Package ‘MAINT.Data’

March 2, 2026

Type Package

Title Model and Analyse Interval Data

Version 2.8.0

Date 2026-03-02

Author Pedro Duarte Silva [aut, cre],
Paula Brito [aut]

Maintainer Pedro Duarte Silva <psilva@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 (>= 3.5.0), Rcpp (>= 1.0.3), methods

Imports MASS, miscTools, robustbase, rrcov, pcaPP, mclust, ggplot2, GGally, sn (>= 1.3.0), withr

Suggests testthat

LinkingTo Rcpp, RcppArmadillo (>= 0.9.500.2.0)

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-03-02 14:40:08 UTC

Contents

MAINT.Data-package 3
Abalone 7

AgrMcDt . . . 7
BestModel-methods . . . 8
Cars . . . 9
ChinaTemp . . . 9
coef–methods . . . 10
ConfMat . . . 11
ConfTests-class . . . 13
cor–methods . . . 14
DACrossVal . . . 15
EMControl . . . 17
EMControl-class . . . 18
extmatrix-class . . . 18
fasttle-methods . . . 19
fulltle-methods . . . 22
getIdtOutl . . . 25
IData . . . 28
IData-class . . . 29
IdtE-class . . . 31
Idtlda-class . . . 32
IdtMANOVA-class . . . 33
IdtMclust-class . . . 35
Idtmclust-methods . . . 37
IdtMxE-class . . . 39
IdtMxNandSNDE-class . . . 40
IdtMxNDE-class . . . 42
IdtMxNDRE-class . . . 43
IdtMxSNDE-class . . . 45
IdtMxtNDE-class . . . 46
IdtNandSNDE-class . . . 47
IdtNDE-class . . . 47
IdtOutl-class . . . 48
IdtOutl-methods . . . 50
Idtqda-class . . . 51
IdtSNDE-class . . . 52
IdtSNgenda-class . . . 52
IdtSngNandSNDE-class . . . 54
IdtSngNDE-class . . . 55
IdtSngNDRE-class . . . 56
IdtSngSNDE-class . . . 58
IdtSNlocda-class . . . 59
lda-methods . . . 60
LoansbyPurpose_minmaxDt . . . 62
LoansbyRiskLvs_minmaxDt . . . 63
LoansbyRiskLvs_qntlDt . . . 64
LRTest-class . . . 65
MANOVA-methods . . . 65
MANOVAPermTest . . . 67
mean–methods . . . 69

mle-methods 70
nycflights 72
pcoordplot-methods 72
plot-methods 74
plotInfCrt-methods 75
qda-methods 77
qHardRoqF 78
RepLOptim 79
Robda-methods 82
RobEstControl 84
RobEstControl-class 87
RobMxtDEst-methods 89
snda-methods 91
stdEr–methods 93
summary-methods 94
testMod-methods 95
var–methods 96
vcov–methods 96

Index **98**

`MAINT.Data-package` *Modelling and Analizing Interval Data*

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, Discriminant Analysis and Gaussian Model Based Clustering.

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 (Duarte Silva, Filzmoser and Brito (2017)) estimation, and statistical tests for the different considered configurations. Methods for (M)ANOVA, Discriminant Analysis (Duarte Silva and Brito (2015)) and model based clustering (Brito, Duarte Silva and Dias (2015)) of this data class are also provided.

Package:	MAINT.Data
Type:	Package
Version:	2.8.0
Date:	2026-03-02
License:	GPL-2
LazyLoad:	yes
LazyData:	yes

Author(s)

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

Maintainer: Pedro Duarte Silva <psilva@porto.ucp.pt>

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P. and Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Brito, P., Duarte Silva, A. P. and Dias, J. G. (2015), Probabilistic Clustering of Interval Data. *Intelligent Data Analysis* **19**(2), 293–313.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Noirhomme-Fraiture, M. and Brito, P. (2011), Far Beyond the Classical Data Models: Symbolic Data Analysis. *Statistical Analysis and Data Mining* **4**(2), 157–170.

Examples

```
# 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 likelhiood 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(stdEr(ChinaWTE.C234))

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

## Not run:
ChinaWTE.C234 <- fasttle(ChinaWT,CovCase=2:4)
cat("Winter temperatures of China -- normal maximum trimmed likelhiood 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 likelhiood estimation results:\n")
print(ChinaWTE.SkN)
cat("Standard Errors of Estimators:\n") ; print(stdEr(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(H1res(ManvChinaWT.C13)),CVrep=1)

#Robust Quadratic Discriminant Analysis

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

## End(Not run)

# Create an Interval-Data object containing the intervals of loan data
# (from the Kaggle Data Science platform) aggregated by loan purpose

LbyPIdt <- IData(LoansbyPurpose_minmaxDt,
  VarNames=c("ln-inc","ln-revolbal","open-acc","total-acc"))

print(LbyPIdt)

## Not run:

#Fit homoscedastic Gaussian mixtures with up to six components

mclustres <- Idtmclust(LbyPIdt,G=1:6)
plotInfCrt(mclustres,legpos="bottomright")
print(mclustres)

#Display the results of the best mixture according to the BIC

summary(mclustres,parameters=TRUE,classification=TRUE)
pcoordplot(mclustres)


## End(Not run)
```

Abalone — *Abalone Data Set*

Description

A interval-valued data set containing 24 units, created from from the Abalone dataset (UCI Machine Learning Repository), after aggregating by sex and age.

Usage

```
data(Abalone)
```

Format

AbdaDF: A data frame containing the original 4177 Abalone individuals described by 7 variables.
AbUnits: A factor with 4177 observations and 24 levels indicating the sex by age combination to which each orginal individual belongs to.
AbaloneIdt: An IData object with 24 observations and 7 interval-valued variables, describing the intervals formed by aggregating the AbdaDF microdata by the AbUnits factor.

AgrMcDt — *Agregate Micro Data*

Description

AgrMcDt creates IData objects by agregating a Data Frame of Micro Data.

Usage

```
AgrMcDt(MicDtDF, agrby, agrcrt="minmax", trnsf = c("RngLog","AllLogit"))
```

Arguments

MicDtDF	A data frame with the original values of the micro data.
agrby	A factor with categories on which the micro data should be aggregated.
agrcrt	The aggregation criterion. Either the 'minmax' string, or a two dimensional vector with the prob. value for the left (lower) percentile, followed by the prob. value for the right (upper) percentile, used in the aggregation.
trnsf	The data transformation to be applied to the interval ranges and (possibly) the MidPoints. Alternatives are: "RngLog" The MidPoints are left unchanged and a log tranformation is applied to the ranges. "AllLogit" A logit transformation is applied to both the MidPoints and the ranges. This option is particularly usefull for dara restricted to the (0,1) domain.

Value

An object of class IData with the data set of Interval-valued variables resulting from the aggregation performed.

See Also

IData

Examples

```
# Create an Interval-Data object by agregating the microdata consisting
# of 336776 NYC flights included in the FlightsDF data frame,
# by the statistical units specified in the FlightsUnits factor.

Flightsminmax <- AgrMcDt(FlightsDF,FlightsUnits)

#Display the first and last observations

head(Flightsminmax)
tail(Flightsminmax)

#Print summary statistics

summary(Flightsminmax)

## Not run:

# Repeat this procedure using now the 10th and 90th percentiles.

Flights1090prcnt <- AgrMcDt(FlightsDF,FlightsUnits,agrcrt=c(0.1,0.9))

#Display the first and last observations

head(Flights1090prcnt)
tail(Flights1090prcnt)

summary(Flights1090prcnt)


## 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 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 meteorologic 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 geographic region of each station.

---

coef-methods *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

```
## 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 list of parameter estimates. The list components depend on the model and parametriztion assumed by the model. For Gaussian models these are respectivelly mu (vector of mean estimates) and Sigma (matrix of covariance estimates). For Skew-Normal models the components are mu, Sigma and gamma1 (one vector of skewness coefficient estimates) for the centred parametrization and the vectors ksi, and alpha, and the matrix Omega for the direct parametrization.

References

Arellano-Valle, R. B. and Azzalini, A. (2008): "The centred parametrization for the multivariate skew-normal distribution". Journal of Multivariate Analysis, Volume 99, Issue 7, 1362-1382.

See Also

stdEr, vcov

Examples

```
# 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"))

ChinaT_NE <- mle(ChinaT)

# Display model estimates

print(coef(ChinaT_NE))

## Not run:

# Estimate Skew-Normal distribution  parameters by maximum likelihood

ChinaT_SNE <- mle(ChinaT,Model="SKNormal")

# Display model estimates

print(coef(ChinaT_SNE,ParType="Centr"))
print(coef(ChinaT_SNE,ParType="Direct"))


## End(Not run)
```

ConfMat *Confussion Matrices for classification results*

Description

'ConfMat' creates confussion matrices from two factor describing, respectively, original classes and predicted classification results

Usage

```
ConfMat(origcl, predcl, otp=c("absandrel","abs","rel"), dec=3)
```

Arguments

origcl	A factor describing the original classes.
predcl	A factor describing the predicted classes.
otp	A string describing the output to be displayed and returned. Alternatives are "absandrel" for two confusion matrices, respectively with absolute and relative frequencies, "abs" for a confusion matrix with absolute frequencies, and "rel" for a confusion matrix relative frequencies.
dec	The number of decimal digits to display in matrices of relative frequencies.

Value

When argument 'otp' is set to "absandrel" (default), a list with two confusion matrices, respectively with absolute and relative frequencies. When argument 'otp' is set to "abs" a confusion matrix with absolute frequencies, and when argument 'otp' is set to "rel" a confusion matrix with relative frequencies.

Author(s)

A. Pedro Duarte Silva

See Also

lda, qda, snda, Roblda, Robqda, DACrossVal

Examples

```
# 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$GeoReg)
ldapred <- predict(ChinaT.lda,ChinaT)$class

# lda resubstitution confusion matrix

ConfMat(ChinaTemp$GeoReg,ldapred)

#Quadratic Discriminant Analysis

ChinaT.qda <- qda(ChinaT,ChinaTemp$GeoReg)
qdapred <- predict(ChinaT.qda,ChinaT)$class

# qda resubstitution confusion matrix

ConfMat(ChinaTemp$GeoReg,qdapred)
```

`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:

ChiSq: Value of the Chi-Square statistics corresponding to the performed test.
df: Degrees of freedom of the Chi-Square statistics.
pvalue: p-value of the Chi-Square statistics value, obtained from the Chi-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`

cor-methods *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

```
## 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

Examples

```
# 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"))

ChinaT_NE <- mle(ChinaT)

# Display correlation estimates
```

```
print(cor(ChinaT_NE))
```

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, dec=3, ...)
```

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.
dec	The number of decimal digits to display in confusion matrices of relative frequencies.
...	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

```
## 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"])

# Linear Discriminant Analysis with covariance case 3

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

# Classical (configuration 1) Quadratic Discriminant Analysis

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

# Quadratic Discriminant Analysis with covariance case 3

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


## End(Not run)
```

EMControl *Constructor function for objects of class EMControl*

Description

This function will create a control object of class EMControl containing the control parameters for the EM algorithm used in estimation of Gaussian mixtures by function Idtmclust.

Usage

```
EMControl(nrep=0, maxiter=1000, convtol=0.01, protol=1e-3, seed=NULL, pertubfct=1,
  k2max=1e6, MaxVarGRt=1e6)
```

Arguments

nrep	Number of replications (different randomly generated starting points) of the EM algorithm.
maxiter	Maximum number of iterations in each replication of the EM algorithm.
convtol	Numeric tolerance for testing the convergence of the EM algorithm. Convergence is assumed when the log-likelihood changes less than convtol.
protol	Numeric tolerance for the mixture proportions. Proportions below protol, considered to be zero, are not allowed.
seed	Starting value for random generator.
pertubfct	Perturbation factor used to control the degree similarity between the alternative randomly generated starting points of the EM algorithm. Increasing (decreasing) the value of pertubfct increases (decreases) the expected difference between the starting points generated.
k2max	Maximal allowed l2-norm condition number for correlation matrices. Solutions in which any component has correlation matrix with condition number above k2max, are considered to be spurious solutions and are eliminated from the EM search.
MaxVarGRt	Maximal allowed ratio of variances across components. Solutions in which any variable has a ratio between its maximal and minimal (across components) variances above MaxVarGRt, are considered to be spurious solutions and are eliminated from the EM search.

Value

An EMControl object

See Also

Idtmclust

EMControl-class — *EM algorithm control parameters for fitting Gaussian mixtures to interval data.*

Description

This class contains the control parameters for the EM algorithm used in estimation of Gaussian mixtures by function `Idtmclust`. .

Objects from the Class

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

Slots

`nrep` Number of replications (different randomly generated starting points) of the EM algorithm.

`maxiter` Maximum number of iterations in each replication of the EM algorithm.

`convtol` Numeric tolerance for testing the convergence of the EM algorithm. Convergence is assumed when the log-likelihood changes less than convtol.

`protol` Numeric tolerance for the mixture proportions. Proportions below protol, considered to be zero, are not allowed.

`seed` Starting value for random generator.

See Also

`EMControl`

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(Sdt,
    CovCase=1:4,
    SelCrit=c("BIC","AIC"),
    alpha=control@alpha,
    nsamp = control@nsamp,
    seed=control@seed,
    trace=control@trace,
    use.correction=control@use.correction,
    ncsteps=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@reweighted,
    k2max = control@k2max,
    otpType=control@otpType,
    control=RobEstControl(), ...)
```

Arguments

`Sdt`	An IData object representing interval-valued units.
`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(Sdt) 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.
`nsamp`	Number of subsets used for initial estimates.

`seed` Initial seed for random generator, like `.Random.seed`, see `"rrcov:rrcov.control"`.

`trace` Logical (or integer) indicating if intermediate results should be printed; defaults to `FALSE`.

`use.correction` whether to use finite sample correction factors; defaults to `TRUE`.

`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, the value of argument 'alpha' is used directly.

`rawMD2Dist` 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', respectivelly for the usual Chi-square, 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" respectivelly for the usual Chi-square, or the Beta and F distributions proposed by Cerioli (2010).

`eta` 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.

`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' nominal level. 'always' – testing all n entitites 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.

`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-overlaping 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-overlaping 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.

`k2max` Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.

`otpType` The amount of output returned by fasttle. Current options are "SetMD2andEst" (default) which returns an 'IdtSngNDRE' object with the fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and "SetMD2EstandPrfSt" wich returns an 'IdtSngNDRE' object with the previous slots plust 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

An object of class `IdtE` with the fasttle estimates, the value of the comparison criterion used to select the covariance configurations, the robust squared Mahalanobis distances, and optionally (if argument 'otpType' is set to true) performance statistics concerning the algorithm execution.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. *Journal of the American Statistical Association* **105** (489), 147–156.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Hardin, J. and Rocke, A. (2005), The Distribution of Robust Distances. *Journal of Computational and Graphical Statistics* **14**, 910–927.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

fulltle, RobEstControl, getIdtOutl, IdtSngNDRE

Examples

```
## 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-square 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 triming 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 triming parameter, a reweighed 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)


## End(Not run)
```

---

fulltle-methods *Methods for Function fulltle in Package 'MAINT.Data'*

Description

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

Usage

```
fulltle(Sdt, 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, k2max=1e6,
                 force = FALSE, ...)
```

Arguments

`Sdt` An IData object representing interval-valued units.

`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(Sdt) 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, the value of argument 'alpha' is used directly.

`rawMD2Dist` 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', respectivelly for the usual Chi-square, 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" respectivelly for the usual Chi-square, 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 reweightening 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 entitites 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.

`k2max` Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.

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

`...` Further arguments to be passed to internal functions of 'fulltle'.

Value

An object of class `IdtE` with the fulltle estimates, the value of the comparison criterion used to select the covariance configurations and the robust squared Mahalanobis distances.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. *Journal of the American Statistical Association* **105** (489), 147–156.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Hardin, J. and Rocke, A. (2005), The Distribution of Robust Distances. *Journal of Computational and Graphical Statistics* **14**, 910–927.

See Also

`fasttle`, `getIdtOutl`

Examples

```
## Not run:

# 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"))

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

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

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

CarsTE2 <- fulltle(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 reweighed MCD estimate
# based on Hardin and Rocke adjusted F distributions, 95% quantiles, and
# the Cerioli Beta and F distributions together with his iterated procedure
# to identify outliers in the first step.

CarsTE3 <- fulltle(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(Sdt, IdtE=NULL, muE=NULL, SigE=NULL,
  eta=0.025, Rewind=NULL, m=length(Rewind),
  RefDist=c("ChiSq","HardRockeAdjF","HardRockeAsF","CerioliBetaF"),
  multiCmpCor=c("never","always","iterstep"),
  outlin=c("MidPandLogR","MidP","LogR"),
  method=c("Distrib","AdjBoxPlot"), range=c(1.5,3.))
```

Arguments

Sdt	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 entitites 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", respectivelly for the usual Chi-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 MidPpoints, or "LogR" if outliers are only present in LogRanges.
method	The method used to fine the cutt-off used to flag outliers. Either "Distrib" if a quantile from a reference disribution is used, or "AdjBoxPlot" if the Adjusted Box Plot of Hubert and Vandervieren (2008) is adopted .
range	The multiple(s) of the modified inter-quartile range distance to the median used to identify outliers, when argument method is set to "AdjBoxPlot". When range is a two-dimensional vector, its first element is used identify moderate outliers,

and the second one to identify severe outliers. This argument is ignored when method is not equal to "AdjBoxPlot".

Value

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

References

Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. *Journal of the American Statistical Association* **105** (489), 147–156.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Hardin, J. and Rocke, A. (2005), The Distribution of Robust Distances. *Journal of Computational and Graphical Statistics* **14**, 910–927.

Hubert, M. and Vandervieren, E. (2008), An adjusted boxplot rule for skewed distributions. *Computational Statistics and Data Analysis* **52** (12), 5186–5201.

See Also

fasttle, fulltle

Examples

```
## Not run:

# 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"))

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

Carstle1 <- fulltle(CarsIdt)

# Get and display the outliers using the classical 97.5% chi-squared quantile cut-offs.

CarsOtl1 <- getIdtOutl(CarsIdt,Carstle1)
print(CarsOtl1)
plot(CarsOtl1)

# Estimate parameters by the fast trimmed maximum likelihood estimator,
# using a two-step procedure to select the trimming parameter, and a reweighed
# based on the 97.5% quantiles of Hardin and Rocke adjusted F distributions.

Carstle2 <- fulltle(CarsIdt,rawMD2Dist="HardRockeAdjF")
```

```
# Get and display the outliers using the 97.5

CarsTtl2 <- getIdtOutl(CarsIdt,Carstle2,RefDist="CerioliBetaF")
print(CarsTtl2)
plot(CarsTtl2)



## End(Not run)
```

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

```
IData(Data, trnsf = c("RngLog","AllLogit"),
Seq = c("LbUb_VarbyVar", "MidPLogR_VarbyVar", "AllLb_AllUb", "AllMidP_AllLogR"),
VarNames=NULL, ObsNames=row.names(Data), NbMicroUnits=integer(0))
```

Arguments

`Data`	A data frame or matrix of interval bounds or MidPoint/LogRange values.
`trnsf`	The data transformation to be applied to the interval ranges and (possibly) the MidPoints. Alternatives are: "RngLog" The MidPoints are left unchanged and a log tranformation is applied to the ranges. "AllLogit" A logit transformation is applied to both the MidPoints and the ranges. This option is particularly usefull for dara restricted to the (0,1) domain.
`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.

`NbMicroUnits` An integer vector with the number of micro data units by interval-valued observation (or an empty vector, if not applicable)

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`, `AgrMcDt`

Examples

```
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 three observations:\n")
print(head(ChinaT,n=3))
cat("\n...\n")
print(tail(ChinaT,n=3))
```

`IData-class` *Class IData*

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

`MidP`**:** 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.

`NObs`**:** Number of entities under analysis (cases)

`NIVar`**:** Number of interval variables

`NbMicroUnits`**:** An integer vector with the number of micro data units by interval-valued observation (or an empty vector, if not applicable)

`Transform`**:** A string specifying the data transformation to be applied to the interval ranges and (possibly) the MidPoints.

Methods

show `signature(object = "IData")`: show S4 method for the IData-class.

nrow `signature(x = "IData")`: returns the number of statistical units (observations).

ncol `signature(x = "IData")`: returns the number of of Interval-valued variables.

dim `signature(x = "IData")`: returns a vector with the of number statistical units as first element, and the number of Interval-valued variables as second element.

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.

names `signature(x = "IData")`: returns column (variable) names for an object of class IData.

MidPoints `signature(Sdt = "IData")`: returns a data frame with MidPoints for an object of class IData.

LogRanges `signature(Sdt = "IData")`: returns a data frame with LogRanges for an object of class IData.

Ranges `signature(Sdt = "IData")`: returns an data frame with Ranges for an object of class IData.

NbMicroUnits `signature(Sdt = "IData")`: returns an integer vector with the number of micro data units by interval-valued observation 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.

plot `signature(x = "IData")`: plot S4 methods for the IData-class.

mle `signature(x = "IData")`: Maximum likelihood estimation.

fasttle `signature(x = "IData")`: Fast trimmed maximum likelihood estimation.

fulltle `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

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Noirhomme-Fraiture, M., Brito, P. (2011), Far Beyond the Classical Data Models: Symbolic Data Analysis. *Statistical Analysis and Data Mining* **4**(2), 157–170.

See Also

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

---

IdtE-class *Class IdtE*

Description

IdtE contains estimation results for the models assumed for single distributions, or mixtures of distributions, underlying data sets of interval-valued entities.

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

logLiks: 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(Sdt = "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

summary `signature(object = "IdtE")`: summary S4 method for the IDtE-class

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

sd `signature(Sdt = "IdtE")`: extracts the standard deviation estimates from objects of class IdtE.

AIC `signature(Sdt = "IdtE")`: extracts the value of the Akaike Information Criterion from objects of class IdtE.

BIC `signature(Sdt = "IdtE")`: extracts the value of the Bayesian Information Criterion from objects of class IdtE.

logLik `signature(Sdt = "IdtE")`: extracts the value of the maximised log-likelihood from objects of class IdtE.

mean `signature(x = "IdtE")`: extracts the mean vector estimate from objects of class IdtE

var `signature(x = "IdtE")`: extracts the variance-covariance matrix estimate from objects of class IdtE

cor `signature(x = "IdtE")`: extracts the correlation matrix estimate from objects of class IdtE

Author(s)

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

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

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 IDdtlda-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

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

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., homoscedastic gaussian) setup.

IdtHetNMANOVA extends IdtMANOVA, assuming a heteroscedastic gaussian set-up.

IdtLocSNMANOVA extends IdtMANOVA, assuming a Skew-Normal location model set-up.

IdtLocNSNMANOVA extends IdtMANOVA, assuming either a homoscedastic 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 heteroscedastic 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.

`ChiSq`**:** Inherited from class `LRTest`. Value of the Chi-Square statistics corresponding to the performed test.

`df`**:** Inherited from class `LRTest`. Degrees of freedom of the Chi-Square statistics.

`pvalue`**:** Inherited from class `LRTest`. p-value of the Chi-Square statistics value, obtained from the Chi-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.

summary `signature(object = "IdtMANOVA")`: summary 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 = "IdtClMANOVA")`: Linear Discriminant Analysis using the estimated model parameters.

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

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

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

snda `signature(x = "IdtLocNSNMANOVA")`: 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 = "IdtGenSNMANOVA")`: Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a general model (i.e., groups differ in all parameters).

snda `signature(x = "IdtGenNSNMANOVA")`: 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.

Author(s)

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

References

Brito, P., Duarte Silva, A. P. (2012): "Modelling Interval Data with Normal and Skew-Normal Distributions". Journal of Applied Statistics, Volume 39, Issue 1, 3-20.

See Also

`MANOVA`, `lda`, `qda`, `snda`, `IData`

`IdtMclust-class` *Class IdtMclust*

Description

IdtMclust contains the results of fitting mixtures of Gaussian distributions to interval data represented by objects of class `IData`.

Slots

`call`: The matched call that created the IdtMclust object

`data`: The IData data object

`NObs`: Number of entities under analysis (cases)

`NIVar`: Number of interval variables

`SelCrit`: The model selection criterion; currently, AIC and BIC are implemented

`Hmcdt`: Indicates whether the optimal model corresponds to a homoscedastic (TRUE) or a hetereocedasic (FALSE) setup

`BestG`: The optimal number of mixture components.

`BestC`: The configuration case of the variance-covariance matrix in the optimal model

`logLiks`: The logarithms of the likelihood function for the different models tried

`logLik`: The logarithm of the likelihood function for the optimal model

`AICs`: The values of the AIC criterion for the different models tried

`aic`: The value of the AIC criterion for the he optimal model

`BICs`: The values of the BIC criterion for the different models tried

`bic`: The value of the BIC criterion for the he optimal model

`parameters` A list with the following components:

- **pro** A vector whose *k*th component is the mixing proportion for the *k*th component of the mixture model.
- **mean** The mean for each component. If there is more than one component, this is a matrix whose kth column is the mean of the *k*th component of the mixture model.
- **covariance** A three-dimensional array with the covariance estimates. If Hmcdt is FALSE (heteroscedastic setups) the third dimension levels run through the BestG mixture components, with one different covariance matrix for each level. Otherwise (homoscedastic setups), there is only one covariance matrix and the size of the third dimension equals one.

`z`: A matrix whose *[i,k]*th entry is the probability that observation *i* in the test data belongs to the *k*th class.

`classification`: The classification corresponding to `z`, i.e. `map(z)`.

`allres`: A list with the detailed results for all models fitted.

Methods

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

summary `signature(object = "IdtMclust")`: summary S4 method for the IdtMclust-class

parameters `signature(x = "IdtMclust")`: retrieves the value of the parameter estimates for the obtained partition

pro `signature(x = "IdtMclust")`: retrieves the value of the estimated mixing proportions for the obtained partition

mean `signature(x = "IdtMclust")`: retrieves the value of the component means for the obtained partition

var `signature(x = "IdtMclust")`: retrieves the value of the estimated covariance matrices for the obtained partition

cor `signature(x = "IdtMclust")`: retrieves the value of the estimated correlation matrices

classification `signature(x = "IdtMclust")`: retrieves the individual class assignments for the obtained partition

SelCrit `signature(x = "IdtMclust")`: retrieves a string specifying the criterion used to find the best model and partition

Hmcdt `signature(x = "IdtMclust")`: returns TRUE if an homecedastic model has been assumed, and FALSE otherwise

BestG `signature(x = "IdtMclust")`: returns the number of components selectd

BestC `signature(x = "IdtMclust")`: retruns the covariance configuration selected

PostProb `signature(x = "IdtMclust")`: retrieves the estimates of the individual posterir probabilities for the obtained partition

BIC `signature(x = "IdtMclust")`: returns the value of the BIC criterion

AIC `signature(x = "IdtMclust")`: returns the value of the AIC criterion

logLik `signature(x = "IdtMclust")`: returns the value of the log-likelihood

Author(s)

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

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Brito, P., Duarte Silva, A. P. and Dias, J. G. (2015), Probabilistic Clustering of Interval Data. *Intelligent Data Analysis* **19**(2), 293–313.

See Also

`Idtmclust`, `plotInfCrt`, `pcoordplot`

`Idtmclust-methods` *Methods for function Idtmclust in Package 'MAINT.Data'*

Description

Performs Gaussian model based clustering for interval data

Usage

```
Idtmclust(Sdt, G = 1:9, CovCase=1:4, SelCrit=c("BIC","AIC"),
  Mxt=c("Hom","Het","HomandHet"), control=EMControl())
```

Arguments

`Sdt`	An IData object representing interval-valued entities.
`G`	An integer vector specifying the numbers of mixture components (clusters) for which the BIC is to be calculated.
`CovCase`	Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
`SelCrit`	The model selection criterion.
`control`	A list of control parameters for EM. The defaults are set by the call `EMControl()`.
`Mxt`	The type of Gaussian mixture assumed by Idtmclust. Alternatives are "Hom" (default) for homoscedastic mixtures, "Het" for heteroscedastic mixtures, and "HomandHet" for both homoscedastic and heteroscedastic mixtures.

Value

An object of class `IdtMclust` providing the optimal (according to BIC) mixture model estimation.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Brito, P., Duarte Silva, A. P. and Dias, J. G. (2015), Probabilistic Clustering of Interval Data. *Intelligent Data Analysis* **19**(2), 293–313.

Fraley, C., Raftery, A. E., Murphy, T. B. and Scrucca, L. (2012), mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. Technical Report No. 597, Department of Statistics, University of Washington.

See Also

`IdtMclust`, `EMControl`, `EMControl`, `plotInfCrt`, `pcoordplot`

Examples

```
## Not run:

# Create an Interval-Data object containing the intervals of loan data
# (from the Kaggle Data Science platform) aggregated by loan purpose

LbyPIdt <- IData(LoansbyPurpose_minmaxDt,
                 VarNames=c("ln-inc","ln-revolbal","open-acc","total-acc"))

print(LbyPIdt)

#Fit homoscedastic Gaussian mixtures with up to nine components

mclustres <- Idtmclust(LbyPIdt)
plotInfCrt(mclustres,legpos="bottomright")
```

```
print(mclustres)

#Display the results of the best mixture according to the BIC

summary(mclustres,parameters=TRUE,classification=TRUE)
pcoordplot(mclustres)

#Repeat the analysus with both homoscedastic and heteroscedastic mixtures up to six components

mclustres1 <- Idtmclust(LbyPIdt,G=1:6,Mxt="HomandHet")
plotInfCrt(mclustres1,legpos="bottomright")
print(mclustres1)

#Display the results of the best heteroscedastic mixture according to the BIC

summary(mclustres1,parameters=TRUE,classification=TRUE,model="HetG2C2")


## End(Not run)
```

IdtMxE-class *Class IdtMxE*

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

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 distribution were estimated. Always set to FALSE in objects of class "IdtMxE"

`Ngrps`: Number of mixture components

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

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`IdtE`, `IdtSngDE`, `IData`, `MANOVA`, `RobMxtDEst`

`IdtMxNandSNDE-class` *Class IdtMxNandSNDE*

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

`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 distribution 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

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`IdtE`, `IdtMxE`, `IdtSngNandSNDE`, `MANOVA`, `RobMxtDEst`, `IData`

IdtMxNDE-class *Class IdtMxNDE*

Description

IdtMxNDE contains the results of a mixture Normal model maximum likelihood parameter estimation, with the four different possible variance-covariance configurations.

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

`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 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 = "IdtMxtNDE")`: Linear Discriminant Analysis using the estimated model parameters.

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

Author(s)

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

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

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 hetereocedasic 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

`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 distribution were estimated. Always set to FALSE in objects of class IdtMxNDRE

`Ngrps`: Inherited from class `IdtMxE`. Number of mixture components

`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 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.

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

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

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 homoscedastic 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

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 distribution 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

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

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 *Class IdtNandSNDE*

Description

IdtNandSNDE is a union of classes IdtSngNandSNDE and IdtMxNandSNDE, used for storing the estimation results of Normal and Skew-Normal modelisations for Interval Data.

Methods

coef signature(coef = "IdtNandSNDE"): extracts parameter estimates from objects of class IdtNandSNDE

stdEr signature(x = "IdtNandSNDE"): extracts standard errors from objects of class IdtNandSNDE

vcov signature(x = "IdtNandSNDE"): extracts an estimate of the variance-covariance matrix of the parameters estimators for objects of class IdtNandSNDE

mean signature(x = "IdtNandSNDE"): extracts the mean vector estimate from objects of class IdtNandSNDE

var signature(x = "IdtNandSNDE"): extracts the variance-covariance matrix estimate from objects of class IdtNandSNDE

cor signature(x = "IdtNandSNDE"): extracts the correlation matrix estimate from objects of class IdtNandSNDE

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

IData, mle, fasttle, fulltle, MANOVA, RobMxtDEst, IdtSngNandSNDE, IdtMxNandSNDE

IdtNDE-class *Class IdtNDE*

Description

IdtNDE is a a union of classes IdtSngNDE, IdtSngNDRE, IdtMxNDE and IdtMxNDRE, used for storing the estimation results of Normal modelizations for Interval Data.

Methods

coef `signature(coef = "IdtNDE")`: extracts parameter estimates from objects of class IdtNDE

stdEr `signature(x = "IdtNDE")`: extracts standard errors from objects of class IdtNDE

vcov `signature(x = "IdtNDE")`: extracts an estimate of the variance-covariance matrix of the parameters estimators for objects of class IdtNDE

mean `signature(x = "IdtNDE")`: extracts the mean vector estimate from objects of class IdtNDE

var `signature(x = "IdtNDE")`: extracts the variance-covariance matrix estimate from objects of class IdtNDE

cor `signature(x = "IdtNDE")`: extracts the correlation matrix estimate from objects of class IdtNDE

sd `signature(Idt = "IdtNDE")`: extracts the standard deviation estimates from objects of class IdtNDE.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`IdtSngNDE`, `IdtSngNDRE`, `IdtMxNDE`, `IdtMxNDRE`, `IdtSNDE`, `IData`, `mle`, `fasttle`, `fulltle`, `MANOVA`, `RobMxtDEst`

`IdtOutl-class` *Class IdtOutl*

Description

A description of interval-valued variable outliers found by the MAINT.Data function `getIdtOutl`.

Slots

`outliers`**:** A vector of indices of the interval data units flaged as outliers.

`MD2`**:** A vector of squared robust Mahalanobis distances for all interval data units.

eta Nominal size of the null hypothesis that a given observation is not an outlier.

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

multiCmpCor Whether a multicomparison correction of the nominal size (eta) for the outliers tests was performed. Alternatives are: 'never' – ignoring the multicomparisons and testing all entities at the 'eta' nominal level. 'always' – testing all n entitites at 1.- (1.-'eta'^(1/n)).

NObs Number of original observations in the original data set.

p Number of total numerical variables (MidPoints and/or LogRanges) that may be responsible for the outliers.

h Size of the subsets over which the trimmed likelihood was maximized when computing the robust Mahalanobis distances.)

boolRewind A logical vector indicanting which of the data units belong to the final trimmed subsetused to compute the tle estimates.)

trshld The treshold(s) used on (squared) Mahalanobis distances to flag interval data unist as outliers

Methods

show `signature(object = "IdtOutl")`: show S4 method for the IdtOutl-class.

plot `signature(x = "IdtOutl")`: plot S4 methods for the IdtOutl-class.

getMahaD2 `signature(x = "IdtOutl")`: retrieves the vector of squared robust Mahalanobis distances for all data units.

geteta `signature(x = "IdtOutl")`: retrieves the nominal size of the null hypothesis used to flag observations as outliers.

getRefDist `signature(x = "IdtOutl")`: retrieves the assumed reference distributions used to find cutoffs defining the observations assumed as outliers.

getmultiCmpCor `signature(x = "IdtOutl")`: retrieves the multicomparison correction used when flaging observations as outliers.

Author(s)

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

References

Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. *Journal of the American Statistical Association* **105** (489), 147–156.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

See Also

`getIdtOutl`, `fasttle`, `fulltle`

`IdtOutl-methods` *Plot method for class IdtOutl in Package 'MAINT.Data'*

Description

Plots robust Mahalanobis distances and outlier cut-offs for an object describing potential outliers in a interval-valued data set

Usage

```
  ## S4 method for signature 'IdtOutl,missing'
plot(x, scale=c("linear","log"), RefDist=getRefDist(x), eta=geteta(x),
  multiCmpCor=getmultiCmpCor(x), Obsnames=TRUE, sqrddist=TRUE, ...)
```

Arguments

`x`	An IData object of class IdtOutl describing potential interval-valued ouliters.
`scale`	The scale of the axis for the robust Mahalanobis distances.
`RefDist`	The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are "ChiSq" and "CerioliBetaF" respectivelly for the usual Chi-squared, and the Beta and F distributions proposed by Cerioli (2010). By default uses the one selected in the creation of the object 'x'.
`eta`	Nominal size of the null hypothesis that a given observation is not an outlier. By default uses the one selected in the creation of the object 'x'.
`multiCmpCor`	Whether a multicomparison correction of the nominal size (eta) for the outliers tests was performed. Alternatives are: 'never' – ignoring the multicomparisons and testing all entities at the 'eta' nominal level. 'always' – testing all n entitites at 1.- (1.-'eta'^(1/n)). By default uses the one selected in the creation of the object 'x'.
`sqrddist`	A booleand flag indicating if whether the squared Mahalanobis distances (default), or only the distances by themselves should be displayed.
`Obsnames`	A booleand flag indicating if the observation names are to be displayed along the horizontal axis.
`...`	Further arguments to be passed to methods.

References

Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. *Journal of the American Statistical Association* **105** (489), 147–156.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.
Journal of Computational and Graphical Statistics **14**, 910–927.

See Also

getIdtOutl, fasttle, fulltle

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(object = "Idtqda")`: Classifies interval-valued observations in conjunction with qda.

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

CovCase `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

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

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 paramenters 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

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

IData, mle, MANOVA, IdtSngSNDE, IdtMxSNDE, IdtNDE

IdtSNgenda-class *Class "IdtSNgenda"*

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 sekwness parameter ("eta") estimates for each group.

`scaling`: For each group g, 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 sekwness parameter ("gamma1") estimates for each group.

`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(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>

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

`MANOVA`, `snda`, `IData`

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

`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 TRUE in objects of class IdtSngNandSNDE

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

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

IData, IdtMxNandSNDE, mle, fasttle, fulltle

IdtSngNDE-class *Class IdtSngNDE*

Description

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

Slots

mleNmuE: Vector with the maximum likelihood mean vectors estimates

mleNmuEse: Vector with the maximum likelihood means' standard errors

CovConfCases: List of the considered configurations

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 IdtSngNDE 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 distribution were estimated. Always set to TRUE in objects of class IdtSngNDE

Extends

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

Methods

No methods defined with class IdtSngNDE in the signature.

Author(s)

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

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

IData, mle, IdtSngNDRE, IdtSngSNDE, IdtMxNDE

IdtSngNDRE-class *Class IdtSngNDRE*

Description

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

Slots

RobNmuE: Matrix with the maximum likelihood mean vectors estimates

CovConfCases: List of the considered configurations

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 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 distribution were estimated. Always set to TRUE in objects of class IdtSngNDRE

`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 tle 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 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.

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

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

See Also

`IData`, `fasttle`, `fulltle`, `IdtSngNDE`, `IdtMxNDRE`

IdtSngSNDE-class *Class IdtSngSNDE*

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 distribution 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>

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

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

`IdtSNlocda-class` *Class "IdtSNlocda"*

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 snda.

show `signature(object = "IdtSNlocda")`: show S4 method for the IDdtlda-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

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

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

```
## 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, k2max=1e6, ... )

## S4 method for signature 'IdtMxtNDE'
lda(x, prior="proportions", selmodel=BestModel(x), egvtol=1.0e-10,
  silent=FALSE, k2max=1e6, ... )

## S4 method for signature 'IdtClMANOVA'
lda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  egvtol=1.0e-10, silent=FALSE, k2max=1e6, ... )

## S4 method for signature 'IdtLocNSNMANOVA'
lda( x, prior="proportions",
 selmodel=BestModel(H1res(x)@NMod), egvtol=1.0e-10, silent=FALSE, k2max=1e6, ... )
```

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 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 whether a warning message should be printed if the method fails.
selmodel	Selected model from a list of candidate models saved in object x.
k2max	Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.
...	Other named arguments.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

qda, snda, Roblda, Robqda, IData, IdtMxtNDE, IdtClMANOVA, IdtLocNSNMANOVA, qda, ConfMat

Examples

```
# 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$GeoReg)
cat("Temperatures of China -- linear discriminant analysis results:\n")
print(ChinaT.lda)
ldapred <- predict(ChinaT.lda,ChinaT)$class
```

```
cat("lda Prediction results:\n")
print(ldapred )
cat("Resubstition confusion matrix:\n")
ConfMat(ChinaTemp$GeoReg,ldapred)

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

CVlda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=lda,CovCase=CovCase(ChinaT.lda))
summary(CVlda[,,"Clerr"])

## End(Not run)
```

`LoansbyPurpose_minmaxDt`

Loans by purpose: minimum and maximum Data Set

Description

This data set consist of the lower and upper bounds of the intervals for four interval characteristics of the loans aggregated by their purpose. The original microdata is available at the Kaggle Data Science platform and consists of 887 383 loan records characterized by 75 descriptors. Among the large set of variables available, we focus on borrowers' income and account and loan information aggregated by the 14 loan purposes, wich are considered as the units of interest.

Usage

```
data(LoansbyPurpose_minmaxDt)
```

Format

A data frame containing 14 observations on the following 8 variables.

ln-inc_min The minimum, for the current loan purpose, of natural logarithm of the self-reported annual income provided by the borrower during registration.

ln-inc_max The maximum, for the current loan purpose, of natural logarithm of the self-reported annual income provided by the borrower during registration.

ln-revolbal_min The minimum, for the current loan purpose, of natural logarithm of the total credit revolving balance.

ln-revolbal_max The maximum, for the current loan purpose, of natural logarithm of the total credit revolving balance.

open-acc_min The minimum, for the current loan purpose, of the number of open credit lines in the borrower's credit file.

open-acc_max The maximum, for the current loan purpose, of the number of open credit lines in the borrower's credit file.

total-acc_min The minimum, for the current loan purpose, of the total number of credit lines currently in the borrower's credit file.

total-acc_max The maximum, for the current loan purpose, of the total number of credit lines currently in the borrower's credit file.

Source

https:www.kaggle.com/wendykan/lending-club-loan-data

`LoansbyRiskLvs_minmaxDt`

Loans by risk levels: minimum and maximum Data Set

Description

This data set consist of the lower and upper bounds of the intervals for four interval characteristics for 35 risk levels (from A1 to G5) of loans. The original microdata is available at the Kaggle Data Science platform and consists of 887 383 loan records characterized by 75 descriptors. Among the large set of variables available, we focus on borrowers' income and account and loan information aggregated by the 35 risk levels wich are considered as the units of interest.

Usage

```
data(LoansbyRiskLvs_minmaxDt)
```

Format

A data frame containing 35 observations on the following 8 variables.

ln-inc_min The minimum, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.

ln-inc_max The maximum, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.

int-rate_min The minimum, for the current risk category, of the interest rate on the loan.

int-rate_max The maximum, for the current risk category, of the interest rate on the loan.

open-acc_min The minimum, for the current risk category, of the number of open credit lines in the borrower's credit file.

open-acc_max The maximum, for the current risk category, of the number of open credit lines in the borrower's credit file.

total-acc_min The minimum, for the current risk category, of the total number of credit lines currently in the borrower's credit file.

total-acc_max The maximum, for the current risk category, of the total number of credit lines currently in the borrower's credit file.

Source

https:www.kaggle.com/wendykan/lending-club-loan-data

LoansbyRiskLvs_qntlDt *Loans by risk levels: ten and ninety per cent quantiles Data Set*

Description

This data set consist of the ten and ninety per cent quantiles of the intervals for four interval characteristics for 35 risk levels (from A1 to G5) of loans. The original microdata is available at the Kaggle Data Science platform and consists of 887 383 loan records characterized by 75 descriptors. Among the large set of variables available, we focus on borrowers' income and account and loan information aggregated by the 35 risk levels wich are considered as the units of interest.

Usage

```
data(LoansbyRiskLvs_qntlDt)
```

Format

A data frame containing 35 observations on the following 8 variables.

ln-inc_q0.10 The ten percent quantile, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.

ln-inc_q0.90 The ninety percent quantile, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.

int-rate_q0.10 The ten percent quantile, for the current risk category, of the interest rate on the loan.

int-rate_q0.90 The ninety percent quantile, for the current risk category, of the interest rate on the loan.

open-acc_q0.10 The ten percent quantile, for the current risk category, of the number of open credit lines in the borrower's credit file.

open-acc_q0.90 The ninety percent quantile, for the current risk category, of the number of open credit lines in the borrower's credit file.

total-acc_q0.10 The ten percent quantile, for the current risk category, of the total number of credit lines currently in the borrower's credit file.

total-acc_q0.90 The ninety percent quantile, for the current risk category, of the total number of credit lines currently in the borrower's credit file.

Source

https:www.kaggle.com/wendykan/lending-club-loan-data

LRTest-class *Class LRTest*

Description

LRTest contains the results of likelihood ratio tests

Slots

`ChiSq`: Value of the Chi-Square statistics corresponding to the performed test

`df`: Degrees of freedom of the Chi-Square statistics

`pvalue`: p-value of the Chi-Square statistics value, obtained from the Chi-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 homoscedastic or heteroscedastic setups. Methods H0res and H1res retrieve the model estimates under the null and alternative hypothesis, and method show displays the MANOVA results.

Usage

```
MANOVA(Sdt, grouping, Model=c("Normal","SKNormal","NrmandSKN"), CovCase=1:4,
SelCrit=c("BIC","AIC"), Mxt=c("Hom","Het","Loc","Gen"),
                CVtol=1.0e-5, k2max=1e6,
OptCntrl=list(), 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 units.
`Sdt`	An IData object representing interval-valued units.
`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" (homoscedastic) and "Het" (heteroscedastic) 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.
`k2max`	Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.
`OptCntrl`	List of optional control parameters to be 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])

#Classical (homoscedastic) MANOVA tests

ManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg)
cat("China, MANOVA by geografical regions results =\n")
print(ManvChina)

#Heteroscedastic MANOVA tests

HetManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,Mxt="Het")
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")
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")
cat("China, Skew-Normal MANOVA (general model)  by geografical regions results =\n")
print(SKNGenManvChina)


## End(Not run)
```

MANOVAPermTest *MANOVA permutation test*

Description

Function MANOVAPermTest performs a MANOVA permutation test allowing for both Gaussian and Skew-Normal distributions and homoscedastic or heteroscedastic setups.

Usage

```
MANOVAPermTest(MANOVAres, Sdt, grouping, nrep=200,
    Model=c("Normal","SKNormal","NrmandSKN"), CovCase=1:4,
   SelCrit=c("BIC","AIC"), Mxt=c("Hom","Het","Loc","Gen"), CVtol=1.0e-5, k2max=1e6,
    OptCntrl=list(), onerror=c("stop","warning","silentNull"), ...)
```

Arguments

MANOVAres	An object representing a MANOVA analysis on interval-valued entities.
Sdt	An IData object representing interval-valued entities.
grouping	Factor indicating the group to which each observation belongs to.
nrep	Number of random generated permutations used to approximate the null distribution of the likelihood ratio statistic.
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" (heteroscedastic) 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.
k2max	Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.
OptCntrl	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.

Details

Function MANOVAPermTest performs a MANOVA permutation test allowing for both Gaussian and Skew-Normal distributions and homoscedastic or heteroscedastic setups. This test is implemented by simulating the null distribution of the MANOVA likelihood ratio statistic, using many random permutations of the observation group labels. It is intended as an alternative of the classical Chi-squares based MANOVA likelihood ratio tests, when small sample sizes cast doubt on the applicability of the Chi-squared distribution. We note that this test may be computationally intensive, in particular when used for the Skw-Normal model.

Value

the p-value of the MANOVA permutation test.

See Also

MANOVA, IdtMANOVA

Examples

```
## Not run:

#Perform a MANOVA of the AbaloneIdt data set, comparing the Abalone variable means
# according to their age

# Create an Interval-Data object containing the Length, Diameter, Height, Whole weight,
# Shucked weight, Viscera weight (VW), and Shell weight (SeW) of 4177 Abalones,
# aggregated by sex and age.
# Note: The original micro-data (imported UCI Machine Learning Repository Abalone dataset)
# is given in the AbaDF data frame, and the corresponding values of the sex by age combinations
# is represented by the AbUnits factor.

AbaloneIdt <- AgrMcDt(AbaDF,AbUnits)

# Create a factor with three levels (Young, Adult and Old) for Abalones with respectively
# less than 10 rings, between 11 and 18 rings, and more than 18 rings.

Agestrg <- substring(rownames(AbaloneIdt),first=3)
AbalClass <- factor(ifelse(Agestrg=="1-3"|Agestrg=="4-6"| Agestrg=="7-9","Young",
  ifelse(Agestrg=="10-12"|Agestrg=="13-15"| Agestrg=="16-18","Adult","Old") ) )

#Perform a classical MANOVA, computing the p-value from the asymptotic Chi-squared distribution
# of the Wilk's lambda statistic

MANOVAres <- MANOVA(AbaloneIdt,AbalClass)
summary(MANOVAres)

#Find a finite sample p-value of the test statistic, using a permutation test.

MANOVAPermTest(MANOVAres,AbaloneIdt,AbalClass)


## End(Not run)
```

---

mean-methods *Methods for function mean in Package 'MAINT.Data'*

Description

S4 methods for function mean. These methods extract estimates of mean vectors for the models fitted to Interval Data.

Usage

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

sd var cor

---

mle-methods *Methods for function mle in Package 'MAINT.Data'*

Description

Performs maximum likelihood estimation for parametric models of interval data

Usage

```
## S4 method for signature 'IData'
mle(Sdt, Model="Normal", CovCase="AllC", SelCrit=c("BIC","AIC"),
  k2max=1e6, OptCntrl=list(), ...)
```

Arguments

`Sdt`	An IData object representing interval-valued units.
`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: The string "AllC" for all possible configurations (default), or a set of integers between 1 and 4.
`SelCrit`	The model selection criterion.
`k2max`	Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.
`OptCntrl`	List of optional control parameters to be passed to the optimization routine. See the documentation of RepLOptim for a description of the available options.
`...`	Other named arguments.

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012): "Modelling Interval Data with Normal and Skew-Normal Distributions". Journal of Applied Statistics, Volume 39, Issue 1, 3-20.

See Also

`IData`, `RepLOptim`

Examples

```
# 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 a Gaussian distribution

ChinaE <- mle(ChinaT)
cat("China maximum likelhiood estimation results =\n")
print(ChinaE)
cat("Standard Errors of Estimators:\n")
print(stdEr(ChinaE))
```

nycflights *New York City flights Data Set*

Description

A interval-valued data set containing 142 units and four interval-valued variables (dep_delay, arr_delay, air_time and distance), created from from the flights data set in the R package nycflights13 (on-time data for all flights that departed the JFK, LGA or EWR airports in 2013), after removing all rows with missing observations, and aggregating by month and carrier.

Usage

```
data(nycflights)
```

Format

FlightsDF: A data frame containing the original 327346 valid (i.e. with non missing values) flights from the nycflights13 package, described by the 4 variables: dep_delay, arr_delay, air_time and distance.
FlightsUnits: A factor with 327346 observations and 142 levels, indicating the month by carrier combination to which each orginal flight belongs to.
FlightsIdt: An IData object with 142 observations and 4 interval-valued variables, describing the intervals formed by agregating the FlightsDF microdata by the 0.05 and 0.95 quantiles of the subsamples formed by FlightsUnits factor.

pcoordplot-methods *Parallel coordinates plot.*

Description

Method pcoordplot displays a parallel coordinates plot, representing the results stored in an IdtMclust-method object.

Usage

```
## S4 method for signature 'IdtMclust'
pcoordplot(x,title="Parallel Coordinate Plot",
Seq=c("AllMidP_AllLogR","MidPLogR_VarbyVar"), model ="BestModel",
legendpar=list(), Components = "All", ...)
```

Arguments

x	An object of type "IdtMclust" representing the the clusterig results of an Interval-valued data set obtained by the function "IdtMclust".
title	The title of the plot.
Seq	The ordering of the coordinates in the plot. Available options are: "AllMidP_AllLogR": all MidPoints followed all LogRanges, in the same variable order. "MidPLogR_VarbyVar": MidPoints followed by LogRanges, variable by variable.
model	A character vector specifying the the model whose solution is to be displayed.
legendpar	A named list with graphical parameters for the plot legend. Currently only the base R 'cex.main' and 'cex.lab' parameters are implemented.
...	Graphical arguments to be passed to methods
Components	The set of components to be compared

See Also

IdtMclust, Idtmclust, plotInfCrt

Examples

```
## Not run:

# Create an Interval-Data object containing the intervals of loan data
# (from the Kaggle Data Science platform) aggregated by loan purpose

LbyPIdt <- IData(LoansbyPurpose_minmaxDt,
                 VarNames=c("ln-inc","ln-revolbal","open-acc","total-acc"))


#Fit homoscedastic Gaussian mixtures with up to ten components

mclustres <- Idtmclust(LbyPIdt,G=1:10)
plotInfCrt(mclustres,legpos="bottomright")

#Display the results of the best mixture according to the BIC

pcoordplot(mclustres)
pcoordplot(mclustres,model="HomG6C1")
pcoordplot(mclustres,model="HomG4C1")



## End(Not run)
```

plot-methods *Methods for function plot in Package 'MAINT.Data'*

Description

S4 methods for function plot. As in the generic plot S3 'graphics' method, these methods plot Interval-valued data contained in IData objects.

Usage

```
## S4 method for signature 'IData,IData'
plot(x, y, type=c("crosses","rectangles"), append=FALSE, ...)
## S4 method for signature 'IData,missing'
plot(x, casen=NULL, layout=c("vertical","horizontal"), append=FALSE, ...)
```

Arguments

x	An object of type IData representing the values of an Interval-value variable.
y	An object of type IData representing the values of a second Interval-value variable, to be displayed along y (vertical) coordinates.
type	What type of plot should de drawn. Alternatives are "crosses" (default) and "rectangles".
append	A boolean flag indicating if the interval-valued variables should be displayed in a new plot, or added to an existing plot.
casen	An optional character string with the case names.
layout	The axes along which the interval-valued variables be displayed. Alternatives are "vertical" (default) and "horizontal".
...	Graphical arguments to be passed to methods.

See Also

IData

Examples

```
## Not run:

# Create an Interval-Data object containing the Length, Diameter, Height, Whole weight,
# Shucked weight, Viscera weight (VW), and Shell weight (SeW) of 4177 Abalones,
# aggregated by sex and age.
# Note: The original micro-data (imported UCI Machine Learning Repository Abalone dataset)
# is given in the AbaDF data frame, and the corresponding values of the sex by age combinations
# is represented by the AbUnits factor.

AbaloneIdt <- AgrMcDt(AbaDF,AbUnits)
```

```
# Dispaly a plot of the Length versus the Whole_weight interval variables

plot(AbaloneIdt[,"Length"],AbaloneIdt[,"Whole_weight"])
plot(AbaloneIdt[,"Length"],AbaloneIdt[,"Whole_weight"],type="rectangles")

# Display the Abalone lengths using different colors to distinguish the Abalones age
# (measured by the number of rings)

# Create a factor with three levels (Young, Adult and Old) for Abalones with
# respectively less than 10 rings, between 11 and 18 rings, and more than 18 rings.

Agestrg <- substring(rownames(AbaloneIdt),first=3)
AbalClass <- factor(ifelse(Agestrg=="1-3"|Agestrg=="4-6"| Agestrg=="7-9","Young",
  ifelse(Agestrg=="10-12"|Agestrg=="13-15"| Agestrg=="16-18","Adult","Old") ) )

plot(AbaloneIdt[AbalClass=="Young","Length"],col="blue",layout="horizontal")
plot(AbaloneIdt[AbalClass=="Adult","Length"],col="green",layout="horizontal",append=TRUE)
plot(AbaloneIdt[AbalClass=="Old","Length"],col="red",layout="horizontal",append=TRUE)
legend("bottomleft",legend=c("Young","Adult","Old"),col=c("blue","green","red"),lty=1)



## End(Not run)
```

`plotInfCrt-methods` *Information criteria plot.*

Description

Method plotInfCrt displays a plot representing the values of an appropriate information criterion (currently either BIC or AIC) for the models whose results are stored in an IdtMclust-method object. A supplementary short output message prints the values of the chosen criterion for the 'nprin' best models.

Usage

```
## S4 method for signature 'IdtMclust'
plotInfCrt(object, crt=object@SelCrit, legpos="right", nprnt=5,
  legendout=TRUE, outlegsize="adjstoscreen", outlegdisp="adjstoscreen",
  legendpar=list(), ...)
```

Arguments

`object`	An object of type "IdtMclust" representing the the clusterig results of an Interval-valued data set obtained by the function "IdtMclust".
`crt`	The information criteria whose values are to be displayed.

legpos	Legend position. Alternatives are "right" (default), "left", "bottomright", "bottomleft", "topright" and "topleft" .
nprnt	Number of solutions for which the value of the information criterio should be printed in an suplmentary short output message.
legendout	A boolean flag indicating if the legend should be placed outside (default) or inside the main plot.
outlegsize	The size (in inches) to be reserved for a legend placed outside the main plot, or the string "adjstoscreen" (default) for an automatic adjustment of the plot and legend sizes.
outlegdisp	The displacement (as a percentage of the main plot size) of the outer margin for a legend placed outside the main plot, or the string "adjstoscreen" (default) for an automatic adjustment of the legend position.
legendpar	A named list with graphical parameters for the plot legend.
...	Graphical arguments to be passed to methods.

See Also

IdtMclust, Idtmclust, pcoordplot

Examples

```
## Not run:

# Create an Interval-Data object containing the intervals of loan data
# (from the Kaggle Data Science platform) aggregated by loan purpose

LbyPIdt <- IData(LoansbyPurpose_minmaxDt,
                 VarNames=c("ln-inc","ln-revolbal","open-acc","total-acc"))

#Fit homoscedastic and heteroscedastic mixtures up to Gaussian mixtures with up to seven components

mclustres <- Idtmclust(LbyPIdt,G=1:7,Mxt="HomandHet")

#Compare de model fit according to the BIC

plotInfCrt(mclustres,legpos="bottomleft")

#Display the results of the best three mixtures according to the BIC

summary(mclustres,parameters=TRUE,classification=TRUE)
pcoordplot(mclustres)
summary(mclustres,parameters=TRUE,classification=TRUE,model="HetG2C2")
summary(mclustres,parameters=TRUE,classification=TRUE,model="HomG6C1")
pcoordplot(mclustres,model="HomG6C1")



## End(Not run)
```

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

```
## S4 method for signature 'IData'
qda( x, grouping, prior="proportions", CVtol=1.0e-5, subset=1:nrow(x),
  CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE, k2max=1e6, ... )

## S4 method for signature 'IdtMxtNDE'
qda(x, prior="proportions", selmodel=BestModel(x), silent=FALSE,
  k2max=1e6, ... )

## S4 method for signature 'IdtHetNMANOVA'
qda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  silent=FALSE, k2max=1e6, ... )

## S4 method for signature 'IdtGenNSNMANOVA'
qda( x, prior="proportions",
  selmodel=BestModel(H1res(x)@NMod), silent=FALSE, k2max=1e6, ... )
```

Arguments

x	An object of class IData, IdtMxtNDE, IdtHetNMANOVA or IdtGenNSNMANOVA 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 wether a warning message should be printed if the method fails.

selmodel	Selected model from a list of candidate models saved in object x.
k2max	Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.
...	Other named arguments.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

lda, snda, Roblda, Robqda, IData, IdtMxtNDE, IdtHetNMANOVA, IdtGenNSNMANOVA, ConfMat

Examples

```
# 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$GeoReg)
cat("Temperatures of China -- qda discriminant analysis results:\n")
print(ChinaT.qda)
cat("Resubstition confusion matrix:\n")
ConfMat(ChinaTemp$GeoReg,predict(ChinaT.qda,ChinaT)$class)

## 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"])


## End(Not run)
```

---

qHardRoqF *Hardin and Rocke F-quantiles*

Description

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

Usage

```
qHardRoqF(p, nobs, 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

Hardin, J. and Rocke, A. (2005), The Distribution of Robust Distances. *Journal of Computational and Graphical Statistics* **14**, 910–927.

See Also

`fasttle`, `fulltle`

`RepLOptim` *Repeated Local Optimization*

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`	Boolean flag indicating whether a numerically evaluated hessian matrix at the optimum should be computed and returned. Not available for the "nlminb" method.
`parmstder`	Boolean flag indicating whether 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 miminum 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'. Subsquent 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 controling the behaviour of the local optimizer can be specified as components of `control` list. This list can include any of the following components:

maxrepet Maximum time of repetions of the same minimum objective value, before RepLOptim is stoped and the current best solution is returned. By default set to 2.

maxnoimprov 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.

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

allrep 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" mehtod, 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 assymptotic 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.
hessegval	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

```
## 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"), k2max=1e6,
  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"), k2max=1e6, 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 l_1 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".
k2max	Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.
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

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

See Also

lda, qda, snda, IData, RobEstControl, ConfMat

Examples

```
# 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$GeoReg)
cat("Temperatures of China -- robust lda discriminant analysis results:\n")
print(ChinaT.rlda)
cat("Resubstition confusion matrix:\n")
ConfMat(ChinaTemp$GeoReg,predict(ChinaT.rlda,ChinaT)$class)

#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"])

#Robust Quadratic Discriminant Analysis

ChinaT.rqda <- Robqda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- robust qda discriminant analysis results:\n")
print(ChinaT.rqda)
cat("Resubstition confusion matrix:\n")
ConfMat(ChinaTemp$GeoReg,predict(ChinaT.rqda,ChinaT)$class)

#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"])


## 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="Twopplusone", outlin="MidPandLogR",
  trialmethod="simple", m=1, reweighted=TRUE, k2max=1e6, otpType="SetMD2andEst")
```

Arguments

`alpha`	Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*nrow(Sdt) 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.
`rawMD2Dist`	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', respectivelly for the usual Chi-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" respectivelly for the usual Chi-squared, 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 reweightening 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'^(1/n)); and 'iterstep' – as sugested 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'.
`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 which are they are the smaller samples that lead to a non-singular covaraince 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-overlaping 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-overlaping 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.

`k2max` Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.

`otpType` The amount of output returned by fasttle.
Current options are "SetMD2andEst" (default) which returns an 'IdtSngNDRE' object with the fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and "SetMD2EstandPrfSt" wich returns an 'IdtSngNDRE' object with the previous slots plust a list of some performance statistics concerning the algorithm execution.

Value

A `RobEstControl` object

References

Brito, P., Duarte Silva, A. P. (2012): "Modelling Interval Data with Normal and Skew-Normal Distributions". Journal of Applied Statistics, Volume 39, Issue 1, 3-20.

Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. *Journal of the American Statistical Association* **105** (489), 147–156.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Hardin, J. and Rocke, A. (2005), The Distribution of Robust Distances. *Journal of Computational and Graphical Statistics* **14**, 910–927.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

`RobEstControl`, `fasttle`, `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 `"rrcov::CovControlMcd"`. Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*nrow(Sdt) 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 `"rrcov::CovControlMcd"`. Number of subsets used for initial estimates.

`scalefn`**:** Inherited from class `"rrcov::CovControlMcd"` and not used in the package 'Maint.Data.'

`maxcsteps`**:** Inherited from class `"rrcov::CovControlMcd"` and not used in the package 'Maint.Data.'

`seed`**:** Inherited from class `"rrcov::CovControlMcd"`. Starting value for random generator. Default is `seed = NULL`.

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

`trace`**,** `tolSolve`**:** Inherited from class `"rrcov::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.

`rawMD2Dist`**:** 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', respectivelly for the usual Chi-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" respectivelly for the usual Chi-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 reweightening 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'^(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'.

`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.

`k2max`: Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.

`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 algorithm. 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, 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-overlaping 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.

`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" wich returns a list with the previous three components plust a list of some performance statistics concerning the algorithm execution.

Extends

Class CovControlMcd, directly. Class CovControl by rrcov::CovControlMcd, distance 2.

Methods

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

References

Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. *Journal of the American Statistical Association* **105** (489), 147–156.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Hardin, J. and Rocke, A. (2005), The Distribution of Robust Distances. *Journal of Computational and Graphical Statistics* **14**, 910–927.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

RobEstControl, fasttle, RobMxtDEst, Roblda, Robqda

RobMxtDEst-methods *Methods for Function RobMxtDEst in Package 'MAINT.Data'*

Description

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

Usage

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

Arguments

Sdt	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" (hetereocedasic).
CovEstMet	Method used to estimate the common covariance matrix. 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 l_1 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.

`l1medpar` List of named arguments to be passed to the function `pcaPP::l1median` (in package pcaPP) used to find the multivariate l_1 medians. Possible components are 'MaxStep', 'ItTol' and 'trace' (see the documentation of `pcaPP::l1median` for details).
If kept at NULL (default) the defaults of `pcaPP::l1median` will be used.

`...` Other named arguments.

Value

An object of class IdtMxNDRE, containing the estimation results.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P., Filzmoser, P. and Brito, P. (2017), Outlier detection in interval data. *Advances in Data Analysis and Classification*, 1–38.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

`IdtMxNDRE`, `RobEstControl`.

Examples

```
# 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"))

## Not run:

# Estimate robustly an homoscedastic mixture, with mixture components defined by regions

ChinaHomMxtRobE <- RobMxtDEst(ChinaT,ChinaTemp$GeoReg)
print(ChinaHomMxtRobE)

# Estimate robustly an heteroscedastic mixture, with mixture components defined by regions

ChinaHetMxtRobE <- RobMxtDEst(ChinaT,ChinaTemp$GeoReg,Mxt="Het")
print(ChinaHetMxtRobE)



## End(Not run)
```

snda-methods *Skew-Normal Discriminant Analysis of Interval Data*

Description

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

Usage

```
## 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"), k2max=1e6, ... )

## S4 method for signature 'IdtLocSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  egvtol=1.0e-10, silent=FALSE, k2max=1e6, ... )

## S4 method for signature 'IdtLocNSNMANOVA'
snda( x, prior="proportions",
 selmodel=BestModel(H1res(x)@SNMod), egvtol=1.0e-10, silent=FALSE, k2max=1e6, ... )

## S4 method for signature 'IdtGenSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  silent=FALSE, k2max=1e6, ... )

## S4 method for signature 'IdtGenNSNMANOVA'
snda( x, prior="proportions",
  selmodel=BestModel(H1res(x)@SNMod), silent=FALSE, k2max=1e6, ... )
```

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.
k2max	Maximal allowed l2-norm condition number for correlation matrices. Correlation matrices with condition number above k2max are considered to be numerically singular, leading to degenerate results.
...	Other named arguments.

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

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

Examples

```
## 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, asssuming 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)
cat("Resubstition confusion matrix:\n")
ConfMat(ChinaTemp$GeoReg,predict(ChinaT.locsnda,ChinaT)$class)

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

CVlocsnda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=snda,Mxt="Loc",
  CovCase=CovCase(ChinaT.locsnda),kfold=3,CVrep=1)

summary(CVlocsnda[,,"Clerr"])

# Skew-Normal based discriminant analysis, asssuming 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)
cat("Resubstition confusion matrix:\n")
ConfMat(ChinaTemp$GeoReg,predict(ChinaT.gensnda,ChinaT)$class)

#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"])


## 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

```
## 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

---

summary-methods *IdtMclust summary method*

Description

summary methods for the classe IdtMclust defined in Package 'MAINT.Data'.

Usage

```
 ## S4 method for signature 'IdtMclust'
summary(object, parameters = FALSE, classification = FALSE, model = "BestModel",
 ShowClassbyOBs = FALSE, ...)
```

Arguments

object	An object of class IdtMclust representing the results of fitting Gaussian mixtures to interval data objects
parameters	A boolean flag indicating if the parameter estimates of the optimal mixture should be displayed
classification	A boolean flag indicating if the crisp classification resulting from the optimal mixture should be displayed
model	A character vector specifying the the model whose solution is to be displayed.
ShowClassbyOBs	A boolean flag indicating if class membership should shown by observation or by class (default)
.	
...	Other named arguments.

See Also

Idtmclust, IdtMclust, plotInfCrt, pcoordplot

testMod-methods *Methods for Function testMod in Package 'MAINT.Data'*

Description

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

Usage

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

Arguments

ModE	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 likelhiood estimation results for Winter quarters:\n")
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)
```

var-methods *Methods for function var in Package 'MAINT.Data'*

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

vcov-methods *Methods for function vcov in Package 'MAINT.Data'*

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

```
## 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`	The 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. Note that this argument is only used in heterocedastic models, i.e. in the `IdtMxNDE`, `IdtMxSNDE` methods when the object slot "Hmcdt" is set to to FALSE.
`...`	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

∗ **AIC**
BestModel-methods, 8
plotInfCrt-methods, 75
∗ **BIC**
BestModel-methods, 8
plotInfCrt-methods, 75
∗ **Discriminant Analysis Interval Data**
lda-methods, 60
qda-methods, 77
Robda-methods, 82
snda-methods, 91
∗ **EM algorithm**
EMControl-class, 18
∗ **Interval Data**
coef-methods, 10
cor-methods, 14
IdtNandSNDE-class, 47
IdtNDE-class, 47
IdtSNDE-class, 52
lda-methods, 60
MAINT.Data-package, 3
mean-methods, 69
pcoordplot-methods, 72
plot-methods, 74
plotInfCrt-methods, 75
qda-methods, 77
Robda-methods, 82
snda-methods, 91
stdEr-methods, 93
var-methods, 96
vcov-methods, 96
∗ **MAINT.Data**
MAINT.Data-package, 3
∗ **MANOVA for Interval Data**
MAINT.Data-package, 3
∗ **MANOVA**
IdtMANOVA-class, 33
MANOVA-methods, 65
MANOVAPermTest, 67
∗ **Parallel coordinates plot**
pcoordplot-methods, 72
∗ **Parametric modelling of Interval Data**
MAINT.Data-package, 3
∗ **RobMxtDEst**
RobMxtDEst-methods, 89
∗ **Statistical tests for Interval Data**
MAINT.Data-package, 3
∗ **Symbolic Data Analysis**
MAINT.Data-package, 3
∗ **classes**
ConfTests-class, 13
EMControl-class, 18
IData-class, 29
IdtE-class, 31
Idtlda-class, 32
IdtMANOVA-class, 33
IdtMclust-class, 35
IdtMxE-class, 39
IdtMxNandSNDE-class, 40
IdtMxNDE-class, 42
IdtMxNDRE-class, 43
IdtMxSNDE-class, 45
IdtMxtNDE-class, 46
IdtNandSNDE-class, 47
IdtNDE-class, 47
Idtqda-class, 51
IdtSNDE-class, 52
IdtSNgenda-class, 52
IdtSngNandSNDE-class, 54
IdtSngNDE-class, 55
IdtSngNDRE-class, 56
IdtSngSNDE-class, 58
IdtSNlocda-class, 59
LRTest-class, 65
RobEstControl-class, 87
∗ **class**
IdtOutl-class, 48
∗ **coef**

coef-methods, 10
∗ **cor**
cor-methods, 14
∗ **cross-validation**
DACrossVal, 15
∗ **datasets**
Abalone, 7
Cars, 9
ChinaTemp, 9
LoansbyPurpose_minmaxDt, 62
LoansbyRiskLvs_minmaxDt, 63
LoansbyRiskLvs_qntlDt, 64
nycflights, 72
∗ **discriminant analysis**
ConfMat, 11
DACrossVal, 15
∗ **error rates**
ConfMat, 11
∗ **interval data**
Abalone, 7
AgrMcDt, 7
BestModel-methods, 8
Cars, 9
ChinaTemp, 9
ConfTests-class, 13
EMControl-class, 18
fasttle-methods, 19
fulltle-methods, 22
IData, 28
IData-class, 29
IdtE-class, 31
Idtlda-class, 32
IdtMANOVA-class, 33
IdtMclust-class, 35
Idtmclust-methods, 37
IdtMxE-class, 39
IdtMxNandSNDE-class, 40
IdtMxNDE-class, 42
IdtMxNDRE-class, 43
IdtMxSNDE-class, 45
IdtMxtNDE-class, 46
IdtOutl-class, 48
IdtOutl-methods, 50
Idtqda-class, 51
IdtSNgenda-class, 52
IdtSngNandSNDE-class, 54
IdtSngNDE-class, 55
IdtSngNDRE-class, 56
IdtSngSNDE-class, 58
IdtSNlocda-class, 59
LoansbyPurpose_minmaxDt, 62
LoansbyRiskLvs_minmaxDt, 63
LoansbyRiskLvs_qntlDt, 64
LRTest-class, 65
MANOVA-methods, 65
MANOVAPermTest, 67
mle-methods, 70
nycflights, 72
RobEstControl-class, 87
RobMxtDEst-methods, 89
summary-methods, 94
testMod-methods, 95
∗ **likelihood ratio test**
LRTest-class, 65
testMod-methods, 95
∗ **likelihood ratio tets**
MANOVA-methods, 65
MANOVAPermTest, 67
∗ **maximum likelihood estimation**
mle-methods, 70
∗ **mean**
mean-methods, 69
∗ **methods**
BestModel-methods, 8
coef-methods, 10
cor-methods, 14
fasttle-methods, 19
fulltle-methods, 22
IdtOutl-methods, 50
lda-methods, 60
MANOVA-methods, 65
mean-methods, 69
mle-methods, 70
pcoordplot-methods, 72
plot-methods, 74
plotInfCrt-methods, 75
qda-methods, 77
Robda-methods, 82
RobMxtDEst-methods, 89
snda-methods, 91
stdEr-methods, 93
summary-methods, 94
testMod-methods, 95
var-methods, 96
vcov-methods, 96
∗ **model based clustering**

EMControl-class, 18
IdtMclust-class, 35
Idtmclust-methods, 37
pcoordplot-methods, 72
plotInfCrt-methods, 75
∗ **outlier detection**
IdtOutl-class, 48
IdtOutl-methods, 50
∗ **permutation tets**
MANOVAPermTest, 67
∗ **plot**
plot-methods, 74
∗ **robust**
RobEstControl-class, 87
∗ **stdEr**
stdEr-methods, 93
∗ **trimmed maximum likelihood estimator**
fasttle-methods, 19
fulltle-methods, 22
IdtOutl-class, 48
IdtOutl-methods, 50
∗ **var**
var-methods, 96
∗ **vcov**
vcov-methods, 96
.Random.seed, *20*

AbaDF (Abalone), 7
Abalone, 7
AbaloneIdt (Abalone), 7
AbUnits (Abalone), 7
AgrMcDt, 7, *29*, *31*
AIC,IdtE-method (IdtE-class), 31
AIC,IdtMclust-method (IdtMclust-class), 35

BestC (IdtMclust-class), 35
BestC,IdtMclust-method (IdtMclust-class), 35
BestG (IdtMclust-class), 35
BestG,IdtMclust-method (IdtMclust-class), 35
BestModel (BestModel-methods), 8
BestModel,IData-method (BestModel-methods), 8
BestModel,IdtE-method (IdtE-class), 31
BestModel-methods, 8
BIC,IdtE-method (IdtE-class), 31
BIC,IdtMclust-method (IdtMclust-class), 35

Cars, 9
ChinaTemp, 9
classification (IdtMclust-class), 35
classification,IdtMclust-method (IdtMclust-class), 35
coef (coef-methods), 10
coef,IdtNandSNDE-method (coef-methods), 10
coef,IdtNDE-method (coef-methods), 10
coef,IdtSNDE-method (coef-methods), 10
coef-methods, 10
coef-methods (coef-methods), 10
ConfMat, 11, *61*, *78*, *83*, *92*
ConfTests, *65*
ConfTests-class, 13
cor, *70*, *96*
cor (cor-methods), 14
cor,IdtMclust-method (IdtMclust-class), 35
cor,IdtMxNDE-method (cor-methods), 14
cor,IdtMxSNDE-method (cor-methods), 14
cor,IdtNandSNDE-method (cor-methods), 14
cor,IdtNDE-method (cor-methods), 14
cor,IdtSNDE-method (cor-methods), 14
cor-methods, 14
cor-methods (cor-methods), 14
CovCase,Idtlda-method (Idtlda-class), 32
CovCase,Idtqda-method (Idtqda-class), 51
CovCase,IdtSNgenda-method (IdtSNgenda-class), 52
CovCase,IdtSNlocda-method (IdtSNlocda-class), 59
CovControl, *88*
CovControlMcd, *87, 88*

DACrossVal, *12*, 15

EMControl, *17*, 17, *18*, *38*
EMControl-class, 18
extmatrix-class, 18

fasttle, *24*, *27*, *31, 32*, *47–49*, *51*, *55*, *57*, *79*, *84*, *86*, *89*
fasttle (fasttle-methods), 19
fasttle,IData-method (fasttle-methods), 19

fasttle-methods, 19
FlightsDF (nycflights), 72
FlightsIdt (nycflights), 72
FlightsUnits (nycflights), 72
fulltle, *22*, *27*, *31*, *32*, *47–49*, *51*, *55*, *57*, *79*
fulltle (fulltle-methods), 22
fulltle,IData-method (fulltle-methods), 22
fulltle-methods, 22

geteta (IdtOutl-class), 48
geteta,IdtOutl-method (IdtOutl-class), 48
getIdtOutl, *22*, *24*, 25, *48*, *49*, *51*
getMahaD2 (IdtOutl-class), 48
getMahaD2,IdtOutl-method (IdtOutl-class), 48
getmultiCmpCor (IdtOutl-class), 48
getmultiCmpCor,IdtOutl-method (IdtOutl-class), 48
getRefDist (IdtOutl-class), 48
getRefDist,IdtOutl-method (IdtOutl-class), 48

H0res (MANOVA-methods), 65
H0res,IdtMANOVA-method (MANOVA-methods), 65
H1res (MANOVA-methods), 65
H1res,IdtMANOVA-method (MANOVA-methods), 65
head,IData-method (IData-class), 29
Hmcdt (IdtMclust-class), 35
Hmcdt,IdtMclust-method (IdtMclust-class), 35

IData, *7*, *8*, *13*, *16*, 28, *29*, *31–33*, *35*, *40*, *41*, *43*, *45–48*, *51–53*, *55–57*, *59–61*, *65*, *71*, *74*, *77*, *78*, *82*, *83*, *91*, *92*
IData-class, 29
IdtClMANOVA, *34*, *60*, *61*
IdtClMANOVA-class (IdtMANOVA-class), 33
IdtE, *9*, *21*, *24*, *39–46*, *54–58*, *95*
IdtE-class, 31
IdtGenNSNMANOVA, *34*, *77*, *78*, *91*
IdtGenNSNMANOVA-class (IdtMANOVA-class), 33
IdtGenSNMANOVA, *34*, *91*, *92*
IdtGenSNMANOVA-class (IdtMANOVA-class), 33
IdtHetNMANOVA, *34*, *77*, *78*
IdtHetNMANOVA-class (IdtMANOVA-class), 33
Idtlda-class, 32
IdtLocNSNMANOVA, *34*, *60*, *61*, *91*, *92*
IdtLocNSNMANOVA-class (IdtMANOVA-class), 33
IdtLocSNMANOVA, *34*, *91*, *92*
IdtLocSNMANOVA-class (IdtMANOVA-class), 33
IdtMANOVA, *67*, *69*
IdtMANOVA-class, 33
IdtMclust, *38*, *73*, *76*, *94*
Idtmclust, *17*, *18*, *37*, *73*, *76*, *94*
Idtmclust (Idtmclust-methods), 37
Idtmclust,IData-method (Idtmclust-methods), 37
IdtMclust-class, 35
Idtmclust-methods, 37
IdtMxE, *40–46*
IdtMxE-class, 39
IdtMxNandSNDE, *47*, *55*
IdtMxNandSNDE-class, 40
IdtMxNDE, *14*, *42*, *45–48*, *56*, *70*, *96*, *97*
IdtMxNDE-class, 42
IdtMxNDRE, *43*, *46–48*, *57*, *90*
IdtMxNDRE-class, 43
IdtMxSNDE, *14*, *45*, *52*, *59*, *70*, *96*, *97*
IdtMxSNDE-class, 45
IdtMxtNDE, *60*, *61*, *77*, *78*
IdtMxtNDE-class, 46
IdtNandSNDE, *10*, *14*, *70*, *96*, *97*
IdtNandSNDE-class, 47
IdtNDE, *14*, *52*, *70*, *96*, *97*
IdtNDE-class, 47
IdtOutl-class, 48
IdtOutl-methods, 50
Idtqda-class, 51
IdtSNDE, *10*, *14*, *48*, *70*, *96*, *97*
IdtSNDE-class, 52
IdtSngDE, *40*, *54*, *55*, *57*, *58*
IdtSngDE-class (IdtE-class), 31
IdtSNgenda-class, 52
IdtSngNandSNDE, *41*, *47*
IdtSngNandSNDE-class, 54
IdtSngNDE, *26*, *43*, *47*, *48*, *57*, *59*
IdtSngNDE-class, 55
IdtSngNDRE, *22*, *26*, *45*, *47*, *48*, *56*

IdtSngNDRE-class, 56
IdtSngSNDE, *46*, *52*, *56*
IdtSngSNDE-class, 58
IdtSNlocda-class, 59
is.IData (IData), 28

lda, *12*, *16*, *31*, *35*, *78*, *83*, *92*
lda (lda-methods), 60
lda,IData-method (lda-methods), 60
lda,IdtClMANOVA-method (lda-methods), 60
lda,IdtLocNSNMANOVA-method (lda-methods), 60
lda,IdtMxtNDE-method (lda-methods), 60
lda-methods, 60
LoansbyPurpose_minmaxDt, 62
LoansbyRiskLvs_minmaxDt, 63
LoansbyRiskLvs_qntlDt, 64
logLik,IdtE-method (IdtE-class), 31
logLik,IdtMclust-method (IdtMclust-class), 35
LogRanges,IData-method (IData-class), 29
LRTest, *13*, *33–35*
LRTest-class, 65

MAINT.Data (MAINT.Data-package), 3
MAINT.Data-package, 3
MANOVA, *31–33*, *35*, *40*, *41*, *43*, *46–48*, *51–53*, *60*, *65*, *69*
MANOVA (MANOVA-methods), 65
MANOVA,IData-method (MANOVA-methods), 65
MANOVA-methods, 65
MANOVAPermTest, 67
matrix, *18*
mean (mean-methods), 69
mean,IdtMclust-method (IdtMclust-class), 35
mean,IdtMxNDE-method (mean-methods), 69
mean,IdtMxSNDE-method (mean-methods), 69
mean,IdtNandSNDE-method (mean-methods), 69
mean,IdtNDE-method (mean-methods), 69
mean,IdtSNDE-method (mean-methods), 69
mean-methods, 69
mean-methods (mean-methods), 69
MidPoints,IData-method (IData-class), 29
mle, *13*, *31*, *32*, *47*, *48*, *52*, *55*, *56*, *59*, *65*
mle (mle-methods), 70
mle,IData-method (mle-methods), 70
mle-methods, 70
NbMicroUnits,IData-method (IData-class), 29
nycflights, 72

parameters (IdtMclust-class), 35
parameters,IdtMclust-method (IdtMclust-class), 35
pcoordplot, *37*, *38*, *76*, *94*
pcoordplot (pcoordplot-methods), 72
pcoordplot,IdtMclust-method (pcoordplot-methods), 72
pcoordplot-methods, 72
plot (plot-methods), 74
plot,IData,IData-method (plot-methods), 74
plot,IData,missing-method (plot-methods), 74
plot,IData-method (plot-methods), 74
plot,IdtOutl,missing-method (IdtOutl-methods), 50
plot-methods, 74
plotInfCrt, *37*, *38*, *73*, *94*
plotInfCrt (plotInfCrt-methods), 75
plotInfCrt,IdtMclust-method (plotInfCrt-methods), 75
plotInfCrt-methods, 75
PostProb (IdtMclust-class), 35
PostProb,IdtMclust-method (IdtMclust-class), 35
predict,Idtlda-method (Idtlda-class), 32
predict,Idtqda-method (Idtqda-class), 51
predict,IdtSNgenda-method (IdtSNgenda-class), 52
predict,IdtSNlocda-method (IdtSNlocda-class), 59
print,IData-method (IData-class), 29
print,IdtOutl-method (IdtOutl-class), 48
pro (IdtMclust-class), 35
pro,IdtMclust-method (IdtMclust-class), 35

qda, *12*, *16*, *31*, *33*, *35*, *51*, *61*, *83*, *92*
qda (qda-methods), 77
qda,IData-method (qda-methods), 77
qda,IdtGenNSNMANOVA-method (qda-methods), 77
qda,IdtHetNMANOVA-method (qda-methods), 77
qda,IdtMxtNDE-method (qda-methods), 77

qda-methods, 77
qHardRoqF, 78

Ranges,IData-method (IData-class), 29
RepLOptim, *67*, *71*, 79
Robda-methods, 82
RobEstControl, *21*, *22*, *83*, *84*, 84, *86*, *89*, *90*
RobEstControl-class, 87
Roblda, *12*, *31*, *33*, *61*, *78*, *84*, *86*, *89*, *92*
Roblda (Robda-methods), 82
Roblda,IData-method (Robda-methods), 82
Roblda-methods (Robda-methods), 82
RobMxtDEst, *31*, *32*, *40*, *41*, *45*, *47*, *48*, *84*, *86*, *89*
RobMxtDEst (RobMxtDEst-methods), 89
RobMxtDEst,IData-method (RobMxtDEst-methods), 89
RobMxtDEst-methods, 89
Robqda, *12*, *31*, *33*, *51*, *61*, *78*, *84*, *86*, *89*, *92*
Robqda (Robda-methods), 82
Robqda,IData-method (Robda-methods), 82
Robqda-methods (Robda-methods), 82

sd, *70*
sd,IdtE-method (IdtE-class), 31
SelCrit (IdtMclust-class), 35
SelCrit,IdtMclust-method (IdtMclust-class), 35
show,ConfTests-method (ConfTests-class), 13
show,IdtE-method (IdtE-class), 31
show,Idtlda-method (Idtlda-class), 32
show,IdtMANOVA-method (MANOVA-methods), 65
show,IdtMclust-method (IdtMclust-class), 35
show,Idtqda-method (Idtqda-class), 51
show,IdtSNgenda-method (IdtSNgenda-class), 52
show,IdtSNlocda-method (IdtSNlocda-class), 59
show,LRTest-method (LRTest-class), 65
snda, *12*, *33*, *35*, *53*, *60*, *61*, *78*, *83*
snda (snda-methods), 91
snda,IData-method (snda-methods), 91
snda,IdtGenNSNMANOVA-method (snda-methods), 91
snda,IdtGenSNMANOVA-method (snda-methods), 91
snda,IdtLocNSNMANOVA-method (snda-methods), 91
snda,IdtLocSNMANOVA-method (snda-methods), 91
snda-methods, 91
stdEr, *11*, *97*
stdEr (stdEr-methods), 93
stdEr,IdtNandSNDE-method (stdEr-methods), 93
stdEr,IdtNDE-method (stdEr-methods), 93
stdEr,IdtSNDE-method (stdEr-methods), 93
stdEr-methods, 93
stdEr-methods (stdEr-methods), 93
summary,IData-method (IData-class), 29
summary,IdtE-method (IdtE-class), 31
summary,IdtMclust-method (summary-methods), 94
summary-methods, 94

tail,IData-method (IData-class), 29
testMod (testMod-methods), 95
testMod,IdtE-method (IdtE-class), 31
testMod-methods, 95

var, *14*, *70*
var (var-methods), 96
var,IdtMclust-method (IdtMclust-class), 35
var,IdtMxNDE-method (var-methods), 96
var,IdtMxSNDE-method (var-methods), 96
var,IdtNandSNDE-method (var-methods), 96
var,IdtNDE-method (var-methods), 96
var,IdtSNDE-method (var-methods), 96
var-methods, 96
var-methods (var-methods), 96
vcov, *11*, *94*
vcov (vcov-methods), 96
vcov,IdtMxNDE-method (vcov-methods), 96
vcov,IdtMxSNDE-method (vcov-methods), 96
vcov,IdtNandSNDE-method (vcov-methods), 96
vcov,IdtNDE-method (vcov-methods), 96
vcov,IdtSNDE-method (vcov-methods), 96
vcov-methods, 96
vcov-methods (vcov-methods), 96