Package 'broomExtra'

February 19, 2021

Type Package

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

Version 4.2.0

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

URL https://indrajeetpatil.github.io/broomExtra/,
https://github.com/IndrajeetPatil/broomExtra

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

Depends R (>= 3.6.0)

Imports broom (>= 0.7.4),
broom.mixed,
dplyr,
magrittr,
parameters (>= 0.11.0),
performance (>= 0.7.0),
rlang,
tibble,
lifecycle

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

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE, roclets = c(``rd'', ``namespace'', ``collate'', ``pkgapi::api_roclet''))

RoxygenNote 7.1.1.9001

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first watcher, parallel*

R **topics documented:**

---

augment   *Retrieve augmented dataframe if it exists.*

Description

Check if a 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 a `NULL`.

Usage

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

Arguments

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

Value

A [tibble::tibble()](#) with information about data points.

Methods

No methods found in currently loaded packages.

See Also

[grouped_augment](#)

Examples

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

---

glance	*Retrieve model summary dataframe if it exists.*

Description

Check if a `glance` method exits 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 `glance_performance()` function.

Usage

```
glance(x, ...)
```

Arguments

x	model or other R object to convert to single-row data frame
...	other arguments passed to methods

Methods

No methods found in currently loaded packages.

See Also

`grouped_glance`, `glance_performance`

Examples

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

---

glance_performance	*Model performance summary dataframes using* `broom` *and* `easystats`.

Description

Stable

Computes indices of model performance for regression models.

Usage

```
glance_performance(x, ...)
```

Arguments

x	model or other R object to convert to single-row data frame
...	other arguments passed to methods

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

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

---

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

Description

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

Usage

```
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.

Methods

No methods found in currently loaded packages.

See Also

[augment](#)

Examples

```
set.seed(123)

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

---

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

Description

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

Usage

```
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.

Methods

No methods found in currently loaded packages.

See Also

[glance](#)

Examples

```
set.seed(123)

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

---

grouped_tidy   *Tidy output from grouped analysis of any function that has* data *argu-*
                                ment in its function call.

Description

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

Usage

```
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.

Methods

No methods found in currently loaded packages.

See Also

[tidy](#)

Examples

```
set.seed(123)

# linear mixed effects model
broomExtra::grouped_tidy(
  data = dplyr::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.

Methods

No methods found in currently loaded packages.

See Also

`grouped_tidy`, `tidy_parameters`

Examples

```
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

Stable

Computes parameters for regression models.

Usage

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

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 atten-
             tion to different naming conventions across these packages. For example, the re-
             quired 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