HW3_ISYE6501

Question 5.1

Using crime data from the file `uscrime.txt` (http://www.statsci.org/data/general/uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html), test to see whether there are any outliers in the last column (number of crimes per 100,000 people). Use the `grubbs.test` function in the `outliers` package in R

For this question, I did some iterations.

The first Grubbs test I did was with 2 opposite outliers, the p value was 1, so I concluded that at least one of the extremes (tail or head) is not an outlier. Then we checked for the upper bound with 1993. The p-value in this case is 0.07, so this could potentially be an outlier. Next, I did a tail test with, and 342 came up as an hypothesis, the p-value is 1 so I will reject the hypothesis that 342 is an outlier.

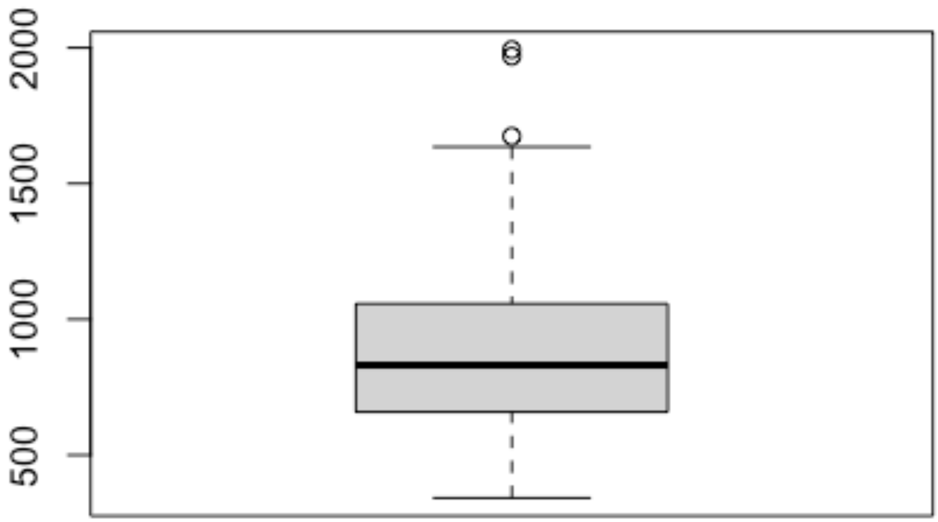
Then I removed 1993 to look for other outliers on the upper bound, 1969 came up and the p-value is 0.02848, so 1969 could also be an outlier. I removed it and tested the data set without 1969, nor 1993. The next value is 1674, the p-value is 0.1781>0.05, it is a high enough p-value to reject the alternative hypothesis. So I will stop here.

The visualization of the data shows that the 2 cities with the highest amount of crimes seem to be outliers but I am not confidence there is enough evidence to remove them from the dataset. Removing two cities , especially from a small data set like this could throw off our analysis completely. I would need to explore more around this to make a firm conclusion.

```
install.packages("outliers")
library(outliers)
library(ggplot2)
uscrime<-read.delim("uscrime.txt",stringsAsFactors = FALSE, header=TRUE)
head(uscrime)

##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Pr
ob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.0846
02
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.0295
99
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.0834
01
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.0158
01
```

```
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.0413
99
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.0342
01
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

```
#Boxplot of the crime column to visually determine if there any outliers
boxplot(uscrime$Crime,xlab="")
```



```
#We are only interested in the crime data
crime_data<-uscrime[,"Crime"]
#null hhypothesis : there are no outliers
#two tail test
grubbs.test(crime_data, type =11)
```

```
##
##  Grubbs test for two opposite outliers
##
```

```
## data:  crime_data
## G = 4.26877, U = 0.78103, p-value = 1
## alternative hypothesis: 342 and 1993 are outliers
```

```
#one tail test
grubbs.test(crime_data, type =10)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime_data
## G = 2.81287, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```

```
#one tail test with opposite tail excluding, 342
grubbs.test(crime_data, type =10,opposite = TRUE)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime_data
## G = 1.45589, U = 0.95292, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

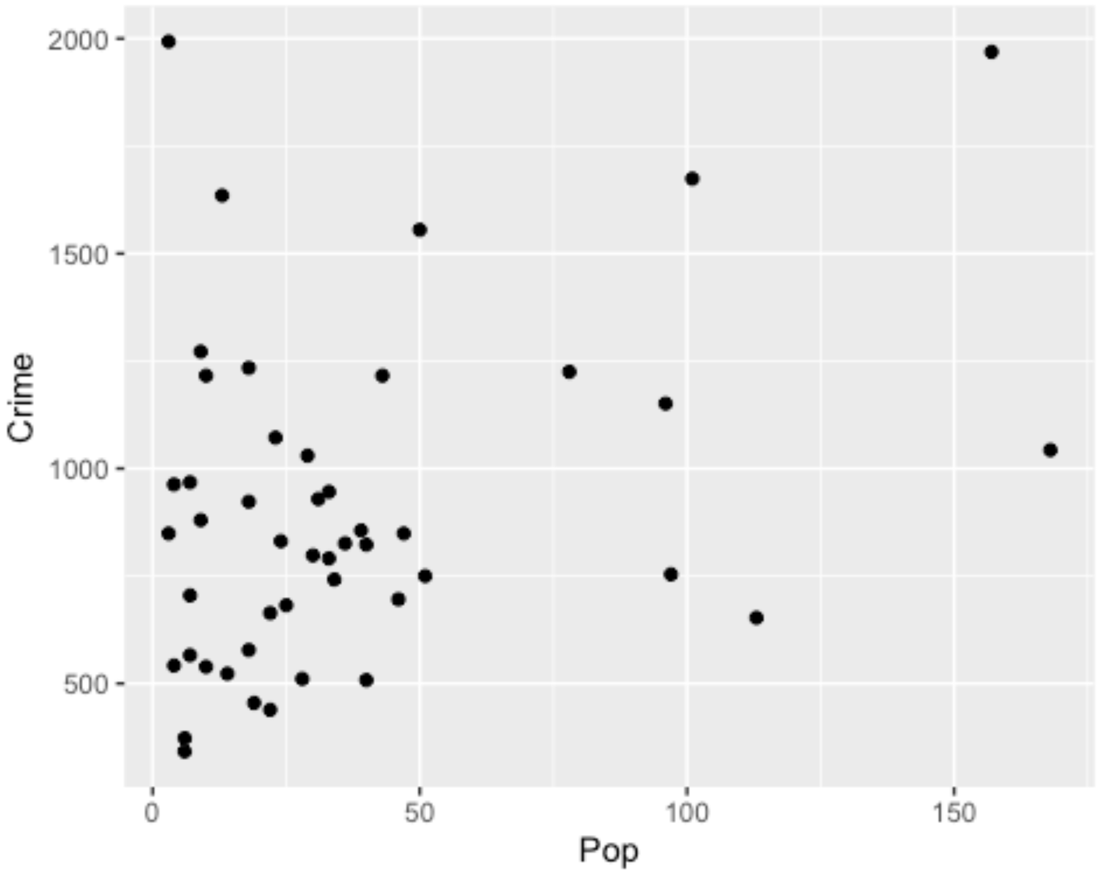
```
#one tail test, excluding 1993
crime_data2<-crime_data[-which.max(crime_data)]
grubbs.test(crime_data2, type =10)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime_data2
## G = 3.06343, U = 0.78682, p-value = 0.02848
## alternative hypothesis: highest value 1969 is an outlier
```

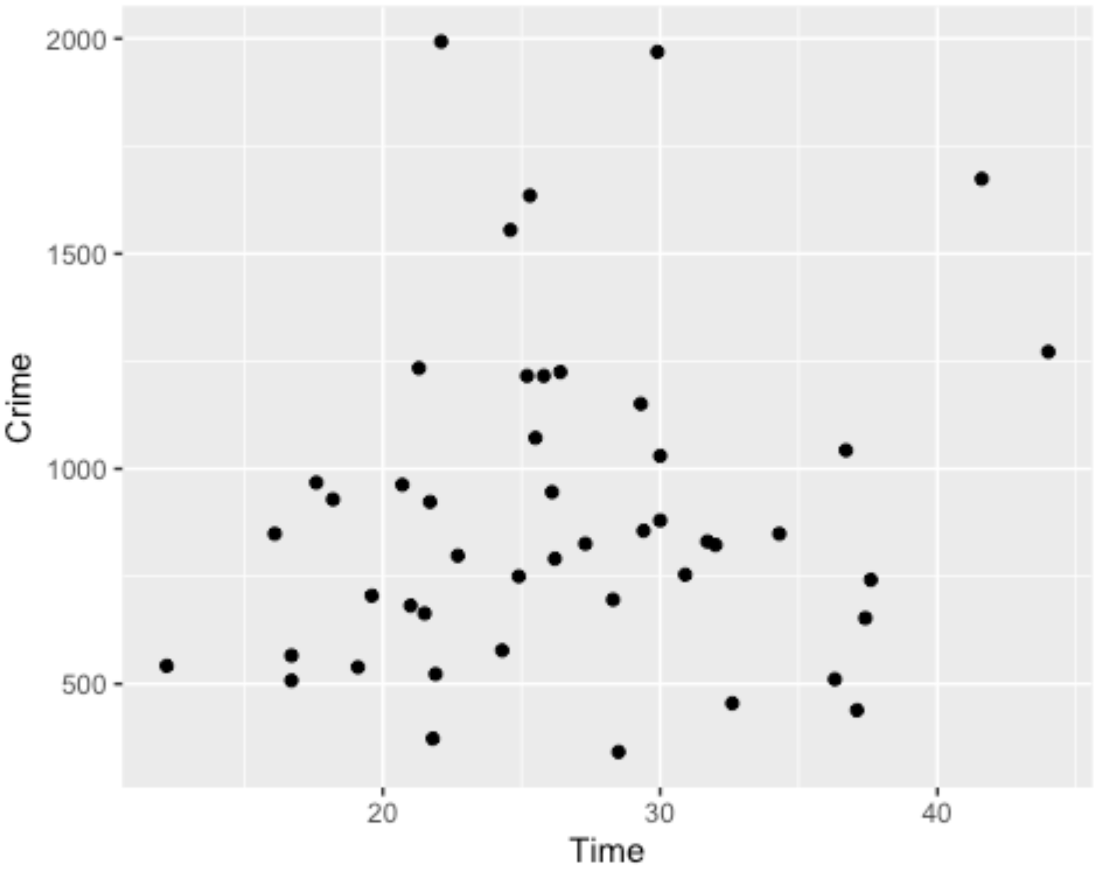
```
#one tail test, excluding 1969
crime_data3<-crime_data2[-which.max(crime_data2)]
grubbs.test(crime_data3, type =10)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime_data3
## G = 2.56457, U = 0.84712, p-value = 0.1781
## alternative hypothesis: highest value 1674 is an outlier
```

```
#Plotting Population vs crime to visualize if there is any outlier
ggplot(data=uscrime,mapping=aes(Pop, Crime))+geom_point()
```



```
#Plotting Crime as a function of time
ggplot(data=uscrime,mapping=aes(Time, Crime))+geom_point()
```



Question 6.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a Change Detection model would be appropriate. Applying the CUSUM technique, how would you choose the critical value and the threshold?

CUSUM technique would be great to monitor different relevant parameters for transformer bushings. Bushings are vital components in high-voltage equipment, they facilitate the passage of an energized, current-carrying conductor through the grounded tank of the transformer. Most bushings have a central conductor wound with alternating layers of paper insulation and conductive foil. When a capacitive layer shorts, the voltage across each layer increases, increasing the leakage current proportionally, which could cause serious damage to the transformer. The bushing monitoring system would continuously monitor the relative power factor change and the capacitance of the bushings and will detect partial discharge. Applying the CUSUM technique, the threshold T will be about 10%, a change of 5% is significant since we are dealing with high voltage so this will be a pretty big number. I would pick a low critical value, approximatively 10%, since we do not want the generator operating close to the threshold and the implications of this failure might be deadly.

Question 6.2

1. Using July through October daily-high-temperature data for Atlanta for 1996 through 2015, use a CUSUM approach to identify when unofficial summer ends (i.e., when the weather starts cooling off) each year. You can get the data that you need from the file `temps.txt` or online, for example at http://www.iweathernet.com/atlanta-weather-records or https://www.wunderground.com/history/airport/KFTY/2015/7/1/CustomHistory.html . You can use R if you'd like, but it's straightforward enough that an Excel spreadsheet can easily do the job too.

I approached tbis problem two ways:

My first approach is to look at the average of temperature of each day taken from 1996 to 2015. With this approach, my goal was to observe, in average, when the weather started cooling down in general. This hypothesis concluded that the last day of summer was October 8th .

```
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.1 ──

## ✓ tibble  3.1.6     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.1.0     ✓ forcats 0.5.1
## ✓ purrr   0.3.4

## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

#Reading the temperature data
temps_data<-read.table("temps.txt",header = TRUE,stringsAsFactors = FALSE,che
ck.names = FALSE)
head(temps_data)

##     DAY 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2
009
## 1 1-Jul   98   86   91   84   89   84   90   73   82   91   93   95   85
95
## 2 2-Jul   97   90   88   82   91   87   90   81   81   89   93   85   87
90
## 3 3-Jul   97   93   91   87   93   87   87   87   86   86   93   82   91
89
## 4 4-Jul   90   91   91   88   95   84   89   86   88   86   91   86   90
91
## 5 5-Jul   89   84   91   90   96   86   93   80   90   89   90   88   88
80
```

```
## 6 6-Jul   93   84   89   91   96   87   93   84   90   82   81   87   82   
87
##   2010 2011 2012 2013 2014 2015
## 1   87   92  105   82   90   85
## 2   84   94   93   85   93   87
## 3   83   95   99   76   87   79
## 4   85   92   98   77   84   85
## 5   88   90  100   83   86   84
## 6   89   90   98   83   87   84
```

```
names(temps_data)
```

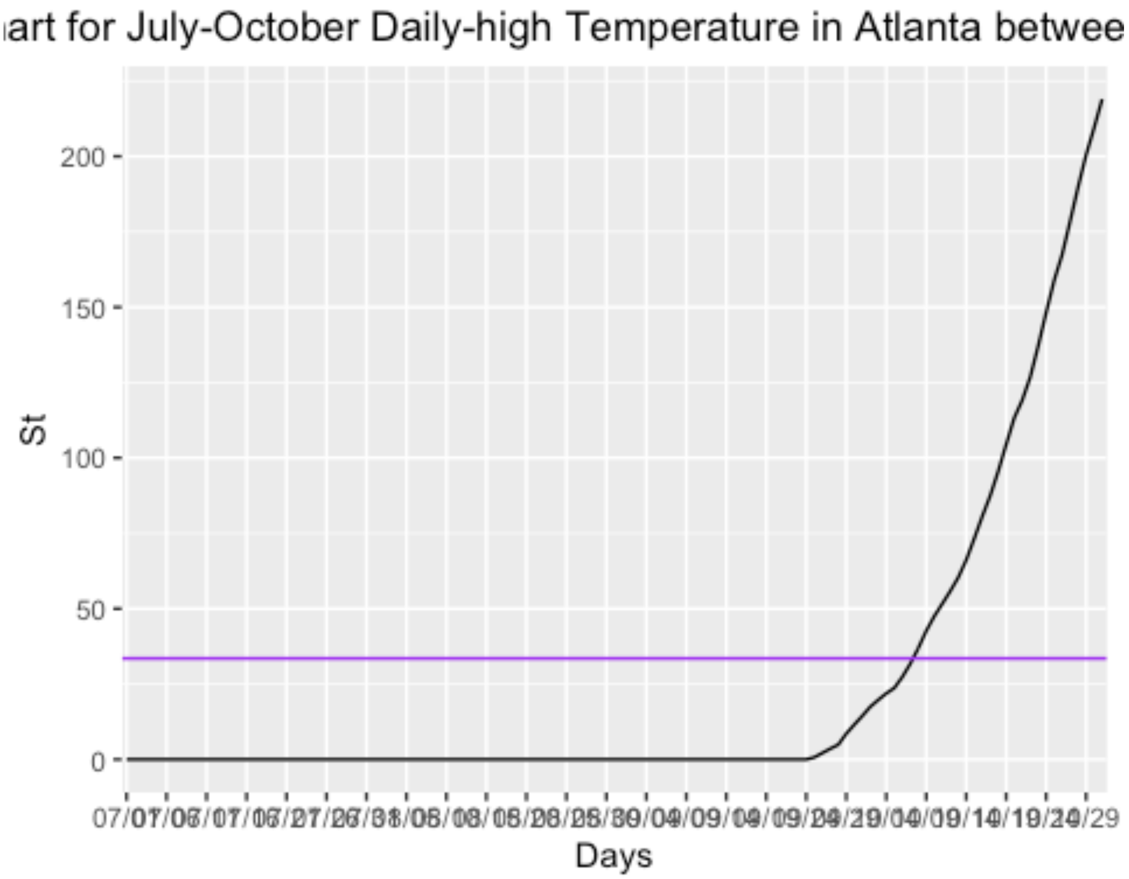
```
##  [1] "DAY"  "1996" "1997" "1998" "1999" "2000" "2001" "2002" "2003" "2004"
## [11] "2005" "2006" "2007" "2008" "2009" "2010" "2011" "2012" "2013" "2014"
## [21] "2015"
```

```
#Adding a column to the data, this column will hold the average value for eac
h day
temps_data<-cbind(temps_data,rowMeans(temps_data[,-1]))
#Renaming the column avg
colnames(temps_data)[ncol(temps_data)]<-"Avg"
head(temps_data)
```

```
##     DAY 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2
009
## 1 1-Jul   98   86   91   84   89   84   90   73   82   91   93   95   85   
95
## 2 2-Jul   97   90   88   82   91   87   90   81   81   89   93   85   87   
90
## 3 3-Jul   97   93   91   87   93   87   87   87   86   86   93   82   91   
89
## 4 4-Jul   90   91   91   88   95   84   89   86   88   86   91   86   90   
91
## 5 5-Jul   89   84   91   90   96   86   93   80   90   89   90   88   88   
80
## 6 6-Jul   93   84   89   91   96   87   93   84   90   82   81   87   82   
87
##   2010 2011 2012 2013 2014 2015   Avg
## 1   87   92  105   82   90   85 88.85
## 2   84   94   93   85   93   87 88.35
## 3   83   95   99   76   87   79 88.40
## 4   85   92   98   77   84   85 88.35
## 5   88   90  100   83   86   84 88.25
## 6   89   90   98   83   87   84 87.85
```

```
#Adding another column to hold values for variable St
temps_data[,"St"]<-NA
#Computing standard deviation for the average column of our data frame
std_temp<-sd(temps_data[,"Avg"])
std_temp
```

```
## [1] 6.701381

#mean is the average of the average column
mean_temp<-mean(temps_data[,"Avg"])
mean_temp

## [1] 83.33902

temps_data$Avg

##   [1] 88.85 88.35 88.40 88.35 88.25 87.85 87.10 89.15 90.05 88.55 87.95 88
.15
##  [13] 87.20 88.20 87.00 88.10 89.20 89.25 90.40 89.40 89.95 89.45 89.05 89
.10
##  [25] 88.00 89.50 89.55 89.95 89.25 89.55 88.15 88.55 88.65 89.55 90.30 91
.15
##  [37] 89.40 88.95 88.75 89.00 89.25 89.20 87.90 88.10 88.30 88.00 88.80 89
.05
##  [49] 90.15 90.30 89.30 89.10 89.40 88.40 87.85 86.50 88.45 87.60 87.15 88
.30
##  [61] 85.80 85.90 85.25 85.25 85.90 85.80 86.20 84.60 84.75 85.25 85.05 85
.25
##  [73] 85.55 85.30 83.10 83.65 83.70 82.25 81.85 81.70 82.40 83.00 81.60 81
.20
##  [85] 82.75 80.40 79.30 78.55 78.55 78.65 76.35 77.00 77.10 76.95 77.70 77
.85
##  [97] 78.20 76.35 75.60 74.80 74.25 75.15 75.85 75.80 75.45 74.20 72.90 72
.65
## [109] 73.10 71.90 71.05 71.25 74.10 72.35 69.65 68.85 69.35 71.40 68.90 68
.60
## [121] 69.35 71.05 70.50

#Chose C as standard deviation/2 and T as 5*standard deviation, I chose these
values of C and T because I am expecting some small noise, and looking at the
year-to-year average, there is about a 10-15 degrees difference between the m
ax and min. And in this case, I want to identify when summer ends, so I want
to be careful about mistakenly identifying some change as randomness.
C<-std_temp/2
C

## [1] 3.35069

T<-5*std_temp
T

## [1] 33.5069

#This is the loop to go through each row of the average column and compute St
temps_data[1,"St"]<-0
for(i in 2:nrow(temps_data)){

    temps_data[i,"St"]<-max(0,(temps_data[i-1,"St"]+mean_temp-temps_data[i,"A
vg"]-C))

}
temps_data$St

##   [1]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##   [7]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [13]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [19]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [25]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [31]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [37]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [43]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [49]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [55]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [61]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [67]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [73]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [79]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
##  [85]   0.000000   0.000000   0.688334   2.126668   3.565002   4.903336
##  [91]   8.541670  11.530004  14.418338  17.456672  19.745006  21.883340
##  [97]  23.671673  27.310007  31.698341  36.886675  42.625009  47.463343
## [103]  51.601677  55.790011  60.328345  66.116679  73.205013  80.543347
## [109]  87.431681  95.520015 104.458349 113.196683 119.085017 126.723351
## [115] 137.061685 148.200019 158.838353 167.426687 178.515020 189.903354
## [121] 200.541688 209.480022 218.968356

temps_l<-which(temps_data$St>T)
#this is the first day we noticed a change in trend
cat("The day a change in trend is detected is:",temps_data[which(temps_data$S
t>T),"DAY"][1])

## The day a change in trend is detected is: 8-Oct

#Frormating data to visualize it easily
temps_data[,"Date"]<-as.Date(temps_data[,"DAY"],"%d-%B")
temps_data[,"Date"]<-format(temps_data[,"Date"],format="%m/%d")


options(repr.plot.width=20, repr.plot.height=10)
ggplot(data = temps_data, aes(x = Date, y =`St`,group=2)) +
  geom_line()+
  geom_hline(yintercept=T,color="purple")+
  scale_x_discrete(breaks = unique(temps_data$Date)[seq(1,125,5)])+
  xlab("Days") +
  ylab("St") +
  ggtitle("CUSUM Chart for July-October Daily-high Temperature in Atlanta bet
ween 1996 and 2015")+
  theme(plot.title = element_text(hjust = 0.5))
```



My second approach was to look at CUSUM for each year individually and look for a trend:

```
temps_data2<-read.table("temps.txt",header = TRUE,stringsAsFactors = FALSE,check.names = FALSE)
Std_dev = array()
i=1
#This value is the average temperature for each year, I used July for my control data, because I am aassuming that July is still in the summer usually.
data_avg= colMeans(temps_data2[1:31,2:21])
Std_dev=sd(data_avg)
#Setting my C and my T arbitrary, I did not want a very large C and I chose a T big enough to trigger when S>T
#Given that we want to evaluate when summer ends unofficially, I assumed that the threshold should be big enough to account for those small temperature variations during the summer. I changed C here because the standard deviation was smaller, but I kept the trigger T at about the same value(5*standarddeviation)
C=Std_dev
T=5*Std_dev
C
```

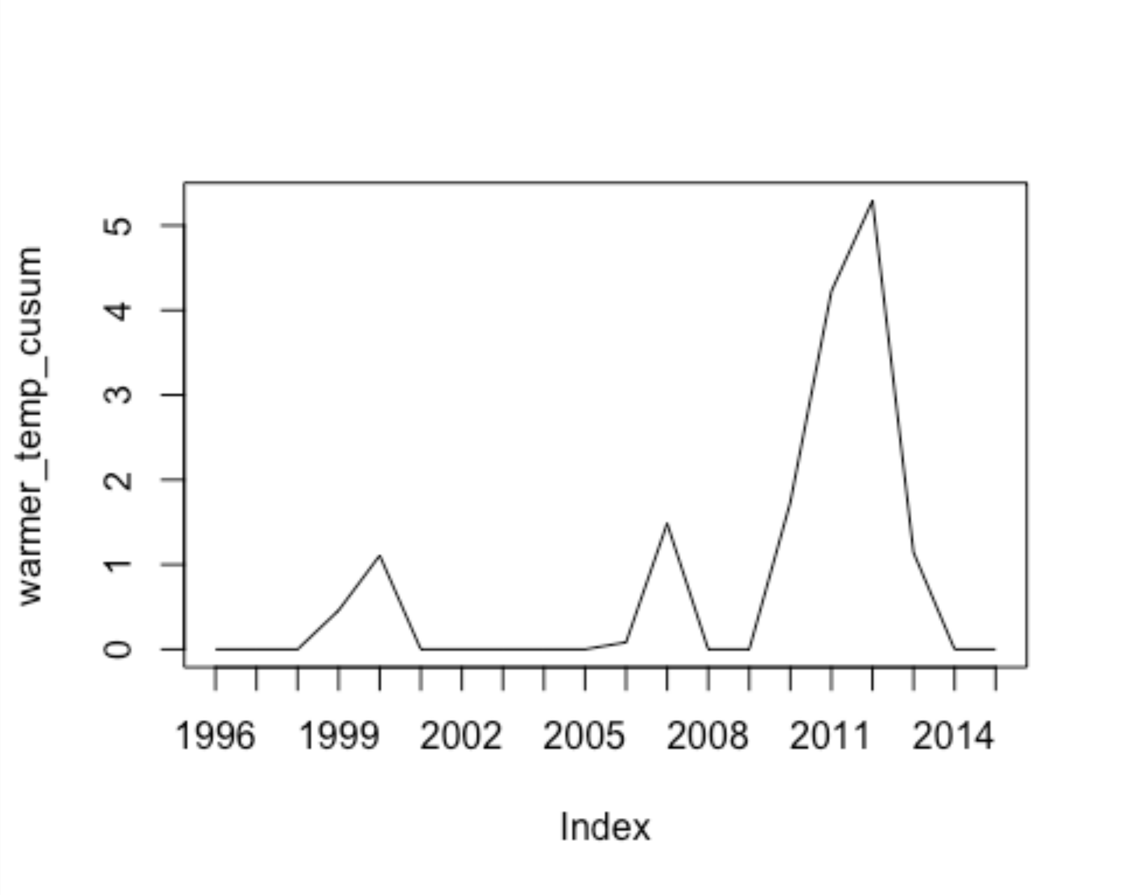
```
## [1] 2.486898
```

```
T

## [1] 12.43449

#Looping to run a CUSUM on each year from 1996 to 2015
for(col in names(temps_data2)[2:ncol(temps_data2)]){
  data=temps_data2[col]
  #average temp in july for each year
  mu=mean(data[1:31,])
  S=0
  for(i in seq(1:dim(data)[1]))
  {
    x=data[i,]
    #using this equation to assess decrease
    S=max(0,S+(mu-x-C))

    if(S>T)
    {
      cat(col,'average temperature',mu,'Approx end summer date: ',temps_data2
[i,1],'\n')
      break
    }
  }
}

## 1996 average temperature 91.19355 Approx end summer date:  27-Jul
## 1997 average temperature 87.25806 Approx end summer date:  31-Jul
## 1998 average temperature 89.70968 Approx end summer date:  3-Aug
## 1999 average temperature 87.64516 Approx end summer date:  13-Jul
## 2000 average temperature 91.74194 Approx end summer date:  25-Jul
## 2001 average temperature 86.74194 Approx end summer date:  2-Sep
## 2002 average temperature 89.25806 Approx end summer date:  12-Jul
## 2003 average temperature 85.58065 Approx end summer date:  7-Sep
## 2004 average temperature 87.83871 Approx end summer date:  10-Aug
## 2005 average temperature 86.93548 Approx end summer date:  5-Oct
## 2006 average temperature 90.19355 Approx end summer date:  7-Jul
## 2007 average temperature 86.41935 Approx end summer date:  16-Sep
## 2008 average temperature 89.16129 Approx end summer date:  13-Aug
## 2009 average temperature 86.64516 Approx end summer date:  30-Aug
## 2010 average temperature 91.25806 Approx end summer date:  4-Jul
## 2011 average temperature 91.93548 Approx end summer date:  16-Jul
## 2012 average temperature 94.09677 Approx end summer date:  14-Jul
## 2013 average temperature 84.70968 Approx end summer date:  7-Jul
## 2014 average temperature 86.6129 Approx end summer date:  22-Jul
## 2015 average temperature 90.06452 Approx end summer date:  4-Jul
```

2. Use a CUSUM approach to make a judgment of whether Atlanta's summer climate has gotten warmer in that time (and if so, when).

For this question, I based my assumption on the conclusion from the previous question. I assumed that summer was July 1st through Oct 8th. After analysis, it looks like it started to get warmer in 2011.

```
temps_data3<-read.table("temps.txt",header = TRUE,stringsAsFactors = FALSE,ch
eck.names = FALSE)
head(temps_data3)

##     DAY 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2
009
## 1 1-Jul   98   86   91   84   89   84   90   73   82   91   93   95   85
95
## 2 2-Jul   97   90   88   82   91   87   90   81   81   89   93   85   87
90
## 3 3-Jul   97   93   91   87   93   87   87   87   86   86   93   82   91
89
## 4 4-Jul   90   91   91   88   95   84   89   86   88   86   91   86   90
91
## 5 5-Jul   89   84   91   90   96   86   93   80   90   89   90   88   88
80
## 6 6-Jul   93   84   89   91   96   87   93   84   90   82   81   87   82
87
##   2010 2011 2012 2013 2014 2015
## 1   87   92  105   82   90   85
## 2   84   94   93   85   93   87
## 3   83   95   99   76   87   79
## 4   85   92   98   77   84   85
## 5   88   90  100   83   86   84
## 6   89   90   98   83   87   84

names(temps_data3)

##  [1] "DAY"  "1996" "1997" "1998" "1999" "2000" "2001" "2002" "2003" "2004"
## [11] "2005" "2006" "2007" "2008" "2009" "2010" "2011" "2012" "2013" "2014"
## [21] "2015"

#mean temperatures per year, I am assuming that summer ends in October 8, so
I am only considering data from July 1st to October 8th.
mean_temp <- colMeans(temps_data3[1:69,-1])
#function to compute the change detection equation
cusum_temperatures <- function(x,C){
c <- numeric(length(x))
mean_x <- as.numeric(mean(x))
#looping through the years to see if there was a change in the summer
for (i in 2:length(x)){
  #equation for increase
diff <- c[i-1] + x[i] - mean_x - C
c[i] <- ifelse(diff>0, diff, 0)
}
return (c)
}
#I chose a small C to be able to detect some change since C defines the shift
from the target I am also assuming I will ignore any data point that is withi
n this parameter, and we are talking about summer temperatures so the change
should not be that significant.
C<-sd(mean_temp)/2
warmer_temp_cusum <- cusum_temperatures(mean_temp,C)
names(warmer_temp_cusum) <- names(mean_temp)
plot(warmer_temp_cusum,xaxt='n',type = "l")
axis(1,at = 1:length(warmer_temp_cusum),labels = names(warmer_temp_cusum))
```



```
#I chose a lower value T in this case because I am assuming the accumulation
of change will not be that significant (we are still in summertime)
T <- 2*sd(mean_temp)
T
```

```
## [1] 3.954001
```

```
for (j in 1:length(warmer_temp_cusum)){
if(warmer_temp_cusum[j] >= T){
print(paste0('weather started getting warmer from year ',names(mean_temp)[j])
)
break
}
}
```

```
## [1] "weather started getting warmer from year 2011"
```