# ISYE6501 HW Wk2 Solution

**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.*

One of projects we are working on, within a global organization, different Business units need to ship packages between different locations. Need to cluster the shipments in order to choose the best logistics service provider for different types of shipments. Predictors can be: weight, service level (express, regular, etc.), location (departure, arrival), office names/department names

**Question 4.2**

The iris data set iris.txt contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Iris ). The response values are only given to see how well a specific method performed and should not be used to build the model. Use 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.

```
# Load package
library(cluster)
library(ggplot2)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
# Load Iris data and explore
df_iris = iris
dim(df_iris)
```

```
## [1] 150   5
```

```
str(df_iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(df_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
##
##
##
```

```
head(df_iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

After exploring the dataset, those 4 variables have different value ranges. Need to scale the variables before building the model

```
# data cleanup: scale
df_scale <- df_iris

for (i in seq(4)){
  df_scale[,i]<-(df_scale[,i]-min(df_scale[,i]))/(max(df_scale[,i])-min(df_scale[,i]))}

head(df_scale)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1   0.22222222   0.6250000   0.06779661  0.04166667  setosa
## 2   0.16666667   0.4166667   0.06779661  0.04166667  setosa
## 3   0.11111111   0.5000000   0.05084746  0.04166667  setosa
## 4   0.08333333   0.4583333   0.08474576  0.04166667  setosa
## 5   0.19444444   0.6666667   0.06779661  0.04166667  setosa
## 6   0.30555556   0.7916667   0.11864407  0.12500000  setosa
```

### Create dataframe to build K-means model

```
# Remove the last variable Species from scaled dataframe to build the K-means model
df_scale_km <- df_scale[,0:4]

# set seed
set.seed(4)
```

```r
# Build k-means model using k = 2-6

k2 <- kmeans(df_scale_km, centers=2, nstart=25)
k3 <- kmeans(df_scale_km, centers=3, nstart=25)
k4 <- kmeans(df_scale_km, centers=4, nstart=25)
k5 <- kmeans(df_scale_km, centers=5, nstart=25)
k6 <- kmeans(df_scale_km, centers=6, nstart=25)
```

After building the K-means model, use the response column to compare the result

```r
table(k2$cluster,df_scale[,5])
```

```
##
##     setosa versicolor virginica
##   1     50          0         0
##   2      0         50        50
```

```r
table(k3$cluster,df_scale[,5])
```

```
##
##     setosa versicolor virginica
##   1      0          3        36
##   2     50          0         0
##   3      0         47        14
```

```r
table(k4$cluster,df_scale[,5])
```

```
##
##     setosa versicolor virginica
##   1      0         23        19
##   2      0          0        29
##   3      0         27         2
##   4     50          0         0
```

```r
table(k5$cluster,df_scale[,5])
```

```
##
##     setosa versicolor virginica
##   1      0         23        19
##   2      0         27         2
##   3     22          0         0
##   4      0          0        29
##   5     28          0         0
```

```r
table(k6$cluster,df_scale[,5])
```

```
##
##     setosa versicolor virginica
##   1      0         27         2
##   2     22          0         0
```

```
##    3      0          0         19
##    4      0         23         17
##    5     28          0          0
##    6      0          0         12
```

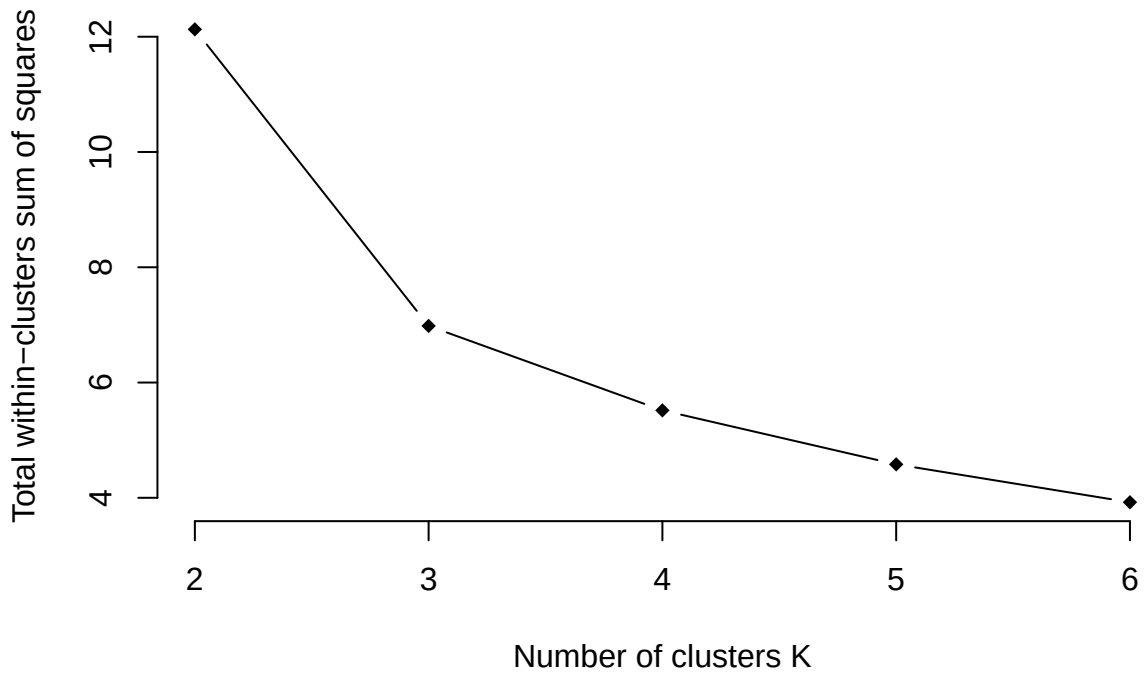
From above, we can see for k<4, the setosa can be clustered all correct. Since we know the response, from the response data, it is confirmed K=3 would be a better result.

To evaluate the total distances to verify K

```
k.values <- 2:6
k.values
```

```
## [1] 2 3 4 5 6
```

```
# extract total within-cluster sum of squares for k clusters
wss_values <- rep(0,5)
for (k in k.values){
  wss_values[k-1]=kmeans(df_scale_km, k, nstart=25 )$tot.withinss
}

# Draw elbow diagram
plot(k.values, wss_values,
       type="b", pch = 18, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
```



From the elbow diagram, it shows the best K would be 3, for k=4-6, the within-cluster sum of squares does not remove effectively. **In conclusion, the k=3 is the best cluster solution.**

### 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.

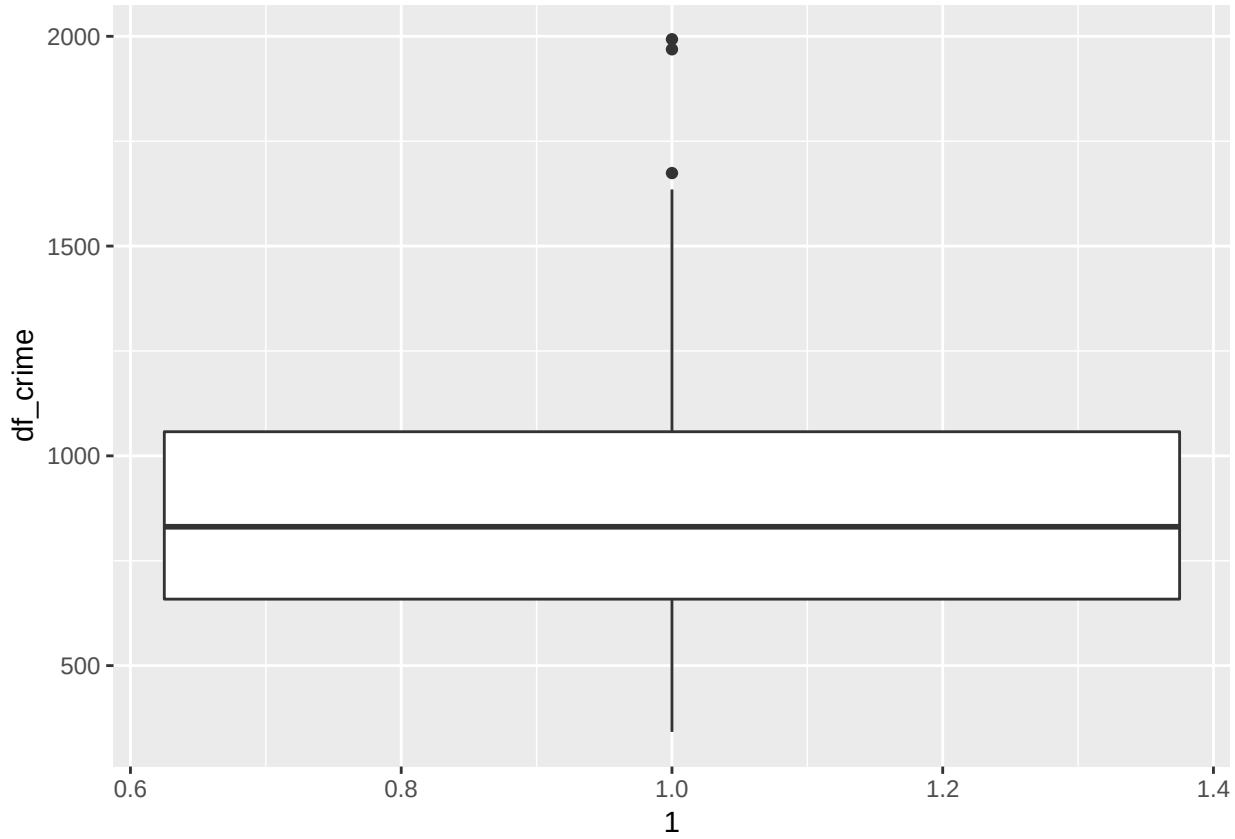
```
# Load package in R
library(outliers)

# Load txt data downloaded from website
df_raw <- read.table('Wk2/uscrime.txt',header=TRUE)
dim(df_raw)
```

```
## [1] 47 16
```

```
# only take the last column to check for outlier
df_crime <- df_raw[,16]
summary(df_crime)
```

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

```
# create boxplot to visulize the data
qplot(y=df_crime, x= 1, geom = "boxplot")
```




From the boxplot above, the two highest points might be outliers. Need to do the Grubb test to verify.

```
# start with two outlier test that set type as 11
test_11 <- grubbs.test(df_crime, type = 11)
test_11
```

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

With p-value = 1, it means it is impossible to have two outliers in the data. So use type = 10 to test for one outlier

```
# use type = 10 to test for one outlier
test_10 <- grubbs.test(df_crime, type = 10)
test_10
```

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

When level of significance is set as 0.05, the P-value is still above the level of significance. In this case, the highest point is not a outlier.

**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?*

Change Detection model can be used to evaluate my daily commute time. My daily commute is over 30 miles and usually around 1 hr. It would help me to identify the best route and best time to leave from home/work. And detect the change when the commute time is getting too long then I can change my route or try different time to leave.

For the CUSUM technique, the critical value should be large to offset the impact of one or two exceptional days with accidents or special occasions. Meanwhile, the threshold should be small to be sensitive to detect the change if the extra commute time happens for many days.

**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).

**See solution of 6.2 in attached Excel Sheet**