Package ‘report’

September 30, 2021

Type Package
Title Automated Reporting of Results and Statistical Models
Version 0.4.0
Maintainer Dominique Makowski <dom.makowski@gmail.com>
Description The aim of the 'report' package is to bridge the gap between R’s output and the formatted results contained in your manuscript. This package converts statistical models and data frames into textual reports suited for publication, ensuring standardization and quality in results reporting.
License GPL-3 | file LICENSE
BugReports https://github.com/easystats/report/issues
Imports bayestestR (>= 0.10.5), effectsize (>= 0.4.5), insight (>= 0.14.1), parameters (>= 0.14.0), performance (>= 0.7.2), datawizard, stats, tools, utils
Suggests BayesFactor, brms, dplyr, httr, ivreg, knitr, lavaan, lme4, logspline, rmarkdown, rstanarm, spelling, survival, testthat
VignetteBuilder knitr
Encoding UTF-8
Language en-US
RoxygenNote 7.1.2
Config/testthat/edition 3
R topics documented:

'report.lme.R' 'report.numeric.R' 'report.sessionInfo.R'
'report.survreg.R' 'report.test_performance.R'
'report.zeroInfl.R' 'report_effectsize.R' 'report_info.R'
'report_intercept.R' 'report_misc.R' 'report_model.R'
'report_parameters.R' 'report_participants.R'
'report_sample.R' 'report_statistics.R' 'report_table.R'
'utils_data.R' 'utils_error_message.R' 'utils_grouped_df.R'

NeedsCompilation no

Author Dominique Makowski [aut, cre] (<https://orcid.org/0000-0001-5375-9967>,
@Dom_Makowski),
Daniel Lüdecke [aut] (<https://orcid.org/0000-0002-8895-3206>,
@strengejacke),
Mattan S. Ben-Shachar [aut] (<https://orcid.org/0000-0002-4287-4801>,
@mattansb),
Indrajeet Patil [aut] (<https://orcid.org/0000-0003-1995-6531>,
@patilindrajeets),
Rudolf Siegel [ctb] (<https://orcid.org/0000-0002-6021-804X>)

Repository CRAN

Date/Publication 2021-09-30 07:30:02 UTC

R topics documented:

as.report_text .................................................. 3
cite_easystats .................................................... 4
data_rename ......................................................... 5
format_algorithm .................................................. 6
format_citation ................................................... 7
format_text ......................................................... 8
report ............................................................... 9
report.aov ........................................................ 11
report.bayesfactor_models ..................................... 13
report.brmsfit .................................................... 15
report.character ................................................ 17
report.compare_performance ................................... 19
report.default ................................................... 21
report.htest ....................................................... 23
report.lavaan ..................................................... 25
report.lm .......................................................... 26
report.sessionInfo ............................................... 29
report.stanreg .................................................... 30
report.test_performance ......................................... 32
report_date ....................................................... 34
report_effectsize ............................................... 35
report_info ....................................................... 36
report_intercept ............................................... 37
as.report_text

Create or test objects of class report.

Description

Allows to create or test whether an object is of the report class.

Usage

as.report_text(x, ...)

as.report(text, table = NULL, plot = NULL, ...)

is.report(x)

as.report_effectsize(x, summary = NULL, prefix = " - ", ...)

as.report_info(x, summary = NULL, ...)

as.report_intercept(x, summary = NULL, ...)

as.report_model(x, summary = NULL, ...)

as.report_parameters(x, summary = NULL, prefix = " - ", ...)

as.report_performance(x, summary = NULL, ...)

as.report_priors(x, summary = NULL, ...)

as.report_random(x, summary = NULL, ...)

as.report_statistics(x, summary = NULL, prefix = " - ", ...)

as.report_table(x, ...)

Index

as.report_text 50
Arguments

x
An arbitrary R object.
...
Args to be saved as attributes.
text
Text obtained via report_text()
table
Table obtained via report_table()
plot
Plot obtained via report_plot(). Not yet implemented.
summary
Add a summary as attribute (to be extracted via summary()).
prefix
The prefix to be displayed in front of each parameter.

Value

A report object or a TRUE/FALSE value.

cite_easystats Cite the easystats ecosystem

Description

A convenient function for those who wish to cite the easystats packages.

Usage

cite_easystats()

Value

An object of class cite_easystats that can be printed, summarized (using summary()), or transformed into a table (using as.data.frame()).

Examples

cite_easystats()
summary(cite_easystats())
as.data.frame(cite_easystats())
data_rename

Convenient dataframe manipulation functionalities

Description
Safe and intuitive functions to manipulate dataframes.

Usage
```
data_rename(data, pattern, replacement, safe = TRUE)
data_findcols(data, pattern = NULL, starts_with = NULL, ends_with = NULL)
data_remove(data, pattern)
data_reorder(data, cols, safe = TRUE)
data_addprefix(data, pattern)
data_addsuffix(data, pattern)
```

Arguments
- `data`: Dataframe.
- `pattern, replacement, starts_with, ends_with`: Character strings.
- `safe`: Do not throw error if for instance the variable to be renamed/removed doesn’t exist.
- `cols`: Vector of column names.

Value
A modified data frame.

Examples
```
library(report)
# Rename columns
data_rename(iris, "Sepal.Length", "length")
# data_rename(iris, "FakeCol", "length", safe=FALSE) # This fails
data_rename(iris, "FakeCol", "length") # This doesn't
data_rename(iris, c("Sepal.Length", "Sepal.Width"), c("length", "width"))

# Find columns names by pattern
data_findcols(iris, starts_with = "Sepal")
data_findcols(iris, ends_with = "Width")
data_findcols(iris, pattern = "\.")
```
# Remove columns
data_remove(iris, "Sepal.Length")

# Reorder columns
data_reorder(iris, c("Species", "Sepal.Length"))
data_reorder(iris, c("Species", "dupa"))

# Add prefix / suffix
data_addprefix(iris, "NEW_")
data_addsuffix(iris, "_OLD")

---

format_algorithm  Convenient formatting of text components

Description
Convenient formatting of text components

Usage
format_algorithm(x)
format_formula(x, what = "conditional")
format_model(x)

Arguments
x  The R object that you want to report (see list of of supported objects above).
what  The name of the item returned by insight::find_formula.

Value
A character string.
A character string.
A character string.

Examples
model <- lm(Sepal.Length ~ Species, data = iris)
format_algorithm(model)

if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Sepal.Width + (1 | Species), data = iris)
  format_algorithm(model)
}
model <- lm(Sepal.Length ~ Species, data = iris)
format_citation

format_formula(model)

if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Sepal.Width + (1 | Species), data = iris)
  format_formula(model)
  format_formula(model, "random")
}
model <- lm(Sepal.Length ~ Species, data = iris)
format_model(model)

if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Sepal.Width + (1 | Species), data = iris)
  format_model(model)
}

---

**format_citation**  
**Citation formatting**

**Description**

Convenience functions to manipulate and format citations. Only works with APA formatted citations, for now.

**Usage**

```r
format_citation(citation, authorsdate = FALSE, short = FALSE, intext = FALSE)
cite_citation(citation)
clean_citation(citation)
```

**Arguments**

- `citation`: A character string of a citation.
- `authorsdate`: Only show authors and date (remove title, journal, etc.).
- `short`: If more than one authors, replace by et al.
- `intext`: Remove brackets around the date (so that it can be placed inside larger parentheses).

**Value**

A character string.
Examples

library(report)


format_citation(citation, authorsdate = TRUE)
format_citation(citation, authorsdate = TRUE, short = TRUE)
format_citation(citation, authorsdate = TRUE, short = TRUE, intext = TRUE)

cite_citation(citation)
clean_citation(citation())

Convenient text formatting functionalities

Description

Convenience functions to manipulate and format text.

Usage

format_text(text, sep = ", ", last = " and ", width = NULL, ...)  
text_fullstop(text)  
text_lastchar(text, n = 1)  
text_concatenate(text, sep = ", ", last = " and ")  
text_paste(text, text2 = NULL, sep = ", ", ...)  
text_remove(text, pattern = "", ...)  
text_wrap(text, width = NULL, ...)

Arguments

text, text2 A character string.
sep Separator.
last Last separator.
width Positive integer giving the target column width for wrapping lines in the output. Can be "auto", in which case it will select 90\ default width.
... Other arguments to be passed to or from other functions.
n The number of characters to find.
pattern Character strings.
Value

A character string.

Examples

library(report)

# Add full stop if missing
text_fullstop(c("something", "something else.\n"))

# Find last characters
text_lastchar(c("ABC", "DEF"), n = 2)

# Smart concatenation
text_concatenate(c("First", "Second", "Last"))

# Remove parts of string
text_remove(c("one!", "two", "three!", ":")

# Wrap text
long_text <- paste(rep("abc ", 100), collapse = "")
cat(text_wrap(long_text, width = 50))

# Paste with optional separator
text_paste(c("A", ",", "B"), c("42", "42", "42"))

---

report

Automatic reporting of R objects

Description

Create reports of different objects. See the documentation for your object's class:

- System and packages (sessionInfo)
- Dataframes and vectors
- Correlations and t-tests (htest)
- ANOVAs (aov, anova, aovlist, ...)
- Regression models (glm, lm, ...)
- Mixed models (glmer, lmer, glmmTMB, ...)
- Bayesian models (stanreg, brms...)
- Bayes factors (from bayestestR)
- Structural Equation Models (SEM) (from lavaan)
- Model comparison (from performance())

Most of the time, the object created by the `report()` function can be further transformed, for instance summarized (using `summary()`), or converted to a table (using `as.data.frame()`).
Usage

\texttt{report(x, \ldots)}

Arguments

\texttt{x} \quad \text{The R object that you want to report (see list of supported objects above).}
\texttt{\ldots} \quad \text{Arguments passed to or from other methods.}

Details

\textbf{Organization:} \texttt{report\_table} and \texttt{report\_text} are the two distal representations of a report, and are the two provided in \texttt{report()}. However, intermediate steps are accessible (depending on the object) via specific functions (e.g., \texttt{report\_parameters}).

\textbf{Output:}

The \texttt{report()} function generates a report-object that contain in itself different representations (e.g., text, tables, plots). These different representations can be accessed via several functions, such as:

- \texttt{as.report\_text(r)}: Detailed text.
- \texttt{as.report\_text(r, summary=TRUE)}: Minimal text giving the minimal information.
- \texttt{as.report\_table(r)}: Comprehensive table including most available indices.
- \texttt{as.report\_table(r, summary=TRUE)}: Minimal table.

Note that for some report objects, some of these representations might be identical.

Value

A list-object of class \texttt{report}, which contains further list-objects with a short and long description of the model summary, as well as a short and long table of parameters and fit indices.

See Also

Specific components of reports (especially for stats models):

- \texttt{report\_table()}
- \texttt{report\_parameters()}
- \texttt{report\_statistics()}
- \texttt{report\_effectsize()}
- \texttt{report\_model()}
- \texttt{report\_priors()}
- \texttt{report\_random()}
- \texttt{report\_performance()}
- \texttt{report\_info()}
- \texttt{report\_text()}

Other types of reports:
report.aov

- `report_system()`
- `report_packages()`
- `report_participants()`
- `report_sample()`
- `report_date()`

Methods:

- `as.report()`

Template file for supporting new models:

- `report.default()`

Examples

```r
library(report)

model <- t.test(mpg ~ am, data = mtcars)
r <- report(model)

# Text
r
summary(r)

# Tables
as.data.frame(r)
summary(as.data.frame(r))
```

---

**Description**

Create reports for ANOVA models.

**Usage**

```r
## S3 method for class 'aov'
report(x, ...)
```

```r
## S3 method for class 'aov'
report_effectsize(x, ...)
```

```r
## S3 method for class 'aov'
report_table(x, ...)
```

```r
## S3 method for class 'aov'
```
report_statistics(x, table = NULL, ...)

## S3 method for class 'aov'
report_parameters(x, ...)

## S3 method for class 'aov'
report_model(x, table = NULL, ...)

## S3 method for class 'aov'
report_info(x, effectsize = NULL, ...)

## S3 method for class 'aov'
report_text(x, table = NULL, ...)

Arguments

x       Object of class aov, anova or aovlist.
...
    Arguments passed to or from other methods.
table   Provide the output of report_table() to avoid its re-computation.
effectsize
        Provide the output of report_effectsize() to avoid its re-computation.

Value

An object of class report().

See Also

Specific components of reports (especially for stats models):

- report_table()
- report_parameters()
- report_statistics()
- report_effectsize()
- report_model()
- report_priors()
- report_random()
- report_performance()
- report_info()
- report_text()

Other types of reports:

- report_system()
- report_packages()
- report_participants()
- report_sample()
• `report_date()`

Methods:

• `as.report()`

Template file for supporting new models:

• `report.default()`

Examples

```r
data <- iris
data$Cat1 <- rep(c("A", "B"), length.out = nrow(data))

model <- aov(Sepal.Length ~ Species * Cat1, data = data)
r <- report(model)
r
summary(r)
as.data.frame(r)
summary(as.data.frame(r))
```

---

`report.bayesfactor_models`

*Reporting Models’ Bayes Factor*

Description

Create reports of Bayes factors for model comparison.

Usage

```r
## S3 method for class 'bayesfactor_models'
report(
  x,
  interpretation = "jeffreys1961",
  exact = TRUE,
  protect_ratio = TRUE,
  ...
)

## S3 method for class 'bayesfactor_inclusion'
report(
  x,
  interpretation = "jeffreys1961",
  exact = TRUE,
  protect_ratio = TRUE,
  ...
)
```
Arguments

- **x**: Object of class `bayesfactor_inclusion`.
- **interpretation**: Effect size interpretation set of rules (see `interpret_bf`).
- **exact**: Should very large or very small values be reported with a scientific format (e.g., `4.24e5`), or as truncated values (as `"> 1000"` and `< 1/1000`).
- **protect_ratio**: Should values smaller than 1 be represented as ratios?
- **...**: Arguments passed to or from other methods.

Value

An object of class `report()`.

See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
- `report_priors()`
- `report_random()`
- `report_performance()`
- `report_info()`
- `report_text()`

Other types of reports:

- `report_system()`
- `report_packages()`
- `report_participants()`
- `report_sample()`
- `report_date()`

Methods:

- `as.report()`

Template file for supporting new models:

- `report.default()`
Examples

library(report)

mo0 <- lm(Sepal.Length ~ 1, data = iris)
mo1 <- lm(Sepal.Length ~ Species, data = iris)
mo2 <- lm(Sepal.Length ~ Species + Petal.Length, data = iris)
mo3 <- lm(Sepal.Length ~ Species * Petal.Length, data = iris)

if (require("bayestestR")) {
  # Bayes factor - models
  BFmodels <- bayesfactor_models(mo1, mo2, mo3, denominator = mo0)
  r <- report(BFmodels)
  r
  as.data.frame(r)

  # Bayes factor - inclusion
  inc_bf <- bayesfactor_inclusion(BFmodels, prior_odds = c(1, 2, 3), match_models = TRUE)
  r <- report(inc_bf)
  r
  as.data.frame(r)
}

---

report.brmsfit  Reporting Bayesian Models from brms

Description

Create reports for Bayesian models. The description of the parameters follows the Sequential Effect eXistence and slgnificance Testing framework (see SEXIT documentation).

Usage

## S3 method for class 'brmsfit'
report(x, ...)

Arguments

x  Object of class lm or glm.

...  Arguments passed to or from other methods.

Value

An object of class report().
See Also

Specific components of reports (especially for stats models):

- report_table()
- report_parameters()
- report_statistics()
- report_effectsize()
- report_model()
- report_priors()
- report_random()
- report_performance()
- report_info()
- report_text()

Other types of reports:

- report_system()
- report_packages()
- report_participants()
- report_sample()
- report_date()

Methods:

- as.report()

Template file for supporting new models:

- report.default()

Examples

library(report)

# Bayesian models
## Not run:
if (require("brms")) {
  model <- brm(mpg ~ qsec + wt, data = mtcars, refresh = 0, iter = 300)
  r <- report(model)
  r
  summary(r)
  as.data.frame(r)
  summary(as.data.frame(r))
}

## End(Not run)
Description

Create reports for data frames.

Usage

```r
## S3 method for class 'character'
report(
  x,
  n_entries = 3,
  levels_percentage = "auto",
  missing_percentage = "auto",
  ...
)

## S3 method for class 'data.frame'
report(
  x,
  n = FALSE,
  centrality = "mean",
  dispersion = TRUE,
  range = TRUE,
  distribution = FALSE,
  levels_percentage = "auto",
  digits = 2,
  n_entries = 3,
  missing_percentage = "auto",
  ...
)

## S3 method for class 'factor'
report(x, levels_percentage = "auto", ...)

## S3 method for class 'numeric'
report(
  x,
  n = FALSE,
  centrality = "mean",
  dispersion = TRUE,
  range = TRUE,
  distribution = FALSE,
  missing_percentage = "auto",
  digits = 2,
  ...
)
```
Arguments

x  The R object that you want to report (see list of supported objects above).
n_entries  Number of different character entries to show. Can be "all".
levels_percentage  Show characters entries and factor levels by number or percentage. If "auto", then will be set to number and percentage if the length if n observations larger than 100.
missing_percentage  Show missing by number (default) or percentage. If "auto", then will be set to number and percentage if the length if n observations larger than 100.
...  Arguments passed to or from other methods.
n  Include number of observations for each individual variable.
centrality  Character vector, indicating the index of centrality (either "mean" or "median").
dispersion  Show index of dispersion (sd if centrality = "mean", or mad if centrality = "median").
range  Show range.
distribution  Show kurtosis and skewness.
digits  Number of significant digits.

Value

An object of class report().

Examples

library(report)

r <- report(iris,
  centrality = "median", dispersion = FALSE,
  distribution = TRUE, missing_percentage = TRUE
)

r

summary(r)

as.data.frame(r)

summary(as.data.frame(r))

if (require("dplyr")) {
  r <- iris %>%
    dplyr::group_by(Species) %>%
    report()

  r

  summary(r)

  as.data.frame(r)

  summary(as.data.frame(r))
}
Description

Create reports for model comparison as obtained by the `performance::compare_performance()` function in the performance package.

Usage

```r
## S3 method for class 'compare_performance'
report(x, ...)

## S3 method for class 'compare_performance'
report_table(x, ...)

## S3 method for class 'compare_performance'
report_statistics(x, table = NULL, ...)

## S3 method for class 'compare_performance'
report_parameters(x, table = NULL, ...)

## S3 method for class 'compare_performance'
report_text(x, table = NULL, ...)
```

Arguments

- `x` Object of class NEW OBJECT.
- `...` Arguments passed to or from other methods.
- `table` Provide the output of `report_table()` to avoid its re-computation.

Value

An object of class `report()`.

See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
report.compare_performance

- report_priors()
- report_random()
- report_performance()
- report_info()
- report_text()

Other types of reports:
- report_system()
- report_packages()
- report_participants()
- report_sample()
- report_date()

Methods:
- as.report()

Template file for supporting new models:
- report.default()

Examples

library(report)
library(performance)

m1 <- lm(Sepal.Length ~ Petal.Length * Species, data = iris)
m2 <- lm(Sepal.Length ~ Petal.Length + Species, data = iris)
m3 <- lm(Sepal.Length ~ Petal.Length, data = iris)

x <- performance::compare_performance(m1, m2, m3)
r <- report(x)
summary(r)
as.data.frame(r)
summary(as.data.frame(r))

# Specific reports
report_table(x)
report_statistics(x)
report_parameters(x)
Template to add report support for new objects

Description

Template file to add report support for new objects. Check-out the vignette on Supporting New Models.

Usage

```r
## Default S3 method:
report(x, ...)
```

```r
## Default S3 method:
report_effectsize(x, ...)
```

```r
## Default S3 method:
report_table(x, ...)
```

```r
## Default S3 method:
report_statistics(x, ...)
```

```r
## Default S3 method:
report_parameters(x, ...)
```

```r
## Default S3 method:
report_intercept(x, ...)
```

```r
## Default S3 method:
report_model(x, ...)
```

```r
## Default S3 method:
report_random(x, ...)
```

```r
## Default S3 method:
report_priors(x, ...)
```

```r
## Default S3 method:
report_performance(x, ...)
```

```r
## Default S3 method:
report_info(x, ...)
```

```r
## Default S3 method:
report_text(x, ...)
```
Arguments

- Object of class `NEW OBJECT`.
- Arguments passed to or from other methods.

Value

An object of class `report()`.

See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
- `report_priors()`
- `report_random()`
- `report_performance()`
- `report_info()`
- `report_text()`

Other types of reports:

- `report_system()`
- `report_packages()`
- `report_participants()`
- `report_sample()`
- `report_date()`

Methods:

- `as.report()`

Template file for supporting new models:

- `report.default()`

Examples

```r
library(report)

# Add a reproducible example instead of the following
model <- lm(Sepal.Length ~ Petal.Length * Species, data = iris)

r <- report(model)

r

summary(r)
as.data.frame(r)
summary(as.data.frame(r))
```
Description

Create reports for htest objects (t.test(), cor.test(), etc.).

Usage

```r
## S3 method for class 'htest'
report(x, ...)

## S3 method for class 'htest'
report_effectsize(x, ...)

## S3 method for class 'htest'
report_table(x, ...)

## S3 method for class 'htest'
report_statistics(x, table = NULL, ...)

## S3 method for class 'htest'
report_parameters(x, table = NULL, ...)

## S3 method for class 'htest'
report_model(x, table = NULL, ...)

## S3 method for class 'htest'
report_info(x, effectsize = NULL, ...)

## S3 method for class 'htest'
report_text(x, table = NULL, ...)
```

Arguments

- **x**: Object of class htest.
- **...**: Arguments passed to or from other methods.
- **table**: Provide the output of `report_table()` to avoid its re-computation.
- **effectsize**: Provide the output of `report_effectsize()` to avoid its re-computation.

Value

An object of class `report()`. 
See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
- `report_priors()`
- `report_random()`
- `report_performance()`
- `report_info()`
- `report_text()`

Other types of reports:

- `report_system()`
- `report_packages()`
- `report_participants()`
- `report_sample()`
- `report_date()`

Methods:

- `as.report()`

Template file for supporting new models:

- `report.default()`

Examples

```r
report(t.test(iris$Sepal.Width, iris$Sepal.Length))
report(t.test(iris$Sepal.Width, iris$Sepal.Length, var.equal = TRUE))
report(t.test(mtcars$mpg ~ mtcars$vs))
report(t.test(mtcars$mpg, mtcars$vs, paired = TRUE))
report(t.test(iris$Sepal.Width, mu = 1))
```
Description

Create a report for lavaan objects.

Usage

```r
## S3 method for class 'lavaan'
report(x, ...)
```

```r
## S3 method for class 'lavaan'
report_performance(x, table = NULL, ...)
```

Arguments

- `x` Object of class lavaan.
- `...` Arguments passed to or from other methods.
- `table` Provide the output of `report_table()` to avoid its re-computation.

Value

An object of class `report()`.

See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
- `report_priors()`
- `report_random()`
- `report_performance()`
- `report_info()`
- `report_text()`

Other types of reports:

- `report_system()`
- `report_packages()`
• `report_participants()`
• `report_sample()`
• `report_date()`

Methods:
• `as.report()`

Template file for supporting new models:
• `report.default()`

Examples

library(report)

# Structural Equation Models (SEM)
if (require("lavaan")) {
  structure <- " ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3
dem60 ~ ind60 "
  model <- lavaan::sem(structure, data = PoliticalDemocracy)
  r <- report(model)
  r
  # summary(r)
  # as.data.frame(r)
  # summary(as.data.frame(r))

  # Specific reports
  report_table(model)
  report_performance(model)
}

---

report.lm  Reporting (General) Linear Models

Description

Create reports for (general) linear models.

Usage

```r
# S3 method for class 'lm'
report(x, include_effectsize = TRUE, effectsize_method = "refit", ...)

# S3 method for class 'lm'
report_effectsize(x, effectsize_method = "refit", ...)

# S3 method for class 'lm'
```
Arguments

x                     Object of class lm or glm.
include_effectsize    If FALSE, won’t include effect-size related indices (standardized coefficients,
etc.).

effectsize_method

See documentation for `effectsize::effectsize()`.

... Arguments passed to or from other methods.

table

Provide the output of `report_table()` to avoid its re-computation.

include_diagnostic

If FALSE, won’t include diagnostic related indices for Bayesian models (ESS, Rhat).

include_intercept

If FALSE, won’t include the intercept.

effectsize

Provide the output of `report_effectsize()` to avoid its re-computation.

parameters

Provide the output of `report_parameters()` to avoid its re-computation.

Value

An object of class `report()`.

See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
- `report_priors()`
- `report_random()`
- `report_performance()`
- `report_info()`
- `report_text()`

Other types of reports:

- `report_system()`
- `report_packages()`
- `report_participants()`
- `report_sample()`
- `report_date()`

Methods:

- `as.report()`

Template file for supporting new models:

- `report.default()`
Examples

```r
library(report)

# Linear models
model <- lm(Sepal.Length ~ Petal.Length * Species, data = iris)
r <- report(model)
r
summary(r)
as.data.frame(r)
summary(as.data.frame(r))

# Logistic models
model <- glm(vs ~ disp, data = mtcars, family = "binomial")
r <- report(model)
r
summary(r)
as.data.frame(r)
summary(as.data.frame(r))

# Mixed models
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
r <- report(model)
r
summary(r)
as.data.frame(r)
summary(as.data.frame(r))
}
```

---

**report.sessionInfo**  
*Report R environment (packages, system, etc.)*

**Description**

Report R environment (packages, system, etc.)

**Usage**

```r
## S3 method for class 'sessionInfo'
report(x, ...)

report_packages(session = NULL, include_R = TRUE, ...)
cite_packages(session = NULL, include_R = TRUE, ...)
report_system(session = NULL)
```
Arguments

x The R object that you want to report (see list of supported objects above).

... Arguments passed to or from other methods.

session A sessionInfo object.

include_R Include R in the citations.

Value

- For report_packages, a data frame of class with information on package name, version and citation.

An object of class report().

Examples

library(report)

session <- sessionInfo()

r <- report(session)

r

summary(r)

as.data.frame(r)

summary(as.data.frame(r))

# Convenience functions

report_packages(include_R = FALSE)

cite_packages(prefix = "> ")

report_system()

---

Report.stanreg Reporting Bayesian Models

Description

Create reports for Bayesian models. The description of the parameters follows the Sequential Effect eXistence and sIgnificance Testing framework (see SEXIT documentation).

Usage

## S3 method for class 'stanreg'

report(x, ...)

Arguments

x Object of class lm or glm.

... Arguments passed to or from other methods.
Value

An object of class report().

See Also

Specific components of reports (especially for stats models):

- report_table()
- report_parameters()
- report_statistics()
- report_effectsize()
- report_model()
- report_priors()
- report_random()
- report_performance()
- report_info()
- report_text()

Other types of reports:

- report_system()
- report_packages()
- report_participants()
- report_sample()
- report_date()

Methods:

- as.report()

Template file for supporting new models:

- report.default()

Examples

library(report)

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(mpg ~ qsec + wt, data = mtcars, refresh = 0, iter = 500)
  r <- report(model)
  summary(r)
  as.data.frame(r)
  summary(as.data.frame(r))
}
report.test_performance

Reporting models comparison

Description

Create reports for model comparison as obtained by the `performance::compare_performance()` function in the performance package.

Usage

```r
## S3 method for class 'test_performance'
report(x, ...)

## S3 method for class 'test_performance'
report_table(x, ...)

## S3 method for class 'test_performance'
report_statistics(x, table = NULL, ...)

## S3 method for class 'test_performance'
report_parameters(x, table = NULL, ...)

## S3 method for class 'test_performance'
report_text(x, table = NULL, ...)
```

Arguments

- `x` Object of class NEW OBJECT.
- `...` Arguments passed to or from other methods.
- `table` Provide the output of `report_table()` to avoid its re-computation.

Value

An object of class `report()`.

See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
report.test_performance

- report_priors()
- report_random()
- report_performance()
- report_info()
- report_text()

Other types of reports:
- report_system()
- report_packages()
- report_participants()
- report_sample()
- report_date()

Methods:
- as.report()

Template file for supporting new models:
- report.default()

Examples

library(report)
library(performance)

m1 <- lm(Sepal.Length ~ Petal.Length * Species, data = iris)
m2 <- lm(Sepal.Length ~ Petal.Length + Species, data = iris)
m3 <- lm(Sepal.Length ~ Petal.Length, data = iris)

x <- performance::test_performance(m1, m2, m3)
r <- report(x)
summary(r)
as.data.frame(r)
summary(as.data.frame(r))

# Specific reports
report_table(x)
report_statistics(x)
report_parameters(x)
**Miscellaneous reports**

**Description**

Other convenient or totally useless reports.

**Usage**

```r
call_report(...)  
call_story(...)  
```

**Arguments**

... Arguments passed to or from other methods.

**Value**

Objects of class `report_text()`.

**See Also**

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
- `report_priors()`
- `report_random()`
- `report_performance()`
- `report_info()`
- `report_text()`

Other types of reports:

- `report_system()`
- `report_packages()`
- `report_participants()`
- `report_sample()`
- `report_date()`

Methods:
Examples

library(report)

report_date()
summary(report_date())
report_story()

---

**report_effectsize**  
_Report the effect size(s) of a model or a test_

Description

Computes, interpret and formats the effect sizes of a variety of models and statistical tests (see list of supported objects in `report()`).

Usage

`report_effectsize(x, ...)`

Arguments

- `x` The R object that you want to report (see list of of supported objects above).
- `...` Arguments passed to or from other methods.

Value

An object of class `report_effectsize()`.

Examples

library(report)

# h-tests
report_effectsize(t.test(iris$Sepal.Width, iris$Sepal.Length))

# ANOVAs
report_effectsize(aov(Sepal.Length ~ Species, data = iris))

# GLMs
report_effectsize(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
report_effectsize(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  report_effectsize(model)
}

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600)
  report_effectsize(model, effectsize_method = "basic")
}

---

**report_info**  
*Report additional information*

**Description**
Reports additional information relevant to the report (see list of supported objects in `report()`).

**Usage**

```r
report_info(x, ...)
```

**Arguments**

- `x`  
The R object that you want to report (see list of supported objects above).
- `...`  
Arguments passed to or from other methods.

**Value**

An object of class `report_info()`.

**Examples**

```r
library(report)
# h-tests
report_info(t.test(iris$Sepal.Width, iris$Sepal.Length))

# ANOVAs
report_info(aov(Sepal.Length ~ Species, data = iris))

# GLMs
report_info(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
report_info(lm(Sepal.Length ~ Petal.Length * Species, data = iris), include_effectsize = TRUE)
report_info(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
if (require("lme4")) {
```
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_info(model)
}

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 300)
  report_info(model)
}

---

**report_intercept**  Report intercept

### Description

Reports intercept of regression models (see list of supported objects in `report()`).

### Usage

```r
report_intercept(x, ...)
```

### Arguments

- `x`  
  The R object that you want to report (see list of supported objects above).

- `...`  
  Arguments passed to or from other methods.

### Value

An object of class `report_intercept()`.

### Examples

```r
library(report)

# GLMs
report_intercept(lm(Sepal.Length ~ Species, data = iris))
report_intercept(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  report_intercept(model)
}

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600)
  ```
report_model

Report the model type

Description

Reports the type of different R objects (see list of supported objects in `report()`).

Usage

```r
report_model(x, table = NULL, ...)
```

Arguments

- `x` The R object that you want to report (see list of supported objects above).
- `table` A table obtained via `report_table()`. If not provided, will run it.
- `...` Arguments passed to or from other methods.

Value

A character string.

Examples

```r
library(report)

# h-tests
report_model(t.test(iris$Sepal.Width, iris$Sepal.Length))

# ANOVA
report_model(aov(Sepal.Length ~ Species, data = iris))

# GLMs
report_model(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
report_model(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  report_model(model)
}

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600)
}```
report_parameters

    report_model(model)

Report the parameters of a model

Description

Creates a list containing a description of the parameters of R objects (see list of supported objects in `report()`).

Usage

    report_parameters(x, ...)

Arguments

x

  The R object that you want to report (see list of supported objects above).

...

  Arguments passed to or from other methods.

Value

A vector.

Examples

library(report)

# Miscellaneous
r <- report_parameters(sessionInfo())
summary(r)

# Data
report_parameters(iris$Sepal.Length)
report_parameters(as.character(round(iris$Sepal.Length, 1)))
report_parameters(iris$Species)
report_parameters(iris)

# h-tests
report_parameters(t.test(iris$Sepal.Width, iris$Sepal.Length))

# ANOVA
report_parameters(aov(Sepal.Length ~ Species, data = iris))

# GLMs
report_parameters(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
report_participants

A helper function to help you format the participants data (age, sex, ...) in the participants section.

Usage

report_participants(
    data,
    age = NULL,
    sex = NULL,
    education = NULL,
    participants = NULL,
    group = NULL,
    spell_n = FALSE,
    digits = 1,
    ...
)

Arguments

data A data frame.
age The name of the column containing the age of the participant.
sex The name of the column containing the sex of the participant. The classes should
     be one of c("Male","M","Female","F"). Note that you can specify other char-
     acters here as well (e.g., "Other"), but the function will report only percentage
     of females, regardless of whether any category other than "Male" is present in
     the data.
education The name of the column containing education information.
participants   The name of the participants' identifier column (for instance in the case of repeated measures).
group           A character vector indicating the name(s) of the column(s) used for stratified description.
spell_n         Fully spell the sample size ("Three participants" instead of "3 participants").
digits           Number of significant digits.
...             Arguments passed to or from other methods.

Value
A character vector with description of the "participants", based on the information provided in data.

Examples

library(report)
data <- data.frame(
  "Age" = c(22, 23, 54, 21, 8, 42),
  "Sex" = c("F", "F", "M", "M", "M", "F")
)

report_participants(data, age = "Age", sex = "Sex")

# Years of education (relative to high school graduation)
data$Education <- c(0, 8, -3, -5, 3, 5)
report_participants(data, age = "Age", sex = "Sex", education = "Education")

# Education as factor
data$Education2 <- c("Bachelor", "PhD", "Highschool",
  "Highschool", "Bachelor", "Bachelor"
)
report_participants(data, age = "Age", sex = "Sex", education = "Education2")

# Repeated measures data
data <- data.frame(
  "Age" = c(22, 22, 54, 54, 8, 8),
  "Sex" = c("F", "F", "M", "M", "F", "F"),
  "Participant" = c("S1", "S1", "S2", "S2", "S3", "S3")
)

report_participants(data, age = "Age", sex = "Sex", participants = "Participant")

# Grouped data
data <- data.frame(
  "Age" = c(22, 22, 54, 54, 8, 8, 42, 42),
  "Participant" = c("S1", "S1", "S2", "S2", "S3", "S3", "S4", "S4"),
)
Description

Investigating the fit of statistical models to data often involves selecting the best fitting model amongst many competing models. This function helps report indices of model fit for various models. Reports the type of different R objects. For a list of supported objects, see `report()`.

Usage

```r
report_performance(x, table = NULL, ...)
```

Arguments

- `x` The R object that you want to report (see list of supported objects above).
- `table` A table obtained via `report_table()`. If not provided, will run it.
- `...` Arguments passed to or from other methods.

Value

An object of class `report_performance()`.

Examples

```r
library(report)

# GLMs
report_performance(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
report_performance(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  report_performance(model)
}
# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600)
  report_performance(model)
}

# Structural Equation Models (SEM)
if (require("lavaan")) {
  structure <- " ind60 =~ x1 + x2 + x3
               dem60 =~ y1 + y2 + y3
               dem60 ~ ind60 "
  model <- lavaan::sem(structure, data = PoliticalDemocracy)
  report_performance(model)
}

---

## report_priors

### Report priors of Bayesian models

**Description**
Reports priors of Bayesian models (see list of supported objects in `report()`).

**Usage**

```r
report_priors(x, ...)”
```

**Arguments**

- `x` The R object that you want to report (see list of of supported objects above).
- `...` Arguments passed to or from other methods.

**Value**

An object of class `report_priors()`.

**Examples**

```r
library(report)

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(mpg ~ disp, data = mtcars, refresh = 0, iter = 1000)
  r <- report_priors(model)
  summary(r)
}
```
Report random effects and factors

Description

Reports random effects of mixed models (see list of supported objects in `report()`).

Usage

```
report_random(x, ...)
```

Arguments

- `x`: The R object that you want to report (see list of supported objects above).
- `...`: Arguments passed to or from other methods.

Value

An object of class `report_random()`.

Examples

```
library(report)

# Mixed models
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  r <- report_random(model)
  summary(r)
}

# Bayesian models
if (require("rstanarm")) {
  model <- stan_lmer(mpg ~ disp + (1 | cyl), data = mtcars, refresh = 0, iter = 1000)
  r <- report_random(model)
  summary(r)
}

## Not run:
if (require("brms")) {
  model <- brm(mpg ~ disp + (1 | cyl), data = mtcars, refresh = 0, iter = 1000)
  r <- report_random(model)
  summary(r)
}

## End(Not run)
```
report_sample

Sample Description

Description
Create sample description table (also referred to as "Table 1").

Usage

```r
report_sample(
  data,
  group_by = NULL,
  centrality = "mean",
  select = NULL,
  exclude = NULL,
  weights = NULL,
  total = TRUE,
  digits = 2,
  ...
)
```

Arguments

- **data** A data frame for which descriptive statistics should be created.
- **group_by** Character vector, indicating the column for possible grouping of the descriptive table.
- **centrality** Character, indicates the statistics that should be calculated for numeric variables. May be "mean" (for mean and standard deviation) or "median" (for median and median absolute deviation) as summary.
- **select** Character vector, with column names that should be included in the descriptive table.
- **exclude** Character vector, with column names that should be excluded from the descriptive table.
- **weights** Character vector, indicating the name of a potential weight-variable. Reported descriptive statistics will be weighted by weight.
- **total** Add a Total column.
- **digits** Number of decimals.
- **...** Arguments passed to or from other methods.

Value

A data frame of class report_sample with variable names and their related summary statistics.
Examples

library(report)

report_sample(iris[, 1:4])
report_sample(iris, select = c("Sepal.Length" , "Petal.Length" , "Species" ))
report_sample(iris, group_by = "Species")

---

**report_statistics**

*Report the statistics of a model*

Description

Creates a list containing a description of the parameters’ values of R objects (see list of supported objects in `report()`). Useful to insert in parentheses in plots or reports.

Usage

```r
report_statistics(x, table = NULL, ...)
```

Arguments

- `x` The R object that you want to report (see list of supported objects above).
- `table` A table obtained via `report_table()`. If not provided, will run it.
- `...` Arguments passed to or from other methods.

Value

An object of class `report_statistics()`.

Examples

```r
library(report)

# Data
report_statistics(iris$Sepal.Length)
report_statistics(as.character(round(iris$Sepal.Length, 1)))
report_statistics(iris$Species)
report_statistics(iris)

# t-tests
report_statistics(t.test(iris$Sepal.Width, iris$Sepal.Length))

# ANOVA
report_statistics(aov(Sepal.Length ~ Species, data = iris))

# GLMs
report_statistics(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
```
report_table

Report a descriptive table

Description

Creates tables to describe different objects (see list of supported objects in report()).

Usage

report_table(x, ...)

Arguments

x

The R object that you want to report (see list of supported objects above).

...

Arguments passed to or from other methods.

Value

An object of class report_table().

Examples

library(report)

# Miscellaneous
r <- report_table(sessionInfo())
summary(r)

# Data
report_table(iris$Sepal.Length)
report_table(as.character(round(iris$Sepal.Length, 1)))
report_table(iris$Species)
report_table(iris)

# h-tests
report_table(t.test(mpg ~ am, data = mtcars))

# ANOVAs
report_table(aov(Sepal.Length ~ Species, data = iris))

# GLMs
report_table(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
report_table(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  report_table(model)
}

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600)
  report_table(model, effectsize_method = "basic")
}

# Structural Equation Models (SEM)
if (require("lavaan")) {
  structure <- " ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3
dem60 ~ ind60 "
  model <- lavaan::sem(structure, data = PoliticalDemocracy)
  report_table(model)
}

---

**report_text**

*Report a textual description of an object*

**Description**

Creates text containing a description of the parameters of R objects (see list of supported objects in `report()`).

**Usage**

`report_text(x, table = NULL, ...)`

**Arguments**

- **x**: The R object that you want to report (see list of of supported objects above).
- **table**: A table obtained via `report_table()`. If not provided, will run it.
- **...**: Arguments passed to or from other methods.
Value

An object of class `report_text()`.

Examples

```r
library(report)

# Miscellaneous
r <- report_text(sessionInfo())
r
summary(r)

# Data
report_text(iris$Sepal.Length)
report_text(as.character(round(iris$Sepal.Length, 1)))
report_text(iris$Species)
report_text(iris)

# t-tests
report_text(t.test(iris$Sepal.Width, iris$Sepal.Length))

# ANOVA
r <- report_text(aov(Sepal.Length ~ Species, data = iris))
r
summary(r)

# GLMs
r <- report_text(lm(Sepal.Length ~ Petal.Length * Species, data = iris))
r
summary(r)

if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  r <- report_text(model)
r
  summary(r)
}

# Bayesian models
if (require("rstanarm")) {
  model <- stan_glm(mpg ~ cyl + wt, data = mtcars, refresh = 0, iter = 600)
  r <- report_text(model)
r
  summary(r)
}
```
Index

ANOVAs, 9
as.report(as.report_text), 3
as.report(), 11, 13, 14, 16, 20, 22, 24, 26, 28, 31, 33, 35
as.report_effectsize(as.report_text), 3
as.report_info(as.report_text), 3
as.report_intercept(as.report_text), 3
as.report_model(as.report_text), 3
as.report_parameters(as.report_text), 3
as.report_performance(as.report_text), 3
as.report_priors(as.report_text), 3
as.report_random(as.report_text), 3
as.report_statistics(as.report_text), 3
as.report_table(as.report_text), 3
Bayes factors, 9
Bayesian models, 9
cite_citation(format_citation), 7
cite_easystats, 4
cite_packages(report.sessionInfo), 29
clean_citation(format_citation), 7
Correlations and t-tests, 9
data_addprefix(data_rename), 5
data_addsuffix(data_rename), 5
data_findcols(data_rename), 5
data_remove(data_rename), 5
data_rename, 5
data_reorder(data_rename), 5
Dataframes and vectors, 9
effectsize::effectsize(), 28
format_algorithm, 6
format_citation, 7
format_formula(format_algorithm), 6
format_model(format_algorithm), 6
format_text, 8
interpret_bf, 14
is.report(as.report_text), 3
mad, 18
Mixed models, 9
Model comparison, 9
performance(), 9
performance::compare_performance(), 19, 32
Regression models, 9
report, 3, 9
report.aov, 11
report.bayesfactor_inclusion
  (report.bayesfactor_models), 13
report.bayesfactor_models, 13
report.brmsfit, 15
report.character, 17
report.compare_performance, 19
report.data.frame(report.character), 17
report.default, 21
report.default(), 11, 13, 14, 16, 20, 22, 24, 26, 28, 31, 33, 35
report.factor(report.character), 17
report.htest, 23
report.lavaan, 25
report.lm, 26
report.numeric(report.character), 17
report.sessionInfo, 29
report.stanreg, 30
report.test_performance, 32
report_date, 34
report_date(), 11, 13, 14, 16, 20, 22, 24, 26, 28, 31, 33, 34
report_effectsize, 35
report_effectsize(), 10, 12, 14, 16, 19, 22, 24, 25, 28, 31, 32, 34, 35
INDEX

report_effectsize.aov (report.aov), 11
report_effectsize.default
  (report.default), 21
report_effectsize.htest (report.htest),  23
report_effectsize.lm (report.lm), 26
report_info(), 10, 12, 14, 16, 20, 22, 24, 25,
  28, 31, 33, 34, 36
report_info.aov (report.aov), 11
report_info.default (report.default), 21
report_info.htest (report.htest),  23
report_info.lm (report.lm), 26
report_intercept(), 37
report_intercept.default
  (report.default), 21
report_intercept.lm (report.lm), 26
report_model(), 10, 12, 14, 16, 19, 22, 24,
  25, 28, 31, 32, 34
report_model.aov (report.aov), 11
report_model.default (report.default), 21
report_model.htest (report.htest),  23
report_model.lm (report.lm), 26
report_packages (report.sessionInfo), 29
report_packages(), 11, 12, 14, 16, 20, 22,
  24, 25, 28, 31, 33, 34
report_parameters, 39
report_parameters(), 10, 12, 14, 16, 19, 22,
  24, 25, 28, 31, 32, 34
report_parameters.aov (report.aov), 11
report_parameters.compare_performance
  (report.compare_performance), 19
report_parameters.default
  (report.default), 21
report_parameters.htest (report.htest),  23
report_parameters.lm (report.lm), 26
report_parameters.test_performance
  (report.test_performance), 32
report_participants, 40
report_participants(), 11, 12, 14, 16, 20,
  22, 24, 26, 28, 31, 33, 34
report_performance, 42
report_performance(), 10, 12, 14, 16, 20,
  22, 24, 25, 28, 31, 33, 34, 42
report_performance.default
  (report.default), 21
report_performance.lavaan
  (report.lavaan), 25
report_performance.lm (report.lm), 26
report_priors, 43
report_priors(), 10, 12, 14, 16, 20, 22, 24,
  25, 28, 31, 33, 34, 43
report_priors.default (report.default), 21
report_random, 44
report_random(), 10, 12, 14, 16, 20, 22, 24,
  25, 28, 31, 33, 34, 44
report_random.default (report.default), 21
report_random.merMod (report.default), 26
report_sample, 45
report_sample(), 11, 12, 14, 16, 20, 22, 24,
  26, 28, 31, 33, 34
report_statistics, 46
report_statistics(), 10, 12, 14, 16, 19, 22,
  24, 25, 28, 31, 32, 34, 46
report_statistics.aov (report.aov), 11
report_statistics.compare_performance
  (report.compare_performance), 19
report_statistics.default
  (report.default), 21
report_statistics.htest (report.htest),  23
report_statistics.lm (report.lm), 26
report_statistics.test_performance
  (report.test_performance), 32
report_story (report_date), 34
report_system (report.sessionInfo), 29
report_system(), 11, 12, 14, 16, 20, 22, 24,
  25, 28, 31, 33, 34
report_table, 47
report_table(), 10, 12, 14, 16, 19, 22, 24,
  25, 28, 31, 32, 34, 47
report_table.aov (report.aov), 11
report_table.compare_performance
  (report.compare_performance), 19
report_table.default (report.default), 21
report_table.htest (report.htest), 23
report_table.lm (report.lm), 26
report_table.test_performance
  (report.test_performance), 32
report_text, 48
report_text(), 10, 12, 14, 16, 20, 22, 24, 25, 28, 31, 33, 34, 49
report_text.aov (report.aov), 11
report_text.compare_performance
  (report.compare_performance), 19
report_text.default (report.default), 21
report_text.h test (report.h test), 23
report_text.lm (report.lm), 26
report_text.test_performance
  (report.test_performance), 32

d s d, 18
sessionInfo, 30
SEXIT documentation, 15, 30
Structural Equation Models (SEM), 9
System and packages, 9

text_concatenate (format_text), 8
text_fullstop (format_text), 8
text_lastchar (format_text), 8
text_paste (format_text), 8
text_remove (format_text), 8
text_wrap (format_text), 8