ISyE 6501-HOMEWORK 3

Group Members:
Jinyu, Huang | jhuang472@gatech.edu | GTID: 903522245
Chengqi, Huang | chengqihuang@gatech.edu | GTID: 903534690
Yuefan, Hu | yuefanhu@gatech.edu | GTID:903543027
Jingyu, Li | alanli@gatech.edu | GTID: 903520148

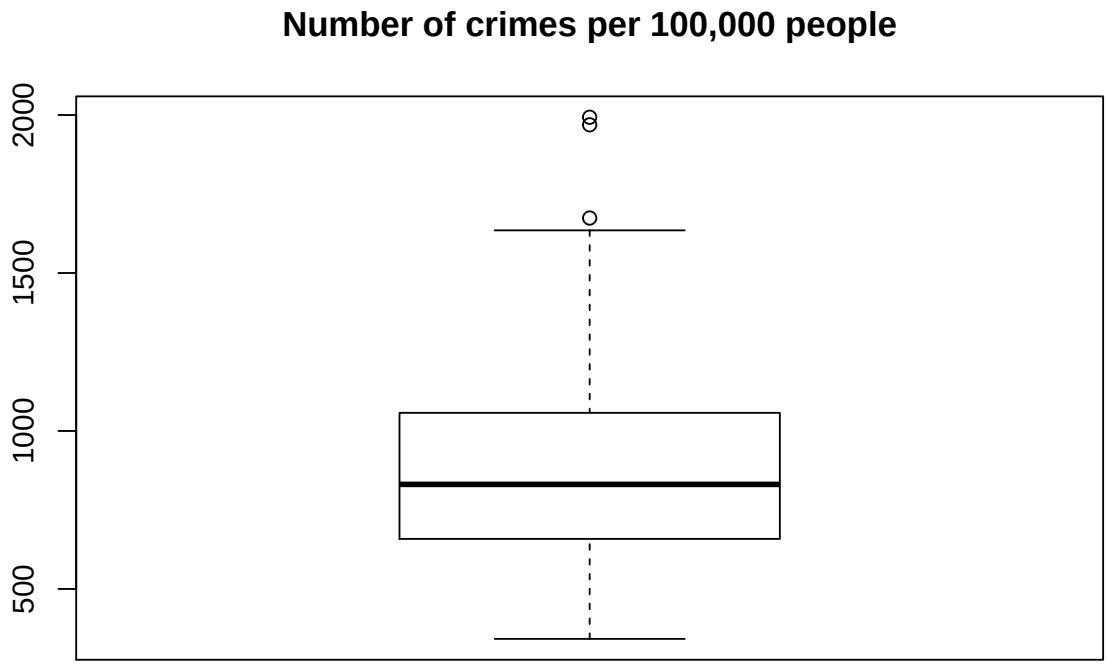
Qusetion 5.1

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.

Answer

We firstly draw a box-and-whisker plot of column *Crime* to get a overall impression on whether or not some potential outliers exist. According to the plot below, there seems to be two outliers on the same side.

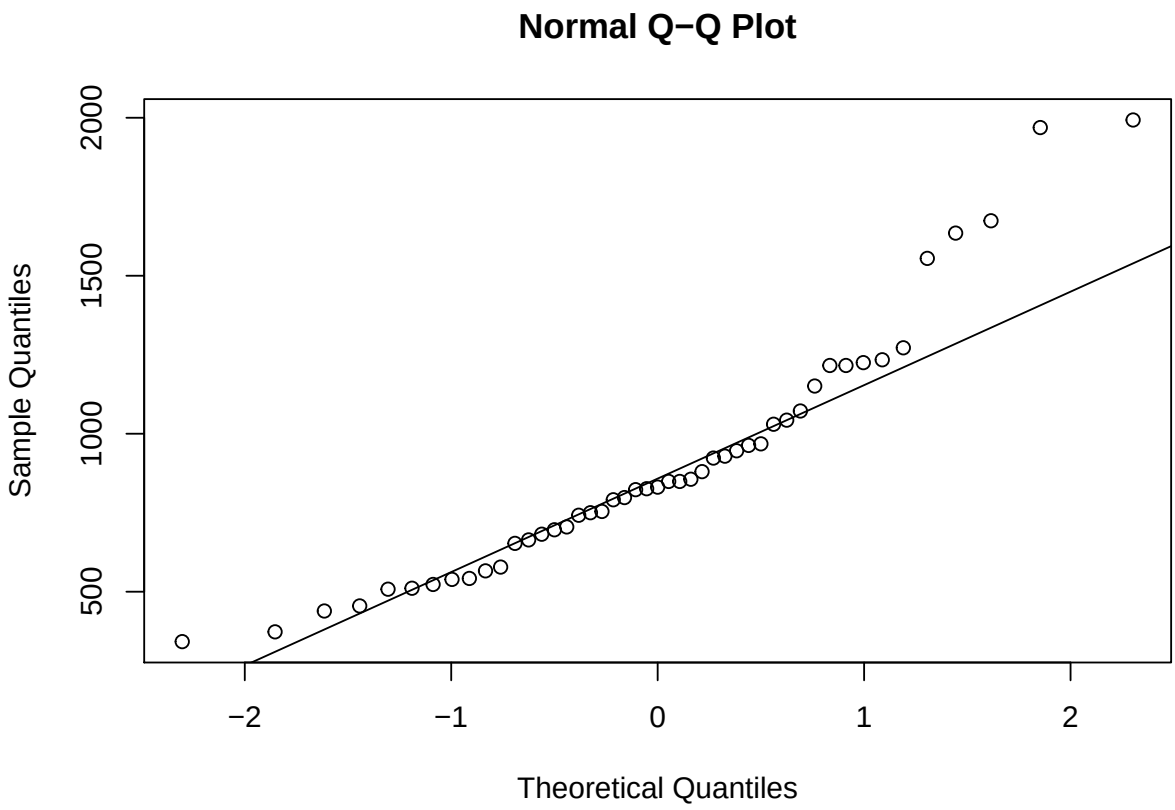
```
> data = read.table("uscrime.txt", header=TRUE)  # import data
> target_column = data[,16]
> boxplot(target_column, main="Number of crimes per 100,000 people")
```



Then because *grubbs.test* is valid when the data is normally distributed(excluding the potential outlier), we draw the QQ Normal plot of our target data. The plot shows the data is light tailed, in which the points in

the middle part generally distribute around the qqline but the first and last quantiles contain fewer data points than we expect with a normal distribution. So in generally, the distribution of our target data meet the requirement for *grubbs.test* and indeed there may be outliers.

```
> qqnorm(target_column)
> qqline(target_column)
```



Then we apply the *grubbs.test* function to check the outliers statistically. We try to set *type=20* in the function, which means the function will test for two outliers in one tail. But this method limits the sample size under 31. So what we can do is to set *type=10* for testing one outlier. Results show that the test is marginally significant, with $p = 0.079(0.05 \leq p \leq 0.1)$. We can reject the null-hypothesis(null-hypothesis: there is no outliers) but the probability of we rejecting wrongly is not that small. We propose that more investigation is needed to check if this data point is an outlier in the sample or is a real meaningful data. For example, we need to collect more data.

```
> library(outliers)
> grubbs.test(target_column, type=10)
```

```
    Grubbs test for one outlier

data:  target_column
G = 2.81287, U = 0.82426, p-value = 0.07887
alternative hypothesis: highest value 1993 is an outlier
```

Qusetion 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?

Answer

For an online game(e.g. Fortnite), daily active users(DAU) is an import indicator to evaluate the performance of the game. And in game operation, developers will add new content package into the game every 2 to 3 months to enrich the experience of the game. So the Change Detectin model can be used to see whether users like the new contents and DAU increases after the releasing. In detail, the CUSUM model is:

$$S_t = max\{0, S_{t-1} + (DAU_t - \mu - C)\}$$

$$Is \;\; S_t \geq T \; ?$$

Because we don't know the exact data of Fortnite, we just explain the model conceptually. μ is the baseline for comparison, which we will use the average DAU of the past month before we release the new content. And for the threshold(T), we think the value we choose is related to our business target. For example, we expect that we want to leverage the new content to increase the game's DAU for 20%, then T can be a number which is 20% of μ. As for the critical value(C), we don't think it's that important compared with the threshold. Considering what we want in business is a continuously increasing trend in DAU, C should be a positive number instead of zero which will make the detection not so sensitive. And if we wish to detect the change after a longer increasing period, C should be larger.

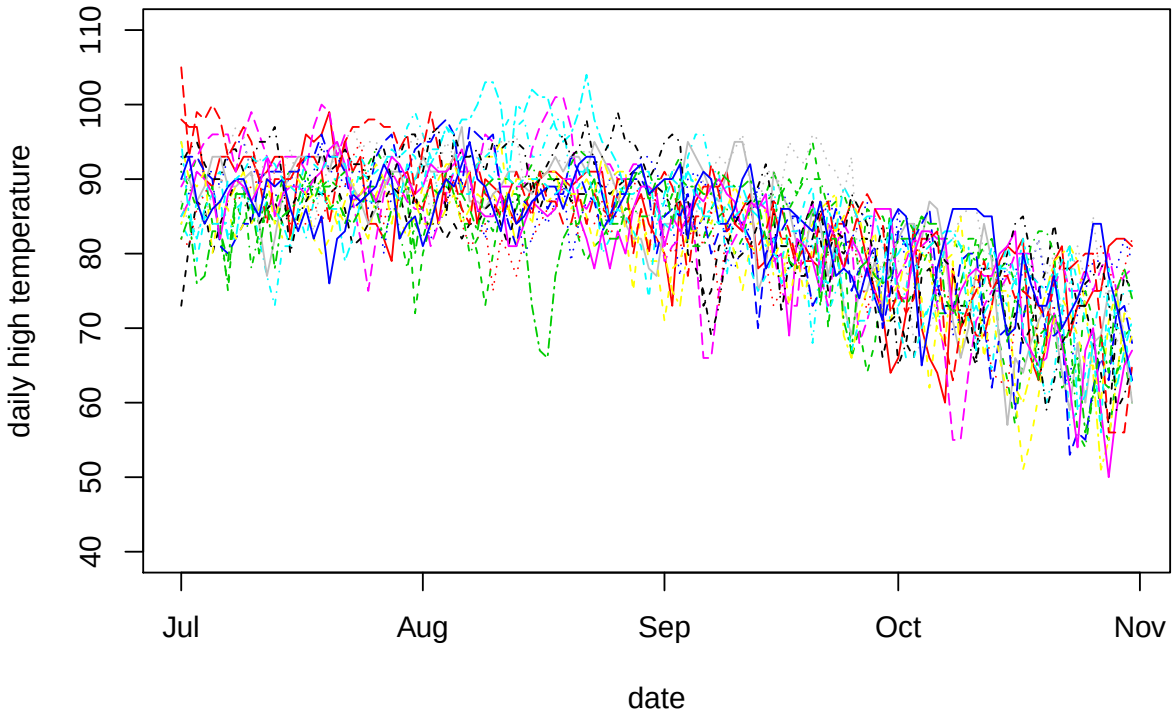
Qusetion 6.2

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. You can use R if you'd like, but it's straightforward enough that an Excel spreadsheet can easily do the job too.

Answer

In order to build our CUSUM model, we have some assumptions to help us to determine the parameters. First of all, we draw the line plot to get a general impression on how daily-high-temperature change from July to October in each year. From the plot and based on our experience, July is certainly in Summer. So we choose the average daily-high-temperature data of July as the baseline(μ) for our model.

```
> data = read.table("temps.txt", header=TRUE)  # import data
> data[,1] = as.Date(data[,1], format="%d-%b") # change column 1's data type to Date
> plot(data[,1], data[,2],
+      type="n",
+      xlab="date",
+      ylab="daily high temperature",
+      ylim=c(40,110))# build the plot
> for (i in seq(2,21)){  # draw lines
+   lines(data[,1], data[,i], col=i, lty=i-1)
+ }
```



Secondly, we assume that if the daily-high-temperature decrease below 72, it means summer ends and weather starts to cool off. So for the threshold in each year's model, $T = 72 - \mu$.

Thirdly, the daily-high-temperature data fluctuate heavly even in July. Based on the range of day-by-day change, we choose $C = 7$. So in all, our CUSUM model for each year is:

$$S_t = min\{0, S_{t-1} - (\mu - x_t - C)\}$$

$$C = 7$$

$$\mu = average(July\ data)$$

$$Is\ S_t \leq 72 - \mu\ ?$$

We build the CUSUM function in R. This function use every year's data as input and return the row number when $S_t \leq 72 - \mu$.

```
> cusum_model = function(data, column_num, C=7){
+   base = mean(data[1:31, column_num])  # July average
+   s = 0  # cusum value
+   for (i in seq(32,123)){  # traverse from Aug-1 to Oct-31
+     new_s = s-(base-data[i, column_num]-C)
+     if (new_s < 0){s = new_s}
+     else {s = 0}
+     if (s <= 72-base){return (i)}  # return row number
+   }
+   return (0)
+ }
```

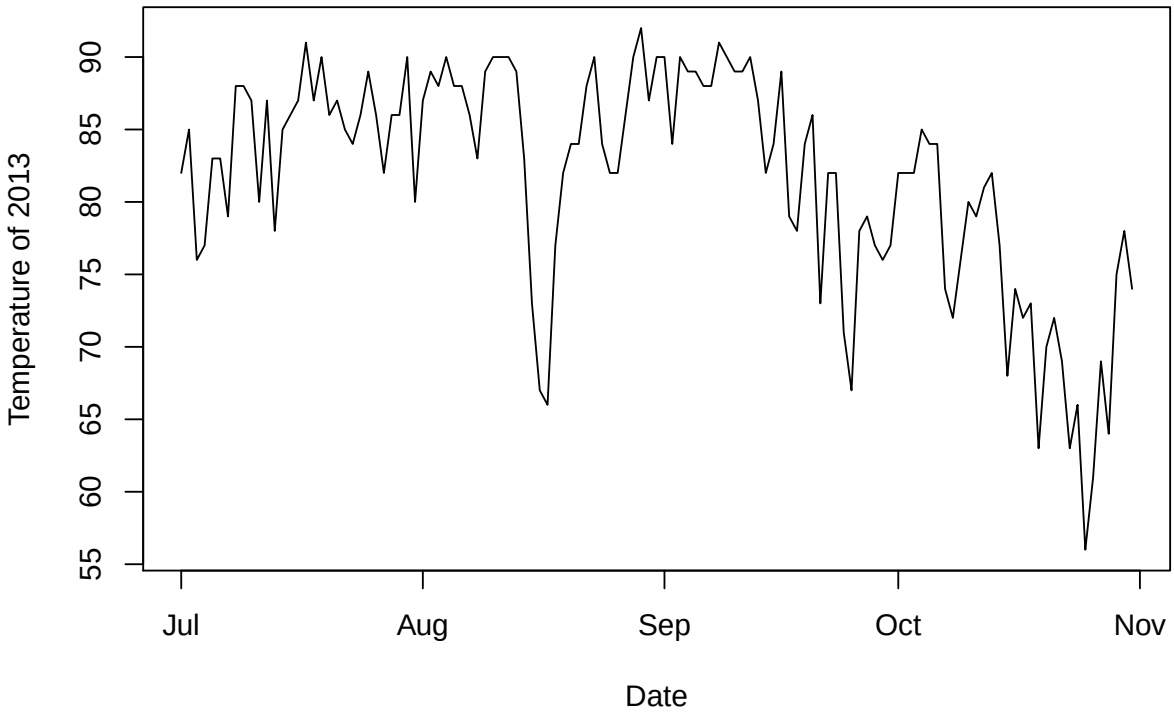
We use the function above to get the date in each year when summer ended. In most of the year, summer ended in September, especially in the second half of September. But the result of 2013 is quite strange that summer ended in the middle August.

```
> for (i in seq(2,21)){
+   row_num = cusum_model(data,i)
+   if (row_num != 0){
+     cat(colnames(data)[i],":",format(data[row_num,1],format="%d-%b"),"\n")
+   }
+   else{print("Summer did not end...")}
+ }
```

```
X1996 : 18-Sep
X1997 : 25-Sep
X1998 : 30-Sep
X1999 : 21-Sep
X2000 : 06-Sep
X2001 : 25-Sep
X2002 : 25-Sep
X2003 : 29-Sep
X2004 : 16-Sep
X2005 : 07-Oct
X2006 : 14-Sep
X2007 : 12-Oct
X2008 : 17-Sep
X2009 : 01-Oct
X2010 : 28-Sep
X2011 : 06-Sep
X2012 : 04-Sep
X2013 : 16-Aug
X2014 : 28-Sep
X2015 : 14-Sep
```

So we draw a plot to check the temperature trend in 2013. As we can see in the plot, there was a low temperature data point during middle August. In detail, from Aug 15 to 17, the daily-high-temperature were 73, 67, and 66. We google online and find that there is news about the uncommon low temperature in Atlanta during that period of time (https://www.csmonitor.com/USA/2013/0816/Atlanta-cold-snap-Why-is-it-sweater-weather-in-the-South). Since the temperature increase again after those days and it was some extreme weather, they should not be considered as the symbol of summer ending.

```
> plot(data[,1],data[,19],type="l",xlab="Date",ylab="Temperature of 2013")
```



So we try another way to regard them as outliers and replace the three data points by the average value of the former 5 days and the later 5 days. We run our CUSUM model again and it detects that the real date of summer ending may around the end of September.

```
> data[,22] = data[,19]
> data[46:48,22] = mean(data[c(41,42,43,44,45,49,50,51,52,53),22])
> colnames(data)[22] = "2013NEW"
> row_num = cusum_model(data,22)
> if (row_num != 0){
+   cat(colnames(data)[22],":",format(data[row_num,1],format="%d-%b"),"\n")
+   }
```

```
2013NEW : 25-Sep
```

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).

Answer

The first step is to calculate the average daily-high-temperature during the summer in each year. And we also replace the X2013 data by the 2013NEW column we get above.

```
> data[,19] = data[,22]  # replace 2013 data with the new data we derive above
> data = data[,-22]  # delete the extra column
```

```
> summer_average = matrix(nrow=20, ncol=2)  # store average temperature data
> summer_average[,1] = seq(1996,2015)
> for (i in seq(2,21)){
+   row_num = cusum_model(data,i)  # row number that summer ends
```

```
+   average_value = mean(data[1:row_num-1,i])  # average temperature during summer
+   summer_average[i-1,2] = average_value
+ }
> cat("average summer daily-high-temperature from 1996 to 2015:","\n");summer_average
```

```
average summer daily-high-temperature from 1996 to 2015:

      [,1]     [,2]
 [1,] 1996 88.48101
 [2,] 1997 86.30233
 [3,] 1998 86.97802
 [4,] 1999 88.75610
 [5,] 2000 90.70149
 [6,] 2001 85.65116
 [7,] 2002 88.30233
 [8,] 2003 84.75556
 [9,] 2004 85.40260
[10,] 2005 86.29592
[11,] 2006 88.76000
[12,] 2007 87.96117
[13,] 2008 87.23077
[14,] 2009 84.93478
[15,] 2010 90.97753
[16,] 2011 92.41791
[17,] 2012 90.75385
[18,] 2013 85.52442
[19,] 2014 86.53933
[20,] 2015 88.72000
```

Considering there are some variance among the weather in different years, taking one year as the baseline is not proper. We use the average value of Year 1996, Year 1997 and Year 1998 as the baseline. Then we set $T = 5$(approximately equal to 5% of baseline) and $C = 2.5$(average year-by-year difference) in our CUSUM model. According to the model, there is no strong trend that Atlanta's summer climate has gotten warmer.

```
> base = mean(summer_average[1:3,2])  # baseline, Year 1996 to 1998
> s = 0  # cusum value
> for (i in seq(4,20)){  # traverse from Year 1999 to 2015
+   new_s = s+(summer_average[i,2]-base-2.5)
+   if (new_s > 0){s = new_s}
+   else {s = 0}
+   if (s >= 5){print(summer_average[i,1]);break}  # print year
+ }
> if (s<5){print("Atlanta's summer climate has not gotten warmer")}
```

```
[1] "Atlanta's summer climate has not gotten warmer"
```