

# KarsTS 1.0

An interface for karstic time series

User's guide



## The KarsTS package

Author, Creator and Maintainer: Marina Sáez Andreu

Thesis advisors: David Benavente García and Soledad Cuezva Robleño

Contributor: Concepción Pla Bru

Date: May, 2017

License: GPL ( $\geq 2$ )

Citation:

Marina Saez (2017). KarsTS: An interface for karstic time series. R package version 1.0.

If you have any problem, email the maintainer ([marinasaez\\_andreu@hotmail.com](mailto:marinasaez_andreu@hotmail.com)). Suggestions for the next version are welcome too. I hope KarsTS helps you with your research. KarsTS took a great effort so, please, do not forget to cite it.

I would like to express my gratitude to my thesis advisors for their support and to my dear colleague C. Pla for proofreading this user's guide.

# **Contents**

1.. Introduction.....	6
2.. Getting started .....	6
3.. Note on NAs ad NaNs .....	6
4.. Data sets.....	7
5.. Parts of the screen .....	7
6.. Menus.....	8
6.1.. Times Series.....	8
Load .....	8
Save .....	8
Remove.....	8
Export .....	8
Rename .....	8
Merge .....	8
List .....	9
Sampling periods.....	9
Cut and resampling .....	9
Aggregate .....	9
Operations.....	10
Round .....	10
Scale .....	10
Difference.....	10
Cumulative sum.....	11
6.2.. Gap Sets.....	11
Load, Save, Remove, Export, Rename, Merge and List.....	11
Gap selection.....	11
Artificial Random Gaps, Artificial Random Gaps and Apply Gaps to Time Series .....	11
Little's MCAR test .....	11
6.3.. Analysis.....	12
Statistics .....	12
Rolling statistics.....	12
Loess decomp. (Loess decomposition).....	12

Loess smooth. (Loess smoothing) .....	13
Normality.....	13
Stationarity .....	13
Linearity.....	13
PCA .....	14
Invariants.....	14
Recurrence matrix .....	15
Cross recurrence matrix .....	15
Joint recurrence matrix .....	15
Determinism.....	15
Laminarity.....	16
6.4.. Plots.....	16
Plot ts.....	16
Remove points.....	17
Get coordinates.....	17
Linear correlation .....	17
Mutual information .....	17
Wind rose .....	17
Histogram .....	18
Phase portraits .....	18
False Nearest Neighbors .....	18
Ed(1) and Ed (2) .....	18
Recurrence plot, cross recurrence and joint recurrence plot. ....	19
6.5.. Filling .....	19
Interpolations.....	19
ARIMA.....	19
Mean Value .....	20
Multiv. Splines (Multivariate splines).....	21
MissForest .....	21
Twins .....	21
ARIMAX .....	22
Check filling .....	22
7.. References.....	22
Appendix I: Functionalities.....	24

Appendix II: Input files format .....	28
Time series: .....	28
.R or .RData. files .....	28
.csv files .....	28
txt files.....	28
Gap sets.....	29
.R or .RData files .....	29
.csv files .....	29
.txt files.....	29
.R o .RData files .....	29
csv files .....	30
txt files.....	30
Loading frequent problems.....	30
Appendix IV: Filling methods overview .....	32
Univariate.....	32
Interpolations.....	32
ARIMA.....	32
Mean value.....	32
Multivariate.....	32
Missing Random Algorithm .....	32
Twins .....	33

## **1.. Introduction**

KarsTS is an R-based interface designed to study karstic time series, although it can be used in other fields. Karstic research typically involves microclimatic time series and gas concentrations ( $\text{CO}_2$ ,  $^{222}\text{Rn}$ ,  $\text{CH}_4$  ...). Many of these time series have a strong non-linear behavior. For this reason, KarsTS gathers a number of linear and non-linear methods in a friendly-user interface: false nearest neighbors, mutual information, phase portraits etc. KarsTS can build and plot recurrence matrices from long time series (thousands of measurements).

Frequently, time series contain gaps because caves are aggressive environments and monitoring systems can suffer technical problems. As a result, the length of these gaps may become significant. KarsTS provides a number of univariate and multivariate filling methods and tools to handle gap sets. For example, it is possible to select a subset of gaps and apply a filling method to it.

A complete list of KarsTS functionalities can be found [here](#).

KarsTS can import and export txt and csv files. The user can access KarsTS data sets from the R console; however, we do not recommend it. KarsTS is intended to be autonomous in order that it is not necessary to use another program at the same time.

## **2.. Getting started**

KarsTS runs both on Windows and Mac.

First usage:

1. If you work on Mac, you may need to update XQuartz.
2. Install R on your computer if it is not already installed.
3. Open R and install the package KarsTS
4. Run the following lines on R command window

```
library(KarsTS) #loads the package
KarsTS() #launches the interface
```

Successive usages: go straight to the third point.

## **3.. Note on NAs ad NaNs**

In R, and, consequently, in KarsTS, missing values are represented as NA (not available). NA is different from NaN (not a number). For example,  $\log_{10}(-1)$  is not a number, whilst a temperature missing observation is a number, even if it is unknown.

## 4.. Data sets

There are three types of data sets in KarSTS

- ❖ Time series
- ❖ Sets of gaps
- ❖ Recurrence matrices

See [Appendix II: Input files format](#).

## 5.. Parts of the screen

The main screen has five menubuttons, which are located in the upper part of the screen.

1. Time Series: tools to manipulate time series
2. Gap Sets: tools to manipulate gap sets
3. Analysis: tools to analyze time series and create recurrence matrices
4. Plots: data visualization and graphic outputs
5. Filling: filling methods

Below the buttons, there is an input panel (at left), an output window (center) and a plot window (right). The output window is editable, that is, the user can write, erase, copy and paste in it. The graphics can also be plotted directly on a pop-up window. In the lower part of the screen four additional buttons allow to: change the working directory, save the outputs to a txt file, get information about the program and open a help file.

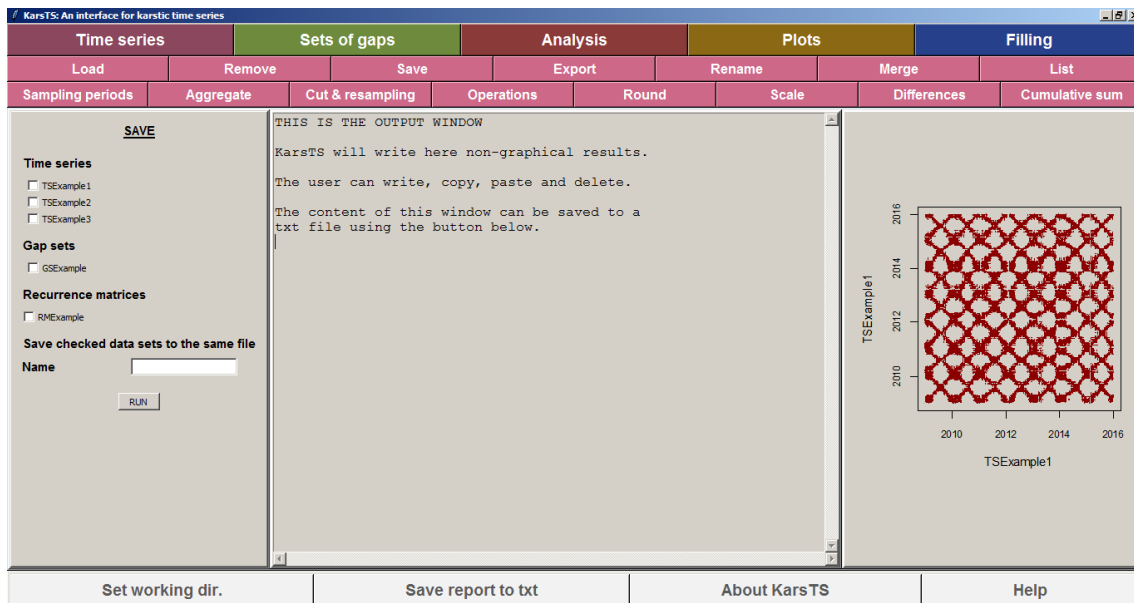


Figure 1: KarSTS main screen

## **6.. Menus**

### **6.1.. Times Series**

#### **Load**

The Load button is used to load time series, sets of gaps and recurrence matrices. KarSTS accepts R, RData, csv and txt file types. The R and RData files can contain more than one data set.

The input panel has two buttons. The upper one loads R files whereas the lower one imports txt and csv files.

In order to load txt or csv files, it is necessary to specify the date format. It is convenient to specify the separator, although it defaults to tabulation for txt files and comma for csv files. Note that csv files can use semicolons as separators, depending on the computer language options (the computer where the file was created).

For more details, see [Appendix II: Input files format](#) and [Loading frequent problems](#).

#### **Save**

Data sets can be saved in separated .R files to the user's working directory. The files are named following the data sets they store. It is possible to save several data sets to a single file too.

#### **Remove**

Data sets will be removed from the environment, not from the user's hard disk.

#### **Export**

KarSTS can export data sets to .csv or .txt files. Each gap set will be stored in a separated file, named after the data set; even though, multiple time series can be saved to a single file.

#### **Rename**

Rename changes the name of a data set. The original data set will not be removed.

#### **Merge**

Only time series and gap sets can be merged.

Time series must fulfill the following conditions in order to be merged:

- They cannot overlay (otherwise, the same time could have two different simultaneous values)
- All of them must have the same sampling period

If there are time gaps between the time series to merge, KarSTS will fill them with NAs.

The requirements for gap sets are the following:

- They must come from time series with the same sampling period
- They must come from time series with the same starting date



KarsTS verifies the times series or gap sets compatibility before merging them.

## List

This button displays a list and a short description of the loaded data sets on the Output Window.

## Sampling periods

This function analyzes the sampling periods and existing gaps in a time series.

When a time series with various sampling periods is imported, KarsTS introduces as many NAs as necessary in order that the whole time series has the minimum sampling period. For the same reason, KarsTS also introduces NAs to fill the time gaps. As a consequence, once the time series is loaded, there are NAs coming from both true missing values and the resampling. The latter result in many gaps with the same length. The user must select the true sampling periods from a checkbox.

The final output of this button is a table in the Output Window. Each row corresponds to a piece of time series that has a unique sampling period and contains either actual values or missing values. The columns of the table correspond to:

- Sampling period (in minutes)
- Initial date
- Final date
- Number of measurements
- Actual values or NAs

## Cut and resampling

This function has two main purposes:

- Creating a time series that is a section of another one
- Creating a time series with a different sampling period. For example, if the sampling period of the input time series is equal to 5 minutes and the resampling is 2, the sampling period of the resulting time series will be equal to 10 minutes. Unlike the Aggregate button, it will not average the measurements.

### Inputs:

- Sampling period: defaults to 1 (the same sampling period)
- Initial date: defaults to the first date in the time series
- Final date: defaults to the last date in the time series

## Aggregate

This button builds a time series of grouped values, using one of these functions: mean, median, minimum, maximum, standard deviation or sum.

### Inputs:

- Time series

- Period (in lags)
- Statistic: mean, median, minimum, maximum, standard deviation or sum.
- NA treatment: see the examples below
- Initial date: see the examples below

### Examples:

Consider a time series with an observation every 30 minutes (sampling period).

- 1) We want to calculate a time series of minimum daily values. The period is equal to 48 (number of measurements in one day) and we select the minimum. Some days might have missing values (NAs). If the option “propagate NAs” is chosen, the days containing some missing value are assigned NAs. On the contrary, with the option “ignore NAs”, the minimum is calculated using the available measurements and a day will be assigned a NA only in case all its measurements are missing. Finally, the starting date of the time series can be also changed. For instance, if the original time series starts at 2009-12-25 00:03:00, we can start at 2009-12-25 00:00:00 for the sake of simplicity.
- 2) We want to shorten the time series in order to save computing time. Thus, we choose a period of 4 and the mean as statistic. We will obtain a time series with 12 values per day, instead of 48. Note that each value is the mean of four measurements. To resample without averaging, use the Cut and resampling button instead.

## Operations

This button is used to perform operations with one or more time series: sum, multiplication, reciprocal, opposite and natural logarithm.

The result of adding or multiplying two or more time series is a single time series, the name of which must be entered by the user. The rest of operations are applied separately to each selected time series and the output time series are assigned default names.

## Round

KarsTS does not automatically round the output time series when a calculation is performed. The input panel shows two rounding options: number of significant digits or number of decimal places. When both options are entered, only the number of significant digits is used.

## Scale

This button scales one or more time series at the same time. There are two options, depending on the parameters used to center and scale the variable: (i) the median and the median absolute deviation (robust scaling) or (ii) the mean and the standard deviation.

## Difference

This button differences one or more time series at the selected lag. The option “Center the time series” interpolates the differenced time series in order to keep the length of the original time series.

## Cumulative sum

It is used to calculate the cumulative sum of one or more time series. As in the difference button, it is possible to center the output time series.

## 6.2.. Gap Sets

### Load, Save, Remove, Export, Rename, Merge and List

See [Times Series](#)

### Gap selection

This button extracts a gap set from the gaps of a time series.

There are four selection criteria: all, minimum size, maximum size and specific gaps. The minimum and maximum sizes are measured in lags. When the Selection button is selected, a list of gaps appears on the output window to enable the selection of specific gaps.

When the user chooses various criteria, they are combined. The next possibilities:

- The criterion “all” annuls the rest.
- When a maximum and a minimum size are entered, only the gaps that meet simultaneously both conditions will be part of the gap set.
- When both a size criterion and specific gaps are selected, the latter will be added to the gap set regardless of their size.

### Artificial Random Gaps, Artificial Random Gaps and Apply Gaps to Time Series

In order to verify the goodness of a filling method on a time series, we recommend creating an artificial set of gaps.

The Artificial Random Gaps button creates a random set of gaps that will not overlay the existing gaps in the time series. The user can choose the number of gaps and their size.

The Artificial Specific Gaps button generates a gap between two dates chosen by the user.

When a set of artificial gaps is created, a new time series containing those gaps is created too. In order to apply the gap set to another time series, use the Apply Gaps to Time Series button.

### Little’s MCAR test

This button performs the MCAR Little’s test. The test null hypothesis is that the values are missing completely at random. KarSTS searches for patterns in the multivariate time series.

Inputs: two time series, at least.

Outputs:

- A table containing the number and percentage of missing values.
- The patterns detected and the mean values of the variables within each pattern.
- Which gaps belong to each pattern.

## 6.3.. Analysis

### Statistics

It displays a table in the Output Window that contains the univariate descriptive statistics of one or more time series.

### Rolling statistics

This button calculates the time series statistics in sliding windows. It returns as many time series as statistics were selected by the user. If the median or the mean are selected, the result is a smoothed time series, which has the same length as the original one.

#### Inputs

- Time series
- Sliding window size: in lags
- Statistics: check one or more
- Estimate tails: the sliding window is centered, which implies that the firsts/lasts values in the time series are not surrounded by a complete, symmetric window. For example, let the window length be 48. The first value in the time series has no values at left, although it has 24 values at right. If we do not estimate tails, the first value in the time series will be NA; otherwise, the first value in the time series will be the median of the first 25 values (24 at right plus the value itself). If we estimate the tails, the function will take longer to run.

### Loess decomp. (Loess decomposition)

It is used to decompose a time series using the loess seasonal decomposition. By default, the time series is decomposed in three components: trend, seasonal component and remainder. They bear the name of the original time series, plus the suffixes Tr, Sea and Rem, respectively.

At each point, the trend is the result of averaging the values in a window around the point (trend window). A larger window yields a smoother trend. In the same way, the seasonal component is calculated in a window (the seasonal window). A large seasonal window results in a seasonal component whose amplitude changes slowly. Optionally, more windows of different sizes can be used to decompose the trend into two or more time series.

#### Inputs:

- Seasonality: number of measurements in one period
- Seasonal window: number of measurements in the seasonal window
- Trend window: number of measurements in the trend window
- Trend decomposition: it allows decomposing the trend into up to seven time series, although usually only two or three will be used. The window size and the name for each output time series are requested. The larger window should be in the uppermost box.

## Loess smooth. (Loess smoothing)

Unlike the loess seasonal decomposition, the loess smoothing does not require a periodical phenomenon. The smoothing is the result of locally fitting a linear model in a window around a point. The model predicts the value of the corresponding central point. By default, the model relates the time series values and the time series times. Moreover, the user can include other variables (predictor time series) in it.

### Inputs:

- Time series
- Alpha: the parameter alpha controls the size of the sliding window. Basically, alpha is the proportion of points in the time series that will be used in each window; therefore, a higher alpha value yields a smoother time series.
- Predictor time series: optional

## Normality

It is used to run a number of stationarity tests. When the user selects more than one time series, KarsTS performs multivariate as well as univariate tests. If the time series has NAs, KarsTS removes them.

Table 1: Normality tests

Test	Type
Cramer von Mises	Univariate
Lilliefors (KS)	Univariate
Anderson-Darling	Univariate
Henze-Zirkler	Multivariate
Mardia (for skewness and kurtosis)	Multivariate

## Stationarity

It performs a number of stationarity tests. If the time series has NAs, KarsTS will remove them, thus affecting the time correlations; therefore, it is convenient to fill the time series before running the tests.

The next table summarizes the null hypotheses:

Table 2: Stationarity tests

Test	H <sub>0</sub>
Philipp-Perron unit root test	It has a unit root (The detrended time series is not stationary)
Box-Ljung test	Independence (There are no self-correlations)
Augmented Dickey-Fuller test (lag=1)	It has a unit root (The detrended time series is not stationary)
KPSS test for Level Stationarity	Level stationarity (The time series is white noise)

## Linearity

This button performs a number of linearity tests. It does not admit missing values.

In case the time series is not seasonal, the delay for the surrogates test must be 1.

The tests and their null hypotheses are the following:

**Table 3: Linearity tests**

Test	Null hypothesis
<b>Surrogates test</b>	Data comes from a linear stochastic process
<b>Teraesvirta's neural network test</b>	Linearity in mean
<b>White neural network test</b>	Linearity in mean
<b>Keenan's one-degree test for nonlinearity</b>	The time series follows some AR process
<b>McLeod-Li test</b>	The time series follows some ARIMA process
<b>Tsay's Test for nonlinearity</b>	The time series follows some AR process
<b>Likelihood ratio test for threshold nonlinearity</b>	The time series follows some AR process (the alternative hypothesis is that it follows a TAR process)

## PCA

This button performs a principal component analysis. The calculation is done by singular value decomposition. The user has to choose the time series and a name for the output components. These will appear as new time series. In addition, KarSTs writes on the output window the variance explained and the loadings.

## Invariants

In this context, the invariants are properties of a time series that do not change once the system has been unfolded completely. Because of that, the invariants can be used to estimate the parameters for a correct embedding, specifically the radius and the embedding dimension. Therefore, in general, the user analyzes the plots and estimates at which radius and embedding dimension the curve becomes horizontal.

As the invariants are calculated for different radii and embedding dimensions, the user has to enter the minimum and maximum embedding dimensions as well as the minimum and maximum radii. Moreover, the user must enter the number of radii at which the invariants will be estimated. For example, for 6 radii ranging from 50 to 50.5, the invariants will be calculated at 50, 50.1, 50.2, 50.3, 50.4, 50.5 and 50.6.

- **Correlation sum and dimension:**  
The correlation dimension measures the fractal dimension of a geometrical object embedded in a phase space. The correlation sum can be interpreted as the probability of finding a neighbor in a ball of radius  $r$  in the phase space.  
The function returns two plots in a single panel. The upper one represents the correlation sum for different radii and embedding dimensions. The user must check that there is a region in this plot where the curves for different embedding dimensions are parallel; otherwise, the estimation of the correlation dimension must be discarded.
- **KS Entropy:**  
It is a measure of complexity; specifically, smaller values of KS Entropy indicate more self-similarity or less noise.  
This option plots KS entropy curves calculated with different embedding dimensions and radii. The radius values are in the X axis.

## Recurrence matrix

Since the calculation of a recurrence matrix can take some time for large time series, it is convenient to save it as soon as it is created.

The function requires some inputs. The following table shows what buttons can be used to estimate these inputs:

Table 4: Tools for getting the recurrence matrices parameters

Inputs	Useful buttons
Embedding dimension	False Nearest Neighbors, Invariants
Delay	Mutual information
Tolerance	Invariants
Theiler window	Mutual information

## Cross recurrence matrix

Everything said about the simple recurrence matrices in the above paragraph applies to the cross ones. However, there are some differences that the user needs to consider.

Whilst a (simple) recurrence matrix compares a time series with itself, a cross recurrence matrix compares two similar variables; therefore, it is necessary to choose two time series. The first one will be assigned to the X-axis and the second one to the Y-axis. KarSTs does not check whether the selected time series are appropriate to create a cross recurrence matrix. If the time series do not reflect similar phenomena, the user should consider a joint recurrence matrix instead.

Unlike a simple recurrence matrix, a cross recurrence matrix does not need to be symmetric. Thus, for the same matrix size, it takes significantly more time to calculate and more memory store them.

## Joint recurrence matrix

A joint recurrence matrix is the result of multiplying element-by-element two or more simple matrices. The Joint recurrence matrix button requires two simple recurrence matrices and the name for the joint one. In order to create a joint recurrence matrix using more than two simple matrices, use the Joint Recurrence Matrix button repeatedly.

## Determinism

The determinism regards the predictability and the degree of periodicity of a time series; it is related to the diagonal lines in the recurrence plot. Although the output is the same for both simple and joint recurrence matrices, its meaning is not exactly the same.

The input is a recurrence matrix. When the recurrence matrix has many ones, this function may take some time to run (the user can check the number of ones with the List button).

In order to reduce tangential movement, the user can enter a minimum length for the diagonal lines to be considered true diagonal lines (the default minimum is 2).

The result is a list of properties in the Output Window.

- Recurrence rate: number of ones/total number of points in the recurrence matrix.
- Determinism: number of points belonging to diagonal lines/number of ones.
- Ratio: points belonging to diagonal lines/points in the recurrence matrix.
- Summary of the lengths of the diagonal lines. The length of the diagonal lines indicates the predictability of the time series. For example, we consider the mean length is 30 points and the sampling period is 30 minutes; it means that, as an average, the series can be predicted 30 measurements (900 minutes) forward.

## Laminarity

In the same way the determinism is related to the diagonals, the laminarity is related to the vertical lines in the recurrence plot. The Laminarity button is analogous to the Determinism button and its outputs are:

1. Recurrence rate: number of ones/total number of points in the recurrence matrix.
2. Laminarity: number of points belonging to vertical lines/number of ones
3. Ratio: points belonging to vertical lines / points in the recurrence matrix.
4. Summary of the lengths of the vertical lines. The length of the vertical lines indicates how long the system remains trapped in the same state. Let say the mean length is 30 points and the sampling period is 30 minutes; it means that, as an average, the system changes its state in 30 points (900 minutes). For example, the state could be the cave being charged or discharged.

## 6.4.. Plots

### Plots

This button plots one or more time series. To draw another plot, press the Plot ts button in order to remove the previous choices.

The user can choose to plot points, lines or both. When plotting only lines, consider that the points surrounded by NAs will not be plotted. For example, if the time series is as follows:

3, NA, 6.7, NA, 44, NA, 3.4, NA, 87, NA, 4.5, NA...,

the graphic will be empty.

The user can order the time series; the ones with lower ranks will be plotted over the others (the default order is the list order).

The plot appears on the main panel (central row, right side) when the PLOT button is used, whereas ZOOM creates a new window. It is not necessary to press PLOT before pressing ZOOM.

When the plot is drawn to a new window, the overlaying curve can be selected in order to zoom a fragment (although, all the time series will be plotted).



## Remove points

This button allows selecting some points on a plot and turning those points into NAs in the corresponding time series. Thus, for example, the user may clean the time series from outliers.

The process takes two steps:

- Mark the points in the plot.  
Select the time series and press “Plot”. Select the points by dragging the mouse over them (the selected points will become red). The points selected will be those within a maximum radius from the cursor (threshold). If the selection of the points is difficult, increase the threshold.
- Create the new time series  
Press “Run” to create the new time series with reddened points turned into NAs.

## Get coordinates

The user selects, one by one, a set of points in a time series. Then, when the button *Remove points* is pressed, the function returns their position, time and value in a table on the Output Window.

First, select the time series and press “Plot”. Select the points on the plot, they will turn green. When the user presses the button “Run”, a table with the required information will appear in the Output Window.

## Linear correlation

It plots the correlation functions. To calculate the autocorrelation function (acf) and the partial autocorrelation function (pacf), the user must select one time series; whereas, the button returns the cross-correlation plot when two time series are selected. NAs are not allowed; therefore, the time series requires a preliminary filling. The correlation will be calculated at delays up to the maximum number of lags, which is one of the inputs.

## Mutual information

It is used to plot the average mutual information (AMI). The AMI is a generalized correlation, therefore, it can detect non-linear correlations. As in the linear correlation case, the user can choose one or two time series. NAs are not allowed; therefore the time series requires a preliminary filling. The AMI will be calculated at delays up to the maximum number of lags chosen by the user.

## Wind rose

It plots a wind rose from directional data. It is necessary to enter the time series that provides the magnitude (for example, wind speed) and the time series that provides the directions (for example, wind direction). The magnitudes must be non-negative and the directions must range from 0 to 360. Optionally, the user can enter the number of bins and the number of ticks.

## Histogram

It is used to plot a histogram. The user can choose the approximate number of bars, otherwise the Friedman-Diaconis formula will be used to estimate it.

## Phase portraits

It is used to plot time series in the phase space. In other words, the time does not appear in any axis, therefore it makes scatterplots of the time series values. The plots can have 2 or 3 dimensions. The time series can be embedded; in order to not embed, set the embedding dimension to 1 and the delay to 0.

Table 5: Phase portraits

Number of time series	Embedding dimension of the time series	Plot dimension
2	1,1	2
3	1,1,1	3
1	2	2
1	3	3
2	1,2	3
2	2,1	3

To use differential embedding, the differenced time series must be calculated previously and treated as independent time series (see the buttons Loess. Smooth and Differences).

## False Nearest Neighbors

The false nearest neighbors plot can be used to estimate the embedding dimension of a time series. An insufficient embedding dimension produces false nearest neighbors; therefore, when the phase space is completely unfolded, the number of neighbors remains constant. Consider that noise can increase the embedding dimension unnecessarily, so it might be convenient to smooth the time series.

The user must provide the following inputs:

- Maximum embedding dimension: the embedding dimension up to which the false nearest neighbors will be calculated. Usually, a maximum embedding dimension of 10 is enough.
- Delay: it depends on the periodicities in the time series.
- Theiler window: it depends on the periodicities in the time series.

Optionally, the user can enter the escape factor and the neighborhood parameter; otherwise the default values will be used.

## Ed(1) and Ed (2)

They are properties of a system that remain constant when the parameters of the phase space are well chosen. Thus, they are useful to choose a radius for the neighborhood and an embedding dimension. The function requires a delay, which depends on the periodical properties of the time series.

## Recurrence plot, cross recurrence and joint recurrence plot.

These buttons plot recurrence matrices either on the main panel ("Plot") or on an independent window ("Zoom"). The user can press Zoom directly.

In case the recurrence matrix is joint, the user has to choose the X and Y labels. The possible labels are the names of the time series that were used to calculate the recurrence matrix.

## 6.5.. Filling

The inputs are time series and, optionally, a gap set to fill. If there is no gap set in the environment or none of them is selected, KarsTS will try to fill all the gaps in the time series. On the Output Window will appear a summary that includes the inputs used, the gaps that could be filled and the ones that remain missing.

## Interpolations

The output is the filled time series, which will be named after the one with NAs, adding the suffix "\_lin" (linear interpolation), "spl" (spline interpolation) or "\_sti" (Stineman's interpolation,).

## ARIMA

This is a univariate filling method that combines a seasonal ARIMA model and a Kalman smoother. The time series needs to be linear, but KarsTS linearity tests do not support missing values. A solution is to perform a temporary filling and to test for linearity. Another option is to try to fill the gaps using the ARIMA button and run the linearity test *a posteriori*.

### Necessary inputs

- Time series to fill
- Period (in lags)
- ARIMA parameters: The "Suggest ARIMA parameters" button fills the entries corresponding to the ARIMA parameters automatically; most times, this is the easiest method to perform the ARIMA filling.

### Optional inputs

- Gap set: if not provided, KarsTS will build a gap set encompassing all NAs in the selected time series.

### Outputs

- Filled time series: It bears the name of the original time series and the suffix "ark", which stands for **ARIMA** and **Kalman**.
- Summary on the output window, including:
  - Period
  - ARIMA model: (AR,I,MA)x(seasonal AR, seasonal I, seasonal MA)
  - Time series
  - Gap set
  - Gap set length
  - Number of remaining gaps (within the gap set)

- A table containing the filled gaps

## Mean Value

Each missing value is replaced by the median (or mean) of the values located in equivalent positions in the periods around. In case none of these values is available, the central missing value cannot be filled. When some of them are available, the central missing value will be filled or not, depending on the threshold set by the user (minimum number of observations to each side). The process can be applied iteratively, so that a filled time series is the input for the next iteration. Consider using a small number of iterations because filling missing values does not increase the amount of information.

The mean value method is recommended for very regular seasonal time series. Yet, note that a missing value can be replaced by the median or mean of the surrounding points by setting the period to one.

### Necessary inputs

- Time series to fill
- Period (in lags)
- Number of periods around: total number of periods to both sides of the period to which the missing value belongs. For example, when the number of periods around is six, the window consists of three periods before and three periods after the central one. Note that the number of periods must be even; otherwise KarSTS will take the next even integer.
- Maximum number of iterations

### Optional inputs

- Gap set: if not provided, KarSTS will build a gap set encompassing all NAs in the selected time series.
- Minimum number of observations (one side): when the number of NAs, at least, at one side of the window exceeds this minimum number of observations, KarSTS does not fill the central NA. This avoids the introduction of a biased filling due to an asymmetric distribution of NAs around the NA to fill. The minimum number of iterations defaults to zero.
- Statistic: median (default) or mean.

### Outputs

- Filled time series: It bears the name of the original time series and the suffix “mvl”.
- Summary on the output window, including:
  - Period
  - Periods around
  - Minimum number of observations to each side

- Number of NAs filled in each iteration
- Time series
- Gap set
- Gap set length
- Number of remaining gaps (within the gap set)
- A table containing the filled gaps

## Multiv. Splines (Multivariate splines)

This multivariate filling method acts on each gap separately. In the first place, it fits a generalized linear model (GAM) where the variable to fill is the response variable. The model is fit in a window around the gap, not in all the time series. The user defines the length of the window in terms of % of the gap length. For example, the gap to fill has 200 missing values. If the user enters “90”, the window will include 180 measurements before the gap and 180 after it. The percentage can be greater than 100. For example, if it is 900 %, the window will include 1800 values before the gap and 1800 after. Note that the same percent will produce windows adapted to the length of the different gaps in the gap set.

The user has to choose between a fixed d.f. regression spline or a penalized regression spline. The allowed smoothing basis are cr (cubic regression spline), tp (thin plate regression spline) or any of the basis allowed in the function te from package mgcv.

## MissForest

MissForest is a multivariate filling method. It does not require the time series to meet any conditions. It performs a non-parametric missing value imputation using a random forest algorithm. The process can be time-consuming, especially when several iterations are performed. The user must enter the maximum number of iterations and the number of trees to grow.

The input time series contain the information needed to fill the gaps. The output time series are the ones to be filled and they are a subset of the input series.

The time series need to be simultaneous and have the same sampling period.

Optionally, the variables can be scaled or embedded. The default setting is not to embed, that is, embedding dimension 1 and delay 0.

## Twins

The Twins Filling Method can be univariate or multivariate (see [Twins](#)).

The principal inputs are the time series to fill and a recurrence matrix, which can be simple or joint.

When the time series to fill was used to create the recurrence matrix, the twins of the missing points themselves cannot be known. In this case, KarSTS uses the twins of the points close to the missing ones (up to a maximum distance).

It is more advisable to create the recurrence matrix using time series other than the one to fill; in this way, the missing values will be filled regardless of how many missing values surround them.

## ARIMAX

It fills gaps using an ARIMA model with external regressor variables (ARIMAX). The user has to choose on or more time series as regressors; apart from this, it works exactly as the ARIMA button.

## Check filling

This button tests the appropriateness of the filling. It compares the observed and the predicted values. Therefore, it can only be applied if the gaps are artificial.

### Inputs

- The time series without any artificial gaps
- The time series with artificial gaps
- The time series with filled artificial gaps

It performs a linear fitting between the observed and the predicted values. The summary of the fitting is written on the Output window. The function also produces two plots to check the goodness of the fit.

This button will lead to reject the filling only when it differs greatly from the original values; therefore, it is recommendable to verify the goodness of the filling by plotting the filled time series and the original one (button Plot ts) too.

## 7.. References

A. Alexander Beaujean (2012). BaylorEdPsych: R Package for Baylor University Educational Psychology Quantitative Courses. R package version 0.5. <https://CRAN.R-project.org/package=BaylorEdPsych>.

Achim Zeileis and Gabor Grothendieck (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. Journal of Statistical Software, 14(6), 1-27. doi:10.18637/jss.v014.i06.

Adrian Trapletti and Kurt Hornik (2017). tseries: Time Series Analysis and Computational Finance. R package version 0.10-40.

Antonio and Fabio Di Narzo (2013). tseriesChaos: Analysis of nonlinear time series. R package version 0.1-13. <https://CRAN.R-project.org/package=tseriesChaos>.

C. Agostinelli and U. Lund (2013). R package 'circular': Circular Statistics (version 0.4-7). URL <https://r-forge.r-project.org/projects/circular/>.

Constantino A. Garcia (2015). nonlinearTseries: Nonlinear Time Series Analysis. R package version 0.2.3. <https://CRAN.R-project.org/package=nonlinearTseries>.

Daniel Adler, Duncan Murdoch and others (2017). rgl: 3D Visualization Using OpenGL. R package version 0.98.1. <https://CRAN.R-project.org/package=rgl>.

Daniel J. Stekhoven (2013). missForest: Nonparametric Missing Value Imputation using Random Forest. R package version 1.4.

Hyndman RJ (2017). `_forecast`: Forecasting functions for time series and linear models\_. R package version 8.0, <URL: <http://github.com/robjhyndman/forecast>>.

James Wettenhall, Philippe Grosjean. R Tcl/Tk recipes.  
<http://www.sciviews.org/recipes/tcltk/toc/> [17-05-2017]

Korkmaz S, Goksuluk D, Zararsiz G. MVN: An R Package for Assessing Multivariate Normality. *The R Journal*. 2014 6(2):151-162.

Luke Tierney (2011). `tkrplot`: TK Rplot. R package version 0.0-23. <https://CRAN.R-project.org/package=tkrplot>.

Philippe Grosjean (2017). `SciViews`: A GUI API for R. UMONS, Mons, Belgium. URL <http://www.sciviews.org/SciViews-R>.

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Robert Marwan, M. Carmen Romano, Marco Thiel, Jürgen Kurths (2007). Recurrence plots for the analysis of complex systems. *Physics Reports* 438, 237-329.

Robert Nau, Statistical forecasting: notes on regression and time series analysis: ARIMA models for time series forecasting . <https://people.duke.edu/~rnau/411home.htm> [17-05-2017]

Ryan Hafen (2016). `stlplus`: Enhanced Seasonal Decomposition of Time Series by Loess. R package version 0.5.1. <https://CRAN.R-project.org/package=stlplus>.

Simon N. Wood (2016). Just Another Gibbs Additive Modeler: Interfacing JAGS and mgcv. *Journal of Statistical Software*, 75(7), 1-15. doi:10.18637/jss.v075.i07.

Tomas Johannesson & Halldor Bjornsson and Icelandic Met. Office; Gabor Grothendieck (2012). `stinepack`: Stineman, a consistently well behaved method of interpolation. R package version 1.3. <https://CRAN.R-project.org/package=stinepack>.

## **Appendix I: Functionalities**

Analyze the distribution of sampling frequencies and NAs in a time series (*Sampling periods*)

Apply an artificial set of gaps to a time series (*Apply gaps to time series*)

Build a time series of daily, monthly... values (*Aggregate*)

Calculate simple, cross and joint recurrence matrices (*Recurrence Matrix, Cross Recurrence Matrix and Joint Recurrence Matrix*)

Calculate the basic statistics of a time series (*Basic Statistics*)

Calculate the basic statistics of a time series in sliding windows (*Rolling Statistics*)

Calculate the reciprocal or the inverse time series (*Operations*)

Check if a time series follows an AR or ARIMA process (*Linearity Tests*)

Check if two time series are correlated (*Linear correlation and Mutual Information*)

Create a set of random artificial gaps (*Artificial Random Gaps*)

Create a specific artificial gap (*Specific Random Gaps*)

Cumulative sum of a time series (*Cumulative sum*)

Cut a piece of time series (*Select by Rg/Freq*)

Differencing a time series (*Differencing*)

Duplicate data sets (*Rename*)

Estimate the correlation dimension (fractal dimension of an object embedded in a phase space) (*Invariants*)

Estimate the correlation sum (probability of finding a neighbor in a ball of radius  $r$  in the phase space) (*Invariants*)

Estimate the delay to embed a time series (*Mutual Information*)

Estimate the determinism (a measure of the periodicity of a time series) (*Determinism*)

Estimate the embedding dimension of a time series (*False Nearest Neighbors,  $Ed(1)$  &  $Ed(2)$  and Invariants*)

Estimate the KS Entropy (measure of complexity) (*Invariants*)

Estimate the laminarity (*Laminarity*)

Estimate the length of the diagonal lines (predictability) (*Determinism*)



Estimate the radius necessary to create a recurrence matrix from a time series (*Invariants*)

Estimate the ratio (points belonging to diagonal lines / points in the recurrence matrix) (*Determinism*)

Estimate the recurrence rate (number of ones/ number of points in the recurrence matrix) (*Determinism and Laminarity*)

Estimate the trapping time (*Laminarity*)

Export data sets to .txt, .csv, and .mat files (*Export*)

Extract the seasonal component of a time series (the component can change over the time) (*Loess decomp.*)

Extract the trend of a seasonal time series (*Loess decomp.*)

Filling methods:

Univariate

ARIMA model (*ARIMA*)

Mean value (*Mean Value*)

Stinemann's, linear and spline interpolations (*Stinemann's interpolation, linear interpolation and spline interpolations*)

Twins (*Twins*)

Multivariate

ARIMAX model (*ARIMAX*)

Forest random algorithm (*Missforest*)

Generalized additive model (*Multiv. Splines*)

Twins (*Twins*)

Filter outliers causing unusually stiff slopes (*Rm. Slope Outliers*)

Find non-linear correlations or autocorrelations (*Mutual information*)

Get the dates, values and indices of one or more points in a plot (*Get coordinates*)

List the data sets to the Output Window along with a short description of each (*List*)

Load data sets (time series, multiple time series, gaps, recurrence matrices) (*Load*)

Merge consecutive pieces of time series (*Merge*)

Merge gap sets (*Merge*)

Perform a linear regression between the real values and the filled artificial gaps (*Check filling*)

Perform a loess seasonal decomposition on a time series (*Loess decomp.*)

Perform a number of linearity tests ( surrogate test, Teraesvirta's, White neural, Keenan's one-degree, McLeod-Li test, Tsay's Test and Likelihood ratio) (*Linearity Tests*)

Perform a number of stationarity tests ( Philipp-Perron unit root test, Box-Ljung test, Augmented Dickey-Fuller test and KPSS test for Level Stationarity) (*Stationarity Tests*)

Plot one or more time series (*Plot ts*)

Plot phase 2 or 3 dimensional phase portraits (*Phase portraits*)

Plot simple, cross and joint recurrence matrices (*Recurrence Plot, Cross Recurrence Plot and Joint Recurrence Plot*)

Plot the average mutual information of two time series (*Mutual Information*)

Plot the histogram of the values of a time series (*Histogram*)

Plot the linear autocorrelation and partial-autocorrelation of a time series (*Linear Correlation*)

Plot the linear correlation and partial-correlation of two time series (*Linear Correlation*)

Plot the self-average mutual information of a time series (*Mutual Information*)

Re-compose a time series (additive or multiplicative) after doing some changes to each component (*Operations*)

Remove data sets from the working environment (*Remove*)

Remove outliers manually (on a plot) (*Remove points*)

Rename data sets (*Rename*)

Resample a time series (or a piece of it) (*Cut and resampling*)

Round a time series to a number of decimal positions (*Round*)

Round a time series to a number of significant digits (*Round*)

Save data sets to the hard disk (*Save*)

Save the output panel content to a txt file (Save to txt)

Scale a time series (*Scale*)

Select a subset of existing NAs in a time series (*Select gaps*)

Smooth a non-seasonal time series using loess (you might take on account other variables to do so) (*Loess smooth.*)

Smooth a seasonal time series (*Loess Decomp and Compose, successively*)

Smooth a time series by averaging the data (ex. daily, monthly...) (*Aggregate*)

Smooth a time series using the mean or the median in sliding windows (*Rollings Statistics*)

Sum up or multiply two or more time series (*Operations*)

Take natural logarithm of a time series (*Operations*)

Test for missingness completely and random (Little's MCAR test)

Two or three-dimensional scatter plots (*Phase portraits*)

## Appendix II: Input files format

KarTS admits the following extensions: .R, .RData, .csv y .txt. Often, time series are stored originally in csv or text files, whilst gap sets and recurrence matrices use to be created with KarTS. The first two types of file can store multiple data sets. However, cvs or txt files can store only a single matrix, gap set, univariate time series or multivariate time series.

### Time series:

Times series can be univariate or multivariate; anyway, times must be located in the first column.

### .R or .RData. files

The files can contain several time series, gap sets and recurrence matrices. Each time series is a data frame with two columns: time and values. Times are POSIXct POSIXt dates with R default format (Y-m-d H:M). KarTS will introduce as many NAs as possible, in order that the difference between two consecutive dates is unique through the whole time series.

### .csv files

They can contain univariate or multivariate time series; in any case, times must be placed in the first column. The rest of the columns can contain numbers, NA (missing value) or empty cell (missing value, as well).

The first column of the first row is reserved for a title (for example, thisPlaceData). The second row must contain the variable names (for example, "time", "temperature", "Rn" etc.). When a csv file is loaded, KarTS decomposes the multivariate time series into univariate ones, assigning them those names ("temperature", "Rn" etc.). The loaded time series are R data frames as described in the previous section.

When KarTS exports a time series to a csv file, the rows containing only missing values are not written in order to save space.

	A	B	C	D		A	B	C	D		A	B	C	D
1	ts_examples				1	ts_examples				1	ts_examples			
2	time	ts_example	ts_example2		2	time	ts_example	ts_example2		2	time	ts_example	ts_example2	
3	11/22/2012 13:30	289.3243	289.2559		3	11/22/2012 13:30	289.3243	289.2559		3	11/22/2012 13:30	289.3243	289.2559	
4	11/22/2012 19:30	289.1798	289.2764		4	11/22/2012 19:30	289.1798	289.2764		4	11/22/2012 19:30	289.1798	289.2764	
5	11/23/2012 1:30	289.1922	289.2558		5	11/23/2012 1:30	289.1922	289.2558		5	11/23/2012 1:30	289.1922	289.2558	
6	11/23/2012 7:30	289.1612	289.2882		6	11/23/2012 7:30	289.1612	289.2882		6	11/23/2012 7:30	289.1612	289.2882	
7	11/23/2012 13:30	289.2083	289.2643		7	11/23/2012 13:30	289.2083	289.2643		7	11/23/2012 13:30	289.2083	289.2643	
8	11/23/2012 19:30	289.2007	289.2352		8	11/23/2012 19:30	289.2007	289.2352		8	11/23/2012 19:30	289.2007	289.2352	
9	11/24/2012 1:30				9	11/24/2012 1:30	NA	NA		9	11/25/2012 1:30	289.263	289.2095	
10	11/24/2012 7:30				10	11/24/2012 7:30	NA	NA		10	11/25/2012 7:30	289.2571	289.3084	
11	11/24/2012 13:30				11	11/24/2012 13:30	NA	NA		11	11/25/2012 13:30	289.2435	289.2585	
12	11/24/2012 19:30				12	11/24/2012 19:30	NA	NA		12	11/25/2012 19:30	289.2681	289.215	
13	11/25/2012 1:30	289.263	289.2095		13	11/25/2012 1:30	289.263	289.2095		13	11/26/2012 1:30	289.1932	289.2371	
14	11/25/2012 7:30	289.2571	289.3084		14	11/25/2012 7:30	289.2571	289.3084		14	11/26/2012 7:30	289.2383	289.2062	
15	11/25/2012 13:30	289.2435	289.2585		15	11/25/2012 13:30	289.2435	289.2585		15	11/26/2012 13:30	289.2855	289.2374	

Figure 2: Examples of time series in csv files. All the options are equivalent: missing values represented by empty cells, NA or missing rows.

### txt files

Times must occupy the first two columns: year, month and day in the first column; hours and minutes in the second one. Apart from this, they work in the same way csv files do.

## Gap sets

### .R or .RData files

Each gap set is a list containing the following elements:

**\$gaps:** rows in the original (univariate) time series that correspond to missing values

**\$tsIni:** original time series initial time

**\$tsEnd:** original time series final time

**\$samPerMin:** original time series sampling period (minutes)

**\$tsLength:** number of elements in the original time series.

**\$tsName:** original time series name

### .csv files

The elements explained above are displayed in columns. The first column of the first row must contain the gap set name; the second one, the columns names (gaps, tsIni etc.)

### .txt files

The initial and final dates are stored in two columns each one. Apart from this, they work in the same way csv files do.

#### Recurrence matrices

Recurrence matrices contain ones and zeros (and NAs). As recurrence matrices can be huge, KarsTS stores only those positions corresponding to ones. Besides, simple or joint recurrence matrices are symmetric, hence only the upper triangle is stored. Note that in a recurrence matrix the main diagonal is the line  $X=Y$ ; the lower triangle is the one located to the left and the upper triangle lies to the right.

### .R or .RData files

Each recurrence matrix is a list. Besides the positions corresponding to ones, it contains additional information about the time series from which it comes. Note that simple recurrence matrices come from a single time series, cross recurrence matrices from two time series and joint recurrence matrices from two or more time series.

**\$ones:** data frame with two columns. The first one refers to the ones matrix columns or X coordinates in the corresponding recurrence plot. The second one contains the ones rows or Y positions.

**\$tol:** tolerances used to create the recurrence matrix

**\$tsName:** original time series names

**\$embDim:** embedding dimensions used to create the matrix

**\$delay:** delays used to create the matrix

**\$dist:** distances used to create the matrix

**\$tsLength:** original time series lengths

**\$samPerSec:** original time series sampling periods

**\$tsIni:** original time series initial date

**\$type:** type of matrix; it can be "simple", "cross" or "joint"

## csv files

The elements described above are stored in columns. The matrix name appears in the first column of the first row; the columns names appear in the second row (that is, X, Y, tol, tsName, etc.)

## txt files

The initial Date is displayed in two columns. Appendix III: Most frequent problems

While KarsTS is performing a costly calculation, the interface might not be visualized correctly. When it is over the interface will recover its normal appearance. It might be necessary to drag the main window a little, though. KarsTS will warn you of potentially costly procedures.

## Loading frequent problems

The most frequent problems when loading time series files created with other programs are the following.

1. Some date is in a wrong place or it is repeated. Note that the time of the time series must be strictly increasing (nevertheless, there can be different sampling periods and time jumps).
2. The date format is not allowed by KarsTS. The allowed formats are:  
m/d/y H:M, d/m/y H:N, Y/m/d H:M, Y/d/m H:M, m-d-y H:M, d-m-y H:N, Y-m-d H:M, Y-d-m H:M.
3. The date format is not unique. Whether the new format is allowed or not, the loading will fail. Note that the date format can change in a distant part of the file.
4. When a row lacks its date, KarsTS will remove the whole row. Therefore, if the file has empty rows, the user does not need to remove them (unless they are located in the first or the second rows, which must contain the general title and the column titles respectively).
5. Some data columns have values that can be interpreted as text:
  - a. Symbols other than NA to indicate missing value (for example, --)
  - b. Cells in scientific notation (They might be in a part of the file the user is not watching)
  - c. Cells with a mistaken separator (for example, 12,5 in a file where the decimal separator is the point)
6. There is something written in other columns
7. If you .csv file has the date (year, month, day) in a column and the time (hours, minutes) in another one, you can sum them up in a single column. Alternatively, you can export the file to txt.
8. The variables names are in the first row, not in the second. The loading will not fail, but the imported time series will have numbers as names.
9. The time separations are not compatible amongst them. For example: 2004-12-5 00:30, 2004-12-5 01:00, 2004-12-5 01:30, 2004-12-5 01:32, 2004-12-5 02:02, 2004-12-5 02:32, 2004-12-5 03:02. In this case, it is necessary to modify it so that the result is:

2004-12-5 00:30, 2004-12-5 01:00, 2004-12-5 01:30, 2004-12-5 01:30, 2004-12-5 02:00. 2004-12-5 02:30, 2004-12-5 03:00.

10. The file was created in a computer with a different date format and the user's Excel interprets wrongly the dates. As a result, the file has more than one date format or the dates are not consecutive.

## **Appendix IV: Filling methods overview**

### **Univariate**

#### **Interpolations**

Interpolation is not a time-series specific method, that is, it does not account for the autocorrelations. However, it is fast and it can be used on gaps smaller than one period.

The Stineman's interpolation is similar to the splines, but it is less sensitive to noise. Thus, in most cases Stineman's interpolation is the best option.

#### **ARIMA**

Fits an ARIMA model to the time series and uses it to predict the missing values. The time series needs to meet some requirements. Linearity tests (as well as correlations and stationarity tests) are useful to verify whether an ARIMA model is appropriate for the time series. The program can suggest values for the ARIMA parameters, otherwise you can estimate them using the classical procedure (for a clear explanation, see <http://people.duke.edu/~rnau/411arim.htm>).

#### **Mean value**

This method requires a time series with a clear, regular period.

Each missing value is replaced by the median (or mean) of the equivalent places in the nearby periods. For example, in a time series with a daily seasonality, a missing value at 15:00 will be replaced with the median of the values of other days at the same hour.

The number of periods to use to calculate the median is chosen by the user. If you choose too many periods, the missing value probably will stop being related to the far away values (consider that the time series can have a trend or a change in the periods amplitude). If you choose too few periods, maybe some of them have missing values too.

### **Multivariate**

#### **Missing Random Algorithm**

A missing random algorithm (missForest, from package missForest by D.J. Stekhoven) is used to perform a multiple imputation. The algorithm classifies the multivariate values, thus producing a number of trees. Once the trees are trained, they are used to predict the values. Finally, the results of the different trees are pooled. The function missForest, by itself, does not account for periodicity because it is not time-series specific; however the program allows the user to introduce embedded time series.



## Twins

Twins are points that represent the same state of a dynamic system.

The state of the system is given by its trajectory through time. As an example, consider that the microclimate of a cave can be described using the  $\text{CO}_2$  concentration and the temperature (Tint) (figure 3). Measuring more variables would be redundant and measuring fewer would be insufficient. The system is given by a vector containing the  $\text{CO}_2$  concentration and the temperature at each time. If we projected the trajectory over the plane that contains the time and a variable axis, we would get the time series corresponding to that variable.

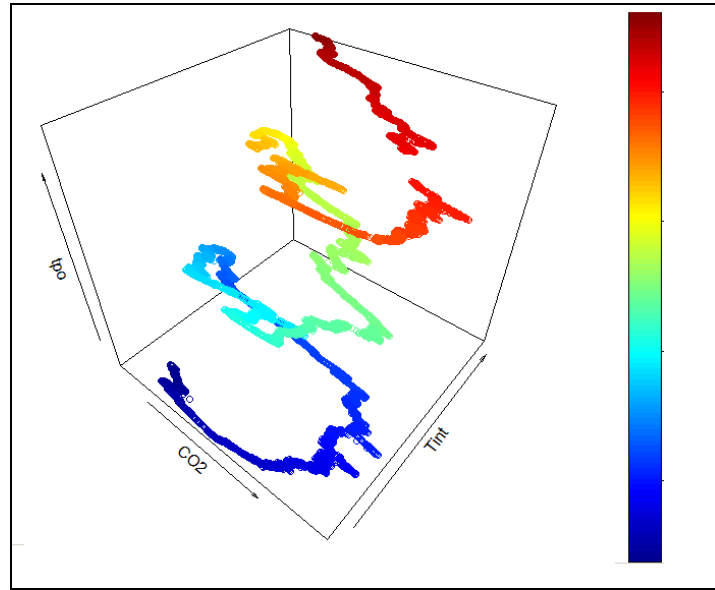


Figure 3: Example of trajectory

When we remove the time, we say we work on the *phase space*. In the previous example, the phase space has 2 dimensions. This can be visualized as a jet aircraft wake because you can see simultaneously all the points the aircraft has passed through (the wake phase space would be three-dimensional).

In order to represent the same state of a system, two points of a time series must have the same value. In practice, having the same value means that their difference is less than a threshold. In a recurrence matrix, it means that they are ones in a column. However, this condition is not sufficient. Twin points are so close that they share also their neighborhood (they have the same points around). Therefore, twin points correspond to identical columns in the recurrence matrix. Twin points are widely used to generate surrogate time series, however, we use them to fill missing values. Note that not all points in a time series have necessarily twins.

### Filling with twins

This method is based on finding twin points in a simple or joint recurrence matrix. Each missing value is filled independently from the others; therefore we will explain the process for a single missing value.

Case 1: joint recurrence matrix; the time series to fill was not used to create it.

Consider:

- the time series to fill (Ts.F)
- m predictor time series (Ts.1, Ts.2..., Ts.m)
- a joint recurrence matrix, JRM, created with the m predictor time series
- the position to fill, i. Ts.F (i) is missing and Ts.1 (i), Ts.2 (i)..., Ts.m (i) are known.

In the first place, the function finds the twins of the ith column in the recurrence matrix JRM. Let  $j_1, j_2, \dots, j_h$  be the twin columns. Note that the columns i,  $j_1, j_2, \dots, j_h$  are identical because they represent twin points (points having the same value and the same neighbors in the phase space). Then, the missing value is replaced by the median of Ts.F ( $j_1$ ), Ts.F ( $j_2$ ), ..., Ts.F ( $j_h$ ):

$$\text{Ts.F (i)} = \text{median (Ts.F (j}_1\text{), Ts.F (j}_2\text{), ..., Ts.F (j}_h\text{))}$$

Case 2: simple recurrence matrix; the time series to fill was not used to create it.

It is a particular case of the previous one, with m=1.

Case 3: joint recurrence matrix; the time series to fill was used to create it.

Consider:

- the time series to fill (Ts.F)
- m predictor time series (Ts.1, Ts.2..., Ts.m)
- a joint recurrence matrix, JRM, created with Ts.F and the m predictor time series
- the position to fill, i. Ts.F (i) is missing and Ts.1 (i), Ts.2 (i)..., Ts.m (i) are known.

The ith column in JRM lacks information because Ts.F (i) is missing and Ts.F was used to build JRM. Therefore, we find the twins of the previous column, that is, the (i-1)th column. Let  $j_1, j_2, \dots, j_h$  be the twin columns. Now, we use Ts.F ( $j_1+1$ ), Ts.F ( $j_2+1$ ), ..., Ts.F ( $j_h+1$ ) to replace the missing value:

$$\text{Ts.F (i)} = \text{median (Ts.F (j}_1\text{+1), Ts.F (j}_2\text{+1), ..., Ts.F (j}_h\text{+1))}$$

In case, Ts.F (i-1) is also missing, we use the (i-2)th column and replace the missing value by Ts.F (i) = median (Ts.F ( $j_1+2$ ), Ts.F ( $j_2+2$ ), ..., Ts.F ( $j_h+2$ )).

The process continues up to a maximum distance, D, chosen by the user.

Case 4: simple recurrence matrix; the time series to fill was used to create it.

It is a particular case of case 3, with m=1.

Note that when the time series to fill was used to create the recurrence matrix, each gap is filled from the outside in. However, when the time series to fill was not used to create the recurrence matrix, each NA can be filled no matter by how many more NA is surrounded. Note also that in case 1, D can be > 0, but it is not necessary.