HW 3

2023-09-05

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.

```
options(repos = c(CRAN = "https://cran.r-project.org"))
install.packages("outliers")
```

```
##
## The downloaded binary packages are in
##  /var/folders/cz/ntt36s2d63lgvz1wy56zv5h80000gn/T//RtmpP6jAly/downloaded_packages
```

```
library(outliers)
```

```
data <- read.table("http://www.statsci.org/data/general/uscrime.txt", stringsAsFactors = FALSE, header =
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
```

```
# Check for missing values
summary(data)
```

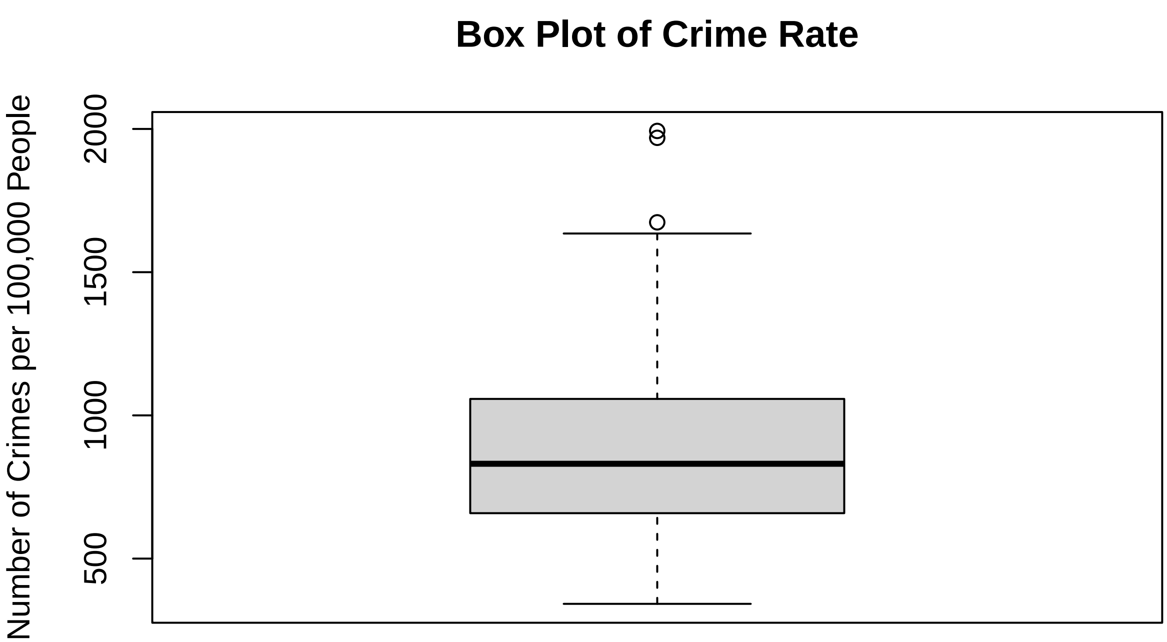
```
##        M               So               Ed              Po1
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80
##  Mean   :13.86   Mean   :0.3404   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
```

```r
# Perform Grubbs' test on the last column
outlier_result <- grubbs.test(data$Crime, type = 10)
#interpret the results
outlier_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
```

```r
# Create a box plot of the last column (CrimeRate)
boxplot(data$Crime, main = "Box Plot of Crime Rate",
        ylab = "Number of Crimes per 100,000 People")
```




```r
# Sort the unique values in descending order
sorted_values <- sort(unique(data$Crime), decreasing = TRUE)
```

```r
# Extract the second and third highest values
second_highest <- sorted_values[2]
third_highest <- sorted_values[3]

# Print the second and third highest values
cat("Second Highest Value:", second_highest, "\n")
```

```
## Second Highest Value: 1969
```

```r
cat("Third Highest Value:", third_highest, "\n")
```

```
## Third Highest Value: 1674
```

After loading and exploring the dataset I didnt come across any apparent missing values. Subsequently I used the grubbs test to detect outliers in the last column. The results returned a p-Vaue of 0.07887.This p-value (0.07887) is greater than the significance level (at 0.05), which suggests that there is not strong evidence to reject the null hypothesis. The null hypothesis in this context would be that the highest value (1993) is not an outlier; it is a part of the dataset and does not significantly differ from the other data points. Based on this result, there is not strong statistical evidence to conclude that the highest value (1993) in the "Crime" variable is an outlier.

Sometimes, even if not statistically significant, extreme values may still be meaningful outliers from a practical standpoint. Whether such a value makes sense in the context of the crime data and problem, would require that I interpret what it means relative to the rest of the data. I constructed a boxplot and it does show to be an extreme data point from the rest of the data. Further analysis, shows that it differs by 24/100000 from the second highest value and by 319 offenses per 100,000 from the third highest value, which may have important implications, this could be due to an unusual spike in crime, public safety concern or policy and resource allocation.

##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?

I work as an analyst and one of our responsibilities is managing a substantial number of investment accounts, where detecting changes in cash flow is very importance. The implementation of a Change Detection model could be a crucial strategy for identifying anomalies that may significantly affect portfolio performance. In sum, our work involves overseeing diverse investment portfolios for a multitude of clients, each with distinct financial goals and risk preferences. The primary objective is to ensure that these portfolios consistently meet their performance expectations. However, the recognition of abrupt and unforeseen fluctuations in cash flow has revealed their substantial influence on the overall performance of specific accounts. To effectively address this challenge, the adoption of a Change Detection model, specifically the Cumulative Sum technique, may be very effective.

The CV (C) plays a pivotal role in determining when the cumulative sum resets to zero. Given the management of a vast number of accounts, setting an appropriate CV that accommodates the diversity of client cash flow patterns is essential. A higher CV lends adaptability to gradual changes while resetting less frequently, which is advantageous for identifying significant deviations. However, a lower CV triggers resets more frequently, proving beneficial for promptly recognizing abrupt changes in cash flow dynamics. In observing change in performance, despite cases of randomness, it doesn't cost to have a false positive, therefore setting the critical value to 0 for a more sensitive method could be beneficial.

The threshold (T) is also a pivotal parameter that governs the model's sensitivity to cash flow deviations. By choosing a relatively low threshold value, we could enhance the model's responsiveness to even subtle changes, enabling the prompt detection of anomalies. This heightened sensitivity proves instrumental in mitigating the potential consequences of unexpected cash flows, be it underperformance or overperformance, as we report back to clients.

##Question 6.2 1. Using July through October daily-high-temperature data for Atlanta for 1996 through

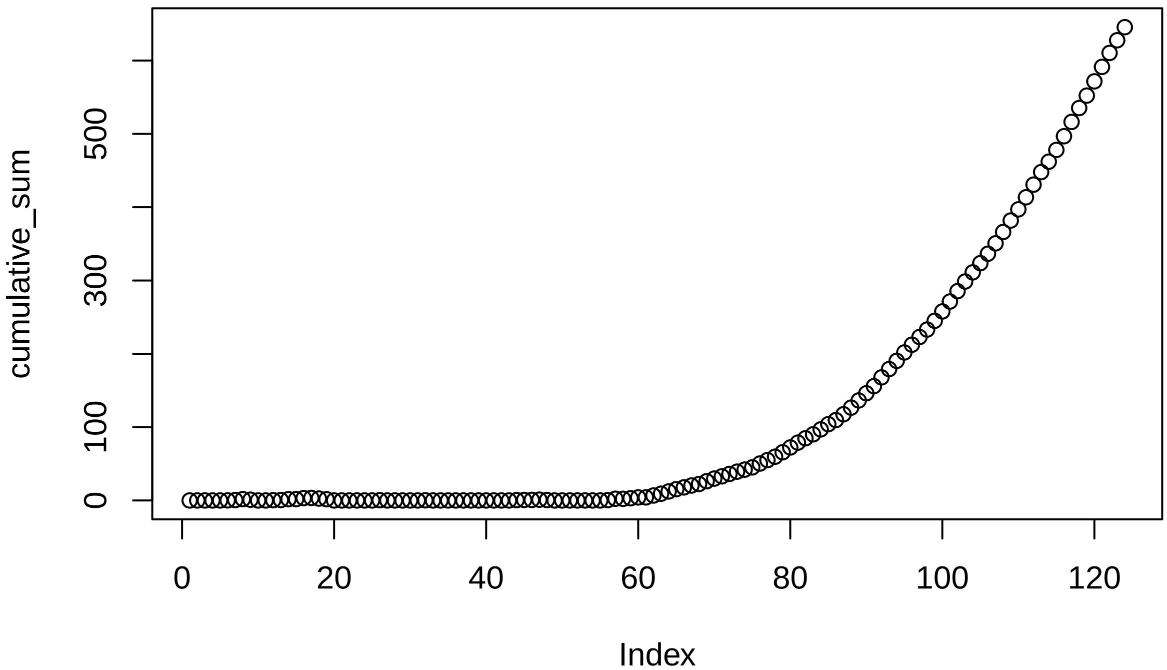
2015, use a CUSUM approach to identify when unofficial summer ends (i.e., when the weather starts cooling off) each year. You can get the data that you need from the file temps.txt or online, for example at http://www.iweathernet.com/atlanta-weather-records or https://www.wunderground.com/history/airport/KFTY/2015/7/1/CustomHistory.html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

#Calculate CUSUM for Identifying the End of Summer Each Year:

In this code, I first calculated the daily average temperatures spanning multiple years and computed the overall mean temperature. Then, I quantified the deviation of each day's average temperature from this mean, introducing a sensitivity control variable called cusum_C. Subsequently, I generated a cumulative sum of these temperature deviations, ensuring it remained non-negative. The cumulative sum helps identify instances when temperature deviations surpass a specified threshold.

After plotting the cumulative sum for visual analysis, I used the which function to identify the indices where the cumulative sum exceeded or equaled 75, signifying significant temperature changes. The index_above_75 variable stored these indices, revealing the days when noteworthy temperature anomalies occurred. Temperatures begin to show a significant difference by October 22nd.

```
temps = read.table('/Users/barovierallybose/Documents/Intro to analytics modeling /HW 3/hw3-FA23/temps.t
# Calculate daily temperature averages for each day across multiple years

# Select all columns containing temperature data (assuming they start from column 2)
temperature_columns <- temps[, 2:ncol(temps)]

# Calculate the daily average temperature by row (each day)
daily_avg_temperatures <- rowMeans(temperature_columns, na.rm = TRUE)

# The result, 'daily_avg_temperatures', is a vector where each element represents the daily average tem
# Calculate the mean of the daily average temperatures
mean_daily_avg_temp <- mean(daily_avg_temperatures)

# Calculate the difference between each day's average temperature and the overall mean
daily_avg_deviation_from_mean <- daily_avg_temperatures - mean_daily_avg_temp

# Set a variable 'cusum_threshold' to control sensitivity (you can adjust this value)
cusum_C <- 5  # You can change this value to control sensitivity

# Subtract the 'cusum_threshold' from the difference score
damimu_minus_C <- daily_avg_deviation_from_mean - cusum_C

# Initialize the cumulative sum with zeros
cumulative_sum <- numeric(length(damimu_minus_C) + 1)

# Loop through each day, calculate the cumulative sum, and update the accumulator
for (i in seq_along(damimu_minus_C)) {
  cumulative_sum[i + 1] <- max(0, cumulative_sum[i] - damimu_minus_C[i])
}
# Plot the cumulative sum
plot(cumulative_sum)
```




```
# Get the index of the maximum cumulative sum that is greater than or equal to 85
index_above_75 <- which(cumulative_sum >= 75)
index_above_75
```

```
##  [1]  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99
## [20] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
## [39] 119 120 121 122 123 124
```

```
# Return the date associated with the index
date_at_index_above_75 <- temps[index_above_75, 1]
date_at_index_above_75
```

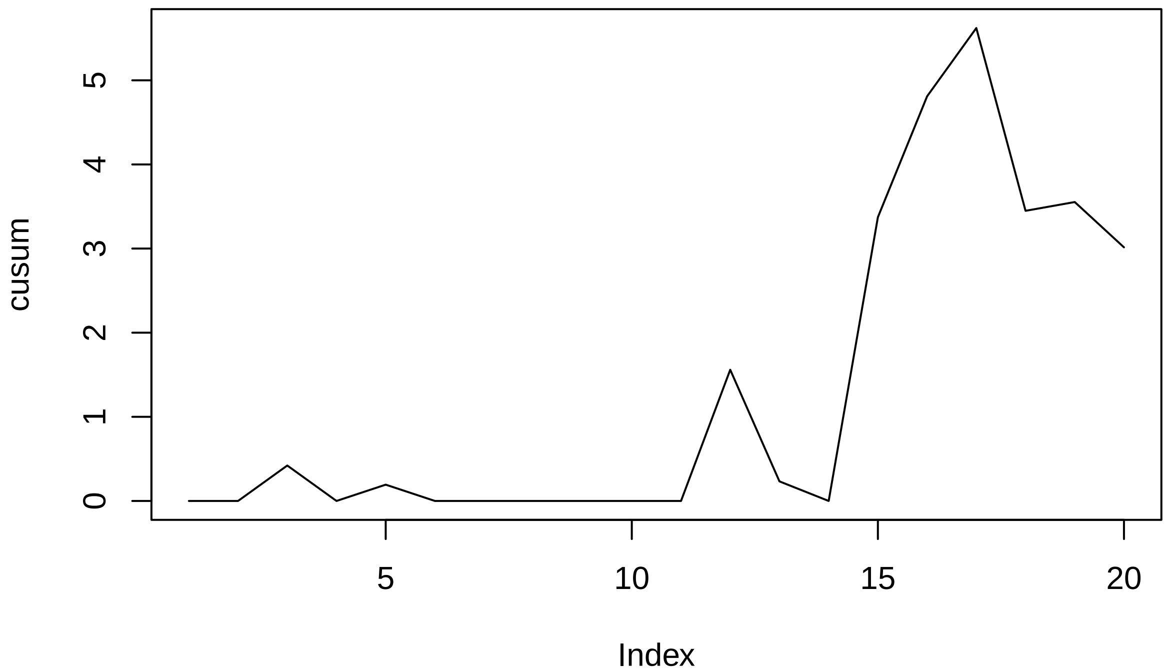
```
##  [1] "19-Sep" "20-Sep" "21-Sep" "22-Sep" "23-Sep" "24-Sep" "25-Sep" "26-Sep"
##  [9] "27-Sep" "28-Sep" "29-Sep" "30-Sep" "1-Oct"  "2-Oct"  "3-Oct"  "4-Oct"
## [17] "5-Oct"  "6-Oct"  "7-Oct"  "8-Oct"  "9-Oct"  "10-Oct" "11-Oct" "12-Oct"
## [25] "13-Oct" "14-Oct" "15-Oct" "16-Oct" "17-Oct" "18-Oct" "19-Oct" "20-Oct"
## [33] "21-Oct" "22-Oct" "23-Oct" "24-Oct" "25-Oct" "26-Oct" "27-Oct" "28-Oct"
## [41] "29-Oct" "30-Oct" "31-Oct" NA
```

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).

I applied a CUSUM analysis spanning multiple years to identify temperature changes. I followed a similar approach as with daily returns, first computing yearly averages and then determining the mean (mu) for each year. Next, I iterated through each year's temperature data to calculate the deviation from the overall average temperature for the entire period. To fine-tune sensitivity, I set the control constant (C) to 0.5 and computed the CUSUM values. Additionally, I visualized the CUSUM control chart to pinpoint variations in individual year means relative to the adjusted target value. By utilizing the "which" function, I detected instances where the cumulative sum surpassed a threshold of 5, leading to the observation that Atlanta's temperatures had indeed experienced a warming trend by the year 2011.

```
# Get average temp for each year
yearly_avgs <- colMeans(temps[-1], na.rm=TRUE)

# Overall mean
```

```r
mu <- mean(yearly_avgs)

# Differences from mean
diffs <- yearly_avgs - mu

# Threshold
C <- 0.5

# Initialize CUSUM
cusum <- diffs - C
cusum[1] <- 0

# Compute CUSUM
for(i in 2:length(cusum)){
  cusum[i] <- max(0, cusum[i-1] + diffs[i] - C)
}

# Plot
plot(cusum, typ="l")
```




```r
# Find crossing
crossing <- which(cusum > 5)[1]

# Print year
print(colnames(temps[crossing]))
```

```
## [1] "X2011"
```