Question 7.1 Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of α (the first smoothing parameter) to be closer to 0 or 1, and why?

Answer:

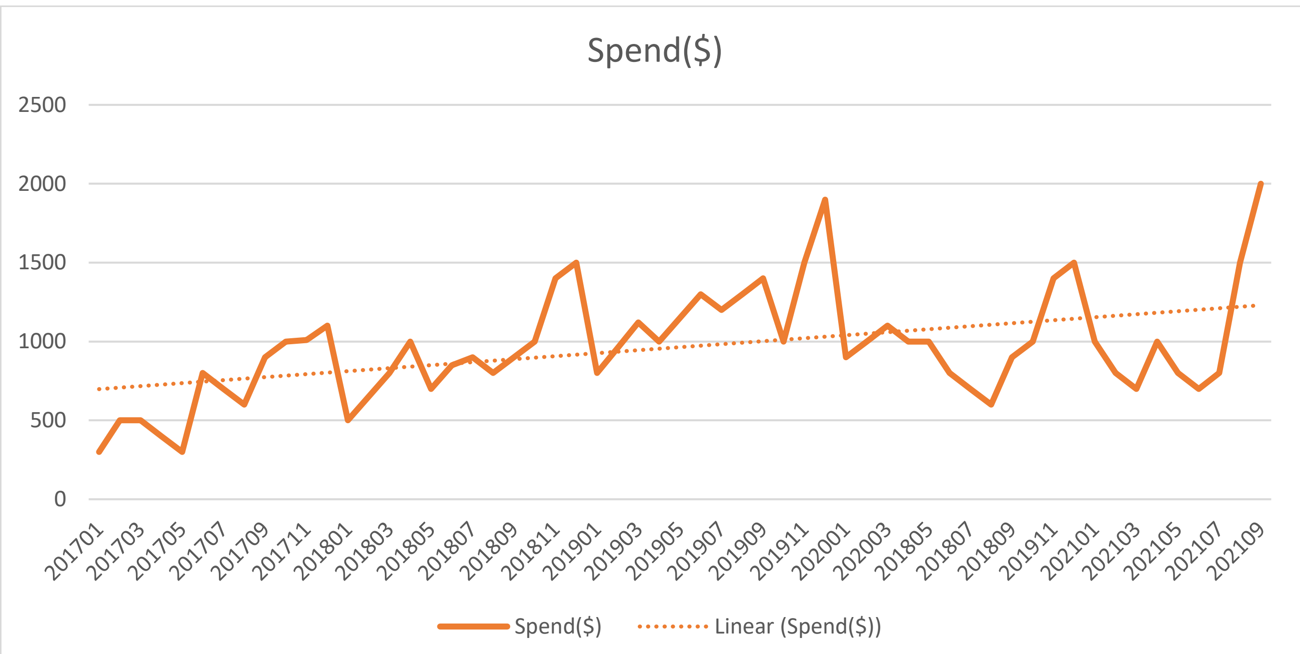
Understand the baseline pattern of my online spending using exponential smoothing.

Data: Time series data with history of 5+ years at monthly interval. That will give me over 60 data points as initial data set to work with.

Alpha selection: As this is related to my spending habits, I will choose the alpha "α" value closer to "1" to adding more weightage to the baseline spending pattern and NOT to the randomness of the current data point. I would like to expand the simple exponential model to accommodate Trend as I see an increasing trend with the spending online.

Seasonality: My spending is going to exhibit cyclic effects, more spending during Christmas and Holiday season, closer to family birthdays and summer vacations. For sake of simplicity, I will choose my cyclic Length as Quarterly spikes (L=3 months). I can add forecast of upcoming month as exponential smoothing is good for short term forecast

Based on a sample dataset I put together in excel, here is how the observed data will look like and it shows trending and seasonality



Question 7.2Using the 20 years of daily high temperature data for Atlanta (July through October) from Question 6.2 (filetemps.txt), build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years. (Part of the point of this assignment is for you to think about how you might use exponential smoothing to answer this question. Feel free to combine it with other models if you'd like to. There's certainly more than one reasonable approach.)

Answer:

I have broken this question into two part (1) Establish an exponential smooth (2) CUSUM to Change detect the summer end

Part#1: Summary of Approach for Exponential Smoothing

1) Prep the data to a time series design to visualize the peaks and valleys of the original data
2) Apply HoltSmooth function
   Obtain the Alpha(smoothing paramters), Beta(Trend) &Gamma(Seasonality) (As Alpha is close to 1, forecast are based on not much randomness in system and depends more on the current data point and not on baseline

3) Obtain the Accuracy (Sum of Squared Errors SSE)
4) Visualize the Smoothed data as line plot (The plot shows the original time series in black, and the forecasts as a red line)
5) Forecast Temp for next 30 points in year 2016 (This was not the ask; I was interested to understand how forecast looks in exponential smoothing)

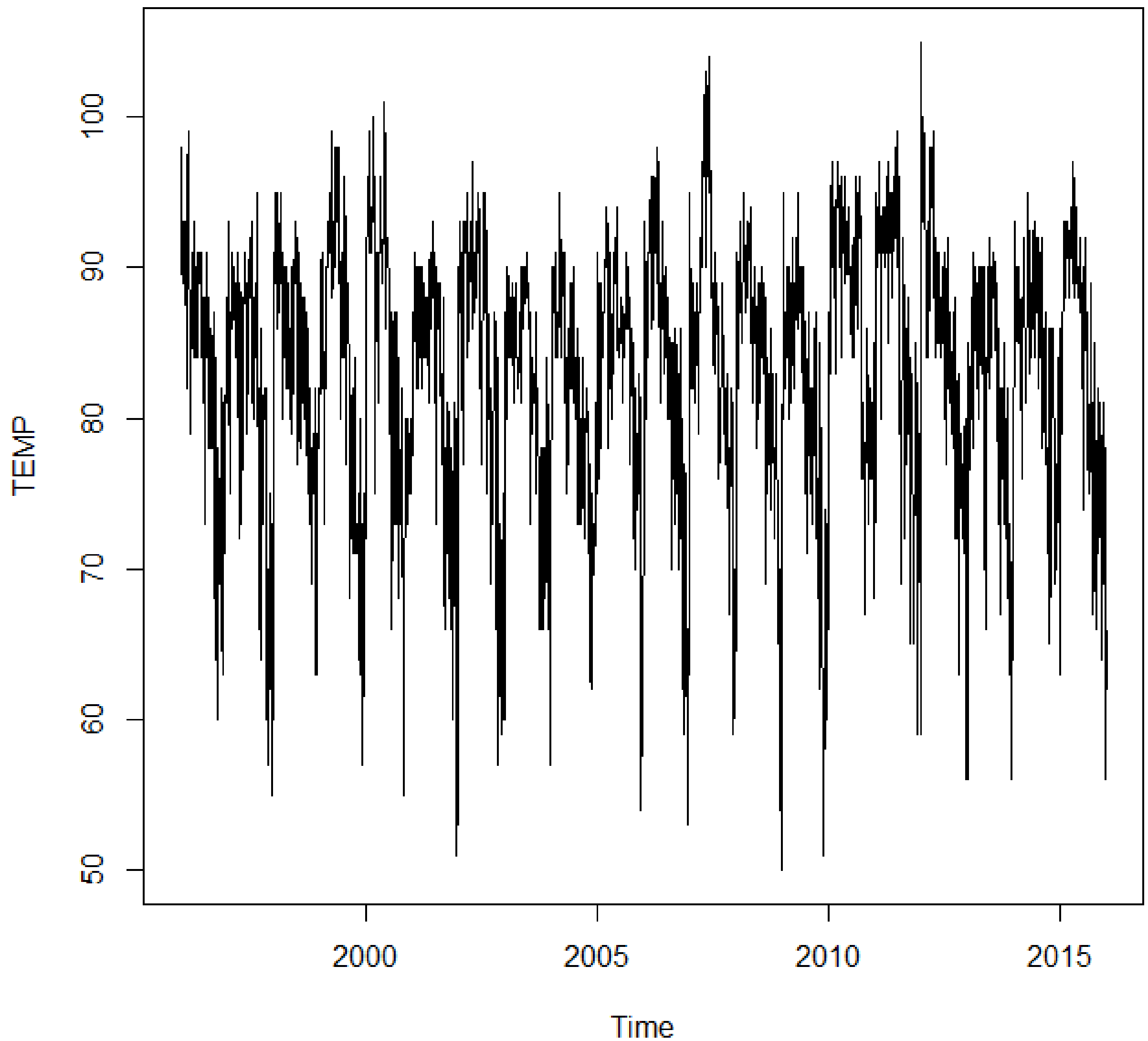
Step#1: Ingest data

```
##########CLEAR##########
rm(list = ls())

##########LIBRARY##########
#install.packages("TSstudio")
library(TSstudio) #Plot time series data
library(zoo)
library("dplyr") # Summarize data



##########INGEST FILE##########

data<- read.table("6_2temps.txt",header=TRUE,stringsAsFactors = FALSE,sep="\t")

head(data,10)
     DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006
   1-Jul    98    86    91    84    89    84    90    73    82    91    93
   2-Jul    97    90    88    82    91    87    90    81    81    89    93
   3-Jul    97    93    91    87    93    87    87    87    86    86    93
   4-Jul    90    91    91    88    95    84    89    86    88    86    91
   5-Jul    89    84    91    90    96    86    93    80    90    89    90
   6-Jul    93    84    89    91    96    87    93    84    90    82    81
   7-Jul    93    75    93    82    96    87    89    87    89    76    80
   8-Jul    91    87    95    86    91    89    89    90    87    88    82
   9-Jul    93    84    95    87    96    91    90    89    88    89    84
.0 10-Jul    93    87    91    87    99    87    91    84    89    78    84
  X2007 X2008 X2009 X2010 X2011 X2012 X2013 X2014 X2015
     95    85    95    87    92   105    82    90    85
     85    87    90    84    94    93    85    93    87
     82    91    89    83    95    99    76    87    79
     86    90    91    85    92    98    77    84    85
     88    88    80    88    90   100    83    86    84
     87    82    87    89    90    98    83    87    84
     82    88    86    94    94    93    79    89    90
     82    90    82    97    94    95    88    90    90
     89    89    84    96    91    97    88    90    91
.0   86    87    84    90    92    95    87    87    93
```

Step#2 : Prep data into Time series

By End of this step, the data for all the years will be organized as time series data to visualize the trending of years in the line plot

```
>
> ##########DATA PREP FOR TIME SERIES ANALYSIS##########
>
> TEMP<- read.table("6_2temps.txt",header=TRUE,stringsAsFactors = FALSE,sep="\t") %>%
+ dplyr::select (., -DAY) %>% # every field expect DAY
+         unlist() %>%
+         as.vector()  %>%# create vector of timeseries years one after another
+         ts(start = 1996, frequency = 123)
>
> head(TEMP,10)
 [1] 98 97 97 90 89 93 93 91 93 93
>
> plot(TEMP) # Plot original data by years
>
```



Step#3 : smooth data using Winter Holt function

As Alpha is close to 1, forecast are based on not much randomness in system and depends more on the current data point and not on baseline

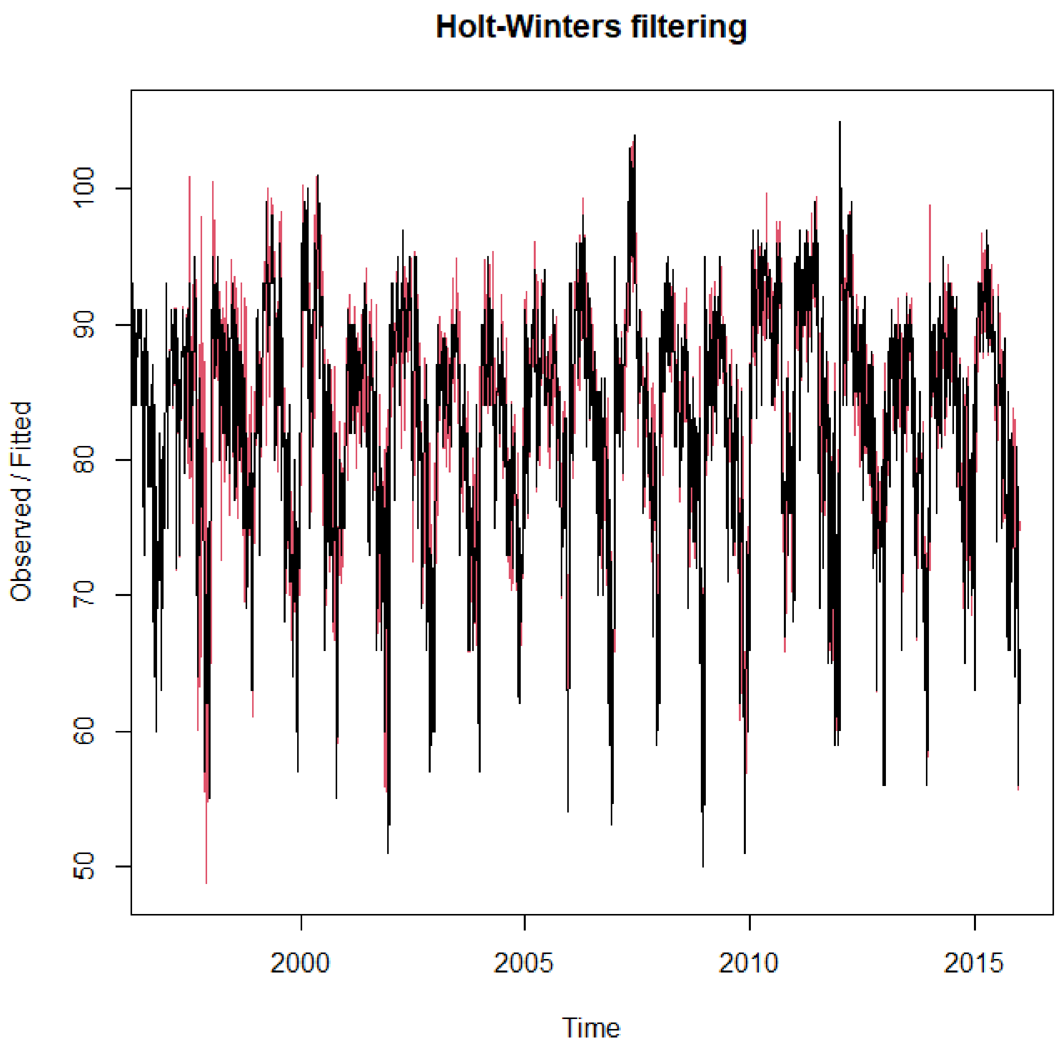
```
>
> #Coefficients for HoltWinters
> holtsmooth
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = TEMP)

Smoothing parameters:
 alpha: 0.6610618
 beta : 0
 gamma: 0.6248076
```

Print the Parameters to see the smoothing factor, trends & seasonality

```
> #print the parameters
> print(paste0('Accuracy(Sum of Squared Errors): ', holtsmooth$SSE))
[1] "Accuracy(Sum of Squared Errors): 66244.2504058465"
>
> #As Alpha is close to 1, forecast are based on not much randomness in system and depends more on the current$
> print(paste0('Holt Winters smoothing(Alpha): ', holtsmooth$alpha))
[1] "Holt Winters smoothing(Alpha): 0.661061754684708"
>
> print(paste0('Holt Winters Trend(Beta): ', holtsmooth$beta))
[1] "Holt Winters Trend(Beta): 0"
> print(paste0('Holt Winters Seasonality(Gamma): ', holtsmooth$gamma))
[1] "Holt Winters Seasonality(Gamma): 0.624807621487671"
```

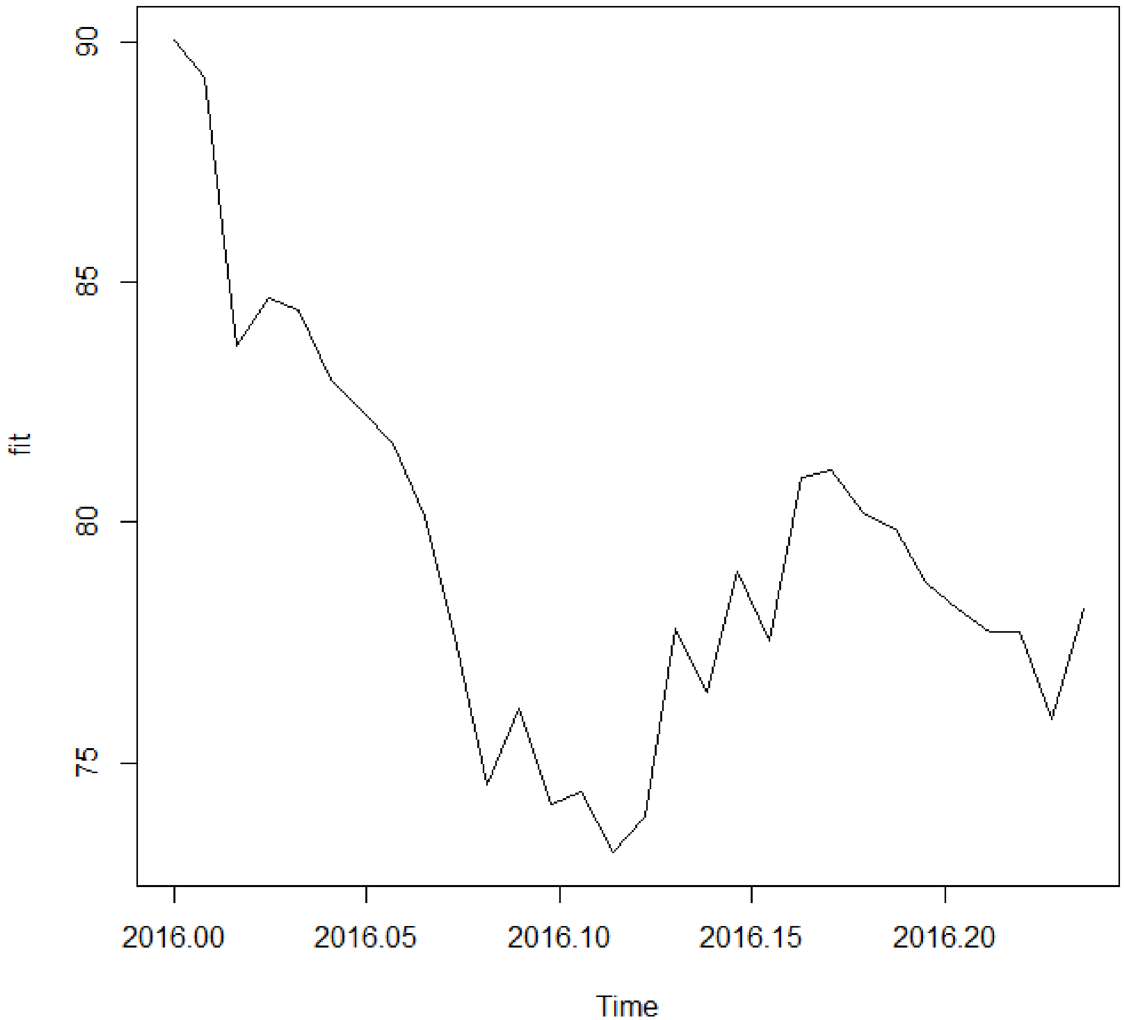
Step#4: Plot the results of Winter Hold smoothing (The plot shows the original time series in black, and the forecasts as a red line.)

```
#Plot the Holt Winters results
plot(holtsmooth) #The plot shows the original time series in black, and the forecasts as a red line.
```



Step#5: predict for 30 more data points

```
>
> #predict future Temp
> predictmn <- predict(holtsmooth, n.ahead=30)
> plot(predictmn)
>
```



Part#2: Summary of Approach to identify if the summer has gotten later in 20 years :

The difference between this assignment vs. previous assignment is we are identify if the summer has got later with the years vs. earlier winter onset in previous assignment.

Hence for change detection, I am using CUSUM approach to identify a temperature drop. I am using Threshold of "82" as end of summer indicator.

Summary of Approach

1) Follow the CUSUM process and obtain the change detection for each year
2) Since the data was same as our previous assignment, I have reused the confidence interval "C"=4 to not have an overly sensitive system that detects random temperature drops during summer.
3) Using this change detected data point for each year , visualize the graph with trend to observe if the statement "summer has got later" is true

   (I plotted a graph to show the summer end dates for each year with year in x-axis and day of the month(As all end dates fall in Oct) to find a pattern. My summarize conclusion

   - **End dates was later in the year from 2010 to 2014. However, 2015 had an early summer end.**
   - **During decades 2000-2010 , there is a slight shift to summer end date to later part of the year**
   - **Looking at trend lines, the upward trend indicating summer moved to a later time is not evident. With that in consideration, I conclude that <u>the summer end dates has NOT gotten later</u>**

)

   Please refer below for details

Step#1: Setting up threshold and confidence value; setup x(t) for each year

```
##########CUSUM to predict if SUMMER ended later in the last 20 years##########

# set up C value
C <- 4
#SET UP Threshold for Temp drop  set to 82.
```

```
>
> #Extract each year
> date_avg96 <- data$X1996
> date_avg97 <- data$X1997
> date_avg98 <- data$X1998
> date_avg99 <- data$X1999
> date_avg00 <- data$X2000
> date_avg01 <- data$X2001
> date_avg02 <- data$X2002
> date_avg03 <- data$X2003
> date_avg04 <- data$X2004
> date_avg05 <- data$X2005
> date_avg06 <- data$X2006
> date_avg07 <- data$X2007
> date_avg08 <- data$X2008
> date_avg09 <- data$X2009
> date_avg10 <- data$X2010
> date_avg11 <- data$X2011
> date_avg12 <- data$X2012
> date_avg13 <- data$X2013
> date_avg14 <- data$X2014
> date_avg15 <- data$X2015
```

Step#2 : Establish CUSUM for each year

1. **Calculate mean and derive standard baseline**

```
> #calculate mean
> mean_x_t96 <- mean(x_t96)
> mean_x_t97 <- mean(x_t97)
> mean_x_t98 <- mean(x_t98)
> mean_x_t99 <- mean(x_t99)
> mean_x_t00 <- mean(x_t00)
> mean_x_t01 <- mean(x_t01)
> mean_x_t02 <- mean(x_t02)
> mean_x_t03 <- mean(x_t03)
> mean_x_t04 <- mean(x_t04)
> mean_x_t05 <- mean(x_t05)
> mean_x_t06 <- mean(x_t06)
> mean_x_t07 <- mean(x_t07)
> mean_x_t08 <- mean(x_t08)
> mean_x_t09 <- mean(x_t09)
> mean_x_t10 <- mean(x_t10)
> mean_x_t11 <- mean(x_t11)
> mean_x_t12 <- mean(x_t12)
> mean_x_t13 <- mean(x_t13)
> mean_x_t14 <- mean(x_t14)
> mean_x_t15 <- mean(x_t15)
>
>
>
>
> mean_x_t96
[1] 83.71545
> mean_x_t97
[1] 81.6748
> mean_x_t98
[1] 84.26016
> mean_x_t99
[1] 83.35772
```

2. **Use CUSUM detect decrease to detech change to temperature**

```
> # as we are seeing decrease in temperature, we calculate mean - data
>
> mean_data96 <- mean_x_t96-date_avg96
> mean_data97 <- mean_x_t97-date_avg97
> mean_data98 <- mean_x_t98-date_avg98
> mean_data99 <- mean_x_t99-date_avg99
> mean_data00 <- mean_x_t00-date_avg00
> mean_data01 <- mean_x_t01-date_avg01
> mean_data02 <- mean_x_t02-date_avg02
> mean_data03 <- mean_x_t03-date_avg03
> mean_data04 <- mean_x_t04-date_avg04
> mean_data05 <- mean_x_t05-date_avg05
> mean_data06 <- mean_x_t06-date_avg06
> mean_data07 <- mean_x_t07-date_avg07
> mean_data08 <- mean_x_t08-date_avg08
> mean_data09 <- mean_x_t09-date_avg09
> mean_data10 <- mean_x_t10-date_avg10
> mean_data11 <- mean_x_t11-date_avg11
> mean_data12 <- mean_x_t12-date_avg12
> mean_data13 <- mean_x_t13-date_avg13
> mean_data14 <- mean_x_t14-date_avg14
> mean_data15 <- mean_x_t15-date_avg15
>
>
>
```

```
> # subtract C from the difference score
> s_t96 <- mean_data96 - C
> s_t97 <- mean_data97 - C
> s_t98 <- mean_data98 - C
> s_t99 <- mean_data99 - C
> s_t00 <- mean_data00 - C
> s_t01 <- mean_data01 - C
> s_t02 <- mean_data02 - C
> s_t03 <- mean_data03 - C
> s_t04 <- mean_data04 - C
> s_t05 <- mean_data05 - C
> s_t06 <- mean_data06 - C
> s_t07 <- mean_data07 - C
> s_t08 <- mean_data08 - C
> s_t09 <- mean_data09 - C
> s_t10 <- mean_data10 - C
> s_t11 <- mean_data11 - C
> s_t12 <- mean_data12 - C
> s_t13 <- mean_data13 - C
> s_t14 <- mean_data14 - C
> s_t15 <- mean_data15 - C
>
>
>
> cusum96 <- append(0, 0)
```

Step#3: For each datapoint, calculate the CUSUM for each year from 1996 -2015. As the length of all the year data point is same, I have used length of one data point to run a loop

```
> for (i in 1:length(s_t96))
+       {
+   ifelse(cusum96[i] + s_t96[i-1] > 0, cusum96[i+1] <- cusum96[i] + s_t96[i-1], cusum96[i+1] <- 0)
+   ifelse(cusum97[i] + s_t97[i-1] > 0, cusum97[i+1] <- cusum97[i] + s_t97[i-1], cusum97[i+1] <- 0)
+   ifelse(cusum98[i] + s_t98[i-1] > 0, cusum98[i+1] <- cusum98[i] + s_t98[i-1], cusum98[i+1] <- 0)
+   ifelse(cusum99[i] + s_t99[i-1] > 0, cusum99[i+1] <- cusum99[i] + s_t99[i-1], cusum99[i+1] <- 0)
+   ifelse(cusum00[i] + s_t00[i-1] > 0, cusum00[i+1] <- cusum00[i] + s_t00[i-1], cusum00[i+1] <- 0)
+   ifelse(cusum01[i] + s_t01[i-1] > 0, cusum01[i+1] <- cusum01[i] + s_t01[i-1], cusum01[i+1] <- 0)
+   ifelse(cusum02[i] + s_t02[i-1] > 0, cusum02[i+1] <- cusum02[i] + s_t02[i-1], cusum02[i+1] <- 0)
+   ifelse(cusum03[i] + s_t03[i-1] > 0, cusum03[i+1] <- cusum03[i] + s_t03[i-1], cusum03[i+1] <- 0)
+   ifelse(cusum04[i] + s_t04[i-1] > 0, cusum04[i+1] <- cusum04[i] + s_t04[i-1], cusum04[i+1] <- 0)
+   ifelse(cusum05[i] + s_t05[i-1] > 0, cusum05[i+1] <- cusum05[i] + s_t05[i-1], cusum05[i+1] <- 0)
+   ifelse(cusum06[i] + s_t06[i-1] > 0, cusum06[i+1] <- cusum06[i] + s_t06[i-1], cusum06[i+1] <- 0)
+   ifelse(cusum07[i] + s_t07[i-1] > 0, cusum07[i+1] <- cusum07[i] + s_t07[i-1], cusum07[i+1] <- 0)
+   ifelse(cusum08[i] + s_t08[i-1] > 0, cusum08[i+1] <- cusum08[i] + s_t08[i-1], cusum08[i+1] <- 0)
+   ifelse(cusum09[i] + s_t09[i-1] > 0, cusum09[i+1] <- cusum09[i] + s_t09[i-1], cusum09[i+1] <- 0)
+   ifelse(cusum10[i] + s_t10[i-1] > 0, cusum10[i+1] <- cusum10[i] + s_t10[i-1], cusum10[i+1] <- 0)
+   ifelse(cusum11[i] + s_t11[i-1] > 0, cusum11[i+1] <- cusum11[i] + s_t11[i-1], cusum11[i+1] <- 0)
+   ifelse(cusum12[i] + s_t12[i-1] > 0, cusum12[i+1] <- cusum12[i] + s_t12[i-1], cusum12[i+1] <- 0)
+   ifelse(cusum13[i] + s_t13[i-1] > 0, cusum13[i+1] <- cusum13[i] + s_t13[i-1], cusum13[i+1] <- 0)
+   ifelse(cusum14[i] + s_t14[i-1] > 0, cusum14[i+1] <- cusum14[i] + s_t14[i-1], cusum14[i+1] <- 0)
+   ifelse(cusum15[i] + s_t15[i-1] > 0, cusum15[i+1] <- cusum15[i] + s_t15[i-1], cusum15[i+1] <- 0)
+ }
>
```

Step#4: This process identifies the data point which exceeds the set threshold "T" =82

```
> which(cusum96 >= 82)
 [1] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
[21] 120 121 122 123 124
> which(cusum97 >= 82)
 [1] 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum98 >= 82)
[1] 116 117 118 119 120 121 122 123 124
> which(cusum99 >= 82)
 [1] 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123
[21] 124
> which(cusum00 >= 82)
 [1] 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121
[21] 122 123 124
> which(cusum01 >= 82)
 [1] 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum02 >= 82)
 [1] 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum03 >= 82)
 [1] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum04 >= 82)
 [1] 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum05 >= 82)
[1] 118 119 120 121 122 123 124
> which(cusum06 >= 82)
 [1] 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum07 >= 82)
[1] 118 119 120 121 122 123 124
> which(cusum08 >= 82)
[1] 117 118 119 120 121 122 123 124
> which(cusum09 >= 82)
 [1] 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum10 >= 82)
 [1]  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
[21] 119 120 121 122 123 124
> which(cusum11 >= 82)
 [1] 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum12 >= 82)
 [1] 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
> which(cusum13 >= 82)
[1] 118 119 120 121 122 123 124
```

Step#5 : Once I get all the data point, I use the first data point as this is the indicator of the summer end

```
> data96 <- data$X1996
> print(paste0('1996 summer end date: ',data[100,1]))
[1] "1996 summer end date: 8-Oct"
> print(paste0('1997 summer end date: ',data[112,1]))
[1] "1997 summer end date: 20-Oct"
> print(paste0('1998 summer end date: ',data[116,1]))
[1] "1998 summer end date: 24-Oct"
> print(paste0('1999 summer end date: ',data[104,1]))
[1] "1999 summer end date: 12-Oct"
> print(paste0('2000 summer end date: ',data[102,1]))
[1] "2000 summer end date: 10-Oct"
> print(paste0('2001 summer end date: ',data[110,1]))
[1] "2001 summer end date: 18-Oct"
> print(paste0('2002 summer end date: ',data[110,1]))
[1] "2002 summer end date: 18-Oct"
> print(paste0('2003 summer end date: ',data[109,1]))
[1] "2003 summer end date: 17-Oct"
> print(paste0('2004 summer end date: ',data[111,1]))
[1] "2004 summer end date: 19-Oct"
> print(paste0('2005 summer end date: ',data[118,1]))
[1] "2005 summer end date: 26-Oct"
> print(paste0('2006 summer end date: ',data[111,1]))
[1] "2006 summer end date: 19-Oct"
> print(paste0('2007 summer end date: ',data[118,1]))
[1] "2007 summer end date: 26-Oct"
> print(paste0('2008 summer end date: ',data[117,1]))
[1] "2008 summer end date: 25-Oct"
> print(paste0('2009 summer end date: ',data[110,1]))
[1] "2009 summer end date: 18-Oct"
> print(paste0('2010 summer end date: ',data[99,1]))
[1] "2010 summer end date: 7-Oct"
> print(paste0('2011 summer end date: ',data[105,1]))
[1] "2011 summer end date: 13-Oct"
> print(paste0('2012 summer end date: ',data[107,1]))
[1] "2012 summer end date: 15-Oct"
> print(paste0('2013 summer end date: ',data[118,1]))
[1] "2013 summer end date: 26-Oct"
> print(paste0('2014 summer end date: ',data[123,1]))
[1] "2014 summer end date: 31-Oct"
> print(paste0('2015 summer end date: ',data[98,1]))
[1] "2015 summer end date: 6-Oct"
>
```

Conclusion: I plotted a graph to show the summer end dates for each year with year in x-axis and day of the month(As all end dates fall in Oct) to find a pattern. From the graph below, here are the observation

1) **Summer end dates was later in the year from 2010 to 2014. However, 2015 had an early summer end.**
2) **During decades 2000-2010 , there is a slight shift to summer end date to later part of the year**

Looking at trend lines, the upward trend indicating summer moved to a later time is not evident. With that in consideration, I conclude that <u>the summer end dates has NOT gotten later</u>