mgcvUI: An Interactive Interface for the mgcv (Generalized Additive Models) Package

William Bert Craytor

Vol. 1, Issue 1 · Summer 2026

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

Contents

1 Introduction **4**
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 . . . 6
1.5 Getting Started . . . 7

2 MLS Input Data Requirements **8**
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

3 General Purpose Mode **10**
3.1 Overview . . . 10
3.2 The Sidebar Workflow . . . 10
3.3 Import from earthUI (Optional) . . . 10
3.4 Main Panel Tabs . . . 11
3.5 Settings Persistence . . . 11
3.6 Dark Mode . . . 11
3.7 Locale & Regional Settings . . . 11

4 Appraisal Mode **13**
4.1 Subject Row Handling . . . 13
4.2 Effective Date & Sale Age . . . 13
4.3 Special Column Designations . . . 13
4.4 RCA Adjustments Overview . . . 13

5 Market Area Analysis Mode **14**
5.1 Differences from Appraisal Mode . . . 14
5.2 When to Use Market Mode . . . 14

6 Variable Selection **15**
6.1 Target Variable . . . 15
6.2 Response Transform . . . 15
6.3 The Predictor Table . . . 15
6.4 Data Type Detection & Overrides . . . 16
6.5 Special Column Types Reference . . . 16

7 Parameter Selection **17**
7.1 Parameter Presets . . . 17
7.2 Family . . . 17
7.3 Method . . . 17
7.4 Gamma (Smoothing Multiplier) . . . 18
7.5 Cross-Validation . . . 18
7.6 Select (Shrinkage) . . . 18
7.7 Default Basis . . . 18
7.8 Default k (Basis Dimension) . . . 18
7.9 Tensor Type . . . 19
7.10 Allowed Interactions . . . 19
7.11 Factor-by-Smooth Interactions . . . 19
7.12 Advanced Parameters . . . 19

8 Fitting the Model **20**
8.1 The Fit Button . . . 20
8.2 What Happens During Fitting . . . 20
8.3 Background Fitting & Progress . . . 20

9 Result Tabs **21**
9.1 Data Preview . . . 21
9.2 Equation . . . 21
9.3 Summary . . . 21
9.4 Variable Importance . . . 21
9.5 Contribution Plots . . . 21
9.5.1 Univariate Smooth Terms — $s(x)$ . . . 22
9.5.2 Factor-by-Smooth Terms — $s(x, \text{by} = \text{factor})$ . . . 22
9.5.3 Interaction Terms — $ti(x_1, x_2)$ and $te(x_1, x_2)$ . . . 22
9.5.4 Parametric Terms — Linear and Factor . . . 22
9.6 Correlation . . . 22
9.7 Diagnostics . . . 22
9.8 RCA Adjustments . . . 22
9.9 Anova . . . 23
9.10 Mgcv Output . . . 23
9.11 Sign Check . . . 23
9.12 Concurvity . . . 23
9.13 Basis Check . . . 23

10 Downloading Data **24**
10.1 Output Columns . . . 24
10.2 CQA Scores . . . 24

11 RCA Calculations & Downloading **25**
11.1 Opening the RCA Dialog . . . 25
11.2 CQA Score Interpolation . . . 25
11.3 Output Columns . . . 25

12 Sales Comparison Grid **26**
12.1 Comp Selection Dialog . . . 26
12.2 Workbook Layout . . . 26
12.3 Sheet Protection & Residual Cells . . . 26

13 Downloading Reports **27**
13.1 Report Formats . . . 27
13.2 Report Contents . . . 27

14 Exporting Model Functions **28**
14.1 R Functions . . . 28
14.2 Python Functions . . . 28
14.3 C++ Header . . . 28
14.4 SQLite Database . . . 28

15 Comparison with earthUI and glmnetUI **29**
15.1 Key Differences . . . 29
15.2 Shared Features . . . 29
15.3 Recommended Workflow . . . 29

16 Demo Dataset: Appraisal_1.csv **30**
16.1 Description . . . 30
16.2 Columns . . . 30
16.3 Suggested Quick Start . . . 30

17 System Requirements & Troubleshooting **32**
17.1 Supported Platforms . . . 32
17.2 R Version . . . 32
17.3 Required R Packages . . . 32
17.4 Optional Dependencies . . . 32
17.5 Platform-Specific Notes . . . 32
17.5.1 macOS . . . 32
17.5.2 Windows . . . 33
17.5.3 Linux . . . 33
17.6 Graceful Degradation . . . 33
17.7 Troubleshooting . . . 33
17.7.1 “PDF option is not available” . . . 33
17.7.2 “Settings will not persist” . . . 34
17.7.3 “Diagnostics plot unavailable” . . . 34
17.7.4 “Report error: number of items to replace. . . ” . . . 34
17.7.5 Port 7880 already in use . . . 34
17.7.6 Package installation fails on Linux . . . 34

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 wiggliness 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)

mgcvUI, earthUI, and glmnetUI are companion applications that use different modeling engines. The following table summarizes the key differences:

Feature	mgcv (GAM)	earth (MARS)	glmnet (Elastic Net)
Model type	Smooth nonlinear (penalized splines)	Piecewise linear (adaptive splines)	Linear (with regularization)
Nonlinearity	Automatic via smooth functions (thin plate, cubic, etc.)	Automatic via hinge functions with data-driven knots	Only through interactions or basis expansion
Variable selection	Optional via double-penalty shrinkage (`select=TRUE`)	Built-in via forward/backward stepwise with GCV pruning	Built-in via L_1 penalty (Lasso)
Interactions	Tensor product smooths (`te()`, `ti()`)	Hinge-based products (interpretable, up to degree 3)	Pairwise cross-products (hard to interpret)
Interpretability	Smooth partial effect curves; very intuitive	g-functions show piecewise linear partial effects per variable	Coefficients are linear weights; easy to explain individually
Best for	Smooth nonlinear relationships, flexible modeling	Automatic nonlinearity and interaction detection	High-dimensional data, variable selection, small samples
RCA adjustments	Smooth adjustments via partial effects	Piecewise linear adjustments via g-functions	Linear per-variable adjustments
Overfitting control	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.

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 βx 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

When you launch mgcvUI, a **Purpose** radio button at the top of the sidebar lets you choose one of three modes:

- **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 Reconciliation by Comparable Adjustment (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 Reconciliation by Comparable Adjustment (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 the package** in R: `install.packages("mgcvUI")` or install from source.
2. **Launch the application**: run `mgcvUI::mgcvUI()` in R. The app opens in your web browser on port 7880. The app remembers your last-used purpose mode and restores it automatically.
3. **Import your data** using the file upload in Section 1 of the sidebar. mgcvUI accepts CSV and Excel files.
4. **Import from earthUI (optional)** — Section 2 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.
5. **Select your Purpose** (General, For Appraisal, or Market Area Analysis). Note that changing the purpose resets both the data import and the earthUI import — you must re-import after switching.
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 Mgcv GAM Model" and review the results in the main panel.
9. **Export** — download predictions as Excel, generate reports, or (in appraisal mode) compute RCA adjustments and sales grids.

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	`contract_date`	Used to compute `sale_age` (days from effective date)
Listing Date	`listing_date`	Used with contract date to compute DOM if no DOM column exists
Days on Market	`dom`	Days on market; displayed in exports
Concessions	`concessions`	Sale concessions; Net SP = Sale Price − Concessions
Living Area (SF)	`living_area`	Enables per-SF residuals (`residual_sf`, `cqa_sf`)
Lot Size	`lot_size`	Site size column
Site Dimensions	`site_dimensions`	Grouped with lot size
Latitude	`latitude`	Rounded to 3 dp
Longitude	`longitude`	Rounded to 3 dp
Area ID	`area`	Market area / neighborhood identifier
Actual Age	`actual_age`	Property age
Effective Age	`effective_age`	Effective property age
Address	`display_only`	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**: Categorical variables like area ID, style, or condition should contain a manageable number of unique values. mgcvUI auto-detects factors but you can override.

> ***Tip:*** *Review the NA column in the predictor table after import. Columns with many missing values may cause rows to be dropped. Columns with ≥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 data import through export:

1. Import Data — File upload accepting CSV and Excel files. For Excel files with multiple sheets, a sheet selector appears. Column names are automatically converted to snake_case.

2. Import from earthUI (optional) — Import an earthUI result `.rds` file to seed the GAM with earth-discovered knot positions. This step is optional — skip it if you do not have an earthUI result to import.

3. Project Output Folder — A text field specifying where downloads are saved (defaults to `~/Downloads`).

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.

7. Download Output — Exports predictions, residuals, CQA scores, and per-variable contributions as an Excel file. Available in all purpose modes.

8. Download Report — Generates a formatted report (Word, PDF, or HTML) saved to the output folder. Steps 8–10 (RCA Adjustments, Sales Grid, Report) appear only in Appraisal/Market modes; in General mode the report step is numbered 8.

3.3 Import from earthUI (Optional)

Section 2 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.

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 Preview	Interactive DataTable of imported data.
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 `gratia`). 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	`gam.check()` 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 last-used purpose mode is also persisted globally and restored when the app is relaunched.

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 as my default** to store your locale preferences globally.

> ***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 you select **For Appraisal** as the Purpose, 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 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 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, 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 ≤ 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 βx). 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 βx 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:

- `weights` — 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
- `listing_date` — Used as a fallback for computing Days on Market
- `dom` — Identifies the Days on Market column

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 (discovery)	tp (thin plate)	k = auto (10), gamma = 1.2, select = TRUE
Earth Pipeline (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
gaussian	Continuous responses (e.g., sale price). Most common.
Gamma	Positive continuous responses with variance proportional to the mean.
poisson	Count data (e.g., number of sales).
binomial	Binary outcomes.
inverse.gaussian	Highly right-skewed positive data.

7.3 Method

The smoothing parameter estimation method:

Method	Description
REML	Restricted Maximum Likelihood (default, recommended). Most robust against overfitting.
GCV.Cp	Generalized Cross-Validation. Tends to select slightly more complex models.
ML	Maximum Likelihood. Similar to REML but can undersmooth.
P-REML	Pearson REML variant.
P-ML	Pearson ML variant.
GACV.Cp	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
tp	Thin plate regression spline	Default. Isotropic (no knot placement needed). Good general-purpose choice.
cr	Cubic regression spline	Allows explicit knot placement. Required for earth knot integration.
ps	P-spline	Evenly-spaced B-spline basis with difference penalty. Computationally efficient.
bs	B-spline	Flexible B-spline basis.

7.8 Default k (Basis Dimension)

The basis dimension k controls the maximum wiggliness 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
ti	`ti(x1, x2)`	Tensor interaction — models only the interaction effect, with main effects handled separately by univariate `s()` terms. Preferred for interpretability.
te	`te(x1, x2)`	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 Preview

Shows the imported data as an interactive DataTable with horizontal scrolling and 15 rows per page.

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(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 — $s(x, \mathbf{by} = \mathbf{factor})$

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 Mgcv 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 ∼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** button (sidebar section 7) exports an Excel file with predictions and diagnostics.

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., `est_sale_price`), back-transformed from log scale if applicable
`residual`	Actual − predicted
`calc_residual`	Verification: actual − sum-of-contributions (back-transformed)
`cqa`	Competitive Quality Adjustment score (0–10 scale)
`residual_sf`	Residual / living area (if `living_area` designated)
`cqa_sf`	CQA calculated from ranking via `residual_sf`

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 (Reconciliation by Comparable Adjustment) workflow is available in **Appraisal** mode only, after fitting the model.

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**: Numeric input, 0.00–9.99, default 5.00.

Click **Generate** to compute and download.

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
`subject_value`	Model prediction + interpolated residual
`subject_cqa`	User-entered CQA score
`<var>_adjustment`	Subject contribution − comp contribution (in dollars)
`residual_adjustment`	Subject residual − comp residual
`net_adjustments`	Sum of all adjustments
`gross_adjustments`	Sum of absolute adjustments
`residual_adj_pct`	Residual adjustment as % of sale price
`net_adj_pct`	Net adjustments as % of sale price
`gross_adj_pct`	Gross adjustments as % of sale price
`adjusted_sale_price`	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 sale age.

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

13.1 Report Formats

Three formats are available:

- **HTML** — Self-contained HTML document via `rmarkdown`. Works on all platforms with no extra software. Recommended default.
- **Word (.docx)** — Formatted document built via the `officer` package. Suitable for editing and distribution. Works on all platforms.
- **PDF** — Generated via `rmarkdown` + LaTeX. Paper size follows the locale setting (Letter or A4). **Requires a LaTeX installation** (see System Requirements below). If no LaTeX is detected, the PDF option is automatically hidden from the format dropdown.

Reports are saved to the output folder specified in Section 3.

13.2 Report Contents

Reports include:

- **Equation**: Model equation with f_i notation for smooth terms and explicit coefficients for linear terms, smooth function definitions table (term, basis, k, knots), and family/method info. Long equations wrap across multiple lines in PDF.
- **Model overview**: R^2, CV R^2, Deviance Explained, AIC, BIC, n, family, method, fitting time
- **Summary tables**: Smooth terms (EDF, F, p-value) and parametric terms (Estimate, Std Error, t, p-value)
- **Variable importance**: Bar chart ranked by F-statistic (smooths) or $|t|$ (parametric)
- **Contribution plots** (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)
- **Correlation matrix** heatmap
- **Diagnostics**: 4-panel residual diagnostics (with `gam.check` fallback if `gratia::appraise` fails) and actual vs predicted scatter
- **ANOVA**: Separate parametric and smooth term tables
- **Concurvity**: Overall (transposed for readability) and pairwise worst case
- **Earth sign consistency** (when earth knots imported)
- **Basis dimension check** and full mgcv output

Each section is independently error-protected — if one section fails, it shows an inline error message and the rest of the report generates normally.

A white checkmark appears on the Download 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`.

The **Export Functions** section (available in the report module) offers checkboxes for R, Python, C++, and SQLite. Click **Download Functions (.zip)** to generate 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_rsq, dev_explained, aic, n_obs, family, method, formula.
- Table `linear_terms`: variable, coefficient.
- Table `intercept`: intercept value.

15 Comparison with earthUI and glmnetUI

mgcvUI, earthUI, and glmnetUI are companion tools 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	`select=TRUE` shrinkage	Stepwise pruning	Lasso penalty
Interactions	Tensor products (te/ti)	Hinge products (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++, SQLite	R, Python, C++, SQLite	—
Report formats	Word, PDF, HTML	Word, PDF, HTML	Word, PDF, HTML

15.2 Shared Features

All three tools provide:

- 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 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.

16.2 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

16.3 Suggested Quick Start

1. Launch mgcvUI: `mgcvUI::mgcvUI()`
2. Import `Appraisal_1.csv` via the file upload
3. Set Purpose to **For Appraisal**
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 = 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 **pandoc** (needed for HTML/PDF reports) 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, bslib	Web application framework and Bootstrap 5 theming
mgcv	GAM model fitting
ggplot2, plotly	Static and interactive contribution plots
gratia	Smooth term extraction and diagnostics
DT	Interactive data tables
officer	Word (.docx) report generation
rmarkdown	HTML and PDF report generation
readr, readxl	CSV and Excel file import
openxlsx	Excel output (predictions, RCA, sales grid)
RSQLite, DBI	Settings persistence (SQLite database)
jsonlite	Variable state serialization

17.4 Optional Dependencies

Component	When Needed
LaTeX (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: `tinytex::install_tinytex()`
sysfonts + showtext	Roboto Condensed font for ggplot2 plots. If unavailable or offline, the app falls back to the system sans-serif font automatically.
earth ($\geq$ 5.3.0)	Only needed for the earthUI import pipeline. Not required for standalone use.

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`.
- **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 `mgcv::gam.check()` in the console as an alternative.
Temp directory not writable	Report generation fails with a clear error message. Resolve by setting `TMPDIR` 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.

— (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.

— (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.

— (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.