DEMO 4 - ON TIME SERIES USE-CASES

QUESTION 1 - REAL-LIFE USE-CASE WARRANTING THE USE OF EXPONENTIAL SMOOTHING

I recently read this brilliant blog written by one of the engineers designing website traction models at Automattic (they are the creators of WordPress), to help their customers design websites that perform better.

They used exponential smoothing and a well-parameterized ARIMA model due to seasonal changes in shopping patterns caused by holidays and festivals months.

Data needed ~

1. Date
2. Price of item
3. Income range
4. Purchase of holiday/festival item on that date

Alpha Estimate ~

They also used several other factors associated with the UI and marketing score of the website in question, and the package that the website was included in.

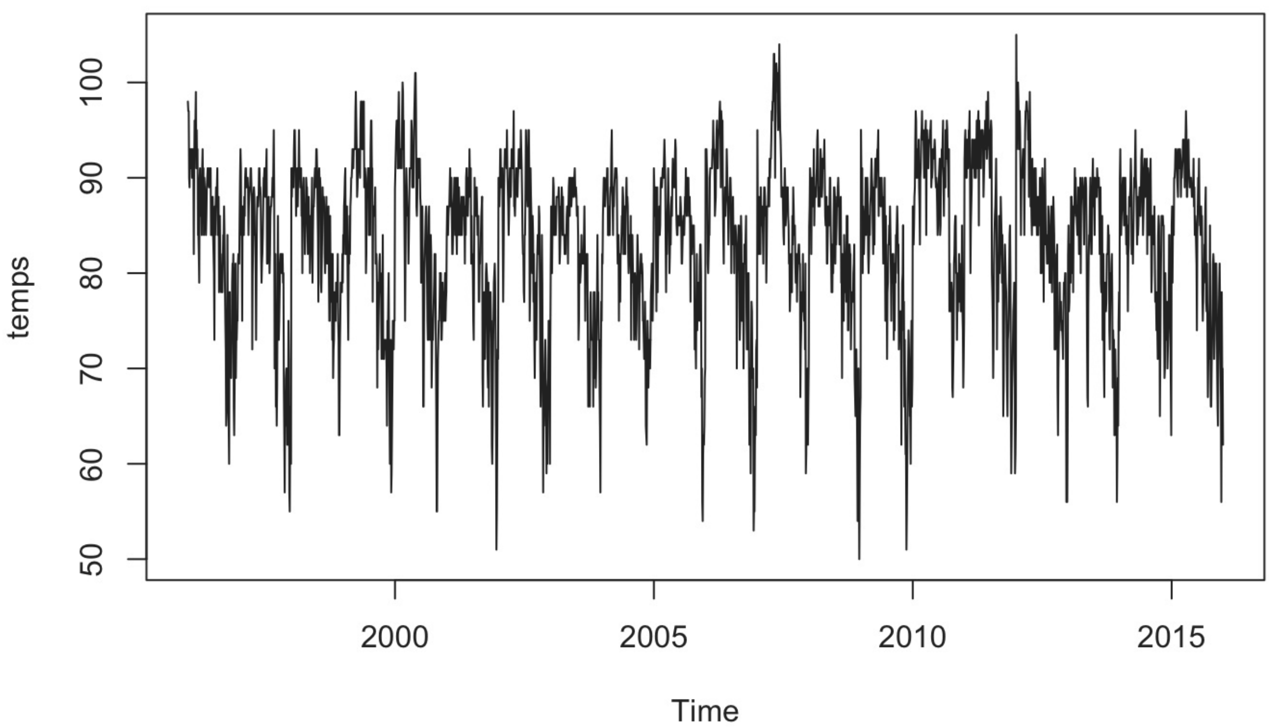
The randomness of shopping within certain holiday time periods (or prior to it) is high since some of the populace may finish their purchases much before, and some may do it just before, and some may do it at both times. Hence, for high random data, the value of alpha should be closer to zero. A good range would be between 0.1 and 0.4.

QUESTION 2 - TEMPERATURE OVERVIEW USING EXPONENTIAL SMOOTHING

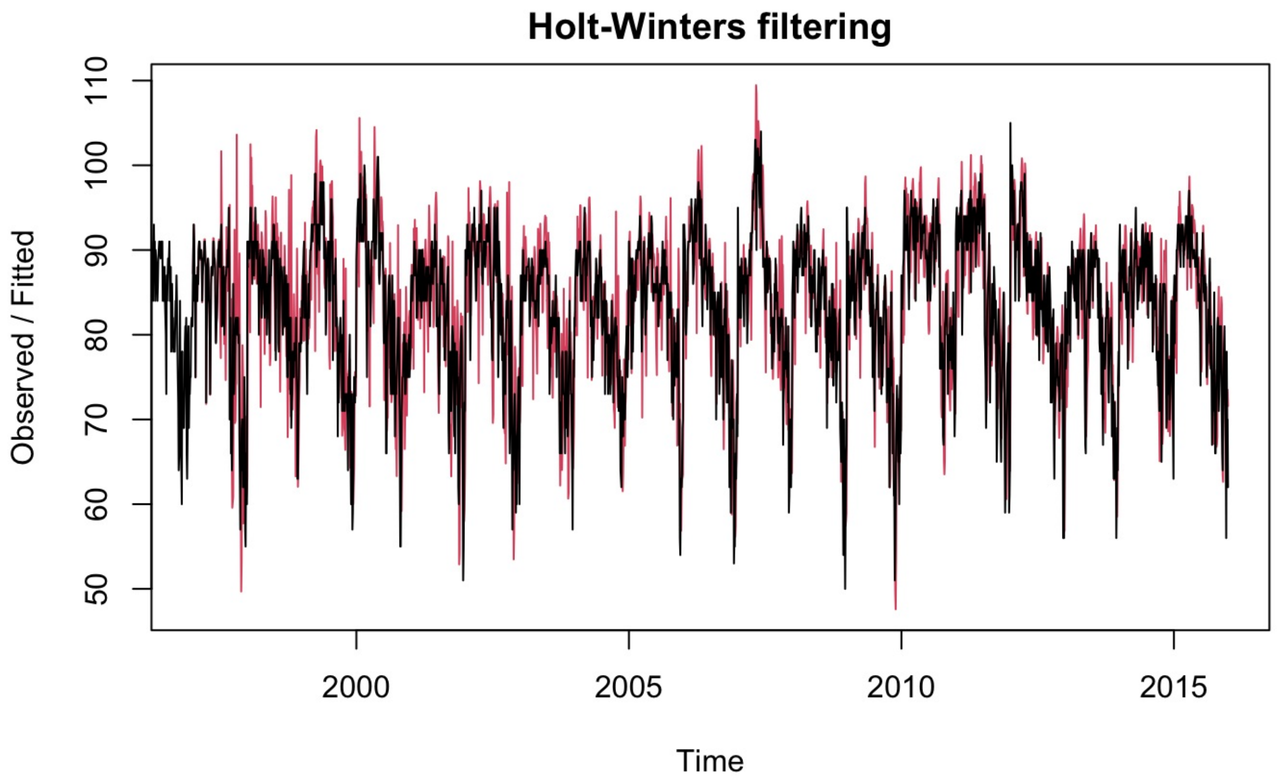
Step-By-Step Analysis ~

1. Extract the data into tabular format and get a good idea of how the data looks like by checking some of the samples.

2. We are looking at temperatures during the same periods over many years and we already know from climate sciences that the temperatures are seasonal with a cyclicity of twelve months. However, to see all these temperatures as time-series data over the years, we will need to unbind it from it's yearly form and plot it as a continuous series of temperatures. We get this when we do that.



1. After this comes the exponential smoothing fit that we apply using the Holts-Winters function. We experiment with different values for the parameters. **Based on a qualitative study performed on randomness in temperatures from day-to-day (as in, not across trends or seasons), the alpha value is fixed between 0.3 to 0.6.** The beta and gamma parameters have been kept at a default value of 0.1 in terms of **multiplicative** factor.



1. After this we perform a data transformation to matrices, assess the mean temperatures for each year, and build a model using the original CUSUM method for change detection, and the smoothened, and actual data values. And we look at the results of when the summer ended each year. **After taking the first year for baseline calculation and for an alpha value of 0.5, we get these results -**

   Year Day

   X1997 30-Sep
   X1998 1-Oct
   X1999 1-Oct
   X2000 1-Oct
   X2001 2-Oct
   X2002 2-Oct
   X2003 3-Oct
   X2004 3-Oct
   X2005 4-Oct
   X2006 4-Oct

   X2007 5-Oct
   X2008 5-Oct
   X2009 5-Oct
   X2010 5-Oct
   X2011 3-Oct
   X2012 3-Oct
   X2013 3-Oct
   X2014 4-Oct
   X2015 4-Oct

Explainability~

By varying the values of alpha, what was discovered was that the summers started getting shorter in all cases, but the earliest change and fastest change could be tracked from 2001 and it extended to 2014, with the 0.5 value giving the first change in day at year 2010.

The end of summer is the day when the seasonal factor falls below the baseline SF factor, which means that the weather is getting cooler. We could use cusum to see at which point in each year that threshold is fulfilled.

The results definitely answer the question that the summers have been getting longer over the years, and NOW we have a more constrained change period now that we used exponential smoothing.

Conclusion -

Alpha = 0.1 - 0.6

Beta = NULL - 0.1

Gamma = NULL - 0.1

seasonal = 'multiplicative'

T = 3 - 5

C = 0.5 - 1.5

Tested within these ranges, the results consistently show that the summers have been getting longer by a rate of approximately 1 day per three years.

In [ ]:

```
#Code for Exponential Smoothin followed by CUSUM modeling

df <- read.table('temps.txt', stringsAsFactors = FALSE, header = TRUE)

#overview of actual data - summary and plot
unbound <- as.vector(unlist(df[,2:21])) #making the data continuous
temps <- ts(data = unbound, frequency=123, start=1996)
summary(temps)
ts.plot(temps)

#fit data using holt-winters function - summary and plot
temps_norm <- HoltWinters(temps, alpha=0.1, beta=0.1, gamma=0.1, seasonal = "multiplicative")
summary(temps_norm)**Data needed ~**
plot(temps_norm)

#Data manipulation to matrices - actual and fitted AND assessment of mean temperature of each year
temps_fit <- matrix(temps_norm$fitted[,4], nrow=123)
head(temps_fit)

temps_smooth <- matrix(temps_norm$fitted[,1], nrow=123)
colnames(temps_fit) <- colnames(df[,3:21])
rownames(temps_fit) <- df[,1]

colnames(temps_smooth) <- colnames(df[,3:21])
rownames(temps_smooth) <- df[,1]

meanSFAllYears <- mean(temps_fit)
meanSFAllYears

meanSFEachYear <- vector()
for (i in 1:ncol(temps_fit)){
  meanSFEachYear[i] = mean(temps_fit[,i])
}
meanSFEachYear

'''
Mean overall -
0.9954727

Mean for each year -
1.0000000
0.9981882
0.9980547
0.9975095
0.9971271
0.9961954
0.9955108
0.9957393
0.9956360
0.9949667
0.9948162
0.9946495
0.9943300
0.9941211
0.9940949
0.9933907
0.9932476
0.9935215
0.9928824
'''

#Checking baseline factor (seasonal factor for year 1996)
meanSF_firstYear <- mean(temps_fit[,1])

#Building the CUSUM model based on first year
cusum_decrease = function(data, mean, T, C){
  results = list()
  cusum = 0
  rowCounter = 1
  while (rowCounter <= nrow(data)){
    current = data[rowCounter,]
    cusum = max(0, cusum + (mean - current - C))
    # print(cusum)
```

```
    if (cusum >= T) {
      results = rowCounter
      break
    }
    rowCounter = rowCounter + 1
    if (rowCounter >= nrow(data)){
      results = NA
      break
    }
  }
  return(results)
}

# T and C
C_var = sd(temps_fit[,1])*1.5
T_var = sd(temps_fit[,1])*3

result_vector = vector()
for (col in 1:ncol(temps_fit)){
  result_vector[col] = cusum_decrease(data = as.matrix(temps_fit[,col]), mean = 1,T = T_var, C = C_var)
}

result_df = data.frame(Year = colnames(temps_fit),Day = df[result_vector,1])
result_df
```

THE END--
