Package ‘sjPlot’
March 27, 2014

Type Package

Encoding UTF-8

Title sjPlot - data visualization for statistics in social science

Version 1.3

Date 2014-03-27

Author Daniel Lüdecke <d.luedecke@uke.de>

Maintainer Daniel Luedecke <d.luedecke@uke.de>

Description Collection of several plotting / table output functions for visualizing data and utility functions.

License GPL-3

Imports ggplot2, MASS, car, foreign, lmtest, plyr, psych, reshape2,scales

NeedsCompilation no

Repository CRAN

Date/Publication 2014-03-27 10:23:04

R topics documented:

  sjPlot-package .................................................. 3
  etc ................................................................. 4
  sjc.cluster ....................................................... 5
  sjc.dend .......................................................... 7
  sjc.elbow ......................................................... 8
  sjc.grpdisc ....................................................... 9
  sjc.kgap .......................................................... 10
  sjc.qclus ........................................................ 11
  sji.convertToLabel .............................................. 16
  sji.convertToValue .............................................. 17
### R topics documented:

- `sji.getValueLabels` .................................................. 18
- `sji.getVariableLabels` .............................................. 19
- `sji.setValueLabels` .................................................. 20
- `sji.setVariableLabels` .............................................. 21
- `sji.SPSS` ............................................................. 22
- `sji.viewSPSS` ......................................................... 23
- `sjp.aov1` ............................................................. 25
- `sjp.chi2` .............................................................. 29
- `sjp.corr` .............................................................. 31
- `sjp.frq` ............................................................... 35
- `sjp.glm` ............................................................... 41
- `sjp.glm.ma` ........................................................... 46
- `sjp.glmm` ............................................................. 47
- `sjp.grpfrq` ............................................................ 51
- `sjp.likert` ............................................................ 58
- `sjp.lm` ................................................................. 63
- `sjp.lm.int` ............................................................. 67
- `sjp.lm.ma` ............................................................. 71
- `sjp.lm1` .............................................................. 72
- `sjp.lmm` .............................................................. 74
- `sjp.pca` ............................................................... 79
- `sjp.reglin` ............................................................ 83
- `sjp.scatter` .......................................................... 84
- `sjp.stackfrq` .......................................................... 88
- `sjp.vif` ............................................................... 93
- `sjp.xtab` .............................................................. 94
- `sjt.corr` ............................................................. 99
- `sjt.df` ............................................................... 102
- `sjt.frq` ............................................................... 105
- `sjt.glm` ............................................................. 109
- `sjt.itemanalysis` ..................................................... 113
- `sjt.lm` ............................................................... 117
- `sjt.pca` ............................................................. 121
- `sjt.stackfrq` .......................................................... 125
- `sjt.xtab` ............................................................. 129
- `sju.adjustPlotRange.y` ............................................. 133
- `sju.aov1.levene` .................................................... 134
- `sju.betaCoef` ......................................................... 135
- `sju.chi2.gof` ......................................................... 136
- `sju.cramer` .......................................................... 136
- `sju.cronbach` ......................................................... 137
- `sju.dicho` ............................................................ 138
- `sju.groupVar` ........................................................ 138
- `sju.groupVarLabels` ............................................... 140
- `sju.mwu` ............................................................. 141
- `sju.phi` .............................................................. 142
- `sju.recode` ........................................................... 143
- `sju.recodeTo` ........................................................ 144
Description

Collection of several plotting and table output functions for data visualization. Results of several statistical analyses (that are commonly used in social sciences) can be visualized using this package, including simple and cross tabulated frequencies, histograms, box plots, (generalized) linear models (forest plots), PCA, correlations, cluster analyses, scatter plots etc.

Furthermore, this package contains some tools that are useful when carrying out data analysis or interpreting data (especially intended for people coming from SPSS and who are new to R). This collection of tool functions contains, for instance, (SPSS) data set import, variable recoding and weighting, statistical tests, determination of cluster groups, interpretation of interaction terms in linear models etc.

What does this package do?

In short, the functions in this package mostly do two things:

1. compute basic or advanced statistical analyses
2. either plot the results as ggplot-diagram or print them as html-table

However, meanwhile the amount of functions has increased, hence you’ll also find some utility functions beside the plotting functions (see below).

How does this package help me?

Basically, this package either helps those users who...

- have difficulties using and/or understanding all possibilities that ggplot offers to create plots, simply by providing intuitive function parameters, which allow for manipulating the appearance of plots;
- don’t want to set up complex ggplot-object each time from the scratch;
- like quick inspections of (basic) statistics via (html-)tables that are shown in the GUI viewer pane or default browser; or
- want easily create beautiful table outputs that can be imported in office applications.

Furthermore, for advanced users, each functions returns either the prepared ggplot-object (in case of sjp-plotting functions) or the HTML-tables (in case of sjt-table-output functions), which than can be manipulated even further (for instance, for ggplot-objects, you can specify certain parameters that cannot be modified via the sjPlot package or html-tables could be integrated into knitr-documents). The functions follow specific naming conventions:
Details

Package: sjPlot
Type: Package
Version: 1.3
Date: 2014-03-27
License: GPL-3

Author(s)
Daniel Lüdecke <d.luedecke@uke.de>

References
Weblog: http://strengejacke.wordpress.com/sjplot-r-package/
Github: https://github.com/sjPlot/devel

efc

Sample dataset from the EUROFAMCARE project

Description
Sample dataset from the EUROFAMCARE project

References
http://www.uke.de/eurofamcare/

See Also
sji.SPSS
sji.viewSPSS
sjt.df
sji.getValueLabels
sji.getVariableLabels
sji.convertToLabel
sjp.frq
Examples

# Attach EFC-data
data(efc)

# Show structure
str(efc)

# Show first rows
head(efc)

# Show variables
## Not run:
sji.viewSPSS(efc)
## End(Not run)

# Show variable labels
sji.getVariableLabels(efc)

# Plot efc-data frame summary
## Not run:
sjt.df(efc, alternateRowColor=TRUE)
## End(Not run)


descendation

sjc.cluster

\textit{Compute hierarchical or kmeans cluster analysis}

Description

Compute hierarchical or kmeans cluster analysis and returns the group association for each observation as vector.

Usage

sjc.cluster(data, groupcount, method = "h", distance = "euclidean",
agglomeration = "ward", iter.max = 20, algorithm = "Hartigan-Wong")

Arguments

data

The data frame containing all variables that should be used for the cluster analysis.

groupcount

The amount of groups (clusters) that should be retrieved. May also be a set of initial (distinct) cluster centres, in case method is "kmeans" (see \texttt{kmeans} for details on centers parameter). If groupcount indicates a number of clusters, following functions may be helpful for estimating the amount of clusters:

\begin{itemize}
  \item Use \texttt{sjc.elbow}-function to determine the group-count depending on the elbow-criterion.
  \item If using kmeans as method, use \texttt{sjc.kgap}-function to determine the group-count according to the gap-statistic.
\end{itemize}
• If using hierarchical as method (default), use `sjc.dend`-function to inspect different cluster group solutions.
• Use `sjc.grpdisc`-function to inspect the goodness of grouping (accuracy of classification).

`method` Indicates the clustering method. If "hclust" (default), a hierarchical clustering using the ward method is computed. Use "kmeans" to compute a k-means clustering. You can specify initial letters only.

`distance` The distance measure to be used when "method" is "hclust" (for hierarchical clustering). This must be one of "euclidean" (default), "maximum", "manhattan", "canberra", "binary" or "minkowski". See `dist`.

`agglomeration` The agglomeration method to be used when "method" is "hclust" (for hierarchical clustering). This should be one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid". Default is "ward". See `hclust`.

`iter.max` the maximum number of iterations allowed. Only applies, if method is "kmeans". See `kmeans` for details on this parameter.

`algorithm` algorithm used for calculating kmeans cluster. Only applies, if method is "kmeans". May be one of "Hartigan-Wong" (default), "Lloyd" (used by SPSS), or "MacQueen". See `kmeans` for details on this parameter.

Value

The group classification for each observation as vector. This group classification can be used for `sjc.grpdisc`-function to check the goodness of classification. The returned vector includes missing values, so it can be appended to the original data frame data.

Note

To get similar results as in SPSS Quick Cluster function, following points have to be considered:

1. Use the `/PRINT INITIAL` option for SPSS Quick Cluster to get a table with initial cluster centers.
2. Create a matrix of this table, by consecutively copying the values, one row after another, from the SPSS output into a matrix and specifying `nrow` and `ncol` parameters.
3. Use `algorithm="Lloyd"`.
4. Use the same amount of `iter.max` both in SPSS and this `sjc.qclus`.

This ensures a fixed initial set of cluster centers (as in SPSS), while `kmeans` in R always selects initial cluster sets randomly.

See Also

`sjc.dend`  
`sjc.grpdisc`  
`sjc.elbow`  
`kmeans`  
`hclust`
Examples

# Hierarchical clustering of mtcars-dataset
groups <- sjc.cluster(mtcars, 5)

# K-means clustering of mtcars-dataset
groups <- sjc.cluster(mtcars, 5, method="k")

sjc.dend  Compute hierarchical cluster analysis and visualize group classification

Description

Computes a hierarchical cluster analysis and plots a hierarchical dendrogram with highlighting rectangles around the classified groups. Can be used, for instance, as visual tool to verify the elbow-criterion (see sjc.elbow).

Usage

sjc.dend(data, groupcount, distance = "euclidean", agglomeration = "ward")

Arguments

data  The data frame containing all variables that should be used for the cluster analysis.
groupcount  The amount of groups (clusters) that should be used.
  • Use sjc.elbow-function to determine the group-count depending on the elbow-criterion.
  • If using kmeans as method, use sjc.kgap-function to determine the group-count according to the gap-statistic.
  • Use sjc.grpdisc-function to inspect the goodness of grouping (accuracy of classification).
  Solutions for multiple cluster groups can be plotted, for instance with "groupcount=c(3:6)".
distance  The distance measure to be used. This must be one of "euclidean" (default), "maximum", "manhattan", "canberra", "binary" or "minkowski". See dist.
agglomeration  The agglomeration method to be used. This should be one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid". Default is "ward". See hclust.

See Also

sjc.cluster
sjc.grpdisc
sjc.elbow
Examples

\# Plot dendrogram of hierarchical clustering of mtcars-dataset
\# and show group classification
sjc.dend(mtcars, 5)

\# Plot dendrogram of hierarchical clustering of mtcars-dataset
\# and show group classification for 2 to 4 groups
sjc.dend(mtcars, 2:4)

sjc.elbow

Compute elbow values of a k-means cluster analysis

Description

Plot elbow values of a k-means cluster analysis. This function computes a k-means cluster analysis on the provided data frame and produces two plots: one with the different elbow values and a second plot that maps the differences between each "step" (i.e. between elbow values) on the y-axis. An increase in the second plot may indicate the elbow criterion.

Usage

sjc.elbow(data, steps = 15, showDiff = FALSE)

Arguments

data The data frame containing all variables that should be used for determining the elbow criteria.
steps The maximum group-count for the k-means cluster analysis for which the elbow-criterion should be displayed. Default is 15.
showDiff If TRUE, an additional plot with the differences between each fusion step of the Elbow criterion calculation is shown. This plot may help identifying the "elbow". Default for this parameter is FALSE.

See Also

sjc.kgap
sjc.dend
sjc.cluster
sjc.grpdisc

Examples

\# plot elbow values of mtcars dataset
sjc.elbow(mtcars)
**sjc.grpdisc**

*Compute a linear discriminant analysis on classified cluster groups*

**Description**

Computes linear discriminant analysis on classified cluster groups. This function plots a bar graph indicating the goodness of classification for each group.

**Usage**

```r
sjc.grpdisc(data, groups, groupcount, showTotalCorrect = TRUE, printPlot = TRUE)
```

**Arguments**

- `data`: The data frame containing all variables that should be used for the check for goodness of classification of a cluster analysis.
- `groups`: The group classification of the cluster analysis that was returned from the `sjc.cluster` function.
- `groupcount`: The amount of groups (clusters) that should be used. Use the `sjc.elbow` function to determine the group-count depending on the elbow-criterion.
- `showTotalCorrect`: If `TRUE` (default), a vertical line indicating the overall goodness of classification is added to the plot, so one can see whether a certain group is below or above the average classification goodness.
- `printPlot`: If `TRUE` (default), plots the results as graph. Use `FALSE` if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

**Value**

(Invisibly) returns an object with

- `data`: the used data frame for plotting,
- `plot`: the ggplot object,
- `accuracy`: a vector with the accuracy of classification for each group,
- `total.accuracy`: the total accuracy of group classification.

**See Also**

- `sjc.dend`
- `sjc.cluster`
- `sjc.elbow`
Examples

```r
# retrieve group classification from hierarchical cluster analysis
# on the mtcars data set (5 groups)
groups <- sjc.cluster(mtcars, 5)

# plot goodness of group classificatoin
sjc.grpdisc(mtcars, groups, 5)
```

**sjc.kgap**

Compute gap statistics for k-means-cluster

**Description**

An implementation of the gap statistic algorithm from Tibshirani, Walther, and Hastie's "Estimating the number of clusters in a data set via the gap statistic". Source code was taken from the `clusGap` function of the cluster-package ([http://cran.r-project.org/web/packages/cluster/index.html](http://cran.r-project.org/web/packages/cluster/index.html)).

**Usage**

```r
sjc.kgap(x, max = 10, B = 100, SE.factor = 1, method = "Tibs2001SEmax", plotResults = TRUE)
```

**Arguments**

- `x`:
  A matrix, where rows are observations and columns are individual dimensions, to compute and plot the gap statistic (according to a uniform reference distribution).

- `max`:
  The maximum number of clusters to consider, must be at least two. Default is 10.

- `B`:
  integer, number of Monte Carlo ("bootstrap") samples. Default is 100.

- `SE.factor`:
  [When method contains "SE"]] Determining the optimal number of clusters, Tibshirani et al. proposed the "1 S.E."-rule. Using an SE.factor f, the "f S.E."-rule is used, more generally.

- `method`:
  A character string indicating how the "optimal" number of clusters, $k^*$, is computed from the gap statistics (and their standard deviations), or more generally how the location $k^*$ of the maximum of $f[k]$ should be determined. Default is "Tibs2001SEmax". Possible value are:
  - "globalmax" simply corresponds to the global maximum, i.e., is which.max(f).
  - "firstmax" gives the location of the first local maximum.
  - "Tibs2001SEmax" uses the criterion, Tibshirani et al.(2001) proposed: "the smallest $k$ such that $f(k) >= f(k+1) - s_{k+1}$". Note that this chooses $k = 1$ when all standard deviations are larger than the differences $f(k+1) - f(k)$.
  - "firstSEmax" is the location of the first $f()$ value which is not larger than the first local maximum minus SE.factor * SE.f[], i.e., within an "f S.E." range of that maximum (see also SE.factor).
• "globalSEmax" (used in Dudoit and Fridlyand (2002), supposedly following Tibshirani's proposition) is the location of the first f() value which is not larger than the global maximum minus SE.factor * SE.f[], i.e., within an "f S.E." range of that maximum (see also SE.factor).

plotResults If TRUE (default), a graph visualizing the gap statistic will be plotted. Use FALSE to omit the plot.

Value
An object containing the used data frame for plotting, the ggplot object and the number of found cluster.

Note
Source code was taken from the clusGap function of the cluster-package (http://cran.r-project.org/web/packages/cluster/index.html).

References

See Also
sjc.elbow
clusGap

Examples
# plot gap statistic and determine best number of clusters
# in mtcars dataset
sjc.kgap(mtcars)

# and in iris dataset
sjc.kgap(iris[,1:4])
Description

Compute a quick kmeans or hierarchical cluster analysis and displays "cluster characteristics" as graph.

1. If method is kmeans, this function first determines the optimal group count via gap statistics (unless parameter groupcount is specified), using the sjc.kgap function.
2. Than a cluster analysis is performed by running the sjc.cluster function to determine the cluster groups.
3. After that, all variables in data are scaled and centered. The mean value of these z-scores within each cluster group is calculated to see how certain characteristics (variables) in a cluster group differ in relation to other cluster groups.
4. These results are shown in a graph.

Usage

sjc.qclus(data, groupcount = NULL, groups = NULL, method = "k",
distance = "euclidean", agglomeration = "ward", iter.max = 20,
algorithm = "Hartigan-Wong", showAccuracy = FALSE, title = NULL,
titleSize = 1.3, titleColor = "black", axisLabels.x = NULL,
axisLabelAngle.x = 0, axisLabelSize = 1.1, axisLabelColor = "gray30",
axisTitle.x = "Cluster group characteristics",
axisTitle.y = "Mean of z-scores", axisTitleColor = "black",
axisTitleSize = 1.3, breakTitleAt = 40, breakLabelsAt = 12,
breakLegendTitleAt = 20, breakLegendLabelsAt = 20, facetCluster = FALSE,
barColor = NULL, barAlpha = 1, colorPalette = "GnBu", barWidth = 0.5,
barSpace = 0.1, barOutline = FALSE, outlineColor = "black",
theme = NULL, borderColor = NULL, axisColor = NULL,
hideLegend = FALSE, showTickMarks = TRUE, showAxisLabels.x = TRUE,
showAxisLabels.y = TRUE, showGroupCount = TRUE,
showAccuracyLabels = FALSE, legendTitle = NULL, legendLabels = NULL,
legendPos = "right", legendSize = 1, legendBorderColor = "white",
legendBackColor = "white", majorGridColor = NULL, minorGridColor = NULL,
hideGrid.x = FALSE, hideGrid.y = FALSE, flipCoordinates = FALSE,
printPlot = TRUE)

Arguments

data                      The data frame containing all variables that should be used for the cluster analysis.
groupcount               The amount of groups (clusters) that should be retrieved. May also be a set of initial (distinct) cluster centres, in case method is "kmeans" (see kmeans for details on centers parameter). By default (NULL), the optimal amount of clusters is calculated using the gap statistics (see sjc.kgap. However, this works only with kmeans as method. If method is "hclust", you have to specify a group-count. Use the sjc.elbow-function to determine the group-count depending on the elbow-criterion. Use sjc.grpdisc-function to inspect the goodness of grouping.
groups

By default, this parameter is NULL and will be ignored. However, if you just want to plot an already existing cluster solution without computing a new cluster analysis, specify groupcount and group. group is a vector of same length as nrow(data) and indicates the group classification of the cluster analysis. The group classification can be computed with the sjc.cluster function.

method

The method for computing the cluster analysis. By default ("kmeans"), a kmeans cluster analysis will be computed. Use "hclust" to compute a hierarchical cluster analysis. You can specify the initial letters only.

distance

The distance measure to be used when "method" is "hclust" (for hierarchical clustering). This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski". See dist. By default, method is "kmeans" and this parameter will be ignored.

agglomeration

The agglomeration method to be used when "method" is "hclust" (for hierarchical clustering). This should be one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid". Default is "ward". See hclust. By default, method is "kmeans" and this parameter will be ignored.

iter.max

the maximum number of iterations allowed. Only applies, if method is "kmeans". See kmeans for details on this parameter.

algorithm

algorithm used for calculating kmeans cluster. Only applies, if method is "kmeans". May be one of "Hartigan-Wong" (default), "Lloyd" (used by SPSS), or "MacQueen". See kmeans for details on this parameter.

showAccuracy

If TRUE, the sjc.grpdisc function will be called, which computes a linear discriminant analysis on the classified cluster groups and plots a bar graph indicating the goodness of classification for each group.

title

Title of diagram as string. Example: title=c("my title")

titleSize

The size of the plot title. Default is 1.3.

titleColor

The color of the plot title. Default is "black".

axisLabels.x

Labels for the x-axis breaks. Example: axisLabels.x=c("Label1", "Label2", "Label3"). Note: If you use the sj1.SPSS function and the sj1.getValueLabels function, you receive a list object with label string. The labels may also be passed as list object. They will be unlisted and converted to character vector automatically.

axisLabelAngle.x

Angle for axis-labels.

axisLabelSize

The size of axis labels of both x and y axis. Default is 1.1, recommended values range between 0.5 and 3.0.

axisLabelColor

User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

axisTitle.x

A label for the x axis. useful when plotting histograms with metric scales where no category labels are assigned to the x axis.

axisTitle.y

A label for the y axis. useful when plotting histograms with metric scales where no category labels are assigned to the y axis.

axisTitleColor

The color of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.
axisTitleSize  the size of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels. Default is 1.3.
breakTitleAt  Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.
breakLabelsAt  Determines how many chars of the labels are displayed in one line and when a line break is inserted into the axis labels.
breakLegendTitleAt  Determines how many chars of the legend title are displayed in one line and when a line break is inserted into the legend title.
breakLegendLabelsAt  Determines how many chars of the legend labels are displayed in one line and when a line break is inserted into the axis labels.
facetCluster  If TRUE, each cluster group will be represented by an own panel. Default is FALSE, thus all cluster groups are plotted in a single graph.
barColor  User defined color for bars.
  • If not specified (NULL), a default color palette will be used for the bar charts.
  • If barColor is "gs", a greyscale will be used.
  • If barColor is "bw", a monochrome white filling will be used.
  • If barColor is "brewer", use the colorPalette parameter to specify a palette of the color brewer.
Else specify your own color values as vector (e.g. barColor=c("#f00000", "#00ff00", "#0080ff")).
barAlpha  Specify the transparancy (alpha value) of bars.
colorPalette  If barColor is "brewer", specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.
barWidth  Width of bars. Recommended values for this parameter are from 0.4 to 1.5
barSpace  Spacing between bars. Default value is 0.1. If 0 is used, the grouped bars are stucked together and have no space in between. Recommended values for this parameter are from 0 to 0.5
barOutline  If TRUE, each bar gets a colored outline. Default is FALSE.
outlineColor  The color of the bar outline. Only applies, if barOutline is set to TRUE.
theme  Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.
  • Use "bw" for a white background with gray grids
  • "classic" for a classic theme (black border, no grids)
  • "minimal" for a minimalistic theme (no border, gray grids) or
  • "none" for no borders, grids and ticks.
borderColor  User defined color of whole diagram border (panel border).
axisColor  User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).
hideLegend  Indicates whether legend (guide) should be shown or not.
showTickMarks  Whether tick marks of axes should be shown or not.
showAxisLabels.x
Whether x axis labels (cluster variables) should be shown or not.

showAxisLabels.y
Whether y axis labels (z scores) should be shown or not.

showGroupCount
if TRUE (default), the count within each cluster group is added to the legend labels (e.g. "Group 1 (n=87)").

showAccuracyLabels
if TRUE, the accuracy-values for each cluster group is added to the legend labels (e.g. "Group 1 (n=87, accuracy=95.3)"). Accuracy is calculated by sjc.grpdisc.

legendTitle
Title of the diagram’s legend.

legendLabels
Labels for the guide/legend. Example: See axisLabels.x. If legendLabels is NULL (default), the standard string "Group <nr>" will be used.

legendPos
The position of the legend, if a legend is drawn. Use "bottom", "top", "left" or "right" to position the legend above, below, on the left or right side of the diagram. Right positioning is default.

legendSize
The text size of the legend. Default is 1. Relative size, so recommended values are from 0.3 to 2.5

legendBorderColor
Color of the legend’s border. Default is "white", so no visible border is drawn.

legendBackColor
Fill color of the legend’s background. Default is "white", so no visible background is drawn.

majorGridColor
Specifies the color of the major grid lines of the diagram background.

minorGridColor
Specifies the color of the minor grid lines of the diagram background.

hideGrid.x
If TRUE, the x-axis-gridlines are hidden. Default is FALSE.

hideGrid.y
If TRUE, the y-axis-gridlines are hidden. Default is FALSE.

flipCoordinates
If TRUE, the x and y axis are swapped.

printPlot
If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value
(Invisibly) returns an object with
- data: the used data frame for plotting,
- plot: the ggplot object,
- groupcount: the number of found cluster (as calculated by sjc.kgap)
- classification: the group classification (as calculated by sjc.cluster), including missing values, so this vector can be appended to the original data frame.
- accuracy: the accuracy of group classification (as calculated by sjc.grpdisc).
Note

To get similar results as in SPSS Quick Cluster function, following points have to be considered:

1. Use the /PRINT INITIAL option for SPSS Quick Cluster to get a table with initial cluster centers.
2. Create a matrix of this table, by consecutively copying the values, one row after another, from the SPSS output into a matrix and specifying nrow and ncol parameters.
3. Use algorithm="Lloyd".
4. Use the same amount of iter.max both in SPSS and this sjc.qclus.

This ensures a fixed initial set of cluster centers (as in SPSS), while kmeans in R always selects initial cluster sets randomly.

See Also

sjc.cluster
sjc.kgap
sjc.elbow
sjc.grpdisc

Examples

# K-means clustering of mtcars-dataset
sjc.qclus(mtcars)

# K-means clustering of mtcars-dataset with 4 pre-defined groups in a faceted panel
sjc.qclus(airquality, groupcount=4, facetCluster=TRUE)

Description

This function converts (replaces) variable values (of factors) with their associated value labels. Might be helpful for factor variables. For instance, if you have a Gender variable with 0/1, and associated labels are male/female, this function would convert all 0 to male and all 1 to female in the data frame.

Usage

sji.convertToLabel(variable)

Arguments

variable A (factor) variable.
**Value**

A factor variable containing with the replaced value labels.

**See Also**

- `sji.convertToValue`
- `sji.getValueLabels`
- `sji.getVariableLabels`
- `sji.SPSS`

**Examples**

```r
data(efc)
print(sji.getValueLabels(efc)['c161sex'])
head(efc$c161sex)
head(sji.convertToLabel(efc$c161sex))

print(sji.getValueLabels(efc)['e42dep'])
table(efc$e42dep)
table(sji.convertToLabel(efc$e42dep))
```

**Description**

This function converts (replaces) factor values with the related factor level index number, thus the factor is converted to a numeric variable.

**Usage**

```r
sji.convertToValue(fac, startAt = 1)
```

**Arguments**

- `fac` A (factor) variable.
- `startAt` the starting index, i.e. numeric value of the variable.

**Value**

A numeric variable with values ranging from `startAt` to `startAt` + length of factor levels.

**See Also**

- `sji.convertToLabel`
- `sji.getValueLabels`
- `sji.getVariableLabels`
- `sji.SPSS`
Examples

data(efc)
test <- sji.convertToLabel(efc$e42dep)
table(test)

table(sji.convertToValue(test))
hist(sji.convertToValue(test,0))

sjii.getValueLabels Retrieve value labels of an SPSS-imported data frame

Description
This function retrieves the value labels of an imported SPSS data set and returns the result as list.

Usage

tsji.getValueLabels(dat)

Arguments

dat a data frame containing imported SPSS data

Value
a list with all value labels from the SPSS dataset

See Also

tsji.SPSS
sjii.getVariableLabels
sjii.convertToLabel
sjii.convertToValue
sjii.setValueLabels

Examples

# import SPSS data set
# mydat <- sji.SPSS("my_spss_data.sav", enc="UTF-8")

# retrieve variable labels
# mydat.var <- sji.getVariableLabels(mydat)

# retrieve value labels
# mydat.val <- sji.getValueLabels(mydat)
**sji.getVariableLabels**  
*Retrieves variable labels of an SPSS-imported data frame*

**Description**

This function retrieves the variable labels of an imported SPSS data set and returns the result as list.

**Usage**

```r
sji.getVariableLabels(dat)
```

**Arguments**

- `dat`  
  A data frame containing imported SPSS data.

**Value**

A list with all variable labels from the SPSS dataset.

**See Also**

- `sji.getValueLabels`
- `sji.setValueLabels`
- `sji.setVariableLabels`
- `sji.SPSS`
- `sji.convertToLabel`
- `sji.convertToValue`

**Examples**

```r
# import SPSS data set
# mydat <- sji.SPSS("my_spss_data.sav", enc="UTF-8")

# retrieve variable labels
# mydat.var <- sji.getVariableLabels(mydat)

# retrieve value labels
# mydat.val <- sji.getValueLabels(mydat)
```
**sjii.setValueLabels**  
*Attach value labels to a variable or vector*

**Description**

This function attaches character labels as "value.labels" attribute to a variable or vector "var". These value labels will be accessed by most of this package’s functions, in order to automatically set values or legend labels.

**Usage**

```
sji.setValueLabels(var, labels)
```

**Arguments**

- **var**: a variable (vector) where labels should be attached. Replaces former value labels.
- **labels**: a character vector of labels that will be attached to "var" by setting the "value.labels" attribute. The length of this character vector must equal the value range of "var", i.e. if "var" has values from 1 to 3, "labels" should have a length of 3.

**Value**

the variable "var" with attached value labels.

**See Also**

- `sjii.SPSS`
- `sjii.getVariableLabels`
- `sjii.convertToLabel`
- `sjii.convertToValue`
- `sjii.getValueLabels`

**Examples**

```
dummy <- sample(1:4, 40, replace=TRUE)
sjp.frq(dummy)

dummy <- sjii.setValueLabels(dummy, c("very low", "low", "mid", "hi"))
sjp.frq(dummy)
```
**sji.setVariableLabels**  
Set variable label(s) to a single variable or data frame

### Description

This function sets variable labels to a single variable or to a set of variables in a data frame. To each variable, the attribute "variable.label" with the related variable name is attached. Most of this package’s function can automatically retrieve the variable name to use it as axis labels or plot title.

### Usage

```
sji.setVariableLabels(x, lab)
```

### Arguments

- **x**  
  A single variable (vector) or data frame with variables.

- **lab**  
  If `x` is a vector (single variable), use a single character string with the variable label for `x`. If `x` is a `data.frame`, use a vector with character labels of same length as `nco1(x)`.

### Value

`x`, with attached "variable.label" attribute(s), which contains the variable name(s).

### See Also

- `sji.getValueLabels`
- `sji.setValueLabels`
- `sji.getVariableLabels`
- `sji.SPSS`
- `sji.convertToLabel`
- `sji.convertToValue`

### Examples

```r
# sample data set, imported from SPSS. Variable labels are attached  
# as attribute to the data frame (so variables currently don't have this attribute)  
data(efc)  
# get variable labels  
variable.labels <- sji.getVariableLabels(efc)  
# set variable labels as attribute to each single variable of data frame  
efc <- sji.setVariableLabels(efc, variable.labels)  

## Not run:  
sjt.frq(efc$e42dep)  
sjt.frq(data.frame(efc$e42dep, efc$e16sex))  
## End(Not run)
```
#……………………………………#
# manually set value and variable labels 
# ………………………………………#
dummy <- sample(1:4, 40, replace=TRUE)
dummy <- sji.setValueLabels(dummy, c("very low", "low", "mid", "hi"))
dummy <- sji.setVariableLabels(dummy, "Dummy-variable")
# auto-detection of value labels by default, auto-detection of 
# variable labels if parameter "title" set to "auto"
sjp.frq(dummy, title="auto")

--

# sjie SPSS

## Import SPSS dataset as data frame into R

### Description
Import data from SPSS, including NA's, value and variable labels.

### Usage

```r
sjie.SPSS(path, enc = NA, autoAttachVarLabels = FALSE)
```

### Arguments

- **path**
The file path to the SPSS dataset.
- **enc**
The file encoding of the SPSS dataset.
- **autoAttachVarLabels**
if TRUE, variable labels will automatically be attached to each variable as "variable.label" attribute. See `sjie.setVariableLabels` for details.

### Value
A data frame containing the SPSS data. retrieve value labels with `sjie.getValueLabels` and variable labels with `sjie.getVariableLabels`.

### Note
This is a wrapper function for `read.spss` of the foreign package, using convenient parameter default settings.

### References

- [http://strengejacke.wordpress.com/2013/02/24/simplify-your-r-workflow-with-functions-rstats/](http://strengejacke.wordpress.com/2013/02/24/simplify-your-r-workflow-with-functions-rstats/)
sji.viewSPSS

See Also

sji.getValueLabels
sji.getVariableLabels
sji.convertToLabel
sji.convertToValue
sji.viewSPSS

Examples

# import SPSS data set
# mydat <- sji.SPSS("my_spss_data.sav", enc="UTF-8")

# retrieve variable labels
# mydat.var <- sji.getVariableLabels(mydat)

# retrieve value labels
# mydat.val <- sji.getValueLabels(mydat)

---

sji.viewSPSS  View SPSS data set structure

Description

Save (or show) content of an imported SPSS data file as HTML table. Similar to the SPSS variable view. This quick overview shows variable ID number, name, label, type and associated value labels. The result can be considered as "codeplan" of the data frame.

Usage

sji.viewSPSS(df, file = NULL, alternateRowColors = TRUE, showType = FALSE,
showValues = TRUE, orderByName = FALSE, breakVariableNamesAt = 50,
encoding = "UTF-8", CSS = NULL, useViewer = TRUE, no.output = FALSE)

Arguments

df  An imported data frame, imported by sji.SPSS function.
file  The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and opened either in the IDE’s viewer pane or in the default web browser.
alternateRowColors  If TRUE, alternating rows are highlighted with a light gray background color.
showType  If TRUE, the variable type is shown in a separate column. Since SPSS variable types are mostly numeric after import, this column is hidden by default.
showValues  If TRUE (default), the variable values and their associated value labels are shown as additional column.
orderByName  If TRUE, rows are ordered according to the variable names. By default, rows (variables) are ordered according to their order in the data frame.
breakVariableNamesAt

Wordwrap for long variable names. Determines how many chars of a variable name are displayed in one line and when a line break is inserted. Default value is 50, use NULL to turn off word wrap.

encoding

The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g.) German umlauts).

CSS

A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:

1. the class-names with "css."-prefix as parameter name and
2. each style-definition must end with a semicolon

Examples:

- css.table='border:2px solid red;' for a solid 2-pixel table border in red.
- css.summary='font-weight:bold;' for a bold fontweight in the summary row.
- css.arc='color:blue;' for a blue text color each 2nd row.

See further examples below.

useViewer

If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

no.output

If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with

- the web page style sheet (page.style),
- the web page content (page.content),
- the complete html-output (output.complete) and
- the html-table with inline-css for use with knitr (knitr)

for further use.

See Also

sjl.SPSS
sjt.df
Examples

```r
# init dataset
data(efc)

# view variables
# Not run:
sji.viewSPSS(efc)
# End(Not run)

# view variables w/o values and value labels
# Not run:
sji.viewSPSS(efc, showValues=FALSE)
# End(Not run)

# view variables including variable typed, order by name
# Not run:
sji.viewSPSS(efc, orderByName=TRUE, showType=TRUE)
# End(Not run)

# User defined style sheet
# Not run:
sji.viewSPSS(efc,
  CSS=list(css.table="border: 2px solid;",
           css.tdata="border: 1px solid;",
           css.arc="color:blue;"))
# End(Not run)
```

---

**sjp.aov1**  
*Plot One-Way-Anova tables*

**Description**

Plot One-Way-Anova table sum of squares (SS) of each factor level (group) against the dependent variable. The SS of the factor variable against the dependent variable (variance within and between groups) is printed to the model summary.

**Usage**

```r
sjp.aov1(depVar, grpVar, meansums = FALSE, type = "dots",
  hideErrorBars = FALSE, title = NULL, titleSize = 1.3,
  titleColor = "black", axisLabels.y = NULL, reverseOrder = FALSE,
  stringIntercept = "( Intercept)", showAxisLabels.y = TRUE,
  axisLabelSize = 1.1, axisLabelColor = "gray30", axisTitle.x = NULL,
  axisTitleSize = 1.2, axisTitleColor = c("#444444"), axisLimits = NULL,
  valueLabelColor = "grey20", valueLabelColorNS = "grey50",
  valueLabelSize = 4.5, valueLabelAlpha = 0.8, axisLabelAngle.x = 0,
  axisLabelAngle.y = 0, errorBarColor = NULL, errorBarWidth = 0,
  ...)```

Arguments

**depVar**
The dependent variable. Will be used with following formula: `aov(depVar ~ grpVar)`

**grpVar**
The grouping variable, as unordered factor. Will be used with following formula: `aov(depVar ~ grpVar)`

**meansums**
If TRUE, the values reported are the true group mean values. If FALSE (default), the values are reported in the standard way, i.e. the values indicate the difference of the group mean in relation to the intercept (reference group).

**type**
Indicates Whether the group means should be plotted as “dots” (aka forest plots, default) or as "bars".

**hideErrorBars**
If TRUE, the error bars that indicate the confidence intervals of the group means are not shown. Only applies if parameter type is "bars". Default value is FALSE.

**title**
Diagram’s title as string. Example: `title=c("my title")` Use "auto" to automatically detect variable names that will be used as title (see `sji.setVariableLabels` for details).

**titleSize**
The size of the plot title. Default is 1.3.

**titleColor**
The color of the plot title. Default is "black".

**axisLabels.y**
Value labels of the grouping variable grpVar that are used for labelling the grouping variable axis. Passed as vector of strings. Example: `axisLabels.y=c("Label1", "Label2", ...)`

**reverseOrder**
If TRUE, the order of the factor categories (groups) is reversed. Default is FALSE.

**stringIntercept**
A string that indicates the reference group (intercept), that is appended to the value label of the grouping variable. Default is "(Intercept)".

**showAxisLabels.y**
Whether y axis text (category value) should be shown (use TRUE) or not. Default is TRUE.

**axisLabelSize**
The size of value labels in the diagram. Default is 4, recommended values range between 2 and 8.

**axisLabelColor**
The color of the category labels (predictor labels). Default is a dark grey (grey30).
axisTitle.x  A label for the x axis. Default is NULL, which means no x-axis title. Use "auto" to automatically detect variable names that will be used as title (see sj.setVariableLabels) for details.

axisTitleColor  The color of the x axis label.

axisTitleSize  The size of the x axis label. Default is 1.2.

axisLimits  Defines the range of the axis where the beta coefficients and their confidence intervals are drawn. By default, the limits range from the lowest confidence interval to the highest one, so the diagram has maximum zoom. Use your own values as 2-value-vector, for instance: limits=c(-0.8,0.8).

valueLabelColor  Colour of the values inside the diagram. Only applies, when parameter showValueLabels is set to TRUE. Use any valid colour value, e.g. valueLabelColor="grey50" or valueLabelColor=c("#cc3366"). Default is "grey20".

valueLabelColorNS  Colour of the non significant values inside the diagram. Only applies, when parameter showValueLabels is set to TRUE. Use any valid colour value, e.g. valueLabelColor="grey50" or valueLabelColor=c("#cc3366"). Default is "grey50".

valueLabelSize  Size of the value labels. Default is 4.5. Recommended Values range from 2 to 8.

valueLabelAlpha  The alpha level (transparancy) of the value labels. Default is 0.8, use any value from 0 to 1.

axisLabelAngle.x  Angle for x-axis-labels, passed as numeric value.

axisLabelAngle.y  Angle for y-axis-labels, passed as numeric value.

errorBarColor  The color of the error bars that indicate the confidence intervals of the group means. Default is NULL, which means that if type is "dots", the pointColors value will be used as error bar color. In case type is "bars", "black" will be used as error bar color.

errorBarWidth  The width of the error bar ends. Default is 0.

errorBarSize  The size of the error bar. Default is 0.8.

errorBarLineType  The linetype of error bars. Default is 1 (solid line).

pointColors  The colours of the points that indicate the mean-value. pointColors is a vector with two values: the first indicating groups with positive means and the second indicating negative means. Default is c("#3366a0", "#aa6633").

pointSize  The size of the points that indicate the mean-value. Default is 3.

barColors  The colors of the bars in bar charts. Only applies if parameter type is "bars". barColors is a vector with two values: the first indicating groups with positive means and the second indicating negative means. Default is c("#3366a0", "#aa6633").

barWidth  The width of the bars in bar charts. Only applies if parameter type is "bars". Default is 0.5.

barAlpha  The alpha value of the bars in bar charts. Only applies if parameter type is "bars". Default is 1.
barOutline  If TRUE, each bar gets a colored outline. Only applies if parameter type is bars. Default is FALSE.

outlineColor  The color of the bar outline. Only applies, if barOutline is set to TRUE. Default is black.

breakTitleAt  Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt  Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

gridBreaksAt  Sets the breaks on the y axis, i.e. at every n’th position a major grid is being printed. Default is NULL, so pretty gridbeaks will be used.

borderColor  User defined color of whole diagram border (panel border).

axisColor  User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

theme  Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.
  • Use “bw” for a white background with gray grids
  • ”classic” for a classic theme (black border, no grids)
  • ”minimal” for a minimalistic theme (no border, gray grids) or
  • ”none” for no borders, grids and ticks.

majorGridColor  Specifies the color of the major grid lines of the diagram background.

minorGridColor  Specifies the color of the minor grid lines of the diagram background.

hideGrid.x  If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y  If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

showTickMarks  Whether tick marks of axes should be shown or not.

showValueLabels  Whether the value labels (mean differences) should be plotted to each dot or not.

labelDigits  The amount of digits for rounding the estimations (see showValueLabels). Default is 2, i.e. estimators have 2 digits after decimal point.

showPValueLabels  Whether the significance levels of each category/group should be appended to values or not.

showModelSummary  If TRUE (default), a summary of the anova model with Sum of Squares between groups (ssb), Sum of Squares within groups (ssw), multiple and adjusted R-square and F-Test is printed to the lower right corner of the diagram. Default is TRUE.

printPlot  If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Insensibly) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).
See Also

sju.aov1.levene

Examples

data(efc)
# note: "grpVar" does not need to be a factor.
# coercion to factor is done by the function
sjp.aov1(efc$c12hour, efc$e42dep)

data(efc)
efc.val <- sji.getValueLabels(efc)
efc.var <- sji.getVariableLabels(efc)
sjp.aov1(efc$c12hour,
        as.factor(efc$e42dep),
        axisLabels.y=efc.val['e42dep'],
        axisTitle.x=efc.var[['c12hour']])

# -------------------------------#
# auto-detection of value labels and variable names
# -------------------------------#
efc <- sji.setVariableLabels(efc, efc.var)
sjp.aov1(efc$c12hour,
        efc$e42dep,
        title="auto",
        axisTitle.x="auto")

ts proj.12hour,
        as.factor(efc$c172code),
        axisLabels.y=efc.val['c172code'],
        title=efc.var[['c12hour']],
        type="bars",
        showTickMarks=FALSE,
        showModelSummary=FALSE,
        axisLabelAngle.x=90)

sjp.chi2  Plot Pearson’s Chi2-Test of multiple contingency tables

Description

Plot Pearson’s Chi2-Test of multiple contingency tables as ellipses or tiles. Requires a data frame
with dichotomous (dummy) variables. Calculation of Chi2-matrix taken from following blog-posting:
http://talesofr.wordpress.com/2013/05/05/ridiculously-photogenic-factors-heatmap-with-p-values
Usage

sjp.chi2(df, title = "Pearson's Chi2-Test of Independence", titleSize = 1.3, 
tintColor = "black", axisLabels = NULL, valueLabelColor = "black", 
valueLabelSize = 4.5, valueLabelAlpha = 1, outlineColor = "black", 
outlineSize = 0.5, axisColor = NULL, axisLabelSize = 1.1, 
axisLabelColor = "gray30", axisLabelAngle.x = 0, axisLabelAngle.y = 0, 
breakTitleAt = 50, breakLabelsAt = 12, hideLegend = TRUE, 
legendTitle = NULL, printPlot = TRUE)

Arguments

df a data frame of (dichotomous) factor variables.
title Title of the diagram, plotted above the whole diagram panel
titleSize The size of the plot title. Default is 1.3.
tintColor The color of the plot title. Default is "black".
axisLabels Labels for the x- and y-axis. axisLabels are detected automatically if each variable has a "variable.label" attribute (see sjisetVariableLabels) for details).
valueLabelColor the color of the value labels (numbers) inside the diagram
valueLabelSize The size of value labels in the diagram. Default is 4.5, recommended values range between 2 and 8
valueLabelAlpha specify the transparency (alpha value) of value labels
outlineColor defines the outline color of geoms (circles or tiles). Default is black.
outlineSize defines the outline size of geoms (circles or tiles). Default is 1.
axisColor user defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border)
axisLabelSize The size of variable labels at the axes. Default is 1.1, recommended values range between 0.5 and 3.0
axisLabelColor user defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels
axisLabelAngle.x angle for x-axis-labels
axisLabelAngle.y angle for y-axis-labels
breakTitleAt Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title
breakLabelsAt Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted
hideLegend show or hide the legend. The legend indicates the strength of correlations by gradient colour fill.
legendTitle the legend title, provided as string, e.g. legendTitle=c("Strength of correlation"). Default is NULL, hence no legend title is used.
printPlot  If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Insisibily) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

References

http://strengejacke.wordpress.com/sjplot-r-package/
http://talesofr.wordpress.com/2013/05/05/ridiculously-photogenic-factors-heatmap-with-p-values/

Examples

# create data frame with 5 dichotomous (dummy) variables
df <- data.frame(as.factor(sample(1:2, 100, replace=TRUE)),
                  as.factor(sample(1:2, 100, replace=TRUE)),
                  as.factor(sample(1:2, 100, replace=TRUE)),
                  as.factor(sample(1:2, 100, replace=TRUE)),
                  as.factor(sample(1:2, 100, replace=TRUE)))

# create variable labels
items <- list(c("Item 1", "Item 2", "Item 3", "Item 4", "Item 5"))

# plot Chi2-contingency-table
sjp.chi2(df, axisLabels=items)

sjp.corr  Plot correlation matrix

Description

Plot correlations as ellipses or tiles. Required parameter is either a data frame or a computed cor-
object. In case of ellipses, the ellipses size indicates the strength of the correlation. Furthermore, blue and red colors indicate positive or negative correlations, where stronger correlations are dark-
ened.

Usage

sjp.corr(data, title = NULL, titleSize = 1.3, titleColor = "black",
         axisLabels = NULL, type = "circle", sortCorrelations = TRUE,
         decimals = 3, missingDeletion = "listwise", corMethod = "spearman",
         geomAlpha = 0.8, valueLabelColor = "black", valueLabelSize = 4.5,
         valueLabelAlpha = 1, circleSize = 15, outlineColor = "black",
         outlineSize = 1, axisColor = NULL, borderColor = NULL,
         axisLabelSize = 1.1, axisLabelColor = "gray30", axisLabelAngle.x = 0,
         axisLabelAngle.y = 0, breakTitleAt = 50, breakLabelsAt = 12,
         hideDiagCircle = TRUE, hideLegend = TRUE, legendTitle = NULL,
showCorrelationValueLabels = TRUE, showCorrelationPValues = TRUE, pvaluesAsNumbers = FALSE, showTickMarks = FALSE, fillColor = NULL, majorGridColor = NULL, minorGridColor = NULL, theme = NULL, printPlot = TRUE)

Arguments

data A correlation object, built with the R-cor-function, or a data frame which correlations should be calculated.
title Title of the diagram, plotted above the whole diagram panel.
titleSize The size of the plot title. Default is 1.3.
tintColor The color of the plot title. Default is "black".
axisLabels Labels for the x- and y-axis. axisLabels are detected automatically if data is a data frame where each variable has a "variable.label" attribute (see sjि.setVariableLabels for details).
type Indicates whether the geoms of correlation values should be plotted as "circle" (default) or as "tile".
sortCorrelations If TRUE (default), the axis labels are sorted according to the correlation strength. If FALSE, axis labels appear in order of how variables were included in the correlation computation or data frame.
decimals Indicates how many decimal values after comma are printed when the values labels are shown. Default is 3. Only applies when showCorrelationValueLabels is TRUE.
missingDeletion Indicates how missing values are treated. May be either "listwise" (default) or "pairwise".
corMethod Indicates the correlation computation method. May be one of "spearman" (default), "pearson" or "kendall".
geomAlpha Specify the transparency (alpha value) of geom objects (circles or tiles). Default is 0.8.
valueLabelColor The color of the value labels (numbers) inside the diagram. Default is "black".
valueLabelSize The size of value labels in the diagram. Default is 4.5, recommended values range between 2 and 8.
valueLabelAlpha Specify the transparency (alpha value) of value labels. Default is 1.
circleSize Specifies the circle size factor. The circle size depends on the correlation value multiplicated with this factor. Default is 15.
outlineColor Defines the outline color of geoms (circles or tiles). Default is black.
outlineSize Defines the outline size of geoms (circles or tiles). Default is 1.
axisColor User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).
borderColor User defined color of whole diagram border (panel border).
axisLabelSize  The size of variable labels at the axes. Default is 1.1, recommended values range between 0.5 and 3.0.

axisLabelColor  User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

axisLabelAngle.x  Angle for x-axis-labels.

axisLabelAngle.y  Angle for y-axis-labels.

breakTitleAt  Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title. Default is 50.

breakLabelsAt  Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted. Default is 12.

hideDiagCircle  If TRUE (default), the geoms of the diagonal correlations (self-correlations with value "1") are not plotted. Only applies if parameter type is "circle".

hideLegend  Show or hide the legend. The legend indicates the strength of correlations by gradient color fill. Default is TRUE, hence the legend is hidden.

legendTitle  The legend title, provided as string, e.g. legendTitle=c("Strength of correlation"). Default is NULL, hence no legend title is used.

showCorrelationValueLabels  Whether correlation values should be plotted to each geom

showCorrelationPValues  Whether significance levels (p-values) of correlations should be plotted to each geom.

pvaluesAsNumbers  If TRUE, the significance levels (p-values) are printed as numbers. If FALSE (default), asterisks are used.

showTickMarks  Whether tick marks should be plotted or not. Default is FALSE.

color  A color palette for filling the geoms. If not specified, the 5th diverging color palette from the color brewer palettes (RdBu) is used, resulting in red colors for negative and blue colors for positive correlations, that become lighter the weaker the correlations are. Use any color palette that is suitable for the scale_fill_gradientn parameter of ggplot2.

majorGridColor  Specifies the color of the major grid lines of the diagram background.

minorGridColor  Specifies the color of the minor grid lines of the diagram background.

theme  Specifies the diagram's background theme. Default (parameter NULL) is a gray background with white grids.

  • Use "bw" for a white background with gray grids
  • "classic" for a classic theme (black border, no grids)
  • "minimal" for a minimalistic theme (no border, gray grids) or
  • "none" for no borders, grids and ticks.

printPlot  If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.
**Value**

(Insisibily) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df) and the original correlation matrix (corr.matrix).

**References**

- [http://strengejacke.wordpress.com/2013/04/18/examples-for-sjplotting-functions-including-correlation](http://strengejacke.wordpress.com/2013/04/18/examples-for-sjplotting-functions-including-correlation)

**Examples**

```r
# create data frame with 5 random variables
df <- as.data.frame(cbind(rnorm(10), rnorm(10), rnorm(10), rnorm(10), rnorm(10)))

# plot correlation matrix using circles
sjp.corr(df)

# plot correlation matrix using square tiles without diagram background
sjp.corr(df, type="tile", theme="none")
```

```r
# Data from the EUROFAMCARE sample dataset
# -----------------------------
data(efc)

# retrieve variable and value labels
varlabs <- sji.getVariableLabels(efc)

# receive first item of COPE-index scale
start <- which(colnames(efc)=="c83cop2")
# receive last item of COPE-index scale
end <- which(colnames(efc)=="c88cop7")

# create data frame with COPE-index scale
df <- as.data.frame(efc[,c(start:end)])
colnames(df) <- varlabs[c(start:end)]

# we have high correlations here, because all items
# belong to one factor. See example from "sjp.pca".
sjp.corr(df, type="tile", theme="none", outlineColor="white", hideLegend=FALSE)
```

```r
# -----------------------------
# auto-detection of labels
# -----------------------------
efc <- sji.setVariableLabels(efc, varlabs)
sjp.corr(efc[,c(start:end)])
```
Plot frequencies of (count) variables

Description

Plot frequencies of a (count) variable as bar graph, histogram, box plot etc. using ggplot.

Usage

sjp.frq(varCount, title = NULL, titleSize = 1.3, titleColor = "black", weightBy = NULL, weightByTitleString = NULL, interactionVar = NULL, maxYlim = FALSE, upperYlim = NULL, order = "none", type = "bars", axisLabels.x = NULL, interactionVarLabels = NULL, axisLabelAngle.x = 0, axisLabelSize = 1.1, axisLabelColor = "gray30", valueLabelSize = 4, valueLabelColor = "black", breakTitleAt = 50, breakLabelsAt = 12, gridBreaksAt = NULL, barWidth = 0.6, dotSize = 4, barColor = NULL, barAlpha = 1, barOutline = FALSE, barOutlineSize = 0.2, innerBoxPlotWidth = 0.15, innerBoxPlotDotSize = 3, borderColor = NULL, axisColor = NULL, outlineColor = "black", majorGridColor = NULL, minorGridColor = NULL, hideGrid.x = FALSE, hideGrid.y = FALSE, showValueLabels = TRUE, showCountValues = TRUE, showPercentageValues = TRUE, showAxisLabels.x = TRUE, showTickMarks = TRUE, showMeanIntercept = FALSE, showMeanValue = TRUE, showStandardDeviation = TRUE, showNormalCurve = FALSE, showStandardNormalCurve = FALSE, adjustNormalCurve.x = FALSE, meanInterceptLineType = 2, meanInterceptLineWidth = 0.5, normalCurveColor = "red", normalCurveSize = 0.8, normalCurveAlpha = 0.4, axisTitle.x = NULL, axisTitle.y = NULL, axisTitleColor = "black", axisTitleSize = 1.3, startAxisAt = "auto", hist.skipZeros = FALSE, autoGroupAt = NULL, theme = NULL, flipCoordinates = FALSE, na.rm = TRUE, printPlot = TRUE)

Arguments

- **varCount**: The variable which frequencies should be plotted.
- **title**: Title of diagram as string. Example: title=c("my title"). Use "auto" to automatically detect variable names that will be used as title (see sj.setVariableLabels for details).
- **titleSize**: The size of the plot title. Default is 1.3.
- **titleColor**: The color of the plot title. Default is "black".
- **weightBy**: A weight factor that will be applied to weight all cases from varCount. Default is NULL, so no weights are used.
- **weightByTitleString**: If a weight factor is supplied via the parameter weightBy, the diagram’s title may indicate this with a remark. Default is NULL, so the diagram’s title will
interactionVar: An interaction variable which can be used for box plots. Divides the observations in varCount into the factors (sub groups) of interactionVar. Only applies when parameter "type" is "box" or "violin" (resp. their alternative strings like "boxplot", "boxplots" or "v").

maxYlim: Indicates how to calculate the maximum limit of the y-axis. If TRUE, the upper y-limit corresponds to the amount of cases, i.e. y-axis for each plot of a data base are the same. If FALSE (default), the maximum y-axis depends on the highest count of a variable's answer category. In this case, the y-axis breaks may change, depending on the variable.

upperYlim: Uses a pre-defined upper limit for the y-axis. Overrides the maxYlim parameter.

order: Determines whether categories on x-axis should be order according to the frequencies or not. Default is "none", so categories are not ordered by frequency. Use "asc" or "desc" for sorting categories ascending or descending in relation to the frequencies.

type: Specifies the type of distribution plot that will be plotted.
- "bar", "bars" or "b" for simple bars (the default setting).
- "dots" or "dot" for a dot plot.
- "h", "hist" or "histogram" for a histogram.
- "line", "lines" or "l" for a histogram with filled area with line.
- "dens", "d" or "density" for a density plot.
- "box", "boxplot" or "boxplots" for box plots.
- "v" or "violin" for violin plots.

axisLabels.x: Labels for the x-axis breaks. Example: axisLabels.x=c("Label1", "Label2", "Label3"). Note: If you use the sji.SPSS function and the sji.getValueLabels function, you receive a list object with label string. The labels may also be passed as list object. They will be unlisted and converted to character vector automatically.

interactionVarLabels: Labels for the x-axis breaks when having interaction variables included. These labels replace the axisLabels.x. Only applies, when using box or violin plots (i.e. "type" is "box" or "violin") and interactionVar is not NULL. Example: See axisLabels.x.

axisLabelAngle.x: Angle for axis-labels.

axisLabelSize: The size of axis labels of both x and y axis. Default is 1.1, recommended values range between 0.5 and 3.0.

valueLabelSize: The size of value labels in the diagram. Default is 4, recommended values range between 2 and 8.

breakTitleAt: Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt: Determines how many chars of the labels are displayed in one line and when a line break is inserted into the axis labels.
gridBreaksAt  Sets the breaks on the y axis, i.e. at every n'th position a major grid is being printed.

barWidth   Width of bars. Default is 0.6, recommended values range from 0.2 to 2.0

dotSize    The size of dots in case of dot-plots (type="dots").

innerBoxPlotWidth  The width of the inner box plot that is plotted inside of violin plots. Only applies if type is "violin". Default value is 0.15

innerBoxPlotDotSize  Size of mean dot inside a violin plot. Applies only when type is set to "violin".

barColor   User defined color for bars. If not specified, a default blue color palette will be used for the bar charts.

barAlpha   Specify the transparency (alpha value) of bars.

axisLabelColor   User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

borderColor   User defined color of whole diagram border (panel border).

axisColor   User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

barOutline   If TRUE, each bar gets a colored outline. Default is FALSE.

barOutlineSize   The size of the bar outlines. Only applies if barOutline is TRUE. Default is 0.2

outlineColor   The color of the bar outline. Only applies, if barOutline is TRUE.

majorGridColor   Specifies the color of the major grid lines of the diagram background.

minorGridColor   Specifies the color of the minor grid lines of the diagram background.

hideGrid.x   If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y   If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

showValueLabels   Whether counts and percentage values should be plotted to each bar. Default is TRUE.

showCountValues   If TRUE (default), count values are be plotted to each bar. If FALSE, count values are removed.

showPercentageValues   If TRUE (default), percentage values are be plotted to each bar, if FALSE, percentage-values are removed.

showAxisLabels.x   Whether x axis labels (category names) should be shown or not.

showAxisLabels.y   Whether y axis labels (count values) should be shown or not.

showTickMarks   Whether tick marks of axes should be shown or not.

showMeanIntercept   Whether a vertical line in histograms is drawn to indicate the mean value of the count variables. Only applies to histogram-charts.

showMeanValue   If TRUE (default value), the mean value is printed to the vertical line that indicates the mean value of the count variables. Only applies to histogram-charts.
showStandardDeviation
   If TRUE, the standard deviation is annotated as shaded rectangle around the mean intercept line. Only applies to histogram-charts.

meanInterceptLineType
   The linetype of the mean intercept line. Only applies to histogram-charts and when showMeanIntercept is TRUE.

meanInterceptLinesize
   The size of the mean intercept line. Only applies to histogram-charts and when showMeanIntercept is TRUE.

showNormalCurve
   If TRUE, a normal curve, which is adjusted to the data, is plotted over the histogram or density plot. Default is FALSE. Only applies when histograms or density plots are plotted (see type).

showStandardNormalCurve
   If TRUE, a normal curve, which is not adjusted to the data (thus representing a "true" standard normal curve, which is, however, still just an approximation), is plotted over the histogram or density plot. Default is FALSE. Only applies when histograms or density plots are plotted (see type).

adjustNormalCurve.x
   If TRUE and showStandardNormalCurve is also TRUE, the x-axis-start of the standard normal curve starts with the x-axis limits of the graph. This is only necessary, if minimum value of varCount is larger than 0 or 1.

normalCurveColor
   Specify the color of the normal curve line. Only applies if showNormalCurve is TRUE.

normalCurveSize
   Specify the size of the normal curve line. Only applies if showNormalCurve is TRUE.

normalCurveAlpha
   Specify the transparency (alpha value) of the normal curve. Only applies if showNormalCurve is TRUE.

valueLabelColor
   The color of the value labels (numbers) inside the diagram.

axisTitle.x
   A label for the x axis. Useful when plotting histograms with metric scales where no category labels are assigned to the x axis. Use "auto" to automatically detect variable names that will be used as title (see sjisetVariableLabels) for details).

axisTitle.y
   A label for the y axis. Useful when plotting histograms with metric scales where no category labels are assigned to the y axis.

axisTitleColor
   The color of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

axisTitleSize
   The size of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels. Default is 1.3.

hist.skipZeros
   If TRUE, zero counts (categories with no answer) in varCount are omitted when drawing histograms, and the mapping is changed to stat_bin. Only applies to histograms (see type). Use this parameter to get identical results to the default
qplot or `geom_histogram` histogram plots of ggplot. You may need to adjust the `barWidth` parameter for better visual results (which, by ggplot-default, is $1/30$ of the x-axis-range).

`startAxisAt` Determines the first value on the x-axis. By default, this value is set to "auto", i.e. the value range on the x axis starts with the lowest value of `varCount`. If you set `startAxisAt` to 1, you may have zero counts if the lowest value of `varCount` is larger than 1 and hence no bars plotted for these values in such cases.

`autoGroupAt` A value indicating at which length of unique values of `varCount` the variable is automatically grouped into smaller units (see `sju.groupVar`). If `varCount` has large numbers of unique values, too many bars for the graph have to be plotted. Hence it's recommended to group such variables. For example, if `autoGroupAt` is 50, i.e. if `varCount` has 50 and more unique values it will be grouped using `sju.groupVar` with `groupsize="auto"` parameter. By default, the maximum group count is 30. However, if `autoGroupAt` is less than 30, `autoGroupAt` groups are built. Default value for `autoGroupAt` is NULL, i.e. auto-grouping is off.

`theme` Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.

- Use "bw" for a white background with gray grids
- "classic" for a classic theme (black border, no grids)
- "minimal" for a minimalistic theme (no border, gray grids) or
- "none" for no borders, grids and ticks.

`flipCoordinates` If TRUE, the x and y axis are swapped. Default is FALSE.

`na.rm` If TRUE, missings are not included in the frequency calculation and diagram plot.

`printPlot` If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

**Value**

(Invisibly) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

**References**


**See Also**

`sjt.frq`

**Examples**

```r
# --------------
# boxplot
# --------------
```
```r
sjp.frq(ChickWeight$weight, type="box")

# histogram
sjp.frq(discoveries, type="hist", showMeanIntercept=TRUE)
# histogram with minimal theme and w/o labels
sjp.frq(discoveries, type="hist", showMeanIntercept=TRUE, theme="minimal", minorGridColor="white", showTickMarks=FALSE, hideGrid.x=TRUE, showValueLabels=FALSE)

# violin plot
sjp.frq(ChickWeight$weight, type="v")

# bar plot
sjp.frq(ChickWeight$Diet)

# bar plot with EUROFAMCARE sample dataset
# dataset was importet from an SPSS-file, using:
# efc <- sji.SPSS("efc.sav", enc="UTF-8")
# data(efc)
# efc.val <- sji.getValueLabels(efc)
# efc.var <- sji.getVariableLabels(efc)
# sjp.frq(as.factor(efc$e15relat),
#      title=efc.var[['e15relat']],
#      axisLabels.x=efc.val[['e15relat']],
#      axisLabelAngle.x=90)

# bar plot with EUROFAMCARE sample dataset
# grouped variable
ageGrp <- sju.groupVarVar(efc$e17age)
ageGrpLab <- sju.groupVarVarLabels(efc$e17age)
# sjp.frq(ageGrp,
#      title=efc.var[['e17age']],
#      axisLabels.x=ageGrpLab)
# minimal theme
# sjp.frq(ageGrp,
#      title=efc.var[['e17age']],
#      axisLabels.x=ageGrpLab, theme="minimal", minorGridColor="white", showTickMarks=FALSE, hideGrid.x=TRUE)
```
sjp.glm

Plot odds ratios (forest plots)

Description

Plot odds ratios with confidence intervals as bar chart or dot plot

Usage

sjp.glm(fit, sortOdds = TRUE, title = NULL, titleSize = 1.3, 
titleColor = "black", axisLabels.y = NULL, axisLabelSize = 1.1, 
axisLabelAngle.x = 0, axisLabelAngle.y = 0, axisLabelColor = "gray30", 
axisTitle.x = "Odds Ratios", axisTitleSize = 1.2, 
axisTitleColor = c("#444444"), axisLimits = NULL, breakTitleAt = 50, 
breakLabelsAt = 12, gridBreaksAt = 0.5, transformTicks = FALSE, 
type = "dots", hideErrorBars = FALSE, errorBarWidth = 0, 
errorBarSize = 0.8, errorBarLineType = 1, pointSize = 3, 
colorPalette = "Paired", barColor = NULL, barWidth = 0.3, 
barAlpha = 1, valueLabelColor = "black", valueLabelSize = 4.5, 
valueLabelAlpha = 1, axisColor = NULL, borderColor = NULL,
Arguments

fit The fitted model of a logistic regression (glm-Object).

sortOdds If TRUE (default), the odds ratios are ordered according their OR value from highest first to lowest last. Use FALSE if you don’t want to change the order of the predictors.

title Diagram’s title as string. Example: title=c("my title")

titlesize The size of the plot title. Default is 1.3.

titleColor The color of the plot title. Default is "black".

axisLabels.y Labels of the predictor variables (independent vars, odds) that are used for labelling the axis. Passed as vector of strings. Example: axisLabels.y=c("Label1", "Label2", "Label3")

Note: If you use the sj.stat function and the sj.getPredictor function, you receive a list object with label string. The labels may also be passed as list object. They will be unlisted and converted to character vector automatically.

axisLabelSize The size of value labels in the diagram. Default is 1.1, recommended values range between 0.7 and 3.0

showAxisLabels.y Whether odds names (predictor labels) should be shown or not.

showTickMarks Whether tick marks of axes should be shown or not.

axistitle.x A label ("title") for the x axis.

axistitleColor The color of the x axis label.

axistitleSize The size of the x axis label.

axisLimits Defines the range of the axis where the beta coefficients and their confidence intervals are drawn. By default, the limits range from the lowest confidence interval to the highest one, so the diagram has maximum zoom. Use your own values as 2-value-vector, for instance: limits=c(-0.8,0.8).

axialLabelAngle.x Angle for axis-labels where the odds ratios are printed. Note that due to the coordinate flip, the actual y-axis with odds ratio labels are appearing on the x-axis.

axialLabelAngle.y Angle for axis-labels where the predictor labels (axisLabels.y) are printed. Note that due to the coordinate flip, the actual x-axis with predictor labels are appearing on the y-axis.

breakTitleAt Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.
**breakLabelsAt** Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

**gridBreaksAt** Sets the breaks on the y axis, i.e. at every n'th position a major grid is being printed. Default is 0.5

**transformTicks** if TRUE, the grid bars have exponential distances, i.e. they visually have the same distance from one grid bar to the next. Default is FALSE which means that grids are plotted on every gridBreaksAt's position, thus the grid bars become narrower with higher odds ratio values.

**type** Indicates Whether the Odds Ratios should be plotted as "dots" (aka forest plots, default) or as "bars".

**hideErrorBars** If TRUE, the error bars that indicate the confidence intervals of the odds ratios are not shown. Only applies if parameter type is bars. Default value is FALSE.

**pointSize** The size of the points that indicate the beta-value. Default is 3.

**barColor** A vector with colors for representing the odds values (i.e. points and error bars in case the parameter type is "dots" or the bar charts in case of "bars"). The first color value indicates odds ratio values larger than 1, the second color value indicates odds ratio values lower or equal to 1. Default colors is a blue/red-scheme. You can also use:

- "bw" or "black" for only one colouring in almost black
- "gray", "grey" or "gs" for a grayscale
- "brewer" for colours from the color brewer palette.

If barColors is "brewer", use the colorPalette parameter to specify a palette of the color brewer. Else specify your own color values as vector (e.g. barColors=c("#f00000", "#00ff00")).

**colorPalette** If parameter barColor is brewer, specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.

**barWidth** The width of the bars in bar charts. only applies if parameter type is bars. Default is 0.5

**barAlpha** The alpha value of the bars in bar charts. only applies if parameter type is bars. Default is 1

**axisLabelColor** Colour of the tick labels at the axis (variable names, odds names).

**valueLabelColor** The colour of the odds values. These values are printed above the plots respectively beside the bar charts. default color is "black".

**valueLabelSize** Size of the value labels. Default is 4.5. Recommended Values range from 2 to 8

**valueLabelAlpha** The alpha level (transparancy) of the value labels. Default is 1, use any value from 0 to 1.

**axisColor** User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

**borderColor** User defined color of whole diagram border (panel border).

**barOutline** If TRUE, each bar gets a colored outline. only applies if parameter type is bars. Default is FALSE.
outlineColor  The color of the bar outline. Only applies, if barOutline is set to TRUE. Default is black.
interceptLineType  The linetype of the intercept line (zero point). Default is 2 (dashed line).
interceptLineColor  The color of the intercept line. Default value is "grey70".
errorBarWidth  The width of the error bar ends. Default is 0
errorBarSize  The size (thickness) of the error bars. Default is 0.8
errorBarLineType  The linetype of error bars. Default is 1 (solid line).
majorGridColor  Specifies the color of the major grid lines of the diagram background.
minorGridColor  Specifies the color of the minor grid lines of the diagram background.
hideGrid.x  If TRUE, the x-axis-gridlines are hidden. Default if FALSE.
hideGrid.y  If TRUE, the y-axis-gridlines are hidden. Default if FALSE.
theme  Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.
  • Use "bw" for a white background with gray grids
  • "classic" for a classic theme (black border, no grids)
  • "minimal" for a minimalistic theme (no border, gray grids) or
  • "none" for no borders, grids and ticks.
flipCoordinates  If TRUE (default), predictors are plotted on the left y-axis and estimate values are plotted on the x-axis.
showIntercept  If TRUE, the intercept of the fitted model is also plotted. Default is FALSE. Please note that due to exp-transformation of estimates, the intercept in some cases can not be calculated, thus the function call is interrupted and no plot printed.
showValueLabels  Whether the beta and standardized beta values should be plotted to each dot or not.
labelDigits  The amount of digits for rounding the estimations (see showValueLabels). Default is 2, i.e. estimators have 2 digits after decimal point.
showPValueLabels  Whether the significance levels of each coefficient should be appended to values or not.
showModelSummary  If TRUE (default), a summary of the regression model with Intercept, R-square, F-Test and AIC-value is printed to the lower right corner of the diagram.
printPlot  If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Insisibily) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).
Note

Based on the script from surefoss: http://www.surefoss.org/dataanalysis/plotting-odds-ratios-aka-a-forrestplot

References

• http://strengejacke.wordpress.com/sjplot-r-package/
• http://strengejacke.wordpress.com/2013/03/22/plotting-lm-and-glm-models-with-ggplot-rstats/

See Also

sjp glm ma

Examples

# prepare dichotomous dependent variable
y <- ifelse(swiss$Fertility<median(swiss$Fertility), 0, 1)

# fit model
fitOR <- glm(y ~ swiss$Education + swiss$Examination + swiss$Infant.Mortality + swiss$Catholic,
             family=binomial(link="logit"))

# print Odds Ratios as dots
sjp glm(fitOR)

# print Odds Ratios as bars
sjp glm(fitOR, type="bars")

# -------------------------------
# Predictors for negative impact
# of care. Data from the EUROFAMCARE
# sample dataset
# -------------------------------
data(efc)
# retrieve predictor variable labels
labs <- sji.getVariableLabels(efc)
predlab <- c(labs[['c161sex']],
            labs[['e42dep']],
            paste0(labs[['c172code']], " (mid)"),
            paste0(labs[['c172code']], " (high)"))

# create binary response
y <- ifelse(efc$neg_c_7<median(efc$neg_c_7), 0, 1)
# create dummy variables for educational status
edu.mid <- ifelse(efc$c172code==2, 1, 0)
edu.high <- ifelse(efc$c172code==3, 1, 0)
# create data frame for fitted model
df <- na.omit(as.data.frame(cbind(y,
                             as.factor(efc$c161sex),
                             as.factor(efc$e42dep),
                             as.factor(edi.mid),
                             as.factor(efc$neg_c_7))))
as.factor(edu.high)))

# fit model
fit <- glm(y ~ ., data=df, family=binomial(link="logit"))
# plot odds
sjp.glm(fit, title=labels[['neg_c_7']], axisLabels.y=predlab)

sjp.glm.ma

Plot model assumptions of glm's

Description

Plots model assumptions of generalized linear models to verify if generalized linear regression is applicable.

Usage

sjp.glm.ma(logreg, showOriginalModelOnly = TRUE)

Arguments

logreg          a fitted glm-model
showOriginalModelOnly

if TRUE (default), only the model assumptions of the fitted model logreg are plotted. if FALSE, the model assumptions of an updated model where outliers are automatically excluded are also plotted.

Value

an updated fitted generalized linear model where outliers are dropped out.

See Also

sjp.glm

Examples

# prepare dichotomous dependent variable
y <- ifelse(swiss$Fertility<median(swiss$Fertility), 0, 1)

# fit model
fitOR <- glm(y ~ swiss$Education + swiss$Examination + swiss$Infant.Mortality + swiss$Catholic,
             family=binomial(link="logit"))

# plot model assumptions
sjp.glm.ma(fitOR)
sjp.glmm

Plot odds ratios (forest plots) of multiple fitted glm's

Description

Plot odds ratios (forest plots) of multiple fitted glm's with confidence intervals in one plot.

Usage

```r
sjp.glmm(..., title = NULL, titleSize = 1.3, titleColor = "black",
        labelDependentVariables = NULL, legendDepVarTitle = "Dependent Variables",
        legendPValTitle = "p-level", stringModel = "Model",
        axisLabelSize = 1.1, axisLabelAngle.x = 0, axisLabelAngle.y = 0,
        axisLabelColor = "gray30", axisTitle.x = "Odds Ratios",
        axisTitleSize = 1.2, axisTitleColor = c("#444444");
        axisLimits = NULL,
        breakTitleAt = 50, breakLabelsAt = 12, breakLegendAt = 20,
        gridBreaksAt = 0.5, transformTicks = FALSE, errorBarWidth = 0,
        errorBarSize = 0.5, errorBarLineType = 1, pointSize = 3,
        modelPlotSpace = 0.4, colorPalette = "Paired", modelColors = NULL,
        valueLabelColor = "black", valueLabelSize = 4, valueLabelAlpha = 1,
        nsAlpha = 1, usePShapes = FALSE, axisColor = NULL, borderColor = NULL,
        interceptLineType = 2, interceptLineColor = "grey70",
        majorGridColor = NULL, minorGridColor = NULL, hideGrid.y = FALSE,
        hideGrid.x = FALSE, theme = NULL, flipCoordinates = TRUE,
        legendPos = "right", legendSize = 1, legendBorderColor = "white",
        legendBackColor = "white", showIntercept = FALSE,
        showAxisLabels.y = TRUE, showTickMarks = TRUE, showValueLabels = TRUE,
        labelDigits = 2, showPValueLabels = TRUE, printPlot = TRUE)
```

Arguments

- `...` One or more fitted glm-objects.
- `title` Diagram’s title as string. Example: `title="my title"`
- `titleSize` The size of the plot title. Default is 1.3.
- `titleColor` The color of the plot title. Default is "black".
- `labelDependentVariables` Labels of the dependent variables of all fitted models which have been used as first parameter(s), provided as char vector.
- `legendDepVarTitle` A character vector used for the title of the dependent variable’s legend. Default is "Dependent Variables".
- `legendPValTitle` A character vector used for the title of the significance level’s legend. Default is "p-level". Only applies if `usePShapes` is TRUE.
**String constant used as legend text for the model names in case no labels for the dependent variables are provided (see labelDependentVariables). Default is "Model".**

**Labels of the predictor variables (independent vars, odds) that are used for labelling the axis. Passed as vector of strings. Example: axisLabels.y=c("Label1", "Label2", "Label3"). Note: If you use the sji.SPSS function and the sji.getValueLabels function, you receive a list object with label strings. The labels may also be passed as list object. They will be unlisted and converted to character vector automatically.**

**The size of value labels in the diagram. Default is 1.1, recommended values range between 0.7 and 3.0**

**Whether odds names (predictor labels) should be shown or not.**

**Whether tick marks of axes should be shown or not.**

**A label ("title") for the x axis.**

**The color of the x axis label.**

**The size of the x axis label.**

**Defines the range of the axis where the beta coefficients and their confidence intervals are drawn. By default, the limits range from the lowest confidence interval to the highest one, so the diagram has maximum zoom. Use your own values as 2-value-vector, for instance: limits=c(-0.8,0.8).**

**Angle for axis-labels where the odds ratios are printed. Note that due to the coordinate flip, the actual y-axis with odds ratios are appearing on the x-axis.**

**Angle for axis-labels where the predictor labels (axisLabels.y) are printed. Note that due to the coordinate flip, the actual x-axis with predictor labels are appearing on the y-axis.**

**Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title**

**Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted**

**Wordwrap for legend, i.e. names of the dependent variables of each fitted model. See parameter labelDependentVariables. Determines how many chars of each dependent variable name is displayed in one line in the legend and when a line break is inserted**

**Sets the breaks on the y axis, i.e. at every n'th position a major grid is being printed. Default is 0.5**

**if TRUE, the grid bars have exponential distances, i.e. they visually have the same distance from one grid bar to the next. Default is FALSE which means that grids are plotted on every gridBreaksAt's position, thus the grid bars become narrower with higher odds ratio values.**

**The size of the points that indicate the beta-value. Default is 3.**

**Defines the space between the dots and error bars of the plotted fitted models. Default is 0.3.**
modelColors

A vector with colors for representing the odds values (i.e. points and error bars) of the different fitted models. Thus, the length of this vector must be equal to the length of supplied fitted models, so each model is represented by its own color. You can use:
- "bw" or "black" for only one colouring in almost black
- "gray", "grey" or "gs" for a grayscale
- "brewer" for colours from the color brewer palette.

If modelColors is "brewer", use the colorPalette parameter to specify a palette of the color brewer. Else specify your own color values as vector (e.g. modelColors=c("#f00000", "#00ff00")).

colorPalette

If parameter modelColors is brewer, specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.

axisLabelColor

Colour of the tick labels at the axis (variable names, odds names).

valueLabelColor

The colour of the odds values. These values are printed above the plots respectively beside the bar charts. default color is "black".

valueLabelSize

Size of the value labels. Default is 4. Recommended Values range from 2 to 8

valueLabelAlpha

The alpha level (transparency) of the value labels. Default is 1, use any value from 0 to 1.

nsAlpha

The alpha level (transparency) of non significant predictors. Points and error bars are affected by this value and plotted with a slight transparency. Default is 1.

usePShapes

If TRUE, significant levels are distinguished by different point shapes and a related legend is plotted. Default is FALSE.

axisColor

User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

borderColor

User defined color of whole diagram border (panel border).

interceptLineType

The linetype of the intercept line (zero point). Default is 2 (dashed line).

interceptLineColor

The color of the intercept line. Default value is "grey70".

errorBarWidth

The width of the error bar ends. Default is 0

errorBarSize

The size (thickness) of the error bars. Default is 0.5

errorBarLineType

The linetype of error bars. Default is 1 (solid line).

majorGridColor

Specifies the color of the major grid lines of the diagram background.

minorGridColor

Specifies the color of the minor grid lines of the diagram background.

hideGrid.x

If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y

If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

theme

Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.
- Use "bw" for a white background with gray grids
• "classic" for a classic theme (black border, no grids)
• "minimal" for a minimalistic theme (no border, gray grids) or
• "none" for no borders, grids and ticks.

flipCoordinates
If TRUE (default), predictors are plotted on the left y-axis and estimate values are plotted on the x-axis.

legendPos
The position of the legend, if a legend is drawn. Use "bottom", "top", "left" or "right" to position the legend above, below, on the left or right side of the diagram. Right positioning is default.

legendSize
The text size of the legend. Default is 1. Relative size, so recommended values are from 0.3 to 2.5

legendBorderColor
Color of the legend’s border. Default is "white", so no visible border is drawn.

legendBackgroundColor
Fill color of the legend’s background. Default is "white", so no visible background is drawn.

showIntercept
If TRUE, the intercept of the fitted model is also plotted. Default is FALSE. Please note that due to exp-transformation of estimates, the intercept in some cases cannot be calculated, thus the function call is interrupted and no plot printed.

showValueLabels
Whether the beta and standardized beta values should be plotted to each dot or not.

labelDigits
The amount of digits for rounding the estimations (see showValueLabels). Default is 2, i.e. estimators have 2 digits after decimal point.

showPValueLabels
Whether the significance levels of each coefficient should be appended to values or not.

printPlot
If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value
(Invisibly) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

References
• http://strengejacke.wordpress.com/sjplot-r-package/
• http://strengejacke.wordpress.com/2014/01/29/comparing-multiple-glm-in-one-graph-rstats/

See Also
sjp.glm
sjt.glm
sjp.glm.ma
sjp.lmm
Examples

```r
# prepare dummy variables for binary logistic regression
y1 <- ifelse(swiss$Fertility < median(swiss$Fertility), 0, 1)
y2 <- ifelse(swiss$Infant.Mortality < median(swiss$Infant.Mortality), 0, 1)
y3 <- ifelse(swiss$Agriculture < median(swiss$Agriculture), 0, 1)

# Now fit the models. Note that all models share the same predictors
# and only differ in their dependent variable (y1, y2 and y3)
fitOR1 <- glm(y1 ~ swiss$Education + swiss$Examination + swiss$Catholic,
              family = binomial(link = "logit"))
fitOR2 <- glm(y2 ~ swiss$Education + swiss$Examination + swiss$Catholic,
              family = binomial(link = "logit"))
fitOR3 <- glm(y3 ~ swiss$Education + swiss$Examination + swiss$Catholic,
              family = binomial(link = "logit"))

# plot multiple models
sjp.glmm(fitOR1, fitOR2, fitOR3)

# plot multiple models with legend labels and point shapes instead of value labels
sjp.glmm(FITOR1, fitOR2, fitOR3,
labelDependentVariables = c("Fertility", "Infant Mortality", "Agriculture"),
showValueLabels = FALSE,
showPValueLabels = FALSE,
usePShapes = TRUE,
nsAlpha = 0.2)
```

---

**sjp.grpfrq**

Plot grouped or stacked frequencies

**Description**

Plot grouped or stacked frequencies of variables as bar/dor graphs, box or violin plots, histograms etc. using ggplot.

**Usage**

```r
sjp.grpfrq(varCount, varGroup, weightBy = NULL, weightByTitleString = NULL,
interactionVar = NULL, type = "bars", dotSize = 4, hideLegend = FALSE,
maxYlim = FALSE, upperYlim = NULL, useFacetGrid = FALSE, title = NULL,
titleSize = 1.3, titleColor = "black", legendTitle = NULL,
axisLabels.x = NULL, axisLabelSize = 1.1, axisLabelColor = "gray30",
axisLabelAngle.x = 0, interactionVarLabels = NULL, legendLabels = NULL,
valueLabelSize = 4, valueLabelColor = "black", breakTitleAt = 50,
breakLabelsAt = 12, breakLegendTitleAt = 20, breakLegendLabelsAt = 20,
gridBreaksAt = NULL, barPosition = "dodge", barWidth = 0.6,
barSpace = 0.1, barColor = NULL, barAlpha = 1,
innerBoxPlotWidth = 0.15, innerBoxPlotDotSize = 3,
colorPalette = "GnBu", lineType = 1, lineSize = 1, lineAlpha = 1,
```
smoothLines = FALSE, bordercolor = NULL, axiscolor = NULL, barOutline = FALSE, outlineColor = "black", majorGridColor = NULL, minorGridColor = NULL, hideGrid.x = FALSE, hideGrid.y = FALSE, showValueLabels = TRUE, showCountValues = TRUE, showPercentageValues = TRUE, showAxisLabels.x = TRUE, showAxisLabels.y = TRUE, showTickMarks = TRUE, showPlotAnnotation = TRUE, showMeanIntercept = FALSE, showMeanValue = TRUE, showStandardDeviation = FALSE, showTableSummary = TRUE, showGroupCount = FALSE, tableSummaryPos = "r", meanInterceptLineType = 2, meanInterceptLineWidth = 0.5, axisTitle.x = NULL, axisTitle.y = NULL, axisTitleColor = "black", axisTitleSize = 1.3, autoGroupAt = NULL, startAxisAt = "auto", theme = NULL, legendPos = "right", legendSize = 1, legendBorderColor = "white", legendBackColor = "white", flipCoordinates = FALSE, na.rm = TRUE, printPlot = TRUE)

Arguments

varCount The variable which frequencies should be plotted. The counts of this variable are along the y-axis, the variable’s categories on the x-axis.

varGroup the grouping variable, where each value represents a single bar chart within each category of the varCount variable.

weightBy A weight factor that will be applied to weight all cases from varCount.

weightByTitleString If a weight factor is supplied via the parameter weightBy, the diagram’s title may indicate this with a remark. Default is NULL, so the diagram’s title will not be modified when cases are weighted. Use a string as parameter, e.g.: weightByTitleString=" (weighted)".

interactionVar An interaction variable which can be used for box plots. Divides each category indicated by varGroup into the factors of interactionVar, so that each category of varGroup is subgrouped into interactionVar’s categories. Only applies when parameter type is box or violin (resp. their alternative strings like "boxplot", "boxplots" or "v").

barPosition Indicates whether bars should be positioned side-by-side (default, or use "dodge" as parameter) or stacked (use "stack" as parameter). If type is "histogram", you can use either "dodge" (default value), which displays the bars side-by-side, or "identity", which results in overlaying bars. In the latter case, it’s recommended to adjust the barAlpha value.

type The plot type. May be one of the following:

- "b", "bar", "bars" (default) for bar charts
- "l", "line", "lines" for line diagram
- "d", "dot", "dots" for dot plots
- "h", "hist", "histogram" for grouped histograms
- "b", "boxplot", "boxplots" for box plots
- "v", "violin" for violin box plots

dotSize Size of dots. Applies only when type is set to "dots".
hideLegend Indicates whether legend (guide) should be shown or not.

maxYlim Indicates how to calculate the maximum limit of the y-axis. If TRUE, the upper y-limit corresponds to the amount of cases, i.e. y-axis for each plot of a data base are the same. If FALSE (default), the maximum y-axis depends on the highest count of a variable’s answer category. In this case, the y-axis breaks may change, depending on the variable.

upperYlim Uses a pre-defined upper limit for the y-axis. Overrides the maxYlim parameter.

useFacetGrid TRUE when bar charts should be plotted as facet grids instead of integrated single bar charts. Ideal for larger amount of groups. This parameter wraps a single panel into varGrpup amount of panels, i.e. each group is represented within a new panel.

title Title of the diagram, plotted above the whole diagram panel. Use “auto” to automatically detect variable names that will be used as title (see sj.setVariableLabels) for details).

titleSize The size of the plot title. Default is 1.3.

titleColor The color of the plot title. Default is "black".

legendTitle Title of the diagram’s legend.

axisLabels.x Labels for the x-axis breaks. Passed as vector of strings. Note: This parameter is not necessary when data was either imported with sj.SPSS or has named factor levels (see examples below). Else, specify parameter like this: axisLabels.x=c("Label1", "Label2", "Label3"). Note: If you use the sj.SPSS function and the sj.getVariableLabels function, you receive a list object with label string. The labels may also be passed as list object. They will be unlisted and converted to character vector automatically.

interactionVarLabels Labels for the x-axis breaks when having interaction variables included. These labels replace the axisLabels.x. Only applies, when using box or violin plots (i.e. "type" is "box" or "violin") and interactionVar is not NULL. Example: See axisLabels.x.

legendLabels Labels for the guide/legend. Example: See axisLabels.x.

axisLabelSize The size of axis labels of both x and y axis. Default is 1.1, recommended values range between 0.5 and 3.0

valueLabelSize The size of value labels in the diagram. Default is 4, recommended values range between 2 and 8

axisLabelAngle.x Angle for axis-labels.

breakTitleAt Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

breakLegendTitleAt Wordwrap for diagram legend title. Determines how many chars of the legend’s title are displayed in one line and when a line break is inserted.
breakLegendLabelsAt
  Wordwrap for diagram legend labels. Determines how many chars of the legend labels are displayed in one line and when a line break is inserted.

gridBreaksAt
  Sets the breaks on the y axis, i.e. at every n’th position a major grid is being printed.

barWidth
  Width of bars. Recommended values for this parameter are from 0.4 to 1.5

innerBoxPlotWidth
  The width of the inner box plot that is plotted inside of violin plots. Only applies if type is "violin". Default value is 0.15

innerBoxPlotDotSize
  Size of mean dot inside a violin plot. Applies only when type is set to "violin".

barSpace
  Spacing between bars. Default value is 0.1. If 0 is used, the grouped bars are stuck together and have no space in between. Recommended values for this parameter are from 0 to 0.5

barColor
  User defined color for bars.
  • If not specified (NULL), a default red-green-yellow color palette will be used for the bar charts.
  • If barColor is "gs", a greyscale will be used.
  • If barColor is "bw", a monochrome white filling will be used.
  • If barColor is "brewer", use the colorPalette parameter to specify a palette of the color brewer.

  Else specify your own color values as vector (e.g. barColor=c("#f00000", "#00ff00", "#0080ff")).

colorPalette
  If barColor is "brewer", specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.

barAlpha
  Specify the transparency (alpha value) of bars.

lineType
  The linetype when using line diagrams. Only applies, when parameter type is set to "lines".

lineSize
  The size of lines in a line diagram. Only applies, when parameter type is set to "lines".

lineAlpha
  The alpha value of lines in a line diagram. Only applies, when parameter type is set to "lines".

smoothLines
  Prints a smooth line curve. Only applies, when parameter type is set to "lines".

axisLabelColor
  User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

borderColor
  User defined color of whole diagram border (panel border).

axisColor
  User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

barOutline
  If TRUE, each bar gets a colored outline. Default is FALSE.

outlineColor
  The color of the bar outline. Only applies, if barOutline is set to TRUE.

majorGridColor
  Specifies the color of the major grid lines of the diagram background.

minorGridColor
  Specifies the color of the minor grid lines of the diagram background.

hideGrid.x
  If TRUE, the x-axis-gridlines are hidden. Default is FALSE.
hideGrid.y

If TRUE, the y-axis-gridlines are hidden. Default is FALSE.

showValueLabels

Whether counts and percentage values should be plotted to each bar. Default is TRUE.

showCountValues

If TRUE (default), count values are be plotted to each bar. If FALSE, count values are removed.

showPercentageValues

If TRUE (default), percentage values are be plotted to each bar, if FALSE, percentage-values are removed.

showAxisLabels.x

Whether x axis labels (category names) should be shown or not.

showAxisLabels.y

Whether y axis labels (count values) should be shown or not.

showTickMarks

Whether tick marks of axes should be shown or not.

showPlotAnnotation

If TRUE, the groups of dots in a dot-plot are highlighted with a shaded rectangle.

showMeanIntercept

if TRUE, a vertical line in histograms is drawn to indicate the mean value of the count variables. Only applies to histogram-charts.

showMeanValue

If TRUE (default value), the mean value is printed to the vertical line that indicates the mean value of the count variables. Only applies to histogram-charts.

showStandardDeviation

If TRUE, the standard deviation is annotated as shaded rectangle around the mean intercept line. Only applies to histogram-charts. The shaded rectangles have borders in the group colors, so it’s easier to see which shaded area belongs to which mean value resp. group

showTableSummary

If TRUE (default), a summary of the cross tabulation with N, Chi-square (see chisq.test), df, Cramer’s V or Phi-value and p-value is printed to the upper right corner of the diagram. If a cell contains expected values lower than five (or lower than 10 if df is 1), the Fisher’s exact test (see fisher.test) is computed instead of Chi-square test. If the table’s matrix is larger than 2x2, Fisher’s exact test with Monte Carlo simulation is computed. Only applies to bar-charts or dot-plots, i.e. when parameter type is either “bars” or “dots”.

showGroupCount

if TRUE, the count within each group is added to the category labels (e.g. ”Cat 1 (n=87)”). Default value is FALSE.

tableSummaryPos

Position of the model summary which is printed when showTableSummary is TRUE. Default is “r”, i.e. it’s printed to the upper right corner. Use “l” for upper left corner.

meanInterceptLineType

The linetype of the mean intercept line. Only applies to histogram-charts and when showMeanIntercept is TRUE.

meanInterceptLineSize

The size of the mean intercept line. Only applies to histogram-charts and when showMeanIntercept is TRUE.
valueLabelColor
The color of the value labels (numbers) inside the diagram.

axisTitle.x
A label for the x axis. Useful when plotting histograms with metric scales where no category labels are assigned to the x axis. Use "auto" to automatically detect variable names that will be used as title (see sj1.setVariableLabels) for details.

axisTitle.y
A label for the y axis. Useful when plotting histograms with metric scales where no category labels are assigned to the y axis.

axisTitleColor
The color of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

axisTitleSize
The size of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

autoGroupAt
A value indicating at which length of unique values of varCount the variable is automatically grouped into smaller units (see sj1.groupVar). If varCount has large numbers of unique values, too many bars for the graph have to be plotted. Hence it's recommended to group such variables. For example, if autoGroupAt is 50, i.e. if varCount has 50 and more unique values it will be grouped using sj1.groupVar with groupsize="auto" parameter. By default, the maximum group count is 30. However, if autoGroupAt is less than 30, autoGroupAt groups are built. Default value for autoGroupAt is NULL, i.e. auto-grouping is off.

startAxisAt
Determines the first value on the x-axis. By default, this value is set to "auto", i.e. the value range on the x axis starts with the lowest value of varCount. If you set startAxisAt to 1, you may have zero counts if the lowest value of varCount is larger than 1 and hence no bars plotted for these values in such cases.

theme
Specifies the diagram's background theme. Default (parameter NULL) is a gray background with white grids.

• Use "bw" for a white background with gray grids
• "classic" for a classic theme (black border, no grids)
• "minimal" for a minimalistic theme (no borders, gray grids) or
• "none" for no borders, grids and ticks.

legendPos
The position of the legend, if a legend is drawn. Use "bottom", "top", "left" or "right" to position the legend above, below, on the left or right side of the diagram. Right positioning is default.

legendSize
The text size of the legend. Default is 1. Relative size, so recommended values are from 0.3 to 2.5

legendBorderColor
Color of the legend's border. Default is "white", so no visible border is drawn.

legendBackgroundColor
Fill color of the legend's background. Default is "white", so no visible background is drawn.

flipCoordinates
If TRUE, the x and y axis are swapped.

na.rm
If TRUE, missings are not included in the frequency calculation and diagram plot.

printPlot
If TRUE (default), plots the results as graph. Use FALSE if you don't want to plot any graphs. In either case, the ggplot-object will be returned as value.
Value

(Insisibly) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

References

- http://strengejacke.wordpress.com/sjplot-r-package/

- http://strengejacke.wordpress.com/2013/03/05/easily-plotting-grouped-bars-with-ggplot

Examples

```r
# histogram plot
sjp.grpfrq(discoveries, sample(1:3, length(discoveries), replace=TRUE), type="hist",
        showValueLabels=FALSE, showMeanIntercept=TRUE)

# histogram with EUROFAMCARE sample dataset
data(efc)
ecf.val <- sji.getValueLabels(efc)
ecf.var <- sji.getVariableLabels(efc)
sjp.grpfrq(ecf$e17age,
ecf$e16sex,
title=ecf.var['e17age'],
legendTitle=ecf.var['e16sex'],
type="hist",
showValueLabels=FALSE)

# boxplot
sjp.grpfrq(ChickWeight$weight, as.numeric(ChickWeight$Diet), type="box")

# violin plot
sjp.grpfrq(sample(1:4, length(PlantGrowth$group), replace=TRUE),
          PlantGrowth$group, type="v")

# grouped bars
sjp.grpfrq(sample(1:2, length(PlantGrowth$group), replace=TRUE),
          PlantGrowth$group, barSpace=0.2)

# grouped bars with EUROFAMCARE sample dataset
# dataset was importet from an SPSS-file, using:
# efc <- sji.SPSS("efc.sav", enc="UTF-8")
data(efc)
# ------------------------------------------
# auto-detection of value labels and variable names
# ------------------------------------------
ecf.var <- sji.getVariableLabels(efc)
ecf <- sji.setVariableLabels(ecf, ecf.var)
# grouped bars using necessary y-limit
sjp.grpfrq(ecf$e42dep, ecf$e16sex, title="auto")
```
# grouped bars using the maximum y-limit
sjp.grpfrq(efc$e42dep, 
  efc$e16sex, 
  title=ecf.var['e42dep'], 
  axisLabels.x=ecf.val[['e42dep']], # not needed for SPSS-data sets 
  legendTitle=ecf.var['e16sex'], 
  legendLabels=ecf.val[['e16sex']]), # not needed for SPSS-data sets 
  maxYlim=TRUE)

# box plots with interaction variable
sjp.grpfrq(efc$e7age, 
  efc$e42dep, 
  interactionVar=efc$e16sex, 
  title=paste(ecf.var['e7age'], "by", ecf.var['e42dep'], "and", ecf.var['e16sex']), 
  axisLabels.x=ecf.val[['e7age']], 
  interactionVarLabels=ecf.val[['e16sex']], 
  legendTitle=ecf.var['e42dep'], 
  legendLabels=ecf.val[['e42dep']], 
  type="box")

# Grouped bar plot ranging from 1 to 28 (though scale starts with 7)
sjp.grpfrq(efc$neg_c_7, efc$e42dep, showValueLabels=FALSE, startAxisAt=1) 
# Same grouped bar plot ranging from 7 to 28
sjp.grpfrq(efc$neg_c_7, efc$e42dep, showValueLabels=FALSE)

---

sjp.likert

Plot likert scales as centered stacked bars

---

Description

Plot likert scales as centered stacked bars. "Neutral" categories (odd-numbered categories) will be removed from the plot.

Usage

sjp.likert(items, legendLabels = NULL, orderBy = NULL, reverseOrder = FALSE, dropLevels = NULL, weightBy = NULL, 
  weightByTitleString = NULL, hideLegend = FALSE, title = NULL, 
  titleSize = 1.3, titleColor = "black", legendTitle = NULL, 
  includeN = TRUE, axisLabels.y = NULL, axisLabelSize = 1.1, 
  axisLabelAngle.x = 0, axisLabelColor = "gray30", valueLabelSize = 4, 
  valueLabelColor = "black", breakTitleAt = 50, breakLabelsAt = 30, 
  breakLegendTitleAt = 30, breakLegendLabelsAt = 28, gridRange = 1, 
  gridBreaksAt = 0.2, diagramMargins = TRUE, barWidth = 0.5, 
  barColor = NULL, colorPalette = "GnBu", barAlpha = 1, 
  borderColor = NULL, axisColor = NULL, barOutline = FALSE, 
  outlineColor = "black", majorGridColor = NULL, minorGridColor = NULL, 
  hideGrid.x = FALSE, hideGrid.y = FALSE, axisTitle.x = NULL, 
  axisTitle.y = NULL, axisTitleColor = "black", axisTitleSize = 1.3,
Arguments

items A data frame with each column representing one likert-item.

legendLabels A list or vector of strings that indicate the likert-scale-categories and which appear as legend text.

orderBy Indicates whether the items should be ordered by total sum of positive or negative answers. Use "pos" to order descending by sum of positive answers, "neg" for sorting descending negative answers or NULL (default) for no sorting.

reverseOrder If TRUE, the item order (positive/negative) are reversed. Default is FALSE.

dropLevels Indicates specific factor levels that should be dropped from the items before the likert scale is plotted. Default is NULL, hence all factor levels are included. Example to drop first factor level: dropLevels=c(1).

weightBy A weight factor that will be applied to weight all cases from items.

weightByTitleString If a weight factor is supplied via the parameter weightBy, the diagram’s title may indicate this with a remark. Default is NULL, so the diagram’s title will not be modified when cases are weighted. Use a string as parameter, e.g.: weightByTitleString=" (weighted)"

hideLegend Indicates whether legend (guide) should be shown or not.

title Title of the diagram, plotted above the whole diagram panel.

titleSize The size of the plot title. Default is 1.3.

titleColor The color of the plot title. Default is "black".

legendTitle Title of the diagram’s legend.

includeN If TRUE (default), the N of each item is included into axis labels.

axisLabels.y Labels for the y-axis (the labels of the items). These parameters must be passed as list! Example: axisLabels.y=list(c("Q1", "Q2", "Q3")) Axis labels will automatically be detected, when they have a "variable.label" attribute (see sjisetVariableLabels for details).

axisLabelSize The size of category labels at the axes. Default is 1.1, recommended values range between 0.5 and 3.0

axisLabelAngle.x Angle for axis-labels.

axisLabelColor User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

valueLabelSize The size of value labels in the diagram. Default is 4, recommended values range between 2 and 8
valueLabelColor
The color of value labels in the diagram. Default is black.

breakTitleAt
Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt
Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

breakLegendTitleAt
Wordwrap for diagram legend title. Determines how many chars of the legend’s title are displayed in one line and when a line break is inserted.

breakLegendLabelsAt
Wordwrap for diagram legend labels. Determines how many chars of the legend labels are displayed in one line and when a line break is inserted.

gridRange
Sets the limit of the x-axis-range. Default is 1, so the x-scale ranges from zero to 100 percent on both sides from the center. Valid values range from 0 (0 percent) to 1 (100 percent).

gridBreaksAt
Sets the breaks on the y axis, i.e. at every n’th position a major grid is being printed. Valid values range from 0 to 1.

diagramMargins
If TRUE (default), the diagram has margins, i.e. the y-axis is not exceeded to the diagram’s boundaries.

barWidth
Width of bars. Recommended values for this parameter are from 0.4 to 1.5

barColor
User defined color for bars. If not specified (NULL), a default red-green color palette for four(!) categories will be used for the bar charts. You can use predefined color-sets that are independent from the amount of categories:

- If barColor is "brown", a brown-marine-palette will be used.
- If barColor is "violet", a violet-green palette will be used.
- If barColor is "pink", a pink-green palette will be used.
- If barColor is "brewer", use the colorPalette parameter to specify a palette of the color brewer.

Else specify your own color values as vector (e.g. barColor=c("darkred", "red", "green", "darkgreen")

colorPalette
If barColor is "brewer", specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.

barAlpha
Specify the transparancy (alpha value) of bars.

borderColor
User defined color of whole diagram border (panel border).

barOutline
If TRUE, each bar gets a colored outline. Default is FALSE.

outlineColor
The color of the bar outline. Only applies, if barOutline is set to TRUE.

majorGridColor
Specifies the color of the major grid lines of the diagram background.

minorGridColor
Specifies the color of the minor grid lines of the diagram background.

hideGrid.x
If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y
If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

axisColor
User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).
axisTitle.x A label for the x axis. Useful when plotting histograms with metric scales where no category labels are assigned to the x axis.

axisTitle.y A label for the y axis. Useful when plotting histograms with metric scales where no category labels are assigned to the y axis.

axisTitleColor The color of the x and y axis labels. refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

axisTitleSize The size of the x and y axis labels. refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

showValueLabels Whether counts and percentage values should be plotted to each bar

jitterValueLabels If TRUE, the value labels on the bars will be "jittered", i.e. they have alternating vertical positions to avoid overlapping of labels in case bars are very short. Default is FALSE.

showItemLabels Whether x axis text (category names) should be shown or not

showTickMarks Whether tick marks of axes should be shown or not

showSeparatorLine If TRUE, a line is drawn to visually "separate" each bar in the diagram.

separatorLineColor The color of the separator line. Only applies, if showSeparatorLine is TRUE

separatorLineSize The size of the separator line. only applies, if showSeparatorLine is TRUE

legendPos The position of the legend. Default is "right". Use one of the following values: "right", "left", "bottom", "top".

legendSize The size of the legend.

legendBorderColor The border color of the legend.

legendBackColor The background color of the legend.

theme Specifies the diagram's background theme. Default (parameter NULL) is a gray background with white grids.
   • Use "bw" for a white background with gray grids
   • "classic" for a classic theme (black border, no grids)
   • "minimal" for a minimalistic theme (no border, gray grids) or
   • "none" for no borders, grids and ticks.

flipCoordinates If TRUE, the x and y axis are swapped.

printPlot If TRUE (default), plots the results as graph. Use FALSE if you don't want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Inisibily) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).
Note

Since package version 1.3, the parameter legendLabels, which represent the value labels, are retrieved automatically if a) the variables in items come from a data frame that was imported with the sji.SPSS function (because then value labels are attached as attributes to the data) or b) when the variables are factors with named factor levels (e.g., see column group in dataset PlantGrowth). However, you still can use own parameters as axis- and legendlabels.


References

http://strengejacke.wordpress.com/sjplot-r-package/


See Also

sjp.stackfrq

Examples

# prepare data for dichotomous likert scale, 5 items
likert_2 <- data.frame(as.factor(sample(1:2, 500, replace=TRUE, prob=c(0.3, 0.7))),
                        as.factor(sample(1:2, 500, replace=TRUE, prob=c(0.6, 0.4))),
                        as.factor(sample(1:2, 500, replace=TRUE, prob=c(0.25, 0.75))),
                        as.factor(sample(1:2, 500, replace=TRUE, prob=c(0.9, 0.1))),
                        as.factor(sample(1:2, 500, replace=TRUE, prob=c(0.35, 0.65))))

# create labels
levels_2 <- list(c("Disagree", "Agree"))

# prepare data for 4-category likert scale, 5 items
likert_4 <- data.frame(as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.2, 0.3, 0.1, 0.4))),
                        as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.5, 0.25, 0.15, 0.1))),
                        as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.25, 0.1, 0.4, 0.25))),
                        as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.1, 0.4, 0.4, 0.1))),
                        as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.35, 0.25, 0.15, 0.25))))

# create labels
levels_4 <- list(c("Strongly disagree", "Disagree", "Agree", "Strongly Agree"))

# prepare data for 6-category likert scale, 5 items
likert_6 <- data.frame(
    as.factor(sample(1:6, 500, replace=TRUE, prob=c(0.2, 0.1, 0.1, 0.3, 0.2, 0.1))),
    as.factor(sample(1:6, 500, replace=TRUE, prob=c(0.15, 0.15, 0.3, 0.1, 0.1, 0.2))),
    as.factor(sample(1:6, 500, replace=TRUE, prob=c(0.2, 0.25, 0.05, 0.2, 0.2, 0.2))),
    as.factor(sample(1:6, 500, replace=TRUE, prob=c(0.2, 0.1, 0.1, 0.4, 0.1, 0.1))),
    as.factor(sample(1:6, 500, replace=TRUE, prob=c(0.1, 0.4, 0.1, 0.3, 0.05, 0.15))))

# create labels
levels_6 <- list(c("Very strongly disagree", "Strongly disagree", "Disagree", "Agree", "Strongly Agree", "Very strongly agree"))

# create item labels
items <- list(c("Q1", "Q2", "Q3", "Q4", "Q5"))

# plot dichotomous likert scale, ordered by "negative" values
sjp.lkert(likert_2, legendLabels=levels_2, axisLabels.y=items, orderBy="neg")

# plot 4-category likert-scale, no order
sjp.lkert(likert_4, legendLabels=levels_4, axisLabels.y=items)

# plot 4-category likert-scale, ordered by positive values,
# in brown color scale and with jittered value labels
sjp.lkert(likert_6, legendLabels=levels_6, barColor="brown",
    axisLabels.y=items, orderBy="pos", jitterValueLabels=TRUE)

sjp.lm

Plot beta coefficients of lm

Description
Plot beta coefficients of linear regressions with confidence intervals as dot plot (forest plot). Additionally, the standardized beta values are plotted as red dots.

Usage

sjp.lm(fit, sort = "beta", title = NULL, titleSize = 1.3,
    titleColor = "black", axisLabels.y = NULL, showAxisLabels.y = TRUE,
    axisLabelSize = 1.1, axisLabelColor = "gray30",
    axisTitle.x = "Estimates", axisTitleSize = 1.4,
    axisTitleColor = c("#444444"), axisLimits = NULL,
    valueLabelColor = "gray20", valueLabelColorNS = "gray50",
    valueLabelSize = 4.5, valueLabelAlpha = 0.8, axisLabelAngle.x = 0,
    axisLabelAngle.y = 0, errorBarColor = "#3366a0", errorBarWidth = 0,
    errorBarSize = 0.8, pointColor = "#3366a0", pointSize = 3,
    pointColorStdBeta = "#cc5553", pointSizeStdBeta = 3,
    stdBetaLineType = 2, stdBetaLineAlpha = 0.3, interceptLineType = 2,
    interceptLineColor = "grey70", breakTitleAt = 50, breakLabelsAt = 12,
    gridBreaksAt = NULL, borderColor = NULL, axisColor = NULL,
    theme = NULL, flipCoordinates = TRUE, majorGridColor = NULL,
    minorGridColor = NULL, hideGrid.x = FALSE, hideGrid.y = FALSE,
    showTickMarks = TRUE, showValueLabels = TRUE, labelDigits = 2,
    showPValueLabels = TRUE, showModelSummary = TRUE,
    showStandardBeta = FALSE, showStandardBetaLine = FALSE,
    printPlot = TRUE)

Arguments

fit The model of the linear regression (lm-Object).
title Diagram’s title as string. Example: title=c("my title")
titleSize The size of the plot title. Default is 1.3.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>titleColor</td>
<td>The color of the plot title. Default is &quot;black&quot;.</td>
</tr>
<tr>
<td>sort</td>
<td>Determines whether the predictors are sorted by beta-values (default, or use &quot;beta&quot; as parameter) or by standardized beta values (use &quot;std&quot;).</td>
</tr>
<tr>
<td>axisLabels.y</td>
<td>Labels of the predictor variables (independent vars) that are used for labelling the axis. Passed as vector of strings. Example: axisLabels.y=c(&quot;Label1&quot;, &quot;Label2&quot;, &quot;Label3&quot;). Note: If you use the sji.spss function and the sji.getValueLabels function, you receive a list object with label string. The labels may also be passed as list object. They will be unlisted and converted to character vector automatically.</td>
</tr>
<tr>
<td>showAxisLabels.y</td>
<td>Whether x axis text (category names, predictor labels) should be shown (use TRUE) or not. Default is TRUE.</td>
</tr>
<tr>
<td>axisLabelSize</td>
<td>The size of value labels in the diagram. Default is 4, recommended values range between 2 and 8.</td>
</tr>
<tr>
<td>axisLabelColor</td>
<td>The color of the category labels (predictor labels). Default is a dark grey (grey30).</td>
</tr>
<tr>
<td>axisTitle.x</td>
<td>A label for the x axis. Default is &quot;Estimates&quot;.</td>
</tr>
<tr>
<td>axisTitleColor</td>
<td>The color of the x axis label. Default is a dark grey.</td>
</tr>
<tr>
<td>axisTitleSize</td>
<td>The size of the x axis label. Default is 1.4.</td>
</tr>
<tr>
<td>axisLimits</td>
<td>Defines the range of the axis where the beta coefficients and their confidence intervals are drawn. By default, the limits range from the lowest confidence interval to the highest one, so the diagram has maximum zoom. Use your own values as 2-value-vector, for instance: limits=c(-0.8,0.8).</td>
</tr>
<tr>
<td>valueLabelColor</td>
<td>Colour of the values (significant beta coefficients) inside the diagrams. Only applies, when parameter showValueLabels is set to TRUE. Use any valid colour value, e.g. valueLabelColor=&quot;grey50&quot; or valueLabelColor=c(&quot;#cc3366&quot;).</td>
</tr>
<tr>
<td>valueLabelColorNS</td>
<td>Colour of the non significant values (non significant beta coefficients) inside the diagrams. Only applies, when parameter showValueLabels is set to TRUE. Use any valid colour value, e.g. valueLabelColor=&quot;grey50&quot; or valueLabelColor=c(&quot;#cc3366&quot;).</td>
</tr>
<tr>
<td>valueLabelSize</td>
<td>Size of the value labels. Default is 4.5. Recommended Values range from 2 to 8</td>
</tr>
<tr>
<td>valueLabelAlpha</td>
<td>The alpha level (transparancy) of the value labels. Default is 0.8, use any value from 0 to 1.</td>
</tr>
<tr>
<td>axisLabelAngle.x</td>
<td>Angle for axis-labels where the estimates are printed. Note that due to the coordinate flip, the acutal y-axis with estimates labels are appearing on the x-axis.</td>
</tr>
<tr>
<td>axisLabelAngle.y</td>
<td>Angle for axis-labels, passed as numeric value.</td>
</tr>
<tr>
<td>errorBarColor</td>
<td>The color of the error bars that indicate the confidence intervals of the beta-coefficients</td>
</tr>
<tr>
<td>errorBarWidth</td>
<td>The width of the error bar ends. Default is 0</td>
</tr>
<tr>
<td>errorBarSize</td>
<td>The size of the error bar. Default is 0.8</td>
</tr>
<tr>
<td>pointColor</td>
<td>The colour of the points that indicate the beta-value.</td>
</tr>
</tbody>
</table>
pointSize
The size of the points that indicate the beta-value. Default is 3.

pointColorStdBeta
The colour of the points that indicate the standardized beta-value.

pointSizeStdBeta
The size of the points that indicate the standardized beta-value. Default is 3.

stdBetaLineType
The standardized beta-value dots are connected by a thin line for a better overview. With this parameter you can specify the line type.

stdBetaLineAlpha
The alpha-value for the line that connects the standardized beta-value dots.

interceptLineType
The linetype of the intercept line (zero point). Default is 2 (dashed line).

interceptLineColor
The color of the intercept line. Default value is "grey70".

breakTitleAt
Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt
Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

gridBreaksAt
Sets the breaks on the y axis, i.e. at every n'th position a major grid is being printed. Default is NULL, so pretty gridbeaks will be used.

borderColor
User defined color of whole diagram border (panel border).

axisColor
User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

theme
Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.

  • Use "bw" for a white background with gray grids
  • "classic" for a classic theme (black border, no grids)
  • "minimal" for a minimalistic theme (no border, gray grids) or
  • "none" for no borders, grids and ticks.

flipCoordinates
If TRUE (default), predictors are plotted on the left y-axis and estimate values are plotted on the x-axis.

majorGridColor
Specifies the color of the major grid lines of the diagram background.

minorGridColor
Specifies the color of the minor grid lines of the diagram background.

hideGrid.x
If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y
If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

showTickMarks
Whether tick marks of axes should be shown or not.

showValueLabels
Whether the beta and standardized beta values should be plotted to each dot or not.

labelDigits
The amount of digits for rounding the estimations (see showValueLabels). Default is 2, i.e. estimators have 2 digits after decimal point.
showPValueLabels

Whether the significance levels of each coefficient should be appended to values or not.

showModelSummary

If TRUE (default), a summary of the regression model with Intercept, R-square, F-Test and AIC-value is printed to the lower right corner of the diagram.

showStandardBeta

Whether or not the dots for the standardized beta values should be plotted to the diagram.

showStandardBetaLine

Whether or not the connecting line for the standardized beta values should be plotted to the diagram. Default is FALSE.

printPlot

If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Insisibily) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

Note

Based on an idea from surefoss: http://www.surefoss.org/dataanalysis/plotting-odds-ratios-aka-a-forrest

References

- http://strengejacke.wordpress.com/sjplot-r-package/

See Also

sjp.lm
sjp.reglin
sjp.lm.int
sjp.scatter
sju.betaCoef

Examples

# fit linear model
fit <- lm(airquality$Ozone ~ airquality$Wind + airquality$Temp + airquality$Solar.R)

# plot estimates with CI and standardized beta-values
sjp.lm(fit, gridBreaksAt=2)

# plot estimates with CI without standardized beta-values
# and with narrower tick marks (because "gridBreaksAt" was not specified)
sjp.lm(fit, showStandardBeta=FALSE)
**Plot interaction terms of linear models**

**Description**

Plot regression curves of significant interaction terms in linear models (lm). Note that beside interaction terms, also the single predictors of each interaction must be included in the fitted model as well. Thus, `lm(dep~pred1*pred2)` will work, but `lm(dep~pred1:pred2)` won’t!

**Usage**

```r
sjp.lm.int(fit, smooth = "none", diff = FALSE, swapPredictors = FALSE, plevel = 0.05, title = NULL, titleSize = 1.3, titleColor = "black", fillColor = "grey", fillAlpha = 0.4, lowerBoundColor = "#3366cc", upperBoundColor = "#cc3300", lineColor = "#33cc66", axisTitle.x = NULL, axisTitle.y = NULL, axisLabelColor = "gray30", axisLabelSize = 1.1, axisTitleColor = "black", axisTitleSize = 1.3, legendLabels = NULL, legendLabelColor = "black", legendLabelSize = 0.9, showValueLabels = FALSE, valueLabelColor = "black", valueLabelSize = 4, valueLabelAlpha = 0.8, breakTitleAt = 50, breakLegendLabelsAt = 20, breakAnnotationLabelsAt = 50, gridBreaksAt = NULL, theme = NULL, showTickMarks = TRUE, showInterceptLines = FALSE, showInterceptLabels = TRUE, interceptLineColor = "#3366cc", estLineColor = "#cc3300", lineLabelSize = 3.7, lineLabelColor = "black", lineLabelString = "(no interaction)", borderColor = NULL, axisColor = NULL, majorGridColor = NULL, minorGridColor = NULL, hideGrid.x = FALSE, hideGrid.y = FALSE, printPlot = TRUE)
```

**Arguments**

- **fit**: the fitted linear model (lm) object, including interaction terms
- **smooth**: smoothes the regression line in case it is not linear. Default is "none", so no smoothing is applied. Use "loess" for loess-smoothing or "lm" to force linear regression lines.
- **diff**: if FALSE (default), the minimum and maximum interaction effects of predictor 2 on predictor 1 are shown (one line each). if TRUE, only the difference between minimum and maximum interaction effect is shown (single line)
- **swapPredictors**: if TRUE, the predictor with less unique values is printed along the x-axis. Default is FALSE, so the predictor with more unique values is printed along the x-axis.
- **plevel**: Indicates at which p-value an interaction term is considered as significant. Default is 0.05 (5 percent).
- **title**: a default title used for the plots. Default value is NULL, which means that each plot’s title includes the dependent variable as well as the names of the interaction terms.
- **titleSize**: The size of the plot title. Default is 1.3.
titleColor        The color of the plot title. Default is "black".
fillColor        fill color of the shaded area between the minimum and maximum lines. Default is "grey". Either set fillColor to NULL or use 0 for fillAlpha if you want to hide the shaded area.
fillAlpha        alpha value (transparency) of the shaded area between the minimum and maximum lines. Default is 0.4. Use either 0 or set fillColor to NULL if you want to hide the shaded area.
lowerBoundColor  the color of the line indicating the lower bound of the interaction term. Default value is "#3366cc" (blue-like)
upperBoundColor  the color of the line indicating the upper bound of the interaction term. Default value is "#cc3300" (red-like)
lineColor        the color of the line indicating the upper difference between lower and upper bound of interaction terms. Only applies if diff is TRUE. Default value is "#33c666" (green-like)
axistitle.x      a default title used for the x-axis. Default value is NULL, which means that each plot’s x-axis uses the predictor’s name as title.
axistitle.y      a default title used for the y-axis. Default value is NULL, which means that each plot’s y-axis uses the dependent variable’s name as title.
axistitleColor   the color value for the axis titles (both x and y). Default value is "black".
axistitleSize    The size of axis titles (both x and y). Default is 1.3, recommended values range between 0.5 and 3.0
legendLabels     Labels for the guide/legend. Default is NULL, so the name of the predictor with min/max-effect is used as legend label.
legendLabelSize  The size of legend labels. Default is 0.9, recommended values range between 0.5 and 3.0
legendLabelColor user defined color for legend labels. If not specified, black will be used for the labels
showValueLabels  if TRUE, value labels are plotted along the lines. Default is FALSE.
valueLabelSize   size of the value labels. Default is 4. Recommended Values range from 2 to 8
valueLabelColor  colour of the values inside the diagrams. Only applies, when parameter showValueLabels is set to TRUE. Use any valid colour value, e.g. valueLabelColor="grey50" or valueLabelColor=c("#cc3366"). Default is "black".
valueLabelAlpha  the alpha level (transparency) of the value labels. Default is 0.8, use any value from 0 to 1.
breakTitleAt  Wordwrap for diagram's title. Determines how many chars of the title are
displayed in one line and when a line break is inserted. Default is 50.
breakLegendLabelsAt  
Wordwrap for diagram legend labels. Determines how many chars of the legend
labels are displayed in one line and when a line break is inserted. Default is 20.
breakAnnotationLabelsAt  
Wordwrap for diagram annotation labels. Determines how many chars of the legend
labels are displayed in one line and when a line break is inserted. Default is 50. Only applies if showInterceptLines is TRUE.
gridBreaksAt  Sets the breaks on the y axis, i.e. at every n’th position a major grid is being
printed. Default is NULL.
theme  specifies the diagram’s background theme. default (parameter NULL) is a gray
background with white grids. Use "bw" for a white background with gray grids,
"classic" for a classic theme (black border, no grids), "minimal" for a minimalistic theme (no border, gray grids) or "none" for no borders, grids and ticks.
showTickMarks  Whether tick marks of axes should be shown or not
showInterceptLines  
If TRUE, the intercept and the estimate of the predictor (reference category of
predictor in case interaction is not present) are plotted.
showInterceptLabels  
If TRUE (default), the intercept lines are labelled. Only applies if showInterceptLines
is TRUE.
interceptLineColor  The line color of the model’s intercept line. Only applies, if showInterceptLines
is TRUE.
estLineColor  The line color of the model’s predictor’s estimate line. Only applies, if showInterceptLines
is TRUE.
lineLabelSize  The size of the intercept line annotations inside the plot. Only applies if showInterceptLines
is TRUE. Default is 3.7.
lineLabelColor  The color of the intercept line annotations inside the plot. Only applies if
showInterceptLines is TRUE. Default is "black".
lineLabelString  Default string for the intercept lines that is appended to the predictor variable
name. By default, this string is "(no interaction)".
borderColor  user defined color of whole diagram border (panel border)
axisColor  user defined color of axis border (y- and x-axis, in case the axes should have
different colors than
majorGridColor  specifies the color of the major grid lines of the diagram background
minorGridColor  specifies the color of the minor grid lines of the diagram background
hideGrid.x  If TRUE, the x-axis-gridlines are hidden. Default if FALSE.
hideGrid.y  If TRUE, the y-axis-gridlines are hidden. Default if FALSE.
printPlot  If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot
any graphs. In either case, the ggplot-object will be returned as value.
Value

(Insisibly) returns the ggplot-objects with the complete plot-list (plot.list) as well as the data frame that were used for setting up the ggplot-objects (df.list).

Note

Beside interaction terms, also the single predictors of each interaction must be included in the fitted model as well. Thus, \( \text{lm} (\text{dep}_1 \text{pred1} \text{pred2}) \) will work, but \( \text{lm} (\text{dep}_1 \text{pred1} : \text{pred2}) \) won’t!

References


See Also

sjp.lm
sjp.reglin
sjp.lm.ma

Examples

```r
# Note that the data sets used in this example may not be perfectly suitable for
# fitting linear models. I just used them because they are part of the R-software.

# fit "dummy" model.
fit <- lm(weight ~ Time * Diet, data=ChickWeight, x=TRUE)

# show summary to see significant interactions
summary(fit)

# plot regression line of interaction terms
sjp.lm.int(fit)
# plot regression line of interaction terms, including value labels
sjp.lm.int(fit, showValueLabels=TRUE)

# load sample data set
data(efc)
# create data frame with variables that should be included
# in the model
df <- as.data.frame(cbind(usage=efc$tot_sc_e,
    sex=efc$c161sex,
    education=efc$c172code,
    burden=efc$neg_c_7,
    dependency=efc$e42dep))

# convert gender predictor to factor
df$sex <- relevel(factor(df$sex), ref="2")
```
# fit "dummy" model
fit <- lm(usage ~ ., data=df, x=TRUE)
summary(fit)

# plot interactions
sjp.lm.int(fit)
# plot interactions, including those with p-value up to 0.1
sjp.lm.int(fit, plevel=0.1, showInterceptLines=TRUE)

sxp.lm.ma

Plot model assumptions of lm's

Description
Plots model assumptions of linear models to verify if linear regression is applicable

Usage
sjp.lm.ma(linreg, showOriginalModelOnly = TRUE, completeDiagnostic = FALSE)

Arguments
linreg a fitted lm-model
showOriginalModelOnly
  if TRUE (default), only the model assumptions of the fitted model linreg are
  plotted. if FALSE, the model assumptions of an updated model where outliers
  are automatically excluded are also plotted.
completeDiagnostic
  if TRUE, additional tests are performed. Default is FALSE

Value
an updated fitted linear model where outliers are dropped out.

See Also
  sjp.lm
  sjp.reglin
  sjp.lm.int

Examples
# fit linear model
fit <- lm(airquality$Ozone ~ airquality$Wind + airquality$Temp + airquality$Solar.R)
fit.updated <- sjp.lm.ma(fit)
Plot regression line of fitted \textit{lm}

**Description**

Plot a regression line with confidence interval for a fitted model with only one predictor (i.e. \textit{lm}(y \sim x)). This function may plot two lines: The resulting linear regression line including confidence interval (in blue) by default, and a loess-smoothed line without confidence interval (in red) if parameter \textit{showloess} is \texttt{true}. The better the linear relationship of predictor and response is, the more both lines should overlap (i.e. the red loess-smoothed line is almost linear).

Furthermore, a scatter plot of response and predictor values is plotted.

**Usage**

\texttt{sjp.lm1(fit, data, title = \texttt{NULL}, titleSize = 1.3, titleColor = "black", breakTitleAt = 50, axisLabel.x = \texttt{NULL}, axisLabel.y = \texttt{NULL}, breakLabelsAt = 12, axisLabelSize = 1.1, axisLabelColor = "gray30", axisTitleSize = 1.4, axisTitleColor = c("#444444"), lineColor = "blue", modsumLabelColor = "grey20", modsumLabelSize = 4.5, showCI = \texttt{TRUE}, cilevel = 0.95, pointAlpha = 0.2, pointColor = "black", showScatterPlot = \texttt{TRUE}, showLoess = \texttt{FALSE}, loessLineColor = "red", showLoessCI = \texttt{FALSE}, loessCILevel = 0.95, showModelSummary = \texttt{TRUE}, borderColor = \texttt{NULL}, axisColor = \texttt{NULL}, theme = \texttt{NULL}, majorGridColor = \texttt{NULL}, minorGridColor = \texttt{NULL}, hideGrid.x = \texttt{FALSE}, hideGrid.y = \texttt{FALSE}, showTickMarks = \texttt{TRUE}, printPlot = \texttt{TRUE})}

**Arguments**

- \textit{fit} The model of the linear regression (\texttt{lm}-Object).
- \textit{data} The data/dataset/dataframe used to fit the model.
- \textit{title} Diagram’s title as string. Example: \textit{title=} \texttt{"my title"}
- \textit{titleSize} The size of the plot title. Default is 1.3.
- \textit{titleColor} The color of the plot title. Default is "black".
- \textit{breakTitleAt} Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.
- \textit{axisLabel.x} Labels of the predictor (independent variable) that is used for labelling the axis. Passed as string. Example: \textit{axisLabel.x=} \texttt{"My Predictor Var"}. Note: If you use the \texttt{sjipls} function and the \texttt{sj.getVariableLabels} function, you receive a character vector with variable label strings. You can use it like so: \textit{axisLabel.x=sjipls(efc)\"quol_5\"}
- \textit{axisLabel.y} Labels of the response (dependent variable) that is used for labelling the axis. Passed as string. Example: \textit{axisLabel.y=} \texttt{"My Dependent Var"}. Note: If you use the \texttt{sjipls} function and the \texttt{sj.getVariableLabels} function, you receive a character vector with variable label strings. You can use it like so: \textit{axisLabel.y=sjipls(efc)\"neg_c_7\"}

- \textit{breakLabelsAt} Wordwrap for diagram labels. Determines how many chars of the label are displayed in one line and when a line break is inserted into the labels.
- \textit{axisLabelSize} The size of the axis labels. Default is 1.1.
- \textit{axisLabelColor} The color of the axis labels. Default is "gray30".
- \textit{axisTitleSize} The size of the axis titles. Default is 1.4.
- \textit{axisTitleColor} The color of the axis titles. Default is \texttt{c\"#444444\"}.
- \textit{lineColor} The color of the regression line. Default is "blue".
- \textit{modsumLabelColor} The color of the model summary. Default is "grey20".
- \textit{modsumLabelSize} The size of the model summary. Default is 4.5.
- \textit{showCI} Whether to show confidence interval. Default is \texttt{TRUE}.
- \textit{cilevel} The level of the confidence interval. Default is 0.95.
- \textit{pointAlpha} The alpha value of the points. Default is 0.2.
- \textit{pointColor} The color of the points. Default is "black".
- \textit{showScatterPlot} Whether to show scatter plot. Default is \texttt{TRUE}.
- \textit{showLoess} Whether to show loess-smoothed line. Default is \texttt{FALSE}.
- \textit{loessLineColor} The color of the loess-smoothed line. Default is "red".
- \textit{showLoessCI} Whether to show confidence interval of the loess-smoothed line. Default is \texttt{FALSE}.
- \textit{loessCILevel} The level of the confidence interval of the loess-smoothed line. Default is 0.95.
- \textit{showModelSummary} Whether to show model summary. Default is \texttt{TRUE}.
- \textit{borderColor} The color of the border. Default is \texttt{NULL}.
- \textit{axisColor} The color of the axes. Default is \texttt{NULL}.
- \textit{theme} The theme of the plot. Default is \texttt{NULL}.
- \textit{majorGridColor} The color of the major grid. Default is \texttt{NULL}.
- \textit{minorGridColor} The color of the minor grid. Default is \texttt{NULL}.
- \textit{hideGrid.x} Whether to hide grid on x-axis. Default is \texttt{FALSE}.
- \textit{hideGrid.y} Whether to hide grid on y-axis. Default is \texttt{FALSE}.
- \textit{showTickMarks} Whether to show tick marks. Default is \texttt{TRUE}.
- \textit{printPlot} Whether to print the plot. Default is \texttt{TRUE}.
breakLabelsAt  Wordwrap for axis labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted
axisLabelSize  The size of axis tick marks  Default is 1.1.
axisLabelColor  The color of the axis tick marks.  Default is a dark grey (grey30).
axisTitleColor  The color of the axis labels (response and predictor label).  Default is a dark grey.
axisTitleSize  The size of the axis label (response and predictor label).  Default is 1.4.
lineColor  The color of the regression line.  Default is "blue”.
modsumLabelColor  Colour of the model summary inside the diagrams. Only applies, when parameter showModelSummary is set to TRUE. Use any valid colour value, e.g. modsumLabelColor="grey50" or valueLabelColor=c(“#cc3366”).
modsumLabelSize  Size of the model summary text.  Default is 4.5. Recommended Values range from 2 to 8
showCI  If TRUE (default), a confidence region for the regression line will be plotted. Use ciLevel to specify the confidence level.
ciLevel  The confidence level of the confidence region. Only applies when showCI is TRUE. Default is 0.95.
pointAlpha  The alpha values of the scatter plot’s point-geoms. Default is 0.2.
pointColor  The color of the scatter plot’s point-geoms. Only applies when showScatterPlot is TRUE. Default is “black”.
showScatterPlot  If TRUE (default), a scatter plot of response and predictor values for each predictor of the fitted model fit is plotted.
showLoess  If TRUE, an additional loess-smoothed line is plotted.
loessLineColor  The color of the loess-smoothed line. Default is "red”. Only applies, if showLoess is TRUE.
showLoessCI  If TRUE, a confidence region for the loess-smoothed line will be plotted. Default is FALSE. Use loessCiLevel to specify the confidence level. Only applies, if showLoess is TRUE.
loessCiLevel  The confidence level of the loess-line’s confidence region. Only applies, if showLoessCI is TRUE. Default is 0.95.
showModelSummary  If TRUE (default), a summary of the regression model with Intercept, R-square, F-Test and AIC-value is printed to the lower right corner of the diagram.
borderColor  User defined color of whole diagram border (panel border).
axisColor  User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).
theme  Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.
    • Use ”bw” for a white background with gray grids
    • "classic” for a classic theme (black border, no grids)
• "minimal" for a minimalistic theme (no border, gray grids) or
• "none" for no borders, grids and ticks.

majorGridColor Specifies the color of the major grid lines of the diagram background.

minorGridColor Specifies the color of the minor grid lines of the diagram background.

hideGrid.x If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

showTickMarks Whether tick marks of axes should be shown or not

printPlot If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Invisibly) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

See Also

sjp.lm
sjp.reglin
sjp.scatter

Examples

# load sample data
data(efc)
# fit model
fit <- lm(neg_c_7 ~ quol_5, data=efc, na.action=na.omit)
# plot regression line
sjp.lm(fit, efc)
# plot regression line with label strings
sjp.lm(fit, efc,
  axisLabel.x=sji.getVariableLabels(efc)['quol_5'],
  axisLabel.y=sji.getVariableLabels(efc)['neg_c_7'],
  showLoess=TRUE)

sjp.lmm

Plot beta coefficients of multiple fitted lm's

Description

Plot beta coefficients (estimates) with confidence intervals of multiple fitted linear models in one plot.
Usage

sjp.lm(..., title = NULL, titleSize = 1.3, titleColor = "black",
labelDependentVariables = NULL, legendDepVarTitle = "Dependent Variables",
legendPValTitle = "p-level", stringModel = "Model", axisLabels.y = NULL,
axisLabelSize = 1.1, axisLabelAngle.x = 0, axisLabelAngle.y = 0,
axisLabelColor = "gray30", axisTitle.x = "Estimates",
axisTitleSize = 1.2, axisTitleColor = c("#444444"), axisLimits = NULL,
breakTitleAt = 50, breakLabelsAt = 12, breakLegendAt = 20,
gridBreaksAt = NULL, errorBarWidth = 0, errorBarSize = 0.5,
errorBarLineType = 1, pointSize = 3, modelPlotSpace = 0.4,
colorPalette = "Paired", modelColors = NULL, valueLabelColor = "black",
valueLabelSize = 4, valueLabelAlpha = 1, nsAlpha = 1,
usePShapes = FALSE, axisColor = NULL, borderColor = NULL,
interceptLineType = 2, interceptLineColor = "grey70",
majorGridColor = NULL, minorGridColor = NULL, hideGrid.x = FALSE,
hideGrid.y = FALSE, theme = NULL, flipCoordinates = TRUE,
legendPos = "right", legendSize = 1, legendBorderColor = "white",
legendBackColor = "white", showIntercept = FALSE,
showAxisLabels.y = TRUE, showTickMarks = TRUE, showValueLabels = TRUE,
labelDigits = 2, showPValueLabels = TRUE, printPlot = TRUE)

Arguments

... One or more fitted lm-objects.
title Diagram’s title as string. Example: title=c("my title")
titleSize The size of the plot title. Default is 1.3.
titleColor The color of the plot title. Default is "black".
labelDependentVariables Labels of the dependent variables of all fitted models which have been used as first parameter(s), provided as char vector.
legendDepVarTitle A character vector used for the title of the dependent variable’s legend. Default is "Dependent Variables".
legendPValTitle A character vector used for the title of the significance level’s legend. Default is "p-level". Only applies if usePShapes is TRUE.
stringModel String constant used as legend text for the model names in case no labels for the dependent variables are provided (see labelDependentVariables). Default is "Model".
axisLabels.y Labels of the predictor variables (independent vars, betas) that are used for labelling the axis. Passed as vector of strings. Example: axisLabels.y=c("Label1", "Label2", "Label3")
Note: If you use the sj.i.SPSS function and the sj.i.getValueLabels function, you receive a list object with label strings. The labels may also be passed as list object. They will be unlisted and converted to character vector automatically.
axisLabelSize The size of value labels in the diagram. Default is 1.1, recommended values range between 0.7 and 3.0
showAxisLabels.y

Whether beta names (predictor labels) should be shown or not.

showTickMarks

Whether tick marks of axes should be shown or not.

axisTitle.x

A label ("title") for the x axis.

axisTitleColor

The color of the x axis label.

axisTitleSize

The size of the x axis label.

axisLimits

Defines the range of the axis where the beta coefficients and their confidence intervals are drawn. By default, the limits range from the lowest confidence interval to the highest one, so the diagram has maximum zoom. Use your own values as 2-value-vector, for instance: limits=c(-0.8,0.8).

axisLabelAngle.x

Angle for axis-labels where the estimates labels are printed. Note that due to the coordinate flip, the actual y-axis with estimates are appearing on the x-axis.

axisLabelAngle.y

Angle for axis-labels where the predictor labels (axisLabels.y) are printed. Note that due to the coordinate flip, the actual x-axis with predictor labels are appearing on the y-axis.

breakTitleAt

Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt

Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

breakLegendAt

Wordwrap for legend, i.e. names of the dependent variables of each fitted model. See parameter labelDependentVariables. Determines how many chars of each dependent variable name is displayed in one line in the legend and when a line break is inserted.

gridBreaksAt

Sets the breaks on the y axis, i.e. at every n'th position a major grid is being printed. Default is NULL, so pretty gridbeaks will be used.

pointSize

The size of the points that indicate the beta-value. Default is 3.

modelPlotSpace

Defines the space between the dots and error bars of the plotted fitted models. Default is 0.3.

modelColors

A vector with colors for representing the beta values (i.e. points and error bars) of the different fitted models. Thus, the length of this vector must be equal to the length of supplied fitted models, so each model is represented by its own color. You can use:

- "bw" or "black" for only one colouring in almost black
- "gray", "grey" or "gs" for a grayscale
- "brewer" for colours from the color brewer palette.

If modelColors is "brewer", use the colorPalette parameter to specify a palette of the color brewer. Else specify your own color values as vector (e.g. modelColors=c("#f00000", "#0ff000").

colorPalette

If parameter modelColors is brewer, specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.

axisLabelColor

Colour of the tick labels at the axis (variable names, beta names).
valueLabelColor
The colour of the beta values. These values are printed above the plots respectively beside the bar charts. Default color is “black”.

valueLabelSize
Size of the value labels. Default is 4. Recommended values range from 2 to 8

valueLabelAlpha
The alpha level (transparency) of the value labels. Default is 1, use any value from 0 to 1.

nsAlpha
The alpha level (transparency) of non significant predictors. Points and error bars are affected by this value and plotted with a slight transparency. Default is 1.

usePShapes
If TRUE, significant levels are distinguished by different point shapes and a related legend is plotted. Default is FALSE.

axisColor
User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

borderColor
User defined color of whole diagram border (panel border).

interceptLineColor
The linetype of the intercept line (zero point). Default is 2 (dashed line).

interceptLineColor
The color of the intercept line. Default value is "grey70".

errorBarWidth
The width of the error bar ends. Default is 0

errorBarSize
The size (thickness) of the error bars. Default is 0.5

errorBarLinetype
The linetype of error bars. Default is Q (solid line).

majorGridColor
Specifies the color of the major grid lines of the diagram background.

minorGridColor
Specifies the color of the minor grid lines of the diagram background.

hideGrid.x
If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y
If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

theme
Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.

• Use "bw" for a white background with gray grids
• "classic" for a classic theme (black border, no grids)
• "minimal" for a minimalistic theme (no border, gray grids) or
• "none" for no borders, grids and ticks.

flipCoordinates
If TRUE (default), predictors are plotted on the left y-axis and estimate values are plotted on the x-axis.

legendPos
The position of the legend, if a legend is drawn. Use "bottom", "top", "left" or "right" to position the legend above, below, on the left or right side of the diagram. Right positioning is default.

legendSize
The text size of the legend. Default is 1. Relative size, so recommended values are from 0.3 to 2.5

legendBorderColor
Color of the legend’s border. Default is "white", so no visible border is drawn.
legendBackColor Fill color of the legend’s background. Default is "white", so no visible background is drawn.

showIntercept If true, the intercept of the fitted model is also plotted. Default is FALSE.

showValueLabels Whether the beta value labels should be plotted.

labelDigits The amount of digits for rounding the estimations (see showValueLabels). Default is 2, i.e. estimators have 2 digits after decimal point.

showPValueLabels Whether the significance levels of each coefficient should be appended to values or not.

printPlot If true (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Insisibly) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

References

http://strengejacke.wordpress.com/sjplot-r-package/

See Also

sjp.lm
sjt.lm
sjp.lm.ma
sjp.glmm

Examples

# prepare dummy variables for binary logistic regression
# Now fit the models. Note that all models share the same predictors
# and only differ in their dependent variable
data(efc)

# fit first model
fit1 <- lm(barthtot ~ c160age + c12hour + c161sex + c172code, data=efc)
# fit second model
fit2 <- lm(neg_c_7 ~ c160age + c12hour + c161sex + c172code, data=efc)
# fit third model
fit3 <- lm(tot_sc_e ~ c160age + c12hour + c161sex + c172code, data=efc)

# plot multiple models
sjp.lmm(fit1, fit2, fit3)

# plot multiple models with legend labels and point shapes instead of value labels
sjp.lmm(fit1, fit2, fit3, axisLabels.y=c("Carer's Age", "Hours of Care"),
```r
sjp.pca(data, numberoffactors = NULL, factorloadingtolerance = 0.1,
plotEigenvalues = FALSE, digits = 2, title = NULL, titleSize = 1.3,
titleColor = "black", axisLabels.y = NULL, type = "tile",
geomAlpha = 0.8, valueLabelColor = "black", valueLabelSize = 4.5,
valueLabelAlpha = 1, circleSize = 10, outlineColor = "black",
outlineSize = 0.2, axisColor = NULL, borderColor = NULL,
axisLabelSize = 1.1, axisLabelColor = "gray30", axisLabelAngle.x = 0,
axisLabelAngle.y = 0, breakTitleAt = 50, breakLabelsAt = 20,
hideLegend = TRUE, legendTitle = NULL, showValueLabels = TRUE,
showTickMarks = FALSE, showCronbachsAlpha = TRUE, fillColor = NULL,
majorGridColor = NULL, minorGridColor = NULL, theme = NULL,
printPlot = TRUE)
```

**Arguments**

- `data`: A data frame with factors (each columns one variable) that should be used to compute a PCA, or a `prcomp` object.
- `numberoffactors`: A predefined number of factors to use for the calculating the varimax rotation. By default, this value is `NULL` and the amount of factors is calculated according to the Kaiser-criteria. See parameter `plotEigenvalues`.
- `factorloadingtolerance`: Specifies the minimum difference a variable needs to have between factor loadings (components) in order to indicate a clear loading on just one factor and not diffusing over all factors. For instance, a variable with 0.8, 0.82 and 0.84 factor loading on 3 possible factors can not be clearly assigned to just one factor.
and thus would be removed from the principal component analysis. By default, the minimum difference of loading values between the highest and 2nd highest factor should be 0.1

plotEigenvalues
If TRUE, a plot showing the Eigenvalues according to the Kaiser criteria is plotted to determine the number of factors.

digits
The amount of decimals used. Default is 2.

title
Title of the diagram, plotted above the whole diagram panel.
	titleSize
The size of the plot title. Default is 1.3.

titleColor
The color of the plot title. Default is "black".

axisLabels.y
The item labels that are printed on the y-axis. If no item labels are provided (default), the data frame's column names are used. Item labels must be a string vector, e.g.: axisLabels.y=c("Var 1", "Var 2", "Var 3").

type
Indicates whether "circle" (default) or "tile" geoms should be used for plotting.

gemAlpha
Specify the transparency (alpha value) of geom objects (circles or tiles). Default is 0.8.

valueLabelColor
The color of the value labels (numbers) inside the diagram. Default is "black".

valueLabelSize
The size of value labels in the diagram. Default is 4.5, recommended values range between 2 and 8.

valueLabelAlpha
Specify the transparency (alpha value) of value labels. Default is 0.8

circleSize
Specifies the circle size factor. The circle size depends on the correlation value multiplicated with this factor. Default is 10.

outlineColor
Defines the outline color of geoms (circles or tiles). Default is "black".

outlineSize
Defines the outline size of geoms (circles or tiles). Default is 1.

axisColor
User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

borderColor
User defined color of whole diagram border (panel border).

axisLabelSize
The size of variable labels at the axes. Default is 1.1, recommended values range between 0.5 and 3.0.

axisLabelColor
User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

axisLabelAngle.x
Angle for x-axis-labels.

axisLabelAngle.y
Angle for y-axis-labels.

breakTitleAt
Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title. Default is 50.

breakLabelsAt
Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted. Default is 12.
hideLegend: Show or hide the legend. The legend indicates the strength of correlations by gradient colour fill. Default is TRUE, hence the legend is hidden.

legendTitle: The legend title, provided as string, e.g. legendTitle=c("Factor loading"). Default is NULL, hence no legend title is used.

showValueLabels: Whether factor loading values should be plotted to each geom. Default is TRUE.

showTickMarks: Whether tick marks should be plotted or not. Default is FALSE.

showCronbachsAlpha: If TRUE (default), the cronbach’s alpha value for each factor scale will be calculated, i.e. all variables with the highest loading for a factor are taken for the reliability test. The result is an alpha value for each factor dimension. Only applies when data is a data frame and no prcomp object.

fillColor: A color palette for filling the geoms. If not specified, the 5th diverging color palette from the color brewer palettes (RdBu) is used, resulting in red colors for negative and blue colors for positive factor loadings, that become lighter the weaker the loadings are. Use any color palette that is suitable for the scale_fill_gradientn parameter of ggplot2.

majorGridColor: Specifies the color of the major grid lines of the diagram background.

minorGridColor: Specifies the color of the minor grid lines of the diagram background.

theme: Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.

  • Use "bw" for a white background with gray grids
  • "classic" for a classic theme (black border, no grids)
  • "minimal" for a minimalistic theme (no border, gray grids) or
  • "none" for no borders, grids and ticks.

printPlot: If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Invisibly) returns a structure with

  • the varimax-rotated factor loading matrix (varim)
  • the column indices of removed variables (for more details see next list item) (removed.colindex)
  • an updated data frame containing all factors that have a clear loading on a specific scale in case data was a data frame (See parameter factorLoadingTolerance for more details) (removed.df)
  • the ggplot-object (plot),
  • the data frame that was used for setting up the ggplot-object (df).

Note

This PCA uses the prcomp function and the varimax rotation.
References

http://strengejacke.wordpress.com/sjplot-r-package/

See Also

sjt.pca
sju.reliability
sjt.itemanalysis
sju.cronbach

Examples

# randomly create data frame with 7 items, each consisting of 4 categories
likert_4 <- data.frame(sample(1:4, 500, replace=TRUE, prob=c(0.2, 0.3, 0.1, 0.4)),
                        sample(1:4, 500, replace=TRUE, prob=c(0.5, 0.25, 0.15, 0.1)),
                        sample(1:4, 500, replace=TRUE, prob=c(0.4, 0.15, 0.25, 0.2)),
                        sample(1:4, 500, replace=TRUE, prob=c(0.25, 0.1, 0.4, 0.25)),
                        sample(1:4, 500, replace=TRUE, prob=c(0.1, 0.4, 0.4, 0.1)),
                        sample(1:4, 500, replace=TRUE),
                        sample(1:4, 500, replace=TRUE, prob=c(0.35, 0.25, 0.15, 0.25)))

# Create variable labels
colnames(likert_4) <- c("V1", "V2", "V3", "V4", "V5", "V6", "V7")

# plot results from PCA as square-tiled "heatmap"
sjp.pca(likert_4)

# manually compute PCA
pca <- prcomp(, retx=TRUE, center=TRUE, scale=TRUE)

# Data from the EUROFAMCARE sample dataset
# -----------------------------
data(efc)

# retrieve variable and value labels
varlabs <- sj.getVariableLabels(efc)

# receive first item of COPE-index scale
start <- which(colnames(efc)="$82cop1")
# receive last item of COPE-index scale
end <- which(colnames(efc)="$90cop9")

# create data frame with COPE-index scale
df <- as.data.frame(efc[,c(start:end)])
 hjjاه

sjp.reglin <- varlabs[c(start:end)]

sjp.pca(df)

# auto-detection of labels

efc <- sjil.setVariableLabels(efc, varlabs)
sjp.pca(efc[,c(start:end)])

sjp.reglin

Plot regression lines for each predictor

Description

Plot regression lines with confidence intervals for each single predictor of a fitted model. This
method extracts all predictors of a fitted model and fits each of them against the response variable.

This function plots two lines: The resulting linear regression line including confidence interval
(in blue) and a loess-smoothed line without confidence interval (in red). The better the linear rela-
tionship of predictor and response is, the more both lines should overlap (i.e. the red loess-smoothed
line is almost linear).

Furthermore, a scatter plot of response and predictor values is plotted.

Usage

sjp.reglin(fit, data, lineColor = "blue", showCI = TRUE, ciLevel = 0.95,
    pointAlpha = 0.2, pointColor = "black", showScatterPlot = TRUE,
    showLoess = TRUE, loessLineColor = "red", showLoessCI = FALSE,
    loessCiLevel = 0.95, printPlot = TRUE)

Arguments

fit The model of the linear regression (lm-Object).
data The data/dataset/dataframe used in the fitted model.
lineColor The color of the regression line. Default is "blue".
showCI If TRUE (default), a confidence region for the regression line will be plotted. Use
ciLevel to specify the confidence level.
ciLevel The confidence level of the confidence region. Only applies when showCI is
    TRUE. Default is 0.95.
pointAlpha The alpha values of the scatter plot’s point-geoms. Default is 0.2.
pointColor The color of the scatter plot’s point-geoms. Only applies when showScatterPlot
    is TRUE. Default is "black".
showScatterPlot If TRUE (default), a scatter plot of response and predictor values for each predic-
tor of the fitted model fit is plotted.
showLoess  If TRUE (default), an additional loess-smoothed line is plotted.
loessLineColor The color of the loess-smoothed line. Default is "red". Only applies if showLoess is TRUE.
showLoessCI If TRUE, a confidence region for the loess-smoothed line will be plotted. Default is FALSE. Use loessCiLevel to specify the confidence level. Only applies, if showLoess is TRUE.
loessCiLevel The confidence level of the loess-line's confidence region. Only applies, if showLoessCI is TRUE. Default is 0.95.
printPlot  If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Insibility) returns the ggplot-objects with the complete plot-list (plot.list) as well as the data frame that were used for setting up the ggplot-objects (df.list).

See Also

sjp.lm
sjp.scatter
sjp.lm.ma
sjp.lm.int

Examples

data(efc)
fit <- lm(tot_sc_e ~ c12hour + e17age + e42dep, data=efc)

# regression line and scatter plot
sjp.reglin(fit, efc)

# regression line w/o scatter plot
sjp.reglin(fit, efc, showScatterPlot=FALSE)

# regression line w/o CI
sjp.reglin(fit, efc, showCI=FALSE)
Usage

sjp.scatter(x, y, grp = NULL, title = NULL, titleSize = 1.3,
titleColor = "black", legendTitle = NULL, legendLabels = NULL,
axisTitle.x = NULL, axisTitle.y = NULL, axisTitleColor = "black",
axisTitleSize = 1.3, axisTickMarkSize = 1, axisTickMarkColor = "grey50",
breakTitleAt = 50, breakLegendTitleAt = 20, breakLegendLabelsAt = 20,
pointAlpha = 0.5, pointSize = 3, pointColors = NULL,
legendPos = "right", legendSize = 1, legendBorderColor = "white",
legendBackColor = "white", showTickMarkLabels.x = TRUE,
showTickMarkLabels.y = TRUE, showTickMarks = TRUE,
majorGridColor = NULL, minorGridColor = NULL, hideGrid.x = FALSE,
hideGrid.y = FALSE, borderColor = NULL, axisColor = NULL,
showGroupFitLine = FALSE, showTotalFitLine = FALSE, showSE = FALSE,
fitmethod = "lm", useJitter = FALSE, autoJitter = TRUE,
jitterRatio = 0.15, showRug = FALSE, hideLegend = FALSE, theme = NULL,
useFacetGrid = FALSE, printPlot = TRUE)

Arguments

x A vector (variable) indicating the x positions.
y A vector (variable) indicating the y positions.
grp A grouping variable. If not NULL, the scatter plot will be grouped. See examples below. Default is NULL, i.e. not grouping is done.
title Title of the diagram, plotted above the whole diagram panel. Use "auto" to automatically detect variable names that will be used as title (see sjisetVariableLabels) for details).
titleSize The size of the plot title. Default is 1.3.
titleColor The color of the plot title. Default is "black".
legendTitle Title of the diagram’s legend.
legendLabels Labels for the guide/legend.
axisTitle.x A label (title) for the x axis. Use "auto" to automatically detect variable names that will be used as title (see sjisetVariableLabels) for details).
axisTitle.y A label (title) for the y axis. Use "auto" to automatically detect variable names that will be used as title (see sjisetVariableLabels) for details).
axisTitleColor The color of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.
axisTitleSize The size of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.
axisTickMarkSize The size of tick mark values of both x and y axis. Default is 1, recommended values range between 0.5 and 3.0
axisTickMarkColor User defined color for tick mark values. If not specified, a default mid gray color will be used for the labels.
breakTitleAt  Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.
breakLegendTitleAt  Wordwrap for diagram legend title. Determines how many chars of the legend’s title are displayed in one line and when a line break is inserted.
breakLegendLabelsAt  Wordwrap for diagram legend labels. Determines how many chars of the legend labels are displayed in one line and when a line break is inserted.
pointAlpha  The alpha values of scattered points. Useful to better cope with overplotting. Default is 0.5
pointSize  The size of scattered points.
pointColors  The color(s) of scattered points. If `grp` is not NULL, groups are indicated by different colors, thus a vector with multiple color values has to be supplied. By default, the `Set1` palette of diverging palette type is chosen (see `http://colorbrewer2.org`).
legendPos  The position of the legend, if a legend is drawn. Use "bottom", "top", "left" or "right" to position the legend above, below, on the left or right side of the diagram. Right positioning is default.
legendSize  The text size of the legend. Default is 1. Relative size, so recommended values are from 0.3 to 2.5
legendBorderColor  Color of the legend’s border. Default is "white", so no visible border is drawn.
legendBackColor  Fill color of the legend’s background. Default is "white", so no visible background is drawn.
showTickMarkLabels.x  Whether x axis tick mark labels should be shown or not.
showTickMarkLabels.y  Whether y axis tick mark labels should be shown or not.
showTickMarks  Whether tick marks of axes should be shown or not.
showGroupFitLine  If TRUE, a fitted line for each group is drawn. See `fitmethod` to change the fit method of the fitted lines.
showTotalFitLine  If TRUE, a fitted line for the overall scatterplot is drawn. See `fitmethod` to change the fit method of the fitted line.
showSE  If TRUE, a shaded region indicating the standard error of the fitted lines will be added.
fitmethod  By default, a linear method ("lm") is used for fitting the fit lines. Possible values are for instance:
  * "lm"
  * "glm"
  * "loess"
  * "auto"
sjp.scatter

(see http://docs.ggplot2.org/current/stat_smooth.html for more details).

useJitter If TRUE, points will be jittered (to avoid overplotting).

autojitter If TRUE, points will be jittered according to an overlap-estimation. A matrix of x and y values is created and the amount of cells (indicating a unique point position) is calculated. If more than 15% (see jitterRatio) of the approximated amount of unique point coordinates seem to overlap, they are automatically jittered.

jitterRatio The ratio of tolerated overlapping (see autojitter). If approximated amount of overlapping points exceed this ration, they are automatically jittered. Default is 0.15. Valid values range between 0 and 1.

showRug If TRUE, a marginal rug plot is displayed in the graph (see http://docs.ggplot2.org/current/geom_rug.html for more details).

hideLegend Indicates whether legend (guide) should be shown or not.

borderColor User defined color of whole diagram border (panel border).

axisColor User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

majorGridColor Specifies the color of the major grid lines of the diagram background.

minorGridColor Specifies the color of the minor grid lines of the diagram background.

hideGrid.x If TRUE, the x-axis-gridlines are hidden. Default is FALSE.

hideGrid.y If TRUE, the y-axis-gridlines are hidden. Default is FALSE.

theme Specifies the diagram’s background theme. Default (parameter NULL) is a gray background with white grids.

• Use "bw" for a white background with gray grids
• "classic" for a classic theme (black border, no grids)
• "minimal" for a minimalistic theme (no border,gray grids) or
• "none" for no borders, grids and ticks.

useFacetGrid TRUE when each scatter plot group should be plotted as single facet instead of an integrated single graph. Only applies if grp is not NULL. Each category of grp will be plotted in an own facet.

printPlot If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.

Value

(Insisibly) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

References

• http://strengejacke.wordpress.com/sjplot-r-package/
• http://strengejacke.wordpress.com/2014/02/28/simply-creating-various-scatter-plots-with-ggplot
See Also

sjp.lm
sjp.reglin

Examples

# load sample data
data(efc)

# simple scatter plot, auto-jittering
sjp.scatter(efc$e16sex, efc$neg_c_7)

# simple scatter plot, no jittering needed
sjp.scatter(efc$c160age, efc$e17age)

# grouped scatter plot
sjp.scatter(efc$c160age, efc$e17age, efc$e42dep)

# grouped and jitted scatter plot with marginal rug plot
sjp.scatter(efc$e16sex, efc$neg_c_7, efc$c172code, showRug=TRUE)

# grouped and labelled scatter plot
sjp.scatter(efc$c160age, efc$e17age, efc$e42dep, title="Scatter Plot",
        legendTitle=sij.getVariableLabels(efc)['e42dep'],
        legendLabels=sij.getValueLabels(efc)['e42dep'],
        axisTitle.x=sij.getVariableLabels(efc)['c160age'],
        axisTitle.y=sij.getVariableLabels(efc)['e17age'],
        showGroupFitLine=TRUE)

# grouped and labelled scatter plot as facets
sjp.scatter(efc$c160age, efc$e17age, efc$e42dep, title="Scatter Plot",
        legendTitle=sij.getVariableLabels(efc)['e42dep'],
        legendLabels=sij.getValueLabels(efc)['e42dep'],
        axisTitle.x=sij.getVariableLabels(efc)['c160age'],
        axisTitle.y=sij.getVariableLabels(efc)['e17age'],
        showGroupFitLine=TRUE, useFacetGrid=TRUE, showSE=TRUE)

# auto-detection of labels
# ---------------------------
efc <- sij.setVariableLabels(efc, sij.getVariableLabels(efc))
sjp.scatter(efc$c160age, efc$e17age, efc$e42dep,
        title="auto", axisTitle.x="auto", axisTitle.y="auto")

sjp.stackfreq

Plot stacked proportional bars

Description

Plot items (variables) of a scale as stacked proportional bars. This function is useful when several items with identical scale/categories should be plotted to compare the distribution of answers.
Usage

sjp.stackfrq(items, legendLabels = NULL, orderBy = NULL, weightBy = NULL, weightByTitleString = NULL, hideLegend = FALSE, reverseOrder = TRUE, title = NULL, titleSize = 1.3, titleColor = "black", legendTitle = NULL, includeN = TRUE, axisLabels.y = NULL, axisLabelSize = 1.1, axisLabelAngle.x = 0, axisLabelColor = "gray30", valueLabelSize = 4, valueLabelColor = "black", breakTitleAt = 50, breakLabelsAt = 30, breakLegendTitleAt = 30, breakLegendLabelsAt = 28, gridBreaksAt = 0.2, diagramMargins = FALSE, barWidth = 0.5, barColor = NULL, colorPalette = "GnBu", barAlpha = 1, borderColor = NULL, axisColor = NULL, barOutline = FALSE, outlineColor = "black", majorGridColor = NULL, minorGridColor = NULL, hideGrid.x = FALSE, hideGrid.y = FALSE, axisTitle.x = NULL, axisTitle.y = NULL, axisTitleColor = "black", axisTitleSize = 1.3, theme = NULL, showTickMarks = FALSE, showValueLabels = TRUE, jitterValueLabels = FALSE, showItemLabels = TRUE, showSeparatorLine = FALSE, separatorLineColor = "grey80", separatorLineSize = 0.3, legendPos = "right", legendSize = 1, legendBorderColor = "white", legendBackColor = "white", flipCoordinates = TRUE, printPlot = TRUE)

Arguments

items A data frame with each column representing one likert-item.
legendLabels A list or vector of strings that indicate the likert-scale-categories and which appear as legend text.
orderBy Indicates whether the items should be ordered by highest count of first or last category of items. Use "first" to order ascending by lowest count of first category, "last" to order ascending by lowest count of last category or NULL (default) for no sorting. In case you want to reverse order (descending from highest count), use reverseOrder parameter.
weightBy A weight factor that will be applied to weight all cases from items.
weightByTitleString If a weight factor is supplied via the parameter weightBy, the diagram’s title may indicate this with a remark. Default is NULL, so the diagram’s title will not be modified when cases are weighted. Use a string as parameter, e.g.: weightByTitleString= "(weighted)".
hideLegend Indicates whether legend (guide) should be shown or not.
reverseOrder If TRUE, the item order on the x-axis is reversed.
title Title of the diagram, plotted above the whole diagram panel.
titleSize The size of the plot title. Default is 1.3.
titleColor The color of the plot title. Default is "black".
legendTitle Title of the diagram’s legend.
includeN If TRUE (default), the N of each item is included into axis labels.
axisLabels.y  Labels for the y-axis (the labels of the items). These parameters must be passed as list! Example: axisLabels.y=list(c("Q1", "Q2", "Q3")) Axis labels will automatically be detected, when they have a "variable.label" attribute (see sjp.setVariableLabels) for details.

axisLabelSize The size of category labels at the axes. Default is 1.1, recommended values range between 0.5 and 3.0

axisLabelAngle.x Angle for axis-labels.

axisLabelColor User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

valueLabelSize The size of value labels in the diagram. Default is 4, recommended values range between 2 and 8

valueLabelColor The color of value labels in the diagram. Default is black.

breakTitleAt Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

breakLegendTitleAt Wordwrap for diagram legend title. Determines how many chars of the legend's title are displayed in one line and when a line break is inserted.

breakLegendLabelsAt Wordwrap for diagram legend labels. Determines how many chars of the legend labels are displayed in one line and when a line break is inserted.

ggridBreaksAt Sets the breaks on the y axis, i.e. at every n'th position a major grid is being printed. Valid values range from 0 to 1.

diagramMargins If TRUE (default), the diagram has margins, i.e. the y-axis is not exceeded to the diagram’s boundaries.

barWidth Width of bars. Recommended values for this parameter are from 0.4 to 1.5

barColor User defined color for bars. If not specified (NULL), a default blue color palette will be used for the bar charts. You can use pre-defined color-sets that are independent from the amount of categories: If barColor is "brewer", use the colorPalette parameter to specify a palette of the color brewer Else specify your own color values as vector (e.g. barColor=c("darkred", "red", "green", "darkgreen"))

colorPalette If barColor is "brewer", specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.

barAlpha Specify the transpareny (alpha value) of bars.

borderColor User defined color of whole diagram border (panel border).

barOutline If TRUE, each bar gets a colored outline. Default is FALSE.

outlineColor The color of the bar outline. Only applies, if barOutline is set to TRUE.

majorGridColor Specifies the color of the major grid lines of the diagram background.

minorGridColor Specifies the color of the minor grid lines of the diagram background.

hideGrid.x If TRUE, the x-axis-gridlines are hidden. Default if FALSE.
hideGrid.y If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

axisColor User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

axisTitle.x A label for the x axis. Useful when plotting histograms with metric scales where no category labels are assigned to the x axis.

axisTitle.y A label for the y axis. Useful when plotting histograms with metric scales where no category labels are assigned to the y axis.

axisTitleColor The color of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

axisTitleSize The size of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

showValueLabels Whether counts and percentage values should be plotted to each bar.

jitterValueLabels If TRUE, the value labels on the bars will be "jittered", i.e. they have alternating vertical positions to avoid overlapping of labels in case bars are very short. Default is FALSE.

showItemLabels Whether x axis text (category names) should be shown or not.

showTickMarks Whether tick marks of axes should be shown or not.

showSeparatorLine If TRUE, a line is drawn to visually "separate" each bar in the diagram.

separatorLineColor The color of the separator line. only applies, if showSeparatorLine is TRUE.

separatorLineWidth The size of the separator line. only applies, if showSeparatorLine is TRUE.

legendPos The position of the legend. Default is "right". Use one of the following values: "right", "left", "bottom", "top".

legendSize The size of the legend.

legendBorderColor The border color of the legend.

legendBackColor The background color of the legend.

theme Specifies the diagram's background theme. Default (parameter NULL) is a gray background with white grids.

- Use "bw" for a white background with gray grids
- "classic" for a classic theme (black border, no grids)
- "minimal" for a minimalistic theme (no border, gray grids)
- "none" for no borders, grids and ticks.

flipCoordinates If TRUE, the x and y axis are swapped.

printPlot If TRUE (default), plots the results as graph. Use FALSE if you don’t want to plot any graphs. In either case, the ggplot-object will be returned as value.
Value

(Insisibily) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

Note

Thanks to Forrest Stevens (http://www.clas.fiu.edu/users/forrest/) for bug fixes

References

- http://strengejacke.wordpress.com/sjplot-r-package/
- http://strengejacke.wordpress.com/2013/07/17/plotting-likert-scales-net-stacked-distributions-

See Also

sjp.likert
sjt.stackfrq

Examples

# ------------------------------
# random sample
# ------------------------------
# prepare data for 4-category likert scale, 5 items
# prepare data for 4-category likert scale, 5 items
likert_4 <- data.frame(as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.2, 0.3, 0.1, 0.4))),
                       as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.5, 0.25, 0.15, 0.1))),
                       as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.25, 0.1, 0.4, 0.25))),
                       as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.1, 0.4, 0.4, 0.1))),
                       as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.35, 0.25, 0.15, 0.25))))

# create labels
levels_4 <- list(c("Independent", "Slightly dependent", "Dependent", "Severely dependent"))

# create item labels
items <- list(c("Q1", "Q2", "Q3", "Q4", "Q5"))

# plot stacked frequencies of 5 (ordered) item-scales
sjp.stackfrq(likert_4, legendLabels=levels_4, axisLabels.y=items)

# ------------------------------
# Data from the EUROFAMCARE sample dataset
# ------------------------------
data(efc)

# receive first item of COPE-index scale
start <- which(colnames(efc)=="c82cop1")

# receive first item of COPE-index scale
end <- which(colnames(efc)=="c90cop9")

# retrieve variable and value labels
sji.getVariableLabels(efc)

# create value labels. We need just one variable of
# the COPE-index scale because they have all the same
# level / categorie / value labels
levels <- vallabs['c82cop1']

# create item labels
items <- list(varlabs[c(start:end)])

sjp.stackfrq(efc[,c(start:end)], legendLabels=levels,
    axisLabels.y=items, jitterValueLabels=TRUE)

# auto-detection of labels
# sjp.setVariableLabels(efc, varlabs)
sjp.stackfrq(efc[,c(start:end)])

---

# sjp.vif

**Plot Variance Inflation Factors of linear models**

**Description**

Plots the Variance Inflation Factors (check for multicollinearity) of (generalized) linear models. Values below 5 are good and indicating no multicollinearity, values between 5 and 10 may be tolerable. Values greater than 10 are not acceptable and indicate multicollinearity between model’s predictors.

**Usage**

sjp.vif(fit)

**Arguments**

- **fit**
  
  The fitted (generalized) linear model which should be checked for multicollinearity.

**Value**

(invisibly) returns the VIF values.

**Examples**

# fit linear model
fit <- lm(airquality$Ozone ~ airquality$Wind + airquality$Temp + airquality$Solar.R)
# plot VIF values
sjp.vif(fit)
Plot contingency tables

Description

Plot proportional crosstables (contingency tables) of two variables as ggplot diagram.

Usage

sjp.xtab(y, x, title = NULL, titleSize = 1.3, titleColor = "black",
legendTitle = NULL, weightBy = NULL, weightByTitleString = NULL,
type = "bars", tableIndex = "col", reverseOrder = FALSE,
maxYlim = TRUE, upperYlim = NULL, axisLabels.x = NULL,
axisLabelColor.x = "darkgray", axisLabelAngle.x = 0,
axisLabelSize.x = 1.1, legendLabels = NULL, valueLabelSize = 4,
valueLabelColor = "black", valueLabelPosOnTop = TRUE,
stringTotal = "Total", breakTitleAt = 50, breakLabelsAt = 12,
bbreakLegendTitleAt = 20, breakLegendLabelsAt = 20, gridBreaksAt = 0.2,
barWidth = 0.7, barSpace = 0.1, barPosition = "dodge",
barColor = NULL, colorPalette = "GnBu", barAlpha = 1, lineType = 1,
lineSize = 1, lineAlpha = 1, lineDotSize = 3, smoothLines = FALSE,
borderColor = NULL, axisColor = NULL, barOutline = FALSE,
outlineColor = "black", majorGridColor = NULL, minorGridColor = NULL,
hideGrid.x = FALSE, hideGrid.y = FALSE, showValueLabels = TRUE,
jitterValueLabels = FALSE, showCategoryLabels = TRUE,
showTickMarks = TRUE, showTableSummary = TRUE, tableSummaryPos = "r",
showTotalColumn = TRUE, hideLegend = FALSE, axisTitle.x = NULL,
axisTitle.y = NULL, axisTitleColor = "black", axisTitleSize = 1.3,
theme = NULL, flipCoordinates = FALSE, printPlot = TRUE)

Arguments

y The variable which proportions (percentage values) should be plotted. The percentage proportion (within table row, table column or complete table, see parameter tableIndex of this variable are along the y-axis, the variable’s categories on the x-axis.

x The grouping variable, where each value represents a single bar chart within each category of the y variable.

weightBy A weight factor that will be applied to weight all cases from y.

weightByTitleString If a weight factor is supplied via the parameter weightBy, the diagram’s title may indicate this with a remark. Default is NULL, so the diagram’s title will not be modified when cases are weighted. Use a string as parameter, e.g.: weightByTitleString=" (weighted)".

type The plot type. may be either "b", "bar", "bars" (default) for bar charts, or "l", "line", "lines" for line diagram.
tableIndex  Indicates which data from the proportional table should be plotted. Use "row" for calculating row percentages, "col" for column percentages and "cell" for cell percentages. Only when tableIndex is "col", an additional bar chart with the total sum of each column (i.e. of each category on the x-axis) can be added with the parameter showTotalColumn.

barPosition  Indicates whether bars should be positioned side-by-side (default, or use "dodge" as parameter) or stacked (use "stack" as parameter).

hideLegend  Indicates whether legend (guide) should be shown or not. Default is FALSE, thus the legend is shown.

reverseOrder  Whether the categories along the x-axis should appear in reversed order or not.

maxYlim  Indicates how to calculate the maximum limit of the y-axis. If TRUE, the y-axes ranges from 0 to 100 depends on the highest percentage value of a variable’s answer category. In this case, the y-axis breaks may change, depending on the variable.

upperYlim  Uses a pre-defined upper limit for the y-axis. Overrides the maxYlim parameter.

title  Title of the diagram, plotted above the whole diagram panel. Use "auto" to automatically detect variable names that will be used as title (see sjisetVariableLabels for details).

titlesize  The size of the plot title. Default is 1.3.

titleColor  The color of the plot title. Default is "black".

legendTitle  Title of the diagram’s legend.

axisLabels.x  Labels for the x-axis breaks.

legendLabels  Labels for the guide/legend.

axisLabelSize.x  The size of category labels at the axes. Default is 1.1, recommended values range between 0.5 and 3.0.

valueLabelSize  The size of value labels in the diagram. Default is 4, recommended values range between 2 and 8.

axisLabelAngle.x  Angle for axis-labels.

breakTitleAt  Wordwrap for diagram title. Determines how many chars of the title are displayed in one line and when a line break is inserted into the title.

breakLabelsAt  Wordwrap for diagram labels. Determines how many chars of the category labels are displayed in one line and when a line break is inserted.

breakLegendTitleAt  Wordwrap for diagram legend title. Determines how many chars of the legend’s title are displayed in one line and when a line break is inserted.

breakLegendLabelsAt  Wordwrap for diagram legend labels. Determines how many chars of the legend labels are displayed in one line and when a line break is inserted.

gridBreaksAt  Sets the breaks on the y axis, i.e. at every n’th position a major grid is being printed. Valid values range from 0 to 1.

barWidth  Width of bars. Recommended values for this parameter are from 0.4 to 1.5.
barSpace  Spacing between bars. If unchanges, the grouped bars are sticked together and have no space in between. Recommended values for this parameter are from 0 to 0.5.

barColor  User defined color for bars.
  - If not specified (NULL), a default red-green-yellow color palette will be used for the bar charts.
  - If barColor is "gs", a greyscale will be used.
  - If barColor is "bw", a monochrome white filling will be used.
  - If barColor is "brewer", use the colorPalette parameter to specify a palette of the color brewer.

Else specify your own color values as vector (e.g. barColor=c("#f00000", "#00ff00", "#0080ff")).

colorPalette  If barColor is "brewer", specify a color palette from the color brewer here. All color brewer palettes supported by ggplot are accepted here.

barAlpha  Specify the transpareny (alpha value) of bars.

lineType  The linetype when using line diagrams. Only applies, when parameter type is set to "lines".

lineSize  The size of lines in a line diagram. Only applies, when parameter type is set to "lines".

lineAlpha  The alpha value of lines in a line diagram. Only applies, when parameter type is set to "lines".

lineDotSize  Size of dots. Only applies, when parameter type is set to "lines".

smoothLines  Prints a smooth line curve. Only applies, when parameter type is set to "lines".

axisLabelColor.x  User defined color for axis labels. If not specified, a default dark gray color palette will be used for the labels.

borderColor  User defined color of whole diagram border (panel border).

axisColor  User defined color of axis border (y- and x-axis, in case the axes should have different colors than the diagram border).

barOutline  If TRUE, each bar gets a colored outline. Default is FALSE.

outlineColor  The color of the bar outline. Only applies, if barOutline is set to TRUE

majorGridColor  specifies the color of the major grid lines of the diagram background

minorGridColor  specifies the color of the minor grid lines of the diagram background

hideGrid.x  If TRUE, the x-axis-gridlines are hidden. Default if FALSE.

hideGrid.y  If TRUE, the y-axis-gridlines are hidden. Default if FALSE.

showValueLabels  Whether counts and percentage values should be plotted to each bar

jitterValueLabels  If TRUE, the value labels on the bars will be "jittered", i.e. they have alternating vertical positions to avoid overlapping of labels in case bars are very short. Default is FALSE.
valueLabelPosOnTop
Whether value labels should be displayed on top of dodged bars or inside the bars. Default is TRUE, i.e. the value labels are displayed on top of the bars. Only applies if parameter barPosition is dodge (default).

stringTotal
The string for the legend label when a total-column is added. Only applies if showTotalColumn is TRUE. Default is "Total".

showCategoryLabels
Whether x axis text (category names) should be shown or not.

showTickMarks
Whether tick marks of axes should be shown or not.

showTableSummary
If TRUE (default), a summary of the cross tabulation with N, Chi-square (see chisq.test), df, Cramer's V or Phi-value and p-value is printed to the upper right corner of the diagram. If a cell contains expected values lower than five (or lower than 10 if df is 1), the Fisher's exact test (see fisher.test) is computed instead of Chi-square test. If the table's matrix is larger than 2x2, Fisher's exact test with Monte Carlo simulation is computed. Only applies to bar-charts or dot-plots, i.e. when parameter type is either "bars" or "dots".

tableSummaryPos
Position of the model summary which is printed when showTableSummary is TRUE. Default is "r", i.e. it's printed to the upper right corner. Use "l" for upper left corner.

showTotalColumn
If tableIndex is "col", an additional bar chart with the sum within each category and its percentages will be added to each category.

valueLabelColor
The color of the value labels (numbers) inside the diagram.

axisTitle.x
A label for the x axis. Useful when plotting histograms with metric scales where no category labels are assigned to the x axis. Use "auto" to automatically detect variable names that will be used as title (see sji.setVariableLabels) for details.

axisTitle.y
A label for the y axis. Useful when plotting histograms with metric scales where no category labels are assigned to the y axis.

axisTitleColor
The color of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

axisTitleSize
The size of the x and y axis labels. Refers to axisTitle.x and axisTitle.y, not to the tick mark or category labels.

theme
Specifies the diagram's background theme. Default (parameter NULL) is a gray background with white grids.
- Use "bw" for a white background with gray grids
- "classic" for a classic theme (black border, no grids)
- "minimal" for a minimalistic theme (no border, gray grids) or
- "none" for no borders, grids and ticks.

flipCoordinates
If TRUE, the x and y axis are swapped.

printPlot
If TRUE (default), plots the results as graph. Use FALSE if you don't want to plot any graphs. In either case, the ggplot-object will be returned as value.
Value

(Invisibility) returns the ggplot-object with the complete plot (plot) as well as the data frame that was used for setting up the ggplot-object (df).

Note

Since package version 1.3, the parameters `axisLabels.x` and `legendLabels`, which represent the value labels, are retrieved automatically if a) the variables `x` and `y` come from a data frame that was imported with the `sj.te.SPSS` function (because then value labels are attached as attributes to the data) or b) when the variables are factors with named factor levels (e.g., see column `group` in dataset `PlantGrowth`). However, you still can use own parameters as axis- and legendlabels.

References


See Also

`sjt.xtab`

Examples

```r
# create 4-category-items
x <- sample(1:4, 100, replace=TRUE)
# create 3-category-items
y <- sample(1:3, 100, replace=TRUE)

# plot "cross tabulation" of x and y
sjp.xtab(y,x)

# plot "cross tabulation" of x and y, including labels
sjp.xtab(y,x, axisLabels.x=c("low", "mid", "high"),
         legendLabels=c("Grp 1", "Grp 2", "Grp 3", "Grp 4"))

# plot "cross tabulation" of x and y as stacked proportional bars
sjp.xtab(y,x, tableIndex="row", barPosition="stack", flipCoordiates=TRUE)

# grouped bars with EUROFAMCARE sample dataset
# dataset was imported from an SPSS-file, using:
# efc <- sj.te.SPSS("efc.sav", enc="UTF-8")
data(efc)
efc.val <- sj.te.getValueLabels(efc)
efc.var <- sj.te.getVariableLabels(efc)

sjp.xtab(efc$e42dep,
          efc$e16sex,
          title=efc.var["e42dep"],
          axisLabels.x=efc.val[['e42dep']],
          legendTitle=efc.var["e16sex"],
          legendLabels=efc.val[['e16sex']])
```
sjt.corr

Show correlations as HTML table

Description

Shows the results of a computed correlation as HTML table. Requires either a data frame or a computed `cor`-object.

Usage

```r
sjt.corr(data, missingDeletion = "pairwise", corMethod = "spearman", showPValues = TRUE, pvaluesAsNumbers = FALSE, fadeNS = TRUE, file = NULL, varlabels = NULL, breakLabelsAt = 40, digits = 3, encoding = "UTF-8", CSS = NULL, useViewer = TRUE, noOutput = FALSE)
```

Arguments

data A correlation object, built with the R-`cor`-function, or a data frame which correlations should be calculated.

missingDeletion Indicates how missing values are treated. May be either "listwise" or "pairwise" (default).

corMethod Indicates the correlation computation method. May be one of "spearman" (default), "pearson" or "kendall".
showPValues  
Whether significance levels (p-values) of correlations should be printed or not.
"pvaluesAsNumbers"
If TRUE, the significance levels (p-values) are printed as numbers. If FALSE (default), asterisks are used.

"fadeNS"
If TRUE (default), non-significant correlation-values appear faded (by using a lighter grey text color).

"file"
The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and openended either in the RStudio View pane or in the default web browser.

"varlabels"
The item labels that are printed along the first column/row. If no item labels are provided (default), the data frame’s column names are used. Item labels must be a string vector, e.g.: varlabels=c("Var 1", "Var 2", "Var 3"). varlabels are detected automatically if data is a data frame where each variable has a "variable.label" attribute (see sji.setVariableLabels() for details).

"breakLabelsAt"
Wordwrap for diagram labels. Determines how many chars of the variable labels are displayed in one line and when a line break is inserted. Default is 40.

"digits"
The amount of digits used the values inside table cells. Default is 2.

"encoding"
The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g. German umlauts).

"CSS"
A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:
1. the class-names with "css.-"-prefix as parameter name and
2. each style-definition must end with a semicolon
Examples:
- css.table='border:2px solid red;' for a solid 2-pixel table border in red.
- css.summary='font-weight:bold;' for a bold fontweight in the summary row.
- css.lasttablerow='border-bottom: 1px dotted blue;' for a blue dotted border of the last table row.

See further examples below.

"useViewer"
If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

"no.output"
If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with
• the web page style sheet (page.style),
• the web page content (page.content),
• the complete html-output (output.complete) and
• the html-table with inline-css for use with knitr (knitr)

for further use.

Note

The HTML tables can either be saved as file and manually opened (specify parameter file) or they can be saved as temporary files and will be displayed in the RStudio Viewer pane (if working with RStudio) or opened with the default web browser. Displaying resp. opening a temporary file is the default behaviour (i.e. file=NULL).

References

• http://strengejacke.wordpress.com/sjplot-r-package/
• http://strengejacke.wordpress.com/2014/03/04/beautiful-table-outputs-in-r-part-2-rstats-sjplot

See Also

sjp.corr

Examples

# create data frame with 5 random variables
df <- as.data.frame(cbind(rnorm(10), rnorm(10), rnorm(10), rnorm(10), rnorm(10)))

# plot correlation matrix using circles
## Not run:
sjt.corr(df)
## End(Not run)

# Data from the EUROFAMCARE sample dataset
# -----------------------------
data(efc)

# retrieve variable and value labels
varlabs <- sji.getVariableLabels(efc)

# receive first item of COPE-index scale
start <- which(colnames(efc)=="c83cop2")
# receive last item of COPE-index scale
end <- which(colnames(efc)=="c88cop7")

# create data frame with COPE-index scale
df <- as.data.frame(efc[,c(start:end)])
colnames(df) <- varlabs[c(start:end)]

# we have high correlations here, because all items
# belong to one factor. See example from "sjp.pca".
## Not run:
sjt.corr(df, pvaluesAsNumbers=TRUE)
## End(Not run)

# -------------------------------
# auto-detection of labels
# -------------------------------
efc <- sji.setVariableLabels(efc, varlabs)
## Not run:
sjt.corr(efc[,c(start:end)])
## End(Not run)

sjt.df  
Show (description of) data frame as HTML table

Description
Show description or the content of data frame (rows and columns) as HTML table, or saves it as file. Helpful if you want a quick overview of a data frame's content. See parameter describe for details. By default, describe is TRUE and a description of the data frame is given, using the describe function of the psych package.

Usage

sjt.df(df, describe = TRUE, file = NULL, alternateRowColors = FALSE, 
orderColumn = NULL, orderAscending = TRUE, title = NULL, 
repeatHeader = FALSE, stringVariable = "Variable", showType = FALSE, 
showCommentRow = FALSE, commentString = "No comment...", 
encoding = "UTF-8", CSS = NULL, useViewer = TRUE, no.output = FALSE)

Arguments

df          A data frame that should be printed.
describe     If TRUE (default), a description of the data frame’s variables is given. The description is retrieved from the describe function of the psych package. If this parameter is FALSE, the data frame’s content (values) is shown.
file         The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and opened either in the RStudio View pane or in the default web browser.
alternateRowColors  If TRUE, alternating rows are highlighted with a light gray background color.
orderColumn  Indicates a column, either by column name or by column index number, that should be ordered. Default order is ascending, which can be changed with orderAscending parameter. Default is NULL, hence the data frame is printed with no specific order. See examples for further details.
orderAscending  If TRUE (default) and orderColumn is not NULL, data frame is ordered according to the specified column in an ascending order. Use FALSE to apply descending order. See examples for further details.

title  A table caption. By default, title is NULL, hence no title will be used.

stringVariable  A string used for the first column name. Default is "Variable".

repeatHeader  If TRUE, the header row will also be added at the bottom at the table. This might be helpful, if you have longer tables and want to see the column names at the end of the table as well.

showType  If TRUE, the variable type is shown in a separate row below the column names.

showCommentRow  If TRUE, an optional comment line can be added to the end / below the table. Use commentString to specify the comment.

commentString  A string that will be added to the end / below the table. Only applies, if showCommentRow is TRUE.

encoding  The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g.) German umlauts).

CSS  A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:
1. the class-names with "css."-prefix as parameter name and
2. each style-definition must end with a semicolon
Examples:
• css.table='border:2px solid red;' for a solid 2-pixel table border in red.
• css.summary='font-weight:bold;' for a bold fontweight in the summary row.
• css.arc='color:blue;' for a blue text color each 2nd row.
See further examples below.

useViewer  If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

no.output  If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with
• the data frame with the description information (data),
• the web page style sheet (page.style),
• the web page content (page.content),
• the complete html-output \( \text{output.complete} \) and
• the html-table with inline-css for use with knitr (knitr)

for further use.

References

• http://strengejacke.wordpress.com/sjplot-r-package/
• http://strengejacke.wordpress.com/2014/03/04/beautiful-table-outputs-in-r-part-2-rstats-sjplot/

See Also

\text{sjj.viewSPSS}

Examples

```r
# init dataset
data(efc)

# plot efc-data frame summary
## Not run:
sjt.df(efc, alternateRowColors=TRUE)
## End(Not run)

# plot content, first 50 rows of first 5 columns of example data set
## Not run:
sjt.df(efc[1:50,1:5], describe=FALSE, stringVariable="Observation")
## End(Not run)

# plot efc-data frame summary, ordered descending by mean-column
## Not run:
sjt.df(efc, orderColumn="mean", orderAscending=FALSE)
## End(Not run)

# plot first 20 rows of first 5 columns of example data set, ordered by column "e42dep" with alternating row colors
## Not run:
sjt.df(efc[1:20,1:5], alternateRowColors=TRUE,
          orderColumn="e42dep", describe=FALSE)
## End(Not run)

# plot first 20 rows of first 5 columns of example data set, ordered by 4th column in descending order.
## Not run:
sjt.df(efc[1:20,1:5], orderColumn=4, orderAscending=FALSE, describe=FALSE)
## End(Not run)

# User defined style sheet
# ------------------------------------------
## Not run:
sjt.df(efc,
```
**Show frequencies as HTML table**

**Description**

Shows (multiple) frequency tables as HTML file, or saves them as file.

**Usage**

```r
sjt.frq(data, file = NULL, weightBy = NULL, variableLabels = NULL, valueLabels = NULL, autoGroupAt = NULL, alternateRowColors = FALSE, stringValue = "value", stringCount = "N", stringPerc = "raw \%", stringValidPerc = "valid \%", stringCumPerc = "cumulative \%", stringMissingValue = "missings", highlightMedian = FALSE, highlightQuartiles = FALSE, skipZeroRows = FALSE, showSummary = TRUE, showSkew = FALSE, showKurtosis = FALSE, skewString = "\&gamma;", kurtosisString = "\&omega;", encoding = "UTF-8", CSS = NULL, useViewer = TRUE, no.output = FALSE)
```

**Arguments**

- **data**
  The variables which frequencies should be printed as table. Either use a single variable (vector) or a data frame where each column represents a variable (see examples).

- **file**
  The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and openend either in the RStudio View pane or in the default web browser.

- **weightBy**
  A weight factor that will be applied to weight all cases from data. default is NULL, so no weights are used.

- **variableLabels**
  A single character vector or a list of character vectors that indicate the variable names of those variables from data and will be used as variable labels in the output. Note that if multiple variables are supplied (as data frame), the variable labels must be supplied as list object (see examples).

- **valueLabels**
  A list of character vectors that indicate the value labels of those variables from data. Note that if multiple variables are supplied (as data frame), the value labels must be supplied as nested list object (see examples).

- **autoGroupAt**
  A value indicating at which length of unique values a variable from data is automatically grouped into smaller units (see `sju.groupVar`). Variables with large numbers of unique values may be too time consuming when a HTML table is created and R would not respond any longer. Hence it's recommended to group...
such variables. Default value is 50, i.e. variables with 50 and more unique values will be grouped using \texttt{sju.groupVar} with \texttt{groupsize="auto"} parameter. By default, the maximum group count is 30. However, if \texttt{autoGroupAt} is less than 30, \texttt{autoGroupAt} groups are built. Default value is \texttt{NULL}, i.e. auto-grouping is turned off.

\texttt{alternateRowColors}

If \texttt{TRUE}, alternating rows are highlighted with a light gray background color.

\texttt{stringValue}

String label for the very first table column containing the values (see \texttt{valueLabels}).

\texttt{stringCount}

String label for the first table data column containing the counts. Default is \texttt{"N"}.

\texttt{stringPerc}

String label for the second table data column containing the percentages, where the count percentages include missing values.

\texttt{stringValidPerc}

String label for the third data table column containing the valid percentages, i.e. the count percentage value excluding possible missing values.

\texttt{stringCumPerc}

String label for the last table data column containing the cumulative percentages.

\texttt{stringMissingValue}

String label for the last table data row containing missing values.

\texttt{highlightMedian}

If \texttt{TRUE}, the table row indicating the median value will be highlighted.

\texttt{highlightQuartiles}

If \texttt{TRUE}, the table row indicating the lower and upper quartiles will be highlighted.

\texttt{skipZeroRows}

If \texttt{TRUE}, rows with only zero-values are not printed. Default is \texttt{FALSE}.

\texttt{showSummary}

If \texttt{TRUE} (default), a summary row with total and valid N as well as mean and standard deviation is shown.

\texttt{showSkew}

If \texttt{TRUE}, the variable’s skewness is added to the summary. The skewness is retrieved from the \texttt{describe} function of the \texttt{psych} package.

\texttt{showKurtosis}

If \texttt{TRUE}, the variable’s kurtosis is added to the summary. The kurtosis is retrieved from the \texttt{describe} function of the \texttt{psych} package.

\texttt{skewString}

A character string, which is used as header for the skew column (see \texttt{showSkew}). Default is lower case Greek gamma.

\texttt{kurtosisString}

A character string, which is used as header for the kurtosis column (see \texttt{showKurtosis}). Default is lower case Greek omega.

\texttt{encoding}

The charset encoding used for variable and value labels. Default is \texttt{"UTF-8"}. Change encoding if specific chars are not properly displayed (e.g.) German umlauts).

\texttt{CSS}

A \texttt{list} with user-defined style-sheet-definitions, according to the official CSS syntax (see \texttt{http://www.w3.org/Style/CSS/}). See return value \texttt{page.style} for details of all style-sheet-classnames that are used in this function. Parameters for this list need:

1. the class-names with \texttt{"css."}-prefix as parameter name and
2. each style-definition must end with a semicolon

Examples:
• css.table='border:2px solid red;' for a solid 2-pixel table border in red.
• css.summary='font-weight:bold;' for a bold fontweight in the summary row.
• css.lasttablerow='border-bottom: 1px dotted blue;' for a blue dotted border of the last table row.

See further examples below.

useViewer If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

no.output If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with

• the web page style sheet (page.style),
• each frequency table as web page content (page.content.list),
• the complete html-output (output.complete) and
• the html-table with inline-css for use with knitr (knitr)

for further use.

Note

The HTML tables can either be saved as file and manually opened (specify parameter file) or they can be saved as temporary files and will be displayed in the RStudio Viewer pane (if working with RStudio) or opened with the default web browser. Displaying resp. opening a temporary file is the default behaviour (i.e. file=NULL).

References

• http://strengejacke.wordpress.com/sjplot-r-package/
• http://strengejacke.wordpress.com/2014/02/20/no-need-for-spss-beautiful-output-in-r-rstats/

See Also

sjp.frq
sjt.xtab

Examples

# load sample data
data(efc)

# retrieve value and variable labels
variables <- sji.getVariableLabels(efc)
values <- sji.getValueLabels(efc)

# show frequencies of "e42dep" in RStudio Viewer Pane
# or default web browser
## Not run:
sjt.fraq(efc$e42dep)
## End(Not run)

# plot and show frequency table of "e42dep" with labels
## Not run:
sjt.fraq(efc$e42dep,
       variableLabels=variables['e42dep'],
       valueLabels=values[['e42dep']])
## End(Not run)

# plot frequencies of e42dep, e16sex and c172code in one HTML file
# and show table in RStudio Viewer Pane or default web browser
## Not run:
sjt.fraq(as.data.frame(cbind(efc$e42dep, efc$e16sex, efc$c172code)),
       variableLabels=list(variables['e42dep'], variables['e16sex'], variables['c172code']),
       valueLabels=list(values['e42dep'], values['e16sex'], values['c172code']))
## End(Not run)

# plot larger scale including zero-counts
# indicating median and quartiles
## Not run:
sjt.fraq(efc$neg_c_7,
       variableLabels=variables['neg_c_7'],
       valueLabels=values['neg_c_7'],
       highlightMedian=TRUE,
       highlightQuartiles=TRUE)
## End(Not run)

## -----------------------------
# auto-detection of labels
## -----------------------------
efc <- sji.setVariableLabels(efc, variables)
## Not run:
sjt.fraq(data.frame(efc$e42dep, efc$e16sex, efc$c172code))
## End(Not run)

## -----------------------------
# User defined style sheet
## -----------------------------
## Not run:
sjt.fraq(efc$e42dep,
       variableLabels=variables['e42dep'],
       valueLabels=values[['e42dep']],
       CSS=list(css.table="border: 2px solid;",
                css.tdata="border: 1px solid;",
                css.firsttablecol="color:#003399; font-weight:bold;"))
## End(Not run)
**sjt.glm**  
*Show (and compare) generalized linear models as HTML table*

---

**Description**

Shows (and compares multiple) generalized linear models (Odds Ratios) as HTML table, or saves them as file. The fitted glm’s should have the same predictor variables and either

- differ only in their response (dependent variable), to see the effect of a specific set of predictors on different responses, or
- all have the same response variables, but differ in their family objects and link function in order to see which model fits best to the data.

See parameter `showFamily` for details and section examples.

**Usage**

```
sjt.glm(..., file = NULL, labelPredictors = NULL,  
labelDependentVariables = NULL, stringPredictors = "Predictors",  
stringDependentVariables = "Dependent Variables",  
showHeaderStrings = FALSE, stringModel = "Model",  
stringIntercept = "(Intercept)", stringObservations = "Observations",  
stringOR = "OR", stringCI = "CI", stringSE = "std. Error",  
stringP = "p", digits.est = 2, digits.p = 3, digits.ci = 2,  
digits.se = 2, digits.summary = 3, exp.coef = TRUE,  
pvaluesAsNumbers = FALSE, boldpvalues = TRUE, showConfInt = TRUE,  
showStdError = FALSE, separateConfColumn = FALSE, newValueConf = TRUE,  
showAbbrHeadline = TRUE, showPseudoR = TRUE, showLogLik = FALSE,  
showAIC = FALSE, showChi2 = FALSE, showFamily = FALSE,  
cellSpacing = 0.2, encoding = "UTF-8", CSS = NULL, useViewer = TRUE,  
no.output = FALSE)
```

**Arguments**

- `...` One or more fitted glm-objects.
- `file` The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and openend either in the RStudio View pane or in the default web browser.
- `labelPredictors` Labels of the predictor variables, provided as char vector.
- `labelDependentVariables` Labels of the dependent variables of all fitted models which have been used as first parameter(s), provided as char vector.
- `stringPredictors` String constant used as headline for the predictor column. Default is "Predictors".
stringDependentVariables
String constant used as headline for the dependent variable columns. Default is "Dependent Variables".

showHeaderStrings
If TRUE, the header strings stringPredictors and stringDependentVariables are shown. By default, they're hidden.

stringModel
String constant used as headline for the model names in case no labels for the dependent variables are provided (see labelDependentVariables). Default is "Model".

stringIntercept
String constant used as headline for the Intercept row default is "Intercept".

stringObservations
String constant used in the summary row for the count of observation (cases). Default is "Observations".

stringOR
String used for the column heading of odds ratio values. Default is "OR".

stringCI
String used for the column heading of confidence interval values. Default is "CI".

stringSE
String used for the column heading of standard error values. Default is "std. Error".

stringP
String used for the column heading of p values. Default is "p".

digits.est
Amount of decimals for estimators.

digits.p
Amount of decimals for p-values.

digits.ci
Amount of decimals for confidence intervals.

digits.se
Amount of decimals for standard error.

digits.summary
Amount of decimals for values in model summary.

exp.coef
If TRUE (default), regression coefficients and confidence intervals are exponentiated (exp(coef(fit))). Use FALSE if you want the non-exponentiated coefficients as they are provided by the summary function.

pvaluesAsNumbers
If TRUE, p-values are shown as numbers. If FALSE (default), p-values are indicated by asterisks.

boldPvalues
If TRUE (default), significant p-values are shown bold faced.

showConfInt
If TRUE (default), the confidence intervall is also printed to the table. Use FALSE to omit the CI in the table.

showStdError
If TRUE, the standard errors are also printed. Default is FALSE.

separateConfColumn
if TRUE, the CI values are shown in a separate table column. Default is FALSE.

newLineConf
If TRUE and separateConfColumn is FALSE, inserts a line break between OR and CI values. If FALSE, CI values are printed in the same line with OR values.

showAbbrHeadline
If TRUE (default), the table data columns have a headline with abbreviations for odds ratios, confidence interval and p-values.

showPseudoR
If TRUE (default), the pseudo R2 values for each model are printed in the model summary. R2cs is the Cox-Snell-pseudo R-square value, R2n is Nagelkerke's pseudo R-square value.
showLogLik  If TRUE, the Log-Likelihood for each model is printed in the model summary. Default is FALSE.

showAIC   If TRUE, the AIC value for each model is printed in the model summary. Default is FALSE.

showChi2  If TRUE, the chi-square value for each model is printed in the model summary. Default is FALSE.

showFamily If TRUE, the family object and link function for each fitted model are printed. Can be used in case you want to compare models with different link functions and same predictors and response, to decide which model fits best. See family for more details. It is recommended to inspect the model AIC (see showAIC) to get a decision help for which model to choose.

cellSpacing  The inner padding of table cells. By default, this value is 0.2 (measure is cm), which is suitable for viewing the table. Decrease this value (0.05 to 0.1) if you want to import the table into Office documents. This is a convenient parameter for the CSS parameter for changing cell spacing, which would be: CSS=list(css.thead="padding:0.2cm;", css.tzdata="padding:0.2cm;").

encoding  The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g.) German umlauts).

CSS  A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:

  1. the class-names with "css."-prefix as parameter name and
  2. each style-definition must end with a semicolon

Examples:

- css.table='border:2px solid red;' for a solid 2-pixel table border in red.
- css.summary='font-weight:bold;' for a bold fontweight in the summary row.
- css.lasttablerow='border-bottom: 1px dotted blue;' for a blue dotted border of the last table row.

See further examples below.

useViewer  If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

no.output  If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with

- the web page style sheet (page.style),
the web page content (page.content),
the complete html-output (output.complete) and
the html-table with inline-css for use with knitr (knitr)

for further use.

Note

The HTML tables can either be saved as file and manually opened (specify parameter file) or they can be saved as temporary files and will be displayed in the RStudio Viewer pane (if working with RStudio) or opened with the default web browser. Displaying resp. opening a temporary file is the default behaviour (i.e. file=NULL).

References

• http://strengejacke.wordpress.com/sjplot-r-package/
• http://strengejacke.wordpress.com/2013/08/20/print-glm-output-to-html-table-rstats/

See Also

sjt.lm
sjp.glm

Examples

# prepare dummy variables for binary logistic regression
y1 <- ifelse(swiss$Fertility<median(swiss$Fertility), 0, 1)
y2 <- ifelse(swiss$Infant.Mortality<median(swiss$Infant.Mortality), 0, 1)
y3 <- ifelse(swiss$Agriculture<median(swiss$Agriculture), 0, 1)

# Now fit the models. Note that both models share the same predictors
# and only differ in their dependent variable (y1, y2 and y3)
fitOR1 <- glm(y1 ~ swiss$Education+swiss$Examination+swiss$Catholic,
             family=binomial(link="logit"))
fitOR2 <- glm(y2 ~ swiss$Education+swiss$Examination+swiss$Catholic,
             family=binomial(link="logit"))
fitOR3 <- glm(y3 ~ swiss$Education+swiss$Examination+swiss$Catholic,
             family=binomial(link="logit"))

# open HTML-table in RStudio Viewer Pane or web browser
## Not run:
sjt.glm(fitOR1, fitOR2, labelDependentVariables=c("Fertility", "Infant Mortality"),
        labelPredictors=c("Education", "Examination", "Catholic"))
## End(Not run)

# open HTML-table in RStudio Viewer Pane or web browser,
# table indicating p-values as numbers
## Not run:
sjt.glm(fitOR1, fitOR2, labelDependentVariables=c("Fertility", "Infant Mortality"),
        labelPredictors=c("Education", "Examination", "Catholic"),
        pvaluesAsNumbers=TRUE)
## sjt.itemanalysis

Show item analysis of an item scale as HTML table
Description

This function performs an item analysis with certain statistics that are useful for scale / index development. The resulting tables are shown in the viewer pane / webbrowser or can be saved as file. Following statistics are computed for each item of a data frame:

- percentage of missing values
- mean value
- standard deviation
- skew
- item difficulty
- item discrimination
- Cronbach’s Alpha if item was removed from scale

If factor.groups is not NULL, the data frame df will be splitted into groups, assuming that factor.groups indicate those columns of the data frame that belong to a certain factor (see return value of function sjt.pca as example for retrieving factor groups for a scale and see examples for more details).

Usage

sjt.itemanalysis(df, factor.groups = NULL, scaleItems = FALSE, alternateRowColors = TRUE, orderColumn = NULL, orderAscending = TRUE, file = NULL, encoding = "UTF-8", CSS = NULL, useViewer = TRUE, no.output = FALSE)

Arguments

df
A data frame with items (from a scale)

factor.groups
If not NULL, the data frame df will be splitted into sub-groups, where the item analysis is carried out for each of these groups. Must be a vector of same length as ncol(df), where each item in this vector represents the group number of the related columns of df. See examples for more details.

scaleItems
If TRUE, the data frame’s vectors will be scaled when calculating the Cronbach’s Alpha value (see sju.reliability). Recommended, when the variables have different measures / scales.

alternateRowColors
If TRUE, alternating rows are highlighted with a light gray background color.

orderColumn
Indicates a column, either by column name or by column index number, that should be ordered. Default order is ascending, which can be changed with orderAscending parameter. Default is NULL, hence the results of the item analysis are printed with no specific order. See examples in sjt.df for further details.

orderAscending
If TRUE (default) and orderColumn is not NULL, data frame is ordered according to the specified column in an ascending order. Use FALSE to apply descending order. See examples in sjt.df for further details.

do
The destination file, which will be in html-format. If no filepath is specified (default), the file will be saved as temporary file and openend either in the RStudio View pane or in the default web browser.
encoding

The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g., German umlauts).

CSS

A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value `page.style` for details of all style-sheet-classnames that are used in this function. Parameters for this list need:

1. the class-names with "css."-prefix as parameter name and
2. each style-definition must end with a semicolon

Examples:

• `css.table='border:2px solid red;'` for a solid 2-pixel table border in red.
• `css.summary='font-weight:bold;'` for a bold fontweight in the summary row.
• `css.arc='color:blue;'` for a blue text color each 2nd row.

See `sjt.df` for further examples.

useViewer

If true, the function tries to show the HTML table in the IDE’s viewer pane. If false or no viewer available, the HTML table is opened in a web browser.

no.output

If true, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

(Invisibly) returns a structure with following elements:

- `df.list`: List of data frames with the item analysis for each sub.group (or complete, if `factor.groups` was NULL)
- `index.scores`: List of standardized scale / index scores of each case (mean value of all scale items for each case) for each sub-group. Note that NA's are removed from this list. Use `df.index.scores` if you want to append the cases' related index scores to the original data frame.
- `df.index.scores`: A data frame with all `index.scores` as column variables. While `index.scores` don't have NA's included, this data frame's row-length equals to the originals data frame's row-length (and thus can be appended)
- `ideal.item.diff`: List of vectors that indicate the ideal item difficulty for each item in each sub-group. Item difficulty only differs when items have different levels.
- `cronbach.values`: List of Cronbach's Alpha values for the overall item scale for each sub-group.
- `knitr.list`: List of html-tables with inline-css for use with knitr for each table (sub-group)
- `knitr`: html-table of all complete output with inline-css for use with knitr
- `complete.page`: Complete html-output.

If `factor.groups` was NULL, each list contains only one element, since just one table is printed for the complete scale indicated by `df`. If `factor.groups` is a vector of group-index-values, the lists contain elements for each sub-group.
Note

- Item difficulty should range between 0.2 and 0.8. Ideal value is \( p + (1-p)/2 \) (which mostly is between 0.5 and 0.8).
- For item discrimination, acceptable values are 0.20 or higher; the closer to 1.00 the better

See Also

\[ \text{sju.cronbach} \]
\[ \text{sju.reliability} \]
\[ \text{sjp.pca} \]
\[ \text{sjt.pca} \]
\[ \text{sjt.df} \]

Examples

```r
# Data from the EUROFAMCARE sample dataset
# --------------------------------------
data(efc)

# retrieve variable and value labels
varlabs <- sji.getVariableLabels(efc)

# receive first item of COPE-index scale
start <- which(colnames(efc) == "c82cop1")
# receive last item of COPE-index scale
end <- which(colnames(efc) == "c90cop9")

# create data frame with COPE-index scale
df <- as.data.frame(efc[, c(start:end)])
colnames(df) <- varlabs[c(start:end)]

## Not run:
sjt.itemanalysis(df)
## End(Not run)

# --------------------------------------
# auto-detection of labels
# --------------------------------------
ecf <- sji.setVariableLabels(efc, varlabs)
## Not run:
sjt.itemanalysis(ecf[, c(start:end)])
## End(Not run)

# --------------------------------------
# Compute PCA on Cope-Index, and perform a
# item analysis for each extracted factor.
# --------------------------------------
## Not run:
factor.groups <- sjt.pca(df, no.output=TRUE)$factor.index
sjt.itemanalysis(df, factor.groups)
```
**sjt.lm**

Show linear regression as HTML table

---

### Description

Shows (multiple) fitted linear models (beta coefficients, std. beta values etc.) as HTML table, or saves them as file. The fitted lm’s should have the same predictor variables and differ only in their response (dependent variable).

### Usage

```r
sjt.lm(..., file = NULL, labelPredictors = NULL,
labelDependentVariables = NULL, stringPredictors = "Predictors",
stringDependentVariables = "Dependent Variables", stringModel = "Model",
showHeaderStrings = FALSE, stringIntercept = "(Intercept)",
stringObservations = "Observations", stringB = "B",
stringSB = "std. Beta", stringCI = "CI", stringSE = "std. Error",
stringP = "p", showConfInt = TRUE, showStdBeta = FALSE,
showStdError = FALSE, digits.est = 2, digits.p = 3, digits.ci = 2,
digits.se = 2, digits.sb = 2, digits.summary = 3,
pvaluesAsNumbers = FALSE, boldpvalues = TRUE,
separateConfColumn = FALSE, newlineConf = TRUE, showAbbrHeadline = TRUE,
showR2 = TRUE, showFstat = FALSE, showAIC = FALSE, cellSpacing = 0.2,
encoding = "UTF-8", CSS = NULL, useViewer = TRUE, no.output = FALSE)
```

### Arguments

- **...** One or more fitted lm-objects.
- **file** The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and openend either in the RStudio View pane or in the default web browser.
- **labelPredictors** Labels of the predictor variables, provided as char vector.
- **labelDependentVariables** Labels of the dependent variables of all fitted models which have been used as first parameter(s), provided as char vector.
- **stringPredictors** String constant used as headline for the predictor column. Default is "Predictors".
- **stringDependentVariables** String constant used as headline for the dependent variable columns. Default is "Dependent Variables".
- **showHeaderStrings** If TRUE, the header strings stringPredictors and stringDependentVariables are shown. By default, they’re hidden.
stringModel String constant used as headline for the model names in case no labels for the dependent variables are provided (see labelDependentVariables). Default is "Model".

stringIntercept String constant used as headline for the Intercept row default is "Intercept".

stringObservations String constant used in the summary row for the count of observation (cases). Default is "Observations".

stringB String used for the column heading of beta coefficients. Default is "B".

stringSB String used for the column heading of standardized beta coefficients. Default is "std. Beta".

stringCI String used for the column heading of confidence interval values. Default is "CI".

stringSE String used for the column heading of standard error values. Default is "std. Error".

stringP String used for the column heading of p values. Default is "p".

showConfInt If TRUE (default), the confidence intervall is also printed to the table. Use FALSE to omit the CI in the table.

showStdBeta If TRUE, the standardized beta-coefficients are also printed. Default is FALSE.

showStdError If TRUE, the standard errors are also printed. Default is FALSE.

digits.eest Amount of decimals for estimators.

digits.p Amount of decimals for p-values.

digits.ci Amount of decimals for confidence intervals.

digits.se Amount of decimals for standard error.

digits.sb Amount of decimals for standardized beta.

digits.summary Amount of decimals for values in model summary.

pvaluesAsNumbers If TRUE, p-values are shown as numbers. If FALSE (default), p-values are indicated by asterisks.

boldpvalues If TRUE (default), significant p-values are shown bold faced.

separateConfColumn if TRUE, the CI values are shown in a separate table column. Default is FALSE.

newlineConf If TRUE and separateConfColumn is FALSE, inserts a line break between B and CI values. If FALSE, CI values are printed in the same line with B values.

showAbbrHeadline If TRUE (default), the table data columns have a headline with abbreviations for beta- and std. beta-values, confidence interval and p-values.

showR2 If TRUE (default), the R2 and adjusted R2 values for each model are printed in the model summary.

showFStat If TRUE, the F-statistics for each model is printed in the model summary. Default is FALSE.

showAIC If TRUE, the AIC value for each model is printed in the model summary. Default is FALSE.
The inner padding of table cells. By default, this value is 0.2 (measure is cm), which is suitable for viewing the table. Decrease this value (0.05 to 0.1) if you want to import the table into Office documents. This is a convenient parameter for the CSS parameter for changing cell spacing, which would be: CSS=list(css.thead="padding:0.2cm;", css.tzdata="padding:0.2cm;").

The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g.) German umlauts).

A list with user-defined style-sheet-definitions, according to the official CSS syntax (see [http://www.w3.org/Style/CSS/](http://www.w3.org/Style/CSS/)). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:

1. the class-names with "css."-prefix as parameter name and
2. each style-definition must end with a semicolon

Examples:

• css.table='border:2px solid red;' for a solid 2-pixel table border in red.
• css.summary='font-weight:bold;' for a bold fontweight in the summary row.
• css.lasttablerow='border-bottom: 1px dotted blue;' for a blue dotted border of the last table row.

See further examples below.

If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with

• the web page style sheet (page.style),
• the web page content (page.content),
• the complete html-output (output.complete) and
• the html-table with inline-css for use with knitr (knitr)

for further use.

Note

The HTML tables can either be saved as file and manually opened (specify parameter file) or they can be saved as temporary files and will be displayed in the RStudio Viewer pane (if working with RStudio) or opened with the default web browser. Displaying resp. opening a temporary file is the default behaviour (i.e. file=NULL).
References

- http://strengejacke.wordpress.com/sjplot-r-package/

See Also

- sjt.glm
- sjp.lm

Examples

```r
# Now fit the models. Note that both models share the same predictors # and only differ in their dependent variable
data(efc)

# fit first model
fit1 <- lm(barthtot ~ c160age + c12hour + c161sex + c172code, data=efc)
# fit second model
fit2 <- lm(neg_c_7 ~ c160age + c12hour + c161sex + c172code, data=efc)

# create and open HTML-table in RStudio Viewer Pane or web browser
## Not run:
sjt.lm(fit1, fit2, labelDependentVariables=c("Barthel-Index", "Negative Impact"), 
    labelPredictors=c("Carer's Age", "Hours of Care", "Carer's Sex", "Educational Status"))
## End(Not run)

# show HTML-table, indicating p-values as numbers
## Not run:
sjt.lm(fit1, fit2, labelDependentVariables=c("Barthel-Index", "Negative Impact"), 
    labelPredictors=c("Carer's Age", "Hours of Care", "Carer's Sex", "Educational Status"), 
    showStdBeta=TRUE, pvaluesAsNumbers=TRUE)
## End(Not run)

# create and open HTML-table in RStudio Viewer Pane or web browser, 
# printing CI in a separate column
## Not run:
sjt.lm(fit1, fit2, labelDependentVariables=c("Barthel-Index", "Negative Impact"), 
    labelPredictors=c("Carer's Age", "Hours of Care", "Carer's Sex", "Educational Status"), 
    separateConfColumn=TRUE)
## End(Not run)

# show HTML-table, indicating p-values as numbers 
# and printing CI in a separate column 
## Not run:
sjt.lm(fit1, fit2, labelDependentVariables=c("Barthel-Index", "Negative Impact"), 
    labelPredictors=c("Carer's Age", "Hours of Care", "Carer's Sex", "Educational Status"), 
    showStdBeta=TRUE, pvaluesAsNumbers=TRUE, separateConfColumn=TRUE)
## End(Not run)

# ---------------------------------------------
# connecting two html-tables
```
Show principal component analysis as HTML table

Description

Performs a principle component analysis on a data frame or matrix and displays the factor solution as HTML table, or saves them as file.

In case a data frame is used as parameter, the Cronbach’s Alpha value for each factor scale will be calculated, i.e. all variables with the highest loading for a factor are taken for the reliability test. The result is an alpha value for each factor dimension.

Usage

sjt.pca(data, numberOfFactors = NULL, factorLoadingTolerance = 0.1,
         file = NULL, varlabels = NULL,
         title = "Principal Component Analysis (with varimax rotation)"
Arguments

data A data frame with factors (each columns one variable) that should be used to compute a PCA, or a `prcomp` object.

numberOffactors A predefined number of factors to use for the calculating the varimax rotation. By default, this value is `NULL` and the amount of factors is calculated according to the Kaiser-criteria. See parameter `plotEigenvalues`.

factorLoadingTolerance Specifies the minimum difference a variable needs to have between factor loadings (components) in order to indicate a clear loading on just one factor and not diffusing over all factors. For instance, a variable with 0.8, 0.82 and 0.84 factor loading on 3 possible factors can not be clearly assigned to just one factor and thus would be removed from the principal component analysis. By default, the minimum difference of loading values between the highest and 2nd highest factor should be 0.1

file The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and openend either in the RStudio View pane or in the default web browser.

varlabels The item labels that are printed in the first column. If no item labels are provided (default), the data frame’s column names are used. Item labels must be a string vector, e.g.: `varlabels=c("Var 1", "Var 2", "Var 3")`.

title A table caption. By default, "Principal Component Analysis (with varimax rotation)" is used as the table’s title.

breakLabelsAt Wordwrap for diagram labels. Determines how many chars of the variable labels are displayed in one line and when a line break is inserted. Default is 20.

digits The amount of digits used the values inside table cells. Default is 2.

showCronbachsAlpha If TRUE (default), the cronbach’s alpha value for each factor scale will be calculated, i.e. all variables with the highest loading for a factor are taken for the reliability test. The result is an alpha value for each factor dimension. Only applies when data is a data frame and no `prcomp` object.

showMSA If TRUE, shows an additional column with the measure of sampling adequacy according dor each component.

showVariance If TRUE, the proportions of variances for each component as well as cumulative variance are shown in the table footer.

alternateRowColors If TRUE, alternating rows are highlighted with a light gray background color.

stringPov The string for the table row that contains the proportions of variances. By default, "Proportion of Variance" will be used.
stringCpov  The string for the table row that contains the cumulative variances. By default, "Cumulative Proportion" will be used.

encoding  The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g.) German umlauts).

CSS  A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:

1. the class-names with "css."-prefix as parameter name and
2. each style-definition must end with a semicolon

Examples:

* css.table='border:2px solid red;' for a solid 2-pixel table border in red.
* css.summary='font-weight:bold;' for a bold fontweight in the summary row.
* css.lasttablerrow='border-bottom:1px dotted blue;' for a blue dotted border of the last table row.

See further examples below.

useViewer  If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

no.output  If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with

* the web page style sheet (page.style),
* the web page content (page.content),
* the complete html-output (output.complete),
* the html-table with inline-css for use with knitr (knitr),
* the factor.index, i.e. the column index of each variable with the highest factor loading for each factor and
* the removed.items, i.e. which variables have been removed because they were outside of the factorLoadingTolerance’s range.

for further use.
Note

The HTML tables can either be saved as file and manually opened (specify parameter file) or they can be saved as temporary files and will be displayed in the RStudio Viewer pane (if working with RStudio) or opened with the default web browser. Displaying resp. opening a temporary file is the default behaviour (i.e. file=NULL).

This PCA uses the `prcomp` function and the `varimax` rotation.

References


See Also

- `sjp.pca`
- `sju.reliability`
- `sji.itemanalysis`
- `sju.cronbach`

Examples

```r
# randomly create data frame with 7 items, each consisting of 4 categories
likert_4 <- data.frame(sample(1:4, 500, replace=TRUE, prob=c(0.2,0.3,0.1,0.4)),
                       sample(1:4, 500, replace=TRUE, prob=c(0.5,0.25,0.15,0.01)),
                       sample(1:4, 500, replace=TRUE, prob=c(0.4,0.15,0.25,0.02)),
                       sample(1:4, 500, replace=TRUE, prob=c(0.25,0.1,0.4,0.25)),
                       sample(1:4, 500, replace=TRUE, prob=c(0.1,0.4,0.4,0.1)),
                       sample(1:4, 500, replace=TRUE),
                       sample(1:4, 500, replace=TRUE, prob=c(0.35,0.25,0.15,0.25)))

# Create variable labels
colnames(likert_4) <- c("V1", "V2", "V3", "V4", "V5", "V6", "V7")

# show table
## Not run:
sjt.pca(likert_4)
## End(Not run)

# Data from the EUROFAMCARE sample dataset
# -----------------------------------------------
data(efc)

# retrieve variable and value labels
varlabs <- sji.getVariableLabels(efc)

# retrieve first item of COPE-index scale
start <- which(colnames(efc)=="c82cop1")
# retrieve last item of COPE-index scale
end <- which(colnames(efc)=="c90cop9")
```
# sjt.stackfrq

Create data frame with COPE-index scale

```r
# create data frame with COPE-index scale
df <- as.data.frame(efc[,c(start:end)])
colnames(df) <- varlabs[c(start:end)]

## Not run:
sjt.pca(df)
## End(Not run)
```

## Description

Show stacked frequencies as HTML table. This function is useful when several items with identical scale/categories should be printed as table to compare their distributions (e.g. when plotting scales like SF, Barthel-Index, Quality-of-Life-scales etc.).

## Usage

```r
sjt.stackfrq(items, weightBy = NULL, title = NULL, varlabels = NULL, breakLabelsAt = 40, valuelabels = NULL, breakValueLabelsAt = 20, orderBy = NULL, reverseOrder = FALSE, alternateRowColors = FALSE, showN = FALSE, showTotalN = FALSE, showNA = FALSE, labelNA = "NA", showSkew = FALSE, showKurtosis = FALSE, skewString = "Skew", kurtosisString = "Kurtosis", file = NULL, encoding = "UTF-8", CSS = NULL, useViewer = TRUE, noOutput = FALSE)
```

## Arguments

- **items**: A `data.frame` with each column representing one (likert- or scale-)item.
- **weightBy**: A weight factor that will be applied to weight all cases from `items`.
- **title**: A table caption.
- **varlabels**: A list or vector of strings with variable names. If not specified, row names of `items` will be used, resp. variable labels will automatically be detected, when they have a "variable.label" attribute (see `sji.setVariableLabels`) for details.
- **breakLabelsAt**: Wordwrap for variable labels. Determines how many chars of the variable labels are displayed in one line and when a line break is inserted. Default is 40.
valueLabels  A list or vector of strings that category/value labels, which appear in the header row.

breakValueLabelsAt  Wordwrap for value labels. Determines how many chars of the value labels are displayed in one line and when a line break is inserted. Default is 20.

orderBy  Indicates whether the items should be ordered by highest count of first or last category of items. Use "first" to order ascending by lowest count of first category, "last" to order ascending by lowest count of last category or NULL (default) for no sorting. You can specify just the initial letter. In case you want to revers order (descending from highest count), use reverseOrder parameter.

reverseOrder  If TRUE, the item order is reversed.

showN  If TRUE, each item’s category N is printed in the table cells.

showTotalN  If TRUE, an additional column with each item’s total N is printed.

showNA  If TRUE, NA’s (missing values) are also printed in the table.

labelNA  The label for the missing column/row.

showSkew  If TRUE, an additional column with each item’s skewness is printed. The skewness is retrieved from the describe function of the psych package.

showKurtosis  If TRUE, an additional column with each item’s kurtosis is printed. The kurtosis is retrieved from the describe function of the psych package.

skewString  A character string, which is used as header for the skew column (see showSkew)). Default is "Skew".

kurtosisString  A character string, which is used as header for the kurtosis column (see showKurtosis)). Default is "Kurtosis".

alternateRowColors  If TRUE, alternating rows are highlighted with a light gray background color.

file  The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and openend either in the RStudio View pane or in the default web browser.

encoding  The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g.) German umlauts).

CSS  A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:

1. the class-names with "css."-prefix as parameter name and
2. each style-definition must end with a semicolon

Examples:

- css.table='border:2px solid red;' for a solid 2-pixel table border in red.
- css.summary='font-weight:bold;' for a bold fontweight in the summary row.
- css.caption='border-bottom: 1px dotted blue;' for a blue dotted border of the last table row.
useViewer | If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

no.output | If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value

Invisibly returns a structure with

- the web page style sheet (page.style),
- the web page content (page.content),
- the complete html-output (output.complete) and
- the html-table with inline-css for use with knitr (knitr)

for further use.

Note

The HTML tables can either be saved as file and manually opened (specify parameter file) or they can be saved as temporary files and will be displayed in the RStudio Viewer pane (if working with RStudio) or opened with the default web browser. Displaying resp. opening a temporary file is the default behaviour (i.e. file=NULL).

References

- http://strengejacke.wordpress.com/sjplot-r-package/

See Also

sjp.stackfreq
sjp.likert

Examples

```r
# random sample
# -----------------------------
likert_4 <- data.frame(as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.2, 0.3, 0.1, 0.4))),
as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.5, 0.25, 0.15, 0.1))),
as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.25, 0.1, 0.4, 0.25))),
as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.1, 0.4, 0.4, 0.1))),
as.factor(sample(1:4, 500, replace=TRUE, prob=c(0.35, 0.25, 0.15, 0.25))))

# create labels
levels_4 <- c("Independent", "Slightly dependent", "Dependent", "Severely dependent")

# create item labels
```
items <- c("Q1", "Q2", "Q3", "Q4", "Q5")

# plot stacked frequencies of 5 (ordered) item-scales
## Not run:
sjt.stackfrq(likert_4, valuelabels=levels_4, varlabels=items)
## End(Not run)

# -----------------------------------------------
# Data from the EUROFAMCARE sample dataset
# -----------------------------------------------
data(efc)
# receive first item of COPE-index scale
start <- which(colnames(efc)="c82cop1")
# receive first item of COPE-index scale
end <- which(colnames(efc)="c90cop9")
# retrieve variable labels
varlabs <- sji.getVariableLabels(efc)

# Note: Parameter "valuelabels" is only needed for datasets
# that have been imported from SPSS.
## Not run:
sjt.stackfrq(efc[,c(start:end)],
varlabels=varlabs[,c(start:end)],
alternateRowColors=TRUE)
## End(Not run)

## Not run:
sjt.stackfrq(efc[,c(start:end)],
varlabels=varlabs[,c(start:end)],
alternateRowColors=TRUE,
showN=TRUE,
showNA=TRUE)
## End(Not run)

# -----------------------------------------------
# auto-detection of labels
# -----------------------------------------------
efc <- sji.setVariableLabels(efc, varlabs)
## Not run:
sjt.stackfrq(efc[,c(start:end)])
## End(Not run)

# -----------------------------------------------
# User defined style sheet
# -----------------------------------------------
## Not run:
sjt.stackfrq(efc[,c(start:end)],
varlabels=varlabs[,c(start:end)],
alternateRowColors=TRUE,
showTotalN=TRUE,
showSkew=TRUE,
showKurtosis=TRUE,
Show contingency tables as HTML table

Description

Shows contingency tables as HTML file in browser or viewer pane, or saves them as file.

Usage

```
sjt.xtab(var.row, var.col, var.grp = NULL, weightBy = NULL, digits = 1, file = NULL, variableLabels = NULL, valueLabels = NULL, breakVariableLabelsAt = 40, breakValueLabelsAt = 20, stringTotal = "Total", showCellPerc = FALSE, showRowPerc = FALSE, showColPerc = FALSE, showExpected = FALSE, showHorizontalLine = FALSE, showSummary = TRUE, showLegend = TRUE, showNA = FALSE, labelNA = "NA", tdcol.n = "black", tdcol.expected = "#339999", tdcol.cell = "#993333", tdcol.row = "#333399", tdcol.col = "#339933", highlightTotal = FALSE, highlightColor = ":f8f8f8", percSign = "&nbsp;%", hundred = "100.0", encoding = "UTF-8", CSS = NULL, useViewer = TRUE, no.output = FALSE)
```

Arguments

- **var.row**: Variable that should be displayed in the table rows.
- **var.col**: Variable that should be displayed in the table columns.
- **var.grp**: An optional grouping variable that splits the data into several groups, depending on the amount of categories. See examples for details.
- **weightBy**: A weight factor that will be applied to weight all cases. Default is NULL, so no weights are used.
- **digits**: The amount of digits used for the percentage values inside table cells. Default is 1.
- **file**: The destination file, which will be in html-format. If no filepath is specified, the file will be saved as temporary file and openend either in the RStudio View pane or in the default web browser.
- **variableLabels**: A character vector of same length as supplied variables, with the associated variable names. Following order is needed: name of **var.row**, name of **var.col**, and - if **var.grp** is not NULL - name of **var.grp**. See examples for more details. **variableLabels** are detected automatically, if **var.row** or **var.col** have a "variable.label" attribute (see **sji.setVariableLabels**) for details.
- **valueLabels**: A list of character vectors that indicate the value labels of the supplied variables. Following order is needed: value labels of **var.row**, value labels of **var.col**, and - if **var.grp** is not NULL - value labels of **var.grp**. **valueLabels** needs to be a **list** object. See examples for more details.
breakVariableLabelsAt
Wordwrap for variable labels. Determines how many chars of the variable labels are displayed in one line and when a line break is inserted. Default is 40.

breakValueLabelsAt
Wordwrap for value labels. Determines how many chars of the value labels are displayed in one line and when a line break is inserted. Default is 20.

stringTotal
String label for the total column / row header.

showCellPerc
If TRUE, cell percentage values are shown.

showRowPerc
If TRUE, row percentage values are shown.

showColPerc
If TRUE, column percentage values are shown.

showExpected
If TRUE, expected values are also shown.

showHorizontalLine
If TRUE, data rows are separated with a horizontal line.

showSummary
If TRUE (default), a summary row with Chi-square statistics (see chisq.test), Cramer’s V or Phi-value etc. is shown. If a cell contains expected values lower than five (or lower than 10 if df is 1), the Fisher’s exact test (see fisher.test) is computed instead of Chi-square test. If the table’s matrix is larger than 2x2, Fisher’s exact test with Monte Carlo simulation is computed.

showLegend
If TRUE (default), the color legend for coloring observed and expected values as well as cell, row and column percentages is shown. See tdcol.n, tdcol.expected, tdcol.cell, tdcol.row and tdcol.col.

showNA
If TRUE, NA’s (missing values) are also printed in the table.

labelNA
The label for the missing column/row.

tdcol.n
Color for highlighting count (observed) values in table cells. Default is black.

tdcol.expected
Color for highlighting expected values in table cells. Default is cyan.

tdcol.cell
Color for highlighting cell percentage values in table cells. Default is red.

tdcol.row
Color for highlighting row percentage values in table cells. Default is blue.

tdcol.col
Color for highlighting column percentage values in table cells. Default is green.

highlightTotal
If TRUE, the total column and row will be highlighted with a different background color. See highlightColor.

highlightColor
If highlightTotal is TRUE, this color value will be used for painting the background of the total column and row. Default is a light grey.

percSign
The percentage sign that is printed in the table cells, in HTML-format. Default is "&nbsp;%", hence the percentage sign has a non-breaking-space after the percentage value.

hundret
Default value that indicates the 100-percent column-sums (since rounding values may lead to non-exact results). Default is "100.0".

encoding
The charset encoding used for variable and value labels. Default is "UTF-8". Change encoding if specific chars are not properly displayed (e.g. German umlauts).

CSS
A list with user-defined style-sheet-definitions, according to the official CSS syntax (see http://www.w3.org/Style/CSS/). See return value page.style for details of all style-sheet-classnames that are used in this function. Parameters for this list need:
1. the class-names with "css."-prefix as parameter name and  
2. each style-definition must end with a semicolon  

Examples:  
- `css.table='border:2px solid red;'` for a solid 2-pixel table border in red.  
- `css.summary='font-weight:bold;'` for a bold fontweight in the summary row.  
- `css.lasttablerow='border-bottom: 1px dotted blue;'` for a blue dotted border of the last table row.

See further examples below.

useViewer  
If TRUE, the function tries to show the HTML table in the IDE’s viewer pane. If FALSE or no viewer available, the HTML table is opened in a web browser.

no.output  
If TRUE, the html-output is neither opened in a browser nor shown in the viewer pane and not even saved to file. This option is useful when the html output should be used in knitr documents. The html output can be accessed via the return value.

Value  
Invisibly returns a structure with  
- the web page style sheet (page.style),  
- the web page content (page.content),  
- the complete html-output (output.complete) and  
- the html-table with inline-css for use with knitr (knitr)

for further use.

Note  
The HTML tables can either be saved as file and manually opened (specify parameter file) or they can be saved as temporary files and will be displayed in the RStudio Viewer pane (if working with RStudio) or opened with the default web browser. Displaying resp. opening a temporary file is the default behaviour (i.e. file=NULL).

Since package version 1.3, the parameter valueLabels, which represent the value labels, is retrieved automatically if a) the variables var.col and var.row come from a data frame that was imported with the sjipl.spss function (because then value labels are attached as attributes to the data) or b) when the variables are factors with named factor levels (e.g., see column group in dataset PlantGrowth). However, you still can use own parameters variable labels.

References  
See Also

sjp.xtab
sju.table.values

Examples

# prepare sample data set
data(efc)
ecf.labels <- sji.getValueLabels(ecf)

# print simple cross table w/o labels
## Not run:
sjt.xtab(ecf$e16sex, ecf$e42dep)
## End(Not run)

# print cross table with labels and expected values
## Not run:
sjt.xtab(ecf$e16sex, ecf$e42dep,
variableLabels=c("Elder's gender", "Elder's dependency"),
valueLabels=list(ecf.labels[['e16sex']], ecf.labels[['e42dep']])),
showExpected=TRUE)
## End(Not run)

# print minimal cross table with labels, total col/row highlighted
## Not run:
sjt.xtab(ecf$e16sex, ecf$e42dep,
variableLabels=c("Elder's gender", "Elder's dependency"),
valueLabels=list(ecf.labels[['e16sex']], ecf.labels[['e42dep']])),
showHorizontalLine=FALSE,
showCellPerc=FALSE,
highlightTotal=TRUE)
## End(Not run)

# auto-detection of labels
# ----------------------
ecf <- sji.setVariableLabels(ecf, sji.getVariableLabels(ecf))
# print cross table with labels and all percentages
## Not run:
sjt.xtab(ecf$e16sex, ecf$e42dep,
showRowPerc=TRUE, showColPerc=TRUE)
## End(Not run)

# print cross table with labels and all percentages, including
# grouping variable
## Not run:
sjt.xtab(ecf$e16sex, ecf$e42dep, ecf$c161sex,
variableLabels=c("Elder's gender",
"Elder's dependency",
"Carer's gender"),
valueLabels=list(ecf.labels[['e16sex']],
ecf.labels[['e42dep']]),
**Description**

This method adjusts the y-range of a ggplot-object, which is useful when value labels are outside of the plot region. A modified ggplot-object will be returned with adjusted y-range so everything should be visible. Note that this function only works on `scale_y_continuous`.

**Usage**

```r
describe_model_summary(your_model)
```

**Arguments**

- **gp**: A ggplot-object. Usually, this will be returned by most of this package’s plotting functions.
- **upperMargin**: Defines the margin of the upper y bound of the plot. This value will be multiplied with the total y range. Default is 1.05, which means that the upper margin of the plot is about 5 percent of the "visible" plot area (i.e. the y-range is 105 percent of the actual needed range to make all object visible).

**Value**

The same ggplot-object, with adjusted y-range, so all graphics and labels should be visible.

**Note**

Note that this function only works on `scale_y_continuous`.

**References**

http://www.r-bloggers.com/setting-axis-limits-on-ggplot-charts/
Examples

```r
# sample data set
data(efc)
# show frequencies of relationship-variable and
# retrieve plot object
gp <- sjp.frq(efc$e15relat, printPlot=FALSE)
# show current plot
plot(gp$plot)
# show adjusted plot
sju.adjustPlotRange.y(gp$plot)
```

---

**sju.aov1.levene**

*Plot Levene-Test for One-Way-Anova*

Description

Plot results of Levene’s Test for Equality of Variances for One-Way-Anova.

Usage

```r
sju.aov1.levene(depVar, grpVar)
```

Arguments

- **depVar**: The dependent variable. Will be used with following formular: `aov(depVar ~ grpVar)`
- **grpVar**: The grouping variable, as unordered factor. Will be used with following formular: `aov(depVar ~ grpVar)`

See Also

`sjp.aov1`, `sju.chi2.gof`, `sju.mwu` and `wilcox.test`, `ks.test`, `kruskal.test`, `t.test`, `chisq.test`, `fisher.test`

Examples

```r
data(efc)
sju.aov1.levene(efc$c12hour, efc$s42dep)
```
sju.betaCoef

Retrieve std. beta coefficients of lm

Description

Returns the standardized beta coefficients of a fitted linear model.

Usage

sju.betaCoef(fit)

Arguments

fit A fitted linear model.

Value

The standardized beta coefficients of the fitted linear model.

Note

"Standardized coefficients refer to how many standard deviations a dependent variable will change, per standard deviation increase in the predictor variable. Standardization of the coefficient is usually done to answer the question of which of the independent variables have a greater effect on the dependent variable in a multiple regression analysis, when the variables are measured in different units of measurement (for example, income measured in dollars and family size measured in number of individuals)." (Source: Wikipedia)

References

http://en.wikipedia.org/wiki/Standardized_coefficient

See Also

sjp.lm
sjt.lm

Examples

# fit linear model
fit <- lm(airquality$Ozone ~ airquality$Wind + airquality$Temp + airquality$Solar.R)
# print std. beta coefficients
sju.betaCoef(fit)
sju.chi2.gof

Performs a Chi-square goodness-of-fit-test

Description

Performs a Chi-square goodness-of-fit-test

Usage

sju.chi2.gof(var, prob, weights = NULL)

Arguments

- var: a numeric vector / variable.
- prob: a vector of probabilities (indicating the population probabilities) of the same length as var’s amount of categories / factor levels. Use nrow(table(var)) to determine the amount of necessary values for prob.
- weights: a vector with weights, used to weight var.

Note

This function is a convenient function for chisq.test, performing goodness-of-fit test.

See Also

sju.mwu, sju.aov1.levene and wilcox.test, ks.test, kruskal.test, t.test, chisq.test, fisher.test

Examples

data(efc)
# differing from population
sju.chi2.gof(efc$e42dep, c(0.3,0.2,0.22,0.28))
# equal to population
sju.chi2.gof(efc$e42dep, prop.table(table(efc$e42dep)))

sju.cramer

Cramer’s V for a contingency table

Description

Compute Cramer’s V for a table with more than 2x2 fields.

Usage

sju.cramer(tab)
Arguments

- `tab`: A simple `table` or `ftable`. Tables of class `xtabs` and other will be coerced to `ftable` objects.

Value

The table's Cramer's V.

See Also

- `sju.table.values`
- `sju.phi`

Examples

```r
tab <- table(sample(1:2, 30, TRUE), sample(1:3, 30, TRUE))
sju.cramer(tab)
```

---

**sju.cronbach**  
*Calculates Cronbach’s Alpha for a matrix*

Description

This function calculates the Cronbach’s alpha value for each column of a data frame or matrix.

Usage

```r
sju.cronbach(df)
```

Arguments

- `df`: A data frame or matrix with more than 2 columns.

Value

The Cronbach’s alpha value for `df`.

Note

For use case, see `sjp.pca` and `sjt.pca`.

See Also

- `sju.reliability`
- `sjt.itemanalysis`
- `sjp.pca`
- `sjt.pca`
### sju.dicho

**Dichotomize variables**

**Description**

Dichotomizes variables into dummy variables (0/1). Dichotomization is either done by median or mean (see `dichBy`).

**Usage**

```r
sju.dicho(var, dichBy = "median", dichVal = -1)
```

**Arguments**

- **var**: The variable that should be dichotomized.
- **dichBy**: Indicates the split criterion where the variable is dichotomized. By default, `var` is split into two groups at the median (`dichBy="median"` or `dichBy="md"`). Further values for `dichBy` are "mean" (or "m"), which splits into groups at the mean of `var`; and "value" (or "v"). In the latter case, you have to specify `dichVal`.
- **dichVal**: Indicates a value where `var` is dichotomized when `dichBy="value"`. Note that `dichVal` is inclusive, i.e. `dichVal=10` will split `var` into one group with values from lowest to 10 and another group with values greater than 10.

**Value**

A dichotomized variable (0/1-coded).

**Examples**

```r
data(efc)
summary(efc$c12hour)
table(sju.dicho(efc$c12hour))
table(sju.dicho(efc$c12hour, "mean"))
table(sju.dicho(efc$c12hour, "value", 30))
```

### sju.groupVar

**Recode count variables into grouped factors**

**Description**

Recode count variables into grouped factors.
Usage

sju.groupVar(var, groupsize = 5, asNumeric = TRUE, rightInterval = FALSE, autoGroupCount = 30)

Arguments

var
The count variable, which should recoded into groups.

groupsize
The group-size, i.e. the range for grouping. By default, for each 5 categories a new group is defined, i.e. groupsize=5. Use groupsize="auto" to automatically resize a variable into a maximum of 30 groups (which is the ggplot-default grouping when plotting histograms). Use autoGroupCount to determin the amount of groups.

asNumeric
If TRUE (default), the recoded variable will be returned as numeric vector. If FALSE, a factor is returned.

rightInterval
If TRUE, grouping starts with the lower bound of groupsize. In this case, groups cover the ranges from 50-54, 55-59, 60-64 etc. If FALSE (default), grouping starts with the upper bound of groupsize. In this case, groups cover the ranges from 51-55, 56-60, 61-65 etc.

autoGroupCount
Sets the maximum number of groups that are defined when auto-grouping is on (groupsize="auto"). Default is 30. If groupsize is not set to "auto", this parameter will be ignored.

Value

A grouped variable, either as numeric or as factor (see parameter asNumeric).

See Also

sju.groupVarLabels

Examples

age <- abs(round(rnorm(100, 65, 20)))
age.grp <- sju.groupVar(age, 10)
hist(age)
hist(age.grp)

# histogram with EUROFAMCARE sample dataset
# variable not grouped
data(efc)
efc.val <- sji.getValueLabels(efc)
efc.var <- sji.getVariableLabels(efc)
sjp(efc$e17age,
   title=efc.var[['e17age']],
   type="h",
   showValueLabels=FALSE)

# bar plot with EUROFAMCARE sample dataset
# grouped variable
sju.groupVarLabels

Create labels for recoded groups

Description

Creates the related labels for the grouped variable created by the sju.groupVar function.

Usage

sju.groupVarLabels(var, groupsize = 5, rightInterval = FALSE, 
autoGroupCount = 30)

Arguments

- **var**  
The scale variable, which should recoded into groups.

- **groupsize**  
The group-size, i.e. the range for grouping. By default, for each 5 categories 
  new group is built, i.e. groupsize=5. Use groupsize="auto" to automatically 
  resize a variable into a maximum of 30 groups (which is the ggplot-default 
  grouping when plotting histograms). Use parameter autoGroupCount to define 
  the amount of groups.

- **rightInterval**  
If TRUE, grouping starts with the lower bound of groupsize. In this case, groups 
cover the ranges from 50-54, 55-59, 60-64 etc. 
If FALSE (default), grouping starts with the upper bound of groupsize. In this 
case, groups cover the ranges from 51-55, 56-60, 61-65 etc.

- **autoGroupCount**  
Sets the maximum number of groups that are built when auto-grouping is on 
(groupsize="auto"). Default is 30. If groupsize is not set to "auto", this 
parameter will be ignored.

Value

A string vector containing labels based on the grouped counts of var, formatted as "from lower 
bound to upper bound", e.g. "10-19" "20-29" "30-39" etc. See example below.

Note

Usually you should use the same values for groupsize and rightInterval as used in the sju.groupVar 
function if you want to create labels for the related recoded variable.
Performs a Mann-Whitney-U-Test

This function performs a Mann-Whitney-U-Test (or Wilcoxon rank sum test, see `wilcox.test`) for the variable `var`, which is divided into groups indicated by `grp` (so the formula `var ~ grp` is used). If `grp` has more than two categories, a comparison between each two groups is performed.

Usage

```
sju.mwu(var, grp, alternative = "two.sided")
```
Arguments

var  A numeric vector / variable, where the Mann-Whitney-U-Test should be applied to.

grp  The grouping variable indicating the groups that should be used for comparison.

alternative  a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.

Value

(Invisibly) returns a data frame with p-values for each group-comparison.

Note

This function calls the wilcox.test with formula. If grp has more than two groups, additionally a Kruskal-Wallis-Test (see kruskal.test) is performed.

See Also

sju.chi2.gof, sju.aov1.levene and wilcox.test, ks.test, kruskal.test, t.test, chisq.test, fisher.test

Examples

data(efc)
# Mann-Whitney-U-Tests for elder's age by elder's dependency.
sju.mwu(efc$s17age, efc$s42dep)

sju.phi  Phi value for a contingency table

Description

Compute Phi value for a contingency table.

Usage

sju.phi(tab)

Arguments

tab  A simple table or ftable. Tables of class xtabs and other will be coerced to ftable objects.

Value

The table's Phi value.
See Also

  sju.table.values
  sju.cramer

Examples

  tab <- table(sample(1:2, 30, TRUE), sample(1:2, 30, TRUE))
  sju.phi(tab)

---

sju.recode  
Recode variable values.

Description

  Recodes the categories of a variables. Wrapper function that calls the recode function from the car package.

Usage

  sju.recode(...)

Arguments

  ...  
  parameters, see recode function from the car package.

Value

  A variable with recoded values.

Examples

  data(efc)
  table(efc$e42dep)
  table(sju.recode(efc$e42dep, "1:2=1;3:4=2"))
Recode variable categories into new values.

Description

Recodes the categories of a variable var into new category values, beginning with the lowest value specified by parameter lowest. Useful if you want to recode dummy variables with 1/2 coding to 0/1 coding, or recoding scales from 1-4 to 0-3 etc.

Usage

sju.recodeTo(var, lowest = 0, highest = -1)

Arguments

- var: The variable (vector) that should be recoded.
- lowest: Indicating the lowest category value after recoding. Default is 0, so the new variable starts with the category value 0.
- highest: If specified and larger than lowest, all category values larger than highest will be set to NA. Default is M, i.e. this parameter is ignored and no NA's will be produced.

Value

A new variable with recoded category values, where lowest indicates the lowest value.

Examples

```r
# recode 1-4 to 0-3
dummy <- sample(1:4, 10, replace=TRUE)
sju.recodeTo(dummy)

# recode 3-6 to 0-3
# note that numeric type is returned
dummy <- as.factor(3:6)
sju.recodeTo(dummy)

# lowest value starting with 1
dummy <- sample(11:15, 10, replace=TRUE)
sju.recodeTo(dummy, 1)

# lowest value starting with 1, highest with 3
# all others set to NA
dummy <- sample(11:15, 10, replace=TRUE)
sju.recodeTo(dummy, 1, 3)
```
sju.reliability

Performs a reliability test on an item scale.

Description

This function calculates the item discriminations (corrected item-total correlations for each item of df with the remaining items) and the Cronbach’s alpha for each item, if it was deleted from the scale.

Usage

sju.reliability(df, scaleItems = FALSE, digits = 3)

Arguments

df
   A data frame with items (from a scale)

scaleItems
   If TRUE, the data frame’s vectors will be scaled. Recommended, when the variables have different measures/scales.

digits
   Amount of digits for Cronbach’s Alpha and correlation values in returned data frame.

Value

A data frame with the corrected item-total correlations (item discrimination) and Cronbach’s alpha (if item deleted) for each item of the scale, or NULL if data frame had too less columns.

Note

This function is similar to a basic reliability test in SPSS. The correlations in the Item-Total-Statistic are a computed correlation of each item against the sum of the remaining items (which are thus treated as one item).

See Also

sju.cronbach
sji.itemanalysis
sjp.pca
sjt.pca
sjt.df

Examples

# ---------------------------------------
# Data from the EUROFAMCARE sample dataset
# ---------------------------------------
data(efc)

# retrieve variable and value labels
sju.setNA

Set NA for specific variable values

Description

This function sets specific values of a variable var as missings (NA).

Usage

sju.setNA(var, values)
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>The variable where new missing values should be defined.</td>
</tr>
<tr>
<td>values</td>
<td>The values that should be replaced with NA’s.</td>
</tr>
</tbody>
</table>

Value

The var with each values of values replaced by an NA.

Examples

```r
# create random variable
dummy <- sample(1:8, 100, replace=TRUE)
# show value distribution
table(dummy)
# set value 1 and 8 as missings
dummy <- sju.setNA(dummy, c(1,8))
# show value distribution, including missings
table(dummy, exclude=NULL)
```

sju.table.values  
Compute table's values

Description

This function calculates a table's cell, row and column percentages as well as expected values and returns all results as lists of tables.

Usage

`sju.table.values(tab, digits = 2)`

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tab</td>
<td>A simple table or ftable of which cell, row and column percentages as well as expected values are calculated. Tables of class xtabs and other will be coerced to ftable objects.</td>
</tr>
<tr>
<td>digits</td>
<td>The amount of digits for the table percentage values.</td>
</tr>
</tbody>
</table>

Value

(invisibly) returns a list with four tables:

1. cell a table with cell percentages of tab
2. row a table with row percentages of tab
3. col a table with column percentages of tab
4. expected a table with expected values of tab
Weight a variable

This function weights the variable `var` by a specific vector of `weights`.

Usage

```r
sju.weight(var, weights)
```

Arguments

- `var` The (unweighted) variable
- `weights` A vector with same length as `var`, which contains weight factors. Each value of `var` has a specific assigned weight in `weights`.

Value

The weighted `var`.

Note

The values of the returned vector are in sorted order, whereas the categories of the original `var` may be spread randomly. Hence, `var` can’t be used, for instance, for further cross tabulation. In case you want to have weighted contingency tables or (grouped) box plots etc., use the `weightBy` parameter of most functions (like in `sjt.xtab` or `sjp.grpfrq`).

See Also

- `sju.weight2`
**sju.weight2**

**Weight a variable**

This function weights the variable `var` by a specific vector of weights. It’s an alternative weight calculation to `sju.weight`, where `sju.weight` usage is recommended. This function sums up all weights values of the associated categories of `var`, whereas the `sju.weight` function uses a `xtabs` formula to weight cases. Thus, this function may return a value with a different length than that from `var`.

**Usage**

```r
sju.weight2(var, weights)
```

**Arguments**

- `var` The (unweighted) variable
- `weights` A vector with same length as `var`, which contains weight factors. Each value of `var` has a specific assigned weight in `weights`.

**Value**

The weighted `var`.

**Note**

The values of the returned vector are in sorted order, whereas the categories of the original `var` may be spread randomly. Hence, `var` can’t be used, for instance, for further cross tabulation. In case you want to have weighted contingency tables or (grouped) box plots etc., use the `weightBy` parameter of most functions (like in `sjt.xtab` or `sjp.grpfrq`).

**See Also**

- `sju.weight`

**Examples**

```r
v <- sample(1:4, 20, TRUE)
table(v)
w <- abs(rnorm(20))
table(sju.weight(v,w))
sju.weight2(v, weights)
```

```r
table(sju.weight2(v, w))
```
Description

Insert line breaks in long character strings. Useful if you want to wordwrap plot labels.

Usage

\texttt{sju\_wordwrap(labels, wrap, linesep = NULL)}

Arguments

- \texttt{labels}: The label(s) (i.e. character string). You can also pass several strings as vector (e.g. labels=c("first long string", "second long string"))
- \texttt{wrap}: The amount of chars per line (i.e. line length)
- \texttt{linesep}: By default, this parameter is NULL and a regular new line string is used. For HTML-needs, for instance, linesep could be "<br>".

Value

New label(s) with line breaks inserted at every \texttt{wrap}'s position.

Examples

\texttt{sju\_wordwrap(c("A very long string", "And another even longer string"), 10)}
Index

*Topic data
  efc, 4
AIC, 111
chisq.test, 55, 97, 130, 134, 136, 142
clusGap, 10, 11
coeff, 110
cor, 31, 32, 99
data.frame, 21, 125
describe, 102, 106, 126
dist, 6, 7, 13
efc, 4
exp, 110
family, 109, 111
fisher.test, 55, 97, 130, 134, 136, 142
ftable, 137, 142, 147
gem_histogram, 39
hclust, 6, 7, 13
kmeans, 5, 6, 12, 13, 16
kruskal.test, 134, 136, 142
ks.test, 134, 136, 142
list, 24, 100, 103, 106, 111, 115, 119, 123, 126, 129, 130
matrix, 6, 16
NA, 126, 130, 147
numeric, 23
PlantGrowth, 62, 98, 131
prcomp, 79, 81, 122, 124
pretty, 28, 65, 76
psych, 102, 106, 126
qplot, 39
read.spss, 22
recode, 143
sjc.cluster, 5, 7–9, 12, 13, 15, 16
sjc.dend, 6, 7, 8, 9
sjc.elbow, 5–7, 8, 9, 11, 12, 16
sjc.grpdisc, 6–8, 9, 12, 13, 15, 16
sjc.kgap, 5, 7, 8, 10, 12, 15, 16
sjc.qclus, 11
sji.convertToLabel, 4, 16, 17–21, 23
sji.convertToValue, 17, 17, 18–21, 23
sji.getValueLabels, 4, 13, 17, 18, 19–23,
  26, 36, 42, 48, 53, 64, 75
sji.getValueVariableLabels, 4, 17, 18, 19,
  20–23, 72
sji.setValueLabels, 18, 19, 20, 21
sji.setValueVariableLabels, 19, 21, 22, 26, 27,
  30, 32, 35, 38, 53, 56, 59, 85, 90, 95,
  97, 100, 125, 129
sji.SPSS, 4, 13, 17–21, 22, 23, 24, 26, 36, 42,
  48, 53, 62, 64, 72, 75, 98, 131
sji.viewSPSS, 4, 23, 23, 104
sjp.aov1, 25, 134
sjp.chi2, 29
sjp.corr, 31, 101
sjp.frq, 4, 35, 107
sjp.glm, 41, 46, 50, 112
sjp.glm.ma, 45, 46, 50
sjp.glmm, 47, 78
sjp.grpfrq, 51, 148, 149
sjp.likert, 58, 92, 127
sjp.lm, 63, 70, 71, 74, 78, 84, 120, 135
sjp.lm.int, 66, 67, 71, 84
sjp.lm.ma, 66, 70, 71, 78, 84
sjp.lm1, 72, 88
sjp.lmm, 50, 74
sjp.pca, 79, 116, 124, 137, 145
sjp.reglin, 66, 70, 71, 74, 83, 88
sjp.scatter, 66, 74, 84, 84
sjp.stackfrq, 62, 88, 127
sjp.vif, 93
sjp.xtab, 94, 132
sjPlot (sjPlot-package), 3
sjPlot-package, 3
sjt.corr, 99
sjt.df, 4, 24, 102, 114–116, 145
sjt.frq, 39, 105
sjt.glm, 50, 109, 120
sjt.itemanalysis, 82, 113, 124, 137, 145
sjt.lm, 78, 112, 117, 135
sjt.pca, 82, 114, 116, 121, 137, 145
sjt.stackfrq, 92, 125
sjt.xtab, 98, 107, 129, 148, 149
sju.adjustPlotRange.y, 133
sju.aov1.levene, 29, 134, 136, 142
sju.betaCoef, 66, 135
sju.chi2.gof, 134, 136, 142
sju.cramer, 136, 143, 148
sju.cronbach, 82, 116, 124, 137, 145
sju.dicho, 138
sju.groupVar, 39, 56, 105, 106, 138, 140, 141
sju.groupVarLabels, 139, 140
sju.mwu, 134, 136, 141
sju.phi, 137, 142, 148
sju.recode, 143
sju.recodeTo, 144
sju.reliability, 82, 114, 116, 124, 137, 145
sju.setNA, 146
sju.table.values, 132, 137, 143, 147
sju.weight, 148, 149
sju.weight2, 148, 149
sju.wordwrap, 150
stat_bin, 38
structure, 24, 81, 100, 103, 107, 111, 119, 123, 127, 131
summary, 110
t.test, 134, 136, 142
table, 137, 142, 147
varimax, 81, 124
wilcox.test, 134, 136, 141, 142
xtabs, 137, 142, 147, 149