Hmrk3

N/A

2023-09-09

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#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.
```

```
#Let’s start with displaying the head of the data frame

data = read.table("uscrime.csv", header = TRUE)
head(data)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      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
```

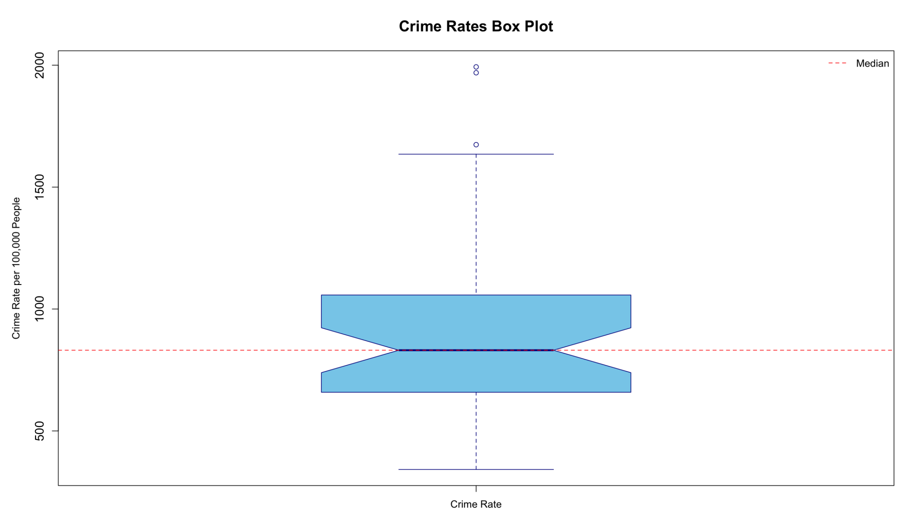
```
#Now let's take a look at some of the summary statistics for the dataframe, as well as some summary statistics for the crime column. We can see here that the minimum value is 342 and #the max value is 1993.

summary(data)
```

```
##        M               So               Ed             Po1
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80
##  Mean   :13.86   Mean   :0.3404   Mean   :10.56   Mean   : 8.50
##  3rd Qu.:14.60   3rd Qu.:1.0000   3rd Qu.:11.45   3rd Qu.:10.45
##  Max.   :17.70   Max.   :1.0000   Max.   :12.20   Max.   :16.60
##       Po2               LF              M.F              Pop
##  Min.   : 4.100   Min.   :0.4800   Min.   : 93.40   Min.   :  3.00
##  1st Qu.: 5.850   1st Qu.:0.5305   1st Qu.: 96.45   1st Qu.: 10.00
##  Median : 7.300   Median :0.5600   Median : 97.70   Median : 25.00
##  Mean   : 8.023   Mean   :0.5612   Mean   : 98.30   Mean   : 36.62
##  3rd Qu.: 9.700   3rd Qu.:0.5930   3rd Qu.: 99.20   3rd Qu.: 41.50
##  Max.   :15.700   Max.   :0.6410   Max.   :107.10   Max.   :168.00
##        NW              U1                U2            Wealth
##  Min.   : 0.20   Min.   :0.07000   Min.   :2.000   Min.   :2880
##  1st Qu.: 2.40   1st Qu.:0.08050   1st Qu.:2.750   1st Qu.:4595
##  Median : 7.60   Median :0.09200   Median :3.400   Median :5370
##  Mean   :10.11   Mean   :0.09547   Mean   :3.398   Mean   :5254
##  3rd Qu.:13.25   3rd Qu.:0.10400   3rd Qu.:3.850   3rd Qu.:5915
##  Max.   :42.30   Max.   :0.14200   Max.   :5.800   Max.   :6890
##       Ineq            Prob              Time           Crime
##  Min.   :12.60   Min.   :0.00690   Min.   :12.20   Min.   : 342.0
##  1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 658.5
##  Median :17.60   Median :0.04210   Median :25.80   Median : 831.0
##  Mean   :19.40   Mean   :0.04709   Mean   :26.60   Mean   : 905.1
##  3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:1057.5
##  Max.   :27.60   Max.   :0.11980   Max.   :44.00   Max.   :1993.0
```

```
summary(data$Crime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   342.0   658.5   831.0   905.1  1057.5  1993.0
```

```
#Let also create a boxplot, where we can visually see the outliers, the median(shown in red) as #well as where the majority of the data lies. We can also deduce the 1st and 3rd quartiles.


boxplot(data$Crime,
        main = "Crime Rates Box Plot",
        ylab = "Crime Rate per 100,000 People",
        col = "skyblue",
        border = "darkblue",
        notch = TRUE,
        names = c("Crime Rate"),
        cex.axis = 1.2,
        cex.main = 1.5,
        xaxt = "n"
)

# Customize x-axis labels (if needed)
axis(side = 1, at = 1, labels = c("Crime Rate"))

# Add a horizontal grid line
abline(h = median(data$Crime), col = "red", lty = 2)

# Add a legend
legend("topright", legend = "Median", col = "red", lty = 2, bty = "n")
```




```
# Here are the results of the grub test. I ran this a few times, with each result displaying slightly different information, where I will explain in the analysis.
```

```
grubbs_result <- grubbs.test(data$Crime, type = 10, opposite = TRUE)
grubbs_result
```

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

```
grubbs_result <- grubbs.test(data$Crime, type = 10, opposite = FALSE)
grubbs_result
```

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

```
grubbs_result <- grubbs.test(data$Crime, type = 11,  opposite = TRUE)
grubbs_result
```

```
##
##  Grubbs test for two opposite outliers
##
## data:  data$Crime
## G = 4.26877, U = 0.78103, p-value = 1
## alternative hypothesis: 342 and 1993 are outliers
```

```
grubbs_result <- grubbs.test(data$Crime, type = 11,  opposite = FALSE)
grubbs_result
```

```
##
##  Grubbs test for two opposite outliers
##
## data:  data$Crime
## G = 4.26877, U = 0.78103, p-value = 1
## alternative hypothesis: 342 and 1993 are outliers
```

```
#Analysis: Ultimately, the grub test determined that 1993 is most likely an outlier, but there is no evidence to claim that 342 is an outlier. Above there are grub tests with type 10 and type 11. Type 10 and Type 11 Grubbs' tests lies in their ability to detect outliers with different characteristics. Type 10 Grubbs' test is designed to specifically identify one high outlier, assuming that there is at most one unusually high value in the dataset. In contrast, Type 11 Grubbs' test is more versatile as it can detect one outlier that is either higher or lower than the rest of the data points. It accommodates the possibility of both high and low outliers, making it suitable for situations where you want to identify any single data point that significantly deviates from the majority of the data. In Grubbs' tests, the opposite parameter with the value TRUE signifies that you are actively searching for potential outliers in your data. It directs the #test to identify data points that significantly deviate from the majority of the data, treating them #as possible outliers. On the other hand, setting opposite to FALSE assumes that there are no outliers of interest, and the test primarily checks whether all data points are consistent with the majority of the data. This choice impacts the focus of the test, with TRUE being suitable when you want to detect outliers and FALSE when you want to confirm the absence of significant outliers. 342 was inside the whiskers of the plot, it wouldn’t really be an outlier. The downside #oo the Grubbs test is that it will only find one outlier for each side unless type 20 is noted, but in that case the test only accepts less than 30 values.
```

```
#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?
```

```
#ANS : Change detection models can be used in a variety of different scenarios to monitor data over time and identify abrupt or significant changes. For example, in healthcare, change detection models can be applied to monitor vital signs of a patient or detect anomalies that may indicate more health issues. A more specific example can be detecting irregular heart rhythms, or changes in blood pressure. A change detection model would be a great resource in a hospital and it can alarm doctors for when a patient is in trouble. To apply the CUSUM technique here, we would need to first start with data collection. This would be done by continuously monitoring the patient's blood pressure. Blood pressure measurements are typically recorded at regular intervals, such as every few minutes, using an automated blood pressure monitor. Then we would need to establish a baseline for the patient's blood pressure. The baseline represents the expected range of blood pressure values for the patient when they are in a stable and healthy condition. It can be determined based on historical blood pressure data for the patient and relevant medical guidelines. For the critical value(h) we can choose something that represents the level of change in blood pressure that is considered significant and warrants an alert. The critical value should align with the clinical significance of the change. For example, you might set a lower critical value if even minor blood pressure fluctuations are concerning in the patient's context, or a higher critical value if larger changes are expected in certain situations. For the threshold(k), a value should be selected carefully to avoid raising false alarms due to normal variations in blood pressure. It should be greater than zero and based on historical data and clinical knowledge. A common approach is to set the threshold slightly above the expected range of normal blood pressure fluctuations for the patient.
```

```
#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.
#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).
```

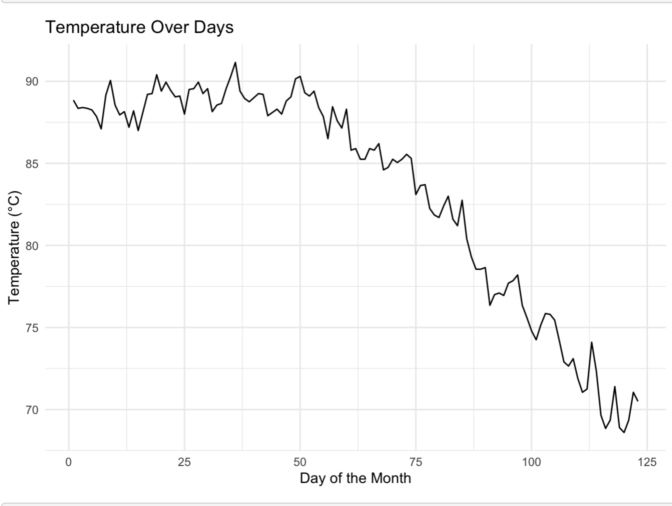
```
#PART 1( 6.1.2)
#For this question, I decided to use R and some simple data manipulation. I first took a look at how the temperature changes throughout the months with some exploratory data analysis and ran some simple summary statistics. As expected, it decreases:
```

```
#Calculating the average of the temperatures
temp_data = read.table("temps.csv", header = TRUE)
temp_data$Average = rowMeans(temp_data[, -1], na.rm = TRUE)
temp_data[,"Csum"]= NA
```

```
summary(temp_data)
```

```
##      DAY                X1996           X1997           X1998
##  Length:123         Min.   :60.00   Min.   :55.00   Min.   :63.00
##  Class :character   1st Qu.:79.00   1st Qu.:78.50   1st Qu.:79.50
##  Mode  :character   Median :84.00   Median :84.00   Median :86.00
##                     Mean   :83.72   Mean   :81.67   Mean   :84.26
##                     3rd Qu.:90.00   3rd Qu.:88.50   3rd Qu.:89.00
##                     Max.   :99.00   Max.   :95.00   Max.   :95.00
##      X1999           X2000            X2001           X2002
##  Min.   :57.00   Min.   : 55.00   Min.   :51.00   Min.   :57.00
##  1st Qu.:75.00   1st Qu.: 77.00   1st Qu.:78.00   1st Qu.:78.00
##  Median :86.00   Median : 86.00   Median :84.00   Median :87.00
##  Mean   :83.36   Mean   : 84.03   Mean   :81.55   Mean   :83.59
##  3rd Qu.:91.00   3rd Qu.: 91.00   3rd Qu.:87.00   3rd Qu.:91.00
##  Max.   :99.00   Max.   :101.00   Max.   :93.00   Max.   :97.00
##      X2003           X2004           X2005           X2006
##  Min.   :57.00   Min.   :62.00   Min.   :54.00   Min.   :53.00
##  1st Qu.:78.00   1st Qu.:78.00   1st Qu.:81.50   1st Qu.:79.00
##  Median :84.00   Median :82.00   Median :85.00   Median :85.00
##  Mean   :81.48   Mean   :81.76   Mean   :83.36   Mean   :83.05
##  3rd Qu.:87.00   3rd Qu.:87.00   3rd Qu.:88.00   3rd Qu.:91.00
##  Max.   :91.00   Max.   :95.00   Max.   :94.00   Max.   :98.00
##      X2007           X2008           X2009           X2010
##  Min.   : 59.0   Min.   :50.00   Min.   :51.00   Min.   :67.00
##  1st Qu.: 81.0   1st Qu.:79.50   1st Qu.:75.00   1st Qu.:82.00
##  Median : 86.0   Median :85.00   Median :83.00   Median :90.00
##  Mean   : 85.4   Mean   :82.51   Mean   :80.99   Mean   :87.21
##  3rd Qu.: 89.5   3rd Qu.:88.50   3rd Qu.:88.00   3rd Qu.:93.00
##  Max.   :104.0   Max.   :95.00   Max.   :95.00   Max.   :97.00
##      X2011           X2012            X2013           X2014
##  Min.   :59.00   Min.   : 56.00   Min.   :56.00   Min.   :63.00
##  1st Qu.:79.00   1st Qu.: 79.50   1st Qu.:77.00   1st Qu.:81.50
##  Median :89.00   Median : 85.00   Median :84.00   Median :86.00
##  Mean   :85.28   Mean   : 84.65   Mean   :81.67   Mean   :83.94
##  3rd Qu.:94.00   3rd Qu.: 90.50   3rd Qu.:88.00   3rd Qu.:89.00
##  Max.   :99.00   Max.   :105.00   Max.   :92.00   Max.   :95.00
##      X2015         Average        Csum
##  Min.   :56.0   Min.   :68.60   Mode:logical
##  1st Qu.:77.0   1st Qu.:77.78   NA's:123
##  Median :85.0   Median :85.90
##  Mean   :83.3   Mean   :83.34
##  3rd Qu.:90.0   3rd Qu.:88.78
##  Max.   :97.0   Max.   :91.15
```

```
ggplot(temp_data, aes(x = 1:length(Average), y = Average)) +
  geom_line() +
  labs(x = "Day of the Month", y = "Temperature (°C)", title = "Temperature Over Days") +
  theme_minimal()
```



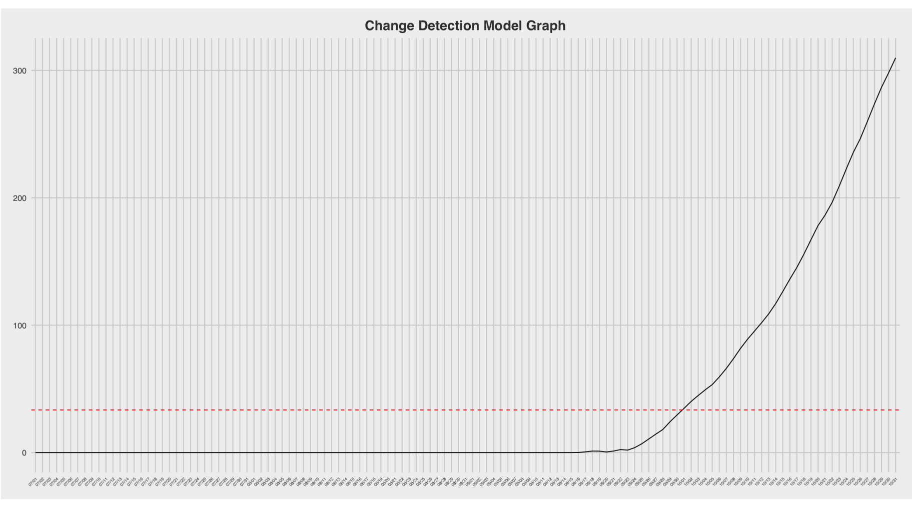

```
# Now we will set C to the standard deviation. The value of C can be set to anything... but I wanted to detect change relatively quickly so I set it to be 0. Sometimes C may be set to 1 or two times the standard deviation as well I set t to 5 times the stdev. These are just based of standard practices.

standard_dev = sd(temp_data[,"Average"])
C = 1
t = 5 * standard_dev
mean_sample<-mean(temp_data[,"Average"])
```

```
#Here is where the beauty of the Csum algorithm occurs. the code calculates a cumulative sum (Csum) based on the values in the Average column of the temp_data data frame, along with mean_sample and C. It ensures that the cumulative sum doesn't go below zero using the max(0, ...) function. The result is stored in the Csum column for each row in temp_data.

temp_data$Csum <- 0

for (i in 2:nrow(temp_data)) {
  temp_data$Csum[i] = max(0, temp_data$Csum[i - 1] + mean_sample - temp_data$Average[i] - C)
}
```

```
#Here we see where the algorithm is able to detect some change with a for loop. We base the threshold at 5 times the standard deviation, which is approximately 34. Based of Csum, the change starts to occur in mid/late October.

for (element in seq_along(temp_data$Csum)){

  if (temp_data$Csum[element] > t) {
    print(temp_data$DAY[element])
  }
}
```

```
## [1] "1-Oct"
## [1] "2-Oct"
## [1] "3-Oct"
## [1] "4-Oct"
## [1] "5-Oct"
## [1] "6-Oct"
## [1] "7-Oct"
## [1] "8-Oct"
## [1] "9-Oct"
## [1] "10-Oct"
## [1] "11-Oct"
## [1] "12-Oct"
## [1] "13-Oct"
## [1] "14-Oct"
## [1] "15-Oct"
## [1] "16-Oct"
## [1] "17-Oct"
## [1] "18-Oct"
## [1] "19-Oct"
## [1] "20-Oct"
## [1] "21-Oct"
## [1] "22-Oct"
## [1] "23-Oct"
## [1] "24-Oct"
## [1] "25-Oct"
## [1] "26-Oct"
## [1] "27-Oct"
## [1] "28-Oct"
## [1] "29-Oct"
## [1] "30-Oct"
## [1] "31-Oct"
```

```
temp_data[,"Date"]<-as.Date(temp_data[,"DAY"],"%d-%B")
temp_data[,"Date"]<-format(temp_data[,"Date"],format="%m/%d")

ggplot(data = temp_data, aes(x = Date, y = Csum, group = 2)) +
  geom_line() +
  geom_hline(yintercept = t, color = "red", linetype = "dashed") +
  labs(x = "Date", y = "Csum", title = "Change Detection Model Graph") +
  theme_fivethirtyeight() +
  theme(
    axis.text.x = element_text(size = 5, angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, size = 16),
    aspect.ratio = 0.5 # Adjust aspect ratio for wider appearance
  )
```




```
#Based of the visualization here, we can see that a large change is detected at the September 29th mark. I used a C value of 0 to detect change faster, but if this is increased then the change will take a little longer to detect and it may go into October a little. I apologize for the x-axis being so small.. but the threshold value hits around late september/ early October.
```

```
#PART 2( 6.2.2)
# Use a CUSUM approach to make a judgment of whether Atlanta’s summer climate has gotten warmer in that time (and if so, when).
```

```
# To now make a judgment of whether Atlanta’s summer climate has gotten warmer in that time, we can use the Csum approach again, just on a a different set of data. Since we determined in the last part of the question that the weather cools down in around late September, we can assume that is when summer ends. Due to this, I only used data points from July 1st to September 29th.
```

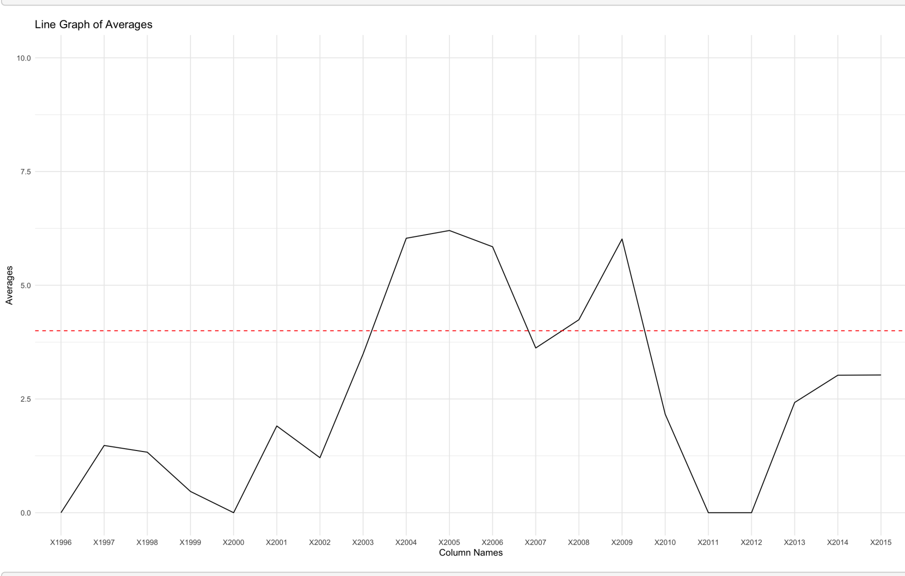
```
new_subset = temp_data[1:91, ]
# Remove the "Average" and "Csum" columns
new_subset <- subset(new_subset, select = -c(Average, Csum, Date))
```

```
#making a seperate dataframe and also calculating averages
column_averages <- colMeans(new_subset[, -1], na.rm = TRUE)
averages_data <- data.frame(
  ColumnNames = names(column_averages),
  Averages = column_averages
)
```

```
#Keeping C at 0 to detect changes faster I tried T values from 1 to 4
standard_dev = sd(averages_data[,"Averages"])
C = 0
t = 4
mean_sample<-mean(averages_data[,"Averages"])
```

```
#Applying Csum method
averages_data$Csum <- 0


for (i in 2:nrow(averages_data)) {
  averages_data$Csum[i] = max(0, averages_data$Csum[i - 1] + mean_sample - averages_data$Averages[i] - C)
}
```

```
#Visualizing the data to make a judgment of whether Atlanta’s summer climate has gotten warmer in that time
ggplot(data = averages_data, aes(x = ColumnNames, y = Csum, group = 2)) +
  geom_line() +  # Add a line plot
  geom_hline(yintercept = t, color = "red", linetype = "dashed") +
  labs(x = "Column Names", y = "Averages", title = "Line Graph of Averages") +
  theme_minimal() +
  ylim(0,10)
```




```
#Based of this plot, it is challenging to determine whether or not Atlanta's summer has gotten warmer. The data I used included averaged temperatures from July-Early October. Based off the threshold value of 4, it seems that the weather changes drastically from 2003 to 2009. After this it goes back down, so it is difficult to see any sort of change detection through the Csum model. We can see that the the temperature is a little hotter through the years of 2003-2009 however. In this case, there may be a need to explore other methods, possibilties or algorithms to see if there is truly a change.
```