Package ‘SmartEDA’

April 25, 2020

Type Package

Title Summarize and Explore the Data

Version 0.3.5

Maintainer Dayanand Ubrangala <daya6489@gmail.com>

Depends R (>= 3.3.0)

Imports ggplot2(>= 2.2.1), sampling(>= 2.8), scales(>= 0.5.0), markdown(>= 1.9), ISLR(>= 1.2), data.table(>= 1.10.4-3), gridExtra(>= 2.3), GGally(>= 1.4.0)

Description Exploratory analysis on any input data describing the structure and the relationships present in the data. The package automatically select the variable and does related descriptive statistics. Analyzing information value, weight of evidence, custom tables, summary statistics, graphical techniques will be performed for both numeric and categorical predictors.

License MIT + file LICENSE

Suggests psych, Hmisc, smbinning, testthat, knitr, covr

Encoding UTF-8

LazyData true

URL https://daya6489.github.io/SmartEDA/

BugReports https://github.com/daya6489/SmartEDA/issues

Repository CRAN

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Dayanand Ubrangala [aut, cre],
        Kiran R [aut, ctb],
        Ravi Prasad Kondapalli [aut, ctb],
        Sayan Putatunda [aut, ctb]

Date/Publication 2020-04-25 06:20:02 UTC
ExpCatStat

**R topics documented:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpCatStat</td>
<td>2</td>
</tr>
<tr>
<td>ExpCatViz</td>
<td>4</td>
</tr>
<tr>
<td>ExpCTable</td>
<td>5</td>
</tr>
<tr>
<td>ExpCustomStat</td>
<td>6</td>
</tr>
<tr>
<td>ExpData</td>
<td>8</td>
</tr>
<tr>
<td>ExpInfoValue</td>
<td>9</td>
</tr>
<tr>
<td>ExpKurtosis</td>
<td>10</td>
</tr>
<tr>
<td>ExpNumStat</td>
<td>11</td>
</tr>
<tr>
<td>ExpNumViz</td>
<td>13</td>
</tr>
<tr>
<td>ExpOutliers</td>
<td>14</td>
</tr>
<tr>
<td>ExpOutQQ</td>
<td>16</td>
</tr>
<tr>
<td>ExpParcoord</td>
<td>16</td>
</tr>
<tr>
<td>ExpReport</td>
<td>18</td>
</tr>
<tr>
<td>ExpSkew</td>
<td>19</td>
</tr>
<tr>
<td>ExpStat</td>
<td>20</td>
</tr>
<tr>
<td>ExpWoeTable</td>
<td>21</td>
</tr>
</tbody>
</table>

**Index**

ExpCatStat

*Function provides summary statistics for all character or categorical columns in the dataframe*

**Description**

This function combines results from weight of evidence, information value and summary statistics.

**Usage**

ExpCatStat(data, Target = NULL, result = "Stat", clim = 10,
            nlim = 10, bins = 10, Pclass = NULL, plot = FALSE, top = 20,
            Round = 2)

**Arguments**

- **data**: dataframe or matrix
- **Target**: target variable
- **result**: "Stat" - summary statistics, "IV" - information value
- **clim**: maximum unique levels for categorical variable. Variables will be dropped if unique levels is higher than clim for class factor/character variable
- **nlim**: maximum unique values for numeric variable.
- **bins**: number of bins (default is 10)
- **Pclass**: reference category of target variable
- **plot**: Information value barplot (default FALSE)
- **top**: for plotting top information values (default value is 20)
- **Round**: round of value
Details

Criteria used for categorical variable predictive power classification are

- If information value is < 0.03 then predictive power = "Not Predictive"
- If information value is 0.3 to 0.1 then predictive power = "Somewhat Predictive"
- If information value is 0.1 to 0.3 then predictive power = "Medium Predictive"
- If information value is >0.3 then predictive power = "Highly Predictive"

Value

This function provides summary statistics for categorical variable

- Stat - Summary statistics includes Chi square test scores, p value, Information values, Cramers V and Degree if association
- IV - Weight of evidence and Information values

Columns description:

- Variable variable name
- Target - Target variable
- class - name of bin (variable value otherwise)
- out0 - number of good observations
- out1 - number of bad observations
- Total - Total values for each category
- pct1 - good observations / total good observations
- pct0 - bad observations / total bad observations
- odds - Odds ratio \((a/b)/(c/d)\)
- woe - Weight of Evidence – calculated as \(\ln(\text{odds})\)
- iv - Information Value - \(\ln(\text{odds}) \times (\text{pct0} - \text{pct1})\)

Author(s)

dubrangala

Examples

```r
# Example 1
## Read mtcars data
# Target variable "am" - Transmission (0 = automatic, 1 = manual)
# Summary statistics
ExpCatStat(mtcars,Target="am",result = "Stat",clim=10,nlim=10,bins=10, Pclass=1,plot=FALSE,top=20,Round=2)
## Information value plot
ExpCatStat(mtcars,Target="am",result = "Stat",clim=10,nlim=10,bins=10, Pclass=1,plot=TRUE,top=20,Round=2)
## Information value for categorical Independent variables
ExpCatStat(mtcars,Target="am",result = "IV",clim=10,nlim=10,bins=10, Pclass=1,plot=FALSE,top=20,Round=2)
```
ExpCatViz  

Distributions of categorical variables

Description

This function automatically scans through each variable and creates bar plot for categorical variable.

Usage

ExpCatViz(data, target = NULL, fname = NULL, clim = 10, col = NULL, margin = 1, Page = NULL, Flip = F, sample = NULL, rdata = FALSE, value = NULL, gtitle = NULL, theme = "Default")

Arguments

data  dataframe or matrix

target  target variable. This is not a mandatory field

fname  output file name. Output will be generated in PDF format

clim  maximum categories to be considered to include in bar graphs

col  define the colors to fill the bars, default it will take sample colours

margin  index, 1 for row based proportions and 2 for column based proportions

Page  output pattern. if Page=c(3,2), It will generate 6 plots with 3 rows and 2 columns

Flip  default vertical bars. It will be used to flip the axis vertical to horizontal

sample  random selection of categorical variable

rdata  to plot bar graph for frequency/aggregated table

value  value column name. This is mandatory if `rdata` is TRUE

gtitle  graph title

theme  adding extra themes, geoms, and scales for `ggplot2` (eg: themes options from ggthemes package)

Value

This function returns collated graphs in grid format in PDF or JPEG format. All the files will be stored in the working directory

- Bar graph for raw data(this function will dynamically pick all the categorical variable and plot the bar chart)
- Bar graph for aggregated data
- Bar graph is a Stacked Bar graph by target variable

See Also

geom_bar
### Examples

```r
## Bar graph for specified variable
mtdata = mtcars
mtdata$carname = rownames(mtcars)
ExpCatViz(data=mtdata,target="carname",col="blue",rdata=TRUE,value="mpg")
n=nrow(mtdata)
ExpCatViz(data=mtdata,target="carname",col=rainbow(n),rdata=TRUE,value="mpg") # Rainbow colour

# Stacked bar chart
ExpCatViz(data=mtdata,target = "gear",col=hcl.colors(3, "Set 2"))
ExpCatViz(data=mtdata,target = "gear",col=c("red", "green", "blue"))

# Bar chart
ExpCatViz(data=mtdata)
ExpCatViz(data=mtdata,col="blue",gtitle = "Barplot")
```

---

### ExpCTable

**Function to create frequency and custom tables**

**Description**

This function will automatically select categorical variables and generate frequency or cross tables based on the user inputs. Output includes counts, percentages, row total and column total.

**Usage**

```
ExpCTable(data, Target = NULL, margin = 1, clim = 10, nlim = 10,
          round = 2, bin = 3, per = FALSE)
```

**Arguments**

- `data` : dataframe or matrix
- `Target` : target variable (dependent variable) if any. Default NULL
- `margin` : margin of index, 1 for row based proportions and 2 for column based proportions
- `clim` : maximum categories to be considered for frequency/custom table. Variables will be dropped if unique levels are higher than 'clim' for class factor/character variable. Default value is 10.
- `nlim` : numeric variable unique limits. Default 'nlim' values is 3, table excludes the numeric variables which is having greater than 'nlim' unique values
- `round` : round off
- `bin` : number of cuts for continuous target variable
- `per` : percentage values. Default table will give counts.

**Details**

This function provides both frequency and custom tables for all categorical features. And output will be generated in data frame.
Value

Frequency tables, Cross tables

Columns description for frequency tables:

- Variable is Variable name
- Valid is Variable values
- Frequency is Frequency
- Percent is Relative frequency
- CumPercent is Cumulative sum of relative frequency

Columns description for custom tables:

- Variable is Variable name
- Category is Variable values
- Count is Number of counts
- Per is Percentages
- Total is Total count

Examples

```r
# Frequency table
ExpCTable(mtcars, Target = NULL, margin = 1, clim = 10, nlim = 3, bin = NULL, per = FALSE)
# Crosstable for Mtcars data
ExpCTable(mtcars, Target = "gear", margin = 1, clim = 10, nlim = 3, bin = NULL, per = FALSE)
```

ExpCustomStat

**Customized summary statistics**

Description

Table of descriptive statistics. Output returns matrix object containing descriptive information on all input variables for each level or combination of levels in categorical/group variable. Also while running the analysis user can filter out the data by individual variable level or across data level.

Usage

```r
ExpCustomStat(data, Cvar = NULL, Nvar = NULL, stat = NULL, gpby = TRUE, filt = NULL, dcast = FALSE, value = NULL)
```
**ExpCustomStat**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>data</code></td>
<td>dataframe or Matrix</td>
</tr>
<tr>
<td><code>Cvar</code></td>
<td>qualitative variables on which to stratify / subgroup or run categorical summaries</td>
</tr>
<tr>
<td><code>Nvar</code></td>
<td>quantitative variables on which to run summary statistics for.</td>
</tr>
<tr>
<td><code>stat</code></td>
<td>descriptive statistics. Specify which summary statistics required (Included all base stat functions like 'mean', 'median', 'max', 'min', 'sum', 'IQR', 'sd', 'var', quantile like P0.1, P0.2 etc'). Also added two more stats here are 'PS' is percentage of shares and 'Prop' is column percentage</td>
</tr>
<tr>
<td><code>gpby</code></td>
<td>default value is True. Group level summary will be created based on list of categorical variable. If summary required at each categorical variable level then keep this option as FALSE</td>
</tr>
<tr>
<td><code>filt</code></td>
<td>filter out data while running the summary statistics. Filter can apply across data or individual variable level using filt option. If there are multiple filters, separate the conditions by using '^'. Ex: <code>Nvar = c(&quot;X1&quot;,&quot;X2&quot;,&quot;X3&quot;,&quot;X4&quot;)</code>, let say we need to exclude data <code>X1&gt;900</code> for <code>X1</code> variable, <code>X2==10</code> for <code>X2</code> variable, <code>Gender !='Male'</code> for <code>X3</code> variable and all data for <code>X4</code> then filt should be, filt = <code>c(&quot;X1&gt;900&quot;^^&quot;X2==10&quot;^^&quot;Gender!=Male&quot;^^all) or c(&quot;X1&gt;900&quot;^^&quot;X2==10&quot;^^Gender!=Male&quot;^^ ^)</code>. In case if you want to keep all data for some of the variable listed in Nvar, then specify inside the filt like <code>^all</code> or <code>^(single space)</code></td>
</tr>
<tr>
<td><code>dcast</code></td>
<td>fast dcast from data.table</td>
</tr>
<tr>
<td><code>value</code></td>
<td>If dcast is TRUE, pass the variable name which needs to come on column</td>
</tr>
</tbody>
</table>

**Details**

Filter unique value from all the numeric variables

Case1: Excluding unique values or outliers values like '999' or '9999' or '888' etc from each selected variables.

Eg: `dat = data.frame(x = c(23,24,34,999,12,12,23,999,45), y = c(1,3,4,999,0,999,0,8,999,0))`  
Exclude 999:  
x = c(23,24,34,12,12,23,45)  
y = c(1,3,4,0,0,8,0)

Case2: Summarise the data with selected descriptive statistics like 'mean' and 'median' or 'sum' and 'variance' etc.

Case3: Aggregate the data with different statistics using group by statement

Case4: Reshape the summary statistics.. etc

The complete functionality of 'ExpCustomStat' function is detailed in vignette help page with example code.

**Value**

summary statistics as dataframe. Usage of this function is detailed in user guide vignettes document.
Examples

## Selected summary statistics 'Count,sum, percentage of shares' for disp and mpg variables by vs, am and gear
ExpCustomStat(mtcars, Cvar=c("vs","am","gear"), Nvar = c("disp","mpg"), stat = c("Count","sum","PS"), gpby = TRUE, filt = NULL)

ExpCustomStat(mtcars, Cvar=c("gear"), Nvar = c("disp","mpg"),
stat = c("Count","sum","var"), gpby = TRUE, filt = "am==1")

ExpCustomStat(mtcars, Cvar = c("gear"), Nvar = c("disp","mpg"),
stat = c("Count","sum","mean","median"), gpby = TRUE, filt = "am==1")

## Selected summary statistics 'Count and fivenum stat for disp and mpg variables by gear
ExpCustomStat(mtcars, Cvar = c("gear"), Nvar = c("disp", "mpg"),
stat = c("Count","min","p0.25","p0.75","max"), gpby = TRUE)

ExpData

Function to generate the overview of a data frame

Description

This function used to produce summaries of data structure and overview of the data frame.

Usage

ExpData(data, type = 1)

Arguments

data a data frame

type Type 1 is overview of the data; Type 2 is structure of the data

Details

This function provides overview and structure of the data frames.

- Type = 1, overview of the data (column names are "Descriptions", "Obs")
- Type = 2, structure of the data (column names are "S.no", "Variable Name", "Variable Type", "

Examples

# Overview of the data
ExpData(data=mtcars, type=1)

# Structure of the data
ExpData(data=mtcars, type=2)
ExpInfoValue

**Description**

Provides information value for each categorical variable (X) against target variable (Y)

**Usage**

ExpInfoValue(X, Y, valueOfGood = NULL)

**Arguments**

- **X**: Independent categorical variable.
- **Y**: Binary response variable, it can take values of either 1 or 0.
- **valueOfGood**: Value of Y that is used as reference category.

**Details**

Information value is one of the most useful technique to select important variables in a predictive model. It helps to rank variables on the basis of their importance. The IV is calculated using the following formula

\[ IV = ( \text{Percentage of Good event} - \text{Percentage of Bad event} ) \times \text{WOE}, \]

where WOE is weight of evidence

\[ \text{WOE} = \log( \text{Percentage of Good event} - \text{Percentage of Bad event} ) \]

Here is what the values of IV mean according to Siddiqi (2006)

- If information value is < 0.03 then predictive power = "Not Predictive"
- If information value is 0.03 to 0.1 then predictive power = "Somewhat Predictive"
- If information value is 0.1 to 0.3 then predictive power = "Medium Predictive"
- If information value is >0.3 then predictive power = "Highly Predictive"

**Value**

Information value (iv) and Predictive power class

- **information value**
- **predictive class**

**See Also**

IV
**Examples**

```r
X = mtcars$gear
Y = mtcars$am
ExpInfoValue(X,Y,valueOfGood = 1)
```

---

**ExpKurtosis**

*Measures of Shape - Kurtosis*

**Description**

Measures of shape to give a detailed evaluation of data. Explains the amount and direction of skew. Kurtosis explains how tall and sharp the central peak is. Skewness has no units: but a number, like a z score.

**Usage**

```r
ExpKurtosis(x, type)
```

**Arguments**

- `x`: A numeric object or data.frame
- `type`: A character which specifies the method of computation. Options are "moment" or "excess"

**Value**

ExpKurtosis returns Kurtosis values

**Author(s)**

dubrangala

**Examples**

```r
ExpKurtosis(mtcars$hp,type="excess")
ExpKurtosis(mtcars$carb,type="moment")
ExpKurtosis(mtcars,type="excess")
```
ExpNumStat

Summary statistics for numerical variables

Description

Function provides summary statistics for all numerical variable. This function automatically scans through each variable and select only numeric/integer variables. Also if we know the target variable, function will generate relationship between target variable and each independent variable.

Usage

ExpNumStat(data, by = "A", gp = NULL, Qnt = NULL, Nlim = 10, MesofShape = 2, Outlier = FALSE, round = 3, dcast = FALSE, val = NULL)

Arguments

data: dataframe or matrix
by: group by A (summary statistics by All), G (summary statistics by group), GA (summary statistics by group and Overall)
gp: target variable if any, default NULL
Qnt: default NULL. Specified quantiles is c(.25,0.75) will find 25th and 75th percentiles
Nlim: numeric variable limit (default value is 10 which means it will only consider those variable having more than 10 unique values and variable type is numeric/integer)
MesofShape: Measures of shapes (Skewness and kurtosis).
Outlier: Calculate the lower hinge, upper hinge and number of outliers
round: round off
dcast: fast dcast from data.table
val: Name of the column whose values will be filled to cast (see Details sections for list of column names)

Details

column descriptions
• Vname is Variable name
• Group is Target variable
• TN is Total sample (inculded NA observations)
• nNeg is Total negative observations
• nPos is Total positive observations
• nZero is Total zero observations
• NegInf is Negative infinite count
• PosInf is Positive infinite count
• NA_value is Not Applicable count
• Per_of_Missing is Percentage of missings
• Min is minimum value
• Max is maximum value
• Mean is average value
• Median is median value
• SD is Standard deviation
• CV is coefficient of variations (SD/mean)*100
• IQR is Inter quartile range
• Qnt is quantile values
• MesofShape is Skewness and Kurtosis
• Outlier is Number of outliers
• Cor is Correlation b/w target and independent variables

**Value**

summary statistics for numeric independent variables

Summary by:

• Only overall level
• Only group level
• Both overall and group level

**Author(s)**
dubrangala

**See Also**
describe.by

**Examples**

# Descriptive summary of numeric variables is Summary by Target variables
ExpNumStat(mtcars,by="G",gp="gear",Qnt=c(0.1,0.2),MesofShape=2,
Outlier=TRUE,round=3)

# Descriptive summary of numeric variables is Summary by Overall
ExpNumStat(mtcars,by="A",gp="gear",Qnt=c(0.1,0.2),MesofShape=2,
Outlier=TRUE,round=3)

# Descriptive summary of numeric variables is Summary by Overall and Group
ExpNumStat(mtcars,by="GA",gp="gear",Qnt=seq(0,1,.1),MesofShape=1,
Outlier=TRUE,round=2)

# Summary by specific statistics for all numeric variables
ExpNumStat(mtcars,by="GA",gp="gear",Qnt=c(0.1,0.2),MesofShape=2,
Outlier=FALSE,round=2,dcast = TRUE,val = "IQR")


ExpNumViz

Distributions of numeric variables

Description

This function automatically scans through each variable and creates density plot, scatter plot and box plot for continuous variable.

Usage

ExpNumViz(data, target = NULL, type = 1, nlim = 3, fname = NULL, col = NULL, Page = NULL, sample = NULL, scatter = FALSE, gtitle = NULL, theme = "Default")

Arguments

data       dataframe or matrix

target     target variable

type       1 (boxplot by category and overall), 2 (boxplot by category only), 3 (boxplot for overall)
nlim       numeric variable unique limit. Default nlim is 3, graph will exclude the numeric variable which is having less than 'nlim' unique value
.fname      output file name
.col        define the fill color for box plot. Number of color should be equal to number of categories in target variable
.Page       output pattern. if Page=c(3,2), It will generate 6 plots with 3 rows and 2 columns
.sample     random selection of plots
.scatter    option to run scatter plot between all the numerical variables (default scatter=FALSE)
gtitle     chart title
.theme      adding extra themes, geoms, and scales for 'ggplot2' (eg: themes options from ggthemes package)

Details

This function automatically scan each variables and generate a graph based on the user inputs. Graphical representation includes scatter plot, box plot and density plots.

- target is continuous then output is scatter plots
- target is categorical then output is box plot
- target is NULL then density plot for all numeric features
- scatter = TRUE generate multiple scatter plot between all the independent contionuos variables with or without group argument
Value
returns collated graphs in PDF or JPEG format

- Univariate plot density plot for all the numeric data with the value of shape of the distribution (Skewness & Kurtosis)
- Bivariate plot correlatin plot for all the numeric data
- Bivariate plot scatter plot between continuous dependent variable and Independent variables
- Box plot by overall sample
- Box plot by stratified sample

See Also
geom_boxplot ggrepthemes geom_density geom_point

Examples

## Generate Boxplot by category
ExpNumViz(iris,target = "Species", type = 2, nlim = 2, 
col = c("red", "green", "blue", "pink"), Page = NULL, sample = 2, scatter = FALSE, 
gtitle = "Box plot: ")
## Generate Density plot
ExpNumViz(iris, nlim = 2, 
col = NULL,Page = NULL, sample = 2, scatter = FALSE, 
gtitle = "Density plot: ")
## Generate Scatter plot by Dependent variable
ExpNumViz(iris,target = "Sepal.Length", type = 1, nlim = 2, 
col = "red", Page = NULL, sample = NULL, scatter = FALSE, 
gtitle = "Scatter plot: ", theme = "Default")
## Generate Scatter plot for all the numerical variables
ExpNumViz(iris,target = "Species", type = 1, nlim = 2, 
col = c("red", "green", "blue"), Page = NULL, sample = NULL, scatter = TRUE, 
gtitle = "Scatter plot: ", theme = "Default")

ExpOutliers

Univariate outlier analysis

Description
this function will run univariate outlier analysis based on boxplot or SD method. The function returns the summary of outlier for selected numeric features and adding new features if there is any outliers

Usage

ExpOutliers(data, varlist = NULL, method = "boxplot", 
treatment = NULL, capping = c(0.05, 0.95), outflag = FALSE)
ExpOutliers

Arguments

data  dataframe or matrix
varlist  list of numeric variable to perform the univariate outlier analysis
method  detect outlier method boxplot or NxStDev (where N is 1 or 2 or 3 std deviations, like 1xStDev or 2xStDev or 3xStDev)
treatment  treating outlier value by mean or median. default NULL
capping  default LL = 0.05 & UL = 0.95cap the outlier value by replacing those observations outside the lower limit with the value of 5th percentile and above the upper limit, with the value of 95th percentile value
outflag  add extreme value flag variable into output data

Details

this function provides both summary of the outlier variable and data

Univariate outlier analysis method

• boxplot is If a data value are below (Q1 minus 1.5x IQR) or boxplot lower whisker or above (Q3 plus 1.5x IQR) or boxplot upper whisker then those points are flagged as outlier value
• Standard Deviation is If a data distribution is approximately normal then about 68 percent of the data values lie within one standard deviation of the mean and about 95 percent are within two standard deviations, and about 99.7 percent lie within three standard deviations. If any data point that is more than 3 times the standard deviation, then those points are flagged as outlier value

Value

Outlier summary includes

• Num of outliers is Number of outlier in each variable
• Lower bound is Q1 minus 1.5x IQR for boxplot; Mean minus 3x StdDev for Standard Deviation method
• Upper bound is Q3 plus 1.5x IQR for boxplot; Mean plus 3x StdDev for Standard Deviation method
• Lower cap is Lower percentile capping value
• Upper cap is Upper percentile capping value

Examples

ExpOutliers(mtcars, varlist = c("mpg", "disp", "wt", "qsec"), method = 'BoxPlot',
capping = c(0.1, 0.9), outflag = TRUE)

ExpOutliers(mtcars, varlist = c("mpg", "disp", "wt", "qsec"), method = '2xStDev',
capping = c(0.1, 0.9), outflag = TRUE)

# Mean imputation or 5th percentile or 95th percentile value capping
ExpOutliers(mtcars, varlist = c("mpg", "disp", "wt", "qsec"), method = 'BoxPlot',
treatment = "mean", capping = c(0.05, 0.95), outflag = TRUE)
ExpOutQQ  
Quantile Quantile Plots

**Description**

This function automatically scans through each variable and creates normal QQ plot also adds a line to a normal quantile quantile plot.

**Usage**

```r
ExpOutQQ(data, nlim = 3, fname = NULL, Page = NULL, sample = NULL)
```

**Arguments**

- `data`: Input dataframe or data.table
- `nlim`: numeric variable limit
- `fname`: output file name. Output will be generated in PDF format
- `Page`: output pattern. if Page=c(3,2), It will generate 6 plots with 3 rows and 2 columns
- `sample`: random number of plots

**Value**

Normal quantile quantile plot

**See Also**

- `geom_qq`

**Examples**

```r
CData = ISLR::Carseats
ExpOutQQ(CData,nlim=10,fname=NULL,Page=c(3,2),sample=4)
```

---

ExpParcoord  
Parallel Co ordinate plots

**Description**

This function creates parallel Co ordinate plots

**Usage**

```r
ExpParcoord(data, Group = NULL, Stsize = NULL, Nvar = NULL, 
Cvar = NULL, scale = NULL)
```
**ExpParcoord**

**Arguments**

- **data**: Input dataframe or data.table
- **Group**: stratification variables
- **Stsize**: vector of startum sample sizes
- **Nvar**: vector of numeric variables, default it will consider all the numeric variable from data
- **Cvar**: vector of categorical variables, default it will consider all the categorical variable
- **scale**: scale the variables in the parallel coordinate plot (Default normalized with minimum of the variable is zero and maximum of the variable is one) (see ggparcoord details for more scale options)

**Details**

The Parallel Co ordinate plots having the functionalities of visualization for sample rows if data size large. Also data can be stratified basis of Target or group variables. It will normalize all numeric variables between 0 and 1 also having other standardization options. It will automatically make dummy (1,0) variables for categorical variables

**Value**

Parallel Co ordinate plots

**See Also**

- [ggparcoord](#)

**Examples**

```r
CData = ISLR::Carseats
# Defualt ExpParcoord function
ExpParcoord(CData,Group=NULL,Stsize=NULL,  
Nvar=c("Price","Income","Advertising","Population","Age","Education"))
# With Stratified rows and selected columns only
ExpParcoord(CData,Group="ShelveLoc",Stsize=c(10,15,20),  
Nvar=c("Price","Income"),Cvar=c("Urban","US"))
# Without stratification
ExpParcoord(CData,Group="ShelveLoc",Nvar=c("Price","Income"),  
Cvar=c("Urban","US"),scale=NULL)
# Scale changed std: univariately, subtract mean and divide by standard deviation
ExpParcoord(CData,Group="US",Nvar=c("Price","Income"),  
Cvar=c("ShelveLoc"),scale="std")
# Selected numeric variables
ExpParcoord(CData,Group="ShelveLoc",Stsize=c(10,15,20),  
Nvar=c("Price","Income","Advertising","Population","Age","Education"))
```
ExpReport

Function to create HTML EDA report

Description
Create a exploratory data analysis report in HTML format

Usage
ExpReport(data, Template = NULL, Target = NULL, label = NULL, theme = "Default", op_file = NULL, op_dir = getwd(), sc = NULL, sn = NULL, Rc = NULL)

Arguments
- data: a data frame
- Template: R markdown template (.rmd file)
- Target: dependent variable. If there is no defined target variable then keep as it is NULL.
- label: target variable descriptions, not a mandatory field
- theme: customized ggplot theme (default SmartEDA theme) (for Some extra themes use Package: ggthemes)
- op_file: output file name (.html)
- op_dir: output path
- sc: sample number of plots for categorical variable. User can decide how many number of plots to depict in html report.
- sn: sample number of plots for numerical variable. User can decide how many number of plots to depict in html report.
- Rc: reference category of target variable. If Target is categorical then Pclass value is mandatory and which should not be NULL

Details
The "ExpReport" function will generate a HTML report for any R data frames.

Note
If the markdown template is ready, you can use that template to generate the HTML report
ExpReport will generate three different types of HTML report based on the Target field

- IF Target = NULL, means there is no defined dependent varaible then it will genreate general EDA report at overall level
- IF Target = continuous, then it will genreate EDA report including univariate and multivarite summary statistics with correlation.
- IF Target = categorical, then it will genreate EDA report including univariate and multivarite summary statistics with chi square, Information values.
**ExpSkew**

**See Also**

`create_report`

**Examples**

```r
### Creating HTML report
### Not run:
library (ggthemes)
# Create report where target variable is categorical
ExpReport(mtcars, Target="gear", label="car", theme=theme_economist(), op_file="Samp1.html", Rc=3)
# Create report where target variable is continuous
ExpReport(mtcars, Target="wt", label="car", theme="Default", op_file="Samp2.html")
# Create report where no target variable defined
ExpReport(mtcars, Target=NULL, label="car", theme=theme_foundation(), op_file="Samp3.html")

### End(Not run)
```

---

**ExpSkew**  
**Measures of Shape - Skewness**

**Description**

Measures of shape to give a detailed evaluation of data. Explains the amount and direction of skew. Kurotis explains how tall and sharp the central peak is. Skewness has no units: but a number, like a z score.

**Usage**

```r
ExpSkew(x, type)
```

**Arguments**

- `x` A numeric object or data.frame
- `type` a character which specifies the method of computation. Options are "moment" or "sample"

**Value**

`ExpSkew` returns Skewness values

**Author(s)**

dubrangala

**Examples**

```r
ExpSkew(mtcars, type="moment")
ExpSkew(mtcars, type="sample")
```
ExpStat

Function provides summary statistics for individual categorical predictors

Description

Provides bivariate summary statistics for all the categorical predictors against target variables. Output includes chi-square value, degrees of freedom, information value, p-value

Usage

ExpStat(X, Y, valueOfGood = NULL)

Arguments

X  Independent categorical variable.
Y  Binary response variable, it can take values of either 1 or 0.
valueOfGood  Value of Y that is used as reference category.

Details

Summary statistics included Pearson’s Chi-squared Test for Count Data, "chisq.test" which performs chi-squared contingency table tests and goodness-of-fit tests. If any NA value present in X or Y variable, which will be considered as NA as in category while computing the contingency table. Also added unique levels for each X categorical variables and degrees of freedom

Value

The function provides summary statistics like

- Unique number of levels
- Chi square statistics
- P value
- df Degrees of freedom
- IV Information value
- Predictive class

See Also

chisq.test

Examples

X = mtcars$carb
Y = mtcars$am
ExpStat(X,Y,valueOfGood = 1)
ExpWoeTable

Function provides summary statistics with weight of evidence

Description

Weight of evidence for categorical (X-independent) variable against Target variable (Y)

Usage

ExpWoeTable(X, Y, valueOfGood = NULL, print = FALSE, Round = 2)

Arguments

X  Independent categorical variable.
Y  Binary response variable, it can take values of either 1 or 0.
valueOfGood Value of Y that is used as reference category.
print  print results
Round  rounds the values

Details

The weight of evidence tells the predictive power of an independent variable in relation to the dependent variable

Value

Weight of evidence summary table

See Also

WOETable

Examples

X = mtcars$gear
Y = mtcars$am
WOE = ExpWoeTable(X,Y,valueOfGood = 1)
Index

chisq.test, 20
create_report, 19

describe.by, 12

ExpCatStat, 2
ExpCatViz, 4
ExpCTable, 5
ExpCustomStat, 6
ExpData, 8
ExpInfoValue, 9
ExpKurtosis, 10
ExpNumStat, 11
ExpNumViz, 13
ExpOutliers, 14
ExpOutQQ, 16
ExpParcoord, 16
ExpReport, 18
ExpSkew, 19
ExpStat, 20
ExpWoeTable, 21

geom_bar, 4
geom_boxplot, 14
geom_density, 14
geom_point, 14
geom_qq, 16
ggparcoord, 17
ggthemes, 14

IV, 9

WOETable, 21