Package 'raterevision'

November 14, 2026

Title Insurance Rate Revision Workflow Tools

Version 0.1.0

Description Provides tools for insurance rate revision workflows around normalized rating factor tables. Functions reshape long-form rating tables into actuary-friendly wide tables, round-trip those tables through Microsoft Excel, apply and review proposed factor changes, rebase and normalize relativities, compare current and proposed rating output, summarize rate impacts, and solve a one-dimensional rate-level adjustment to meet a target aggregate rate change. The package is designed to work naturally with 'ratingtables' while remaining usable with ordinary data frames and arbitrary user-supplied rating code.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports openxlsx2, stats, utils

Suggests knitr, rmarkdown, ratingtables (>= 0.2.0), testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

URL https://github.com/gs-actuary/raterevision

BugReports https://github.com/gs-actuary/raterevision/issues

NeedsCompilation no

Author Greg Sollenberger [aut, cre, cph]

Maintainer Greg Sollenberger <yagreg7@yahoo.com>

Repository CRAN

Date/Publication 2026-11-14 09:20:02 UTC

Contents

compare_rating_outputs . . . 2
diagnose_rating_output . . . 3
extract_rate_tables . . . 4
normalize_factor_set . . . 5
profile_rate_workbook . . . 6
rate_change_distribution . . . 7
rate_plan_diff . . . 7
rate_tables_long . . . 8
rate_tables_wide . . . 9
read_rate_workbook . . . 9
rebase_factor_set . . . 10
replace_factor_set . . . 11
solve_rate_target . . . 11
summarize_rate_change . . . 12
update_factors . . . 13
validate_rate_plan . . . 14
write_rate_workbook . . . 14

Index **16**

`compare_rating_outputs` *Compare current and proposed rating outputs*

Description

Standardizes current/proposed output into one long comparison table. It can consume the wide `indicated_<coverage>` columns produced by `ratingtables`, or already-long output with an explicit premium and coverage column.

Usage

```
compare_rating_outputs(
  current,
  proposed,
  id_cols,
  premium_cols = NULL,
  premium_prefix = "indicated_",
  premium_col = NULL,
  coverage_col = NULL,
  keep_cols = NULL
)
```

Arguments

current	Current rating output.
proposed	Proposed rating output.
id_cols	Columns uniquely identifying a rated record.
premium_cols	Optional wide premium columns. If omitted, common columns beginning with premium_prefix are detected.
premium_prefix	Prefix removed from wide premium columns to obtain the coverage label. Defaults to "indicated_".
premium_col	For long input, the premium column name.
coverage_col	For long input, the coverage/peril column name.
keep_cols	Additional columns from proposed output to carry into the comparison (for example charter or territory).

Value

Long comparison data with current/proposed premium and changes.

---

diagnose_rating_output

Diagnose differences between a reference rater and a candidate rater

---

Description

Uses grouped discrepancy scans and lightweight statistical models to identify rating variables associated with missing or incorrect candidate premiums. The models are diagnostic tools, not production predictive models: no tuning grid or cross-validation is performed, and the default print method reports only high-level findings.

Usage

```
diagnose_rating_output(
  comparison,
  predictors = NULL,
  absolute_tolerance = 0.01,
  relative_tolerance = 1e-06,
  top_n = 5L,
  model_sample = 50000L,
  max_model_levels = 40L,
  interaction_top = 4L,
  seed = 1L
)
```

Arguments

comparison	Output from compare_rating_outputs().
predictors	Optional character vector of columns to investigate. By default all non-ID, non-premium columns are used, including coverage.
absolute_tolerance	Dollar tolerance below which a premium difference is treated as a match.
relative_tolerance	Relative tolerance applied to the absolute reference premium. A row is wrong only when its absolute error exceeds both tolerances.
top_n	Number of variables retained in the high-level suspect table.
model_sample	Maximum number of rows used by the background models.
max_model_levels	Maximum factor levels retained in a background model; rarer levels are collapsed to <OTHER>. Numeric variables with many unique values are quantile-binned for diagnostic modeling.
interaction_top	Number of leading variables considered for pairwise interaction diagnostics. Set to 0 to skip interaction models.
seed	Random seed used only when down-sampling for background models.

Details

The input is normally created by compare_rating_outputs(), with all useful rating variables carried through via its keep_cols argument. In a migration workflow, current_premium should represent the trusted/reference premium and proposed_premium the candidate ratingtables premium.

Value

An object of class raterevision_rater_diagnosis containing summary counts, variable/level diagnostics, optional coverage_summary and coverage_level_details (when coverage is present), model signals, interaction signals, and the fitted lightweight models for advanced inspection.

---

extract_rate_tables *Extract candidate rate tables from a profiled workbook*

Description

Reads the candidate regions selected in a profile_rate_workbook() result and returns a named list of rectangular data frames. The object intentionally uses the same list-oriented interface as rate_tables_wide(), while retaining source-sheet/range metadata so that a migration can be reviewed and refined.

Usage

```
extract_rate_tables(
  profile,
  file = attr(profile, "file"),
  only_included = TRUE,
  min_confidence = NULL
)
```

Arguments

profile	A raterevision_workbook_profile returned by profile_rate_workbook(). The profile may be edited before extraction.
file	Optional workbook path. By default the source path stored on the profile is used.
only_included	Logical; when TRUE, extract only rows where include is TRUE.
min_confidence	Optional additional confidence threshold. Set to NULL to rely only on the editable include column.

Details

This function does not claim that the extracted rectangles are already valid normalized ratingtables factor tables. It is an ingestion step: uncertain candidates should be reviewed, renamed, resized, or excluded in the profile before extraction.

Value

A named list of rectangular data frames with class raterevision_tables. A manifest is stored in the manifest attribute.

---

normalize_factor_set *Normalize the weighted mean of a factor set*

Description

Scales factors so that their weighted mean equals target_mean within each grouping. Equal weights are used by default. If real portfolio exposures are desired, merge or align those weights to the factor rows first and supply them through weights.

Usage

```
normalize_factor_set(
  factor_table,
  term_name,
  weights = NULL,
  target_mean = 1,
  by = NULL
)
```

Arguments

factor_table	Normalized factor table.
term_name	Term to normalize.
weights	NULL for equal weights, a numeric vector aligned to the selected term rows, or the name of a numeric column in factor_table.
target_mean	Desired weighted mean, usually 1.
by	Optional grouping columns. Defaults to metadata, coverage, term name, and variable-name columns.

Value

The full factor table with the selected term normalized.

---

profile_rate_workbook *Profile an Excel workbook for candidate rating tables*

Description

Scans an arbitrary .xlsx workbook and identifies rectangular regions that look like rating tables. The profiler is deliberately heuristic: it uses connected nonblank regions, header structure, numeric density, and nearby labels to make educated guesses. The returned profile is intended to be reviewed and, when necessary, edited before extraction.

Usage

```
profile_rate_workbook(
  file,
  sheets = NULL,
  min_cells = 3L,
  min_rows = 1L,
  min_cols = 2L,
  include_threshold = 0.5,
  title_lookback = 3L
)
```

Arguments

file	Path to an .xlsx, .xlsm, or .xlsb workbook.
sheets	Optional character vector of sheet names to scan. By default all sheets are scanned except sheets beginning with an underscore.
min_cells	Minimum number of nonblank cells required for a candidate.
min_rows	Minimum number of occupied rows required for a candidate.
min_cols	Minimum number of occupied columns required for a candidate.
include_threshold	Confidence threshold used to populate the editable include column. Candidates below the threshold remain in the profile.
title_lookback	Number of rows above a candidate searched for a nearby descriptive title.

Value

A data frame with class `raterevision_workbook_profile`. Each row is a candidate table. The `include`, `table_name`, `extract_range`, and `header_row` fields may be edited before calling `extract_rate_tables()`.

---

rate_change_distribution *Summarize the distribution of rate impacts*

Description

Summarize the distribution of rate impacts

Usage

```
rate_change_distribution(
  comparison,
  breaks = c(-Inf, -0.2, -0.1, -0.05, 0, 0.05, 0.1, 0.2, Inf),
  by = NULL
)
```

Arguments

`comparison`	Output from `compare_rating_outputs()`.
`breaks`	Numeric rate-change breakpoints, including desired infinities.
`by`	Optional additional grouping columns.

Value

Rate-change distribution by bucket and optional groups.

---

rate_plan_diff *Compare current and proposed factor tables*

Description

Produces a row-level audit of added, removed, changed, and unchanged factor keys. `factor_row_id` is ignored by default because it is not a rating key.

Usage

```
rate_plan_diff(
  current,
  proposed,
  key_cols = NULL,
  tolerance = sqrt(.Machine$double.eps),
  include_unchanged = TRUE
)
```

Arguments

current	Current normalized factor table.
proposed	Proposed normalized factor table.
key_cols	Optional key columns. By default all non-value, non-internal columns from either table are used.
tolerance	Numeric tolerance for treating values as unchanged.
include_unchanged	Logical; retain unchanged rows.

Value

A data frame with current/proposed values, changes, and `status`.

---

rate_tables_long *Convert wide rate tables back to normalized long format*

Description

Reconstructs a normalized factor table from the object produced by `rate_tables_wide()`. This is the non-Excel reverse transformation used by `read_rate_workbook()`.

Usage

```
rate_tables_long(
  tables,
  manifest = attr(tables, "manifest"),
  add_factor_row_id = TRUE
)
```

Arguments

tables	A `raterevision_tables` object or a named list of data frames.
manifest	Optional manifest. Required if the `manifest` attribute has been removed.
add_factor_row_id	Logical; add a fresh sequential `factor_row_id`.

Value

A normalized long-format factor table.

rate_tables_wide *Convert a normalized factor table to wide rate tables*

Description

Converts a ratingtables-style normalized factor table into a named list of actuary-friendly data frames. Coverages become columns. One-way and higher interactions remain long by default; selected two-way interactions can be displayed as matrices with one interaction variable down rows and the other across columns. Matrix layout is intentionally prohibited for interactions deeper than two variables.

Usage

```
rate_tables_wide(
  factor_table,
  filters = NULL,
  matrix_terms = character(),
  sort_rows = TRUE
)
```

Arguments

factor_table	A normalized factor table containing coverage, term_name, term_value, and optional variable1/level1, variable2/level2, ... columns.
filters	Optional named list of exact-value filters, for example list(state = "IL", rate_eff_date = as.Date("2026-01-01")).
matrix_terms	Character vector of two-way term_name values to display in matrix form. All other terms use coverage-wide long format.
sort_rows	Logical; sort metadata and level columns using numeric-aware ordering where possible.

Value

A named list of data frames with class raterevision_tables. A machine-readable manifest is stored in the manifest attribute.

read_rate_workbook *Read an Excel rate workbook*

Description

Reads a workbook created by write_rate_workbook(). By default the visible tables are reconstructed into a normalized long factor table suitable for ratingtables or further programmatic revision.

Usage

```
read_rate_workbook(file, return = c("long", "tables"), validate = TRUE)
```

Arguments

file	Input .xlsx path.
return	Either "long" (default) or "tables".
validate	Logical; validate the reconstructed long factor table.

Value

A normalized factor table or raterevision_tables object.

---

rebase_factor_set *Rebase a factor set to a selected rating level*

Description

Divides all factors within each applicable rate-set/coverage group by the factor at a selected reference level. This makes the selected level equal to 1.000 while preserving all relativities. Rebasing is a presentation/basis operation; it is distinct from normalize_factor_set(), which controls a weighted average level.

Usage

```
rebase_factor_set(factor_table, term_name, base_level, by = NULL)
```

Arguments

factor_table	Normalized factor table.
term_name	Term to rebase.
base_level	For a one-way term, a scalar level. For an interaction, a named vector or list mapping variable names to reference levels.
by	Optional grouping columns. The default preserves all metadata, coverage, term name, and variable-name columns.

Value

The full factor table with the selected term rebased.

replace_factor_set *Replace complete factor-table slices*

Description

Replaces complete slices of a current factor table with rows from a replacement table. This differs from `update_factors()`, which changes only explicitly supplied cells. By default a slice is identified by rate-set metadata, coverage, term name, and variable-name columns; level columns are intentionally excluded so omitted levels are removed.

Usage

```
replace_factor_set(current, replacement, term_names = NULL, slice_cols = NULL)
```

Arguments

`current`	Current normalized factor table.
`replacement`	Replacement rows.
`term_names`	Optional terms to take from `replacement`.
`slice_cols`	Optional columns defining a complete factor-set slice.

Value

A normalized factor table.

solve_rate_target *Solve a target aggregate rate change*

Description

Finds a single numeric parameter that makes an arbitrary proposed rating workflow achieve a desired aggregate rate change from a supplied current premium. The solver does not know or care whether the proposed workflow uses the same specification, a structurally different specification, driver averaging, custom R code, or territory-specific base rates. The user defines the one-dimensional adjustment inside `rate_function`.

Usage

```
solve_rate_target(
  current_premium,
  target,
  rate_function,
  interval = c(0.5, 1.5),
  premium_extractor = NULL,
  tol = 1e-08,
```

```
  maxiter = 100L,
  extend_interval = c("no", "yes", "upX", "downX"),
  parameter_name = "multiplier"
)
```

Arguments

current_premium	Current aggregate premium, a positive finite scalar.
target	Desired aggregate rate change as a decimal, e.g. 0.06.
rate_function	Function of one numeric parameter. It may return a premium scalar/vector directly, or any object handled by premium_extractor.
interval	Initial two-value search interval for the parameter.
premium_extractor	Optional function that converts the result of rate_function to a single aggregate premium.
tol	Root-finding tolerance.
maxiter	Maximum root-finding iterations.
extend_interval	Passed to stats::uniroot(extendInt = ...); one of "no", "yes", "upX", or "downX".
parameter_name	Label used in printed output.

Details

A common territory-base use is to let x multiply every territory-specific base rate, preserving the proposed territorial shape while solving one overall rate-level parameter. If several base rates are allowed to move independently against one aggregate target, the problem is underdetermined and should not be passed to this scalar solver. Independent targets (for example one target per coverage) can be solved by calling this function once per independently adjustable parameter.

Value

A raterevision_target_solution object.

---

summarize_rate_change *Summarize aggregate rate change*

Description

Summarize aggregate rate change

Usage

```
summarize_rate_change(comparison, by = "coverage", exposure_col = NULL)
```

Arguments

comparison	Output from compare_rating_outputs().
by	Grouping columns, typically coverage, charter, or both. Use NULL for one overall row.
exposure_col	Optional exposure column to sum.

Value

Aggregate current/proposed premium and rate change by group.

---

update_factors *Apply sparse factor updates*

Description

Updates only the factor keys supplied in updates. New keys are rejected by default so that a typo cannot silently create a rating level.

Usage

```
update_factors(current, updates, key_cols = NULL, allow_new_levels = FALSE)
```

Arguments

current	Current normalized factor table.
updates	Data frame containing key columns and term_value.
key_cols	Optional matching columns. By default all update columns other than term_value and factor_row_id are used.
allow_new_levels	Logical; allow unmatched update rows to be appended.

Value

Updated normalized factor table.

validate_rate_plan *Validate a normalized rate factor table*

Description

Performs revision-oriented checks without duplicating the full rating-spec validation in `ratingtables`. It checks required columns, factor values, duplicate keys, variable/level slot consistency, and optionally positivity for known multiplicative terms.

Usage

```
validate_rate_plan(factor_table, multiplicative_terms = NULL, error = TRUE)
```

Arguments

`factor_table`	Normalized factor table.
`multiplicative_terms`	Optional character vector of terms whose values must be strictly positive.
`error`	Logical; stop if any errors are found.

Value

A `raterevision_validation` data frame. If `error = TRUE`, validation errors are also raised as an error.

write_rate_workbook *Write rate tables to an Excel workbook*

Description

Writes normalized factors or a `rate_tables_wide()` list to an `.xlsx` workbook. The workbook contains one visible sheet per rate table, an optional instruction sheet, and a very-hidden `_raterevision` manifest used for a deterministic round trip back into R.

Usage

```
write_rate_workbook(
  x,
  file,
  filters = NULL,
  matrix_terms = character(),
  overwrite = TRUE,
  include_readme = TRUE
)
```

Arguments

x	A normalized factor table or raterevision_tables object.
file	Output .xlsx path.
filters	Optional filters passed to rate_tables_wide() when x is a long factor table.
matrix_terms	Two-way interaction terms to display as matrices.
overwrite	Logical; overwrite an existing file.
include_readme	Logical; include a visible _README sheet.

Value

The file path, invisibly.

Index

compare_rating_outputs, 2
compare_rating_outputs(), *4*, *7*, *13*

diagnose_rating_output, 3

extract_rate_tables, 4
extract_rate_tables(), *7*

normalize_factor_set, 5
normalize_factor_set(), *10*

profile_rate_workbook, 6
profile_rate_workbook(), *4, 5*

rate_change_distribution, 7
rate_plan_diff, 7
rate_tables_long, 8
rate_tables_wide, 9
rate_tables_wide(), *4*, *8*, *14, 15*
read_rate_workbook, 9
read_rate_workbook(), *8*
rebase_factor_set, 10
replace_factor_set, 11

solve_rate_target, 11
summarize_rate_change, 12

update_factors, 13
update_factors(), *11*

validate_rate_plan, 14

write_rate_workbook, 14
write_rate_workbook(), *9*