Package ‘corrViz’

June 30, 2023

Type Package

Title Visualise Correlations

Version 0.1.0

Author Alan Inglis

Maintainer Alan Inglis <alan.n.inglis@gmail.com>

Description An investigative tool designed to help users visualize correlations between variables in their datasets. This package aims to provide an easy and effective way to explore and visualize these correlations, making it easier to interpret and communicate results.

License GPL (>= 2)

Encoding UTF-8

Imports ggplot2, stats, grDevices, visNetwork, plotly, DendSer, gganimate, igraph, ggraph, circlize, GGally, purrr, shiny

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, scales, gifski

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-30 11:40:07 UTC

R topics documented:

`animSolar` *animSolar*

Description

This function creates an animated solar system plot of correlations between variables in a dataset.

Usage

```
animSolar(
  mat,
  sun = NULL,
  export = FALSE,
  num_frames = 100,
  path = NULL,
  gif_name = "solar_system.gif",
  fps = 60
)
```

Arguments

`mat` A square correlation matrix to visualise.

`sun` A character string specifying the column name in the dataset to be treated as the 'sun' in the solar system plot.

`export` A logical value specifying whether to export the animation as a GIF file, default is FALSE.

`num_frames` An integer specifying the number of frames in the animation, default is 100.

`path` A character string specifying the directory path where the GIF file will be saved, default is NULL.

`gif_name` A character string specifying the name of the GIF file. Must be in the format "myFile.gif"

`fps` An integer specifying the frames per second for the animation. Default is 60 and is only used when exporting a gif via `export = TRUE`.

Details

In a solar system correlation plot, the dependent variable of interest is positioned at the center, represented as the sun. The explanatory variables are depicted as planets orbiting around the sun, with their distance from the sun corresponding to the absolute value of their correlation with the dependent variable. Therefore, the greater the distance of a planet from the sun, the weaker the correlation between the explanatory variable and the dependent variable.

The num_frames argument is used to select the number of frames. Setting this to a low value will produce the plot quicker, however having a low number of frames will result in the "planets" jumping as the frames transition. Additionally, a low values of num_frames will affect the orbit of the animation when setting export = FALSE. This differs from the fps argument which sets the number of frames to play per second for use when exporting a gif.

Value

An animated solar system plot displaying correlations.

Examples

```
cm <- cor(mtcars)

animSolar(mat = cm,
          sun = 'mpg',
          export = FALSE,
          num_frames = 25)
```

corrBarplot *corrBarplot*

Description

This function creates a either a static or interactive bar plot of correlations between variables in a dataset.

Usage

```
corrBarplot(
  mat,
  interactive = TRUE,
  pal = colorRampPalette(c("cornflowerblue", "white", "tomato"))(100)
)
```

Arguments

mat	A square correlation matrix to visualise.
interactive	A logical value specifying whether to create an interactive ggplotly plot, default is TRUE
pal	A colour palette for the bar plot, default is colorRampPalette(c("cornflowerblue", "white", "tomato"))(100).

Details

Creates a static or interactive bar plot displaying correlation values. By hovering mouse over a bar, the variables and correlation value is shown.

Value

A static or interactive bar plot displaying correlations.

Examples

```
cm <- cor(mtcars)

corrBarplot(mat = cm,
            interactive = TRUE)
```

corrBubble *corrBubble*

Description

This function creates an interactive bubble plot of correlations between variables in a dataset.

Usage

```
corrBubble(
  mat,
  display = c("all", "upper", "lower"),
  pal = colorRampPalette(c("cornflowerblue", "white", "tomato"))(100)
)
```

Arguments

`mat`	A square correlation matrix to visualise.
`display`	A character vector specifying which part of the correlation matrix to display: 'all', 'upper', or 'lower', default is 'all'.
`pal`	A color palette for the bubble plot.

Details

Creates an interactive bubble plot displaying correlation values. By hovering mouse over a cell, the variables and correlation value is shown.

Value

An interactive bubble plot displaying correlations.

Examples

```
cm <- cor(mtcars)

corrBubble(mat = cm,
           display = 'all')
```

---

corrChord *corrChord*

Description

This function creates a chord plot of correlations between variables in a dataset.

Usage

```
corrChord(mat, threshold = 0, circle = FALSE)
```

Arguments

mat	A square correlation matrix to visualise.
threshold	A numeric value indicating the minimum absolute correlation value to display in the plot.
circle	A logical value indicating whether to use a circular layout (TRUE) or linear layout (FALSE), default is FALSE.

Details

When using a large amount of data, this plot can quickly become over complicated. It is recommended to filter the correlations using the `threshold` argument to simplify the visualisation.

Value

A chord plot displaying correlations.

Examples

```
cm <- cor(mtcars)

corrChord(mat = cm,
          threshold = 0.8)

corrChord(mat = cm,
          threshold = 0.8,
          circle = TRUE)
```

`corrCircle` *corrCircle*

Description

This function creates a circular plot of correlations between variables in a dataset.

Usage

```
corrCircle(mat, threshold = 0, ticks = FALSE)
```

Arguments

`mat` A square correlation matrix to visualise.

`threshold` A numeric value indicating the minimum absolute correlation value to display in the plot.

`ticks` A logical value indicating whether to display ticks (TRUE) or not (FALSE), default is FALSE.

Details

When using a large amount of data, this plot can quickly become over complicated. It is recommended to filter the correlations using the `threshold` argument to simplify the visualisation.

Value

A circular chord plot object displaying the correlations between variables.

Examples

```
cm <- cor(mtcars)

corrCircle(mat = cm,
         threshold = 0.8)
```

corrGrid — *Plot a correlation grid*

Description

Create a correlation grid plot to visualize correlations among the columns of a dataset

Usage

```
corrGrid(
  mat,
  display = c("all", "upper", "lower"),
  type = c("square", "circle", "text", "pie"),
  showDiag = "TRUE",
  pal = colorRampPalette(c("darkblue", "white", "darkred"))(100)
)
```

Arguments

`mat`	A square correlation matrix to visualise.
`display`	A character string, specifying the display type, one of "all", "upper", or "lower" (default: "all").
`type`	A character string, specifying the shape of the correlation coefficients, one of "square", "circle", or "text" (default: "square").
`showDiag`	A logical value, if TRUE (default), the diagonal of the correlation matrix is shown.
`pal`	A color palette function, used for the correlation coefficient colors.

Value

A correlation grid plot

Examples

```
cm <- cor(mtcars)
corr_grid_plot <- corrGrid(mat = cm,
                            type = 'square')
```

corrHeatmap *corrHeatmap*

Description

This function creates an interactive heatmap of correlations between variables in a dataset.

Usage

```
corrHeatmap(
  mat,
  display = c("all", "upper", "lower"),
  reorder = TRUE,
  pal = colorRampPalette(c("darkblue", "white", "darkred"))(100)
)
```

Arguments

`mat`	A square correlation matrix to visualise.
`display`	A character vector specifying which part of the correlation matrix to display: ’all’, ’upper’, or ’lower’, default is ’all’.
`reorder`	A logical value indicating whether to reorder the heatmap based on hierarchical clustering, default is TRUE.
`pal`	A color palette for the heatmap.

Details

Creates an interactive heatmap displaying correlation values. By hovering mouse over a cell, the variables and correlation value is shown.

Value

An interactive heatmap plot displaying correlations.

Examples

```
cm <- cor(mtcars)

corrHeatmap(mat = cm,
          display = 'all')
```

corrNetwork *corrNetwork*

Description

Creates an interactive Correlation Network Visualization

Usage

```
corrNetwork(
  mat,
  threshold = 0,
  layout = "layout_nicely",
  width = "100%",
  height = "400px",
  physics = TRUE
)
```

Arguments

`mat`	A square correlation matrix to visualise.
`threshold`	A numeric value indicating the minimum absolute correlation value to display in the plot.
`layout`	Any `igraph` layout to display the network.
`width`	The width of the viewing window.
`height`	The height of the viewing window.
`physics`	A logical value indicating whether to use physics-based layout. Default is TRUE.

Details

Each node in the network represents a variable where the width of the connecting edges represent the absolute value of the correlation. Positive correlations have red coloured edges whereas negative correlations have blue coloured edges.

Value

A network plot displaying correlations.

Examples

```
ci <- cor(iris[1:4])
corrNetwork(mat = ci, threshold = 0.5)

# Another example
cm <- cor(mtcars)

corrNetwork(mat = cm,
           threshold = 0.8,
           layout = 'layout_on_grid',
           physics = FALSE)
```

corrPairs *corrPairs*

Description

This function creates a pairwise correlation plot with annotated correlation coefficients and optional coloring by a specified variable. The plot can be interactive or static.

Usage

```
corrPairs(
  data,
  method = c("pearson", "kendall", "spearman"),
  interactive = TRUE,
  col_by = NULL
)
```

Arguments

data — A data frame containing the variables to be plotted.

method — A character string specifying the correlation method. One of "pearson", "kendall", or "spearman". Default is "pearson".

interactive — A logical value indicating whether the output plot should be interactive (TRUE) or static (FALSE). Default is TRUE.

col_by — An optional character string specifying the name of the column in the data frame to be used for coloring points. Default is NULL.

Value

A ggplotly object (if interactive = TRUE) or a ggplot object (if interactive = FALSE) displaying the pairwise correlation plot with annotated correlation coefficients and optional coloring by the specified variable.

Examples

```
corrPairs(data = mtcars[,1:4],
        method = "pearson",
        interactive = TRUE,
        col_by = "cyl")
```

corrSankey *corrSankey*

Description

Create an interactive Sankey diagram to visualize correlations

Usage

```
corrSankey(mat, threshold = 0, colour = FALSE)
```

Arguments

`mat`	A square correlation matrix to visualise.
`threshold`	A numeric value indicating the minimum absolute correlation value to include in the diagram. Default is 0 (include all correlations).
`colour`	A logical value indicating whether to color the links based on positive or negative correlation. Default is FALSE (links are grey).

Details

This function generates a Sankey diagram for a given data frame, correlation method, and correlation threshold, with an optional colour parameter.

Value

A plotly Sankey diagram object.

Examples

```
cm <- cor(mtcars)
corrSankey(mat = cm, threshold = 0.6)
corrSankey(mat = cm, threshold = 0.8, colour = TRUE)
```

corrShiny *corrShiny*

Description

Correlation Explorer Shiny App. This function creates a Shiny application to explore the correlation between variables in a given dataset.

Usage

```
corrShiny(
  data,
  x_var,
  y_var,
  color_var = NULL,
  size_var = NULL,
  correlation_method = "pearson"
)
```

Arguments

`data`	A data frame with the variables to be analyzed.
`x_var`	The name of the variable to be plotted on the X-axis.
`y_var`	The name of the variable to be plotted on the Y-axis.
`color_var`	The name of the variable to be used for coloring the points on the scatter plot.
`size_var`	The name of the variable to be used for sizing the points on the scatter plot.
`correlation_method`	The method to be used for computing the correlation coefficient, must be one of "pearson", "spearman" or "kendall".

Value

A Shiny app that displays a scatter plot and the correlation coefficient between two variables.

corrSolar *corrSolar*

Description

This function creates a solar system plot of correlations between variables in a dataset.

Usage

```
corrSolar(mat, sun = NULL)
```

Arguments

`mat`	A square correlation matrix to visualise.
`sun`	A character string specifying the column name in the dataset to be treated as the 'sun' in the solar system plot.

Details

In a solar system correlation plot, the dependent variable of interest is positioned at the center, represented as the sun. The explanatory variables are depicted as planets orbiting around the sun, with their distance from the sun corresponding to the absolute value of their correlation with the dependent variable. Therefore, the greater the distance of a planet from the sun, the weaker the correlation between the explanatory variable and the dependent variable.

Value

An solar system plot displaying correlations.

Examples

```
cm <- cor(mtcars)
corrSolar(mat = cm,
          sun = 'mpg')
```

matrix2long *matrix2long*

Description

Convert a Matrix to Long Format.

Usage

```
matrix2long(mat)
```

Arguments

`mat`	A matrix to be converted into long format.

Details

This function converts a matrix into a long format data frame. The resulting data frame contains four columns: row, column, value, and id. The 'id' column assigns a unique identifier to each column group, making it easier to identify and analyze the data by column groups.

Value

A data frame in long format with columns: row, column, value, and id.

Examples

```
# Create a matrix
mat <- matrix(data = 1:9,
              nrow = 3,
              ncol = 3,
              dimnames = list(c("A", "B", "C"),
                              c("X", "Y", "Z")))

long_format <- matrix2long(mat)
long_format

# Using correlation matrix
matrix2long(cor(mtcars))
```

Index