

Package: kssa (via r-universe)

August 20, 2024

Title Known Sub-Sequence Algorithm

Version 0.0.1

Maintainer Iván Felipe Benavides <pipeben@gmail.com>

Description Implements the Known Sub-Sequence Algorithm <doi:10.1016/j.aaf.2021.12.013>, which helps to automatically identify and validate the best method for missing data imputation in a time series. Supports the comparison of multiple state-of-the-art algorithms.

License AGPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

URL <https://github.com/pipeben/kssa>

BugReports <https://github.com/pipeben/kssa/issues>

Depends R (>= 4.0)

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports magrittr, ggplot2, rlang, methods, forecast, imputeTS, stats, zoo, Metrics, dplyr, missMethods

Date 2022-06-14

Repository <https://pipeben.r-universe.dev>

RemoteUrl <https://github.com/pipeben/kssa>

RemoteRef HEAD

RemoteSha 774a8a6b91d1da6fc1b9bd8bd668bf0f05552f3d

Contents

get_imputations	2
kssa	3
kssa_plot	5

get_imputations	<i>get_imputations function</i>
-----------------	---------------------------------

Description

Function to get imputations from methods compared by kssa

Usage

```
get_imputations(x_ts, methods = "all", seed = 1234)
```

Arguments

x_ts	A ts object with missing data to be imputed
methods	A string or string vector indicating the method or methods
seed	Numeric. Any number

Value

A list of imputed time series with the selected methods

Examples

```
# Get imputed values for airgap_na_ts with the methods of
# Create 20% random missing data in tsAirgapComplete time series from imputeTS
set.seed(1234)
library("imputeTS")
library("kssa")
airgap_na <- missMethods::delete_MCAR(as.data.frame(tsAirgapComplete), 0.2)

# Convert co2_na to time series object
airgap_na_ts <- ts(airgap_na, start = c(1959, 1), end = c(1997, 12), frequency = 12)

my_imputations <- get_imputations(airgap_na_ts, methods = "all")

# my_imputations contains the imputed time series with all methods.
# Access it and choose the one from the best method for your purposes

my_imputations$seadec
plot.ts(my_imputations$seadec)
```

kssa

*kssa Algorithm***Description**

Run the Known Sub-Sequence Algorithm to compare the performance of imputation methods on a time series of interest

Usage

```
kssa(
  x_ts,
  start_methods,
  actual_methods,
  segments = 5,
  iterations = 10,
  percentmd = 0.2,
  seed = 1234
)
```

Arguments

<code>x_ts</code>	Time series object <code>ts</code> containing missing data (NA)
<code>start_methods</code>	String vector. The method or methods to start the algorithm. Same as for <code>actual_methods</code>
<code>actual_methods</code>	The imputation methods to be compared and validated. It can be a string vector containing the following 11 imputation methods: <ul style="list-style-type: none"> • "all" - compare among all methods automatically - Default • "auto.arima" - State space representation of an ARIMA model • "StructTS" - State space representation of a structural model • "seadec" - Seasonal decomposition with Kalman smoothing • "linear_i" - Linear interpolation • "spline_i" - Spline interpolation • "stine_i" - Stineman interpolation • "simple_ma" - Simple moving average • "linear_ma" - Linear moving average • "exponential_ma" - Exponential moving average • "locf" - Last observation carried forward • "stl" - Seasonal and trend decomposition with Loess For further details on these imputation methods please check packages imputeTS and forecast
<code>segments</code>	Integer. Into how many segments the time series will be divided
<code>iterations</code>	Integer. How many iterations to run

percentmd	Numeric. Percentage of missing data. Must match with the true percentage of missing data in x_ts
seed	Numeric. Random seed to choose

Value

A list of results to be plotted with function [kssa_plot](#) for easy interpretation

References

Benavides, I. F., Santacruz, M., Romero-Leiton, J. P., Barreto, C., & Selvaraj, J. J. (2022). Assessing methods for multiple imputation of systematic missing data in marine fisheries time series with a new validation algorithm. *Aquaculture and Fisheries*. [Full text publication](#).

Examples

```
# Create 20% random missing data in tsAirgapComplete time series from imputeTS
set.seed(1234)
library("kssa")
library("imputeTS")
airgap_na <- missMethods::delete_MCAR(as.data.frame(tsAirgapComplete), 0.2)

# Convert co2_na to time series object
airgap_na_ts <- ts(airgap_na, start = c(1959, 1), end = c(1997, 12), frequency = 12)

# Apply the kssa algorithm with 5 segments,
# 10 iterations, 20% of missing data, and
# compare among all available methods in the package.
# Remember that percentmd must match with
# the real percentage of missing data in the
# input co2_na_ts time series

results_kssa <- kssa(airgap_na_ts,
  start_methods = "all",
  actual_methods = "all",
  segments = 5,
  iterations = 10,
  percentmd = 0.2
)

# Print and check results
results_kssa

# For an easy interpretation of kssa results
# please use function kssa_plot
```

kssa_plot	<i>kssa_plot function</i>
-----------	---------------------------

Description

Function to plot the results of kssa for easy interpretation

Usage

```
kssa_plot(results, type, metric)
```

Arguments

results	An object with results produced with function kssa
type	A character value with the type of plot to show. It can be "summary" or "complete".
metric	A character with the performance metric to be plotted. It can be "rmse", "mase", "cor", or "smape" <ul style="list-style-type: none">• "rmse" - Root Mean Squared Error (default choice)• "mase" - Mean Absolute Scaled Error• "smape" - Symmetric Mean Absolute Percentage Error• "cor" - Pearson correlation coefficient

For further details on these metrics please check package Metrics

Value

A plot of kssa results in which imputation methods are ordered from lower to higher (left to right) error.

Examples

```
# Plot the results obtained in the example from function kssa

# Create 20% random missing data in tsAirgapComplete time series from imputeTS
set.seed(1234)
library("kssa")
library("imputeTS")
airgap_na <- missMethods::delete_MCAR(as.data.frame(tsAirgapComplete), 0.2)

# Convert co2_na to time series object
airgap_na_ts <- ts(airgap_na, start = c(1959, 1), end = c(1997, 12), frequency = 12)

# Apply the kssa algorithm with 5 segments,
# 10 iterations, 20% of missing data, and
# compare among all available methods in the package.
# Remember that percentmd must match with
# the real percentage of missing data in the
```

```
# input co2_na_ts time series

results_kssa <- kssa(airgap_na_ts,
  start_methods = "all",
  actual_methods = "all",
  segments = 5,
  iterations = 10,
  percentmd = 0.2
)

kssa_plot(results_kssa, type = "complete", metric = "rmse")

# Conclusion: Since kssa_plot is ordered from lower to
# higher error (left to right), method 'linear_i' is the best to
# impute missing data in airgap_na_ts. Notice that method 'locf' is the worst

# To obtain imputations with the best method, or any method of preference
# please use function get_imputations
```

Index

forecast, [3](#)

get_imputations, [2](#)

imputeTS, [3](#)

kssa, [3](#), [5](#)

kssa_plot, [4](#), [5](#)

ts, [3](#)