Package ‘imputeFin’

February 20, 2021

Title  Imputation of Financial Time Series with Missing Values and/or Outliers
Version 0.1.2
Date 2021-02-19

Description  Missing values often occur in financial data due to a variety of reasons (errors in the collection process or in the processing stage, lack of asset liquidity, lack of reporting of funds, etc.). However, most data analysis methods expect complete data and cannot be employed with missing values. One convenient way to deal with this issue without having to redesign the data analysis method is to impute the missing values. This package provides an efficient way to impute the missing values based on modeling the time series with a random walk or an autoregressive (AR) model, convenient to model log-prices and log-volumes in financial data. In the current version, the imputation is univariate-based (so no asset correlation is used). In addition, outliers can be detected and removed.

The package is based on the paper:

Maintainer  Daniel P. Palomar <daniel.p.palomar@gmail.com>

URL  https://CRAN.R-project.org/package=imputeFin,
     https://github.com/dppalomar/imputeFin,
     https://www.danielpalomar.com,
     https://doi.org/10.1109/TSP.2019.2899816,
     https://doi.org/10.1109/TSP.2020.3033378

BugReports  https://github.com/dppalomar/imputeFin/issues

License  GPL-3
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.1
**imputeFin-package**

**imputeFin: Imputation of Financial Time Series with Missing Values.**

**Description**

Missing values often occur in financial data due to a variety of reasons (errors in the collection process or in the processing stage, lack of asset liquidity, lack of reporting of funds, etc.). However, most data analysis methods expect complete data and cannot be employed with missing values. One convenient way to deal with this issue without having to redesign the data analysis method is to impute the missing values. This package provides an efficient way to impute the missing values based on modeling the time series with a random walk or an autoregressive (AR) model, convenient to model log-prices and log-volumes in financial data. In the current version, the imputation is univariate-based (so no asset correlation is used). In addition, outliers can be detected and removed.

**Functions**

`fit_AR1_Gaussian, impute_AR1_Gaussian, fit_AR1_t, impute_AR1_t, plot_imputed`
fit_AR1_Gaussian

Data

\texttt{ts_AR1_Gaussian, ts_AR1_t}

Help

For a quick help see the README file: GitHub-README. For more details see the vignette: CRAN-vignette.

Author(s)

Junyan Liu, Rui Zhou, and Daniel P. Palomar

References


fit_AR1_Gaussian \hspace{1cm} \text{Fit Gaussian AR(1) model to time series with missing values and/or outliers}

Description

Estimate the parameters of a univariate Gaussian AR(1) model to fit the given time series with missing values and/or outliers. For multivariate time series, the function will perform a number of individual univariate fittings without attempting to model the correlations among the time series. If the time series does not contain missing values, the maximum likelihood (ML) estimation is done in one shot. With missing values, the iterative EM algorithm is employed for the estimation until converge is achieved.

Usage

\begin{verbatim}
fit_AR1_Gaussian(
  y,
  random_walk = FALSE,
  zero_mean = FALSE,
  remove_outliers = FALSE,
  outlier_prob_th = 0.001,
  verbose = TRUE,
  return_iterates = FALSE,
  return_condMeanCov = FALSE,
  tol = 1e-08,
  maxiter = 100
)
\end{verbatim}
Arguments

*y* Time series object coercible to either a numeric vector or numeric matrix (e.g., `zoo` or `xts`) with missing values denoted by `NA`.

`random_walk` Logical value indicating if the time series is assumed to be a random walk so that \( \phi_1 = 1 \) (default is `FALSE`).

`zero_mean` Logical value indicating if the time series is assumed zero-mean so that \( \phi_0 = 0 \) (default is `FALSE`).

`remove_outliers` Logical value indicating whether to detect and remove outliers.

`outlier_prob_th` Threshold of probability of observation to declare an outlier (default is `1e-3`).

`verbose` Logical value indicating whether to output messages (default is `TRUE`).

`return_iterates` Logical value indicating if the iterates are to be returned (default is `FALSE`).

`return_condMeanCov` Logical value indicating if the conditional mean and covariance matrix of the time series (excluding the leading and trailing missing values) given the observed data are to be returned (default is `FALSE`).

`tol` Positive number denoting the relative tolerance used as stopping criterion (default is `1e-8`).

`maxiter` Positive integer indicating the maximum number of iterations allowed (default is 100).

Value

If the argument *y* is a univariate time series (i.e., coercible to a numeric vector), then this function will return a list with the following elements:

`phi0` The estimate for \( \phi_0 \) (real number).

`phi1` The estimate for \( \phi_1 \) (real number).

`sigma2` The estimate for \( \sigma^2 \) (positive number).

`phi0_iterates` Numeric vector with the estimates for \( \phi_0 \) at each iteration (returned only when `return_iterates = TRUE`).

`phi1_iterates` Numeric vector with the estimates for \( \phi_1 \) at each iteration (returned only when `return_iterates = TRUE`).

`sigma2_iterates` Numeric vector with the estimates for \( \sigma^2 \) at each iteration (returned only when `return_iterates = TRUE`).

`f_iterates` Numeric vector with the objective values at each iteration (returned only when `return_iterates = TRUE`).

`cond_mean_y` Numeric vector (of same length as argument *y*) with the conditional mean of the time series (excluding the leading and trailing missing values) given the observed data (returned only when `return_condMeanCov = TRUE`).
cond_cov_y Numeric matrix (with number of columns/rows equal to the length of the argument y) with the conditional covariance matrix of the time series (excluding the leading and trailing missing values) given the observed data (returned only when return_condMeanCov = TRUE).

index_miss Indices of missing values imputed.

index_outliers Indices of outliers detected/corrected.

If the argument y is a multivariate time series (i.e., with multiple columns and coercible to a numeric matrix), then this function will return a list with each element as in the case of univariate y corresponding to each of the columns (i.e., one list element per column of y), with the following additional elements that combine the estimated values in a convenient vector form:

phi0_vct Numeric vector (with length equal to the number of columns of y) with the estimates for phi0 for each of the univariate time series.

phi1_vct Numeric vector (with length equal to the number of columns of y) with the estimates for phi1 for each of the univariate time series.

sigma2_vct Numeric vector (with length equal to the number of columns of y) with the estimates for sigma2 for each of the univariate time series.

Author(s)

Junyan Liu and Daniel P. Palomar

References


See Also

impute_AR1_Gaussian, fit_AR1_t

Examples

library(imputeFin)
data(ts_AR1_Gaussian)
y_missing <- ts_AR1_Gaussian$y_missing
fitted <- fit_AR1_Gaussian(y_missing)
Fit Student's t AR(1) model to time series with missing values and/or outliers

Description

Estimate the parameters of a univariate Student's t AR(1) model to fit the given time series with missing values and/or outliers. For multivariate time series, the function will perform a number of individual univariate fittings without attempting to model the correlations among the time series. If the time series does not contain missing values, the maximum likelihood (ML) estimation is done via the iterative EM algorithm until converge is achieved. With missing values, the stochastic EM algorithm is employed for the estimation (currently the maximum number of iterations will be executed without attempting to check early converge).

Usage

```r
fit_AR1_t(
  y,
  random_walk = FALSE,
  zero_mean = FALSE,
  fast_and_heuristic = TRUE,
  remove_outliers = FALSE,
  outlier_prob_th = 0.001,
  verbose = TRUE,
  return_iterates = FALSE,
  return_condMean_Gaussian = FALSE,
  tol = 1e-08,
  maxiter = 100,
  n_chain = 10,
  n_thin = 1,
  K = 30
)
```

Arguments

- **y**: Time series object coercible to either a numeric vector or numeric matrix (e.g., `zoo` or `xts`) with missing values denoted by `NA`.
- **random_walk**: Logical value indicating if the time series is assumed to be a random walk so that \( \phi_1 = 1 \) (default is `FALSE`).
- **zero_mean**: Logical value indicating if the time series is assumed zero-mean so that \( \phi_0 = 0 \) (default is `FALSE`).
- **fast_and_heuristic**: Logical value indicating whether a heuristic but fast method is to be used to estimate the parameters of the Student’s t AR(1) model (default is `TRUE`).
- **remove_outliers**: Logical value indicating whether to detect and remove outliers.
outlier_prob_th
Threshold of probability of observation to declare an outlier (default is 1e-3).

verbose
Logical value indicating whether to output messages (default is TRUE).

return_iterates
Logical value indicating if the iterates are to be returned (default is FALSE).

return_condMean_Gaussian
Logical value indicating if the conditional mean and covariance matrix of the
time series (excluding the leading and trailing missing values) given the ob-
served data are to be returned (default is FALSE).

tol
Positive number denoting the relative tolerance used as stopping criterion (de-
default is 1e-8).

maxiter
Positive integer indicating the maximum number of iterations allowed (default
is 100).

n_chain
Positive integer indicating the number of the parallel Markov chains in the
stochastic EM method (default is 10).

n_thin
Positive integer indicating the sampling period of the Gibbs sampling in the
stochastic EM method (default is 1). Every n_thin-th samples is used. This is
aimed to reduce the dependence of the samples.

K
Positive number controlling the values of the step sizes in the stochastic EM
method (default is 30).

Value
If the argument y is a univariate time series (i.e., coercible to a numeric vector), then this function
will return a list with the following elements:

phi0
The estimate for phi0 (real number).

phi1
The estimate for phi1 (real number).

sigma2
The estimate for sigma^2 (positive number).

nu
The estimate for nu (positive number).

phi0_iterates
Numeric vector with the estimates for phi0 at each iteration (returned only when
return_iterates = TRUE).

phi1_iterates
Numeric vector with the estimates for phi1 at each iteration (returned only when
return_iterates = TRUE).

sigma2_iterates
Numeric vector with the estimates for sigma^2 at each iteration (returned only when
return_iterates = TRUE).

nu_iterate
Numeric vector with the estimates for nu at each iteration (returned only when
return_iterates = TRUE).

f_iterates
Numeric vector with the objective values at each iteration (returned only when
return_iterates = TRUE).

cond_mean_y_Gaussian
Numeric vector (of same length as argument y) with the conditional mean of the
time series (excluding the missing values at the head and tail) given the observed
data based on Gaussian AR(1) model (returned only when return_condMean_Gaussian
= TRUE).
index_miss Indices of missing values imputed.

index_outliers Indices of outliers detected/corrected.

If the argument \( y \) is a multivariate time series (i.e., with multiple columns and coercible to a numeric matrix), then this function will return a list with each element as in the case of univariate \( y \) corresponding to each of the columns (i.e., one list element per column of \( y \)), with the following additional elements that combine the estimated values in a convenient vector form:

\[
\begin{align*}
\phi_0 \_vct & \quad \text{Numeric vector (with length equal to the number of columns of } y \text{) with the estimates for } \phi_0 \text{ for each of the univariate time series.} \\
\phi_1 \_vct & \quad \text{Numeric vector (with length equal to the number of columns of } y \text{) with the estimates for } \phi_1 \text{ for each of the univariate time series.} \\
\sigma^2 \_vct & \quad \text{Numeric vector (with length equal to the number of columns of } y \text{) with the estimates for } \sigma^2 \text{ for each of the univariate time series.} \\
\nu \_vct & \quad \text{Numeric vector (with length equal to the number of columns of } y \text{) with the estimates for } \nu \text{ for each of the univariate time series.}
\end{align*}
\]

Author(s)

Junyan Liu and Daniel P. Palomar

References


See Also

impute_AR1_t, fit_AR1_Gaussian, fit_VAR_t

Examples

```r
library(imputeFin)
data(ts_AR1_t)
y_missing <- ts_AR1_t$y_missing
fitted <- fit_AR1_t(y_missing)
```

| fit_VAR_t | Fit Student's t VAR model to time series with missing values and/or outliers |
Description

Estimate the parameters of a Student’s t vector autoregressive model

\[ y_t = \phi_0 + \sum_{i=1}^{p} \Phi_i y_{t-i} + \epsilon_t \]

to fit the given time series with missing values. If the time series does not contain missing values, the maximum likelihood (ML) estimation is done via the iterative EM algorithm until convergence is achieved. With missing values, the stochastic EM algorithm is employed for the estimation (currently the maximum number of iterations will be executed without attempting to check early converge).

Usage

```r
fit_VAR_t(Y, p = 1, omit_missing = FALSE, parallel_max_cores = max(1, parallel::detectCores() - 1), verbose = FALSE, return_iterates = FALSE, initial = NULL, L = 10, maxiter = 50, ptol = 0.001, partition_groups = TRUE, K = round(maxiter/3))
```

Arguments

- `Y`: Time series object coercible to either a numeric matrix (e.g., zoo or xts) with missing values denoted by NA.
- `p`: Positive integer indicating the order of the VAR model.
- `omit_missing`: Logical value indicating whether to use the omit-variable method, i.e., excluding the variables with missing data from the analysis (default is `FALSE`).
- `parallel_max_cores`: Positive integer indicating the maximum number of cores used in the parallel computing, only valid when `partition_groups = TRUE` (default is 1).
- `verbose`: Logical value indicating whether to report in console the information of each iteration.
- `return_iterates`: Logical value indicating whether to return the parameter estimates at each iteration (default is `FALSE`).
- `initial`: List with the initial values of the parameters of the VAR model, which may contain some or all of the following elements:
  - `nu` (ν): A positive number as the degrees of freedom,
• \( \phi_0 (\phi_0) \) - a numerical vector of length \( \text{ncol}(Y) \) as the interception of VAR model,
• \( \Phi_i (\Phi_i) \) - a list of \( p \) matrices of dimension \( \text{ncol}(Y) \) as the autoregressive coefficient matrices,
• \( \Sigma (\Sigma) \) - a positive definite of dimension \( \text{ncol}(Y) \) as the scatter matrix.

\( L \)  Positive integer with the number of Markov chains (default is 10).

\( \text{maxiter} \)  Positive integer with the number of maximum iterations (default is 100).

\( \text{ptol} \)  Non-negative number with the tolerance to determine the convergence of the (stochastic) EM method.

\( \text{partition\_groups} \)  Logical value indicating whether to partition \( Y \) into groups (default is \text{TRUE}).

\( K \)  Positive integer indicating the values of the step sizes in the stochastic EM method.

**Value**

A list with the following elements:

- \( \nu \)  The estimate for \( \nu \).
- \( \phi_0 \)  The estimate for \( \phi_0 \).
- \( \Phi_i \)  The estimate for \( \Phi_i \).
- \( \Sigma \)  The estimate for scatter matrix, i.e., \( \Sigma \).
- \( \text{converged} \)  A logical value indicating whether the method has converged.
- \( \text{iter\_usage} \)  A number indicating how many iteration has been used.
- \( \text{elapsed\_times} \)  A numerical vector indicating how much is consumed in each iteration.
- \( \text{elapsed\_time} \)  A number indicating how much time is consumed overall.
- \( \text{elapsed\_time\_per\_iter} \)  A number indicating how much time is consumed for each iteration in average.
- \( \text{iterates\_record} \)  A list as the records of parameter estimates of each iteration, only returned when \( \text{return\_iterates} = \text{TRUE} \).

**Author(s)**

Rui Zhou and Daniel P. Palomar

**References**


**See Also**

`fit\_AR1\_t`
Examples

```r
library(imputeFin)
data(ts_VAR_t)
fitted <- fit_VAR_t(Y = ts_VAR_t$Y, p = 2, parallel_max_cores = 2)
```

---

**impute_AR1_Gaussian**

*Impute missing values of time series based on a Gaussian AR(1) model*

**Description**

Impute inner missing values (excluding leading and trailing ones) of time series by drawing samples from the conditional distribution of the missing values given the observed data based on a Gaussian AR(1) model as estimated with the function `fit_AR1_Gaussian`. Outliers can be detected and removed.

**Usage**

```r
impute_AR1_Gaussian(
  y,
  n_samples = 1,
  random_walk = FALSE,
  zero_mean = FALSE,
  remove_outliers = FALSE,
  outlier_prob_th = 0.001,
  verbose = TRUE,
  return_estimates = FALSE,
  tol = 1e-10,
  maxiter = 100
)
```

**Arguments**

- `y` - Time series object coercible to either a numeric vector or numeric matrix (e.g., `zoo` or `xts`) with missing values denoted by `NA`.
- `n_samples` - Positive integer indicating the number of imputations (default is 1).
- `random_walk` - Logical value indicating if the time series is assumed to be a random walk so that `phi1 = 1` (default is `FALSE`).
- `zero_mean` - Logical value indicating if the time series is assumed zero-mean so that `phi0 = 0` (default is `FALSE`).
- `remove_outliers` - Logical value indicating whether to detect and remove outliers.
- `outlier_prob_th` - Threshold of probability of observation to declare an outlier (default is `1e-3`).

---

**Description**

Impute inner missing values (excluding leading and trailing ones) of time series by drawing samples from the conditional distribution of the missing values given the observed data based on a Gaussian AR(1) model as estimated with the function `fit_AR1_Gaussian`. Outliers can be detected and removed.

**Usage**

```r
impute_AR1_Gaussian(
  y,
  n_samples = 1,
  random_walk = FALSE,
  zero_mean = FALSE,
  remove_outliers = FALSE,
  outlier_prob_th = 0.001,
  verbose = TRUE,
  return_estimates = FALSE,
  tol = 1e-10,
  maxiter = 100
)
```

**Arguments**

- `y` - Time series object coercible to either a numeric vector or numeric matrix (e.g., `zoo` or `xts`) with missing values denoted by `NA`.
- `n_samples` - Positive integer indicating the number of imputations (default is 1).
- `random_walk` - Logical value indicating if the time series is assumed to be a random walk so that `phi1 = 1` (default is `FALSE`).
- `zero_mean` - Logical value indicating if the time series is assumed zero-mean so that `phi0 = 0` (default is `FALSE`).
- `remove_outliers` - Logical value indicating whether to detect and remove outliers.
- `outlier_prob_th` - Threshold of probability of observation to declare an outlier (default is `1e-3`).
verbose Logical value indicating whether to output messages (default is TRUE).

return_estimates Logical value indicating if the estimates of the model parameters are to be returned (default is FALSE).

tol Positive number denoting the relative tolerance used as stopping criterion (default is 1e-8).

maxiter Positive integer indicating the maximum number of iterations allowed (default is 100).

Value

By default (i.e., for n_samples = 1 and return_estimates = FALSE), the function will return an imputed time series of the same class and dimensions as the argument y with one new attribute recording the locations of missing values (the function plot_imputed will make use of such information to indicate the imputed values), as well as locations of outliers removed.

If n_samples > 1, the function will return a list consisting of n_sample imputed time series with names: y_imputed.1, y_imputed.2, etc.

If return_estimates = TRUE, in addition to the imputed time series y_imputed, the function will return the estimated model parameters:

phi0 The estimate for phi0 (numeric scalar or vector depending on the number of time series).

phi1 The estimate for phi1 (numeric scalar or vector depending on the number of time series).

sigma2 The estimate for sigma2 (numeric scalar or vector depending on the number of time series).

Author(s)

Junyan Liu and Daniel P. Palomar

References


See Also

plot_imputed, fit_AR1_Gaussian, impute_AR1_t
Examples

```r
library(imputeFin)
data(ts_AR1_Gaussian)
y_missing <- ts_AR1_Gaussian$y_missing
y_imputed <- impute_AR1_Gaussian(y_missing)
plot_imputed(y_imputed)
```

---

**impute_AR1_t**

**Impute missing values of time series based on a Student’s t AR(1) model**

**Description**

Impute inner missing values (excluding leading and trailing ones) of time series by drawing samples from the conditional distribution of the missing values given the observed data based on a Student’s t AR(1) model as estimated with the function `fit_AR1_t`. Outliers can be detected and removed.

**Usage**

```r
impute_AR1_t(
  y, 
  n_samples = 1, 
  random_walk = FALSE, 
  zero_mean = FALSE, 
  fast_and_heuristic = TRUE, 
  remove_outliers = FALSE, 
  outlier_prob_th = 0.001, 
  verbose = TRUE, 
  return_estimates = FALSE, 
  tol = 1e-08, 
  maxiter = 100, 
  K = 30, 
  n_burn = 100, 
  n_thin = 50
)
```

**Arguments**

- **y**: Time series object coercible to either a numeric vector or numeric matrix (e.g., zoo or xts) with missing values denoted by NA.
- **n_samples**: Positive integer indicating the number of imputations (default is 1).
- **random_walk**: Logical value indicating if the time series is assumed to be a random walk so that \( \phi_1 = 1 \) (default is FALSE).
- **zero_mean**: Logical value indicating if the time series is assumed zero-mean so that \( \phi_0 = 0 \) (default is FALSE).
**impute_AR1_t**

- **fast_and_heuristic**: Logical value indicating whether a heuristic but fast method is to be used to estimate the parameters of the Student’s t AR(1) model (default is `TRUE`).
- **remove_outliers**: Logical value indicating whether to detect and remove outliers.
- **outlier_prob_th**: Threshold of probability of observation to declare an outlier (default is `1e-3`).
- **verbose**: Logical value indicating whether to output messages (default is `TRUE`).
- **return_estimates**: Logical value indicating if the estimates of the model parameters are to be returned (default is `FALSE`).
- **tol**: Positive number denoting the relative tolerance used as stopping criterion (default is `1e-8`).
- **maxiter**: Positive integer indicating the maximum number of iterations allowed (default is `100`).
- **K**: Positive number controlling the values of the step sizes in the stochastic EM method (default is `30`).
- **n_burn**: Positive integer controlling the length of the burn-in period of the Gibbs sampling (default is `100`). The first (`n_burn * n_thin`) samples generated will be ignored.
- **n_thin**: Positive integer indicating the sampling period of the Gibbs sampling in the stochastic EM method (default is `1`). Every `n_thin`-th samples is used. This is aimed to reduce the dependence of the samples.

**Value**

By default (i.e., for `n_samples = 1` and `return_estimates = FALSE`), the function will return an imputed time series of the same class and dimensions as the argument `y` with one new attribute recording the locations of missing values (the function `plot_imputed` will make use of such information to indicate the imputed values), as well as locations of outliers removed.

If `n_samples > 1`, the function will return a list consisting of `n_sample` imputed time series with names: `y_imputed.1`, `y_imputed.2`, etc.

If `return_estimates = TRUE`, in addition to the imputed time series `y_imputed`, the function will return the estimated model parameters:

- **phi0**: The estimate for `phi0` (numeric scalar or vector depending on the number of time series).
- **phi1**: The estimate for `phi1` (numeric scalar or vector depending on the number of time series).
- **sigma2**: The estimate for `sigma2` (numeric scalar or vector depending on the number of time series).
- **nu**: The estimate for `nu` (numeric scalar or vector depending on the number of time series).

**Author(s)**

Junyan Liu and Daniel P. Palomar
**impute_OHLC**

Impute missing values of an OHLC time series on a rolling window basis based on a Gaussian AR(1) model.

**Description**

Impute inner missing values (excluding leading and trailing ones) of an OHLC time series on a rolling window basis. This is a wrapper of the functions `impute_AR1_Gaussian` and `impute_rolling_AR1_Gaussian`.

**Usage**

```r
impute_OHLC(
  y_OHLC,
  rolling_window = 252,
  remove_outliers = FALSE,
  outlier_prob_th = 0.001,
  tol = 1e-10,
  maxiter = 100
)
```

**Arguments**

- **y_OHLC**
  
  Time series object coercible to a numeric matrix (e.g., zoo or xts) with four columns denoting the prices Op, Hi, Lo, Cl.

- **rolling_window**
  
  Rolling window length (default is 252).

- **remove_outliers**
  
  Logical value indicating whether to detect and remove outliers.

- **outlier_prob_th**
  
  Threshold of probability of observation to declare an outlier (default is 1e-3).

**References**


**See Also**

`plot_imputed`, `fit_AR1_t`, `impute_AR1_Gaussian`
**impute_rolling_AR1_Gaussian**

Impute missing values of time series on a rolling window basis based on a Gaussian AR(1) model

**Description**

Impute inner missing values (excluding leading and trailing ones) of time series on a rolling window basis. This is a wrapper of the function `impute_AR1_Gaussian`.

**Usage**

```r
impute_rolling_AR1_Gaussian(
  y,
  rolling_window = 252,
  random_walk = FALSE,
  zero_mean = FALSE,
  remove_outliers = FALSE,
  outlier_prob_th = 0.001,
  tol = 1e-10,
  maxiter = 100
)
```

**Arguments**

- `y`  
  Time series object coercible to either a numeric vector or numeric matrix (e.g., zoo or xts) with missing values denoted by NA.

- `rolling_window`  
  Rolling window length (default is 252).

- `random_walk`  
  Logical value indicating if the time series is assumed to be a random walk so that \( \phi_1 = 1 \) (default is FALSE).

- `zero_mean`  
  Logical value indicating if the time series is assumed to have zero mean.

- `remove_outliers`  
  Logical value indicating if the time series is assumed to have no outliers.

- `outlier_prob_th`  
  Positive number denoting the relative tolerance used as stopping criterion (default is 1e-8).

- `maxiter`  
  Positive integer indicating the maximum number of iterations allowed (default is 100).

**Value**

Imputed OHLC prices.

**Author(s)**

Daniel P. Palomar

**See Also**

`impute_AR1_Gaussian`, `impute_rolling_AR1_Gaussian`
Plot single imputed time series (as returned by functions `impute_AR1_Gaussian` and `impute_AR1_t`), highlighting the imputed values in a different color.

### Usage

```r
code_snippet
```

### Description

Plot single imputed time series (as returned by functions `impute_AR1_Gaussian` and `impute_AR1_t`), highlighting the imputed values in a different color.

### Value

Same as `impute_AR1_Gaussian` for the case `n_samples = 1` and `return_estimates = FALSE`.
Arguments

- **y_imputed**: Imputed time series (can be any object coercible to a numeric vector or a numeric matrix). If it has the attribute "index_miss" (as returned by any of the imputation functions `impute_AR1_Gaussian` and `impute_AR1_t`), then it will highlight the imputed values in a different color.

- **column**: Positive integer indicating the column index to be plotted (only valid if the argument `y_imputed` is coercible to a matrix with more than one column). Default is 1.

- **title**: Title of the plot (default is "Imputed time series").

- **color_imputed**: Color for the imputed values (default is "red").

- **type**: Type of plot. Valid options: "ggplot2" and "simple". Default is "ggplot2" (the package ggplot2 must be installed).

Author(s)

Daniel P. Palomar

Examples

```r
library(imputeFin)
data(ts_AR1_t)
y_missing <- ts_AR1_t$y_missing
y_imputed <- impute_AR1_t(y_missing)
plot_imputed(y_missing, title = "Original time series with missing values")
plot_imputed(y_imputed)
```

---

**ts_AR1_Gaussian**  
*Synthetic AR(1) Gaussian time series with missing values*

Description

Synthetic AR(1) Gaussian time series with missing values for estimation and imputation testing purposes.

Usage

```r
data(ts_AR1_Gaussian)
```

Format

List with the following elements:

- **y_missing**: 300 x 3 zoo object with three AR(1) Gaussian time series along the columns: the first column contains a time series with 10% consecutive missing values; the second column contains a time series with 10% missing values randomly distributed; and the third column contains the union of the previous missing values.
**ts_AR1_t**

Value of \( \phi_0 \) used to generate the time series.

Value of \( \phi_1 \) used to generate the time series.

Value of \( \sigma_2 \) used to generate the time series.

---

**ts_AR1_t**

*Synthetic AR(1) Student’s t time series with missing values*

---

**Description**

Synthetic AR(1) Student’s t time series with missing values for estimation and imputation testing purposes.

**Usage**

data(ts_AR1_t)

**Format**

List with the following elements:

- **y_missing** 300 x 3 zoo object with three AR(1) Student’s t time series along the columns: the first column contains a time series with 10% consecutive missing values; the second column contains a time series with 10% missing values randomly distributed; and the third column contains the union of the previous missing values.

- **phi0** Value of \( \phi_0 \) used to generate the time series.

- **phi1** Value of \( \phi_1 \) used to generate the time series.

- **sigma2** Value of \( \sigma_2 \) used to generate the time series.

- **nu** Value of \( \nu \) used to generate the time series.

---

**ts_VAR_t**

*Synthetic Student’s t VAR data with missing values*

---

**Description**

Synthetic Student’s t VAR data with missing values for estimation and imputation testing purposes.

**Usage**

data(ts_VAR_t)
Format

List with the following elements:

- **Y**: 200 x 3 zoo object as a Student’s t VAR time series.
- **phi0**: True value of the constant vector in the VAR model.
- **Phii**: True value of the coefficient matrix in the VAR model.
- **scatter**: True value of the scatter matrix (of the noise distribution) in the VAR model.
- **nu**: True value of the degrees of freedom (of the noise distribution) in the VAR model.
Index

* dataset
  ts_AR1_Gaussian, 18
  ts_AR1_t, 19
  ts_VAR_t, 19
  fit_AR1_Gaussian, 2, 3, 8, 11, 12
  fit_AR1_t, 2, 3, 6, 10, 13, 15
  fit_VAR_t, 8, 8
  impute_AR1_Gaussian, 2, 5, 11, 15–18
  impute_AR1_t, 2, 8, 12, 13, 17, 18
  impute_OHLC, 15
  impute_rolling_AR1_Gaussian, 15, 16, 16
  imputeFin-package, 2
  plot_imputed, 2, 12, 14, 15, 17, 17
  ts_AR1_Gaussian, 3, 18
  ts_AR1_t, 3, 19
  ts_VAR_t, 19