Package: findviews (via r-universe)

September 2, 2024

Type Package

Title A View Generator for Multidimensional Data

Version 0.1.4.9000

Author Thibault Sellam

Maintainer Thibault Sellam <thibault.sellam@gmail.com>

Description A tool to explore wide data sets, by detecting, ranking and plotting groups of statistically dependent columns.

License MIT + file LICENSE

LazyData TRUE

Imports shiny, ggplot2 (>= 2.0.0), scales, grDevices, gridExtra, stats, grid

Suggests testthat

RoxygenNote 5.0.1

URL https://github.com/tsellam/findviews

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

RemoteUrl https://github.com/tsellam/findviews

RemoteRef HEAD

RemoteSha 294a3fa018262646e6e09ce3329e8b9d3cbf2633

Contents

`findviews` *Views of a multidimensional dataset.*

Description

`findviews` detects and plots groups of mutually dependent columns. It is based on Shiny and ggplot.

Usage

```
findviews(data, view_size_max = NULL, clust_method = "complete", ...)
```

Arguments

`data`	Data frame or matrix to be processed
`view_size_max`	Maximum number of columns in the views. If set to `NULL`, findviews uses `log2(ncol(data))`, rounded upwards and capped at 5.
`clust_method`	Character describing a clustering method, used internally by `hclust`. Example values are "complete", "single" or "average".
`...`	Optional Shiny parameters, used in Shiny's `runApp` function.

Details

The function `findviews` takes a data frame or a matrix as input. It computes the pairwise dependency between the columns, detects clusters in the resulting structure and displays the results with a Shiny app.

`findviews` processes numerical and categorical data separately. It excludes the columns with only one value, the columns in which all the values are distinct (e.g., primary keys), and the columns with more than 75% missing values.

`findviews` computes the dependency between the columns differently depending on their type. It uses Pearson's coefficient of correlation for numerical data, and Cramer's V for categorical data.

To cluster the columns, `findviews` uses the function `hclust`, R's implementation of agglomerative hierarchical clustering. The parameter `clust_method` specifies which flavor of agglomerative clustering to use. The number of clusters is determined by the parameter `view_size_max`.

Examples

```
## Not run:
findviews(mtcars)
findviews(mtcars, view_size_max = 4,  port = 7000)

## End(Not run)
```

findviews_core *Views of a multidimensional dataset, non-Shiny version*

Description

findviews_core generates views of a multidimensional data set. It produces the same results as findviews, but does *not* present them with a Shiny app.

Usage

```
findviews_core(data, view_size_max = NULL, clust_method = "complete")
```

Arguments

data	Data frame or matrix to be processed
view_size_max	Maximum number of columns in the views. If set to NULL, findviews uses log2(ncol(data)), rounded upwards and capped at 5.
clust_method	Character describing a clustering method, used internally by hclust. Example values are "complete", "single" or "average".

Details

findviews_core takes a data frame or a matrix as input. It computes the pairwise dependency between the columns and detects clusters in the resulting structure. See the documentation of findviews for more details.

The difference between findviews and findviews_core is that the former presents its results with a Shiny app, while the latter simply outputs them as R stuctures.

Examples

```
findviews_core(mtcars)
findviews_core(mtcars, view_size_max = 4)
```

findviews_to_compare *Views of a multidimensional dataset, ranked by their differentiation power.*

Description

findviews_to_compare detects views on which two arbitrary sets of rows differ. It plots the results with ggplot and Shiny.

Usage

```
findviews_to_compare(group1, group2, data, view_size_max = NULL,
  clust_method = "complete", ...)
```

Arguments

group1	Logical vector of size nrow(data), which describes the first group to compare. The value TRUE at position i indicates the the i-th row of data belongs to the group.
group2	Logical vector, which describes the second group to compare. The value TRUE at position i indicates the the i-th row of data belongs to the group.
data	Data frame or matrix to be processed
view_size_max	Maximum number of columns in the views. If set to NULL, findviews uses log2(ncol(data)), rounded upwards and capped at 5.
clust_method	Character describing a clustering method, used internally by hclust. Example values are "complete", "single" or "average".
...	Optional Shiny parameters, used in Shiny's runApp function.

Details

The function findviews_to_compare takes two groups of rows as input and detects views on which the statistical distribution of those two groups differ.

To detect the set of views, findviews_to_compare eliminates the rows which are present in neither group and applies findviews.

To evaluate the differentiation power of the views, findviews computes the histograms of the two groups to be compared, and computes their dissimilarity them with the Euclidean distance.

This method is loosely based on the following paper:

```
Fast, Explainable View Detection to Characterize Exploration Queries
Thibault Sellam, Martin Kersten
SSDBM, 2016
```

Examples

```
## Not run:
findviews_to_compare(mtcars$mpg >= 20 , mtcars$mpg < 20 , mtcars)

## End(Not run)
```

`findviews_to_compare_core` *Views of a multidimensional dataset, ranked by their differentiation power, non-Shiny version*

Description

`findviews_to_compare_core` detects views on which two arbitrary sets of tuples are well separated. It produces the same results as `findviews_to_compare`, but does *not* present them with a Shiny app.

Usage

```
findviews_to_compare_core(group1, group2, data, view_size_max = NULL,
  clust_method = "complete")
```

Arguments

`group1`	Logical vector of size `nrow(data)`, which describes the first group to compare. The value `TRUE` at position i indicates the the i-th row of `data` belongs to the group.
`group2`	Logical vector, which describes the second group to compare. The value `TRUE` at position i indicates the the i-th row of `data` belongs to the group.
`data`	Data frame or matrix to be processed
`view_size_max`	Maximum number of columns in the views. If set to `NULL`, findviews uses `log2(ncol(data))`, rounded upwards and capped at 5.
`clust_method`	Character describing a clustering method, used internally by `hclust`. Example values are "complete", "single" or "average".

Details

The function `findviews_to_compare_core` takes two groups of tuples as input, and detects views on which the statistical distribution of those two groups is different. See the documentation of `findviews_to_compare` for more details.

The difference between `findviews_to_compare` and `findviews_to_compare_core` is that the former presents its results with a Shiny app, while the latter simply outputs them as R stuctures.

Examples

```
findviews_to_compare_core(mtcars$mpg >= 20 , mtcars$mpg < 20 , mtcars)
```

`findviews_to_predict` *Views of a multidimensional dataset, ranked by their prediction power.*

Description

`findviews_to_predict` detects groups of mutually dependent columns, ranks them by predictive power, and plots them with Shiny and ggplot.

Usage

```
findviews_to_predict(target, data, view_size_max = NULL,
  clust_method = "complete", ...)
```

Arguments

`target`	Name of the variable to be predicted.
`data`	Data frame or matrix to be processed
`view_size_max`	Maximum number of columns in the views. If set to `NULL`, findviews uses `log2(ncol(data))`, rounded upwards and capped at 5.
`clust_method`	Character describing a clustering method, used internally by `hclust`. Example values are "complete", "single" or "average".
`...`	Optional Shiny parameters, used in Shiny's `runApp` function.

Details

The function `findviews_to_predict` takes a data set and a target variable as input. It detects clusters of statistically dependent columns in the data set - e.g., views - and ranks those groups according to how well they predict the target variable.

To detect the views, `findviews_to_predict` relies on `findviews`. To evaluate their predictive power, it uses the *mutual information* between the joint distribution of the columns and that of the target variable. Internally, `findviews_to_predict` discretizes all the continuous variables with equi-width binning.

Note: `findviews_to_predict` removes the column to be predicted (the target column) from the dataset before it creates the column groups. Hence, the views it returns may be different from those return by calling by `findviews` directly on the dataset.

Examples

```
## Not run:
findviews_to_predict('mpg', mtcars)
findviews_to_predict('mpg', mtcars, view_size_max = 4)

## End(Not run)
```

`findviews_to_predict_core` *Views of a multidimensional dataset, ranked by their prediction power, non-Shiny version.*

Description

`findviews_to_predict_core` detects groups of mutually dependent columns, and ranks them by their predictive power. It produces the same results as `findviews_to_predict`, but does *not* present them with a Shiny app.

Usage

```
findviews_to_predict_core(target, data, view_size_max = NULL,
  clust_method = "complete")
```

Arguments

`target`	Name of the variable to be predicted.
`data`	Data frame or matrix to be processed
`view_size_max`	Maximum number of columns in the views. If set to `NULL`, findviews uses `log2(ncol(data))`, rounded upwards and capped at 5.
`clust_method`	Character describing a clustering method, used internally by `hclust`. Example values are "complete", "single" or "average".

Details

The function `findviews_to_predict_core` takes a data set and a target variable as input. It detects clusters of statistically dependent columns in the data set - e.g., views - and ranks those groups according to how well they predict the target variable. See the documentation of `findviews_to_predict` for more details.

The difference between `findviews_to_predict` and `findviews_to_predict_core` is that the former presents its results with a Shiny app, while the latter simply outputs them as R stuctures.

Examples

```
findviews_to_predict_core('mpg', mtcars)
findviews_to_predict_core('mpg', mtcars, view_size_max = 4)
```

Index