Package ‘faoutlier’

October 13, 2022

Version  0.7.6
Type    Package
Title   Influential Case Detection Methods for Factor Analysis and
         Structural Equation Models
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 (Chalmers, 2015, <doi:10.1177/0146621615597894>;
Depends  R (>= 3.0.2), sem, mvtnorm, parallel
Imports  methods, lattice, lavaan, mirt (>= 1.32.1), MASS, pbapply (>=
         1.3-0)
ByteCompile yes
LazyLoad yes
LazyData yes
Repository CRAN
License GPL (>= 2)
URL    https://github.com/philchalmers/faoutlier
RoxygenNote  7.1.1
NeedsCompilation no
Author  Phil Chalmers [aut, cre]
Date/Publication 2021-01-10 15:50:05 UTC

R topics documented:

faoutlier .......................................................... 2
forward.search ................................................... 2
gCD ................................................................. 5
GOF ................................................................. 8
holzinger ......................................................... 10
faoutlier  

Influential case detection methods for FA and SEM

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>

References


forward.search  

Forward search algorithm for outlier detection

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.
forward.search

Usage

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

## S3 method for class 'forward.search'
print(x, ncases = 10, stat = "GOF", ...)

## S3 method for class 'forward.search'
plot(
  x,
  y = NULL,
  stat = "GOF",
  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 class(model) is a sem (semmod), or lavaan (character), then
a confirmatory approach is performed instead

criteria character strings indicating the forward search method Can contain 'GOF' for
goodness of fit 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?

... additional parameters to be passed

x an object of class forward.search

ncases number of final cases to print in the sequence

stat type of statistic to use. Could be 'GOF', 'RMR', or 'gCD' for the model chi
squared value, root mean square residual, or generalized Cook's distance, re-
spectively

y a null value ignored by plot
forward.search

  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>

References


See Also
gCD, GOF, LD, robustMD, setCluster

Examples

## Not run:

#run all internal gCD and GOF 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 <- sem::specifyModel()
F1 -> Remndrs, lam11
F1 -> SntComp, lam21
F1 -> WrdMean, lam31
F2 -> MissNum, lam41
F2 -> MxdArit, lam52
**gCD**

**Generalized Cook's Distance**

**Description**

Compute generalize Cook's distances (gCD's) for exploratory and confirmatory FA. Can return DFBETA matrix if requested. If mirt is used, then the values will be associated with the unique response patterns instead.

**Usage**

```r
library(gCD)

# Example data and model
holzinger <- HolzingerSimpleStructure()
holzinger.outlier <- HolzingerSimpleStructure(outlier = TRUE)
model <- 'F1 =~ Remndrs + SntComp + WrdMean
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)

# Confirmatory with lavaan
model <- 'F1 =~ Remndrs + SntComp + WrdMean
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)
```
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 per-
formed instead

progress logical; display the progress of the computations in the console?

additional parameters to be passed

x an object of class gCD

ncases number of extreme cases to display

DFBETAS logical; return DFBETA matrix in addition to gCD? If TRUE, a list is returned

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

doi: 10.1177/0146621615597894

assumption testing for exploratory and confirmatory factor analysis. Frontiers in Psychology, 3,

See Also

LD, obs.resid, robustMD, setCluster
Examples

```r
## 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 <- sem::specifyModel()
F1 -> Remndrs, lam11
F1 -> SntComp, lam21
F1 -> WrdMean, 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 + WrdMean
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)

# categorical data with mirt
library(mirt)
data(LSAT7)
dat <- expand.table(LSAT7)
model <- mirt.model('F = 1-5')
result <- gCD(dat, model)
```
plot(result)

mod <- mirt(dat, model)
res <- residuals(mod, type = 'exp')
cbind(res, gCD=round(result$gCD, 3))

## End(Not run)

---

### Description

Compute Goodness of Fit distances between models when removing the $i_{th}$ case. If mirt is used, then the values will be associated with the unique response patterns instead.

### Usage

```r
GOF(data, model, M2 = TRUE, progress = TRUE, ...)
```

```r
## S3 method for class 'GOF'
print(x, ncases = 10, digits = 5, ...)
```

```r
## S3 method for class 'GOF'
plot(
  x,
  y = NULL,
  main = "Goodness of Fit Distance",
  type = c("p", "h"),
  ylab = "GOF",
  absolute = FALSE,
  ...
)
```

### 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. Finally, if the model is defined with `mirt::mirt.model()` then distances will be computed for categorical data with the mirt package
- **M2**: logical; use the M2 statistic for when using mirt objects instead of G2?
- **progress**: logical; display the progress of the computations in the console?
- **...**: additional parameters to be passed
x an object of class GOF
ncases number of extreme cases to display
digits number of digits to round in the printed result
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
absolute logical; use absolute values instead of deviations?

Details

Note that GOF 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


See Also
gCD, LD, obs.resid, robustMD, setCluster

Examples

## Not run:
#run all GOF functions using multiple cores
setCluster()

#Exploratory
nfact <- 3
(GOFresult <- GOF(holzinger, nfact))
(GOFresult.outlier <- GOF(holzinger.outlier, nfact))
plot(GOFresult)
plot(GOFresult.outlier)

## include a progress bar
GOFresult <- GOF(holzinger, nfact, progress = TRUE)
holzinger

Description of holzinger data

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

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

References


---

holzinger.outlier  
*Description of holzinger data with 1 outlier*

---

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>

References


**Description**

Compute likelihood distances between models when removing the $i_{th}$ case. If there are no missing data then the GOF will often provide equivalent results. If mirt is used, then the values will be associated with the unique response patterns instead.

**Usage**

```r
LD(data, model, progress = TRUE, ...)  
print(x, ncases = 10, digits = 5, ...)  
plot(x, y = NULL, main = "Likelihood Distance", type = c("p", "h"), ylab = "LD", absolute = FALSE, ...)  
```

**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. Finally, if the model is defined with `mirt::mirt.model()` then distances will be computed for categorical data with the mirt package
- `progress`: logical; display the progress of the computations in the console?
- `...`: additional parameters to be passed
- `x`: an object of class LD
- `ncases`: number of extreme cases to display
- `digits`: number of digits to round in the printed result
- `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
- `absolute`: 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


See Also

gCD, GOF, obs.resid, robustMD, setCluster

Examples

```r
## 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)

## add a progress meter
LDresult <- LD(holzinger, nfact, progress = TRUE)
```

```r
#Confirmatory with sem
model <- sem::specifyModel()
F1 -> Remndrs, lam11
F1 -> SntComp, lam21
F1 -> WrdMean, lam31
F2 -> MissNum, lam42
F2 -> MxdArit, lam52
F2 -> OddWrds, lam62
F3 -> Boots, lam73
F3 -> Gloves, lam83
F3 -> Hatchts, lam93
```
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 class(model) is a sem (semmod), or lavaan (character), then
a confirmatory approach is performed instead
...
additional parameters to be passed
x an object of class obs.resid
restype type of residual used, either 'obs' for observation value (inner product), 'res'
or 'std_res' for unstandardized and standardized for each variable, respec-
tively
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

doi: 10.1177/0146621615597894
assumption testing for exploratory and confirmatory factor analysis. Frontiers in Psychology, 3,

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 <- sem::specifyModel()
F1 -> Remndrs, lam11
F1 -> SntComp, lam21
F1 -> WrdMean, 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 + WrdMean
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, ncases = 10, 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 `cov.rob`)
...: additional arguments to pass to `MASS::cov.rob()`
x: an object of class `robmah`
ncases: number of extreme cases to print
digits: number of digits to round in the final result
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


### See Also

gCD, obs.resid, LD

### Examples

```r
## 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 an 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 designs, for example) it is not recommended to define a cluster.

Usage

setCluster(spec, ..., remove = FALSE)

Arguments

- **spec**: input that is passed to parallel::makeCluster(). If no input is given the maximum number of available local cores will be used
- **...**: additional arguments to pass to parallel::makeCluster
- **remove**: logical; remove previously defined cluster object?

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

References


Examples

```r
## Not run:

#make 4 cores available for parallel computing
setCluster(4)

#stop and remove cores
setCluster(remova

#use all available cores
setCluster()
```
## End (Not run)
Index

* cooks
  gCD, 5
  GOF, 8
  LD, 12
* covariance
  obs.resid, 14
  robustMD, 16
* data
  holzinger, 10
  holzinger.outlier, 11
* forward.search
  forward.search, 2
* package
  faoutlier, 2
* parallel
  setCluster, 18

cov.rob, 17
faoutlier, 2
forward.search, 2
gCD, 4, 5, 9, 13, 15, 17
GOF, 4, 8, 12, 13
holzinger, 10
holzinger.outlier, 11
LD, 4, 6, 9, 12, 15, 17
obs.resid, 6, 9, 13, 14, 17
plot.forward.search (forward.search), 2
plot.gCD (gCD), 5
plot.GOF (GOF), 8
plot.LD (LD), 12
plot.obs.resid (obs.resid), 14
plot.robmah (robustMD), 16
print.forward.search (forward.search), 2
print.gCD (gCD), 5
print.GOF (GOF), 8

print.LD (LD), 12
print.obs.resid (obs.resid), 14
print.robmah (robustMD), 16

robustMD, 4, 6, 9, 13, 15, 16
setCluster, 4, 6, 9, 13, 18