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

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

1 Introduction

1.1 History and Background

1.1.1 The Lasso

The **Lasso** (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator) 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 α 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 (`select=TRUE`)
Interactions	Pairwise cross-products (hard to interpret)	Hinge-based products (interpretable, up to degree 3)	Tensor product smooths (`te()`, `ti()`)
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	`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. 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, nlambda, 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 ($\lvert\beta\rvert \times \mathrm{sd}(x)$). 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
Variable	Column name
Type	Data type dropdown: `numeric`, `integer`, `character`, `Date`, `POSIXct`
Include	Checkbox — include this column as a predictor in the model
Factor	Checkbox — treat the variable as a categorical factor (creates dummy columns in the model matrix)
Force	Checkbox — force into model (penalty factor = 0, never dropped by lasso)
Special	Dropdown (appraisal/market only) — see Special Column Types Reference below
Sign	Expected coefficient sign: positive, negative, or either
NAs	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
Cross-validation	Recommended. Tests many lambda values using k-fold CV.
Manual	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
gaussian	Continuous responses (e.g., sale price). Most common.
binomial	Binary outcomes (e.g., sold/not sold).
poisson	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 ≥ 0
- A "negative" sign forces the coefficient ≤ 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
- **GR²** — 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 ± 1 standard error. Dashed vertical lines mark λ_{min} (blue) and λ_{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
`est_<target>`	Model prediction (e.g., `est_sale_price`)
`residual`	Actual − predicted
`cqa`	Condition-Quality-Appeal score (0–10 scale)
`residual_sf`	Residual / living area (if `living_area` designated)
`cqa_sf`	CQA calculated from ranking via `residual_sf`
`<var>_contribution`	Per-variable contribution to prediction
`basis`	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 ~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 `TMPDIR` 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.

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.

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.