assignment_11.2.2_ChattapadhyayKausik

Kausik Chattapadhyay

2022-11-11

Assignment

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 data/clustering-data.csv.

Question A:

Plot the dataset using a scatter plot.

Answer for A:

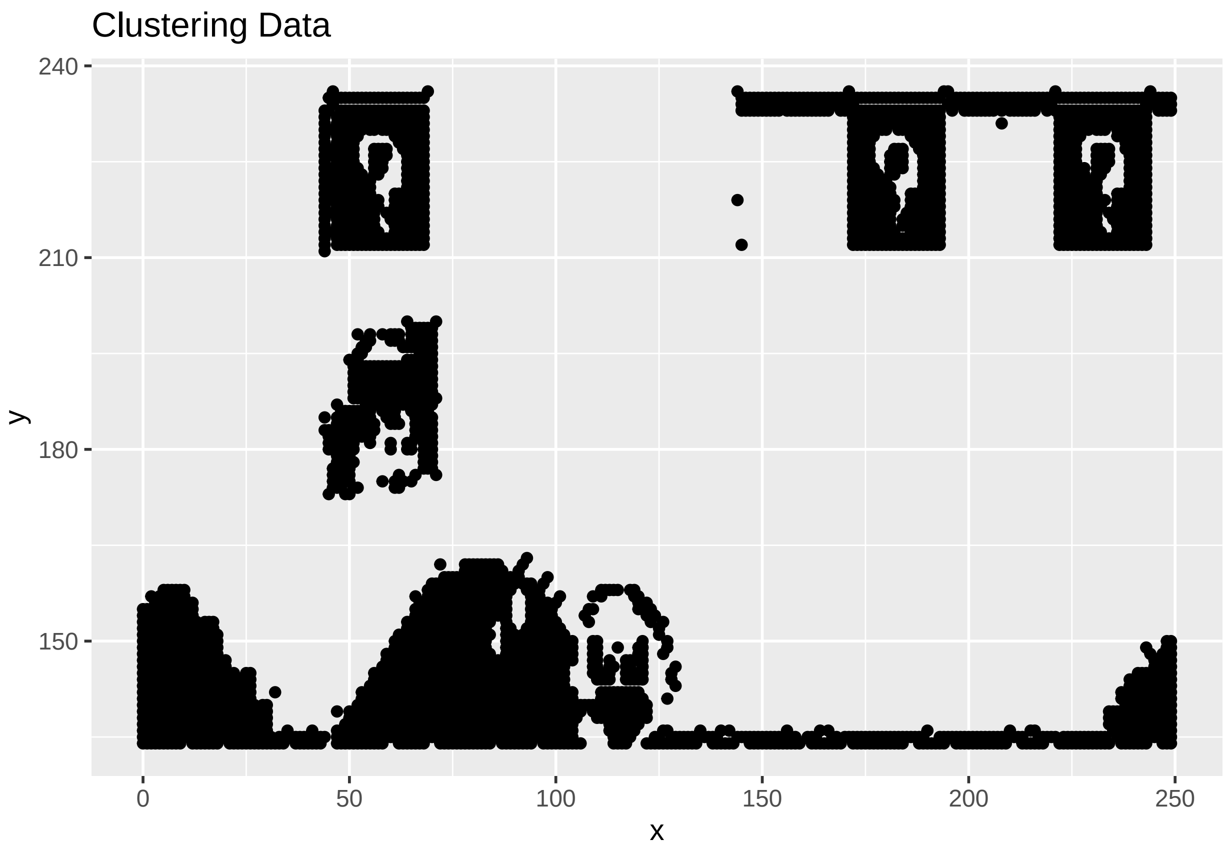
```
## Set the working directory to the root of your DSC 520 directory
setwd("/Users/kausik/desktop/MS Data Science/DSC 520/dsc520-stats-r-assignments")

#Load the `clustering-data.csv`

clustering_data <- read.csv("data/clustering-data.csv")
head(clustering_data)
```

```
##     x   y
## 1  46 236
## 2  69 236
## 3 144 236
## 4 171 236
## 5 194 236
## 6 195 236
```

```
summary(clustering_data)
```

```
##        x               y        
##  Min.   :  0.0   Min.   :134.0  
##  1st Qu.: 56.0   1st Qu.:141.0  
##  Median : 82.0   Median :154.0  
##  Mean   :109.6   Mean   :175.7  
##  3rd Qu.:180.0   3rd Qu.:218.0  
##  Max.   :249.0   Max.   :236.0  
```

```r
#plot the data point
library(ggplot2)
ggplot(data = clustering_data, aes(y = y, x = x)) + geom_point() +
    ggtitle("Clustering Data")
```

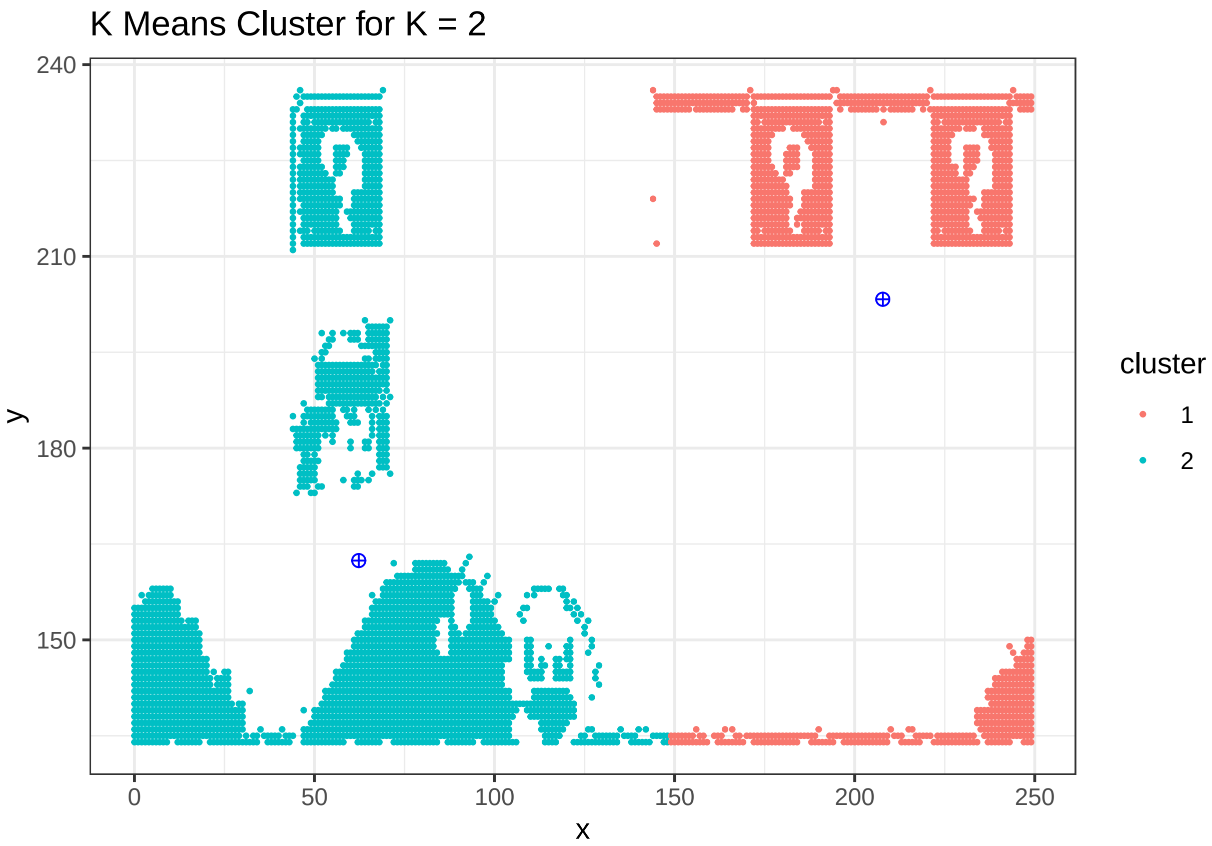


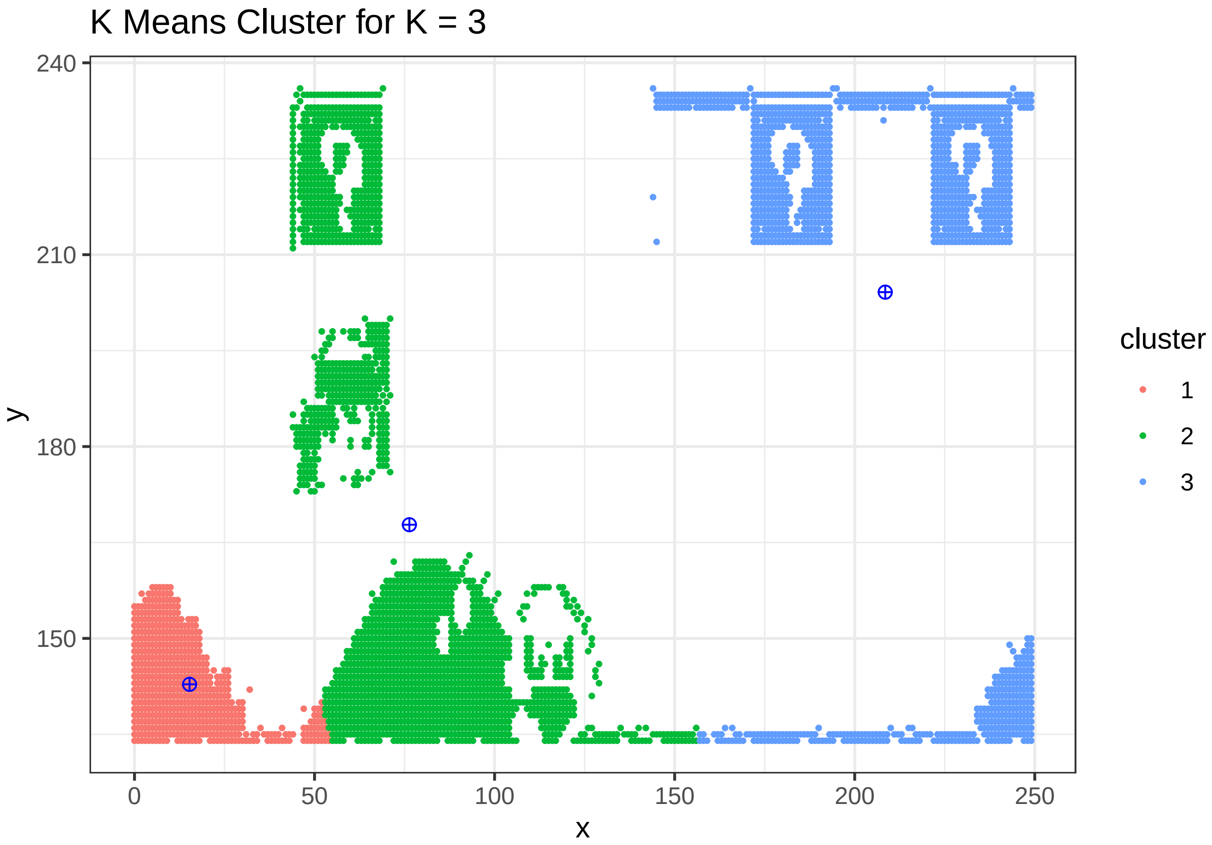
Question B. **Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.**

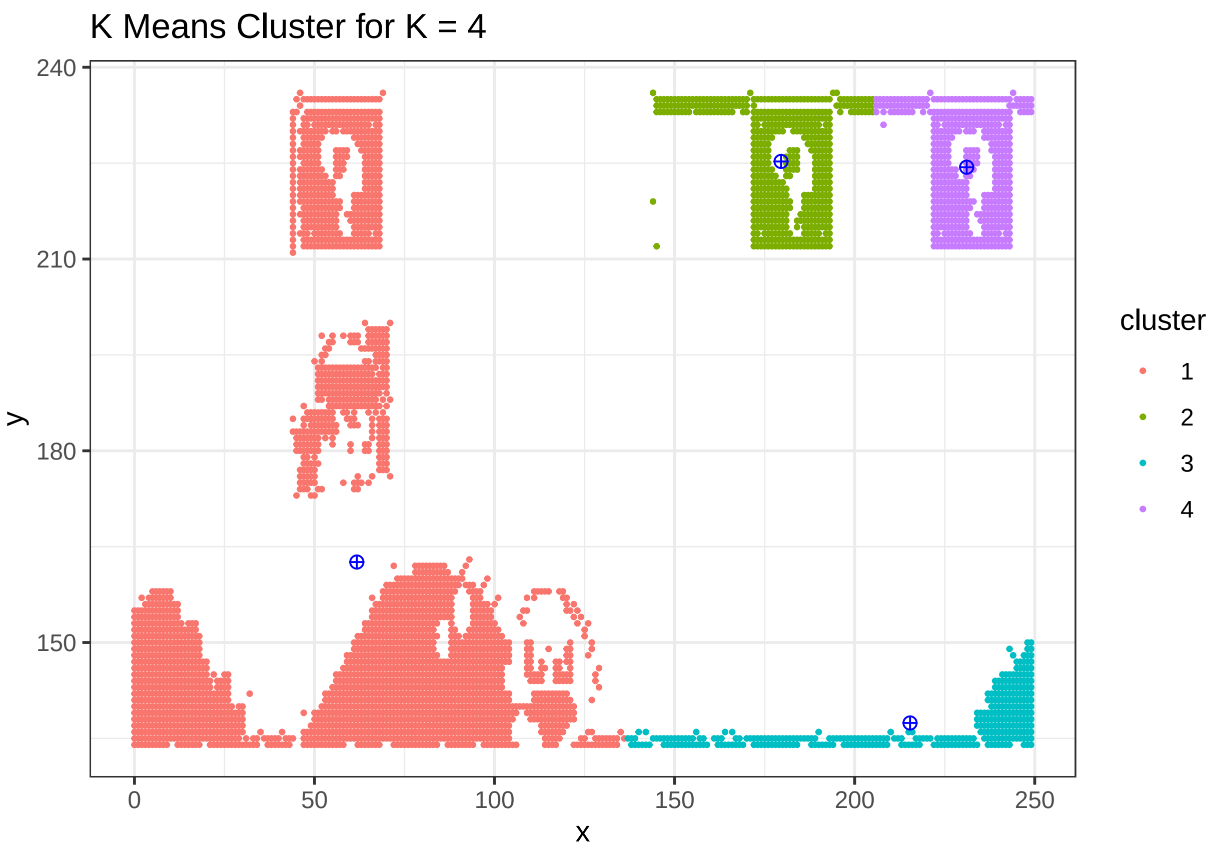
Answer for B:

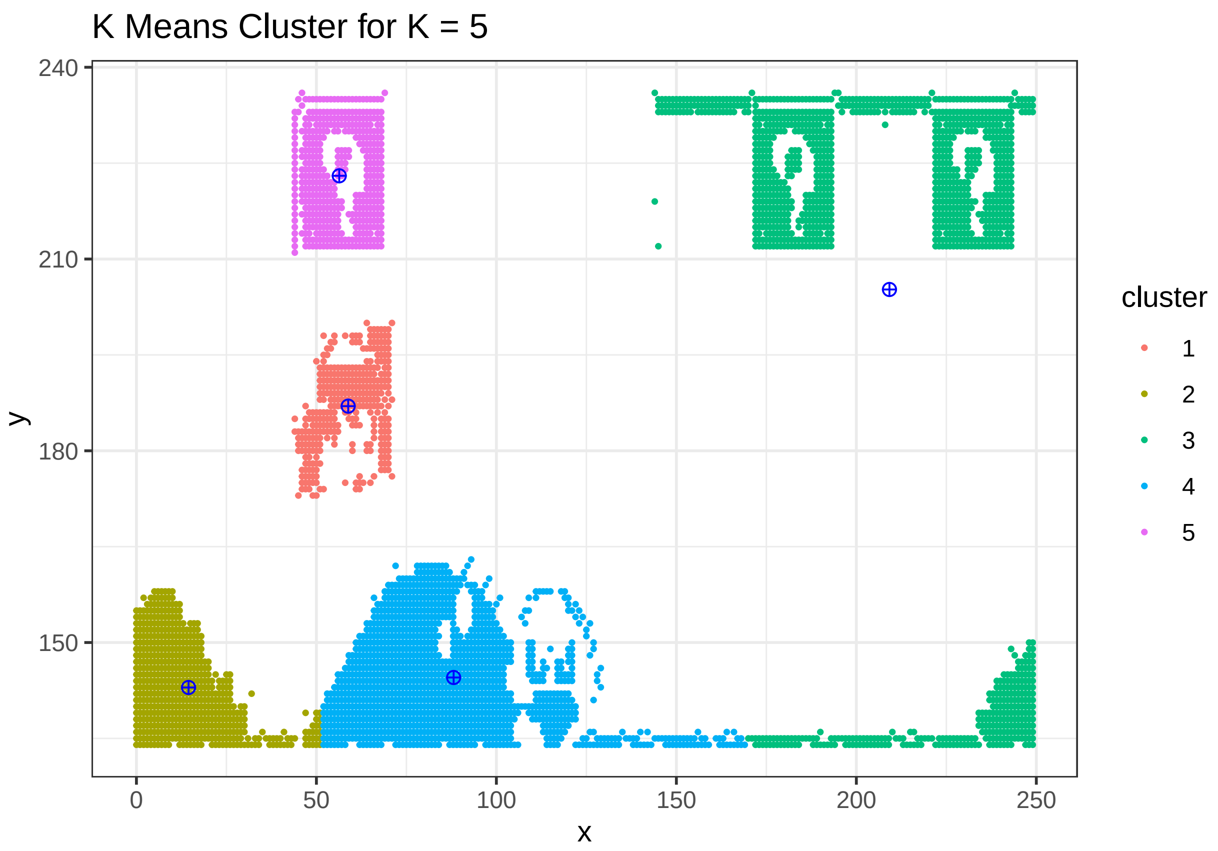
```r
cluster_matrix <- data.matrix(clustering_data)
wss <- (nrow(cluster_matrix) -1) * sum(apply(cluster_matrix,2,var))
total.withinss_values <- NULL
average_distance <- NULL
kmean_values<- NULL
for(i in 2:12){
   wss[i] <- sum(kmeans(cluster_matrix,centers=i)$tot.withinss)

   cdata <- clustering_data
   cdata.kmeanscluster <- kmeans(cdata, i)
   cdata$cluster <- as.factor(cdata.kmeanscluster$cluster)

   p <- ggplot(data = cdata, aes(x = x, y = y, color = cluster)) + geom_point(size = 0.5) + geom_point(
 color = "blue", shape = 10, size = 2) + ggtitle(paste("K Means Cluster for K = ", i, sep ="")) + theme
```

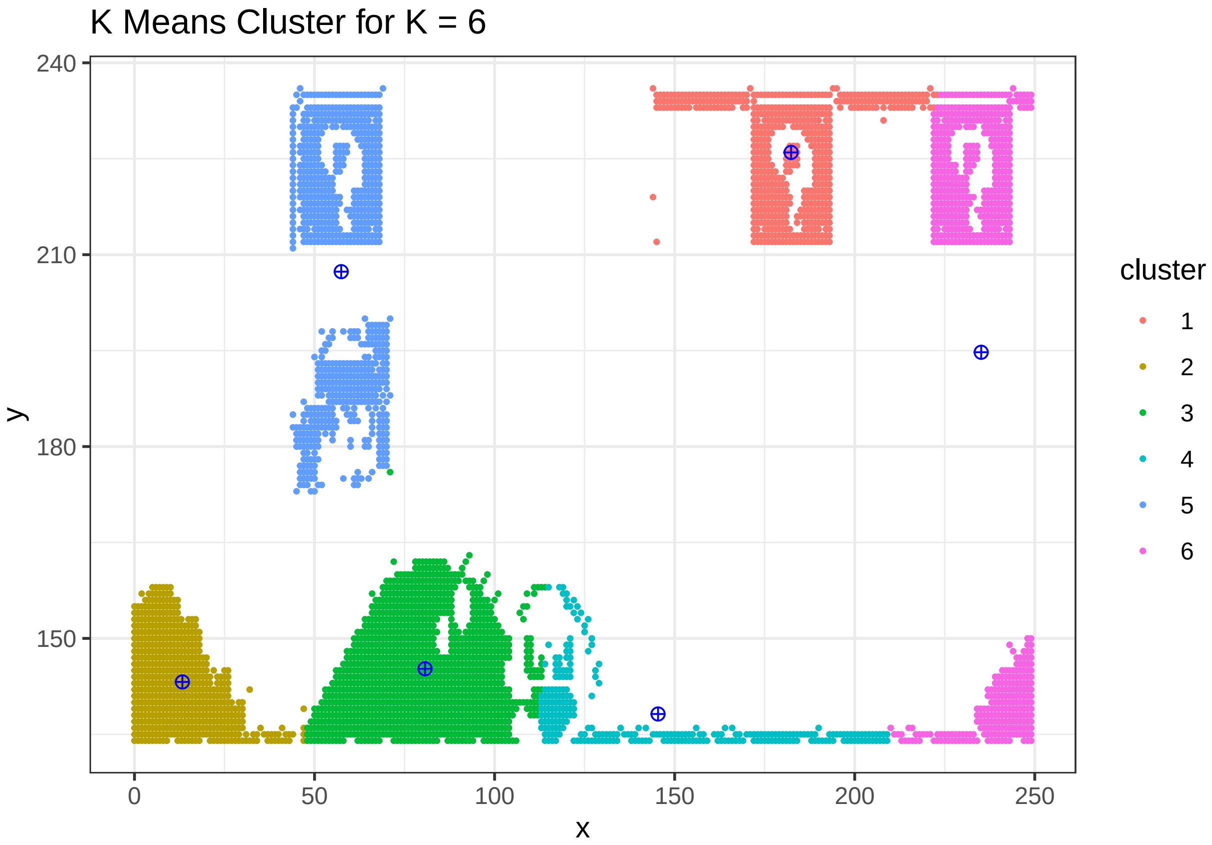
```
  #display graph
  print(p)

  kmean_values<- c(kmean_values, i)
  x.distance <- cdata.kmeanscluster$centers[cdata$cluster] - cdata$x
  y.distance <- cdata.kmeanscluster$centers[cdata$cluster] - cdata$y
  total.distance <- sqrt((x.distance ** 2) + (y.distance ** 2))
  average_distance <- c(average_distance, mean(total.distance))
  total.withinss_values <- c(total.withinss_values, cdata.kmeanscluster$tot.withinss)

}
```

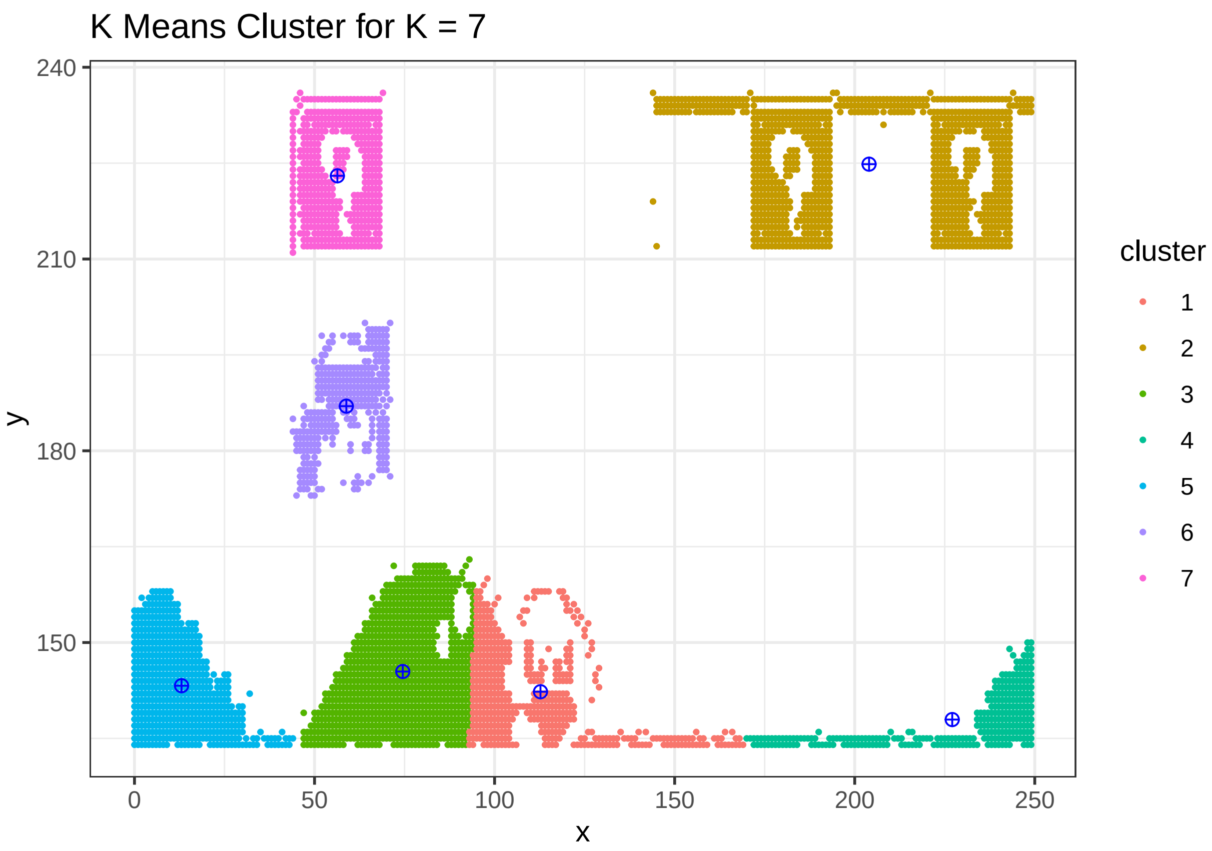


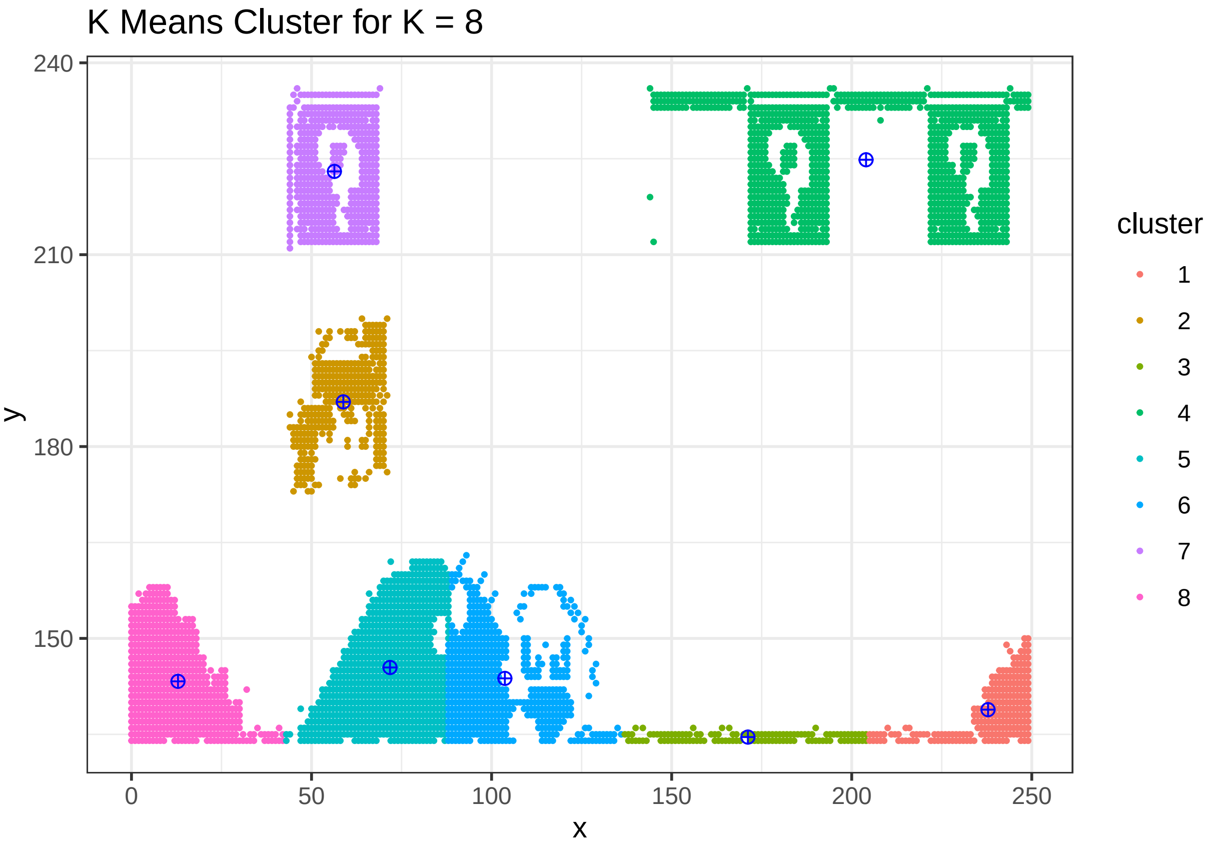

K Means Cluster for K = 3
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3

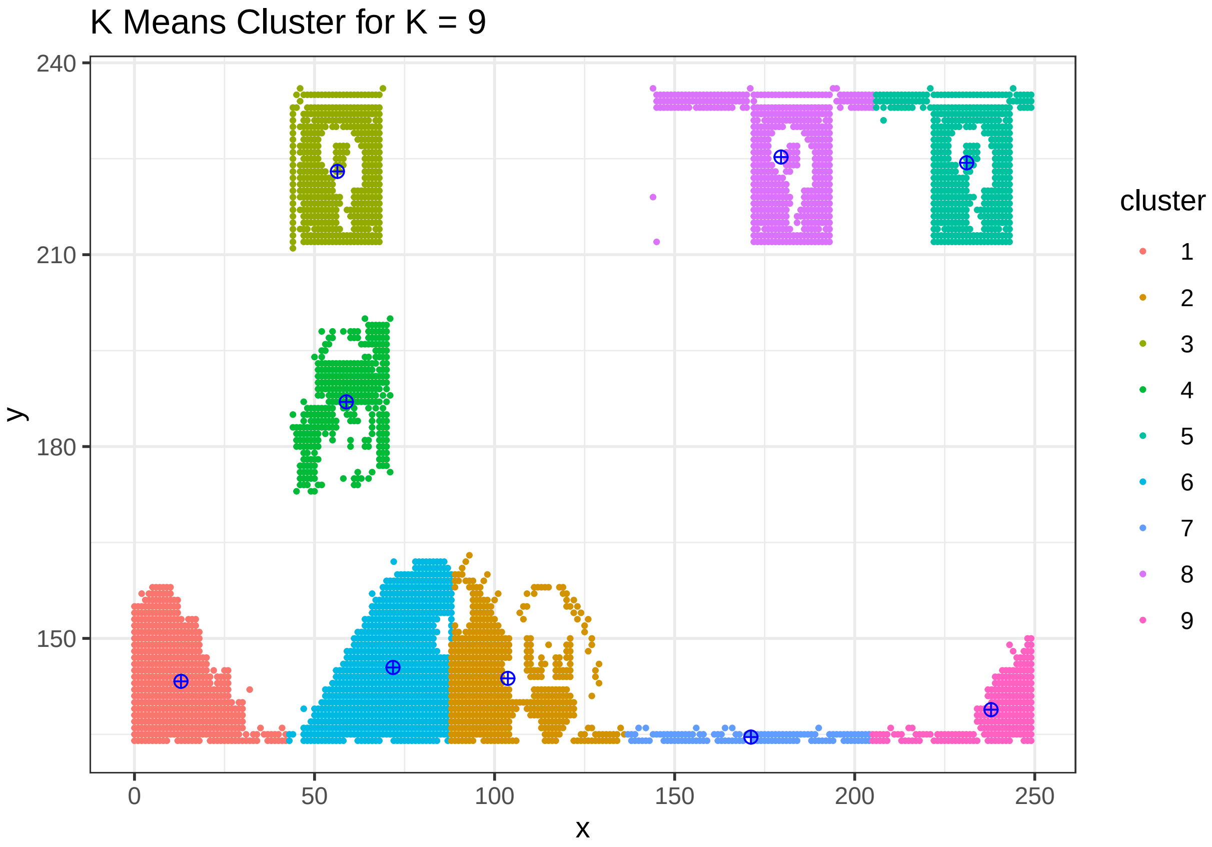

K Means Cluster for K = 4
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4

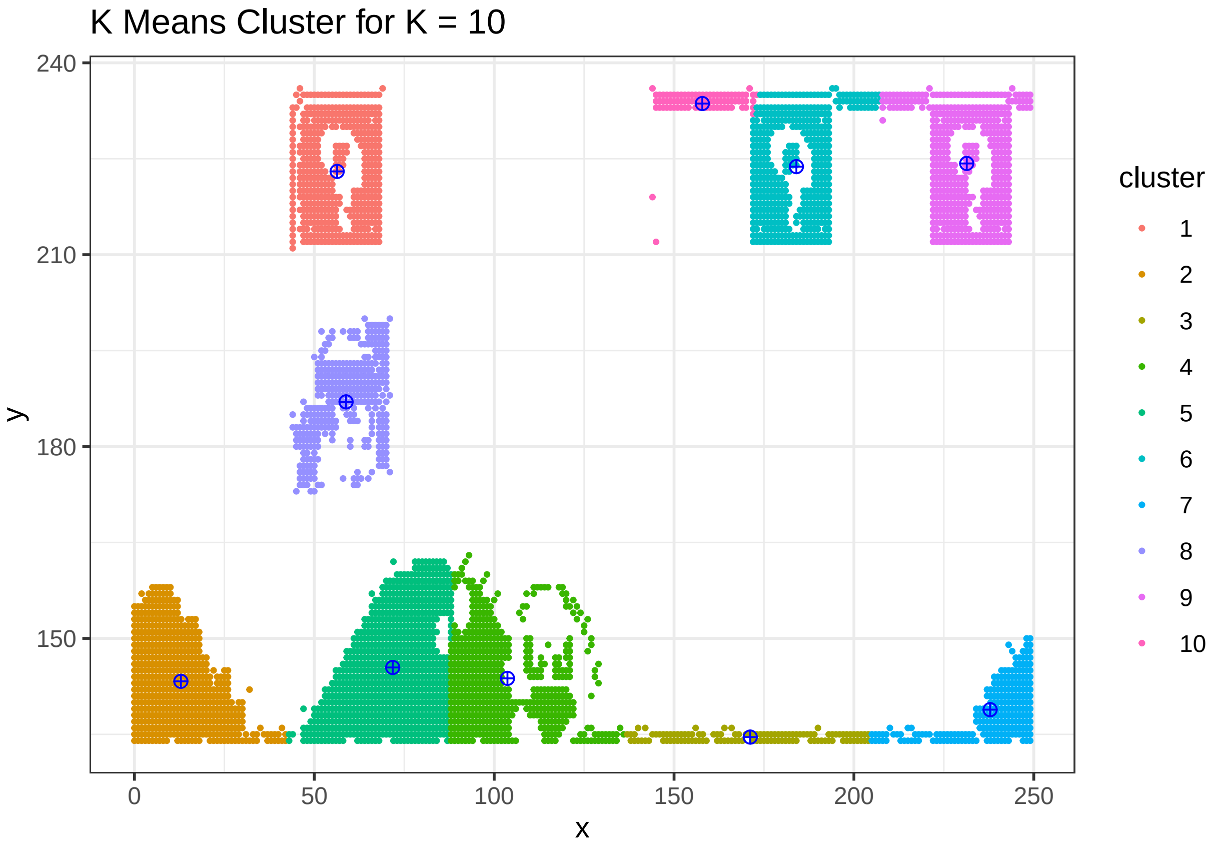

K Means Cluster for K = 5
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5

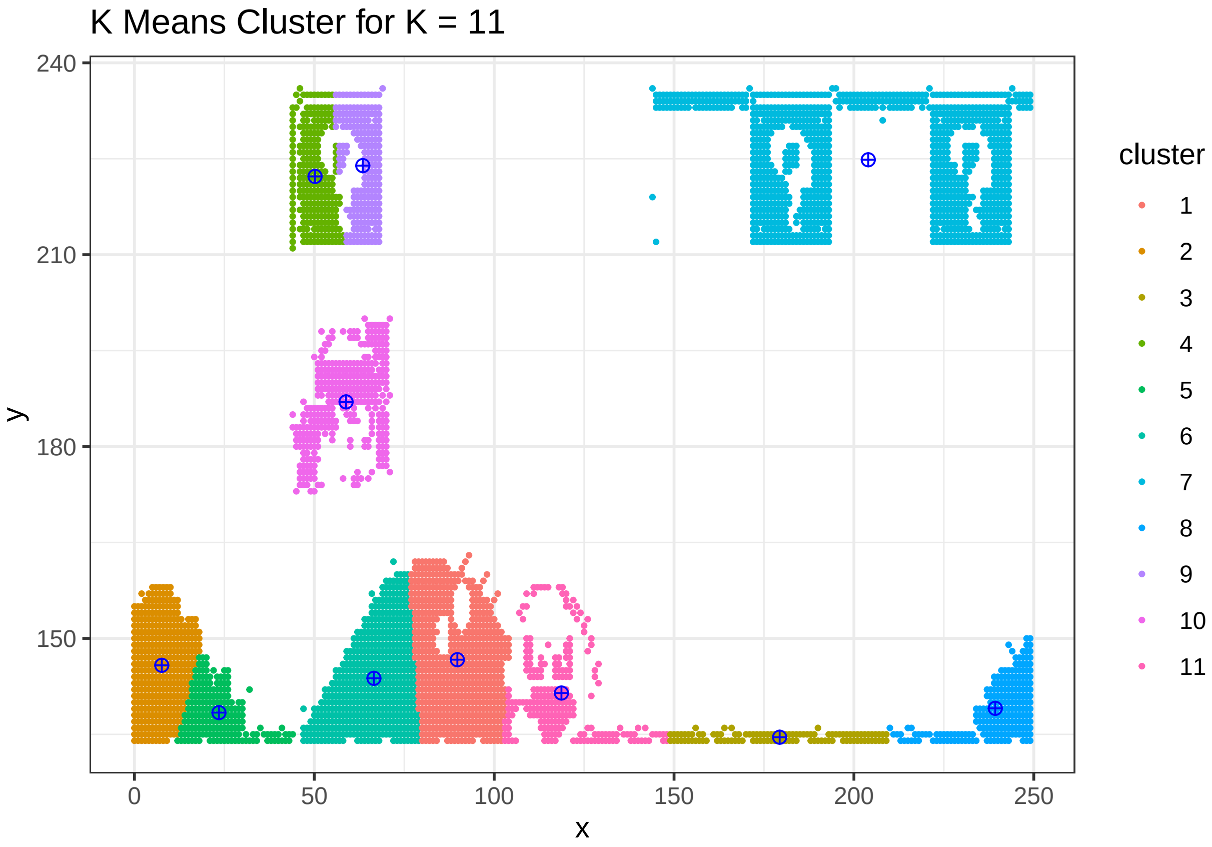

K Means Cluster for K = 6
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6

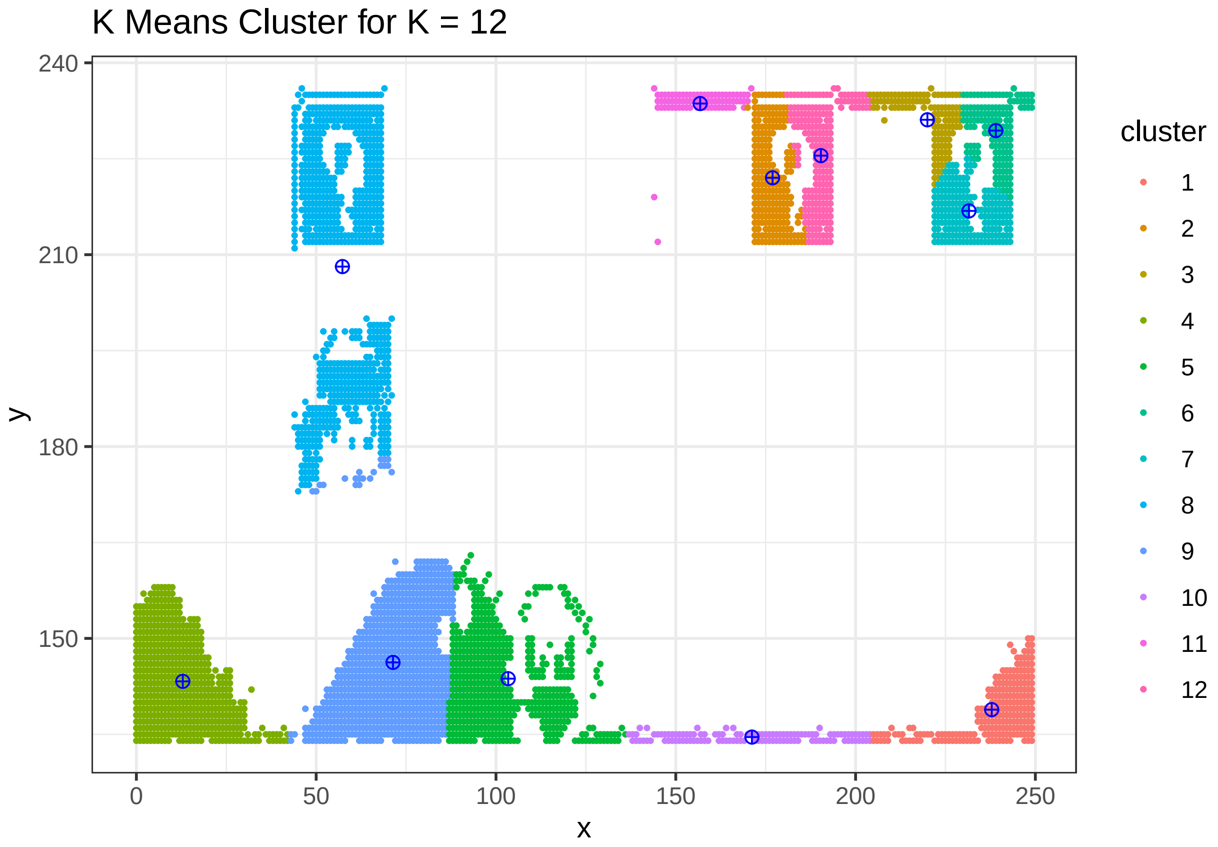

K Means Cluster for K = 7
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7


K Means Cluster for K = 8
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7
8


K Means Cluster for K = 9
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7
8
9


K Means Cluster for K = 10
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7
8
9
10


K Means Cluster for K = 11
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7
8
9
10
11



Question C.1:

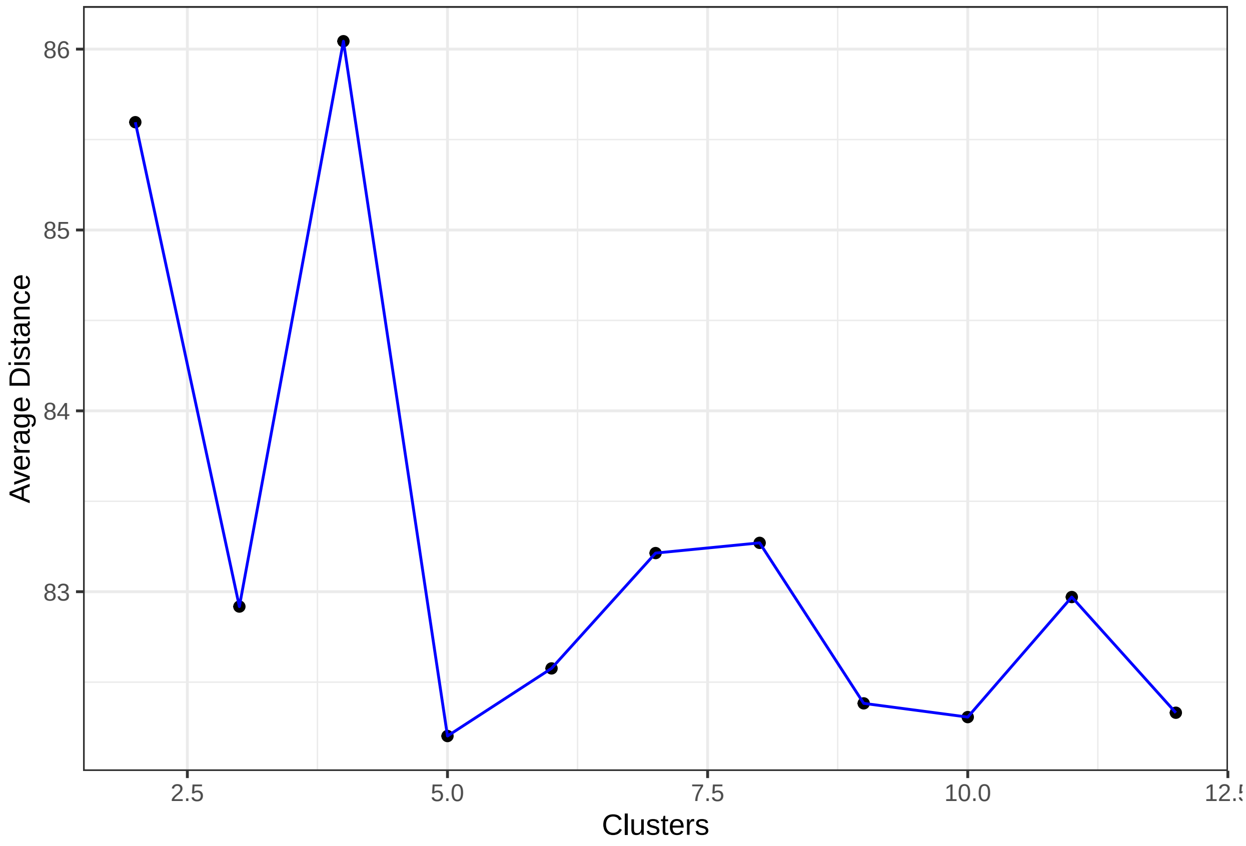
As k-means is an unsupervised algorithm, you cannot compute the accuracy as there are no correct values to compare the output to. Instead, you will use the average distance from the center of each cluster as a measure of how well the model fits the data. To calculate this metric, simply compute the distance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

Answer For C.1

```
avg_distdata <- data.frame(kmean_values, average_distance)
avg_distdata
```

```
##   kmean_values average_distance
## 1            2         85.59601
## 2            3         82.91766
## 3            4         86.04381
## 4            5         82.20191
## 5            6         82.57580
## 6            7         83.21349
## 7            8         83.26995
## 8             9         82.38256
## 9            10         82.30661
## 10           11         82.97045
## 11           12         82.33073
```

```
ggplot(data = avg_distdata, aes(x=kmean_values, y=average_distance)) + xlab("Clusters") + ylab("Average
```



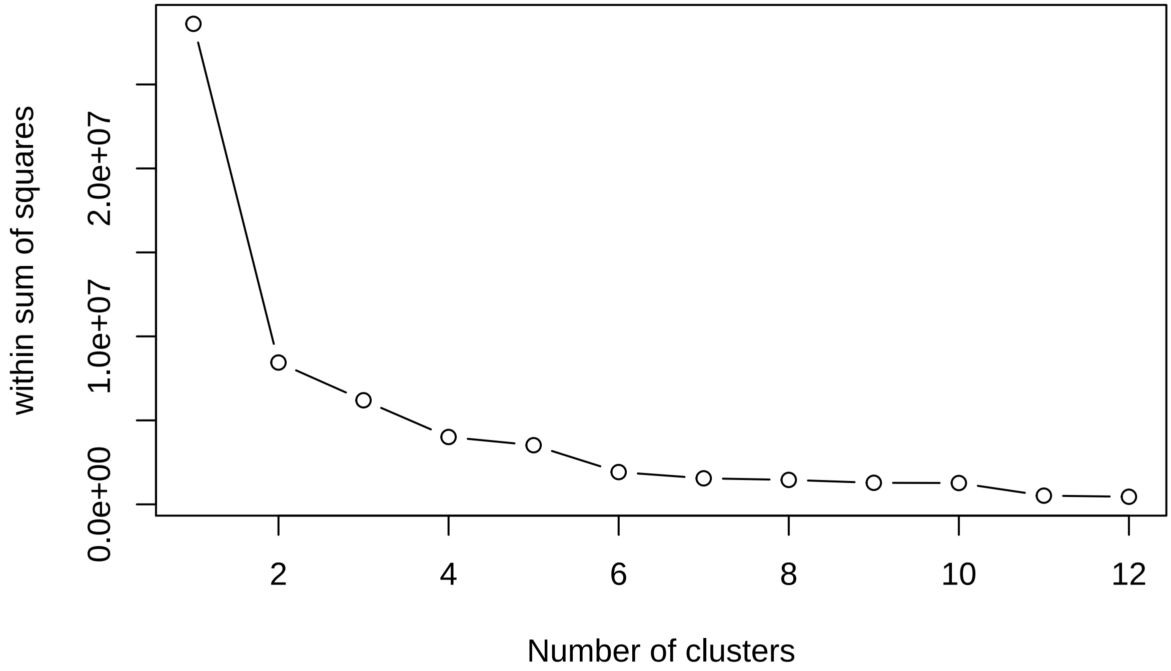
Question C.2:

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?

Answer For C.2

Looking the graph generated from average and within sum of squares, the elbow point seems to lie between 7 and 9.

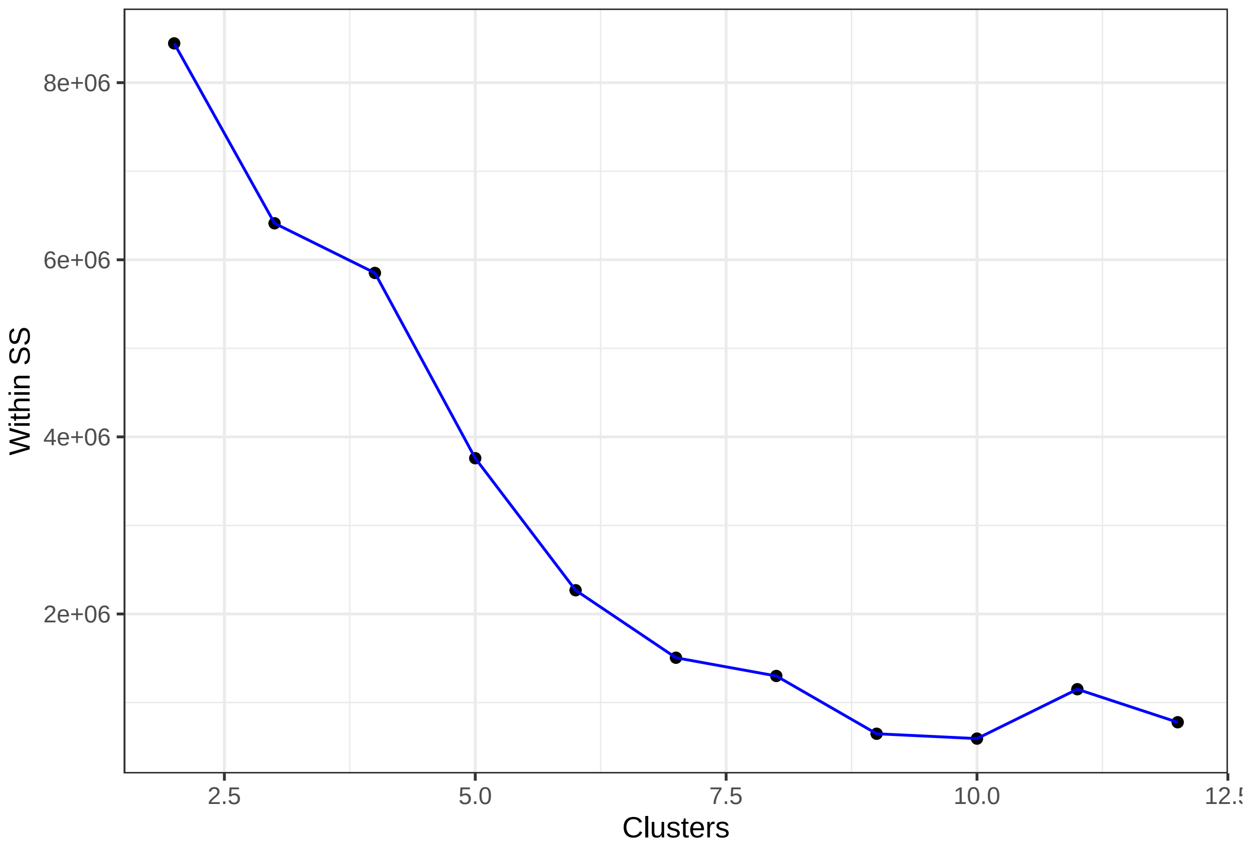
```
#Plot SS
plot(1:12, wss, type="b",xlab="Number of clusters",ylab="within sum of squares")
```



```r
totalWithinSS <- data.frame(kmean_values, total.withinss_values)
totalWithinSS
```

```
##    kmean_values total.withinss_values
## 1             2             8443681.1
## 2             3             6411644.9
## 3             4             5851331.1
## 4             5             3758773.6
## 5             6             2267917.2
## 6             7             1505729.6
## 7             8             1299901.3
## 8             9              647331.9
## 9            10              592654.1
## 10           11             1150200.5
## 11           12              777149.9
```

```r
ggplot(data = totalWithinSS, aes(x=kmean_values, y=total.withinss_values)) + xlab("Clusters") + ylab("W
```



8e+06
6e+06
4e+06
2e+06
Within SS
2.5
5.0
7.5
10.0
12.5
Clusters