

# Package: tidySummaries (via r-universe)

May 20, 2026

**Type** Package

**Title** Tidy Statistical Summaries for Exploratory Data Analysis

**Version** 0.1.0

**Description** Provides a tidy set of functions for summarising data, including descriptive statistics, frequency tables with normality testing, and group-wise significance testing. Designed for fast, readable, and easy exploration of both numeric and categorical data.

**Maintainer** Kleanthis Koupidis <kleanthis.koupidis@gmail.com>

**URL** <https://github.com/kleanthisk10/tidySummaries>

**BugReports** <https://github.com/kleanthisk10/tidySummaries/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 4.1.0)

**Imports** magrittr, tidyr, dplyr, tibble, purrr, stats, crayon, rlang

**RoxygenNote** 7.3.2

**Suggests** ggplot2, testthat, knitr, rmarkdown

**Config/pak/sysreqs** libicu-dev

**Repository** <https://kleanthisk10.r-universe.dev>

**Date/Publication** 2025-05-25 12:04:34 UTC

**RemoteUrl** <https://github.com/kleanthisk10/tidysummaries>

**RemoteRef** HEAD

**RemoteSha** f0580b23bb9626ff91bb1544962682cdd484e1c0

## Contents

select_non_numeric_cols . . . . .	2
select_numeric_cols . . . . .	3
str_replace_many . . . . .	3
summarise_boxplot_stats . . . . .	4
summarise_coef_of_variation . . . . .	5
summarise_correlation . . . . .	5
summarise_frequency . . . . .	6
summarise_group_stats . . . . .	7
summarise_kurtosis . . . . .	8
summarise_skewness . . . . .	9
summarise_statistics . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

select\_non\_numeric\_cols  
*Select Non-Numeric Columns*

---

### Description

Returns a tibble with only the non-numeric columns of the input, and optionally drops rows with NAs.

### Usage

```
select_non_numeric_cols(dataset, remove_na = FALSE)
```

### Arguments

dataset            A vector, matrix, data frame, or tibble.  
remove\_na        Logical. If TRUE, rows with any NA values will be dropped. Default is FALSE.

### Value

A tibble with only non-numeric columns.

### Examples

```
select_non_numeric_cols(iris)

df <- tibble::tibble(a = 1:6, b = c("x", "y", NA, NA, "z", NA))
select_non_numeric_cols(df, remove_na = TRUE)
```

---

select\_numeric\_cols     *Select Numeric Columns*

---

**Description**

Returns a tibble with only the numeric columns of the input, and optionally drops rows with NAs.

**Usage**

```
select_numeric_cols(dataset, remove_na = FALSE)
```

**Arguments**

dataset             A vector, matrix, data frame, or tibble.  
remove\_na           Logical. If TRUE, rows with any NA values will be dropped. Default is FALSE.

**Value**

A tibble with only numeric columns.

**Examples**

```
select_numeric_cols(iris)
```

---

str\_replace\_many     *Multiple Pattern-Replacement Substitutions*

---

**Description**

Applies multiple regular expression substitutions to a character vector or a specific column of a data frame. Performs replacements sequentially

**Usage**

```
str_replace_many(x, pattern, replacement, column = NULL, ...)
```

**Arguments**

x                    A character vector or a data frame containing the text to modify.  
pattern              A character vector of regular expressions to match.  
replacement         A character vector of replacement strings, same length as 'pattern'.  
column               Optional. If 'x' is a data frame, the name of the character column to apply the replacements to.  
...                   Additional arguments passed to 'gsub()', such as 'ignore.case = TRUE'.

**Value**

- If 'x' is a character vector, returns a modified character vector. - If 'x' is a data frame, returns the data frame with the specified column modified.

**Examples**

```
# Example on a character vector
text <- c("The cat and the dog", "dog runs fast", "no animals")
str_replace_many(text, pattern = c("cat", "dog"), replacement = c("lion", "wolf"))

# Example on a data frame
library(tibble)
df <- tibble(id = 1:3, text = c("The cat sleeps", "dog runs fast", "no pets"))
str_replace_many(df, pattern = c("cat", "dog"), replacement = c("lion", "wolf"), column = "text")
```

---

summarise\_boxplot\_stats

*Summarise Boxplot Statistics with Outliers*

---

**Description**

Computes the five-number summary (min, Q1, median, Q3, max), interquartile range (IQR), range, and outliers for each numeric variable in a data frame or a numeric vector.

**Usage**

```
summarise_boxplot_stats(x)
```

**Arguments**

x                    A numeric vector, matrix, data frame, or tibble.

**Value**

A tibble with columns: 'variable', 'min', 'q1', 'median', 'q3', 'max', 'iqr', 'range', 'n\_outliers', 'outliers'.

**Examples**

```
summarise_boxplot_stats(iris)
summarise_boxplot_stats(iris$Sepal.Width)
summarise_boxplot_stats(data.frame(a = c(rnorm(98), 10, NA)))
```

---

summarise\_coef\_of\_variation  
*Summarise Coefficient of Variation*

---

**Description**

Calculates the coefficient of variation ( $CV = sd / mean$ ) for numeric vectors, matrices, data frames, or tibbles.

**Usage**

```
summarise_coef_of_variation(x)
```

**Arguments**

x                    A numeric vector, matrix, data frame, or tibble.

**Value**

A tibble: - If input has one numeric column or is a numeric vector: a tibble with a single value.  
- If input has multiple numeric columns: a tibble with variable names and coefficient of variation values.

**Examples**

```
summarise_coef_of_variation(iris)
summarise_coef_of_variation(iris$Petal.Length)
summarise_coef_of_variation(data.frame(a = rnorm(100), b = runif(100)))
```

---

summarise\_correlation *Summarise Correlation Matrix with Optional Significance Tests*

---

**Description**

Computes correlations between numeric variables of a data frame, or between two vectors. Optionally tests statistical significance (p-value)

**Usage**

```
summarise_correlation(  
  x,  
  y = NULL,  
  method = c("pearson", "kendall", "spearman"),  
  cor_test = FALSE  
)
```

**Arguments**

x	A numeric vector, matrix, data frame, or tibble.
y	Optional. A second numeric vector, matrix, or data frame (same dimensions as 'x').
method	Character. One of "pearson" (default), "kendall", or "spearman".
cor_test	Logical. If TRUE, uses 'cor.test()' and includes p-values. If FALSE, uses 'cor()' only.

**Value**

A tibble with variables, correlations, and optionally p-values. Significant results ( $p < 0.05$ ) are printed in red in the console.

**Examples**

```
summarise_correlation(iris)
summarise_correlation(iris$Sepal.Length, iris$Petal.Length, cor_test = TRUE)
```

---

summarise\_frequency    *Summarise Frequency Table*

---

**Description**

Computes the frequency and relative frequency (or percentage) of factor or character variables in a data frame or vector.

**Usage**

```
summarise_frequency(
  data,
  select = NULL,
  as_percent = FALSE,
  sort_by = NULL,
  top_n = Inf
)
```

**Arguments**

data	A character/factor vector, or a data frame/tibble.
select	Optional. One or more variable names to compute frequencies for. If NULL, all factor/character columns are used.
as_percent	Logical. If TRUE, relative frequencies are returned as percentages (%). Default is FALSE (proportions).
sort_by	Optional. If "N", sorts by frequency; if "group", sorts alphabetically; or "%N" (if as_percent = TRUE). Default is no sorting.
top_n	Integer. Show only the top N values

**Value**

A tibble with the following columns:

- variable** The name of the variable.
- group** The group/category values of the variable.
- N** The count (frequency) of each group.
- %N** The proportion or percentage of each group.

**Examples**

```
summarise_frequency(iris, select = "Species")
summarise_frequency(iris, as_percent = TRUE, sort_by = "N", top_n = 2)
summarise_frequency(data.frame(group = c("A", "A", "B", "C", "A")), as_percent = TRUE)
```

---

summarise\_group\_stats *Summarize Grouped Statistics*

---

**Description**

Groups a data frame by one or more variables and summarizes the selected numeric columns using basic statistic functions. Handles missing values by replacement with zero or removal of rows.

**Usage**

```
summarise_group_stats(
  df,
  group_var,
  values,
  m_functions = c("mean", "sd", "length"),
  replace_na = FALSE,
  remove_na = FALSE
)
```

**Arguments**

- df** A data frame or tibble containing the data.
- group\_var** A character vector of column names to group by.
- values** A character vector of numeric column names to summarize.
- m\_functions** A character vector of functions to apply (e.g., "mean", "sd", "length"). Default is c("mean", "sd", "length").
- replace\_na** Logical. If TRUE, missing values in numeric columns are replaced with 0. Default is FALSE.
- remove\_na** Logical. If TRUE, rows with missing values in group or value columns are removed. Default is FALSE.

**Value**

A tibble with grouped and summarized results.

**Examples**

```
summarise_group_stats(iris, group_var = "Species",
  values = c("Sepal.Length", "Petal.Width"))
summarise_group_stats(mtcars,
  group_var = c("cyl", "gear"),
  values = c("mpg", "hp"), remove_na = TRUE)
```

---

summarise_kurtosis	<i>Summarise Kurtosis</i>
--------------------	---------------------------

---

**Description**

Calculates the kurtosis (default: **excess kurtosis**) of numeric vectors, matrices, data frames, or tibbles. Supports both the "standard" and "unbiased" methods and optionally returns **raw kurtosis**.

**Usage**

```
summarise_kurtosis(x, method = c("standard", "unbiased"), excess = TRUE)
```

**Arguments**

x	A numeric vector, matrix, data frame, or tibble.
method	Character. Method for kurtosis calculation: "standard" (default) or "unbiased".
excess	Logical. If TRUE (default), returns <b>excess kurtosis</b> (minus 3); if FALSE, returns <b>raw kurtosis</b> .

**Value**

A tibble: - If input has one numeric column (or is a vector), a single-row tibble. - If input has multiple numeric columns, a tibble with variable names and kurtosis values.

**Examples**

```
summarise_kurtosis(iris)
summarise_kurtosis(iris, method = "unbiased")
summarise_kurtosis(iris, excess = FALSE) # Raw kurtosis
summarise_kurtosis(iris$Sepal.Width)
```

---

summarise_skewness	<i>Summarise Skewness</i>
--------------------	---------------------------

---

### Description

Calculates skewness for numeric vectors, matrices, data frames, or tibbles using Pearson's moment coefficient.

### Usage

```
summarise_skewness(x)
```

### Arguments

x                    A numeric vector, matrix, data frame, or tibble.

### Value

A tibble: - If input has one numeric column or is a numeric vector: a tibble with a single value. - If input has multiple numeric columns: a tibble with variable names and skewness values.

### Examples

```
summarise_skewness(iris)
summarise_skewness(as.vector(iris$Sepal.Width))
summarise_skewness(data.frame(a = rnorm(100), b = rgamma(100, 2)))
```

---

summarise_statistics	<i>Summarise Descriptive Statistics with Optional Testing</i>
----------------------	---------------------------------------------------------------

---

### Description

Computes descriptive statistics for numeric data. Optionally groups by a variable and includes Shapiro-Wilk and group significance testing. Can color console output for significant differences.

### Usage

```
summarise_statistics(
  data,
  group_var = NULL,
  normality_test = FALSE,
  group_test = FALSE,
  show_colors = TRUE
)
```

**Arguments**

<code>data</code>	A numeric vector, matrix, or data frame.
<code>group_var</code>	Optional. A character name of a grouping variable.
<code>normality_test</code>	Logical. If TRUE, performs Shapiro-Wilk test for normality.
<code>group_test</code>	Logical. If TRUE and ‘group_var’ is set, performs group-wise significance tests (t-test, ANOVA, etc.).
<code>show_colors</code>	Logical. If TRUE and ‘group_test’ is TRUE, prints colored console output for significant group results. Default is TRUE.

**Value**

A tibble with descriptive statistics and optional test results per numeric variable.

**Examples**

```
summarise_statistics(iris, group_var = "Species", group_test = TRUE)
```

# Index

`select_non_numeric_cols`, 2  
`select_numeric_cols`, 3  
`str_replace_many`, 3  
`summarise_boxplot_stats`, 4  
`summarise_coef_of_variation`, 5  
`summarise_correlation`, 5  
`summarise_frequency`, 6  
`summarise_group_stats`, 7  
`summarise_kurtosis`, 8  
`summarise_skewness`, 9  
`summarise_statistics`, 9