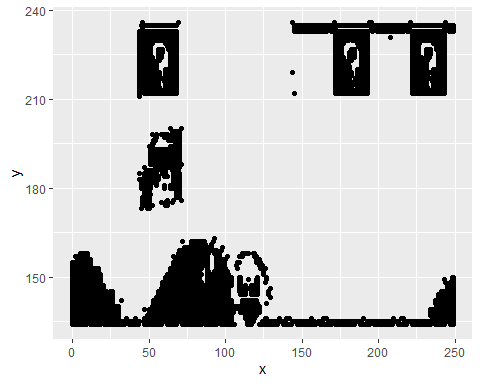
Assignment 16

Kolekar, Shilpa

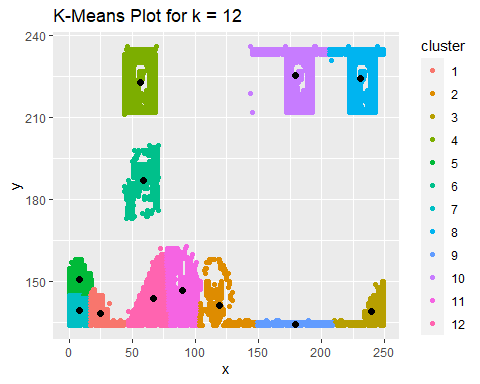
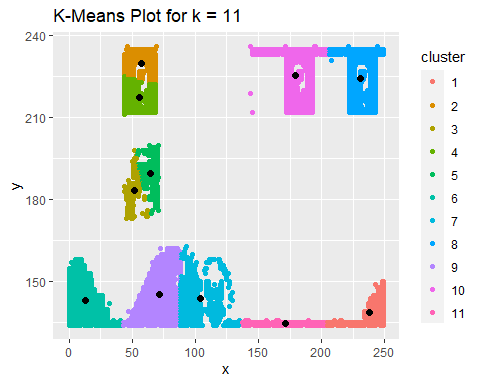
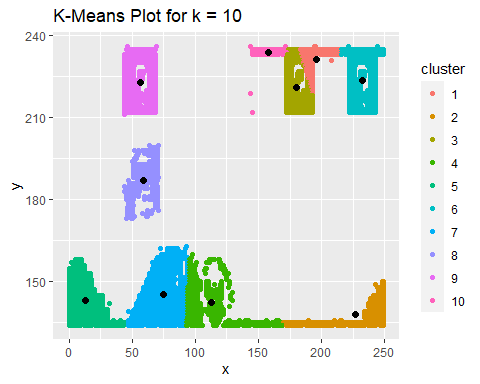
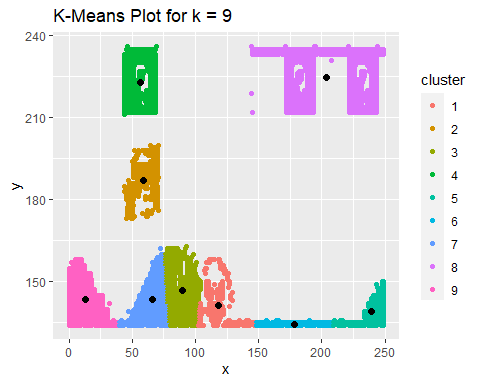
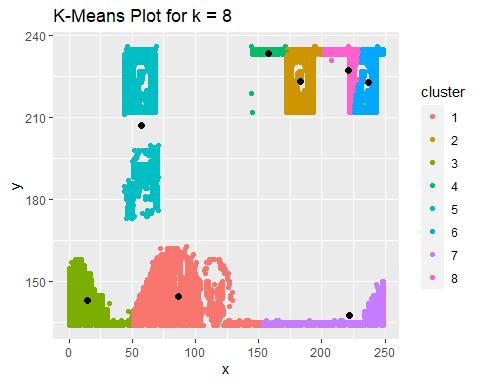
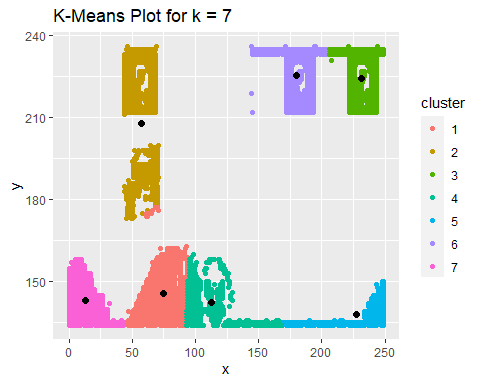
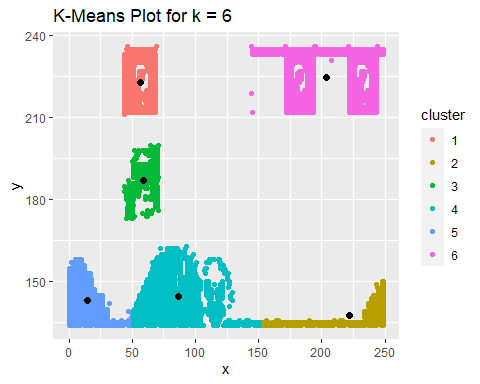
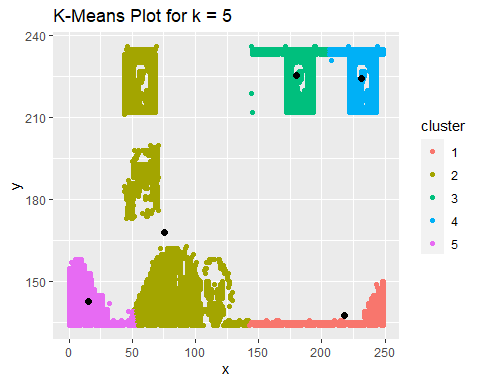
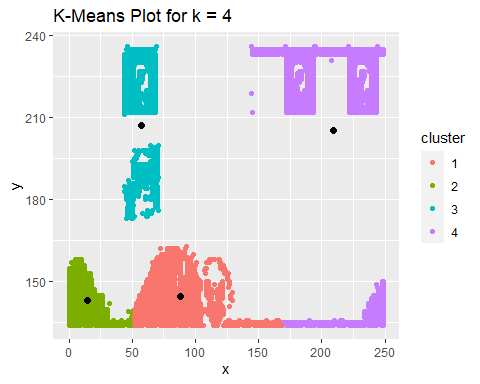
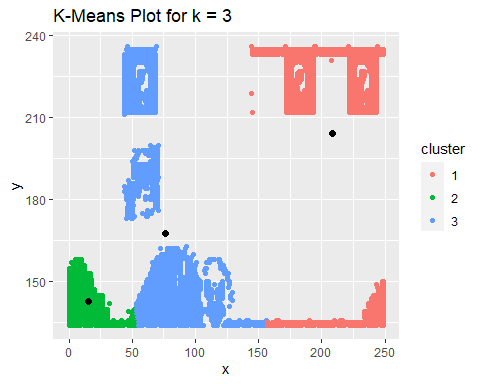
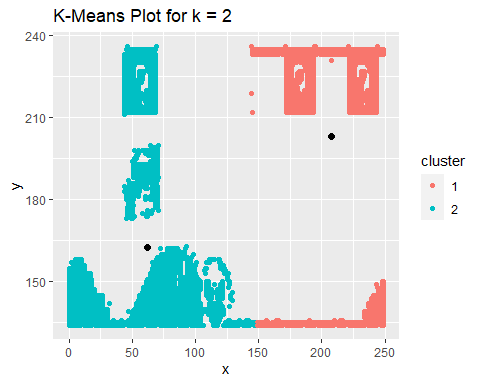
November 1st, 2020

# a. Plot the dataset using a scatter plot.

# working directory path to access binary-classifier-data.CSV file from local drive  
setwd("C:/Users/shilp/Documents/GitHub/dsc520/data")  
  
# Read the files `clustering-data.csv`   
cluster\_df <- read.csv("clustering-data.csv")  
  
library('ggplot2')  
ggplot(cluster\_df, aes(x=x, y=y)) + geom\_point()

 # b. Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

set.seed(20)  
total\_withinss <- NULL  
avg\_distance <- NULL  
for (n in 2:12)  
{  
 temp <- cluster\_df  
 clusters <- kmeans(temp, n)  
 temp$cluster <- as.factor(clusters$cluster)  
 plot = ggplot(data=temp, aes(x=x, y=y, color = cluster)) + geom\_point() + geom\_point(data = as.data.frame(clusters$centers), color = "black", size = 2) + ggtitle(paste("K-Means Plot for k =", n))  
 print(plot)  
   
 # Needed for later part of the assignment  
 x\_distance <- clusters$centers[temp$cluster] - temp$x  
 y\_distance <- clusters$centers[temp$cluster] - temp$y  
 tot\_distance <- sqrt((x\_distance \*\* 2) + (y\_distance \*\* 2))  
 avg\_distance <- c(avg\_distance, mean(tot\_distance))  
 total\_withinss <- c(total\_withinss, clusters$tot.withinss)  
}



# c. As k-means is an unsupervised algorithm, you cannot compute the accuracy as there are no correct values to compare the output to. Instead, you will use the average distance from the center of each cluster as a measure of how well the model fits the data. To calculate this metric, simply compute the distance of each data point to the center of the cluster it is assigned to and take the average value of all of those distances.

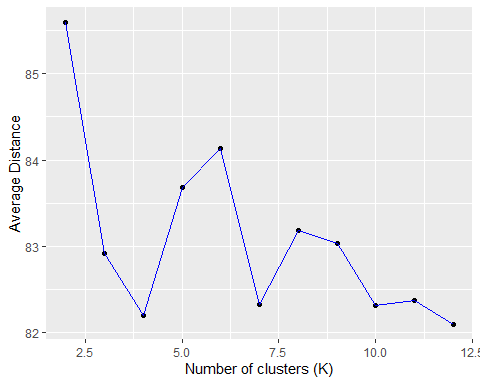
# Calculate this average distance from the center of each cluster for each value of k and plot it as a line chart where k is the x-axis and the average distance is the y-axis.

# One way of determining the “right” number of clusters is to look at the graph of k versus average distance and finding the “elbow point”. Looking at the graph you generated in the previous example, what is the elbow point for this dataset?

k\_values <- c(2:12)  
distance\_df <- data.frame(k\_values, avg\_distance)  
distance\_df

## k\_values avg\_distance  
## 1 2 85.59601  
## 2 3 82.91766  
## 3 4 82.20270  
## 4 5 83.68432  
## 5 6 84.12965  
## 6 7 82.32686  
## 7 8 83.18385  
## 8 9 83.02901  
## 9 10 82.30974  
## 10 11 82.36781  
## 11 12 82.09636

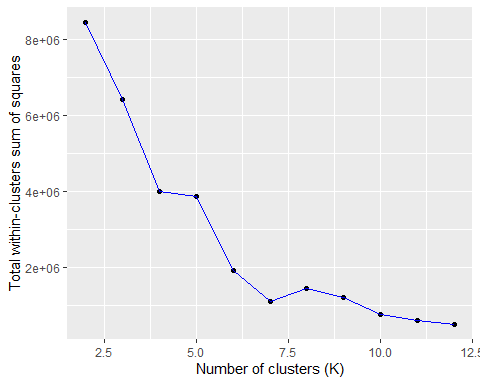
ggplot(data = distance\_df, aes(x=k\_values, y=avg\_distance)) + xlab("Number of clusters (K)") + ylab("Average Distance") + geom\_point() + geom\_line(color = "blue")



withinss\_df <- data.frame(k\_values, total\_withinss)  
withinss\_df

## k\_values total\_withinss  
## 1 2 8443681.1  
## 2 3 6411644.9  
## 3 4 4009678.4  
## 4 5 3856788.5  
## 5 6 1920708.1  
## 6 7 1102869.9  
## 7 8 1436982.5  
## 8 9 1207285.9  
## 9 10 770743.2  
## 10 11 611735.3  
## 11 12 504963.4

ggplot(data = withinss\_df, aes(x=k\_values, y=total\_withinss)) + xlab("Number of clusters (K)") + ylab("Total within-clusters sum of squares") + geom\_point() + geom\_line(color = "blue")



Based on the above plots, the elbow point is 7.