Package ‘rrcovNA’

September 5, 2016

Date 2016-09-03
Title Scalable Robust Estimators with High Breakdown Point for Incomplete Data
Version 0.4-9
Author Valentin Todorov <valentin.todorov@chello.at>
Description Robust Location and Scatter Estimation and Robust Multivariate Analysis with High Breakdown Point for Incomplete Data.
Maintainer Valentin Todorov <valentin.todorov@chello.at>
Depends rrcov (>= 1.3-7), robustbase (>= 0.92-1), methods
Imports stats4, lattice, cluster, norm
Suggests grid
LazyLoad yes
License GPL (>= 2)
NeedsCompilation yes
Repository CRAN

Date/Publication 2016-09-05 08:26:55

R topics documented:

bush10 ............................................... 2
ces ...................................................... 3
CovNA-class ......................................... 4
CovNAClassic ........................................ 6
CovNAClassic-class .................................. 7
CovNAMcd ............................................ 8
CovNAMcd-class ..................................... 9
CovNAOgk ........................................... 11
CovNAOgk-class .................................... 13
CovNARobust ........................................ 14
CovNARobust-class ................................. 15
bush10  

*bush10*  

*Campbell Bushfire Data with added missing data items (10 percent)*

**Description**

This data set is based on the bushfire data set which was used by Campbell (1984) to locate bushfire scars - see `bushfire` in package robustbase. The original dataset contains satellite measurements on five frequency bands, corresponding to each of 38 pixels. The data set is very well studied (Maronna and Yohai, 1995; Maronna and Zamar, 2002). There are 12 clear outliers: 33-38, 32, 7-11 and 12 and 13 are suspect.

**Usage**

`data(bush10)`

**Format**

A data frame with 38 observations on 6 variables.

The original data set consists of 38 observations in 5 variables. Based on it four new data sets are created in which some of the data items are replaced by missing values with a simple "missing completely at random " mechanism. For this purpose independent Bernoulli trials are realized for each data item with a probability of success 0.1 where success means that the corresponding item is set to missing.)

**Source**


Examples

```r
## The following code will result in exactly the same output
## as the one obtained from the original data set
data(bush10)
plot(bush10)
CovNAMcd(bush10)

## Not run:
## This is the code with which the missing data were created:
## Creates a data set with missing values (for testing purposes)
## from a complete data set 'x'. The probability of
## each item being missing is 'pr'.
##
getmiss <- function(x, pr=0.1){
  library(Rlab)
  n <- nrow(x)
  p <- ncol(x)
  bt <- rbern(n*p, pr)
  btmat <- matrix(bt, nrow=n)
  btmiss <- ifelse(btmat==1, NA, 0)
  x+btmiss
}

## End(Not run)
```

---

ces

**Consumer Expenditure Survey Data**

Description

This data set has been derived from the Quarterly Interview Survey of the Consumer Expenditure Survey (CES) undertaken by the U.S. Department of Labor, Bureau of Labor Statistics and is available at [http://econ.lse.ac.uk/courses/ec220/G/iedata/ces/](http://econ.lse.ac.uk/courses/ec220/G/iedata/ces/) where also more details about this survey can be found. The original data set comprises 869 households in 34 variables of which one is unique ID, five characterize the size of the household, further 6 variables contain other characteristics of the household like age, education ethnicity, etc. and 22 variables represent the household expenditures. We will consider a reduced set of only 8 expenditure variables. This reduced data set was analyzed by Hubert at al. (2009) in the context of PCA and the first step of the analysis showed that all variables are highly skewed. They applied the robust PCA method of Serneels and Verdonck based on the EM algorithm, since some of the data are incomplete.

Usage

```r
data(ces)
```
CovNA-class

Format

A data frame with 869 observations on the following 8 variables:

EXP  Total household expenditure
FDHO  Food and nonalcoholic beverages consumed at home
FDAW  Food and nonalcoholic beverages consumed away from home
SHEL  Housing expenditure
TELE  Telephone services
CLOT  Clothing
HEAL  Health care
ENT   Entertainment

Source

http://econ.lse.ac.uk/courses/ec220/G/jedata/ces/

References


Examples

data(ces)
summary(ces)
plot(ces)

CovNA-class  Class "CovNA" – a base class for estimates of multivariate location and scatter for incomplete data

Description

The class CovNA represents an estimate of the multivariate location and scatter of a data set. The objects of class CovNA contain the classical estimates and serve as base for deriving other estimates, i.e. different types of robust estimates.

Objects from the Class

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

Slots

call: Object of class "language"
cov: covariance matrix
center: location
n.obs: number of observations used for the computation of the estimates
mah: mahalanobis distances
det: determinant
flag: flags (FALSE if suspected an outlier)
method: a character string describing the method used to compute the estimate: "Classic"
singularity: a list with singularity information for the covariance matrix (or NULL of not singular)
x: data

Extends

Class "Cov", directly.

Methods

gDistance signature(obj = "CovNA"): distances
gFlag signature(obj = "CovNA"): Flags observations as outliers if the corresponding mahalanobis distance is larger then qchisq(prob, p) where prob defaults to 0.975.
summary signature(object = "CovNA"): calculate summary information

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


Examples

showClass("CovNA")
CovNAClassic

Classical Estimates of Multivariate Location and Scatter for incomplete data (EM Algorithm)

Description

Computes the classical estimates of multivariate location and scatter. Returns an S4 class CovNAClassic with the estimated center, cov, Mahalanobis distances and weights based on these distances.

Usage

CovNAClassic(x, unbiased=TRUE)
CovNA(x, unbiased=TRUE)

Arguments

x a matrix or data frame. As usual, rows are observations and columns are variables.
unbiased whether to return the unbiased estimate of the covariance matrix. Default is unbiased = TRUE

Value

An object of class "CovNAClassic".

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


See Also

Cov-class, CovClassic-class, CovNAClassic-class

Examples

data(bush10)
cv <- CovNAClassic(bush10)
cv
summary(cv)
CovNAClassic-class

Class "CovNAClassic" - classical estimates of multivariate location and scatter for incomplete data (EM algorithm)

Description

The class CovNAClassic represents an estimate of the multivariate location and scatter of an incomplete data set. The class CovNAClassic objects contain the classical estimates.

Objects from the Class

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

Slots

call: Object of class "language"
cov: covariance matrix
center: location
n.obs: number of observations used for the computation of the estimates
mah: mahalanobis distances
method: a character string describing the method used to compute the estimate: "Classic"
singularity: a list with singularity information for the covariance matrix (or NULL of not singular)
x: data

Methods

plot signature(x = "CovNAClassic"): plot the object

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


Examples

data(bush10)
cv <- CovNAClassic(bush10)
cv
summary(cv)
CovNAMcd

**Robust Location and Scatter Estimation via MCD for incomplete data**

### Description

Computes a robust multivariate location and scatter estimate with a high breakdown point for incomplete data, using the 'Fast MCD' (Minimum Covariance Determinant) estimator.

### Usage

`CovNAMcd(x, alpha = 1/2, nsamp = 500, seed = NULL, trace = FALSE, use.correction = TRUE, impMeth = c("norm", "seq", "rseq"), control)`

### Arguments

- **x**: a matrix or data frame.
- **alpha**: numeric parameter controlling the size of the subsets over which the determinant is minimized, i.e., `alpha*n` observations are used for computing the determinant. Allowed values are between 0.5 and 1 and the default is 0.5.
- **nsamp**: number of subsets used for initial estimates or 'best' or 'exact'. Default is `nsamp = 500`. For `nsamp="best"` exhaustive enumeration is done, as long as the number of trials does not exceed 5000. For 'exact', exhaustive enumeration will be attempted however many samples are needed. In this case a warning message will be displayed saying that the computation can take a very long time.
- **seed**: starting value for random generator. Default is `seed = NULL`.
- **trace**: whether to print intermediate results. Default is `trace = FALSE`.
- **use.correction**: whether to use finite sample correction factors. Default is `use.correction=TRUE`.
- **impMeth**: select imputation method to use - choose one of "norm", "seq" or "rseq". The default is "norm".
- **control**: a control object (S4) of class `CovControlMcd-class` containing estimation options - same as these provided in the function specification. If the control object is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object.

### Details

This function computes the minimum covariance determinant estimator of location and scatter and returns an S4 object of class `CovMcd-class` containing the estimates. The implementation of the function is similar to the existing R function `covMcd()` which returns an S3 object. The MCD method looks for the `h(n/2)` observations (out of `n`) whose classical covariance matrix has the lowest possible determinant. The raw MCD estimate of location is then the average of these `h` points, whereas the raw MCD estimate of scatter is their covariance matrix, multiplied by a consistency factor and a finite sample correction factor (to make it consistent at the normal model.
and unbiased at small samples. Both rescaling factors are returned also in the vector `raw.cnp2` of length 2. Based on these raw MCD estimates, a reweighting step is performed which increases the finite-sample efficiency considerably - see Pison et al. (2002). The rescaling factors for the reweighted estimates are returned in the vector `cnp2` of length 2. Details for the computation of the finite sample correction factors can be found in Pison et al. (2002). The finite sample corrections can be suppressed by setting `use.correction=FALSE`. The implementation in `rrcov` uses the Fast MCD algorithm of Rousseeuw and Van Driessen (1999) to approximate the minimum covariance determinant estimator.

**Value**

An S4 object of class `CovNAMcd-class` which is a subclass of the virtual class `CovNARobust-class`.

**Author(s)**

Valentin Todorov <valentin.todorov@chello.at>

**References**


**Examples**

```r
data(bush10)
mcd <- CovNAMcd(bush10)
mcd
summary(mcd)

plot(mcd)
plot(mcd, which="pairs")
plot(mcd, which="xydistance")
plot(mcd, which="xyqqchi2")
```

---

**Description**

This class, derived from the virtual class "CovRobust" accomodates MCD Estimates of multivariate location and scatter computed by the ‘Fast MCD’ algorithm.
Objects from the Class

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

Slots

- `alpha`: Object of class "numeric" - the size of the subsets over which the determinant is minimized (the default is \((n+p+1)/2\))
- `quan`: Object of class "numeric" - the number of observations on which the MCD is based. If `quan` equals `n.obs`, the MCD is the classical covariance matrix.
- `best`: Object of class "Uvector" - the best subset found and used for computing the raw estimates. The size of `best` is equal to `quan`
- `raw.cov`: Object of class "matrix" - the raw (not reweighted) estimate of location
- `raw.center`: Object of class "vector" - the raw (not reweighted) estimate of scatter
- `raw.mah`: Object of class "Uvector" - mahalanobis distances of the observations based on the raw estimate of the location and scatter
- `raw wt`: Object of class "Uvector" - weights of the observations based on the raw estimate of the location and scatter
- `raw.cn2`: Object of class "numeric" - a vector of length two containing the consistency correction factor and the finite sample correction factor of the raw estimate of the covariance matrix
- `cn2`: Object of class "numeric" - a vector of length two containing the consistency correction factor and the finite sample correction factor of the final estimate of the covariance matrix.
- `iter, crit, wt`: from the "CovRobust" class.
- `call, cov, center, n.obs, mah, method, singularity, X`: from the "Cov" class.

Extends

Class "CovRobust", directly. Class "Cov", by class "CovRobust".

Methods

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

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


See Also

`CovMcd, Cov-class, CovRobust-class`
Examples

```r
showClass("CovNAOmd")
```

---

### Description

Computes a robust multivariate location and scatter estimate with a high breakdown point for incomplete data, using the pairwise algorithm proposed by Marona and Zamar (2002) which in turn is based on the pairwise robust estimator proposed by Gnanadesikan-Kettenring (1972).

### Usage

```r
CovNAOgk(x, niter = 2, beta = 0.9, impMeth = c("norm", "seq", "rseq"), control)
```

### Arguments

- `x`: a matrix or data frame.
- `niter`: number of iterations, usually 1 or 2 since iterations beyond the second do not lead to improvement.
- `beta`: coverage parameter for the final reweighted estimate.
- `impMeth`: select imputation method to use - choose one of "norm", "seq" or "rseq". The default is "norm".
- `control`: a control object (S4) of class CovControlGk-class containing estimation options - same as those provided in the function specification. If the control object is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object. The control object contains also functions for computing the robust univariate location and dispersion estimate `mrob` and for computing the robust estimate of the covariance between two random variables `vrob`.

### Details

The method proposed by Marona and Zamar (2002) allows to obtain positive-definite and almost affine equivariant robust scatter matrices starting from any pairwise robust scatter matrix. The default robust estimate of covariance between two random vectors used is the one proposed by Gnanadesikan and Kettenring (1972) but the user can choose any other method by redefining the function in slot `vrob` of the control object CovControlGk. Similarly, the function for computing the robust univariate location and dispersion used is the tau scale defined in Yohai and Zamar (1998) but it can be redefined in the control object.

The estimates obtained by the OGK method, similarly as in CovMcd are returned as ‘raw’ estimates. To improve the estimates a reweighting step is performed using the coverage parameter beta and these reweighted estimates are returned as ‘final’ estimates.
Value

An S4 object of class `CovNAOgk-class` which is a subclass of the virtual class `CovNARobust-class`.

Note

If the user does not specify a scale and covariance function to be used in the computations or specifies one by using the arguments `smrob` and `svrob` (i.e. the names of the functions as strings), a native code written in C will be called which is by far faster than the R version.

If the arguments `mrob` and `vrob` are not NULL, the specified functions will be used via the pure R implementation of the algorithm. This could be quite slow.

See `CovControlOgk` for details.

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


See Also

`CovNAMcd`

Examples

data(bush10)
CovNAOgk(bush10)

```r
### the following three statements are equivalent
c1 <- CovNAOgk(bush10, niter=1)
c2 <- CovNAOgk(bush10, control = CovControlOgk(niter=1))

### direct specification overrides control one:
c3 <- CovNAOgk(bush10, beta=0.95,
    control = CovControlOgk(beta=0.99))
c1
```

```
CovNAOgk-class

OGK Estimates of Multivariate Location and Scatter for incomplete data

Description

This class, derived from the virtual class "CovRobust" accomodates OGK Estimates of multivariate location and scatter computed by the algorithm proposed by Marona and Zamar (2002).

Objects from the Class

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

Slots

- `raw.cov`: Object of class "matrix" the raw (not reweighted) estimate of covariance matrix
- `raw.center`: Object of class "vector" - the raw (not reweighted) estimate of the location vector
- `raw.mah`: Object of class "Uvector" - mahalanobis distances of the observations based on the raw estimate of the location and scatter
- `raw.wt`: Object of class "Uvector" - weights of the observations based on the raw estimate of the location and scatter
- `iter, crit, wt`: from the "CovRobust" class.
- `call, cov, center, n.obs, mah, method, singularity, X`: from the "Cov" class.

Extends

Class "CovRobust", directly. Class "Cov", by class "CovRobust".

Methods

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

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


See Also

`CovMcd-class, CovMest-class`
Examples

showClass("CovNAOgk")

\[\text{Description}\]

Computes a robust multivariate location and scatter estimate with a high breakdown point for incomplete data, using one of the available estimators.

\[\text{Usage}\]

\[\text{CovNARobust(x, control, impMeth=c("norm", "seq", "rseq"))}\]

\[\text{Arguments}\]

- \(x\): a matrix or data frame.
- \(\text{control}\): a control object (S4) for one of the available control classes, e.g. \text{CovControlMcd-class}, \text{CovControlOgk-class}, \text{CovControlSest-class}, etc., containing estimation options. The class of this object defines which estimator will be used. Alternatively a character string can be specified which names the estimator - one of auto, sde, mcd, ogk, m, mve, sfast, surreal, bisquare, rocke. If 'auto' is specified or the argument is missing, the function will select the estimator (see below for details)
- \(\text{impMeth}\): select imputation method to use - choose one of "norm", "seq" or "rseq". The default is "norm"

\[\text{Details}\]

This function is based on imputation and than estimation with a selected high breakdown point method. Thus first imputation with the selected method will be performed and then the function \text{CovRobust} will be called. For details see \text{CovRobust}.

\[\text{Value}\]

An object derived from a \text{CovRobust} object, depending on the selected estimator.

\[\text{Author(s)}\]

Valentin Todorov <valentin.todorov@chello.at>

\[\text{References}\]


CovNARobust-class

Examples

data(bush10)
CovNARobust(bush10)
CovNARobust(bush10, CovControlSeest())

CovNARobust-class

Class "CovNARobust" - virtual base class for robust estimates of multivariate location and scatter for incomplete data

Description

CovNARobust is a virtual base class used for deriving the concrete classes representing different robust estimates of multivariate location and scatter for incomplete data. Here are implemented the standard methods common for all robust estimates like show, summary and plot. The derived classes can override these methods and can define new ones.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

iter: number of iterations used to compute the estimates

crit: value of the criterion function

wt: weights

call, cov, center, n.obs, mah, method, singularity, X: from the "Cov" class.

Extends

Classes "CovNA" and "CovRobust", directly.

Methods

plot signature(x = "CovNARobust"): plot the object

summary signature(object = "CovNARobust"): display additional information for the object

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References

See Also

CovNA-class, CovNAMcd-class, CovNAQgk-class, CovNASde-class, CovNASest-class

Examples

data(hbk)
hbk.x <- data.matrix(hbk[, 1:3])
cv <- CovMest(hbk.x)  # it is not possible to create an object of
# class CovRobust, since it is a VIRTUAL class
cv
summary(cv)  # summary method for class CovRobust
plot(cv)  # plot method for class CovRobust

CovNASde  Stahel-Donoho Estimates of Multivariate Location and Scatter for incomplete data

Description

Compute a robust estimate of location and scale using the Stahel-Donoho projection based estimator

Usage

CovNASde(x, nsamp, maxres, tune = 0.95, eps = 0.5, prob = 0.99,
impMeth = c("norm", "seq", "rseq"), seed = NULL, trace = FALSE, control)

Arguments

x  a matrix or data frame.
nsamp  a positive integer giving the number of resamples required; nsamp may not be reached if too many of the p-subsamples, chosen out of the observed vectors, are in a hyperplane. If nsamp = 0 all possible subsamples are taken. If nsamp is omitted, it is calculated to provide a breakdown point of eps with probability prob.
maxres  a positive integer specifying the maximum number of resamples to be performed including those that are discarded due to linearly dependent subsamples. If maxres is omitted it will be set to 2 times nsamp.
tune  a numeric value between 0 and 1 giving the fraction of the data to receive non-zero weight. Defaults to 0.95
prob  a numeric value between 0 and 1 specifying the probability of high breakdown point; used to compute nsamp when nsamp is omitted. Defaults to 0.99.
impMeth  select imputation method to use - choose one of "norm", "seq" or "rseq". The default is "norm"
eps  a numeric value between 0 and 0.5 specifying the breakdown point; used to compute nsamp when nresamp is omitted. Defaults to 0.5.
seed starting value for random generator. Default is seed = NULL.
trace whether to print intermediate results. Default is trace = FALSE.
control a control object (S4) of class `CovControlSde-class` containing estimation options - same as those provided in the function specification. If the control object is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object.

Value

An S4 object of class `CovNASde-class` which is a subclass of the virtual class `CovNARobust-class`.

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


Examples

data(bush10)
CovNASde(bush10)

```r
## the following four statements are equivalent
c0 <- CovNASde(bush10)
c1 <- CovNASde(bush10, nsamp=2000)
c2 <- CovNASde(bush10, control = CovControlSde(nsamp=2000))
c3 <- CovNASde(bush10, control = new("CovControlSde", nsamp=2000))

## direct specification overrides control one:
c4 <- CovNASde(bush10, nsamp=100,
    control = CovControlSde(nsamp=2000))

c1
summary(c1)
```
CovNASde-class

Stahel-Donoho Estimates of Multivariate Location and Scatter for incomplete data

Description

This class, derived from the virtual class "CovRobust" accommodates Stahel-Donoho estimates of multivariate location and scatter.

Objects from the Class

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

Slots

iter, crit, wt: from the "CovRobust" class.
call, cov, center, n.obs, mah, method, singularity, X: from the "Cov" class.

Extends

Class "CovRobust", directly. Class "Cov", by class "CovRobust".

Methods

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

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


See Also

CovSde, Cov-class, CovRobust-class

Examples

showClass("CovNASde")
Description

Computes S-Estimates of multivariate location and scatter based on Tukey’s biweight function for incomplete data using a fast algorithm similar to the one proposed by Salibian-Barrera and Yohai (2006) for the case of regression. Alternatively, the Ruppert’s SURREAL algorithm, bisquare or Rocke type estimation can be used.

Usage

```r
CovNASest(x, bdp = 0.5, arp = 0.1, eps = 1e-5, maxiter = 120,
          nsamp = 500, impMeth = c("norm", "seq", "rseq"), seed = NULL,
          trace = FALSE, tolsolve = 1e-13,
          scalefn, method = c("sfast", "surreal", "bisquare", "rocke", "suser", "sdet"), control,
          t0, S0, initcontrol)
```

Arguments

- `x`: a matrix or data frame.
- `bdp`: a numeric value specifying the required breakdown point. Allowed values are between \((n - p)/(2 \times n)\) and 1 and the default is `bdp = 0.5`.
- `arp`: a numeric value specifying the asymptotic rejection point (for the Rocke type S estimates), i.e. the fraction of points receiving zero weight (see Rocke (1996)). Default is `arp = 0.1`.
- `eps`: a numeric value specifying the relative precision of the solution of the S-estimate (bisquare and Rocke type). Default is to `eps = 1e-5`.
- `maxiter`: maximum number of iterations allowed in the computation of the S-estimate (bisquare and Rocke type). Default is `maxiter = 120`.
- `nsamp`: the number of random subsets considered. Default is `nsamp = 500`.
- `impMeth`: select imputation method to use - choose one of "norm", "seq" or "rseq". The default is "norm".
- `seed`: starting value for random generator. Default is `seed = NULL`.
- `trace`: whether to print intermediate results. Default is `trace = FALSE`.
- `tolsolve`: numeric tolerance to be used for inversion (solve) of the covariance matrix in `mahalanobis`.
- `scalefn`: function to compute a robust scale estimate or character string specifying a rule determining such a function. Used for computing the "deterministic" S-estimates (method="sdet"). If `scalefn` is missing or is `NULL`, the function is selected depending on the data set size, following the recommendation of Hubert et al. (2012) - `Qn` if \(n \leq 1000\) and `scaleTau2` otherwise.
method  Which algorithm to use: 'sfast'=FAST-S, 'surreal'=SURREAL, 'bisquare', 'rocke' or 'sdet', which will invoke the deterministic algorithm of Hubert et al. (2012).

control  a control object (S4) of class CovControlSest-class containing estimation options - same as these provided in the function specification. If the control object is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object.

t₀  optional initial HBDP estimate for the center
S₀  optional initial HBDP estimate for the covariance matrix
initcontrol  optional control object to be used for computing the initial HBDP estimates

Details

Computes biweight multivariate S-estimator of location and scatter. The computation will be performed by one of the following algorithms:

FAST-S  An algorithm similar to the one proposed by Salibian-Barrera and Yohai (2006) for the case of regression
SURREAL  Ruppert’s SURREAL algorithm when method is set to 'surreal'
BISQUARE  Bisquare S-Estimate with method set to 'bisquare'
ROCKE  Rocke type S-Estimate with method set to 'rocke'.

Value

An S4 object of class CovNASest-class which is a subclass of the virtual class CovNARobust-class.

Author(s)

Valentin Todorov <valentin.todorov@chello.at>, Matias Salibian-Barrera <matias@stat.ubc.ca> and Victor Yohai <vyohai@dm.uba.ar>. See also the code from Kristel Joossens, K.U. Leuven, Belgium and Ella Roelant, Ghent University, Belgium.

References


Examples

library(rrcov)
data(bush10)
CovNASest(bush10)
```r
## the following four statements are equivalent
c0 <- CovNASest(bush10)
c1 <- CovNASest(bush10, bdp = 0.25)
c2 <- CovNASest(bush10, control = CovControlSest(bdp = 0.25))
c3 <- CovNASest(bush10, control = new("CovControlSest", bdp = 0.25))

## direct specification overrides control one:
c4 <- CovNASest(bush10, bdp = 0.40,
control = CovControlSest(bdp = 0.25))
c1
summary(c1)

## Use the SURREAL algorithm of Ruppert
cr <- CovNASest(bush10, method="surreal")
cr

## Use Bisquare estimation
cr <- CovNASest(bush10, method="bisquare")
cr

## Use Rocke type estimation
cr <- CovNASest(bush10, method="rocke")
cr
```

---

**CovNASest-class**


### Description

This class, derived from the virtual class "CovRobust" accommodates \textit{S} Estimates of multivariate location and scatter computed by the ‘Fast \textit{S}’ or ‘SURREAL’ algorithm.

### Objects from the Class

Objects can be created by calls of the form \texttt{new("CovSest", ...)}, but the usual way of creating \texttt{CovSest} objects is a call to the function \texttt{CovSest} which serves as a constructor.

### Slots

\texttt{iter, crit, wt}: from the "CovRobust" class.

\texttt{call, cov, center, n.obs, mah, method, singularity, X}: from the "Cov" class.

### Extends

Class "CovRobust", directly. Class "Cov", by class "CovRobust".
Methods

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

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


See Also

*CovSest, Cov-class, CovRobust-class*

Examples

showClass("CovNASest")

---

**impNorm**

*Impute missing multivariate normal data*

Description

Draws missing elements of a data matrix under the multivariate normal model and a user-supplied parameter.

Usage

impNorm(x)

Arguments

x the original incomplete data matrix.

Details

This function simply uses *imp.norm* from package *norm*.

Value

A matrix of the same form as x, but with all missing values filled in with simulated values drawn from their predictive distribution given the observed data and the specified parameter.

References

See Section 5.4.1 of Schafer (1996).
impSeq

See Also

prelim.norm, makeparam.norm, and rngseed.

Examples

data(bush10)
impNorm(bush10) #impute missing data under the MLE

impSeq

Sequential imputation of missing values

Description

Impute missing multivariate data using sequential algorithm

Usage

impSeq(x)

Arguments

x the original incomplete data matrix.

Details

SEQimpute starts from a complete subset of the data set Xc and estimates sequentially the missing values in an incomplete observation, say x*, by minimizing the determinant of the covariance of the augmented data matrix X* = [Xc; x']. Then the observation x* is added to the complete data matrix and the algorithm continues with the next observation with missing values.

Value

a matrix of the same form as x, but with all missing values filled in sequentially.

References


Examples

data(bush10)
impSeq(bush10) # impute sequentially missing data
impSeqRob  

Robust sequential imputation of missing values

Description

Impute missing multivariate data using robust sequential algorithm

Usage

impSeqRob(x, alpha=0.9)

Arguments

x  
the original incomplete data matrix.

alpha  
The default is alpha=0.9.

Details

seqimpute starts from a complete subset of the data set Xc and estimates sequentially the missing values in an incomplete observation, say x*, by minimizing the determinant of the covariance of the augmented data matrix X* = [Xc; x']. Then the observation x* is added to the complete data matrix and the algorithm continues with the next observation with missing values. Since seqimpute uses the sample mean and covariance matrix it will be vulnerable to the influence of outliers and it is improved by plugging in robust estimators of location and scatter. One possible solution is to use the outlyingness measure as proposed by Stahel (1981) and Donoho (1982) and successfully used for outlier identification in Hubert et al. (2005). We can compute the outlyingness measure for the complete observations only but once an incomplete observation is imputed (sequentially) we could compute the outlyingness measure for it too and use it to decide if this observation is an outlier or not. If the outlyingness measure does not exceed a predefined threshold the observation is included in the further steps of the algorithm.

Value

a matrix of the same form as x, but with all missing values filled in sequentially.

References


Examples

data(bush10)
impSeqRob(bush10) # impute sequentially missing data
Description

Computes classical and robust principal components for incomplete data using an EM algorithm as
described by Serneels and Verdonck (2008)

Usage

```R
PcaNA(x, ...)  # Default S3 method:
PcaNA(x, k = ncol(x), kmax = ncol(x), conv=1e-10, maxiter=100,
method=c("cov", "locantore", "hubert", "grid", "proj", "class"), cov.control=NULL,
scale = FALSE, signflip = TRUE, crit.pca.distances = 0.975, trace=FALSE, ...)
```

Arguments

- `formula`: a formula with no response variable, referring only to numeric variables.
- `data`: an optional data frame (or similar: see `model.frame`) containing the variables
  in the formula `formula`.
- `subset`: an optional vector used to select rows (observations) of the data matrix `x`.
- `na.action`: a function which indicates what should happen when the data contain NAs. The
default is set by the `na.action` setting of `options`, and is `na.fail` if that is
unset. The default is `na.omit`.
- ... arguments passed to or from other methods.
- `x`: a numeric matrix (or data frame) which provides the data for the principal components analysis.
- `k`: number of principal components to compute. If `k` is missing, or `k = 0`, the
  algorithm itself will determine the number of components by finding such `k` that
  `l_k/l_1 >= 10. E - 3` and `\sum_{j=1}^{k} l_j / \sum_{j=1}^{r} l_j >= 0.8`. It is preferable to investigate
  the scree plot in order to choose the number of components and then run again.
  Default is `k=ncol(x)`.
- `kmax`: maximal number of principal components to compute. Default is `kmax=10`. If `k`
  is provided, `kmax` does not need to be specified, unless `k` is larger than 10.
- `conv`: convergence criterion for the EM algorithm. Default is `conv=1e-10`.
- `maxiter`: maximal number of iterations for the EM algorithm. Default is `maxiter=100`.
- `method`: which PC method to use (classical or robust) - "class" means classical PCA and
  one of the following "locantore", "hubert", "grid", "proj", "cov" specifies a robust PCA method. If the method is "cov" - i.e. PCA based on a robust covariance
  matrix - the argument `cov.control` can specify which method for computing the (robust) covariance matrix will be used. Default is `method="locantore"`. 
cov.control | control object in case of robust PCA based on a robust covariance matrix.
scale | a logical value indicating whether the variables should be scaled to have unit variance (only possible if there are no constant variables). As a scale function mad is used but alternatively, a vector of length equal the number of columns of x can be supplied. The value is passed to scale and the result of the scaling is stored in the scale slot. Default is scale = FALSE
signflip | a logical value indicating whether to try to solve the sign indeterminancy of the loadings - ad hoc approach setting the maximum element in a singular vector to be positive. Default is signflip = FALSE
crit.pca.distances | criterion to use for computing the cutoff values for the orthogonal and score distances. Default is 0.975.
trace | whether to print intermediate results. Default is trace = FALSE

Details

PcaNA, serving as a constructor for objects of class PcaNA-class is a generic function with "formula" and "default" methods. For details see the relevant references.

Value

An S4 object of class PcaNA-class which is a subclass of the virtual class Pca-class.

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References

Serneels S & Verdonck T (2008), Principal component analysis for data containing outliers and missing elements. Computational Statistics and Data Analysis, 52(3), 1712–1727.


Examples

```r
## 1. With complete data
## PCA of the bushfire data
data(bushfire)
pca <- PcaNA(bushfire)
pca

## Compare with the classical PCA
prcomp(bushfire)

## or
PcaNA(bushfire, method="class")

## If you want to print the scores too, use
```
### PcaNA-class

**Class “PcaNA” Principal Components for incomplete data**

**Description**

Contains the results of the computations of classical and robust principal components for incomplete data using an EM algorithm as described by Serneels and Verdonck (2008)
Objects from the Class

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

Slots

call, center, scale, loadings, eigenvalues, scores, k, sd, od, cutoff.sd, cutoff.od, flag, n.obs: from the "Pca" class.

Ximp: the data matrix with imputed missing values

Extends

Class "Pca", directly.

Methods

getQuan signature(obj = "PcaNA"): ...

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References


See Also

PcaRobust-class, Pca-class, PcaClassic, PcaClassic-class

Examples

showClass("PcaNA")

---

SummaryCovNA-class

Class "SummaryCovNA" - summary of "CovNA" objects

Description

The "CovNA" object plus some additional summary information

Objects from the Class

Objects can be created by calls of the form `new("SummaryCovNA", ...) but most often by invoking 'summary' on a "CovNA" object. They contain values meant for printing by 'show'.
Slots
No Slots defined with class "SummaryCovNA" in the signature.

Methods
No Methods defined with class "SummaryCovNA" in the signature.

Author(s)
Valentin Todorov <valentin.todorov@chello.at>

References

See Also
CovNA-class

Examples
`showClass("SummaryCovNA")`
Author(s)
Valentin Todorov <valentin.todorov@chello.at>

References

See Also
CovRobust-class, SummaryCov-class

Examples
```r
data(hbk)
hbk.x <- data.matrix(hbk[, 1:3])
cv <- CovMest(hbk.x)
cv
summary(cv)
```
Index

*Topic **classes**

CovNA-class, 4
CovNAClassic, 6
CovNAClassic-class, 7
CovNAMcd-class, 9
CovNAOgk-class, 13
CovNARobust-class, 15
CovNASde-class, 18
CovNASest-class, 21
impNorm, 22
impSeq, 23
impSeqRob, 24
SummaryCovNA-class, 28
SummaryCovNARobust-class, 29

*Topic **datasets**

bush10, 2
ces, 3

*Topic **multivariate**

CovNA-class, 4
CovNAClassic, 6
CovNAClassic-class, 7
CovNAMcd, 8
CovNAMcd-class, 9
CovNAOgk, 11
CovNAOgk-class, 13
CovNARobust, 14
CovNARobust-class, 15
CovNASde, 16
CovNASde-class, 18
CovNASest, 19
CovNASest-class, 21
impNorm, 22
impSeq, 23
impSeqRob, 24
PcaNA, 25
PcaNA-class, 27
SummaryCovNARobust-class, 29

*Topic **robust**

CovNA-class, 4

CovNAClassic, 6
CovNAClassic-class, 7
CovNAMcd, 8
CovNAMcd-class, 9
CovNAOgk, 11
CovNAOgk-class, 13
CovNARobust, 14
CovNARobust-class, 15
CovNASde, 16
CovNASde-class, 18

bush10, 2
bushfire, 2
ces, 3
Cov, 5, 10, 13, 15, 18, 21
CovContrologk, 12
CovMcd, 10
covMcd, 8
CovNA, 15
CovNA (CovNAClassic), 6
CovNA-class, 4
CovNAClassic, 6
CovNAClassic-class, 7
CovNAMcd, 8, 12
CovNAMcd-class, 9
CovNAOgk, 11
CovNAOgk-class, 13
CovNARobust, 14
CovNARobust-class, 15
CovNASde, 16
CovNASde-class, 18
CovNASeSt, 19
CovNASeSt-class, 21
CovRRobust, 10, 13–15, 18, 21
CovSDe, 18
CovSeS, 22

function, 19

getDistance, CovNA-method (CovNA-class), 4
getFlag, CovNA-method (CovNA-class), 4
getQuan, PcaNA-method (PcaNA-class), 27

impNorm, 22
impSeq, 23
impSeqRob, 24

mahalanobis, 19
makeparam.norm, 23
model.frame, 25

na.fail, 25
na.omit, 25

options, 25

Pca, 28
PcaClassic, 28
PcaNA, 25
PcaNA-class, 27
plot, CovNAClassic, missing-method (CovNAClassic-class), 7
plot, CovNARobust, missing-method (CovNARobust-class), 15
prelim.norm, 23

Qn, 19

rngseed, 23

scaleTau2, 19
show, SummaryCovNA-method (SummaryCovNA-class), 28
show, SummaryCovNARobust-method (SummaryCovNARobust-class), 29
solve, 19

summary, CovNA-method (CovNA-class), 4
summary, CovNARobust-method (CovNARobust-class), 15
SummaryCovNA-class, 28
SummaryCovNARobust-class, 29