# Homework 2

5/27/2020

## Question 4.1

**Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use.**

In my work doing loyalty analysis for a retail company, one of the projects I worked on was to build segmentation for their members that shopped in their beauty category so they could understand their customers' purchase behaviour and use the segmentation to drive incremental sales through targeted offers and mass promotions. At the time, I used a simpler RFM (recency, frequency, monetary) segmentation, but a clustering model would have been a much more sophisticated and statisically sound solution to segmenting customers.

Some of the indicators I would include:

- % of customer's beauty sales penetration in L12M
- % of customer's beauty transaction penetration in L12M
- Total beauty sales in L12M
- Total transactions in L12M
- Beauty Categories Shopped

## Question 4.2

**Use iris data set the R function kmeans to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type.**

Before starting the problem, I want to explore the data set first.

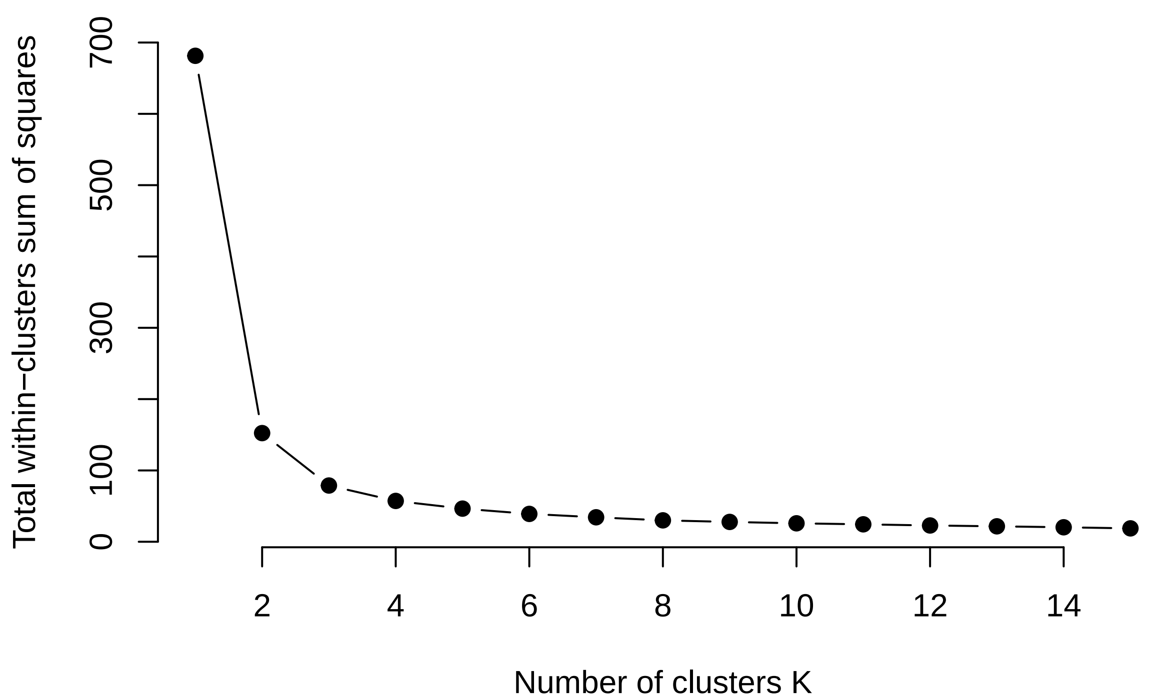
```r
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 
```

To choose the optimal value of k, I will create an elbow diagram. Since the data points are segmented, I am doing this to validate that the optimal k is 3.

```
set.seed(8010)
k_max <- 15
data <- iris[,1:4]
elbow_plot <- sapply(1:k_max,
              function(k){kmeans(data, k, nstart=50,iter.max = 15 )$tot.withinss})

plot(1:k_max, elbow_plot,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




As expected, the elbow of the diagram is where k=3 so we'll use that as the optimal k.

Now we have to decide what predictors to use. There are many combinations of 2, 3 or 4 predictors I could choose from. I want to look at the correlation matrix between the predictors to see if we need to keep all 4 predictors.

```
cor(iris[,1:4])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

It looks like all predictors except Sepal Width have strong positive correlations (ie - are greater than 0.8). Among the three predictors left, Sepal Length only has an r value of 0.82 and 0.87 with Petal Length and Petal Width, respectively. But Petal Length and Petal Width ahve a very high r value of 0.96 so we definitely want to keep those predictors.

I will try 2 kmeans models: one using all 3 and another using only 2 (Petal Width and Petal Length). I'll choose the one with the best accuracy and within cluster sum of squares.

```
set.seed(0)
kmeans_model <- kmeans(iris[c("Sepal.Length","Petal.Length", "Petal.Width")], 3, nstart = 20)
kmeans_model
```

```
## K-means clustering with 3 clusters of sizes 62, 50, 38
##
## Cluster means:
##   Sepal.Length Petal.Length Petal.Width
## 1     5.901613     4.393548    1.433871
## 2     5.006000     1.462000    0.246000
## 3     6.850000     5.742105    2.071053
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3 3 1 3 3 3 3 3
## [112] 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3 3 3 1 3 3 3 3 1 3 3 3 1 3 3 3 1 3
## [149] 3 1
##
## Within cluster sum of squares by cluster:
## [1] 34.46613  8.11020 20.76579
##  (between_SS / total_SS =  90.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Now that I have my clusters determined, I have to assign a flower to them. I'll do this by comparing averages of the labeled data to the output of my model.

```
aggregate(iris[c("Sepal.Length", "Petal.Length", "Petal.Width", "Sepal.Width")], list(iris$Species), mea
```

```
##      Group.1 Sepal.Length Petal.Length Petal.Width Sepal.Width
## 1     setosa        5.006        1.462       0.246       3.428
## 2 versicolor        5.936        4.260       1.326       2.770
## 3  virginica        6.588        5.552       2.026       2.974
```

Based on the averages above, Cluster 1 = Versicolor, Cluster 2 = Setosa and Cluster 3 = Virginica Showing the confusion matrix and calculating accuracy:

```
cluster_res <- kmeans_model$cluster
cluster_res <- mapvalues(cluster_res, c(1, 2, 3), c("versicolor", "setosa", "virginica"))

conf_table = table(cluster_res, iris$Species)
conf_table
```

```
##
## cluster_res  setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48        14
##   virginica       0          2        36
```

```r
accuracy = sum(cluster_res == iris$Species) / nrow(iris)
accuracy
```

```
## [1] 0.8933333
```

Repeating the steps with only Petal Length and Petal Width as predictors:

```r
set.seed(0)
kmeans_model2 <- kmeans(iris[c("Petal.Length", "Petal.Width")], 3, nstart = 20)
kmeans_model2
```

```
## K-means clustering with 3 clusters of sizes 52, 50, 48
##
## Cluster means:
##   Petal.Length Petal.Width
## 1     4.269231    1.342308
## 2     1.462000    0.246000
## 3     5.595833    2.037500
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 1 3 3 3 3
## [112] 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3
## [149] 3 3
##
## Within cluster sum of squares by cluster:
## [1] 13.05769  2.02200 16.29167
##  (between_SS / total_SS =  94.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
cluster_res <- kmeans_model2$cluster
cluster_res <- mapvalues(cluster_res, c(1, 2, 3), c("versicolor", "setosa", "virginica"))

conf_table = table(cluster_res, iris$Species)
conf_table
```

```
##
## cluster_res  setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         4
##   virginica       0          2        46
```

```r
accuracy = sum(cluster_res == iris$Species) / nrow(iris)
accuracy
```

```
## [1] 0.96
```

Using only these two predictors, the within cluster sum of squares and accuracy is higher than using these two + Sepal Length. It looks like the model has some difficulty distinguishing between the versicolor and virginica series, likely because the average petal length and width for those two species are much closer to each other than to setosa.

### Conclusion

Based on the analysis above, my suggested value of k is 3 and the best predictors are Petal Width and Length. This clustering configuration has an accuracy of 96%.

## Question 5.1

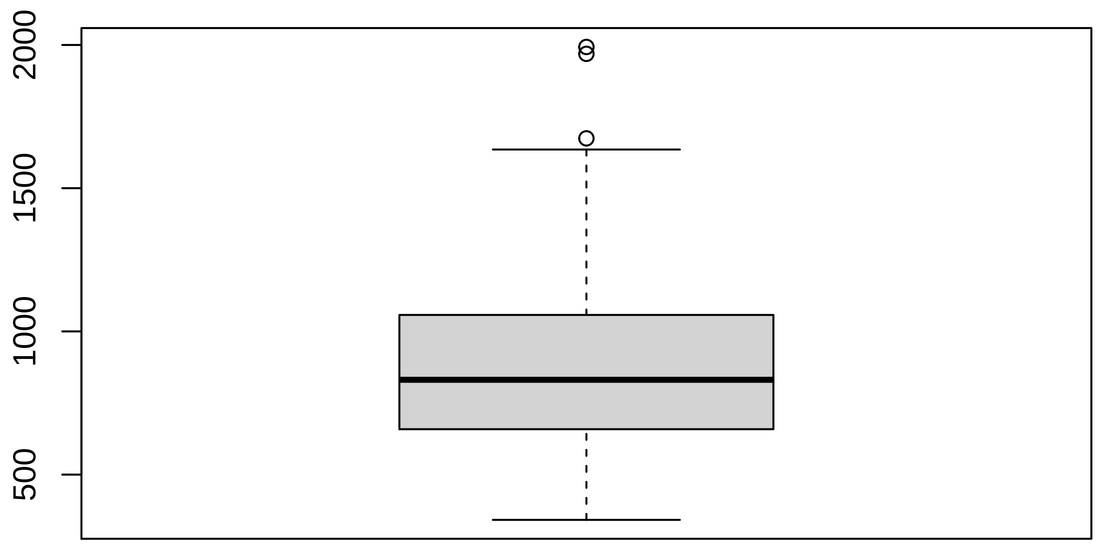
**Using crime data, test to see whether there are any outliers in the last column (number of crimes per 100,000 people). Use the grubbs.test function in the outliers package in R.**

Importing the data and looking at descriptive stats:

```r
data_uscrime <- read.delim("http://www.statsci.org/data/general/uscrime.txt")
summary(data_uscrime)
```

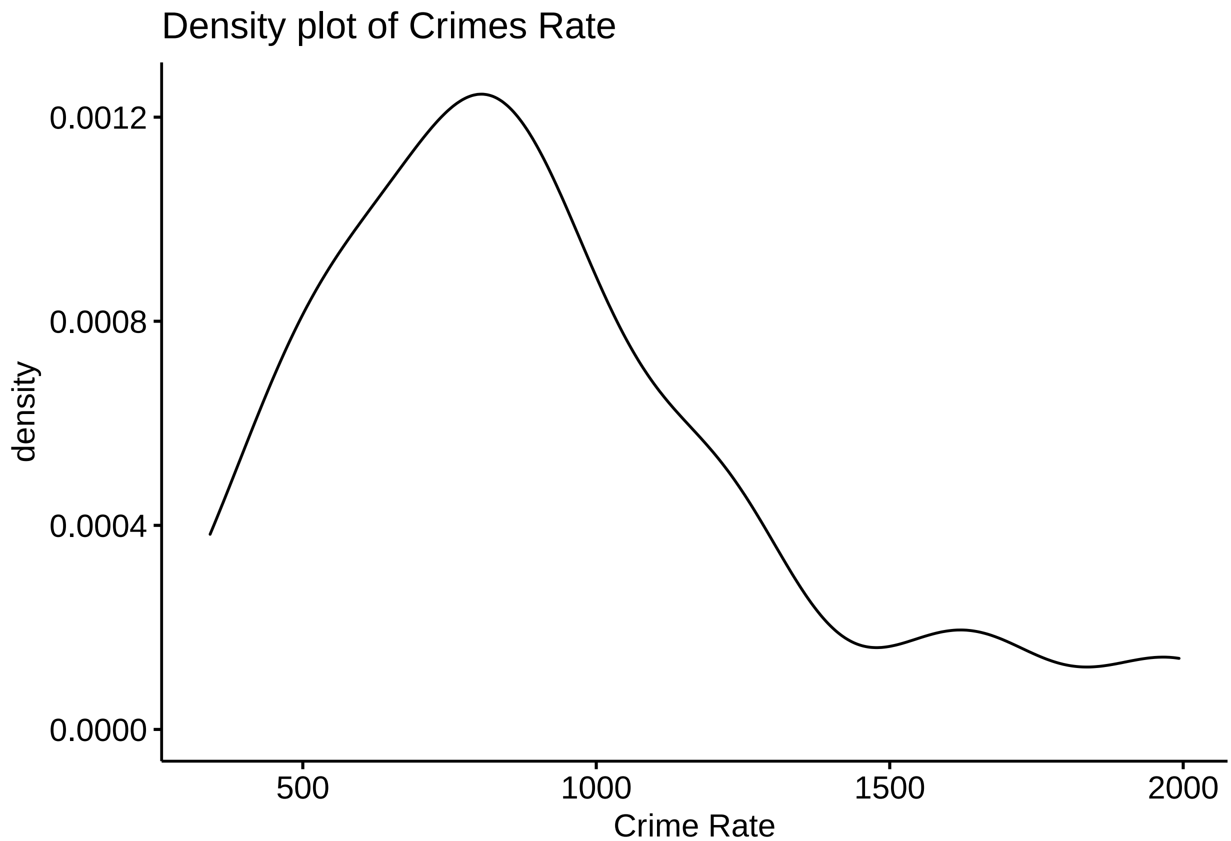
```
##        M               So               Ed             Po1
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80
##  Mean   :13.86   Mean   :0.3404   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
boxplot(data_uscrime[,16])
```




Based on the box and whisker plot above, it looks like there might be a few outliers with a high crime rate. Before doing the Grubbs test, we have to check if the data is normally distributed.

```r
ggdensity(data_uscrime$Crime,
          main = "Density plot of Crimes Rate",
          xlab = "Crime Rate")
```




The data is definitely not normally distributed as it has a long right tail and is positvely skewed. I am using the Shapiro-Wilk test to check statistically that Crime Rate data is different from a normal distribution.

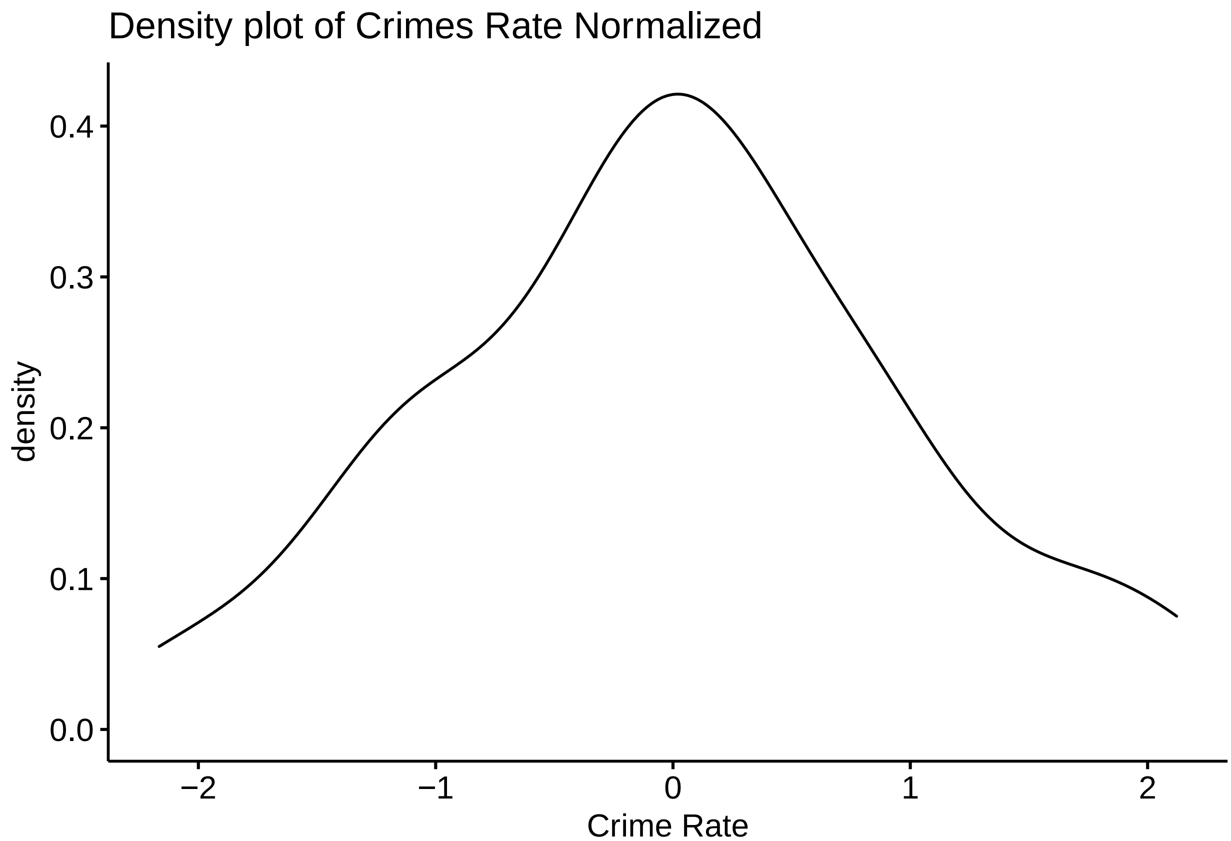
```
shapiro.test(data_uscrime$Crime)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_uscrime$Crime
## W = 0.91273, p-value = 0.001882
```

Since the p value is less than 0.05, we reject the null hypothesis meaning that the Crime data is indeed different from a normal distribution.

Transforming data to normal distribution and performing the Shapiro Wilk test to check:

```
norm <- bestNormalize(data_uscrime$Crime)
data_uscrime$Crime_norm <- predict(norm)
ggdensity(data_uscrime$Crime_norm,
          main = "Density plot of Crimes Rate Normalized",
          xlab = "Crime Rate")
```




```
shapiro.test(data_uscrime$Crime_norm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_uscrime$Crime_norm
## W = 0.98709, p-value = 0.8778
```

Here, we can visually see that the data is normally distributed and the p value > 0.05.

Now that the data is normalized, I can perform the Grubbs test to check for two outliers in opposite tails.

```
grubbs.test(data_uscrime$Crime_norm, type = 11)
```

```
##
##  Grubbs test for two opposite outliers
##
## data:  data_uscrime$Crime_norm
## G = 4.28791, U = 0.80013, p-value = 1
## alternative hypothesis: -2.16544076509641 and 2.12246850934268 are outliers
```

The p value in the Grubbs test is greater than 0.05 which means we fail to reject the null hypothesis and these two points are not considered outliers. These points correspond to the states with min and max Crime rate, as per the summary stats pulled in the beginning of the question. Any data points within this min/max range then are not outliers either. This leads me to believe that there are no statistical outliers in this data.

## 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?**

In my everyday life, I could use a change detection model to track my spending in my chequing account. I have an account that waives the monthly fee if I keep a minimum amount of money in the account. This model would be useful to tell me as my balance is decreasing, when I should stop spending money so that I don't have to pay the monthly fee.

The threshold would be the minimum amount of money I need to keep in order to waive the fee. The critical value would be $100 so that the model will notify me when my chequing balance is within $100 of the minimum threshold.

## 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.**

Please see tab 6.2.1 in the attached Excel sheet.

For each year, I chose the average temperature in July as the baseline for summer temperature. For each day, I calculated S_t = max(0, S_t-1 + (mean - X_t - C)) and played around with different combinations of C and T that give realistic results of when we would expect temperatures to decrease (mid to late August, early September). I settled on C = 2 and T = 30.

Here are the dates in each year S_t is above T = 30:

| Year | Date |
|---|---|
| 1996 | 01-Aug |
| 1997 | 09-Aug |
| 1998 | 14-Aug |
| 1999 | 14-Jul |
| 2000 | 26-Aug |
| 2001 | 03-Sep |
| 2002 | 31-Aug |
| 2003 | 11-Sep |
| 2004 | 12-Aug |
| 2005 | 06-Oct |
| 2006 | 03-Sep |
| 2007 | 20-Sep |
| 2008 | 23-Aug |
| 2009 | 01-Sep |
| 2010 | 05-Jul |
| 2011 | 06-Sept |
| 2012 | 10-Aug |
| 2013 | 17-Aug |
| 2014 | 25-Sept |
| 2015 | 04-Jul |

Based on these results, temperatures start cooling down mid August to early September. There are a few anomalys like 2010 and 2015 where it looks like the temperature cools very early in July. Upon further investigation, the temperature in the beginning ofJuly for each of these years is low compared to the average used. It's possible that summer had a late start in those years as it took longer in these years for the temperature to increase.

### Question 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).**

Please see tab 6.2.2 in the attached Excel sheet.

For this problem, I will be calculating S_t = max(0, S_t-1 + (X_t - mean - C)) for each year in order to determine whether or not Atlanta's summer climate has gotten warmer over 20 years and what year it started to increase.

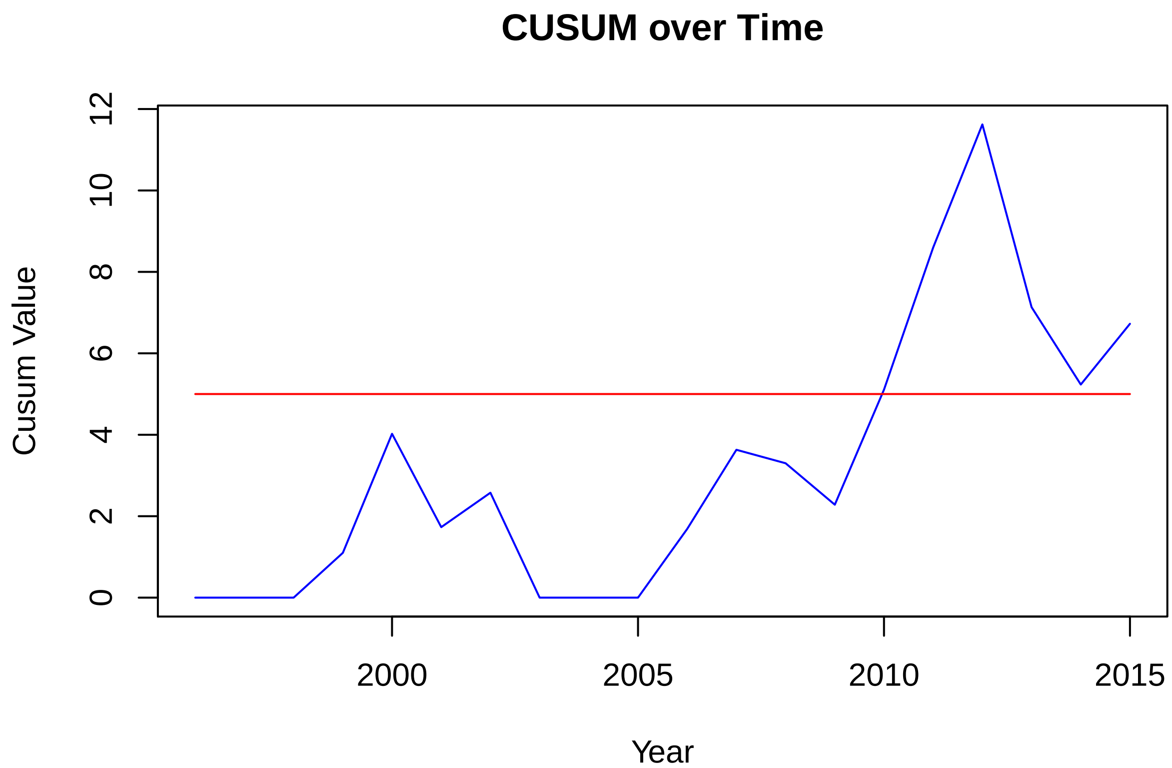
First, I need to determine a static timeframe that I can compare the average temperature in each year. I'm going to take the median of the dates in the table above to use as the unofficial end date of summer and compare each year's average temperature between July 1 and August 20.

I decided to keep C = 0. Higher values of C desensitizes the model too much to detect any changes.

Trying to choose a T value, I first tried T = 4. With this threshold, the model told me that summer temperatures start increasing in 2000. This is true only for the year 2000; the summer temperatures in subsequent years are lower than that in 2000. I decided to use T = 5 as my threshold instead. As a result, the model tells me that the temperature in Atlanta has indeed gotten warmer in the summer starting in 2010.

```
cusum = c(0,    0,  0,  1.09999999999999,   4.0235294117647,    1.7313725490196,    2.57647058823528,
years = c(1996, 1997,   1998,   1999,   2000,   2001,   2002,   2003,   2004,   2005,   2006,   2007,
t = c(5,    5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5)

plot(x = years, y = cusum, type = "l", col = "blue", main = "CUSUM over Time", xlab = "Year", ylab = "Cu
lines(x = years, y = t, type = "l", col = "red")
```


CUSUM over Time
Cusum Value
0
2
4
6
8
10
12
2000
2005
2010
2015
Year