Package 'GSE'

September 25, 2014

Type Package

Title Robust Estimation of Multivariate Location and Scatter in the Presence of Missing Data

Version 3.1

Date 2014-09-24

Author Andy Leung, Mike Danilov, Victor Yohai, Ruben Zamar

Maintainer Andy Leung <andy.leung@stat.ubc.ca>

Description Robust Estimation of Multivariate Location and Scatter in the Presence of Missing Data

License GPL (>= 2)

Depends R (>= 3.0.0), Rcpp (>= 0.10.0), MASS, methods, ggplot2

Suggests lattice, rrcov, xtable

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-09-25 10:27:53

R topics documented:

`auto` *Automobile data*

Description

This data set is taken from UCI repository, see reference. Past usage includes price prediction of cars using all numeric and boolean attributes (Kibler et al., 1989).

Usage

```
data(auto)
```

Format

A data frame with 205 observations on the following 26 variables, of which 15 are quantitative and 11 are categorical. The following description is extracted from UCI repository (Frank and Asuncion, 2010):

`Normalized-losses`	the relative average loss payment per insured vehicle year; ranged from 65 to 256
`Make`	Vehicle's make
`Fuel-type`	diesel, gas
`Aspiration`	std, turbo
`Num-of-doors`	four, two
`Body-style`	hardtop, wagon, sedan, hatchback, convertible
`Drive-wheels`	4wd, fwd, rwd
`Engine-location`	front, rear
`Wheel-base`	continuous from 86.6 120.9
`Length`	continuous from 141.1 to 208.1
`Width`	continuous from 60.3 to 72.3
`Height`	continuous from 47.8 to 59.8
`Curb-weight`	continuous from 1488 to 4066
`Engine-type`	dohc, dohcv, l, ohc, ohcf, ohcv, rotor

Num-of-cylinders	eight, five, four, six, three, twelve, two
Engine-size	continuous from 61 to 326
Fuel-system	1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi
Bore	continuous from 2.54 to 3.94
Stroke	continuous from 2.07 to 4.17
Compression-ratio	continuous from 7 to 23
Horsepower	continuous from 48 to 288
Peak-rpm	continuous from 4150 to 6600
City-mpg	continuous from 13 to 49
Highway-mpg	continuous from 16 to 54
Price	continuous from 5118 to 45400
Symboling	assigned insurance risk rating: -3, -2, -1, 0, 1, 2, 3

Source

The original data have been taken from the UCI Repository Of Machine Learning Databases at

- http://archive.ics.uci.edu/ml/datasets/Automobile.

References

Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Kibler, D., Aha, D.W., & Albert,M. (1989). Instance-based prediction of real-valued attributes. Computational Intelligence, Vol 5, 51–57.

---

boston *Boston Housing Data*

Description

Housing data for 506 census tracts of Boston from the 1970 census. The dataframe boston contains the corrected data by Harrison and Rubinfeld (1979). The data was for a few minor errors and augmented with the latitude and longitude of the observations. The original data can be found in the references below.

Usage

```
data(boston)
```

Format

The original data are 506 observations on 14 variables, medv being the target variable:

cmedv	corrected median value of owner-occupied homes in USD 1000's
crim	per capita crime rate by town
indus	proportion of non-retail business acres per town

nox	nitric oxides concentration (parts per 10 million)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted distances to five Boston employment centres
rad	index of accessibility to radial highways
tax	full-value property-tax rate per USD 10,000
ptratio	pupil-teacher ratio by town
b	$1000(B - 0.63)^2$ where B is the proportion of blacks by town
lstat	percentage of lower status of the population

Source

The original data have been taken from the UCI Repository Of Machine Learning Databases at

- http://www.ics.uci.edu/~mlearn/MLRepository.html,

the corrected data have been taken from Statlib at

- http://lib.stat.cmu.edu/datasets/

See Statlib and references there for details on the corrections. Both were converted to R format by Friedrich Leisch.

References

Harrison, D. and Rubinfeld, D.L. (1978). Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management*, **5**, 81–102.

Gilley, O.W., and R. Kelley Pace (1996). On the Harrison and Rubinfeld Data. *Journal of Environmental Economics and Management*, **31**, 403–405. [Provided corrections and examined censoring.]

Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Pace, R. Kelley, and O.W. Gilley (1997). Using the Spatial Configuration of the Data to Improve Estimation. *Journal of the Real Estate Finance and Economics*, **14**, 333–340. [Added georeferencing and spatial estimation.]

---

calcium *Calcium data*

Description

The Calcium data is from the article by Holcomb and Spalsbury (2005). The dataset used for class was compiled by Boyd, Delost, and Holcomb (1998) for the use of a study to determine if significant gender differences existed between subjects 65 years of age and older with regard to calcium, inorganic phosphorous, and alkaline phosphatase levels. Although the original data from Boyd, Delost, and Holcomb (1998) had observations needing investigation, Holcomb and Spalsbury (2005) further massaged the original data to include data problems and issues that have arisen in other research projects for pedagogical purposes.

Usage

```
data(calcium)
```

Format

A data frame with 178 observations on the following 8 variables.

obsno	Patient Observation Number
age	Age in years
sex	1=Male, 2=Female
alkphos	Alkaline Phosphatase International Units/Liter
lab	1=Metpath; 2=Deyor; 3=St. Elizabeth's; 4=CB Rouche; 5=YOH; 6=Horizon
cammol	Calcium mmol/L
phosmmol	Inorganic Phosphorus mmol/L
agegroup	Age group 1=65-69; 2=70-74; 3=75-79; 4=80-84; 5=85-89 Years

Source

The original data have been taken from the Journal of Statistics Education Databases at

- http://www.amstat.org/publications/jse/datasets/calcium.dat.txt,

the corrected data have been taken from Statlib at

- http://www.amstat.org/publications/jse/datasets/calciumgood.dat.txt

References

Boyd, J., Delost, M., and Holcomb, J., (1998). Calcium, phosphorus, and alkaline phosphatase laboratory values of elderly subjects, *Clinical Laboratory Science*, 11, 223-227.

Holcomb, J., and Spalsbury, A. (2005), Teaching Students to Use Summary Statistics and Graphics to Clean and Analyze Data. *Journal of Statistics Education*, 13, Number 3.

Examples

```
## Not run:
data(calcium)
## remove the categorical variables
calcium.cts <- subset(calcium, select=-c(obsno, sex, lab, agegroup) )
res <- GSE(calcium.cts)
getOutliers(res)
## able to identify majority of the contaminated cases identified
## in the reference

## End(Not run)
```

CovEM *Gaussian MLE of mean and covariance*

Description

Computes the Gaussian MLE via EM-algorithm for missing data.

Usage

```
CovEM(x, tol=0.001, maxiter=1000, print.step=0)
```

Arguments

x	a matrix or data frame. May contain missing values, but cannot contain columns with completely missing entries.
tol	tolerance level for the maximum relative change of the estimates. Default is 0.001.
maxiter	maximum iteration for the EM algorithm. Default is 1000.
print.step	this argument determines the level of printing which is done during the estimation process. The default value is 1 means that the number of iteration at convergence is printed out; a value of 0 means no printing occurs, except for error messages.

Value

An S4 object of class CovRobMiss-class. The output S4 object contains the following slots:

mu	Estimated location. Can be accessed via getLocation.
S	Estimated scatter matrix. Can be accessed via getScatter.
pmd	Squared partial Mahalanobis distances. Can be accessed via getDist.
pmd.adj	Adjusted squared partial Mahalanobis distances. Can be accessed via getDistAdj.
pu	Dimension of the observed entries for each case. Can be accessed via getDim.
call	Object of class "language". Not meant to be accessed.
x	Input data matrix. Not meant to be accessed.
p	Column dimension of input data matrix. Not meant to be accessed.
estimator	Character string of the name of the estimator used. Not meant to be accessed.

Author(s)

Mike Danilov, Andy Leung <andy.leung@stat.ubc.ca>

CovRobMiss-class *Class "CovRobMiss" – a superclass for the robust estimates of location and scatter for missing data*

Description

The Superclass of all the objects output from the various robust estimators of location and scatter for missing data, which includes Generalized S-estimator GSE, Extended Minimum Volumn Ellipsoid emve, and Huberized Pairwise HuberPairwise. It can also be constructed using the code partial.mahalanobis.

Objects from the Class

Objects can be created by calls of the form new("CovRobMiss", ...), but the best way of creating CovRobMiss objects is a call to either of the folowing functions:GSE, emve, HuberPairwise, and partial.mahalanobis, which all serve as a constructor.

Slots

mu Estimated location. Can be accessed via getLocation.

S Estimated scatter matrix. Can be accessed via getScatter.

pmd Square partial Mahalanobis distances. Can be accessed via getDist.

pmd.adj Adjusted square partial Mahalanobis distances. Can be accessed via getDistAdj.

pu Dimension of the observed entries for each case. Can be accessed via getDim.

call Object of class "language". Not meant to be accessed.

x Input data matrix. Not meant to be accessed.

p Column dimension of input data matrix. Not meant to be accessed.

estimator Character string of the name of the estimator used. Not meant to be accessed.

Methods

show signature(object = "CovRobMiss"): display the object

summary signature(object = "CovRobMiss"): calculate summary information

plot signature(object = "CovRobMiss", cutoff = "numeric"): plot the object. See plot

getDist signature(object = "CovRobMiss"): return the squared partial Mahalanobis distances

getDistAdj signature(object = "CovRobMiss"): return the adjusted squared partial Mahalanobis distances

getDim signature(object = "CovRobMiss"): return the dimension of observed entries for each case

getLocation signature(object = "CovRobMiss"): return the estimated location vector

getScatter signature(object = "CovRobMiss", cutoff = "numeric"): return the estimated scatter matrix

getMissing signature(object = "CovRobMiss"): return the case number with completely missing data, if any

getOutliers signature(object = "CovRobMiss", cutoff = "numeric"): return the case number(s) adjusted squared distances above (1 - cutoff)th quantile of chi-square p-degrees of freedom.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

See Also

GSE, emve, HuberPairwise, partial.mahalanobis

CovRobMissSc-class *Class "CovRobMissSc" – a subclass of "CovRobMiss" with scale estimate*

Description

The Superclass of the GSE-class and emve-class objects.

Objects from the Class

Objects can be created by calls of the form new("CovRobMissSc", ...), but the best way of creating CovRobMissSc objects is a call to either of the folowing functions:GSE or emve.

Slots

mu Estimated location. Can be accessed via getLocation.

S Estimated scatter matrix. Can be accessed via getScatter.

sc Estimated M-scale (either GS-scale or MVE-scale). Can be accessed via getScale.

pmd Square partial Mahalanobis distances. Can be accessed via getDist.

pmd.adj Adjusted square partial Mahalanobis distances. Can be accessed via getDistAdj.

pu Dimension of the observed entries for each case. Can be accessed via getDim.

call Object of class "language". Not meant to be accessed.

x Input data matrix. Not meant to be accessed.

p Column dimension of input data matrix. Not meant to be accessed.

estimator Character string of the name of the estimator used. Not meant to be accessed.

Extends

Class "CovRobMiss", directly.

Methods

In addition to methods inheritedfrom the class "CovRobMiss":

signature(object = "CovRobMissSc"): return the GS-scale or MVE-scale of the best candidate.

Author(s)

getScale Andy Leung <andy.leung@stat.ubc.ca>

See Also

GSE, CovRobMiss-class

---

emve *Extended Minimum Volume Ellipsoid (EMVE) in the presence of missing data*

Description

Computes the Extended S-Estimate (ESE) version of the minimum volume ellipsoid (EMVE), which is used as an initial estimator in Generlized S-Estimator (GSE) for missing data by default.

Usage

```
emve(x, n.resample=100, maxits=5, n.sub.size)
```

Arguments

x	a matrix or data frame. May contain missing values, but cannot contain columns with completely missing entries.
n.resample	integer indicating the number of subsamples. Default is 100. Must be at least 20.
maxits	integer indicating the maximum number of iterations of Gaussian MLE calculation for each subsample. Default is 5.
n.sub.size	optional integer indicating the size of a subsample. Must be at least the column dimension of the data.

Details

This function computes EMVE as described in Danilov et al. (2012). Subsampling algorithm is used for computing EMVE. By default, we take $N = 100$ subsamples of size $n_0 = p/(1-\alpha)$, where p is the dimension of the data and α is the fraction of missing data. The subsample size n_0 must be chosen to be larger than p to avoid singularity. Only the best 10 candidates of estimated location and scatter from N subsamples will be output.

In the algorithm, there exists a concentration step in which Gaussian MLE is computed for 50% of the data points using the classical EM-algorithm multiplied by a scalar factor. This step is repeated for each subsample. As the computation can be heavy as the number of subsample increases, we set by default the maximum number of iteration of classical EM-algorithm (i.e. maxits) as 5. Users are encouraged to refer to Danilov et al. (2012) for details about the algorithm and Rubin and Little (2002) for the classical EM-algorithm for missing data.

Value

An S4 object of class emve-class which is a subclass of the virtual class CovRobMissSc-class. The output S4 object contains the following slots:

mu	Estimated location. Can be accessed via getLocation.
S	Estimated scatter matrix. Can be accessed via getScatter.
sc	Estimated EMVE scale. Can be accessed via getScale.
cand.sc	EMVE scales for the top 10 candidates. Can be accessed via getCandidates.
cand.mu	a matrix of dimension 10 by p of the estimated EMVE location for the top 10 candidates, where p is the dimensi
cand.S	an array of dimension p by p by 10 of the estimated EMVE scatter for the top 10 candidates. Can be accessed vi
pmd	Squared partial Mahalanobis distances. Can be accessed via getDist.
pmd.adj	Adjusted squared partial Mahalanobis distances. Can be accessed via getDistAdj.
pu	Dimension of the observed entries for each case. Can be accessed via getDim.
call	Object of class "language". Not meant to be accessed.
x	Input data matrix. Not meant to be accessed.
p	Column dimension of input data matrix. Not meant to be accessed.
estimator	Character string of the name of the estimator used. Not meant to be accessed.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>, Ruben H. Zamar, Mike Danilov, Victor J. Yohai

References

Danilov, M., Yohai, V.J., Zamar, R.H. (2012). Robust Esimation of Multivariate Location and Scatter in the Presence of Missing Data. *Journal of the American Statistical Association* **107**, 1178–1186.

Rubin, D.B. and Little, R.J.A. (2002). *Statistical analysis with missing data* (2nd ed.). New York: Wiley.

See Also

GSE, emve-class

Examples

```
## Not run:
set.seed(12345)
n <- 100; p <- 10
A <- matrix(0.9, p, p); diag(A) <- 1
x <- mvrnorm(n, rep(0,p), A)

## Introduce 10
pcont <- 0.1; dcont <- 10
A.svd <- svd(A)
U <- A.svd$u

## Contaminated points in the direction corresponding to smallest eigenvalue
vv <- A.svd$u[,p]
```

```
uu <- 1/sqrt( t(vv)
vv <- vv*uu
ncont <- rbinom(n,1,pcont)
if( sum(ncont) > 0 )
x[which(ncont== 1),] <- dcont * matrix(vv,sum(ncont),p,byrow=T)

## Introduce 10
pmiss <- 0.1
nmiss <- matrix(rbinom(n*p,1,pmiss), n,p)
x[ which( nmiss == 1 ) ] <- NA

## Use EMVE
res <- emve(x)

## returns the list of candidate estimates
res.cand <- getCandidates(res)
str(res.cand)

## the element 'cand.Sigma' as the best 10 scatter
## in arrays with the 3rd dimension corresponds to their
## rankings, e.g. best candidate
res.cand$cand.S[,,1]
res.cand$S ## the best candidate also available as 'mu','S'

## EMVE results could pass on to GSE without recomputing
## the initial estimate as follows:
res.GSE <- GSE(x, mu0=res.cand$mu, S0=res.cand$S)

## End(Not run)
```

`emve-class` *Extended Minimum Volume Ellipsoid (EMVE) in the presence of missing data.*

Description

Class of Extended Minimum Volume Ellipsoid. It has the superclass of `CovRobMissSc`.

Objects from the Class

Objects can be created by calls of the form `new("emve", ...)`, but the best way of creating `emve` objects is a call to the function `emve` which serves as a constructor.

Slots

`mu` Estimated location. Can be accessed via `getLocation`.

`S` Estimated scatter matrix. Can be accessed via `getScatter`.

`sc` Estimated EMVE scale. Can be accessed via `getScale`.

`cand.sc` EMVE scales for the top 10 candidates. Can be accessed via `getCandidates`.

`cand.mu` a matrix of dimension 10 by p of the estimated EMVE location for the top 10 candidates, where p is the dimension of the data. Can be accessed via `getCandidates`.

`cand.S` an array of dimension p by p by 10 of the estimated EMVE scatter for the top 10 candidates. Can be accessed via `getCandidates`.

`pmd` Squared partial Mahalanobis distances. Can be accessed via `getDist`.

`pmd.adj` Adjusted squared partial Mahalanobis distances. Can be accessed via `getDistAdj`.

`pu` Dimension of the observed entries for each case. Can be accessed via `getDim`.

`call` Object of class `"language"`. Not meant to be accessed.

`x` Input data matrix. Not meant to be accessed.

`p` Column dimension of input data matrix. Not meant to be accessed.

`estimator` Character string of the name of the estimator used. Not meant to be accessed.

Extends

Class `"CovRobMissSc"`, directly.

Methods

The following methods are defined with the superclass "CovRobMiss":

show `signature(object = "CovRobMiss")`: display the object

summary `signature(object = "CovRobMiss")`: calculate summary information

plot `signature(object = "CovRobMiss", cutoff = "numeric")`: plot the object. See `plot`

getDist `signature(object = "CovRobMiss")`: return the squared partial Mahalanobis distances

getDistAdj `signature(object = "CovRobMiss")`: return the adjusted squared partial Mahalanobis distances

getDim `signature(object = "CovRobMiss")`: return the dimension of observed entries for each case

getLocation `signature(object = "CovRobMiss")`: return the estimated location vector

getScatter `signature(object = "CovRobMiss", cutoff = "numeric")`: return the estimated scatter matrix

getMissing `signature(object = "CovRobMiss")`: return the case number(s) with completely missing data, if any

getOutliers `signature(object = "CovRobMiss", cutoff = "numeric")`: return the case number(s) adjusted squared distances above `(1 - cutoff)`th quantile of chi-square p-degrees of freedom.

In addition to above, the following methods are defined with the class "CovRobMissSc":

getScale `signature(object = "CovRobMissSc")`: return the MVE scale of the best candidate

In addition to above, the following methods are defined with the class "emve":

getCandidates `signature(object = "emve")`: return a list of the top 10 candidates of EMVE estimates: `cand.sc` (a vector of EMVE scales), `cand.mu` (a matrix of EMVE location), and `cand.S` (an array of EMVE scatter). See `emve` for details.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

See Also

emve, CovRobMissSc-class, CovRobMiss-class

---

geochem *Geochemical Data*

Description

Geochemical data analyzed by Smith et al (1984). The variables in the data measures the contents (in parts per million) for 20 chemical elements (e.g., Copper and Zinc) in 53 samples of rocks in Western Australia.

Usage

```
data(geochem)
```

Format

The data contains 53 observations on 20 variables corresponding to the 20 chemical elements.

References

Smith, R.E., Campbell, N.A., Licheld, A. (1984). Multivariate statistical techniques applied to pisolitic laterite geochemistry at Golden Grove, Western Australia. *Journal of Geochemical Exploration*, **22**, 193–216.

Agostinelli, C., Leung, A. , Yohai, V.J., and Zamar, R.H. (2014) Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. arXiv:1406.6031[math.ST]

Examples

```
## Not run:
library(ICSNP)
library(rrcov)

data(geochem)
n <- nrow(geochem)
p <- ncol(geochem)

# MLE
res.ML <- list(mu=colMeans(geochem), S=cov(geochem))

# Tyler's M
geochem.med <- apply(geochem,2,median,na.rm=TRUE)
res.Tyler <- tyler.shape(geochem, location=geochem.med)
```

```
res.Tyler <- res.Tyler*(median(mahalanobis( geochem, geochem.med, res.Tyler))/qchisq(0.5, df=p) )
res.Tyler <- list(mu=geochem.med, S=res.Tyler)

# Rocke's Covariace
res.Rock <- CovSest(geochem, method="rocke")
res.Rock <- list(mu=res.Rock@center, S=res.Rock@cov)

# Fast-MCD
res.FMCD <- CovMcd( geochem)
res.FMCD <- list(mu=res.FMCD@center, S=res.FMCD@cov)

# MVE
res.MVE <- CovMve( geochem)
res.MVE <- list(mu=res.MVE@center, S=res.MVE@cov)

# S-estimator with bisquare rho function
res.S <- CovSest(geochem, method="bisquare")
res.S <- list(mu=res.S@center, S=res.S@cov)

# Fast-S
res.FS <- CovSest(geochem)
res.FS <- list(mu=res.FS@center, S=res.FS@cov)

# 2SGS
res.2SGS <- TSGS( geochem )
res.2SGS <- list(mu=res.2SGS@mu, S=res.2SGS@S)

# Combine all the results
geochem.res <- list(ML=res.ML, Tyler=res.Tyler, Rocke=res.Rock, MCD=res.FMCD,
MVE=res.MVE, FS=res.FS, MVES=res.S, TSGS=res.2SGS)

## Compare LRT distances between different estimators
res.tab <- data.frame(  LRT.to.2SGS=c(slrt( res.ML$S, res.2SGS$S),
slrt( res.Tyler$S, res.2SGS$S),
slrt( res.Rock$S, res.2SGS$S),
slrt( res.FMCD$S, res.2SGS$S),
slrt( res.MVE$S, res.2SGS$S),
slrt( res.FS$S, res.2SGS$S),
slrt( res.S$S, res.2SGS$S),
slrt( res.2SGS$S, res.2SGS$S) ))
row.names(res.tab) <- c("ML","Tyler","Rocke","MCD","MVE","FS","MVES","TSGS")

# Calculate proportion of outliers cellwise
pairwise.mahalanobis <- function(x, mu, S){
# function that computes pairwise mahalanobis distances
p <- ncol(x)
pairs.md <- c()
for(i in 1:(p-1)) for(j in (i+1):p)
pairs.md <- c(pairs.md, mahalanobis( x[,c(i,j)], mu[c(i,j)], S[c(i,j),c(i,j)]))
pairs.md
}
res.tab$Full <- res.tab$Pairs <- res.tab$Cell <- NA
for(i in names(geochem.res) ){
```

```
## Identify cellwise outliers
uni.dist <- sweep(sweep(geochem, 2, geochem.res[[i]]$mu, "-"), 2,
sqrt(diag(geochem.res[[i]]$S)), "/")^2
uni.dist.stat <- mean(uni.dist > qchisq(.99^(1/(n*p)), 1))
res.tab$Cell[ which( row.names(res.tab) == i)] <- round(uni.dist.stat,3)

## Identify pairwise outliers
pair.dist <- pairwise.mahalanobis( geochem, geochem.res[[i]]$mu, geochem.res[[i]]$S)
pair.dist.stat <- mean(pair.dist > qchisq(0.99^(1/(n*choose(p,2))), 2))
res.tab$Pairs[ which( row.names(res.tab) == i)] <- round(pair.dist.stat,3)

## Identify any large global MD
full.dist <- mahalanobis( geochem, geochem.res[[i]]$mu, geochem.res[[i]]$S)
full.dist.stat <- mean(full.dist > qchisq(0.99^(1/n), p))
res.tab$Full[ which( row.names(res.tab) == i)] <- round(full.dist.stat,3)
}
res.tab

## End(Not run)
```

`get-methods` *Accessor methods to the essential slots of classes CovRobMiss, TSGS, GSE, emve, and HuberPairwise*

Description

Accessor methods to the slots of objects of classes `CovRobMiss`, `TSGS`, `GSE`, `emve`, and `HuberPairwise`

Usage

```
getLocation(object)
getScatter(object)
getDist(object)
getDistAdj(object)
getDim(object)
getMissing(object)
getOutliers(object, cutoff)
getScale(object)
getCandidates(object)
getInitial(object)
getFiltDat(object)
```

Arguments

`object`	an object of any of the following classes `CovRobMiss-class`, `GSE-class`, `emve-class`, and `HuberPairwise-class`. For function `getScale`, only `GSE-class` objects are allowed. For function `getCandidates`, only `emve-class` objects are allowed.
`cutoff`	optional argument for `getOutliers` - quantiles of chi-square to be used as a threshold for outliers detection, defaults to 0.99

Details

getLocation `signature(object = "CovRobMiss")`: return the estimated location vector

getScatter `signature(object = "CovRobMiss", cutoff = "numeric")`: return the estimated scatter matrix

getDist `signature(object = "CovRobMiss")`: return the squared partial Mahalanobis distances

getDistAdj `signature(object = "CovRobMiss")`: return the adjusted squared partial Mahalanobis distances

getDim `signature(object = "CovRobMiss")`: return the dimension of observed entries for each case

getMissing `signature(object = "CovRobMiss")`: return the case number with completely missing data, if any

getOutliers `signature(object = "CovRobMiss", cutoff = "numeric")`: return the case number(s) adjusted squared distances above `(1 - cutoff)`th quantile of chi-square p-degrees of freedom.

getScale `signature(object = "CovRobMissSc")`: return either the estimated generalized S-scale or MVE-scale. See GSE and emve for details.

getCandidates `signature(object = "emve")`: return a list of the top 10 candidates of EMVE estimates. The list includes `cand.mve.scale` (a vector of EMVE scales), `cand.mu` (a matrix of EMVE location), and `cand.Sigma` (an array of EMVE scatter). See see emve for details.

getInitial `signature(object = "GSE")`: return a list of estimated initials with `mu0` as the initial location and `S0` as the initial covariance matrix.

getFiltDat `signature(object = "TSGS")`: return filtered data matrix from the first step of 2SGS.

Examples

```
## Not run:
data(boston)
res <- GSE(boston)

## extract estimated location
getLocation(res)

## extract estimated scatter
getScatter(res)

## extract estimated adjusted distances
getDistAdj(res)

## extract outliers
getOutliers(res)

## End(Not run)
```

GSE *Generalized S-Estimator in the presence of missing data*

Description

Computes the Generalized S-Estimate (GSE) – a robust estimate of location and scatter for data with contamination and missingness.

Usage

```
GSE(x, tol=1e-5, maxiter=500, init="emve", tol.scale=1e-4, miter.scale=30,
print.step=1, mu0, S0, ...)
```

Arguments

x	a matrix or data frame. May contain missing values, but cannot contain columns with completely missing entries.
tol	tolerance for the convergence criterion. Default is 1e-5.
maxiter	maximum number of iterations for the GSE algorithm. Default is 500.
init	type of initial estimator. Currently this can either be "emve" (emve), "qc", or "huber" (see emve and HuberPairwise). Default is "emve". If mu0 and S0 are provided, this argument is ignored.
tol.scale	tolerance for the computation of the GS-scale. Default is 1e-4.
miter.scale	maximum number of iterations for the computation the GS-scale. Default is 30.
print.step	this argument determines the level of printing which is done during the estimation process. The default value is 1 means that the number of iteration at convergence is printed out; a value of 0 means no printing occurs, except for error messages; and a value of 2 means that the relative change of scale and estimated GSE scale at each iteration is printed out.
mu0	optional vector of initial location estimate
S0	optional matrix of initial scatter estimate
...	optional arguments for computing the initial estimates (see emve, HuberPairwise).

Details

This function computes GSE as described in Danilov et al. (2012). The estimator requires a robust positive definite initial estimator. This initial estimator is required to "re-scale" the partial square mahalanobis distance for the different missing pattern, in which a single scale parameter is not enough. This function currently allows two main different initial estimators: EMVE (the default; see emve and Huberized Pairwise (see HuberPairwise). GSE using the latter estimator when the tuning constant $c0$ is 0 is referred to as QGSE in Danilov et al. (2012). Numerical results have shown that GSE with EMVE as initial has better performance (in both efficiency and robustness), but computing time can be longer.

Value

An S4 object of class GSE-class which is a subclass of the virtual class CovRobMissSc-class. The output S4 object contains the following slots:

mu	Estimated location. Can be accessed via getLocation.
S	Estimated scatter matrix. Can be accessed via getScatter.
sc	Generalized S-scale (GS-scale). Can be accessed via getScale.
pmd	Squared partial Mahalanobis distances. Can be accessed via getDist.
pmd.adj	Adjusted squared partial Mahalanobis distances. Can be accessed via getDistAdj.
pu	Dimension of the observed entries for each case. Can be accessed via getDim.
mu0	Estimated initial location. Can be accessed via getInitial.
S0	Estimated initial scatter matrix. Can be accessed via getInitial.
ximp	Input data matrix with missing values imputed using best linear predictor. Not meant to be accessed.
weights	Weights used in the estimation of the location. Not meant to be accessed.
weightsp	First derivative of the weights used in the estimation of the location. Not meant to be accessed.
iter	Number of iterations till convergence. Not meant to be accessed.
eps	relative change of the GS-scale at convergence. Not meant to be accessed.
call	Object of class "language". Not meant to be accessed.
x	Input data matrix. Not meant to be accessed.
p	Column dimension of input data matrix. Not meant to be accessed.
estimator	Character string of the name of the estimator used. Not meant to be accessed.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>, Ruben H. Zamar, Mike Danilov, Victor J. Yohai

References

Danilov, M., Yohai, V.J., Zamar, R.H. (2012). Robust Esimation of Multivariate Location and Scatter in the Presence of Missing Data. *Journal of the American Statistical Association* **107**, 1178–1186.

See Also

emve, HuberPairwise, GSE-class

Examples

```
## Not run:
set.seed(12345)
n <- 100; p <- 10
A <- matrix(0.9, p, p); diag(A) <- 1
x <- mvrnorm(n, rep(0,p), A)

## Introduce 10% contamination points
pcont <- 0.1; dcont <- 10
A.svd <- svd(A)
U <- A.svd$u %*% diag( sqrt(A.svd$d) ) %*% t(A.svd$v)
```

```
## Contaminated points in the direction corresponding to smallest eigenvalue
vv <- A.svd$u[,p]
uu <- 1/sqrt( t(vv) %*% solve(A, vv) )
vv <- vv*uu
ncont <- rbinom(n,1,pcont)
if( sum(ncont) > 0 )
x[which(ncont== 1),] <- dcont * matrix(vv,sum(ncont),p,byrow=T)

## Introduce 10% missingness
pmiss <- 0.1
nmiss <- matrix(rbinom(n*p,1,pmiss), n,p)
x[ which( nmiss == 1 ) ] <- NA

## Using EMVE as initial
res.emve <- GSE(x)
summary(res.emve) ## summary of the output
plot(res.emve) ## plot of the output
slrt( getScatter(res.emve), A) ## LRT distances to the true covariance

## Using QC as initial
res.qc <- GSE(x, init="qc")
summary(res.qc)
plot(res.qc)
slrt( getScatter(res.qc), A) ## in general performs worse than if EMVE used as initials

## End(Not run)
```

GSE-class *Generalized S-Estimator in the presence of missing data*

Description

Class of Generalized S-Estimator. It has the superclass of CovRobMissSc.

Objects from the Class

Objects can be created by calls of the form new("GSE", ...), but the best way of creating GSE objects is a call to the function GSE which serves as a constructor.

Slots

mu Estimated location. Can be accessed via getLocation.

S Estimated scatter matrix. Can be accessed via getScatter.

sc Generalized S-scale (GS-scale). Can be accessed via getScale.

pmd Square partial Mahalanobis distances. Can be accessed via getDist.

pmd.adj Adjusted square partial Mahalanobis distances. Can be accessed via getDistAdj.

pu Dimension of the observed entries for each case. Can be accessed via getDim.

`mu0` Estimated initial location. Can be accessed via `getInitial`.

`S0` Estimated initial scatter matrix. Can be accessed via `getInitial`.

`ximp` Input data matrix with missing values imputed using best linear predictor. Not meant to be accessed.

`weights` Weights used in the estimation of the location. Not meant to be accessed.

`weightsp` First derivative of the weights used in the estimation of the location. Not meant to be accessed.

`iter` Number of iterations till convergence. Not meant to be accessed.

`eps` relative change of the GS-scale at convergence. Not meant to be accessed.

`call` Object of class `"language"`. Not meant to be accessed.

`x` Input data matrix. Not meant to be accessed.

`p` Column dimension of input data matrix. Not meant to be accessed.

`estimator` Character string of the name of the estimator used. Not meant to be accessed.

Extends

Class `"CovRobMissSc"`, directly.

Methods

The following methods are defined with the superclass "CovRobMiss":

show `signature(object = "CovRobMiss")`: display the object

summary `signature(object = "CovRobMiss")`: calculate summary information

plot `signature(object = "CovRobMiss", cutoff = "numeric")`: plot the object. See `plot`

getDist `signature(object = "CovRobMiss")`: return the squared partial Mahalanobis distances

getDistAdj `signature(object = "CovRobMiss")`: return the adjusted squared partial Mahalanobis distances

getDim `signature(object = "CovRobMiss")`: return the dimension of observed entries for each case

getLocation `signature(object = "CovRobMiss")`: return the estimated location vector

getScatter `signature(object = "CovRobMiss", cutoff = "numeric")`: return the estimated scatter matrix

getMissing `signature(object = "CovRobMiss")`: return the case number(s) with completely missing data, if any

getOutliers `signature(object = "CovRobMiss", cutoff = "numeric")`: return the case number(s) adjusted squared distances above `(1 - cutoff)`th quantile of chi-square p-degrees of freedom.

In addition to above, the following methods are defined with the class "CovRobMissSc":

getScale `signature(object = "CovRobMissSc")`: return the GS scale

In addition to above, the following methods are defined with the class "GSE":

getInitial `signature(object = "GSE")`: return a list of 'mu0' and 'S0', the initial estimates

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

See Also

GSE, CovRobMissSc-class, CovRobMiss-class

---

horse *Horse-colic data*

Description

This is a modified version of the original data set (taken from UCI repository, see reference), where only quantitative variables are considered. This data set is about horse diseases where the task is to determine if the lesion of the horse was surgical or not. It contains rows with completely missing values except for ID and must be removed by the users. They are kept mainly for pedagogical purposes.

Usage

```
data(horse)
```

Format

A data frame with 368 observations on the following 7 variables are quantitative and 1 categorical. The first variable is a numeric id.

Hospital_Number	numeric id, i.e. the case number assigned to the horse (may not be unique if the horse is tr
Rectal_temperature	rectal temperature in degree celcius
Pulse	the heart rate in beats per minute; normal rate is 30-40 for adults
Respiratory_rate	respiratory rate; normal rate is 8 to 10
Nasogastric_reflux_PH	scale is from 0 to 14 with 7 being neutral; normal values are in the 3 to 4 range
Packed_cell_volume	the number of red cells by volume in the blood; normal range is 30 to 50
Total_protein	normal values lie in the 6-7.5 (gms/dL) range
Abdomcentesis_total_protein	Values are in gms/dL
surgical_leison	was the problem (lesion) surgical?; 1 = yes, 2 = no

Source

The original data have been taken from the Journal of Statistics Education Databases at

- http://archive.ics.uci.edu/ml/datasets/Horse+Colic,

References

Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Examples

```
## Not run:
data(horse)
horse.cts <- horse[,-c(1,9)] ## remove the id and categorical variable
res <- GSE(horse.cts)
plot(res, which="dd", xtrans="log10",ytrans="log10")
getOutliers(res)

## End(Not run)
```

HuberPairwise *Quadrant Covariance and Huberized Pairwise Scatter*

Description

Computes the Quadrant Covariance (QC) or Huberized Pairwise Scatter as described in Alqallaf et al. (2002).

Usage

```
HuberPairwise( x, psi="huber", c0=1.345, computePmd=TRUE)
```

Arguments

x	a matrix or data frame. May contain missing values, but cannot contain columns with completely missing entries.
psi	loss function to be used in computing pairwise scatter. Default is "huber". If psi="sign", this yields QC. Other value includes "huber".
c0	tuning constant for the huber function. c0=0 would yield QC. Default is c0=1.345. This parameter is unnecessary if psi='sign'.
computePmd	logical indicating whether to compute partial Mahalanobis distances (pmd) and adjusted pmd. Default is TRUE.

Details

As described in Alqallaf et al. (2002), this estimator requires a robust scale estimate and a location M-estimate, which will be used to transform the data through a loss-function to be outlier-free. Currently, this function takes MADN (normalized MAD) and median as the robust scale and location estimate to save computation time. By default, the loss function psi is a sign function, but users are encouraged to also try Huberized scatter with the loss function as $\psi_c(x) = min(max(-c, x), c), c > 0, c = 1.345$. The function does not adjust for intrinsic bias as described in Alqallaf et al. (2002). Missing values will be replaced by the corresponding column's median.

Value

An S4 object of class HuberPairwise-class which is a subclass of the virtual class CovRobMiss-class. The output S4 object contains the following slots:

mu	Estimated location. Can be accessed via getLocation.
S	Estimated scatter matrix. Can be accessed via getScatter.
pmd	Squared partial Mahalanobis distances. Can be accessed via getDist.
pmd.adj	Adjusted squared partial Mahalanobis distances. Can be accessed via getDistAdj.
pu	Dimension of the observed entries for each case. Can be accessed via getDim.
R	Estimated correlation matrix. Not meant to be accessed.
call	Object of class "language". Not meant to be accessed.
x	Input data matrix. Not meant to be accessed.
p	Column dimension of input data matrix. Not meant to be accessed.
estimator	Character string of the name of the estimator used. Not meant to be accessed.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

References

Alqallaf, F.A., Konis, K. P., R. Martin, D., Zamar, R. H. (2002). Scalable Robust Covariance and Correlation Estimates for Data Mining. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton.

HuberPairwise-class *Quadrant Covariance and Huberized Pairwise Scatter*

Description

Class of Quadrant Covariance and Huberized Pairwise Scatter. It has the superclass of CovRobMiss.

Objects from the Class

Objects can be created by calls of the form new("HuberPairwise", ...), but the best way of creating HuberPairwise objects is a call to the function HuberPairwise which serves as a constructor.

Slots

mu Estimated location. Can be accessed via getLocation.

S Estimated scatter matrix. Can be accessed via getScatter.

pmd Squared partial Mahalanobis distances. Can be accessed via getDist.

pmd.adj Adjusted squared partial Mahalanobis distances. Can be accessed via getDistAdj.

pu Dimension of the observed entries for each case. Can be accessed via getDim.

R Estimated correlation matrix. Not meant to be accessed.

call Object of class "language". Not meant to be accessed.

x Input data matrix. Not meant to be accessed.

p Column dimension of input data matrix. Not meant to be accessed.

estimator Character string of the name of the estimator used. Not meant to be accessed.

Extends

Class "CovRobMiss", directly.

Methods

No methods defined with class "HuberPairwise" in the signature.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

See Also

HuberPairwise, CovRobMiss-class

`partial.mahalanobis` *Partial Square Mahalanobis Distance*

Description

Computes the partial square Mahalanobis distance for all observations in x. Let $\mathbf{x} = (x_{i1}, ..., x_{ip})'$ be a p-dimensional random vector and $\mathbf{u} = (u_{i1}, ..., u_{ip})'$ be a p-dimensional vectors of zeros and ones indicating which entry is missing: 0 as missing and 1 as observed. Then partial mahalanobis distance is given by:

$d(\mathbf{x}, \mathbf{u}, \mathbf{m}, \mathbf{\Sigma}) = (\mathbf{x^{(u)}} - \mathbf{m^{(u)}})'(\mathbf{\Sigma^{(u)}})^{-1}(\mathbf{x^{(u)}} - \mathbf{m^{(u)}})$.

With no missing data, this function is equivalent to mahalanobis distance.

Usage

```
partial.mahalanobis(x, mu, Sigma)
```

Arguments

x	a matrix or data frame. May contain missing values, but cannot contain columns with completely missing entries.
mu	location estimate
Sigma	scatter estimate. Must be positive definite

Value

An S4 object of class CovRobMiss-class. The output S4 object contains the following slots:

mu	Estimated location. Can be accessed via getLocation.
S	Estimated scatter matrix. Can be accessed via getScatter.
pmd	Squared partial Mahalanobis distances. Can be accessed via getDist.
pmd.adj	Adjusted squared partial Mahalanobis distances. Can be accessed via getDistAdj.
pu	Dimension of the observed entries for each case. Can be accessed via getDim.
call	Object of class "language". Not meant to be accessed.
x	Input data matrix. Not meant to be accessed.
p	Column dimension of input data matrix. Not meant to be accessed.
estimator	Character string of the name of the estimator used. Not meant to be accessed.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

Examples

```
## Not run:
## suppose we would like to compute pmd for an MLE
x <- matrix(rnorm(1000),100,10)
U <- matrix(rbinom(1000,1,0.1),100,10)
x <- x * ifelse(U==1,NA,1)
## compute MLE (i.e. EM in this case)
res <- CovEM(x)
## compute pmd
res.pmd <- partial.mahalanobis(x, mu=getLocation(res), S=getScatter(res))
summary(res.pmd)
plot(res.pmd, which="index")

## End(Not run)
```

---

plot-methods *Plot methods for objects of class 'CovRobMiss'*

Description

Plot methods for objects of class 'CovRobMiss'. The following plots are available:

- chi-square qqplot for adjusted square partial Mahalanobis distances

- index plot for adjusted square partial Mahalanobis distances

- distance-distance plot comparing the adjusted distances based on classical MLE and robust estimators

Cases with completely missing data will be dropped out. Outliers are identified using some pre-specific cutoff value, for instance 99% quantile of chi-square with p degrees of freedom, where p is the column dimension of the data. Identified outliers can also be retrieved using getOutliers with an optional argument of cutoff, ranged from 0 to 1.

Usage

```
## S4 method for signature 'CovRobMiss'
plot(x, which = c("all","distance","qqchi2", "dd"),
which=c("all", "distance", "qqchisq", "dd"),
ask = (which=="all" && dev.interactive(TRUE)),
cutoff, ...)
```

Arguments

x	an object of class "CovRobMiss"
which	Which plot to show? Default is which="all".
ask	logical; if 'TRUE', the user is *ask*ed before each plot, see 'par(ask=.)'. Default is ask = which=="all" && dev.interactive().
cutoff	The cutoff value for the distances.
...	Additional arguments to be passed over to control the coordinates. See coord_trans.

Examples

```
## Not run:
data(boston)
res <- GSE(boston)

## plot all graphs
plot(res)

## plot individuals plots
plot(res, which="qqchisq")
plot(res, which="distance")
plot(res, which="dd")

## control the coordinates, e.g. log10 transform the y-axis
plot(res, which="qqchisq", ytrans="log10", xtrans="log10")
plot(res, which="distance", ytrans="log10")
plot(res, which="dd", ytrans="log10", xtrans="log10")

## End(Not run)
```

---

simulation-tools *Data generator for simulation study on cell- and case-wise contamination*

Description

Includes the data generator for the simulation study on cell- and case-wise contamination that appears on Agostinelli et al. (2014).

Usage

```
generate.randcorr(cond, p, tol=1e-5, maxits=100)

generate.cellcontam(n, p, cond, contam.size, contam.prop)

generate.casecontam(n, p, cond, contam.size, contam.prop)
```

Arguments

cond	desired condition number of the random correlation matrix. The correlation matrix will be used to generate multivariate normal samples in `generate.cellcontam` and `generate.cellcontam`.
tol	tolerance level for the condition number of the random correlation matrix. Default is `1e-5`.
maxits	integer indicating the maximum number of iterations until the condition number of the random correlation matrix is within a tolerance level. Default is 100.
n	integer indicating the number of observations to be generated.
p	integer indicating the number of variables to be generated.
contam.size	size of cell- or case-wise contamination. For cell-wise outliers, random cells in a data matrix are replaced by `contam.dist`. For case-wise outliers, random cases in a data matrix are replaced by `contam.dist` times v where v
contam.prop	proportion of cell- or case-wise contamination.

Details

Details about how the correlation matrix is randomly generated and how the contaminated data is generated can be found in Agostinelli et al. (2014).

Value

`generate.randcorr` gives the random correlation matrix in dimension `p` and with condition number `cond`.

`generate.cellcontam` and `generate.casecontam` give the multivariate normal sample that is either cell-wise or case-wise contaminated as described in Agostinelli et al. (2014). The contaminated sample is returned as components of a list with components

- `x` multivariate normal sample with cell- or case-wise contamination.
- `u` `n` by `p` matrix of 0's and 1's with 1's correspond to an outlier. A row of 1's correspond to a case-wise outlier.
- `A` random correlation matrix with a specified condition number.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>, Claudio Agostinelli, Ruben H. Zamar, Victor J. Yohai

References

Agostinelli, C., Leung, A. , Yohai, V.J., and Zamar, R.H. (2014) Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. arXiv:1406.6031[math.ST]

See Also

TSGS

`slrt` *LRT-based distances between matrices*

Description

LRT-distance that we use to evaluate the performance of our covariance estimates.

Usage

```
slrt(S, trueS)
```

Arguments

`S`	estimated covariance matrix
`trueS`	true covariance matrix.

Details

Note that this is not actually a distance in a sense that slrt(M1,M2) != slrt(M2,M1)

Value

scalar LRT-distance

Author(s)

Mike Danilov

References

Seber, G.A. (2004) Multivariate observations, Wiley

Danilov, M. (2010). Robust Estimation of Multivariate Scatter under Non-Affine Equivariant Scenarios. Ph.D. thesis, Department of Statistics, University of British Columbia.

SummaryCov-class *Class "SummaryCov" - displaying summary of "CovRobMiss" objects*

Description

Displays summary information for CovRobMiss-class objects

Objects from the Class

Objects can be created by calls of the form new("SummaryCov", ...).

Slots

obj: CovRobMiss-class object

evals: Eigenvalues and eigenvectors of the covariance or correlation matrix

Methods

show signature(object = "SummaryCov"): display the object

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

TSGS *Two-Step Generalized S-Estimator for cell- and case-wise outliers*

Description

Computes the Two-Step Generalized S-Estimate (2SGS) – a robust estimate of location and scatter for data with cell-wise and case-wise contamination.

Usage

```
TSGS(x, alpha=0.99, it=TRUE, ...)

.gy.filt(x, alpha, it=TRUE)
```

Arguments

x	a matrix or data frame.
alpha	quantile of the reference distribution used in tail comparison in the first step. A standard normal is used as the reference distribution. Default value is 0.99.
it	logical, whether the filtering is repeated until no additional cell-wise outliers are identified in the first step. Default value is TRUE.
...	optional arguments to be used in the computation of GSE in the second step (see GSE).

Details

This function computes 2SGS as described in Agostinelli et al. (2014). The procedure has two steps:

In Step I, the method filters (i.e., flags and removes) cell-wise outliers using Gervini-Yohai approach. Outliers are flagged and replaced by missing values (NA) when the empirical tail distribution is heavier than a reference distribution. A standard normal is currently used as a reference distribution as suggested in Agostinelli et al. (2014). The filtering step can be called on its own by using the function `GSE:::.gy.filt(x, alpha, it=FALSE)`.

In Step II, the method applies GSE (see GSE), which has been specifically designed to deal with incomplete multivariate data with case-wise outliers, to the filted data coming from Step I.

The application to the Chemical data set analyzed in Agostinelli et al. (2014) can be found in geochem.

The tools that were used to generate contaminated data in the simulation study in Agostinelli et al. (2014) can be found in generate.cellcontam and generate.casecontam.

Value

The following gives the major slots in the output S4 object:

mu	Estimated location. Can be accessed via getLocation.
S	Estimated scatter matrix. Can be accessed via getScatter.
xf	Filtered data matrix from the first step of 2SGS. Can be accessed via getFiltDat.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>, Claudio Agostinelli, Ruben H. Zamar, Victor J. Yohai

References

Agostinelli, C., Leung, A. , Yohai, V.J., and Zamar, R.H. (2014) Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. arXiv:1406.6031[math.ST]

See Also

GSE, generate.cellcontam, generate.casecontam

Examples

```
## Not run:
set.seed(12345)

# Generate 5% cell-wise contaminated normal data
# using a random correlation matrix with condition number 100
x <- generate.cellcontam(n=100, p=10, cond=100, contam.size=5, contam.prop=0.05)

## Using MLE
slrt( cov(x$x), x$A)
```

```
## Using Fast-S
slrt( rrcov:::CovSest(x$x)@cov, x$A)

## Using 2SGS
slrt( GSE:::TSGS(x$x)@S, x$A)


# Generate 5% case-wise contaminated normal data
# using a random correlation matrix with condition number 100
x <- generate.casecontam(n=100, p=10, cond=100, contam.size=15, contam.prop=0.05)

## Using MLE
slrt( cov(x$x), x$A)

## Using Fast-S
slrt( rrcov:::CovSest(x$x)@cov, x$A)

## Using 2SGS
slrt( GSE:::TSGS(x$x)@S, x$A)


## End(Not run)
```

`TSGS-class` *Two-Step Generalized S-Estimator for cell- and case-wise outliers*

Description

Class of Two-Step Generalized S-Estimator. It has the superclass of `GSE`.

Objects from the Class

Objects can be created by calls of the form `new("TSGS", ...)`, but the best way of creating `TSGS` objects is a call to the function `TSGS` which serves as a constructor.

Slots

`mu` Estimated location. Can be accessed via `getLocation`.

`S` Estimated scatter matrix. Can be accessed via `getScatter`.

`xf` Filtered data matrix from the first step of 2SGS. Can be accessed via `getFiltDat`.

Extends

Class `"GSE"`, directly.

Methods

In addition to the methods defined in the superclass "GSE", the following methods are also defined:

getFiltDat `signature(object = "TSGS")`: return the filtered data matrix.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>

See Also

TSGS, GSE, GSE-class

`wages` *Wages and Hours*

Description

The data are from a national sample of 6000 households with a male head earning less than USD 15,000 annually in 1966. The data were clasified into 39 demographic groups for analysis. The study was undertaken in the context of proposals for a guaranteed annual wage (negative income tax). At issue was the response of labor supply (average hours) to increasing hourly wages. The study was undertaken to estimate this response from available data.

Usage

```
data(wages)
```

Format

A data frame with 39 observations on the following 10 variables:

HRS	Average hours worked during the year
RATE	Average hourly wage (USD)
ERSP	Average yearly earnings of spouse (USD)
ERNO	Average yearly earnings of other family members (USD)
NEIN	Average yearly non-earned income
ASSET	Average family asset holdings (Bank account, etc.) (USD)
AGE	Average age of respondent
DEP	Average number of dependents
RACE	Percent of white respondents
SCHOOL	Average highest grade of school completed

Source

DASL library http://lib.stat.cmu.edu/DASL/Datafiles/wagesdat.html

References

D.H. Greenberg and M. Kosters, (1970). Income Guarantees and the Working Poor, The Rand Corporation.

Index