TSAIB tutorial

Mathias Høxbro Juel Vendt

2022-05-02

Contents

Preface

The software package TSAIB was developed as the main deliverable for a synthesis project regarding the study-line: Earth and Space Physics and Engineering, with a specialization in Earth Observation, at DTU.

Installation

The software is implemented in the open source language "R" which can be found at https://www.r-project.org/. The source code to the "R" software package TSAIB is located at GitHub at https://github.com/MathiasVendt/TSAIB

From GitHub

The easiest way to install the package is directly from Github by using the R devtools package. The R-version should be 2.10 or higher.

1. Open R
2. Install the devtools library by typing

```
install.packages('devtools')
```

3. Install the TSAIB package

```
devtools::install_github("MathiasVendt/TSAIB")
```

Or by using the `remotes` library like this:

1. Open R
2. Load in `remotes` library

```
library(remotes)
```

3. Install the TSAIB package

```
install_github("MathiasVendt/TSAIB")
```

Dependencies

The packages depends on the following R libraries which can be installed from R with the function `install.packages`. Hence, to install the package "ncdf4"use the following command for the R window:

```
install.packages("ncdf4")
```

- ncdf4
- ggplot2
- ggpub
- forecast

A special library is also needed, and will be mentioned later in this tutorial

- TMB (Template Model Builder)

Together with a compiler tool, `RTools`. This can be installed by going to https://cran.r-project.org/, click on 'Download R for Windows', then 'Rtools', and select the latest version of `RTools` to download. After downloading has completed run the installer with default settings.

Getting started with R

The section gives a short introduction to R, which is useful to new R users.

R tutorials can be found at the r-project web side https://cran.r-project.org/manuals.html

Help pages can be accessed by typing "?" in front of a given function. If we want to access the help for the function `sum` we write

```
?sum
```

To start the web based help interface

```
help.start()
```

To exit R write

```
q()
```

Introduction to the package "TSAIB"

TSAIB is a R package aimed at getting statistical insight, correct for intermission bias and performing time series analysis of satellite observations consisting of sea-level anomalies measured in the arctic ocean. A variety of functions is provided in order to determine a time series model based on the analyses and insights from the analysis functions. In this introduction, examples are given to illustrate how to use the package.

To load the package simply write:

```
library(TSAIB)
```

Using "TSAIB" for analysis, visualization and manipulation of data

This section gives a step by step guide on how to utilize the package for analyzing the data, and the order in which the functions are used are recommended.

Load data

The test data set is called `TSdata`, and is included when the package is downloaded. When the TSAIB library is loaded, info about the data set can be found by the command:

```
?TSdata
```

In order to load the data into the R environment, the data set is assigned a name, e.g. "TSdata" like this:

```
TSdata=TSdata
```

The included data "TSdata" is a list containing 5 elements: Longitudes, latitudes, dates, measurements and a 3x3x325 construct, containing 325 measurements from a 3x3 data point grid. The data is extracted with the `GridTSExtract` function from the TSAIB library, and is done as:

```
TSdata=GridTSExtract(nco)
```

Where `nco` is the NetCDF data from which TSdata is extracted, and documentation on the data is listed in the Data appendix.

The user can also import their own data from a NetCDF file using the `GridTSExtract` function, and more info on how to use the function on a specific NetCDF file is shown by:

```
?GridTSExtract
```

Gather simple statistics about the data

It is now desired to get a brief overview of the data, including statistics, distributions and plotting. This can be achieved using the `TSdiagnostics` function. For this example, a meaned 30 point grid area (`TS_area`) is analyzed, which is centered around (lon,lat)=(0,73.25). To access sub elements of a construct in R, use `$` as a seperator between the parent object, and it's sub-directory. The function wont work for NaN elements, and if the extracted data contains a few of those, the `nanrem` option in the `TSdiagnostics` function is set to `"TRUE"`, which will omit any NaN elements for the analysis.

```
TSdiagnostics(TS_area)
```

qplot Histogram

qplot Boxplot

qplot Density

qplot Scatter
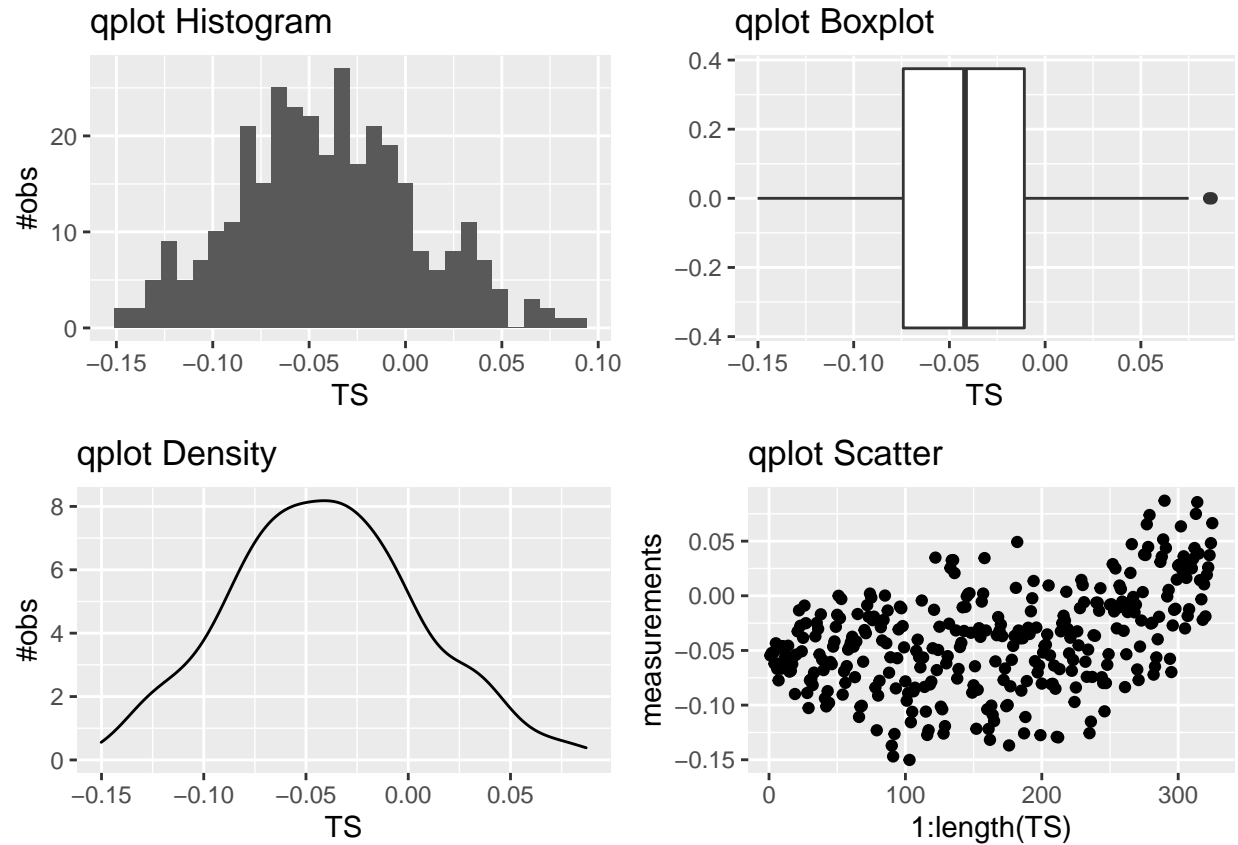

```
#> Number of NaN's in the data set
#> [1] 0
#> Number of objects in the data set
#> [1] 325
#> Fraction of the data set which is NaN's
#> [1] 0
#> mean
#> [1] -0.04076835
#> standard deviation
#> [1] 0.04648211
#> median
#> [1] -0.04184615
#> quantile
#>          0%         25%         50%         75%        100%
#> -0.15015656 -0.07411409 -0.04184615 -0.01085852  0.08699045
#> sum
#> [1] -13.24971
#>
#>  One Sample t-test
#>
#> data:  TS
#> t = -15.812, df = 324, p-value < 2.2e-16
#> alternative hypothesis: true mean is not equal to 0
#> 95 percent confidence interval:
#>  -0.0458408 -0.0356959
#> sample estimates:
```

```
#>   mean of x
#> -0.04076835
```

Removing NaN

If it is decided to remove NaN objects, the function `TSnanrem` can be used to do this, with a variety of options of how to deal with NaN. For this example, the NaN's of `TSmatrix` are to be replaced with the mean of the time series:

```
TSnanrem(TSdata$TSmatrix, method = "mean")
```

Intermission bias correction

Some times, the data consists of measurements from different satellite missions, and these measurements will have different biases and need to be aligned. This calls for correction of intermission biases, and if the bias is known, it can be accounted for using the `IB` function. The input to this function is a time series struct `TSM`, a date vector, a bias vector containing up to two different biases `c(bias1,bias2)` and lastly, a bias index struct, with the same dimensions as `TSM`, containing zeros for the restrained time series, and integers 1 & 2, for bias1 and bias2 respectively. In the following example, the same 3x3x325 time series as used before, has been added a bias of `c(1,-2)`, and is called `TimeSeriesMatrixBias`, with the corresponding bias index struct `biasid`.

```
IB(TimeSeriesMatrixBias,TSdata$date,c(1,-2),biasid)
```
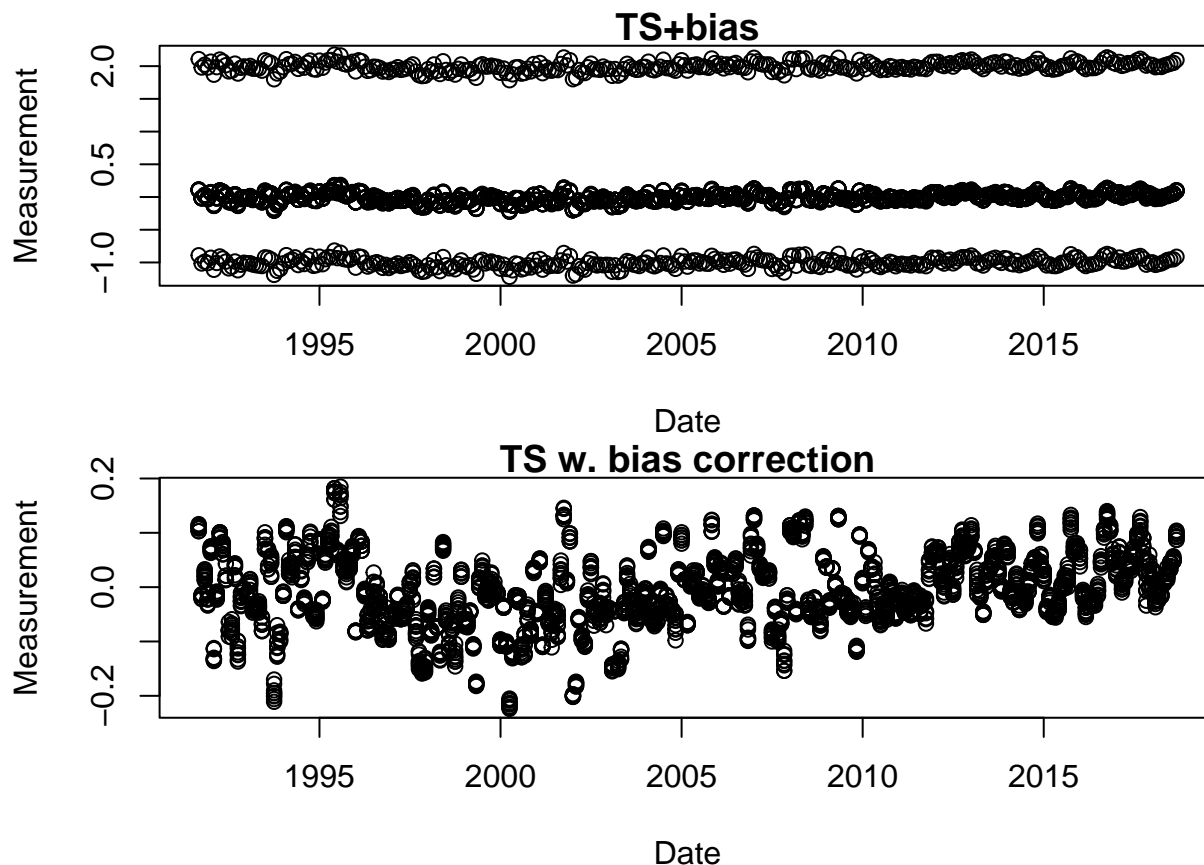
Time series analysis plots

It is possible to get an insight of the time series with the `TSAplots` function. This function utilizes R's inbuilt differencing function `diff`, auto correlation function `acf` and partial auto correlation function `pacf`, to display a collection of plots to determine the time series characteristics. It is possible to choose a transformation of the data, being either `sqrt` or `log` transformations, and even to save the plots in a variety of formats. The default settings is used, which doesn't transform the data or save the plots:

```
TSAplots(TS_area,TSdata$date)
```
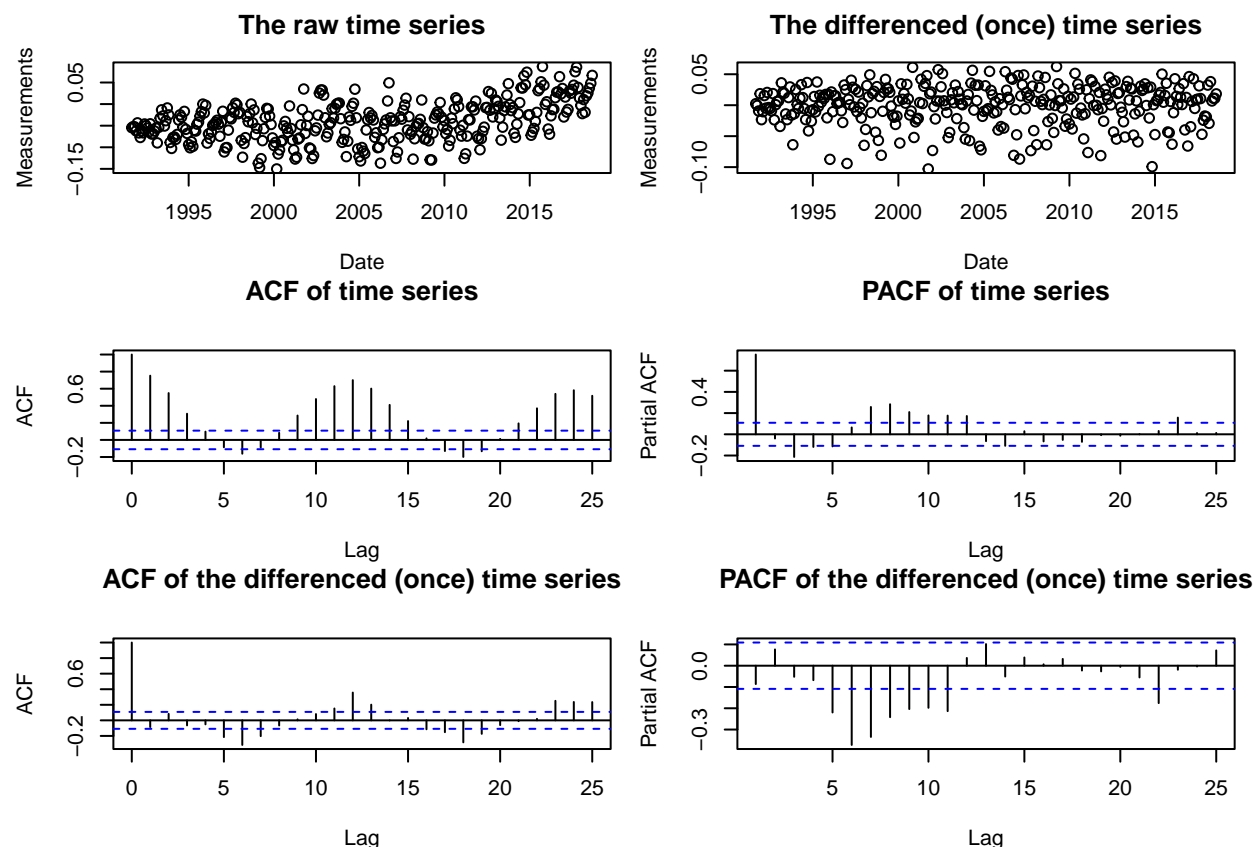

A "tail-off" decay is observed in "ACF of time series", indicating an AR(p) part, and a cut-off is observed at lag 3 on "PACF of time series", indicating an AR(p=3) part could be reasonably assumed. The second row of plots show the ACF and PACF of the differenced time series (differenced once!), and the slow decay in the first ACF plot is clearly removed. Differencing once (d=1) is then a reasonable assumption. After differencing, a positive spike is seen at lag 12 on the "ACF of the differenced (once) time series"-plot, which could indicate a seasonality of 12 (S=12). This assumption is likely to be true, as the data used for this example is monthly observations of sea-level anomalies.

Using "TSAIB" for model estimation

To estimate a model which is describing the time series at hand, the `TSAIB` package containes several functions with different approches:

Iterative scanning of ARIMA(p,d,q)x(P,D,Q)s models

From the time series analysis functions, some indications of adequate parameter estimates for ARIMA models may be present, and if so, the `ARIMAbuilder` function could be useful. The inputs are the maximum number of p,q,P,Q parts in the arima model, and a fixed level of differencing: d and D. The output will display the top 5 best models, based on the Bayesian Information Criteria (BIC). BIC is chosen as it, according to (Madsen 2008), yields a consistent estimate of the model order, and is given by:

$$\text{BIC} = N\text{log}\hat{\sigma}_\epsilon^2 + \log(N)(p + q + P + Q + d + D) \tag{1}$$

with N being the number of observations in the data set. By default, all parameters are set to 1, with seasonality set to 12 (i.e.: ARIMA(1,1,1)x(1,1,1)_12). For this example, however, the reasonable parameter guesses from `TSAplots` are used: ARIMA(p=3,d=1,q=?)x(P=?,D=?,Q=?)_S=12. Estimating ARIMA models is not an exact science, and ACF and PACF plots on real time series data is sometimes ambiguous. Hence the missing parameters are to be set to a reasonable guess of the maximum parameter value. The higher the parameter range, the more processing is needed to run `ARIMAbuilder`. To make the processing relatively fast, the remaining parameter estimates are set to 1: ARIMA(3,1,1,1,1,1)_12

```
ARIMAbuilder(TS_area,TSdata$date,3,1,1,1,1,1,12)
#> [1] 0 1 0 0 1 0
#> [1] 0 1 0 0 1 1
#> [1] 0 1 0 1 1 0
#> [1] 0 1 0 1 1 1
#> [1] 0 1 1 0 1 0
#> [1] 0 1 1 0 1 1
#> [1] 0 1 1 1 1 0
#> [1] 0 1 1 1 1 1
#> [1] 1 1 0 0 1 0
#> [1] 1 1 0 0 1 1
#> [1] 1 1 0 1 1 0
#> [1] 1 1 0 1 1 1
#> [1] 1 1 1 0 1 0
#> [1] 1 1 1 0 1 1
#> [1] 1 1 1 1 1 0
#> [1] 1 1 1 1 1 1
#> [1] 2 1 0 0 1 0
#> [1] 2 1 0 0 1 1
#> [1] 2 1 0 1 1 0
#> [1] 2 1 0 1 1 1
#> [1] 2 1 1 0 1 0
#> [1] 2 1 1 0 1 1
#> [1] 2 1 1 1 1 0
#> [1] 2 1 1 1 1 1
#> [1] 3 1 0 0 1 0
#> [1] 3 1 0 0 1 1
#> [1] 3 1 0 1 1 0
#> [1] 3 1 0 1 1 1
#> [1] 3 1 1 0 1 0
#> [1] 3 1 1 0 1 1
#> [1] 3 1 1 1 1 0
#> [1] 3 1 1 1 1 1
```

TS:Red, #1model:Black, Pred:Blue, 95% Conf.: Green
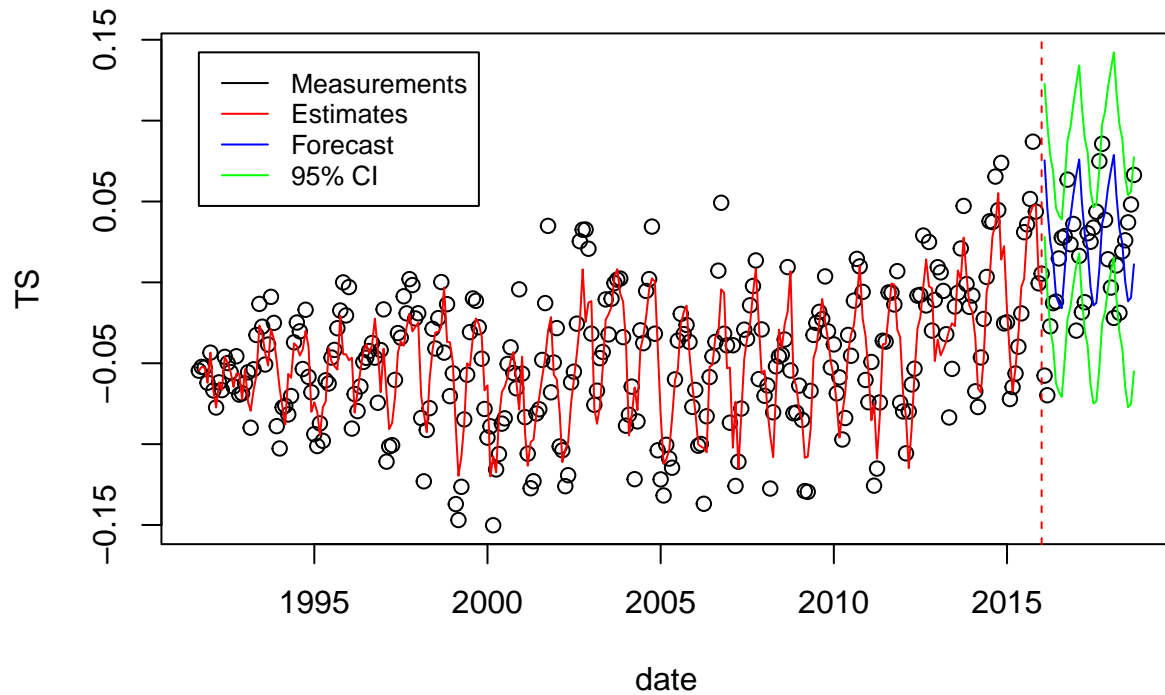
```
#> $'#1(p,q,P,Q)'
#>      dim1 dim2 dim3 dim4
#> [1,]    1    1    1    1
#>
#> $BIC1
#> [1] -2301.147
#>
#> $'#2(p,q,P,Q)'
#>      dim1 dim2 dim3 dim4
#> [1,]    2    1    1    1
#>
#> $BIC2
#> [1] -2297.367
#>
#> $'#3(p,q,P,Q)'
#>      dim1 dim2 dim3 dim4
#> [1,]    3    1    1    1
#>
#> $BIC3
#> [1] -2295.258
#>
#> $'#4(p,q,P,Q)'
#>      dim1 dim2 dim3 dim4
#> [1,]    1    1    0    1
#>
#> $BIC4
```

```
#> [1] -2284.659
#>
#> $'#5(p,q,P,Q)'
#>      dim1 dim2 dim3 dim4
#> [1,]    2    1    0    1
#>
#> $BIC5
#> [1] -2281.368
```

It is seen that ARIMA(3,1,1)x(1,1,1)_12 is on third place, ARIMA(2,1,1)x(1,1,1)_12 is on second place, and the best model based on BIC is: ARIMA(1,1,1)x(1,1,1)_12, and that is shown in the figure above.
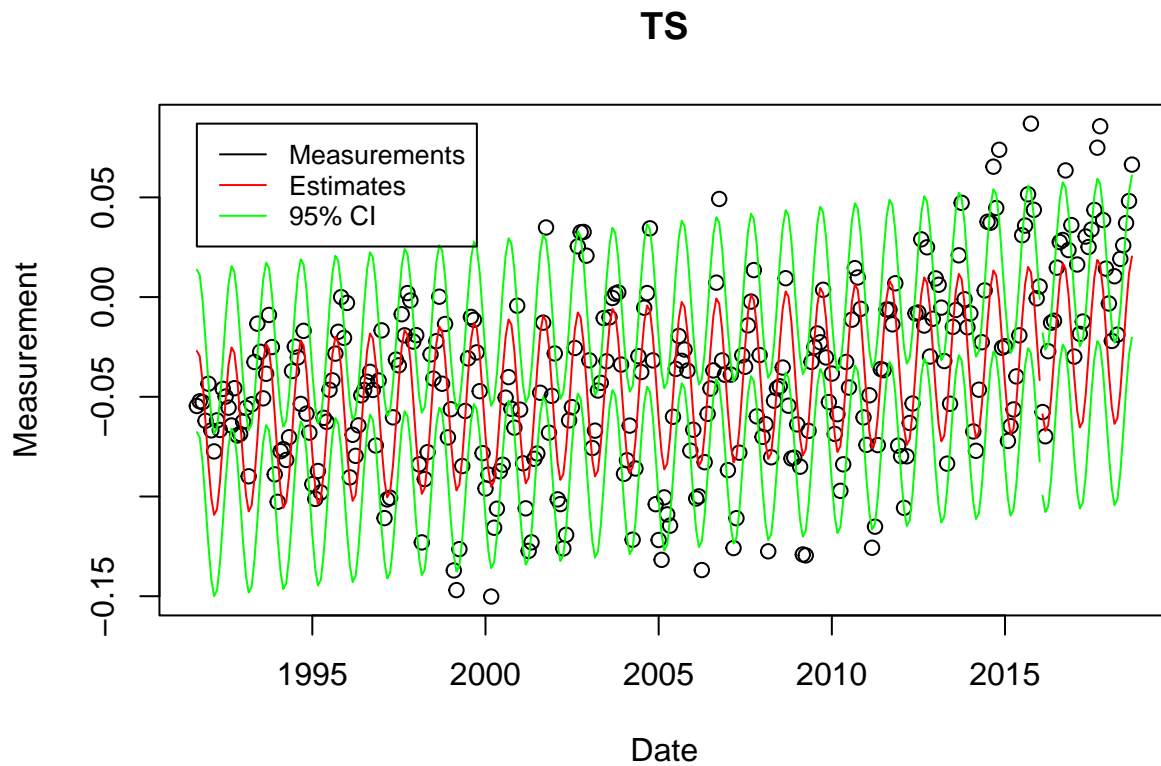
Simple time series model estimation, using OLS

The function **TSSmodel** estimates a simple model of the form:

$$Y_t = \alpha + \beta_t t + \beta_s \sin(\frac{2\pi}{p}t) + \beta_c \cos(\frac{2\pi}{p}t) \tag{2}$$

where the alpha and the betas are the estimated parameters, and p and t is set to 12 and 1 respectively by default:

```
TSSmodel(TS_area,TSdata$date)
```

TS


```
#> [1] "Parameter estimates:"
```

```
#> [1] "alpha"
#> [1] -0.06869383
#> [1] "beta_t (trend)"
#> [1] 0.0001462904
#> [1] "beta_s (sinus)"
#> [1] 0.02571606
#> [1] "beta_c (cosinus)"
#> [1] 0.03311466
```

Using TMB to estimate a RW model, and account for intermission bias

TMB stands for "Template Model Builder", and is a C++ interface for automatic differentiation that implements Laplace's method to estimate a time series model. For more info on TMB and how to install the package, please visit https://github.com/kaskr/adcomp.

Selected TMB functions have been implemented into TSAIB, in order to efficiently estimate a Random Walk (RW) model, based on likelihood functions. These TMB functions are gathered in one TSAIB function, called `TMBTSA`. This function has three inputs: TSM (a struct with time series data), date (the usual vector containing time stamps) and biasid (a struct with the same dimensions as TSM, containing the ones where the biased data is). `TMBTSA` can handle one bias at a time, and doesn't need to know the exact bias value, in contrast to the `IB` function, as it will estimate constant bias in the process. Furthermore, TSM doesn't have to be corrected for NaN, as `TMBTSA` will handle those as well.

For the next example, a synthetic time series struct `TimeSeriesMatrixBiasTMB` is used. As with the demonstration for the `IB` function, this is an altered version of `TSdata$TSmatrix`, with a single bias added this time. An illustration of the synthetic data is shown below:

TS with bias
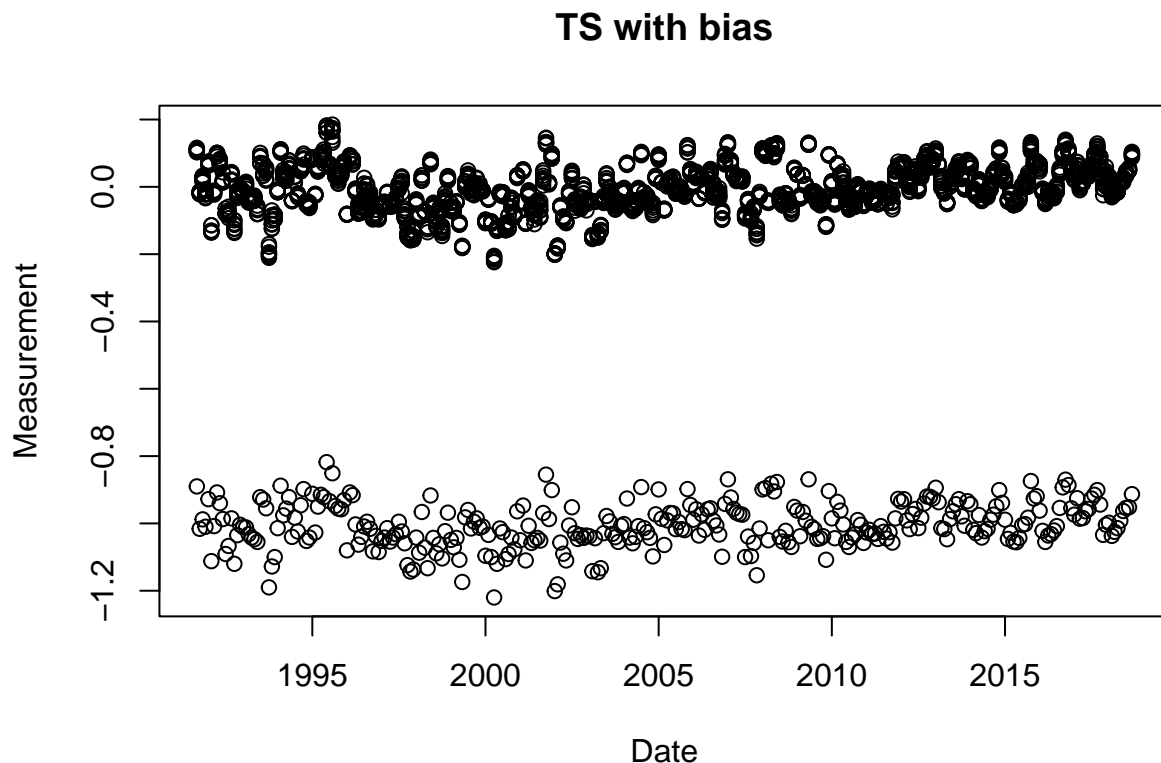
The bias index that matches `TimeSeriesMatrixBiasTMB` is `biasidTMB`. Together with the date vector `TSdata$date`, `TimeSeriesMatrixBiasTMB` and `biasidTMB` are inserted into `TMBTSA` as so:

```
TMBTSA(TimeSeriesMatrixBiasTMB,TSdata$date,biasidTMB)
```

The iterative messages from the process has been left out, as it's excessive information for this tutorial. The estimated RW model is shown below:
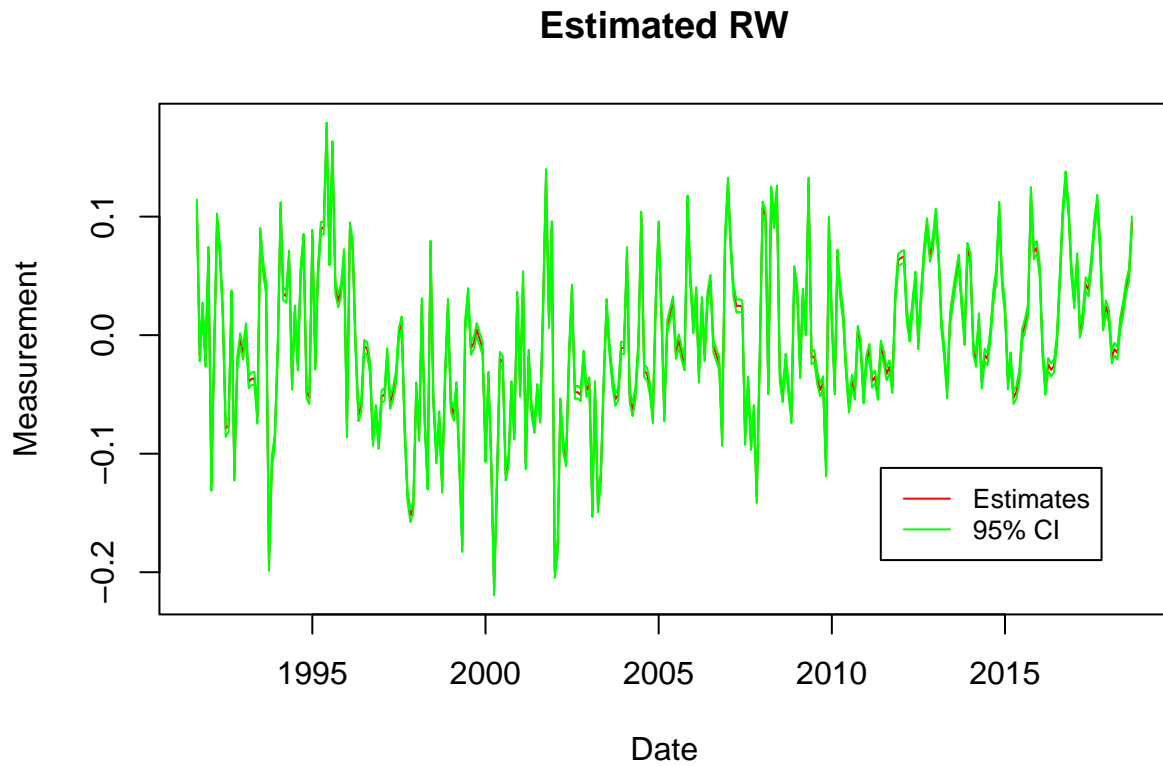
To compare the estimates with the data without bias, `TSdata$TSmatrix` is plottet together with the estimates below.

TS without bias + Estimated RW
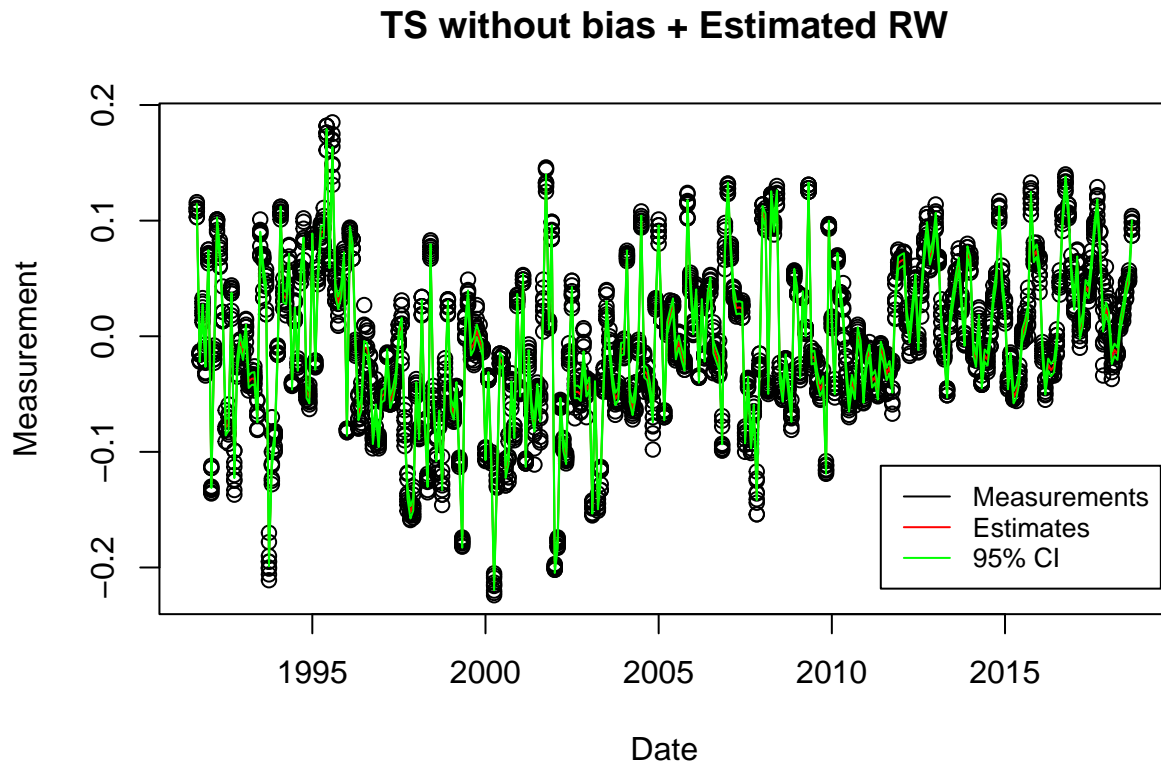
`TMBTSA` assumes the data to be independent and uncorrelated, and since the test data used in this example is satellite measurements of a certain area, there might be spacial correlation between the observations per time stamp. This impacts the calculation for the standard deviation, and it is can be seen in the figures, where the 95% CI lies extremely close to the estimates, making the estimates hard to distinguish from the CI.

Conclusion

Discussion

Appendices

Data set used for demonstration of the TSAIB package:

TSdata

```
?TSdata
 Description
  An extracted list using the "GridTSExtract" function, from the data set:
  Rose, S.K.; Andersen, O.B.; Passaro, M.; Ludwigsen, C.A.; Schwatke,
  C. Arctic Ocean Sea Level Record from the Complete Radar
  Altimetry Era: 1991-2018. Remote Sens. 2019, 11, 1672.
  https://www.mdpi.com/2072-4292/11/14/1672
```

```
Usage
 TSdata
Format
 A large list containing 5 elements, which are:
 longitude: containing longitudes from -180:180, by increments of 0.5 degrees
 (length: 720)
 latitude: containing latitudes from 65:81.5, by increments of 0.25 degrees (length: 67)
 date: containing years from 1991:2019, by increments of 1/12 (length:325)
 measurements: containing sea_level_anomaly measurements for each increments of:
 longitude, latitude and date. (dimension: [67,720,325])
 TSmatrix: containing a 3x3x325 grid of measurements centered around (lon,lat)=(-165,74)
```

TSdata2

```
?TSdata2
 Description
 An extracted list using the "GridTSExtract" function, from the data set:
 Rose, S.K.; Andersen, O.B.; Passaro, M.; Ludwigsen, C.A.; Schwatke,
 C. Arctic Ocean Sea Level Record from the Complete Radar
 Altimetry Era: 1991-2018. Remote Sens. 2019, 11, 1672.
 https://www.mdpi.com/2072-4292/11/14/1672
Usage
 TSdata2
Format
 A large list containing 5 elements, which are:
 longitude: Containing longitudes from -180:180, by increments of 0.5 degrees
 (length: 720)
 latitude: Containing latitudes from 65:81.5, by increments of 0.25 degrees (length: 67)
 date: Containing years from 1991:2019, by increments of 1/12 (length:325)
 measurements: Containing sea_level_anomaly measurements for each increments of:
 longitude, latitude and date. (dimension: [67,720,325])
 TSmatrix: Containing a 61x61x325 grid of measurements centered around (lon,lat)=(0,73)
```

TS_area

```
?TS_area
 Description
 A time series containing the meaned and NaN removed time series, from using the
 "TSnanrem" function on "TSdata$TSmatrix"
Usage
 TS_area
Format
 A numerical vector with length: 325
```

TimeSeriesMatrixBias

```
?TimeSeriesMatrixBias
 Description
 A time series matrix with added biases: 1 and -2, on the data: "TSdata$TSmatrix"
Usage
 TimeSeriesMatrixBias
Format
 A numerical struct with dimensions 3x3x325
```

biasid

```
?biasid
 Description
  A bias index containing bias indicies: 1 and 2, matching the synthetic bias data
  "TimeSeriesMatrixBias"
 Usage
  biasid
 Format
  A numerical struct with dimensions 3x3x325
```

TimeSeriesMatrixBiasTMB

```
?TimeSeriesMatrixBias
 Description
  A time series matrix with added biases: 1, on the data: "TSdata$TSmatrix"
 Usage
  TimeSeriesMatrixBiasTMB
 Format
  A numerical struct with dimensions 3x3x325
```

biasidTMB

```
?biasidTMB
 Description
  A bias index containing bias index: 1, matching the synthetic bias data
  "TimeSeriesMatrixBiasTMB"
 Usage
  biasidTMB
 Format
  A numerical struct with dimensions 3x3x325
```

DEMOpllam

```
?DEMOpllam
 Description
  RW model estimates, calculated using the TMBTSA function on "TimeSeriesMatricBiasTMB"
  and "biasidTMB"
 Usage
  DEMOpllam
 Format
  A numerical vector with length: 325
```

DEMOplsdlam

```
?DEMOplsdlam
 Description
  RW model standard deviations, calculated using the "TMBTSA" function on
  "TimeSeriesMatricBiasTMB" and "biasidTMB"
 Usage
  DEMOplsdlam
 Format
  A numerical vector with length: 325
```

DEMOtval

```
?DEMOtval
 Description
  T-value for the 95% confidence interval for the TMBTSA function
 Usage
  DEMOtval
 Format
  A numerical value
```

Functions

GridTSExtract

```
?GridTSExtract
 Description
  Extracts a time series from a NetCDF lon&lat coordinate grid
 Usage
GridTSExtract(
  nco,
  lonid = "longitude",
  latid = "latitude",
  timeid = "date",
  measurementsid = "sea_level_anomaly",
  coord = c(-165, 74),
  radius = 0,
  dlon = 2,
  dlat = 4
)
 Arguments
  nco: The NetCDF object (open with "nc_open" from the "ncdf4" library)
  lonid: The variable id for longitude vector. e.g.: "longitude"
  latid: The variable id for latitude vector. e.g.: "latitude"
  timeid: The variable id for the time vector. e.g.: "date"
  measurementsid: The variable id for the measurements vector. e.g.: "sea_level_anomaly"
  coord: The center coordinate for the TS grid area. e.g.: c(lon,lat) i.e. c(-164,74)
  radius: The square radius of the TS grid. e.g: 0=1x1 grid, 1=3x3 grid, 2=5x5 grid etc.
  dlon: Number of data points for each degree longitude
  dlat:Number of data points for each degree latitude
 Value: Time series from grid
```

TSdiagnostics

```
?TSdiagnostics
 Description
  Shows basic statistics and characteristics of the time series data
 Usage
  TSdiagnostics(TS, nanrem = "FALSE")
 Arguments
  TS:The time series to be analyzed
  nanrem: Set NaN to be removed or not. e.g.: nanrem="TRUE"
 Value: Basic statistics and characteristics
```

TSnanrem

```
?TSnanrem
 Description
  Removes or imputes NaN from the time series data set, with a specified method of choice
 Usage
  TSnanrem(TS, method = "mean", value_nr = 0)
 Arguments
  TS: The time series to remove NaN from
  method: The imputation method. e.g.: "omit", "mean", "median" or "value"
  value_nr: The specific value for the "value" imputing method. e.g.: value_nr=0
 Value: Time series with NaNs removed or replaced
```

IB

```
?IB
 Description
  Corrects a struct of timeseries for up to two intermission biases
 Usage
  IB(TSM, date, biasvec = c(1, -2), biasid)
 Arguments
  TSM: The time series struct with the intermission bias
  date: The time axis in the time series (dates, seconds, intervals)
  biasvec: The vector containging up to 2 different bias. If the data only contains one7
  bias, insert that value instead of the vector
  biasid: A struct with the same dimensions as TSM, with integers indicating the bias:
  0 for the restrained time series, 1 for the first bias and 2 for the second bias.
Value: Time series struct with bias correction
```

TSAplots

```
?TSAplots
 Description
  Plots and saves the ACF and PACF for a pre-defined time series, with and without data
  transformations (log, sqrt), and differencing. Input type: Time series
 Usage
  TSAplots(TS, date, trans = "NONE", saveas = "NONE")
 Arguments
  TS: The time series for the TSAplots function
  date: The time axis in the time series (dates, seconds, intervals)
  trans: data transform parameter: trans='log' for log transform, trans='sqrt' for sqrt
  transform, and trans='NONE' for no transform
  saveas: specifies the file format to save the plots in the current working directory.
  Possible formats is: PDF, JPEG, TIFF, BMP and PNG. e.g.: "pdf" or "jpeg".
  If no format is specified, it will not save the plots
 Value: Plots from TSAplots
```

TSSmodel

```
?TSSmodel
 Description
  Estimates a simple time series model
```

```
Usage
 TSSmodel(TS, date, p = 12, t = 1, testsize = round(length(date)/10))
Arguments
 TS: The time series for the TSSmodel function
 date: The time axis in the time series (dates, seconds, intervals)
 p: The period. e.g.: p=12, for 12 months in a year
 t: The time steps. e.g.: t=1, for monthly time steps, with period 12
 testsize: The number of data points to be used for testing the model
Value: Simple TS model
```

ARIMAbuilder

```
?ARIMAbuilder
 Description
  Lists the top 5 estimated ARIMA/SARIMA models of the form: ARIMA(p,d,q)x(P,D,Q),
  based on specified parameter values and ranges
 Usage
  ARIMAbuilder(TS, p = 1, d = 1, q = 1, P = 1, D = 1, Q = 1, S = 12)
 Arguments
  TS: The time series to build the models upon (remember to transform before
  using this function!)
  p: The maximum AR value, estimated from plot analysis
  d: The fixed differencing, often 0, 1 or 2
  q:    The maximum MA value, estimated from plot analysis
  P:    The maximum seasonal AR value, estimated from plot analysis
  D:    The fixed seasonal differencing, often 0, 1 or 2
  Q:    The maximum seasonal MA value, estimated from plot analysis
  S:    The seasonality/period of the seasonal differencing
 Value: list of top 5 ARIMA/SARIMA models
```

TMBTSA

```
?TMBTSA
 Description
  Utilizes TMB, which is a C++ interface for automatic differentiation that implements
  the Laplace method, to estimate a time series model
 Usage
  TMBTSA(TSM, date, biasid)
 Arguments
  TSM: The time series struct, with dimensions (x/lon,y/lat,timesteps)
  date: The time axis in the time series (i.e., dates, seconds, intervals)
  biasid: A struct with the same dimensions as TSM, with integers indicating the bias:
  0 for the restrained time series and 1 for the bias. If no bias is present, biasid
  contains only zeros.
 Value: RW model
```

References

Madsen, Henrik. 2008. "Time Series Analysis." http://www.http://henrikmadsen.org/books/time-series-analysis/.