

# Package ‘tidystats’

January 3, 2019

**Title** Create a Tidy Statistics Output File

**Version** 0.3

**Description** Produce a data file containing the output of statistical models and assist with a workflow aimed at writing scientific papers using 'R Markdown'. Supported statistical functions include: `t.test()`, `cor.test()`, `lm()`, `glm()`, `aov()`, `anova()`, and several others. The package is based on tidy data principles and the 'tidyverse' (Wickham, 2017).

**Depends** R (>= 3.5.0)

**Imports** tibble, readr, tidyr, dplyr, magrittr, purrr, stringr, knitr, kableExtra, miniUI, shiny, rlang

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**LazyData** true

**Suggests** testthat, lme4, lmerTest, psych

**NeedsCompilation** no

**Author** Willem Sleegers [aut, cre]

**Maintainer** Willem Sleegers <w.sleegers@me.com>

**Repository** CRAN

**Date/Publication** 2019-01-03 01:10:08 UTC

## R topics documented:

<code>add_stats</code> . . . . .	3
<code>add_stats.data.frame</code> . . . . .	4
<code>add_stats.default</code> . . . . .	5
<code>add_stats.matrix</code> . . . . .	7
<code>add_stats_to_model</code> . . . . .	8
<code>copy_to_clipboard_script</code> . . . . .	9
<code>correlation_table</code> . . . . .	9
<code>count_data</code> . . . . .	10

cox . . . . .	11
css_style . . . . .	12
describe_data . . . . .	12
inspect . . . . .	13
inspect.default . . . . .	14
inspect.list . . . . .	14
inspect_click_script . . . . .	15
read_stats . . . . .	15
rename_columns . . . . .	15
report . . . . .	16
report_anova . . . . .	17
report_chi_squared . . . . .	18
report_correlation . . . . .	18
report_descriptives . . . . .	19
report_fisher . . . . .	20
report_glm . . . . .	20
report_lm . . . . .	21
report_lmm . . . . .	22
report_p_value . . . . .	23
report_rma . . . . .	23
report_statistic . . . . .	24
report_table_lm . . . . .	24
report_t_test . . . . .	25
report_wilcoxon . . . . .	25
stats_list_to_df . . . . .	26
tidy_count_data . . . . .	26
tidy_describe_data . . . . .	27
tidy_stats . . . . .	28
tidy_stats.anova . . . . .	29
tidy_stats.aov . . . . .	30
tidy_stats.aovlist . . . . .	30
tidy_stats.confint . . . . .	31
tidy_stats.glm . . . . .	31
tidy_stats.htest . . . . .	32
tidy_stats.lm . . . . .	33
tidy_stats.lmerMod . . . . .	33
tidy_stats.lmerModLmerTest . . . . .	34
tidy_stats.psych . . . . .	35
tidy_stats.rma . . . . .	36
write_stats . . . . .	36

---

add_stats	<i>Add statistical output to a tidy stats list</i>
-----------	--

---

### Description

add\_stats adds output to a tidystats list. It can take either the output of a statistical test as input or a data frame. See Details for more information on adding data frames.

### Usage

```
add_stats(results, output, identifier = NULL, type = NULL,
          confirmatory = NULL, notes = NULL, class = NULL)
```

### Arguments

results	A tidystats list.
output	Output of a statistical test or a data frame. If a data frame is provided, it must already be in a tidy format.
identifier	A character string identifying the model. Automatically created if not provided.
type	A character string indicating the type of test. One of "hypothesis", "manipulation check", "contrast", "descriptives", or "other". Can be abbreviated.
confirmatory	A boolean to indicate whether the statistical test was confirmatory (TRUE) or exploratory (FALSE). Can be NA.
notes	A character string to add additional information. Some statistical tests produce notes information, which will be overwritten if notes are provided.
class	A character string to indicate which function was used to produce the output. See 'Details' for a list of supported functions.

### Details

Some statistical functions produce unidentifiable output, which means tidystats cannot figure out how to tidy the data. To add these results, you can provide a class via the class argument or you can manually tidy the results yourself and add the resulting data frame via add\_stats().

A list of supported classes are: - confint

### Examples

```
# Create an empty list to store the results in
results <- list()

# Example: t-test
model_t_test <- t.test(extra ~ group, data = sleep)
results <- add_stats(results, model_t_test, identifier = "t_test")

# Example: correlation
x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
```

```

y <- c( 2.6, 3.1, 2.5, 5.0, 3.6, 4.0, 5.2, 2.8, 3.8)

model_correlation <- cor.test(x, y)

# Add output to the results list, only storing the correlation and p-value
results <- add_stats(results, model_correlation, identifier = "correlation")

# Example: Regression
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)

model_lm <- lm(weight ~ group)

# Add output to the results list, with notes
results <- add_stats(results, model_lm, identifier = "regression", notes =
"regression example")

# Example: ANOVA
model_aov <- aov(yield ~ block + N * P * K, npk)

results <- add_stats(results, model_aov, identifier = "ANOVA")

# Example: Within-subjects ANOVA
model_aov_within <- aov(extra ~ group + Error(ID/group), data = sleep)

results <- add_stats(results, model_aov_within, identifier = "ANOVA_within")

# Example: Manual chi-squared test of independence
library(tibble)

x_squared_data <- tibble(
  statistic = c("X-squared", "df", "p"),
  value = c(5.4885, 6, 0.4828),
  method = "Chi-squared test of independence"
)

results <- add_stats(results, x_squared_data, identifier = "x_squared")

```

---

add\_stats.data.frame    *add\_stats data frame function*

---

## Description

add\_stats.data.frame is a function to add a tidy data frame of results to a tidystats list. tidystats does not support all possible statistical tests, so it may not be able to produce tidy output of a statistical model. The best solution for now is to tidy the output of a statistical test yourself, creating a tidy data frame, and then use add\_stats, which will call this function, to add it to the tidystats list.

**Usage**

```
## S3 method for class 'data.frame'
add_stats(results, output, identifier = NULL,
          type = NULL, confirmatory = NULL, notes = NULL, class = NULL)
```

**Arguments**

results	A tidystats list.
output	A data frame that contains statistical output in a tidy format.
identifier	A character string identifying the model. Automatically created if not provided.
type	A character string indicating the type of test. One of "hypothesis", "manipulation check", "contrast", "descriptives", or "other". Can be abbreviated.
confirmatory	A boolean to indicate whether the statistical test was confirmatory (TRUE) or exploratory (FALSE). Can be NA.
notes	A character string to add additional information. Some statistical tests produce notes information, which will be overwritten if notes are provided.
class	Unused.

**Examples**

```
# Create an empty list to store the results in
results <- list()

# Example: Manual chi-squared test of independence
x_squared_data <- data.frame(
  statistic = c("X-squared", "df", "p"),
  value = c(5.4885, 6, 0.4828),
  method = "Chi-squared test of independence"
)

# Add results to results
results <- add_stats(results, x_squared_data)
```

---

add\_stats.default      *add\_stats default function*

---

**Description**

add\_stats.default is the default add\_stats function, which takes the output of a statistical test, tidies the output, and adds it to a tidystats list.

**Usage**

```
## Default S3 method:
add_stats(results, output, identifier = NULL,
          type = NULL, confirmatory = NULL, notes = NULL, class = NULL)
```

**Arguments**

results	A tidy stats list.
output	Output of a statistical test.
identifier	A character string identifying the model. Automatically created if not provided.
type	A character string indicating the type of test. One of "hypothesis", "manipulation check", "contrast", "descriptives", or "other". Can be abbreviated.
confirmatory	A boolean to indicate whether the statistical test was confirmatory (TRUE) or exploratory (FALSE). Can be NA.
notes	A character string to add additional information. Some statistical tests produce notes information, which will be overwritten if notes are provided.
class	A character string to indicate which function was used to produce the output. See 'Details' for a list of supported functions.

**Examples**

```
# Create an empty list to store the results in
results <- list()

# Example: t-test
model_t_test <- t.test(extra ~ group, data = sleep)
results <- add_stats(results, model_t_test, identifier = "t_test")

# Example: correlation
x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
y <- c( 2.6,  3.1,  2.5,  5.0,  3.6,  4.0,  5.2,  2.8,  3.8)

model_correlation <- cor.test(x, y)

# Add output to the results list, only storing the correlation and p-value
results <- add_stats(results, model_correlation, identifier = "correlation")

# Example: Regression
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl", "Trt"))
weight <- c(ctl, trt)

model_lm <- lm(weight ~ group)

# Add output to the results list, with notes
results <- add_stats(results, model_lm, identifier = "regression", notes =
  "regression example")

# Example: ANOVA
model_aov <- aov(yield ~ block + N * P * K, npk)

results <- add_stats(results, model_aov, identifier = "ANOVA")

# Example: Within-Subjects ANOVA
```

```

model_aov_within <- aov(extra ~ group + Error(ID/group), data = sleep)

results <- add_stats(results, model_aov_within, identifier = "ANOVA_within")

```

---

add\_stats.matrix      *add\_stats matrix function*

---

## Description

add\_stats.matrix is a function to add a matrix to a tidystats list. Some statistical functions (e.g., confint) return a matrix. Normally this would not be enough information for tidystats to figure out how to tidy the results. However, if you run add\_stats and provide a class, tidystats may be able to tidy the output. See 'Details' for a list of supported classes.

## Usage

```

## S3 method for class 'matrix'
add_stats(results, output, identifier = NULL,
          type = NULL, confirmatory = NULL, notes = NULL, class = NULL)

```

## Arguments

results	A tidystats list.
output	A matrix that contains statistical output.
identifier	A character string identifying the model. Automatically created if not provided.
type	A character string indicating the type of test. One of "hypothesis", "manipulation check", "contrast", "descriptives", or "other". Can be abbreviated.
confirmatory	A boolean to indicate whether the statistical test was confirmatory (TRUE) or exploratory (FALSE). Can be NA.
notes	A character string to add additional information. Some statistical tests produce notes information, which will be overwritten if notes are provided.
class	A character string to indicate which function was used to produce the output. See 'Details' for a list of supported functions.

## Details

Supported classes include: confint

## Examples

```

# Create an empty list to store the results in
results <- list()

# Example: Confidence intervals on a regression model

```

```

ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)

model_lm <- lm(weight ~ group)
model_lm_confint <- confint(model_lm)

# Add output to the results list
results <- add_stats(results, model_lm_confint, identifier = "lm_confint",
  class = "confint")

```

---

add\_stats\_to\_model      *Add statistical output to a model in a tidy stats list*

---

## Description

add\_stats\_to\_model adds output to a model in a tidy results list. Sometimes you have to run additional analyses on the output of a statistical test, so you want to add these results to an existing model in a tidy stats list.

## Usage

```
add_stats_to_model(results, output, identifier, class = NULL)
```

## Arguments

results	A tidystats list.
output	Output of a statistical test.
identifier	A character string identifying the model.
class	A character string to indicate which function was used to produce the output. See 'Details' for a list of supported functions.

## Details

Supported classes are: - confint

## Examples

```

# Create an empty list to store the results in
results <- list()

# Conduct a regression analysis
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))

```



```
weight <- c(ct1, trt)

model <- lm(weight ~ group)

# Add output to the results list
results <- add_stats(results, model, identifier = "M1")

# Get confidence intervals of the model
model_CIs <- confint(model)

# Add it to the results list
results <- add_stats_to_model(results, model_CIs, identifier = "M1",
  class = "confint")
```

---

copy\_to\_clipboard\_script

*Load in Javascript code to copy content to the clipboard.*

---

### Description

A helper function to load in Javascript code to copy content to the clipboard, used in the inspect() function.

---

correlation\_table      *Create a correlation table*

---

### Description

Function to create a correlation table.

### Usage

```
correlation_table(correlations, triangle = "lower", p_values = NULL,
  diagonal = NULL, labels = NULL, digits = 2)
```

### Arguments

correlations	A data frame or matrix containing the correlations.
triangle	A character string indicating whether the lower, upper, or both halves should be visible, must be one of 'lower', 'upper', or 'both'. You can specify just the initial letter.
p_values	A data frame or matrix containing the p-values.
diagonal	A vector of values to be placed on the diagonal. Default is a vector of 1s.
labels	A vector of strings containing the labels for each variable.
digits	An integer indicating the number of decimal places.

## Examples

```
# Add examples
```

---

count_data	<i>Count the total of observations</i>
------------	--

---

## Description

count\_data returns the number of observations for categorical variables.

## Usage

```
count_data(data, ..., na.rm = TRUE)
```

## Arguments

data	A data frame.
...	One or more unquoted (categorical) column names from the data frame, separated by commas.
na.rm	Logical. Should missing values (including NaN) be removed?

## Details

The data frame can be grouped using **dplyr**'s `group_by` so that the total of observations will be calculated for each group level.

## Examples

```
library(dplyr)

# 1 variable
count_data(cox, sex)

# 2 variables
count_data(cox, condition, sex)

# 1 variable, 1 group
cox %>%
  group_by(condition) %>%
  count_data(sex)
```

---

COX *Data of a replication study of C.R. Cox, J. Arndt, T. Pyszczynski, J. Greenberg, A. Abdollahi, S. Solomon (2008, JPSP, 94(4), Exp. 6)*

---

### Description

This is the data of a replication study performed by J. Wissink, G. Hoogendoorn, H. Brohmer, M. Verschoor, J. Krijnen, and M. Zeelenberg as part of the Reproducibility Project: Psychology. The target of replication was the finding in Experiment 6 of Cox et al. (2008) that participants who scored low on avoidance but high on anxiety demonstrated an increased relative preference for a parent after mortality salience as opposed to dental pain,  $b = -32.04$ ,  $SE = 14.47$ ,  $t = -2.22$ ,  $p = .03$ .

### Usage

COX

### Format

A data frame with 200 rows and 67 variables.

**ID** Participant identifier

**sex** The participant's sex

**age** The participant's age

**condition** The experimental condition: mortality salience or dental pain

**avoidance** Attachment avoidance score as assessed with the Relationship Scales Questionnaire

**anxiety** Attachment anxiety score as assessed with the Relationship Scales Questionnaire

**affect\_positive** Sum of positive PANAS items

**affect\_negative** Sum of negative PANAS items

**call\_parent** Minutes allocated (out of 100) to call a parent

**call\_siblings** Minutes allocated (out of 100) to call a sibling

**call\_partner** Minutes allocated (out of 100) to call a romantic partner

**call\_friend** Minutes allocated (out of 100) to call a close friend

### Details

For more information on the Reproducibility Project: Psychology, see <https://osf.io/ezcu/>. The individual scores on the PANAS and RSQ items are not included in this dataset.

### Source

<https://osf.io/5tbxf/>

## References

Cox, C.R., Arndt, J., Pyszczynski, T., Greenberg, J., Abdollahi, A., & Solomon, S. (2008) Terror management and adults' attachment to their parents: The safe haven remains. *Journal of Personality and Social Psychology*, 94(4), 696-717, <https://dx.doi.org/10.1037/0022-3514.94.4.696>

---

css_style	<i>Load in CSS code to style HTML content</i>
-----------	---

---

## Description

A helper function to load in CSS code to style HTML content, used in the `inspect()` function.

---

describe_data	<i>Calculate common descriptive statistics</i>
---------------	--

---

## Description

`describe_data` returns a set of common descriptive statistics (e.g., n, mean, sd) for numeric variables.

## Usage

```
describe_data(data, ..., na.rm = TRUE)
```

## Arguments

<code>data</code>	A data frame.
<code>...</code>	One or more unquoted (numerical) column names from the data frame, separated by commas.
<code>na.rm</code>	Logical. Should missing values (including NaN) be removed?

## Details

The data set can be grouped using **dplyr**'s `group_by` so that descriptives will be calculated for each group level.

Skew and kurtosis are based on the `skewness` and `kurtosis` functions of the **moments** package (Komsta & Novomestky, 2015).

**Examples**

```

library(dplyr)

# 1 variable
describe_data(cox, avoidance)

# 1 variable, 1 group
cox %>%
  group_by(condition) %>%
  describe_data(avoidance)

# 2 variables
describe_data(cox, avoidance, anxiety)

# 2 variables, 1 group
cox %>%
  group_by(condition) %>%
  describe_data(avoidance, anxiety)

# 1 variable, 2 groups
cox %>%
  group_by(condition, sex) %>%
  describe_data(avoidance)

# 2 variables, 2 groups
cox %>%
  group_by(condition, sex) %>%
  describe_data(avoidance, anxiety)

```

---

inspect	<i>Inspect the output of (a) statistical model(s) via an interactive Shiny app.</i>
---------	---

---

**Description**

inspect is a function to inspect the output of one or more statistical models. The function will open a Shiny widget in the Viewer pane. This allows the user to visually inspect the model output, as well as copy the results in APA format.

**Usage**

```
inspect(results, ...)
```

**Arguments**

results	A tidystats list or the output of a statistical test.
...	A variable number of identifiers in order to limit the number of models shown when a tidystats list is provided.

---

inspect.default	<i>Inspect (a) statistical model(s) output</i>
-----------------	--

---

### Description

inspect is a function to inspect the output of a statistical model via the Viewer pane.

### Usage

```
## Default S3 method:  
inspect(results, ...)
```

### Arguments

results	The output of a statistical test.
...	Unused

### Examples

```
# Run a statistical test  
model <- t.test(extra ~ group, data = sleep)  
  
# Inspect the output  
# inspect(model)
```

---

inspect.list	<i>Inspect (a) statistical model(s) added to your tidystats list</i>
--------------	--

---

### Description

inspect is a function to inspect one or more models that are part of a tidystats list. The function will open a Shiny widget in the Viewer pane, which will show the results of one or more models. This allows the user to visually inspect the model output, as well as copy the results in APA format.

### Usage

```
## S3 method for class 'list'  
inspect(results, ...)
```

### Arguments

results	A tidystats list.
...	Identifiers of models to show in the viewer.

---

inspect\_click\_script    *Load in Javascript code to figure out clicks in the inspect() function*

---

**Description**

A helper function to load in Javascript code to figure out what was clicked on, used in the inspect() function.

---

read\_stats                    *Read a .csv file that was produced with write\_stats*

---

**Description**

read\_stats can convert a .csv file containing the statistical results that was produced with write\_stats. It returns a list containing the results, with the identifier as the name for each list element.

**Usage**

```
read_stats(file)
```

**Arguments**

file                    Path to the tidy stats data file

**Examples**

```
results <- read_stats(system.file("results.csv", package = "tidystats"))
```

---

rename\_columns            *Rename statistics columns*

---

**Description**

Renames the statistics columns.

**Usage**

```
rename_columns(x)
```

**Arguments**

x                    The output of a statistical model, converted to a data frame.

**Details**

This function matches each column name with an entry in a list of statistical names and renames the column if necessary. The goal is to create a naming scheme that makes statistical labels as consistent as possible.

---

report	<i>Report function</i>
--------	------------------------

---

**Description**

report is a general function that returns Markdown code of a statistical test in 6th edition APA style.

**Usage**

```
report(identifier, term = NULL, term_nr = NULL, var = NULL,
       group = NULL, statistic = NULL,
       results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the model.
term	A character string indicating which term you want to report the statistics of.
term_nr	A number indicating which term you want to report the the statistics of.
var	A character string identifying the variable.
group	A character string identifying the group.
statistic	A character string of a statistic you want to extract from a model.
results	A tidy stats list.

**Details**

report calls a specific report function dependent on the type of statistical test that is supplied. The 'method' column of the statistical test is used to determine which report function to run.

**Examples**

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the list as the default list
options(tidystats_list = results)

# Example: t-test
report("t_test_one_sample")
report("t_test_welch")
```



```

# Example: correlation
report("correlation_pearson")
report("correlation_spearman")

# Example: ANOVA
report("aov_two_way", term = "condition")
report("aov_two_way", term = "sex")

# Example: Linear models
report("lm_simple", term = "conditionmortality salience")
report("lm_simple", term_nr = 2)
report("lm_simple", group = "model")

```

---

report_anova	<i>Report method for ANOVA models</i>
--------------	---------------------------------------

---

## Description

Function to report ANOVA output in APA style.

## Usage

```
report_anova(identifier, group = NULL, term = NULL, term_nr = NULL,
  results = getOption("tidystats_list"))
```

## Arguments

identifier	A character string identifying the model.
group	A character string indicating the group containing the statistics you want to report.
term	A character string indicating the term you want to report.
term_nr	A number indicating the term you want to report.
results	A tidystats list.

## Examples

```

# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the list as the default list
options(tidystats_list = results)

# Report results
report_anova("aov_two_way", term = "condition")
report_anova("aov_two_way", term = "sex")

```

---

report\_chi\_squared      *Report function for a chi-squared test*

---

### Description

Function to report a chi-squared test in APA style.

### Usage

```
report_chi_squared(identifier, results = getOption("tidystats_list"))
```

### Arguments

identifier      A character string identifying the model.  
 results          A tidystats list.

### Examples

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default tidystats list in options()
options(tidystats_list = results)

# Report results
report(identifier = "chi_squared")
report(identifier = "chi_squared_yates")
```

---

report\_correlation      *Report function for correlations*

---

### Description

Function to report a correlation in APA style.

### Usage

```
report_correlation(identifier, term, term_nr,
  results = getOption("tidystats_list"))
```

### Arguments

identifier      A character string identifying the model.  
 term            A character string indicating the term you want to report.  
 term\_nr        A number indicating the term you want to report.]  
 results        A tidystats list.

**Examples**

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default tidystats list in options()
options(tidystats_list = results)

# Report results
report(identifier = "correlation_pearson")
report(identifier = "correlation_spearman")
report(identifier = "correlation_kendall")
```

---

report\_descriptives    *Report descriptives helper functions*

---

**Description**

Report descriptives helper functions

**Usage**

```
M(identifier, var = NULL, group = NULL,
  results = getOption("tidystats_list"))

SD(identifier, var = NULL, group = NULL,
  results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the descriptives.
var	A character string identifying the exact variable, if needed.
group	A character string identifying the group, if needed.
results	A tidystats list.

**Examples**

```
# Read in a list of results
descriptives <- read_stats(system.file("descriptives.csv",
  package = "tidystats"))

options(tidystats_list = descriptives)

# Report the mean
M("D4_avoidance")
M("D5_avoidance_anxiety", var = "avoidance")

# Report the standard deviation
```

```
SD("D4_avoidance")
```

---

report_fisher	<i>Report function for Fisher's Exact Tests for Count Data</i>
---------------	--

---

### Description

Function to report Fisher's Exact Tests for Count Data in APA style.

### Usage

```
report_fisher(identifier, results = getOption("tidystats_list"))
```

### Arguments

identifier	A character string identifying the model.
results	A tidystats list.

### Examples

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default tidystats list in options()
options(tidystats_list = results)

# Report results
report(identifier = "fisher_test")
report(identifier = "fisher_test_hybrid")
report(identifier = "fisher_test_simulated_p")
```

---

report_glm	<i>Report method for generalized linear models</i>
------------	--

---

### Description

Function to report a generalized linear models in APA style.

### Usage

```
report_glm(identifier, group = NULL, term = NULL, term_nr = NULL,
  results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the model.
group	A character string indicating the group containing the statistics you want to report.
term	A character string indicating the term you want to report.
term_nr	A number indicating the term you want to report.
results	A tidystats list.

**Examples**

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default results list
options(tidystats_list = results)

# Report results
report("glm_gaussian", term = "Prewt")
report("glm_gamma", term = "log(u)")
report("glm_poisson", term_nr = 2)
```

---

report\_lm

*Report method for linear regression models*


---

**Description**

Function to report a regression in APA style.

**Usage**

```
report_lm(identifier, group = NULL, term = NULL, term_nr = NULL,
  results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the model.
group	A character string indicating the group containing the statistics you want to report.
term	A character string indicating the term you want to report.
term_nr	A number indicating the term you want to report.
results	A tidystats list.

## Examples

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default results list
options(tidystats_list = results)

# Report results
report("lm_simple", term = "conditionmortality salience")
report("lm_simple", term_nr = 2)
report("lm_simple", group = "model")
```

---

report_lmm	<i>Report method for linear mixed models</i>
------------	--

---

## Description

Function to report a linear mixed model in APA style.

## Usage

```
report_lmm(identifier, group = NULL, term = NULL, term_nr = NULL,
  results = getOption("tidystats_list"))
```

## Arguments

identifier	A character string identifying the model.
group	A character string indicating the group containing the statistics you want to report.
term	A character string indicating the term you want to report.
term_nr	A number indicating the term you want to report.
results	A tidystats list.

## Examples

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default results list
options(tidystats_list = results)

# Example: regression term
report("lme4_lme", term = "Days")
report("lmerTest_lme", term = "Days")
```

---

report_p_value	<i>Report p-value function</i>
----------------	--------------------------------

---

**Description**

A helper function to report p values in APA style.

**Usage**

```
report_p_value(p_value)
```

**Arguments**

p_value	One or more p-values.
---------	-----------------------

**Examples**

```
report_p_value(0.532)
report_p_value(0.0432)
report_p_value(0.0000121)
```

---

report_rma	<i>Report method for metafor's rma models</i>
------------	---

---

**Description**

Function to report a meta-analysis in APA style.

**Usage**

```
report_rma(identifier, group = NULL, term = NULL, term_nr = NULL,
  results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the model.
group	A character string identifying the group you want to report the statistics of.
term	A character string indicating which term you want to report the statistics of.
term_nr	A number indicating which term you want to report the the statistics of.
results	A tidystats list.

---

report_statistic	<i>Report a single statistic</i>
------------------	----------------------------------

---

**Description**

A helper function to report single statistics in APA style.

**Usage**

```
report_statistic(statistic, value)
```

**Arguments**

statistic	A character string indicating the kind of statistic (e.g., p, t, df).
value	The value of the statistic to be reported.

**Examples**

```
report_statistic("p", 0.0001)
report_statistic("t", 4.2325)
report_statistic("r", 0.213)
```

---

report_table_lm	<i>Report table method for linear regression models</i>
-----------------	---

---

**Description**

Function create a table in order to display results of a regression.

**Usage**

```
report_table_lm(identifier, terms = NULL, term_nrs = NULL,
  statistics = NULL, include_model = TRUE, term_labels = NULL,
  results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the model.
terms	A character vector indicating which terms you want to report in the table.
term_nrs	A numeric vector indicating which terms you want to report in the table.
statistics	A character vector of statistics you want to display in the table.
include_model	Include or exclude model statistics (e.g., R squared).
term_labels	A character vector to change the labels for the terms.
results	A tidystats list.



---

report_t_test	<i>Report function for t-tests</i>
---------------	------------------------------------

---

**Description**

Function to report a t-test in APA style.

**Usage**

```
report_t_test(identifier, results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the model.
results	A tidystats list.

**Examples**

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default tidystats list in options()
options(tidystats_list = results)

# Report results
report(identifier = "t_test_one_sample")
report(identifier = "t_test_two_sample")
report(identifier = "t_test_welch")
report(identifier = "t_test_paired")
```

---

report_wilcoxon	<i>Report function for Wilcoxon Rank Sum and Signed Rank Tests</i>
-----------------	--

---

**Description**

Function to report Wilcoxon Rank Sum and Signed Rank Tests in APA style.

**Usage**

```
report_wilcoxon(identifier, results = getOption("tidystats_list"))
```

**Arguments**

identifier	A character string identifying the model.
results	A tidystats list.

**Examples**

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Set the default tidystats list in options()
options(tidystats_list = results)

# Report results
report(identifier = "wilcoxon_rank_sum")
report(identifier = "wilcoxon_signed_rank")
```

---

stats_list_to_df	<i>Convert a tidy stats list to a data frame</i>
------------------	--

---

**Description**

stats\_list\_to\_df converts a tidy stats list to a data frame.

**Usage**

```
stats_list_to_df(results)
```

**Arguments**

results            A tidy stats list.

**Examples**

```
# Read in a list of results
results <- read_stats(system.file("results.csv", package = "tidystats"))

# Convert list to a data frame
stats_list_to_df(results)
```

---

tidy_count_data	<i>Convert count data to a tidy data frame</i>
-----------------	--

---

**Description**

tidy\_count\_data returns a tidy data frame of count statistics created with **tidystats**' count\_data.

**Usage**

```
tidy_count_data(count_data)
```

**Arguments**

count\_data      a data frame created with tidystats' count\_data.

**Examples**

```
library(dplyr)

# Calculate counts
condition_n <- count_data(cox, condition)

# Create a tidy data frame of the count data
tidy_count_data(condition_n)

# With a grouping variable:
cox %>%
  group_by(sex) %>%
  count_data(condition) %>%
  tidy_count_data()
```

---

tidy\_describe\_data      *Convert descriptives to a tidy data frame*

---

**Description**

tidy\_describe\_data returns a tidy data frame of descriptive statistics created with **tidystats'** describe\_data.

**Usage**

```
tidy_describe_data(descriptives)
```

**Arguments**

descriptives      A data frame created with tidystats' describe\_data.

**Examples**

```
library(dplyr)

# Calculate descriptives
descriptives <- describe_data(sleep, extra)

# Create a tidy data frame of the descriptives
tidy_describe_data(descriptives)

# With a grouping variable:
sleep %>%
  group_by(group) %>%
```

```
describe_data(extra) %>%
tidy_describe_data()
```

---

tidy\_stats

*Create a tidy stats data frame from a statistical output object*


---

### Description

tidy\_stats is a generic function that calls one of the specific tidy\_stats functions dependent on the type of model (e.g., htest, lm, aov).

### Usage

```
tidy_stats(model)
```

### Arguments

model                    Output of a statistical test

### Examples

```
# Example: t-test
model_t_test <- t.test(extra ~ group, data = sleep)
tidy_stats(model_t_test)

# Example: correlation
x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
y <- c( 2.6,  3.1,  2.5,  5.0,  3.6,  4.0,  5.2,  2.8,  3.8)

model_correlation <- cor.test(x, y)
tidy_stats(model_correlation)

# Example: Regression
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)

model_lm <- lm(weight ~ group)
tidy_stats(model_lm)

# Example: ANOVA
model_aov <- aov(yield ~ block + N * P * K, npk)
tidy_stats(model_aov)

# Example: Within-subjects ANOVA
model_aov_within <- aov(extra ~ group + Error(ID/group), data = sleep)
tidy_stats(model_aov_within)
```

---

tidy_stats.anova	<i>Create a tidy stats data frame from an anova object</i>
------------------	--

---

## Description

tidy\_stats.anova takes an anova object and tidies the output into a data frame.

## Usage

```
## S3 method for class 'anova'  
tidy_stats(model)
```

## Arguments

model            Output of anova().

## Examples

```
# Regression example  
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)  
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)  
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))  
weight <- c(ctl, trt)  
  
model_0 <- lm(weight ~ 1)  
model_1 <- lm(weight ~ group)  
  
tidy_stats(anova(model_0))  
tidy_stats(anova(model_1))  
tidy_stats(anova(model_0, model_1))  
tidy_stats(anova(model_0, model_1, test = "Chisq"))  
  
# Logistic regression example  
counts <- c(18,17,15,20,10,20,25,13,12)  
outcome <- gl(3,1,9)  
treatment <- gl(3,3)  
  
glm.D93 <- glm(counts ~ outcome + treatment, family = poisson())  
  
tidy_stats(anova(glm.D93))
```

---

tidy_stats.aov	<i>Create a tidy stats data frame from an aov object</i>
----------------	--

---

**Description**

tidy\_stats.aov takes an aov object and converts the object to a tidy stats data frame.

**Usage**

```
## S3 method for class 'aov'  
tidy_stats(model)
```

**Arguments**

model                    Output of aov without within-subject factors.

**Examples**

```
# Conduct an ANOVA  
model_aov <- aov(yield ~ block + N * P * K, npk)  
  
# Tidy stats  
tidy_stats(model_aov)
```

---

tidy_stats.aovlist	<i>Create a tidy stats data frame from an aovlist object</i>
--------------------	--

---

**Description**

tidy\_stats.aovlist takes an aovlist object and converts the object to a tidy stats data frame.

**Usage**

```
## S3 method for class 'aovlist'  
tidy_stats(model)
```

**Arguments**

model                    Output of aov() including within-subject factors.

**Examples**

```
# Conduct a within-subjects ANOVA  
model_aov_within <- aov(extra ~ group + Error(ID/group), data = sleep)  
  
# Tidy stats  
tidy_stats(model_aov_within)
```

---

tidy\_stats.confint      *tidy\_stats method for confint output.*

---

### Description

Creates a tidystats data frame from the output of confint.

### Usage

```
## S3 method for class 'confint'  
tidy_stats(model)
```

### Arguments

model                      The output of confint().

### Details

This method should not be called directly.

---

tidy\_stats.glm              *Create a tidy stats data frame from a glm object*

---

### Description

tidy\_stats.glm takes a glm object and converts the object to a tidy stats data frame.

### Usage

```
## S3 method for class 'glm'  
tidy_stats(model)
```

### Arguments

model                      Output of glm().

### Examples

```
# Get data  
counts <- c(18,17,15,20,10,20,25,13,12)  
outcome <- gl(3,1,9)  
treatment <- gl(3,3)  
d.AD <- data.frame(treatment, outcome, counts)  
  
# Run model  
glm.D93 <- glm(counts ~ outcome + treatment, family = poisson())
```

```
# Tidy stats  
tidy_stats(glm.D93)
```

---

tidy_stats.htest	<i>Create a tidy stats data frame from an htest object</i>
------------------	--

---

## Description

tidy\_stats.htest takes an htest object and converts the object to a tidy stats data frame.

## Usage

```
## S3 method for class 'htest'  
tidy_stats(model)
```

## Arguments

model            Output of t.test().

## Examples

```
# Conduct a t-test  
model_t_test <- t.test(extra ~ group, data = sleep)  
tidy_stats(model_t_test)  
  
# Conduct a correlation  
x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)  
y <- c( 2.6,  3.1,  2.5,  5.0,  3.6,  4.0,  5.2,  2.8,  3.8)  
  
model_correlation <- cor.test(x, y)  
tidy_stats(model_correlation)  
  
# Conduct a chi-square test  
M <- as.table(rbind(c(762, 327, 468), c(484, 239, 477)))  
  
model_chi_square <- chisq.test(M)  
tidy_stats(model_chi_square)
```



---

tidy_stats.lm	<i>Create a tidy stats data frame from an lm object</i>
---------------	---

---

**Description**

tidy\_stats.lm takes an lm object and converts the object to a tidy stats data frame.

**Usage**

```
## S3 method for class 'lm'  
tidy_stats(model)
```

**Arguments**

model            Output of lm().

**Examples**

```
# Conduct a regression  
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)  
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)  
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))  
weight <- c(ctl, trt)  
  
model_lm <- lm(weight ~ group)  
tidy_stats(model_lm)
```

---

tidy_stats.lmerMod	<i>Create a tidy stats data frame from an lmerMod object</i>
--------------------	--

---

**Description**

tidy\_stats.lmerMod takes an lmerMod object and converts the object to a tidy stats data frame.

**Usage**

```
## S3 method for class 'lmerMod'  
tidy_stats(model)
```

**Arguments**

model            Output of lme4's lmer().

## Examples

```
# Load package
library(lme4)

# Conduct a linear mixed model
model_lmer <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

# Tidy stats
tidy_stats(model_lmer)
```

---

tidy\_stats.lmerModLmerTest

*Create a tidy stats data frame from an lmerModLmerTest object*

---

## Description

tidy\_stats.lmerModLmerTest takes an lmerModLmerTest object and converts the object to a tidy stats data frame.

## Usage

```
## S3 method for class 'lmerModLmerTest'
tidy_stats(model)
```

## Arguments

model                      Output of lmerTest's lmer().

## Examples

```
# Load packages
library(lme4)
library(lmerTest)

# Conduct a linear mixed model
model_lmerTest <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

# Tidy stats
tidy_stats(model_lmerTest)
```

---

tidy_stats.psych	<i>tidy_stats method for psych's alpha objects</i>
------------------	--

---

## Description

Creates a tidystats data frame for a psych's alpha object.

## Usage

```
## S3 method for class 'psych'  
tidy_stats(model)
```

## Arguments

model            An psych alpha object

## Examples

```
# Load packages  
library(psych)  
library(dplyr)  
  
# Create an empty list to store results in  
results <- list()  
  
# Example: Cronbach's alpha  
alpha_agreeableness <- bfi %>%  
  select(A1, A2, A3, A4, A5) %>%  
  alpha(check.keys = TRUE, warnings = FALSE)  
  
# Tidy stats  
tidy_stats(alpha_agreeableness)  
  
# Example: Correlations  
cors_agreeableness <- bfi %>%  
  select(A1, A2, A3, A4, A5) %>%  
  corr.test()  
  
# Tidy stats  
tidy_stats(cors_agreeableness)
```

---

tidy_stats.rma	<i>Create a tidy stats data frame from an rma object from the metafor package</i>
----------------	---

---

### Description

tidy\_stats.rma takes an rma object and converts the object to a tidy stats data frame.

### Usage

```
## S3 method for class 'rma'  
tidy_stats(model)
```

### Arguments

model            Output of metafor::rma().

---

write_stats	<i>Save the results in a tidy stats list to a .csv file</i>
-------------	---

---

### Description

write\_stats converts a tidy stats results list to a data frame and then saves the data to a .csv file.

### Usage

```
write_stats(results, path)
```

### Arguments

results            A tidy stats list.  
path                Path or connection to write to.

### Details

The prettyNum function is used to format the numbers before saving them to disk. This is to prevent saving numbers with many decimals.

**Examples**

```
library(dplyr)

# Create an empty list to store the results in
results <- list()

# Conduct statistical tests
model1 <- t.test(anxiety ~ condition, data = cox)
model2 <- cor.test(cox$anxiety, cox$avoidance)

# Add output to the results list
results <- results %>%
  add_stats(model1, identifier = "M1") %>%
  add_stats(model2, identifier = "M2")

# Save the results
dir <- tempdir()
write_stats(results, file.path(dir, "results.txt"))
```

# Index

## \*Topic **datasets**

- cox, 11
- add\_stats, 3
- add\_stats.data.frame, 4
- add\_stats.default, 5
- add\_stats.matrix, 7
- add\_stats\_to\_model, 8
- copy\_to\_clipboard\_script, 9
- correlation\_table, 9
- count\_data, 10
- cox, 11
- css\_style, 12
- describe\_data, 12
- inspect, 13
- inspect.default, 14
- inspect.list, 14
- inspect\_click\_script, 15
- M(report\_descriptives), 19
- read\_stats, 15
- rename\_columns, 15
- report, 16
- report\_anova, 17
- report\_chi\_squared, 18
- report\_correlation, 18
- report\_descriptives, 19
- report\_fisher, 20
- report\_glm, 20
- report\_lm, 21
- report\_lmm, 22
- report\_p\_value, 23
- report\_rma, 23
- report\_statistic, 24
- report\_t\_test, 25
- report\_table\_lm, 24
- report\_wilcoxon, 25
- SD(report\_descriptives), 19
- stats\_list\_to\_df, 26
- tidy\_count\_data, 26
- tidy\_describe\_data, 27
- tidy\_stats, 28
- tidy\_stats.anova, 29
- tidy\_stats.aov, 30
- tidy\_stats.aovlist, 30
- tidy\_stats.confint, 31
- tidy\_stats.glm, 31
- tidy\_stats.htest, 32
- tidy\_stats.lm, 33
- tidy\_stats.lmerMod, 33
- tidy\_stats.lmerModLmerTest, 34
- tidy\_stats.psych, 35
- tidy\_stats.rma, 36
- write\_stats, 36