

# Package ‘report’

September 30, 2021

**Type** Package

**Title** Automated Reporting of Results and Statistical Models

**Version** 0.4.0

**Maintainer** Dominique Makowski <dom.makowski@gmail.com>

**Description** The aim of the 'report' package is to bridge the gap between R's output and the formatted results contained in your manuscript. This package converts statistical models and data frames into textual reports suited for publication, ensuring standardization and quality in results reporting.

**License** GPL-3 | file LICENSE

**URL** <https://easystats.github.io/report/>

**BugReports** <https://github.com/easystats/report/issues>

**Imports** bayestestR (>= 0.10.5), effectsize (>= 0.4.5), insight (>= 0.14.1), parameters (>= 0.14.0), performance (>= 0.7.2), datawizard, stats, tools, utils

**Suggests** BayesFactor, brms, dplyr, httr, ivreg, knitr, lavaan, lme4, logspline, rmarkdown, rstanarm, spelling, survival, testthat

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.1.2

**Config/testthat/edition** 3

**Collate** 'cite\_easystats.R' 'format\_algorithm.R' 'format\_citation.R' 'format\_formula.R' 'format\_model.R' 'format\_text.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\_performance.R' 'report.data.frame.R' 'report.default.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\_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\_sample.R' 'report\_statistics.R' 'report\_table.R'  
 'utils\_data.R' 'utils\_error\_message.R' 'utils\_grouped\_df.R'

**NeedsCompilation** no

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

**Repository** CRAN

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

## R topics documented:

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

report_model . . . . .	38
report_parameters . . . . .	39
report_participants . . . . .	40
report_performance . . . . .	42
report_priors . . . . .	43
report_random . . . . .	44
report_sample . . . . .	45
report_statistics . . . . .	46
report_table . . . . .	47
report_text . . . . .	48

<b>Index</b>	<b>50</b>
--------------	-----------

---

as.report_text	<i>Create or test objects of class <a href="#">report</a>.</i>
----------------	--

---

## 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 <code>report_text()</code>
table	Table obtained via <code>report_table()</code>
plot	Plot obtained via <code>report_plot()</code> . Not yet implemented.
summary	Add a summary as attribute (to be extracted via <code>summary()</code> ).
prefix	The prefix to be displayed in front of each parameter.

**Value**

A report object or a TRUE/FALSE value.

---

cite_easystats	<i>Cite the easystats ecosystem</i>
----------------	-------------------------------------

---

**Description**

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

**Usage**

```
cite_easystats()
```

**Value**

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

**Examples**

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

---

data_rename	<i>Convenient dataframe manipulation functionalities</i>
-------------	--

---

### Description

Safe and intuitive functions to manipulate dataframes.

### Usage

```
data_rename(data, pattern, replacement, safe = TRUE)

data_findcols(data, pattern = NULL, starts_with = NULL, ends_with = NULL)

data_remove(data, pattern)

data_reorder(data, cols, safe = TRUE)

data_addprefix(data, pattern)

data_addsuffix(data, pattern)
```

### Arguments

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

### Value

A modified data frame.

### Examples

```
library(report)

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

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

```
# Remove columns
data_remove(iris, "Sepal.Length")

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

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

---

format_algorithm	<i>Convenient formatting of text components</i>
------------------	---

---

## 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 <code>insight::find_formula</code> .

## 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)
  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

<code>citation</code>	A character string of a citation.
<code>authorsdate</code>	Only show authors and date (remove title, journal, etc.).
<code>short</code>	If more than one authors, replace by et al.
<code>intext</code>	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())
```

---

format\_text

*Convenient text formatting functionalities*


---

**Description**

Convenience functions to manipulate and format text.

**Usage**

```
format_text(text, sep = ", ", last = " and ", width = NULL, ...)

text_fullstop(text)

text_lastchar(text, n = 1)

text_concatenate(text, sep = ", ", last = " and ")

text_paste(text, text2 = NULL, sep = ", ", ...)

text_remove(text, pattern = "", ...)

text_wrap(text, width = NULL, ...)
```

**Arguments**

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



**Value**

A character string.

**Examples**

```
library(report)

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

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

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

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

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

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

---

report

*Automatic reporting of R objects*

---

**Description**

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

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

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

**Usage**

```
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

```
library(report)

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

# Text
r
summary(r)

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

---

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 <a href="#">interpret_bf</a> ).
exact	Should very large or very small values be reported with a scientific format (e.g., 4.24e5), or as truncated values (as "> 1000" and "< 1/1000").
protect_ratio	Should values smaller than 1 be represented as ratios?
...	Arguments passed to or from other methods.

**Value**

An object of class [report\(\)](#).

**See Also**

Specific components of reports (especially for stats models):

- [report\\_table\(\)](#)
- [report\\_parameters\(\)](#)
- [report\\_statistics\(\)](#)
- [report\\_effectsize\(\)](#)
- [report\\_model\(\)](#)
- [report\\_priors\(\)](#)
- [report\\_random\(\)](#)
- [report\\_performance\(\)](#)
- [report\\_info\(\)](#)
- [report\\_text\(\)](#)

Other types of reports:

- [report\\_system\(\)](#)
- [report\\_packages\(\)](#)
- [report\\_participants\(\)](#)
- [report\\_sample\(\)](#)
- [report\\_date\(\)](#)

Methods:

- [as.report\(\)](#)

Template file for supporting new models:

- [report.default\(\)](#)

## Examples

```
library(report)

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

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

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

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

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

---

report.brmsfit

*Reporting Bayesian Models from brms*

---

## Description

Create reports for Bayesian models. The description of the parameters follows the Sequential Effect eXistence and 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**

```
library(report)

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

## End(Not run)
```



**Description**

Create reports for data frames.

**Usage**

```
## 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**

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

**Value**

An object of class `report()`.

**Examples**

```
library(report)

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

if (require("dplyr")) {
  r <- iris %>%
    dplyr::group_by(Species) %>%
    report()
  r
  summary(r)
  as.data.frame(r)
  summary(as.data.frame(r))
}
```

---

`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

<code>x</code>	Object of class NEW OBJECT.
<code>...</code>	Arguments passed to or from other methods.
<code>table</code>	Provide the output of <code>report_table()</code> 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.htest	<i>Reporting htest objects (Correlation, t-test...)</i>
--------------	---

---

## 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 <code>report_table()</code> to avoid its re-computation.
effectsize	Provide the output of <code>report_effectsize()</code> 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**

```
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))
```



---

report.lavaan	<i>Reports of Structural Equation Models (SEM)</i>
---------------	--

---

## 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

```
library(report)

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

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

---

report.lm

*Reporting (General) Linear Models*

---

## Description

Create reports for (general) linear models.

## Usage

```
## 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 <code>effectsize::effectsize()</code> .
...	Arguments passed to or from other methods.
table	Provide the output of <code>report_table()</code> 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 <code>report_effectsize()</code> to avoid its re-computation.
parameters	Provide the output of <code>report_parameters()</code> 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
if (require("lme4")) {
  model <- lme4::lmer(Sepal.Length ~ Petal.Length + (1 | Species), data = iris)
  r <- report(model)
  r
  summary(r)
  as.data.frame(r)
  summary(as.data.frame(r))
}
```

---

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

---

## Description

Report R environment (packages, system, etc.)

## Usage

```
## 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 <a href="#">sessionInfo</a> 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 <code>lm</code> or <code>glm</code> .
...	Arguments passed to or from other methods.

**Value**

An object of class `report()`.

**See Also**

Specific components of reports (especially for stats models):

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

Other types of reports:

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

Methods:

- `as.report()`

Template file for supporting new models:

- `report.default()`

**Examples**

```
library(report)

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

```
## 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 <code>report_table()</code> 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	<i>Miscellaneous reports</i>
-------------	------------------------------

---

**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
```

```

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

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

```

---

report\_info

*Report additional information*


---

## Description

Reports additional information relevant to the report (see list of supported objects in [report\(\)](#)).

## Usage

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

## Arguments

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

## Value

An object of class [report\\_info\(\)](#).

## Examples

```

library(report)

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

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

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

# Mixed models
if (require("lme4")) {

```

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

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

---

report_intercept	<i>Report intercept</i>
------------------	-------------------------

---

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

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

```
  report_intercept(model)
}
```

---

report_model	<i>Report the model type</i>
--------------	------------------------------

---

## 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 <code>report_table()</code> . 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\_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 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
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

```

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

```

## Arguments

data	A data frame.
age	The name of the column containing the age of the participant.
sex	The name of the column containing the sex of the participant. The classes should be one of <code>c("Male", "M", "Female", "F")</code> . Note that you can specify other characters here as well (e.g., "Other"), but the function will report only percentage of females, regardless of whether any category other than "Male" is present in the data.
education	The name of the column containing education information.



participants	The name of the participants' identifier column (for instance in the case of repeated measures).
group	A character vector indicating the name(s) of the column(s) used for stratified description.
spell_n	Fully spell the sample size ("Three participants" instead of "3 participants").
digits	Number of significant digits.
...	Arguments passed to or from other methods.

**Value**

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

**Examples**

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

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

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

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

# Repeated measures data
data <- data.frame(
  "Age" = c(22, 22, 54, 54, 8, 8),
  "Sex" = c("F", "F", "M", "M", "F", "F"),
  "Participant" = c("S1", "S1", "s2", "s2", "s3", "s3")
)

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

# Grouped data
data <- data.frame(
  "Age" = c(22, 22, 54, 54, 8, 8, 42, 42),
  "Sex" = c("F", "F", "M", "M", "F", "F", "M", "M"),
  "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",
  participants = "Participant",
  group = "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	<i>Report the model's quality and fit indices</i>
--------------------	---

---

## Description

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

## Usage

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

## Arguments

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

## Value

An object of class `report_performance()`.

## Examples

```

library(report)

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

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

```

```

}

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

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

```

---

report\_priors

*Report priors of Bayesian models*


---

## Description

Reports priors of Bayesian models (see list of supported objects in [report\(\)](#)).

## Usage

```
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)
}

```

---

report_random	<i>Report random effects and factors</i>
---------------	--

---

**Description**

Reports random effects of mixed models (see list of supported objects in [report\(\)](#)).

**Usage**

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

**Arguments**

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

**Value**

An object of class [report\\_random\(\)](#).

**Examples**

```
library(report)

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

---

report_sample	<i>Sample Description</i>
---------------	---------------------------

---

### Description

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

### Usage

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

### Arguments

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

### Value

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

## Examples

```
library(report)

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

---

report_statistics	<i>Report the statistics of a model</i>
-------------------	---

---

## 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 supported objects above).
table	A table obtained via <code>report_table()</code> . 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
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	<i>Report a descriptive table</i>
--------------	-----------------------------------

---

## Description

Creates tables to describe different objects (see list of supported objects in [report\(\)](#)).

## Usage

```
report_table(x, ...)
```

## Arguments

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

## Value

An object of class [report\\_table\(\)](#).

## Examples

```
library(report)

# Miscellaneous
r <- report_table(sessionInfo())
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_glm(Sepal.Length ~ Species, data = iris, refresh = 0, iter = 600)
  report_table(model, effectsize_method = "basic")
}

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

```

---

report\_text

*Report a textual description of an object*


---

## Description

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

## Usage

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

## Arguments

x	The R object that you want to report (see list of supported objects above).
table	A table obtained via <code>report_table()</code> . 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

ANOVAs, [9](#)  
as.report (as.report\_text), [3](#)  
as.report(), [11](#), [13](#), [14](#), [16](#), [20](#), [22](#), [24](#), [26](#),  
[28](#), [31](#), [33](#), [35](#)  
as.report\_effectsize (as.report\_text), [3](#)  
as.report\_info (as.report\_text), [3](#)  
as.report\_intercept (as.report\_text), [3](#)  
as.report\_model (as.report\_text), [3](#)  
as.report\_parameters (as.report\_text), [3](#)  
as.report\_performance (as.report\_text),  
[3](#)  
as.report\_priors (as.report\_text), [3](#)  
as.report\_random (as.report\_text), [3](#)  
as.report\_statistics (as.report\_text), [3](#)  
as.report\_table (as.report\_text), [3](#)  
as.report\_text, [3](#)

Bayes factors, [9](#)  
Bayesian models, [9](#)

cite\_citation (format\_citation), [7](#)  
cite\_easystats, [4](#)  
cite\_packages (report.sessionInfo), [29](#)  
clean\_citation (format\_citation), [7](#)  
Correlations and t-tests, [9](#)

data\_addprefix (data\_rename), [5](#)  
data\_addsuffix (data\_rename), [5](#)  
data\_findcols (data\_rename), [5](#)  
data\_remove (data\_rename), [5](#)  
data\_rename, [5](#)  
data\_reorder (data\_rename), [5](#)  
Dataframes and vectors, [9](#)

effectsize::effectsize(), [28](#)

format\_algorithm, [6](#)  
format\_citation, [7](#)  
format\_formula (format\_algorithm), [6](#)  
format\_model (format\_algorithm), [6](#)  
format\_text, [8](#)

interpret\_bf, [14](#)  
is.report (as.report\_text), [3](#)

mad, [18](#)  
Mixed models, [9](#)  
Model comparison, [9](#)

performance(), [9](#)  
performance::compare\_performance(), [19](#),  
[32](#)

Regression models, [9](#)  
report, [3](#), [9](#)  
report(), [12](#), [14](#), [15](#), [18](#), [19](#), [22](#), [23](#), [25](#), [28](#),  
[30–32](#), [35–39](#), [42–44](#), [46–48](#)  
report.aov, [11](#)  
report.bayesfactor\_inclusion  
(report.bayesfactor\_models), [13](#)  
report.bayesfactor\_models, [13](#)  
report.brmsfit, [15](#)  
report.character, [17](#)  
report.compare\_performance, [19](#)  
report.data.frame (report.character), [17](#)  
report.default, [21](#)  
report.default(), [11](#), [13](#), [14](#), [16](#), [20](#), [22](#), [24](#),  
[26](#), [28](#), [31](#), [33](#), [35](#)  
report.factor (report.character), [17](#)  
report.htest, [23](#)  
report.lavaan, [25](#)  
report.lm, [26](#)  
report.numeric (report.character), [17](#)  
report.sessionInfo, [29](#)  
report.stanreg, [30](#)  
report.test\_performance, [32](#)  
report\_date, [34](#)  
report\_date(), [11](#), [13](#), [14](#), [16](#), [20](#), [22](#), [24](#), [26](#),  
[28](#), [31](#), [33](#), [34](#)  
report\_effectsize, [35](#)  
report\_effectsize(), [10](#), [12](#), [14](#), [16](#), [19](#), [22](#),  
[24](#), [25](#), [28](#), [31](#), [32](#), [34](#), [35](#)

- report\_effectsize.aov (report.aov), 11
- report\_effectsize.default (report.default), 21
- report\_effectsize.htest (report.htest), 23
- report\_effectsize.lm (report.lm), 26
- report\_info, 36
- report\_info(), 10, 12, 14, 16, 20, 22, 24, 25, 28, 31, 33, 34, 36
- report\_info.aov (report.aov), 11
- report\_info.default (report.default), 21
- report\_info.htest (report.htest), 23
- report\_info.lm (report.lm), 26
- report\_intercept, 37
- report\_intercept(), 37
- report\_intercept.default (report.default), 21
- report\_intercept.lm (report.lm), 26
- report\_model, 38
- report\_model(), 10, 12, 14, 16, 19, 22, 24, 25, 28, 31, 32, 34
- report\_model.aov (report.aov), 11
- report\_model.default (report.default), 21
- report\_model.htest (report.htest), 23
- report\_model.lm (report.lm), 26
- report\_packages (report.sessionInfo), 29
- report\_packages(), 11, 12, 14, 16, 20, 22, 24, 25, 28, 31, 33, 34
- report\_parameters, 39
- report\_parameters(), 10, 12, 14, 16, 19, 22, 24, 25, 28, 31, 32, 34
- report\_parameters.aov (report.aov), 11
- report\_parameters.compare\_performance (report.compare\_performance), 19
- report\_parameters.default (report.default), 21
- report\_parameters.htest (report.htest), 23
- report\_parameters.lm (report.lm), 26
- report\_parameters.test\_performance (report.test\_performance), 32
- report\_participants, 40
- report\_participants(), 11, 12, 14, 16, 20, 22, 24, 26, 28, 31, 33, 34
- report\_performance, 42
- report\_performance(), 10, 12, 14, 16, 20, 22, 24, 25, 28, 31, 33, 34, 42
- report\_performance.default (report.default), 21
- report\_performance.lavaan (report.lavaan), 25
- report\_performance.lm (report.lm), 26
- report\_priors, 43
- report\_priors(), 10, 12, 14, 16, 20, 22, 24, 25, 28, 31, 33, 34, 43
- report\_priors.default (report.default), 21
- report\_random, 44
- report\_random(), 10, 12, 14, 16, 20, 22, 24, 25, 28, 31, 33, 34, 44
- report\_random.default (report.default), 21
- report\_random.merMod (report.lm), 26
- report\_sample, 45
- report\_sample(), 11, 12, 14, 16, 20, 22, 24, 26, 28, 31, 33, 34
- report\_statistics, 46
- report\_statistics(), 10, 12, 14, 16, 19, 22, 24, 25, 28, 31, 32, 34, 46
- report\_statistics.aov (report.aov), 11
- report\_statistics.compare\_performance (report.compare\_performance), 19
- report\_statistics.default (report.default), 21
- report\_statistics.htest (report.htest), 23
- report\_statistics.lm (report.lm), 26
- report\_statistics.test\_performance (report.test\_performance), 32
- report\_story (report\_date), 34
- report\_system (report.sessionInfo), 29
- report\_system(), 11, 12, 14, 16, 20, 22, 24, 25, 28, 31, 33, 34
- report\_table, 47
- report\_table(), 10, 12, 14, 16, 19, 22, 24, 25, 28, 31, 32, 34, 47
- report\_table.aov (report.aov), 11
- report\_table.compare\_performance (report.compare\_performance), 19
- report\_table.default (report.default), 21
- report\_table.htest (report.htest), 23

report\_table.lm (report.lm), 26  
report\_table.test\_performance  
    (report.test\_performance), 32  
report\_text, 48  
report\_text(), 10, 12, 14, 16, 20, 22, 24, 25,  
    28, 31, 33, 34, 49  
report\_text.aov (report.aov), 11  
report\_text.compare\_performance  
    (report.compare\_performance),  
    19  
report\_text.default (report.default), 21  
report\_text.htest (report.htest), 23  
report\_text.lm (report.lm), 26  
report\_text.test\_performance  
    (report.test\_performance), 32

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

text\_concatenate (format\_text), 8  
text\_fullstop (format\_text), 8  
text\_lastchar (format\_text), 8  
text\_paste (format\_text), 8  
text\_remove (format\_text), 8  
text\_wrap (format\_text), 8