024, n° 22-24.169.** (Discussed in professional commentary through 2025.)

The Court of Cassation held, in the context of inheritance-tax *valeur vénale* (market value) disputes, that a sale of the inherited property occurring after the taxable event (the death of the *de cujus*) may not be used by the heirs to revise downward the *valeur vénale* originally declared, even where the post-event sale price is significantly lower and the sale occurs within months of the death. The decision applies the principle that *valeur vénale* for inheritance-tax purposes is determined as of the date of the taxable event and is not corrected by reference to post-event transactions in the same property.

*Implication for appraisers.* The decision confirms a strict date-of-valuation discipline in French inheritance-tax practice and limits the relevance of post-event sales as retrospective evidence of value. Appraisers preparing French inheritance-tax valuations should develop comprehensive comparable-sales evidence based on transactions *at or before* the valuation date, and should not present post-event sales of the subject property itself as confirmatory evidence. The same reasoning is likely to extend to gift tax (*droits de donation*) and to wealth-tax (*IFI*) contexts.

**Cour de cassation, 3<sup>e</sup> Chambre civile, 8 January 2026, n° 23-22.803.** Amicable (non-judicial) expert reports.

The Third Civil Chamber of the Court of Cassation issued a ruling redefining the legal value of amicable (non-judicial) expert reports in French civil proceedings. The decision follows the broader procedural reform effected by the décret of 18 July 2025, which made mediation and amicable dispute-resolution the default rule in civil procedure. The Court clarified that amicable expert reports prepared at the contractual instance of a single party may nonetheless be received as evidence and accorded probative weight, subject to the contradictory principle and to the court’s evaluation of the expert’s independence and methodology.

*Implication for appraisers.* The decision strengthens the evidentiary position of *expertises immobilières* prepared outside the framework of court-appointed judicial experts (*experts judiciaires*), provided the report observes the methodological standards of the profession and the expert’s independence is documented. For French appraisers working on contested valuations in inheritance, divorce, or commercial disputes, the decision creates a stronger foundation for client-commissioned reports to be cited and relied upon in subsequent litigation. Particular care to documenting independence (absence of subordination, economic interest, or family relationship with the client) and to applying the methods prescribed by professional standards becomes correspondingly more important.

## Germany

**Bundesfinanzhof, judgments of 12 November 2025 (II R 25/24, II R 31/24, II R 3/25).** **The *Grundsteuer Bundesmodell* trilogy—constitutionality of the reformed property-tax valuation system.** The Second Senate of the German Federal Fiscal Court (*Bundesfinanzhof*, BFH) ruled in three parallel revision proceedings that the provisions of the income-capitalization valuation method (*Ertragswertverfahren*) applied under the so-called *Bundesmodell* for the valuation of residential property in eleven German federal states for purposes of the reformed property tax (*Grundsteuer*) effective from 1 January 2025 are constitutional. The plaintiffs were apartment owners in North Rhine-Westphalia, Saxony, and Berlin who challenged their assessments on the grounds that the regimented valuation rules violate the general equality principle (Article 3(1) of the Basic Law) and produce systematically inaccurate values.

The BFH held that the legislature had created a valuation system *conceptually oriented toward market value* (*konzeptionell einer Verkehrswertorientierung folgt*) and designed to approximate, on average across all valued objects, the “objectively realistic property value” within a corridor of the fair market value (*gemeiner Wert*). The Court found that the legislature had not exceeded the margin permitted to it in balancing the goals of the valuation scheme against the inequalities necessarily attendant to such a scheme, and that the significant weight assigned to avoiding a renewed “valuation backlog” through automated periodic updating was constitutionally permissible. The BFH also emphasized that taxpayers retain the option, established by earlier BFH *Beschlüsse* of 27 May 2024 (II B 78/23 and II B 79/23), to demonstrate a lower fair market value (*gemeiner Wert*) through a qualified appraisal under § 198 BewG, with the limitation that the lower value must be substantiated by an appraisal report from a publicly appointed real estate appraiser applying recognized valuation methods.

A constitutional complaint (*Verfassungsbeschwerde*) supported by *Bund der Steuerzahler Deutschland* and *Haus & Grund Deutschland* is now pending before the Federal Constitutional Court (*Bundesverfassungsgericht*) under file number 1 BvR 472/26.

*Implication for appraisers.* The trilogy is the most significant German real estate valuation jurisprudence of the survey period. For appraisers, the principal practical consequence is the elevated demand for § 198 BewG “lower-value” appraisals (*Verkehrswertgutachten zum Nachweis des niedrigeren gemeinen Werts*) in cases where the *Grundsteuerwert* determined under the *Bundesmodell* appears to exceed the actual market value of the subject property. The BFH’s explicit endorsement of this pathway, combined with its general deference to the *Bundesmodell*, means that the appraisal report is now the operative mechanism for individual taxpayer relief from systematic overvaluation. Appraisers should expect a sustained increase in instructions for such reports and should ensure compliance with ImmoWertV 2021 methodology and the requirements of § 198 BewG. The pending constitutional complaint at the BVerfG (1 BvR 472/26) will be material to monitor for any further changes to this framework, and is expected to be a principal subject of the next quarterly installment.

**Bundesfinanzhof, judgment of 7 October 2025, IX R 26/24. Purchase-price allocation between land and building (*Kaufpreisaufteilung*).** In an income-tax case involving the depreciation (*Absetzung für Abnutzung*, AfA) basis of a residential rental property, the Ninth Senate of the Federal Fiscal Court addressed the methodology for allocating the contractual purchase price between the non-depreciable land component (*Grund und Boden*) and the depreciable building component (*Gebäude*). The plaintiffs had challenged the lower court’s reliance on *Bodenrichtwerte* (standard land values) on the grounds that the underlying data basis was not reliably documented for the specific location of the subject property.

*Implication for appraisers.* The decision continues a line of BFH jurisprudence (notably BFH II R 25/24 of 12 November 2025 cited above, and earlier BFH IX R 12/21 of 20 September 2022) confirming that ImmoWertV is the source of recognized valuation principles for purchase-price allocation. The case also illustrates the importance of the data basis underlying *Bodenrichtwerte*: where the *Gutachterausschüsse* (expert appraisal committees) do not have sufficient comparable-sales data for the specific location, the *Bodenrichtwert* may be challenged on data-quality grounds. For appraisers preparing purchase-price allocations or depreciation-basis opinions, this means that documentation of the *Bodenrichtwert* data basis—including the underlying *Kaufpreissammlung*—becomes part of the defensible analysis.

## United Kingdom

***Mehrunnisa Alikhan Ahmed Khan v Wolverhampton City Council.*** Upper Tribunal (Lands Chamber), 2025.

The Upper Tribunal (Lands Chamber) confirmed that the proper valuation date for assessing compulsory purchase compensation is the date on which the property vests in the acquiring authority under a general vesting declaration (GVD). The Tribunal also addressed the relationship between contemporaneous and retrospective valuations: in the case at hand, the local authority’s initial valuation made immediately after the GVD was £150,000, while a second valuation prepared in 2025 during tribunal proceedings, with reference to the assumed vesting date three months after the owner’s notice, was £120,000. Notwithstanding the Tribunal’s confirmation of the legal valuation date, it preferred the figure calculated closer in time to the date the GVD was made, resulting in a higher payment to the claimant.

*Implication for appraisers.* The decision is methodologically instructive for UK appraisers working in compulsory purchase contexts. While the legal valuation date is the vesting date, the Tribunal’s

preference for the contemporaneous valuation over a retrospective valuation prepared years later, on the same legal valuation date, illustrates that contemporaneous market evidence carries practical weight even where it would not formally be admissible at the vesting date itself. Appraisers should ensure that valuation reports prepared close to a GVD are documented with sufficient supporting evidence to serve as the primary record, anticipating that later tribunal proceedings may rely on the contemporaneous report in preference to retrospective reconstruction.

# Recent Latin America Court Cases

*Recent Court Decisions • The Valuation Engineer*

William Bert Craytor • Vol. 1, Issue 1 • Summer 2026

DOI: 10.67330/ywgzhg47

---

## Scope and Sources

This survey is drawn from publicly available primary sources where accessible—most notably the Inter-American Court of Human Rights case repository ([corteidh.or.cr](http://corteidh.or.cr)), the Brazilian Supreme Court (STF) and Superior Court of Justice (STJ) portals ([portal.stf.jus.br](http://portal.stf.jus.br), [stj.jus.br](http://stj.jus.br)), the Mexican *Semanario Judicial de la Federación* ([sjfsemanal.scjn.gob.mx](http://sjfsemanal.scjn.gob.mx)), the Argentine *Corte Suprema de Justicia de la Nación* ([csjn.gov.ar](http://csjn.gov.ar)), the Chilean *Poder Judicial* portal ([poderjudicial.cl](http://poderjudicial.cl)) and *Diario Constitucional*, and the Colombian *Corte Constitucional* ([corteconstitucional.gov.co](http://corteconstitucional.gov.co))— and from secondary commentary in professional and academic publications. Decisions in Spanish and Portuguese have been read in the original and are described in English; case captions retain their original spelling and diacritics.

Subscription-only legal-research databases (vLex, La Ley, Thomson Reuters Proview Latin America) are not used. National-court decisions below the supreme-court level are surveyed selectively, with priority given to decisions of clear precedential or methodological significance. Case citations herein should be verified against the court’s official report before any reliance for practice or for further publication.

This baseline installment treats Argentina, Brazil, Chile, Colombia, and Mexico as the principal national jurisdictions. Peru, Venezuela, Ecuador, Bolivia, Paraguay, Uruguay, and the Central American and Caribbean jurisdictions are not separately surveyed in this installment but will be added in subsequent quarterly installments as material warrants. Venezuelan property-restitution and Cuban sanctions-related valuation work will be surveyed in a future installment because of their methodological distinctiveness and their relevance to cross-border practice.

## Regional Context

Unlike the European Union survey, this Latin American survey has no direct analog to the supranational regulatory layer represented by the EU AI Act and the recast Energy Performance of Buildings Directive. Latin American real estate appraisal regulation is overwhelmingly national; the principal supranational influences are the Inter-American Court of Human Rights’ fundamental-rights jurisprudence under Article 21 of the American Convention (Right to Property) and the adoption of International Financial Reporting Standards (particularly IFRS 13, Fair Value Measurement) by most regional jurisdictions for purposes of financial reporting. Two methodological themes nonetheless run across the regional jurisprudence and warrant mention here:

**Valuation of indigenous communal property.** The most distinctive methodological challenge in Latin American appraisal practice is the valuation of indigenous and Afro-descendant communal property in restitution, expropriation, and compensation contexts. Inter-American Court jurisprudence has consistently held that indigenous communal property is constitutionally protected on the

same footing as individual private property, but the methods by which such property is valued—where there are typically no comparable arms-length transactions, where the property is communally and inalienably held, and where the cultural and spiritual significance of the territory is integral to its value—remain underdeveloped. As of late 2025, twenty-four Inter-American Court judgments in favor of indigenous peoples remained pending or only partially complied with, several involving outstanding compensation calculations that turn on unresolved valuation methodology.

**Inflation, currency, and indexation in expropriation compensation.** Several regional jurisdictions, notably Argentina, require valuation methodologies and indemnification calculations to contend with high inflation, foreign-currency tasaciones, and complex indexation regimes. The Argentine Supreme Court has held that expropriation indemnification originally expressed in foreign currency may not be converted to local currency at an artificial parity rate, and that the valuation must reflect the property’s real economic value at the time of indemnification. Similar concerns apply in Venezuela (in connection with restitution and sanctions-related valuations) and periodically in Brazil during periods of elevated inflation. For appraisers practicing across regional jurisdictions, these inflation-and-currency considerations are integral to the valuation methodology in a way that has no European or United States parallel.

## Notes for Subsequent Installments

Quarterly installments following this baseline will focus on changes: new Inter-American Court decisions, dispositions of pending matters identified here, developments in the implementation of Colombia’s Jurisdicción Agraria y Rural toward its 2027 operational start, and Member State developments under IFRS 13 fair-value-measurement practice. Peru, Venezuela, Ecuador, and the Central American jurisdictions will be added as material warrants. Particular attention will be given to the methodological development of indigenous communal-property valuation, where the Inter-American Court’s substantive holdings have outpaced the practical valuation methodology available to appraisers and compensation authorities.

## Case Summaries by Jurisdiction

### Inter-American Court of Human Rights

*Maya Q’eqchi’ Agua Caliente vs. Guatemala and Comunidad Garífuna de San Juan y sus Miembros vs. Honduras.* Inter-American Court of Human Rights, judgments of 2024, discussed in regional jurisprudence through 2025.

In two contemporaneous judgments the Inter-American Court ruled on the obligations of Guatemala and Honduras to demarcate, title, and prevent the dispossession of ancestral territories belonging to indigenous and Afro-descendant communities. The Maya Q’eqchi’ Agua Caliente community had obtained a definitive property title to its territory only after forty-five years of administrative and judicial action; the Garífuna community of San Juan in Honduras faced ongoing dispossession of ancestral coastal lands. Both judgments addressed the State’s obligations under Article 21 (right to property) of the American Convention to recognize, demarcate, and protect indigenous communal property. Regional commentators have observed that the Inter-American Court’s restitution framework in these decisions is somewhat more restrictive than in earlier judgments, particularly in its treatment of “innocent third parties” (*terceros inocentes*) who acquired overlapping rights without notice of the indigenous claim.

*Implication for appraisers.* The decisions reinforce that indigenous and Afro-descendant communal property is constitutionally protected throughout the Inter-American system and that valuation of such property in restitution or compensation contexts cannot rely on conventional arms-length comparable sales (which typically do not exist) without methodological supplementation. Appraisers commissioned in indigenous-restitution contexts should anticipate that the valuation must integrate three distinct components: the underlying market value of land of comparable physical characteristics in the region; the opportunity costs and improvements attributable to indigenous use; and the non-economic cultural and spiritual significance of the territory, which Inter-American Court jurisprudence treats as a compensable dimension of the right to property. The methodology for the third component remains underdeveloped and is a continuing subject of Inter-American Court reasoning.

**Mapuche / Consejo de Todas las Tierras vs. Chile.** Inter-American Court of Human Rights, judgment of 2024 (discussed in 2025 jurisprudence).

The Inter-American Court condemned the Chilean State for human-rights violations against 135 members of the Mapuche people in connection with the territorial reivindication actions of the Consejo de Todas las Tierras (1989–1992) and the application of anti-terrorism legislation to indigenous territorial claims. The judgment extends the doctrine of *Norín Catrimán and others vs. Chile* (2014) and explicitly recognizes the right of indigenous peoples to self-determination in connection with territorial claims.

*Implication for appraisers.* While the principal holdings of the case concern criminal-procedural protections rather than valuation methodology directly, the recognition of indigenous self-determination in connection with territory creates a procedural overlay that affects valuation work for Chilean indigenous-territorial restitution and related contexts. Appraisers commissioned in such contexts should ensure that the methodology incorporates appropriate consultation processes and recognizes the community’s self-determination over how its territorial values are characterized.

## Argentina

**Argentine doctrine on the Tribunal de Tasaciones de la Nación.** Consolidated through 2025 jurisprudence.

The Argentine Supreme Court of Justice (CSJN) and lower federal tribunals have continued to develop the doctrine surrounding valuations performed by the Tribunal de Tasaciones de la Nación (TTN) in expropriation and *expropiación inversa* matters. The established CSJN doctrine (Fallos 313:82, repeatedly applied through 2024–2025) holds that TTN valuations carry presumptive weight by reason of the technical capacity of the tribunal’s members, the evidentiary basis on which they are founded, and the uniformity of practice; departure from a TTN valuation requires the showing of manifest error or omission. Federal tribunals applying this doctrine in 2024–2025—including the Juzgado Federal de Oberá in *Sosa v. Entidad Binacional Yacyretá* (June 2024, *expropiación inversa* arising from the Yacyretá hydroelectric reservoir)—continue to rely on TTN valuations as the operative anchor for indemnification.

*Implication for appraisers.* For Argentine appraisers, the practical consequence is that private appraisal reports submitted in opposition to TTN valuations face a substantial evidentiary burden. To displace or modify a TTN valuation, the private report must demonstrate methodological error or omission in the TTN’s work, not merely an alternative valuation conclusion. Appraisers preparing such reports should focus the methodological critique on the TTN’s comparable-sales selection,

adjustment derivation, and treatment of local market conditions, rather than on the magnitude of the value difference alone.

**Foreign-currency tasaciones and indemnification parity.** The CSJN doctrine that expropriation indemnification originally expressed in foreign currency may not be converted to Argentine pesos at an artificial one-to-one parity remains operative through 2025. The doctrine protects expropriated owners from indemnifications that, due to inflation or currency policy, fail to reflect the property's real economic value at the time of indemnification. The principle has been periodically reaffirmed in 2024–2025 in connection with long-running expropriation proceedings.

*Implication for appraisers.* Argentine appraisers preparing valuations for expropriation matters that span periods of significant inflation or currency-policy change should explicitly document the currency basis of the valuation and should consider the appropriate indexation methodology between the valuation date and the prospective date of payment. Reports that fail to address inflation and currency risk are unlikely to be persuasive to tribunals applying CSJN indemnification doctrine.

## Brazil

**Supremo Tribunal Federal, RE 922.144 (Tema 865 of general repercussion).** Brazilian Supreme Federal Tribunal, judgment of 2024 with continuing 2025 application.

The STF held that indemnification in *desapropriação* proceedings is constitutionally compatible with the federal *precatório* regime established by Article 100 of the Federal Constitution. Specifically, the difference between the value initially deposited by the public authority (for purposes of *imissão provisória na posse*) and the value ultimately determined by judicial expertise must be paid through the *precatório* system rather than as an immediate complementary deposit. The decision balances the constitutional command of *justa, prévia e em dinheiro* compensation (Article 5, XXIV) against the orderly-payment requirements of the *precatório* system.

*Implication for appraisers.* The decision has direct practical consequences for the initial-deposit valuation prepared by appraisers acting for expropriating authorities. Because any underestimation in the initial deposit will be paid later through the *precatório* system (which may involve substantial delay), there is a structural incentive for plaintiffs to challenge initial valuations as understated. Appraisers preparing initial-deposit valuations should document the methodology with particular care to defend the valuation against subsequent judicial expertise, anticipating that the gap between the initial deposit and the eventual judicial determination will be subject to compensatory interest (*juros compensatórios*) under STF jurisprudence and the Decreto-Lei 3.365/1941 framework.

**Superior Tribunal de Justiça, REsp 2.129.162 (Tema repetitivo).** Brazilian Superior Court of Justice, judgment of May 2025.

The STJ resolved a repetitive controversy concerning attorneys' fees in cases where the expropriating authority desists from the *desapropriação* action. The Court established that fees in desistance cases must follow the framework of Article 27, § 1, of Decreto-Lei 3.365/1941 rather than the general procedural rules of the *Código de Processo Civil*. The decision arose from a Cemig (Companhia Energética de Minas Gerais) action seeking an administrative easement for an electricity distribution line.

*Implication for appraisers.* The decision is procedural rather than methodological, but it has implications for the structure of appraiser engagement in administrative-easement and *desapropriação*

matters. Appraisers retained as expert witnesses (*peritos judiciais*) in such proceedings should ensure that fee arrangements are documented with reference to the applicable decree-law framework rather than to general procedural rules.

**Superior Tribunal de Justiça, doctrine on *juros compensatórios*.** Continuing 2025 application.

The STJ has maintained its doctrine that compensatory interest (*juros compensatórios*) in *desapropriação* matters incide only after the judicial decision on the title of the expropriated property. The interest applies to the difference between the initial deposit and the final indemnification value, at the historically established rate of twelve percent per year (Súmula 618 of the STF), and it serves to compensate the expropriated owner for the anticipated loss of possession before the final indemnification is paid.

*Implication for appraisers.* Brazilian appraisers should anticipate that the valuation gap they identify in judicial expertise—between the public authority’s initial deposit and the appraised market value—becomes the base for compensatory interest calculations. The economic significance of the gap is therefore greater than the nominal difference suggests, particularly in long proceedings.

## Chile

**Corte Suprema, Rol N° 31.423-2025.** Chilean Supreme Court, judgment of September 2025.

The Chilean Supreme Court rejected a cassation appeal brought by a real estate developer (inmobiliaria) against the State Treasury (*Fisco de Chile*) in a special expropriation proceeding under Article 12 of Decree-Law N° 2.186. The plaintiff had argued that the valuation set by the Commission of Experts (*Comisión de Peritos*) failed to reflect the property’s highest and best use (*mayor y mejor uso*) and the value of existing constructions and complementary works, and had additionally sought indemnification for lost profits (*lucro cesante*) on the ground of an existing lease contract that would have generated future rental income. The Court held that the plaintiff’s arguments amounted to a disagreement with the lower court’s weighing of evidence rather than to a true infringement of the legal norms governing valuation, and that the lower court’s reasoning—which relied on the Commission of Experts’ evidence in conjunction with documentary and testimonial evidence—was adequately founded.

*Implication for appraisers.* The decision illustrates the substantial deference Chilean Supreme Court doctrine accords to the Commission of Experts’ valuation in expropriation proceedings, and the limited scope for cassation review of valuation conclusions on the ground of methodological disagreement alone. Appraisers preparing opposing valuations in Chilean expropriation matters should focus on demonstrable methodological error in the Commission’s work rather than on alternative-valuation reasoning, and should be aware that *lucro cesante* claims based on existing lease arrangements face substantial evidentiary requirements.

**Corte Suprema, Rol N° (Coquimbo expropriation).** Chilean Supreme Court, judgment of 17 July 2025.

The Chilean Supreme Court ratified an appellate decision raising indemnification in a Coquimbo expropriation matter to UF 2,969.75 (based on a private appraiser’s report) against the State’s argued amount of UF 2,349.07 (based on inscribed sales from the *Conservador de Bienes Raíces*). The Court emphasized that the appellate court’s reliance on the private appraiser’s report, considered

together with testimonial evidence and an additional appraisal report on a nearby property, was a permissible exercise of *sana crítica* (reasoned weighing of evidence) and that cassation review does not extend to second-guessing such evidentiary weighing.

*Implication for appraisers.* The Coquimbo decision is the counterpart to Rol N° 31.423-2025: where the Commission's valuation prevails when challenged by a developer, a private appraiser's higher valuation can also prevail when sustained by adequate methodological reasoning and corroborative evidence. The practical lesson for Chilean appraisers is that the methodology and documentary support for a private appraisal are decisive, and that inscribed-sales evidence alone is insufficient to displace a well-supported private appraisal that draws on additional comparables and contextual evidence.

#### **Tribunal Constitucional, October 2025.** *Bienes raíces* tasation under the Tax Code.

The Chilean Tribunal Constitucional rejected a constitutional challenge to the provisions of the Código Tributario governing the tasation of real estate (*bienes raíces*) for purposes of the impuesto territorial. Dissenting ministers argued that the impugned norms provided inadequate mechanisms for the executed party to object to the tasation, compromising the constitutional guarantee of a rational and just procedure.

*Implication for appraisers.* The decision preserves the existing administrative tasation framework for Chilean real estate tax, but the dissenting reasoning identifies a vulnerability in the current framework that may be revisited in future constitutional or administrative-law challenges. Chilean appraisers preparing valuations in connection with real estate tax disputes should monitor the post-decision policy environment, as administrative reforms to improve taxpayer-objection mechanisms appear plausible.

## **Colombia**

#### **Corte Constitucional, approval of the Jurisdicción Agraria y Rural.** Colombian Constitutional Court, 2025 decision.

The Colombian Constitutional Court approved the constitutional review of the legislation creating the new Jurisdicción Agraria y Rural, a specialized rural-land jurisdiction within the Colombian judicial branch. The Court approved key articles defining the object of the reform, its integration into the judicial branch, the creation of specialized tribunals and courts, and the budgetary aspects. The new jurisdiction is scheduled to begin operations in 2027 and is designed to resolve land-restitution disputes, agrarian-property disputes, and matters arising from the implementation of the 2016 peace accord. As of the date of approval, agrarian disputes affecting more than 61,000 hectares are pending in the existing administrative and judicial system.

*Implication for appraisers.* The new jurisdiction will become the principal forum for valuation disputes arising from land-restitution proceedings and rural-property matters in Colombia. Colombian appraisers practicing in restitution contexts (including those working for the Unidad de Restitución de Tierras and the Instituto Geográfico Agustín Codazzi) should anticipate that the new tribunals will develop their own valuation jurisprudence and that methodological standards may evolve from the current administratively-driven framework. Practitioners should monitor the operational launch in 2027 and the development of the tribunals' initial caseload.

**Existing IGAC-based valuation framework for land-restitution.** Continuing 2025 application.

Colombian land-restitution practice continues to rely on the *avalúo comercial* prepared by the Instituto Geográfico Agustín Codazzi (IGAC) as the principal basis for compensation in restitution and substitution contexts, supplemented by avalúos prepared by Lonjas de Propiedad Raíz where required. The framework is articulated in Constitutional Court decision SU-034 of 2018 (on the procedural requirements for compliance) and in subsequent decisions including T-120 of 2024 (on the relationship between IGAC valuation and individual beneficiary preferences in compensation-in-kind situations).

*Implication for appraisers.* For appraisers working in Colombian land-restitution, the institutional framework places IGAC valuations at the operative center of the compensation calculation, with Lonja de Propiedad Raíz avalúos serving as a supplementary or controversy-resolution mechanism. The forthcoming Jurisdicción Agraria y Rural may alter this institutional architecture, and the relative weight of IGAC versus Lonja valuations is one of the issues likely to be developed by the new tribunals.

## Mexico

**Suprema Corte de Justicia de la Nación, ISR liability on avalúo–price differential.** Mexican Supreme Court, decision announced in early 2026.

The Pleno of the Mexican Supreme Court (SCJN) declared constitutional Articles 125, 130(IV), and 132 of the Ley del Impuesto sobre la Renta (ISR), which provide that when the avalúo of a real estate acquisition exceeds the contractual purchase price by more than ten percent, the difference is treated as accumulable income to the purchaser and is therefore subject to ISR. The Court held that the rule does not violate the constitutional principles of proportionality and equity in taxation, on the reasoning that the acquisition of property at a price materially below market value represents an objective increase in the purchaser’s patrimony and therefore properly constitutes a taxable economic benefit.

*Implication for appraisers.* The decision has substantial practical consequences for Mexican real estate transactions and for the appraisal practice supporting them. Avalúos performed by *corredores públicos* or other authorized professionals now have direct fiscal consequences for purchasers when the avalúo exceeds the contractual price by more than ten percent. Appraisers should anticipate increased scrutiny of methodology in transactional avalúos and increased demand for avalúos that document the comparable-sales basis with particular rigor. The decision also implicitly elevates the consequence of methodological choice in selecting between alternative valuation approaches, since the choice may now affect the purchaser’s tax liability.

**Suprema Corte de Justicia de la Nación, Amparo en revisión 391/2024.** Mexican Supreme Court Second Chamber, judgment of 25 June 2025.

The Mexican Supreme Court Second Chamber resolved a constitutional challenge by a mining company to a Decree affecting the legal framework for expropriation of mining lots and related ancillary rights (temporary occupation, easements, water-concession preferences, and lease-renewal provisions). The Court held that the Decree did not violate the principle of non-retroactivity because the affected provisions concerned mere expectations of future rights rather than acquired rights.

*Implication for appraisers.* The decision is significant for Mexican mineral-rights and mining-property valuation. The Court's characterization of mining-lot extension rights and related provisions as "mere expectations" rather than acquired rights affects the valuation of mining concessions and the assessment of regulatory risk in their valuation. Appraisers preparing valuations of mining properties or related rights in Mexico should explicitly address the regulatory-risk dimension of the valuation and should avoid treating contingent or extension-dependent values as if they were fully-vested.