

Package ‘equatiomatic’

August 20, 2026

Title Transform Models into 'LaTeX' Equations

Version 0.4.9

Maintainer Philippe Grosjean <phgrosjean@sciviews.org>

Description The goal of 'equatiomatic' is to reduce the pain associated with writing 'LaTeX' formulas from fitted models. The primary function of the package, `extract_eq()`, takes a fitted model object as its input and returns the corresponding 'LaTeX' code for the model.

License CC BY 4.0

Depends R (>= 3.3.0)

URL <https://github.com/datalorax/equatiomatic>,
<https://datalorax.github.io/equatiomatic/>

BugReports <https://github.com/datalorax/equatiomatic/issues>

Imports broom (>= 0.7.0), broom.mixed, shiny, knitr, rmarkdown, stats, utils

Suggests covr, shinyWidgets, forecast (>= 8.13), ggplot2 (>= 3.3.3), latex2exp (>= 0.4.0), lme4, MASS, ordinal, parsnip, recipes, workflows, testthat (>= 3.0.0), gtsummary, spelling

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Config/testthat/edition 3

Config/testthat/parallel true

Language en-US

NeedsCompilation no

Author Daniel Anderson [aut] (ORCID: <<https://orcid.org/0000-0003-4699-4680>>),
Andrew Heiss [aut] (ORCID: <<https://orcid.org/0000-0002-3948-3914>>),
Jay Sumners [aut],

Joshua Rosenberg [ctb] (ORCID: <<https://orcid.org/0000-0003-2170-0447>>),
Jonathan Sidi [ctb] (ORCID: <<https://orcid.org/0000-0002-4222-1819>>),
Ellis Hughes [ctb] (ORCID: <<https://orcid.org/0000-0003-0637-4436>>),
Thomas Fung [ctb] (ORCID: <<https://orcid.org/0000-0003-2601-0728>>),
Reza Norouzian [ctb] (ORCID: <<https://orcid.org/0000-0002-0531-6886>>),
Indrajeet Patil [ctb] (ORCID: <<https://orcid.org/0000-0003-1995-6531>>),
Quinn White [ctb] (ORCID: <<https://orcid.org/0000-0001-5399-0237>>),
David Kane [ctb],
Philippe Grosjean [cre] (ORCID:
<<https://orcid.org/0000-0002-2694-9471>>),
Teun van den Brand [ctb] (ORCID:
<<https://orcid.org/0000-0002-9335-7468>>),
Guyliann Engels [ctb] (ORCID: <<https://orcid.org/0000-0001-9514-1014>>)

Repository CRAN
Date/Publication 2026-08-19 12:50:09 UTC

Contents

arrests	2
equation	3
extract_eq	5
format.equation	9
hsb	10
knit_print.equation	10
penguins	11
polls	12
preview_eq	13
print.equation	13
renderEq	14
simple_ts	15
sim_longitudinal	15
ts_reg_list	16

Index	17
--------------	-----------

arrests	<i>Arrest data from Gelman & Hill</i>
---------	---

Description

Arrest data from Gelman & Hill’s book, used in Chapter 6 (and others). The data have been aggregated by precinct and race/ethnicity, with the sum of prior arrests and stops calculated. You can download the original data here: <http://www.stat.columbia.edu/~gelman/arm/examples/police/>

Usage

arrests

Format

A tibble with 225 rows and 4 variables:

precinct An integer denoting the precinct identification number.

eth A factor with the coded race/ethnicity

stops The number of police stops

arrests The number of prior arrests (this is used as an offset variable in the book)

equation	<i>Get a 'LaTeX' equation from a model or from 'LaTeX' code in a R Markdown or Quarto document</i>
----------	--

Description

Extract or create a 'LaTeX' equation to describe a model, or directly from 'LaTeX' code. All objects supported by `extract_eq()` are also supported by the default method description. If labels (and units) are defined as attributes to the data, they are automatically used in the equation in place of the variable names.

Usage

```
equation(object, ...)

## Default S3 method:
equation(
  object,
  auto.labs = TRUE,
  origdata = NULL,
  labs = NULL,
  swap_var_names = NULL,
  ...
)

## S3 method for class 'character'
equation(object, ...)

eq__(object, ...)

eq_(object, ...)

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

Arguments

object	An object with a model whose the equation is constructed. If a character object is provided, it is concatenated into a single character string and the equation class, otherwise it is not transformed (it is supposed to be a valid 'LaTeX' equation). Remember that backslashes must be doubled in regular R strings, or use the <code>r"(...)"</code> notation, which may be more comfortable here, see examples.
...	Further parameters passed to <code>extract_eq()</code> .
auto.labs	If TRUE (by default), use labels (and units) automatically from data or <code>origdata=</code> .
origdata	The original data set this model was fitted to. By default it is NULL and no label is used.
labs	Labels to change the names of elements in the term column of the table. By default it is NULL and no term is changed.
swap_var_names	Change the variable names for these values, regardless of the values in <code>auto.labs=</code> or <code>labs=</code> that are ignored if this argument is used. Provide a named character string with name being the variables and strings the new names. You can use <code>^</code> or <code>_</code> to indicate next character, or next integer should be super- or subscript in the equation.
x	An inline_equation object generated by <code>eq_()</code> or <code>eq__()</code> .

Details

There are slight differences between `equation()`, `eq_()` and `eq__()`:

- `equation()` returns a string with 'LaTeX' code and prints 'LaTeX' code at the R console.
- `eq_()` returns the 'LaTeX' code surrounded by a dollar sign `$. . .$` and is suitable to build inline equations in R Markdown/Quarto documents by using inline R code. It prints the rendered inline equation in the RStudio Viewer or in the browser. So it is advised to rapidly preview the resulting equation.
- `eq__()` returns the 'LaTeX' code *not* surrounded by dollar signs in a single character string. It also prints the rendered inline equation in the RStudio Viewer or in the browser. It should be used in an R inline chunk inside a `$$...$$` construct in a Markdown text. The result is a display equation that can also be cross referenced in Quarto in the usual way if you label it, e.g., you use `$$...$$ {#eq-label}`.

Value

An object of class `c("equation", "character")`.

Examples

```
iris_lm <- lm(data = iris, Petal.Length ~ Sepal.Length + Species)
summary(iris_lm)
equation(iris_lm)
# Providing directly the LaTeX code of the equation (variance of a sample)
equation("S^2_y = \\sum_{i=1}^{n-1} \\frac{(y_i - \\bar{Y})^2}{n}")
# Easier to write like this (avoiding the double backslashes):
eq1 <- equation(r"(S^2_y = \\sum_{i=1}^{n-1} \\frac{(y_i - \\bar{Y})^2}{n})")
```

```

# Print raw text:
eq1
# Get a preview of the equation
if (interactive())
  eq__(eq1)
# The same function can be used inside a `$$...$$ {#eq-label}` construct in
# R Markdown or Quarto to calculate a display equation that is also recognized
# by the cross referencing system of Quarto.
# Get a string suitable for inclusion inline in R Markdown with `r eq__(eq1)`
# (inside Markdown text and without the dollar signs around it)
# For inline equations in a Markdown text, you are better to use `r eq_(eq1)`
if (interactive())
  eq_(eq1)

```

extract_eq

'LaTeX' equation for R models

Description

Extract the variable names from a model to produce a 'LaTeX' equation. Supports any model where there is a `broom::tidy()` method. This is a generic function with methods for `lmerMod` objects obtained with `lme4::lmer()`, `glmerMod` objects with `lme4::glmer()`, `forecast_ARIMA` with `forecast::Arima()` and default, with the later further covering most "base" R models implemented in `broom::tidy()` like `lm` objects with `stats::lm()`, `glm` objects with `stats::glm()` or `polr` objects with `MASS::polr()`. The default method also supports `clm` objects obtained with `ordinal::clm()`.

Usage

```

extract_eq(
  model,
  intercept = "alpha",
  greek = "beta",
  greek_colors = NULL,
  subscript_colors = NULL,
  var_colors = NULL,
  var_subscript_colors = NULL,
  raw_tex = FALSE,
  swap_var_names = NULL,
  swap_subscript_names = NULL,
  ital_vars = FALSE,
  label = NULL,
  index_factors = FALSE,
  show_distribution = FALSE,
  wrap = FALSE,
  terms_per_line = 4,
  operator_location = "end",
  align_env = "aligned",

```

```

use_coefs = FALSE,
coef_digits = 2,
fix_signs = TRUE,
font_size = NULL,
mean_separate = NULL,
return_variances = FALSE,
se_subscripts = FALSE,
logit_notation = FALSE,
...
)

```

Arguments

model	A fitted model
intercept	How should the intercept be displayed? Default is "alpha", but can also accept "beta", in which case the it will be displayed as beta zero.
greek	What notation should be used for coefficients? Currently only accepts "beta" (with plans for future development). Can be used in combination with raw_tex to use any notation, e.g., " $\hat{\beta}$ ".
greek_colors	The colors of the greek notation in the equation. Must be a single color (named or HTML hex code) or a vector of colors (which will be recycled if smaller than the number of terms in the model). When rendering to PDF, I suggest using HTML hex codes, as not all named colors are recognized by LaTeX, but equationmatic will internally create the color definitions for you if HTML codes are supplied. Note that this is not yet implemented for mixed effects models (lme4).
subscript_colors	The colors of the subscripts for the greek notation. The argument structure is equivalent to greek_colors (i.e., see above for more detail).
var_colors	The color of the variable names. This takes a named vector of the form <code>c("variable" = "color")</code> . For example <code>c("bill_length_mm" = "#00d4fa", "island" = "#00fa85")</code> . Colors can be names (e.g., "red") or HTML hex codes, as shown in the example.
var_subscript_colors	The colors of the factor subscripts for categorical variables. The interface for this is equivalent to var_colors, and all subscripts for a given variable will be displayed in the provided color. For example, the code <code>c("island" = "green")</code> would result in the subscripts for "Dream" and "Torgersen" being green (assuming "Biscoe" was the reference group).
raw_tex	Logical. Is the greek code being passed to denote coefficients raw tex code?
swap_var_names	A vector of the form <code>c("old_var_name" = "new name")</code> . For example: <code>c("bill_length_mm" = "Bill Length (MM)")</code> .
swap_subscript_names	A vector of the form <code>c("old_subscript_name" = "new name")</code> . For example: <code>c("f" = "Female")</code> .
ital_vars	Logical, defaults to FALSE. Should the variable names not be wrapped in the <code>\operatorname{}</code> command?

label	A label for the equation, which can then be used for in-text references. See example here . Note that this only works for PDF output . The in-text references also must match the label exactly, and must be formatted as <code>\ref{eq: label}</code> , where label is a place holder for the specific label. Notice the space after the colon before the label. This also must be there, or the cross-reference will fail.
index_factors	Logical, defaults to FALSE. Should the factors be indexed, rather than using subscripts to display all levels?
show_distribution	Logical. When fitting a logistic or probit regression, should the binomial distribution be displayed? Defaults to FALSE.
wrap	Logical, defaults to FALSE. Should the terms on the right-hand side of the equation be split into multiple lines? This is helpful with models with many terms.
terms_per_line	Integer, defaults to 4. The number of right-hand side terms to include per line. Used only when wrap is TRUE.
operator_location	Character, one of “end” (the default) or “start”. When terms are split across multiple lines, they are split at mathematical operators like +. If set to “end”, each line will end with a trailing operator (+ or -). If set to “start”, each line will begin with an operator.
align_env	TeX environment to wrap around equation. Must be one of aligned, aligned*, align, or align*. Defaults to aligned.
use_coefs	Logical, defaults to FALSE. Should the actual model estimates be included in the equation instead of math symbols?
coef_digits	Integer, defaults to 2. The number of decimal places to round to when displaying model estimates.
fix_signs	Logical, defaults to TRUE. If disabled, coefficient estimates that are negative are preceded with a "+" (e.g. $5(x) + -3(z)$). If enabled, the "+ -" is replaced with a "-" (e.g. $5(x) - 3(z)$).
font_size	The font size of the equation. Defaults to default of the output format. Takes any of the standard LaTeX arguments (see here).
mean_separate	Currently only support for lmer models. Should the mean structure be inside or separated from the normal distribution? Defaults to NULL, in which case it will become TRUE if there are more than three fixed-effect parameters. If TRUE, the equation will be displayed as, for example, $\text{outcome} \sim N(\mu, \sigma)$; $\mu = \alpha + \beta_1(\text{wave})$. If FALSE, this same equation would be $\text{outcome} \sim N(\alpha + \beta, \sigma)$.
return_variances	Logical. When use_coefs = TRUE with a mixed effects model (e.g., <code>lme4::lmer()</code>), should the variances and co-variances be returned? If FALSE (the default) standard deviations and correlations are returned instead.
se_subscripts	Logical. If se_subscripts = TRUE then the equation will include the standard errors below each coefficient. This is supported for lm and glm models.
logit_notation	Logical. If TRUE, then the equation uses a more compact notation for a logistic regression.
...	Additional arguments (for future development; not currently used).

Details

The different methods all use the same arguments, but not all arguments are suitable to all models. Check here above to determine if a feature is implemented for a given model.

Value

A character of class “equation”.

Examples

```
# Simple model
mod1 <- lm(mpg ~ cyl + disp, mtcars)
extract_eq(mod1)

# Include all variables
mod2 <- lm(mpg ~ ., mtcars)
extract_eq(mod2)

# Works for categorical variables too, putting levels as subscripts
data("penguins", package = "equatiomatic")
mod3 <- lm(body_mass_g ~ bill_length_mm + species, penguins)
extract_eq(mod3)

set.seed(8675309)
d <- data.frame(
  cat1 = rep(letters[1:3], 100),
  cat2 = rep(LETTERS[1:3], each = 100),
  cont1 = rnorm(300, 100, 1),
  cont2 = rnorm(300, 50, 5),
  out = rnorm(300, 10, 0.5)
)
mod4 <- lm(out ~ ., d)
extract_eq(mod4)

# Don't italicize terms
extract_eq(mod1, ital_vars = FALSE)

# Wrap equations in an "aligned" environment
extract_eq(mod2, wrap = TRUE)

# Wider equation wrapping
extract_eq(mod2, wrap = TRUE, terms_per_line = 4)

# Include model estimates instead of Greek letters
extract_eq(mod2, wrap = TRUE, terms_per_line = 2, use_coefs = TRUE)

# Don't fix doubled-up "+ -" signs
extract_eq(mod2, wrap = TRUE, terms_per_line = 4, use_coefs = TRUE, fix_signs = FALSE)

# Use indices for factors instead of subscripts
extract_eq(mod2, wrap = TRUE, terms_per_line = 4, index_factors = TRUE)
```

```
# Use other model types, like glm
set.seed(8675309)
d <- data.frame(
  out = sample(0:1, 100, replace = TRUE),
  cat1 = rep(letters[1:3], 100),
  cat2 = rep(LETTERS[1:3], each = 100),
  cont1 = rnorm(300, 100, 1),
  cont2 = rnorm(300, 50, 5)
)
mod5 <- glm(out ~ ., data = d, family = binomial(link = "logit"))
extract_eq(mod5, wrap = TRUE)
# More compact logit notation
extract_eq(mod5, wrap = TRUE, logit_notation = TRUE)
```

format.equation	<i>Format 'LaTeX' equations</i>
-----------------	---------------------------------

Description

Format 'LaTeX' equations built with [extract_eq](#).

Usage

```
## S3 method for class 'equation'
format(x, ..., latex = knitr::is_latex_output())
```

Arguments

x	'LaTeX' equation built with extract_eq
...	not used
latex	Logical, whether the output is LaTeX or not. The default value uses knitr::is_latex_output() to determine the current output format.

Value

A character string with the equation formatted either as proper LaTeX code, or as a display equation tag (surrounded by `$$...$$`) for R Markdown or Quarto documents.

hsb

*A subset of the full 1982 High School and Beyond Survey***Description**

This is the dataset used throughout Raudenbush & Bryk (2002).

Usage

hsb

Format

A tibble with 7185 rows and 8 variables:

sch.id An integer denoting the school identification number. There are 160 unique schools.

math Individual students' math score.

size The number of students in the school.

sector A dummy variable (integer) denoting whether the school is public (sector = 0) or catholic (sector = 1). There are 90 public schools and 70 catholic.

meanses A group-mean centered SES variable at the school level

minority A dummy variable indicating if the student was coded as white (minority = 0) or not (minority = 1).

female A dummy variable indicating if the student was coded as female (female = 1) or not (female = 0).

ses A student-level composite variable indicating the students' socio-economic status.

knit_print.equation

*Print 'LaTeX' equations in R Markdown environments***Description**

Print 'LaTeX' equations built with [extract_eq](#) nicely in R Markdown environments.

Usage

```
## S3 method for class 'equation'
knit_print(
  x,
  ...,
  tex_packages = "\\renewcommand*\\familydefault{\\rmdefault}"
)
```

Arguments

<code>x</code>	'LaTeX' equation built with <code>extract_eq</code>
<code>...</code>	not used
<code>tex_packages</code>	A string with LaTeX code to include in the header, usually to include LaTeX packages in the output.

Value

A string with the equation formatted according to R Markdown's output format (different output for HTML, PDF, docx, gfm, markdown_strict). The format is detected automatically, so, you do not have to worry about it.

penguins	<i>Size measurements for adult foraging penguins near Palmer Station, Antarctica</i>
----------	--

Description

Data originally from `palmerpenguins`. Includes measurements for penguin species, island in Palmer Archipelago, size (flipper length, body mass, bill dimensions), and sex.

Usage

```
penguins
```

Format

A tibble with 344 rows and 8 variables:

species a factor denoting penguin species (Adélie, Chinstrap and Gentoo)

island a factor denoting island in Palmer Archipelago, Antarctica (Biscoe, Dream or Torgersen)

bill_length_mm a number denoting bill length (millimeters)

bill_depth_mm a number denoting bill depth (millimeters)

flipper_length_mm an integer denoting flipper length (millimeters)

body_mass_g an integer denoting body mass (grams)

sex a factor denoting penguin sex (female, male)

year an integer denoting the study year (2007, 2008, or 2009)

Source

Adélie penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Adélie penguins (*Pygoscelis adeliae*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative [doi:10.6073/pasta/98b16d7d563f265cb52372c8ca99e60f](https://doi.org/10.6073/pasta/98b16d7d563f265cb52372c8ca99e60f)

Gentoo penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Gentoo penguin (*Pygoscelis papua*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative [doi:10.6073/pasta/7fca67fb28d56ee2ffa3d9370ebda689](https://doi.org/10.6073/pasta/7fca67fb28d56ee2ffa3d9370ebda689)

Chinstrap penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Chinstrap penguin (*Pygoscelis antarcticus*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 6. Environmental Data Initiative [doi:10.6073/pasta/c14dfcfada8ea13a17536e73eb6fbe9e](https://doi.org/10.6073/pasta/c14dfcfada8ea13a17536e73eb6fbe9e)

Originally published in: Gorman KB, Williams TD, Fraser WR (2014) Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus *Pygoscelis*). PLoS ONE 9(3): e90081. [doi:10.1371/journal.pone.0090081](https://doi.org/10.1371/journal.pone.0090081)

polls

The polls data from Gelman and Hill ()

Description

This is the dataset used in Gelman & Hill's book, Data Analysis Using Regression and Multi-level/Hierarchical Models. They are polling data on the presidential election from 1988, collected one week before the election. You can download all the data from the book here: <http://www.stat.columbia.edu/~gelman/arm/> Note that this is only a few of the variables from the original data supplied with the book.

Usage

polls

Format

A tibble with 13,544 rows and 7 variables:

state An integer denoting the state identification number.

edu An ordered factor stating the education level of the respondent

age An unordered factor stating the age of range of the respondent

female A dummy variable (integer) denoting whether the respondent was coded as male (female = 0) or female (female = 1).

black A dummy variable (integer) denoting whether the respondent was coded as Black (black = 1) or not Black (black = 0).

weight A sampling weight

bush Whether the respondent stated they were in favor of voting for George Bush Sr.

preview_eq	<i>Preview an equation in a web browser</i>
------------	---

Description

Preview 'LaTeX' equations built with `extract_eq`.

Usage

```
preview_eq(x, ...)
```

Arguments

x	'LaTeX' equation built with <code>extract_eq</code>
...	not used

Value

The path to the temporary html file that was created to preview the equation is returned invisibly.

Examples

```
if (interactive()) {
  mod1 <- lm(mpg ~ cyl + disp, mtcars)
  eq1 <- extract_eq(mod1)
  eq1 # Not that nice
  preview_eq(eq1)
  # or easier...
  preview_eq(mod1)
}
```

print.equation	<i>Print 'LaTeX' equations</i>
----------------	--------------------------------

Description

Print 'LaTeX' equations built with `extract_eq`.

Usage

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

Arguments

x	'LaTeX' equation built with <code>extract_eq</code>
...	not used

Value

The unmodified object 'x' is returned invisibly. The function is used for its side effect of printing the equation.

renderEq	<i>Display equations in shiny apps</i>
----------	--

Description

[Experimental] These are a set of functions designed to help render equations in Shiny applications (see the vignette about Shiny).

Usage

```
renderEq(expr, env = parent.frame(), quoted = FALSE, outputArgs = list())

eqOutput(outputId)
```

Arguments

expr	An R expression, specifically a call to extract_eq()
env	The environment
quoted	Is the expression quoted?
outputArgs	list of output arguments
outputId	The identifier of the output from the server. Should be passed as a string.

Value

Render the equation in a suitable way for Shiny for [renderEq\(\)](#) in an [eqOutput\(\)](#) equation output element that can be included in a panel.

Functions

- [renderEq\(\)](#): Rendering function
- [eqOutput\(\)](#): Output function

simple_ts	<i>Simple simulated time series data</i>
-----------	--

Description

Output from `set.seed(42); simple_ts <- ts(rnorm(1000), freq = 4)`. This is included primarily for unit testing.

Usage

```
simple_ts
```

Format

A tibble with 1000 rows and 8 variables:

Qtr1 First quarter simulated values.

Qtr2 Second quarter simulated values.

Qtr3 Third quarter simulated values.

Qtr4 Fourth quarter simulated values.

sim_longitudinal	<i>Simulated longitudinal data</i>
------------------	------------------------------------

Description

Data are simulated to be similar to longitudinal data collected within schools/districts.

Usage

```
sim_longitudinal
```

Format

A tibble with 1000 rows and 8 variables:

sid An integer denoting the individual student. There are 100 students.

school An integer denoting the school. There are 15 schools.

district An integer denoting the school district. There are 5 districts.

group A character variable denoting the instructional level of the student, low, medium, or high.

treatment A factor indicating whether the student received the intervention treatment (0 = no treatment received; 1 = treatment received).

prop_low The proportion of student in the school in the low instructional group.

wave The assessment wave. Each student has nine waves of data collection.

score The individual students' score at the given wave.

ts_reg_list	<i>Simulated data for time-series regression</i>
-------------	--

Description

Output from `set.seed(42); ts_reg_list <- list(x1 = rnorm(1000), x2 = rnorm(1000), ts_rnorm = rnorm(1000))`.

Usage

ts_reg_list

Format

A tibble with 1000 rows and 8 variables:

x1 Random normal simulated data.

x2 Random normal simulated data.

ts_rnorm Random normal simulated data.

Index

- * **datasets**
 - arrests, [2](#)
 - hsb, [10](#)
 - penguins, [11](#)
 - polls, [12](#)
 - sim_longitudinal, [15](#)
 - simple_ts, [15](#)
 - ts_reg_list, [16](#)
- arrests, [2](#)
- broom::tidy(), [5](#)
- eq_ (equation), [3](#)
- eq__ (equation), [3](#)
- eqOutput (renderEq), [14](#)
- eqOutput(), [14](#)
- equation, [3](#)
- extract_eq, [5](#), [9–11](#), [13](#)
- extract_eq(), [3](#), [4](#), [14](#)
- forecast::Arima(), [5](#)
- format.equation, [9](#)
- hsb, [10](#)
- knit_print.equation, [10](#)
- knitr::is_latex_output(), [9](#)
- lme4::glmer(), [5](#)
- lme4::lmer(), [5](#)
- lmer, [7](#)
- MASS::polr(), [5](#)
- ordinal::clm(), [5](#)
- penguins, [11](#)
- polls, [12](#)
- preview_eq, [13](#)
- print.equation, [13](#)

- print.inline_equation (equation), [3](#)
- renderEq, [14](#)
- renderEq(), [14](#)
- sim_longitudinal, [15](#)
- simple_ts, [15](#)
- stats::glm(), [5](#)
- stats::lm(), [5](#)
- ts_reg_list, [16](#)