Package ‘sumvar’

June 13, 2025

Title Summarise Continuous, Date and Categorical Variables, Check for Duplicates and Missing Data

Version 0.1

Description Explore continuous, date and categorical variables. 'sumvar' aims to bring the ease and simplicity of the ``sum'' and ``tab'' functions from 'stata'.

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, ggplot2, lubridate, magrittr, patchwork, purrr, rlang, scales, stats, tibble, tidyr, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/alstockdale/sumvar,
https://alstockdale.github.io/sumvar/

BugReports https://github.com/alstockdale/sumvar/issues

License MIT + file LICENSE

VignetteBuilder knitr

NeedsCompilation no

Author Alexander Stockdale [aut, cre]

Maintainer Alexander Stockdale <a.stockdale@liverpool.ac.uk>

Repository CRAN

Date/Publication 2025-06-13 20:00:02 UTC

Contents

dist_date *Summarize and visualize a date variable*

Description

Summarises the minimum, maximum, median, and interquartile range of a date variable, optionally stratified by a grouping variable. Produces a histogram and (optionally) a density plot.

Usage

```
dist_date(data, var, by = NULL)
```

Arguments

`data`	A data frame or tibble.
`var`	The date variable to summarise.
`by`	Optional grouping variable.

Value

A tibble with summary statistics for the date variable.

See Also

`dist_sum` for continuous variables.

Examples

```
# Example ungrouped
df <- tibble::tibble(
  dt = as.Date("2020-01-01") + sample(0:1000, 100, TRUE)
)
dist_date(df, dt)

# Example grouped
df2 <- tibble::tibble(
  dt = as.Date("2020-01-01") + sample(0:1000, 100, TRUE),
  grp = sample(1:2, 100, TRUE)
)
dist_date(df2, dt, grp)
# Note this function accepts a pipe from dplyr eg. df %>% dist_date(date_var, group_var)
```

dist_sum *Explore a continuous variable.*

Description

Summarises the median, interquartile range, mean, standard deviation, confidence intervals of the mean and produces a density plot, stratified by a second grouping variable.

Provides frequentist hypothesis tests for comparison between groups: T test and Wilcoxon rank sum for 2 groups, Anova and Kruskall wallis test for 3 or more groups.

The function accepts an input from a dplyr pipe "%>%" and outputs the results as a tibble.

Usage

```
dist_sum(data, var, by = NULL)
```

Arguments

data	The data frame or tibble
var	The variable you would like to summarise
by	The grouping variable

Value

A tibble with a summary of the variable frequency (n), number of missing observations (n_miss), median, interquartile range, mean, SD, 95% confidence intervals of the mean (using the Z distribution), and density plots.

Shows the T test (p_ttest) and Wilcoxon rank sum (p_wilcox) hypothesis tests when there are two groups And an Anova test (p_anova) and Kruskal-Wallis test (p_kruskal) when there are three or more groups.

Examples

```
example_data <- dplyr::tibble(id = 1:100, age = rnorm(100, mean = 30, sd = 10),
                              group = sample(c("a", "b", "c", "d"),
                              size = 100, replace = TRUE))
dist_sum(example_data, age, group)
example_data <- dplyr::tibble(id = 1:100, age = rnorm(100, mean = 30, sd = 10),
                             sex = sample(c("male", "female"),
                             size = 100, replace = TRUE))
dist_sum(example_data, age, sex)
summary <- dist_sum(example_data, age, sex) # Save summary statistics as a tibble.
```

dup *Explore duplicate and missing data*

Description

Provides an integer value for the number of duplicates found within a variable The function accepts an input from a dplyr pipe "%>%" and outputs the results as a tibble.

eg. example_data %>% dup(variable)

Usage

```
dup(data, var = NULL)
```

Arguments

`data`	The data frame or tibble
`var`	The variable to assess

Value

A tibble with the number and percentage of duplicate values found, and the number of missing values (NA), together with percentages.

Examples

```
example_data <- dplyr::tibble(id = 1:200, age = round(rnorm(200, mean = 30, sd = 50), digits=0))
example_data$age[sample(1:200, size = 15)] <- NA  # Replace 15 values with missing.
dup(example_data, age)
# It is also possible to pass a whole database to dup and it will explore all variables.
example_data <- dplyr::tibble(age = round(rnorm(200, mean = 30, sd = 50), digits=0),
                              sex = sample(c("Male", "Female"), 200, TRUE),
                    favourite_colour = sample(c("Red", "Blue", "Purple"), 200, TRUE))
example_data$age[sample(1:200, size = 15)] <- NA  # Replace 15 values with missing.
example_data$sex[sample(1:200, size = 32)] <- NA  # Replace 32 values with missing.
dup(example_data)
```

sumvar *sumvar: Summarise Continuous and Categorical Variables in R*

Description

The sumvar package explores continuous and categorical variables. sumvar brings the ease and simplicity of the "sum" and "tab" functions from Stata to R.

- To explore a continuous variable, use `dist_sum()`. You can stratify by a grouping variable: `df %>% dist_sum(var, group)`
- To explore dates, use `dist_date()`; usage is the same as `dist_sum()`.
- To summarise a single categorical variable use `tab1()`, e.g. `df %>% tab1(var)`. For a two-way table, use `tab()`, e.g. `df %>% tab(var1, var2)`. Both include options for frequentist hypothesis tests.
- Explore duplicates and missing values with with `dup()`.

All functions are tidyverse/dplyr-friendly and accept the `%>%` pipe, outputting results as a tibble. You can save outputs for further manipulation, e.g. `summary <- df %>% dist_sum(var)`.

Author(s)

Maintainer: Alexander Stockdale `<a.stockdale@liverpool.ac.uk>`

See Also

Useful links:

- https://github.com/alstockdale/sumvar
- https://alstockdale.github.io/sumvar/
- Report bugs at https://github.com/alstockdale/sumvar/issues

`tab` *Create a cross-tabulation of two categorial variables*

Description

Creates a "n x n" cross-tabulation of two categorical variables, with row percentages. Includes options for adding frequentist hypothesis testing.

The function accepts an input from a dplyr pipe "%>%" and outputs the results as a tibble.

eg. example_data %>% tab(variable1, variable2)

Usage

```
tab(data, variable1, variable2, test = "none")
```

Arguments

`data`	The data frame or tibble
`variable1`	The first categorical variable
`variable2`	The second categorical variable
`test`	Optional frequentist hypothesis test, use test=exact for Fisher's exact or test=chi for Chi squared

Value

A tibble with a cross-tabulation of frequencies and row percentages

Examples

```
example_data <- dplyr::tibble(id = 1:100, group1 = sample(c("a", "b", "c", "d"),
                                                  size = 100, replace = TRUE),
                                                  group2= sample(c("male", "female"),
                                                  size = 100, replace = TRUE))
example_data$group1[sample(1:100, size = 10)] <- NA  # Replace 10 with missing
tab(example_data, group1, group2)
summary <- tab(example_data, group1, group2) # Save summary statistics as a tibble.
```

---

tab1 *Summarise a categorial variable*

Description

Summarises frequencies and percentages for a categorical variable.

The function accepts an input from a dplyr pipe "%>%" and outputs the results as a tibble. eg. example_data %>% tab1(variable)

Usage

```
tab1(data, variable, dp = 1)
```

Arguments

`data`	The data frame or tibble
`variable`	The categorical variable you would like to summarise
`dp`	The number of decimal places for percentages (default=2)

Value

A tibble with frequencies and percentages

Examples

```
example_data <- dplyr::tibble(id = 1:100, group = sample(c("a", "b", "c", "d"),
                                                  size = 100, replace = TRUE))
example_data$group[sample(1:100, size = 10)] <- NA  # Replace 10 with missing
tab1(example_data, group)
summary <- tab1(example_data, group) # Save summary statistics as a tibble.
```

Index