Package ‘report’

December 7, 2023

Type Package
Title Automated Reporting of Results and Statistical Models
Version 0.5.8
Maintainer Rémi Thériault <remi.theriault@mail.mcgill.ca>
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 MIT + file LICENSE
BugReports https://github.com/easystats/report/issues
Depends R (>= 3.6)
Imports bayestestR (>= 0.13.1), effectsize (>= 0.8.6), insight (>=
0.19.7), parameters (>= 0.21.3), performance (>= 0.10.8),
datawizard (>= 0.9.0), stats, tools, utils
Suggests brms, ivreg, knitr, lavaan, lme4, dplyr, rmarkdown, rstanarm,
survival, modelbased, emmeans, testthat
VignetteBuilder knitr
Encoding UTF-8
Language en-US
RoxygenNote 7.2.3.9000
Config/testthat/edition 3
Config/Needs/website rstudio/bslib, r-lib/pkgdown,
easystats/easystatstemplate
Collate 'cite_easystats.R' 'format_algorithm.R' 'format_citation.R'
'format_formula.R' 'format_model.R' 'reexports.R'
'report-package.R' 'utils_combine_tables.R' 'report.lm.R'
'report.MixMod.R' 'report_text.R' 'report.R' 'report.hypothesis.R'
'report.aov.R' 'report.bayesfactor_models.R' 'report.lme4.R'
R topics documented:

'as.report_text'                  3
'cite_easystats'                 4
'format_algorithm'               6
'format_citation'               7
'report'                        8
'report.aov'                     10
'report.bayesfactor_models'     12
'report.brmsfit'                 14
'report.character'              15
'report.compare_performance'    17
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, ...)

Arguments

x
An arbitrary R object.

... Arg 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(
  packages = "easystats",
  format = c("text", "markdown", "biblatex"),
  intext_prefix = TRUE,
  intext_suffix = "."
)

## S3 method for class 'cite_easystats'
summary(object, what = "all", ...)

## S3 method for class 'cite_easystats'
print(x, what = "all", ...)
Arguments

packages  A character vector of packages to cite. Can be "all" for all easystats packages or a vector with specific package names.

format  The format to generate citations. Can be "text" for plain text, "markdown" for markdown citations and CSL bibliography (recommended for writing in RMarkdown), or "biblatex" for BibLaTeX citations and bibliography.

intext_prefix  A character vector of length 1 containing text to include before in-text citations. If TRUE, defaults to "Analyses were conducted using the easystats collection of packages ". If FALSE or NA, no prefix is included.

intext_suffix  A character vector of length 1 containing text to include after in-text citations. Defaults to ".". If FALSE or NA, no suffix is included.

what  What elements of the citations to print, can be "all", "intext", or "refs".

...  Not used. Included for compatibility with the generic function.

x, object  A "cite_easystats" object to print.

Value

A list of class "cite_easystats" with elements:

- intext: In-text citations in the requested format
- refs: References or bibliography in the requested format

Examples

# Cite just the 'easystats' umbrella package:
cite_easystats()
summary(cite_easystats(), what = "all")

# Cite every easystats package:
cite_easystats(packages = "all")
summary(cite_easystats(packages = "all"), what = "all")

# Cite specific packages:
cite_easystats(packages = c("modelbased", "see"))
summary(cite_easystats(packages = c("modelbased", "see")), what = "all")

# To cite easystats packages in an RMarkdown document, use:

## In-text citations:
print(cite_easystats(format = "markdown"), what = "intext")

## Bibliography (print with the 'output = 'asis'' option on the code chunk)
print(cite_easystats(format = "markdown"), what = "refs")
**format_algorithm**  
Convenient formatting of text components

**Description**  
Convenient formatting of text components

**Usage**  

```r
format_algorithm(x)
format_formula(x, what = "conditional")
format_model(x)
```

**Arguments**
- `x` The R object that you want to report (see list 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**

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

# Mixed models
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Sepal.Width + (1 | Species), data = iris)
format_algorithm(model)

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

# Mixed models
library(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_citation

# Mixed models
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Sepal.Width + (1 | Species), data = iris)
format_model(model)

---

**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**

```r
library(report)

format_citation(citation, authorsdate = TRUE)
format_citation(citation, authorsdate = TRUE, short = TRUE)
format_citation(citation, authorsdate = TRUE, short = TRUE, intext = TRUE)
```
Automatic reporting of R objects

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

Usage
report(x, ...)

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

Details
- 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()`).

Organization: `report_table` and `report_text` are the two distal representations of a report, and are the two provided in `report()`. However, intermediate steps are accessible (depending on the object) via specific functions (e.g., `report_parameters`).

Output:
The `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:
- `as.report_text(r)`: Detailed text.
• as.report_text(r, summary=TRUE): Minimal text giving the minimal information.
• as.report_table(r): Comprehensive table including most available indices.
• 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 `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):

- `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)

model <- t.test(mtcars$mpg ~ mtcars$am)
r <- report(model)

# Text
r
summary(r)

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

---

**report.aov**

*Reporting ANOVAs*

**Description**

Create reports for ANOVA models.

**Usage**

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

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

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

## 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

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

## 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

```r
library(bayestestR)
# Bayes factor - models
mo0 <- lm(Sepal.Length ~ 1, data = iris)
m1 <- lm(Sepal.Length ~ Species, data = iris)
m2 <- lm(Sepal.Length ~ Species + Petal.Length, data = iris)
m3 <- lm(Sepal.Length ~ Species * Petal.Length, data = iris)
BFmodels <- bayesfactor_models(m1, m2, m3, denominator = mo0)

r <- report(BFmodels)
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 slIgnificance Testing framework (see SEXIT documentation).

**Usage**

```r
## 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.character

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

Methods:
- as.report()

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

Examples

# Bayesian models
library(brms)
model <- suppressWarnings(brm(mpg ~ qsec + wt, data = mtcars, refresh = 0, iter = 300))
r <- report(model, verbose = FALSE)
r
summary(r)
as.data.frame(r)
summary(as.data.frame(r))

---

**report.character**

**Reporting Datasets and Dataframes**

### Description

Create reports for data frames.

### Usage

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

```r
## 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 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
```
r <- report(iris,
            centrality = "median", dispersion = FALSE,
            distribution = TRUE, missing_percentage = TRUE
)
r
summary(r)
as.data.frame(r)
summary(as.data.frame(r))
```

# grouped analysis using `{dplyr}` package

```r
library(dplyr)
r <- iris %>%
  group_by(Species) %>%
  report()
r
summary(r)
as.data.frame(r)
summary(as.data.frame(r))
```

---

**report.compare_performance**

*Reporting models comparison*

**Description**

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

**Usage**

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

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

Description

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

Usage

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

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

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

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

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

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

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

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

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

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

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

## Default S3 method:
report_text(x, ...)

Arguments

x 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.estimate_contrasts

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

Methods:
- as.report()

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

Examples

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))

report.estimate_contrasts

Reporting estimate_contrasts objects

Description

Create reports for estimate_contrasts objects.

Usage

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

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

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

Arguments

x Object of class estimate_contrasts.
...
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

```r
library(modelbased)
model <- lm(Sepal.Width ~ Species, data = iris)
contr <- estimate_contrasts(model)
report(contr)
```
Reporting htest objects (Correlation, t-test...)

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

# t-tests
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), verbose = FALSE)
report(t.test(iris$Sepal.Width, mu = 1))

# Correlations
report(cor.test(iris$Sepal.Width, iris$Sepal.Length))
Description

Create a report for lavaan objects.

Usage

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

## 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

```r
# Structural Equation Models (SEM)
library(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
suppressWarnings(report_table(model))
suppressWarnings(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'
```

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

```r
```
Arguments

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

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
library(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 session information

Description

Report R environment (packages, system, etc.)

Usage

## 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 slgNificance 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

```r
# Bayesian models
library(rstanarm)
model <- suppressWarnings(stan_glm(mpg ~ qsec + wt, data = mtcars, refresh = 0, iter = 500))
r <- report(model)
r
summary(r)
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 '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_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))
Miscellaneous reports

Description

Other convenient or totally useless reports.

Usage

\[
\text{report\_date}(\ldots)
\]

\[
\text{report\_story}(\ldots)
\]

Arguments

\[\ldots\]

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()
Methods:

- as.report()

Template file for supporting new models:

- report.default()

Examples

```r
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**

```r
report_effectsize(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_effectsize()`.

**Examples**

```r
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
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_effectsize(model)

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

---

**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)

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

# ANOVAs
```
# 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
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_info(model)

# Bayesian models
library(rstanarm)
model <- suppressWarnings(stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 300))
report_info(model)

---

Report intercept

**Description**

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

**Usage**

`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

library(report)

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

# Mixed models
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (I | Species), data = iris)
report_intercept(model)

# Bayesian models
library(rstanarm)
model <- suppressWarnings(stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600))
report_intercept(model)

---

report_model  Report the model type

Description

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

Usage

report_model(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

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
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_model(model)

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

---

**report_parameters**

**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**

```r
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**

```r
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)

# t-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_parameters(lm(Petal.Width ~ Species, data = iris), include_intercept = FALSE)
report_parameters(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_parameters(model)

# Bayesian models
library(rstanarm)
model <- suppressWarnings(stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600))
report_parameters(model)
```

---

**report_participants**  
*Reporting the participant data*
Description

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

Usage

```r
report_participants(
  data,
  age = NULL,
  sex = NULL,
  gender = NULL,
  education = NULL,
  country = NULL,
  race = NULL,
  participants = NULL,
  group = NULL,
  spell_n = FALSE,
  digits = 1,
  threshold = 10,
  ...
)
```

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 characters here as well (e.g., "Intersex"), but the function will group all individuals in those groups as "Other".
gender The name of the column containing the gender of the classes should be one of c("Man", "M", "Male", "Woman", "W", F", "Female", "Non-Binary", "N"). Note that you can specify other characters here as well (e.g., "Gender Fluid"), but the function will group all individuals in those groups as "Non-Binary".
education The name of the column containing education information.
country The name of the column containing country information.
race The name of the column containing race/ethnicity 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 Logical, fully spell the sample size (“Three participants” instead of “3 participants”).
digits Number of significant digits.
threshold Percentage after which to combine, e.g., countries (default is 10%, so countries that represent less than 10% will be combined in the "other" category).
... 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("Intersex", "F", "M", "M", "NA", NA),
  "Gender" = c("N", "W", "W", "M", "NA", NA)
)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", gender = "Gender",
  education = "Education"
)

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

# Country
data <- data.frame(
  "Age" = c(22, 23, 54, 21, 8, 42, 18, 32, 24, 27, 45),
  "Country" = c("USA", NA, "Canada", "Canada", "India", "Germany",
    "USA", "USA", "USA", "USA", "Canada"
  )
)report_participants(data)

# Country, control presentation treshold
report_participants(data, threshold = 5)

# Race/ethnicity
data <- data.frame(
  "Age" = c(22, 23, 54, 21, 8, 42, 18, 32, 24, 27, 45),
    "White", "Asian", "Southeast Asian", "Mixed"
  )
)
report_performance

Report the model's quality and fit indices

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

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

# Mixed models
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_performance(model)

# Bayesian models
library(rstanarm)
model <- suppressWarnings(stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600))
report_performance(model)

# Structural Equation Models (SEM)
library(lavaan)
structure <- "ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3
dem60 ~ ind60"
model <- lavaan::sem(structure, data = PoliticalDemocracy)
suppressWarnings(report_performance(model))
```
**report_random**

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

**Usage**

```r
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**

```r
# Bayesian models
library(rstanarm)
model <- stan_glm(mpg ~ disp, data = mtcars, refresh = 0, iter = 1000)
r <- report_random(model)
r
summary(r)
```

---

**report_random**  
*Report random effects and factors*

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

**Usage**

```r
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

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

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

library(brms)
model <- suppressWarnings(brm(mpg ~ disp + (1 | cyl), data = mtcars, refresh = 0, iter = 1000))
r <- report_random(model)
r
summary(r)

---

`report_s`

Report S- and p-values in easy language.

Description

Reports interpretation of S- and p-values in easy language.

Usage

`report_s(s = NULL, p = NULL, test_value = 0, test_parameter = "parameter")`

Arguments

- **s**: An S-value. Either `s` or `p` must be provided.
- **p**: A p-value. Either `s` or `p` must be provided.
The value of the test parameter under the null hypothesis.

The name of the test parameter under the null hypothesis.

A string with the interpretation of the S- or p-value.

Examples

```r
report_s(s = 1.5)
report_s(p = 0.05)
```

Description

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

Usage

```r
report_sample(
  data,
  group_by = NULL,
  centrality = "mean",
  ci = NULL,
  ci_method = "wilson",
  ci_correct = FALSE,
  select = NULL,
  exclude = NULL,
  weights = NULL,
  total = TRUE,
  digits = 2,
  n = FALSE,
  ...
)
```

Arguments

- **data**
  A data frame for which descriptive statistics should be created.

- **group_by**
  Character vector, indicating the column(s) for possible grouping of the descriptive table. Note that weighting (see weights) does not work with more than one grouping column.

- **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.
ci
Level of confidence interval for relative frequencies (proportions). If not NULL, confidence intervals are shown for proportions of factor levels.

ci_method
Character, indicating the method how to calculate confidence intervals for proportions. Currently implemented methods are "wald" and "wilson". Note that "wald" can produce intervals outside the plausible range of [0, 1], and thus it is recommended to prefer the "wilson" method. The formulae for the confidence intervals are:

- "wald":

\[ p \pm z \sqrt{\frac{p(1 - p)}{n}} \]

- "wilson":

\[ \frac{2np + z^2 \pm z \sqrt{z^2 + 4npq}}{2(n + z^2)} \]

where \( p \) is the proportion (of a factor level), \( q \) is 1 - \( p \), \( z \) is the critical z-score based on the interval level, and \( n \) is the length of the vector (cf. Newcombe 1998, Wilson 1927).

ci_correct
Logical, if TRUE, applies continuity correction. See Newcombe 1998 for different correction-methods based on the chosen ci_method.

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.

n
Logical, actual sample size used in the calculation of the reported descriptive statistics (i.e., without the missing values).

...
Arguments passed to or from other methods.

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

References

Examples

```r
library(report)

report_sample(iris[, 1:4])
report_sample(iris, select = c("Sepal.Length", "Petal.Length", "Species"))
report_sample(iris, group_by = "Species")
report_sample(airquality, group_by = "Month", n = TRUE, total = FALSE)

# confidence intervals for proportions
set.seed(123)
d <- data.frame(x = factor(sample(letters[1:3], 100, TRUE, c(0.01, 0.39, 0.6))))
report_sample(d, ci = 0.95, ci_method = "wald") # ups, negative CI
report_sample(d, ci = 0.95, ci_method = "wilson") # negative CI fixed
report_sample(d, ci = 0.95, ci_correct = TRUE) # continuity correction
```

---

**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$Species)
report_statistics(iris)
```
# h-tests
report_statistics(t.test(iris$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_statistics(glm(vs ~ disp, data = mtcars, family = "binomial"))

# Mixed models
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_statistics(model)

# Bayesian models
library(rstanarm)
model <- suppressWarnings(stan_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600))
report_statistics(model)

---

**report_table**  
Report a descriptive table

**Description**

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

**Usage**

```r
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

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

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

# t-tests
report_table(t.test(mtcars$mpg ~ mtcars$am))

# 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
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
report_table(model)

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

# Structural Equation Models (SEM)
library(lavaan)
structure <- "ind60 =~ x1 + x2 + x3
    dem60 =~ y1 + y2 + y3
    dem60 ~ ind60"
model <- lavaan::sem(structure, data = PoliticalDemocracy)
suppressWarnings(report_table(model))
```
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 \texttt{report()}).

Usage

\begin{verbatim}
report_text(x, table = NULL, ...)
\end{verbatim}

Arguments

\begin{itemize}
\item \textit{x}  The R object that you want to report (see list of of supported objects above).
\item \textit{table}  A table obtained via \texttt{report_table()}. If not provided, will run it.
\item \textit{...}  Arguments passed to or from other methods.
\end{itemize}

Value

An object of class \texttt{report_text()}. 

Examples

\begin{verbatim}
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)

# h-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)
\end{verbatim}
library(lme4)
model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)

r <- report_text(model)
r
summary(r)

# Bayesian models
library(rstanarm)
model <- suppressWarnings(stan_glm(mpg ~ cyl + wt, data = mtcars, refresh = 0, iter = 600))

r <- report_text(model)
r
summary(r)
Index

ANO Vas, 8
as_report (as_report_text), 3
as_report (), 9, 11, 13, 15, 18, 21, 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, 8
Bayesian models, 8
cite_citation (format_citation), 7
cite_easystats, 4
cite_packages (report.sessionInfo), 29
clean_citation (format_citation), 7
Correlations and t-tests, 8
Dataframes and vectors, 8
effectsize::effectsize(), 28
format_algorithm, 6
format_citation, 7
format_formula (format_algorithm), 6
format_model (format_algorithm), 6
interpret_bf, 12
is_report (as_report_text), 3
mad, 16
Mixed models, 8
Model comparison, 8
performance(), 8
performance::compare_performance(), 17, 32
print.cite_easystats (cite_easystats), 4
Regression models, 8
report, 3, 8
report (), 11, 13, 14, 17, 18, 20, 22, 23, 25, 28, 30–32, 35–39, 43, 45, 49, 50, 52
report.aov, 10
report.bayesfactor_inclusion (report.bayesfactor_models), 12
report.bayesfactor_models, 12
report.brmsfit, 14
report.character, 15
report.compare_performance, 17
report.data.frame (report.character), 15
report.default, 19
report.default (), 9, 11, 13, 15, 18, 21, 22, 24, 26, 28, 31, 33, 35
report.estimate_contrasts, 21
report.factor (report.character), 15
report.htest, 23
report.lavaan, 25
report.lm, 26
report.numeric (report.character), 15
report.sessionInfo, 29
report.stanreg, 30
report.test_performance, 32
report_date, 34
report_date (), 9, 11, 13, 15, 18, 21, 22, 24, 26, 28, 31, 33, 35
report_effectsize, 35
report_effectsize(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33–35
report_effectsize.aov (report.aov), 10
report_effectsize.default (report.default), 19
report_effectsize.htest (report.htest), 23
report_effectsize.lm (report.lm), 26
report_info, 36
report_info(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34, 36
report_info.aov (report.aov), 10
report_info.default (report.default), 19
report_info.htest (report.htest), 23
report_info.lm (report.lm), 26
report_intercept, 37
report_intercept(), 37
report_intercept.default
(report.default), 19
report_intercept.lm (report.lm), 26
report_model, 38
report_model(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34
report_model.aov (report.aov), 10
report_model.default (report.default), 19
report_model.htest (report.htest), 23
report_model.lm (report.lm), 26
report_packages (report.sessionInfo), 29
report_packages(), 9, 11, 13, 15, 18, 21, 22, 24, 25, 28, 31, 33, 34
report_parameters, 39
report_parameters(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34
report_parameters.aov (report.aov), 10
report_parameters.compare_performance
(report.compare_performance), 17
report_parameters.default
(report.default), 19
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(), 9, 11, 13, 15, 18, 21, 22, 24, 26, 28, 31, 33, 34
report_performance, 43
report_performance(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34, 44
report_performance.default
(report.default), 19
report_performance.lavaan
(report.lavaan), 25
report_performance.lm (report.lm), 26
report_priors, 44
report_priors(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34, 45
report_priors.default (report.default), 19
report_random, 45
report_random(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34, 45
report_random.default (report.default), 19
report_random.merMod (report.lm), 26
report_sample, 47
report_sample(), 9, 11, 13, 15, 18, 21, 22, 24, 26, 28, 31, 33, 35
report_statistics, 49
report_statistics(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34, 49
report_statistics.aov (report.aov), 10
report_statistics.compare_performance
(report.compare_performance), 17
report_statistics.default
(report.default), 19
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(), 9, 11, 13, 15, 18, 20, 22, 24, 25, 28, 31, 33, 34
report_table, 50
report_table(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34, 50
report_table.aov (report.aov), 10
report_table.compare_performance
(report.compare_performance), 17
report_table.default (report.default), 19
report_table.estimate_contrasts
(report.estimate_contrasts), 21
report_table.htest (report.htest), 23
report_table.lm (report.lm), 26
report_table.test_performance
(report.test_performance), 32  
report_text, 52  
report_text(), 9, 11, 13, 14, 18, 20, 22, 24, 25, 28, 31, 33, 34, 52  
report_text.aov(report.aov), 10  
report_text.compare_performance  
  (report.compare_performance), 17  
report_text.default(report.default), 19  
report_text.estimate_contrasts  
  (report.estimate_contrasts), 21  
report_text.htest(report.htest), 23  
report_text.lm(report.lm), 26  
report_text.test_performance  
  (report.test_performance), 32  

sd, 16  
sessionInfo, 30  
SEEXIT documentation, 14, 30  
Structural Equation Models (SEM), 8  
summary.cite_easystats  
  (cite_easystats), 4  
System and packages, 8