Exercise 16: Clustering

Michael Hotaling

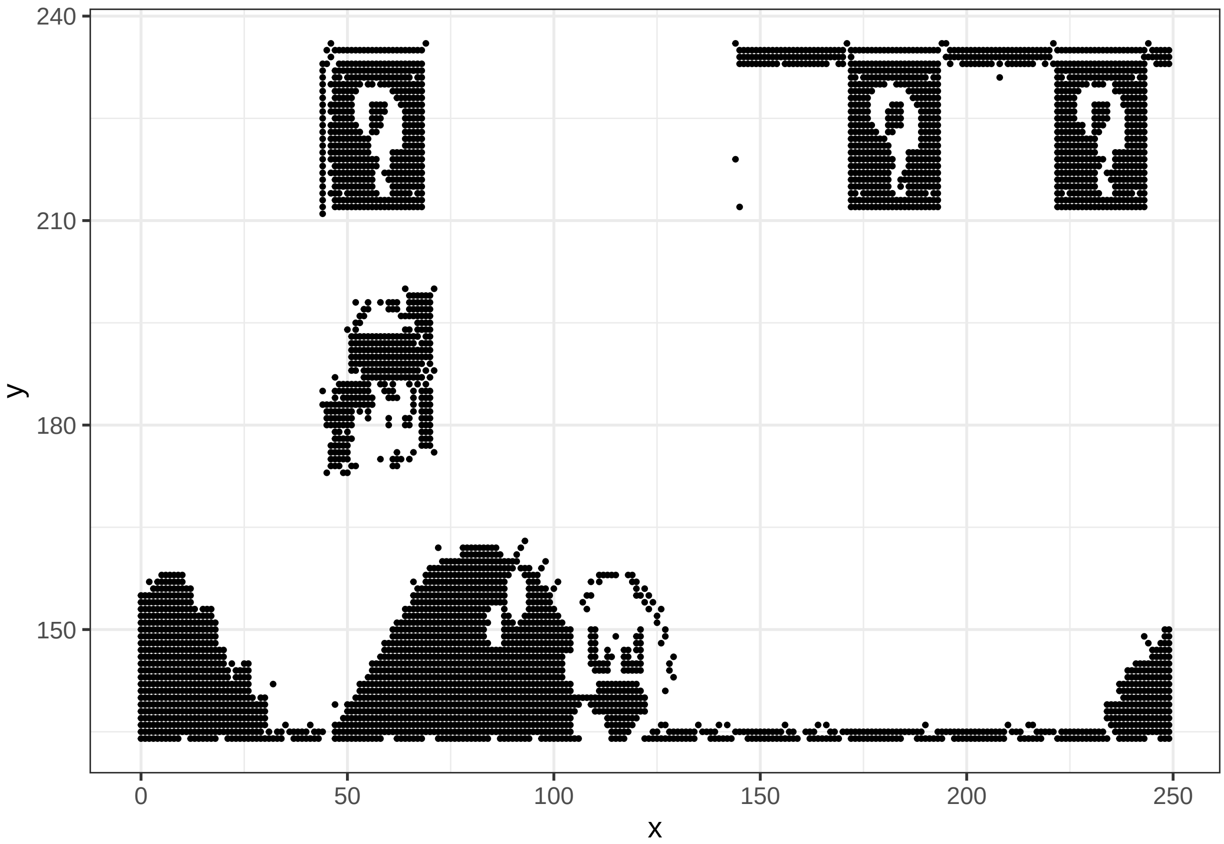
10/28/2020

Exercise 16: Clustering

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.

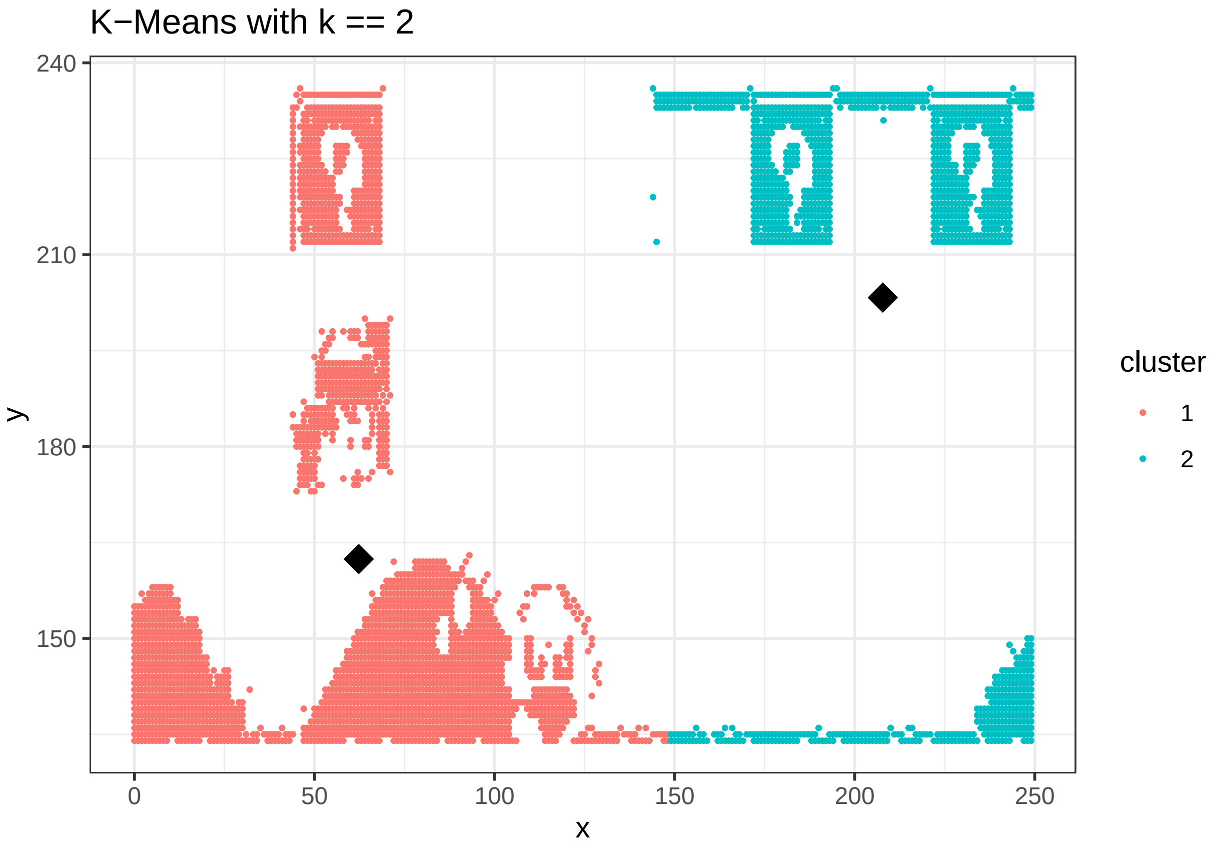
a. Plot the dataset using a scatter plot.

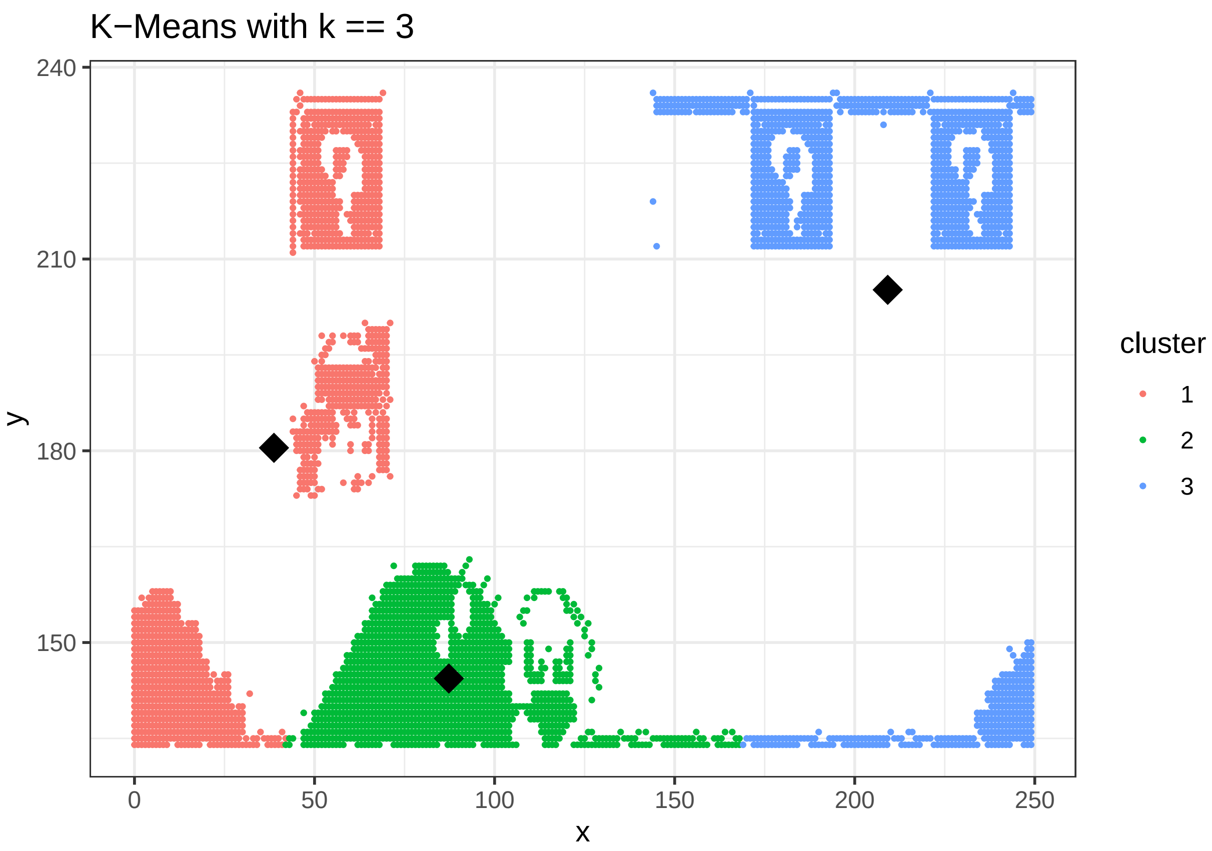
```
library(ggplot2)
library(knitr)
library(pander)
library(factoextra)

df <- read.csv("clustering-data.csv")

ggplot(data = df, aes(x = x, y = y)) +
geom_point(size = 0.5) +
theme_bw()
```

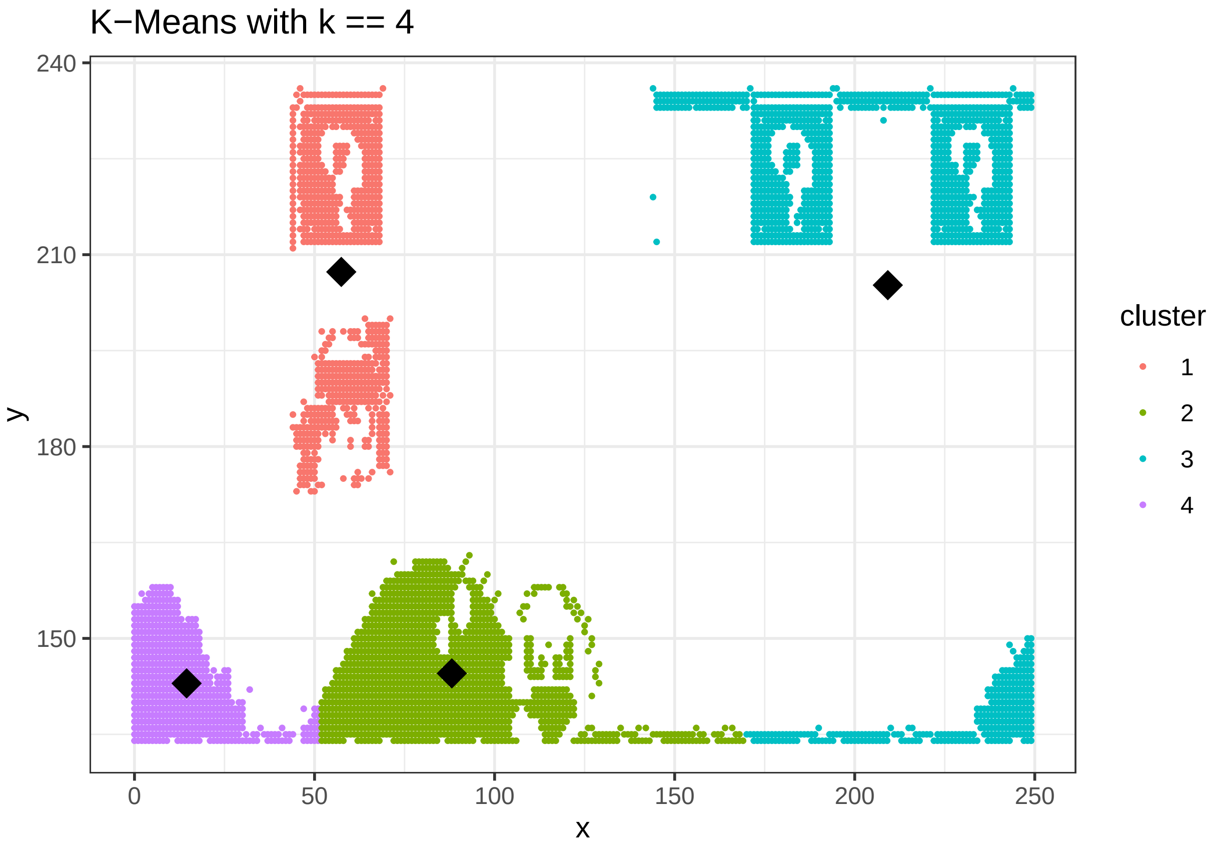


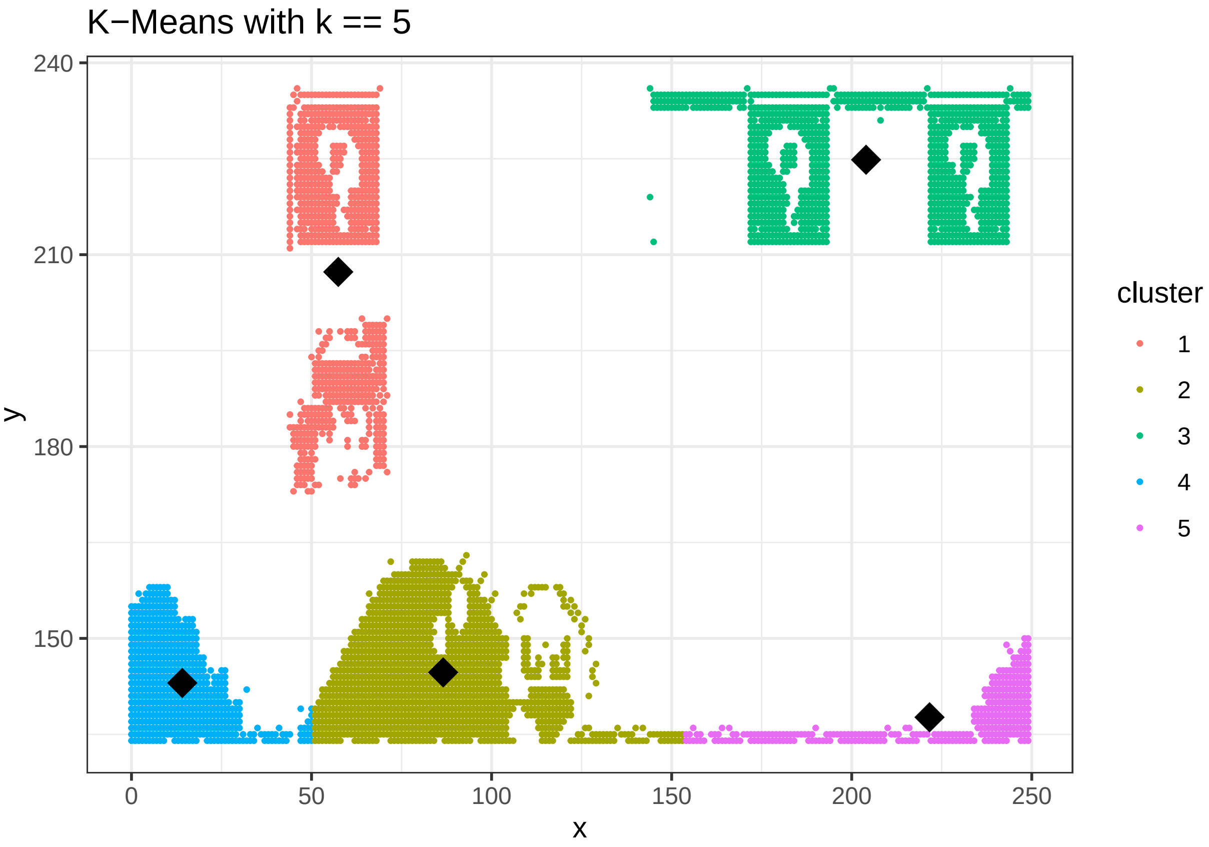
b. Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

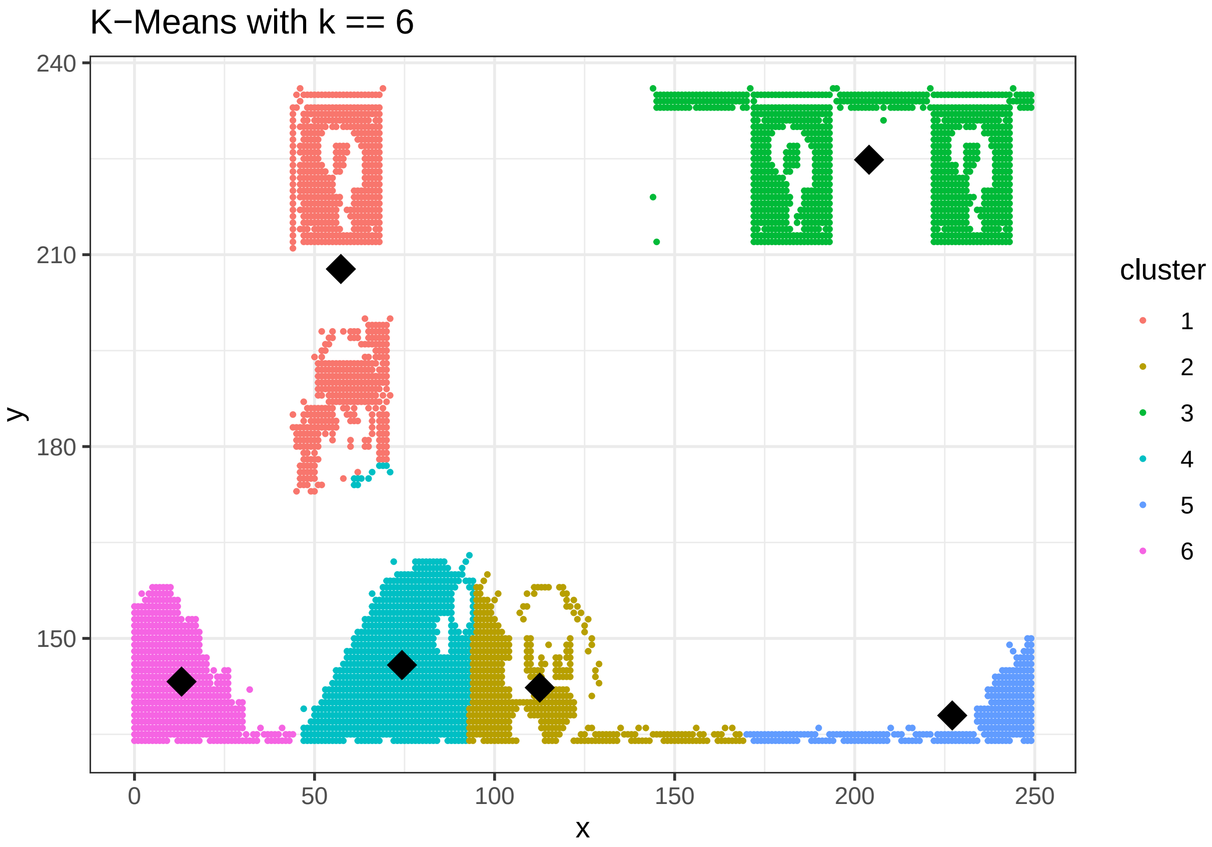
```
for(i in 2:12){
  set.seed(1)
  df <- read.csv("clustering-data.csv")
  df.cluster <- kmeans(df, i)

  df$cluster <- as.factor(df.cluster$cluster)
  p <- ggplot(data = df,
              aes(x = x,
                  y = y,
                  color = cluster)) +
  geom_point(size = 0.5) +
  geom_point(data = as.data.frame(df.cluster$centers),
             color = "black",
             shape = 18,
             size = 5) +
  ggtitle(paste("K-Means with k == ", i, sep ="")) +
  theme_bw()
  print(p)
}
```

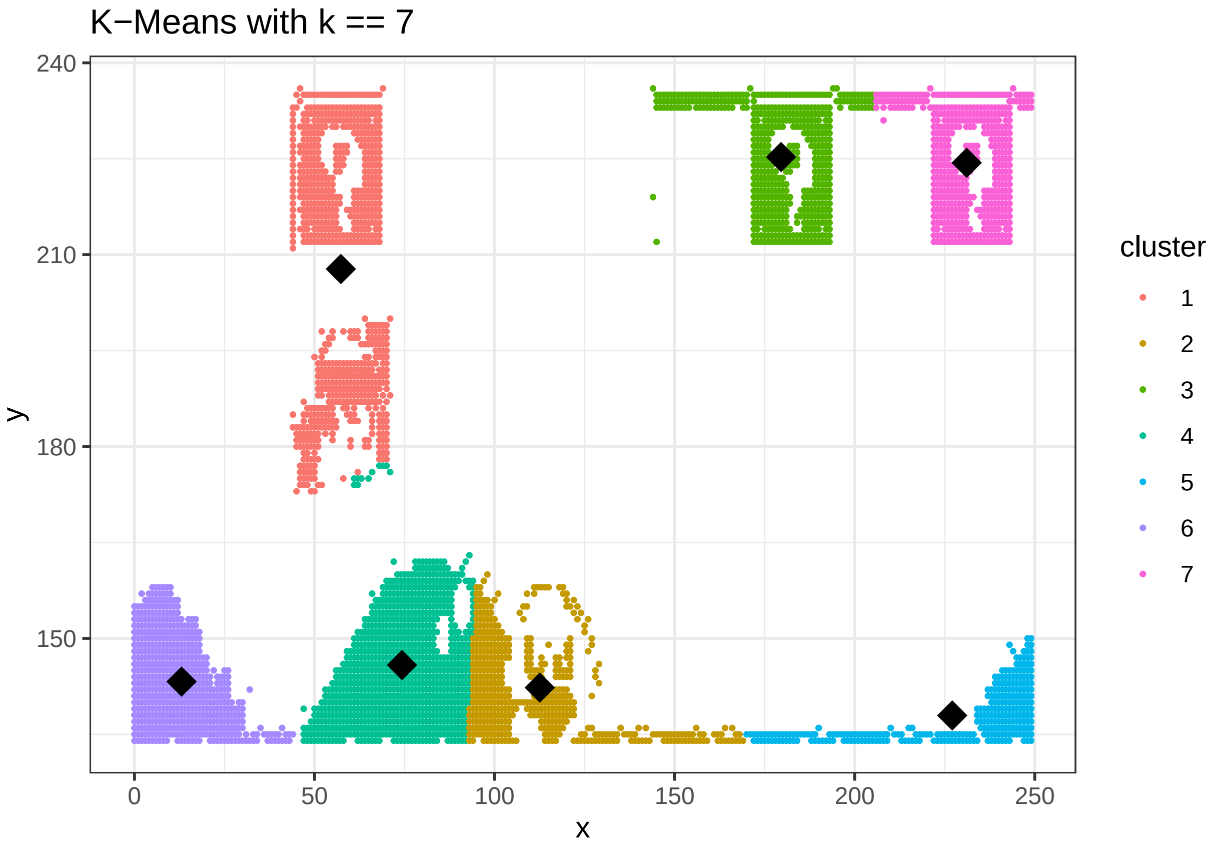

K−Means with k == 2
240
210
180
150
y
0
50
100
150
200
250
x
cluster
● 1
● 2

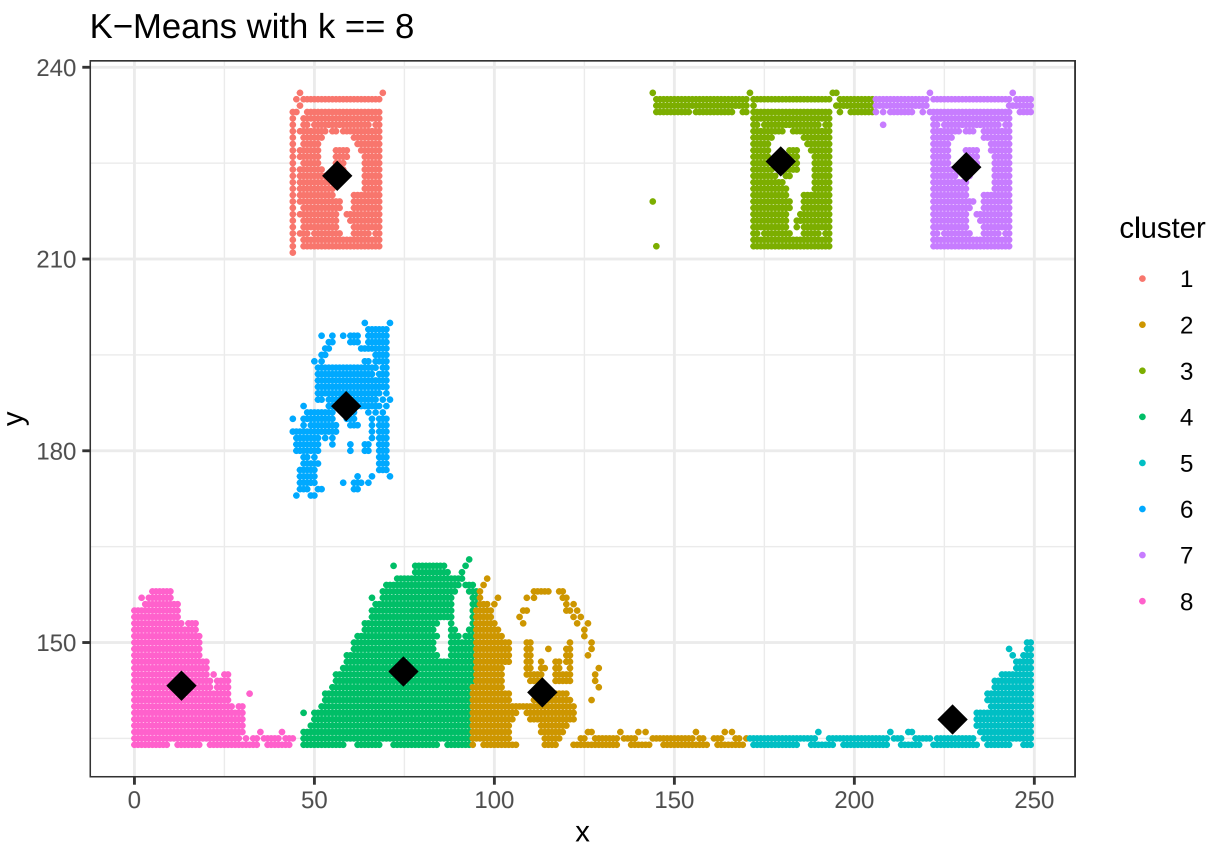

K−Means with k == 3
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3

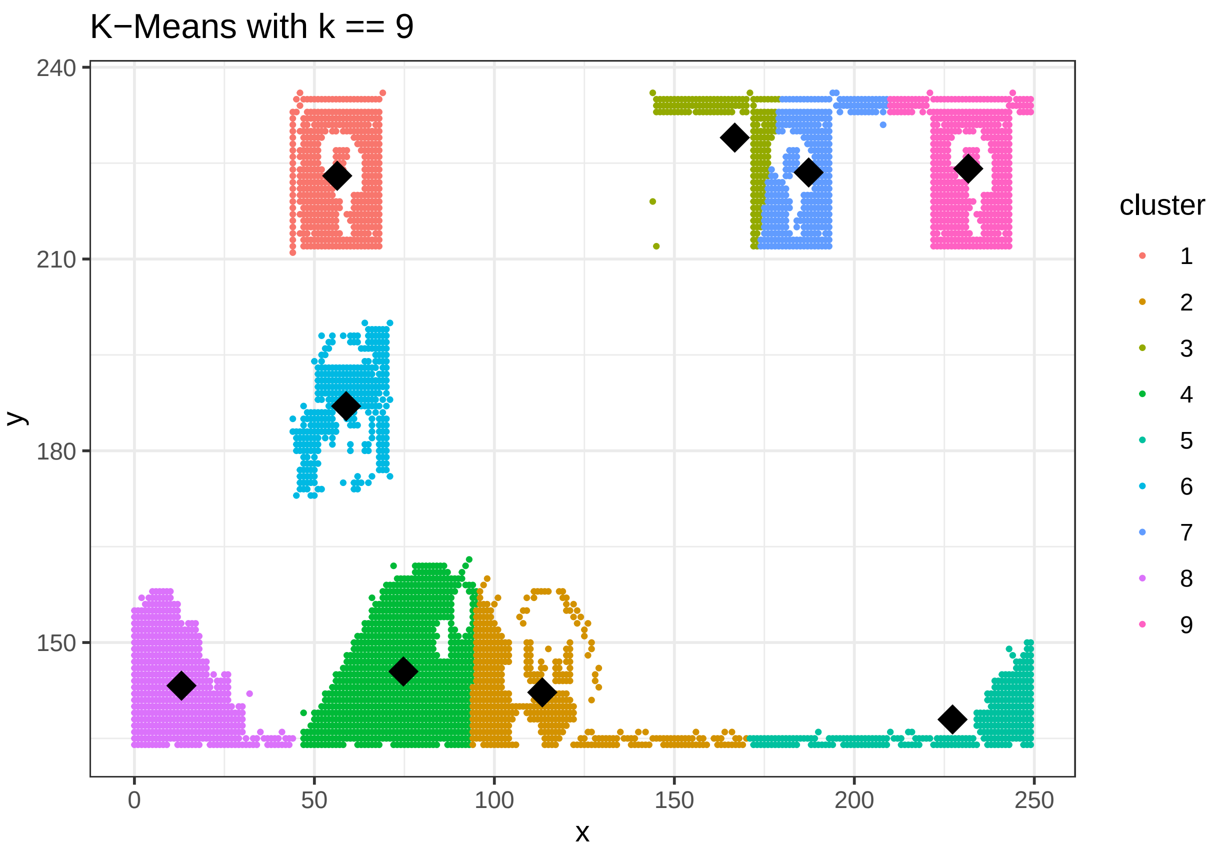

K−Means with k == 4
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4

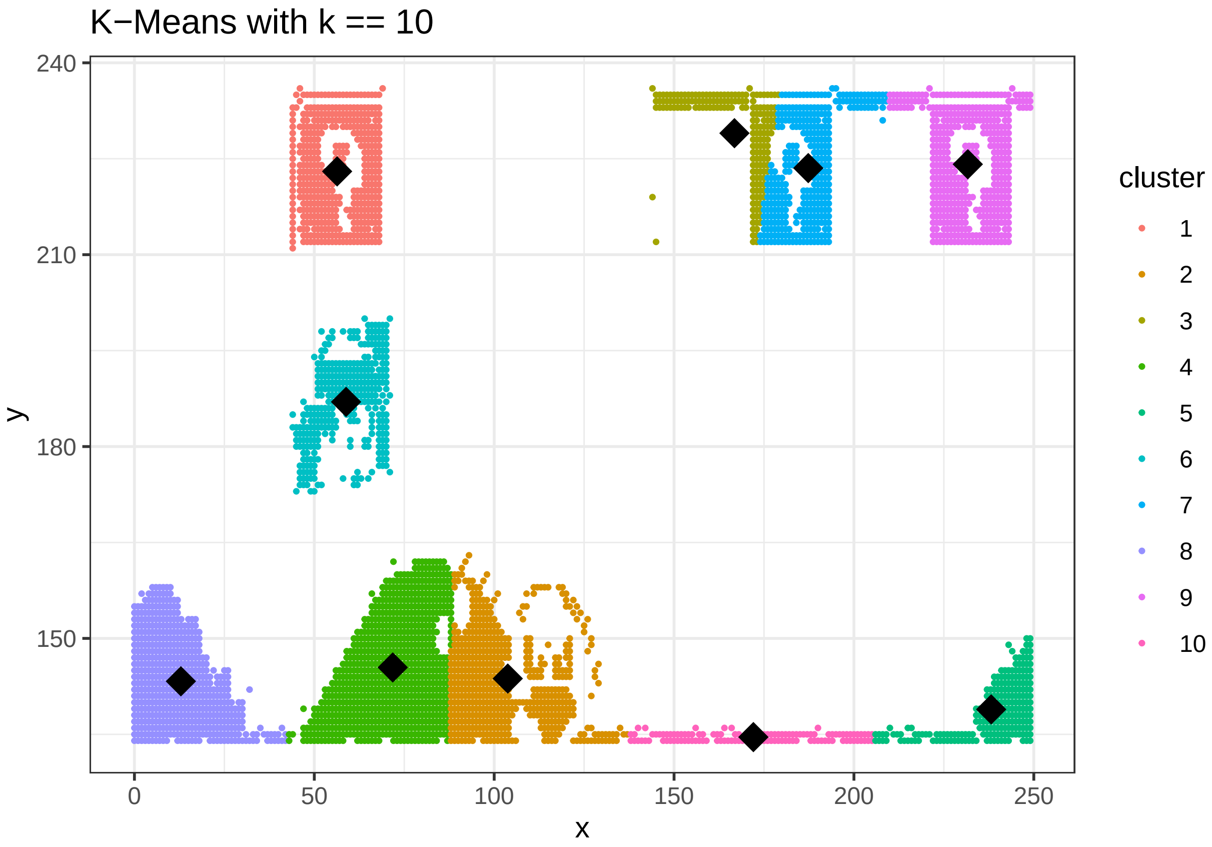

K−Means with k == 5
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5

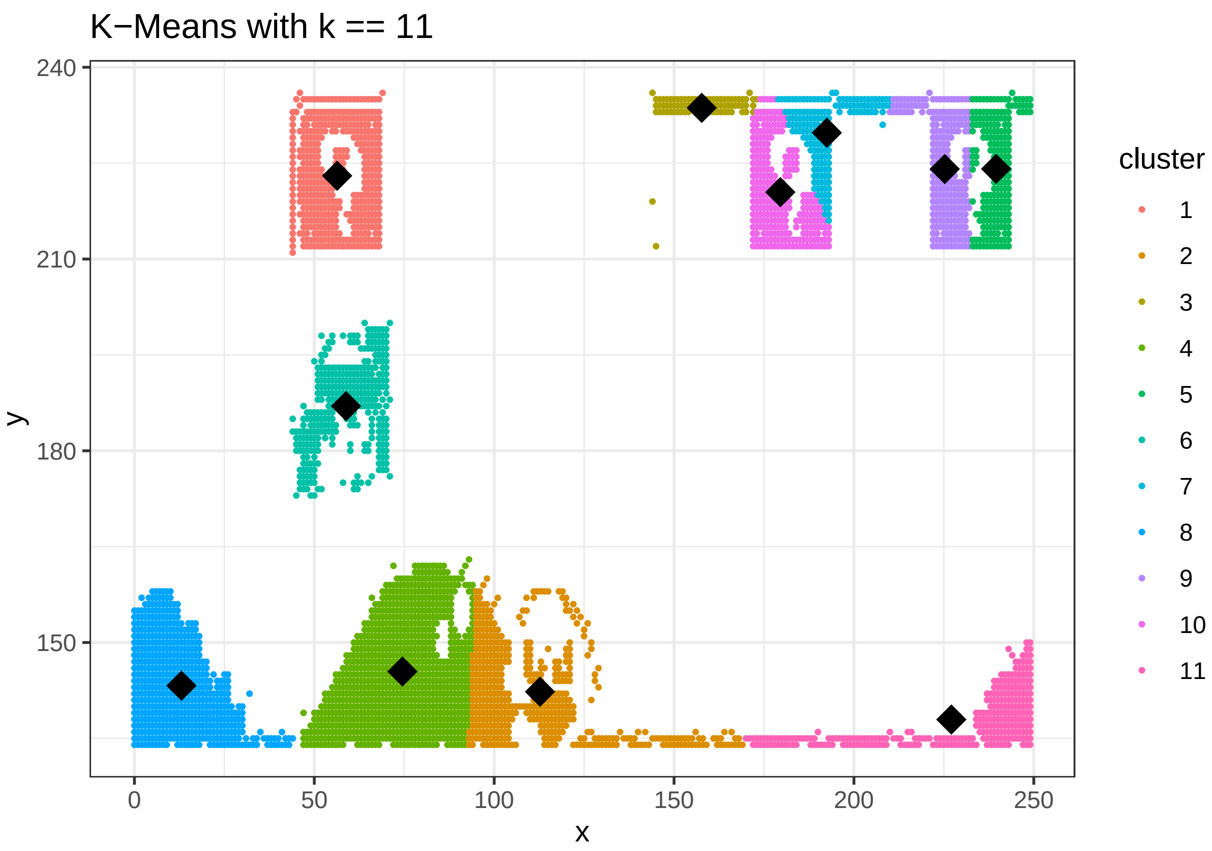

K−Means with k == 6
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6

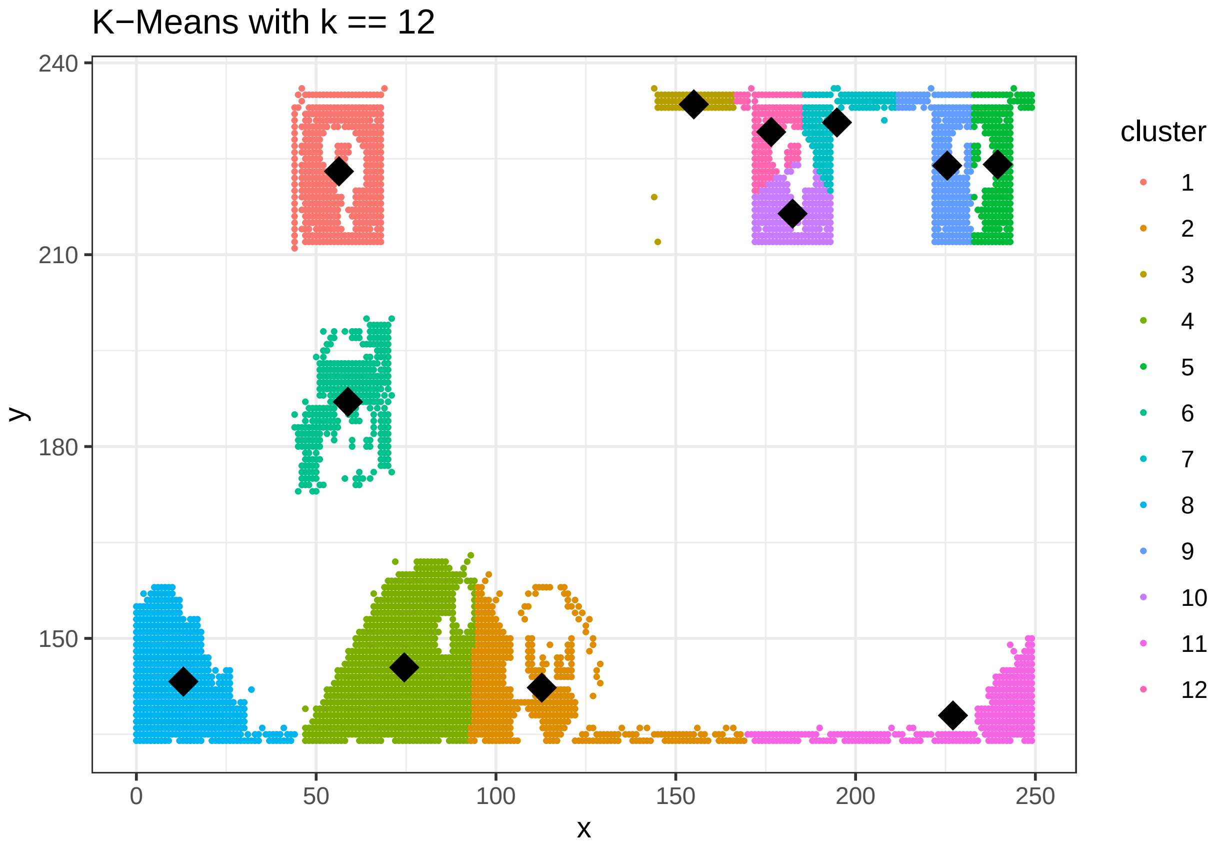

K−Means with k == 7
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7


K−Means with 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 with 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 with 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

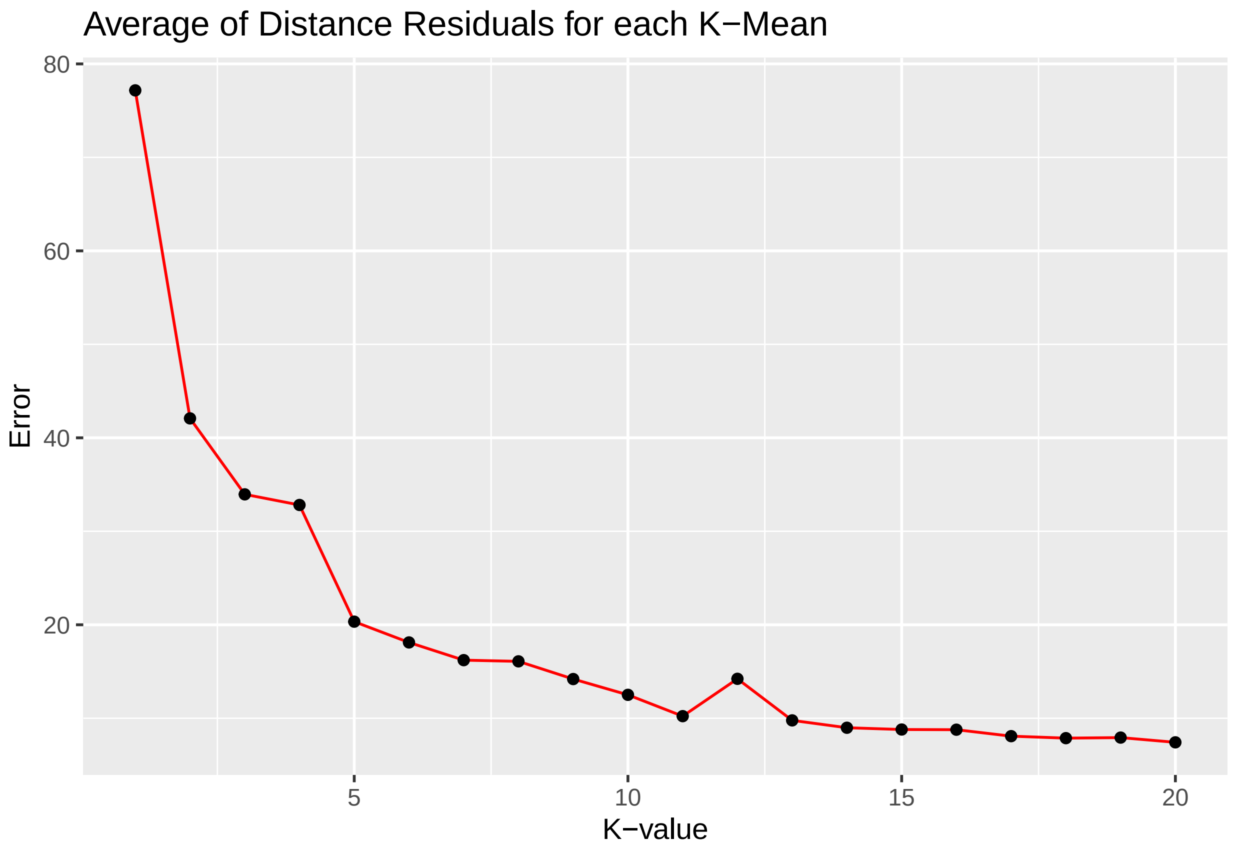

K−Means with k == 11
240
210
180
150
y
0
50
100
150
200
250
x
cluster
1
2
3
4
5
6
7
8
9
10
11



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.

```
set.seed(1)
errors <- NULL
pos <- 1

ks <- 1:20

for( i in ks){
  df <- read.csv("clustering-data.csv")

  df.cluster <- kmeans(df, centers = i)
  df$cluster <- as.factor(df.cluster$cluster)

  df$x.dist <- df.cluster$centers[df$cluster,"x"] - df$x
  df$y.dist <- df.cluster$centers[df$cluster,"y"] - df$y
  df$tot.dist <- sqrt((df$x.dist ** 2) + (df$y.dist ** 2))

  errors[pos] <- mean(df$tot.dist)
  pos <- pos + 1
}
```

```
error.df <- data.frame(ks,errors)

ggplot(data = error.df, aes(x = ks, y = errors)) +
geom_line(color = "red") +
ggtitle("Average of Distance Residuals for each K-Mean") +
xlab("K-value") +
ylab("Error") +
geom_point()
```



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?

There isn't a well defined elbow point using this data but it's approximately around k == 5. As k increases, the error continues to drop. A k value of 8 visually looks like a good fit, but some general domain knowledge of how the data should be clustered should be taken into account since the data is so strange.