Package 'OutliersO3'

March 23, 2019

Type Package

Title Draws Overview of Outliers (O3) Plots

Version 0.6.2

Date 2019-03-27

Author Antony Unwin

Maintainer Antony Unwin <unwin@math.uni-augsburg.de>

Description
Potential outliers are identified for all combinations of a dataset's variables. O3 plots are described in Unwin(2019) <doi:10.1080/10618600.2019.1575226>. The available methods are HDoutliers() from the package 'HDoutliers', FastPCS() from the package 'FastPCS', mvBACON() from 'robustX', adjOutlyingness() from 'robustbase', DectectDeviatingCells() from 'cellWise', covMcd() from 'robustbase'.

Depends R (>= 3.3.0)

Imports stats, utils, grDevices, rlist, ggplot2, dplyr, tidyr, forcats, HDoutliers, robustbase, robustX, FastPCS, cellWise (>= 2.1.0), GGally, memisc

License GPL (>= 2)

Encoding UTF-8

LazyData true

Suggests knitr, gridExtra, rmarkdown, languageR

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-23 22:36:30 UTC

R topics documented:

`Election2005` *Election2005 data*

Description

A data set for the German 'Bundestag' election of 2005. It includes information about the elections in 2005 and in 2002 separately for each of the 299 constituencies and also demographic and other information about the constituencies themselves.

Usage

```
data(Election2005)
```

Format

A data frame with 299 observations on 70 variables. The variables of the data set are:

- 1-4: general information about the constituencies (ID and name of district, Bundesland)
- 5-40: demographic and economic information
- 41 - 48: general information about the elections in 2009 and 2005
  - `WBerechE`: number of eligible voters
  - `WE`: votes cast
  - `UngZE`: invalid second votes
  - `GulZE`: valid second votes
- 49- 70: results of the five biggest parties in 2009 and 2005 (smaller parties are summarized in the variable `Rest`). Everything with a 'V' or 'v' at the end is from the election in 2005. 'ze' and 'zv' at the end refer to second votes.

Details

This dataset has been taken from the package **mbgraphic** of Katrin Grimm.

Source

Original source was http://www.bundeswahlleiter.de.

Examples

```
dim(Election2005)
str(Election2005)
```

O3plotColours *Set colours for O3 plots*

Description

Provides a colour scheme for O3 plots.

Usage

```
O3plotColours(colours = c("khaki", "yellow", "red", "lightgreen",
              "lightblue", "red", "slategray1", "slategray2",
              "slategray3", "slategray4", "orange", "red"), colors)
```

Arguments

colours	A set of colours for the three kinds of plot. There are 12 in all and the defaults are khaki, yellow, red, lightgreen, lightblue, red, slategray1, slategray2, slategray3, slategray4, orange, red. The first three (1-3) are for plots with three different tolerance levels; the next three (4-6) are for plots comparing results for two methods; the final six (7-12) are for plots combining results of from three to six methods. If results from m methods are combined in one plot and m is more than two, then red is always used for m methods agreeing and the rest of the colour scale is shifted up accordingly.
colors	To allow users to write 'colors' instead of 'colours'.

Details

O3plotColours is provided for assigning colours for O3plots.

Value

A named list of colours.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

See Also

O3plotM and O3plotT

Examples

```
c1 <- O3prep(stackloss, k1=2, method=c("HDo", "BAC"), tolHDo=0.025, tolBAC=0.01)
c2 <- O3plotM(c1)
c2$gO3
col1 <- O3plotColours(colours=c("khaki", "yellow", "red", "darkseagreen", "gold1",
"red", "slategray1", "slategray2", "slategray3", "slategray4", "orange", "red"))
c3 <- O3plotM(c1, O3control=col1)
c3$gO3
```

`O3plotM` *Draws an Overview of Outliers (O3) plot for more than one method and parallel coordinate plots*

Description

Function for drawing Overview of Outliers (O3) plots for comparing outlier methods and for drawing supporting parallel coordinate plots.

Usage

```
O3plotM(outResults, caseNames=paste0("X", 1:nrow(outResults$data)),
 sortVars=TRUE, coltxtsize=14, O3control=O3plotColours())
```

Arguments

`outResults`	a list for each method, and within that for each variable combination, of the variables used, the indices of cases identified as outliers, and the outlier distances for all cases in the dataset.
`caseNames`	the ID variable used to identify the cases in an O3 plot, the default is the rownames from the dataset (so that they will then just be X7, X11, etc.)
`sortVars`	sort the variable columns by how often the variables occur in combinations, otherwise keep the variable order in the dataset
`coltxtsize`	set the size of text for column names in O3 plots (useful if there are so many columns that names overlap)
`O3control`	A list of colours for O3 plots. If omitted, `O3plotColours` gives the defaults.

Details

This function takes the output from `O3prep` and draws an O3 plot. If there are only two methods, then the default colours are red if both methods identify the case as an outlier for a variable combination and blue or green if only one method does. With more than two methods the default colours are red if all methods identify the outlier, orange if all but one method do, and shades of slategray otherwise.

The two parallel coordinate plots, one using the raw data and one using outlier distances, are examples of what can be done to explore the results in more detail. If you want these plots with other

highlighting then you can use outsTable with either the dataset or the Cs array to draw them using ggparcoord from **GGally** or whatever graphics tool you prefer.

The plots produced are ggplot objects so that you can work with them—to some extent—directly. In particular, plot margins can be set using `+ theme(plot.margin = unit(c(t, r, b, l), ''cm''))`, which is useful when the cases are labelled with the caseNames option.

Value

nOut	numbers of outliers found by each method
gpcp	a parallel coordinate plot of the dataset with cases ever found to be outliers coloured red
gO3	an O3 plot
gO3x	an O3 plot for three or more methods in which outliers identified by only one method for a variable combination are ignored.
gMethods	a parallel coordinate plot of the outlier distances calculated by each method for the full dataset with cases ever found to be outliers coloured red
outsTable	a table of all outliers found by case, variable combination, and method. The variable combination labels are a binary coding in the original order of the variables in the dataset.
Cs	a three-dimensional array of methods by variable combinations by cases of the outlier distances calculated.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

See Also

O3plotColours, HDoutliers in **HDoutliers**, FastPCS in **FastPCS**, mvBACON in **robustX**, adjOutlyingness in **robustbase**, DDC in **cellWise**, covMcd in **robustbase**

Examples

```
c1 <- O3prep(stackloss, k1=2, method=c("HDo", "BAC"), tolHDo=0.025, tolBAC=0.01)
c2 <- O3plotM(c1)
c2$nOut
c2$gpcp
c2$gO3

## Not run:
b1 <- O3prep(stackloss, method=c("HDo", "BAC", "DDC"), tolHDo=0.025, tolBAC=0.01, tolDDC=0.05)
b2 <- O3plotM(b1)
b2$nOut
b2$gpcp
b2$gO3
b2$outsTable

## End(Not run)
```

```
# It is advisable with large datasets to check the number of outliers identified (nOut)
# before drawing graphics.  Occasionally methods find very many outliers.
## Not run:
data(diamonds, package="ggplot2")
data <- diamonds[1:5000, c(1, 5, 6, 8:10)]
pPa <- O3prep(data, method=c("PCS", "adjOut"), tolPCS=0.01, toladj=0.01, boxplotLimits=10)
pPx <- O3plotM(pPa)
pPx$nOut

## End(Not run)
```

`O3plotT` *Draws an Overview of Outliers (O3) plot for one method and parallel coordinate plots*

Description

Function for drawing Overview of Outliers (O3) plots for one method and up to 3 tolerance levels and for drawing supporting parallel coordinate plots.

Usage

```
O3plotT(outResults, caseNames=paste0("X", 1:nrow(outResults$data)),
 sortVars=TRUE, coltxtsize=14, O3control=O3plotColours())
```

Arguments

`outResults`	a list for each tolerance level, and within that for each variable combination, of the variables used, the indices of cases identified as outliers, and the outlier distances for all cases in the dataset.
`caseNames`	the ID variable used to identify the cases in an O3 plot, the default is the row-names from the dataset (so that they will then just be X7, X11, etc.)
`sortVars`	sort the variable columns by how often the variables occur in combinations, otherwise keep the variable order in the dataset
`coltxtsize`	set the size of text for column names in O3 plots (useful if there are so many columns that names overlap)
`O3control`	A list of colours for O3 plots. If omitted, `O3plotColours` gives the defaults.

Details

This function takes the output from `O3prep` and draws an O3 plot with up to 3 different tolerance levels. The default colours are khaki for the least strict tolerance level, yellow for the next, and red for the strictest.

The two parallel coordinate plots, one using the raw data and one using outlier distances, are examples of what can be done to explore the results in more detail. If you want these plots with other highlighting then you can use outsTable with either the dataset or the Ds array to draw them using `ggparcoord` from **GGally** or whatever graphics tool you prefer.

The plots produced are ggplot objects so that you can work with them—to some extent—directly. In particular, plot margins can be set using `+ theme(plot.margin = unit(c(t, r, b, l), ''cm''))`, which is useful when the cases are labelled with the caseNames option.

Value

`nOut`	numbers of outliers found for the specified tolerance levels
`gpcp`	a parallel coordinate plot of all the data with cases ever found to be outliers coloured red
`gO3`	an O3 plot
`gCombs`	a parallel coordinate plot of the outlier distances calculated for each variable combination for the full dataset with cases found to be outliers at the strictest tolerance level coloured red
`outsTable`	a table of all outliers found by case, variable combination, and tolerance level. The variable combination labels are a binary coding in the original order of the variables in the dataset.
`Ds`	a three-dimensional array of tolerance levels by variable combinations by cases of the outlier distances calculated.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

See Also

`O3plotColours`, `HDoutliers` in **HDoutliers**, `FastPCS` in **FastPCS**, `mvBACON` in **robustX**, `adjOutlyingness` in **robustbase**, `DDC` in **cellWise**, `covMcd` in **robustbase**

Examples

```
a0 <- O3prep(stackloss, method="PCS", tols=0.05, boxplotLimits=3)
a1 <- O3plotT(a0)
a1$nOut
a1$gO3

b0 <- O3prep(stackloss, method="BAC", k1=2, tols=0.01, boxplotLimits=6)
b1 <- O3plotT(b0)
b1$nOut
b1$gpcp
b1$gO3

## Not run:
a2 <- O3prep(stackloss, method="PCS", tols=c(0.1, 0.05, 0.01), boxplotLimits=c(3, 6, 10))
a3 <- O3plotT(a2)
a3$nOut
a3$gpcp
a3$gO3
a3$outsTable

## End(Not run)
```

O3prep *Identify outliers for different combinations of variables*

Description

Check the dataset and parameters prior to analysis. Identify outliers for the variable combinations and methods/tolerance levels specified. Prepare input for the two plotting functions `O3plotT` and `O3plotM`.

Usage

```
O3prep(data, k1=1, K=ncol(data), method="HDo", tols=0.05, boxplotLimits=c(6, 10, 12),
     tolHDo=0.05, tolPCS=0.01, tolBAC=0.001, toladj=0.05, tolDDC=0.01, tolMCD=0.000001)
```

Arguments

`data`	dataset to be checked for outliers
`k1`	lowest number of variables in a combination
`K`	highest number of variables in a combination
`method`	method(s) used for identifying outliers (up to six can be used)
`tols`	outlier tolerance level(s) when only one method is specified, up to three can be used. For consistent use of the argument, it is transformed for some of the methods. See details below of how the argument is applied for each approach.
`boxplotLimits`	up to three boxplot limits are used, matching the number of tolerance levels, if a method does not apply for a single variable.
`tolHDo`	an individual outlier tolerance level for the HDoutliers method. The default in **HDoutliers**, alpha, is 0.05.
`tolPCS`	an individual outlier tolerance level for the FastPCS method. This equals (1-alpha) for the argument in **FastPCS**, where the default is 0.5.
`tolBAC`	an individual outlier tolerance level for the mvBACON method. The default for alpha in **robustX** is 0.95. This seems high, but it is divided by n, the dataset size.
`toladj`	an individual outlier tolerance level for the adjOutlyingness method. This equals (1-alpha.cutoff) for the argument in **robustbase**, where the default is 0.75.
`tolDDC`	an individual outlier tolerance level for the DDC method. This equals (1-tolProb) for the argument in **cellWise**, where the default is 0.99.
`tolMCD`	an individual outlier tolerance level for the covMcd method. The default is 0.025 (based on the help page for plot.mcd in **robustbase**). This is NOT the alpha argument in `covMcd`, which is used for determining subset size and set to 0.9 in OutliersO3.

Details

To check outliers for all possible combinations of variables choose k1=1 and K=number of variables in the dataset (the default).

The optional methods are "HDo" `HDoutliers` (from **HDoutliers**), "PCS" `FastPCS` (**FastPCS**), "BAC" `mvBACON` (**robustX**), "adjOut" `adjOutlyingness` (**robustbase**), "DDC" `DDC` (**Cellwise**), "MCD" `covMcd` (**robustbase**). References for all these methods can be found on their help pages, linked below. (Note that **Cellwise** has renamed its function `DetectDeviatingCells`. Since version 2.1.0 `DDC` is used instead.)

If only one method is specified, then up to three tolerance levels (tols) and three boxplot limits (boxplotLimits) can be specified.

`tol` is the argument determining outlyingness and should be set low, as in `HDoutliers` and `mvBACON`, where it is called alpha, and in `covMcd`. For the other methods `(1-tol)` is used. In `DDC` the argument is called tolProb. Using the same tolerance level for all methods does not make them directly comparable, which is why it is recommended to set them individually when drawing a comparative O3 plot. The defaults suggested on the methods' help pages mostly found too many outliers and so other defaults have been set. Users need to decide for themselves, possibly dependent on the dataset they are analysing.

Methods "HDo", "mvBACON", "adjOut", and "MCD" can analyse single variables. For the other methods boxplot limits are used for single variables and any case > (Q3 + boxplotLimit*IQR) or < (Q1 - boxplotLimit*IQR) is classed an outlier, where `boxplotLimit` is the limit specified.

Value

`data`	the dataset analysed
`nw`	the number of variable combinations analysed
`mm`	the outlier methods used
`tols`	the individual tolerance levels for the outlier methods used (if more than one), otherwise up to 3 tolerance levels used for one method
`outList`	a list for each method/tolerance level, and within that for each variable combination, of the variables used, the indices of cases identified as outliers, and the outlier distances for all cases in the dataset.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

See Also

`HDoutliers` in **HDoutliers**, `FastPCS` in **FastPCS**, `mvBACON` in **robustX**, `adjOutlyingness` in **robustbase**, `DDC` in **cellWise**, `covMcd` in **robustbase**

Examples

```
a0 <- O3prep(stackloss, method="PCS", tols=0.05, boxplotLimits=3)

b0 <- O3prep(stackloss, method=c("BAC", "adjOut"), k1=2, tols=0.01, boxplotLimits=6)
```

```
## Not run:
a1 <- O3prep(stackloss, method="PCS", tols=c(0.1, 0.05, 0.01), boxplotLimits=c(3, 6, 10))

b1 <- O3prep(stackloss, method=c("HDo", "BAC", "DDC"), tolHDo=0.025, tolBAC=0.01, tolDDC=0.05)

## End(Not run)
```

`OutliersO3` *OutliersO3: displays and compares potential outliers*

Description

Up to six different methods can be used to identify potential outliers in a dataset. An O3 (Overview of Outliers) plot and supporting parallel coordinate plots are drawn.

Details

An Overview of Outliers (O3) plot is the result of checking for potential outliers for every possible combination of (numeric) variables. It shows which cases are identified as outliers for each combination for which any outliers are found.

The available methods are "HDo" `HDoutliers` (from **HDoutliers**), "PCS" `FastPCS` (**FastPCS**), "BAC" `mvBACON` (**robustX**), "adjOut" `adjOutlyingness` (**robustbase**), "DDC" `DDC` (**Cellwise**), "MCD" `covMcd` (**robustbase**). Outlier tolerance levels can be set individually for each method.

Plots can be drawn for a single method with up to 3 tolerance levels, for two methods showing which cases are found by one or both, or for up to 6 methods showing how often a case is identified as an outlier.

If a method cannot be used for single variables, then boxplot limits defined by the argument boxplotLimits are used.

If only one method is specified, then up to three tolerance levels (tols) and boxplot limits (boxplotLimits) can be specified.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

Thanks are due to Bill Venables for some key coding advice.

Index