Package: SimtablR (via r-universe)

November 21, 2026

Title Easy Publication-Ready Tables and Regression Analysis

Version 3.1.1

Description Streamlines the creation of descriptive frequency tables ('Table 1'), diagnostic test accuracy evaluations (sensitivity, specificity, predictive values), and multi-outcome regression summaries. Features a grammar for publication-ready epidemiological tables built on design-aware effect measures, prevalence and odds ratio calculations, and seamless integration with 'flextable' for exporting results to 'Microsoft Word' and 'PowerPoint'.

License MIT + file LICENSE

Encoding UTF-8

Imports cli, digest, generics, rlang, sandwich, stats, tidyselect, utils

Suggests dplyr, flextable, ggplot2, gt, haven, knitr, openxlsx, labelled, logistf, officer, pROC, rmarkdown, survival, EValue, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Depends R (>= 4.1.0)

LazyData true

URL https://github.com/MatheusTG-14/SimtablR, https://MatheusTG-14.github.io/SimtablR/

BugReports https://github.com/MatheusTG-14/SimtablR/issues

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

NeedsCompilation no

Author Matheus Trabuco Gonzalez [aut, cre]

Maintainer Matheus Trabuco Gonzalez <matheustrabucogonzalez@gmail.com>

Repository https://cranhaven.r-universe.dev

Date/Publication 2026-11-21 02:02:00 UTC

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/SimtablR

RemoteSha 6346655cc5fbf774789ef8e0206051cfbc0bc3eb

RemoteSubdir SimtablR

Contents

add_effect . . . 3
adjust . . . 4
advise . . . 5
as.data.frame.simtab_audit . . . 5
as.data.frame.simtab_rbind_tb . . . 6
as.data.frame.simtab_report . . . 7
as.data.frame.simtab_result . . . 7
as_flextable.simtab_codebook . . . 8
as_flextable.simtab_rbind_tb . . . 9
as_flextable.simtab_result . . . 10
as_gt . . . 10
as_methods . . . 11
autoplot.simtab_result . . . 12
autoplot.simtab_spec . . . 13
codebook . . . 13
describe . . . 14
design . . . 15
diag_test . . . 15
e_value . . . 18
engine . . . 19
epitabl . . . 19
evaluate . . . 21
export_docx . . . 22
export_plot . . . 23
export_pptx . . . 24
export_regtab_csv . . . 25
export_regtab_xlsx . . . 26
export_xlsx . . . 26
flow . . . 27
fmt . . . 28
glance.simtab_result . . . 29
is_simtab . . . 29
journal_style . . . 30
label . . . 32
list_engines . . . 32
list_journals . . . 33
measure . . . 33
missingness . . . 34

model-accessors . . . 35
model_info . . . 36
overall . . . 37
plot.simtab_diag . . . 38
print.simtab_rbind_tb . . . 38
print.simtab_result . . . 39
rbind.simtab_tb . . . 40
register_engine . . . 41
register_journal . . . 42
regtab . . . 43
repro_manifest . . . 45
roc . . . 46
sensitivity . . . 48
set_design . . . 49
set_summary . . . 50
simtab . . . 50
simtab_completions . . . 51
simtab_errors . . . 52
simtablr . . . 53
simtablr_guidance . . . 54
simtablr_references . . . 55
stard . . . 58
stratify . . . 59
strobe . . . 59
style . . . 60
survtab . . . 60
table1 . . . 62
tb . . . 66
test . . . 70
tidy.simtab_result . . . 71
validate . . . 71
why . . . 72

Index **73**

`add_effect` *Add or replace the effect-measure column(s) of a table1*

Description

Recomputes `tab` with a crude (and optionally adjusted) effect-measure column. Equivalent to passing `measure=`/`adjust=` to `table1()` directly; calling it when an effect column already exists replaces it.

Usage

```
add_effect(tab, measure, adjust.for = NULL, ref = NULL, conf.level = NULL, ...)
```

Arguments

tab	A table1 object.
measure	One of "OR", "PR", "RR".
adjust.for	Optional covariate names for an adjusted column.
ref	Optional reference level(s); see table1().
conf.level	Optional confidence level; defaults to the table's.
...	Ignored.

Value

The recomputed table1 object.

See Also

table1(), test()

Examples

```
data(epitabl)
table1(epitabl, c("sex", "smoking"), by = "adjudicated_acs") |> add_effect("OR")
```

---

adjust *Capture adjustment covariates*

Description

Capture adjustment covariates

Usage

```
adjust(spec, ...)
```

Arguments

spec	A simtab_spec.
...	Tidyselect expressions for adjustment covariates.

Value

A modified simtab_spec.

Examples

```
adjust(simtab(epitabl), age, sex)
```

advise *Run educator advice rules*

Description

Evaluates all applicable rules against a computed result. Rule failures are caught and skipped so advice can never break analysis.

Usage

```
advise(result, audit = FALSE)
```

Arguments

result	A computed simtab_result or simtab_report.
audit	Logical. FALSE (default) returns the fired advice entries. TRUE runs every applicable rule regardless of the guidance dial and returns a simtab_audit report of checked, fired, and silent rules (single results only).

Value

A list of advice entries deduplicated by rule id, or a simtab_audit object when audit = TRUE.

See Also

simtablr_references for the full references behind the citations shown in advice, and simtablr_guidance() to control display.

Examples

```
res <- tb(epitabl, sex, diabetes)
advise(res)
advise(res, audit = TRUE)
```

as.data.frame.simtab_audit *Coerce a SimtablR audit to a data.frame*

Description

Returns the checked-rules table ($checked) with a fired logical column joined on, so the fired/silent distinction shown by print() is available as a single rectangular table.

Usage

```
## S3 method for class 'simtab_audit'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	A simtab_audit.
row.names, optional, ...	Ignored; accepted for S3 consistency.

Value

A data.frame with one row per checked rule.

Examples

```
res <- tb(epitabl, sex, diabetes)
as.data.frame(advise(res, audit = TRUE))
```

---

as.data.frame.simtab_rbind_tb

Convert rbind_tb to Data Frame

Description

Convert rbind_tb to Data Frame

Usage

```
## S3 method for class 'simtab_rbind_tb'
as.data.frame(x, row.names = NULL, optional = FALSE, tidy = FALSE, ...)
```

Arguments

x	An rbind_tb object.
row.names	NULL or a character vector giving the row names for the data frame.
optional	Logical. If TRUE, setting row names and converting column names is optional.
tidy	Logical. If TRUE, returns a long-format tidy data frame.
...	Additional arguments.

Value

A data.frame.

Examples

```
t1 <- tb(epitabl, sex, diabetes)
t2 <- tb(epitabl, hypertension, diabetes)
as.data.frame(rbind(t1, t2))
```

as.data.frame.simtab_report

Coerce a SimtablR report to a data.frame

Description

A `simtab_report` bundles items that can have different shapes (for example `simtablr()`'s descriptive Table 1 and association Table 2), so there is no single natural rectangular table for a report in general. This default method instead returns a named list of `as.data.frame()` results, one per item, mirroring `as_gt.simtab_report()` and `as_flextable.simtab_report()`. Subclasses whose items always share one shape, such as `sensitivity()`, provide their own rectangular `as.data.frame()` method instead of using this default.

Usage

```
## S3 method for class 'simtab_report'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A `simtab_report`.
`row.names, optional, ...`	Passed to each item's `as.data.frame()`.

Value

A named list of data.frames, one per report item.

Examples

```
rep <- simtablr(epitabl, outcome = mace_event, exposure = renal_impairment,
                vars = c(age, sex), design = "cohort")
as.data.frame(rep)
```

as.data.frame.simtab_result

Convert a SimtablR result or spec to a data frame

Description

Convert a SimtablR result or spec to a data frame

Usage

```
## S3 method for class 'simtab_result'
as.data.frame(x, row.names = NULL, optional = FALSE, tidy = FALSE, ...)

## S3 method for class 'simtab_spec'
as.data.frame(x, row.names = NULL, optional = FALSE, tidy = FALSE, ...)
```

Arguments

x	A simtab_result or simtab_spec.
row.names	Unused.
optional	Unused.
tidy	Logical. FALSE returns the formatted display frame; TRUE returns the long numeric table.
...	Passed to subclass renderers. For regtab, vif = TRUE adds maximum VIF-equivalent columns by outcome.

Value

A data.frame.

Examples

```
res <- tb(epitabl, sex, diabetes)
as.data.frame(res)
as.data.frame(res, tidy = TRUE)
```

---

as_flextable.simtab_codebook

Convert a SimtablR codebook to a flextable

Description

Convert a SimtablR codebook to a flextable

Usage

```
## S3 method for class 'simtab_codebook'
as_flextable(x, ...)
```

Arguments

x	A simtab_codebook object.
...	Additional arguments passed to flextable::flextable().

Value

A styled `flextable` object.

Examples

```
if (requireNamespace("flextable", quietly = TRUE)) {
  cb <- codebook(epitabl[, c("age", "sex", "diabetes")])
  flextable::as_flextable(cb)
}
```

---

as_flextable.simtab_rbind_tb

Convert simtab_rbind_tb Object to Flextable

Description

Convert simtab_rbind_tb Object to Flextable

Usage

```
as_flextable.simtab_rbind_tb(x, footnotes = NULL, ...)
```

Arguments

`x`	A `simtab_rbind_tb` object.
`footnotes`	Optional character vector of footer lines appended to the table.
`...`	Additional arguments passed to `flextable::flextable()`.

Value

A `flextable` object.

Examples

```
if (requireNamespace("flextable", quietly = TRUE)) {
  t1 <- tb(epitabl, sex, diabetes)
  t2 <- tb(epitabl, hypertension, diabetes)
  as_flextable.simtab_rbind_tb(rbind(t1, t2))
}
```

as_flextable.simtab_result *Convert a SimtablR result or spec to a flextable*

Description

Convert a SimtablR result or spec to a flextable

Usage

```
as_flextable.simtab_result(x, ...)

as_flextable.simtab_spec(x, ...)

as_flextable.simtab_report(x, ...)
```

Arguments

x	A simtab_result or simtab_spec.
...	Passed to the subclass flextable renderer.

Value

A flextable object, or a named list of flextable objects for a simtab_report.

Examples

```
if (requireNamespace("flextable", quietly = TRUE)) {
  res <- tb(epitabl, sex, diabetes)
  flextable::as_flextable(res)
}
```

as_gt *Convert a SimtablR object to a gt table*

Description

as_gt() is a soft-dependency renderer for HTML and Quarto workflows. It auto-computes a simtab_spec, then dispatches by result subclass. Built-in results with specialised renderers retain their display features; every other result with an as.data.frame renderer is converted from its display data frame. Extension engines can register a custom as_gt renderer.

Usage

```
as_gt(x, ...)

## S3 method for class 'simtab_spec'
as_gt(x, ...)

## S3 method for class 'simtab_result'
as_gt(x, ...)

## S3 method for class 'simtab_rbind_tb'
as_gt(x, ...)

## S3 method for class 'simtab_report'
as_gt(x, ...)
```

Arguments

x	A SimtablR result or spec.
...	Passed to `gt::gt()`.

Value

A `gt_tbl` object, or a named list of `gt_tbl` objects for a `simtab_report`.

Examples

```
if (requireNamespace("gt", quietly = TRUE)) {
  res <- tb(epitabl, sex, diabetes)
  as_gt(res)
}
```

---

as_methods *Generate a manuscript methods sentence*

Description

Builds concise manuscript prose from the decisions recorded in a computed SimtablR result. Passing a bare `simtab_spec` computes it first, matching the other render/extract verbs.

Usage

```
as_methods(x, ...)
```

Arguments

x	A `simtab_result` or `simtab_spec`.
...	Ignored.

Value

A character string of length one.

Examples

```
res <- tb(epitabl, sex, diabetes)
as_methods(res)
```

`autoplot.simtab_result`

Autoplot a computed SimtablR result

Description

Dispatches to the engine's registered `autoplot` renderer.

Usage

```
## S3 method for class 'simtab_result'
autoplot(object, ...)
```

Arguments

`object`	A `simtab_result`.
`...`	Passed to the engine's autoplot renderer.

Value

A `ggplot` object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  r <- roc(epitabl, poc_hstn_value, adjudicated_acs)
  ggplot2::autoplot(r)
}
```

autoplot.simtab_spec *Plot a computed SimtablR spec*

Description

Plot a computed SimtablR spec

Usage

```
## S3 method for class 'simtab_spec'
autoplot(object, ...)
```

Arguments

object	A simtab_spec.
...	Passed to the computed result's autoplot method.

Value

A ggplot object when the computed result supports autoplot.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  sp <- regtab(epitabl, "adjudicated_acs", ~ age + sex, family = binomial())$spec
  ggplot2::autoplot(sp)
}
```

codebook *Build a SimtablR codebook*

Description

Build a SimtablR codebook

Usage

```
codebook(x, ...)
```

Arguments

x	A data frame, simtab_spec, simtab_result, or simtab_report.
...	Ignored.

Details

For reports, a publication label recorded consistently by one or more items is used. If report items record conflicting labels for the same variable, `codebook()` warns and uses the neutral label stored on the source column rather than choosing an arbitrary item by position.

Value

A `simtab_codebook` data frame with one row per variable.

Examples

```
cb <- codebook(epitabl[, c("age", "sex", "diabetes")])
head(cb)
```

---

describe *Capture variables to describe*

Description

Capture variables to describe

Usage

```
describe(spec, ...)
```

Arguments

`spec`	A `simtab_spec`.
`...`	Tidyselect expressions for variables to describe.

Value

A modified `simtab_spec`.

Examples

```
describe(simtab(epitabl), age, sex)
```

design *Read a recorded study design*

Description

Returns the study design recorded by `set_design()`, without requiring callers to read the internal `"simtablr.design"` attribute directly. On a data frame this reads the advisory attribute; on a `simtab_spec` or `simtab_result` it reads the specification's `design` field, falling back to the source data frame's attribute for a `simtab_result` (matching what `repro_manifest()` reports as `design`/`design_source`). It does not indicate whether the design took part in effect-measure resolution; see `resolved_from` on `spec$effect` or `design_used` in `repro_manifest()` for that.

Usage

```
design(x)
```

Arguments

`x`	A data.frame, `simtab_spec`, or `simtab_result`.

Value

A single design string, or `NULL` if none is recorded.

Examples

```
cohort <- set_design(epitabl, "cohort")
design(cohort)
```

diag_test *Assess the accuracy of a binary diagnostic test*

Description

`diag_test()` compares a binary index test with a binary reference standard and reports the confusion matrix with sensitivity, specificity, predictive values, likelihood ratios, and related metrics, each with a confidence interval. Positive levels are detected automatically but can be set with `positive` and `test_positive`. For a continuous marker use `roc()`.

Usage

```
diag_test(
  data,
  test,
  ref,
  positive = NULL,
  test_positive = NULL,
  conf.level = 0.95,
  ci = c("exact", "wilson"),
  d = 2,
  percent = TRUE
)
```

Arguments

data	A data frame.
test	The index test column: a bare name or string. Must have two levels.
ref	The reference standard column: a bare name or string. Must have two levels.
positive	The level of ref that means "disease present". If NULL, common labels such as "Yes", "1", or "Positive" are detected, falling back to the last level.
test_positive	The level of test that means "test positive". If NULL, positive is reused when test has that level; otherwise it is detected the same way.
conf.level	Number between 0 and 1. Confidence level for intervals.
ci	String. Interval method for the proportions: "exact" (Clopper-Pearson) or "wilson".
d	Integer. Decimal places for all displayed estimates and intervals.
percent	Logical. If TRUE, show sensitivity, specificity, predictive values, accuracy, and prevalence as percentages. Ratios and indices are always shown as decimals.

Details

Statistical methods:

Sensitivity, specificity, PPV, NPV, accuracy, and prevalence are proportions from the confusion matrix, with Clopper-Pearson intervals by default or Wilson intervals with ci = "wilson". Predictive values and accuracy depend on the prevalence in data. Likelihood ratios use the log-method interval for a ratio of proportions (Simel et al., 1991; Altman et al., 2000). The diagnostic odds ratio uses the Woolf logit interval, with standard error $\sqrt{1/TP + 1/FP + 1/FN + 1/TN}$ (Glas et al., 2003). No continuity correction is applied, so both intervals are NA when any cell of the matrix is zero. Cohen's kappa measures chance-corrected agreement between test and reference, with the Fleiss, Cohen & Everitt (1969) standard error. The Youden index and F1 score are reported without intervals.

Missing data:

Rows missing either the test or the reference are dropped, and a message reports how many.

Modifying the result:

Use `fmt()` to change decimals or percentages, `plot()` for a fourfold display, and `ggplot2::autoplot()` with `type = "matrix"` or `type = "metrics"` for a heatmap or a plot of metrics with intervals. `as.data.frame(x, tidy = TRUE)` returns the unrounded metrics.

Value

A `simtab_result` of class `simtab_diag`. Print it to see the confusion matrix and metrics, convert it with `as.data.frame()`, or save it with `export_docx()`, `export_pptx()`, or `export_xlsx()`. Unrounded results are stored in `$data`.

References

Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4), 404–413. doi:10.1093/biomet/26.4.404.

Simel, D. L., Samsa, G. P., & Matchar, D. B. (1991). Likelihood ratios with confidence: sample size estimation for diagnostic test studies. *Journal of Clinical Epidemiology*, 44(8), 763–770. doi:10.1016/08954356(91)90128V.

Altman, D. G., Machin, D., Bryant, T. N., & Gardner, M. J. (2000). *Statistics with Confidence* (2nd ed.). BMJ Books.

Glas, A. S., Lijmer, J. G., Prins, M. H., Bonsel, G. J., & Bossuyt, P. M. M. (2003). The diagnostic odds ratio: a single indicator of test performance. *Journal of Clinical Epidemiology*, 56(11), 1129–1135. doi:10.1016/S08954356(03)00177X.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46. doi:10.1177/001316446002000104.

Fleiss, J. L., Cohen, J., & Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5), 323–327. doi:10.1037/h0028106.

See Also

`roc()` for continuous markers, `plot.simtab_diag()`, and simtablr_references for all references cited by SimtablR.

Examples

```
# Point-of-care troponin against adjudicated ACS
substudy <- subset(epitabl, diagnostic_substudy == "Yes")
acc <- diag_test(substudy, test = poc_hstn_positive, ref = adjudicated_acs)
acc

# Wilson intervals, decimals instead of percentages
diag_test(
  substudy, poc_hstn_positive, adjudicated_acs,
  positive = "Yes", test_positive = "Positive",
  ci = "wilson", percent = FALSE
)

# Metrics as a data frame
as.data.frame(acc)
```

`e_value` *Compute E-values for SimtablR ratio estimates*

Description

Computes VanderWeele-Ding E-values for ratio-scale estimates and confidence limits. Risk ratios are used directly. For common outcomes, odds ratios use the square-root approximation and hazard ratios use the VanderWeele-Ding conversion `(1 - 0.5^sqrt(HR)) / (1 - 0.5^sqrt(1/HR))`; with `rare = TRUE` the supplied ratio is treated as a rare-outcome risk-ratio approximation.

Usage

```
e_value(x, ...)

## S3 method for class 'simtab_result'
e_value(x, measure = NULL, rare = FALSE, ...)
```

Arguments

`x`	A computed SimtablR result containing ratio estimates.
`...`	Reserved for future options.
`measure`	Optional ratio measure override: `"RR"`, `"OR"`, or `"HR"`.
`rare`	Logical. Treat OR/HR estimates as rare-outcome approximations to risk ratios instead of applying the square-root approximation.

Details

E-values require positive, finite ratio estimates and confidence limits from a supported SimtablR result; unsupported scales fail with a classed input condition. Missing or non-finite source estimates are not converted into evidence. An E-value is a sensitivity-analysis threshold, not proof that uncontrolled confounding is absent, and must be interpreted with the identification assumptions of the parent analysis.

Value

A `simtab_e_value` result with raw E-value numerics.

References

VanderWeele, T. J., & Ding, P. (2017). Sensitivity analysis in observational research: introducing the E-value. *Annals of Internal Medicine*, 167(4), 268–274. doi:10.7326/M162607.

Examples

```
ratio_result <- tb(epitabl, diabetes, adjudicated_acs, or, ref = "No")
e_value(ratio_result)
```

engine *Select a registered computation engine*

Description

Selects the registered engine used when a specification is computed. Because the engine determines the numerical evidence, applying `engine()` to a computed result updates its stored specification and recomputes through the same path as other evidence-changing verbs.

Usage

```
engine(spec, name)
```

Arguments

`spec`	A `simtab_spec` or `simtab_result`.
`name`	A single registered engine name. See `list_engines()`.

Value

A modified `simtab_spec`, or a recomputed `simtab_result`.

See Also

`register_engine()`, `evaluate()`

Examples

```
engine(simtab(epitabl), "descriptive")
```

epitabl *ESTROBE-ACS Teaching Cohort*

Description

A fully synthetic prospective cohort of 1,500 adults enrolled after an emergency-department presentation with suspected acute coronary syndrome (ACS). The data were constructed to demonstrate descriptive, design-aware, diagnostic, regression, survival, sensitivity, and reporting workflows in SimtablR. They do not contain real patient records.

Usage

```
epitabl
```

Format

A data frame with 1,500 rows and 22 variables:

participant_id Unique study participant identifier (`ACS0001` to `ACS1500`).

age Age at the index presentation, in years.

sex Sex recorded for clinical assessment (`Female`, `Male`).

bmi Body mass index in kg/m2; contains clinical missing values.

smoking Smoking status (`Never`, `Former`, `Current`).

hypertension History of hypertension (`No`, `Yes`).

diabetes History of diabetes (`No`, `Yes`).

dyslipidemia History of dyslipidemia (`No`, `Yes`).

cholesterol Total cholesterol in mg/dL; contains clinical missing values.

presentation_hours Hours from symptom onset to emergency department presentation.

systolic_bp Systolic blood pressure at presentation, in mmHg.

renal_impairment Renal impairment (eGFR below 60 mL/min/1.73 m2; `No`, `Yes`).

dialysis Maintenance dialysis (`No`, `Yes`); rare clinical exposure.

adjudicated_acs Independent reference diagnosis of acute coronary syndrome (`No`, `Yes`).

diagnostic_substudy Enrolled in point-of-care diagnostic substudy (`No`, `Yes`).

poc_hstn_value Point-of-care high-sensitivity troponin in ng/L.

poc_hstn_positive Binary point-of-care troponin result (`Negative`, `Positive`).

length_of_stay Index hospital length of stay, in days.

ed_visits Emergency department revisits during 1-year follow-up.

rehospitalized Hospital readmission during 1-year follow-up (`No`, `Yes`).

mace_time_days Time to MACE or censoring after presentation, in days.

mace_event Observed major adverse cardiovascular event before censoring (`No`, `Yes`).

Details

ESTROBE-ACS enrolled 1,500 adult participants presenting with chest pain or an ischemic equivalent and undergoing evaluation for suspected ACS.

The primary reference diagnosis is independent clinical adjudication (`adjudicated_acs`). A point-of-care high-sensitivity troponin substudy (`diagnostic_substudy`) evaluated continuous troponin concentrations (`poc_hstn_value`) and binary test positivity (`poc_hstn_positive`).

In-hospital course and secondary outcomes are captured by hospital length of stay (`length_of_stay`), 1-year emergency department revisits (`ed_visits`), and 1-year hospital readmission (`rehospitalized`).

Survival follow-up covers all 1,500 participants up to 365 days from presentation. Major adverse cardiovascular event (MACE) status is recorded by `mace_event` with follow-up time in `mace_time_days`.

Source

Synthetic data generated by `data-raw/generate-epitabl.R`.

Examples

```
data(epitabl)

table1(
  epitabl,
  c(age, sex, smoking, renal_impairment),
  by = adjudicated_acs
)

diagnostic <- subset(epitabl, diagnostic_substudy == "Yes")
diag_test(
  diagnostic,
  test = poc_hstn_positive,
  ref = adjudicated_acs,
  positive = "Yes"
)
```

`evaluate` *Evaluate a SimtablR analysis specification*

Description

Turns an inert `simtab_spec` into a `simtab_result` by running the shared, engine-agnostic validator, resolving one registered engine, running that engine's own hard-error validator (if any), and wrapping the engine's raw numeric output in the namespaced result subclass the engine registered. Passing an existing `simtab_result` recomputes from its stored specification.

Usage

```
evaluate(spec)
```

Arguments

`spec`	A `simtab_spec`, or a `simtab_result` to recompute.

Value

A `simtab_result`.

Examples

```
sp <- simtab(epitabl) |> describe(hypertension) |> stratify(adjudicated_acs)
res <- evaluate(sp)
res
```

`export_docx` *Export a SimtablR table to a Word (.docx) file*

Description

Export a SimtablR table to a Word (.docx) file

Usage

```
export_docx(x, path, footnotes = NULL, methods = FALSE, overwrite = FALSE, ...)
```

Arguments

`x`	A SimtablR object with an `as_flextable()` method (e.g. from `table1()` or `tb()`).
`path`	Output file path. A missing `.docx` suffix is added; any other suffix is rejected.
`footnotes`	Optional character vector passed to table renderers that support footnotes.
`methods`	Logical; when `TRUE`, append recorded `as_methods()` prose after the exported table(s).
`overwrite`	Logical. Existing files are protected by default; pass `TRUE` to replace the destination explicitly.
`...`	Passed to `flextable::save_as_docx()` (e.g. page properties).

Details

Exports are written to a temporary file in the destination directory and published only after the backend succeeds. Existing files are never changed unless `overwrite = TRUE`.

Value

Invisibly returns the normalized output path.

See Also

`export_pptx()`, `export_xlsx()`, `table1()`

Examples

```
## Not run:
data(epitabl)
table1(epitabl, c("age", "sex"), by = "adjudicated_acs") |>
  export_docx(tempfile(fileext = ".docx"))

## End(Not run)
```

export_plot *Export a SimtablR plot to an image file*

Description

Writes a plot through `ggplot2::ggsave()`. When `width` and `height` are not supplied, the plot's own recommended dimensions are used; an explicit value always overrides the recommendation.

Usage

```
export_plot(
  x,
  path,
  width = NULL,
  height = NULL,
  dpi = 300,
  overwrite = FALSE,
  ...
)
```

Arguments

`x`	A `ggplot` object, or a SimtablR object with an `autoplot()` method.
`path`	Output file path; the extension selects the device. When absent, `.png` is added. Supported suffixes are `.png`, `.pdf`, `.svg`, `.jpeg`, `.jpg`, `.tiff`, `.tif`, `.bmp`, `.eps`, `.ps`, `.tex`, `.wmf`, and `.emf`; device availability still depends on the platform.
`width, height`	Canvas size in inches. Default `NULL`, meaning the plot's recommended dimensions.
`dpi`	Resolution for raster devices. Default `300`.
`overwrite`	Logical. Existing files are protected by default; pass `TRUE` to replace the destination explicitly.
`...`	Passed to `ggplot2::ggsave()`.

Value

Invisibly returns the normalized output path.

See Also

`export_docx()`, `export_pptx()`

Examples

```
## Not run:
data(epitabl)
roc(epitabl, poc_hstn_value, adjudicated_acs) |>
  export_plot(tempfile(fileext = ".png"))

## End(Not run)
```

`export_pptx` *Export a SimtablR table to a PowerPoint (.pptx) file*

Description

Export a SimtablR table to a PowerPoint (.pptx) file

Usage

```
export_pptx(x, path, font_size = 14, overwrite = FALSE, ...)
```

Arguments

`x`	A SimtablR object with an `as_flextable()` method.
`path`	Output file path. A missing `.pptx` suffix is added; any other suffix is rejected.
`font_size`	Font size applied before export. Default `14`.
`overwrite`	Logical. Existing files are protected by default; pass `TRUE` to replace the destination explicitly.
`...`	Passed to `flextable::save_as_pptx()`.

Value

Invisibly returns the normalized output path.

See Also

`export_docx()`, `export_xlsx()`

Examples

```
## Not run:
data(epitabl)
table1(epitabl, c("age", "sex"), by = "adjudicated_acs") |>
  export_pptx(tempfile(fileext = ".pptx"))

## End(Not run)
```

`export_regtab_csv` *Export regtab results to CSV*

Description

Export regtab results to CSV

Usage

```
export_regtab_csv(x, file, overwrite = FALSE, ...)
```

Arguments

`x`	A `regtab` result.
`file`	Output file path. A missing `.csv` suffix is added.
`overwrite`	Logical. Existing files are protected by default; pass `TRUE` to replace the destination explicitly.
`...`	Passed to `utils::write.csv()`.

Details

The file is completed in the destination directory before it is published. Backend failures remove partial output and preserve any existing destination.

Value

Invisibly returns `x`.

Examples

```
## Not run:
mod <- regtab(epitabl, outcomes = "rehospitalized", predictors = ~ age + sex)
export_regtab_csv(mod, tempfile(fileext = ".csv"))

## End(Not run)
```

export_regtab_xlsx *Export regtab results to Excel*

Description

Export regtab results to Excel

Usage

```
export_regtab_xlsx(x, file, overwrite = FALSE, ...)
```

Arguments

x	A regtab result.
file	Output file path. A missing .xlsx suffix is added.
overwrite	Logical. Existing files are protected by default; pass TRUE to replace the destination explicitly.
...	Passed to export_xlsx().

Value

Invisibly returns x.

Examples

```
## Not run:
mod <- regtab(epitabl, outcomes = "rehospitalized", predictors = ~ age + sex)
export_regtab_xlsx(mod, tempfile(fileext = ".xlsx"))

## End(Not run)
```

export_xlsx *Export a SimtablR table to an Excel (.xlsx) file*

Description

Writes three worksheets: a long machine-readable tidy sheet with numeric statistic cells, the formatted display sheet, and a data dictionary, named machine-readable, display, and data-dictionary. Reports write one display sheet per item instead of a single display sheet.

Usage

```
export_xlsx(x, path, overwrite = FALSE, ...)
```

Arguments

x	A SimtablR object with an as.data.frame() method.
path	Output file path. A missing .xlsx suffix is added; any other suffix is rejected.
overwrite	Logical. Existing files are protected by default; pass TRUE to replace the destination explicitly.
...	Passed to openxlsx::writeData().

Value

Invisibly returns the normalized output path.

See Also

export_docx(), export_pptx()

Examples

```
## Not run:
data(epitabl)
table1(epitabl, c("age", "sex"), by = "adjudicated_acs") |>
  export_xlsx(tempfile(fileext = ".xlsx"))

## End(Not run)
```

---

flow *Build a participant flow table from recorded facts*

Description

flow() assembles source, subset, rendered-cohort, complete-case, and model Ns from a result or report without fitting or recomputing anything.

Usage

```
flow(x, ...)

## S3 method for class 'simtab_flow'
as_flextable(x, ...)

## S3 method for class 'simtab_flow'
autoplot(object, ...)
```

Arguments

x	A simtab_flow object.
...	Ignored.
object	A simtab_flow object.

Value

A simtab_flow object.

Examples

```
res <- tb(epitabl, sex, diabetes)
flow(res)
```

---

fmt *Set render formatting options*

Description

Set render formatting options

Usage

```
fmt(
  spec,
  d = NULL,
  conf_pct = NULL,
  labels = NULL,
  percent = NULL,
  big_mark = NULL,
  decimal_mark = NULL
)
```

Arguments

spec	A simtab_spec.
d	Decimal places for percentages/estimates, or NULL to leave unchanged.
conf_pct	Confidence-interval percentage label, or NULL to leave unchanged.
labels	Optional named character labels, merged by variable name through label().
percent	Logical, or NULL to leave unchanged. When TRUE, diagnostic and ROC proportion metrics render as percentages. Other engines ignore this field.
big_mark	Character inserted between every three digits of the integer part of rendered numbers (e.g. "," yields 4,391.2), or NULL to leave unchanged. Default "" (no grouping). Currently honoured by tb().
decimal_mark	Character used as the radix point in rendered numbers (e.g. "," for many European locales), or NULL to leave unchanged. Default ".". Must differ from big_mark. Currently honoured by tb().

Value

A modified simtab_spec.

Examples

```
fmt(simtab(epitabl), d = 1, conf_pct = 95)
fmt(simtab(epitabl), big_mark = ",", decimal_mark = ".")
```

---

glance.simtab_result *Glance a SimtablR result*

Description

Glance a SimtablR result

Usage

```
## S3 method for class 'simtab_result'
glance(x, ...)

## S3 method for class 'simtab_spec'
glance(x, ...)
```

Arguments

x	A simtab_result or simtab_spec.
...	Ignored.

Value

A one-row summary data.frame, where available.

Examples

```
fit <- regtab(epitabl, "adjudicated_acs", ~ age + sex, family = binomial())
generics::glance(fit)
```

---

is_simtab *Test whether an object is a SimtablR result, optionally of a given preset*

Description

Namespaced subclasses (simtab_tb, simtab_table1, simtab_regtab, simtab_diag, simtab_roc, ...) are the supported way to test result identity; bare legacy tags (kept in the class vector for external, backward-compatible inherits() callers) should not be tested directly in new package code. Use is_simtab(x, preset) instead.

Usage

```
is_simtab(x, preset = NULL)
```

Arguments

x	Any object.
preset	Optional character engine/preset name (e.g. "tb", "roc").

Value

TRUE/FALSE.

Examples

```
res <- tb(epitabl, sex, diabetes)
is_simtab(res)
is_simtab(res, preset = "tb")
is_simtab(iris)
```

---

journal_style *Create a SimtablR style / journal preset*

Description

Builds a simtab_style object: the single specification that controls both the text conventions of a table (how counts, percentages, confidence intervals and p-values are written) and its visual flextable theme. Pass the result to the style argument of table1(), or register it under a name with register_journal() so it can be referred to as style = "myjournal".

Usage

```
journal_style(
  np_template = NULL,
  count_header = "n (%)",
  digits_cont = 1,
  digits_est = 2,
  ci_sep = " - ",
  ci_parens = "()",
  est_template = NULL,
  pval_thresh = 0.001,
  pval_digits = 3,
  pval_upper = NULL,
  pval_leading_zero = TRUE,
  pval_adaptive = FALSE,
  percent_sign = TRUE,
  flex = NULL
)
```

Arguments

np_template	Character template for a categorical "count (percent)" cell, using {n} and {p} placeholders. If NULL, derived from percent_sign ("{n} ({p}%)" or "{n} ({p})").
count_header	Character appended to categorical variable headers to note the cell convention, e.g. "n (%)" or "No. (%)".
digits_cont	Integer decimals for continuous summaries. Default 1.
digits_est	Integer decimals for effect-measure estimates. Default 2.
ci_sep	Character separating confidence-interval bounds (and IQR bounds), e.g. " - ", "\u2013", " to ". Default " - ". Where the separator would be ambiguous it is replaced for that interval only: a comma separator becomes "; " when a bound contains a comma (a decimal comma or grouping mark), and a dash separator becomes " to " when a bound is negative.
ci_parens	Two characters wrapping the confidence interval, e.g. "()" or "[]". Default "()".
est_template	Character template for an estimate-with-CI string, using {est}, {lower}, {upper}, {sep} and the bracket placeholders {lp}/{rp}. If NULL, "{est} {lp}{lower}{sep}{upper}{rp}".
pval_thresh	Numeric; p-values below this print as "<thresh". Default 0.001.
pval_digits	Integer decimals for p-values. Default 3.
pval_upper	Optional numeric ceiling; p-values above this print as ">upper". Default NULL.
pval_leading_zero	Logical; whether p-values keep a leading zero before the decimal point. Default TRUE.
pval_adaptive	Logical; whether p-values above 0.01 use two decimals and smaller p-values use pval_digits. Default FALSE.
percent_sign	Logical; whether the default np_template includes a %. Ignored when np_template is supplied. Default TRUE.
flex	A function of one argument function(ft) ... that styles and returns a flextable, applied by as_flextable() on a table1 result. If NULL, the SimtablR house theme (booktabs styling).

Value

An object of class "simtab_style".

See Also

register_journal(), list_journals(), table1()

Examples

```
s <- journal_style(count_header = "No. (%)", ci_sep = " to ", percent_sign = FALSE)
```

label *Set display labels*

Description

Set display labels

Usage

```
label(spec, ..., labels = NULL)
```

Arguments

spec	A simtab_spec.
...	Named character labels.
labels	Optional named character vector of labels.

Details

Label edits merge by variable name: supplied labels replace matching entries and all unspecified recorded labels remain unchanged. On computed results, this is a presentation-only edit; raw evidence is retained and the result call records the edit.

Value

A modified simtab_spec.

Examples

```
label(simtab(epitabl), age = "Age, years", sex = "Sex")
```

list_engines *List registered SimtablR engines*

Description

List registered SimtablR engines

Usage

```
list_engines()
```

Value

A character vector of registered engine names.

See Also

register_engine()

Examples

```
list_engines()
```

`list_journals` *List available journal style presets*

Description

List available journal style presets

Usage

```
list_journals()
```

Value

A character vector of registered preset names.

See Also

journal_style(), register_journal()

Examples

```
list_journals()
```

`measure` *Set an effect measure*

Description

Set an effect measure

Usage

```
measure(spec, m, ref = NULL, conf.level = 0.95, adjust = NULL)
```

Arguments

spec	A simtab_spec.
m	A single effect-measure name.
ref	Optional reference level: one level shared by every variable, or a named list with one level per variable, e.g. list(sex = "Female", smoking = "Never").
conf.level	Confidence level between 0 and 1.
adjust	Optional tidyselect adjustment covariates, captured as a convenience for the builder register.

Value

A modified simtab_spec.

Examples

```
measure(simtab(epitabl), "OR", conf.level = 0.90)
```

---

missingness *Set missing-data policy*

Description

Set missing-data policy

Usage

```
missingness(spec, display = NULL, denominator = NULL, model = NULL)
```

Arguments

spec	A simtab_spec.
display	Whether to display missingness rows.
denominator	Missing-data denominator policy, "available" or "complete".
model	Missing-data model policy, "drop", "fail", or "explicit".

Value

A modified simtab_spec.

Examples

```
missingness(simtab(epitabl), display = TRUE, denominator = "available")
```

model-accessors *Access retained model evidence*

Description

`regtab()` and `survtab()` are reporting-evidence objects rather than mutable fitted-model objects. These methods expose the conventional model evidence that SimtablR retains: link-scale coefficients, confidence intervals, formulas, analysed observation counts, and covariance matrices.

Usage

```
## S3 method for class 'simtab_result'
coef(object, outcome = NULL, ...)

## S3 method for class 'simtab_result'
vcov(object, outcome = NULL, complete = TRUE, ...)

## S3 method for class 'simtab_result'
formula(x, outcome = NULL, ...)

## S3 method for class 'simtab_result'
nobs(object, outcome = NULL, ...)

## S3 method for class 'simtab_result'
confint(object, parm, level = 0.95, outcome = NULL, ...)
```

Arguments

`object`, `x`	A computed `regtab()` or `survtab()` result.
`outcome`	Optional single outcome name. Omit it for the conventional single-outcome value or a predictably named multi-outcome collection.
`...`	Unused.
`complete`	Retained for compatibility with `stats::vcov()`. SimtablR returns the complete copied covariance matrix.
`parm`	Optional coefficient names or positions for `stats::confint()`.
`level`	Confidence level for `stats::confint()`.

Details

A single-outcome result returns the conventional vector, matrix, formula, or scalar. A multi-outcome result returns a list named by outcome, except `stats::nobs()` which returns a named integer vector. Supply `outcome` to select one outcome. Unknown, non-scalar, and failed-outcome selections raise `simtab_error_model`.

Confidence intervals replay the inference represented by the result. Wald intervals can be returned at another `level` from copied coefficient and covariance evidence. Firth profile intervals are retained

only at the level used to fit the result; requesting another level requires refitting and is therefore rejected.

SimtablR deliberately does not retain fitted values, residuals, response vectors, mutable fitted-model objects, or a prediction/forecasting surface. Use `stats::glm()`, `survival::coxph()`, or `logistf::logistf()` directly when those model-object workflows are required.

Value

`coef()` returns a named numeric vector or named list of vectors; `confint()` and `vcov()` return a matrix or named list of matrices; `formula()` returns a formula or named list of formulas; `nobs()` returns an integer scalar or named integer vector.

Examples

```
data(epitabl)
fit <- regtab(
  epitabl,
  outcomes = "rehospitalized",
  predictors = ~ age + sex,
  family = binomial("logit"),
  robust = FALSE
)
coef(fit)
confint(fit)
formula(fit)
nobs(fit)
vcov(fit)
```

---

model_info *Inspect retained model convergence information*

Description

Returns the stable model-information rows retained by `regtab()` or `survtab()`, including analysed N, estimator, convergence, boundary, failure, and error fields where applicable. Unlike the conventional accessors, `model_info()` may select a failed outcome because its purpose is to inspect that failure.

Usage

```
model_info(object, outcome = NULL, ...)
```

Arguments

`object`	A computed `regtab()` or `survtab()` result.
`outcome`	Optional single outcome name.
`...`	Unused.

Value

A data frame with one row per requested model.

Examples

```
data(epitabl)
fit <- regtab(
  epitabl, "rehospitalized", ~ age + sex,
  family = binomial("logit"), robust = FALSE
)
model_info(fit)
```

`overall` *Toggle the overall column*

Description

Toggle the overall column

Usage

```
overall(spec, value = TRUE)
```

Arguments

`spec`	A `simtab_spec`.
`value`	`TRUE` or `FALSE`.

Value

A modified `simtab_spec`.

Examples

```
overall(simtab(epitabl), FALSE)
```

plot.simtab_diag *Plot diagnostic test results*

Description

Draws a fourfold display of the retained confusion matrix with sensitivity and specificity annotated on the bottom margin.

Usage

```
## S3 method for class 'simtab_diag'
plot(x, col = c("#ffcccc", "#ccffcc"), main = "Confusion Matrix", ...)
```

Arguments

x	A simtab_diag result.
col	Character vector of length 2. Fill colours for the negative and positive quadrants respectively. Default: c("#ffcccc", "#ccffcc").
main	Character. Plot title. Default: "Confusion Matrix".
...	Additional arguments passed to graphics::fourfoldplot().

Value

Invisibly returns x.

Examples

```
d <- diag_test(epitabl, poc_hstn_positive, adjudicated_acs,
               positive = "Yes", test_positive = "Positive")
plot(d)
```

print.simtab_rbind_tb *Print Method for simtab_rbind_tb Objects*

Description

Print Method for simtab_rbind_tb Objects

Usage

```
## S3 method for class 'simtab_rbind_tb'
print(x, digits = NULL, ...)
```

Arguments

x	A simtab_rbind_tb object.
digits	Minimum number of significant digits to be printed.
...	Additional arguments.

Value

Invisibly returns x.

Examples

```
t1 <- tb(epitabl, sex, diabetes)
t2 <- tb(epitabl, hypertension, diabetes)
print(rbind(t1, t2))
```

---

print.simtab_result *Print a computed SimtablR result*

Description

Dispatches to the engine's registered print renderer. Falls back to an honest generic (header + as_data_frame renderer if available + advice) when the engine registered no print renderer.

Usage

```
## S3 method for class 'simtab_result'
print(x, ...)
```

Arguments

x	A simtab_result.
...	Passed to subclass renderers. For table1 and regtab, details = TRUE restores the decorative result summary and stored call. For regtab, vif = TRUE adds VIF-equivalent columns to the display frame.

Value

Invisibly returns x.

Examples

```
res <- tb(epitabl, sex, diabetes)
print(res)
```

`rbind.simtab_tb` *Combine tb Objects by Rows*

Description

Vertical stacking of `tb` objects to create multi-variable tables.

Usage

```
## S3 method for class 'simtab_tb'
rbind(..., deparse.level = 1)
```

Arguments

`...`	Objects of class `simtab_tb` to be combined.
`deparse.level`	Integer controlling label deparsing (unused).

Details

This method is registered as an S3 method on `base::rbind()` (it does not mask `base::rbind`), so `rbind(tb1, tb2)` dispatches here while ordinary matrix/data.frame `rbind()` is unaffected.

The resulting `rbind_tb` object combines multiple bivariate tables sharing a common stratifying column into a stacked summary table.

Value

A combined object of class `c("simtab_rbind_tb", "rbind_tb", "simtab")`.

See Also

`tb()`

Examples

```
t1 <- tb(epitabl, sex, diabetes)
t2 <- tb(epitabl, hypertension, diabetes)
rbind(t1, t2)
```

register_engine *Register a SimtablR computation engine*

Description

Stores the full engine contract — compute function, result subclass, optional legacy class tag, optional hard-error validator, a named subset of renderer verbs, and an optional engine-options validator — under a case-insensitive name. A user-registered engine gets print/as.data.frame/ as_flextable/as_gt/tidy/glance/autoplo dispatch through `evaluate()`'s shared `simtab_result` methods with no further S3 ceremony.

Usage

```
register_engine(
  name,
  compute,
  subclass = paste0("simtab_", .normalise_registry_name(name)),
  legacy_tag = NULL,
  validate = NULL,
  renderers = list(),
  engine_opts = NULL,
  verbs = NULL,
  overwrite = FALSE
)
```

Arguments

`name`	Character engine name, e.g. `"roc"`.
`compute`	Function with signature `function(spec, data)` returning `list(data = ..., meta = ...)`.
`subclass`	Character result subclass. Defaults to `paste0("simtab_", name)`.
`legacy_tag`	Optional extra class tag kept for backward-compatible `inherits()` checks. Never used for SimtablR's own dispatch.
`validate`	Optional function `function(spec) -> invisible(spec)` for hard, engine-specific errors only.
`renderers`	Named list of renderer functions, names drawn from `.renderer_verbs()` (`print`, `as_data_frame`, `as_flextable`, `as_gt`, `tidy`, `glance`, `autoplot`, `as_methods`). Each function takes `function(x, ...)`.
`engine_opts`	Optional function `function(opts) -> invisible(opts)` validating the engine's spec-level options.
`verbs`	Optional character vector of the evidence-changing verbs the engine uses: any of `"stratify"`, `"adjust"`, `"measure"`, `"test"`, `"set_summary"`, and `"missingness"`. Applying another of these verbs to a result from this engine warns and returns the result unchanged. `NULL` (the default) accepts every verb. `label()`, `fmt()`, and `style()` always apply.

overwrite	Logical. Replacing a built-in engine (descriptive, bivariate, glm, accuracy, roc, cox, e_value) is an error unless overwrite = TRUE, because it changes what table1(), tb(), and the other presets compute. Re-registering your own engine needs no flag.

Value

Invisibly, name.

See Also

list_engines(), is_simtab()

Examples

```
## Not run:
register_engine("dummy", compute = function(spec, data) list(data = data, meta = list()))

## End(Not run)
```

---

register_journal *Register a named journal style preset*

Description

Stores a simtab_style in SimtablR's preset registry under name so it can be used as style = name in table1() and the add_*() helpers. Registering a name that already exists overwrites it.

Usage

```
register_journal(name, style)
```

Arguments

name	Character preset name (case-insensitive).
style	A simtab_style object from journal_style().

Value

Invisibly, name.

See Also

journal_style(), list_journals()

Examples

```
register_journal("mylab", journal_style(count_header = "No. (%)", ci_sep = " to "))
list_journals()
```

regtab *Fit one regression model per outcome*

Description

`regtab()` fits the same set of predictors to one or more outcomes and returns the estimates side by side in a single publication-style table, one column per outcome. The default Poisson-log model with robust standard errors gives rate ratios for counts and prevalence or risk ratios for 0/1 outcomes (modified Poisson); use `family = binomial()` for odds ratios or `gaussian()` for mean differences. For crude estimates alongside a descriptive table use `table1()`.

Usage

```
regtab(
  data,
  outcomes,
  predictors,
  family = poisson(link = "log"),
  offset = NULL,
  robust = TRUE,
  method = "glm",
  exponentiate = NULL,
  labels = NULL,
  predictor_labels = NULL,
  d = 2,
  conf.level = 0.95,
  include_intercept = FALSE,
  p_values = FALSE,
  design = NULL
)
```

Arguments

`data`	A data frame.
`outcomes`	Character vector of outcome column names. Each outcome gets its own model with the same predictors.
`predictors`	The right-hand side of the model, as a one-sided formula (`~ age + sex`) or a string (`"age * sex + smoking"`). Write transformations such as centring directly in the formula.
`family`	A `stats::family()` object. Poisson-log needs a numeric outcome (counts or 0/1); `binomial()` also accepts two-level factors.
`offset`	Optional person-time column: a bare name or string. With a Poisson-log model, `offset(log(offset))` is added so estimates become incidence-rate ratios.
`robust`	Standard errors: `TRUE` for HC0 robust errors, `FALSE` for model-based errors, or one of `"HC0"`, `"HC1"`, `"HC2"`, `"HC3"`. Prefer `"HC3"` in small samples (Long & Ervin, 2000).

method	String. "glm" fits with stats::glm(); "firth" fits Firth penalised logistic regression with the logistf package, for binomial-logit models only.
exponentiate	Logical. Whether to report exponentiated estimates. If NULL, they are exponentiated for Poisson, binomial, and quasi- families and left on the original scale for Gaussian models.
labels	Named character vector of display labels for outcomes, e.g. c(ed_visits = "ED revisits").
predictor_labels	Named character vector of display labels for model terms, e.g. c(sexMale = "Male sex").
d	Integer. Decimal places for estimates and intervals.
conf.level	Number between 0 and 1. Confidence level for intervals.
include_intercept	Logical. If TRUE, show the intercept row.
p_values	Logical. If TRUE, add a p-value column for each outcome.
design	String. Optional study design recorded on the result. With a Poisson-log model and an offset, it lets the estimate be labelled as an incidence-rate ratio.

Details

Statistical methods:

Each outcome is fitted with stats::glm() and the requested family. Intervals are Wald intervals on the link scale, using HC0 sandwich standard errors by default; a Poisson-log model with robust errors on a 0/1 outcome is the modified Poisson approach for prevalence and risk ratios (Zou, 2004). method = "firth" uses penalised-likelihood (profile) inference instead, so the HC options do not apply.

For models with two or more predictor terms, generalized variance inflation factors (Fox & Monette, 1992) are stored on the result. Show them with as.data.frame(fit, vif = TRUE) or generics::glance(fit, vif = TRUE).

Missing data:

Rows with a missing outcome or predictor are dropped from that outcome's model, so the N can differ between outcomes. It is shown in the table and in model_info().

Modifying the result:

coef(), confint(), vcov(), formula(), and nobs() work on the result, returning one entry per outcome or a single one with outcome = "name". model_info() reports convergence and failed models, generics::tidy() returns one row per term, and ggplot2::autoplot() draws a forest plot.

Limitations:

The result is a reporting table, not a fitted model: it keeps no fitted values or residuals and cannot predict. For conditional logistic regression, prediction, or model diagnostics, fit stats::glm(), logistf::logistf(), or survival::clogit() directly.

Value

A `simtab_result` of class `simtab_regtab`. Print it to see the formatted table, convert it with `as.data.frame()`, or save it with `export_docx()`, `export_pptx()`, or `export_xlsx()`. Unrounded results are stored in `$data`.

References

Zou, G. (2004). A modified Poisson regression approach to prospective studies with binary data. *American Journal of Epidemiology*, 159(7), 702–706. doi:10.1093/aje/kwh090.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1), 27–38. doi:10.1093/biomet/80.1.27.

Fox, J., & Monette, G. (1992). Generalized collinearity diagnostics. *Journal of the American Statistical Association*, 87(417), 178–183. doi:10.1080/01621459.1992.10475190.

Long, J. S., & Ervin, L. H. (2000). Using heteroscedasticity consistent standard errors in the linear regression model. *The American Statistician*, 54(3), 217–224. doi:10.1080/00031305.2000.10474549.

See Also

`model_info()` for convergence, `survtab()` for time-to-event outcomes, `table1()` for crude and adjusted effects in a descriptive table, and simtablr_references for all references cited by SimtablR.

Examples

```
# Rate ratios for two count outcomes (Poisson, robust SEs)
fit <- regtab(
  epitabl,
  outcomes = c("ed_visits", "length_of_stay"),
  predictors = ~ age + sex + smoking
)
fit

# Odds ratios for a binary outcome, with p-values
regtab(
  epitabl, "rehospitalized", ~ age + sex + diabetes,
  family = binomial(), p_values = TRUE
)

# One row per term, for further processing
generics::tidy(fit)
```

`repro_manifest` *Build a reproducibility manifest*

Description

Captures the runtime and recorded analysis decisions needed to reproduce a SimtablR result. The data hash and ruleset version are read from the result; the data are not re-hashed.

Usage

```
repro_manifest(result)
```

Arguments

result	A simtab_result or simtab_spec.

Value

A simtab_manifest list. design and design_source describe any study design recorded on the specification or the source data frame, whether or not it took part in resolving measure; design_used distinguishes the two, and is TRUE only when resolved_from is "design". A design recorded only on the data frame (set_design() on a data.frame) is advisory and never sets design_used; see set_design().

Examples

```
res <- tb(epitabl, sex, diabetes)
repro_manifest(res)
```

---

roc *Evaluate continuous markers with ROC curves*

Description

roc() measures how well one or more continuous markers discriminate a binary outcome. It reports the area under the ROC curve (AUC) with a confidence interval and, by default, the Youden-optimal cutpoint with its sensitivity, specificity, and predictive values. With two or more markers, their AUCs are compared pairwise with the DeLong test. For a test that is already binary use diag_test(). Requires the pROC package.

Usage

```
roc(
  data,
  marker,
  outcome,
  positive = NULL,
  direction = c("auto", ">", "<"),
  cutpoint = c("youden", "none"),
  conf.level = 0.95,
  d = 2,
  percent = TRUE
)
```

Arguments

data	A data frame.
marker	One or more numeric marker columns: a bare name, c() of names, or a tidyselect expression.
outcome	The binary outcome column: a bare name or string.
positive	The level of outcome that means "disease present". If NULL, common labels such as "Yes", "1", or "Positive" are detected, falling back to the last level.
direction	String. Which marker values indicate disease: "<" if higher values do, ">" if lower values do, or "auto" to let pROC choose by comparing the group medians.
cutpoint	String. "youden" reports the cutpoint that maximises sensitivity + specificity - 1; "none" reports the AUC only.
conf.level	Number between 0 and 1. Confidence level for AUC intervals.
d	Integer. Decimal places for displayed estimates.
percent	Logical. If TRUE, show the cutpoint's sensitivity, specificity, and predictive values as percentages. The AUC and the cutpoint itself are always shown as decimals.

Details

Statistical methods:

ROC curves and AUCs are computed with pROC (Robin et al., 2011), using DeLong intervals for the AUC and the paired DeLong test to compare markers measured on the same patients (DeLong et al., 1988). A cutpoint chosen from the same data is optimistic: its sensitivity and specificity will usually be lower in new patients (Ewald, 2006), and a note says so. The AUC describes discrimination only, not calibration.

Missing data:

Rows with a missing outcome or marker value are dropped, across all markers at once, so every marker is evaluated on the same patients. A message reports how many rows were removed. Each marker needs both outcome classes among the remaining rows.

Modifying the result:

Use fmt() to change decimals or percentages and ggplot2::autoplot() to draw the ROC curves. The curve coordinates and DeLong comparisons are stored in $data.

Value

A simtab_result of class simtab_roc. Print it to see the AUC table and comparisons, convert it with as.data.frame(), or save it with export_docx(), export_pptx(), or export_xlsx(). Unrounded results are stored in $data.

References

Robin, X., Turck, N., Hainard, A., et al. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12, 77. doi:10.1186/147121051277.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3), 837–845. doi:10.2307/2531595.

Ewald, B. (2006). Post hoc choice of cut points introduced bias to diagnostic research. *Journal of Clinical Epidemiology*, 59(8), 798–801. doi:10.1016/j.jclinepi.2005.11.025.

See Also

`diag_test()` for binary tests and simtablr_references for all references cited by SimtablR.

Examples

```
if (requireNamespace("pROC", quietly = TRUE)) {
  substudy <- subset(epitabl, diagnostic_substudy == "Yes")

  # AUC and Youden cutpoint for point-of-care troponin
  roc(substudy, poc_hstn_value, adjudicated_acs)

  # Compare two markers with the paired DeLong test
  fit <- roc(substudy, c(poc_hstn_value, systolic_bp), adjudicated_acs,
             positive = "Yes")
  fit

  # AUC only, no data-driven cutpoint
  roc(substudy, poc_hstn_value, adjudicated_acs, cutpoint = "none")
}
```

`sensitivity` *Recompute reviewer-loop sensitivity variations*

Description

`sensitivity()` takes a computed result and recomputes named variations of the recorded specification. Variations are either functions that take a `simtab_spec` and return a modified `simtab_spec`, or supported shorthand values such as `denominator = "complete"` and `measure = "OR"`.

Usage

```
sensitivity(result, ...)

## S3 method for class 'simtab_sensitivity'
as_flextable(x, ...)
```

Arguments

result	A computed simtab_result.
...	Named sensitivity variations. Unnamed variations are rejected.
x	A simtab_sensitivity report.

Value

A simtab_sensitivity report.

Examples

```
res <- tb(epitabl, renal_impairment, adjudicated_acs, measure = "rr", ref = "No")
sensitivity(res, measure = "or")
```

---

set_design *Set study design metadata*

Description

Records a study design on a data frame or a simtab_spec. The two forms behave differently: on a simtab_spec, the design participates in evaluate()'s effect-measure resolution, so a call such as set_design(spec, "cohort") |> evaluate() can pick RR over an unset measure. On a data frame, the design is recorded only as an attribute for educator advice (methodological guidance rules can read it); it is *not* consulted when resolving an effect measure, even for a tb()/table1() call built directly from that data frame. To resolve a measure from design, pass design = to the direct function or call set_design() on a simtab_spec (see vignette("study-design-effect-measures")).

Usage

```
set_design(data, design)
```

Arguments

data	A data.frame or simtab_spec.
design	Study design, e.g. "cross_sectional", "cohort", or "case_control".

Value

data with a SimtablR design attribute, or a modified simtab_spec.

Examples

```
set_design(simtab(epitabl), "cohort")
cohort_data <- set_design(epitabl, "cohort") # advisory only; see Details
```

set_summary *Set the default summary statistic*

Description

Set the default summary statistic

Usage

```
set_summary(spec, stat = c("auto", "median", "mean"), .by_var = NULL)
```

Arguments

spec	A simtab_spec.
stat	Summary statistic, one of "auto", "median", or "mean".
.by_var	Optional variable for a per-variable override.

Value

A modified simtab_spec.

Examples

```
set_summary(describe(simtab(epitabl), age), "median")
```

simtab *Start a SimtablR analysis specification*

Description

Captures data once into a shared data reference and returns an inert simtab_spec. Printing the returned object shows the analysis plan; it never performs estimation.

Usage

```
simtab(data)
```

Arguments

data	A data frame.

Value

A simtab_spec object.

Examples

```
spec <- simtab(epitabl)
spec
```

`simtab_completions` *Opt-in RStudio autocomplete for SimtablR calls*

Description

Installs (or removes) a session-wide completer that offers column-name and terse-flag completions inside calls to `tb()`, `table1()`, `regtab()`, `diag_test()`, `roc()`, and `survtab()` — e.g. `tb(df, var<TAB>)`.

Usage

```
simtab_completions(enable = TRUE, quiet = FALSE)
```

Arguments

`enable`	Logical. `TRUE` (default) installs the completer; `FALSE` removes it and restores whatever `custom.completer` held before.
`quiet`	Logical. Suppress the confirmation message. Default `FALSE`.

Details

This works by setting `rc.options(custom.completer = )`, the only completion-extension hook RStudio's editor honors for package authors. Because that option is **session-global**, enabling it affects completion behavior everywhere in the session, not just inside SimtablR calls: on any line that isn't a recognized SimtablR call, this completer declines the token, and RStudio falls back to its own completion engine (or to a previously installed custom completer, which is called first).

Value

Invisibly, a list with `enabled` (logical), `mode` (`"throw"` when installed, `"unsupported"` when the host cannot support it), and `host` (`"rstudio"` or `"other"`).

Supported hosts

RStudio only. Declining a token relies on the host restoring native completion, which RStudio does and a plain console does not; on any other front end (Rterm, radian, batch), enabling the completer would leave non-SimtablR tokens with no completions at all. `simtab_completions()` therefore declines to install outside RStudio and tells you so, rather than degrading completion for the rest of the session.

Never installed automatically — SimtablR's `.onLoad` does not call this. Call it yourself, e.g. from `.Rprofile`: `if (interactive()) SimtablR::simtab_completions()`.

Examples

```
simtab_completions(enable = FALSE, quiet = TRUE)
```

---

simtab_errors *SimtablR condition classes*

Description

SimtablR errors inherit from simtab_error. More specific classes identify the failing surface so callers can assert on behavior without matching prose: simtab_error_input for invalid user-supplied arguments at a public entry point, simtab_error_binding for variable binding and role-selection failures, simtab_error_spec for invalid SimtablR specifications, simtab_error_engine for engine validation or compute failures, simtab_error_flag for invalid table flags, simtab_error_render for missing renderer methods, and simtab_error_dependency for a suggested package that is needed but not installed. simtab_error_export identifies unsafe paths, destination collisions, and failed file writes. simtab_error_model identifies unsupported model access and invalid outcome selectors on retained model evidence.

Details

Messages follow one house form: an x line stating the problem, an i line giving the context that produced it, and a v line naming the next action.

{} expressions in the message are evaluated in the calling function's environment, so a caller can interpolate its own locals directly rather than pre-building the string with sprintf().

Value

This documentation-only topic evaluates to NULL; condition helpers signal classed errors rather than returning values.

Examples

```
tryCatch(
  tb(epitabl, non_existent_var, diabetes),
  simtab_error = function(e) message("Caught simtab error")
)
```

`simtablr` *Build a manuscript-ready SimtablR report in one call*

Description

`simtablr()` composes existing presets into a transparent `simtab_report`. It creates a cohort-description Table 1 and an exposure-to-outcome Table 2 using the same `table1()` effect path as direct users. No new estimation is performed by the orchestrator.

Usage

```
simtablr(
  data,
  outcome,
  exposure,
  vars = NULL,
  adjust = NULL,
  design = NULL,
  test = TRUE,
  style = "default",
  ...
)
```

Arguments

`data`	A data.frame.
`outcome`	Bare column name for the outcome.
`exposure`	Bare column name for the exposure.
`vars`	Optional tidyselect expression of Table 1 variables. Defaults to all columns except `outcome` and `exposure`.
`adjust`	Optional tidyselect expression of adjustment covariates for Table 2.
`design`	Optional per-call study design used by the existing design-aware measure resolver.
`test`	Logical or test name passed to `table1()`.
`style`	Display style passed to `table1()`.
`...`	Additional `table1()` arguments. Effect-specific arguments such as `measure` are applied to Table 2 only.

Value

A `simtab_report` with named `table1` and `table2` results.

Examples

```
## Not run:
data(epitabl)
simtablr(epitabl, outcome = adjudicated_acs, exposure = smoking,
         vars = c(age, sex, smoking), adjust = c(age, sex),
         design = "cross_sectional")

## End(Not run)
```

`simtablr_guidance` *Set or query SimtablR educator guidance*

Description

Guidance changes only which stored advice is displayed. `"default"` shows severity 3-4 advice and counts hidden severity 1-2 notes; `"important"` and its compatibility alias `"quiet"` show severity 3-4 advice only; `"teaching"` shows severity 1-4 advice with external citations and recorded rationale; `"strict"` shows severity 1-4 advice in compact prose; and `"off"` hides automatic advice. Severity 0 remains audit-only. When a result has only one non-audit advice entry, that entry is shown under every profile except `"off"` instead of being reported as hidden.

Usage

```
simtablr_guidance(level = NULL)
```

Arguments

`level`	One of `"default"`, `"important"`, `"quiet"`, `"teaching"`, `"strict"`, or `"off"`. If omitted, returns the current level.

Details

Guidance is a session-level display setting read when a stored result is printed. Call `simtablr_guidance()` separately; do not place it inside the `...` of `table1()` or `tb()`. Changing guidance does not recompute results.

Value

The active guidance level, invisibly when setting.

See Also

simtablr_references for the full references behind the citations shown in advice.

Examples

```
previous <- simtablr_guidance()
result <- table1(epitabl, "sex")
simtablr_guidance("teaching")
print(result)
simtablr_guidance(previous)
suppressMessages(print(result))
```

`simtablr_references` *References cited by SimtablR*

Description

Full references for every method and recommendation that SimtablR cites, both on its help pages and in its educator advice. Advice messages show a short tag such as "(Ver Hoef & Boveng 2007)"; each entry below starts with that tag in bold, so you can look it up here. Use `simtablr_guidance("teaching")` to show the citation on every advice message, or `advise(result)` to list the advice for a stored result.

Details

Effect measures and association tests:

Anscombe 1956. Anscombe, F. J. (1956). On estimating binomial response relations. *Biometrika*, 43(3/4), 461–464. doi:10.1093/biomet/43.34.461.

Barros & Hirakata 2003. Barros, A. J. D., & Hirakata, V. N. (2003). Alternatives for logistic regression in cross-sectional studies: an empirical comparison of models that directly estimate the prevalence ratio. *BMC Medical Research Methodology*, 3, 21. doi:10.1186/14712288321.

Bender & Lange 2001. Bender, R., & Lange, S. (2001). Adjusting for multiple testing: when and how? *Journal of Clinical Epidemiology*, 54(4), 343–349. doi:10.1016/S08954356(00)003140.

Campbell 2007. Campbell, I. (2007). Chi-squared and Fisher-Irwin tests of two-by-two tables with small sample recommendations. *Statistics in Medicine*, 26(19), 3661–3675. doi:10.1002/sim.2832.

Fisher 1935. Fisher, R. A. (1935). *The Design of Experiments*. Oliver & Boyd.

Greenland & Robins 1985. Greenland, S., & Robins, J. M. (1985). Estimation of a common effect parameter from sparse follow-up data. *Biometrics*, 41(1), 55–68. doi:10.2307/2530643.

Haldane 1956. Haldane, J. B. S. (1956). The estimation and significance of the logarithm of a ratio of frequencies. *Annals of Human Genetics*, 20(4), 309–311. doi:10.1111/j.1469-1809.1955.tb01285.x.

Katz et al. 1978. Katz, D., Baptista, J., Azen, S. P., & Pike, M. C. (1978). Obtaining confidence intervals for the risk ratio in cohort studies. *Biometrics*, 34(3), 469–474. doi:10.2307/2530610.

Knol 2011. Knol, M. J., Le Cessie, S., Algra, A., Vandenbroucke, J. P., & Groenwold, R. H. H. (2012). Overestimation of risk ratios by odds ratios in trials and cohort studies: alternatives to logistic regression. *Canadian Medical Association Journal*, 184(8), 895–899. Published online 2011. doi:10.1503/cmaj.101715.

Pearson 1900. Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed

to have arisen from random sampling. *Philosophical Magazine, Series 5*, 50(302), 157–175. doi:10.1080/14786440009463897.

Rothman, Greenland & Lash 2008. Rothman, K. J., Greenland, S., & Lash, T. L. (2008). *Modern Epidemiology* (3rd ed.). Lippincott Williams & Wilkins.

van Elteren 1960. van Elteren, P. H. (1960). On the combination of independent two-sample tests of Wilcoxon. *Bulletin of the International Statistical Institute*, 37, 351–361.

Woolf 1955. Woolf, B. (1955). On estimating the relation between blood group and disease. *Annals of Human Genetics*, 19(4), 251–253. doi:10.1111/j.14691809.1955.tb01348.x.

Zou 2004. Zou, G. (2004). A modified Poisson regression approach to prospective studies with binary data. *American Journal of Epidemiology*, 159(7), 702–706. doi:10.1093/aje/kwh090.

Descriptive statistics and group balance:

Altman 1991. Altman, D. G. (1991). *Practical Statistics for Medical Research*. Chapman & Hall.

Austin 2009. Austin, P. C. (2009). Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine*, 28(25), 3083–3107. doi:10.1002/sim.3697.

Bland & Altman 1996. Bland, J. M., & Altman, D. G. (1996). Statistics notes: Transforming data. *BMJ*, 312(7033), 770. doi:10.1136/bmj.312.7033.770.

Yang & Dalton 2012. Yang, D., & Dalton, J. E. (2012). A unified approach to measuring the effect size between two groups using SAS. *SAS Global Forum 2012*, Paper 335-2012. https://support.sas.com/resources/papers/proceedings12/335-2012.pdf.

Regression models:

Firth 1993. Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1), 27–38. doi:10.1093/biomet/80.1.27.

Fox & Monette 1992. Fox, J., & Monette, G. (1992). Generalized collinearity diagnostics. *Journal of the American Statistical Association*, 87(417), 178–183. doi:10.1080/01621459.1992.10475190.

Hanley, Negassa, Edwardes & Forrester 2003. Hanley, J. A., Negassa, A., Edwardes, M. D., & Forrester, J. E. (2003). Statistical analysis of correlated data using generalized estimating equations: an orientation. *American Journal of Epidemiology*, 157(4), 364–375. doi:10.1093/aje/kwf215.

Heinze & Schemper 2002. Heinze, G., & Schemper, M. (2002). A solution to the problem of separation in logistic regression. *Statistics in Medicine*, 21(16), 2409–2419. doi:10.1002/sim.1047.

Long & Ervin 2000. Long, J. S., & Ervin, L. H. (2000). Using heteroscedasticity consistent standard errors in the linear regression model. *The American Statistician*, 54(3), 217–224. doi:10.1080/00031305.2000.10474549.

Riley 2019. Riley, R. D., Snell, K. I. E., Ensor, J., et al. (2019). Minimum sample size for developing a multivariable prediction model: Part II - binary and time-to-event outcomes. *Statistics in Medicine*, 38(7), 1276–1296. doi:10.1002/sim.7992.

van Smeden 2016. van Smeden, M., de Groot, J. A. H., Moons, K. G. M., et al. (2016). No rationale for 1 variable per 10 events criterion for binary logistic regression analysis. *BMC Medical Research Methodology*, 16, 163. doi:10.1186/s1287401602673.

van Smeden 2019. van Smeden, M., Moons, K. G. M., de Groot, J. A. H., et al. (2019). Sample size for binary logistic prediction models: beyond events per variable criteria. *Statistical Methods in Medical Research*, 28(8), 2455–2474. doi:10.1177/0962280218784726.

Ver Hoef & Boveng 2007. Ver Hoef, J. M., & Boveng, P. L. (2007). Quasi-Poisson vs. negative binomial regression: how should we model overdispersed count data? *Ecology*, 88(11), 2766–2772. doi:10.1890/070043.1.

Westreich & Greenland 2013. Westreich, D., & Greenland, S. (2013). The Table 2 fallacy: presenting and interpreting confounder and modifier coefficients. *American Journal of Epidemiology*, 177(4), 292–298. doi:10.1093/aje/kws412.

Survival analysis:

Cox 1972. Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society, Series B*, 34(2), 187–202. doi:10.1111/j.25176161.1972.tb00899.x.

Grambsch & Therneau 1994. Grambsch, P. M., & Therneau, T. M. (1994). Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, 81(3), 515–526. doi:10.1093/biomet/81.3.515.

Diagnostic accuracy and ROC curves:

Altman et al. 2000. Altman, D. G., Machin, D., Bryant, T. N., & Gardner, M. J. (2000). *Statistics with Confidence* (2nd ed.). BMJ Books.

Clopper & Pearson 1934. Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4), 404–413. doi:10.1093/biomet/26.4.404.

Cohen 1960. Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46. doi:10.1177/001316446002000104.

DeLong et al. 1988. DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3), 837–845. doi:10.2307/2531595.

Ewald 2006. Ewald, B. (2006). Post hoc choice of cut points introduced bias to diagnostic research. *Journal of Clinical Epidemiology*, 59(8), 798–801. doi:10.1016/j.jclinepi.2005.11.025.

Fleiss, Cohen & Everitt 1969. Fleiss, J. L., Cohen, J., & Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5), 323–327. doi:10.1037/h0028106.

Glas et al. 2003. Glas, A. S., Lijmer, J. G., Prins, M. H., Bonsel, G. J., & Bossuyt, P. M. M. (2003). The diagnostic odds ratio: a single indicator of test performance. *Journal of Clinical Epidemiology*, 56(11), 1129–1135. doi:10.1016/S08954356(03)00177X.

Hanley & McNeil 1982. Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1), 29–36. doi:10.1148/radiology.143.1.7063747.

Robin et al. 2011. Robin, X., Turck, N., Hainard, A., et al. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12, 77. doi:10.1186/147121051277. This is also the source for the "pROC direction = 'auto'" advice tag.

Simel et al. 1991. Simel, D. L., Samsa, G. P., & Matchar, D. B. (1991). Likelihood ratios with confidence: sample size estimation for diagnostic test studies. *Journal of Clinical Epidemiology*, 44(8), 763–770. doi:10.1016/08954356(91)90128V.

Sensitivity analysis:

VanderWeele & Ding 2017. VanderWeele, T. J., & Ding, P. (2017). Sensitivity analysis in observational research: introducing the E-value. *Annals of Internal Medicine*, 167(4), 268–274. doi:10.7326/M162607.

Reporting guidelines:

STROBE item 14. von Elm, E., Altman, D. G., Egger, M., et al. (2007). The Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) statement: guidelines for reporting observational studies. *PLoS Medicine*, 4(10), e296. doi:10.1371/journal.pmed.0040296. Item 14 asks for the number of participants with missing data for each variable.

TRIPOD 2015. Collins, G. S., Reitsma, J. B., Altman, D. G., & Moons, K. G. M. (2015). Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): the TRIPOD statement. *BMJ*, 350, g7594. doi:10.1136/bmj.g7594.

See Also

`advise()` and `simtablr_guidance()` for educator advice.

`stard` *Build a STARD reporting checklist*

Description

Build a STARD reporting checklist

Usage

```
stard(x, ...)
```

Arguments

`x`	A SimtablR diagnostic or ROC result, report, or spec.
`...`	Ignored.

Value

A `simtab_checklist` object.

Examples

```
d <- diag_test(epitabl, poc_hstn_positive, adjudicated_acs,
               positive = "Yes", test_positive = "Positive")
stard(d)
```

stratify *Capture a stratifying variable*

Description

Capture a stratifying variable

Usage

```
stratify(spec, by)
```

Arguments

spec	A simtab_spec.
by	A single data-masked variable.

Value

A modified simtab_spec.

Examples

```
stratify(describe(simtab(epitabl), age, sex), adjudicated_acs)
```

strobe *Build a STROBE reporting checklist*

Description

Build a STROBE reporting checklist

Usage

```
strobe(x, ...)

## S3 method for class 'simtab_checklist'
as_flextable(x, ...)
```

Arguments

x	A simtab_checklist object.
...	Ignored.

Value

A simtab_checklist object.

Examples

```
res <- tb(epitabl, sex, diabetes)
strobe(res)
```

`style` *Set or update a SimtablR style*

Description

Set or update a SimtablR style

Usage

```
style(x, journal)
```

Arguments

`x`	A SimtablR object.
`journal`	A single style or journal preset name.

Value

A modified SimtablR object.

Examples

```
style(simtab(epitabl), "default")
```

`survtab` *Fit a Cox proportional hazards model*

Description

`survtab()` fits a Cox model to time-to-event data and returns a publication-style table of hazard ratios with confidence intervals and p-values. Give the follow-up time, the event indicator, and the predictors; the proportional hazards assumption is checked automatically. For outcomes without follow-up time use `regtab()`.

Usage

```
survtab(
  data,
  time,
  event,
  predictors,
  design = NULL,
  conf.level = 0.95,
  d = 2,
  labels = NULL,
  style = "default"
)
```

Arguments

data	A data frame.
time	Follow-up time column: a bare name or string.
event	Event indicator column: a bare name or string. May be 0/1, logical, or a two-level factor whose second level is the event.
predictors	The right-hand side of the model, as a one-sided formula (`~ age + sex`) or a character vector of column names.
design	String. Optional study design recorded on the result.
conf.level	Number between 0 and 1. Confidence level for intervals.
d	Integer. Decimal places for hazard ratios and intervals.
labels	Named character vector of display labels for model terms, e.g. `c(age = "Age (years)", sexMale = "Male sex")`. Variable labels already stored in `data` are used by default.
style	String. A journal preset name (see `list_journals()`) that controls formatting such as p-values.

Details

Statistical methods:

The model is fitted with `survival::coxph()` using the Efron method for ties. Hazard ratios have Wald confidence intervals. The proportional hazards assumption is tested with the Grambsch-Therneau test on Schoenfeld residuals (`survival::cox.zph()`); when any term fails, a warning is shown with the result. Concordance and the likelihood-ratio test p-value are available from `generics::glance()`.

Missing data:

Rows with a missing time, event, or predictor are dropped. The number of rows and events analysed is shown in the table header and in `model_info()`.

Modifying the result:

`coef()`, `confint()`, `vcov()`, `formula()`, and `nobs()` work on the result. `model_info()` reports convergence, `generics::tidy()` returns one row per term, and `ggplot2::autoplot()` draws a forest plot.

Limitations:

The result is a reporting table, not a fitted model: it keeps no residuals or fitted values and cannot predict. For stratified or time-varying models, survival curves, or residual diagnostics, fit `survival::coxph()` directly.

Value

A `simtab_result` of class `simtab_cox`. Print it to see the formatted table, convert it with `as.data.frame()`, or save it with `export_docx()`, `export_pptx()`, or `export_xlsx()`. Unrounded results are stored in `$data`.

References

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society, Series B*, 34(2), 187–202. doi:10.1111/j.25176161.1972.tb00899.x.

Grambsch, P. M., & Therneau, T. M. (1994). Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, 81(3), 515–526. doi:10.1093/biomet/81.3.515.

See Also

`regtab()` for outcomes without follow-up time, `model_info()` for convergence, and simtablr_references for all references cited by SimtablR.

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {
  # Hazard ratios for major adverse cardiovascular events
  fit <- survtab(
    epitabl,
    time = mace_time_days, event = mace_event,
    predictors = ~ age + sex + diabetes
  )
  fit

  # Concordance, likelihood-ratio test, and events analysed
  generics::glance(fit)
}
```

---

table1 *Describe a cohort across many variables (Table 1)*

Description

`table1()` builds the descriptive "Table 1" of a study in one call: numeric variables as mean (SD) or median (IQR), categorical variables as counts and percentages. Add by to get one column per group, optionally with p-values and crude or adjusted effect measures (PR, RR, or OR). For a single cross-tabulation use `tb()`; for full regression output use `regtab()`.

Usage

```
table1(
  data,
  vars,
  by = NULL,
  ...,
  flags = character(),
  measure = NULL,
  adjust = NULL,
  test = FALSE,
  summary = "auto",
  overall = TRUE,
  missing = TRUE,
  denominator = c("available", "complete"),
  na_model = c("drop", "fail", "explicit"),
  ref = NULL,
  var.type = NULL,
  labels = NULL,
  style = "default",
  d = NULL,
  conf.level = 0.95,
  design = NULL
)
```

Arguments

data	A data frame.
vars	Variables to describe: bare names in c(), a character vector, or a tidyselect expression such as starts_with("cm_").
by	Optional grouping variable: a bare name or string. Produces one column per group.
...	Terse flags such as p or or, written after by; see *Terse flags*. Only flags are allowed here.
flags	Character vector of terse flags, e.g. c("p", "or"). The programmatic form of the bare flags in ....
measure	String. Effect measure to add: "PR", "RR", or "OR". Requires a binary by, whose last level is taken as the event.
adjust	Optional covariates for an adjusted effect-measure column, given like vars. Requires measure.
test	Logical or string. TRUE adds a p-value column with an automatically chosen test. A string forces one test: "chisq", "fisher", "mcnemar", or "trend" for categorical variables, and "t", "wilcoxon", "anova", or "kruskal" for numeric ones.
summary	String. Summary for numeric variables: "auto" chooses between "mean" (mean and SD) and "median" (median and IQR) based on sample size and skewness. A named list sets it per variable, e.g. list(age = "mean", bmi = "median").

overall	Logical. If TRUE, add an "Overall" column for the whole cohort.
missing	Logical. If TRUE, add a "Missing" count row under each variable that has missing values. This does not change any other number.
denominator	String. Which rows the percentages and summaries use: "available" uses every non-missing value of each variable; "complete" uses only rows complete on all of vars and by.
na_model	String. How adjusted models handle missing values: "drop" removes incomplete rows, "fail" stops with an error, and "explicit" treats missing categorical predictors as a "(Missing)" level.
ref	Reference level for effect measures: one level for all variables, or a named list per variable. If NULL, each variable's first level is used.
var.type	Force variable types: "continuous" or "categorical", either one string for all variables or a named vector such as c(score = "continuous"). If NULL, types are detected automatically (see *Statistical methods*).
labels	Named character vector of display labels, e.g. c(smoking = "Smoking status"). Variable labels already stored in data are used by default.
style	Display style: a journal preset name (see list_journals()), a journal_style() object, "n_pct" or "pct_n", a template using {n} and {p}, or a named list per variable.
d	Integer. Decimal places for percentages and continuous summaries. If NULL (the default), percentages use one decimal and continuous summaries use the style's digits_cont (one decimal for the built-in journals).
conf.level	Number between 0 and 1. Confidence level for effect measure intervals.
design	String. Study design used to choose the effect measure when measure is not set: "cross_sectional" gives PR, "cohort" gives RR, and "case_control" gives OR.

Details

Statistical methods:

Crude effect measures for categorical variables use the same estimators as tb(): the Katz log interval for prevalence and risk ratios (Katz et al., 1978) and the Woolf logit interval for odds ratios (Woolf, 1955). For numeric variables, and for every adjusted estimate, a separate generalized linear model is fitted for each described variable, with the adjust covariates added. Odds ratios come from logistic regression. Prevalence and risk ratios come from log-binomial regression, falling back to modified Poisson regression with robust standard errors when that model does not converge (Barros & Hirakata, 2003; Zou, 2004). PR and RR are computed identically; only the label differs.

With test = TRUE, 2x2 tables use the N-1 chi-squared test (Campbell, 2007) and larger tables the Pearson chi-squared test, without continuity correction; Fisher's exact test is chosen only when an expected count is below 1. Numeric variables are compared with Welch's t-test or ANOVA when summarised by the mean, and with the Wilcoxon or Kruskal-Wallis test when summarised by the median.

Numeric variables are treated as continuous unless they look like coded categories: exactly two distinct whole numbers, or at most seven distinct whole numbers with at least 20 observations. Use var.type to override.

Missing data:

By default each variable is described using its own non-missing values, and a "Missing" row reports how many are absent (STROBE item 14). Tests are always computed on observed values. `denominator = "complete"` restricts the whole table to complete cases, so the displayed numbers can change. `na_model` controls only the adjusted models.

Modifying the result:

Multiplicity adjustment, paired tests, and standardized mean differences (Austin, 2009) are set afterwards with `test()`, e.g. `table1(df, vars, by = group, test = TRUE) |> test(p.adjust = "holm", smd = TRUE)`. SMDs are reported as absolute values and need a two-level `by`; numeric variables use the difference in means over the square root of the unweighted average of the two group variances, and categorical variables the multinomial generalisation of Yang & Dalton (2012). With `paired = TRUE`, two equal-sized groups are treated as matched pairs in row order. Use `add_effect()` to add or replace the effect-measure columns and `fmt()` to change decimals.

Value

A `simtab_result` of class `simtab_table1`. Print it to see the formatted table, convert it with `as.data.frame()`, or save it with `export_docx()`, `export_pptx()`, or `export_xlsx()`. Unrounded results are stored in `$data`.

Terse flags

Flags may be supplied as bare words in `...` after `by`, or programmatically with `flags =`. Anything else in `...` is treated as a typo.

`col` Column percentages (the default).

`row`, `cell`, `perc` Not yet supported by `table1()`.

`pr`, `rr`, `or` Add the corresponding crude effect measure.

`p` Add a p-value from an automatically chosen test, like `test = TRUE`.

`miss` Show missing values.

The old flags `rp` and `m` still work as aliases for `pr` and `miss` but are deprecated.

References

Austin, P. C. (2009). Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine*, 28(25), 3083–3107. doi:10.1002/sim.3697.

Barros, A. J. D., & Hirakata, V. N. (2003). Alternatives for logistic regression in cross-sectional studies: an empirical comparison of models that directly estimate the prevalence ratio. *BMC Medical Research Methodology*, 3, 21. doi:10.1186/14712288321.

Campbell, I. (2007). Chi-squared and Fisher-Irwin tests of two-by-two tables with small sample recommendations. *Statistics in Medicine*, 26(19), 3661–3675. doi:10.1002/sim.2832.

Katz, D., Baptista, J., Azen, S. P., & Pike, M. C. (1978). Obtaining confidence intervals for the risk ratio in cohort studies. *Biometrics*, 34(3), 469–474. doi:10.2307/2530610.

Woolf, B. (1955). On estimating the relation between blood group and disease. *Annals of Human Genetics*, 19(4), 251–253. doi:10.1111/j.14691809.1955.tb01348.x.

Zou, G. (2004). A modified Poisson regression approach to prospective studies with binary data. *American Journal of Epidemiology*, 159(7), 702–706. doi:10.1093/aje/kwh090.

See Also

`tb()` for a single cross-tabulation, `add_effect()` and `test()` to modify the table, `journal_style()` for formatting, and simtablr_references for all references cited by SimtablR.

Examples

```
# Describe the whole cohort
table1(epitabl, c(age, sex, smoking, bmi))

# Compare groups, with p-values
table1(epitabl, c(age, sex, smoking), by = adjudicated_acs, flags = "p")

# Crude and age- and sex-adjusted prevalence ratios
table1(
  epitabl, c(smoking, diabetes, hypertension),
  by = adjudicated_acs, measure = "PR", adjust = c(age, sex)
)

# Add standardized mean differences after the fact
table1(epitabl, c(age, sex), by = adjudicated_acs, test = TRUE) |>
  test(smd = TRUE)
```

---

tb *Cross-tabulate one or two variables*

Description

`tb()` builds a frequency table for one variable, or a cross-tabulation of two, with optional percentages, an association test, and crude effect measures (PR, RR, or OR). Put the exposure first (rows) and the outcome second (columns); a numeric first variable is summarised as mean (SD) or median (IQR) within each outcome group. For many variables at once use `table1()`; for adjusted estimates use `regtab()`.

Usage

```
tb(
  data,
  ...,
  m = FALSE,
  miss = FALSE,
  d = 1,
  big_mark = "",
  decimal_mark = ".",
  style = "n_pct",
  style.rp = "{rp} ({lower} - {upper})",
```

```
  style.or = "{or} ({lower} - {upper})",
  test = FALSE,
  subset = NULL,
  strat = NULL,
  rp = FALSE,
  or = FALSE,
  ref = NULL,
  conf.level = 0.95,
  var.type = NULL,
  summary = "auto",
  flags = NULL,
  labels = NULL,
  measure = NULL,
  design = NULL
)
```

Arguments

data	A data frame, or a single vector to tabulate on its own.
...	One or two variables to tabulate: bare names, strings, or tidyselect expressions such as `all_of(v)`. The first becomes the rows and the second the columns. Terse flags such as `row` or `or` may follow; see *Terse flags*.
m	Deprecated; use `miss` instead.
miss	Logical. If `TRUE`, show missing values as their own row and column. Same as the `miss` flag.
d	Integer. Decimal places for percentages and continuous summaries. Effect measures always use two decimals.
big_mark	String inserted between every three digits, e.g. `","` prints `4391` as `4,391`.
decimal_mark	String used as the decimal point, e.g. `","`. Must differ from `big_mark`.
style	String. How counts and percentages are shown: `"n_pct"` (`12 (5.0%)`), `"pct_n"` (`5.0% (12)`), or a template using `{n}` and `{p}`, such as `"{n} [{p}%]"`.
style.rp	String template for prevalence and risk ratios, using `{rp}`, `{lower}`, and `{upper}`.
style.or	String template for odds ratios, using `{or}`, `{lower}`, and `{upper}`.
test	Logical or string. `TRUE` adds a p-value from an automatically chosen test; or force `"chisq"`, `"fisher"`, or `"mcnemar"`. See *Statistical methods* for how the test is chosen.
subset	A logical expression evaluated in `data` to keep only some rows, e.g. `subset = age >= 65`.
strat	A variable to stratify by: a bare name or string. The table is repeated within each stratum and, if an effect measure is requested, a Mantel-Haenszel pooled estimate is added.
rp	Deprecated; use the `pr` flag or `measure = "PR"` instead.
or	Logical. If `TRUE`, add odds ratios. Same as the `or` flag.
ref	Reference level of the row (exposure) variable for effect measures, given as a level name or its position. If `NULL`, the first level is used and a note is printed.

`conf.level`	Number between 0 and 1. Confidence level for effect measure intervals.
`var.type`	Force variable types: `"continuous"` or `"categorical"`, either one string for all variables or a named vector such as `c(score = "continuous")`. If `NULL`, types are detected automatically (see *Statistical methods*).
`summary`	String. Summary for numeric variables: `"auto"` chooses between `"mean"` (mean and SD) and `"median"` (median and IQR) based on sample size and skewness.
`flags`	Character vector of terse flags, e.g. `c("row", "or", "p")`. The programmatic form of the bare flags in `...`.
`labels`	Named character vector of display labels, e.g. `c(smoking = "Smoking status")`. Variable labels already stored in `data` are used by default.
`measure`	String. Effect measure to add: `"PR"`, `"RR"`, or `"OR"`. Overrides any measure implied by `design`.
`design`	String. Study design used to choose the effect measure when none is requested: `"cross_sectional"` gives PR, `"cohort"` gives RR, and `"case_control"` gives OR.

Details

Statistical methods:

Effect measures compare each row level with the reference level (`ref`), taking the last column level as the event. Prevalence and risk ratios use the Katz log interval (Katz et al., 1978) and odds ratios the Woolf logit interval (Woolf, 1955). With `strat`, stratum-specific tables are pooled with the Mantel-Haenszel estimator and tested with the Cochran-Mantel-Haenszel test. Homogeneity across strata is tested with the Breslow-Day test for odds ratios and, for prevalence and risk ratios, with Cochran's Q on the inverse-variance weighted stratum log ratios. A stratified numeric variable is compared within strata: by the F-test for the group term in a linear model with stratum when summarised by the mean, and by the van Elteren stratified Wilcoxon test (van Elteren, 1960) for two groups when summarised by the median.

With `test = TRUE`, 2x2 tables use the N-1 chi-squared test (Campbell, 2007) and larger tables the Pearson chi-squared test, without continuity correction. Fisher's exact test is chosen automatically only when an expected count is below 1; request it with `test = "fisher"`. Numeric variables are compared with the t-test or ANOVA when summarised by the mean, and with the Wilcoxon or Kruskal-Wallis test when summarised by the median.

Numeric variables are treated as continuous unless they look like coded categories: exactly two distinct whole numbers, or at most seven distinct whole numbers with at least 20 observations. Use `var.type` to override.

Missing data:

Missing values are excluded from counts, percentages, tests, and effect measures. Use the `miss` flag to display them as a separate category.

Modifying the result:

Paired tests and multiplicity adjustment are set afterwards with `test()`, e.g. `tb(df, value, group, test = TRUE) |> test(paired = TRUE)`. Use `fmt()` to change decimals and `rbind()` to stack several `tb()` tables that share the same column variable.

Value

A `simtab_result` of class `simtab_tb`. Print it to see the formatted table, convert it with `as.data.frame()`, or save it with `export_docx()`, `export_pptx()`, or `export_xlsx()`. Unrounded results are stored in `$data`.

Terse flags

Flags may be supplied as bare words in `...` alongside the selected variables, or programmatically with `flags =`.

`row` Row percentages.

`col` Column percentages.

`cell`, `perc` Percentages of the table total. `perc` reads better in one-variable tables.

`pr`, `rr`, `or` Add the corresponding crude effect measure.

`p` Add a p-value from an automatically chosen test, like `test = TRUE`.

`miss` Show missing values.

The old flags `rp` and `m` still work as aliases for `pr` and `miss` but are deprecated.

References

Campbell, I. (2007). Chi-squared and Fisher-Irwin tests of two-by-two tables with small sample recommendations. *Statistics in Medicine*, 26(19), 3661–3675. doi:10.1002/sim.2832.

Katz, D., Baptista, J., Azen, S. P., & Pike, M. C. (1978). Obtaining confidence intervals for the risk ratio in cohort studies. *Biometrics*, 34(3), 469–474. doi:10.2307/2530610.

Woolf, B. (1955). On estimating the relation between blood group and disease. *Annals of Human Genetics*, 19(4), 251–253. doi:10.1111/j.14691809.1955.tb01348.x.

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine, Series 5*, 50(302), 157–175. doi:10.1080/14786440009463897.

Fisher, R. A. (1935). *The Design of Experiments*. Oliver & Boyd.

See Also

`table1()` for many variables at once, `regtab()` for adjusted effect measures, `diag_test()` for diagnostic accuracy, and simtablr_references for all references cited by SimtablR.

Examples

```
# Frequencies of one variable
tb(epitabl, smoking)

# Row percentages, a p-value, and crude prevalence ratios
tb(epitabl, smoking, adjudicated_acs, flags = c("row", "pr", "p"), ref = "Never")

# The same request with bare flags (shorthand)
tb(epitabl, smoking, adjudicated_acs, row, pr, p, ref = "Never")
```

```
# Odds ratio stratified by sex, with a Mantel-Haenszel pooled estimate
tb(epitabl, renal_impairment, adjudicated_acs, strat = sex, flags = "or", ref = "No")

# A numeric variable summarised by group
tb(epitabl, age, adjudicated_acs, test = TRUE)
```

`test` *Set a comparison test*

Description

Set a comparison test

Usage

```
test(spec, method = "auto", p.adjust = NULL, paired = NULL, smd = NULL)
```

Arguments

`spec`	A `simtab_spec`.
`method`	Test method. Defaults to `"auto"`. Categorical methods are `"chisq"`, `"fisher"`, `"mcnemar"`, and `"trend"`; continuous methods are `"t"`, `"wilcoxon"`, `"anova"`, and `"kruskal"`. Use `"none"` or `FALSE` to clear the comparison test. When `method` is omitted but `p.adjust`, `paired`, or `smd` is supplied, the existing test choice is kept, so `table1(...) |> test(smd = TRUE)` adds SMDs without adding p-values.
`p.adjust`	Multiplicity adjustment method, or `NULL` to leave unchanged.
`paired`	Logical; whether paired tests are requested.
`smd`	Logical; whether standardized mean differences are requested.

Value

A modified `simtab_spec`.

Examples

```
test(stratify(describe(simtab(epitabl), age), adjudicated_acs), "wilcoxon")
```

tidy.simtab_result *Tidy a SimtablR result*

Description

Tidy a SimtablR result

Usage

```
## S3 method for class 'simtab_result'
tidy(x, ...)

## S3 method for class 'simtab_spec'
tidy(x, ...)
```

Arguments

x	A simtab_result or simtab_spec.
...	Ignored.

Value

A long data.frame. For table1, this matches as.data.frame(x, tidy = TRUE).

Examples

```
res <- tb(epitabl, sex, diabetes)
generics::tidy(res)
```

validate *Validate a SimtablR specification before compute*

Description

Keeps only engine-agnostic checks: spec shape and effect-measure existence. Per-engine hard errors live in that engine's own validate hook, registered via register_engine(), and run by evaluate() after this shared check.

Usage

```
validate(spec)
```

Arguments

spec	A simtab_spec.

Value

Invisibly, spec.

Examples

```
sp <- simtab(epitabl) |> describe(hypertension)
validate(sp)
```

---

why *Report why SimtablR made each recorded decision*

Description

why() prints the decisions already recorded on a spec, result, or report. Passing a simtab_spec is inert: it describes the planned analysis without computing it.

Usage

```
why(x, ...)
```

Arguments

x	A simtab_spec, simtab_result, or simtab_report.
...	Ignored.

Value

A simtab_explanation object, invisibly when printed.

Examples

```
res <- tb(epitabl, sex, diabetes)
why(res)
```

Index

∗ **datasets**
epitabl, 19
∗ **errors**
simtab_errors, 52

add_effect, 3
add_effect(), *65, 66*
adjust, 4
advise, 5
advise(), *58*
as.data.frame.simtab_audit, 5
as.data.frame.simtab_rbind_tb, 6
as.data.frame.simtab_report, 7
as.data.frame.simtab_result, 7
as.data.frame.simtab_spec *(*as.data.frame.simtab_result*)*, 7
as_flextable.simtab_checklist *(*strobe*)*, 59
as_flextable.simtab_codebook, 8
as_flextable.simtab_flow *(*flow*)*, 27
as_flextable.simtab_rbind_tb, 9
as_flextable.simtab_report *(*as_flextable.simtab_result*)*, 10
as_flextable.simtab_report(), *7*
as_flextable.simtab_result, 10
as_flextable.simtab_sensitivity *(*sensitivity*)*, 48
as_flextable.simtab_spec *(*as_flextable.simtab_result*)*, 10
as_gt, 10
as_gt.simtab_report(), *7*
as_methods, 11
as_methods(), *22*
autoplot.simtab_flow *(*flow*)*, 27
autoplot.simtab_result, 12
autoplot.simtab_spec, 13

codebook, 13
coef.simtab_result *(*model-accessors*)*, 35
confint.simtab_result *(*model-accessors*)*, 35

describe, 14
design, 15
diag_test, 15
diag_test(), *46*, *48*, *51*, *69*

e_value, 18
engine, 19
epitabl, 19
evaluate, 21
evaluate(), *19*, *41*, *71*
export_docx, 22
export_docx(), *17*, *23, 24*, *27*, *45*, *47*, *62*, *65*, *69*
export_plot, 23
export_pptx, 24
export_pptx(), *17*, *22, 23*, *27*, *45*, *47*, *62*, *65*, *69*
export_regtab_csv, 25
export_regtab_xlsx, 26
export_xlsx, 26
export_xlsx(), *17*, *22*, *24*, *26*, *45*, *47*, *62*, *65*, *69*

flextable::flextable(), *8*
flow, 27
fmt, 28
fmt(), *17*, *47*, *65*, *68*
formula.simtab_result *(*model-accessors*)*, 35

glance.simtab_result, 29
glance.simtab_spec *(*glance.simtab_result*)*, 29
graphics::fourfoldplot(), *38*

is_simtab, 29

is_simtab(), *42*

journal_style, 30
journal_style(), *33*, *42*, *64*, *66*

label, 32
label(), *28*
list_engines, 32
list_engines(), *19*, *42*
list_journals, 33
list_journals(), *31*, *42*, *61*, *64*

measure, 33
missingness, 34
model-accessors, 35
model_info, 36
model_info(), *44, 45*, *61, 62*

nobs.simtab_result *(*model-accessors*)*, 35

overall, 37

plot.simtab_diag, 38
plot.simtab_diag(), *17*
print.simtab_rbind_tb, 38
print.simtab_result, 39

rbind(), *68*
rbind.simtab_tb, 40
register_engine, 41
register_engine(), *19*, *33*, *71*
register_journal, 42
register_journal(), *30, 31*, *33*
regtab, 43
regtab(), *51*, *60*, *62*, *66*, *69*
repro_manifest, 45
roc, 46
roc(), *15*, *17*, *51*

sensitivity, 48
sensitivity(), *7*
set_design, 49
set_design(), *46*
set_summary, 50
simtab, 50
simtab_completions, 51
simtab_errors, 52
simtablr, 53
simtablr_guidance, 54
simtablr_guidance(), *5*, *58*
simtablr_references, *5*, *17*, *45*, *48*, *54*, 55, *62*, *66*, *69*
stard, 58
stats::confint(), *35*
stats::family(), *43*
stats::glm(), *36*, *44*
stats::nobs(), *35*
stats::vcov(), *35*
stratify, 59
strobe, 59
style, 60
survtab, 60
survtab(), *45*, *51*

table1, 62
table1(), *3, 4*, *22*, *30, 31*, *42, 43*, *45*, *51*, *66*, *69*
tb, 66
tb(), *22*, *28*, *40*, *42*, *51*, *62*, *64*, *66*
test, 70
test(), *4*, *65, 66*, *68*
tidy.simtab_result, 71
tidy.simtab_spec *(*tidy.simtab_result*)*, 71

utils::write.csv(), *25*

validate, 71
vcov.simtab_result *(*model-accessors*)*, 35

why, 72