

# Package ‘modelsummary’

June 13, 2020

**Type** Package

**Title** Summary Tables for Statistical Models: Beautiful, Customizable, and Publication-Ready

**Description** Create beautiful and customizable tables to summarize several statistical models side-by-side. This package supports dozens of model types and can produce tables in HTML, LaTeX, Markdown, Word, PowerPoint, Excel, RTF, JPG, or PNG. Tables can easily be embedded in 'Rmarkdown' or 'knitr' dynamic documents.

**Version** 0.4.0

**URL** <https://vincentarelbundock.github.io/modelsummary>

**BugReports** <https://github.com/vincentarelbundock/modelsummary/issues>

**Depends** R (>= 3.4.0)

**Imports** broom,  
checkmate,  
dplyr,  
generics,  
gt (>= 0.2.0),  
knitr (>= 1.16),  
magrittr,  
kableExtra,  
purrr,  
rmarkdown (>= 1.6.0),  
stringr,  
tibble,  
tidyr (>= 1.0.0)

**Suggests** broom.mixed,  
flextable,  
huxtable,  
lmtest,  
MASS,  
officer,  
sandwich,  
testthat

**License** GPL-3

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 7.1.0.9000

R topics documented:

clean_latex . . . . .	2
extract . . . . .	2
glance_custom . . . . .	4
gof_map . . . . .	5
knit_latex . . . . .	5
modelsummary . . . . .	6
msummary . . . . .	8
tidy_custom.default . . . . .	11
<b>Index</b>	<b>12</b>

---

clean_latex	<i>Deprecated function</i>
-------------	----------------------------

---

Description

The ‘gt::as\_latex’ function does not (yet) produce compilable LaTeX code. This function used to clean up LaTeX output to allow compilation to PDF. This function is now deprecated since ‘modelsummary’ currently supports ‘kableExtra’, which has a mature LaTeX rendering engine.

Usage

```
clean_latex(...)
```

Arguments

... catch everything

---

extract	<i>Extract and combine estimates and goodness-of-fit statistics from several statistical models.</i>
---------	------------------------------------------------------------------------------------------------------

---

Description

Extract and combine estimates and goodness-of-fit statistics from several statistical models.

Usage

```
extract(  
  models,  
  statistic = "std.error",  
  statistic_override = NULL,  
  statistic_vertical = TRUE,  
  conf_level = 0.95,  
  coef_map = NULL,  
  coef_omit = NULL,  
  gof_map = modelsummary::gof_map,  
  gof_omit = NULL,  
  add_rows = NULL,
```

```

    add_rows_location = NULL,
    stars = FALSE,
    fmt = "%.3f",
    estimate = "estimate",
    ...
  )

```

## Arguments

<code>models</code>	a single model object or a (potentially named) list of models to summarize
<code>statistic</code>	string name of the statistic to include in parentheses <ul style="list-style-type: none"> <li>• Typical values: "conf.int", "std.error", "statistic", "p.value"</li> <li>• Alternative values: any column name produced by <code>'broom::tidy(model)'</code></li> </ul>
<code>statistic_override</code>	manually override the uncertainty estimates. This argument accepts three types of input: <ul style="list-style-type: none"> <li>• a function or list of functions of <code>length(models)</code> which produce variance-covariance matrices with row and column names equal to the names of your coefficient estimates. For example, <code>'R'</code> supplies the <code>'vcov'</code> function, and the <code>'sandwich'</code> package supplies <code>'vcovHC'</code>, <code>'vcovHAC'</code>, etc.</li> <li>• a list of <code>length(models)</code> variance-covariance matrices with row and column names equal to the names of your coefficient estimates.</li> <li>• a list of <code>length(models)</code> vectors with names equal to the names of your coefficient estimates. Numeric vectors are formatted according to <code>'fmt'</code> and placed in brackets, character vectors printed as given.</li> </ul>
<code>statistic_vertical</code>	TRUE if statistics should be printed below estimates. FALSE if statistics should be printed beside estimates.
<code>conf_level</code>	confidence level to use for confidence intervals
<code>coef_map</code>	named character vector. Names refer to the original variable names. Values refer to the variable names that will appear in the table. Coefficients which are omitted from this vector will be omitted from the table. The table will be ordered in the same order as this vector.
<code>coef_omit</code>	string regular expression. Omits all matching coefficients from the table (using <code>'stringr::str_detect'</code> ).
<code>gof_map</code>	data.frame with four columns: <code>'raw'</code> , <code>'clean'</code> , <code>'fmt'</code> , and <code>'omit'</code> . See <code>'model-summary::gof_map'</code>
<code>gof_omit</code>	string regular expression. Omits all matching gof statistics from the table (using <code>'stringr::str_detect'</code> ).
<code>add_rows</code>	list of character vectors, each of length equal to the number of models + 1.
<code>add_rows_location</code>	integer or NULL. custom rows will be added to the bottom of the table if this parameter is NULL, or after the position set by this integer.
<code>stars</code>	to indicate statistical significance <ul style="list-style-type: none"> <li>• FALSE (default): no significance stars.</li> <li>• TRUE: <code>*=.1</code>, <code>**=.05</code>, <code>***=.01</code></li> <li>• Named numeric vector for custom stars such as <code>'c('*' = .1, '+' = .05)'</code></li> </ul>

fmt	string which specifies how numeric values will be rounded. This string is passed to the ‘sprintf’ function. ‘%.3f’ will keep 3 digits after the decimal point with trailing zero. ‘%.5f’ will keep 5 digits. ‘%.3e’ will use exponential notation. See ‘?sprintf’ for more options.
estimate	character name of the estimate to display. Must be a column name in the dataframe produced by ‘tidy(model)’. In the vast majority of cases, the default value of this argument should not be changed.
...	all other arguments are passed to the ‘tidy’ method used to extract estimates from the model. For example, this allows users to set ‘exponentiate=TRUE’ to exponentiate logistic regression coefficients.

## Value

tibble

## Examples

```
library(modelsummary)
data(trees)
models <- list()
models[['Bivariate']] <- lm(Girth ~ Height, data = trees)
models[['Multivariate']] <- lm(Girth ~ Height + Volume, data = trees)
extract(models)
```

---

glance_custom	<i>Extract custom information from a model object and turn it into a tidy tibble with a single row.</i>
---------------	---------------------------------------------------------------------------------------------------------

---

## Description

glance\_custom methods always return either a one-row data frame (except on ‘NULL’, which returns an empty data frame). This

## Usage

```
glance_custom(x)
```

## Arguments

x                      model or other R object to convert to single-row data frame

## Methods

No methods found in currently loaded packages.

---

gof\_map

*Data.frame used to clean up and format goodness-of-fit statistics*


---

### Description

By default, this data frame is passed to the 'gof\_map' argument of the 'msummary' or 'modelsummary' functions. Users can modify this data frame to customize the list of statistics to display and their format. See example below.

### Usage

```
gof_map
```

### Format

data.frame with 4 columns of character data: raw, clean, fmt, omit

### Examples

```
library(modelsummary)
mod <- lm(wt ~ drat, data = mtcars)
gm <- modelsummary::gof_map
gm$omit[gm$raw == 'deviance'] <- FALSE
gm$fmt[gm$raw == 'r.squared'] <- "%.5f"
msummary(mod, gof_map = gm)
```

---

knit\_latex

*Deprecated function*


---

### Description

The 'gt::as\_latex' function does not (yet) produce compilable LaTeX code. This function used to clean up LaTeX output to allow compilation to PDF. This function is now deprecated since 'modelsummary' currently supports 'kableExtra', which has a mature LaTeX rendering engine.

### Usage

```
knit_latex(...)
```

### Arguments

... catch everything

**Description**

Beautiful, customizable summaries of statistical models

**Usage**

```
modelssummary(
  models,
  output = "default",
  fmt = "%.3f",
  statistic = "std.error",
  statistic_override = NULL,
  statistic_vertical = TRUE,
  conf_level = 0.95,
  stars = FALSE,
  coef_map = NULL,
  coef_omit = NULL,
  gof_map = modelssummary::gof_map,
  gof_omit = NULL,
  add_rows = NULL,
  add_rows_location = NULL,
  title = NULL,
  notes = NULL,
  estimate = "estimate",
  filename = NULL,
  subtitle = NULL,
  ...
)
```

**Arguments**

- |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| models | a single model object or a (potentially named) list of models to summarize                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| output | filename or object type (string) <ul style="list-style-type: none"> <li>• Supported filename extensions: .html, .tex, .md, .txt, .png, .jpg.</li> <li>• Supported object types: "default", "html", "markdown", "latex", "gt", "kable-Extra", "huxtable", "flextable".</li> <li>• When a file name is supplied to the 'output' argument, the table is written immediately to file. If you want to customize your table by post-processing it with functions provided by the 'gt' or 'kableExtra' packages, you need to choose a different output format (e.g., "gt", "latex", "html", "markdown"), and you need to save the table after post-processing using the 'gt::gtsave', 'kable::save_kable', or 'cat' functions.</li> </ul> |
| fmt    | string which specifies how numeric values will be rounded. This string is passed to the 'sprintf' function. '%.3f' will keep 3 digits after the decimal point with trailing zero. '%.5f' will keep 5 digits. '%.3e' will use exponential notation. See '?sprintf' for more options.                                                                                                                                                                                                                                                                                                                                                                                                                                                |

statistic	string name of the statistic to include in parentheses <ul style="list-style-type: none"> <li>• Typical values: "conf.int", "std.error", "statistic", "p.value"</li> <li>• Alternative values: any column name produced by 'broom::tidy(model)'</li> </ul>
statistic_override	manually override the uncertainty estimates. This argument accepts three types of input: <ul style="list-style-type: none"> <li>• a function or list of functions of length(models) which produce variance-covariance matrices with row and column names equal to the names of your coefficient estimates. For example, 'R' supplies the 'vcov' function, and the 'sandwich' package supplies 'vcovHC', 'vcovHAC', etc.</li> <li>• a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates.</li> <li>• a list of length(models) vectors with names equal to the names of your coefficient estimates. Numeric vectors are formatted according to 'fmt' and placed in brackets, character vectors printed as given.</li> </ul>
statistic_vertical	TRUE if statistics should be printed below estimates. FALSE if statistics should be printed beside estimates.
conf_level	confidence level to use for confidence intervals
stars	to indicate statistical significance <ul style="list-style-type: none"> <li>• FALSE (default): no significance stars.</li> <li>• TRUE: *=.1, **=.05, ***=.01</li> <li>• Named numeric vector for custom stars such as 'c('*' = .1, '+' = .05)'</li> </ul>
coef_map	named character vector. Names refer to the original variable names. Values refer to the variable names that will appear in the table. Coefficients which are omitted from this vector will be omitted from the table. The table will be ordered in the same order as this vector.
coef_omit	string regular expression. Omits all matching coefficients from the table (using 'stringr::str_detect').
gof_map	data.frame with four columns: 'raw', 'clean', 'fmt', and 'omit'. See 'modelssummary::gof_map'
gof_omit	string regular expression. Omits all matching gof statistics from the table (using 'stringr::str_detect').
add_rows	list of character vectors, each of length equal to the number of models + 1.
add_rows_location	integer or NULL. custom rows will be added to the bottom of the table if this parameter is NULL, or after the position set by this integer.
title	string
notes	list or vector of notes to append to the bottom of the table.
estimate	character name of the estimate to display. Must be a column name in the dataframe produced by 'tidy(model)'. In the vast majority of cases, the default value of this argument should not be changed.
filename	This argument was deprecated in favor of the 'output' argument.
subtitle	This argument is deprecated. Use 'title' or the 'tab_header' function from the 'gt' package.
...	all other arguments are passed to the 'tidy' method used to extract estimates from the model. For example, this allows users to set 'exponentiate=TRUE' to exponentiate logistic regression coefficients.

**Value**

a 'gt' table object.

**Examples**

```
# load data and estimate models
data(trees)
models <- list()
models[['Bivariate']] <- lm(Girth ~ Height, data = trees)
models[['Multivariate']] <- lm(Girth ~ Height + Volume, data = trees)

# simple table
msummary(models)

# confidence intervals, p values, or t-stats instead of standard errors
msummary(models, statistic = 'conf.int', conf_level = 0.99)
msummary(models, statistic = 'p.value', conf_level = 0.99)
msummary(models, statistic = 'statistic', conf_level = 0.99)

# rename and re-order coefficients
msummary(models, coef_map = c('Volume' = 'Large', 'Height' = 'Tall'))

# titles
msummary(models, title = 'This is the title')

# title with italicized text
msummary(models, title = gt::md('This is the title'))

# notes at the bottom of the table (here, the second note includes markdown bold characters)
msummary(models, notes = list('A first note', gt::md('A bold note')))

# modify list of GOF statistics and their format using the built-in
# 'gof_map' data frame as a starting point
gof_custom <- modelsummary::gof_map
gof_custom$omit[gof_custom$raw == 'deviance'] <- FALSE
gof_custom$fmt[gof_custom$raw == 'r.squared'] <- "%.5f"
msummary(models, gof_map = gof_custom)
```

---

msummary

*Beautiful, customizable summaries of statistical models*


---

**Description**

‘msummary()’ is a shortcut to ‘modelsummary()’

**Usage**

```
msummary(
  models,
  output = "default",
  fmt = "%.3f",
```



```

statistic = "std.error",
statistic_override = NULL,
statistic_vertical = TRUE,
conf_level = 0.95,
stars = FALSE,
coef_map = NULL,
coef_omit = NULL,
gof_map = modelsummary::gof_map,
gof_omit = NULL,
add_rows = NULL,
add_rows_location = NULL,
title = NULL,
notes = NULL,
estimate = "estimate",
filename = NULL,
subtitle = NULL,
...
)

```

## Arguments

models	a single model object or a (potentially named) list of models to summarize
output	filename or object type (string) <ul style="list-style-type: none"> <li>Supported filename extensions: .html, .tex, .md, .txt, .png, .jpg.</li> <li>Supported object types: "default", "html", "markdown", "latex", "gt", "kableExtra", "huxtable", "flextable".</li> <li>When a file name is supplied to the 'output' argument, the table is written immediately to file. If you want to customize your table by post-processing it with functions provided by the 'gt' or 'kableExtra' packages, you need to choose a different output format (e.g., "gt", "latex", "html", "markdown"), and you need to save the table after post-processing using the 'gt::gtsave', 'kable::save_kable', or 'cat' functions.</li> </ul>
fmt	string which specifies how numeric values will be rounded. This string is passed to the 'sprintf' function. '%.3f' will keep 3 digits after the decimal point with trailing zero. '%.5f' will keep 5 digits. '%.3e' will use exponential notation. See '?sprintf' for more options.
statistic	string name of the statistic to include in parentheses <ul style="list-style-type: none"> <li>Typical values: "conf.int", "std.error", "statistic", "p.value"</li> <li>Alternative values: any column name produced by 'broom::tidy(model)'</li> </ul>
statistic_override	manually override the uncertainty estimates. This argument accepts three types of input: <ul style="list-style-type: none"> <li>a function or list of functions of length(models) which produce variance-covariance matrices with row and column names equal to the names of your coefficient estimates. For example, 'R' supplies the 'vcov' function, and the 'sandwich' package supplies 'vcovHC', 'vcovHAC', etc.</li> <li>a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates.</li> <li>a list of length(models) vectors with names equal to the names of your coefficient estimates. Numeric vectors are formatted according to 'fmt' and placed in brackets, character vectors printed as given.</li> </ul>

<code>statistic_vertical</code>	TRUE if statistics should be printed below estimates. FALSE if statistics should be printed beside estimates.
<code>conf_level</code>	confidence level to use for confidence intervals
<code>stars</code>	to indicate statistical significance <ul style="list-style-type: none"> <li>• FALSE (default): no significance stars.</li> <li>• TRUE: *=.1, **=.05, ***=.01</li> <li>• Named numeric vector for custom stars such as <code>c('*' = .1, '+' = .05)</code></li> </ul>
<code>coef_map</code>	named character vector. Names refer to the original variable names. Values refer to the variable names that will appear in the table. Coefficients which are omitted from this vector will be omitted from the table. The table will be ordered in the same order as this vector.
<code>coef_omit</code>	string regular expression. Omits all matching coefficients from the table (using <code>'stringr::str_detect'</code> ).
<code>gof_map</code>	data.frame with four columns: <code>'raw'</code> , <code>'clean'</code> , <code>'fmt'</code> , and <code>'omit'</code> . See <code>'model-summary::gof_map'</code>
<code>gof_omit</code>	string regular expression. Omits all matching gof statistics from the table (using <code>'stringr::str_detect'</code> ).
<code>add_rows</code>	list of character vectors, each of length equal to the number of models + 1.
<code>add_rows_location</code>	integer or NULL. custom rows will be added to the bottom of the table if this parameter is NULL, or after the position set by this integer.
<code>title</code>	string
<code>notes</code>	list or vector of notes to append to the bottom of the table.
<code>estimate</code>	character name of the estimate to display. Must be a column name in the dataframe produced by <code>'tidy(model)'</code> . In the vast majority of cases, the default value of this argument should not be changed.
<code>filename</code>	This argument was deprecated in favor of the <code>'output'</code> argument.
<code>subtitle</code>	This argument is deprecated. Use <code>'title'</code> or the <code>'tab_header'</code> function from the <code>'gt'</code> package.
<code>...</code>	all other arguments are passed to the <code>'tidy'</code> method used to extract estimates from the model. For example, this allows users to set <code>'exponentiate=TRUE'</code> to exponentiate logistic regression coefficients.

## Value

a `'gt'` table object.

## Examples

```
# load data and estimate models
data(trees)
models <- list()
models[['Bivariate']] <- lm(Girth ~ Height, data = trees)
models[['Multivariate']] <- lm(Girth ~ Height + Volume, data = trees)

# simple table
msummary(models)
```

```

# confidence intervals, p values, or t-stats instead of standard errors
msummary(models, statistic = 'conf.int', conf_level = 0.99)
msummary(models, statistic = 'p.value', conf_level = 0.99)
msummary(models, statistic = 'statistic', conf_level = 0.99)

# rename and re-order coefficients
msummary(models, coef_map = c('Volume' = 'Large', 'Height' = 'Tall'))

# titles
msummary(models, title = 'This is the title')

# title with italicized text
msummary(models, title = gt::md('This is the title'))

# notes at the bottom of the table (here, the second note includes markdown bold characters)
msummary(models, notes = list('A first note', gt::md('A bold note')))

# modify list of GOF statistics and their format using the built-in
# 'gof_map' data frame as a starting point
gof_custom <- modelsummary::gof_map
gof_custom$omit[gof_custom$raw == 'deviance'] <- FALSE
gof_custom$fmt[gof_custom$raw == 'r.squared'] <- "%.5f"
msummary(models, gof_map = gof_custom)

```

---

tidy_custom.default	<i>Extract custom information from a model object and turn it into a tidy tibble</i>
---------------------	--------------------------------------------------------------------------------------

---

## Description

Extract custom information from a model object and turn it into a tidy tibble

## Usage

```
## Default S3 method:
tidy_custom(x)
```

## Arguments

**x** An object to be converted into a tidy [tibble::tibble()].

## Value

A [tibble::tibble()] with information about model components.

# Index

## \* **datasets**

gof\_map, [5](#)

clean\_latex, [2](#)

extract, [2](#)

glance\_custom, [4](#)

gof\_map, [5](#)

knit\_latex, [5](#)

modelsummary, [6](#)

msummary, [8](#)

tidy\_custom.default, [11](#)