Package 'CoImp'

December 20, 2024

Title Parametric and Non-Parametric Copula-Based Imputation Methods

Date 2024-12-20

Version 2.0.2

Author Francesca Marta Lilja Di Lascio [aut, cre],
Aurora Gatto [aut],
Simone Giannerini [aut]

Depends R (>= 4.2.0), methods, stats, copula, cluster

Imports nnet, gtools, locfit

Description Copula-based imputation methods: parametric and non-parametric algorithms for missing multivariate data through conditional copulas.

Maintainer Francesca Marta Lilja Di Lascio <marta.dilascio@unibz.it>

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2024-12-20 14:10:02 UTC

Contents

CoImp *Copula-Based Imputation Method*

Description

Imputation method based on conditional copula functions.

Usage

```
CoImp(X, n.marg = ncol(X), x.up = NULL, x.lo = NULL, q.up = rep(0.85, n.marg),
            q.lo = rep(0.15, n.marg), type.data = "continuous", smoothing =
            rep(0.5, n.marg), plot = TRUE, model = list(normalCopula(0.5,
            dim=n.marg),  claytonCopula(10, dim=n.marg), gumbelCopula(10,
            dim=n.marg), frankCopula(10, dim=n.marg), tCopula(0.5,
            dim=n.marg,...), rotCopula(claytonCopula(10,dim=n.marg),
            flip=rep(TRUE,n.marg)),...), start. = NULL, ...)
```

Arguments

X	a data matrix with missing values. Missing values should be denoted with NA.
n.marg	the number of variables in X.
x.up	a numeric vector of length n.marg with the upper value of each margin used in the Hit or Miss method. Specify either x.up xor q.up.
x.lo	a numeric vector of length n.marg with the lower value of each margin used in the Hit or Miss method. Specify either x.lo xor q.lo.
q.up	a numeric vector of length n.marg with the probability of the quantile used to define x.up for each margin. Specify either x.up xor q.up.
q.lo	a numeric vector of length n.marg with the probability of the quantile used to define x.lo for each margin. Specify either x.lo xor q.lo.
type.data	the nature of the variables in X: discrete or continuous.
smoothing	values for the nearest neighbour component of the smoothing parameter of the lp function.
plot	logical: if TRUE plots the estimated marginal densities and a bar plot of the percentages of missing and available data for each margin.
model	a list of copula models to be used for the imputation, see the Details section. This should be one of normal and t (with dispstr as in the copula package), frank, clayton, gumbel, and rotated copulas. As in fitCopula, itau fitting coerced tCopula to 'df.fixed=TRUE'.
start.	a numeric vector of starting values for the parameter optimization via optim.
...	further parameters for fitCopula, lp and further graphical arguments.

Details

CoImp is an imputation method based on conditional copula functions that allows to impute missing observations according to the multivariate dependence structure of the generating process without any assumptions on the margins. This method can be used independently from the dimension and the kind (monotone or non monotone) of the missing patterns.

Brief description of the approach:

1. estimate both the margins and the copula model on available data by means of the semi-parametric sequential two-step inference for margins;
2. derive conditional density functions of the missing variables given non-missing ones through the corresponding conditional copulas obtained by using the Bayes' rule;
3. impute missing values by drawing observations from the conditional density functions derived at the previous step. The Monte Carlo method used is the Hit or Miss.

The estimation approach for the copula fit is semiparametric: a range of nonparametric margins and parametric copula models can be selected by the user.

Value

An object of S4 class "CoImp", which is a list with the following elements:

`Missing.data.matrix`
: the original missing data matrix to be imputed.

`Perc.miss`
: the matrix of the percentage of missing and available data.

`Estimated.Model`
: the estimated copula model on the available data.

`Estimation.Method`
: the estimation method used for the copula `Estimated.Model`.

`Index.matrix.NA`
: matrix indices of the missing data.

`Smooth.param`
: the smoothing parameter alpha selected on the basis of the AIC.

`Imputed.data.matrix`
: the imputed data matrix.

`Estimated.Model.Imp`
: the estimated copula model on the imputed data matrix.

`Estimation.Method.Imp`
: the estimation method used for the copula `Estimated.Model.Imp`.

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>, Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

NPCoImp, MCAR, MAR, fitCopula, lp.

Examples

```
## generate data from a 4-variate Frank copula with different margins

set.seed(21)
n.marg <- 4
theta  <- 5
copula <- frankCopula(theta, dim = n.marg)
mymvdc <- mvdc(copula, c("norm", "gamma", "beta","gamma"), list(list(mean=7, sd=2),
list(shape=3, rate=2), list(shape1=4, shape2=1), list(shape=4, rate=3)))
n      <- 20
x.samp <- copula::rMvdc(n, mymvdc)

# randomly introduce univariate and multivariate missing

perc.mis    <- 0.3
set.seed(11)
miss.row    <- sample(1:n, perc.mis*n, replace=TRUE)
miss.col    <- sample(1:n.marg, perc.mis*n, replace=TRUE)
miss        <- cbind(miss.row,miss.col)
x.samp.miss <- replace(x.samp,miss,NA)

# impute missing values

imp <- CoImp(x.samp.miss, n.marg=n.marg, smoothing = rep(0.6,n.marg), plot=TRUE,
       type.data="continuous", model=list(normalCopula(0.5, dim=n.marg),
       frankCopula(10, dim=n.marg), gumbelCopula(10, dim=n.marg)));

# methods show and plot

show(imp)
plot(imp)
```

```
## Not run:
## generate data from a 3-variate Clayton copula and introduce missing by
## using the MCAR function and try to impute through a rotated copula

set.seed(11)
n.marg <- 3
theta  <- 5
copula <- claytonCopula(theta, dim = n.marg)
mymvdc <- mvdc(copula, c("beta", "beta", "beta"), list(list(shape1=4, shape2=1),
            list(shape1=.5, shape2=.5), list(shape1=2, shape2=3)))
n      <- 50
x.samp <- copula::rMvdc(n, mymvdc)

# randomly introduce MCAR univariate and multivariate missing

perc.miss <- 0.15
setseed   <- set.seed(13)
x.samp.miss <- MCAR(x.samp, perc.miss, setseed)
x.samp.miss <- x.samp.miss@"db.missing"

# impute missing values

imp <- CoImp(x.samp.miss, n.marg=n.marg, smoothing = c(0.45,0.2,0.5), plot=TRUE,
        q.lo=rep(0.1,n.marg), q.up=rep(0.9,n.marg), model=list(claytonCopula(0.5,
        dim=n.marg),  rotCopula(claytonCopula(0.5,dim=n.marg))));

# methods show and plot

show(imp)
plot(imp)

## End(Not run)
```

`CoImp-class` *Class "CoImp"*

Description

A class for `CoImp` and its extensions

Objects from the Class

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

Slots

`Missing.data.matrix`**:** Object of class `"matrix"`. Original missing data matrix to be imputed.

`Perc.miss`**:** Object of class `"matrix"`. Missing and available data percentage for each variable.

`Estimated.Model`: Object of class `"list"`. The list contains:

`model`	the copula model selected and estimated on the complete cases.
`dimension`	the dimension of the `model`.
`parameter`	the estimated dependence parameter of the `model`.
`number`	the index of the estimated model in the list of models given in input.

`Estimation.Method`: Object of class `"character"`. The estimation method used for the copula model in `Estimated.Model`. Allowed methods are in `fitCopula`.

`Index.matrix.NA`: Object of class `"matrix"`. Matrix of row and column indexes of missing data.

`Smooth.param`: Object of class `"numeric"`. The values of the nearest neighbor component of the smoothing parameter of the `lp` function.

Imputed.data.matrix Object of class `"matrix"`. The imputed data matrix.

Estimated.Model.Imp Object of class `"list"`. The list contains:

`model`	the copula model selected and estimated on the imputed cases.
`dimension`	the dimension of the `model`.
`parameter`	the estimated dependence parameter of the `model`.
`number`	the index of the estimated model in the list of models given in input.

Estimation.Method.Imp Object of class `"character"`.The estimation method used for the copula model in `Estimated.Model.Imp`. Allowed methods are in `fitCopula`.

Methods

plot `signature(x = "CoImp", y = "missing")`: ...

show `signature(object = "CoImp")`: ...

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>, Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

NPCoImp, MCAR, MAR, fitCopula.

Examples

```
showClass("CoImp")
```

MAR *Generation of multivariate missing at random (MAR) data*

Description

Introduction of artificial missing at random (MAR) data in a given data set. Missing values are multivariate and have generic pattern.

Usage

```
MAR(db.complete, perc.miss = 0.3, setseed = 13, mcols = NULL, ...)
```

Arguments

db.complete	the complete data matrix.
perc.miss	the percentage of missing values to be generated.
setseed	the seed for the generation of the missing values.
mcols	the index of the columns in which to introduce MAR values.
...	further parameters for fitCopula.

Details

MAR introduce artificial missing at random values in a given complete data set. Missing values are univariate and multivariate and have generic pattern.

Value

An object of S4 class "MAR", which is a list with the following element:

perc.record.missing	Object of class "numeric". A percentage value.
db.missing	Object of class "matrix". A data set with artificial multivariate MAR.

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>,

Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

CoImp, NPCoImp, fitCopula.

Examples

```
# generate data from a 4-variate Gumbel copula with different margins

set.seed(11)
n.marg <- 4
theta  <- 5
copula <- frankCopula(theta, dim = n.marg)
mymvdc <- mvdc(copula, c("norm", "gamma", "beta","gamma"), list(list(mean=7, sd=2),
list(shape=3, rate=2), list(shape1=4, shape2=1), list(shape=4, rate=3)))
n      <- 30
x.samp <- rMvdc(n, mymvdc)

# apply MAR by introducing 30% of missing data

mar    <- MAR(db.complete = x.samp, perc.miss = 0.3, setseed = 11)

mar
```

MAR-class *Class "MAR"*

Description

A class for MAR and its extensions

Objects from the Class

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

Slots

perc.record.missing: Object of class "numeric". A percentage value.

db.missing: Object of class "matrix". A data set with artificial multivariate MAR with generic pattern.

Methods

show signature(object = "MAR"): ...

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>, Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

CoImp, NPCoImp.

Examples

```
showClass("MAR")
```

MCAR *Generation of multivariate MCAR data*

Description

Introduction of artificial missing completely at random (MCAR) data in a given data set. Missing values are multivariate and have generic pattern.

Usage

```
MCAR(db.complete, perc.miss = 0.3, setseed = 13, mcols = NULL, ...)
```

Arguments

db.complete	the complete data matrix.
perc.miss	the percentage of missing value to be generated.
setseed	the seed for the generation of the missing values.
mcols	the index of the columns in which to introduce MCAR values.
...	further parameters for fitCopula.

Details

MCAR introduce artificial missing completely at random values in a given complete data set. Missing values are multivariate and have generic pattern.

Value

An object of S4 class "MCAR", which is a list with the following element:

db.missing	Object of class "matrix". A data set with artificial multivariate MCAR.

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>,

Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

CoImp, NPCoImp, fitCopula.

Examples

```
# generate data from a 4-variate Gumbel copula with different margins

set.seed(11)
n.marg <- 4
theta  <- 5
```

```
copula <- frankCopula(theta, dim = n.marg)
mymvdc <- mvdc(copula, c("norm", "gamma", "beta","gamma"), list(list(mean=7, sd=2),
list(shape=3, rate=2), list(shape1=4, shape2=1), list(shape=4, rate=3)))
n      <- 30
x.samp <- rMvdc(n, mymvdc)

# apply MCAR by introducing 30% of missing data

mcar   <- MCAR(db.complete = x.samp, perc.miss = 0.3, setseed = 11)

mcar

# same example as above but introducing missing only in the first and third column

mcar2   <- MCAR(db.complete = x.samp, perc.miss = 0.3, setseed = 11, mcols=c(1,3))

mcar2
```

MCAR-class *Class "MCAR"*

Description

A class for MCAR and its extensions

Objects from the Class

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

Slots

db.missing: Object of class "matrix". A data set with artificial multivariate MCAR.

Methods

show signature(object = "MCAR"): ...

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>,

Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

CoImp, NPCoImp.

Examples

```
showClass("MCAR")
```

NPCoImp *Non-Parametric Copula-Based Imputation Method*

Description

Imputation method based on empirical conditional copula functions.

Usage

```
NPCoImp(X, Psi=seq(0.05,0.45,by=0.05), smoothing="beta", K=7, method="gower")
```

Arguments

X	a data matrix with missing values. Missing values should be denoted with NA.
Psi	vector of probabilities to assess the symmetry/asymmetry of the empirical conditional copula (ecc) function and find the best quantile for the imputation (see below for details).
smoothing	the character string specifying the type of smoothing of the empirical copula. Default is "beta" (empirical beta copula) but also "none" (the original empirical copula) can be used.
K	the number of data matrix rows more similar to the missing one that are used for the imputation.
method	the distance measure used for the imputation, among Euclidean, Manhattan, Canberra, Gower, and two based on the Kendall-correlation coefficient (see below for details).

Details

NPCoImp is a non-parametric imputation method based on the empirical conditional copula function. To choose the best quantile for the imputation it assesses the (a)symmetry of the empirical conditional copula and it uses the K pseudo-observations more similar to the missing one. The NPCoImp allows the imputation of missing observations according to the multivariate dependence structure of the data generating process without any assumptions on the margins. This method can be used independently from the dimension and the kind (monotone or non monotone) of the missing patterns. Brief description of the approach:

1. estimate the empirical (beta) conditional copula of the missing observation(s) given the available ones;
2. evaluate the (a)symmetry of the empirical conditional copula around 0.5 (see the paper in the references for details);
3. select the quantile of the empirical conditional copula on the basis of its (a)symmetry. Therefore:
   - symmetry: we impute through the median of the empirical conditional copula;
   - negative asymmetry: we impute with a quantile on the left tail of the ecc (see the paper in the references for details);
   - positive asymmetry: we impute with a quantile on the right tail of the ecc (see the paper in the references for details);
4. select the K pseudo-observations closest to the imputed one and the corresponding original observations;
5. impute missing values by replacing them from the average of the original observations derived at the previous step.

Value

An object of S4 class "NPCoImp", which is a list with the following elements:

`Imputed.matrix` the imputed data matrix.

`Selected.quantile.alpha`
the quantile selected for the imputation and its order alpha.

`numFlat` the number of possible flat empirical conditional copulas, i.e. when ecc is always zero.

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>, Aurora Gatto <aurora.gatto@unibz.it>

References

Di Lascio, F.M.L, Gatto A. (202x) "A non-parametric conditional copula-based imputation method". Under review.

See Also

`CoImp`, `MCAR`, `MAR`.

Examples

```
## generate data from a 4-variate Frank copula with different margins

set.seed(21)
n.marg <- 4
theta  <- 5
copula <- frankCopula(theta, dim = n.marg)
mymvdc <- mvdc(copula, c("norm", "gamma", "beta","gamma"), list(list(mean=7, sd=2),
list(shape=3, rate=2), list(shape1=4, shape2=1), list(shape=4, rate=3)))
n      <- 20
x.samp <- copula::rMvdc(n, mymvdc)

# randomly introduce univariate and multivariate missing

perc.mis    <- 0.25
set.seed(14)
miss.row    <- sample(1:n, perc.mis*n, replace=TRUE)
miss.col    <- sample(1:n.marg, perc.mis*n, replace=TRUE)
miss        <- cbind(miss.row,miss.col)
x.samp.miss <- replace(x.samp,miss,NA)
x.samp.miss
probs <- seq(0.05,0.45,by=0.1)
ndist <- 7
dist.meth <- "gower"

# impute missing values
NPimp <- NPCoImp(X=x.samp.miss, Psi=probs, smoothing="beta", K=ndist,
                   method=dist.meth)

# methods show

show(NPimp)

## Not run:
## generate data from a 3-variate Clayton copula and introduce missing by
## using the MCAR function and try to impute through a rotated copula

set.seed(11)
n.marg <- 3
theta  <- 5
copula <- claytonCopula(theta, dim = n.marg)
mymvdc <- mvdc(copula, c("beta", "beta", "beta"), list(list(shape1=4, shape2=1),
                list(shape1=.5, shape2=.5), list(shape1=2, shape2=3)))
n      <- 50
x.samp <- copula::rMvdc(n, mymvdc)

# randomly introduce MCAR univariate and multivariate missing

perc.miss <- 0.15
setseed   <- set.seed(13)
x.samp.miss <- MCAR(x.samp, perc.miss, setseed)
x.samp.miss <- x.samp.miss@"db.missing"
```

```
probs <- seq(0.05,0.45,by=0.05)
ndist <- 7
dist.meth <- "gower"

# impute missing values

NPimp2 <- NPCoImp(X=x.samp.miss, Psi=probs, smoothing="beta", K=ndist,
                    method=dist.meth)

# methods show and plot

show(NPimp2)

## End(Not run)
```

`NPCoImp-class` *Class "NPCoImp"*

Description

A class for `NPCoImp` and its extensions

Objects from the Class

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

Slots

`Imputed.matrix` Object of class `"matrix"`. The imputed data matrix.

`Selected.quantile.alpha` Object of class `"vector"`. The quantile selected for the imputation and its order alpha.

`numFlat` Object of class `"numeric"`. The number of possible flat empirical conditional copulas, i.e. when the function cannot be empirically estimated.

Methods

show `signature(object = "NPCoImp")`: ...

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>, Aurora Gatto <aurora.gatto@unibz.it>

References

Di Lascio, F.M.L, Gatto A. (202x) "A non-parametric conditional copula-based imputation method". Under review.

See Also

CoImp, MCAR, MAR.

Examples

```
showClass("NPCoImp")
```

PerfMeasure *Performance measures for evaluating the goodness of an imputed database*

Description

Set of measures useful to evaluate the goodness of the used imputation method.

Usage

```
PerfMeasure(db.complete, db.imputed, db.missing, n.marg = 2, model =
                        list(normalCopula(0.5, dim=n.marg), claytonCopula(10,
                    dim=n.marg), gumbelCopula(10, dim=n.marg), frankCopula(10,
                        dim=n.marg), tCopula(0.5, dim=n.marg,...),
                  rotCopula(claytonCopula(10,dim=n.marg),flip=rep(TRUE,n.marg)),
                        ...), ...)
```

Arguments

db.complete	the complete data matrix.
db.imputed	the imputed data matrix.
db.missing	the data matrix with NA data.
n.marg	the number of variables in db.complete.
model	a list of copula models to be used for the imputation. See the Details section. This should be one of normal and t (with dispstr as in the copula package), frank, clayton, gumbel, and rotated copulas. As in fitCopula, itau fitting coerced tCopula to 'df.fixed=TRUE'.
...	further parameters for fitCopula.

Details

PerfMeasure computes some measures useful for evaluating the goodness of the used imputation method. PerfMeasure requires in input the imputed, the complete and the missing data matrix and gives in output five different measures of performance. See below for details

Value

An object of S4 class "PerfMeasure", which is a list with the following elements:

MARE	Object of class "numeric". The mean (on the replications performed) of the absolute relative error between the imputed and the corresponding original value.
RB	Object of class "numeric". The relative bias of the estimator for the dependence parameter.
RRMSE	Object of class "numeric". The relative root mean squared error of the estimator for the dependence parameter.
TID	Object of class "vector". Upper and lower tail dependence indexes for bivariate copulas. Original function is in tailIndex.

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>,

Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

CoImp, NPCoImp, MCAR, MAR.

Examples

```
## Not run:
# generate data from a 4-variate Gumbel copula with different margins

set.seed(11)
n.marg <- 4
theta  <- 5
copula <- frankCopula(theta, dim = n.marg)
mymvdc <- mvdc(copula, c("norm", "gamma", "beta","gamma"), list(list(mean=7, sd=2),
 list(shape=3, rate=2), list(shape1=4, shape2=1), list(shape=4, rate=3)))
n      <- 20
x.samp <- rMvdc(n, mymvdc)
```

```
# randomly introduce univariate and multivariate missing

perc.mis    <- 0.3
set.seed(11)
miss.row    <- sample(1:n, perc.mis*n, replace=TRUE)
miss.col    <- sample(1:n.marg, perc.mis*n, replace=TRUE)
miss        <- cbind(miss.row,miss.col)
x.samp.miss <- replace(x.samp,miss,NA)

# impute missing values

imp <- CoImp(x.samp.miss, n.marg=n.marg, smoothing=rep(0.6,n.marg), plot=TRUE,
            type.data="continuous");
imp

# apply PerfMeasure to the imputed data set

pm <- PerfMeasure(db.complete=x.samp, db.missing=x.samp.miss,
            db.imputed=imp@"Imputed.data.matrix", n.marg=4)
pm
str(pm)

## End(Not run)
```

`PerfMeasure-class` *Class "PerfMeasure"*

Description

A class for `PerfMeasure` and its extensions

Objects from the Class

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

Slots

`MARE`: Object of class `"numeric"`. The mean (on the replications performed) of the absolute relative error between the imputed and the corresponding original value.

`RB`: Object of class `"numeric"`. The relative bias of the estimator for the dependence parameter.

`RRMSE`: Object of class `"numeric"`. The relative root mean squared error of the estimator for the dependence parameter.

`TID`: Object of class `"vector"`. Upper and lower tail dependence indexes for bivariate copulas. Original function is in `tailIndex`.

Methods

show `signature(object = "PerfMeasure")`: ...

Author(s)

F. Marta L. Di Lascio <marta.dilascio@unibz.it>, Simone Giannerini <simone.giannerini@unibo.it>

References

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2015) "Exploring Copulas for the Imputation of Complex Dependent Data". Statistical Methods & Applications, 24(1), p. 159-175. DOI 10.1007/s10260-014-0287-2.

Di Lascio, F.M.L., Giannerini, S. and Reale, A. (2014) "Imputation of complex dependent data by conditional copulas: analytic versus semiparametric approach", Book of proceedings of the 21st International Conference on Computational Statistics (COMPSTAT 2014), p. 491-497. ISBN 9782839913478.

Bianchi, G. Di Lascio, F.M.L. Giannerini, S. Manzari, A. Reale, A. and Ruocco, G. (2009) "Exploring copulas for the imputation of missing nonlinearly dependent data". Proceedings of the VII Meeting Classification and Data Analysis Group of the Italian Statistical Society (Cladag), Editors: Salvatore Ingrassia and Roberto Rocci, Cleup, p. 429-432. ISBN: 978-88-6129-406-6.

See Also

`CoImp`, `NPCoImp`, `MCAR`, `MAR`.

Examples

```
showClass("PerfMeasure")
```

Index