s2dverification: Time series visualization function

N. Manubens, A. Hunter

September 26, 2016

1 PlotTimeSeries

This is a preliminary report describing the updated time series plotting function, *PlotTimeSeries*, and presents several examples that highlight its improved functionality. The new function is an agglomerate of the previous time series plotting functions in the R package *s2dverification*, with additional functionality. The function is still evolving, to ensure that it is compatible with the Common Data Model (CDM) R data structure which is currently being discussed within the QA4Seas project.

In brief, the *PlotTimeSeries* function takes arrays of any number of (named) dimensions and creates scatter plots or draws curves along them, as well as plotting their means and standard errors. The inputted data can include metadata that is automatically represented on the plot, such as the name of the plotted variable, the start dates of a set of provided seasonal forecasts, the date at each forecast time step or the names of the members. The user can manually specify which dimensions to colour or style along, but the function is designed to anticipate the user's requirements. The advantages of the new function are its increased ease-of-use and versatility, as well as the improved quality of the final plots due to the use of the *ggplot2* package.
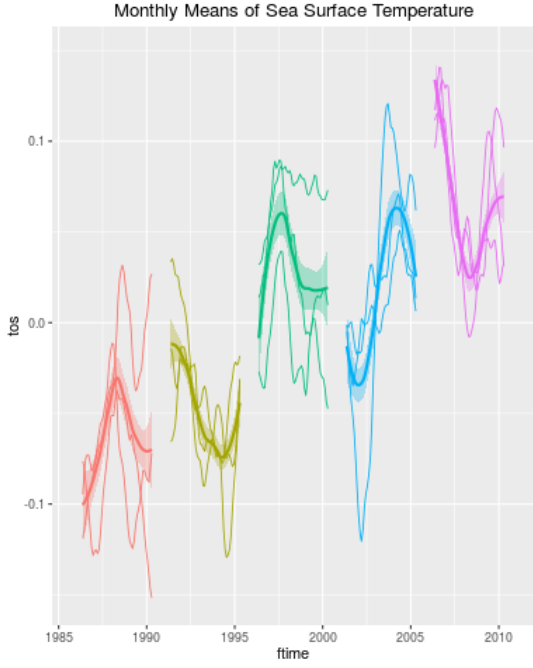
2 Examples

The default settings of the *PlotTimeSeries* function assume that the data to be plotted is either in the CDM format or it is an array of any number of dimensions, and the desired output is a scatter plot with different colours for the start dates and different shapes for the model members, with a smoothing function fitted along the start dates (Loess) and its standard error. However, if these dimensions cannot be automatically found in the data provided, the default settings are disregarded. The function also accepts bare R arrays without dimension names or meta-data, or can be adjusted to recognize any custom convention for the dimension names, allowing for its application to other research fields than climate prediction.

Plots of the mean sea surface temperature anomalies are shown in Fig. 1, using the sample dataset that is loaded with the code in the Appendix.
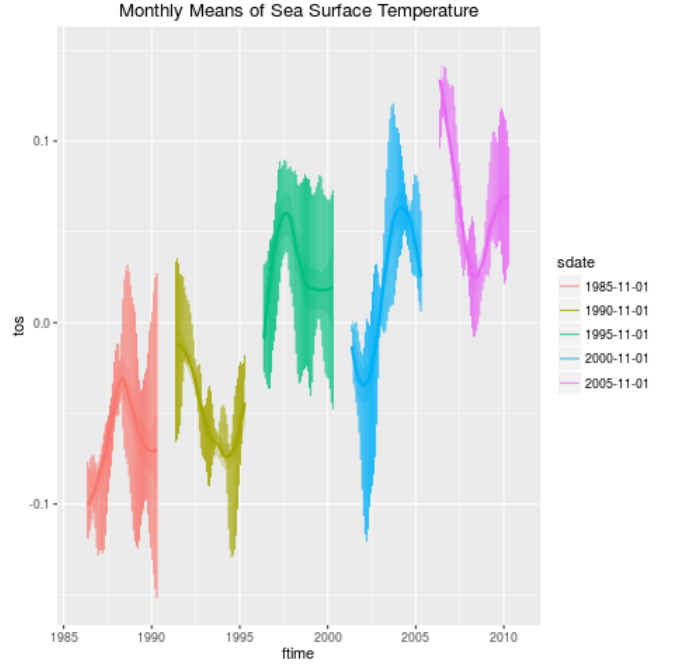
```
PlotTimeSeries(ano_exp)
PlotTimeSeries(ano_exp, curves_along = 'member')
PlotTimeSeries(ano_exp, curves_along = NA)
PlotTimeSeries(ano_exp, curves_along = 3)
```

The function allows the user to plot individual model members with or without the mean, or to display only the mean and the maximum and minimum.
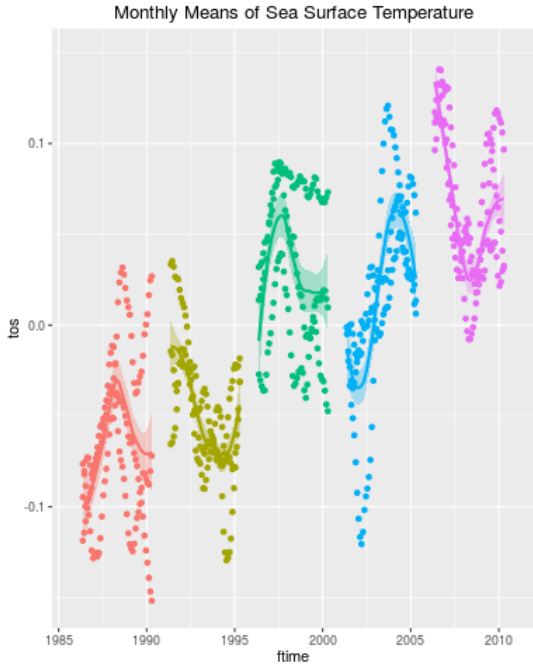
Although the default setting is for the function to colour along the start dates, the user can change this to any of the named dimensions, as in Fig 2.
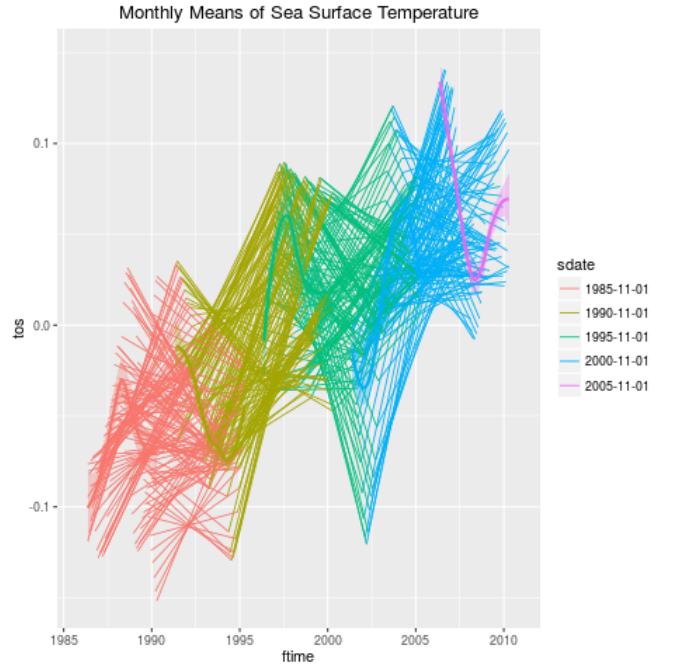
Figure 1: Drift-corrected anomalies of monthly sea surface temperature averages over the whole globe, from the i00k experiment run at BSC-CNS using the EC-Earth (v2.3.0) climate model, initialized every November 1st for the start dates 1985, 1990, 1995, 2000 and 2005. The forecast duration is 5 years
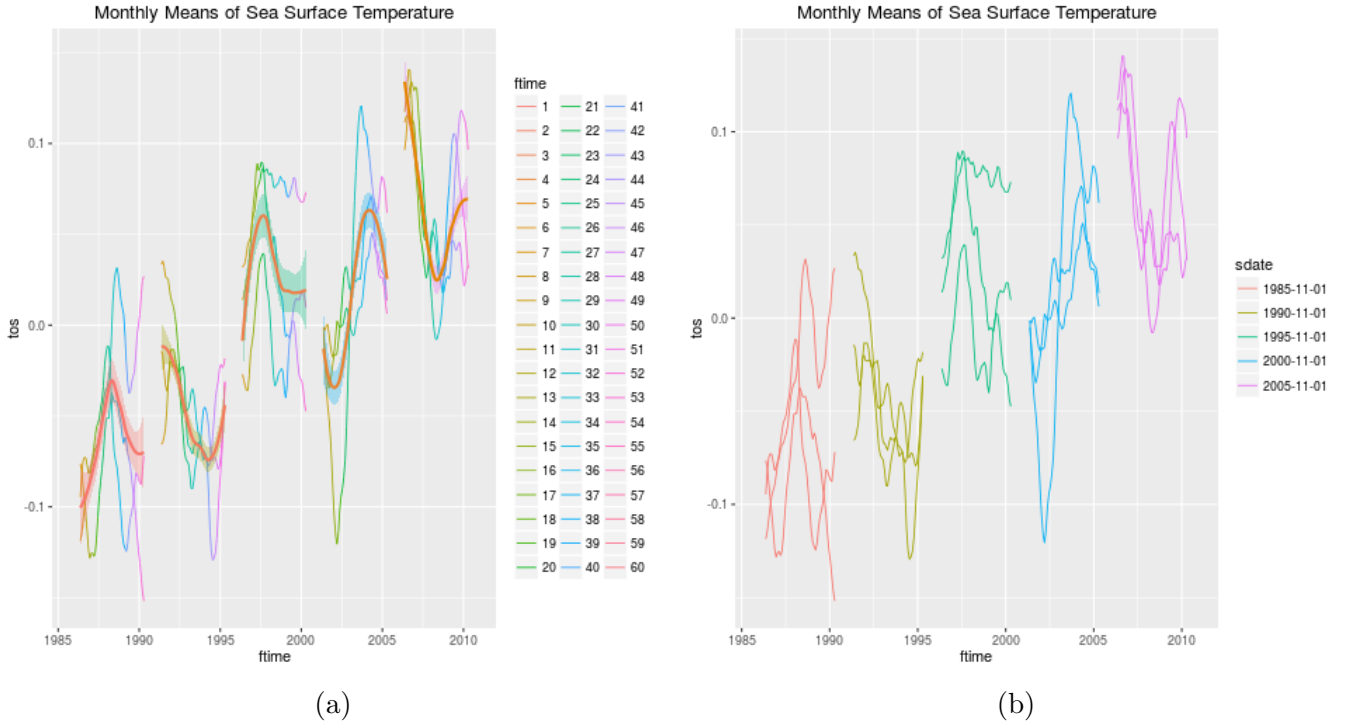
Figure 2: Drift-corrected anomalies of monthly sea surface temperature averages over the whole globe, from the i00k experiment run at BSC-CNS using the EC-Earth (v2.3.0) climate model, initialized every November 1st for the start dates 1985, 1990, 1995, 2000 and 2005. The forecast duration is 5 years

3 Future work

The function is being refined to accommodate a wide range of user requirements. Specifically, the user will also be able to plot the confidence intervals and errors bars, when these values are included in the data. This work is being conducted alongside changes currently being made to the *s2dverification* and *downscaleR* packages to ensure a common R data structure.

4 Appendix

R code for reproducing the figures.

```
# Loading example data
library(s2dverification)
data_path <- system.file('sample_data', package = 's2dverification')
expA <- list(name = 'experiment', path = file.path(data_path,
'model/$EXP_NAME$/$STORE_FREQ$_mean/$VAR_NAME$_3hourly',
'$VAR_NAME$_$START_DATE$.nc'))
obsX <- list(name = 'observation', path = file.path(data_path,
'$OBS_NAME$/$STORE_FREQ$_mean/$VAR_NAME$',
'$VAR_NAME$_$YEAR$$MONTH$.nc'))

# Now we are ready to use Load().
startDates <- c('19851101', '19901101', '19951101', '20001101', '20051101')
sampleData <- Load('tos', list(expA), list(obsX), startDates,
output = 'areave', latmin = 27, latmax = 48,
lonmin = -12, lonmax = 40)
```

```
# Doing example calculations
clim <- Clim(sampleData$mod, sampleData$obs)
ano_exp <- Ano(sampleData$mod, clim$clim_exp)
ano_exp <- Smoothing(ano_exp)
ano_obs <- Ano(sampleData$obs, clim$clim_obs)
ano_obs <- Smoothing(ano_obs)

# Transforming to 'hypothetical' common data structure
ano_exp <- list(Data = ano_exp,
Variable = sampleData$Variable,
Datasets = sampleData$Datasets$exp,
Dates = sampleData$Dates)
class(ano_exp) <- 'multiCDM'
ano_obs <- list(Data = ano_obs,
Variable = sampleData$Variable,
Datasets = sampleData$Datasets$obs,
Dates = sampleData$Dates)
class(ano_obs) <- 'multiCDM'

# Building extra objects with data for only 1 startdate (exp2) and for
# only 1 forecast time step (exp3)
exp2 <- ano_exp
exp2$Data <- Subset(exp2$Data, 'sdate', 1, drop = 'selected')
exp2$Dates$start <- exp2$Dates$start[1:60]
exp2$Dates$end <- exp2$Dates$end[1:60]
exp3 <- ano_exp
exp3$Data <- Subset(exp3$Data, 'ftime', 30, drop = 'selected')
exp3$Dates$start <- exp3$Dates$start[seq(1, 300, 60)]
exp3$Dates$end <- exp3$Dates$end[seq(1, 300, 60)]


source('PlotTimeSeries.R')
PlotTimeSeries(ano_exp)
PlotTimeSeries(ano_exp, curves_along = 'member')
PlotTimeSeries(ano_exp, curves_along = NA)
PlotTimeSeries(ano_exp, curves_along = 3)
PlotTimeSeries(ano_exp, colour_along = 'ftime')
PlotTimeSeries(ano_exp, mean_along = NA)
```