Package: AnalysisLin (via r-universe)

October 26, 2024

Type Package

Title Exploratory Data Analysis

Version 0.1.0

Description A quick and effective data exploration toolkit. It provides essential features, including a descriptive statistics table for a quick overview of your dataset, interactive distribution plots to visualize variable patterns, Principal Component Analysis for dimensionality reduction and feature analysis, missing value imputation methods, and correlation analysis.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

Date 2024-01-22

Imports Hmisc, ggplot2, plotly, stats, caret, htmltools, DT, magrittr, RANN

Repository https://zhiweilin27.r-universe.dev

RemoteUrl https://github.com/zhiweilin27/analysislin

RemoteRef HEAD

RemoteSha 49d9388b2bf275079275f8b645335105ffdb38e8

Contents

`automate_pca` *Principle Component Analysis*

Description

Principle Component Analysis

Usage

```
automate_pca(data, scale = TRUE, variance_threshold = 0.9, scree_plot = TRUE)
```

Arguments

`data`	input dataset
`scale`	a logical argument(default TRUE) that determines if appling standardized scaling to the dataset
`variance_threshold`	an argument(default is 0.9) for set a variance_threshold
`scree_plot`	a logical argument(default TRUE) that determines if scree plot is generated

Value

rotation and score of PCA and scree plot(optional)

Examples

```
automate_pca(data(mtcars))
```

bar_plot *Bar Plots for Categorical Variables*

Description

This function generates bar plots for all categorical variables in the input data frame. Bar plots offer a visual representation of the distribution of categorical variables, making it easy to understand the frequency of each category. They are particularly useful for exploring patterns, identifying dominant categories, and comparing the relative frequencies of different levels within each variable.

Usage

```
bar_plot(
  data,
  fill = "skyblue",
  color = "black",
  width = 0.7,
  subplot = FALSE,
  nrow = 2,
  margin = 0.1
)
```

Arguments

data	The input data frame containing categorical variables.
fill	Fill color for the bars (default: "skyblue").
color	Border color of the bars (default: "black").
width	Width of the bars (default: 0.7).
subplot	A logical argument (default: FALSE) indicating whether to create subplots.
nrow	Number of rows for subplots (if subplot is TRUE, default: 2).
margin	Margin for subplots (if subplot is TRUE, default: 0.1).

Value

A list of bar plots.

Examples

```
data(iris)
bar_plot(iris)
```

categoric_plot *Categorical Variables Plots*

Description

Categorical Variables Plots

Usage

```
categoric_plot(
  data,
  pie = TRUE,
  pie_legend = TRUE,
  pie_legend_size = 0.5,
  pie_legend_position = "bottom",
  pie_inset = c(0, -0.15),
  bar = TRUE,
  bar_legend = TRUE,
  bar_legend_size = 0.5,
  bar_legend_position = "bottom",
  bar_inset = c(0, -0.4),
  n_col = 1,
  bar_width = 0.8,
  bar_height = NULL
)
```

Arguments

data	input data
pie	a logical argument(default TRUE) that determines if pie plot is generated
pie_legend	a logical argument(default TRUE) that determines if a legend of pie is generated
pie_legend_size	an argument(default is 0.5) that determines size of the legend
pie_legend_position	an argument(dafult is bottom) that determines the position of the legend
bar	a logical argument(default TRUE) that determines if bar plot is generated
bar_legend	a logical argument(default TRUE) that determines if a legend of bar is generated
bar_legend_size	an argument(default is 0.5) that determines size of the legend
bar_legend_position	an argument(dafult is bottom) that determines the position of the legend
n_col	an argument that determine how many plot being put on the same rows
bar_width	width of bar plot
bar_height	height of bar plot

Value

a list of pie charts

Examples

```
categoric_plot(data(mtcars))
```

---

corr_cluster *Correlation Clustering*

Description

This function performs hierarchical clustering on a correlation matrix, providing insights into the relationships between variables. It generates a dendrogram visualizing the hierarchical clustering of variables based on their correlation patterns.

Usage

```
corr_cluster(data, type = "pearson", method = "complete", hclust_method = NULL)
```

Arguments

`data`	Input data frame.
`type`	The type of correlation to be computed. It can be "pearson", "kendall", or "spearman".
`method`	The method for hierarchical clustering. It can be "complete", "single", "average", "ward.D", "ward.D2", "mcquitty", "median", or "centroid".
`hclust_method`	The hierarchical clustering method. It can be "complete", "single", "average", "ward.D", "ward.D2", "mcquitty", "median", or "centroid".

Value

A dendrogram visualizing the hierarchical clustering of variables based on the correlation matrix.

Examples

```
data(mtcars)
corr_cluster(data = mtcars, type = 'pearson', method = 'complete')
```

corr_matrix *Correlation Matrix*

Description

Column 1: Row names representing Variable 1 in the correlation test.

Column 2: Column names representing Variable 2 in the correlation test.

Column 3: The correlation coefficients quantifying the strength and direction of the relationship.

Column 4: The p-values associated with the correlations, indicating the statistical significance of the observed relationships. Lower p-values suggest stronger evidence against the null hypothesis.

The table provides valuable insights into the relationships between variables, helping to identify statistically significant correlations.

Usage

```
corr_matrix(
  data,
  type = "pearson",
  corr_plot = FALSE,
  sig.level = 0.01,
  highlight = FALSE
)
```

Arguments

data	Input dataset.
type	Pearson or Spearman correlation, default is Pearson.
corr_plot	Generate a correlation matrix plot, default is false.
sig.level	Significant level. Default is 0.01.
highlight	Highlight p-value(s) that is less than sig.level, default is FALSE

Value

A data frame which contains row names, column names, correlation coefficients, and p-values.

A plot of the correlation if corrplot is set to be true.

Examples

```
data(mtcars)
corr_matrix(mtcars, type = 'pearson')
```

`dens_plot` *Numerical Variables Density Plots*

Description

This function generates density plots for all numerical variables in the input data frame. It offers a vivid and effective visual summary of the distribution of each numerical variable, helping in a quick understanding of their central tendency, spread, and shape.

Usage

```
dens_plot(
  data,
  fill = "skyblue",
  color = "black",
  alpha = 0.7,
  subplot = FALSE,
  nrow = 2,
  margin = 0.1
)
```

Arguments

`data`	The input data frame containing numerical variables.
`fill`	The fill color of the density plot (default: "skyblue").
`color`	The line color of the density plot (default: "black").
`alpha`	The transparency of the density plot (default: 0.7).
`subplot`	A logical argument (default: FALSE) indicating whether to create subplots.
`nrow`	Number of rows for subplots (if subplot is TRUE, default: 2).
`margin`	Margin for subplots (if subplot is TRUE, default: 0.1).

Value

A list of density plots.

Examples

```
data(mtcars)
dens_plot(mtcars)
```

desc_stat *Descriptive Statistics*

Description

desc_stat() function calculates various key descriptive statistics for each variables in the provided data set. The function computes the count, number of unique values, duplicate count, number of missing values, null rate, data type, minimum value, 25th percentile, mean, median, 75th percentile, maximum value, standard deviation, kurtosis, skewness, and jarque_pvalue for each variable.

Usage

```
desc_stat(
  data,
  count = TRUE,
  unique = TRUE,
  duplicate = TRUE,
  null = TRUE,
  null_rate = TRUE,
  type = TRUE,
  min = TRUE,
  p25 = TRUE,
  mean = TRUE,
  median = TRUE,
  p75 = TRUE,
  max = TRUE,
  sd = TRUE,
  skewness = FALSE,
  kurtosis = FALSE,
  shapiro = FALSE,
  kolmogorov = FALSE,
  anderson = FALSE,
  lilliefors = FALSE,
  jarque = FALSE
)
```

Arguments

data	input dataset
count	An logical argument(default TRUE) that determines if count is included in the output
unique	An logical argument(default TRUE) that determines if unique is included in the output
duplicate	An logical argument(default TRUE) that determines if duplicate is included in the output

null	An logical argument(default TRUE) that determines if null is included in the output
null_rate	An logical argument(default TRUE) that determines if null_rate is included in the output
type	An logical argument(default TRUE) that determines if type is included in the output
min	An logical argument(default TRUE) that determines if min is included in the output
p25	An logical argument(default TRUE) that determines if p25 is included in the output
mean	An logical argument(default TRUE) that determines if mean is included in the output
median	An logical argument(default TRUE) that determines if median is included in the output
p75	An logical argument(default TRUE) that determines if p75 is included in the output
max	An logical argument(default TRUE) that determines if max is included in the output
sd	An logical argument(default TRUE) that determines if sd is included in the output
skewness	An logical argument(default FALSE) that determines if skewness is included in the output
kurtosis	An logical argument(default FALSE) that determines if kurtosis is included in the output
shapiro	An logical argument(default FALSE) that determines if shapiro p-value is included in the output
kolmogorov	An logical argument(default FALSE) that determines if kolmogorov p-value is included in the output
anderson	An logical argument(default FALSE) that determines if anderson p-value is included in the output
lilliefors	An logical argument(default FALSE) that determines if lilliefors p-value is included in the output
jarque	An logical argument(default FALSE) that determines if jarque p-value is included in the output

Value

A data frame which summarizes the characteristics of a data set

Examples

```
data(mtcars)
desc_stat(mtcars)
```

`hist_plot` *Histogram Plot for Numerical Variables*

Description

This function generates histogram plots for all numerical variables in the input data frame. It offers a vivid and effective visual summary of the distribution of each numerical variable, helping in a quick understanding of their central tendency, spread, and shape.

Usage

```
hist_plot(
  data,
  fill = "skyblue",
  color = "black",
  alpha = 0.7,
  subplot = FALSE,
  nrow = 2,
  margin = 0.1
)
```

Arguments

`data`	The input data frame containing numerical variables.
`fill`	The fill color for the histogram bars (default: "skyblue").
`color`	The border color for the histogram bars (default: "black").
`alpha`	The alpha (transparency) value for the histogram bars (default: 0.7).
`subplot`	A logical argument (default: FALSE) indicating whether to create subplots for each variable.
`nrow`	Number of rows for subplots (used when subplot is TRUE, default: 2).
`margin`	Margin for subplots (used when subplot is TRUE, default: 0.1).

Value

A list of histogram plot.

Examples

```
hist_plot(data = mtcars, fill = "skyblue", color = "black", alpha = 0.7, subplot = FALSE)
```

impute_missing *Missing Value Imputation*

Description

This function performs missing value imputation in the input data using various methods. The available imputation methods are:

- "mean": Imputes missing values with the mean of the variable. - "median": Imputes missing values with the median of the variable. - "mode": Imputes missing values with the mode of the variable (for categorical data). - "locf": Imputes missing values using the Last Observation Carried Forward method. - "knn": Imputes missing values using the k-Nearest Neighbors algorithm (specify k).

Usage

```
impute_missing(data, method = "mean", k = NULL)
```

Arguments

data	Input data.
method	Method of handling missing values: "mean," "median," "mode," "locf," or "knn."
k	Value of the number of neighbors to be checked (only for knn method). Default is NULL.

Value

a data frame with imputed missing values

Examples

```
data(airquality)
impute_missing(airquality, method='mean')
```

missing_values_plot *Missing Values Plot*

Description

This function generates plots to visualize missing values in a data frame. It includes two types of plots: - A percentage plot: Displays the percentage of missing values for each variable, allowing quick identification of variables with high missingness. - A row plot: Illustrates the distribution of missing values across rows, providing insights into patterns of missingness.

Usage

```
missing_values_plot(df, percentage = TRUE, row = TRUE)
```

Arguments

df	The input data frame.
percentage	A logical argument (default: TRUE) to generate a percentage plot.
row	A logical argument (default: TRUE) to generate a row plot.

Value

A list of plots, including a percentage plot and/or a row plot.

Examples

```
data("airquality")
missing_values_plot(df = airquality, percentage = TRUE, row = TRUE)
```

---

numeric_plot *Numerical Variables Distribution*

Description

Numerical Variables Distribution

Usage

```
numeric_plot(data, hist = TRUE, prob = FALSE, dens = FALSE)
```

Arguments

data	input data
hist	a logical argument(default TRUE) that determines if histogram is generated
prob	a logical argument(default FALSE) that determines it is a probability histogram or relative frequency histogram.
dens	a logical argument(default FALSE) that determine if density line is generated

Value

a list of plots

Examples

```
numeric_plot(data(mtcars),prob=T,dens=T)
```

pca *Principal Component Analysis (PCA)*

Description

This function performs Principal Component Analysis (PCA) on the input data, providing a detailed analysis of variance, eigenvalues, and eigenvectors. It offers options to generate a scree plot for visualizing variance explained by each principal component and a biplot to understand the relationship between variables and observations in reduced dimensions.

Usage

```
pca(
  data,
  variance_threshold = 0.9,
  center = TRUE,
  scale = FALSE,
  scree_plot = FALSE,
  biplot = FALSE,
  choices = 1:2,
  groups = NULL,
  length_scale = 1,
  scree_legend = TRUE,
  scree_legend_pos = c(0.7, 0.5)
)
```

Arguments

data	Numeric matrix or data frame containing the variables for PCA.
variance_threshold	Proportion of total variance to retain (default: 0.90).
center	Logical, indicating whether to center the data (default: TRUE).
scale	Logical, indicating whether to scale the data (default: FALSE).
scree_plot	Logical, whether to generate a scree plot (default: FALSE).
biplot	Logical, whether to generate a biplot (default: FALSE).
choices	Numeric vector of length 2, indicating the principal components to plot in the biplot.
groups	Optional grouping variable for coloring points in the biplot.
length_scale	Scaling factor for adjusting the length of vectors in the biplot (default: 1).
scree_legend	Logical, indicating whether to show legend in scree plot (default: True).
scree_legend_pos	A vector c(x, y) to adjust the position of the legend.

Value

A list containing: - summary_table: A matrix summarizing eigenvalues and cumulative variance explained. - scree_plot: A scree plot if scree_plot is TRUE. - biplot: A biplot if biplot is TRUE.

Examples

```
data(mtcars)
pca_result <- pca(mtcars, scree_plot = TRUE, biplot = TRUE)
pca_result$summary_table
pca_result$scree_plot
pca_result$biplot
```

---

pie_plot *Pie Plots for Categorical Variables*

Description

This function generates pie charts for categorical variables in the input data frame using plotly. Pie plots offer a visual representation of the distribution of categorical variables, making it easy to understand the frequency of each category. They are particularly useful for exploring patterns, identifying dominant categories, and comparing the relative frequencies of different levels within each variable.

Usage

```
pie_plot(data)
```

Arguments

data	The input data frame containing categorical variables.

Value

A list of pie charts.

Examples

```
data(iris)
pie_plot(iris)
```

qq_plot *QQ Plots for Numerical Variables*

Description

This function generates QQ plots for all numerical variables in the input data frame. QQ plots are valuable for assessing the distributional similarity between observed data and a theoretical normal distribution. It acts as a guide, revealing deviations from the expected norm, outliers, and the contours of distribution tails.

Usage

```
qq_plot(data, color = "skyblue", subplot = FALSE, nrow = 2, margin = 0.1)
```

Arguments

data	The input data frame containing numerical variables.
color	The color of the QQ plot line (default: "skyblue").
subplot	A logical argument (default: FALSE) indicating whether to create subplots.
nrow	Number of rows for subplots (if subplot is TRUE, default: 2).
margin	Margin for subplots (if subplot is TRUE, default: 0.1).

Value

A list of QQ plots.

Examples

```
data(mtcars)
qq_plot(mtcars)
```

Index