Package ‘report’

February 22, 2022

Type Package
Title Automated Reporting of Results and Statistical Models
Version 0.5.1
Maintainer Dominique Makowski <dom.makowski@gmail.com>
Description The aim of the 'report' package is to bridge the gap between R’s output and the formatted results contained in your manuscript. This package converts statistical models and data frames into textual reports suited for publication, ensuring standardization and quality in results reporting.
License GPL-3 | file LICENSE
BugReports https://github.com/easystats/report/issues
Imports bayestestR (>= 0.11.5), effectsize (>= 0.6.0), insight (>= 0.16.0), parameters (>= 0.16.0), performance (>= 0.8.0), datawizard (>= 0.2.3), stats, tools, utils
Suggests BayesFactor, brms, htr, ivreg, knitr, lavaan, lme4, logspline, dplyr, rmarkdown, rstanarm, spelling, survival, testthat
VignetteBuilder knitr
Encoding UTF-8
Language en-US
RoxygenNote 7.1.2
Config/testthat/edition 3
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

```r
cite_easystats()
```

Value

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

Examples

```r
cite_easystats()
summary(cite_easystats())
as.data.frame(cite_easystats())
```
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 supported objects above).
what The name of the item returned by insight::find_formula.

Value

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

Examples

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

cite_citation(citation)
clean_citation(citation())
```
**Description**

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

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

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

**Usage**

`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**

**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(mpg ~ am, data = mtcars)
r <- report(model)

# Text
r
summary(r)

# Tables
```
**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)

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(report)

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

if (require("bayestestR")) {
  # Bayes factor - models
  BFmodels <- bayesfactor_models(mo1, mo2, mo3, denominator = mo0)

  r <- report(BFmodels)
  r
  as.data.frame(r)

  # Bayes factor - inclusion
  inc_bf <- bayesfactor_inclusion(BFmodels, prior_odds = c(1, 2, 3), match_models = TRUE)

  r <- report(inc_bf)
  r
  as.data.frame(r)
}
```
Description

Create reports for Bayesian models. The description of the parameters follows the Sequential Effect eXistence and sIgnificance 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_system()`
- `report_packages()
- `report_participants()
- `report_sample()
- `report_date()`
Methods:

- `as.report()`

Template file for supporting new models:

- `report.default()`

Examples

```r
library(report)

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

---

**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",
  ...)

## S3 method for class 'data.frame'
report(
  x,
  n = FALSE,
  centrality = "mean",
  dispersion = TRUE,
  ...)
```
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

library(report)

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

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

# grouped analysis using `{dplyr}` package
if (require("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

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

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

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

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

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

- Object of class NEW OBJECT.
- Arguments passed to or from other methods.
- 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(report)
library(performance)

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

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

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

---

**Description**

Template file to add report support for new objects. Check-out the vignette on [Supporting New Models](https://example.com).

**Usage**

```r
## 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 default

```r
report.default
```

### Code Snippet

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

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

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

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

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

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

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

### Arguments

- `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

```r
library(report)

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

---

**report.htest**  
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:
Template file for supporting new models:

- `as.report()`

Examples

```r
# 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))
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
library(report)

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

  # Specific reports
  report_table(model)
  report_performance(model)
}
```
report.lm

Reporting (General) Linear Models

Description

Create reports for (general) linear models.

Usage

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

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

## S3 method for class 'lm'
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, ...)
```

```
Arguments

x
include_effectsize
effectsize_method
table
include_diagnostic
include_intercept
effectsize
parameters

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
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
r <- report(model)
r
  summary(r)
as.data.frame(r)
  summary(as.data.frame(r))
}
Report R environment (packages, system, etc.)

Description

Report R environment (packages, system, etc.)

Usage

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

report_packages(session = NULL, include_R = TRUE, ...)

cite_packages(session = NULL, include_R = TRUE, ...)

report_system(session = NULL)
```

Arguments

- `x`: The R object that you want to report (see list of supported objects above).
- `...`: Arguments passed to or from other methods.
- `session`: A `sessionInfo` object.
- `include_R`: Include R in the citations.

Value

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

An object of class `report()`.

Examples

```r
library(report)

session <- sessionInfo()

r <- report(session)

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

summary(as.data.frame(r))

# Convenience functions
report_packages(include_R = FALSE)
cite_packages(prefix = "> ")
```
report.stanreg

---

**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**

```r
# 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.test_performance**

- report.sample()
- report.date()

Methods:
- as.report()

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

**Examples**

```r
library(report)

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

---

**report.test_performance**

*Reporting models comparison*

**Description**

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

**Usage**

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

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

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

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

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

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

Value

An object of class `report()`.

See Also

Specific components of reports (especially for stats models):

- `report_table()`
- `report_parameters()`
- `report_statistics()`
- `report_effectsize()`
- `report_model()`
- `report_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)
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**

```r
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_effectsize

- 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, ...)

Report additional information

Description

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

Usage

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

Arguments

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

Value

An object of class `report_info()`.

Examples

```r
library(report)

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

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

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

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

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

An object of class `report_intercept()`. 
Examples

library(report)

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

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

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

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

---

**Report the parameters of a model**

**Description**

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

**Usage**

`report_parameters(x, ...)`

**Arguments**

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

**Value**

A vector.

**Examples**

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

# Bayesian models
if (require("rstanarm")) {
  model <- 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,
  participants = NULL,
)```
group = NULL,
spell_n = FALSE,
digits = 1,
...)

Arguments

data A data frame.
age The name of the column containing the age of the participant.
sex The name of the column containing the sex of the participant. The classes should be one of c("Male","M","Female","F"). Note that you can specify other 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 participant. The classes should be one of c("Man","M","Woman","F","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.
participants The name of the participants' identifier column (for instance in the case of repeated measures).
group A character vector indicating the name(s) of the column(s) used for stratified description.
spell_n Fully spell the sample size ("Three participants" instead of "3 participants").
digits Number of significant digits.
... Arguments passed to or from other methods.

Value

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

Examples

library(report)
data <- data.frame(
  "Age" = c(22, 23, 54, 21, 8, 42),
  "Sex" = c("Intersex", "F", "M", "M", "M", "F"),
  "Gender" = c("N", "W", "W", "M", "M", "M")
)
report_participants(data, age = "Age", sex = "Sex")

# Years of education (relative to high school graduation)
data$Education <- c(0, 8, -3, -5, 3, 5)
report_participants(data,
  age = "Age", sex = "Sex", gender = "Gender",
education = "Education")
# Education as factor
```r
data$Education2 <- c("Bachelor", "PhD", "Highschool", "Highschool", "Bachelor", "Bachelor")
```
```r
report_participants(data, age = "Age", sex = "Sex", gender = "Gender", education = "Education2")
```

# Repeated measures data
```r
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")
)
```
```r
report_participants(data, age = "Age", sex = "Sex", gender = "Gender", participants = "Participant")
```

# Grouped data
```r
data <- data.frame(
  "Age" = c(22, 22, 54, 54, 8, 8, 42, 42),
  "Sex" = c("I", "I", "M", "M", "F", "F", "F", "F"),
  "Participant" = c("S1", "S1", "s2", "s2", "s3", "s3", "s4", "s4"),
)
```
```r
report_participants(data, age = "Age", sex = "Sex", gender = "Gender", participants = "Participant", group = "Condition")
```

# Spell sample size
```r
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

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

Arguments

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

Value

An object of class `report_performance()`.

Examples

```r
library(report)

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

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

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

# Structural Equation Models (SEM)
if (require("lavaan") && packageVersion("effectsize") >= "0.6.0.1" && FALSE) {
  structure <- " ind60 =~ x1 + x2 + x3
  dem60 =~ y1 + y2 + y3
  dem60 ~ ind60 "
  model <- lavaan::sem(structure, data = PoliticalDemocracy)
  report_performance(model)
}
```
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 supported objects above).
...        Arguments passed to or from other methods.

Value
An object of class report_priors().

Examples
library(report)

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

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

Usage
report_random(x, ...)

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

An object of class `report_random()`.

Examples

```r
library(report)

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

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

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

## End(Not run)
```

---

<table>
<thead>
<tr>
<th>report_sample</th>
<th>Sample Description</th>
</tr>
</thead>
</table>

Description

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

Usage

```r
report_sample(
  data,
  group_by = NULL,
  centrality = "mean",
  select = NULL,
  exclude = NULL,
  weights = NULL,
)```
Arguments

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

Value

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

Examples

```r
library(report)

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

Description

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

Usage

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

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

Value

An object of class `report_statistics()`.

Examples

```r
library(report)

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

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

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

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

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

# Bayesian models
if (require("rstanarm")) {
  model <- 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 supported objects above).
- `...` Arguments passed to or from other methods.

**Value**

An object of class `report_table()`.

**Examples**

```r
library(report)

# 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(mpg ~ am, data = mtcars))

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

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

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

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

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

---

report_text  

Report a textual description of an object

Description

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

Usage

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

Arguments

x          The R object that you want to report (see list of 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)

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

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

ANOVA, 7
as.report(as.report_text), 3
as.report(), 8, 10, 12, 14, 17, 20, 22, 23, 26, 29, 30, 32
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
as.report_text, 3
Bayes factors, 7
Bayesian models, 7
cite_citation(format_citation), 6
cite_easystats, 4
cite_packages(report.sessionInfo), 27
clean_citation(format_citation), 6
Correlations and t-tests, 7
Dataframes and vectors, 7
effectsize::effectsize(), 25
format_algorithm, 5
format_citation, 6
format_formula(format_algorithm), 5
format_model(format_algorithm), 5
interpret_bf, 11
is_report(as.as.report_text), 3
mad, 15
Mixed models, 7
Model comparison, 7
performance(), 7
performance::compare_performance(), 16, 29
Regression models, 7
report(), 3, 7
report(), 9, 11, 13, 15, 17, 19, 21, 22, 25, 27, 28, 30, 32–36, 39, 41, 43, 44, 46
report.aov, 9
report.bayesfactor_inclusion(report.bayesfactor_models), 11
report.bayesfactor_models, 11
report.brmsfit, 13
report.character, 14
report.compare_performance, 16
report.data.frame(report.character), 14
report.default, 18
report.default(), 8, 10, 12, 14, 17, 20, 22, 23, 26, 29, 30, 32
report.factor(report.character), 14
report.htest, 20
report.lavaan, 22
report.lm, 24
report.numeric(report.character), 14
report.sessionInfo, 27
report.stanreg, 28
report.test_performance, 29
report_date, 31
report_date(), 8, 10, 12, 13, 17, 20, 21, 23, 26, 29, 30, 32
report_effectsize, 32
report_effectsize(), 8, 10, 12, 13, 17, 19, 21, 23, 25, 28, 30, 32, 33
report_effectsize.aov(report.aov), 9
report_effectsize.default(report.default), 18
report_effectsize.htest(report.htest), 20
report_effectsize.lm(report.lm), 24
report_info, 33
INDEX

report_info(), 8, 10, 12, 13, 17, 19, 21, 23, 26, 28, 30, 32, 34
report_info.aov (report.aov), 9
report_info.default (report.default), 18
report_info.htest (report.htest), 20
report_info.lm (report.lm), 24
report_intercept(), 34
report_intercept.default (report.default), 18
report_intercept.lm (report.lm), 24
report_model(), 8, 10, 12, 13, 17, 19, 21, 23, 25, 28, 30, 32
report_model.aov (report.aov), 9
report_model.default (report.default), 18
report_model.htest (report.htest), 20
report_model.lm (report.lm), 24
report_packages (report.sessionInfo), 27
report_packages(), 8, 10, 12, 13, 17, 20, 21, 23, 26, 28, 30, 32
report_parameters(), 8, 10, 12, 13, 17, 19, 21, 23, 25, 28, 30, 31
report_parameters.aov (report.aov), 9
report_parameters.compare_performance (report.compare_performance), 16
report_parameters.default (report.default), 18
report_parameters.htest (report.htest), 20
report_parameters.lm (report.lm), 24
report_parameters.test_performance (report.test_performance), 29
report_participants(), 8, 10, 12, 13, 17, 20, 21, 23, 26, 28, 30, 32
report_performance(), 8, 10, 12, 13, 17, 19, 21, 23, 26, 28, 30, 32, 40
report_performance.default (report.default), 18
report_performance.lavaan (report.lavaan), 22
report_performance.lm (report.lm), 24
report_priors(), 8, 10, 12, 13, 17, 19, 21, 23, 25, 28, 30, 32, 41
report_priors.default (report.default), 18
report_random, 41
report_random(), 8, 10, 12, 13, 17, 19, 21, 23, 25, 28, 30, 32, 42
report_random.default (report.default), 18
report_random.merMod (report.lm), 24
report_sample(), 8, 10, 12, 13, 17, 20, 21, 23, 26, 29, 30, 32
report_sample.aov (report.aov), 9
report_statsitics(), 8, 10, 12, 13, 17, 19, 21, 23, 25, 28, 30, 32, 44
report_statistics.aov (report.aov), 9
report_statistics.compare_performance (report.compare_performance), 16
report_statistics.default (report.default), 18
report_statistics.htest (report.htest), 20
report_statistics.lm (report.lm), 24
report_statistics.test_performance (report.test_performance), 29
report_story (report.date), 31
report_system (report.sessionInfo), 27
report_system(), 8, 10, 12, 13, 17, 19, 21, 23, 26, 28, 30, 32
report_table(), 8, 10, 11, 13, 17, 19, 21, 23, 25, 28, 30, 31, 45
report_table.aov (report.aov), 9
report_table.compare_performance (report.compare_performance), 16
report_table.default (report.default), 18
report_table.htest (report.htest), 20
report_table.lm (report.lm), 24
report_table.test_performance (report.test_performance), 29
report_text(), 46
report_text.aov (report.aov), 9
report_text.compare_performance
  (report.compare_performance), 16
report_text.default (report.default), 18
report_text.h.test (report.h.test), 20
report_text.lm (report.lm), 24
report_text.test.performance
  (report.test.performance), 29

sd, 15
sessionInfo, 27
SEIXT documentation, 13, 28
Structural Equation Models (SEM), 7
System and packages, 7