Package 'report'

July 10, 2024

Type Package

Title Automated Reporting of Results and Statistical Models

Version 0.5.9

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

URL https://easystats.github.io/report/

BugReports https://github.com/easystats/report/issues

Depends R (>= 3.6)

Imports bayestestR (>= 0.13.2), effectsize (>= 0.8.8), insight (>= 0.20.1), parameters (>= 0.22.0), performance (>= 0.12.0), datawizard (>= 0.11.0), stats, tools, utils

Suggests BayesFactor, brms, ivreg, knitr, lavaan, lme4, dplyr, rmarkdown, rstanarm, survival, modelbased, emmeans, testthat (>= 3.2.1)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

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' 'report.BFBayesFactor.R' 'utils_combine_tables.R' 'report.lm.R' 'report.MixMod.R'

'report_text.R' 'report.R' 'report.htest.R' 'report.aov.R'
'report.bayesfactor_models.R' 'report.lme4.R'
'report.stanreg.R' 'report.brmsfit.R' 'report.character.R'
'report.compare.loo.R' 'report.compare_performance.R'
'report.coxph.R' 'report.data.frame.R' 'report.default.R'
'report.estimate_contrasts.R' 'report.factor.R' 'report.glm.R'
'report.glmmTMB.R' 'report.ivreg.R' 'report.lavaan.R'
'report.lme.R' 'report.numeric.R' 'report.sessionInfo.R'
'report.survreg.R' 'report.test_performance.R'
'report.zeroinfl.R' 'report_effectsize.R' 'report_htest_chi2.R'
'report_htest_cor.R' 'report_htest_fisher.R'
'report_htest_friedman.R' 'report_htest_kruskal.R'
'report_htest_ttest.R' 'report_htest_wilcox.R' 'report_info.R'
'report_intercept.R' 'report_misc.R' 'report_model.R'
'report_parameters.R' 'report_participants.R'
'report_performance.R' 'report_priors.R' 'report_random.R'
'report_s.R' 'report_sample.R' 'report_statistics.R'
'report_table.R' 'utils_error_message.R' 'utils_grouped_df.R'
'utils_misspelled_variables.R' 'zzz.R'

NeedsCompilation no

Author Dominique Makowski [aut] (<https://orcid.org/0000-0001-5375-9967>, @Dom_Makowski),
Daniel Lüdecke [aut] (<https://orcid.org/0000-0002-8895-3206>, @strengejacke),
Indrajeet Patil [aut] (<https://orcid.org/0000-0003-1995-6531>, @patilindrajeets),
Rémi Thériault [aut, cre] (<https://orcid.org/0000-0003-4315-6788>, @rempsyc),
Mattan S. Ben-Shachar [aut] (<https://orcid.org/0000-0002-4287-4801>, @mattansb),
Brenton M. Wiernik [aut] (<https://orcid.org/0000-0001-9560-6336>, @bmwiernik),
Rudolf Siegel [ctb] (<https://orcid.org/0000-0002-6021-804X>),
Camden Bock [ctb] (<https://orcid.org/0000-0002-3907-7748>)

Repository CRAN

Date/Publication 2024-07-10 17:40:02 UTC

Contents

`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.
...	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* pacakges 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

```
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)


# 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_model(model)


# Mixed models
library(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

```
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)

citation <- "Makowski, D., Ben-Shachar, M. S., Patil, I., & Ludecke, D. (2020).
Methods and Algorithms for Correlation Analysis in R. Journal of Open Source
Software, 5(51), 2306."

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())
```

---

report *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 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

```
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

```
## 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

```
library(bayestestR)
# Bayes factor - models
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)
BFmodels <- bayesfactor_models(mo1, mo2, mo3, 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.BFBayesFactor` *Reporting* `BFBayesFactor` *objects from the* `BayesFactor` *package*

Description

Interpretation of the Bayes factor output from the `BayesFactor` package.

Usage

```
## S3 method for class 'BFBayesFactor'
report(x, h0 = "H0", h1 = "H1", ...)

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

Arguments

`x`	An object of class `BFBayesFactor`.
`h0`, `h1`	Names of the null and alternative hypotheses.
`...`	Other arguments to be passed to effectsize::interpret_bf and insight::format_bf.
`table`	A `parameters` table (this argument is meant for internal use).

Examples

```
library(BayesFactor)

rez <- BayesFactor::ttestBF(iris$Sepal.Width, iris$Sepal.Length)
report_statistics(rez, exact = TRUE) # Print exact BF
report(rez, h0 = "the null hypothesis", h1 = "the alternative")

rez <- BayesFactor::correlationBF(iris$Sepal.Width, iris$Sepal.Length)
report(rez)
```

`report.brmsfit` *Reporting Bayesian Models from brms*

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 '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

```
# 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

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

`report.compare.loo` *Reporting Bayesian Model Comparison*

Description

Automatically report the results of Bayesian model comparison using the `loo` package.

Usage

```
## S3 method for class 'compare.loo'
report(x, include_IC = TRUE, include_ENP = FALSE, ...)
```

Arguments

`x`	An object of class brms::loo_compare.
`include_IC`	Whether to include the information criteria (IC).
`include_ENP`	Whether to include the effective number of parameters (ENP).
`...`	Additional arguments (not used for now).

Details

The rule of thumb is that the models are "very similar" if |elpd_diff| (the absolute value of elpd_diff) is less than 4 (Sivula, Magnusson and Vehtari, 2020). If superior to 4, then one can use the SE to obtain a standardized difference (Z-diff) and interpret it as such, assuming that the difference is normally distributed. The corresponding p-value is then calculated as `2 * pnorm(-abs(Z-diff))`. However, note that if the raw ELPD difference is small (less than 4), it doesn't make much sense

to rely on its standardized value: it is not very useful to conclude that a model is much better than another if both models make very similar predictions.

Value

Objects of class report_text().

Examples

```
library(brms)

m1 <- brms::brm(mpg ~ qsec, data = mtcars)
m2 <- brms::brm(mpg ~ qsec + drat, data = mtcars)
m3 <- brms::brm(mpg ~ qsec + drat + wt, data = mtcars)

x <- brms::loo_compare(
  brms::add_criterion(m1, "loo"),
  brms::add_criterion(m2, "loo"),
  brms::add_criterion(m3, "loo"),
  model_names = c("m1", "m2", "m3")
)
report(x)
report(x, include_IC = FALSE)
report(x, include_ENP = TRUE)
```

---

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)
```

---

report.default *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

```
## 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_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

```
library(modelbased)
model <- lm(Sepal.Width ~ Species, data = iris)
contr <- estimate_contrasts(model)
report(contr)
```

`report.htest` *Reporting* `htest` *objects (Correlation, t-test...)*

Description

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

Usage

```
## 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))
```

report.lavaan *Reports of Structural Equation Models (SEM)*

Description

Create a report for lavaan objects.

Usage

```
## 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

```
# 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

```
## 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'
```

```
report_table(x, include_effectsize = TRUE, ...)

## S3 method for class 'lm'
report_statistics(
  x,
  table = NULL,
  include_effectsize = TRUE,
  include_diagnostic = TRUE,
  ...
)

## S3 method for class 'lm'
report_parameters(
  x,
  table = NULL,
  include_effectsize = TRUE,
  include_intercept = TRUE,
  ...
)

## S3 method for class 'lm'
report_intercept(x, table = NULL, ...)

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

## S3 method for class 'lm'
report_performance(x, table = NULL, ...)

## S3 method for class 'lm'
report_info(
  x,
  effectsize = NULL,
  include_effectsize = FALSE,
  parameters = NULL,
  ...
)

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

## S3 method for class 'merMod'
report_random(x, ...)
```

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

```
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 R environment (packages, system, etc.)*

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 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

```
# 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)
```

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

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

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

`report_date` *Miscellaneous reports*

Description

Other convenient or totally useless reports.

Usage

```
report_date(...)

report_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:

- as.report()

Template file for supporting new models:

- report.default()

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
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")
```

`report_info` *Report additional information*

Description

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

Usage

```
report_info(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_info()`.

Examples

```
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
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` *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 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 + (1 | 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

```
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

```
report_parameters(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

A `vector`.

Examples

```
library(report)

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

```
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_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

```
report_participants(
  data,
  age = NULL,
  sex = NULL,
```

```
  gender = NULL,
  education = NULL,
  country = NULL,
  race = NULL,
  participants = NULL,
  by = NULL,
  spell_n = FALSE,
  digits = 1,
  threshold = 10,
  group = NULL,
  ...
)
```

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).
by	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).
group	Deprecated. Use by instead.
...	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),
  "Sex" = c("Intersex", "F", "F", "M", "M", "M", "F", "F", "F", "F", "F"),
  "Gender" = c("N", "W", "W", "M", "M", "M", "W", "W", "W", "W", "W"),
  "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),
  "Sex" = c("Intersex", "F", "F", "M", "M", "M", "F", "F", "F", "F", "F"),
  "Gender" = c("N", "W", "W", "M", "M", "M", "W", "W", "W", "W", "W"),
  "Race" = c(
    "Black", NA, "White", "Asian", "Black", "Arab", "Black",
    "White", "Asian", "Southeast Asian", "Mixed"
  )
)
report_participants(data)

# Race/ethnicity, control presentation treshold
report_participants(data, threshold = 5)
```

```
# Repeated measures data
data <- data.frame(
  "Age" = c(22, 22, 54, 54, 8, 8),
  "Sex" = c("I", "F", "M", "M", "F", "F"),
  "Gender" = c("N", "W", "W", "M", "M", "M"),
  "Participant" = c("S1", "S1", "s2", "s2", "s3", "s3")
)
report_participants(data, age = "Age", sex = "Sex", gender = "Gender", participants = "Participant")

# Grouped data
data <- data.frame(
  "Age" = c(22, 22, 54, 54, 8, 8, 42, 42),
  "Sex" = c("I", "I", "M", "M", "F", "F", "F", "F"),
  "Gender" = c("N", "N", "W", "M", "M", "M", "Non-Binary", "Non-Binary"),
  "Participant" = c("S1", "S1", "s2", "s2", "s3", "s3", "s4", "s4"),
  "Condition" = c("A", "A", "A", "A", "B", "B", "B", "B")
)

report_participants(data,
  age = "Age",
  sex = "Sex",
  gender = "Gender",
  participants = "Participant",
  by = "Condition"
)

# Spell sample size
paste(
  report_participants(data, participants = "Participant", spell_n = TRUE),
  "were recruited in the study by means of torture and coercion."
)
```

`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 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

```
# 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_priors *Report priors of Bayesian models*

Description

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

Usage

```
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

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

```
report_random(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_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.
`test_value`	The value of the test parameter under the null hypothesis.
`test_parameter`	The name of the test parameter under the null hypothesis.

Value

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

Examples

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

`report_sample` *Sample Description*

Description

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

Usage

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

Arguments

`data`	A data frame for which descriptive statistics should be created.
`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, it 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).
group_by	Deprecated. Use by instead.
...	Arguments passed to or from other methods.

Value

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

References

- Newcombe, R. G. (1998). Two-sided confidence intervals for the single proportion: comparison of seven methods. Statistics in Medicine. 17 (8): 857–872
- Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association. 22 (158): 209–212

Examples

```
library(report)

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

```
report_statistics(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_statistics()`.

Examples

```
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

```
report_table(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_table()`.

Examples

```
# 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)

# h-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_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

```
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)



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