# Package ‘MAINT.Data’

June 13, 2017

**Type**  Package  
**Title**  Model and Analyse Interval Data  
**Version**  1.1.2  
**Date**  2017-06-12  
**Author**  Pedro Duarte Silva <psilva@porto.ucp.pt>, Paula Brito <mpbrito.fep.up.pt>  
**Maintainer**  Pedro Duarte Silva <psilva@porto.ucp.pt>  
**Description**  Implements methodologies for modelling interval data by Normal and Skew-Normal distributions, considering appropriate parameterizations of the variance-covariance matrix that takes into account the intrinsic nature of interval data, and lead to four different possible configuration structures. The Skew-Normal parameters can be estimated by maximum likelihood, while Normal parameters may be estimated by maximum likelihood or robust trimmed maximum likelihood methods.  
**License**  GPL-2  
**LazyLoad**  yes  
**LazyData**  yes  
**Depends**  R (>= 3.1.0), Rcpp (>= 0.11.0), sn (>= 1.3.0)  
**Imports**  MASS, methods, miscTools, robustbase, rrcov, pcaPP  
**LinkingTo**  Rcpp, RcppEigen  
**NeedsCompilation**  yes  
**Repository**  CRAN  
**Date/Publication**  2017-06-13 07:16:22 UTC

```r
R topics documented:

<table>
<thead>
<tr>
<th>Package</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAINT.Data-package</td>
<td>3</td>
</tr>
<tr>
<td>BestModel-methods</td>
<td>6</td>
</tr>
<tr>
<td>Cars</td>
<td>7</td>
</tr>
<tr>
<td>ChinaTemp</td>
<td>7</td>
</tr>
<tr>
<td>coef-methods</td>
<td>8</td>
</tr>
</tbody>
</table>
```
## R topics documented:

- `ConfTests-class` ........................................ 9
- `cor--methods` ........................................ 10
- `DACrossVal` ............................................ 11
- `extmatrix-class` ....................................... 13
- `fasttle-methods` ....................................... 13
- `fulltle-methods` ....................................... 18
- `getIdtOutl` ............................................ 21
- `IData` .................................................. 22
- `IData-class` ............................................ 23
- `IdtE-class` ............................................. 25
- `Idtlda-class` .......................................... 26
- `IdtMANOVA-class` ...................................... 27
- `IdtMxE-class` .......................................... 29
- `IdtMxNandSNDE-class` .................................. 30
- `IdtMxNDE-class` ....................................... 31
- `IdtMxNDRE-class` ...................................... 33
- `IdtMxSNDE-class` ...................................... 34
- `IdtMxtNDE-class` ...................................... 35
- `IdtNandSNDE-class` .................................... 36
- `IdtNDNE-class` ........................................ 36
- `Idtqda-class` ........................................... 37
- `IdtSNDE-class` .......................................... 38
- `IdtSNgenda-class` ...................................... 39
- `IdtSngNandSNDE-class` .................................. 40
- `IdtSngNDE-class` ....................................... 41
- `IdtSngNDRE-class` ..................................... 42
- `IdtSngSNDE-class` ...................................... 44
- `IdtSNlocda-class` ...................................... 45
- `lda-methods` ............................................ 46
- `LRTest-class` ........................................... 48
- `MANOVA-methods` ....................................... 49
- `mle-methods` ........................................... 51
- `qda-methods` ........................................... 52
- `qHardRoqF` ............................................ 54
- `RepLOptim` ............................................. 55
- `Robda-methods` ......................................... 57
- `RobEstControl` ......................................... 60
- `RobEstControl-class` ................................... 62
- `RobMxDEst-methods` .................................... 64
- `snda-methods` ......................................... 66
- `stdEr--methods` ....................................... 68
- `testMod-methods` ...................................... 69
- `var--methods` .......................................... 70
- `vcov--methods` ........................................ 71

## Index 73
Description

MAINT.DATA implements methodologies for modelling Interval Data by Normal and Skew-Normal distributions, considering four different possible configurations structures for the variance-covariance matrix. It introduces a data class for representing interval data and includes functions and methods for parametric modelling and analysing of interval data. It performs maximum likelihood and trimmed maximum likelihood estimation, statistical tests, as well as (M)ANOVA and Discriminant Analysis.

Details

In the classical model of multivariate data analysis, data is represented in a data-array where n “individuals” (usually in rows) take exactly one value for each variable (usually in columns). Symbolic Data Analysis (see, e.g., Noirhomme-Fraiture and Brito (2011)) provides a framework where new variable types allow to take directly into account variability and/or uncertainty associated to each single “individual”, by allowing multiple, possibly weighted, values for each variable. New variable types - interval, categorical multi-valued and modal variables - have been introduced.

We focus on the analysis of interval data, i.e., where elements are described by variables whose values are intervals. Parametric inference methodologies based on probabilistic models for interval variables are developed in Brito and Duarte Silva (2011) where each interval is represented by its midpoint and log-range, for which Normal and Skew-Normal (Azzalini and Dalla Valle (1996)) distributions are assumed. The intrinsic nature of the interval variables leads to special structures of the variance-covariance matrix, which are represented by four different possible configurations. MAINT.DATA implements the proposed methodologies in R, introducing a data class for representing interval data; it includes functions for modelling and analysing interval data, in particular maximum likelihood and trimmed maximum likelihood (see, e.g. Hadi and Luceno (1997)) estimation, and statistical tests for the different considered configurations. Methods for (M)ANOVA and Discriminant Analysis (Duarte Silva and Brito (2015)) of this data class are also provided.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

Maintainer: Pedro Duarte Silva <psilva@porto.ucp.pt>
MAINT.Data-package

References


Examples

```r
# Create an Interval-Data object containing the intervals for 899 observations on the temperatures by quarter in 60 Chinese meteorological stations.
ChinaT <- IData(ChinaTemp[1:8], VarNames=c("T1","T2","T3","T4"))

# Display the first and last observations
head(ChinaT)
tail(ChinaT)

# Print summary statistics
summary(ChinaT)

# Create a new data set considering only the Winter (1st and 4th) quarter intervals
ChinaWT <- ChinaT[,c(1,4)]

# Estimate normal distribution parameters by maximum likelihood, assuming the classical (unrestricted) covariance configuration Case 1
ChinaWTE.C1 <- mle(ChinaWT,CovCase=1)
cat("Winter temperatures of China -- normal maximum likelihood estimation results:\n")
print(ChinaWTE.C1)
cat("Standard Errors of Estimators:\n") ; print(stdEr(ChinaWTE.C1))

# Estimate normal distribution parameters by maximum likelihood, assuming that one of the C2, C3 or C4 restricted covariance configuration cases hold
ChinaWTE.C234 <- mle(ChinaWT,CovCase=2:4)
cat("Winter temperatures of China -- normal maximum likelihood estimation results:\n")
print(ChinaWTE.C234)
```
cat("Standard Errors of Estimators:\n") ; print(stderr(ChinaWTE.C234))

# Estimate normal distribution parameters robustly by fast maximum trimmed likelihood,
# assuming that one of the C2, C3 or C4 restricted covariance configuration cases hold

## Not run:
ChinaWTE.C234 <- fastttle(ChinaWT,CovCase=2:4)
cat("Winter temperatures of China -- normal maximum trimmed likelihood estimation results:\n")
print(ChinaWTE.C234)

# Estimate skew-normal distribution parameters
ChinaWTE.SkN <- mle(ChinaWT,Model="SKNormal")
cat("Winter temperatures of China -- Skew-Normal maximum likelihood estimation results:\n")
print(ChinaWTE.SkN)
cat("Standard Errors of Estimators:\n") ; print(stderr(ChinaWTE.SkN))

## End(Not run)

#MANOVA tests assuming that configuration case 1 (unrestricted covariance)
# or 3 (MidPoints independent of Log-Ranges) holds.
ManvChinaWT.C13 <- MANOVA(ChinaWT,ChinaTemp$GeoReg,CovCase=c(1,3))
cat("Winter temperatures of China -- MANOVA by geografical regions results:\n")
print(ManvChinaWT.C13)

#Linear Discriminant Analysis
ChinaWT.lda <- lda(ManvChinaWT.C13)
cat("Winter temperatures of China -- linear discriminant analysis results:\n")
print(ChinaWT.lda)
cat("lda Prediction results:\n")
print(predict(ChinaWT.lda,ChinaWT)$class)

## Not run:
#Estimate error rates by ten-fold cross-validation
CVlda <- DACrossVal(ChinaWT,ChinaTemp$GeoReg,TrainAlg=lda,
CovCase=BestModel(HIres(ManvChinaWT.C13)),CVrep=1)
summary(CVlda[,,"Clerr"])
glerrors <-
apply(CVlda[,,"Nk"]*CVlda[,,"Clerr"],1,sum)/apply(CVlda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glerrors),"\n")

#Robust Quadratic Discriminant Analysis
ChinaWT.rqda <- Robqda(ChinaWT,ChinaTemp$GeoReg)
cat("Winter temperatures of China -- robust quadratic discriminant analysis results:\n")
print(ChinaWT.rqda)
cat("robust qda prediction results:\n")
print(predict.qda(ChinaWT.rqda,ChinaWT)$class)

## End(Not run)
# Create an Interval-Data object containing the intervals for characteristics
# of 27 cars models.

CarsIdt <- IData(Cars1:8,VarNames=c("Price","EngineCapacity","TopSpeed","Acceleration"))

# Display the first and last observations
head(CarsIdt)
tail(CarsIdt)

# Estimate normal distribution parameters

CarsNE <- mle(CarsIdt)
cat("Cars data -- normal maximum likelihood estimation results:\n")
print(CarsNE)
cat("Standard Errors of Estimators:\n") ; print(stdErr(CarsNE))

# Estimate normal distribution parameters robustly by full maximum trimmed likelihood,

## Not run:
CarsTE <- fulltlt(CarsIdt)
cat("Cars data -- normal maximum trimmed likelihood estimation results:\n")
print(CarsTE)

# Estimate parameters searching through normal and Skew-Normal distributions.

CarsNSNE <- mle(CarsIdt,Model="NrmndSKN")
cat("Cars data -- Maximum likelihood estimation results:\n")
print(CarsNSNE)
cat("Standard Errors of Estimators:\n") ; print(stdErr(CarsNSNE))

## End(Not run)

---

**BestModel-methods**

*Methods for function BestModel in Package ‘MAINT.Data’*

**Description**

Selects the best model according to the chosen selection criterion (currently, BIC or AIC)

**Usage**

```
BestModel(ModE,SelCrit=c("IdtCrt","BIC","AIC"))
```

**Arguments**

- **ModE**
  
  An object of class `IdtE` representing the estimates of a model fitted to a data set of interval-value variables
### SelCrit

The model selection criterion. “IdtCrt” stands for the criterion originally used in the ModE estimation, while “BIC” and “AIC” represent respectively the Bayesian and Akaike information criteria.

### Value

An integer with the index of the model chosen by the selection criterion.

---

### Cars

**Cars Data Set**

### Description

This data set consist of the intervals for four characteristics (Price, EngineCapacity, TopSpeed and Acceleration) of 27 cars models partitioned into four different classes (Utilitarian, Berlina, Sportive and Luxury).

### Usage

data(Cars)

### Format

A data frame containing 27 observations on 9 variables, the first eight with the lower and upper bounds of the interval characteristics for 27 car models, the last one a factor indicating the model class.

---

### ChinaTemp

**China Temperatures Data Set**

### Description

This data set consist of the intervals of observed temperatures (Celsius scale) in each of the four quarters, Q_1 to Q_4, of the years 1974 to 1988 in 60 Chinese meteorological stations; one outlier observation (YinChuan_1982) has been discarded. The 60 stations belong to different regions in China, which therefore define a partition of the 899 stations-year combinations.

### Usage

data(ChinaTemp)

### Format

A data frame containing 899 observations on 9 variables, the first eight with the lower and upper bounds of the temperatures by quarter in the 899 stations-year combinations, the last one a factor indicating the geographical region of each station.
Methods for function coef in Package 'MAINT.Data'

Description

S4 methods for function coef. As in the generic coef S3 'stats' method, these methods extract parameter estimates for the models fitted to Interval Data.

Usage

```r
## S4 method for signature 'IdtNDE'
coef(object, selmodel=BestModel(object), ...)
## S4 method for signature 'IdtSNDE'
coef(object, selmodel=BestModel(object), ParType=c("Centr", "Direct", "All"), ...)
## S4 method for signature 'IdtNandSNDE'
coef(object, selmodel=BestModel(object), ParType=c("Centr", "Direct", "All"), ...)
```

Arguments

- `object` An object representing a model fitted to interval data.
- `selmodel` Selected model from a list of candidate models saved in object.
- `ParType` Parameterization of the Skew-Normal distribution. Only used when object has class `IdtSNDE` or `IdtNandSNDE` and in this latter case when argument "selmodel" chooses a Skew-Normal model. Alternatives are “Centr” for centred parameters, “Direct” for direct parameters and “All”, for both types of parameters. See Arellano-Valle and Azzalini (2008) for details.
- `...` Additional arguments for method functions.

Value

A vector of parameter estimates.

References


See Also

`stdEr`, `vcov`
ConfTests-class

Class "Configuration Tests"

Description

ConfTests contains a list of the results of statistical likelihood-ratio tests that evaluate the goodness-of-fit of restricted models against more general ones. Currently, the models implemented are those based on the Normal and Skew-Normal distributions, with the four alternative variance-covariance matrix configurations.

Slots

**TestRes:** List of test results; each element is an object of class LRTest, with the following components:

- **QuiSq:** Value of the Qui-Square statistics corresponding to the performed test.
- **df:** Degrees of freedom of the Qui-Square statistics.
- **pvalue:** p-value of the Qui-Square statistics value, obtained from the Qui-Square distribution with df degrees of freedom.
- **H0logLik:** Logarithm of the Likelihood function under the null hypothesis.
- **H1logLik:** Logarithm of the Likelihood function under the alternative hypothesis.

**RestModels:** The restricted model (corresponding to the null hypothesis)

**FullModels:** The full model (corresponding to the alternative hypothesis)

Methods

- **show** signature(object = "ConfTests"): show S4 method for the ConfTests-class

Author(s)

- Pedro Duarte Silva <psilva@porto.ucp.pt>
- Paula Brito <mpbrito.fep.up.pt>

See Also

- **mle, iData, LRTest**
Methods for function cor in Package ‘MAINT.Data’

Description

S4 methods for function cor. These methods extract estimates of correlation matrices for the models fitted to Interval Data.

Usage

```r
## S4 method for signature 'IdtNDE'
cor(x)
## S4 method for signature 'IdtSNDE'
cor(x)
## S4 method for signature 'IdtNandSNDE'
cor(x)
## S4 method for signature 'IdtMxNDE'
cor(x)
## S4 method for signature 'IdtMxSNDE'
cor(x)
```

Arguments

- `x` An object representing a model fitted to interval data.

Value

For the `IdtNDE`, `IdtSNDE` and `IdtNandSNDE` methods or `IdtMxNDE`, `IdtMxSNDE` methods with slot “Hmcdt” equal to TRUE: a matrix with the estimated correlations.

For the `IdtMxNDE`, and `IdtMxSNDE` methods with slot “Hmcdt” equal to FALSE: a three-dimensional array with a matrix with the estimated correlations for each group at each level of the third dimension.

See Also

- `var`
DACrossVal

Cross Validation for Discriminant Analysis Classification Rules

Description

‘DACrossVal’ evaluates the performance of a Discriminant Analysis training sample algorithm by k-fold Cross-Validation.

Usage

DACrossVal(data, grouping, TrainAlg, EvalAlg=EvalClrule,
Strfolds=TRUE, kfold=10, CVrep=20, prior="proportions", loo=FALSE, ...)

Arguments

data               Matrix, data frame or Interval Data object of observations.
grouping           Factor specifying the class for each observation.
TrainAlg           A function with the training algorithm. It should return an object that can be used as input to the argument of ‘EvalAlg’.
EvalAlg            A function with the evaluation algorithm. By default set to ‘EvalClrule’ which returns a list with components “err” (estimates of error rates by class) and “Nk” (number of out-sample observations by class). This default can be used for all ‘TrainAlg’ arguments that return an object with a predict method returning a list with a ‘class’ component (a factor) containing the classification results.
Strfolds           Boolean flag indicating if the folds should be stratified according to the original class proportions (default), or randomly generated from the whole training sample, ignoring class membership.
kfold              Number of training sample folds to be created in each replication.
CVrep              Number of replications to be performed.
prior              The prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
loo                A boolean flag indicating if a leave-one-out strategy should be employed. When set to “TRUE” overrides the kfold and CVrep arguments.
...                Further arguments to be passed to ‘TrainAlg’ and ‘EvalAlg’.

Value

A three dimensional array with the number of tested observations, and estimated classification errors for each combination of fold and replication tried. The array dimensions are defined as follows:
The first dimension runs through the different fold-replication combinations.
The second dimension represents the classes.
The third dimension has two named levels representing respectively the number of observations tested (“Nk”), and the estimated classification errors (“Clerr”).
## Author(s)

A. Pedro Duarte Silva

## See Also

`lda`, `qda`, `IData`

## Examples

```r
## Not run:

# Compare performance of linear and quadratic discriminant analysis with
# Covariance cases c1 and c4 on the ChinaT data set by 5-fold cross-validation
# replicated twice

# Create an Interval-Data object containing the intervals for 899 observations
# on the temperatures by quarter in 60 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8])

# Classical (configuration 1) Linear Discriminant Analysis

CVldaC1 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=lda,CovCase=1,kfold=5,CVrep=2)
summary(CVldaC1[,"Clerr"])

gberrors <- apply(CVldaC1[,"Nk"]*CVldaC1[,"Clerr"],1,sum)/apply(CVldaC1[,"Nk"],1,sum)
cat("Average global classification error =",mean(gberrors),"\n")

# Linear Discriminant Analysis with covariance case 3

CVldaC4 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=lda,CovCase=3,kfold=5,CVrep=2)
summary(CVldaC4[,"Clerr"])

gberrors <- apply(CVldaC4[,"Nk"]*CVldaC4[,"Clerr"],1,sum)/apply(CVldaC4[,"Nk"],1,sum)
cat("Average global classification error =",mean(gberrors),"\n")

# Classical (configuration 1) Quadratic Discriminant Analysis

CVqdaC1 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=qda,CovCase=1,kfold=5,CVrep=2)
summary(CVqdaC1[,"Clerr"])

gberrors <- apply(CVqdaC1[,"Nk"]*CVqdaC1[,"Clerr"],1,sum)/apply(CVqdaC1[,"Nk"],1,sum)
cat("Average global classification error =",mean(gberrors),"\n")

# Quadratic Discriminant Analysis with covariance case 3

CVqdaC4 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=qda,CovCase=3,kfold=5,CVrep=2)
summary(CVqdaC4[,"Clerr"])

gberrors <- apply(CVqdaC4[,"Nk"]*CVqdaC4[,"Clerr"],1,sum)/apply(CVqdaC4[,"Nk"],1,sum)
cat("Average global classification error =",mean(gberrors),"\n")

## End(Not run)
```
extmatrix-class

Class “extmatrix”

Description

“extmatrix” is a simple extension of the base matrix class, that that accepts NULL objects as members.

Extends

Class matrix, directly.

fasttle-methods

Methods for Function fasttle in Package ‘MAINT.Data’

Description

Performs maximum trimmed likelihood estimation by the fasttle algorithm

Usage

fasttle(Idt, CovCase=1:4, SelCrit=c("BIC","AIC"), alpha=control@alpha, nsamp = control@nsamp, seed=control@seed, trace=control@trace, use.correction=control@use.correction, nsteps=control@ncsteps, getalpha=control@getalpha, rawMD2Dist=control@rawMD2Dist, MD2Dist=control@MD2Dist, eta=control@eta, multiCmpCor=control@multiCmpCor, getkdblstar=control@getkdblstar, outlin=control@outlin, trialmethod=control@trialmethod, m=control@m, reweighted = control@rewighted, otpType=control@otpType, control=RobEstControl(), ...)
### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Idt</strong></td>
<td>An IData object representing interval-valued entities.</td>
</tr>
<tr>
<td><strong>CovCase</strong></td>
<td>Configuration of the variance-covariance matrix: a set of integers between 1 and 4.</td>
</tr>
<tr>
<td><strong>SelCrit</strong></td>
<td>The model selection criterion.</td>
</tr>
<tr>
<td><strong>alpha</strong></td>
<td>Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*nrow(Idt) observations are used for computing the trimmed likelihood. Note that when argument ‘getalpha’ is set to “TwoStep” the final value of ‘alpha’ is estimated by a two-step procedure and the value of argument ‘alpha’ is only used to specify the size of the samples used in the first step. Allowed values are between 0.5 and 1.</td>
</tr>
<tr>
<td><strong>nsamp</strong></td>
<td>Number of subsets used for initial estimates.</td>
</tr>
<tr>
<td><strong>seed</strong></td>
<td>Initial seed for random generator, like <code>.Random.seed</code>, see <code>rrcov.control</code>.</td>
</tr>
<tr>
<td><strong>trace</strong></td>
<td>Logical (or integer) indicating if intermediate results should be printed; defaults to FALSE.</td>
</tr>
<tr>
<td><strong>use.correction</strong></td>
<td>Whether to use finite sample correction factors; defaults to TRUE.</td>
</tr>
<tr>
<td><strong>ncsteps</strong></td>
<td>The maximum number of concentration steps used each iteration of the fasttle algorithm.</td>
</tr>
<tr>
<td><strong>getalpha</strong></td>
<td>Argument specifying if the ‘alpha’ parameter (roughly the percentage of the sample used for computing the trimmed likelihood) should be estimated from the data, or if the value of the argument ‘alpha’ should be used instead. When set to “TwoStep”, ‘alpha’ is estimated by a two-step procedure with the value of argument ‘alpha’ specifying the size of the samples used in the first step. Otherwise, with the value of argument ‘alpha’ is used directly.</td>
</tr>
<tr>
<td><strong>rawMD2Dist</strong></td>
<td>The assumed reference distribution of the raw MCD squared distances, which is used to find to cutoffs defining the observations kept in one-step reweighted MCD estimates. Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual Qui-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005).</td>
</tr>
<tr>
<td><strong>MD2Dist</strong></td>
<td>The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual Qui-squared, or the Beta and F distributions proposed by Cerioli (2010).</td>
</tr>
<tr>
<td><strong>eta</strong></td>
<td>Nominal size for the null hypothesis that a given observation is not an outlier. Defines the raw MCD Mahalanobis distances cutoff used to choose the observations kept in the reweightening step.</td>
</tr>
<tr>
<td><strong>multiCmpCor</strong></td>
<td>Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at ‘eta’ nominal level. ‘always’ – testing all n entities at 1.- (1-’eta’^n/(1/n)); and ‘iterstep’ – use the iterated rule proposed by Cerioli (2010), i.e., make an initial set of tests using the nominal size 1.- (1-’eta’^n/(1/n)), and if no outliers are detected stop. Otherwise, make a second step testing for outliers at the ‘eta’ nominal level.</td>
</tr>
</tbody>
</table>
**fasttle-methods**

getkdblstar  Argument specifying the size of the initial small (in order to minimize the probability of outliers) subsets. If set to the string “Twopplusone” (default) the initial sets have twice the number of interval-value variables plus one (i.e., they are the smaller samples that lead to a non-singular covariance estimate). Otherwise, an integer with the size of the initial sets.

outlin  The type of outliers to be considered. “MidPandLogR” if outliers may be present in both MidPpoints and LogRanges, “MidP” if outliers are only present in MidPpoints, or “LogR” if outliers are only present in LogRanges.

trialmethod  The method to find a trial subset used to initialize each replication of the fasttle algorithm. The current options are “simple” (default) that simply selects ‘kdblstar’ observations at random, and “Poolm” that divides the original sample into ‘m’ non-overlapping subsets, applies the ‘simple trial’ and the refinement methods to each one of them, and merges the results into a trial subset.

m  Number of non-overlapping subsets used by the trial method when the argument of ‘trialmethod’ is set to ‘Poolm’.

reweighted  Should a (Re)weighted estimate of the covariance matrix be used in the computation of the trimmed likelihood or just a “raw” covariance estimate; default is (Re)weighting.

otptyp e  The amount of output returned by fasttle. Current options are “OnlyEst” (default) where only an ‘IdtE’ object with the fasttle estimates is returned, “SetMD2andEst” which returns a list with an ‘IdtE’ object of fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and “SetMD2EstandPrfSt” which returns a list with the previous three components plus a list of some performance statistics concerning the algorithm execution.

control  a list with estimation options - this includes those above provided in the function specification. See RobEstControl for the defaults. If control 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.

...  Further arguments to be passed to internal functions of fasttle.

**Value**

If argument ‘otptype’ is set to “OnlyEst”, an object of class IdtE with the fasttle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

If argument ‘otptype’ is set to “SetMD2andEst” a list with the following components:

sol  An object of class IdtE with the fasttle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

rawset  A vector with the trimmed subset elements used to compute the raw (not reweighted) MCD covariance estimate for the chosen configuration.

rewhtdset  A vector with the final trimmed subset elements used to compute the fasttle estimates.
RobMD2  A vector with the robust squared Mahalanobis distances used to select the trimmed subset.
cnp2  A vector of length two containing the consistency correction factor and the finite sample correction factor of the final estimate of the covariance matrix.
raw.cov  A matrix with the raw MCD estimator used to compute the robust squared Mahalanobis distances of RobMD2.
raw.cnp2  A vector of length two containing the consistency correction factor and the finite sample correction factor of the raw estimate of the covariance matrix.

If argument `otpType` is set to “SetMD2EstandPrfSt” a list with the following components:

sol  An object of class `IdtE` with the fasttle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.
rawSet  A vector with the trimmed subset elements used to compute the raw (not reweighted) MCD covariance estimate for the chosen configuration.
RewghtdSet  A vector with the final trimmed subset elements used to compute the fasttle estimates.
RobMD2  A vector with the robust squared Mahalanobis distances used to select the trimmed subset.
cnp2  A vector of length two containing the consistency correction factor and the finite sample correction factor of the final estimate of the covariance matrix.
raw.cov  A matrix with the raw MCD estimator used to compute the robust squared Mahalanobis distances of RobMD2.
raw.cnp2  A vector of length two containing the consistency correction factor and the finite sample correction factor of the raw estimate of the covariance matrix.
PerfSt  A list with the following components:

**RepSteps**: A list with one component by Covariance Configuration, containing a vector with the number of refinement steps performed by the fasttle algorithm by replication.

**RepLogLik**: A list with one component by Covariance Configuration, containing a vector with the best log-likelihood found be fasttle algorithm by replication.

**StpLogLik**: A list with one component by Covariance Configuration, containing a matrix with the evolution of the log-likelihoods found be fasttle algorithm by replication and refinement step.

**References**


**See Also**

`fulltle`, `RobEstControl`, `getIdtOutl`

**Examples**

```r
## Not run:
# Create an Interval-Data object containing the intervals of temperatures by quarter
# for 899 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8])

# Estimate parameters by the fast trimmed maximum likelihood estimator,
# using a two-step procedure to select the trimming parameter, a reweighted
# MCD estimate, and the classical 97.5% chi-squared quantile cut-offs.

Chinafasttle1 <- fasttle(ChinaT)
cat("China maximum trimmed likelihood estimation results =\n")
print(Chinafasttle1)

# Estimate parameters by the fast trimmed maximum likelihood estimator, using
# the trimming parameter that maximizes breakdown, and a reweighted MCD estimate
# based on the 97.5% quantiles of Hardin and Rocke adjusted F distributions.

Chinafasttle2 <- fasttle(ChinaT, alpha=0.5, getalpha=FALSE, rawMD2Dist="HardRockeAdjF")
cat("China maximum trimmed likelihood estimation results =\n")
print(Chinafasttle2)

# Estimate parameters by the fast trimmed maximum likelihood estimator, using a two-step procedure
# to select the trimming parameter, a reweighted MCD estimate based on Hardin and Rocke adjusted
# F distributions, and 95% quantiles, and the Cerioli Beta and F distributions together
# with Cerioli iterated procedure to identify outliers in the first step.

Chinafasttle3 <- fasttle(ChinaT, rawMD2Dist="HardRockeAdjF", eta=0.05, MD2Dist="CerioliBetaF",
                        multiCmpCor="iterstep")
cat("China maximum trimmed likelihood estimation results =\n")
print(Chinafasttle3)
```
## fulltle-methods

### Methods for Function `fulltle` in Package `MAINT.Data`

**Description**

Performs maximum trimmed likelihood estimation by an exact algorithm (full enumeration of all k-trimmed subsets)

**Usage**

```r
fulltle(Idt, CovCase=1:4, SelCrit=c("BIC","AIC"), alpha=0.75, use.correction=TRUE, getalpha="TwoStep", rawMD2Dist=c("ChiSq","HardRockeAsF","HardRockeAdjF"), MD2Dist=c("ChiSq","CerioliBetaF"), eta=0.025, multiCmpCor=c("never","always","iterstep"), outlin=c("MidPandLogR","MidP","LogR"), reweighted=TRUE, otpType=c("OnlyEst","SetMD2andEst"), force=FALSE, ...)
```

**Arguments**

- `Idt`: An `IData` object representing interval-valued entities.
- `CovCase`: Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
- `SelCrit`: The model selection criterion.
- `alpha`: Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly `alpha*nrow(Idt)` observations are used for computing the trimmed likelihood. Note that when argument ‘getalpha’ is set to “TwoStep” the final value of ‘alpha’ is estimated by a two-step procedure and the value of argument ‘alpha’ is only used to specify the size of the samples used in the first step. Allowed values are between 0.5 and 1.
- `use.correction`: whether to use finite sample correction factors; defaults to TRUE.
- `getalpha`: Argument specifying if the ‘alpha’ parameter (roughly the percentage of the sample used for computing the trimmed likelihood) should be estimated from the data, or if the value of the argument ‘alpha’ should be used instead. When set to “TwoStep”, ‘alpha’ is estimated by a two-step procedure with the value of argument ‘alpha’ specifying the size of the samples used in the first step. Otherwise, with the value of argument ‘alpha’ is used directly.
- `rawMD2Dist`: The assumed reference distribution of the raw MCD squared distances, which is used to find cutoffs defining the observations kept in one-step reweighted MCD estimates. Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual Qui-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005).
**MD2Dist**
The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual Qui-squared, and the Beta and F distributions proposed by Cerioli (2010).

**eta**
Nominal size of the null hypothesis that a given observation is not an outlier. Defines the raw MCD Mahalanobis distances cutoff used to choose the observations kept in the reweighting step.

**multiCmpCor**
Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at the ‘eta’ nominal level. ‘always’ – testing all n entities at 1.- (1.-’eta’^-(1/n)); and ‘iterstep’ – use the iterated rule proposed by Cerioli (2010), i.e., make an initial set of tests using the nominal size 1.- (1.-’eta’^-(1/n)), and if no outliers are detected stop. Otherwise, make a second step testing for outliers at the ‘eta’ nominal level.

**outlin**
The type of outliers to be consideres. “MidPandLogR” if outliers may be present in both MidPpoints and LogRanges, “MidP” if outliers are only present in MidPpoints, or “LogR” if outliers are only present in LogRanges.

**reweighted**
should a (Re)weighted estimate of the covariance matrix be used in the computation of the trimmed likelihood or just a “raw” covariance estimate; default is (Re)weighting.

**otpType**
The amount of output returned by fulltle. Current options are “OnlyEst” (default) where only an ‘IdtE’ object with the fulltle estimates is returned and “SetMD2andEst” which returns a list with an ‘IdtE’ object of fulltle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances.

**force**
A boolean flag indicating whether, for moderate or large data sets the algorithm should proceed anyway, regardless of an expected long execution time, due to exponential explosions in the number of different subsets that need to be evaluated by fulltle

Further arguments to be passed to internal functions of ‘fulltle’.

**Value**
If argument ‘otpType’ is set to “OnlyEst”, an object of class `IdtE` with the fulltle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

If argument ‘otpType’ is set to “SetMD2andEst” a list with the following components:

**sol**
An object of class `IdtE` with the fulltle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

**RewghtdSet**
A vector with the final trimmed subset elements used to compute the fasttle estimates.

**RobMD2**
A vector with the robust squared Mahalanobis distances used to select the trimmed subset.
cnp2  A vector of length two containing the consistency correction factor and the finite sample correction factor of the final estimate of the covariance matrix.

raw.cov  A matrix with the raw MCD estimator used to compute the robust squared Mahalanobis distances of RobMD2.

raw.cnp2  A vector of length two containing the consistency correction factor and the finite sample correction factor of the raw estimate of the covariance matrix.

References


See Also

`fasttle`, `getIdOutl`

Examples

```r
# Create an Interval-Data object containing the intervals for characteristics # of 27 cars models.

CarsIdt <- IData(Cars[1:8], VarNames=c("Price","EngineCapacity","TopSpeed","Acceleration"))

# Display the first and last observations

print(head(CarsIdt))

print(tail(CarsIdt))

# Estimate parameters by the full trimmed maximum likelihood estimator, # using a two-step procedure to select the trimming parameter, a reweighted # MCD estimate, and the classical 97.5% chi-squared quantile cut-offs.

CarsTE1 <- fulltle(CarsIdt)
cat("Cars data -- normal maximum trimmed likelihood estimation results:\n")
print(CarsTE1)

## Not run:
```
# Estimate parameters by the full trimmed maximum likelihood estimator, using
# the trimming parameter that maximizes breakdown, and a reweighted MCD estimate
# based on the 97.5% quantiles of Hardin and Rocke adjusted F distributions.

carsTE2 <- fulltte(carsIdt, alpha=0.5, getalpha=FALSE, rawMD2Dist="HardRockeAdjF")
cat("Cars data -- normal maximum trimmed likelihood estimation results:\n")
print(carsTE2)

# Estimate parameters by the full trimmed maximum likelihood estimator, using
# a two-step procedure to select the trimming parameter, and a reweighted MCD estimate
# based on Hardin and Rocke adjusted F distributions, 97.5% quantiles, and
# the Cerioli Beta and F distributions together with his iterated procedure
# to identify outliers in the first step.

carsTE3 <- fulltte(carsIdt, rawMD2Dist="HardRockeAdjF", eta=0.05, MD2Dist="CerioliBetaF",
                   multiCmpCor="iterstep")
cat("Cars data -- normal maximum trimmed likelihood estimation results:\n")
print(carsTE3)

## End(Not run)

### getIdtOutl

#### Get Interval Data Outliers

**Description**

Identifies outliers in a data set of Interval-valued variables

**Usage**

getIdtOutl(Idt, IdtE=NULL, muE=NULL, SigE=NULL, 
eta=0.025, Rewind=IdtE$RewghtdSet, m=length(Rewind),
RefDist=c("ChiSq","HardRockeAdjF","HardRockeAsF","CerioliBetaF"),
multiCmpCor=c("never","always","iterstep"),
outlin=c("MidPandLogR","MidP","LogR"))

**Arguments**

- **Idt** An IData object representing interval-valued entities.
- **IdtE** Ao object of class IdtSngNDRE or IdtSngNDE containing mean and covariance estimates.
- **muE** Vector with the mean estimates used to find Mahalanobis distances. When specified, it overrides the mean estimate supplied in “IdtE”.
- **SigE** Matrix with the covariance estimates used to find Mahalanobis distances. When specified, it overrides the covariance estimate supplied in “IdtE”.
- **eta** Nominal size of the null hypothesis that a given observation is not an outlier.
Rewind

A vector with the subset of entities used to compute trimmed mean and covariance estimates when using a reweighted MCD. Only used when the ‘RefDist’ argument is set to “CerioliBetaF.”

m

Number of entities used to compute trimmed mean and covariance estimates when using a reweighted MCD. Not used when the ‘RefDist’ argument is set to “ChiSq.”

multiCmpCor

Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at the ‘eta’ nominal level. ‘always’ – testing all n entities at 1.- (1.-‘eta’^(1/n)); and ‘iterstep’ – use the iterated rule proposed by Cerioli (2010), i.e., make an initial set of tests using the nominal size 1.- (1.-‘eta’^(1/n)), and if no outliers are detected stop. Otherwise, make a second step testing for outliers at the ‘eta’ nominal level.

RefDist

The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq”, “HardRockeAsF”, “HardRockeAdjF” and “CerioliBetaF”, respectively for the usual Qui-squared, the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005), and the Beta and F distributions proposed by Cerioli (2010).

outlin

The type of outliers to be considered. “MidPandLogR” if outliers may be present in both MidPpoints and LogRanges, “MidP” if outliers are only present in MidP-points, or “LogR” if outliers are only present in LogRanges.

Value

A vector with the indices of the entities identified as outliers.

References


See Also

data, fulltle

IData

Interval Data objects

Description

IData creates IData objects from data frames of interval bounds or MidPoint/LogRange values of the interval-valued observations.
### Usage

```r
IData(Data, 
    Seq = c("LbUb_VarbyVar", "MidPLogR_VarbyVar", "AllLb_AllUb", "AllMidP_AllLogR"), 
    VarNames=NULL, ObsNames=row.names(Data))
```

### Arguments

- **Data**: a data frame of interval bounds or MidPoint/LogRange values.
- **Seq**: the format of ‘Data’ data frame. Available options are:
  - “LbUb_VarbyVar”: lower bounds followed by upper bounds, variable by variable.
  - “MidPLogR_VarbyVar”: MidPoints followed by LogRanges, variable by variable.
  - “AllLb_AllUb”: all lower bounds followed by all upper bounds, in the same variable order.
  - “AllMidP_AllLogR”: all MidPoints followed all LogRanges, in the same variable order.
- **VarNames**: An optional vector of names to be assigned to the Interval-Valued Variables.
- **ObsNames**: An optional vector of names assigned to the individual observations.

### Details

Objects of class `IData` describe a data set of ‘NObs’ observations on ‘NIVar’ Interval-valued variables. This function creates an interval-data object from a data-frame with either the lower and upper bounds of the observed intervals or by their midpoints and log-ranges.

### See Also

`IData`

### Examples

```r
ChinaT <- IData(ChinaTemp[1:8], VarNames=c("T1","T2","T3","T4"))
cat("Summary of the ChinaT IData object:\n") ; print(summary(ChinaT))
cat("ChinaT first ant last observations:\n")
print(head(ChinaT,n=3))
cat("\n...
")
print(tail(ChinaT,n=3))
```

### Description

A data-array of interval-valued data is an array where each of the NObs rows, corresponding to each entity under analysis, contains the observed intervals of the NIVar descriptive variables.
Slots

MidR: A data-frame of the midpoints of the observed intervals
LogR: A data-frame of the logarithms of the ranges of the observed intervals
ObsNames: An optional vector of names assigned to the individual observations.
VarNames: An optional vector of names to be assigned to the Interval-Valued Variables.
N0bs: Number of entities under analysis (cases)
NIVar: Number of interval variables

Methods

show signature(object = "IData"): show S4 method for the IData-class.
naow signature(x = "IData"): returns the number of entities (observations).
nocol signature(x = "IData"): returns the number of Interval Variables.
rownames signature(x = "IData"): returns the row (entity) names for an object of class IData.
colnames signature(x = "IData"): returns column (variable) names for an object of class IData.
head signature(x = "IData"): head S4 method for the IData-class.
tail signature(x = "IData"): tail S4 method for the IData-class.
mle signature(x = "IData"): Maximum likelihood estimation.
fastt1le signature(x = "IData"): Fast trimmed maximum likelihood estimation.
fullt1le signature(x = "IData"): Exact trimmed maximum likelihood estimation.
RobMxtDEst signature(x = "IData"): Robust estimation of distribution mixtures for interval-valued data.
MANOVA signature(x = "IData"): MANOVA tests on the interval-valued data.
lda signature(x = "IData"): Linear Discriminant Analysis using maximum likelihood parameter estimates of Gaussian mixtures.
qda signature(x = "IData"): Quadratic Discriminant Analysis using maximum likelihood parameter estimates of Gaussian mixtures.
Roblda signature(x = "IData"): Linear Discriminant Analysis using robust estimates of location and scatter.
Robqda signature(x = "IData"): Quadratic Discriminant Analysis using robust estimates of location and scatter.
snda signature(x = "IData"): Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>
References


See Also

`IData, mle, fasttle, fulltle, RobMxtDEst, MANOVA, lda, qda, Roblda, Robqda`

IdtE-class  

**Description**

IdtE contains estimation results for the models assumed for single distributions, or mixtures of distributions, underlying data sets of interval-valued entities. *IdtSngDE* extends IdtE assuming that the data can be characterized by a unique distribution, *i.e.*, not considering partitions of entities into different classes.

**Slots**

- **ModelNames**: The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)
- **ModelType**: Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented
- **ModelConfig**: Configuration of the variance-covariance matrix: Case 1 through Case 4
- **NIVar**: Number of interval variables
- **SelCrit**: The model selection criterion; currently, AIC and BIC are implemented
- **logLik**: The logarithms of the likelihood function for the different cases
- **AICs**: Value of the AIC criterion
- **BICs**: Value of the BIC criterion
- **BestModel**: Bestmodel indicates the best model according to the chosen selection criterion
- **SngD**: Boolean flag indicating whether a single or a mixture of distribution were estimated
Methods

**BestModel** signature(Idt = "IdtE"): Selects the best model according to the chosen selection criterion (currently, AIC or BIC)

**show** signature(object = "IdtE"): show S4 method for the IDtE-class

**testMod** signature(Idt = "IdtE"): Performs statistical likelihood-ratio tests that evaluate the
goodness-of-fit of a nested model against a more general one.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References


See Also

mle, fasttle, fulltle, MANOVA, RobMxtDEst, IData

---

**Idtlda-class**

Class "Idtlda"

Description

Idtlda contains the results of Linear Discriminant Analysis for the interval data

Slots

prior: Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.

means: Matrix with the mean vectors for each group

scaling: Matrix which transforms observations to discriminant functions, normalized so that within groups covariance matrix is spherical.

N: Number of observations

CovCase: Configuration case of the variance-covariance matrix: Case 1 through Case 4

Methods

**predict** signature(object = "Idtlda"): Classifies interval-valued observations in conjunction with lda.

**show** signature(object = "Idtlda"): show S4 method for the IDtlda-class

**CovCase** signature(object = "Idtlda"): Returns the configuration case of the variance-covariance matrix
Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References


See Also

`qda`, `MANOVA`, `Roblda`, `Robqda`, `snda`, `Idata`

---

**IdtMANOVA-class**

*Class IdtMANOVA*

**Description**

IdtMANOVA extends `LRTest` directly, containing the results of MANOVA tests on the interval-valued data. This class is not used directly, but is the basis for different specializations according to the model assumed for the distribution in each group. In particular, the following specializations of IdtMANOVA are currently implemented:

- **IdtClMANOVA** extends IdtMANOVA, assuming a classical (i.e., homocedastic gaussian) setup.
- **IdtHetMANOVA** extends IdtMANOVA, assuming a heterocedastic gaussian set-up.
- **IdtLocSNMANOVA** extends IdtMANOVA, assuming a Skew-Normal location model set-up.
- **IdtLocNSNMANOVA** extends IdtMANOVA, assuming either a homocedastic gaussian or Skew-Normal location model set-up.
- **IdtGenSNMANOVA** extends IdtMANOVA, assuming a Skew-Normal general model set-up.
- **IdtGenNSNMANOVA** extends IdtMANOVA, assuming either a heterocedastic gaussian or Skew-Normal general model set-up.
Slots

NIVar: Number of interval variables.
grouping: Factor indicating the group to which each observation belongs to.
H0res: Model estimates under the null hypothesis.
H1res: Model estimates under the alternative hypothesis.
QuiSq: Inherited from class LRTest. Value of the Qui-Square statistics corresponding to the performed test.
df: Inherited from class LRTest. Degrees of freedom of the Qui-Square statistics.
pvalue: Inherited from class LRTest. p-value of the Qui-Square statistics value, obtained from the Qui-Square distribution with df degrees of freedom.
H0logLik: Inherited from class LRTest. Logarithm of the Likelihood function under the null hypothesis.
H1logLik: Inherited from class LRTest. Logarithm of the Likelihood function under the alternative hypothesis.

Methods

show signature(object = "IdtMANOVA"): show S4 method for the IdtMANOVA-classes.
H0res signature(object = "IdtMANOVA"): retrieves the model estimates under the null hypothesis.
H1res signature(object = "IdtMANOVA"): retrieves the model estimates under the alternative hypothesis.
lda signature(x = "IdtCMANOVA"): Linear Discriminant Analysis using the estimated model parameters.
lda signature(x = "IdtLocNSMANOVA"): Linear Discriminant Analysis using the estimated model parameters.
qda signature(x = "IdtHetNSMANOVA"): Quadratic Discriminant Analysis using the estimated model parameters.
qda signature(x = "IdtGenNSMANOVA"): Quadratic Discriminant Analysis using the estimated model parameters.
snda signature(x = "IdtLocNSMANOVA"): Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a "location" model (i.e., groups differ only in location parameters).
snda signature(x = "IdtGenNSMANOVA"): Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a general model (i.e., groups differ in all parameters).
snda signature(x = "IdtGenNSMANOVA"): Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a general model (i.e., groups differ in all parameters).

Extends

Class LRTest, directly.
IdtMxE-class

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References


See Also

MANOVA, lda, qda, snda, IData

IdtMxE-class  

Description

IdtMxE extends the IdtE class, assuming that the data can be characterized by a mixture of distributions, for instances considering partitions of entities into different groups.

Slots

grouping: Factor indicating the group to which each observation belongs to
ModelNames: Inherited from class IdtE. The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)
ModelType: Inherited from class IdtE. Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented.
ModelConfig: Inherited from class IdtE. Configuration of the variance-covariance matrix: Case 1 through Case 4
NIVar: Inherited from class IdtE. Number of interval variables
SelCrit: Inherited from class IdtE. The model selection criterion; currently, AIC and BIC are implemented
logLik: Inherited from class IdtE. The logarithms of the likelihood function for the different cases
AICs: Inherited from class IdtE. Value of the AIC criterion
BICs: Inherited from class IdtE. Value of the BIC criterion
BestModel: Inherited from class IdtE. Bestmodel indicates the best model according to the chosen selection criterion
SngD: Inherited from class IdtE. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class "IdtMxE"

Ngrps: Number of mixture components
**IdtMxNandSNDE-class**

**Description**

IdtMxNandSNDE contains the results of a mixture model estimation; Normal an Skew-Normal models are considered, with the four different possible variance-covariance configurations.

**Slots**

- **nmod**: Estimates of the mixture model for the Gaussian case
- **snmod**: Estimates of the mixture model for the Skew-Normal case
- **grouping**: Inherited from class *IdtMxE*. Factor indicating the group to which each observation belongs to
- **ModelNames**: Inherited from class *IdtE*. The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)
- **ModelType**: Inherited from class *IdtE*. Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented
- **ModelConfig**: Inherited from class *IdtE*. Configuration case of the variance-covariance matrix: Case 1 through Case 4
- **nivar**: Inherited from class *IdtE*. Number of interval variables
- **selcrit**: Inherited from class *IdtE*. The model selection criterion; currently, AIC and BIC are implemented

**Extends**

Class *IdtE*, directly.

**Methods**

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

**Author(s)**

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

**References**


**See Also**

*IdtE, IdtSngDE, IData, MANOVA, RobMxtDEst*
logLiks: Inherited from class IdtE. The logarithms of the likelihood function for the different cases

AICs: Inherited from class IdtE. Value of the AIC criterion

BICs: Inherited from class IdtE. Value of the BIC criterion

BestModel: Inherited from class IdtE. Indicates the best model according to the chosen selection criterion

SngD: Inherited from class IdtE. Boolean flag indicating whether a single or a mixture of distributions were estimated. Always set to FALSE in objects of class IdtMxNandSNDE

Ngrps: Inherited from class IdtMxE. Number of mixture components

**Extends**

Class IdtMxE, directly. Class IdtE, by class IdtMxE, distance 2.

**Methods**

No methods defined with class IdtMxNandSNDE in the signature.

**Author(s)**

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

**References**


**See Also**

IdtE, IdtMxE, IdtSngNandSNDE, MANOVA, RobMxtDEst, IData
Slots

hmcdt: Indicates whether we consider an homocedastic (TRUE) or a hetereocedasic model (FALSE)

mlenmue: Matrix with the maximum likelihood mean vectors estimates by group (each row refers to a group)

mlenmuese: Matrix with the maximum likelihood means' standard errors by group (each row refers to a group)

CovConfCases: List of the considered configurations

grouping: Inherited from class IdtMxE. Factor indicating the group to which each observation belongs to

ModelNames: Inherited from class IdtE. The model acronym formed by a "N", indicating a Normal model, followed by the configuration (Case 1 through Case 4)

ModelType: Inherited from class IdtE. Indicates the model; always set to "Normal" in objects of the IdtMxNDE class

ModelConfig: Inherited from class IdtE. Configuration case of the variance-covariance matrix: Case 1 through Case 4

NIVar: Inherited from class IdtE. Number of interval variables

SelCrit: Inherited from class IdtE. The model selection criterion; currently, AIC and BIC are implemented

logLik: Inherited from class IdtE. The logarithms of the likelihood function for the different cases

AICs: Inherited from class IdtE. Value of the AIC criterion

BICs: Inherited from class IdtE. Value of the BIC criterion

BestModel: Inherited from class IdtE. Indicates the best model according to the chosen selection criterion

SngD: Inherited from class IdtE. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class IdtMxNDE

ngrps: Inherited from class IdtMxE. Number of mixture components

Extends

Class IdtMxE, directly. Class IdtE, by class IdtMxE, distance 2.

Methods

lda signature(x = "IdtMxNDE"): Linear Discriminant Analysis using the estimated model parameters.

qda signature(x = "IdtMxNDE"): Quadratic Discriminant Analysis using the estimated model parameters.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>
IdtMxNDRE-class

References

See Also
IdtE, IdtMxE, IdtMxNDRE, IdtSngNDE, IData, MANOVA

IdtMxNDRE-class  Class IdtMxNDE

Description
IdtMxNDRE contains the results of a mixture Normal model robust parameter estimation, with the four different possible variance-covariance configurations.

Slots
hmcdt: Indicates whether we consider an homocedastic (TRUE) or a heterocedastic model (FALSE)
RobNmuE: Matrix with the robust mean vectors estimates by group (each row refers to a group)
CovConfCases: List of the considered configurations
grouping: Inherited from class IdtMxE. Factor indicating the group to which each observation belongs to
ModelNames: Inherited from class IdtE. The model acronym formed by a ”N”, indicating a Normal model, followed by the configuration (Case 1 through Case 4)
ModelType: Inherited from class IdtE. Indicates the model; always set to ”Normal” in objects of the IdtMxNDRE class
ModelConfig: Inherited from class IdtE. Configuration case of the variance-covariance matrix: Case 1 through Case 4
NIVar: Inherited from class IdtE. Number of interval variables
SelCrit: Inherited from class IdtE. The model selection criterion; currently, AIC and BIC are implemented
logLik: Inherited from class IdtE. The logarithms of the likelihood function for the different cases
AICs: Inherited from class IdtE. Value of the AIC criterion
BICs: Inherited from class IdtE. Value of the BIC criterion
BestModel: Inherited from class IdtE. Indicates the best model according to the chosen selection criterion
SngD: Inherited from class IdtE. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class IdtMxNDRE
ngrps: Inherited from class IdtMxE. Number of mixture components
IdtMxSNDE-class

**Extends**
Class IdtMxE, directly. Class IdtE, by class IdtMxE, distance 2.

**Methods**
No methods defined with class IdtMxNDRE in the signature.

**Author(s)**
Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

**References**


**See Also**
IdtE, IdtMxE, IdtMxNDE, IdtSngNDRE, RobMxtDest, IData

---

**IdtMxSNDE-class**

**Class IdtMxSNDE**

**Description**
IdtMxSNDE contains the results of a mixture model estimation for the Skew-Normal model, with the four different possible variance-covariance configurations.

**Slots**
- **hmcdt**: Indicates whether we consider an homocedastic location model (TRUE) or a general model (FALSE)
- **CovConfCases**: List of the considered configurations
- **grouping**: Inherited from class IdtMxE. Factor indicating the group to which each observation belongs to
- **ModelNames**: Inherited from class IdtE. The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)
- **ModelType**: Inherited from class IdtE. Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented
- **ModelConfig**: Inherited from class IdtE. Configuration case of the variance-covariance matrix: Case 1 through Case 4
IdtMxtNDE-class

NIVar: Inherited from class IdtE. Number of interval variables
SelCrit: Inherited from class IdtE. The model selection criterion; currently, AIC and BIC are implemented
logLik: Inherited from class IdtE. The logarithms of the likelihood function for the different cases
AICs: Inherited from class IdtE. Value of the AIC criterion
BICs: Inherited from class IdtE. Value of the BIC criterion
BestModel: Inherited from class IdtE. Indicates the best model according to the chosen selection criterion
SngD: Inherited from class IdtE. Boolean flag indicating whether a single or a mixture of distributions were estimated. Always set to FALSE in objects of class IdtMxSNDE
Ngrps: Inherited from class IdtMxE. Number of mixture components

Extends

Class IdtMxE, directly. Class IdtE, by class IdtMxE, distance 2.

Methods

No methods defined with class IdtMxSNDE in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References


See Also

IdtE, IdtMxE, IdtSngSNDE, MANOVA, IData

IdtMxtNDE-class

Class IdtMxtNDE

Description

IdtMxtNDE is an union of classes IdtMxNDE and IdtMxNDRE, containing the results of mixture Normal model parameter estimation by maximum likelihood (IdtMxNDE) or robust (IdtMxNDRE) methods.

See Also

IdtE, IdtMxE, IdtMxNDE, IdtMxNDRE
IdtNandSNDE-class

Description

IdtNandSNDE is a union of classes \texttt{IdtSngNandSNDE} and \texttt{IdtMxNandSNDE}, used for storing the estimation results of Normal and Skew-Normal modelizations for Interval Data.

Methods

\begin{itemize}
  \item \texttt{coef} signature(coef = "IdtNandSNDE"): extracts parameter estimates from objects of class \texttt{IdtNandSNDE}
  \item \texttt{stdEr} signature(x = "IdtNandSNDE"): extracts standard errors from objects of class \texttt{IdtNandSNDE}
  \item \texttt{vcov} signature(x = "IdtNandSNDE"): extracts an estimate of the variance-covariance matrix of the parameters estimators for objects of class \texttt{IdtNandSNDE}
  \item \texttt{mean} signature(x = "IdtNandSNDE"): extracts the mean vector estimate from objects of class \texttt{IdtNandSNDE}
  \item \texttt{var} signature(x = "IdtNandSNDE"): extracts the variance-covariance matrix estimate from objects of class \texttt{IdtNandSNDE}
  \item \texttt{cor} signature(x = "IdtNandSNDE"): extracts the correlation matrix estimate from objects of class \texttt{IdtNandSNDE}
\end{itemize}

References


See Also

\texttt{IData, mle, fasttle, fulltle, MANOVA, RobMxtDEst, IdtSngNandSNDE, IdtMxNandSNDE}

IdtNDE-class

Description

IdtNDE is a union of classes \texttt{IdtSngNDE}, \texttt{IdtSngNDSRE}, \texttt{IdtMxNDE} and \texttt{IdtMxNDSRE}, used for storing the estimation results of Normal modelizations for Interval Data.
Methods

- `coef` signature: `signature(coef = "IdtNDE")`: extracts parameter estimates from objects of class `IdtNDE`.
- `stdEr` signature: `signature(x = "IdtNDE")`: extracts standard errors from objects of class `IdtNDE`.
- `vcov` signature: `signature(x = "IdtNDE")`: extracts an estimate of the variance-covariance matrix of the parameters estimators for objects of class `IdtNDE`.
- `mean` signature: `signature(x = "IdtNDE")`: extracts the mean vector estimate from objects of class `IdtNDE`.
- `var` signature: `signature(x = "IdtNDE")`: extracts the variance-covariance matrix estimate from objects of class `IdtNDE`.
- `cor` signature: `signature(x = "IdtNDE")`: extracts the correlation matrix estimate from objects of class `IdtNDE`.

References


See Also

- `idtsngnde`, `idtsngndre`, `idtmxnde`, `idtmxndre`, `idtsnde`, `idata`, `mle`, `fasttle`, `fulltle`, `MANOVA`.

Idtqda-class

Class "Idtqda"

Description

Idtqda contains the results of Quadratic Discriminant Analysis for the interval data.

Slots

- `prior`: Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.
- `means`: Matrix with the mean vectors for each group.
- `scaling`: A three-dimensional array. For each group, g, `scaling[,g]` is a matrix which transforms interval-valued observations so that within-groups covariance matrix is spherical.
- `ldet`: Vector of half log determinants of the dispersion matrix.
- `lev`: Levels of the grouping factor.
- `CovCase`: Configuration case of the variance-covariance matrix: Case 1 through Case 4.

Methods

- `predict` signature: `signature(object = "Idtqda")`: Classifies interval-valued observations in conjunction with qda.
- `show` signature: `signature(object = "Idtqda")`: show S4 method for the `Idtqda-class`.
- `CovCase` signature: `signature(object = "Idtqda")`: Returns the configuration case of the variance-covariance matrix.
Author(s)
Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References


See Also
qda, MANOVA, Robqda, IData

---

IdtSNDE-class  
Class "IdtSNDE"

Description
IdtSNDE is a class union of classes IdtSngSNDE and IdtMxSNDE, used for storing the estimation results of Skew-Normal modelizations for Interval Data.

Methods
- **coef** signature(coef = "IdtSNDE"): extracts parameter estimates from objects of class IdtSNDE
- **stdEr** signature(x = "IdtSNDE"): extracts standard errors from objects of class IdtSNDE
- **vcov** signature(x = "IdtSNDE"): extracts an asymptotic estimate of the variance-covariance matrix of the parameters estimators for objects of class IdtSNDE
- **mean** signature(x = "IdtSNDE"): extracts the mean vector estimate from objects of class IdtSNDE
- **var** signature(x = "IdtSNDE"): extracts the variance-covariance matrix estimate from objects of class IdtSNDE
- **cor** signature(x = "IdtSNDE"): extracts the correlation matrix estimate from objects of class IdtSNDE

References

Description

IdtSNgenda contains the results of discriminant analysis for the interval data, based on a general Skew-Normal model.

Slots

prior: Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.
ksi: Matrix with the direct location parameter ("ksi") estimates for each group.
eta: Matrix with the direct scaled skweness parameter ("eta") estimates for each group.
scaling: For each group \( g \), \( \text{scaling}[.,g] \) is a matrix which transforms interval-valued observations so that in each group the scale-association matrix ("Omega") is spherical.
mu: Matrix with the centred location parameter ("mu") estimates for each group.
gamma1: Matrix with the centred skweness parameter ("gamma1") estimates for each group.
ldet: Vector of half log determinants of the dispersion matrix.
lev: Levels of the grouping factor.

Methods

predict signature(object = "IdtSNgenda"): Classifies interval-valued observations in conjunction with snda.

show signature(object = "IdtSNgenda"): show S4 method for the IdtSNgenda-class

CovCase signature(object = "IdtSNgenda"): Returns the configuration case of the variance-covariance matrix

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>
IdtSngNandSNDE-class

References


See Also

MANOVA, snda, IDdata

---

### IdtSngNandSNDE-class

*Class IdtSngNandSNDE*

**Description**

IdtSngNandSNDE contains the results of a single class model estimation for the Normal and the Skew-Normal distributions, with the four different possible variance-covariance configurations.

**Slots**

- `nmod`: Estimates of the single class model for the Gaussian case
- `snmod`: Estimates of the single class model for the Skew-Normal case
- `modelnames`: Inherited from class `IdtE`. The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)
- `modeltype`: Inherited from class `IdtE`. Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented
- `modelconfig`: Inherited from class `IdtE`. Configuration of the variance-covariance matrix: Case 1 through Case 4
- `nivar`: Inherited from class `IdtE`. Number of interval variables
- `selcrit`: Inherited from class `IdtE`. The model selection criterion; currently, AIC and BIC are implemented
- `logliks`: Inherited from class `IdtE`. The logarithms of the likelihood function for the different cases
- `aic`: Inherited from class `IdtE`. Value of the AIC criterion
- `bic`: Inherited from class `IdtE`. Value of the BIC criterion
- `bestmodel`: Inherited from class `IdtE`. Bestmodel indicates the best model according to the chosen selection criterion
- `sng0`: Inherited from class `IdtE`. Boolean flag indicating whether a single or a mixture of distributions were estimated. Always set to TRUE in objects of class IdtSngNandSNDE
**IdtSngNDE-class**

**Extends**

Class `IdtSngDE`, directly. Class `IdtE`, by class `IdtSngDE`, distance 2.

**Methods**

No methods defined with class `IdtSngNandSNDE` in the signature.

**Author(s)**

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

**References**


**See Also**

`IData`, `IdtMxNandSNDE`, `mle`, `fasttle`, `fulltle`
AICs: Inherited from class \texttt{IdtE}. Value of the AIC criterion
BICs: Inherited from class \texttt{IdtE}. Value of the BIC criterion
BestModel: Inherited from class \texttt{IdtE}. Bestmodel indicates the best model according to the chosen selection criterion
SngD: Inherited from class \texttt{IdtE}. Boolean flag indicating whether a single or a mixture of distributions were estimated. Always set to \texttt{TRUE} in objects of class \texttt{IdtSngNDE}

**Extends**

Class \texttt{IdtSngDE}, directly. Class \texttt{IdtE}, by class \texttt{IdtSngDE}, distance 2.

**Methods**

No methods defined with class \texttt{IdtSngNDE} in the signature.

**Author(s)**

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

**References**


**See Also**

\texttt{IData}, \texttt{mle}, \texttt{IdtSngNDRE}, \texttt{IdtSngSNDE}, \texttt{IdtMxNDE}

---

**IdtSngNDRE-class**

*Class \texttt{IdtSngNDRE}*

**Description**

Contains the results of a single class robust estimation for the Normal distribution, with the four different possible variance-covariance configurations.

**Slots**

\texttt{RobNmuE}: Matrix with the maximum likelihood mean vectors estimates
\texttt{CovConfCases}: List of the considered configurations
\texttt{ModelNames}: Inherited from class \texttt{IdtE}. The model acronym formed by a "N", indicating a Normal model, followed by the configuration (Case 1 through Case 4)
\texttt{ModelType}: Inherited from class \texttt{IdtE}. Indicates the model; always set to "Normal" in objects of the \texttt{IdtSngNDRE} class
IdtSngNDRE-class

ModelConfig: Inherited from class IdtE. Configuration of the variance-covariance matrix: Case 1 through Case 4

NIVar: Inherited from class IdtE. Number of interval variables

SelCrit: Inherited from class IdtE. The model selection criterion; currently, AIC and BIC are implemented

logLiks: Inherited from class IdtE. The logarithms of the likelihood function for the different cases

AICs: Inherited from class IdtE. Value of the AIC criterion

BICs: Inherited from class IdtE. Value of the BIC criterion

BestModel: Inherited from class IdtE. Bestmodel indicates the best model according to the chosen selection criterion

SngD: Inherited from class IdtE. Boolean flag indicating whether a single or a mixture of distributions were estimated. Always set to TRUE in objects of class IdtSngNDRE

Extends

Class IdtSngDE, directly. Class IdtE, by class IdtSngDE, distance 2.

Methods

No methods defined with class IdtSngNDRE in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References


See Also

IData, fasttle, fulltle, IdtSngNDE, IdtMxNDRE
IdtSngSNDE-class

**Description**

Contains the results of a single class maximum likelihood estimation for the Skew-Normal distribution, with the four different possible variance-covariance configurations.

**Slots**

- **CovConfCases**: List of the considered configurations
- **ModelNames**: The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration Case (C1 to C4) for the covariance matrix
- **ModelNames**: Inherited from class IdtE. The model acronym formed by a "SN", indicating a skew-normal model, followed by the configuration (Case 1 through Case 4)
- **ModelType**: Inherited from class IdtE. Indicates the model; always set to "SkewNormal" in objects of the IdtSngSNDE class
- **ModelConfig**: Inherited from class IdtE. Configuration case of the variance-covariance matrix: Case 1 through Case 4
- **NIVar**: Inherited from class IdtE. Number of interval variables
- **SelCrit**: Inherited from class IdtE. The model selection criterion; currently, AIC and BIC are implemented
- **logLiks**: Inherited from class IdtE. The logarithms of the likelihood function for the different cases
- **AICs**: Inherited from class IdtE. Value of the AIC criterion
- **BICs**: Inherited from class IdtE. Value of the BIC criterion
- **BestModel**: Inherited from class IdtE. Indicates the best model according to the chosen selection criterion
- **SngD**: Inherited from class IdtE. Boolean flag indicating whether a single or a mixture of distributions were estimated. Always set to TRUE in objects of class IdtSngSNDE

**Extends**

Class IdtSngDE, directly. Class IdtE, by class IdtSngDE, distance 2.

**Methods**

No methods defined with class IdtSngSNDE in the signature.

**Author(s)**

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>
**IdtSNlocda-class**

**References**


**See Also**

`mle`, `IData`, `IdtSngNDE`, `IdtMxSNDE`

---

**Description**

`IdtSNlocda` contains the results of Discriminant Analysis for the interval data, based on a location Skew-Normal model.

**Slots**

- `prior`: Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.
- `ksi`: Matrix with the direct location parameter ("ksi") estimates for each group.
- `eta`: Vector with the direct scaled skewness parameter ("eta") estimates.
- `scaling`: Matrix which transforms observations to discriminant functions, normalized so that the within groups scale-association matrix ("Omega") is spherical.
- `mu`: Matrix with the centred location parameter ("mu") estimates for each group.
- `gamma1`: Vector with the centred skewness parameter ("gamma1") estimates.
- `n`: Number of observations.
- `CovCase`: Configuration case of the variance-covariance matrix: Case 1 through Case 4

**Methods**

- `predict` signature(`object` = "IdtSNlocda"): Classifies interval-valued observations in conjunction with `lda`.
- `show` signature(`object` = "IdtSNlocda"): show S4 method for the `IDdtilda-class`
- `CovCase` signature(`object` = "IdtSNlocda"): Returns the configuration case of the variance-covariance matrix

**Author(s)**

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>
References


See Also

`snda`, `MANOVA`, `IData`

### lda-methods

**Linear Discriminant Analysis of Interval Data**

**Description**

`lda` performs linear discriminant analysis of Interval Data based on classic estimates of a mixture of Gaussian models.

**Usage**

```r
## S4 method for signature 'IData'
lda(x, grouping, prior="proportions", CVtol=1.0e-5, egvtol=1.0e-10,
    subset=1:nrow(x), CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE, ... )

## S4 method for signature 'IdtMxtNDE'
lda(x, prior="proportions", selmodel=BestModel(x), egvtol=1.0e-10,
    silent=FALSE, ... )

## S4 method for signature 'IdtClMANOVA'
lda( x, prior="proportions", selmodel=BestModel(H1res(x)),
    egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IdtLocNSNMANOVA'
lda( x, prior="proportions",
    selmodel=BestModel(H1res(x)@NMod), egvtol=1.0e-10, silent=FALSE, ... )
```

**Arguments**

- `x` An object of class `IData`, `IdtMxtNDE`, `IdtClMANOVA` or `IdtLocNSNMANOVA` with either the original Interval Data, an estimate of a mixture of gaussian models for Interval Data, or the results of an Interval Data MANOVA, from which the discriminant analysis will be based.
grouping Factor specifying the class for each observation.
prior The prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
CVtol Tolerance level for absolute value of the coefficient of variation of non-constant variables. When a MidPoint or LogRange has an absolute value within-groups coefficient of variation below CVtol, it is considered to be a constant.
egvtol Tolerance level for the eigenvalues of the product of the inverse within by the between covariance matrices. When an eigenvalue has an absolute value below egvtol, it is considered to be zero.
subset An index vector specifying the cases to be used in the analysis.
CovCase Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit The model selection criterion.
silent A boolean flag indicating whether a warning message should be printed if the method fails.
selmodel Selected model from a list of candidate models saved in object x.

References


See Also

qda, snda, Roblda, Robqda, IData, IdtMxtNDE, IdtClMANOVA, IdtLocNSMANOVA

Examples

```r
# Create an Interval-Data object containing the intervals for 899 observations
# on the temperatures by quarter in 60 Chinese meteorological stations.
ChinaT <- IData(ChinaTemp[1:8], VarNames=c("T1","T2","T3","T4"))

# Linear Discriminant Analysis
ChinaT.lda <- lda(ChinaT, ChinaTemp$GeoReg)
cat("Temperatures of China -- linear discriminant analysis results:\n")
print(ChinaT.lda)
cat("lda Prediction results:\n")
print(predict(ChinaT.lda, ChinaT)$class)

## Not run:
# Estimate error rates by ten-fold cross-validation replicated 20 times
```
LRTest-class

LRTest contains the results of likelihood ratio tests

Slots

QuiSq: Value of the Qui-Square statistics corresponding to the performed test
df: Degrees of freedom of the Qui-Square statistics
pvalue: p-value of the Qui-Square statistics value, obtained from the Qui-Square distribution with df degrees of freedom
H0logLik: Logarithm of the Likelihood function under the null hypothesis
H1logLik: Logarithm of the Likelihood function under the alternative hypothesis

Methods

show signature(object = "LRTest"): show S4 method for the LRTest-class

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

See Also

mle, IData, ConfTests, MANOVA
MANOVA-methods

Methods for Function MANOVA in Package ‘MAINT.Data’

Description

Function MANOVA performs MANOVA tests based on likelihood ratios allowing for both Gaussian and Skew-Normal distributions and homocedastic or heterocedastic setups. Methods H0res and H1res retrieve the model estimates under the null and alternative hypothesis, and method show displays the MANOVA results.

Usage

MANOVA(Idt, grouping, Model=c("Normal","SKNormal","NrmandSKN"), CovCase=1:4, SelCrit=c("BIC","AIC"), Mxt=c("Hom","Het","Loc","Gen"), CVtol=1.0e-5, OptCtrl=1, onerror=c("stop","warning","silentNull"), ...)

## S4 method for signature 'IdtMANOVA'
H0res(object)
## S4 method for signature 'IdtMANOVA'
H1res(object)
## S4 method for signature 'IdtMANOVA'
show(object)

Arguments

object An object representing a MANOVA analysis on interval-valued entities.
Idt An IData object representing interval-valued entities.
grouping Factor indicating the group to which each observation belongs to.
Model The joint distribution assumed for the MidPoint and LogRanges. Current alternatives are “Normal” for Gaussian distributions, “SKNormal” for Skew-Normal and “NrmandSKN” for both Gaussian and Skew-Normal distributions.
CovCase Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit The model selection criterion.
Mxt Indicates the type of mixing distributions to be considered. Current alternatives are “Hom” (homocedastic) and “Het” (heterocedastic) for Gaussian models, “Loc” (location model – groups differ only on their location parameters) and “Gen” “Loc” (general model – groups differ on all parameters) for Skew-Normal models.
CVtol Tolerance level for absolute value of the coefficient of variation of non-constant variables. When a MidPoint or LogRange has an absolute value within-groups coefficient of variation below CVtol, it is considered to be a constant.
OptCtrl

List of optional control parameters to passed to the optimization routine. See the documentation of RepLOptim for a description of the available options.

onerror

Indicates whether an error in the optimization algorithm should stop the current call, generate a warning, or return silently a NULL object.

...

Other named arguments.

Value

An object of class IdtMANOVA, containing the estimation and test results.

See Also

IdtMANOVA, RepLOptim

Examples

#Create an Interval-Data object containing the intervals of temperatures by quarter # for 899 Chinese meteorological stations.
ChinaT <- IData(ChinaTemp[1:8])

#MANOVA tests assuming that one of C2, C3 or C4 restricted configuration cases hold
#Classical (homocedastic) MANOVA tests
ManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,CovCase=2:4)
cat("China, MANOVA by geografical regions results =\n")
print(ManvChina)

#Heterocedastic MANOVA tests
HetManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,Mxt="Het",CovCase=2:4)
cat("China, heterocedastic MANOVA by geografical regions results =\n")
print(HetManvChina)

#Skew-Normal based MANOVA assuming the the groups differ only according to location parameters
## Not run:
SKNLocManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,Model="SKNormal",Mxt="Loc",CovCase=2:4)
cat("China, Skew-Normal MANOVA (location model) by geografical regions results =\n")
print(SKNLocManvChina)

#Skew-Normal based MANOVA assuming the the groups may differ in all parameters
SKNGenManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,Model="SKNormal",Mxt="Gen",CovCase=2:4)
cat("China, Skew-Normal MANOVA (general model) by geografical regions results =\n")
print(SKNGenManvChina)

## End(Not run)
Description

Performs maximum likelihood estimation for parametric models of interval data

Usage

```r
## S4 method for signature 'IData'
mle(Idt, Model=c("Normal","SNNormal","NrmandSKN"), CovCase=1:4,
    SelCrit=c("BIC","AIC"), OptCtrl=list(), ...)
```

Arguments

- `Idt`: An IData object representing interval-valued entities.
- `Model`: The joint distribution assumed for the MidPoint and LogRanges. Current alternatives are "Normal" for Gaussian, distributions, "SNNormal" for Skew-Normal and "NrmandSKN" for both Gaussian and Skew-Normal distributions.
- `CovCase`: Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
- `SelCrit`: The model selection criterion.
- `OptCtrl`: List of optional control parameters to passed to the optimization routine. See the documentation of RepLOptim for a description of the available options.
- `...`: Other named arguments.

References


See Also

`IData`, `RepLOptim`

Examples

```r
# Create an Interval-Data object containing the intervals of temperatures by quarter # for 899 Chinese meteorological stations.
ChinaT <- IData(ChinaTemp[1:8])

# Estimate parameters by maximum likelihood, assuming that one of # the C2, C3 or C4 restricted Covariance configurations holds
qda-methods

Quadratic Discriminant Analysis of Interval Data

Description

qda performs quadratic discriminant analysis of Interval Data based on classic estimates of a mixture of Gaussian models.

Usage

```r
# S4 method for signature 'IData'
quda(x, grouping, prior="proportions", CVtol=1.0e-5, subset=1:nrow(x),
    CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE, ... )

# S4 method for signature 'IdtMxtNDE'
quda(x, prior="proportions", selmodel=BestModel(x), silent=FALSE, ... )

# S4 method for signature 'IdtHetMANOVA'
quda( x, prior="proportions", selmodel=BestModel(H1res(x)),
    silent=FALSE, ... )

# S4 method for signature 'IdtGenNSMANOVA'
quda( x, prior="proportions",
    selmodel=BestModel(H1res(x)@NMod), silent=FALSE, ... )
```

Arguments

- **x**: An object of class `IData`, `IdtMxtNDE`, `IdtHetMANOVA` or `IdtGenNSMANOVA` with either the original Interval Data, and estimate of a mixture of gaussian models for Interval Data, or the results of a Interval Data heterocedastic MANOVA, from which the discriminant analysis will be based.
- **grouping**: Factor specifying the class for each observation.
- **prior**: The prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
- **CVtol**: Tolerance level for absolute value of the coefficient of variation of non-constant variables. When a MidPoint or LogRange has an absolute value within-groups coefficient of variation below CVtol, it is considered to be a constant.
subset
An index vector specifying the cases to be used in the analysis.

CovCase
Configuration of the variance-covariance matrix: a set of integers between 1 and 4.

SelCrit
The model selection criterion.

silent
A boolean flag indicating whether a warning message should be printed if the method fails.

selmodel
Selected model from a list of candidate models saved in object x.

... Other named arguments.

References

See Also
*lda, snda, Roblda, Robqda, IData, IdtMxtNDE, IdtHetNMANOVA, IdtGenNSNMANOVA*

Examples

```r
# Create an Interval-Data object containing the intervals for 899 observations
# on the temperatures by quarter in 60 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8], VarNames=c("T1","T2","T3","T4"))

# Quadratic Discriminant Analysis

ChinaT.qda <- qda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- qda discriminant analysis results:\n")
print(ChinaT.qda)

## Not run:
# Estimate error rates by ten-fold cross-validation replicated 20 times

CVqda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=qda,CovCase=CovCase(ChinaT.qda))
summary(CVqda[,,"Clerr"])
gberrors <-
apply(CVqda[,,"Nk"]*CVqda[,,"Clerr"],1,sum)/apply(CVqda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(gberrors),"\n")

## End(Not run)
```
Description

p-quantiles of the Hardin and Rocke (2005) scaled F distribution for squared Mahalanobis distances based on raw MCD covariance estimators

Usage

```r
qHardRoqF(p, nob, nvar, h=floor((nobs+nvar+1)/2), adj=TRUE, lower.tail=TRUE, log.p=FALSE)
```

Arguments

- `p` Vector of probabilities.
- `nobs` Number of observations used in the computation of the raw MCD Mahalanobis squared distances.
- `nvar` Number of variables used in the computation of the raw MCD Mahalanobis squared distances.
- `h` Number of observations kept in the computation of the raw MCD estimate.
- `adj` logical; if TRUE (default) returns the quantile of the adjusted distribution. Otherwise returns the quantile of the asymptotic distribution.
- `lower.tail` logical; if TRUE (default), probabilities are P(X <= x) otherwise, P(X > x)
- `log.p` logical; if TRUE, probabilities p are given as log(p).

Value

The quantile of the appropriate scaled F distribution.

References


See Also

`fasttle`, `fulltle`
Description

‘RepLOptim’ Tries to minimize a function calling local optimizers several times from different random starting points.

Usage

RepLOptim(start, parsd, fr, gr=NULL, inphess=NULL, ..., method="nlminb",
lower=NULL, upper=NULL, rethess=FALSE, parmstder=FALSE, control=list())

Arguments

start       Vector of starting points used in the first call of the local optimizer.
parsd       Vector of standard deviations for the parameter distribution generating starting points for the local optimizer.
fr          The function to be minimized. If method is neither “nlminb” or “L-BFGS-B”, fr should accept a lbound and an ubound arguments for the parameter bounds, and should enforce these bounds before calling the local optimization routine.
gr          A function to return the gradient for the “nlminb”, “BFGS”, “CG” and L-BFGS-B methods. If it is ‘NULL’, a finite-difference approximation will be used. For the “SANN” method it specifies a function to generate a new candidate point. If it is ‘NULL’ a default Gaussian Markov kernel is used.
inphess     A function to return the hessian for the “nlminb” method. Must return a square matrix of order ‘length(parmean)’ with the different hessian elements in its lower triangle. It is ignored if method component of the control list is not set to its “nlminb” default.
...          Further arguments to be passed to ‘fr’, ‘gr’ and ‘inphess’.
method      The method to be used. See ‘Details’.
lower       Vector of parameter lower bounds. Set to ‘-Inf’ (no bounds) by default.
upper       Vector of parameter upper bounds. Set to ‘Inf’ (no bounds) by default.
rethess     Bolean flag indicating wether a numerically evaluated hessian matrix at the optimum should be computed and returned. Not available for the “nlminb” method.
parmstder   Bolean flag indicating wether parameter assymptotic standard errors based on the inverse hessian approximation to the Fisher information matrix should be computed and returned. Only available if hessian is set to TRUE and if a local minimum with a positive-definite hessian was indeed found. This requirement may fail if ‘nrep’ and ‘niter’ (and maybe ‘neval’) are not large enough, and for non-trivial problems of moderate or high dimensionality may never be satisfied because of numerical difficulties.
control     A list of control parameters. See below for details.
Details

‘RepLOptim’ Tries to minimize a function by calling local optimizers several times from different starting points. The starting point used in the first call the the local optimizer is the value of the argument ‘start’. Subsequent calls use starting points generated from uniform distributions of independent variates with means equal to the current best parameter values and standard deviations equal to the values of the argument ‘parsd’. If parameter bounds are specified and the uniform limits implied by ‘parsd’ violate those bounds, these limits are replaced by the corresponding bounds.

The choice of the local optimizer is made by value of the ‘method’ argument. This argument can be a function object implementing the optimizer or a string describing an available R method. In the latter case current alternatives are: “nlminb” (default) for the ‘nlminb’ port routine, “nlm” for the ‘nlm’ function and “Nelder-Mead”, “L-BFGS-B”, “CG”, “L-BFGS-B” and “SANN” for the corresponding methods of the ‘optim’ function.

Arguments for controlling the behaviour of the local optimizer can be specified as components of control list. This list can include any of the following components:

maximum Maximum time of repetitions of the same minimum objective value, before RepLOptim is stopped and the current best solution is returned. By default set to 2.

maximum Maximum number of times the local optimizer is called without improvements in the minimum objective value, before RepLOptim is stopped and the current best solution is returned. By default set to 50.

maximum Maximum number of times the local optimizer is called and returns a valid solution before RepLOptim is stopped and the current best solution is returned. By default set to 250.

all Total maximum number of replications (including those leading to non-valid solutions) performed. By default equals ten times the value of maxreplic. Ignored when objbnd is set to ‘Inf’.

maxiter Maximum number of iterations performed in each call to the local optimizer. By default set to 500 except with the “SANN” method, when by default is set to 1500.

maxeval Maximum number of function evaluations (nlminb method only) performed in each call to the nlminb optimizer. By defaults set to 1000.

RLOtol The relative convergence tolerance of the local optimizer. The local optimizer stops if it is unable to reduce the value by a factor of ‘RLOtol * (abs(val) + reltol)’ at a step. Ignored when method is set to “nlm”. By default set to the square root of the computer precision, i.e. to ‘sqrt(.Machine$double.eps)’.
hesEgtol  Numerical tolerance used to ensure that the hessian is non-singular. If the last eigenvalue of the hessian is positive but the ratio between it and the first eigenvalue is below hesEgtol the hessian is considered to be semi-definite and the parameter asymptotic standard errors are not computed. By default set to the square root of the computer precision, i.e. to ‘sqrt(.Machine$double.eps)’.

objbnd  Upper bound for the objective. Only solutions leading to objective values below objbnd are considered as valid.

Value

A list with the following components:

- **par**
  - The best result found for the parameter vector.

- **val**
  - The best value (minimum) found for the function fr.

- **vallist**
  - A vector with the best values found for each starting point.

- **iterations**
  - Number the iterations performed by the local optimizer in the call that generated the best result.

- **vallis**
  - A vector with the best values found for each starting point.

- **counts**
  - Number of times the function fr was evaluated in the call that generated the result returned.

- **convergence**
  - Code with the convergence status returned by the local optimizer.

- **message**
  - Message generated by the local optimizer.

- **hessian**
  - Numerically evaluated hessian of fr at the result returned. Only returned when the parameter hessian is set to TRUE.

- **hesseval**
  - Eigenvalues of the hessian matrix. Used to confirm if a local minimum was indeed found. Only returned when the parameter hessian is set to TRUE.

- **stderrors**
  - Assymptotic standard deviations of the parameters based on the observed information matrix. Only returned when the parse parameter is set to true and the hessian is indeed positive definite.

Author(s)

A. Pedro Duarte Silva

Robda-methods  

Robust Discriminant Analysis of Interval Data

Description

Roblda and Robqda perform linear and quadratic discriminant analysis of Interval Data based on robust estimates of location and scatter.
**Usage**

```r
### S4 method for signature 'IData'
Roblda(x, grouping, prior="proportions", CVtol=1.0e-5, egvtol=1.0e-10,
subset=1:nrow(x), CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE,
CovEstMet=c("Pooled","Globdev"), SngDMet=c("fasttle","fulltle"),
Robcontrol=RobEstControl(), ... )

### S4 method for signature 'IData'
Robqda(x, grouping, prior="proportions", CVtol=1.0e-5,
subset=1:nrow(x), CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE,
SngDMet=c("fasttle","fulltle"), Robcontrol=RobEstControl(), ... )
```

**Arguments**

- **x**: An object of class `IData` with the original Interval Data.
- **grouping**: Factor specifying the class for each observation.
- **prior**: The prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
- **CVtol**: Tolerance level for absolute value of the coefficient of variation of non-constant variables. When a MidPoint or LogRange has an absolute value within-groups coefficient of variation below CVtol, it is considered to be a constant.
- **egvtol**: Tolerance level for the eigenvalues of the product of the inverse within by the between covariance matrices. When a eigenvalue has an absolute value below egvtol, it is considered to be zero.
- **subset**: An index vector specifying the cases to be used in the analysis.
- **CovCase**: Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
- **SelCrit**: The model selection criterion.
- **silent**: A boolean flag indicating wether a warning message should be printed if the method fails.
- **CovEstMet**: Method used to estimate the common covariance matrix in Roblda (Robust linear discriminant analysis). Alternatives are “Pooled” (default) for a pooled average of the the robust within-groups covariance estimates, and “Globdev” for a global estimate based on all deviations from the groups multivariate l1 medians. See Todorov and Filzmoser (2009) for details.
- **SngDMet**: Algorithm used to find the robust estimates of location and scatter. Alternatives are “fasttle” (default) and “fulltle”.
- **Robcontrol**: A control object (S4) of class `RobEstControl-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.
- **...**: Other named arguments.
References


See Also

`lda`, `qda`, `snda`, `IData`, `RobEstControl`

Examples

```r
# Create an Interval-Data object containing the intervals for 899 observations on the temperatures by quarter in 60 Chinese meteorological stations.
ChinaT <- IData(ChinaTemp[1:8], VarNames=c("T1","T2","T3","T4"))

# Robust Linear Discriminant Analysis

## Not run:
ChinaT.rlda <- Roblda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- robust lda discriminant analysis results:
"
print(ChinaT.rlda)

# Estimate error rates by ten-fold cross-validation with 5 replications
CVrlda <- DACrossVal(ChinaT,ChinaTemp$GeoReg, TrainAlg=Roblda, CovCase=CovCase(ChinaT.rlda), CVrep=5)

summary(CVrlda[,,"Clerr"])

glberrors <-
  apply(CVrlda[,,"Nk"]*CVrlda[,,"Clerr"],1,sum)/apply(CVrlda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"
"

# Robust Quadratic Discriminant Analysis

ChinaT.rqda <- Robqda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- robust qda discriminant analysis results:
"
print(ChinaT.rqda)

# Estimate error rates by ten-fold cross-validation with 5 replications
CVrqda <- DACrossVal(ChinaT,ChinaTemp$GeoReg, TrainAlg=Robqda, CovCase=CovCase(ChinaT.rqda), CVrep=5)

summary(CVrqda[,,"Clerr"])
```
glberrors <-
  apply(CVrqda[,"Nk"]*CVrqda[,"Clerr"],1,sum)/apply(CVrqda[,"Nk"],1,sum)

cat("Average global classification error =",mean(glberrors),"\n")

## End(Not run)

---

RobEstControl

**Constructor function for objects of class RobEstControl**

**Description**

This function will create a control object of class RobEstControl containing the control parameters for the robust estimation functions fasttle, RobMxtDest, Roblda and Robqda.

**Usage**

RobEstControl(alpha=0.75, nsamp=500, seed=NULL, trace=FALSE, use.correction=TRUE, ncsteps=200, getalpha="TwoStep", rawMD2Dist="Chisq", MD2Dist="Chisq", eta=0.025, multiCmpCor="never", getkdblstar="TwoPlusOne", outline="MidPandLogR", trialmethod="simple", m=1, reweighted=TRUE, otpType="OnlyEst")

**Arguments**

- **alpha**
  Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*nrow(Idt) observations are used for computing the trimmed likelihood. Allowed values are between 0.5 and 1. Note that when argument ‘getalpha’ is set to “TwoStep” the final value of ‘alpha’ is estimated by a two-step procedure and the value of argument ‘alpha’ is only used to specify the size of the samples used in the first step.

- **nsamp**
  Number of subsets used for initial estimates.

- **seed**
  Starting value for random generator.

- **trace**
  Whether to print intermediate results.

- **use.correction**
  Whether to use finite sample correction factors.

- **ncsteps**
  The maximum number of concentration steps used each iteration of the fasttle algorithm.

- **getalpha**
  Argument specifying if the ‘alpha’ parameter (roughly the percentage of the sample used for computing the trimmed likelihood) should be estimadted from the data, or if the value of the argument ‘alpha’ should be used instead. When set to “TwoStep”, ‘alpha’ is estimated by a two-step procedure with the value of argument ‘alpha’ specifying the size of the samples used in the first step. Otherwise the value of argument ‘alpha’ is used directly.
The assumed reference distribution of the raw MCD squared distances, which is used to find cutoffs defining the observations kept in one-step reweighted MCD estimates. Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual Qui-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005).

The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual Qui-squared, the Beta and F distributions proposed by Cerioli (2010).

Nominal size of the null hypothesis that a given observation is not an outlier. Defines the raw MCD Mahalanobis distances cutoff used to choose the observations kept in the reweighting step.

Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at ‘eta’. ‘always’ – testing all n entities at 1.-(1.−‘eta’)^1/n; and ‘iterstep’ – as suggested by Cerioli (2010), make an initial set of tests using the nominal size 1.-(1.−‘eta’)^1/n, and if no outliers were detected stop. Otherwise, make a second step testing for outliers at ‘eta’.

Argument specifying the size of the initial small (in order to minimize the probability of outliers) subsets. If set to the string “Twopplusone” (default) the initial sets have twice the number of interval-value variables plus one which are they the smaller samples that lead to a non-singular covariance estimate). Otherwise, an integer with the size of the initial sets.

The type of outliers to be considered. “MidPandLogR” if outliers may be present in both MidPpoints and LogRanges, “MidP” if outliers are only present in MidPpoints, or “LogR” if outliers are only present in LogRanges.

The method to find a trial subset used to initialize each replication of the fasttle algorithm. The current options are “simple” (default) that simply selects ‘kdblstar’ observations at random, and “Poolm” that divides the original sample into ‘m’ non-overlapping subsets, applies the ‘simple trial’ and the refinement methods to each one of them, and merges the results into a trial subset.

Number of non-overlapping subsets used by the trial method when the argument of ‘trialmethod’ is set to ‘Poolm’.

Should a (Re)weighted estimate of the covariance matrix be used in the computation of the trimmed likelihood or just a “raw” covariance estimate; default is (Re)weighting.

The amount of output returned by fasttle. Current options are “OnlyEst” (default) where only an ‘IdtE’ object with the fasttle estimates is returned, “SetMD2andEst” which returns a list with an ‘IdtE’ object of fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and “SetMD2EstandPrfSt” which returns a list with the previous three components plus a list of some performance statistics concerning the algorithm execution.
Value

A RobEstControl object

References


See Also

RobEstControl, fasttlle, RobMxtDEst, Roblda, Robqda

---

**RobEstControl-class**

Class `RobEstControl` - contains control parameters for the robust estimation of parametric interval data models.

Description

This class extends the CovControlMcd class and contains control parameters for the robust estimation of parametric interval data models.

Objects from the Class

Objects can be created by calls of the form `new("RobEstControl", ...)` or by calling the constructor-function RobEstControl.

Slots

alpha: Inherited from class "CovControlMcd". Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*nrow(Idt) observations are used for computing the trimmed likelihood. Allowed values are between 0.5 and 1. Note that when argument ‘getalpha’ is set to “TwoStep” the final value of ‘alpha’ is estimated by a two-step procedure and the value of argument ‘alpha’ is only used to specify the size of the samples used in the first step.

nsamp: Inherited from class "CovControlMcd". Number of subsets used for initial estimates.

scalefn: Inherited from class "CovControlMcd" and not used in the package ‘Maint.Data.’

maxcsteps: Inherited from class "CovControlMcd" and not used in the package ‘Maint.Data.’
seed: Inherited from class "CovControlMcd". Starting value for random generator. Default is
  seed = NULL.

use.correction: Inherited from class "CovControlMcd". Whether to use finite sample correction
  factors. Default is use.correction=TRUE.

trace, tolSolve: Inherited from class "CovControl".

ncsteps: The maximum number of concentration steps used each iteration of the fasttle algorithm.

getalpha: Argument specifying if the ‘alpha’ parameter (roughly the percentage of the sample
  used for computing the trimmed likelihood) should be estimated from the data, or if the value
  of the argument ‘alpha’ should be used instead. When set to “TwoStep”, ‘alpha’ is estimated
  by a two-step procedure with the value of argument ‘alpha’ specifying the size of the samples
  used in the first step. Otherwise, with the value of argument ‘alpha’ is used directly.

rawMDDist: The assumed reference distribution of the raw MCD squared distances, which is
  used to find to cutoffs defining the observations kept in one-step reweighted MCD estimates.
  Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual
  Qi-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and
  Rocke (2005).

MD2Dist The assumed reference distributions used to find cutoffs defining the observations as-
  sumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual
  Qi-squared, and the Beta and F distributions proposed by Cerioli (2010).

eta Nominal size of the null hypothesis that a given observation is not an outlier. Defines the raw
  MCD Mahalanobis distances cutoff used to choose the observations kept in the reweighting
  step.

multiCmpCor Whether a multicomparison correction of the nominal size (eta) for the outliers
  tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and
  testing all entities at ‘eta’, ‘always’ – testing all n entitites at 1.- (1.-’eta’^N(1/n)); and ‘iterstep’
  – as sugested by Cerioli (2010), make an initial set of tests using the nominal size 1.-
  ‘eta’^N(1/n), and if no outliers were detected stop. Otherwise, make a second step testing for
  outliers at ‘eta’.

getkdblstar: Argument specifying the size of the initial small (in order to minimize the proba-
  bility of outliers) subsets. If set to the string “Twopplusone” (default) the initial sets have twice
  the number of interval-value variables plus one (i.e., they are the smaller samples that lead to
  a non-singular covariance estimate). Otherwise, an integer with the size of the initial sets.

outlin: The type of outliers to be consideres. “MidPandLogR” if outliers may be present in both
  MidPpoints and LogRanges, “MidP” if outliers are only present in MidPpoints, or “LogR” if
  outliers are only present in LogRanges.

trialmethod: The method to find a trial subset used to initialize each replication of the fasttle al-
  gorithm. The current options are “simple” (default) that simply selects ‘kdblstar’ observations
  at random, and “Poolm” that divides the original sample into ‘m’ non-overlaping subsets, ap-
  plies the ‘simple trial’ and the refinement methods to each one of them, and merges the results
  into a trial subset.

m: Number of non-overlaping subsets used by the trial method when the argument of ‘trialmethod’
  is set to ‘Poolm’.

rewighted: Should a (Re)weighted estimate of the covariance matrix be used in the computation
  of the trimmed likelihood or just a “raw” covariance estimate; default is (Re)weighting.
otpType: The amount of output returned by fasttle. Current options are “OnlyEst” (default) where only an ‘IdtE’ object with the fasttle estimates is returned, “SetMD2andEst” which returns a list with an ‘IdtE’ object of fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and “SetMD2EstandPrfSt” which returns a list with the previous three components plus a list of some performance statistics concerning the algorithm execution.

Extends

Class `CovControlMcd`, directly. Class `CovControl` by `CovControlMcd`, distance 2.

Methods

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

References


See Also

`RobEstControl`, `fasttle`, `RobMxtDEst`, `Roblda`, `Robqda`

---

RobMxtDEst estimates mixtures of distribution for interval-valued data using robust methods.

Usage

```r
## S4 method for signature 'IData'
RobMxtDEst(Idt, grouping, Mxt=c("Hom","Het"), CovEstMet=c("Pooled","Globdev"), CovCase=1:4, SelCrit=c("BIC","AIC"), Robcontrol=RobEstControl(), l1medpar=NULL, ...)
```
Arguments

idt An IData object representing interval-valued entities.
grouping Factor indicating the group to which each observation belongs to.
mxt Indicates the type of mixing distributions to be considered. Current alternatives are “Hom” (homocedastic) and “Het” (heterocedastic).
CovEstMet Method used to estimate the common covariance matrix. Alternatives are “Pooled” (default) for a pooled average of the robust within-groups covariance estimates, and “Globdev” for a global estimate based on all deviations from the groups multivariate l1 medians.
See Todorov and Filzmoser (2009) for details.
CovCase Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit The model selection criterion.
Robcontrol A control object (S4) of class RobEstControl-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.
11medpar List of named arguments to be passed to the function pcaPP::11median (in package pcaPP) used to find the multivariate l1 medians. Possible components are ‘MaxStep’, ‘ItTol’ and ‘trace’ (see the documentation of pcaPP::11median for details).
If kept at NULL (default) the defaults of pcaPP::11median will be used.

Value

An object of class IdtMxNDRE, containing the estimation results.

References


See Also

IdtMxNDRE, RobEstControl.
Description

$snda$ performs discriminant analysis of Interval Data based on estimates of mixtures of Skew-Normal models.

Usage

```r
## S4 method for signature 'IData'
snda(x, grouping, prior="proportions", CVtol=1.0e-5, subset=1:nrow(x),
    CovCase=1:4, SelCrit=c("BIC","AIC"), Mxt=c("Loc","Gen"), ... )

## S4 method for signature 'IdtLocSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)),
    egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IdtLocNSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)@SNMod), egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IdtGenSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)), silent=FALSE, ... )

## S4 method for signature 'IdtGenNSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)@SNMod), silent=FALSE, ... )
```

Arguments

- `x` An object of class `IData, IdtLocSNMANOVA, IdtLocNSNMANOVA, IdtGenSNMANOVA` or `IdtGenNSNMANOVA` with either the original Interval Data, or the results of a Interval Data Skew-Normal MANOVA, from which the discriminant analysis will be based.
- `grouping` Factor specifying the class for each observation.
- `prior` The prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
- `CVtol` Tolerance level for absolute value of the coefficient of variation of non-constant variables. When a MidPoint or LogRange has an absolute value within-groups coefficient of variation below CVtol, it is considered to be a constant.
- `subset` An index vector specifying the cases to be used in the analysis.
 CovCase  Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
 SelCrit  The model selection criterion.
 Mxt  Indicates the type of mixing distributions to be considered. Current alternatives are “Loc” (location model – groups differ only on the location parameters of a Skew-Normal model) and “Gen” (general model – groups differ on all parameters of a Skew-Normal models).
 silent  A boolean flag indicating whether a warning message should be printed if the method fails.
 selmodel  Selected model from a list of candidate models saved in object x.
 egvtol  Tolerance level for the eigenvalues of the product of the inverse within by the between covariance matrices. When a eigenvalue has an absolute value below egvtol, it is considered to be zero.
 ...  Other named arguments.

References


See Also

`lda`, `qda`, `Roblda`, `Robqda`, `IData`, `IdtLocSNMANOVA`, `IdtLocNSNMANOVA`, `IdtGenSNMANOVA`, `IdtGenSNMANOVA`

Examples

```r
## Not run:

# Create an Interval-Data object containing the intervals for 899 observations # on the temperatures by quarter in 60 Chinese meteorological stations.
ChinaT <- IData(ChinaTemp[1:8], VarNames=c("T1","T2","T3","T4"))

# Skew-Normal based discriminant analysis, assuming that the different regions differ # only in location parameters
ChinaT.locsnda <- snda(ChinaT,ChinaTemp$GeoReg,Mxt="Loc")
cat("Temperatures of China -- SkewNormal location model discriminant analysis results:\n") print(ChinaT.locsnda)

# Estimate error rates by three-fold cross-validation without replication
CVlocsnda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=locsnda,Mxt="Loc")
```
CovCase=CovCase(ChinaT.locsnda),kfold=3,CVrep=1)

summary(CVlocnsnda[,"Clerr"])

glerrors <-
   apply(CVlocnsnda[,"Nk"]+CVlocnsnda[,"Clerr"],1,sum)/apply(CVlocnsnda[,"Nk"],1,sum)

cat("Average global classification error =",mean(glerrors),"\n")

# Skew-Normal based discriminant analysis, assuming that the different regions may differ
# in all SkewNormal parameters

ChinaT.gensnda <- snda(ChinaT,ChinaTemp$GeoReg,Mxt="Gen")

cat("Temperatures of China -- SkewNormal general model discriminant analysis results:\n")

print(ChinaT.gensnda)

#Estimate error rates by three-fold cross-validation without replication

CVgensnda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=snda,Mxt="Gen",
   CovCase=CovCase(ChinaT.gensnda),kfold=3,CVrep=1)

summary(CVgensnda[,"Clerr"])

glerrors <-
   apply(CVgensnda[,"Nk"]+CVgensnda[,"Clerr"],1,sum)/apply(CVgensnda[,"Nk"],1,sum)

cat("Average global classification error =",mean(glerrors),"\n")

## End(Not run)

---

**stdEr--methods**

Methods for function `stdEr` in Package `MAINT.Data`

**Description**

S4 methods for function `stdEr`. As in the generic `stdEr` S3 `miscTools` method, these methods extract standard errors of the parameter estimates, for the models fitted to Interval Data.

**Usage**

```r
## S4 method for signature 'IdtNDE'
stdEr(x, selmodel=BestModel(x), ...)
## S4 method for signature 'IdtSNDE'
stdEr(x, selmodel=BestModel(x), ...)
## S4 method for signature 'IdtNandSNDE'
stdEr(x, selmodel=BestModel(x), ...)
```
Arguments

x                  An object representing a model fitted to interval data.
selmodel           Selected model from a list of candidate models saved in object x.
...                Additional arguments for method functions.

Value

A vector of the estimated standard deviations of the parameter estimators.

See Also

vcov

description

Performs statistical likelihood-ratio tests that evaluate the goodness-of-fit of a nested model against a more general one.

Usage

testMod(Model, RestMod = Mod@ModelConfig[2]:length(Mod@ModelConfig), FullMod = "Next")

Arguments

Model               An object of class IdtE representing the estimates of a model fitted to a data set of interval-value variables
RestMod             Indices of the restricted models being evaluated in the NULL hypothesis
FullMod             Either indices of the general models being evaluated in the alternative hypothesis or the strings "Next" (default) or "All". In the former case a Restricted model is always compared against the most parsimonious alternative that encompasses it, and in latter all possible comparisons are performed

Value

An object of class ConfTests with the results of the tests performed
Examples

# Create an Interval-Data object containing the intervals of temperatures by quarter
# for 899 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8])

# Estimate by maximum likelihood the parameters of Gaussian models
# for the Winter (1st and 4th) quarter intervals

ChinaWTE <- mle(ChinaT[,c(1,4)])
cat("China maximum likelihood estimation results for Winter quarters:

print(ChinaWTE)

# Perform Likelihood-Ratio tests comparing models with consecutive nested Configuration
testMod(ChinaWTE)

# Perform Likelihood-Ratio tests comparing all possible models
testMod(ChinaWTE, FullMod="All")

# Compare model with covariance Configuration case 3 (MidPoints independent of LogRanges)
# against model with covariance Configuration 1 (unrestricted covariance)
testMod(ChinaWTE, RestMod=3, FullMod=1)

---

Description

S4 methods for function var. These methods extract estimates of variance-covariance matrices for
the models fitted to Interval Data.

Usage

## S4 method for signature 'IdtNDE'
var(x)
## S4 method for signature 'IdtSNDE'
var(x)
## S4 method for signature 'IdtNandSNDE'
var(x)
## S4 method for signature 'IdtMxNDE'
var(x)
## S4 method for signature 'IdtMxSNDE'
var(x)
Arguments

x  An object representing a model fitted to interval data.

Value

For the `IdtNDE`, `IdtSNDE` and `IdtNandSNDE` methods or `IdtMxNDE`, `IdtMxSNDE` methods with slot “Hmcdt” equal to TRUE: a matrix with the estimated covariances.
For the `IdtMxNDE` and `IdtMxSNDE` methods with slot “Hmcdt” equal to FALSE: a three-dimensional array with a matrix with the estimated covariances for each group at each level of the third dimension.

See Also

cor

Description

S4 methods for function vcov. As in the generic vcov S3 ‘stats’ method, these methods extract variance-covariance estimates of parameter estimators, for the models fitted to Interval Data.

Usage

```r
## S4 method for signature 'IdtNDE'
vcov(object, selmodel=BestModel(object), ...)  
## S4 method for signature 'IdtSNDE'
vcov(object, selmodel=BestModel(object), ...)  
## S4 method for signature 'IdtNandSNDE'
vcov(object, selmodel=BestModel(object), ...)  
## S4 method for signature 'IdtMxNDE'
vcov(object, selmodel=BestModel(object), group=NULL, ...)  
## S4 method for signature 'IdtMxSNDE'
vcov(object, selmodel=BestModel(object), group=NULL, ...)  
```

Arguments

object          An object representing a model fitted to interval data.

selmodel        Selected model from a list of candidate models saved in object.

group           Ther group for each the estimated parameter variance-covariance will be returned. If NULL (default), “vcov” will return a three-dimensional array with a matrix of the estimated covariances between the parameter estimates for each group at each level of the third dimension.

...             Additional arguments for method functions.
Value
For the `IdtNDE`, `IdtSNDE` and `IdtNandSNDE` methods or `IdtMxNDE`, `IdtMxSNDE` methods with slot “Hmcdt” equal to TRUE: a matrix of the estimated covariances between the parameter estimates. For the `IdtMxNDE`, and `IdtMxSNDE` methods with slot “Hmcdt” equal to FALSE: if argument “group” is set to NULL, a three-dimensional array with a matrix of the estimated covariances between the parameter estimates for each group at each level of the third dimension. If argument “group” is set to an integer, the matrix with the estimated covariances between the parameter estimates, for the group chosen.

See Also

`stdEr`
Index

*Topic **AIC**
  BestModel-methods, 6

*Topic **BIC**
  BestModel-methods, 6

*Topic **Discriminant Analysis Interval Data**
  lda-methods, 46
  qda-methods, 52
  Robda-methods, 57
  snda-methods, 66

*Topic **Interval Data**
  coef--methods, 8
  cor--methods, 10
  IdtNandSNDE-class, 36
  IdtNDE-class, 36
  IdtSNDE-class, 38
  lda-methods, 46
  MAINT.Data-package, 3
  qda-methods, 52
  Robda-methods, 57
  snda-methods, 66
  stdErr--methods, 68
  var--methods, 70
  vcov--methods, 71

*Topic **MAINT.Data**
  MAINT.Data-package, 3

*Topic **MANOVA for Interval Data**
  MAINT.Data-package, 3

*Topic **MANOVA**
  IdtMANOVA-class, 27
  MANOVA-methods, 49

*Topic **Parametric modelling of Interval Data**
  MAINT.Data-package, 3

*Topic **RobMxtDEst**
  RobMxtDEst-methods, 64

*Topic **Statistical tests for Interval Data**
  MAINT.Data-package, 3

*Topic **Symbolic Data Analysis**
  MAINT.Data-package, 3

*Topic **classes**
  ConfTests-class, 9
  IData-class, 23
  IdtE-class, 25
  Idtlda-class, 26
  IdtMANOVA-class, 27
  IdtMxE-class, 29
  IdtMxNandSNDE-class, 30
  IdtMxNDE-class, 31
  IdtMxNDRE-class, 33
  IdtMxSNDE-class, 34
  IdtMxtnDE-class, 35
  IdtNandSNDE-class, 36
  IdtNDE-class, 36
  Idtqda-class, 37
  IdtSNDE-class, 38
  IdtSNgenda-class, 39
  IdtSngNandSNDE-class, 40
  IdtSngNDE-class, 41
  IdtSngNDRE-class, 42
  IdtSngSNDE-class, 44
  IdtSNlocda-class, 45
  LRTest-class, 48
  RobEstControl-class, 62

*Topic **coef**
  coef--methods, 8

*Topic **cor**
  cor--methods, 10

*Topic **cross-validation**
  DACrossVal, 11

*Topic **datasets**
  Cars, 7
  ChinaTemp, 7

*Topic **discriminant analysis**
  DACrossVal, 11

*Topic **interval data**
  BestModel-methods, 6
Cars, 7
ChinaTemp, 7
ConfTests-class, 9
fasttle-methods, 13
fulltle-methods, 18
IData, 22
IData-class, 23
ldtE-class, 25
ldtlda-class, 26
IdtMANOVA-class, 27
ldtMxE-class, 29
ldtMxNandSNDE-class, 30
ldtMxNDE-class, 31
ldtMxNDRE-class, 33
ldtMxSNDE-class, 34
ldtMxtNDE-class, 35
ldtqda-class, 37
ldtSNagenda-class, 39
ldtSngNandSNDE-class, 40
ldtSngNDE-class, 41
ldtSngNDRE-class, 42
ldtSngSNDE-class, 44
ldtSNlocda-class, 45
LRTest-class, 48
MANOVA-methods, 49
mle-methods, 51
RobEstControl-class, 62
RobMxtDEst-methods, 64
testMod-methods, 69
*Topic likelihood ratio test
LRTest-class, 48
testMod-methods, 69
*Topic likelihood ratio tests
MANOVA-methods, 49
*Topic maximum likelihood estimation
mle-methods, 51
*Topic methods
BestModel-methods, 6
coeF(coef--methods), 8
coeF,IdtNandSNDE-method (coef--methods), 8
coeF,IdtNDE-method (coef--methods), 8
coeF,IdtSNDE-method (coef--methods), 8
coeF--methods, 8
coeF-methods (coef--methods), 8
ConfTests, 48
ConfTests-class, 9
cor, 71
cor (cor--methods), 10
cor,IdtMxNDE-method (cor--methods), 10
cor,IdtMxSNDE-method (cor--methods), 10
cor,IdtNandSNDE-method (cor--methods), 10
cor,IdtNDE-method (cor--methods), 10
cor,IdtSNDE-method (cor--methods), 10
cor--methods, 10
cor-methods (cor--methods), 10
CovCase,Idtlda-method (Idtlda-class), 26
CovCase,ldtqda-method (ldtqda-class), 37
RobMxtDEst-methods, 64
snda-methods, 66
stdEr--methods, 68
testMod-methods, 69
var--methods, 70
vcov--methods, 71
*Topic robust
RobEstControl-class, 62
*Topic stdEr
stdEr--methods, 68
*Topic trimmed maximum likelihood estimator
fasttle-methods, 13
corr-fulltle-methods, 18
corr-var--methods, 70
*Topic vcov
vcov--methods, 71
Random.seed, 14
BestModel (BestModel-methods), 6
BestModel,IData-method (BestModel-methods), 6
BestModel,ldtE-method (ldtE-class), 25
BestModel-methods, 6
BestModel-methods, 6
Cars, 7
ChinaTemp, 7
coeF (coefficient methods), 8
coeF,IdtNandSNDE-method (coefficient methods), 8
coeF,IdtNDE-method (coefficient methods), 8
coeF,IdtSNDE-method (coefficient methods), 8
coeF--methods, 8
coeF-methods (coefficient methods), 8
ConfTests, 48
ConfTests-class, 9
cor, 71
cor (cor--methods), 10
cor,IdtMxNDE-method (cor--methods), 10
cor,IdtMxSNDE-method (cor--methods), 10
cor,IdtNandSNDE-method (cor--methods), 10
cor,IdtNDE-method (cor--methods), 10
cor,IdtSNDE-method (cor--methods), 10
cor--methods, 10
cor-methods (cor--methods), 10
CovCase,Idtlda-method (Idtlda-class), 26
CovCase,ldtqda-method (ldtqda-class), 37
RobMxtDEst-methods, 64
snda-methods, 66
stdEr--methods, 68
testMod-methods, 69
var--methods, 70
vcov--methods, 71
*Topic robust
RobEstControl-class, 62
*Topic stdEr
stdEr--methods, 68
*Topic trimmed maximum likelihood estimator
fasttle-methods, 13
corr-fulltle-methods, 18
corr-var--methods, 70
*Topic vcov
vcov--methods, 71
Random.seed, 14
BestModel (BestModel-methods), 6
BestModel,IData-method (BestModel-methods), 6
BestModel,ldtE-method (ldtE-class), 25
BestModel-methods, 6
BestModel-methods, 6
Cars, 7
ChinaTemp, 7
coeF (coefficient methods), 8
coeF,IdtNandSNDE-method (coefficient methods), 8
coeF,IdtNDE-method (coefficient methods), 8
coeF,IdtSNDE-method (coefficient methods), 8
coeF--methods, 8
coeF-methods (coefficient methods), 8
ConfTests, 48
ConfTests-class, 9
cor, 71
cor (cor--methods), 10
cor,IdtMxNDE-method (cor--methods), 10
cor,IdtMxSNDE-method (cor--methods), 10
cor,IdtNandSNDE-method (cor--methods), 10
cor,IdtNDE-method (cor--methods), 10
cor,IdtSNDE-method (cor--methods), 10
cor--methods, 10
cor-methods (cor--methods), 10
CovCase,Idtlda-method (Idtlda-class), 26
CovCase,ldtqda-method (ldtqda-class), 37
INDEX

CovCase, IdtSNgenda-method (IdtSNgenda-class), 39
CovCase, IdtSNloca-method (IdtSNloca-class), 45
CovControl, 64
CovControlMcd, 64
DACrossVal, 11
extmatrix-class, 13
fasttle, 20, 22, 25, 26, 36, 37, 41, 43, 54, 60, 62, 64
fasttle (fasttle-methods), 13
fasttle, IData-method (fasttle-methods), 13
fasttle-methods, 13
fulltle, 17, 22, 25, 26, 36, 37, 41, 43, 54
fulltle (fulltle-methods), 18
fulltle, IData-method (fulltle-methods), 18
fulltle-methods, 18
getIdtOutl, 17, 20, 21
H0res (MANOVA-methods), 49
H0res, IdtMANOVA-method (MANOVA-methods), 49
H1res (MANOVA-methods), 49
H1res, IdtMANOVA-method (MANOVA-methods), 49
head, IData-method (IData-class), 23
IData-class, 23
IdtCLMANOVA, 27, 46, 47
IdtCLMANOVA-class (IdtMANOVA-class), 27
IdtE, 6, 15, 16, 19, 29-35, 40-44, 69
IdtE-class, 25
IdtGenNSMANOVA, 27, 52, 53, 66
IdtGenNSMANOVA-class (IdtMANOVA-class), 27
IdtGenSNMANOVA, 27, 66, 67
IdtGenSNMANOVA-class (IdtMANOVA-class), 27
IdtHetMANOVA, 27, 52, 53
IdtHetMANOVA-class (IdtMANOVA-class), 27
Idtlda-class, 26
IdtLocNSMANOVA, 27, 46, 47, 66, 67
IdtLocNSMANOVA-class (IdtMANOVA-class), 27
IdtLocSNMANOVA, 27, 66, 67
IdtLocSNMANOVA-class (IdtMANOVA-class), 27
IdtMANOVA, 50
IdtMANOVA-class, 27
IdtMxE, 30-35
IdtMxE-class, 29
IdtMxNandSNDE, 36, 41
IdtMxNandSNDE-class, 30
IdtMxNDE, 10, 32, 34-37, 42, 71, 72
IdtMxNDE-class, 31
IdtMxNDRE, 33, 35-37, 43, 65
IdtMxNDRE-class, 33
IdtMxSNDE, 10, 35, 38, 39, 45, 71, 72
IdtMxSNDE-class, 34
IdtMxTNDRE, 46, 47, 52, 53
IdtMxTNDRE-class, 35
IdtNandSNDE, 8, 10, 71, 72
IdtNandSNDE-class, 36
IdtNDE, 10, 39, 71, 72
IdtNDE-class, 36
Idtqda-class, 37
IdtSNDE, 8, 10, 37, 71, 72
IdtSNDE-class, 38
IdtSngDE, 25, 30, 41-44
IdtSngDE-class (Idt-class), 25
IdtSngDE-class, 39
IdtSngNandSNDE, 31, 36
IdtSngNandSNDE-class, 40
IdtSngNDE, 21, 33, 36, 37, 43, 45
IdtSngNDE-class, 41
IdtSngNDE-class, 34, 36, 37, 42
IdtSngNDE-class, 42
IdtSngSNDE, 35, 38, 39, 42
IdtSngSNDE-class, 44
IdtSNloca-class, 45
is.IData (IData), 22
lda, 12, 25, 29, 53, 59, 67
lda, IData-method (lda-methods), 46
lda, IdtCLMANOVA-method (lda-methods), 46
lda, IdtLocNSMANOVA-method (lda-methods), 46
lda, IdtMxTNDRE-method (lda-methods), 46
lda-methods, 46
LRTest, 9, 27, 28
LRTest-class, 48
MAINT.Data (MAINT.Data-package), 3
MAINT.Data-package, 3
MANOVA, 25–27, 29–31, 33, 35–40, 46, 48
MANOVA (MANOVA-methods), 49
MANOVA, IData-method (MANOVA-methods), 49
MANOVA-methods, 49
matrix, 13
mle, 9, 25, 26, 36, 37, 39, 41, 42, 45, 48
mle (mle-methods), 51
mle, IData-method (mle-methods), 51
mle-methods, 51
predict, Idtlda-method (Idtlda-class), 26
predict, Idtqda-method (Idtqda-class), 37
predict, IdtSNagenda-method (IdtSNagenda-class), 39
predict, IdtSNloca-method (IdtSNloca-class), 45
predict, IdtSNloca-method (IdtSNloca-class), 45
print, IData-method (IData-class), 23
qda, 12, 25, 27, 29, 38, 47, 59, 67
qda, IData-method (qda-methods), 52
qda, IdtGenNSMMANOVA-method (qda-methods), 52
qda, IdtHetNMANOVA-method (qda-methods), 52
qda, IdtMixtNDE-method (qda-methods), 52
qda-methods, 52
qHardRoQf, 54
ReplOptim, 50, 51, 55
Roblda-methods, 57
RobEstControl, 15, 17, 59, 60, 60, 62, 64, 65
RobEstControl-class, 62
Roblda, 25, 27, 47, 53, 60, 62, 64, 67
Roblda (Roblda-methods), 57
Roblda, IData-method (Roblda-methods), 57
Roblda-methods (Roblda-methods), 57
RobMxtDEst, 25, 26, 30, 31, 34, 36, 37, 60, 62, 64
RobMxtDEst (RobMxtDEst-methods), 64
RobMxtDEst, IData-method (RobMxtDEst-methods), 64
RobMxtDEst-methods, 64
Robqda, 25, 27, 38, 47, 53, 60, 62, 64, 67
Robqda (Robqda-methods), 57
Robqda, IData-method (Robqda-methods), 57
Robqda-methods (Robqda-methods), 57
rrocov.control, 14
show, ConfTests-method (ConfTests-class), 9
show, IdtE-method (IdtE-class), 25
show, Idtlda-method (Idtlda-class), 26
show, IdtMANOVA-method (MANOVA-methods), 49
show, Idtqda-method (Idtqda-class), 37
show, IdtSNagenda-method (IdtSNagenda-class), 39
show, IdtSNloca-method (IdtSNloca-class), 45
show, LRTtest-method (LRTtest-class), 48
snda, 27, 29, 40, 46, 47, 53, 59
snda, IData-method (snda-methods), 66
snda, IdtGenNSMMANOVA-method (snda-methods), 66
snda, IdtGenNSMOMANOVA-method (snda-methods), 66
snda, IdtLNlocNSMANNANOV-method (snda-methods), 66
snda-methods, 66
stdEr, 8, 72
stdEr (stdEr-methods), 68
stdEr, IdtNandSNDE-method (stdEr-methods), 68
stdEr, IdtNDE-method (stdEr-methods), 68
stdEr, IdtSNDE-method (stdEr-methods), 68
stdEr-methods, 68
summary, IData-method (IData-class), 23
tail, IData-method (IData-class), 23
testMod (testMod-methods), 69
testMod, IdtE-method (IdtE-class), 25
testMod-methods, 69
var, 10
var (var-methods), 70
var, IdtMxNDE-method (var-methods), 70
var, IdtMxSNDE-method (var-methods), 70
var, IdtNandSNDE-method (var-methods), 70
var, IdtNDE-method (var-methods), 70
INDEX

var, IdtSNDE-method (var--methods), 70
var--methods, 70
var-methods (var--methods), 70
vcov, 8, 69
vcov (vcov--methods), 71
vcov, IdtMxNDE-method (vcov--methods), 71
vcov, IdtMxSNDE-method (vcov--methods), 71
vcov, IdtNandSNDE-method
   (vcov--methods), 71
vcov, IdtNDE-method (vcov--methods), 71
vcov, IdtSNDE-method (vcov--methods), 71
vcov--methods, 71
vcov-methods (vcov--methods), 71