Exercise 16: Clustering

Daniel Angel

February 14, 2021

Assignment

Labeled data is not always available. For these types of datasets, you can use unsupervised algorithms to extract structure. The k-means clustering algorithm and the k nearest neighbor algorithm both use the Euclidean distance between points to group data points. The difference is the k-means clustering algorithm does not use labeled data.

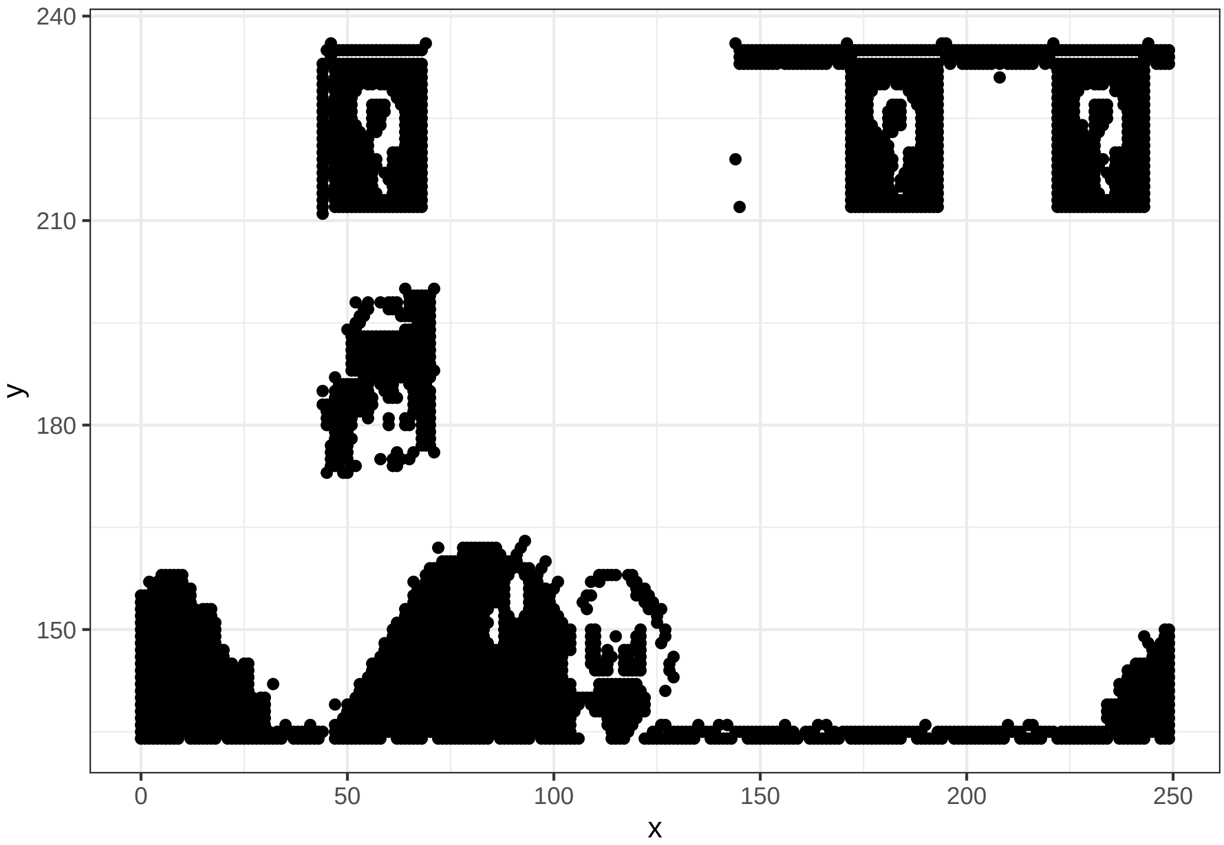
In this problem, you will use the k-means clustering algorithm to look for patterns in an unlabeled dataset. The dataset for this problem is found at data/clustering-data.csv.

```
setwd("C:/Users/Danny/Documents")
clustering_df <- read.csv("data/clustering-data.csv")
```

Question a

Plot the dataset using a scatter plot.

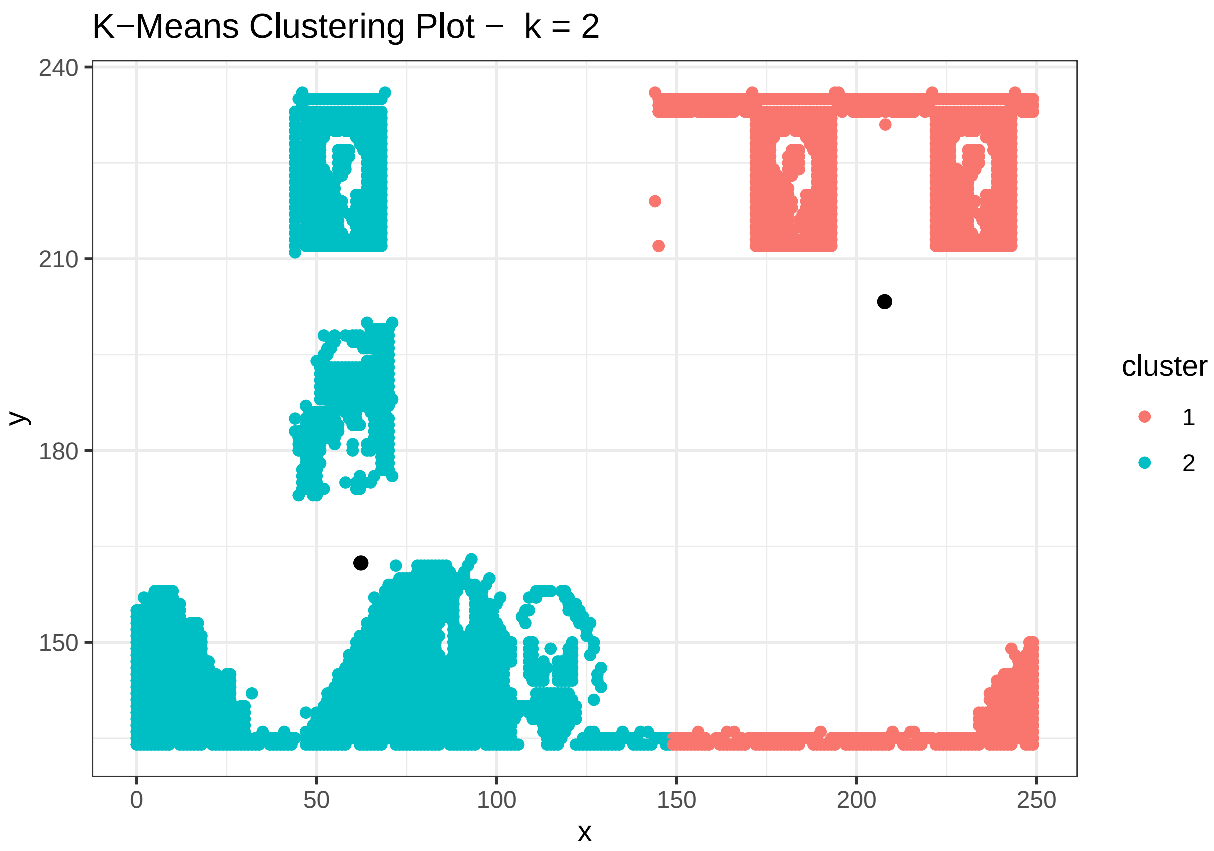
```
ggplot(clustering_df, aes(x=x, y=y)) + geom_point() + theme_bw()
```

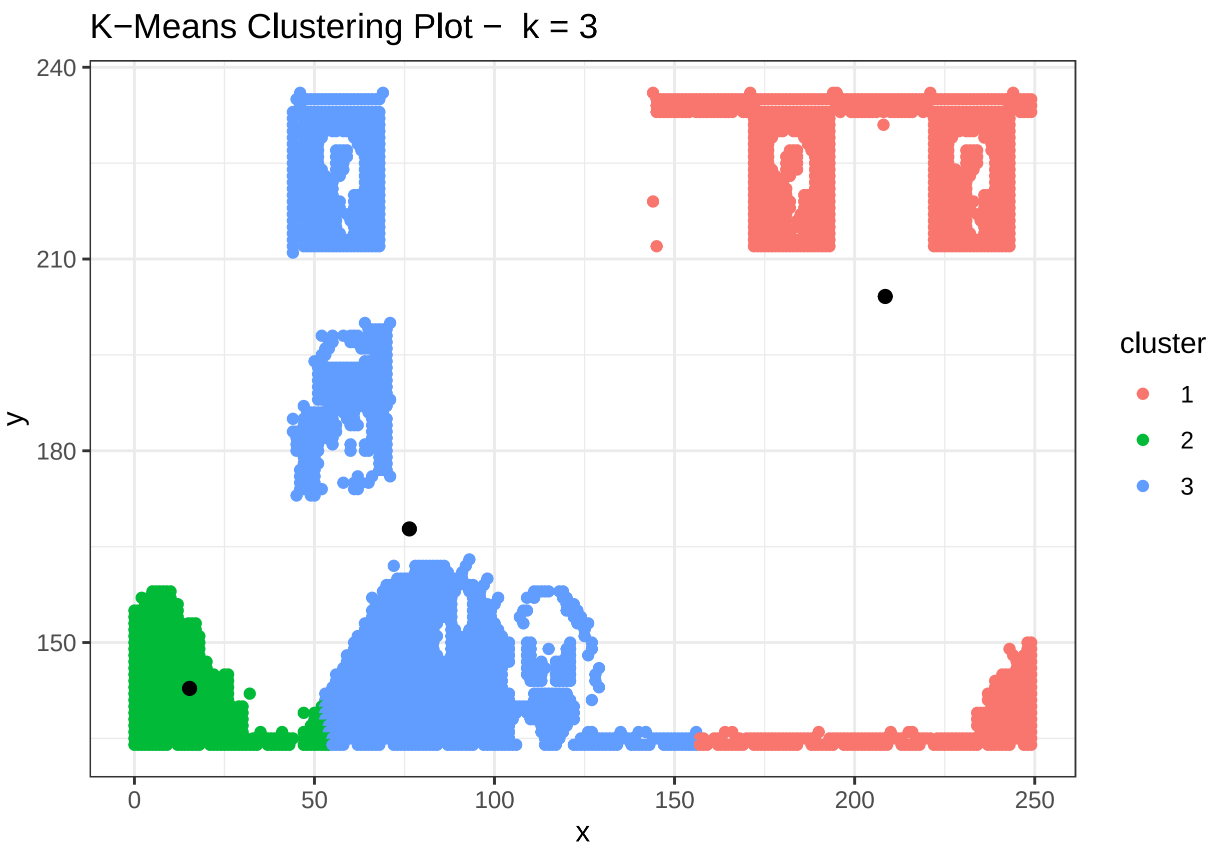


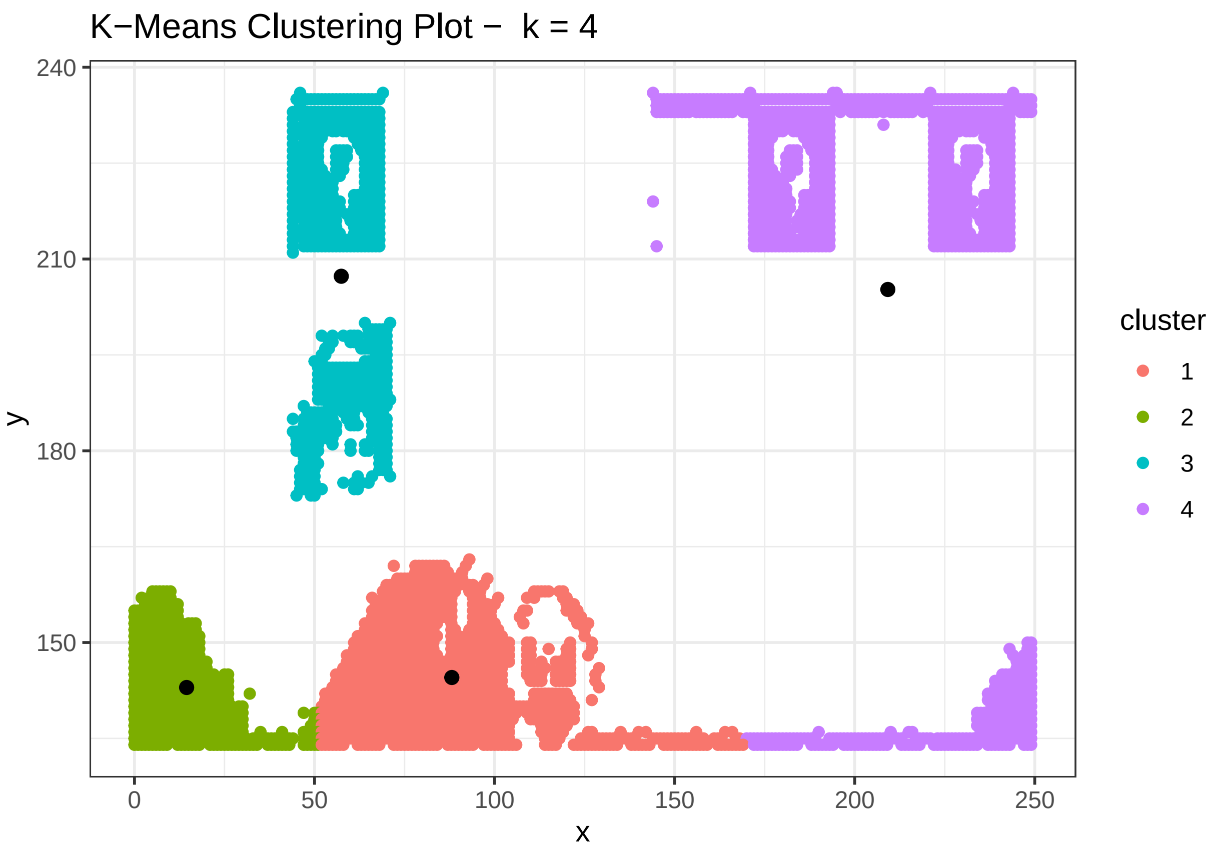
Question b

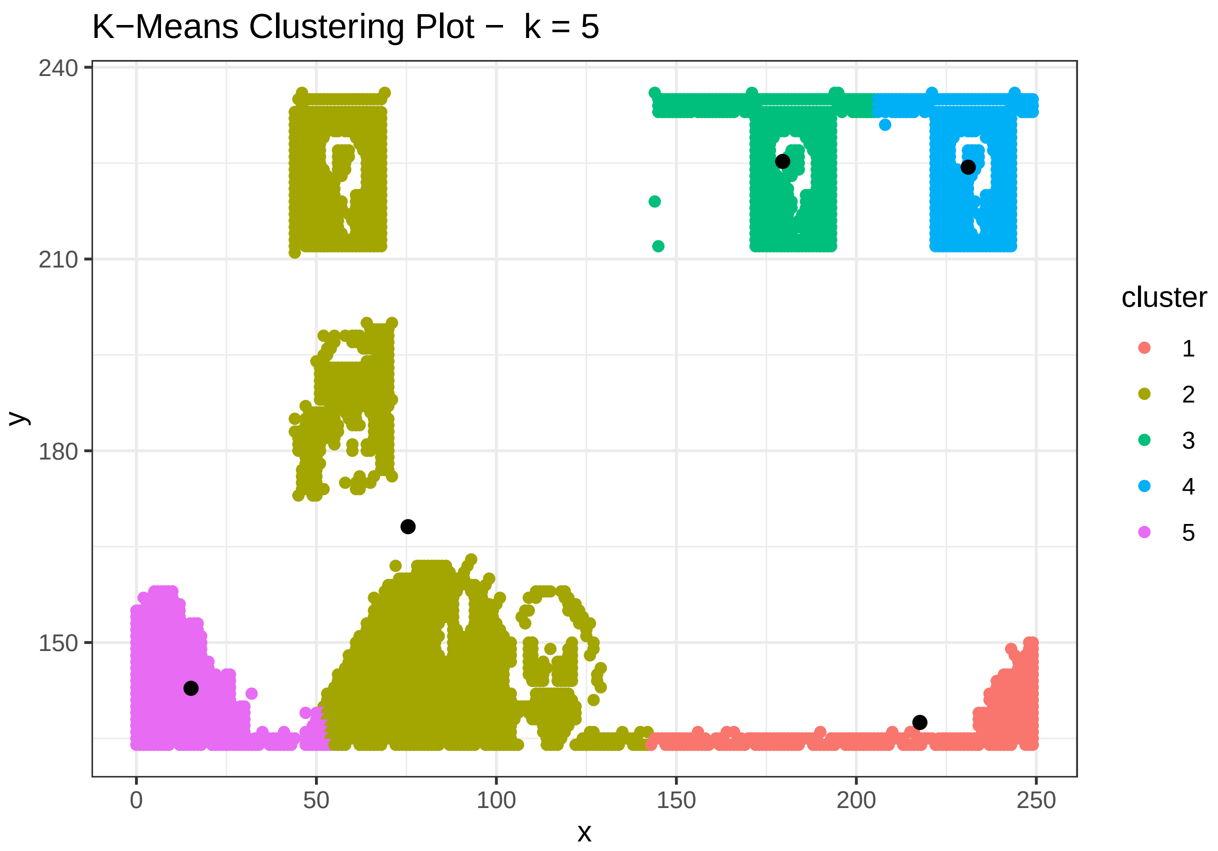
Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

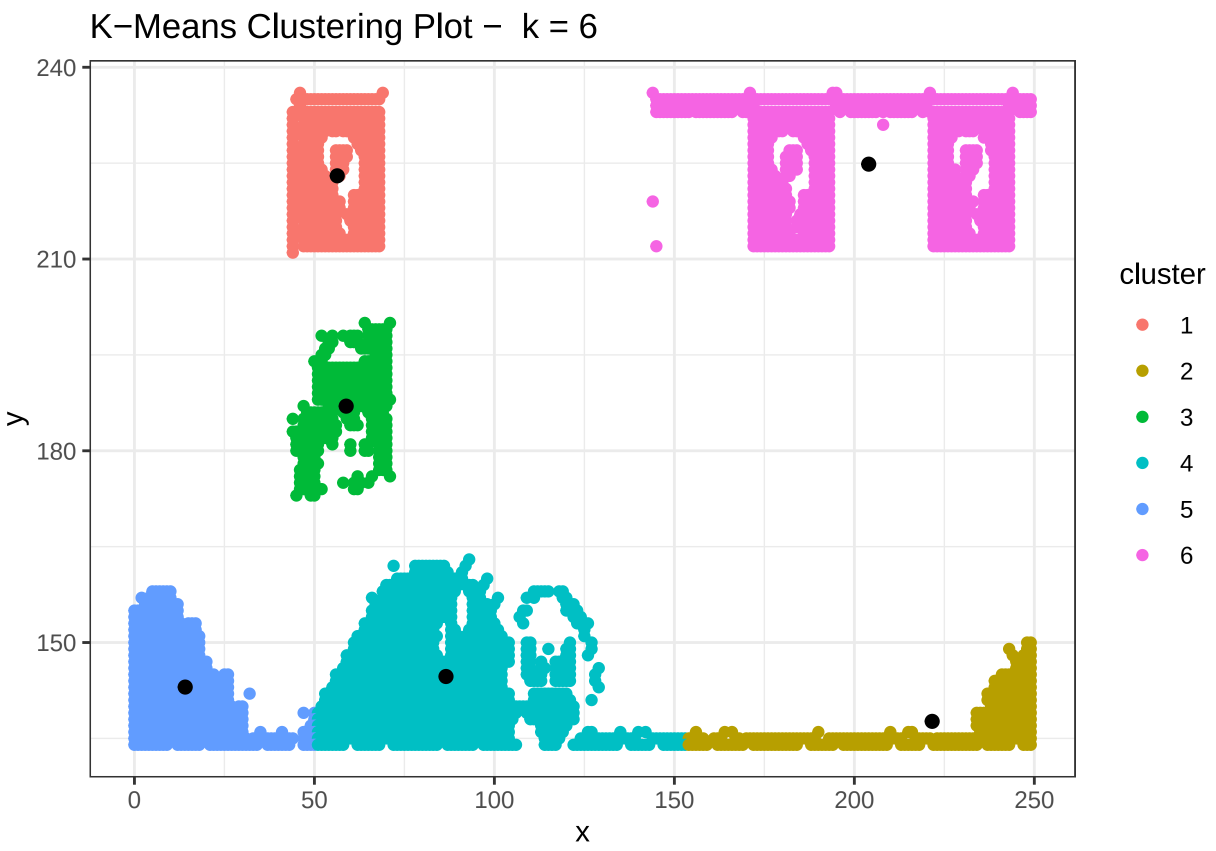
```
set.seed(20)
k_values <- c(2:12)
total_wss <- NULL
average_distance <- NULL
for (i in 1:length(k_values))
{
  cluster_df <- clustering_df
  clusters <- kmeans(cluster_df, k_values[i])
  cluster_df$cluster <- as.factor(clusters$cluster)
  x = ggplot(data = cluster_df, aes(x=x, y=y, color = cluster)) + geom_point() + theme_bw() + geom_poin
  print(x)

  # following values are calculated for question c
  x_dist <- clusters$centers[cluster_df$cluster] - cluster_df$x
  y_dist <- clusters$centers[cluster_df$cluster] - cluster_df$y
  tot_dist <- sqrt((x_dist ** 2) + (y_dist ** 2))
  average_distance <- c(average_distance, mean(tot_dist))
  total_wss <- c(total_wss, clusters$tot.withinss)
}
```

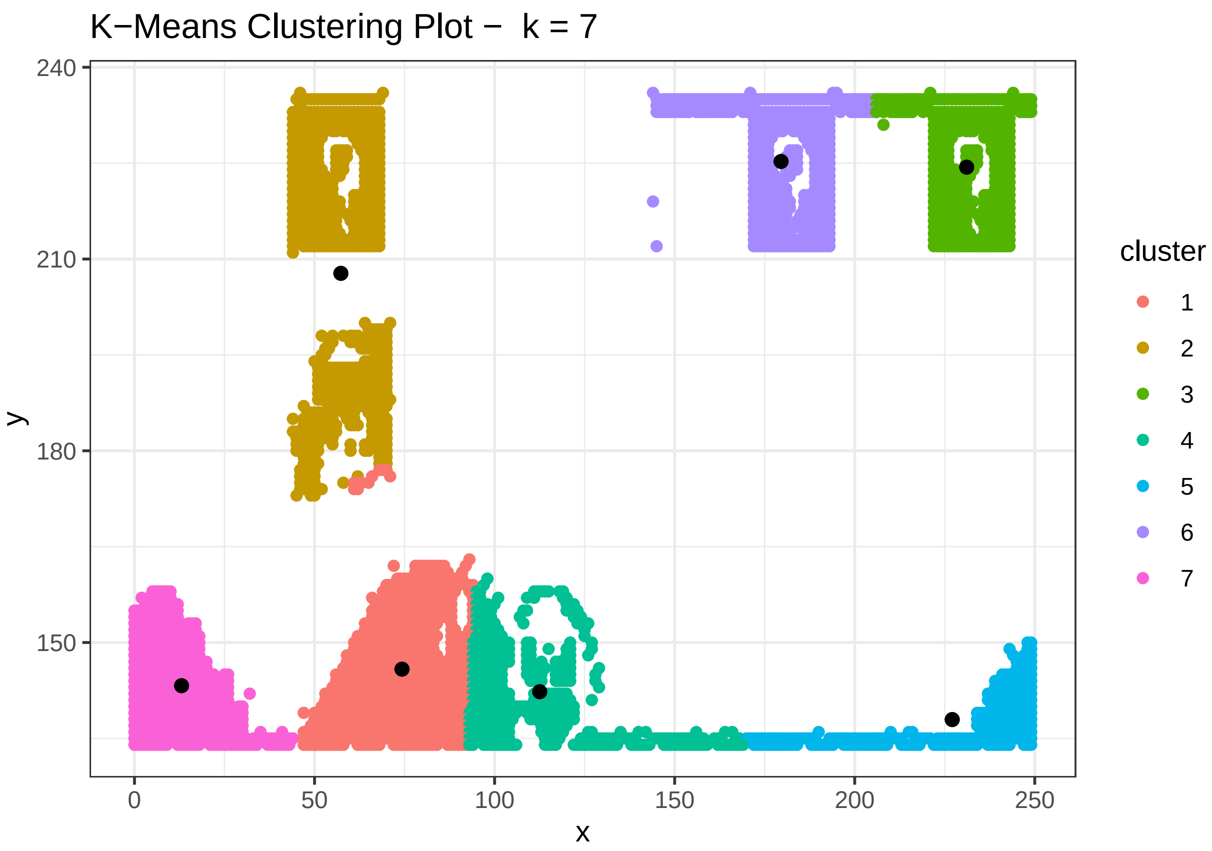

K−Means Clustering Plot − k = 2
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2

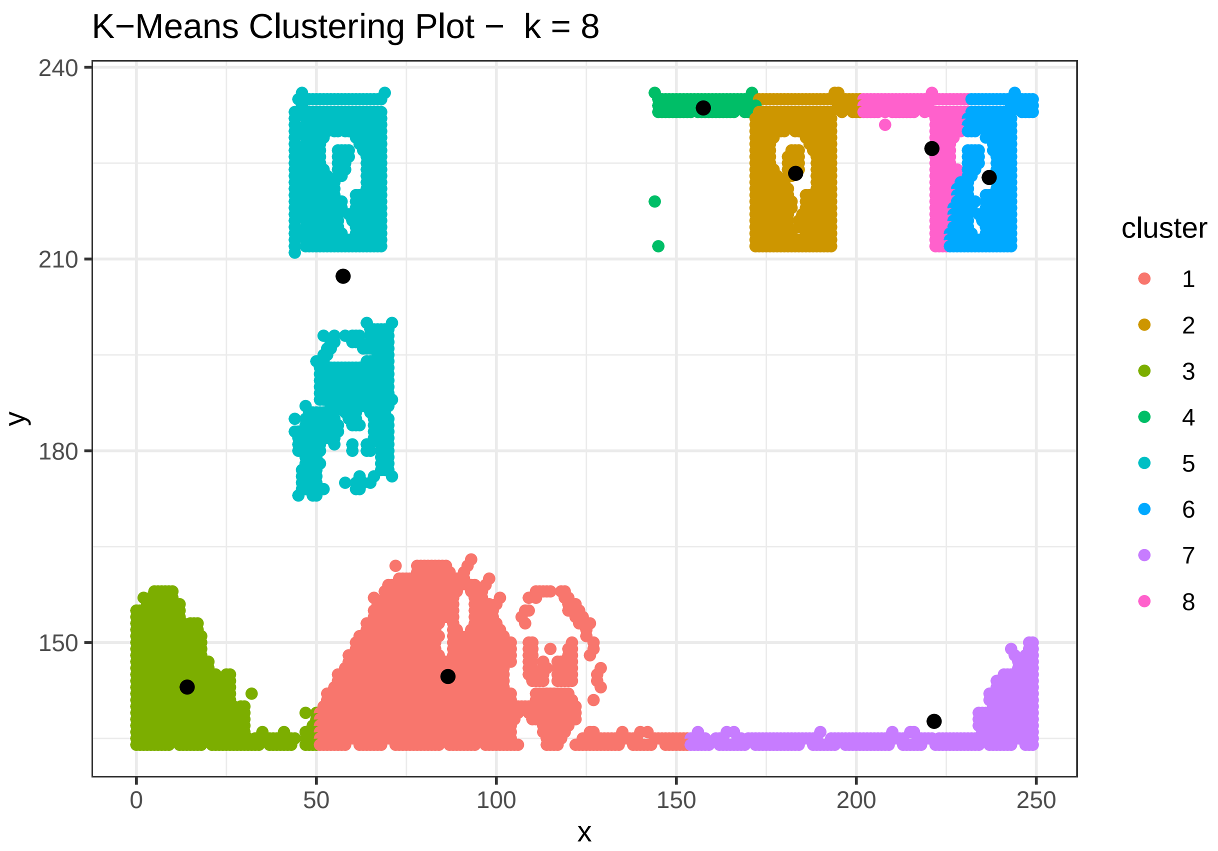

K−Means Clustering Plot − k = 3
y
240
210
180
150
x
0
50
100
150
200
250
cluster
1
2
3

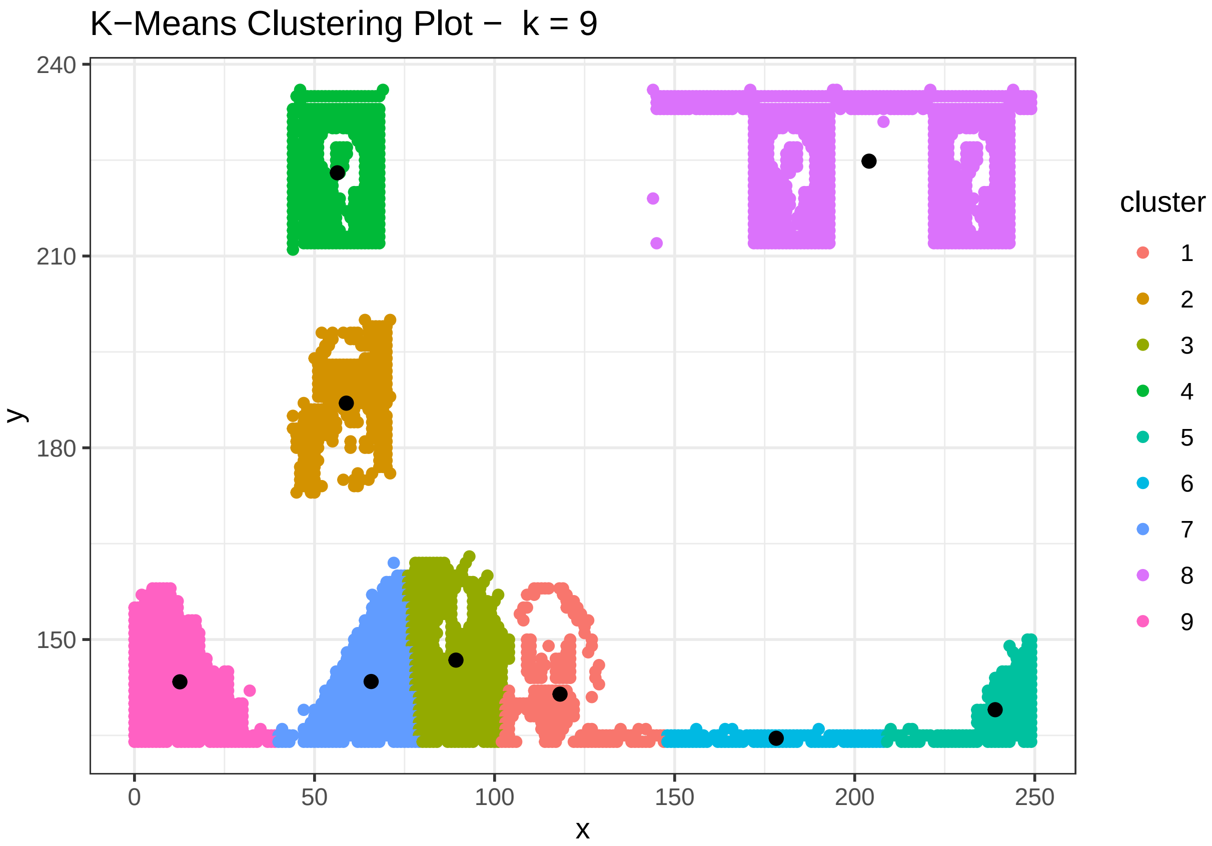

K−Means Clustering Plot − k = 4
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4

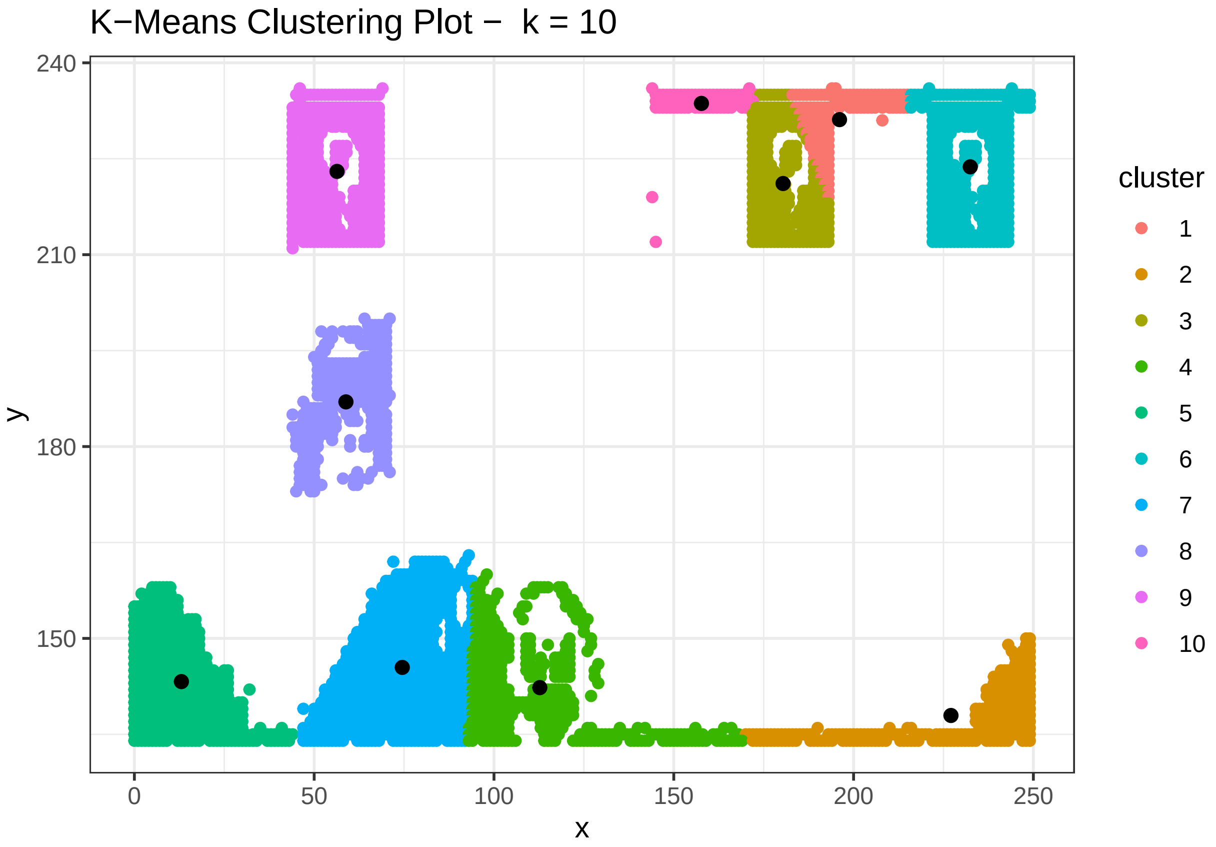

K−Means Clustering Plot − k = 5
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5

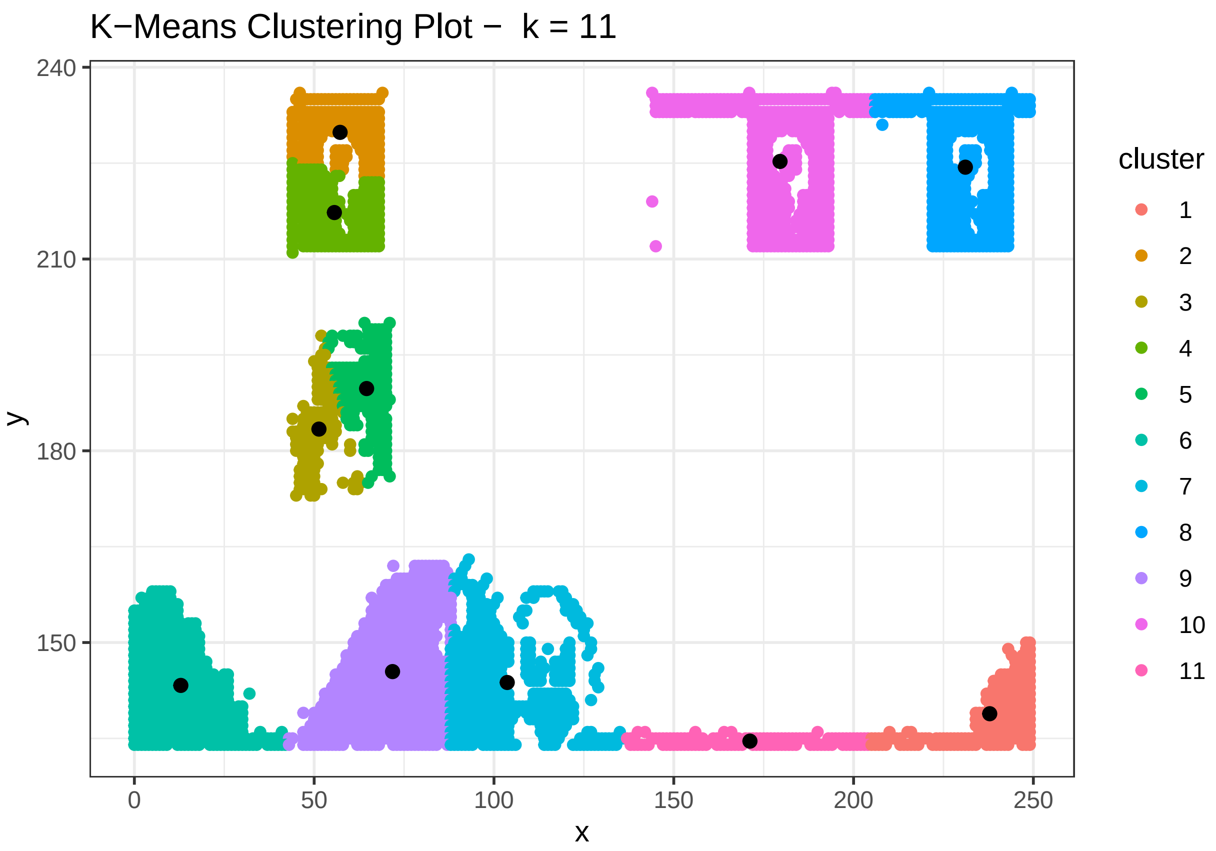

K−Means Clustering Plot − k = 6
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6

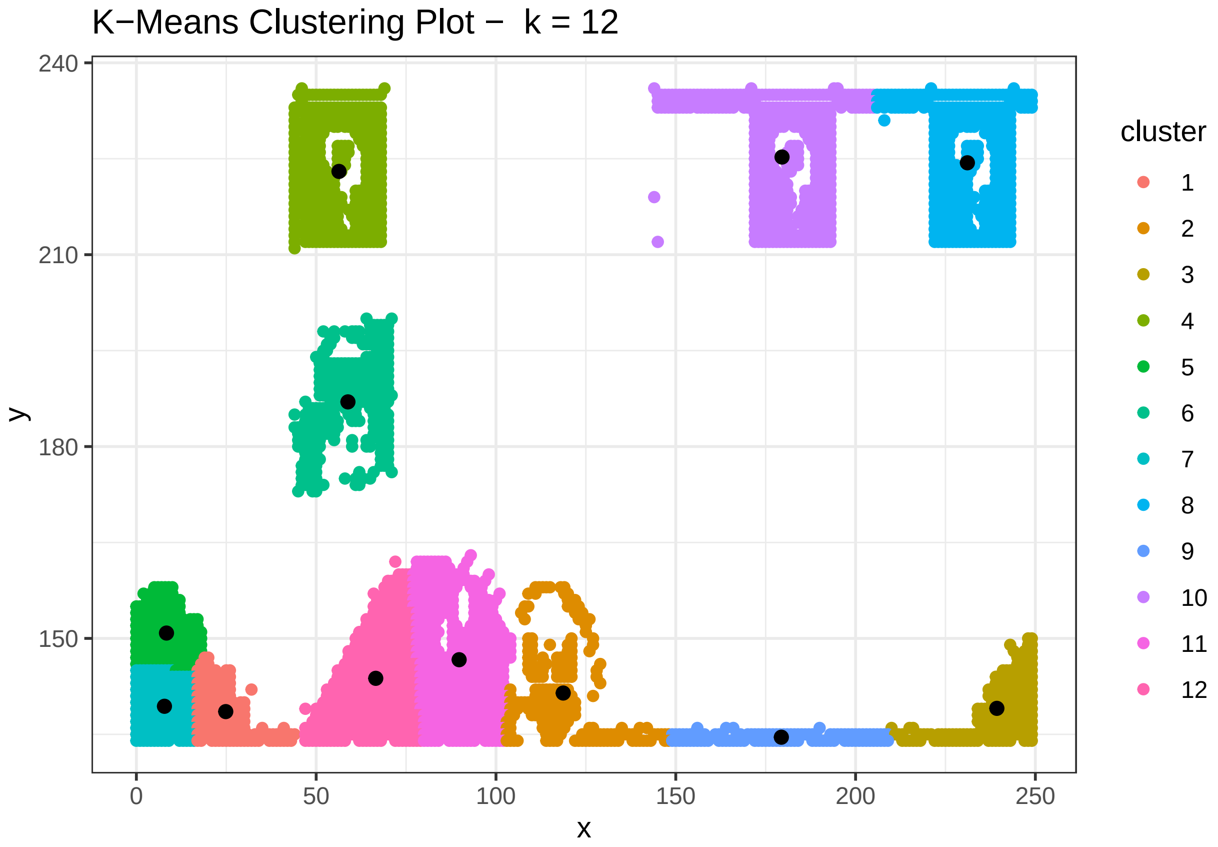

K−Means Clustering Plot − k = 7
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7


K−Means Clustering Plot − 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 Clustering Plot − 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 Clustering Plot − 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 Clustering Plot − 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

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.

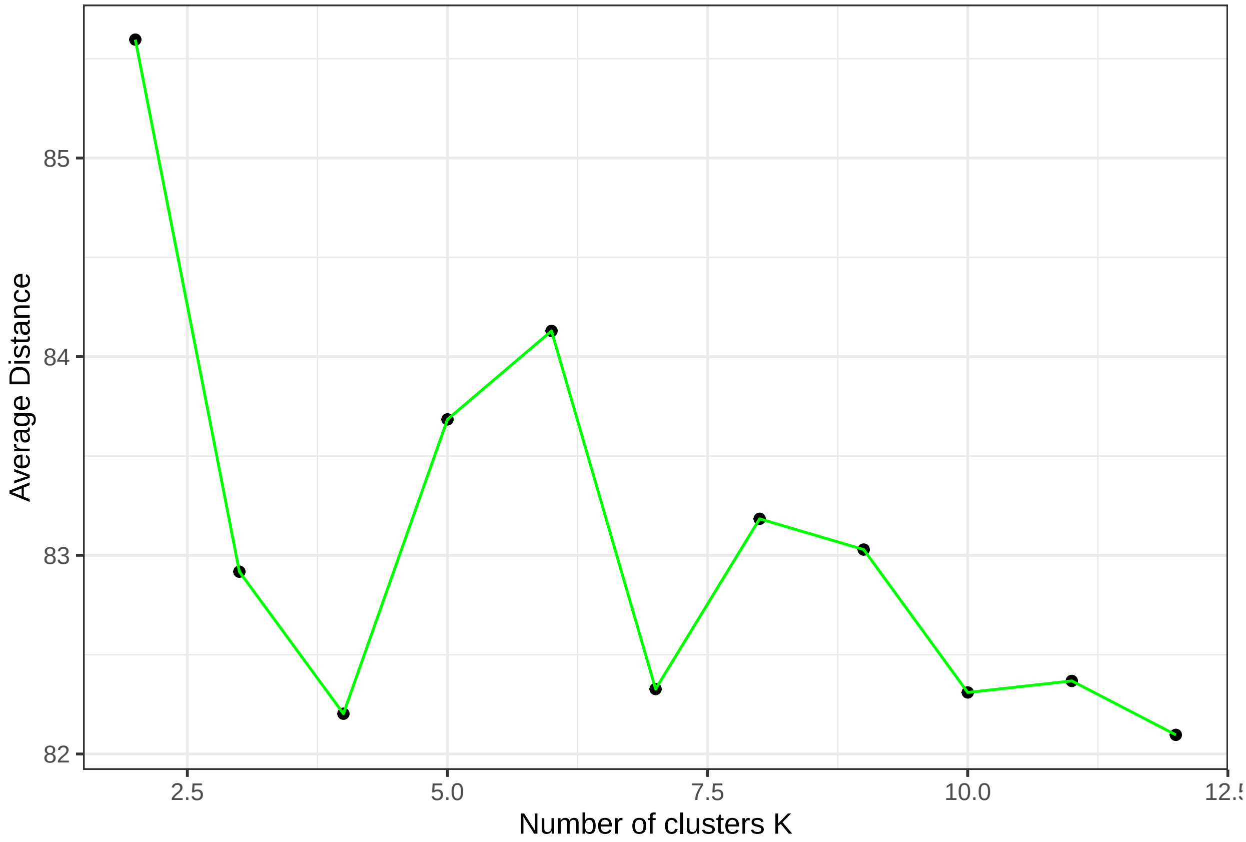
Calculate this average distance from the center of each cluster for each value of k and plot it as a line chart where k is the x-axis and the average distance is the y-axis.

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?

```
dist_elbow_df <- data.frame(k_values, average_distance)
dist_elbow_df
```

```
##    k_values average_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
```

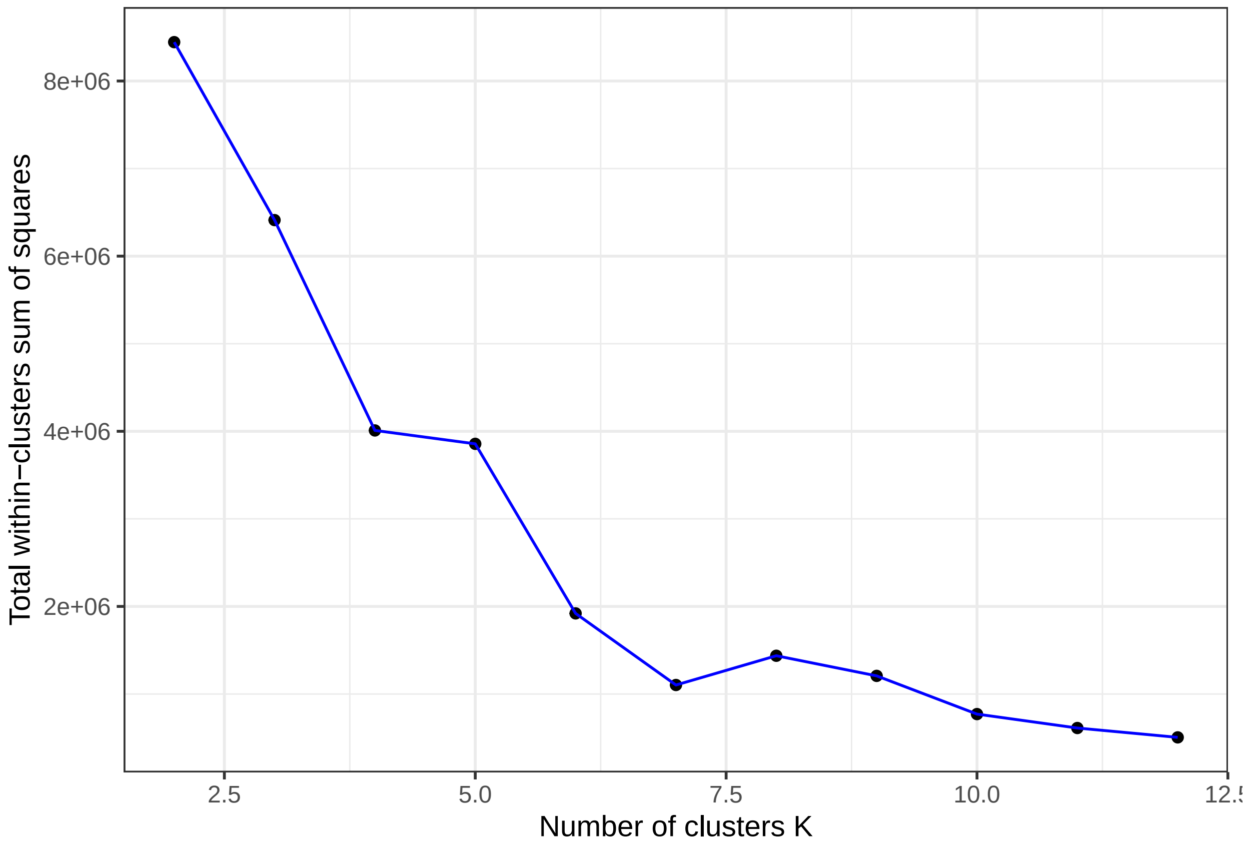
```
ggplot(data = dist_elbow_df, aes(x=k_values, y=average_distance)) + xlab("Number of clusters K") + ylab
```



```
wss_df <- data.frame(k_values, total_wss)
wss_df
```

```
##    k_values total_wss
## 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 = wss_df, aes(x=k_values, y=total_wss)) + xlab("Number of clusters K") + ylab("Total within-
```



I think something is wrong with my distance calculation but I am not positive. I think I might be calculating the average x and average y in the clusters rather than the difference from the centers but I'm not sure how to fix it other than by removing the iteration loop over the range of k=2 to k=12 and using that to nail down the behavior of the values created by the kmeans() function.

If I look at the graph displaying Within Sums of Squares there is an obvious "elbow" around 6 or 7. If I look at the K-Means Cluster Plots I believe K=7 is the superior number of clusters.