Package 'imputeTS'

March 20, 2018

Version 2.6

Date 2018-03-20

Title Time Series Missing Value Imputation

Description Imputation (replacement) of missing values
in univariate time series.
Offers several imputation functions
and missing data plots.
Available imputation algorithms include:
'Mean', 'LOCF', 'Interpolation',
'Moving Average', 'Seasonal Decomposition',
'Kalman Smoothing on Structural Time Series models',
'Kalman Smoothing on ARIMA models'.

Author Steffen Moritz

Maintainer Steffen Moritz <steffen.moritz10@gmail.com>

LazyData yes

Type Package

ByteCompile TRUE

BugReports https://github.com/SteffenMoritz/imputeTS/issues

URL https://github.com/SteffenMoritz/imputeTS

Repository CRAN

Depends R (>= 3.0.1)

Imports stats, stinepack, graphics, grDevices, forecast, Rcpp

Suggests testthat, utils

License GPL-3

VignetteBuilder utils

RoxygenNote 6.0.1

LinkingTo Rcpp

NeedsCompilation yes

Date/Publication 2018-03-20 18:08:55 UTC

R topics documented:

`imputeTS-package` *imputeTS-package description*

Description

The imputeTS package is a collection of algorithms and tools for univariate time series imputation.

Details

The imputeTS package specializes on (univariate) time series imputation. It offers several different imputation algorithm implementations. Beyond the imputation algorithms the package also provides plotting and printing functions of missing data statistics.

The package is easy to use:

- To impute (fill all missing values) in a time series `x`, run:
`> na.interpolation(x)`

- To plot missing data statistics for a time series `x`, run:
`> plotNA.distribution(x)`

- To print missing data statistics for a time series x, run:
> statsNA(x)

Every other imputation function (starting with na.'algorithm name') and plotting function (starting with plotNA.'plot name') work the same way as in this example.

References

Moritz, Steffen, et al. "Comparison of different Methods for Univariate Time Series Imputation in R." arXiv preprint arXiv:1510.03924 (2015).

`na.interpolation` *Missing Value Imputation by Interpolation*

Description

Uses either linear, spline or stineman interpolation to replace missing values.

Usage

```
na.interpolation(x, option = "linear", ...)
```

Arguments

`x`	Numeric Vector (`vector`) or Time Series (`ts`) object in which missing values shall be replaced
`option`	Algorithm to be used. Accepts the following input: • "linear" - for linear interpolation using approx • "spline" - for spline interpolation using spline • "stine" - for Stineman interpolation using stinterp
`...`	Additional parameters to be passed through to approx or spline interpolation functions

Details

Missing values get replaced by values of a approx, spline or stinterp interpolation.

Value

Vector (`vector`) or Time Series (`ts`) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

References

Johannesson, Tomas, et al. (2015). "Package stinepack".

See Also

na.kalman, na.locf, na.ma, na.mean, na.random, na.replace, na.seadec, na.seasplit

Examples

```
#Prerequisite: Create Time series with missing values
x <- ts(c(2,3,4,5,6,NA,7,8))

#Example 1: Perform linear interpolation
na.interpolation(x)

#Example 2: Perform spline interpolation
na.interpolation(x, option ="spline")

#Example 3: Perform stine interpolation
na.interpolation(x, option ="stine")
```

---

na.kalman *Missing Value Imputation by Kalman Smoothing and State Space Models*

Description

Uses Kalman Smoothing on structural time series models (or on the state space representation of an arima model) for imputation.

Usage

```
na.kalman(x, model = "StructTS", smooth = TRUE, nit = -1, ...)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced
model	Model to be used. With this parameter the State Space Model (on which KalmanSmooth is performed) can be chosen. Accepts the following input: • "auto.arima" - For using the state space representation of arima model (using auto.arima) • "StructTS" - For using a structural model fitted by maximum likelihood (using StructTS)

	For both auto.arima and StructTS additional parameters for model building can be given with the ... parameter Additionally it is also possible to use a user created state space model (See code Example 5). This state space model could for example be obtained from another R package for structural time series modeling. Furthermore providing the state space representation of a arima model from arima is also possible. But it is important to note, that user created state space models must meet the requirements specified under KalmanLike. This means the user supplied state space model has to be in form of a list with at least components T, Z, h , V, a, P, Pn. (more details under KalmanLike)
`smooth`	if `TRUE` - `KalmanSmooth` is used for estimation, if `FALSE` - `KalmanRun` is used. Since KalmanRun is often considered extrapolation KalmanSmooth is usually the better choice for imputation.
`nit`	Parameter from Kalman Filtering (see KalmanLike). Usually no need to change from default.
`...`	Additional parameters to be passed through to the functions that build the State Space Models (StructTS or auto.arima).

Details

The KalmanSmoother used in this function is `KalmanSmooth`. It operates either on a `Basic Structural Model` obtained by `StructTS` or the state space representation of a ARMA model obtained by `auto.arima`.

For an detailed explanation of Kalman Filtering and Space Space Models the following literature is a good starting point:

- *G. Welch, G. Bishop, An Introduction to the Kalman Filter. SIGGRAPH 2001 Course 8, 1995*
- *Harvey, Andrew C. Forecasting, structural time series models and the Kalman filter. Cambridge university press, 1990*
- *Grewal, Mohinder S. Kalman filtering. Springer Berlin Heidelberg, 2011*

Value

Vector (`vector`) or Time Series (`ts`) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

References

Hyndman RJ and Khandakar Y (2008). "Automatic time series forecasting: the forecast package for R". Journal of Statistical Software, 26(3).

See Also

`na.interpolation`, `na.locf`, `na.ma`, `na.mean`, `na.random`, `na.replace`, `na.seadec`, `na.seasplit`

Examples

```
#Example 1: Perform imputation with KalmanSmoother and state space representation of arima model
na.kalman(tsAirgap)

#Example 2: Perform imputation with KalmanRun and state space representation of arima model
na.kalman(tsAirgap, smooth = FALSE)

#Example 3: Perform imputation with KalmanSmooth and StructTS model
na.kalman(tsAirgap, model ="StructTS", smooth = TRUE)

#Example 4: Perform imputation with KalmanSmooth and StructTS model with additional parameters
na.kalman(tsAirgap, model ="StructTS", smooth = TRUE, type ="trend")

#Example 5:  Perform imputation with KalmanSmooth and user created model
usermodel <- arima(tsAirgap,order = c(1,0,1))$model
na.kalman(tsAirgap,model = usermodel)
```

---

na.locf *Missing Value Imputation by Last Observation Carried Forward*

Description

Replaces each missing value with the most recent present value prior to it (Last Observation Carried Forward- LOCF). Optionally this can also be done starting from the back of the series (Next Observation Carried Backward - NOCB).

Usage

```
na.locf(x, option = "locf", na.remaining = "rev")
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced
option	Algorithm to be used. Accepts the following input: • "locf" - for Last Observation Carried Forward • "nocb" - for Next Observation Carried Backward
na.remaining	Method to be used for remaining NAs. • "keep" - to return the series with NAs • "rm" - to remove remaining NAs • "mean" - to replace remaining NAs by overall mean • "rev" - to perform nocb / locf from the reverse direction

Details

Replaces each missing value with the most recent present value prior to it (Last Observation Carried Forward- LOCF). This can also be done from the reverse direction -starting from the back (Next Observation Carried Backward - NOCB). Both options have the issue, that NAs at the beginning (or for nocb at the end) of the time series cannot be imputed (since there is no last value to be carried forward present yet). In this case there are remaining NAs in the imputed time series. Since this only concerns very few values at the beginning of the series, na.remaining offers some quick solutions to get a series without NAs back.

Value

Vector (vector) or Time Series (ts) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.ma, na.mean, na.random, na.replace, na.seadec, na.seasplit

Examples

```
#Prerequisite: Create Time series with missing values
x <- ts(c(NA,3,4,5,6,NA,7,8))

#Example 1: Perform LOCF
na.locf(x)

#Example 2: Perform NOCF
na.locf(x, option = "nocb")

#Example 3: Perform LOCF and remove remaining NAs
na.locf(x, na.remaining = "rm")
```

---

na.ma *Missing Value Imputation by Weighted Moving Average*

Description

Missing value replacement by weighted moving average. Uses semi-adaptive window size to ensure all NAs are replaced.

Usage

```
na.ma(x, k = 4, weighting = "exponential")
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced
k	integer width of the moving average window. Expands to both sides of the center element e.g. k=2 means 4 observations (2 left, 2 right) are taken into account. If all observations in the current window are NA, the window size is automatically increased until there are at least 2 non-NA values present.
weighting	Weighting to be used. Accepts the following input: • "simple" - Simple Moving Average (SMA) • "linear" - Linear Weighted Moving Average (LWMA) • "exponential" - Exponential Weighted Moving Average (EWMA)

Details

In this function missing values get replaced by moving average values. Moving Averages are also sometimes referred to as "moving mean", "rolling mean", "rolling average" or "running average".

The mean in this implementation taken from an equal number of observations on either side of a central value. This means for an NA value at position i of a time series, the observations i-1,i+1 and i+1, i+2 (assuming a window size of k=2) are used to calculate the mean.

Since it can in case of long NA gaps also occur, that all values next to the central value are also NA, the algorithm has a semi-adaptive window size. Whenever there are less than 2 non-NA values in the complete window available, the window size is incrementally increased, till at least 2 non-NA values are there. In all other cases the algorithm sticks preset window size.

There are options for using Simple Moving Average (SMA), Linear Weighted Moving Average (LWMA) and Exponential Weighted Moving Average (EWMA).

SMA: all observations in the window are equally weighted for calculating the mean.

LWMA: weights decrease in arithmetical progression. The observations directly next to a central value i, have weight 1/2, the observations one further away (i-2,i+2) have weight 1/3, the next (i-3,i+3) have weight 1/4, ...

EWMA: uses weighting factors which decrease exponentially. The observations directly next to a central value i, have weight 1/2^1, the observations one further away (i-2,i+2) have weight 1/2^2, the next (i-3,i+3) have weight 1/2^3, ...

Value

Vector (vector) or Time Series (ts) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.locf, na.mean, na.random, na.replace, na.seadec, na.seasplit

Examples

```
#Example 1: Perform imputation with simple moving average
na.ma(tsAirgap, weighting = "simple")

#Example 2: Perform imputation with exponential weighted moving average
na.ma(tsAirgap)

#Example 3: Perform imputation with exponential weighted moving average, window size 6
na.ma(tsAirgap, k=6)
```

`na.mean` *Missing Value Imputation by Mean Value*

Description

Missing value replacement by mean values. Different means like median, mean, mode possible.

Usage

```
na.mean(x, option = "mean")
```

Arguments

`x`	Numeric Vector (`vector`) or Time Series (`ts`) object in which missing values shall be replaced
`option`	Algorithm to be used. Accepts the following input: • "mean" - take the mean for imputation • "median" - take the median for imputation • "mode" - take the mode for imputation

Details

Missing values get replaced by overall mean values. The function calculates the mean, median or mode over all the non-NA values and replaces all NAs with this value. Option 'mode' replaces NAs with the most frequent value in the time series. If two or more values occur equally frequent, the function imputes with the lower value. That's why 'mode' is not the best option for decimal values.

Value

Vector (`vector`) or Time Series (`ts`) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.locf, na.ma, na.random, na.replace, na.seadec, na.seasplit

Examples

```
#Prerequisite: Create Time series with missing values
x <- ts(c(2,3,4,5,6,NA,7,8))

#Example 1: Perform imputation with the overall mean
na.mean(x)

#Example 2: Perform imputation with overall median
na.mean(x, option ="median")
```

---

na.random *Missing Value Imputation by Random Sample*

Description

Replaces each missing value by drawing a random sample between two given bounds.

Usage

```
na.random(x, lowerBound = min(x, na.rm = TRUE), upperBound = max(x, na.rm =
  TRUE))
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced
lowerBound	Lower bound for the random samples (default is min(data))
upperBound	Upper bound for the random samples (default is max(data))

Details

Replaces each missing value by drawing a random sample between two given bounds. The default bounds are the minimum and the maximum value in the non-NAs from the time series. Function uses runif function to get the random values.

Value

Vector (vector) or Time Series (ts) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.locf, na.ma, na.mean, na.replace, na.seadec, na.seasplit

Examples

```
#Prerequisite: Create Time series with missing values
x <- ts(c(2,3,NA,5,6,NA,7,8))

#Example 1: Replace all NAs by random values that are between min and max of the input time series
na.random(x)

#Example 2: Replace all NAs by random values between 1 and 10
na.random(x, lowerBound = 1, upperBound = 10)
```

---

na.remove *Remove Missing Values*

Description

Removes all missing values from a time series.

Usage

```
na.remove(x)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced

Details

Removes all missing values from a input time series. This shortens the time series by the number of missing values in the series. Should be handled with care, because this can affect the seasonality of the time series. Seasonal patterns might be destroyed. Independent from the input, this function only returns a vector. (the time information of a resulting time series object wouldn't be correct any more).

Value

Vector (vector)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.locf, na.ma, na.mean, na.random, na.replace, na.seadec, na.seasplit

Examples

```
#Example 1: Remove all NAs
#Create Time series with missing values
x <- ts(c(2,3,NA,5,6,NA,7,8))

#Remove all NAs
na.remove(x)


#Example 2: Remove all NAs in tsAirgap
na.remove(tsAirgap)
```

---

na.replace *Replace Missing Values by a Defined Value*

Description

Replaces all missing values with a given value.

Usage

```
na.replace(x, fill = 0)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced
fill	Value used to replace the missing values

Value

Vector (vector) or Time Series (ts) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.locf, na.ma, na.mean, na.random, na.seadec, na.seasplit

Examples

```
#Prerequisite: Create Time series with missing values
x <- ts(c(2,3,NA,5,6,NA,7,8))

#Example 1: Replace all NAs with 3.5
na.replace(x, fill = 3.5 )

#Example 2: Replace all NAs with 0
na.replace(x, fill = 0 )
```

---

na.seadec *Seasonally Decomposed Missing Value Imputation*

Description

Removes the seasonal component from the time series, performs imputation on the deseasonalized series and afterwards adds the seasonal component again.

Usage

```
na.seadec(x, algorithm = "interpolation", ...)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced
algorithm	Algorithm to be used after decomposition. Accepts the following input: • "interpolation" - Imputation by Interpolation • "locf" - Imputation by Last Observation Carried Forward • "mean" - Imputation by Mean Value • "random" - Imputation by Random Sample • "kalman" - Imputation by Kalman Smoothing and State Space Models • "ma" - Imputation by Weighted Moving Average
...	Additional parameters for these algorithms that can be passed through. Look at na.interpolation, na.locf, na.random, na.mean for parameter options.

Value

Vector (vector) or Time Series (ts) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.locf, na.ma, na.mean, na.random, na.replace, na.seasplit

Examples

```
#Example 1: Perform seasonal imputation using algorithm = "interpolation"
na.seadec(tsAirgap, algorithm = "interpolation")

#Example 2: Perform seasonal imputation using algorithm = "mean"
na.seadec(tsAirgap, algorithm = "mean")
```

---

na.seasplit *Seasonally Splitted Missing Value Imputation*

Description

Splits the times series into seasons and afterwards performs imputation separately for each of the resulting time series datasets (each containing the data for one specific season).

Usage

```
na.seasplit(x, algorithm = "interpolation", ...)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object in which missing values shall be replaced
algorithm	Algorithm to be used after splits. Accepts the following input: • "interpolation" - Imputation by Interpolation • "locf" - Imputation by Last Observation Carried Forward • "mean" - Imputation by Mean Value • "random" - Imputation by Random Sample • "kalman" - Imputation by Kalman Smoothing and State Space Models • "ma" - Imputation by Weighted Moving Average
...	Additional parameters for these algorithms that can be passed through. Look at na.interpolation, na.locf, na.random, na.mean for parameter options.

Value

Vector (vector) or Time Series (ts) object (dependent on given input at parameter x)

Author(s)

Steffen Moritz

See Also

na.interpolation, na.kalman, na.locf, na.ma, na.mean, na.random, na.replace, na.seadec

Examples

```
#Example 1: Perform seasonal splitted imputation using algorithm = "interpolation"
na.seasplit(tsAirgap, algorithm = "interpolation")

#Example 2: Perform seasonal splitted imputation using algorithm = "mean"
na.seasplit(tsAirgap, algorithm = "mean")
```

---

plotNA.distribution *Visualize Distribution of Missing Values*

Description

Visualize the distribution of missing values within a time series.

Usage

```
plotNA.distribution(x, colPoints = "steelblue",
  colBackgroundMV = "indianred2", main = "Distribution of NAs",
  xlab = "Time", ylab = "Value", pch = 20, cexPoints = 0.8,
  col = "black", ...)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object containing NAs
colPoints	Color of the points for each observation
colBackgroundMV	Color for the background for the NA sequences
main	Main label for the plot
xlab	Label for x axis of the plot
ylab	Label for y axis of plot
pch	Plotting 'character', i.e., symbol to use.
cexPoints	character (or symbol) expansion: a numerical vector.
col	Color for the lines.
...	Additional graphical parameters that can be passed through to plot

Details

This function visualizes the distribution of missing values within a time series. Therefore, the time series is plotted and whenever a value is NA the background is colored differently. This gives a nice overview, where in the time series most of the missing values occur.

Author(s)

Steffen Moritz

See Also

plotNA.distributionBar, plotNA.gapsize, plotNA.imputations

Examples

```
#Example 1: Visualize the missing values in x
x <- ts(c(1:11, 4:9,NA,NA,NA,11:15,7:15,15:6,NA,NA,2:5,3:7))
plotNA.distribution(x)

#Example 2: Visualize the missing values in tsAirgap time series
plotNA.distribution(tsAirgap)
```

---

plotNA.distributionBar

Visualize Distribution of Missing Values (Barplot)

Description

Visualization of missing values in barplot form. Especially useful for time series with a lot of observations.

Usage

```
plotNA.distributionBar(x, breaks = nclass.Sturges(x), breaksize = NULL,
  percentage = TRUE, legend = TRUE, axis = TRUE, space = 0,
  col = c("indianred2", "green2"), main = "Distribution of NAs",
  xlab = "Time Lapse", ylab = NULL, ...)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object containing NAs
breaks	Defines the number of bins to be created. Default number of breaks is calculated by nclass.Sturges using Sturges' formula. If the breaksize parameter is set to a value different to NULL this parameter is ignored.
breaksize	Defines how many observations should be in one bin. The required number of overall bins is afterwards calculated automatically. This parameter if used overwrites the breaks parameter.
percentage	Whether the NA / non-NA ration should be given as percent or absolute numbers
legend	If TRUE a legend is shown at the bottom of the plot. A custom legend can be obtained by setting this parameter to FALSE and using legend function

axis	If TRUE a x-axis with labels is added. A custom axis can be obtained by setting this parameter to FALSE and using axis function
space	The amount of space (as a fraction of the average bar width) left before each bar.
col	A vector of colors for the bars or bar components.
main	Main title for the plot
xlab	Label for x axis of the plot
ylab	Label for y axis of plot
...	Additional graphical parameters that can be passed through to barplot

Details

This function visualizes the distribution of missing values within a time series. In comparison to the plotNA.distribution function this is not done by plotting each observation of the time series separately Instead observations for time intervals are represented as bars. For these intervals information about the amount of missing values are shown. This has the advantage, that also for large time series a plot which is easy to overview can be created.

Author(s)

Steffen Moritz

See Also

plotNA.distribution, plotNA.gapsize, plotNA.imputations

Examples

```
#Example 1: Visualize the missing values in tsNH4 time series
plotNA.distributionBar(tsNH4)

#Example 2: Visualize the missing values in tsHeating time series
plotNA.distributionBar(tsHeating, breaks = 20)
```

---

plotNA.gapsize *Visualize Distribution of NA gap sizes*

Description

Visualize Distribution of NA gap sizes (NAs in a row) in a time series

Usage

```
plotNA.gapsize(x, limit = 10, byTotalNA = FALSE, legend = TRUE,
  col = c("indianred", "steelblue"),
  xlab = "Ranking of the different gap sizes", ylab = "Number",
  main = "Occurrence of gap sizes (NAs in a row)", cex.names = 0.7,
  horiz = FALSE, axes = TRUE, beside = TRUE, las = 1, ...)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object containing NAs
limit	Specifies how many of the top gap sizes are shown in the plot.
byTotalNA	For byTotalNA = TRUE the top gap sizes according to their overall weight are shown. (occurrence * gap size) For byTotalNA = FALSE the top gap sizes are shown by their number of occurrence. (occurrence)
legend	If TRUE a legend is shown at the bottom of the plot. A custom legend can be obtained by setting this parameter to FALSE and using legend function
col	A vector of colors for the bars or bar components.
xlab	Label for x axis of the plot
ylab	Label for y axis of plot
main	Main title for the plot
cex.names	Expansion factor for axis names (bar labels).
horiz	A logical value. If FALSE, the bars are drawn vertically with the first bar to the left. If TRUE, the bars are drawn horizontally with the first at the bottom.
axes	Logical. If TRUE, a vertical (or horizontal, if horiz is true) axis is drawn.
beside	A logical value. If FALSE, the columns of height are portrayed as stacked bars, and if TRUE the columns are portrayed as juxtaposed bars.
las	Numeric in 0,1,2,3; the style of axis labels. 0:always parallel to the axis, 1:always horizontal, 2:always perpendicular to the axis, 3:always vertical.
...	Additional graphical parameters that can be passed through to barplot

Details

This plotting function can be used to visualize the length of the NA gaps (NAs in a row) in a time series. It shows a ranking of which gap sizes occur most often. This ranking can be ordered by total NAs for this gap size (occurrence * gap length) or by occurrence of the gap size. The outcome will be somethink like in the time series 2NAs in a row occurred 27times, 4NAs in a row occurred 11 times, 7NAs in a row occurred 5 times, 1NA in a row occurred 3 times, ...

Author(s)

Steffen Moritz

See Also

plotNA.distribution,plotNA.distributionBar, plotNA.imputations

Examples

```
#Example 1: Visualize the top gap sizes in tsNH4
plotNA.gapsize(tsNH4)

#Example 2: Visualize the top gap sizes in tsAirgap
plotNA.gapsize(tsAirgap)
```

plotNA.imputations *Visualize Imputed Values*

Description

Visualize the imputed values in a time series.

Usage

```
plotNA.imputations(x.withNA, x.withImputations, x.withTruth = NULL,
  legend = TRUE, main = "Visualization Imputed Values", xlab = "Time",
  ylab = "Value", colWithTruth = "green3", colLines = "black",
  colWithImputations = "indianred2", colWithNA = "steelblue2",
  ylim = c(min(c(x.withImputations, x.withTruth), na.rm = TRUE),
  max(c(x.withImputations, x.withTruth), na.rm = TRUE)), pch = 20,
  cex = 0.8, ...)
```

Arguments

x.withNA	Numeric Vector or Time Series (ts) object with NAs before imputation
x.withImputations	Numeric Vector or Time Series (ts) object with NAs replaced by imputed values
x.withTruth	Numeric Vector or Time Series (ts) object with the real values. (can be set to NULL if not known)
legend	If TRUE a legend is shown at the bottom of the plot. A custom legend can be obtained by setting this parameter to FALSE and using legend function
main	Main title for the plot
xlab	Label for x axis of the plot
ylab	Label for y axis of plot
colWithTruth	Defines the color of the real values (truth) for the NA values.
colLines	Defines the color of the lines connecting non-NA observations.
colWithImputations	Defines the color for the imputed values.
colWithNA	Defines the color of the non-NA observations.
ylim	the y limits of the plot
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
...	Additional graphical parameters that can be passed through to plot

Details

This plot can be used, to visualize the imputed values for a time series. Therefore, the imputed values (filled NA gaps) are shown in a different color than the other values. If the real values (truth) behind the NA gaps are known these are also added in a different color.

Author(s)

Steffen Moritz

See Also

plotNA.distribution,plotNA.distributionBar, plotNA.gapsize

Examples

```
#Example 1: Visualize the values that were imputed by na.mean in the time series
impMean.Airgap <- na.mean(tsAirgap)
plotNA.imputations(tsAirgap, impMean.Airgap)


#Example 2: Visualize the values that were imputed by na.locf and the true values in the time series
impLOCF.Airgap <- na.locf(tsAirgap)
plotNA.imputations(tsAirgap, impLOCF.Airgap, tsAirgapComplete)
```

---

statsNA *Print Statistics about Missing Values*

Description

Print summary stats about the distribution of missing values in a univariate time series.

Usage

```
statsNA(x, bins = 4, printOnly = TRUE)
```

Arguments

x	Numeric Vector (vector) or Time Series (ts) object containing NAs
bins	Split number for bin stats. Number of bins the time series gets divided into. For each bin information about amount/percentage of missing values is printed. Default value is 4 - what means stats about the 1st,2nd,3rd,4th quarter of the time series are shown.
printOnly	Choose if the function Prints or Returns. For printOnly = TRUE the function has no return value and just prints out missing value stats. If printOnly is changed to FALSE, nothing is printed and the function returns a list. Print gives a little bit more information, since the returned list does not include "Stats for Bins" and "overview NA series"

Details

Prints the following information about the missing values in the time series:

- "Length of time series" - Number of observations in the time series (including NAs)
- "Number of Missing Values" - Number of missing values in the time series
- "Percentage of Missing Values" - Percentage of missing values in the time series
- "Stats for Bins" - Number/percentage of missing values for the split into bins
- "Longest NA gap" - Longest series of consecutive missing values (NAs in a row) in the time series
- "Most frequent gap size" - Most frequent occurring series of missing values in the time series
- "Gap size accounting for most NAs" - The series of consecutive missing values that accounts for most missing values overall in the time series
- "Overview NA series" - Overview about how often each series of consecutive missing values occurs. Series occurring 0 times are skipped

It is furthermore, important to note, that you are able to choose whether the function returns a list or prints the information only. (see description of parameter "printOnly")

Value

A `list` containing the stats. Beware: Function gives only a return value if printOnly = FALSE.

Author(s)

Steffen Moritz

See Also

`plotNA.distribution`,`plotNA.distributionBar`, `plotNA.gapsize`

Examples

```
#Example 1: Print stats about the missing data in tsNH4
statsNA(tsNH4)

#Example 2: Return list with stats about the missing data in tsAirgap
statsNA(tsAirgap, printOnly= FALSE)
```

`tsAirgap` *Time series of monthly airline passengers (with NAs)*

Description

Monthly totals of international airline passengers, 1949 to 1960. This time series contains missing values. In the package included is also the `tsAirgapComplete` time series providing the true values for the missing values.

Usage

```
tsAirgap
```

Format

Time Series (`ts`) with 144 rows including 13 NAs.

Details

The dataset originates from Box & Jenkins (see citation) and is a commonly used example in time series analysis literature.

It characteristics (strong trend, strong seasonal behavior) make it also a great example for time series imputation. Thus the version with inserted NA gaps was created under the name tsAirgap.

In order to use this series for comparing imputation algorithm results, there are two time series provided. One series without missing values, which can be used as ground truth. Another series with NAs, on which the imputation algorithms can be applied.

There are the two time series:

- tsAirgap - The time series with NAs.
- tsAirgapComplete - Time series without NAs.

Source

Box, G. E. P., Jenkins, G. M., Reinsel, G. C. and Ljung, G. M. (2015). Time series analysis: forecasting and control. Fifth Edition. John Wiley \& Sons.

See Also

`tsHeating`, `tsNH4`

`tsAirgapComplete` *Time series of monthly airline passengers (complete)*

Description

Monthly totals of international airline passengers, 1949 to 1960. This time series provides the truth for the missing values of the `tsAirgap` time series. Thus it is identical to the tsAirgap time series except that no value is missing.

Usage

```
tsAirgapComplete
```

Format

Time Series (`ts`) with 144 rows.

Details

The dataset originates from Box & Jenkins (see citation) and is a commonly used example in time series analysis literature.

It characteristics (strong trend, strong seasonal behavior) make it also a great example for time series imputation. Thus the version with inserted NA gaps was created under the name tsAirgap.

In order to use this series for comparing imputation algorithm results, there are two time series provided. One series without missing values, which can be used as ground truth. Another series with NAs, on which the imputation algorithms can be applied.

There are the two time series:

- tsAirgap - The time series with NAs.
- tsAirgapComplete - Time series without NAs.

Source

Box, G. E. P., Jenkins, G. M., Reinsel, G. C. and Ljung, G. M. (2015). Time series analysis: forecasting and control. Fifth Edition. John Wiley \& Sons.

See Also

`tsHeating`, `tsNH4`

tsHeating *Time series of a heating systems supply temperature (with NAs)*

Description

Time series of a heating systems supply temperature. Measured from 18.11.2013 - 05:12:00 to 13.01.2015 - 15:08:00 in 1 minute steps. This time series contains missing values. In the package included is also the tsHeatingComplete time series providing the true values for the missing values.

Usage

```
tsHeating
```

Format

Time Series (ts) with 606837 rows including 57391 NAs.

Details

The time series originates from the GECCO Industrial Challenge 2015. This Challenge was about "Recovering missing information in heating system operating data". Goal was to impute missing values in heating system sensor data as accurate as possible. (http://www.spotseven.de/gecco-challenge/gecco-challenge-2015/)

In order to use this series for comparing imputation algorithm results, there are two time series provided. One series without missing values, which can be used as ground truth. Another series with NAs, on which the imputation algorithms can be applied. The NAs thereby were inserted according to patterns found in similar time series.

There are the two time series:

- tsHeating - The time series with NAs.
- tsHeatingComplete - Time series without NAs.

Source

http://www.spotseven.de/gecco-challenge/gecco-challenge-2015/

See Also

tsAirgap, tsNH4

tsHeatingComplete *Time series of a heating systems supply temperature (complete)*

Description

Time series of a heating systems supply temperature. Measured from 18.11.2013 - 05:12:00 to 13.01.2015 - 15:08:00 in 1 minute steps. This time series provides the truth for the missing values of the tsHeating time series. Thus it is identical to the heating time series except that no value is missing.

Usage

```
tsHeatingComplete
```

Format

Time Series (ts) with 606837 rows.

Details

The time series originates from the GECCO Industrial Challenge 2015. This Challenge was about "Recovering missing information in heating system operating data". Goal was to impute missing values in heating system sensor data as accurate as possible. (http://www.spotseven.de/gecco-challenge/gecco-challenge-2015/)

In order to use this series for comparing imputation algorithm results, there are two time series provided. One series without missing values, which can be used as ground truth. Another series with NAs, on which the imputation algorithms can be applied. The NAs thereby were inserted according to patterns found in similar time series.

There are the two time series:

- tsHeating - The time series with NAs.
- tsHeatingComplete - Time series without NAs.

Source

http://www.spotseven.de/gecco-challenge/gecco-challenge-2015/

See Also

tsAirgap, tsNH4

tsNH4 *Time series of NH4 concentration in a wastewater system (with NAs)*

Description

Time series of NH4 concentration in a wastewater system. Measured from 30.11.2010 - 16:10 to 01.01.2011 - 6:40 in 10 minute steps. This time series contains missing values. In the package included is also the tsNH4Complete time series providing the true values for the missing values.

Usage

```
tsNH4
```

Format

Time Series (ts) with 4552 rows including 883 NAs.

Details

The time series is derived from the dataset of the GECCO Industrial Challenge 2014.

In order to use this series for comparing imputation algorithm results, there are two time series provided. One series without missing values, which can be used as ground truth. Another series with NAs, on which the imputation algorithms can be applied. The NAs thereby were inserted according to patterns found in similar time series.

There are the two time series:

- tsNH4 - The time series with NAs.
- tsNH4Complete - Time series without NAs.

Source

http://www.spotseven.de/gecco-challenge/gecco-challenge-2014/

See Also

tsAirgap,tsHeating

`tsNH4Complete` *Time series of NH4 concentration in a wastewater system (complete)*

Description

Time series of NH4 concentration in a wastewater system. Measured from 30.11.2010 - 16:10 to 01.01.2011 - 6:40 in 10 minute steps. This time series provides the truth for the missing values of the `tsNH4` time series. Thus it is identical to the heating time series except that no value is missing.

Usage

```
tsNH4Complete
```

Format

Time Series (`ts`) with 4552 rows.

Details

The time series is derived from the dataset of the GECCO Industrial Challenge 2014.

In order to use this series for comparing imputation algorithm results, there are two time series provided. One series without missing values, which can be used as ground truth. Another series with NAs, on which the imputation algorithms can be applied. The NAs thereby were inserted according to patterns found in similar time series.

There are the two time series:

- tsNH4 - The time series with NAs.
- tsNH4Complete - Time series without NAs.

Source

http://www.spotseven.de/gecco-challenge/gecco-challenge-2014/

See Also

`tsAirgap`,`tsHeating`

Index