Package ‘modelsummary’

April 5, 2020

Type Package

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

Description Create beautiful, customizable, publication-ready summary tables for statistical models. 'modelsummary' leverages the power of the 'gt' and 'broom' packages. It can produce tables in HTML, RTF, JPG, and LaTeX formats (text/markdown/ascii tables coming soon).

Version 0.2.0

URL https://github.com/vincentarelbundock/modelsummary

BugReports https://github.com/vincentarelbundock/modelsummary/issues

Depends R (>= 3.4.0)

Imports dplyr (>= 0.7.0), generics (>= 0.0.2), broom (>= 0.5.1), tidyr (>= 0.8.0), stringr (>= 1.3.0), purrr (>= 0.2.1), checkmate (>= 1.8.5), magrittr (>= 1.5), tibble (>= 1.4.2), gt (>= 0.2.0)

Suggests testthat, sandwich, lmtest, knitr, MASS, mice

License GPL-3

Encoding UTF-8

LazyData false

RoxygenNote 7.1.0

NeedsCompilation no

Author Vincent Arel-Bundock [aut, cre]

Maintainer Vincent Arel-Bundock <vincent.arel-bundock@umontreal.ca>

Repository CRAN

Date/Publication 2020-04-05 16:20:02 UTC

R topics documented:

clean_latex .................................................. 2
extract ....................................................... 3
Utility function to cleanup \LaTeX output from gt and ensures that it compiles with \LaTeX

Description

The `gt::as_latex` function is still in development, rather feature poor, and prone to breakage. This function is a stopgap measure which adds a little functionality and "cleans-up" some of the LaTeX output to avoid common compilation errors. In time, as upstream improves, the goal is to deprecate this function.

Usage

```r
clean_latex(
  tab,
  label = NULL,
  latex_env = "longtable",
  gof_regex = "^Num Obs."
)
```

Arguments

- `tab`: table object produced by `modelsummary` or `gt`
- `label`: string will be inserted as a `label`
- `latex_env`: the default \LaTeX environment is `longtable`. Fix this argument to `table` if you want to use a tabular nested inside a table.
- `gof_regex`: regex which identifies the first GOF statistic. Used to figure out where to insert a midrule to separate coefficients from GOFs.

Value

a string object with \LaTeX code
Extract and combine estimates and goodness-of-fit statistics from several statistical models.

Usage

```r
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"
)
```

Arguments

- **models**: a single model object or a (potentially named) list of models to summarize
- **statistic**: string name of the statistic to include in parentheses below estimates. Must be either "conf.int", or one of the column names produced by the `broom::tidy` function. Typical values include: "std.error", "conf.int", "statistic", "p.value". A character vector will stack several uncertainty estimates on top of one another (in different rows).
- **statistic_override**: manually override the uncertainty estimates. This argument accepts three types of input:
  - 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.
  - a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates.
  - a list of length(models) numeric vectors with names equal to the names of your coefficient estimates.
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

coeff_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.

coeff_omit
   string regular expression. Omits all matching coefficients from the table (using 'stringr::str_detect').

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.

stars
   FALSE for no significance stars. TRUE for default significance stars (*=.1, **=.05, ***=.01). Named numeric vector for custom significance stars. For example, ‘c('*' = .1, '+' = .05)’

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.

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.mira

Description
Glance a multiple imputation 'mice' pooled object

Usage

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

Arguments

x An object with multiply-imputed models from 'mice' (class: 'mira')

... extra arguments (not used)

Value
a tibble with one row

Note
If x contains 'lm' models, R2 is included in the output

See Also
Other tidiers: tidy.mira()

Examples

library(mice)
data <- airquality
data[4:10,3] <- rep(NA,7)
data[1:5,4] <- NA
tmp <- mice(data,m=5, seed=500, printFlag = FALSE)
mod <- with(tmp, lm(Ozone ~ Solar.R + Wind))
glance(mod)
**gof_map**

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

**Description**

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

**Usage**

```r
gof_map
```

**Format**

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

---

**knit_latex**

*Utility function to cleanup LaTeX output from gt and ensures that it compiles with latex and that it knits well with ‘knitr’*

**Description**

The `gt::as_latex` function is still in development, rather feature poor, and prone to breakage when using `knitr`. This function is a stopgap measure which adds a little functionality and "cleans-up" some of the LaTeX output to avoid common compilation errors. In time, as upstream improves, the goal is to deprecate this function.

**Usage**

```r
knit_latex(tab, label = NULL, latex_env = "longtable")
```

**Arguments**

- `tab`: table object produced by `modelsummary` or `gt`
- `label`: string will be inserted as a `label`
- `latex_env`: the default LaTeX environment is longtable. Fix this argument to 'table' if you want to use a tabular nested inside a table.

**Details**

LaTeX compilation requires the following packages: booktabs, caption, longtable

**Value**

an object of class ‘knit_asis’. The first element of this object (`x[[1]]`) contains raw LaTeX code.
Description

Beautiful, customizable summaries of statistical models

Usage

```r
modelsummary(
  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,
  fmt = ".%3f",
  stars = FALSE,
  stars_note = TRUE,
  title = NULL,
  subtitle = NULL,
  notes = NULL,
  add_rows = NULL,
  add_rows_location = NULL,
  filename = NULL
)
```

Arguments

models  a single model object or a (potentially named) list of models to summarize

statistic  string name of the statistic to include in parentheses below estimates. Must be either "conf.int", or one of the column names produced by the `broom::tidy` function. Typical values include: "std.error", "conf.int", "statistic", "p.value". A character vector will stack several uncertainty estimates on top of one another (in different rows).

statistic_override  manually override the uncertainty estimates. This argument accepts three types of input:
  * 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.
- a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates.
- a list of length(models) numeric vectors with names equal to the names of your coefficient estimates.

**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

**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**

**gof_omit**
string regular expression. Omits all matching gof statistics from the table (using `stringr::str_detect`).

**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.

**stars**
FALSE for no significance stars. TRUE for default significance stars (‘*’=.1, ‘**’=.05, ‘***’=.01). Named numeric vector for custom significance stars. For example, ‘c(‘**’ = .1, ‘+’ = .05)’

**stars_note**
logical include a note at the bottom of the table to describe the contents of the ‘stars’ argument. The note will be omitted if ‘stars==NULL’

**title**
string

**subtitle**
string

**notes**
list of notes to append to the bottom of the table.

**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.

**filename**
the file name to create on disk. Ensure that an extension compatible with the output types is provided (‘.html’, ‘.tex’, ‘.ltx’, ‘.rtf’). Read ‘?gt::gtsave’ for further details. When the table produced by ‘modelsummary’ is post-processed by another ‘gt’ function, users need to use the ‘gtsave’ function from the ‘gt’ package; using the ‘filename’ argument will produce an error.

**Value**
a ‘gt’ table object.
**Examples**

```r
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 and subtitles
msummary(models, title = 'This is the title', subtitle = 'And a subtitle')

# 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')))
```

**msummary**

**Beautiful, customizable summaries of statistical models**

**Description**

Beautiful, customizable summaries of statistical models

**Usage**

```r
msummary(
  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,
fmt = "%.3f",
stars = FALSE,
stars_note = TRUE,
title = NULL,
subtitle = NULL,
notes = NULL,
add_rows = NULL,
add_rows_location = NULL,
filename = NULL
)

Arguments

models a single model object or a (potentially named) list of models to summarize
statistic string name of the statistic to include in parentheses below estimates. Must be either "conf.int", or one of the column names produced by the `broom::tidy` function. Typical values include: "std.error", "conf.int", "statistic", "p.value". A character vector will stack several uncertainty estimates on top of one another (in different rows).
statistic_override manually override the uncertainty estimates. This argument accepts three types of input:
- 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.
- a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates.
- a list of length(models) numeric vectors with names equal to the names of your coefficient estimates.
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
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 'model_summary::gof_map'
gof_omit string regular expression. Omits all matching gof statistics from the table (using 'stringr::str_detect').
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.

FALSE for no significance stars. TRUE for default significance stars (*=.1,
**=.05, ***=.01). Named numeric vector for custom significance stars. For example, ‘c('*' = .1, '+' = .05)’

logical include a note at the bottom of the table to describe the contents of the ‘stars’ argument. The note will be omitted if ‘stars==NULL’

string

string

list of notes to append to the bottom of the table.

list of character vectors, each of length equal to the number of models + 1.

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.

the file name to create on disk. Ensure that an extension compatible with the output types is provided (‘.html’, ‘.tex’, ‘.ltx’, ‘.rtf’). Read ‘?gt::gtsave’ for further details. When the table produced by ‘modelsummary’ is post-processed by another ‘gt’ function, users need to use the ‘gtsave’ function from the ‘gt’ package; using the ‘filename’ argument will produce an error.

a ‘gt’ table object.

# 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 and subtitles
msummary(models, title = 'This is the title', subtitle = 'And a subtitle')
# 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')))

---

**tidy.mira**  

_Tidy a multiple imputation ‘mice’ object_

**Description**

Tidy a multiple imputation ‘mice’ object

**Usage**

```r
## S3 method for class ‘mira’
tidy(x, ...)
```

**Arguments**

- `x`  
  An object with multiply-imputed models from ‘mice’ (class: ‘mira’)

- `...`  
  extra arguments (not used, unless confidence intervals are requested for multiple ‘lm’ models)

**Value**

a tibble with one row per term

**Note**

Available stats in mipo object:

- estimate
- ubar
- b
- t
- dfcom
- df
- riv
- lambda
- fmi
If `mira` object consists of `lm`-models, additional results can be returned to facilitate side-by-side tables with other `lm`-model:

- p.value
- conf.low (if called with conf.int = TRUE)
- conf.high (if called with conf.int = TRUE)

**See Also**

Other tidiers: `glance.mira()`

**Examples**

```r
library(mice)
data <- airquality
data[4:10,3] <- rep(NA,7)
data[1:5,4] <- NA
tmp <- mice(data,m=5, seed=500, printFlag = FALSE)
mod <- with(tmp, lm(Temp~ Ozone+Solar.R+Wind))
tidy(mod)
```

---

**Description**

Tidy multiple imputations of `lm` models created with `mice`

**Usage**

```r
## S3 method for class 'mira.lm'
tidy(x, conf.int = TRUE, conf.level = 0.95, ...)
```

**Arguments**

- `x` A `mira` object containing multiple lm-models based on `mice` imputations.
- `conf.int` Logical. Should confidence intervals be returned. Defaults to true.
- `conf.level` Confidence level for intervals. Defaults to .95
- `...` extra arguments (not used)
Note

Available stats in result:

- estimate
- ubar
- b
- t
- dfcom
- df
- riv
- lambda
- fmi
- p.value
- conf.low (if called with conf.int = TRUE)
- conf.high (if called with conf.int = TRUE)
Index

*Topic datasets
  gof_map, 6

clean_latex, 2

extract, 3

glance.mira, 5, 13
  gof_map, 6

knit_latex, 6

modelsummary, 7

msummary, 9

tidy.mira, 5, 12

tidy.mira.lm, 13