---  
title: "WK9\_Exercise16\_Clustering"  
author: "SumbarajuAditya"  
date: "2/11/2021"  
output: word\_document  
---

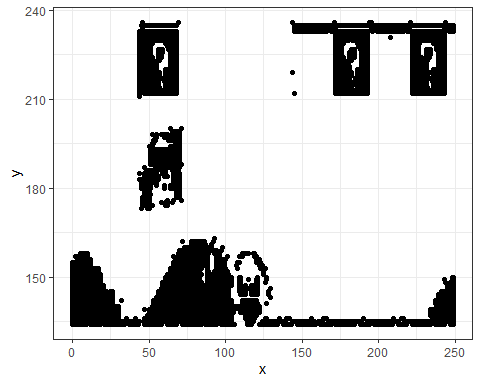
**Dataset** In this problem, you will use the k-means clustering algorithm to look for patterns in an unlabeled dataset. The dataset for this problem is found at C:/BU/DSC520/assignment\_repo/dsc520/data/clustering-data.csv

## Q-a: Plot the dataset using a scatter plot

library(ggplot2)  
ds\_clustering\_df <- read.csv("C:/BU/DSC520/assignment\_repo/dsc520/data/clustering-data.csv")  
str(ds\_clustering\_df)

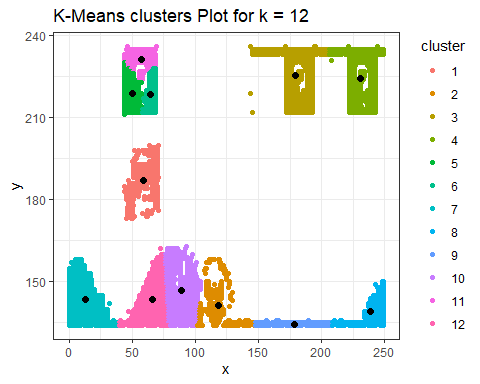
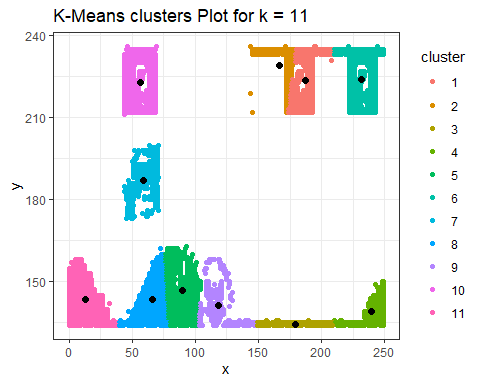
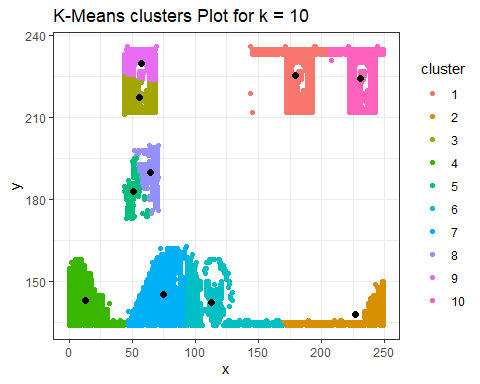
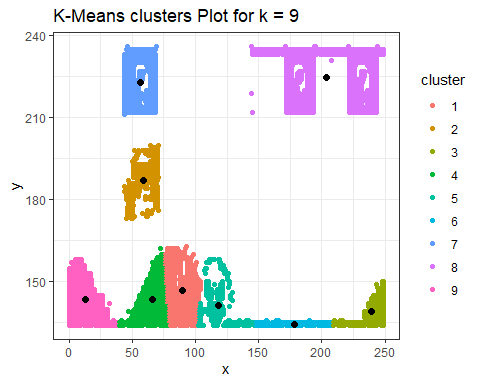
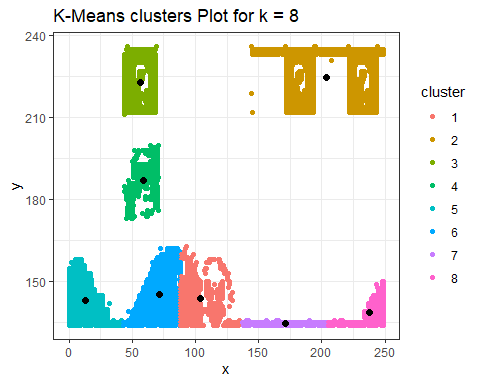
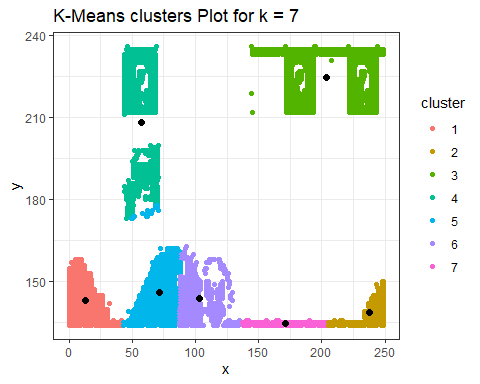
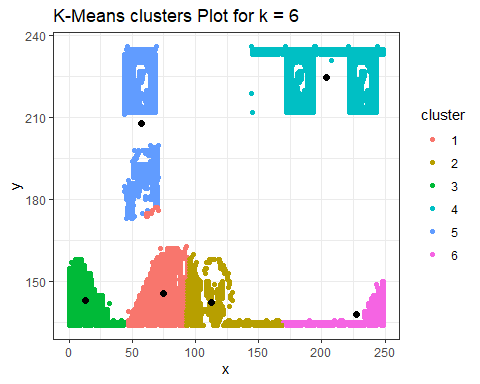
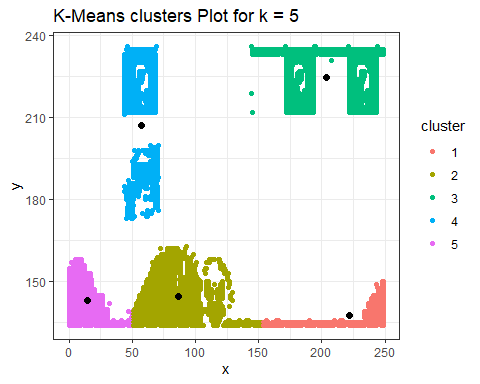
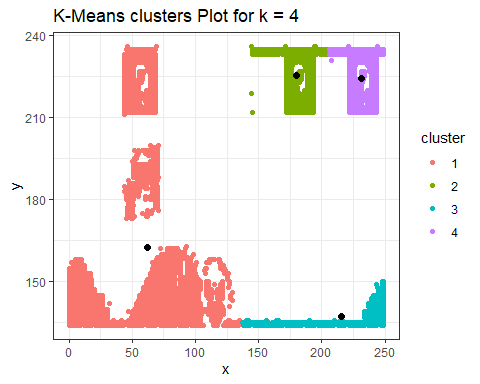
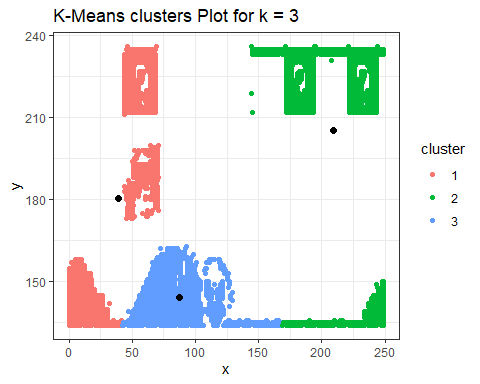
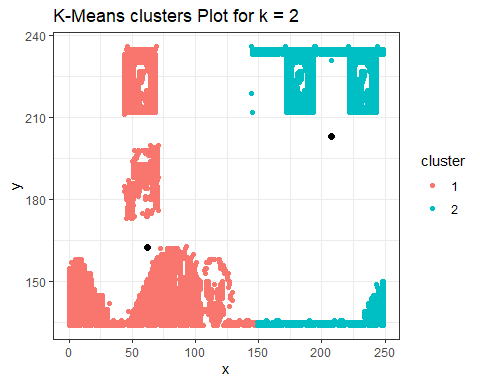
## 'data.frame': 4022 obs. of 2 variables:  
## $ x: int 46 69 144 171 194 195 221 244 45 47 ...  
## $ y: int 236 236 236 236 236 236 236 236 235 235 ...

ggplot(ds\_clustering\_df, aes(x=x, y=y)) + geom\_point() + theme\_bw()



## Q-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.

k\_values <- c(2:12)  
  
#loop to pass K- Values from 2 to 12 and print the plots  
for (i in 1:length(k\_values))  
{  
 ds\_clustering\_df2 <- ds\_clustering\_df  
 k\_clusters <- kmeans(ds\_clustering\_df2, k\_values[i])  
 ds\_clustering\_df2$cluster <- as.factor(k\_clusters$cluster)  
 plots = ggplot(data = ds\_clustering\_df2, aes(x=x, y=y, color = cluster)) + geom\_point() + theme\_bw() + geom\_point(data = as.data.frame(k\_clusters$centers), color = "black", size = 2) + ggtitle(paste("K-Means clusters Plot for k =", k\_values[i]))  
 print(plots)  
  
}



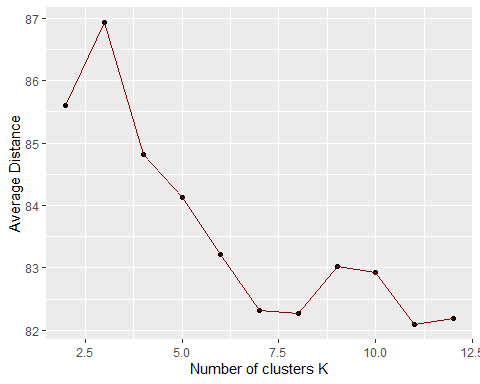
## Q-c

**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.**

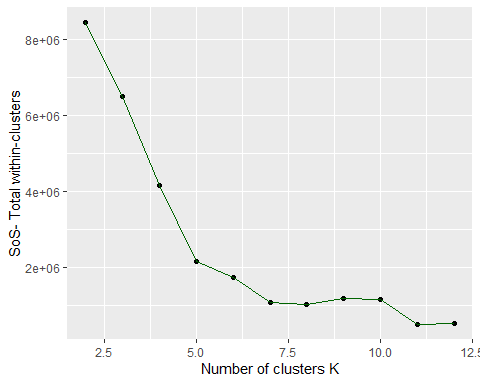
**Calculate this average distance from the center of each cluster for each value of k and plot it as a line chart where k is the x-axis and the average distance is the y-axis.**

**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)  
tot\_withinss <- NULL  
avg\_distance <- NULL  
  
#loop to pass K- Values from 2 to 12 and print the plots  
for (i in 1:length(k\_values))  
{  
 ds\_clustering\_df3 <- ds\_clustering\_df  
 k\_clusters <- kmeans(ds\_clustering\_df3, k\_values[i])  
 ds\_clustering\_df3$cluster <- as.factor(k\_clusters$cluster)  
  
 x\_dist <- k\_clusters$centers[ds\_clustering\_df3$cluster] - ds\_clustering\_df3$x  
 y\_dist <- k\_clusters$centers[ds\_clustering\_df3$cluster] - ds\_clustering\_df3$y  
 tot\_dist <- sqrt((x\_dist \*\* 2) + (y\_dist \*\* 2))  
 avg\_distance <- c(avg\_distance, mean(tot\_dist))  
 tot\_withinss <- c(tot\_withinss, k\_clusters$tot.withinss)  
}  
# get the average distance and find the elbow point  
distance\_elbow\_df <- data.frame(k\_values, avg\_distance)  
ggplot(data = distance\_elbow\_df, aes(x=k\_values, y=avg\_distance)) + xlab("Number of clusters K") + ylab("Average Distance") + geom\_point() + geom\_line(color = "dark red")



withinss\_elbow\_df <- data.frame(k\_values, tot\_withinss)  
ggplot(data = withinss\_elbow\_df, aes(x=k\_values, y=tot\_withinss)) + xlab("Number of clusters K") + ylab("SoS- Total within-clusters") + geom\_point() + geom\_line(color = "dark green")



From the above plots at the graph of k versus average distance and the graph of k versus total sum of squares of within-clusters; The elbow point equates to 5