Package ‘broomExtra’

October 27, 2021

Type Package

Title Enhancements for ’broom’ and ’easystats’ Package Families

Version 4.3.0

Maintainer Indrajeet Patil <patilindrajeet.science@gmail.com>

Description Provides helper functions that assist in data analysis workflows involving regression analyses. The goal is to combine the functionality offered by different set of packages (’broom’, ’broom.mixed’, ’parameters’, and ’performance’) through a common syntax to return tidy dataframes containing model parameters and performance measure summaries. The ’grouped_’ variants of the generics provides a convenient way to execute functions across a combination of grouping variable(s) in a dataframe.

License GPL-3 | file LICENSE


BugReports https://github.com/IndrajeetPatil/broomExtra/issues

Depends R (>= 3.6.0)

Imports broom, broom.mixed, dplyr, magrittr, parameters (>= 0.15.0), performance (>= 0.8.0), rlang

Suggests generics, lavaan, lme4, MASS, rmarkdown, spelling, testthat

Encoding UTF-8

Language en-US

RoxygenNote 7.1.2

Config/testthat/edition 3

NeedsCompilation no

Author Indrajeet Patil [aut, cre, cph]
<https://orcid.org/0000-0003-1995-6531>, @patilindrajeets

Repository CRAN

Date/Publication 2021-10-27 09:40:02 UTC
The `augment` function is used to retrieve an augmented dataframe if it exists. Here is the documentation:

**Description**

Checks if an `augment` method exits for a given object, either in `{broom}` or in `{broom.mixed}`. If it does, return the model summary dataframe; if not, return `NULL`.

**Usage**

```r
augment(x, ...)  
```

**Arguments**

- `x`: Model object or other R object with information to append to observations.
- `...`: Additional arguments to `augment` method.

**Value**

A `tibble::tibble()` with information about data points.

**See Also**

- `grouped_augment`

**Examples**

```r
set.seed(123)
lm.mod <- lm(Sepal.Length ~ Species, iris)
broomExtra::augment(lm.mod)
```
glance

Retrieve model summary dataframe if it exists.

Description

Checks if a glance method exists for a given object, either in \{broom\} or in \{broom.mixed\}. If it does, return the model summary dataframe, if not, return a NULL. In this case, you can try the \texttt{glance_performance()} function.

Usage

\begin{verbatim}
glance(x, ...)
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{x} model or other R object to convert to single-row data frame
\item \texttt{...} other arguments passed to methods
\end{itemize}

See Also

\begin{itemize}
\item \texttt{grouped\_glance, glance\_performance}
\end{itemize}

Examples

\begin{verbatim}
set.seed(123)
lm.mod <- lm(Sepal.Length ~ Species, iris)
broomExtra::glance(lm.mod)
\end{verbatim}

---

glance_performance

Model performance summary dataframes using \{broom\} and \{easystats\}

Description

Computes indices of model performance for regression models.

Usage

\begin{verbatim}
glance_performance(x, ...)
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{x} model or other R object to convert to single-row data frame
\item \texttt{...} other arguments passed to methods
\end{itemize}
grouped_augment

Details

The function will attempt to get these details either using `broom::glance()` or `performance::model_performance()`. If both function provide model performance measure summaries, the function will try to combine them into a single dataframe. Measures for which these two packages have different naming conventions, both will be retained.

Value

A data frame (with one row) and one column per "index".

Examples

```r
set.seed(123)
mod <- lm(mpg ~ wt + cyl, data = mtcars)
broomExtra::glance_performance(mod)
```

Description

Augmented data from grouped analysis of any function that has `data` argument in its function call.

Usage

```r
grouped_augment(data, grouping.vars, ..f, ..., augment.args = list())
```

Arguments

- **data**: Dataframe (or tibble) from which variables are to be taken.
- **grouping.vars**: Grouping variables.
- **..f**: A function, or function name as a string.
- **...**: `<dynamic>` Arguments for `.fn`.
- **augment.args**: A list of arguments to be used in the relevant S3 method.

Value

A `tibble::tibble()` with information about data points.

See Also

`augment`
Examples

```r
set.seed(123)
library(dplyr)

# linear mixed effects model
grouped_glance(
data = mutate(MASS::Aids2, interval = death - diag),
grouping.vars = sex,
..f = lme4::lmer,
formula = interval ~ age + (1 | status),
control = lme4::lmerControl(optimizer = "bobyqa")
)
```

---

Description

Model summary output from grouped analysis of any function that has `data` argument in its function call.

Usage

```r
grouped_glance(data, grouping.vars, ..f, ...)
```

Arguments

- `data`: Dataframe (or tibble) from which variables are to be taken.
- `grouping.vars`: Grouping variables.
- `..f`: A function, or function name as a string.
- `...`: `<dynamic>` Arguments for `.fn`.

See Also

- `glance`

Examples

```r
set.seed(123)
library(dplyr)

# linear mixed effects model
grouped_glance(
data = mutate(MASS::Aids2, interval = death - diag),
grouping.vars = sex,
..f = lme4::lmer,
formula = interval ~ age + (1 | status),
control = lme4::lmerControl(optimizer = "bobyqa")
)```
grouped_tidy

**Grouped tidy analysis**

Description

Tidy output from grouped analysis of any function that has `data` argument in its function call.

Usage

```r
grouped_tidy(data, grouping.vars, ..f, ..., tidy.args = list())
```

Arguments

- `data`: Dataframe (or tibble) from which variables are to be taken.
- `grouping.vars`: Grouping variables.
- `..f`: A function, or function name as a string.
- `...`: <dynamic> Arguments for `.fn`.
- `tidy.args`: A list of arguments to be used in the relevant S3 method.

Value

A `tibble::tibble()` with information about model components.

See Also

- `tidy`

Examples

```r
set.seed(123)
library(dplyr)

# linear mixed effects model
grouped_tidy(
  data = mutate(MASS::Aids2, interval = death - diag),
  grouping.vars = sex,
  ..f = lme4::lmer,
  formula = interval ~ age + (1 | status),
  control = lme4::lmerControl(optimizer = "bobyqa"),
  tidy.args = list(conf.int = TRUE, conf.level = 0.99)
)
```
tidy

Retrieve tidy dataframe if it exists

Description

Checks if a tidy method exits for a given object, either in \{broom\} or in \{broom.mixed\}. If it does, it turn an object into a tidy tibble, if not, return a NULL. In this case, you can try the tidy_parameters() function.

Usage

tidy(x, ...)

Arguments

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

... Additional arguments to tidying method.

Value

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

See Also

grouped_tidy, tidy_parameters

Examples

```r
set.seed(123)
lm.mod <- lm(Sepal.Length ~ Species, iris)
broomExtra::tidy(x = lm.mod, conf.int = TRUE)
```

tidy_parameters

Tidy dataframes of model parameters using \{broom\} and \{easystats\}

Description

Computes parameters for regression models.

Usage

tidy_parameters(x, conf.int = TRUE, ...)

tidy_parameters

Arguments

- `x`: An object to be converted into a tidy tibble::tibble()
- `conf.int`: Indicating whether or not to include a confidence interval in the tidied output (defaults to TRUE).
- `...`: Additional arguments that will be passed to parameters::model_parameters() or broom::tidy(), whichever method works. Note that you should pay attention to different naming conventions across these packages. For example, the required confidence interval width is specified using ci argument in parameters::model_parameters, while using conf.level in broom::tidy.

Details

The function will attempt to get these details first using parameters::model_parameters(), and if this fails, then using broom::tidy().

Value

A data frame of indices related to the model’s parameters.

Examples

```
set.seed(123)
mod <- lm(mpg ~ wt + cyl, data = mtcars)
broomExtra::tidy_parameters(mod)
```
Index

augment, 2, 4
broom::glance(), 4
broom::tidy(), 8
dynamic, 4–6
glance, 3, 5
glance_performance, 3, 3
glance_performance(), 3
grouped_augment, 2, 4
grouped_glance, 3, 5
grouped_tidy, 6, 7

parameters::model_parameters(), 8
performance::model_performance(), 4
tibble::tibble(), 2, 4, 6–8
tidy, 6, 7
tidy_parameters, 7, 7
tidy_parameters(), 7