

Package ‘faoutlier’

July 2, 2014

Version 0.4

Date 2013-04-15

Type Package

Title Influential case detection methods for factor analysis and SEM

Maintainer Phil Chalmers <rphilip.chalmers@gmail.com>

Description Tools for detecting and summarize influential cases that can affect exploratory and confirmatory factor analysis models as well as structural equation models more generally.

Depends R (>= 2.14), parallel, sem

Imports lattice, lavaan, MASS

Suggests testthat

ByteCompile yes

LazyLoad yes

LazyData yes

Repository CRAN

License GPL (>= 2)

URL <https://github.com/philchalmers/faoutlier>

Author Phil Chalmers [aut, cre]

NeedsCompilation no

Date/Publication 2014-05-24 08:18:50

R topics documented:

faoutlier	2
forward.search	2
gCD	4
holzinger	7
holzinger.outlier	7
LD	7
obs.resid	9
robustMD	11
setCluster	12

Index	14
--------------	-----------

faoutlier	<i>Influential case detection methods for FA and SEM</i>
-----------	--

Description

Influential case detection methods for factor analysis and SEM

Details

Implements robust Mahalanobis methods, generalized Cook's distances, likelihood ratio tests, model implied residuals, and various graphical methods to help detect and summarize influential cases that can affect exploratory and confirmatory factor analyses.

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

forward.search	<i>Forward search algorithm for outlier detection</i>
----------------	---

Description

The forward search algorithm begins by selecting a homogeneous subset of cases based on a maximum likelihood criteria and continues to add individual cases at each iteration given an acceptance criteria. By default the function will add cases that contribute most to the likelihood function and that have the closest robust Mahalanobis distance, however model implied residuals may be included as well.

Usage

```
forward.search(data, model, criteria = c("LD", "mah"), n.subsets = 1000,
  p.base = 0.4, print.messages = TRUE, ...)

## S3 method for class 'forward.search'
print(x, stat = "X2", ...)

## S3 method for class 'forward.search'
plot(x, y = NULL, stat = "X2",
  main = "Forward Search", type = c("p", "h"), ylab = "obs.resid", ...)
```

Arguments

data	matrix or data.frame
model	if a single numeric number declares number of factors to extract in exploratory factor analysis. If <code>class(model)</code> is a sem (semmod), or lavaan (character), then a confirmatory approach is performed instead
criteria	character strings indicating the forward search method Can contain 'LD' for likelihood distance, 'mah' for Mahalanobis distance, or 'res' for model implied residuals
n.subsets	a scalar indicating how many samples to draw to find a homogeneous starting base group
p.base	proportion of sample size to use as the base group
print.messages	logical; print how many iterations are remaining?
x	an object of class <code>forward.search</code>
stat	type of statistic to use. Could be 'X2', 'RMR', or 'gCD' for the model chi squared value, root mean square residual, or generalized Cook's distance, respectively
...	additional parameters to be passed
y	a null value ignored by plot
main	the main title of the plot
type	type of plot to use, default displays points and lines
ylab	the y label of the plot

Details

Note that `forward.search` is not limited to confirmatory factor analysis and can apply to nearly any model being studied where detection of influential observations is important.

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

See Also

[gCD](#), [LD](#), [robustMD](#), [setCluster](#)

Examples

```

## Not run:

#run all internal gCD and LD functions using multiple cores
setCluster()

#Exploratory
nfact <- 3
(FS <- forward.search(holzinger, nfact))
(FS.outlier <- forward.search(holzinger.outlier, nfact))
plot(FS)
plot(FS.outlier)

#Confirmatory with sem
model <- specifyModel()
  F1 -> Remndrs,   lam11
  F1 -> SntComp,  lam21
  F1 -> WrdsMean, lam31
  F2 -> MissNum,  lam41
  F2 -> MxdArit,  lam52
  F2 -> OddWrds,  lam62
  F3 -> Boots,    lam73
  F3 -> Gloves,   lam83
  F3 -> Hatchts,  lam93
  F1 <-> F1,   NA,   1
  F2 <-> F2,   NA,   1
  F3 <-> F3,   NA,   1

(FS <- forward.search(holzinger, model))
(FS.outlier <- forward.search(holzinger.outlier, model))
plot(FS)
plot(FS.outlier)

#Confirmatory with lavaan
model <- 'F1 =~ Remndrs + SntComp + WrdsMean
F2 =~ MissNum + MxdArit + OddWrds
F3 =~ Boots + Gloves + Hatchts'

(FS <- forward.search(holzinger, model))
(FS.outlier <- forward.search(holzinger.outlier, model))
plot(FS)
plot(FS.outlier)

## End(Not run)

```

Description

Compute generalize Cook's distances (gCD's) for exploratory and confirmatory FA. Can return DFBETA matrix if requested.

Usage

```
gCD(data, model, ...)

## S3 method for class 'gCD'
print(x, head = 0.05, DFBETAS = FALSE, ...)

## S3 method for class 'gCD'
plot(x, y = NULL, main = "Generalized Cook Distance",
      type = c("p", "h"), ylab = "gCD", ...)
```

Arguments

data	matrix or data.frame
model	if a single numeric number declares number of factors to extract in exploratory factor analysis (requires complete dataset, i.e., no missing). If class(model) is a sem (semmod), or lavaan (character), then a confirmatory approach is performed instead
x	an object of class gCD
head	a ratio of how many extreme gCD cases to display
DFBETAS	logical; attach DFBETA matrix attribute to returned result?
...	additional parameters to be passed
y	a NULL value ignored by the plotting function
main	the main title of the plot
type	type of plot to use, default displays points and lines
ylab	the y label of the plot

Details

Note that gCD is not limited to confirmatory factor analysis and can apply to nearly any model being studied where detection of influential observations is important.

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

References

Flora, D. B., LaBrish, C. & Chalmers, R. P. (2012). Old and new ideas for data screening and assumption testing for exploratory and confirmatory factor analysis. *Frontiers in Psychology*, 3, 1-21.

See Also

[LD](#), [obs.resid](#), [robustMD](#), [setCluster](#)

Examples

```
## Not run:

#run all gCD functions using multiple cores
setCluster()

#Exploratory
nfact <- 3
(gCDresult <- gCD(holzinger, nfact))
(gCDresult.outlier <- gCD(holzinger.outlier, nfact))
plot(gCDresult)
plot(gCDresult.outlier)

#-----
#Confirmatory with sem
model <- specifyModel()
  F1 -> Remndrs,   lam11
  F1 -> SntComp,   lam21
  F1 -> WrdsMean, lam31
  F2 -> MissNum,   lam41
  F2 -> MxdArit,   lam52
  F2 -> OddWrds,   lam62
  F3 -> Boots,     lam73
  F3 -> Gloves,    lam83
  F3 -> Hatchts,   lam93
  F1 <-> F1,   NA,   1
  F2 <-> F2,   NA,   1
  F3 <-> F3,   NA,   1

(gCDresult2 <- gCD(holzinger, model))
(gCDresult2.outlier <- gCD(holzinger.outlier, model))
plot(gCDresult2)
plot(gCDresult2.outlier)

#-----
#Confirmatory with lavaan
model <- 'F1 =~ Remndrs + SntComp + WrdsMean
F2 =~ MissNum + MxdArit + OddWrds
F3 =~ Boots + Gloves + Hatchts'

(gCDresult2 <- gCD(holzinger, model, orthogonal=TRUE))
(gCDresult2.outlier <- gCD(holzinger.outlier, model, orthogonal=TRUE))
plot(gCDresult2)
plot(gCDresult2.outlier)

## End(Not run)
```

holzinger	<i>Description of holzinger data</i>
-----------	--------------------------------------

Description

A sample of 100 simulated cases from the infamous Holzinger dataset using 9 variables.

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

holzinger.outlier	<i>Description of holzinger data with 1 outlier</i>
-------------------	---

Description

A sample of 100 simulated cases from the infamous Holzinger dataset using 9 variables, but with 1 outlier added to the dataset. The first row was replaced by adding 2 to five of the observed variables (odd-numbered items) and subtracting 2 from the other four observed variables (even-numbered items).

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

LD	<i>Likelihood Distance</i>
----	----------------------------

Description

Compute likelihood distances between models when removing the i_{th} case.

Usage

```
LD(data, model, ...)  
  
## S3 method for class 'LD'  
print(x, ncases = 10, digits = 5, ...)  
  
## S3 method for class 'LD'  
plot(x, y = NULL, main = "Likelihood Distance", type = c("p",  
  "h"), ylab = "LD", absolute = FALSE, ...)
```

Arguments

<code>data</code>	matrix or data.frame
<code>model</code>	if a single numeric number declares number of factors to extract in exploratory factor analysis (requires complete dataset, i.e., no missing). If <code>class(model)</code> is a sem (semmod), or lavaan (character), then a confirmatory approach is performed instead
<code>x</code>	an object of class LD
<code>ncases</code>	number of extreme cases to display
<code>digits</code>	number of digits to round in the printed result
<code>...</code>	additional parameters to be passed
<code>y</code>	a NULL value ignored by the plotting function
<code>type</code>	type of plot to use, default displays points and lines
<code>main</code>	the main title of the plot
<code>ylab</code>	the y label of the plot
<code>absolute</code>	logical; use absolute values instead of deviations?

Details

Note that LD is not limited to confirmatory factor analysis and can apply to nearly any model being studied where detection of influential observations is important.

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

References

Flora, D. B., LaBrish, C. & Chalmers, R. P. (2012). Old and new ideas for data screening and assumption testing for exploratory and confirmatory factor analysis. *Frontiers in Psychology*, 3, 1-21.

See Also

[gCD](#), [obs.resid](#), [robustMD](#), [setCluster](#)

Examples

```
## Not run:

#run all LD functions using multiple cores
setCluster()

#Exploratory
nfact <- 3
(LDresult <- LD(holzinger, nfact))
(LDresult.outlier <- LD(holzinger.outlier, nfact))
plot(LDresult)
```



```

plot(LDresult.outlier)

#-----
#Confirmatory with sem
model <- specifyModel()
  F1 -> Remndrs,   lam11
  F1 -> SntComp,  lam21
  F1 -> WrldMean, lam31
  F2 -> MissNum,  lam42
  F2 -> MxdArit,  lam52
  F2 -> OddWrds,  lam62
  F3 -> Boots,    lam73
  F3 -> Gloves,   lam83
  F3 -> Hatchts,  lam93
  F1 <-> F1,   NA,   1
  F2 <-> F2,   NA,   1
  F3 <-> F3,   NA,   1

(LDresult <- LD(holzinger, model))
(LDresult.outlier <- LD(holzinger.outlier, model))
plot(LDresult)
plot(LDresult.outlier)

#-----
#Confirmatory with lavaan
model <- 'F1 =~ Remndrs + SntComp + WrldMean
F2 =~ MissNum + MxdArit + OddWrds
F3 =~ Boots + Gloves + Hatchts'

(LDresult <- LD(holzinger, model, orthogonal=TRUE))
(LDresult.outlier <- LD(holzinger.outlier, model, orthogonal=TRUE))
plot(LDresult)
plot(LDresult.outlier)

## End(Not run)

```

obs.resid

Model predicted residual outliers

Description

Compute model predicted residuals for each variable using regression estimated factor scores.

Usage

```
obs.resid(data, model, ...)
```

```
## S3 method for class 'obs.resid'
print(x, restype = "obs", ...)
```

```
## S3 method for class 'obs.resid'
plot(x, y = NULL, main = "Observed Residuals",
     type = c("p", "h"), restype = "obs", ...)
```

Arguments

data	matrix or data.frame
model	if a single numeric number declares number of factors to extract in exploratory factor analysis. If <code>class(model)</code> is a sem (semmod), or lavaan (character), then a confirmatory approach is performed instead
x	an object of class <code>obs.resid</code>
restype	type of residual used, either 'obs' for observation value (inner product), 'res' or 'std_res' for unstandardized and standardized for each variable, respectively
...	additional parameters to be passed
y	a NULL value ignored by the plotting function
main	the main title of the plot
type	type of plot to use, default displays points and lines

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

References

Flora, D. B., LaBrish, C. & Chalmers, R. P. (2012). Old and new ideas for data screening and assumption testing for exploratory and confirmatory factor analysis. *Frontiers in Psychology*, 3, 1-21.

See Also

[gCD](#), [LD](#), [robustMD](#)

Examples

```
## Not run:
data(holzinger)
data(holzinger.outlier)

#Exploratory
nfact <- 3
(ORresult <- obs.resid(holzinger, nfact))
(ORresult.outlier <- obs.resid(holzinger.outlier, nfact))
plot(ORresult)
plot(ORresult.outlier)

#-----
```

```

#Confirmatory with sem
model <- specifyModel()
  F1 -> Remndrs,   lam11
  F1 -> SntComp,   lam21
  F1 -> WrdsMean, lam31
  F2 -> MissNum,   lam41
  F2 -> MxdArit,   lam52
  F2 -> OddWrds,   lam62
  F3 -> Boots,     lam73
  F3 -> Gloves,    lam83
  F3 -> Hatchts,   lam93
  F1 <-> F1,   NA,   1
  F2 <-> F2,   NA,   1
  F3 <-> F3,   NA,   1

(ORresult <- obs.resid(holzinger, model))
(ORresult.outlier <- obs.resid(holzinger.outlier, model))
plot(ORresult)
plot(ORresult.outlier)

#-----
#Confirmatory with lavaan
model <- 'F1 =~ Remndrs + SntComp + WrdsMean
F2 =~ MissNum + MxdArit + OddWrds
F3 =~ Boots + Gloves + Hatchts'

(obs.resid2 <- obs.resid(holzinger, model, orthogonal=TRUE))
(obs.resid2.outlier <- obs.resid(holzinger.outlier, model, orthogonal=TRUE))
plot(obs.resid2)
plot(obs.resid2.outlier)

## End(Not run)

```

robustMD

Robust Mahalanobis

Description

Obtain Mahalanobis distances using the robust computing methods found in the MASS package. This function is generally only applicable to models with continuous variables.

Usage

```

robustMD(data, method = "mve", ...)

## S3 method for class 'robmah'
print(x, gt = 0, digits = 5, ...)

## S3 method for class 'robmah'
plot(x, y = NULL, type = "xyplot", main, ...)

```

Arguments

data	matrix or data.frame
method	type of estimation for robust means and covariance (see <code>cov.rob</code>)
digits	number of digits to round in the final result
x	an object of class robmah
gt	only print values with MD's greater than gt
...	additional arguments to pass to <code>MASS::cov.rob()</code>
y	empty parameter passed to plot
type	type of plot to display, can be either 'qqplot' or 'xyplot'
main	title for plot. If missing titles will be generated automatically

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

References

Flora, D. B., LaBrish, C. & Chalmers, R. P. (2012). Old and new ideas for data screening and assumption testing for exploratory and confirmatory factor analysis. *Frontiers in Psychology*, 3, 1-21.

See Also

[gCD](#), [obs.resid](#), [LD](#)

Examples

```
## Not run:
data(holzinger)
output <- robustMD(holzinger)
output
plot(output)
plot(output, type = 'qqplot')

## End(Not run)
```

setCluster

Define a parallel cluster object to be used in internal functions

Description

This function defines a object that is placed in a relevant internal environment defined in faoutlier. Internal functions will utilize this object automatically to capitalize on parallel processing architecture. The object defined is a call from `parallel::makeCluster()`. Note that if you are defining other parallel objects (for simulation desings, for example) it is not recommended to define a cluster.

Usage

```
setCluster(ncores, remove = FALSE)
```

Arguments

<code>ncores</code>	number of cores to be used in the returned object which is passed to <code>parallel::makeCluster()</code> . If no input is given the maximum number of available cores will be used
<code>remove</code>	logical; remove previously defined cluster object?

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

Examples

```
## Not run:  
  
#make 4 cores available for parallel computing  
setCluster(4)  
  
#' #stop and remove cores  
setCluster(remove = TRUE)  
  
#use all available cores  
setCluster()  
  
## End(Not run)
```

Index

- *Topic **cooks**
 - gCD, [4](#)
 - LD, [7](#)
- *Topic **covariance**
 - obs.resid, [9](#)
 - robustMD, [11](#)
- *Topic **data**
 - holzinger, [7](#)
 - holzinger.outlier, [7](#)
- *Topic **forward.search**
 - forward.search, [2](#)
- *Topic **package**
 - faoutlier, [2](#)
- *Topic **parallel**
 - setCluster, [12](#)

[cov.rob](#), [12](#)

[faoutlier](#), [2](#)
[faoutlier-package \(faoutlier\)](#), [2](#)
[forward.search](#), [2](#)

[gCD](#), [3](#), [4](#), [8](#), [10](#), [12](#)

[holzinger](#), [7](#)
[holzinger.outlier](#), [7](#)

[LD](#), [3](#), [6](#), [7](#), [10](#), [12](#)

[obs.resid](#), [6](#), [8](#), [9](#), [12](#)

[plot.forward.search \(forward.search\)](#), [2](#)
[plot.gCD \(gCD\)](#), [4](#)
[plot.LD \(LD\)](#), [7](#)
[plot.obs.resid \(obs.resid\)](#), [9](#)
[plot.robmah \(robustMD\)](#), [11](#)
[print.forward.search \(forward.search\)](#), [2](#)
[print.gCD \(gCD\)](#), [4](#)
[print.LD \(LD\)](#), [7](#)
[print.obs.resid \(obs.resid\)](#), [9](#)
[print.robmah \(robustMD\)](#), [11](#)

[robustMD](#), [3](#), [6](#), [8](#), [10](#), [11](#)

[setCluster](#), [3](#), [6](#), [8](#), [12](#)