Package ‘DescriptiveStats.OBeu’

January 20, 2019

Type Package
Title Descriptive Statistics 'OpenBudgets.eu'
Version 1.3.1
Date 2019-01-21
Description Estimate and return the needed parameters for visualizations designed for 'OpenBudgets.eu' <http://openbudgets.eu/> datasets. Calculate descriptive statistical measures in budget data of municipalities across Europe, according to the 'OpenBudgets.eu' data model. There are functions for measuring central tendency and dispersion of amount variables along with their distributions and correlations and the frequencies of categorical variables for a given dataset. Also, can be used generally to extract visualization parameters, convert them to 'JSON' format and use them as input in a different graphical interface.

Maintainer Kleanthis Koupidis <koupidis@okfn.gr>
URL https://github.com/okgreece/DescriptiveStats.OBeu
BugReports https://github.com/okgreece/DescriptiveStats.OBeu/issues
License GPL-2 | file LICENSE
Encoding UTF-8
LazyData true
Imports dplyr, graphics, grDevices, jsonlite, magrittr, RCurl, reshape, stats
RoxygenNote 6.1.1
Suggests curl, knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Kleanthis Koupidis [aut, cre],
Aikaterini Chatzopoulou [aut],
Charalampos Bratsas [aut]
Repository CRAN
Date/Publication 2019-01-20 22:50:03 UTC
**R topics documented:**

- `compare.stats` ................................................. 2
- `CV` .................................................................. 3
- `ds.analysis` ....................................................... 3
- `ds.box` .............................................................. 5
- `ds.boxplot` .......................................................... 6
- `ds.correlation` .................................................... 7
- `ds.frequency` ....................................................... 8
- `ds.hist` .............................................................. 9
- `ds.kurtosis` ......................................................... 10
- `ds.skewness` ....................................................... 11
- `ds.statistics` ...................................................... 12
- `multisub` ........................................................... 13
- `nums` ................................................................ 14
- `open_spending.ds` ................................................ 15
- `sample_json_link_openspending` .............................. 16
- `Wuppertal_df` ..................................................... 17
- `Wuppertal_openspending` ....................................... 17

**Index**

<table>
<thead>
<tr>
<th>compare.stats</th>
<th>Group and compare summaries statistics to a data frame</th>
</tr>
</thead>
</table>

**Description**

Extract and return a data frame with the columns that include only numeric values

**Usage**

```r
compare.stats(df, group_var, values, m_functions)
```

**Arguments**

- `df` numeric vector or matrix or dataframe
- `group_var` character vector of variables to group the data
- `values` numeric or integer variables
- `m_functions` functions to apply in values

**Value**

This function returns a data frame with the selected group_vars and the result of m_functions applied in the selected values.

**Author(s)**

Kleanthis Koupidis
CV

**Coefficient of variation**

**Description**

Extract and return a data frame with the columns that include only numeric values.

**Usage**

CV(x)

**Arguments**

x A numeric vector or matrix or dataframe

**Value**

This function returns a vector with the coefficient of variance for the input vector, matrix or dataframe.

**Author(s)**

Kleanthis Koupidis

---

d.s. analysis

**Calculation of some Descriptive Tasks**

**Description**

The function calculates the basic descriptive measures, the correlation and the boxplot parameters of all the numerical variables and the frequencies of all the nominal variables.

**Usage**

d.s. analysis(data, c.out = 1.5, box.width = 0.15, outliers = TRUE, hist.class = "Sturges", corr.method = "pearson", fr.select = NULL, tojson = FALSE)

**Arguments**

data The input data
c.out Determines the length of the "whiskers" plot. If it is equal to zero no outliers will be returned.
box.width The width level is determined 0.15 times the square root of the size of the input data.
outliers
If TRUE the outliers will be computed at the selected "c.out" level (default is 1.5 times the Interquartile Range).

hist.class
The method or the number of classes for the histogram.

corr.method
The correlation coefficient method to compute: "pearson" (default), "kendall" or "spearman".

fr.select
One or more nominal variables to calculate their corresponding frequencies.

tojson
If TRUE the results are returned in json format

Details
This function returns a list with the basic statistics, the parameters needed to visualize a boxplot and a histogram, it also provides the frequencies of non numerical data of the input dataset and the correlation coefficient. The input of this function can be a matrix or data frame.

Value
A list or json file with the following components:

- descriptives The descriptive measures
- boxplot The statistics of the boxplot
- histogram The histogram parameters
- frequencies The frequencies and the relative frequencies of factors/characters of the input dataset
- correlation The correlation coefficient

Author(s)
Kleanthis Koupidis, Charalampos Bratsas

See Also
open_spendingNds

Examples

# iris data frame as input with the default parameters
ds.analysis(iris)

# using iris data frame with different parameters
ds.analysis(iris, c.out = 1, box.width = 0.20, outliers = TRUE, tojson = TRUE)

# using iris data frame with different parameters
# fr.select parameter specified as Species
# OpenBudgets.eu Dataset Example:
ds.analysis(Wuppertal_df, c.out = 2, box.width = 0.15, outliers = FALSE, tojson = FALSE)
ds.box  

Boxplot Parameters of a numeric vector

Description
This function calculates the statistical measures needed to visualize the boxplot of a numeric vector.

Usage

ds.box(x, c = 1.5, c.width = 0.15, out = TRUE, toJSON = FALSE)

Arguments

x  The input numeric vector

c  Determines the length of the "whiskers" plot. If it is equal to zero or out=F, no outliers will be returned.

c.width  The width level is determined 0.15 times the square root of the size of the input vector

out  If TRUE the outliers will be computed at the selected "c" level (default is 1.5 times the Interquartile Range).

tojson  If TRUE the results are returned in json format

Details
This function returns a list with the parameters needed to visualize a boxplot.

Value
Returns a list or a json file with the following components:

• lo.whisker The extreme of the lower whisker
• lo.hinge The lower "hinge"
• median The median
• up.hinge The upper "hinge"
• up.whisker The extreme of the upper whisker
• box.width The width of the box (default is 0.15 times the square root of the size of the vector)
• lo.out The values of any data points which lie below the extreme of the lower whisker
• up.out The values of any data points which lie above the extreme of the upper whisker
• n The non-NA observations of the vector

Author(s)
Kleanthis Koupidis, Charalampos Bratsas
See Also

ds_analysis, open_spending.ds

Examples

# with vector as an input and the default parameters
vec <- as.vector(iris$Sepal.Width)
d.s.box(vec)

# with vector as an input and the different parameters
vec <- as.vector(iris$Sepal.Width)
d.s.box(vec, c = 3, c.width = 0.20, out = FALSE, tojson = FALSE)

# OpenBudgets.eu Dataset Example:
amounts <- as.vector(Wuppertal_df$Amount)
d.s.box(amounts, c = 1.5, c.width = 0.20, out = TRUE)

d.s.boxplot

Boxplot Parameters of a matrix or data frame

Description

This function calculates the statistics of the boxplot for the input matrix or data frame.

Usage

d.s.boxplot(data, out.level = 1.5, width = 0.15, out1 = TRUE, tojson = FALSE)

Arguments

data The input numeric matrix or data frame.
out.level Determines the length of the "whiskers" plot. If it is equal to zero or "outl" is set to F, no outliers will be returned.
width The width level is determined 0.15 times the square root of the size of the input data.
outl If TRUE the outliers will be computed at the selected "out.level" level (default is 1.5 times the Interquartile Range).
tojson If TRUE the results are returned in json format

Details

This function returns as a list object the statistical parameters needed to visualize boxplot.

Value

Returns a list with the extracted components of ds.box for each variable/column of the input data.
ds.correlation

Author(s)
Aikaterini Chatzopoulou, Kleanthis Koupidis

See Also
ds.box, ds.analysis, open_spending.ds

Examples

# with matrix as an input and the default parameters
Matrix <- cbind(Uni05 = rnorm(200)/21, Norm = rnorm(200),
    '5T' = rt(200, df = 5), Gam2 = rgamma(200, shape = 2))
d.s.boxplot(Matrix, out.level = 1.5, width = 0.15, outl = TRUE, tojson = FALSE)

# iris data frame as an input, different parameters and json output
d.s.boxplot(iris, out.level = 2, width = 0.25, outl = FALSE, tojson = TRUE)

# OpenBudgets.eu Dataset Example:
d.s.boxplot(Wuppertal_df$Amount, out.level = 2.5, width = 0.15,
    outl = TRUE, tojson = FALSE)

d.s.correlation

Correlation Coefficient of a dataframe

Description
This function calculates the correlation coefficient of the input vectors, matrix or data frame. By default, the correlation coefficient of pearson is computed.

Usage
d.s.correlation(x, y = NULL, cor.method = "pearson", tojson = FALSE)

Arguments

x          A numeric vector, matrix or data frame
y          A vector, matrix or data frame with same dimension as x. By default it is equal with NULL.
cor.method The correlation coefficient method to compute: "pearson" (default), "kendall" or "spearman".
tojson     If TRUE the results are returned in json format, default returns a data frame

Details
This function returns an upper triangle matrix with the correlation coefficients of the input data. The correlation coefficient of pearson is computed, by default. Other options are "kendall" or "spearman".
ds.frequency

Author(s)
Aikaterini Chatzopoulou, Kleanthis Koupidis, Charalampos Bratsas

See Also
ds.analysis, open_spending.ds

Examples

# iris data frame as an input and the default parameters
ds.correlation(iris, cor.method = "pearson", tojson = FALSE)

# with matrix as an input, different parameters and json output
Matrix <- cbind(UNI = (1:200)/21, Norm = rnorm(200),
   '5T' = rt(200, df = 5), Gam2 = rgamma(200, shape = 2))
ds.correlation(Matrix, cor.method = "kendall", tojson = TRUE)

---

### Description

This function calculates the frequencies and the relative frequencies of factors/characters of the input dataset.

### Usage

ds.frequency(data, select = NULL, tojson = FALSE)

### Arguments

- **data**: A vector, matrix or data frame which includes at least one factor/character.
- **select**: Select one or more specific nominal variables to calculate their corresponding frequencies, if it’s not specified the result corresponds to frequencies of every factor variable in the data.
- **tojson**: If TRUE the results are returned in json format, default returns a list

### Details

This function returns a list with the frequencies and relative frequencies of factors/characters of the input dataset.

### Author(s)

Kleanthis Koupidis, Charalampos Bratsas
ds.hist

See Also
ds.analysis, open_spending.ds

Examples

# iris data frame as an input and a selected column to calculate its frequencies
ds.frequency(iris, select = "Species", tojson = FALSE)

# iris data frame as an input without a selected column and json output
ds.frequency(iris, tojson = TRUE)

# OpenBudgets.eu Dataset Example:
ds.frequency(Wuppertal_df, select = "Produkt", tojson = FALSE)

---

ds.hist  

Histogram breaks and frequencies

Description

This function computes the histogram parameters of the numeric input vector. The default for
breaks is the value resulted from Sturges algorithm.

Usage

ds.hist(x, breaks = "Sturges", tojson = FALSE)

Arguments

x  
The input numeric vector, matrix or data frame
breaks  
The method or the number of classes for the histogram
tojson  
If TRUE the results are returned in json format, default returns a list

Details

The possible values for breaks are Sturges see nclass.Sturges, Scott see nclass.scott and FD
or Freedman Diaconis nclass.FD which are in package grDevices.

Value

A list or json file with the following components:

- cuts The boundaries of the histogram classes
- density The density of each histogram class
- normal.curve.x Abscissa of the normal curve
- normal.curve.y Ordinate of the normal curve
ds.kurtosis

• fit.line.x Abscissa of the data density curve
• fit.line.y Ordinate of the data density curve
• mean The average value of the input vector
• median The median value of the input data

Author(s)
Kleanthis Koupidis, Charalampos Bratsas

See Also
ds.analysis, open_spending.ds

Examples

# with a vector as an input and the defaults parameters
vec <- as.vector(iris$Sepal.Width)
d.s.hist(vec)

# OpenBudgets.eu Dataset Example:
d.s.hist(Wuppertal_df$Amount, tojson = TRUE)

d.s.kurtosis Calculation of Kurtosis

Description
This function calculates kurtosis of the input vector, matrix or data frame.

Usage
d.s.kurtosis(x, tojson = FALSE)

Arguments
x A numeric vector, matrix or data frame.
tojson If TRUE the results are returned in json format

Details
This function returns the kurtosis, based on a scaled version of the fourth moment, of numbers of the input data.

Author(s)
Aikaterini Chatzopoulou, Charalampos Bratsas
ds.skewness

See Also
dskewness, ds.statistics, ds.analysis, open_spending.ds

Examples

# with a matrix as an input
Matrix <- cbind(Uni05 = (1:200)/21, Norm = rnorm(200),
               `5T` = rt(200, df = 5), Gam2 = rgamma(200, shape = 2))
dskurtosis(Matrix, tojson = FALSE)

# with iris data frame as an input
dskurtosis(iris, tojson = FALSE)

# with a vector as an input and json output
vec <- as.vector(iris$Sepal.Width)
dskurtosis(vec, tojson = TRUE)

# OpenBudgets.eu Dataset Example:
dskurtosis(Wuppertal_df, tojson = FALSE)

----------------------------------------
ds.skewness       Calculation of Skewness

Description

This function calculates skewness of the input vector, matrix or data frame.

Usage

dskewness(x, tojson = FALSE)

Arguments

x          A numeric vector, matrix or data frame.
tojson     If TRUE the results are returned in json format

Details

This function returns the skewness, also known as Pearson’s moment coefficient of skewness, of numbers of the input data.

Author(s)

Aikaterini Chatzopoulou

See Also
dskurtosis, ds.statistics, ds.analysis, open_spending.ds
Examples

# with a matrix as an input
Matrix <- cbind(Uni05 = (1:200)/21, Norm = rnorm(200),
               '5T' = rt(200, df = 5), Gam2 = rgamma(200, shape = 2))
ds.skewness(Matrix, tojson = FALSE)

# with iris data frame as an input
ds.skewness(iris, tojson = FALSE)

# with a vector as an input and json output
vec <- as.vector(iris$Sepal.Width)
ds.skewness(vec, tojson = TRUE)

# OpenBudgets.eu Dataset Example:
ds.skewness(Wuppertal_df, tojson = FALSE)

---

d.s.statistics

Calculation of the Statistic Measures

Description

This function calculates the basic descriptive measures of the input dataset.

Usage

ds.statistics(data, tojson = FALSE)

Arguments

data A numeric vector, matrix or data frame
tojson If TRUE the results are returned in json format, default returns a list

Details

This function returns the following values of the input data: minimum, maximum, range, mean, median, first and third quantiles, variance, standart deviation, skewness and kurtosis.

Value

A list or json file with the following components:

- Min The minimum observed value of the input data
- Max The maximum observed value of the input data
- Range The range, defined as the difference of the maximum and the minimum value.
- Mean The average value of the input data
- Median The median value of the input data
multisub

- Quantiles The 25% and 75% percentiles
- Variance The variance of the input data
- Standard Deviation The standard deviation of the input data
- Skewness The Skewness of the input data
- Kurtosis The Kurtosis of the input data

Author(s)
Aikaterini Chatzopoulou, Kleanthis Koupidis, Charalampos Bratsas

See Also
open_spending.ds

Examples

# with matrix as an input and json outpout
Matrix <- cbind(Uni05 = (1:200)/21, Norm = rnorm(200), 
`'5T'` = rt(200, df = 5), Gam2 = rgamma(200, shape = 2))
ds.statistics(Matrix, tojson = TRUE)

# with vector as an input
vec <- as.vector(iris$Sepal.Width)
ds.statistics(vec, tojson = FALSE)

# with iris data frame as an input
ds.statistics(iris, tojson = FALSE)

# OpenBudgets.eu Dataset Example:
ds.statistics(Wuppertal_df$Amount, tojson = TRUE)

Description

Extract and return a data frame with the columns that include only numeric values

Usage

multisub(pattern, replacement, x, ...)

Multiple replacement
nums

Arguments

- **pattern**: Character string vector containing a regular expression to be matched in the given character vector.
- **replacement**: A character vector of equal length with the pattern to be replaced.
- **x**: A character vector or an object where the matches are
- **...**: other parameters to pass

Value

This function returns a character vector with the replacements.

Author(s)

Kleanthis Koupidis

---

**nums**  
*Select the numeric columns of a given dataset*

Description

Extract and return a data frame with the columns that include only numeric values.

Usage

nums(data)

Arguments

- **data**: A numeric vector, matrix or data frame.

Value

This function returns a data frame with the numeric columns of the input dataset.

Author(s)

Kleanthis Koupidis
Examples

# with data frame as input
nums(iris)

# with vector as input
vec <- as.vector(iris$Sepal.Width)
nums(vec)

# with matrix as input
Matrix <- cbind(Uni05 = (1:200)/21, Norm = rnorm(200),
    '5T' = rt(200, df = 5), Gam2 = rgamma(200, shape = 2))
nums(Matrix)

# OpenBudgets.eu Dataset Example:
head(nums(Wuppertal_df))

---

open_spending.ds

Read and Calculate the Basic Information for Basic Descriptive Tasks from Open Spending and Rudolf APIs.

---

Description

Extract and analyze the input data provided from Open Spending API of OpenBudgets.eu, using the ds.analysis function.

Usage

open_spending.ds(json_data, dimensions = NULL, amounts = NULL, measured.dimensions = NULL, coef.outl = 1.5, box.outliers = TRUE, box.width = 0.15, cor.method = "pearson", freq.select = NULL)

Arguments

json_data The json string, URL or file from Open Spending API
dimensions The dimensions of the input data
amounts The measures of the input data
measured.dimensions The dimensions to which correspond amount/numeric variables
coef.outl Determines the length of the "whiskers" plot. If it is equal to zero no outliers will be returned.
box.outliers If TRUE the outliers will be computed at the selected "coef.outl" level (default is 1.5 times the Interquartile Range).
box.width The width level is determined 0.15 times the square root of the size of the input data.
cor.method The correlation coefficient method to compute: "pearson" (default), "kendall" or "spearman".
freq.select One or more nominal variables to calculate their corresponding frequencies.
Details

This function is used to read data in json format from Open Spending and Rudolf APIs., in order to implement some basic descriptive tasks through ds.analysis function.

Value

A json string with the resulted parameters of the ds.analysis function.

Author(s)

Kleanthis Koupidis

See Also

ds.analysis

Examples

# OpenBudgets.eu Dataset Example:
# open_spending.ds(json_data = Wuppertal_openspending,
#     dimensions = "functional_classification_3.Produktgruppe|date_2.Year",
#     amounts = "Amount")

---

sample_json_link_openspending

Sample data from Open Spending

Description

Sample data of Revised Budget phase amounts

- The year (2016) of the recorded approved budget phase amounts
- The revised budget phase amounts of 2016
- The original amounts of this year
- The functional classification description
- The functional classification code

Format

A link with the json format data

Source

http://next.openspending.org/
Wuppertal_df

Wuppertal Fiscal Data extracted from Open Spending API

Description

This dataset contains the budget of Wuppertal for 2009 to 2020

- The product ID
- The account type
- The kind
- The year these amounts were measured
- The amount
- The product area ID
- The product group ID
- The product
- The product area
- The product group

Format

A data frame with the previous characteristics as columns

Source

http://next.openspending.org/api/3/cubes/4b6d969e07ef7a86aa54e539fc127a14:wuppertalhaushalt/facts

Wuppertal_openspending

Wuppertal Fiscal Data extracted from Open Spending API

Description

This dataset contains the budget of Wuppertal for 2009 to 2020

- The product ID
- The account type
- The kind
- The year these amounts were measured
- The amount
- The product area ID
- The product group ID
- The product
- The product area
- The product group
Format

A link with the json format data

Source

http://next.openspending.org/api/3/cubes/4b6d969e07ef7a86aa54e539fc127a14:wuppertalhaushalt/facts
Index

compare.stats, 2
CV, 3

ds.analysis, 3, 6–11, 16
ds.box, 5, 6, 7
ds.boxplot, 6
ds.correlation, 7
ds.frequency, 8
ds.hist, 9
ds.kurtosis, 10, 11
ds.skewness, 11, 11
ds.statistics, 11, 12

multisub, 13

nclass.FD, 9
nclass.scott, 9
nclass.Sturges, 9
nums, 14

open_spending.ds, 4, 6–11, 13, 15

sample_json_link_openspending, 16

Wuppertal_df, 17
Wuppertal_openspending, 17