Package ‘modelsummary’

July 15, 2019

Type Package

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

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). The 'gt' package is hosted on 'Github' by the 'RStudio' organization: <http://github.com/rstudio/gt>.

Version 0.1.0

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)

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

License GPL-3

Encoding UTF-8

LazyData false

RoxygenNote 6.1.1

NeedsCompilation no

Author Vincent Arel-Bundock [aut, cre]

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

Repository CRAN

Date/Publication 2019-07-15 10:20:03 UTC

R topics documented:

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

Description

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

Usage

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

`statistic_override`
manually override the uncertainy 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.

`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 ‘modelsummary::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.

`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)
```

extract_estimates *Extract estimates and statistics from a single model*

Description

Extract estimates and statistics from a single model

Usage

```
extract_estimates(model, statistic = "std.error",
  statistic_override = NULL, conf_level = 0.95, fmt = "%.3f",
  stars = FALSE)
```

Arguments

`model` object type with an available ‘tidy‘ method.

`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".

`statistic_override` manually override the uncertainy 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.

`conf_level` confidence level to use for confidence intervals

`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)‘

Value

data.frame with side-by-side model summaries

extract_gof *Extract goodness-of-fit statistics from a single model*

Description

Extract goodness-of-fit statistics from a single model

Usage

```
extract_gof(model, fmt = "%.3f", gof_map = NULL)
```

Arguments

`model`	object type with an available ‘glance‘ method.
`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.
`gof_map`	data.frame with four columns: ‘raw‘, ‘clean‘, ‘fmt‘, and ‘omit‘. See ‘modelsummary::gof_map‘

Value

tibble with goodness-of-fit statistics

extract_statistic_override *Allow users to override uncertainty estimates importFrom broom tidy*

Description

Allow users to override uncertainty estimates importFrom broom tidy

Usage

```
extract_statistic_override(model, statistic_override,
  statistic = "std.error")
```

Arguments

`model` object type with an available ‘tidy‘ method.

`statistic_override`
manually override the uncertainy 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` 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".

Value

a numeric vector of test statistics

glance.mira *Glance a multiple imputation ‘mice‘ pooled object*

Description

Glance a multiple imputation ‘mice‘ pooled object

Usage

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

Arguments

`x` An object returned by one of the ‘mice::pool‘ function.

`...` extra arguments (not used)

Value

a tibble with one row

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

```
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 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

```
knit_latex(tab, label = NULL)
```

Arguments

`tab`	table object produced by ‘modelsummary‘ or ‘gt‘
`label`	string will be inserted as a ‘label‘

Value

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

modelsummary *Beautiful, customizable summaries of statistical models*

Description

Beautiful, customizable summaries of statistical models

Usage

```
modelsummary(models, statistic = "std.error",
  statistic_override = NULL, 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, 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".

`statistic_override` manually override the uncertainy 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.

`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 ‘modelsummary::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.

`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

```
# 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')))
```

msummary *Beautiful, customizable summaries of statistical models*

Description

Beautiful, customizable summaries of statistical models

Usage

```
msummary(models, statistic = "std.error", statistic_override = NULL,
  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, 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".

`statistic_override`
manually override the uncertainy 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.

`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 ‘modelsummary::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.

`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

```
# 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')))
```

rounding — *Convert numeric values to strings using the ‘sprintf‘ function. NA, NaN, -Inf, and Inf are replaced by an empty string.*

Description

Convert numeric values to strings using the ‘sprintf‘ function. NA, NaN, -Inf, and Inf are replaced by an empty string.

Usage

```
rounding(x, fmt = "%.3f")
```

Arguments

`x`	a numeric vector to be converted to string
`fmt`	a character vector of format strings which will be fed to the ‘sprintf‘ function. See ?sprintf for details.

Value

a rounded number as character

sanity_checks — *internal function to check the sanity of user input*

Description

internal function to check the sanity of user input

Usage

```
sanity_checks(models, statistic = "std.error",
  statistic_override = NULL, conf_level = 0.95, coef_map = NULL,
  coef_omit = NULL, gof_map = NULL, gof_omit = NULL, fmt = "%.3f",
  stars = NULL, stars_note = TRUE, title = NULL, subtitle = NULL,
  notes = NULL, add_rows = 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".

`statistic_override`

manually override the uncertainy 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.

`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 ‘modelsummary::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.

`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

error if sanity checks fail

statistic_override_function *Use the statistic_override function to extract std.error*

Description

Use the statistic_override function to extract std.error

Usage

```
statistic_override_function(model, statistic_override)
```

Arguments

`model`	object type with an available ‘tidy‘ method.
`statistic_override`	manually override the uncertainy 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.

Value

tibble

statistic_override_lmtest *Use the lmtest::coeftest function to extract uncertainty estimates*

Description

Use the lmtest::coeftest function to extract uncertainty estimates

Usage

```
statistic_override_lmtest(model, statistic_override)
```

Arguments

`model` object type with an available ‘tidy‘ method.

`statistic_override`
manually override the uncertainy 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.

Value

tibble

`statistic_override_matrix` *Use the statistic_override matrix to extract std.error*

Description

Use the statistic_override matrix to extract std.error

Usage

```
statistic_override_matrix(model, statistic_override)
```

Arguments

`model` object type with an available ‘tidy‘ method.

`statistic_override`
manually override the uncertainy 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.

Value

tibble

`statistic_override_vector`

Use the statistic_override vector to extract std.error/p.value/statistic

---

Description

Use the statistic_override vector to extract std.error/p.value/statistic

Usage

```
statistic_override_vector(model, statistic_override, statistic)
```

Arguments

`model` object type with an available ‘tidy‘ method.

`statistic_override`
manually override the uncertainy 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` 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".

Value

tibble

`tidy.mira` *Tidy a multiple imputation ‘mice‘ pooled object*

Description

Tidy a multiple imputation ‘mice‘ pooled object

Usage

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

Arguments

`x` An object returned by one of the ‘mice::pool‘ function.

`...` extra arguments (not used)

Value

a tibble with one row per term

Note

Available stats in mipo object:

- estimate
- ubar
- b
- t
- dfcom
- df
- riv
- lambda
- fmi

See Also

Other tidiers: `glance.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(Temp~ Ozone+Solar.R+Wind))
tidy(mod)
```

Index