# Package 'ezr'

October 13, 2022

**Title** Easy Use of R via Shiny App for Basic Analyses of Experimental Data

**Version** 0.1.5

**Description** Runs a Shiny App in the local machine for basic statistical and graphical analyses. The point-and-click interface of Shiny App enables obtaining the same analysis outputs (e.g., plots and tables) more quickly, as compared with typing the required code in R, especially for users without much experience or expertise with coding. Examples of possible analyses include tabulating descriptive statistics for a variable, creating histograms by experimental groups, and creating a scatter plot and calculating the correlation between two variables.

**License** GPL-3

**URL** https://github.com/jinkim3/ezr

**BugReports** https://github.com/jinkim3/ezr/issues

**Imports** data.table, DT, ggplot2, ggridges, moments, shiny, shinydashboard, stats, weights

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Jin Kim [aut, cre] (<https://orcid.org/0000-0002-5013-3958>)

**Maintainer** Jin Kim <jin.m.kim@yale.edu>

**Repository** CRAN

**Date/Publication** 2020-11-25 10:30:08 UTC

## R topics documented:

---

`desc_stats` *Descriptive statistics*

---

## Description

Returns descriptive statistics for a numeric vector.

## Usage

```
desc_stats(vector = NULL, notify_na_count = NULL)
```

## Arguments

| | |
|---|---|
| `vector` | a numeric vector |
| `notify_na_count` | if `TRUE`, notify how many observations were removed due to missing values. By default, NA count will be printed only if there are any NA values. |

## Value

a named numeric vector

## Examples

```
desc_stats(1:100)
desc_stats(c(1:100, NA))
```

---

`histogram_by_group` *Histogram by group*

---

## Description

Creates histograms by group to compare distributions

### Usage

```
histogram_by_group(
  data = NULL,
  iv_name = NULL,
  dv_name = NULL,
  order_of_groups_top_to_bot = NULL,
  number_of_bins = 40,
  space_between_histograms = 0.15
)
```

### Arguments

| | |
|---|---|
| data | a data object (a data frame or a data.table) |
| iv_name | name of the independent variable |
| dv_name | name of the dependent variable |
| order_of_groups_top_to_bot | a character vector indicating the desired presentation order of levels in the independent variable (from the top to bottom). Omitting a group in this argument will remove the group in the set of histograms. |
| number_of_bins | number of bins for the histograms (default = 40) |
| space_between_histograms | space between histograms (minimum = 0, maximum = 1, default = 0.15) |

### Value

a ggplot object

### Examples

```
histogram_by_group(data = mtcars, iv_name = "cyl", dv_name = "mpg")
histogram_by_group(data = mtcars, iv_name = "cyl", dv_name = "mpg",
order_of_groups_top_to_bot = c("8", "4"), number_of_bins = 10,
space_between_histograms = 0.5)
```

---

## pretty_round_p_value *Pretty round p-value*

---

### Description

Pretty round p-value

### Usage

```
pretty_round_p_value(
  p_value_vector = NULL,
  round_digits_after_decimal = 3,
  include_p_equals = FALSE
)
```

### Arguments

| | |
|---|---|
| p_value_vector | one number or a numeric vector |
| round_digits_after_decimal | round to nth digit after decimal |
| include_p_equals | if TRUE, output will be a string of mathematical expression including "p", e.g., "p < .01" |

### Examples

```
pretty_round_p_value(p_value_vector = 0.049,
round_digits_after_decimal = 2, include_p_equals = FALSE)
pretty_round_p_value(c(0.0015, 0.0014), include_p_equals = TRUE)
```

---

## scatterplot *Scatterplot*

---

### Description

Creates a scatter plot and calculates a correlation between two variables

### Usage

```
scatterplot(
  data = NULL,
  x_var_name = NULL,
  y_var_name = NULL,
  point_label_var_name = NULL,
  weight_var_name = NULL,
  alpha = 1,
  annotate_stats = FALSE,
  line_of_fit_type = "lm",
  ci_for_line_of_fit = FALSE,
  x_axis_label = NULL,
  y_axis_label = NULL,
  point_labels_size_range = c(3, 12),
  jitter_x_percent = 0,
  jitter_y_percent = 0
)
```

### Arguments

| | |
|---|---|
| data | a data object (a data frame or a data.table) |
| x_var_name | name of the variable that will go on the x axis |
| y_var_name | name of the variable that will go on the y axis |

point_label_var_name
: name of the variable that will be used to label individual observations

weight_var_name
: name of the variable by which to weight the individual observations for calculating correlation and plotting the line of fit

alpha
: opacity of the dots (0 = completely transparent, 1 = completely opaque)

annotate_stats
: if TRUE, the correlation and p-value will be annotated at the top of the plot

line_of_fit_type
: if line_of_fit_type = "lm", a regression line will be fit; if line_of_fit_type = "loess", a local regression line will be fit; if line_of_fit_type = "none", no line will be fit

ci_for_line_of_fit
: if ci_for_line_of_fit = TRUE, confidence interval for the line of fit will be shaded

x_axis_label
: alternative label for the x axis

y_axis_label
: alternative label for the y axis

point_labels_size_range
: minimum and maximum size for dots on the plot when they are weighted

jitter_x_percent
: horizontally jitter dots by a percentage of the range of x values

jitter_y_percent
: vertically jitter dots by a percentage of the range of y values

## Value

a ggplot object

## Examples

```
scatterplot(data = mtcars, x_var_name = "wt", y_var_name = "mpg")
scatterplot(data = mtcars, x_var_name = "wt", y_var_name = "mpg",
point_label_var_name = "hp", weight_var_name = "drat",
annotate_stats = TRUE)
scatterplot(data = mtcars, x_var_name = "wt", y_var_name = "mpg",
point_label_var_name = "hp", weight_var_name = "cyl",
annotate_stats = TRUE)
```

---

# se_of_mean *Standard error of the mean*

---

## Description

Standard error of the mean

## Usage

```
se_of_mean(vector, na.rm = TRUE, notify_na_count = NULL)
```

## Arguments

| | |
|---|---|
| vector | a numeric vector |
| na.rm | if TRUE, NA values will be removed before calculation |
| notify_na_count | if TRUE, notify how many observations were removed due to missing values. By default, NA count will be printed only if there are any NA values. |

## Examples

```
se_of_mean(c(1:10, NA))
```

---

# start_ezr *Start ezr*

---

## Description

Starts the ezr program on the local machine

## Usage

```
start_ezr(
  data_for_ezr = NULL,
  sigfig = 3,
  select_list_max = 1e+05,
  ezr_saved_analysis_file_name = "ezr_saved_analysis.csv",
  ezr_run_analysis_file_name = "ezr_run_analysis.csv"
)
```

## Arguments

| | |
|---|---|
| data_for_ezr | a data object (a data frame or a data.table) |
| sigfig | number of significant digits to round to |
| select_list_max | maximum number of variable names to display for dropdown menus |
| ezr_saved_analysis_file_name | name of the .csv file on which saved analysis will be recorded (default = "ezr_saved_analysis.csv") |
| ezr_run_analysis_file_name | name of the .csv file on which all conducted analyses will be recorded (default = "ezr_run_analysis.csv") |

## Value

There will be no output from this function. Rather, the ezr program will open on a new tab or window of the local machine's web browser

## Examples

```
if (interactive()) {start_ezr(data = mtcars)}
```

---

`tabulate_vector` *Tabulate vector*

---

## Description

Shows frequency and proportion of unique values in a table format

## Usage

```
tabulate_vector(
  vector = NULL,
  na.rm = TRUE,
  sort_by_decreasing_count = NULL,
  sort_by_increasing_count = NULL,
  sort_by_decreasing_value = NULL,
  sort_by_increasing_value = NULL,
  total_included = TRUE,
  sigfigs = NULL,
  round_digits_after_decimal = NULL,
  output_type = "dt"
)
```

## Arguments

| | |
|---|---|
| `vector` | a character or numeric vector |
| `na.rm` | if `TRUE`, NA values will be removed before calculating frequencies and proportions. |
| `sort_by_decreasing_count` | if `TRUE`, the output table will be sorted in the order of decreasing frequency. |
| `sort_by_increasing_count` | if `TRUE`, the output table will be sorted in the order of increasing frequency. |
| `sort_by_decreasing_value` | if `TRUE`, the output table will be sorted in the order of decreasing value. |
| `sort_by_increasing_value` | if `TRUE`, the output table will be sorted in the order of increasing value. |
| `total_included` | if `TRUE`, the output table will include a row for total counts. |
| `sigfigs` | number of significant digits to round to |

`round_digits_after_decimal`
: round to nth digit after decimal (alternative to `sigfigs`)

`output_type`
: if `output_type = "df"`, return a data.frame. By default, `output_type = "dt"`, which will return a data.table.

## Value

a data.table or data.frame

## Examples

```
tabulate_vector(c("a", "b", "b", "c", "c", "c", NA))
tabulate_vector(c("a", "b", "b", "c", "c", "c", NA),
sort_by_increasing_count = TRUE)
tabulate_vector(c("a", "b", "b", "c", "c", "c", NA),
sort_by_decreasing_value = TRUE)
tabulate_vector(c("a", "b", "b", "c", "c", "c", NA),
sort_by_increasing_value = TRUE)
tabulate_vector(c("a", "b", "b", "c", "c", "c", NA),
sigfigs = 4)
tabulate_vector(c("a", "b", "b", "c", "c", "c", NA),
round_digits_after_decimal = 1)
tabulate_vector(c("a", "b", "b", "c", "c", "c", NA),
output_type = "df")
```

# Index