Package ‘mtsdi’

October 13, 2022

Version 0.3.5
Date 2018-01-02
Author Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>
Maintainer Washington Junger <wjunger@ims.uerj.br>
Depends R (>= 3.0.0),utils,stats,gam,splines
Title Multivariate Time Series Data Imputation
Description This is an EM algorithm based method for imputation of missing values in multivariate normal time series. The imputation algorithm accounts for both spatial and temporal correlation structures. Temporal patterns can be modeled using an ARIMA(p,d,q), optionally with seasonal components, a non-parametric cubic spline or generalized additive models with exogenous covariates. This algorithm is specially tailored for climate data with missing measurements from several monitors along a given region.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2018-01-23 21:52:30 UTC

R topics documented:
edaprep .......................................................... 2
elapsedtime .................................................. 3
getmean ....................................................... 4
miss .............................................................. 5
mkjnw ........................................................... 5
mnimput ....................................................... 6
mstats .......................................................... 9
plot.mtsdi ................................................... 10
predict.mtsdi ............................................... 11
print.mtsdi ................................................ 12
print.summary.mtsdi ..................................... 13
summary.mtsdi ............................................ 14

Index 16
edaprep  

Dataset Preparation for Analysis

Description

Prepare the dataset for exploratory data analysis

Usage

edaprep(dataset)

Arguments

dataset  dataset with missing observations

Details

It replaces missing observation with the vector mean.

Value

It returns dataset filled in with NA

Author(s)

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

See Also

mnimput, getmean, edaprep

Examples

data(miss)
c <- edaprep(miss)
**elapsedtime**

**Elapsed Time**

---

**Description**

Compute the elapsed time between start time and end time

**Usage**

```
elapsedtime(st, et)
```

**Arguments**

- **st**: starting time
- **et**: ending time

**Details**

It returns the time the process took to run.

**Value**

String of the form `hh:mm:ss`

**Note**

It is not intended to be called directly.

**Author(s)**

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

**See Also**

`mnimput`
getmean

Row Means Estimates

Description
Estimate the row mean from a mtsdi object regarding a fixed number of imputed values

Usage
getmean(object, weighted=TRUE, mincol=1, maxconsec=3)

Arguments
- object: imputation object
- weighted: If TRUE, weights returned by mnimput will be used for mean computation
- mincol: integer for the minimum number of valid values by row
- maxconsec: integer for the maximum number of consecutive missing values in a column

Details
It is useful just in case one wants row mean estimated. If log transformation was used, mean is adjusted accordingly.

Value
A vector of row means with length \( n \), where \( n \) is the number of observations.

Author(s)
Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

See Also
- mnimput
- getmean
- edaprep

Examples
```r
data(miss)
f <- ~c31+c32+c33+c34+c35
i <- mnimput(f,miss,eps=1e-3,ts=TRUE, method="spline", sp.control=list(df=c(7,7,7,7,7)))
m <- getmean(i,2)
```
**miss**

_Sample Dataset_

**Description**

A small sample dataset for the tutorial on data imputation

**Usage**

```r
data(miss)
```

**Format**

A data frame with 24 observations on the following 5 variables.

- `c31` a numeric vector with 1 missing observation
- `c32` a numeric vector with 1 missing observation
- `c33` a numeric vector with 6 missing observations
- `c34` a numeric vector with 3 missing observations
- `c35` a numeric vector with 3 missing observations

**Examples**

```r
data(miss)
```

**mkjnw**

_Example from Johnson & Wichern’s Book_

**Description**

Create a data matrix from the Johnson & Wichern’s book

**Usage**

```r
mkjnw()
```

**Details**

This function creates a data matrix from the Johnson & Wichern’s book.

**Value**

It returns a data matrix.
Author(s)

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

References


See Also

mnimput

Examples

d <- mkjnw()

Description

Perform the modified EM algorithm imputation on a normal multivariate dataset

Usage

`mnimput(formula, dataset, by = NULL, log = FALSE, log.offset = 1, eps = 1e-3, maxit = 1e2, ts = TRUE, method = "spline", sp.control = list(df = NULL, weights = NULL), ar.control = list(order = NULL, period = NULL), ga.control = list(formula, weights = NULL), f.eps = 1e-6, f.maxit = 1e3, ga.bf.eps = 1e-6, ga.bf.maxit = 1e3, verbose = FALSE, digits = getOption("digits"))`

Arguments

- `formula`: formula indicating the missing data frame, for instance, `~X1+X2+X3+...+Xp`
- `dataset`: data with missing values to be imputated
- `by`: factor for variance windows. Default is NULL for a single variance matrix
- `log`: logical. If TRUE data will be transformed into log scale. Default is FALSE
- `log.offset`: If log is TRUE, log values will be shifted by this offset. Default is 1
- `eps`: stop criterion
- `maxit`: maximum number of iterations
- `ts`: logical. TRUE if is time series
- `method`: method for univariate time series filtering. It may be smooth, gam or arima. See Details
- `sp.control`: list for Spline smooth control. See Details
`mnimput`

- `ar.control` list for ARIMA fitting control. See Details
- `ga.control` list for GAM fitting control. See Details
- `f.eps` convergence criterion for the ARIMA filter. See `arima`
- `f.maxit` maximum number of iterations for the ARIMA filter. See `arima`
- `ga.bf.eps` convergence criterion for the backfitting algorithm of GAM models. See `gam`
- `ga.bf.maxit` maximum number of iterations for the backfitting algorithm of GAM models. See `gam`
- `verbose` if TRUE convergence information on each iteration is printed. Default is FALSE
- `digits` an integer indicating the decimal places. If not supplied, it is taken from `options`

**Details**

This is a modified version of the EM algorithm for imputation of missing values. It is also applicable to time series data. When it is explicited the time series attribute through the argument `ts`, missing values are estimated accounting for both correlation between time series and time structure of the series itself. Several filters can be used for prediction of the mean vector in the E-step.

One can select the method for the univariate time series filtering by the argument `method`. The default method is "spline". In this case a smooth spline is fitted to each of the time series at each iteration. Some parameters can be passed to `smooth.spline` through `sm.control`. `df` is a vector as long as the number of columns in `dataset` holding fixed degrees of freedom of the splines. If NULL, the degrees of freedom of each spline are chosen by cross-validation. If `df` has length 1, this values is recycled for all the covariates. `weights` must be a matrix of the same size of `dataset` with the weights for `smooth.spline`. If NULL, all the observations will have weights equal to 1.

Other possibility for time series filtering is to fitting an ARIMA model for each of the time series by setting method to "arima". The ARIMA models must be identified before using this function, nonetheless. `arima` function can be partially controlled through `ar.control`. Each column of order must hold the corresponding `(p, d, q)` parameters for each univariate time series if `period` is NULL. If `period` is not NULL, order must also hold the multiplicative seasonality parameters, so each column of order takes the form `(p, d, q, P, D, Q)`. `period` is the multiplicative seasonality period. `f.eps` and `f.maxit` control de convergence of the ARIMA fitting algorithm. Convergence problems due non stationarity may arise when using this option.

Last but not least, a very interesting approach to modelling temporal patterns to use a full fledged regression model. It is possible to use generalised additive (or linear) models with exogenous variates to proper filtering of time patterns. One must set method to `gam` and supply a vector of formulas in `ga.control`. One must supply one formula for each covariate. Using covariates that are part of the formula of the imputation model may yield some colinearity among the variates. See `gam` and `glm` for details. In order to use regression models for the level, set `method` to "gam"

Simulations have shown that the algorithm is stable and yields good results on imputation of normal data.

**Value**

The function returns an object of class `mtsdi` containing

- `call` function call
- `dataset` imputed dataset
muhat     estimated mean vector
sigmahat   estimated covariance matrix
missings   vector holding the number of missing values on each row
iterations number of iterations until convergence or reach maxit
convergence convergence value. See Details
converged  a logical indicating if the algorithm converged
time       elapsed time of the process

Author(s)

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

References


See Also

mnimput, predict.mtsdi, edaprep

Examples

data(miss)
f <- ~c31+c32+c33+c34+c35
## one-window covariance
i <- mnimput(f,miss,eps=1e-3,ts=TRUE, method="spline",sp.control=list(df=c(7,7,7,7,7)))
summary(i)

## two-window covariances
b<-c(rep("year1",12),rep("year2",12))
ii <- mnimput(f,miss,by=b,eps=1e-3,ts=TRUE, method="spline",sp.control=list(df=c(7,7,7,7,7)))
summary(ii)
**Description**

Carry out some statistics from the incomplete dataset

**Usage**

```
mstats(dataset)
```

**Arguments**

- `dataset` dataset with missing for description

**Details**

This function computes the proportion of missing observations in a given dataset by rows and columns.

**Value**

A list containing

- `rows` number of missing in each row
- `columns` number of missing in each column
- `pattern` the pattern of the missing values

**Author(s)**

Washington Junger `<wjunger@ims.uerj.br>` and Antonio Ponce de Leon `<ponce@ims.uerj.br>`

**See Also**

- `mnimput`, `getmean`, `edaprep`

**Examples**

```
data(miss)
mstats(miss)
```
**plot.mtsdi**

**Plot the Imputed Matrix**

**Description**

This function produces a plot with imputed values and the estimated level for each of the columns in the imputed matrix.

**Usage**

```r
## S3 method for class 'mtsdi'
plot(x, vars = "all", overlay = TRUE, level = TRUE, points = FALSE, leg.loc = "topright", horiz = FALSE, at.once = FALSE, ...)
```

**Arguments**

- `x` an object of the class `mtsdi`
- `vars` a vector with the variables to plot
- `overlay` logical. If TRUE, observed values are plot over the imputed ones
- `level` logical. If TRUE, the level is plot
- `points` logical. If TRUE, points on the observed and imputed values are plot
- `leg.loc` a list with x and y coordinates for the legend or a quoted string. Default is "topright". See Details
- `horiz` logical. If TRUE, the legend will horizontal oriented
- `at.once` logical. If TRUE, all the variables are plot in separate windows at once
- `...` further options for function `plot`

**Details**

The `leg.loc` option may also be specified by setting one of the following quoted strings "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", or "center". This places the legend on the inside of the plot frame at the given location with the orientation set by `horiz`. See `legend` for further details.

**Author(s)**

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

**See Also**

`mnimput`
Examples

data(miss)
f <- ~c31+c32+c33+c34+c35
i <- mnimput(f,miss,eps=1e-3,ts=TRUE, method="spline", sp.control=list(df=c(7,7,7,7,7)))
plot(i)

predict.mtsdi Imputed Dataset Extraction

Description

Extract imputed dataset from a mtsdi object

Usage

## S3 method for class 'mtsdi'
predict(object, ...)

Arguments

object imputation object
...

further options passed to the generic function predict

Details

If log transformation was used, dataset is back transformed accordingly.

Value

A vector of of rows mean with length n, where n is the number of observations.

Author(s)

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

References

See Also

mnimput, getmean, edaprep

Examples

data(miss)
f <- ~c31+c32+c33+c34+c35
i <- mnimput(f,miss,eps=1e-3,ts=TRUE, method="spline",sp.control=list(df=c(7,7,7,7)))
predict(i)

print.mtsdi

Print Model Output

Description

Printing method for the imputation model

Usage

## S3 method for class 'mtsdi'
print(x, digits = getOption("digits"), ...)

Arguments

x an object of class summary.mtsdi
digits an integer indicating the decimal places. If not supplied, it is taken from options
...

Value

This function does not return a value.

Author(s)

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

See Also

mnimput

Examples

data(miss)
f <- ~c31+c32+c33+c34+c35
i <- mnimput(f,miss,eps=1e-3,ts=TRUE, method="spline",sp.control=list(df=c(7,7,7,7)))
predict(i)
print.summary.mtsdi

### Description

Printing method for the summary

### Usage

```r
## S3 method for class 'summary.mtsdi'
print(x, digits = getOption("digits"), print.models = TRUE, ...)
```

### Arguments

- **x**: an object of class `summary.mtsdi`
- **print.models**: a logical indicating that time filtering models should also be printed
- **digits**: an integer indicating the decimal places. If not supplied, it is taken from `options`
- **...**: further options passed from `summary.mtsdi`

### Value

This function does not return a value.

### Author(s)

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

### See Also

`mnimput`

### Examples

```r
data(miss)
f <- ~c31+c32+c33+c34+c35
i <- mnimput(f,miss,eps=1e-3,ts=TRUE, method="spline",sp.control=list(df=c(7,7,7,7)))
summary(i)
```
**Summary.mtsdi**

### Description

Print summary information on the imputation object

### Usage

```r
## S3 method for class 'mtsdi'
summary(object, ...)  

## S3 method for class 'mtsdi'
print(object, ...)  
```

### Arguments

- `object` an object of class `mtsdi`
- `...` further options passed to `print.summary.mtsdi`

### Value

The function returns a list containing

- `call` function call
- `muhat` estimated mean vector
- `sigmahat` estimated covariance matrix
- `iterations` number of iterations used
- `convergence` relative difference of covariance determinant reached
- `time` time used in the process
- `models` details on the models used for time filtering
- `log` a logical indicating that data are log transformed
- `log.offset` offset used in the log transformation in order to avoid zeros

### Author(s)

Washington Junger <wjunger@ims.uerj.br> and Antonio Ponce de Leon <ponce@ims.uerj.br>

### References

See Also

mnimput.predict

Examples

data(miss)
f <- ~c31+c32+c33+c34+c35
i <- mnimput(f,miss,eps=1e-3,ts=TRUE, method="spline", sp.control=list(df=c(7,7,7,7)))
summary(i)
Index

* NA
  edaprep, 2
  elapsedtime, 3
  getmean, 4
  mkjnw, 5
  mnimput, 6
  mstats, 9
  plot.mtsdi, 10
  predict.mtsdi, 11
  print.mtsdi, 12
  print.summary.mtsdi, 13
  summary.mtsdi, 14

* datasets
  miss, 5

* multivariate
  edaprep, 2
  elapsedtime, 3
  getmean, 4
  mkjnw, 5
  mnimput, 6
  mstats, 9
  plot.mtsdi, 10
  predict.mtsdi, 11
  print.mtsdi, 12
  print.summary.mtsdi, 13
  summary.mtsdi, 14

* smooth
  edaprep, 2
  elapsedtime, 3
  getmean, 4
  mkjnw, 5
  mnimput, 6
  mstats, 9
  plot.mtsdi, 10
  predict.mtsdi, 11
  print.mtsdi, 12
  print.summary.mtsdi, 13
  summary.mtsdi, 14

* ts
  edaprep, 2
  elapsedtime, 3
  getmean, 4
  mkjnw, 5
  mnimput, 6
  mstats, 9
  plot.mtsdi, 10
  predict.mtsdi, 11
  print.mtsdi, 12
  print.summary.mtsdi, 13
  summary.mtsdi, 14

arima, 7
edaprep, 2, 2, 4, 8, 9, 12
elapsedtime, 3
gam, 7
getmean, 2, 4, 4, 9, 12
glm, 7
legend, 10
miss, 5
mkjnw, 5
mnimput, 2–4, 6, 6, 8–10, 12, 13, 15
mstats, 9
options, 7, 12, 13
plot, 10
plot.mtsdi, 10
predict, 11, 15
predict.mtsdi, 8, 11
print, 12
print.mtsdi, 12
print.summary.mtsdi, 13, 14
smooth.spline, 7
summary.mtsdi, 13, 14