

# Package ‘GSE’

January 20, 2025

**Type** Package

**Title** Robust Estimation in the Presence of Cellwise and Casewise Contamination and Missing Data

**Version** 4.2-1

**Date** 2022-12-13

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

**Maintainer** Claudio Agostinelli <claudio.agostinelli@unitn.it>

**Description** Robust Estimation of Multivariate Location and Scatter in the Presence of Cellwise and Casewise Contamination and Missing Data.

**License** GPL (>= 2)

**Depends** R (>= 3.1.0), Rcpp (>= 0.10.0), MASS

**Imports** rrcov, robustbase, cellWise, ggplot2, methods

**LinkingTo** Rcpp, RcppArmadillo

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2022-12-13 09:20:02 UTC

## Contents

auto . . . . .	2
boston . . . . .	3
calcium . . . . .	4
CovEM . . . . .	6
CovRobMiss-class . . . . .	7
CovRobMissSc-class . . . . .	8
emve . . . . .	9
emve-class . . . . .	11
geochem . . . . .	12
get-methods . . . . .	14
GSE . . . . .	16
GSE-class . . . . .	18

gy.filt . . . . .	20
horse . . . . .	21
HuberPairwise . . . . .	22
HuberPairwise-class . . . . .	24
ImpS . . . . .	25
partial.mahalanobis . . . . .	26
plot-methods . . . . .	27
simulation-tools . . . . .	28
slrt . . . . .	30
SummaryCovGSE-class . . . . .	30
TSGS . . . . .	31
TSGS-class . . . . .	33
wages . . . . .	34

<b>Index</b>	<b>36</b>
--------------	-----------

---

auto	<i>Automobile data</i>
------	------------------------

---

### 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:

<code>cmdev</code>	corrected median value of owner-occupied homes in USD 1000's
<code>crim</code>	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/~mllearn/MLRepository.html>,

the corrected data have been taken from Statlib at

- <https://dasl.datadescription.com/> (originally downloaded from lib.stat.cmu.edu/DASL/)

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/~mllearn/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://jse.amstat.org/datasets/calcium.dat.txt> (originally downloaded from [www.amstat.org/publications/jse](http://www.amstat.org/publications/jse))

the corrected data have been taken from Statlib at

- <http://jse.amstat.org/datasets/calciumgood.dat.txt> (originally downloaded from [www.amstat.org/publications/jse/datasets/calciumgood.dat.txt](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)
```

**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.

**Value**

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

mu	Estimated location. Can be accessed via <code>getLocation</code> .
S	Estimated scatter matrix. Can be accessed via <code>getScatter</code> .
pmd	Squared partial Mahalanobis distances. Can be accessed via <code>getDist</code> .
pmd.adj	Adjusted squared partial Mahalanobis distances. Can be accessed via <code>getDistAdj</code> .
pu	Dimension of the observed entries for each case. Can be accessed via <code>getDim</code> .
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	<i>Class "CovRobMiss" – a superclass for the robust estimates of location and scatter for missing data</i>
------------------	--

---

## 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 following 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	<i>Class "CovRobMissSc" – a subclass of "CovRobMiss" with scale estimate</i>
--------------------	--

---

### 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 following 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 inherited from 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	<i>Extended Minimum Volume Ellipsoid (EMVE) in the presence of missing data</i>
------	---

---

**Description**

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

**Usage**

```
emve(x, maxits=5, sampling=c("uniform", "cluster"), n.resample, n.sub.size, seed)
```

**Arguments**

<code>x</code>	a matrix or data frame. May contain missing values, but cannot contain columns with completely missing entries.
<code>maxits</code>	integer indicating the maximum number of iterations of Gaussian MLE calculation for each subsample. Default is 5.
<code>sampling</code>	which sampling scheme is to use: 'uniform' or 'cluster' (see Leung and Zamar, 2016). Default is 'uniform'.
<code>n.resample</code>	integer indicating the number of subsamples. Default is 15 for clustering-based subsampling and 500 for uniform subsampling.
<code>n.sub.size</code>	integer indicating the sizes of each subsample. Default is $2(p+1)/a$ for clustering-based subsampling and $(p+1)/a$ for uniform subsampling, where $a$ is proportion of non-missing cells.
<code>seed</code>	optional starting value for random generator. Default is <code>seed = 1000</code> .

## Details

This function computes EMVE as described in Danilov et al. (2012). Two subsampling schemes can be used for computing EMVE: uniform subsampling and the clustering-based subsampling as described in Leung and Zamar (2016). For uniform subsampling, the number of subsamples must be large to ensure high breakdown point. For clustering-based subsampling, the number of subsamples can be smaller. The subsample size  $n_0$  must be chosen to be larger than  $p$  to avoid singularity.

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:

<code>mu</code>	Estimated location. Can be accessed via <code>getLocation</code> .
<code>S</code>	Estimated scatter matrix. Can be accessed via <code>getScatter</code> .
<code>sc</code>	Estimated EMVE scale. Can be accessed via <code>getScale</code> .
<code>pmd</code>	Squared partial Mahalanobis distances. Can be accessed via <code>getDist</code> .
<code>pmd.adj</code>	Adjusted squared partial Mahalanobis distances. Can be accessed via <code>getDistAdj</code> .
<code>pu</code>	Dimension of the observed entries for each case. Can be accessed via <code>getDim</code> .
<code>call</code>	Object of class "language". Not meant to be accessed.
<code>x</code>	Input data matrix. Not meant to be accessed.
<code>p</code>	Column dimension of input data matrix. Not meant to be accessed.
<code>estimator</code>	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 Estimation of Multivariate Location and Scatter in the Presence of Missing Data. *Journal of the American Statistical Association* **107**, 1178–1186.
- Leung, A. and Zamar, R.H. (2016). Multivariate Location and Scatter Matrix Estimation Under Cellwise and Casewise Contamination. Submitted.
- Rubin, D.B. and Little, R.J.A. (2002). *Statistical analysis with missing data* (2nd ed.). New York: Wiley.

## See Also

[GSE](#), [emve-class](#)

---

emve-class	<i>Extended Minimum Volume Ellipsoid (EMVE) in the presence of missing data.</i>
------------	--

---

## 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`.

`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  $(1 - \text{cutoff})$ 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

### 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)

# Roche's Covariance
res.Rock <- CovSest(geochem, method="roche")
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, seed=999 )
res.2SGS <- list(mu=res.2SGS@mu, S=res.2SGS@S)

# Combine all the results
geochem.res <- list(ML=res.ML, Tyler=res.Tyler, Roche=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$$, res.2SGS$$),
slrt( res.S$$, res.2SGS$$),
slrt( res.2SGS$$, res.2SGS$$ ))
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(obj)
getFiltDat(object)

```

### Arguments

**obj, object** an object of any of the following classes [CovRobMiss-class](#), [GSE-class](#), [emve-class](#), and [HuberPairwise-class](#). For function `getScale` the package defines a method for objects of class [GSE-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  $(1 - \text{cutoff})$ 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.

**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-4, maxiter=150, method=c("bisquare", "rocke"),
    init=c("emve", "qc", "huber", "imputed", "emve_c"), mu0, S0, ...)
```

## Arguments

<code>x</code>	a matrix or data frame. May contain missing values, but cannot contain columns with completely missing entries.
<code>tol</code>	tolerance for the convergence criterion. Default is 1e-4.
<code>maxiter</code>	maximum number of iterations for the GSE algorithm. Default is 150.
<code>method</code>	which loss function to use: 'bisquare', 'rocke'.
<code>init</code>	type of initial estimator. Currently this can either be "emve" (EMVE with uniform sampling, see Danilov et al., 2012), "qc" (QC, see Danilov et al., 2012), "huber" (Huber Pairwise, see Danilov et al., 2012), "imputed" (Imputed S-estimator, see the rejoinder in Agostinelli et al., 2015), or "emve_c" (EMVE_C with cluster sampling, see Leung and Zamar, 2016). Default is "emve". If $\mu_0$ and $S_0$ are provided, this argument is ignored.
<code>mu0</code>	optional vector of initial location estimate
<code>S0</code>	optional matrix of initial scatter estimate
<code>...</code>	optional arguments for computing the initial estimates (see <a href="#">emve</a> , <a href="#">HuberPairwise</a> ).

## Details

This function computes GSE (Danilov et al., 2012) and GRE (Leung and Zamar, 2016). 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 initial estimators: EMVE (the default; see [emve](#) and Huberized Pairwise (see [HuberPairwise](#))). GSE using Huberized Pairwise with sign psi function 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 <a href="#">getLocation</a> .
S	Estimated scatter matrix. Can be accessed via <a href="#">getScatter</a> .
sc	Generalized S-scale (GS-scale). Can be accessed via <a href="#">getScale</a> .
pmd	Squared partial Mahalanobis distances. Can be accessed via <a href="#">getDist</a> .
pmd.adj	Adjusted squared partial Mahalanobis distances. Can be accessed via <a href="#">getDistAdj</a> .
pu	Dimension of the observed entries for each case. Can be accessed via <a href="#">getDim</a> .
mu0	Estimated initial location.
S0	Estimated initial scatter matrix.
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](mailto:andy.leung@stat.ubc.ca)>, Ruben H. Zamar, Mike Danilov, Victor J. Yohai

**References**

- Agostinelli, C., Leung, A. , Yohai, V.J., and Zamar, R.H. (2015) Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. *TEST*.
- Danilov, M., Yohai, V.J., Zamar, R.H. (2012). Robust Estimation of Multivariate Location and Scatter in the Presence of Missing Data. *Journal of the American Statistical Association* **107**, 1178–1186.
- Leung, A. and Zamar, R.H. (2016). Multivariate Location and Scatter Matrix Estimation Under Cellwise and Casewise Contamination. Submitted.

**See Also**

[emve](#), [HuberPairwise](#), [GSE-class](#), [generate.casecontam](#)

**Examples**

```
set.seed(12)

## generate 10-dimensional data with 10% casewise contamination
n <- 100
p <- 10
A <- matrix(0.9, p, p)
```

```
diag(A) <- 1
x <- generate.casecontam(n, p, cond=100, contam.size=10, contam.prop=0.1, A=A)$x

## introduce 5% missingness
pmiss <- 0.05
nmiss <- matrix(rbinom(n*p,1,pmiss), n,p)
x[ which( nmiss == 1 ) ] <- NA

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

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

---

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.

`S0` Estimated initial scatter matrix.

`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

### Author(s)

Andy Leung <[andy.leung@stat.ubc.ca](mailto:andy.leung@stat.ubc.ca)>

### See Also

[GSE](#), [CovRobMissSc-class](#), [CovRobMiss-class](#)

---

 gy.filt

*Gervini-Yohai filter for detecting cellwise outliers*


---

**Description**

Flags cellwise outliers detected using Gervini-Yohai filter as described in Agostinelli et al. (2015) and Leung and Zamar (2016).

**Usage**

```
gy.filt(x, alpha=c(0.95,0.85), bivarQt=0.99, bivarCellPr=0.1, miter=5)
```

**Arguments**

x	a matrix or data frame.
alpha	a vector of the quantiles of the univariate and bivariate reference distributions, respectively. Filtering turns off when alpha is 0. For univariate filtering only, $\alpha=c(0.95,0)$ . Default value is $c(0.95,0.85)$ .
bivarQt	quantile of the binomial model for the number of flagged pairs in the bivariate filter. Default is 0.99.
bivarCellPr	probability of the binomial model for the number of flagged pairs in the bivariate filter. Default is 0.1.
miter	maximum number of iteration of filtering. Default value is 5.

**Details**

This function implements the univariate filter and the univariate-plus-bivariate filter as described in Agostinelli et al. (2015) and Leung and Zamar (2016), respectively.

In the univariate filter, outliers are flagged by comparing the empirical tail distribution of each marginal with a reference (normal) distribution using Gervini-Yohai approach.

In the univariate-plus-bivariate filter, outliers are first flagged by applying the univariate filter. Then, the bivariate filter is applied to flag any additional outliers. In the bivariate filter, outliers are flagged by comparing the empirical tail distribution of each bivariate marginal with a reference (chi-square with 2 d.f.) distribution using Gervini-Yohai approach. The number of flagged pairs associated with each cell approximately follows a binomial model under Independent Cellwise Contamination Model. A cell is additionally flagged if the number of flagged pairs exceeds a large quantile of the binomial model.

**Value**

a matrix or data frame of the filtered data.

**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. (2015) Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. TEST.

Leung, A. and Zamar, R.H. (2016). Multivariate Location and Scatter Matrix Estimation Under Cellwise and Casewise Contamination. Submitted.

## See Also

[TSGS](#), [generate.cellcontam](#)

## Examples

```
set.seed(12345)

# Generate 5% cell-wise contaminated normal data
x <- generate.cellcontam(n=100, p=10, cond=100, contam.size=5, contam.prop=0.05)$x

## Using univariate filter only
xna <- gy.filt(x, alpha=c(0.95,0))
mean(is.na(xna))

## Using univariate-and-bivariate filter
xna <- gy.filt(x, alpha=c(0.95,0.95))
mean(is.na(xna))
```

---

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", xlog10=TRUE, ylog10=TRUE)
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=c("huber","sign"), 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".

<code>c0</code>	tuning constant for the huber function. <code>c0=0</code> would yield QC. Default is <code>c0=1.345</code> . This parameter is unnecessary if <code>psi='sign'</code> .
<code>computePmd</code>	logical indicating whether to compute partial Mahalanobis distances ( <code>pmd</code> ) and adjusted <code>pmd</code> . Default is <code>TRUE</code> .

## 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:

<code>mu</code>	Estimated location. Can be accessed via <code>getLocation</code> .
<code>S</code>	Estimated scatter matrix. Can be accessed via <code>getScatter</code> .
<code>pmd</code>	Squared partial Mahalanobis distances. Can be accessed via <code>getDist</code> .
<code>pmd.adj</code>	Adjusted squared partial Mahalanobis distances. Can be accessed via <code>getDistAdj</code> .
<code>pu</code>	Dimension of the observed entries for each case. Can be accessed via <code>getDim</code> .
<code>R</code>	Estimated correlation matrix. Not meant to be accessed.
<code>call</code>	Object of class "language". Not meant to be accessed.
<code>x</code>	Input data matrix. Not meant to be accessed.
<code>p</code>	Column dimension of input data matrix. Not meant to be accessed.
<code>estimator</code>	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](mailto:andy.leung@stat.ubc.ca)>

### See Also

[HuberPairwise](#), [CovRobMiss-class](#)



---

ImpS	<i>Imputed S-estimator</i>
------	----------------------------

---

**Description**

Computes the simple three-step estimator as described in the rejoinder of Agostinelli et al. (2015).

**Usage**

```
ImpS(x, alpha=0.95, method=c("bisquare","rocke"), init=c("emve","emve_c"), ...)
```

**Arguments**

x	a matrix or data frame.
alpha	quantile of the reference distribution in the univariate filter step (see <a href="#">gy.filt</a> ). Default is 0.95.
method	which loss function to use: 'bisquare', 'rocke'.
init	type of initial estimator. Currently this can either be "emve" (EMVE with uniform sampling, see Danilov et al., 2012) or "emve_c" (EMVE_C with cluster sampling, see Leung and Zamar, 2016). Default is "emve".
...	optional, additional arguments to be passed to <a href="#">GSE</a> .

**Details**

This function computes the simple three-step estimator as described in the rejoinder in Agostinelli et al. (2015). The procedure has three steps:

In Step I, the method flags and removes cell-wise outliers using the Gervini-Yohai univariate only filter (see [gy.filt](#)).

In Step II, the method imputes the filtered cells using coordinate-wise medians.

In Step III, the method applies MVE-S to the filtered and imputed data from Step II (see [GSE](#)).

**Value**

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

mu	Estimated location. Can be accessed via <a href="#">getLocation</a> .
S	Estimated scatter matrix. Can be accessed via <a href="#">getScatter</a> .
xf	Filtered data matrix from the first step of 2SGS. Can be accessed via <a href="#">getFiltDat</a> .

**Author(s)**

Andy Leung <[andy.leung@stat.ubc.ca](mailto: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. (2015) Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. TEST.

**See Also**

[GSE, gy.filt](#)

---

partial.mahalanobis    *Partial Square Mahalanobis Distance*

---

**Description**

Computes the partial square Mahalanobis distance for all observations in  $\mathbf{x}$ . Let  $\mathbf{x} = (x_{i1}, \dots, x_{ip})'$  be a  $p$ -dimensional random vector and  $\mathbf{u} = (u_{i1}, \dots, u_{ip})'$  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:

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

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

**Usage**

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

**Arguments**

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

**Value**

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

<code>mu</code>	Estimated location. Can be accessed via <a href="#">getLocation</a> .
<code>S</code>	Estimated scatter matrix. Can be accessed via <a href="#">getScatter</a> .
<code>pmd</code>	Squared partial Mahalanobis distances. Can be accessed via <a href="#">getDist</a> .
<code>pmd.adj</code>	Adjusted squared partial Mahalanobis distances. Can be accessed via <a href="#">getDistAdj</a> .
<code>pu</code>	Dimension of the observed entries for each case. Can be accessed via <a href="#">getDim</a> .
<code>call</code>	Object of class "language". Not meant to be accessed.
<code>x</code>	Input data matrix. Not meant to be accessed.
<code>p</code>	Column dimension of input data matrix. Not meant to be accessed.
<code>estimator</code>	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 = 0.99, xlog10 = FALSE, ylog10 = FALSE)
```

**Arguments**

x	an object of class "CovRobMiss"
which	Which plot to show? Default is which="all".
ask	logical; if 'TRUE', the user is <i>asked</i> before each plot, see 'par(ask=.)'. Default is ask = which=="all" && dev.interactive().
cutoff	The quantile cutoff for the distances. Default is 0.99.
xlog10	Base-10 logged x-axis? Default is FALSE.
ylog10	Base-10 logged y-axis? Default is FALSE.

**Examples**

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

## plot all graphs
plot(res)

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

## control the coordinates, e.g. log10 transform the y-axis
plot(res, which="qqchisq", xlog10=TRUE, ylog10=TRUE)
plot(res, which="index", ylog10=TRUE)
plot(res, which="dd", xlog10=TRUE, ylog10=TRUE)

## 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, A=NULL)

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

**Arguments**

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

**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	<i>LRT-based distances between matrices</i>
------	---

---

**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  $\text{slrt}(M1, M2) \neq \text{slrt}(M2, M1)$

**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.

---

SummaryCovGSE-class	<i>Class "SummaryCovGSE" - displaying summary of "CovRobMiss" objects</i>
---------------------	---

---

**Description**

Displays summary information for [CovRobMiss-class](#) objects

**Objects from the Class**

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

**Slots**

obj: `CovRobMiss-class` object

evals: Eigenvalues and eigenvectors of the covariance or correlation matrix

**Methods**

`show` signature(object = "SummaryCovGSE"): 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, filter=c("UBF-DDC", "UBF", "DDC", "UF"),
     partial.impute=FALSE, tol=1e-4, maxiter=150, method=c("bisquare", "rocke"),
     init=c("emve", "qc", "huber", "imputed", "emve_c"), mu0, S0)
```

**Arguments**

<code>x</code>	a matrix or data frame.
<code>filter</code>	the filter to be used in the first step (see Leung et al. (2016)). Default is 'UBF-DDC'. For all filters, the default parameters are used.
<code>partial.impute</code>	whether partial imputation is used prior to estimation (see details). The default is FALSE.
<code>tol</code>	tolerance for the convergence criterion. Default is 1e-4.
<code>maxiter</code>	maximum number of iterations for the GSE algorithm. Default is 150.
<code>method</code>	which loss function to use: 'bisquare', 'rocke'.
<code>init</code>	type of initial estimator. Currently this can either be "emve" (EMVE with uniform sampling, see Danilov et al., 2012), "qc" (QC, see Danilov et al., 2012), "huber" (Huber Pairwise, see Danilov et al., 2012), "imputed" (Imputed S-estimator, see the rejoinder in Agostinelli et al., 2015), or "emve_c" (EMVE_C with cluster sampling, see Leung and Zamar, 2016). Default is "emve". If $\mu_0$ and $S_0$ are provided, this argument is ignored.
<code><math>\mu_0</math></code>	optional vector of initial location estimate
<code><math>S_0</math></code>	optional matrix of initial scatter estimate

## Details

This function computes 2SGS as described in Agostinelli et al. (2015) and Leung and Zamar (2016). The procedure has two major steps:

In Step I, the method filters (i.e., flags and removes) cell-wise outliers using Gervini-Yohai univariate filter (Agostinelli et al., 2015) or univariate-bivariate filter (Leung et al., 2016) or univariate-bivariate-plus-DDC filter (Leung et al., 2016; Rousseeuw and Van den Bossche, 2016). The filtering step can be called on its own by using the function `gy.filt` or `DDC`.

In Step II, the method applies GSE or GRE (GSE with a Rocke-type loss function), which has been specifically designed to deal with incomplete multivariate data with case-wise outliers, to the filtered data coming from Step I. The second step can be called on its own by using the function `GSE`.

The 2SGS method is intended for continuous variables, and requires that the number of observations  $n$  be relatively larger than 5 times the number of variables  $p$  for desirable performance (see the rejoinder in Agostinelli et al., 2015). In our numerical studies, for  $n$  too small relative to  $p$ , 2SGS may experience a lack of convergence, especially for filtered data sets with a proportion of complete observations less than  $1/2 + (p+1)/n$ . To overcome this problem, partial imputation prior to estimation is proposed (see the rejoinder in Agostinelli et al., 2015). The procedure is rather ad hoc, but initial numerical experiments show that partial imputation may work. Further research on this topic is still needed. By default, partial imputation is not used, unless specified.

In general, we warn users to use 2SGS with caution for data set with  $n$  relatively smaller than 5 times  $p$ .

The application to the chemical data set analyzed in Agostinelli et al. (2015) can be found in `geochem`.

The tools that were used to generate contaminated data in the simulation study in Agostinelli et al. (2015) can be found in `generate.cellcontam` and `generate.casecontam`.

## Value

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

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

## 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. (2015) Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. TEST.

Leung, A., Yohai, V.J., Zamar, R.H. (2016). Multivariate Location and Scatter Matrix Estimation Under Cellwise and Casewise Contamination. arXiv:1609.00402.

Rousseeuw P.J., Van den Bossche W. (2016). Detecting deviating data cells. arXiv:1601.07251



**See Also**

[GSE](#), [gy.filt](#), [DDC](#), [generate.cellcontam](#), [generate.casecontam](#)

**Examples**

```
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( 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( TSGS(x$x)@S, x$A)
```

---

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](mailto: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 classified 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>), the dataset is not anymore available at this source.

**References**

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

# Index

- \* **classes**
  - CovRobMiss-class, 7
  - CovRobMissSc-class, 8
  - emve-class, 11
  - GSE-class, 18
  - HuberPairwise-class, 24
  - SummaryCovGSE-class, 30
  - TSGS-class, 33
- \* **datasets**
  - auto, 2
  - boston, 3
  - calcium, 4
  - geochem, 12
  - horse, 21
  - wages, 34
- \* **get**
  - get-methods, 14
- \* **methods**
  - get-methods, 14
  - plot-methods, 27
- auto, 2
- boston, 3
- calcium, 4
- CovEM, 6
- CovRobMiss, 8, 24
- CovRobMiss-class, 7
- CovRobMissSc, 11, 19
- CovRobMissSc-class, 8
- DDC, 32, 33
- emve, 7, 8, 9, 12, 15–17
- emve-class, 11
- generate.casecontam, 17, 32, 33
- generate.casecontam (simulation-tools), 28
- generate.cellcontam, 21, 32, 33
- generate.cellcontam (simulation-tools), 28
- generate.randcorr (simulation-tools), 28
- geochem, 12, 32
- get-methods, 14
- getDim, 6–8, 10, 11, 17, 18, 23, 24, 26
- getDim (get-methods), 14
- getDim, CovRobMiss-method (CovRobMiss-class), 7
- getDim-methods (get-methods), 14
- getDist, 6–8, 10, 11, 17, 18, 23, 24, 26
- getDist (get-methods), 14
- getDist, CovRobMiss-method (CovRobMiss-class), 7
- getDist-methods (get-methods), 14
- getDistAdj, 6–8, 10, 11, 17, 18, 23, 24, 26
- getDistAdj (get-methods), 14
- getDistAdj, CovRobMiss-method (CovRobMiss-class), 7
- getDistAdj-methods (get-methods), 14
- getFiltDat, 25, 32, 34
- getFiltDat (get-methods), 14
- getFiltDat, TSGS-method (TSGS-class), 33
- getFiltDat-methods (get-methods), 14
- getLocation, 6–8, 10, 11, 17, 18, 23–26, 32, 34
- getLocation (get-methods), 14
- getLocation, CovRobMiss-method (CovRobMiss-class), 7
- getLocation-methods (get-methods), 14
- getMissing (get-methods), 14
- getMissing, CovRobMiss-method (CovRobMiss-class), 7
- getMissing-methods (get-methods), 14
- getOutliers, 27
- getOutliers (get-methods), 14
- getOutliers, CovRobMiss-method (CovRobMiss-class), 7
- getOutliers-methods (get-methods), 14

getScale, [8](#), [10](#), [11](#), [17](#), [18](#)  
getScale (get-methods), [14](#)  
getScale, CovRobMissSc-method  
    (CovRobMissSc-class), [8](#)  
getScale-methods (get-methods), [14](#)  
getScatter, [6–8](#), [10](#), [11](#), [17](#), [18](#), [23–26](#), [32](#), [34](#)  
getScatter (get-methods), [14](#)  
getScatter, CovRobMiss-method  
    (CovRobMiss-class), [7](#)  
getScatter-methods (get-methods), [14](#)  
GSE, [7–10](#), [15](#), [16](#), [19](#), [25](#), [26](#), [32–34](#)  
GSE-class, [18](#)  
gy.filt, [20](#), [25](#), [26](#), [32](#), [33](#)

horse, [21](#)  
HuberPairwise, [7](#), [8](#), [16](#), [17](#), [22](#), [24](#)  
HuberPairwise-class, [24](#)

ImpS, [25](#)

partial.mahalanobis, [7](#), [8](#), [26](#)  
plot, [7](#), [11](#), [19](#)  
plot (plot-methods), [27](#)  
plot, CovRobMiss, missing-method  
    (plot-methods), [27](#)  
plot, CovRobMiss-method (plot-methods),  
    [27](#)  
plot-method (plot-methods), [27](#)  
plot-methods, [27](#)

show, CovRobMiss-method  
    (CovRobMiss-class), [7](#)  
show, SummaryCovGSE-method  
    (SummaryCovGSE-class), [30](#)  
simulation-tools, [28](#)  
slrt, [30](#)  
summary, CovRobMiss-method  
    (CovRobMiss-class), [7](#)  
SummaryCovGSE-class, [30](#)

TSGS, [21](#), [29](#), [31](#), [34](#)  
TSGS-class, [33](#)

wages, [34](#)