ISYE6501

Homework #4

7.1]

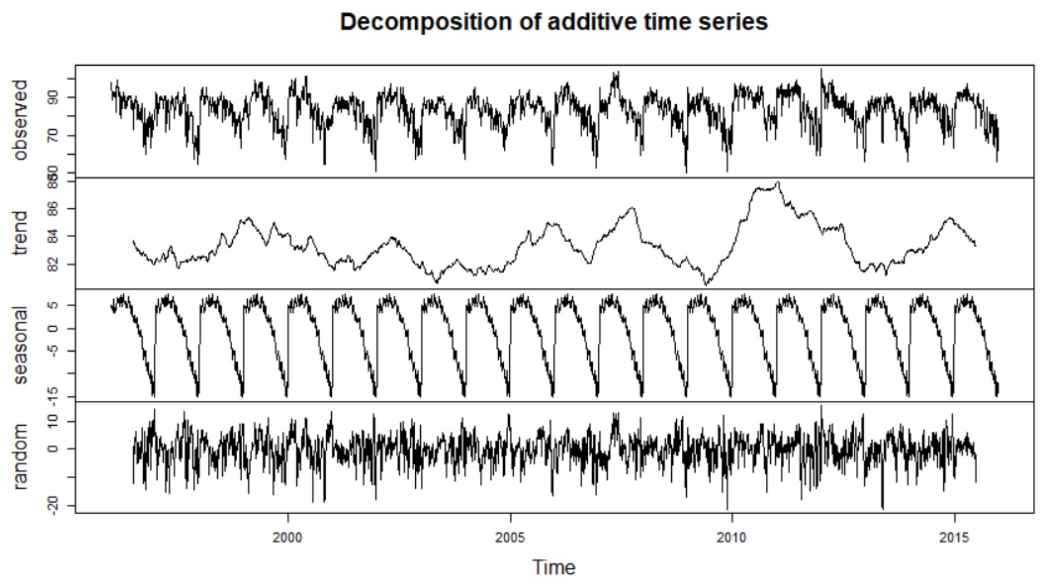
An exponential smoothing model can be used to forecast the monthly sales numbers in any given industry. The data collection would essentially be the daily Purchase orders (PO's) placed and paid by customers to order any given product.

The intention would be to highlight if there is an increased or decreased demand in the market during holiday seasons for the given product.

I would expect the alpha value to be closer to 0 assuming there are no trends and the only seasonality is holiday vs non holiday month. An exponential smoothing model using Holtwinters can be used to smooth out the data

7.2]

The data set containing the temperature by day for each year between 1997 to 2015 was first converted into a timeseries vector. Plot [1] was generated decomposing the timeseries vector of the raw data to visually analyze if any trends, seasonality's or randomness exist



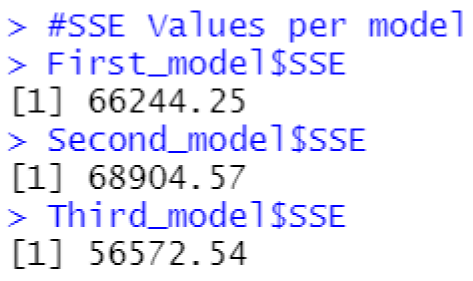
Plot [1]

Based on the decomposed data the linearity in the trend chart suggests that the data has no evident trends but does show clear effects of Seasonality (based on seasonal chart) and a lot of noise due to randomness (based on random chart).

To prepare the data for CUSUM analysis to detect if the unofficial summer had gotten later in the 20years, the data was then passed through HoltWinters model. Three iterations were generated:

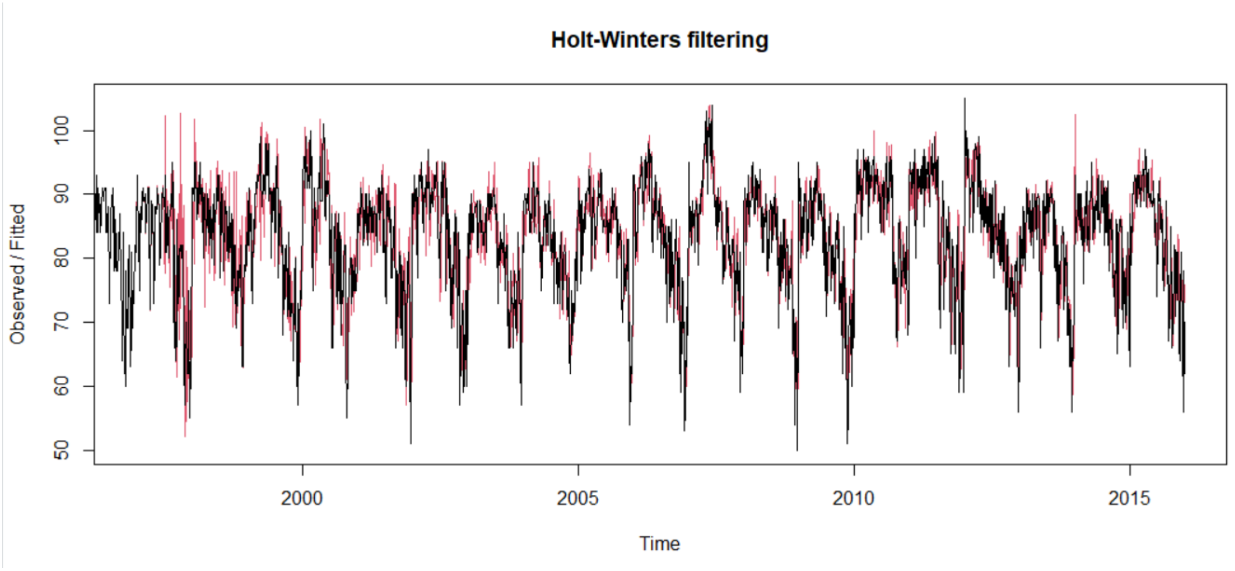
i) First_model=Triple Exponential Smoothing using additive type seasonality
ii) Second_model=Triple Exponential Smoothing using multiplicative type seasonality
iii) Third_model=Double Exponential Smoothing assuming no seasonality

Each model was then checked for the Least Sum of Squared Errors (SSE) to select the appropriate model



```
> #SSE Values per model
> First_model$SSE
[1] 66244.25
> Second_model$SSE
[1] 68904.57
> Third_model$SSE
[1] 56572.54
```

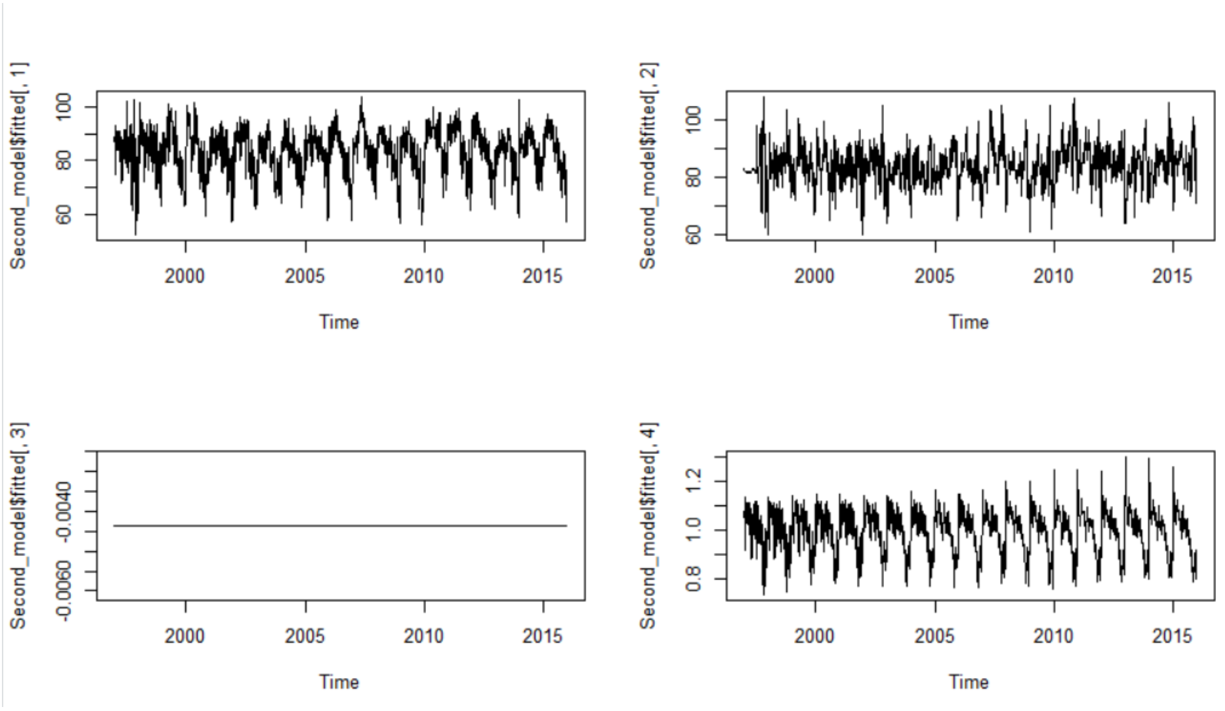
Based on the SSE data the Third_model seems to be the ideal fit for smoothing the data. However, Third_model assumes no seasonality in the data set. Based on the visual analysis done in Plot[1], this seems to be counterintuitive. I was unable to find out why the model would generate the lowest SSE but since the seasonality in the dataset was very evident I chose Second_model to smooth the data. Plot[2] shows the observed (original) dataset in black Vs the Fitted (Second_Model Prediction) dataset in red



Plot[2]

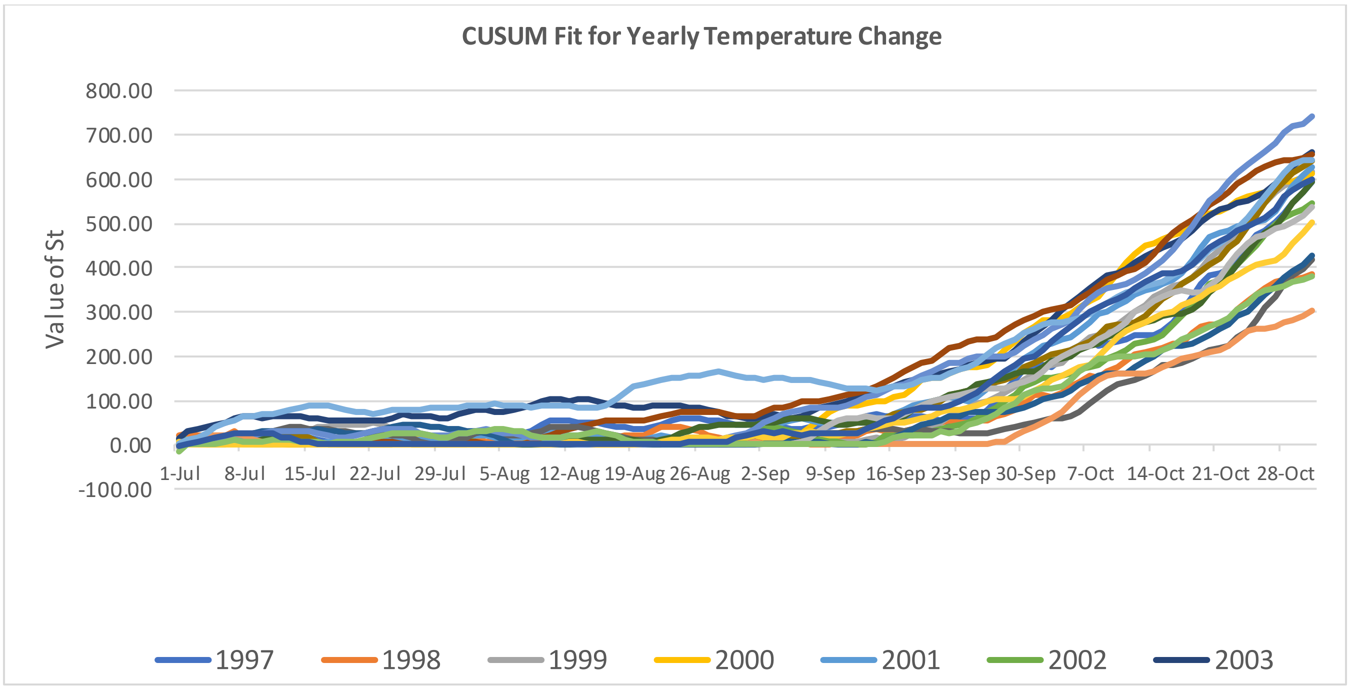
We notice that the predicted values are noisy in the beginning of time near 1997 but the model gets a better fit as it gets more data points to train along the timeline from 1997 to 2015. We also notice between time period 1997 to 1998 the model does not predict any values. This is due to seasonality. Since the model requires one complete cycle to finish before it can detect any patterns/seasons, it starts predicting values from the second cycle.

The fitted values from Second_model were then plotted to check for randomness, trends and seasonality individually in Plot[3] which shows that we see no trends (Plot[3]: [,3]) and an evident seasonality (Plot[3]: [,4]). There still exists some randomness in the prediction (Plot[3]: [,2]), however ever dataset will have some element of randomness to it.



Plot[3]

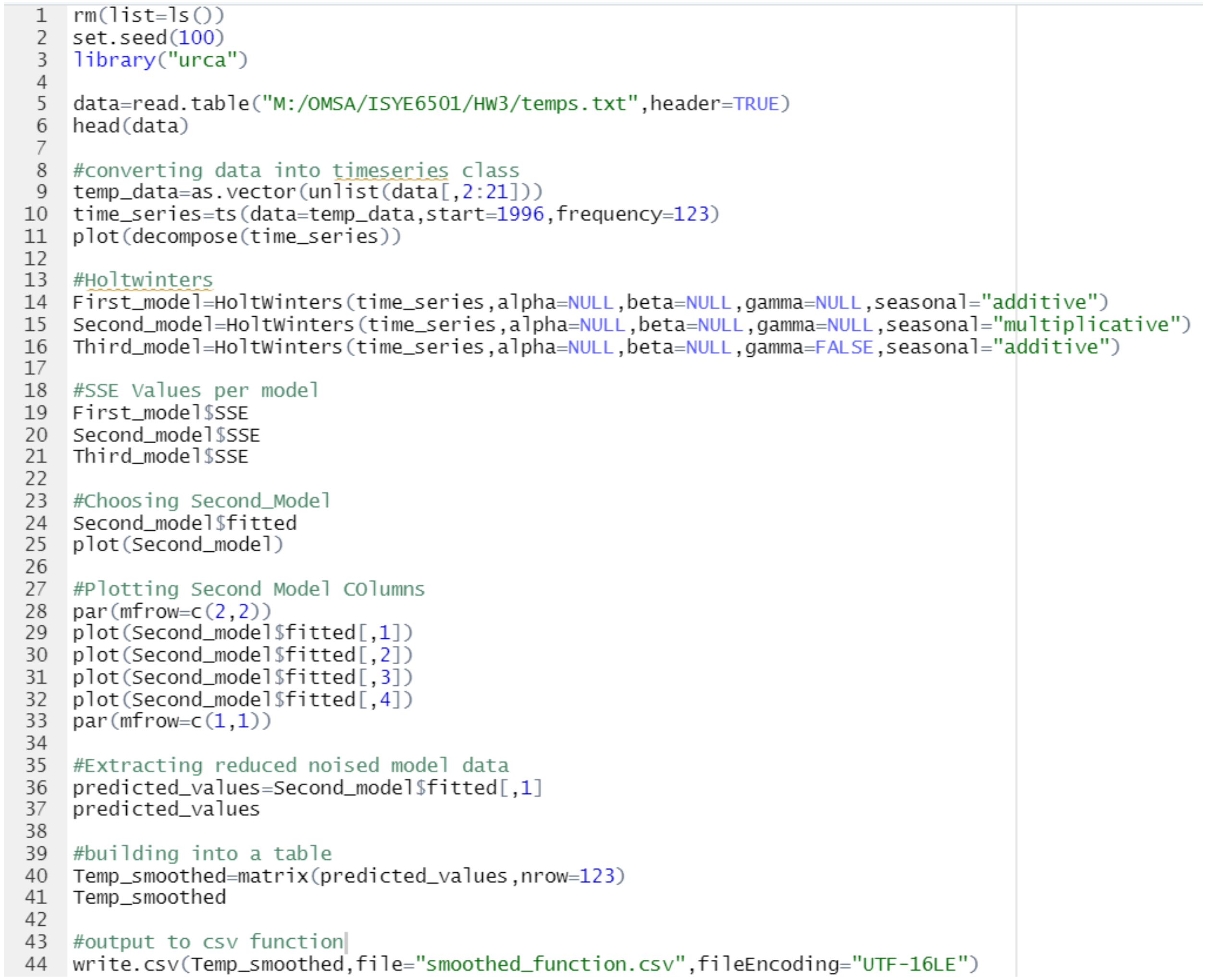
The fitted values from the Second_model were exported in a csv file and then CUSUM method was used to Plot[4] using a Mu value of 88.3, standard deviation measured was 6.45, c 1.6 and a T of 38.78 (6 times the standard deviation). CUSUM method was used to calculate St values per year and then plotted together to see if any direct correlation was graphically observed on the unofficial end of summer being late in the past 19 year(1 year data was never predicted due to cyclical nature of model prediction)



Plot[4]

Based on Plot[4] there did not seem to be any delay in unofficial end of summer being late. This could be due to the fact that the selected parameters while doing CUSUM analysis heavily weigh on the outcome of the analysis.

R Code used for analysis:



```r
rm(list=ls())
set.seed(100)
library("urca")

data=read.table("M:/OMSA/ISYE6501/HW3/temps.txt",header=TRUE)
head(data)

#converting data into timeseries class
temp_data=as.vector(unlist(data[,2:21]))
time_series=ts(data=temp_data,start=1996,frequency=123)
plot(decompose(time_series))

#Holtwinters
First_model=HoltWinters(time_series,alpha=NULL,beta=NULL,gamma=NULL,seasonal="additive")
Second_model=HoltWinters(time_series,alpha=NULL,beta=NULL,gamma=NULL,seasonal="multiplicative")
Third_model=HoltWinters(time_series,alpha=NULL,beta=NULL,gamma=FALSE,seasonal="additive")

#SSE Values per model
First_model$SSE
Second_model$SSE
Third_model$SSE

#Choosing Second_Model
Second_model$fitted
plot(Second_model)

#Plotting Second Model COlumns
par(mfrow=c(2,2))
plot(Second_model$fitted[,1])
plot(Second_model$fitted[,2])
plot(Second_model$fitted[,3])
plot(Second_model$fitted[,4])
par(mfrow=c(1,1))

#Extracting reduced noised model data
predicted_values=Second_model$fitted[,1]
predicted_values

#building into a table
Temp_smoothed=matrix(predicted_values,nrow=123)
Temp_smoothed

#output to csv function
write.csv(Temp_smoothed,file="smoothed_function.csv",fileEncoding="UTF-16LE")
```