

# Exercise 16: Clustering

Patkhedkar Ninad

2020-11-02

## Exercise 16: Clustering

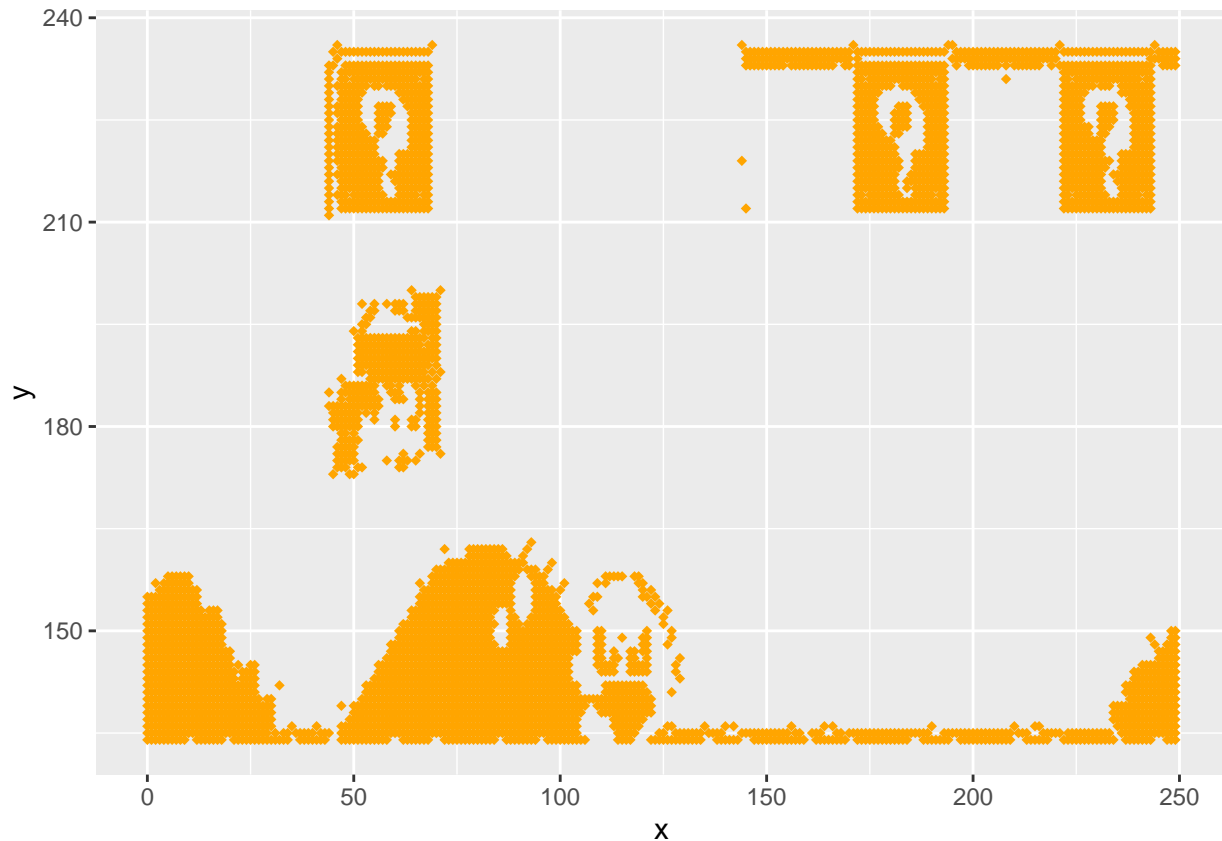
In this problem, you will use the k-means clustering algorithm to look for patterns in an unlabeled dataset. The dataset for this problem is found at `data/clustering-data.csv`.

- a. Plot the dataset using a scatter plot.

```
library(ggplot2)
library(ggfortify)
library(caTools)
library(knitr)
library(pander)
library(class)
library(tidyverse)

setwd("/cloud/project/")

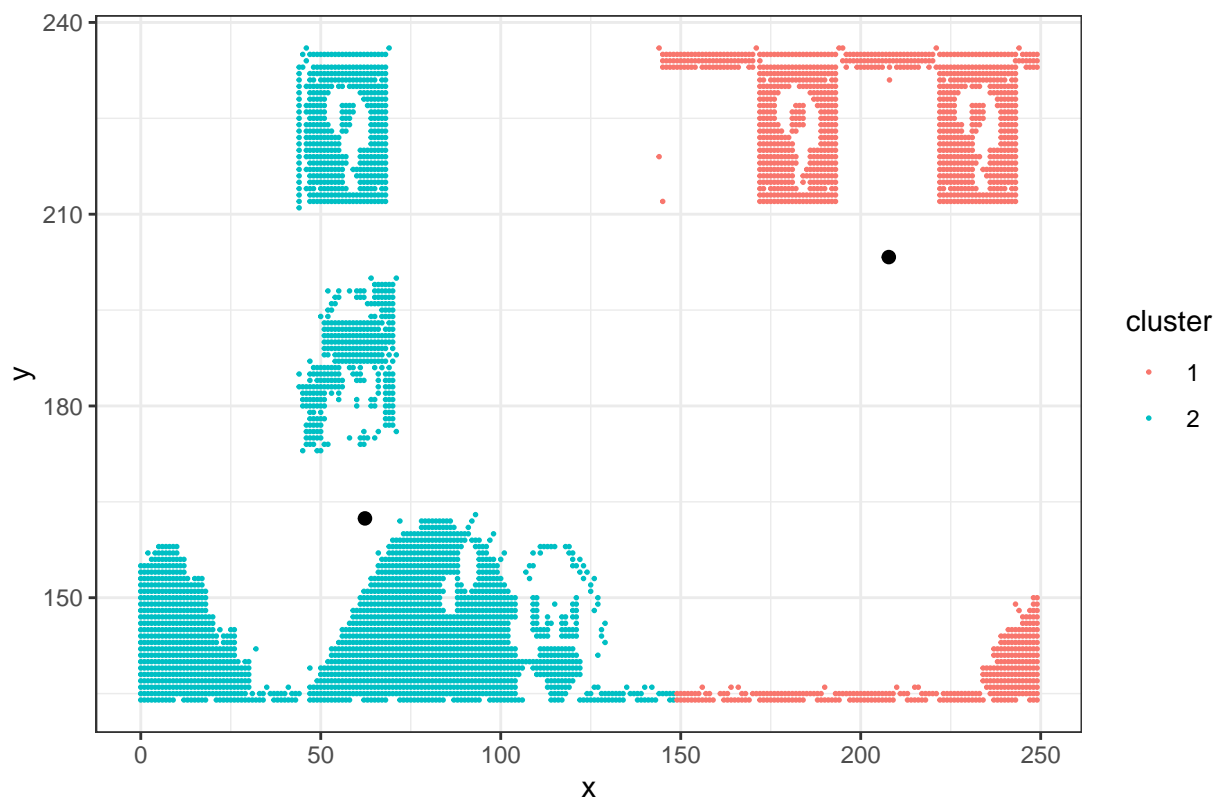
ex16_data <- read.csv("data/clustering-data.csv")
ggplot(ex16_data, aes(x=x, y=y)) +
  geom_point(shape=18, color="orange")
```



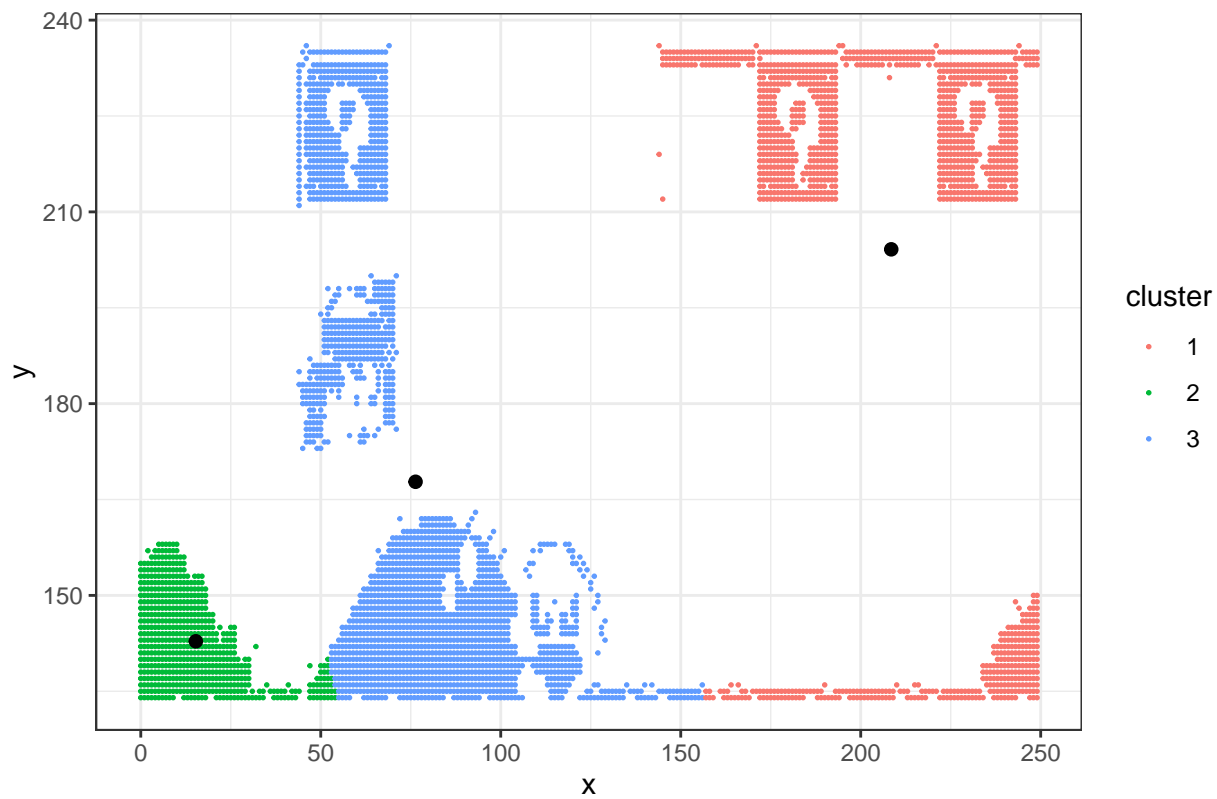
- b. Fit the dataset using the k-means algorithm from  $k=2$  to  $k=12$ . Create a scatter plot of the resultant clusters for each value of  $k$ .

```
for(i in 2:12){
  set.seed(123)
  ex16_data.cluster <- kmeans(ex16_data, i)
  ex16_data$cluster <- as.factor(ex16_data.cluster$cluster)
  p <- ggplot(data = ex16_data,
             aes(x = x,
                 y = y,
                 color = cluster)) +
    geom_point(size = 0.3) +
    geom_point(data = as.data.frame(ex16_data.cluster$centers),
              color = "black",
              shape = 20,
              size = 3) +
    ggtitle(paste("K-Means with k == ", i, sep = "")) +
    theme_bw()
  print(p)
}
```

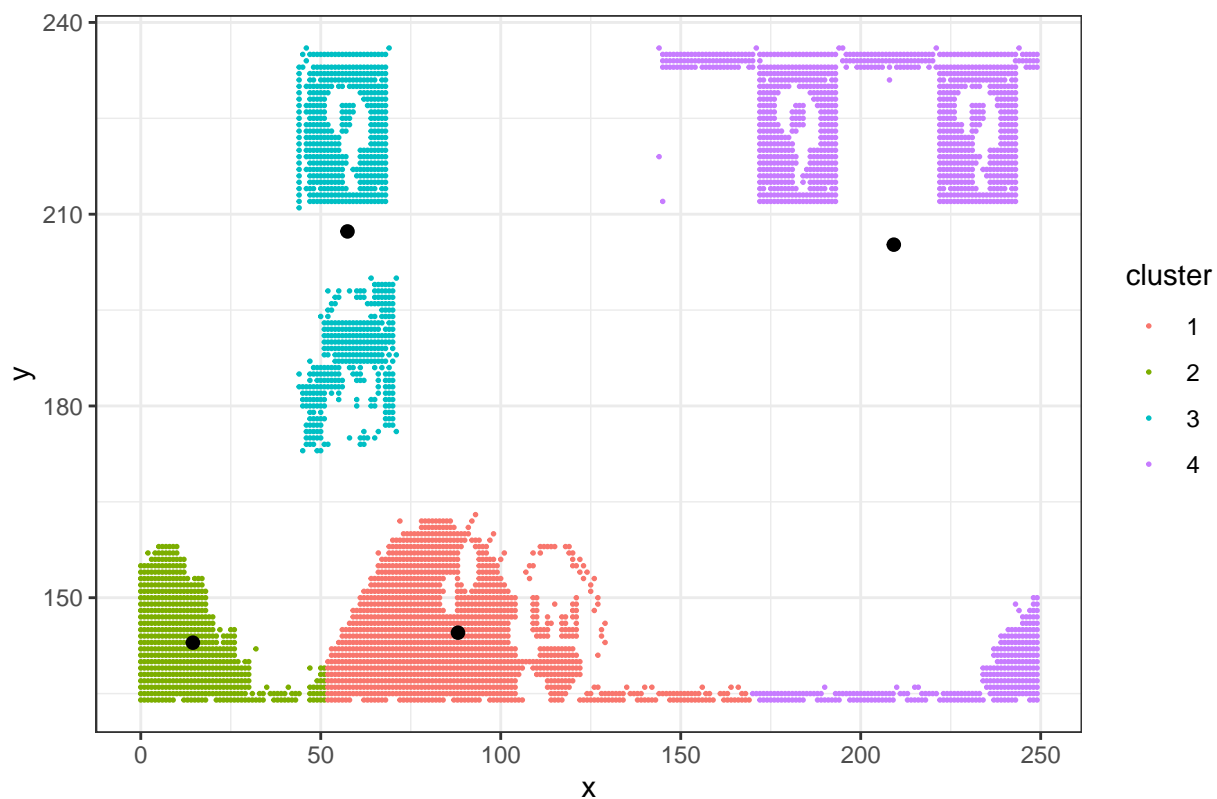
K-Means with k == 2



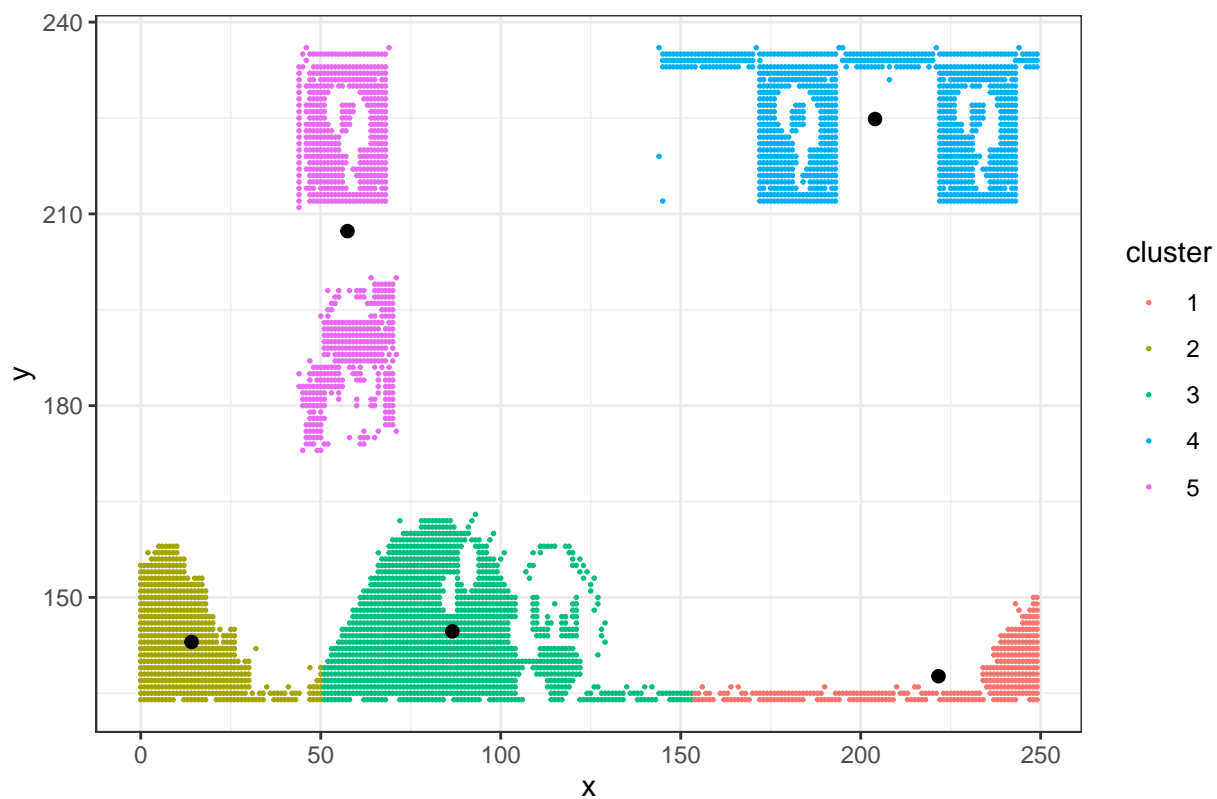
K-Means with k == 3



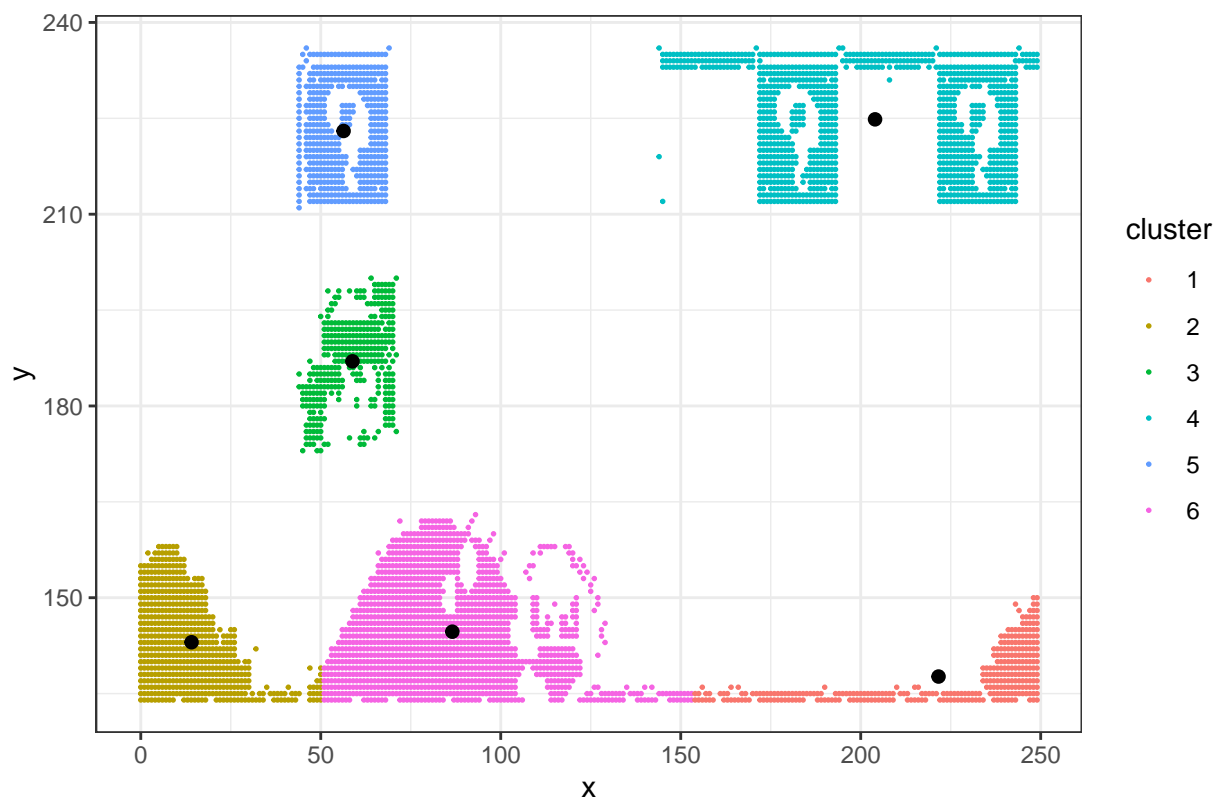
## K-Means with $k == 4$



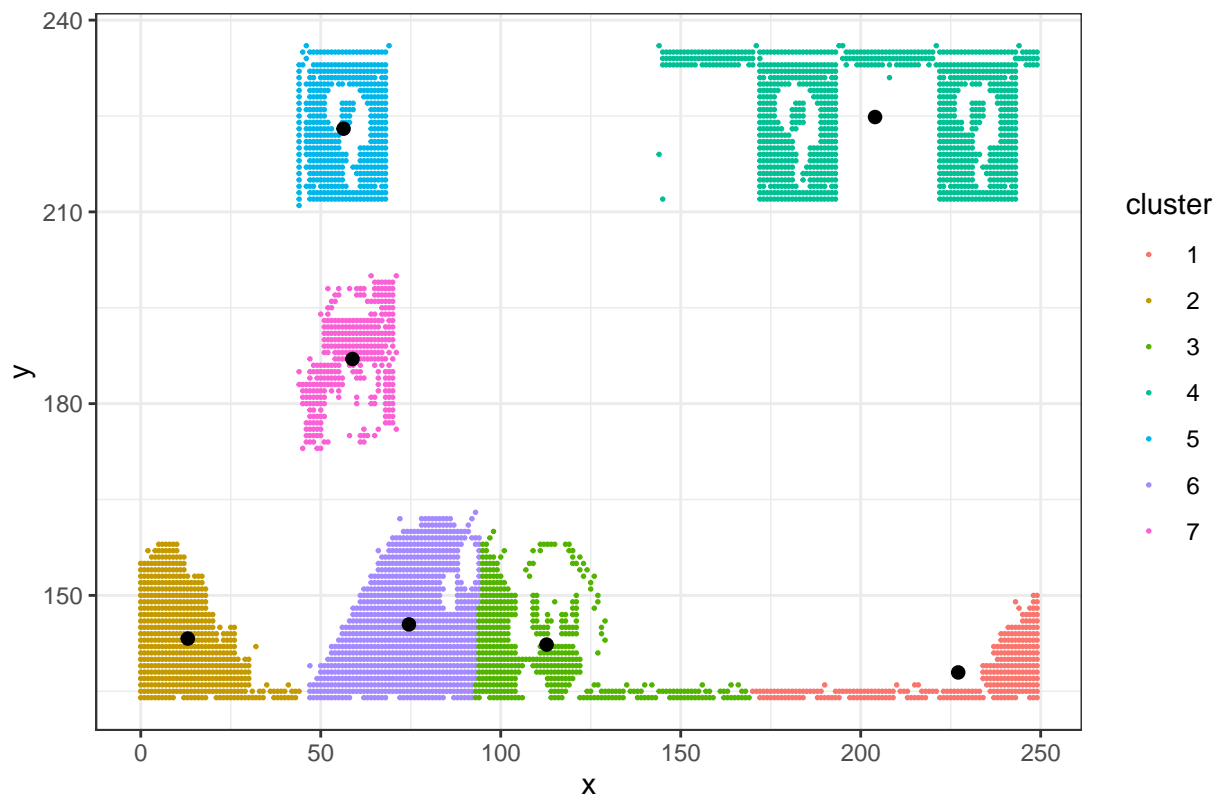
## K-Means with $k = 5$



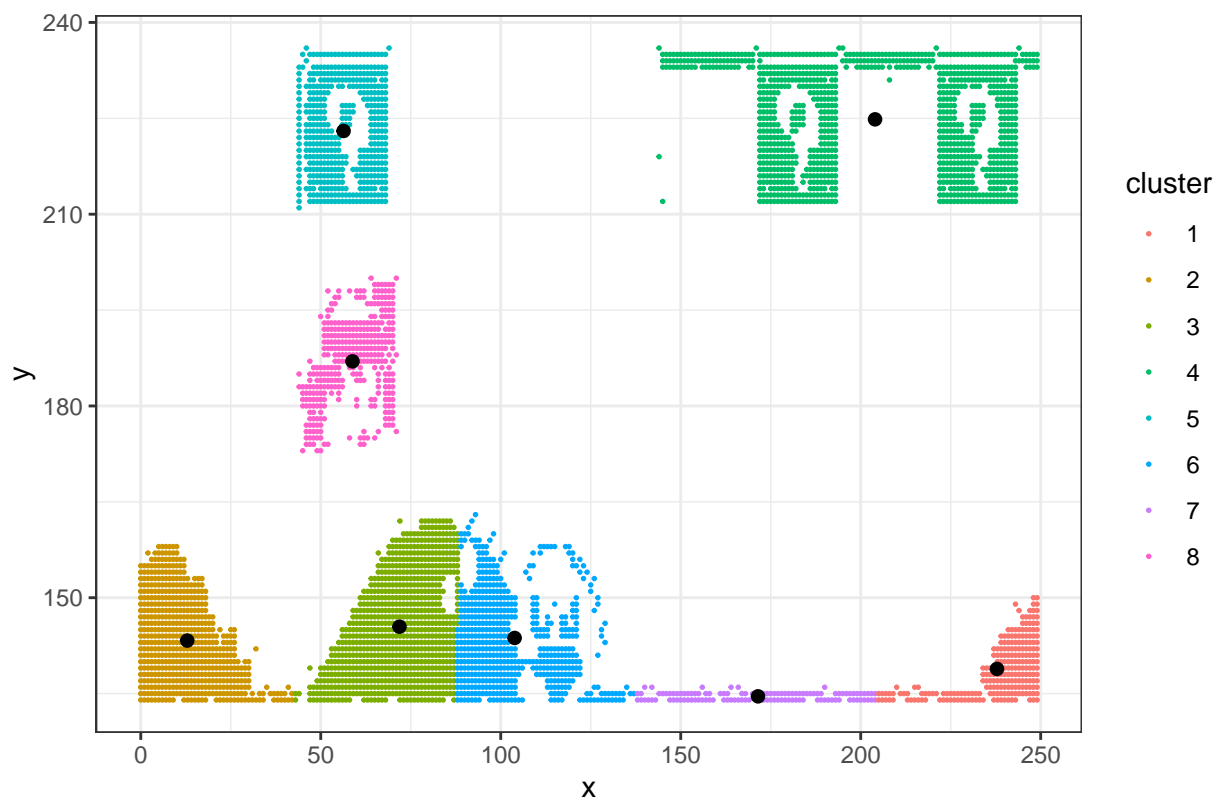
K-Means with k == 6



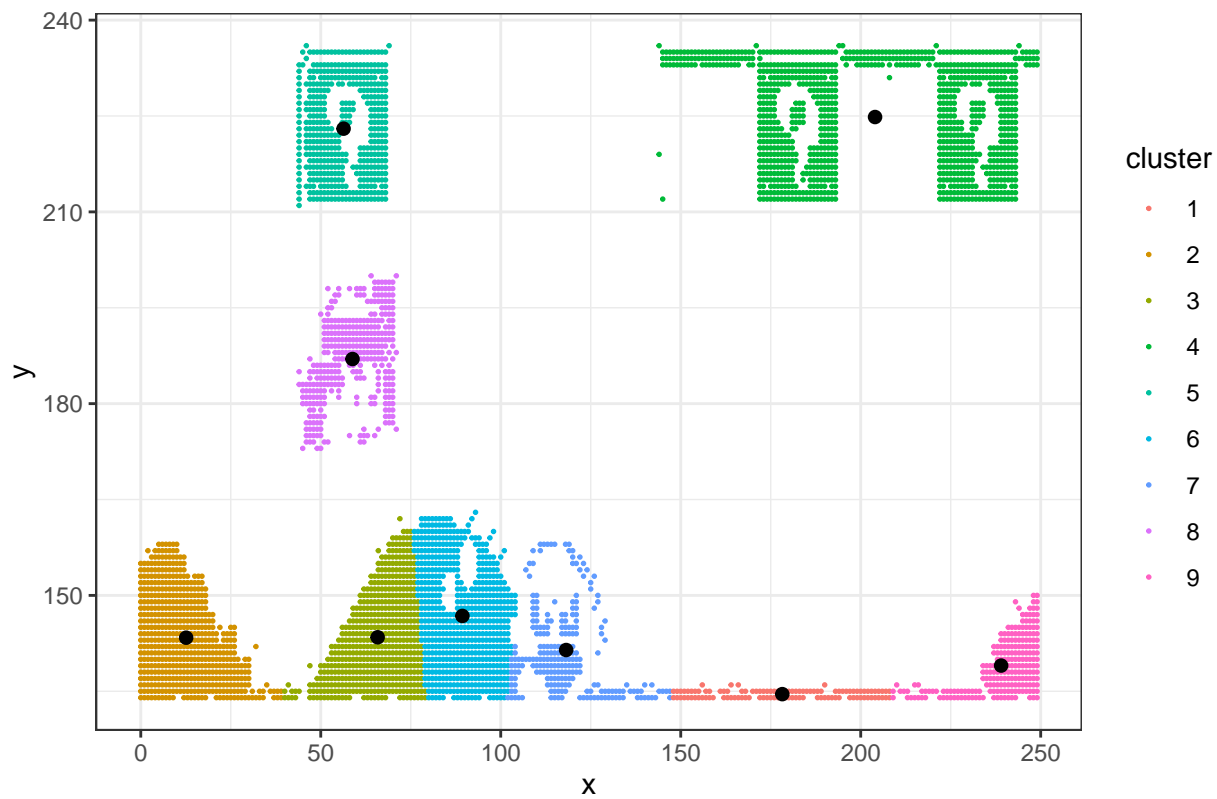
K-Means with k == 7



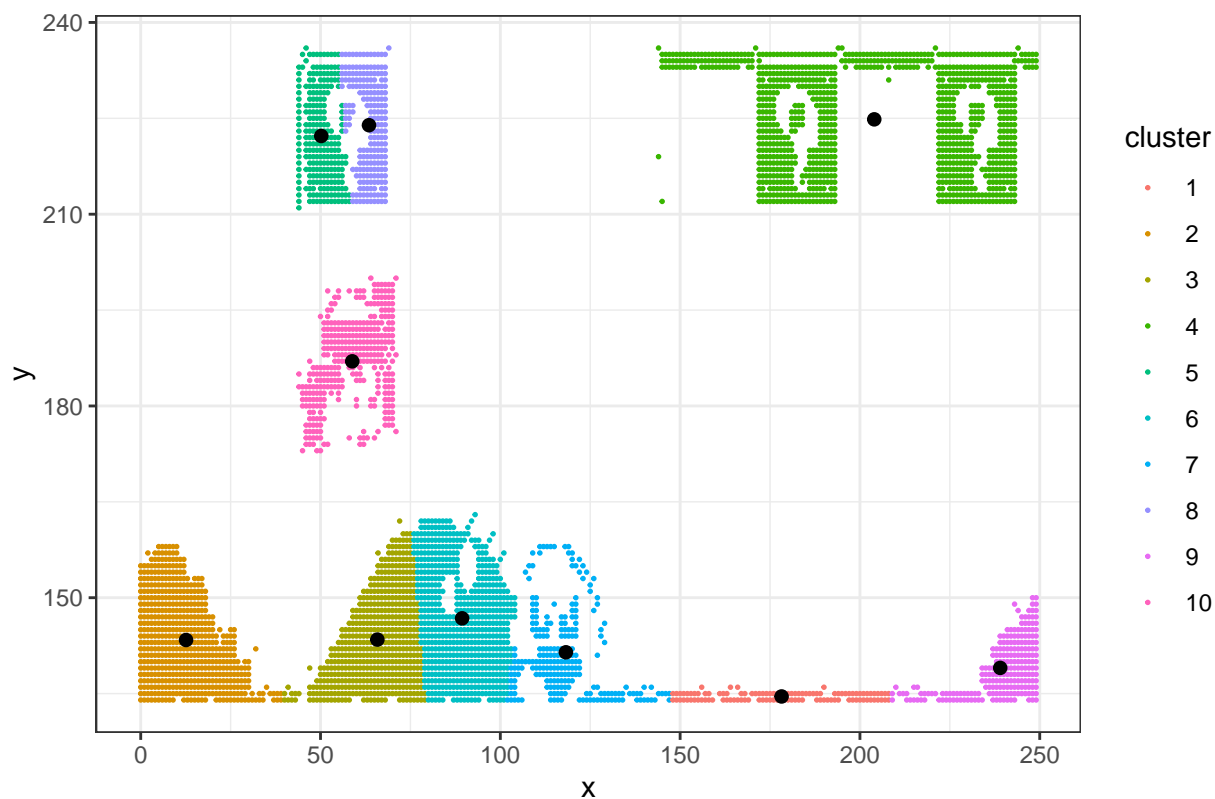
K-Means with k == 8



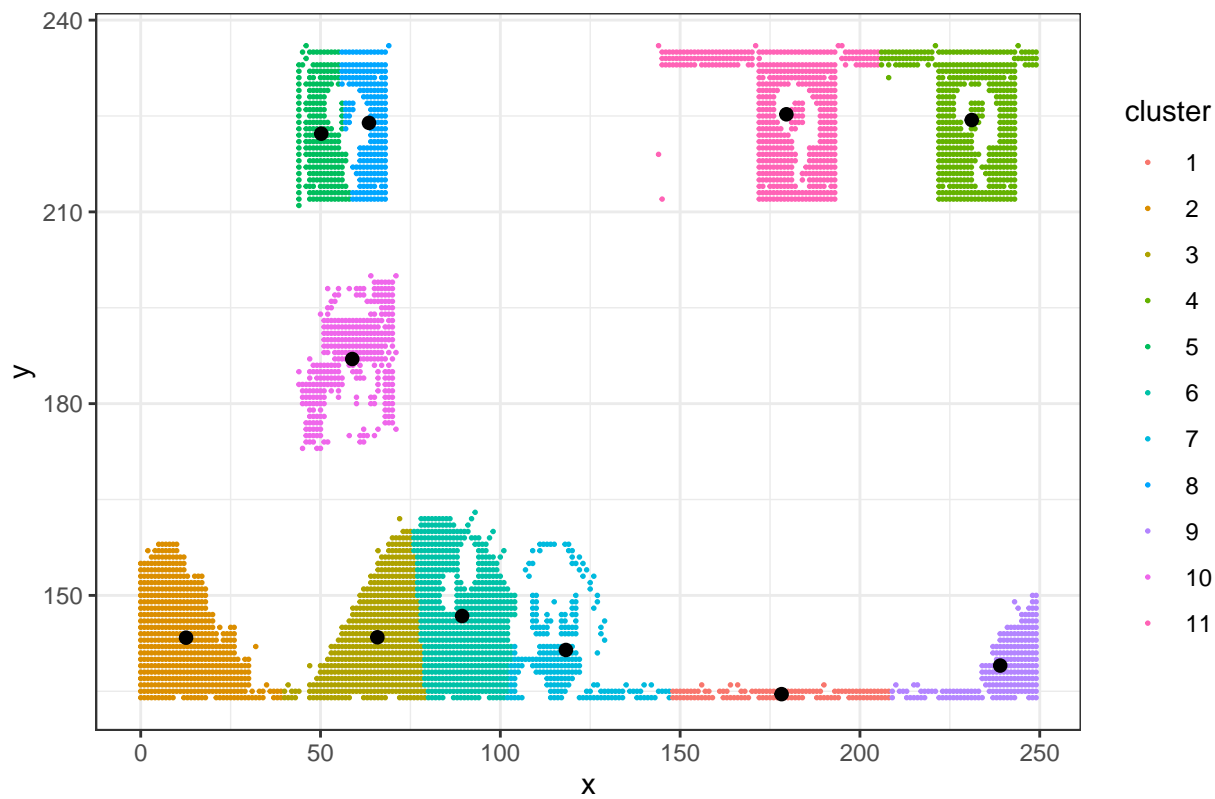
K-Means with k == 9

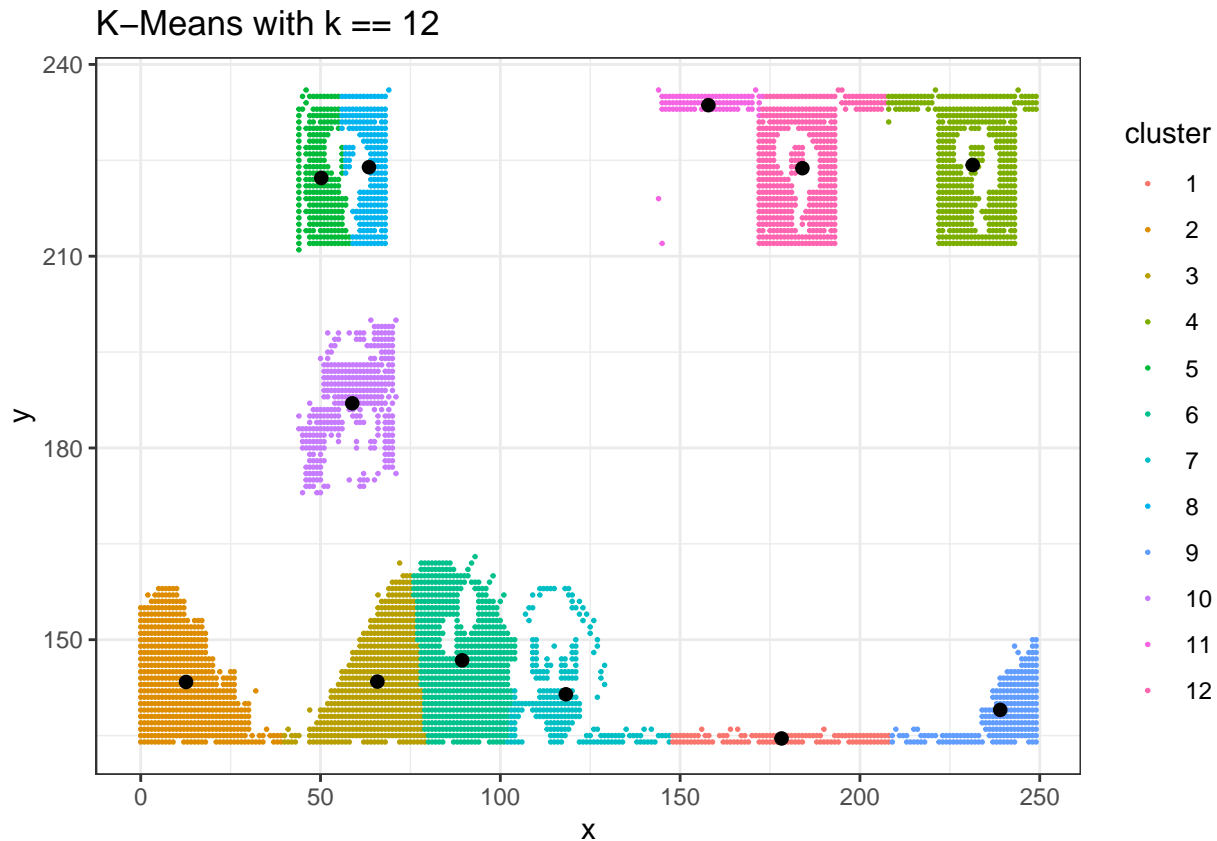


K-Means with k == 10



K-Means with k == 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.

```
# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(ex16_data, k, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 2 to k = 12
k.values <- 2:12

# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)
wss_values
```

```
## [1] 8483034.1 6052521.4 4036085.2 2198242.5 1530858.3 1277444.3 861067.4
## [8] 808431.9 556036.4 507429.2 463772.2
```

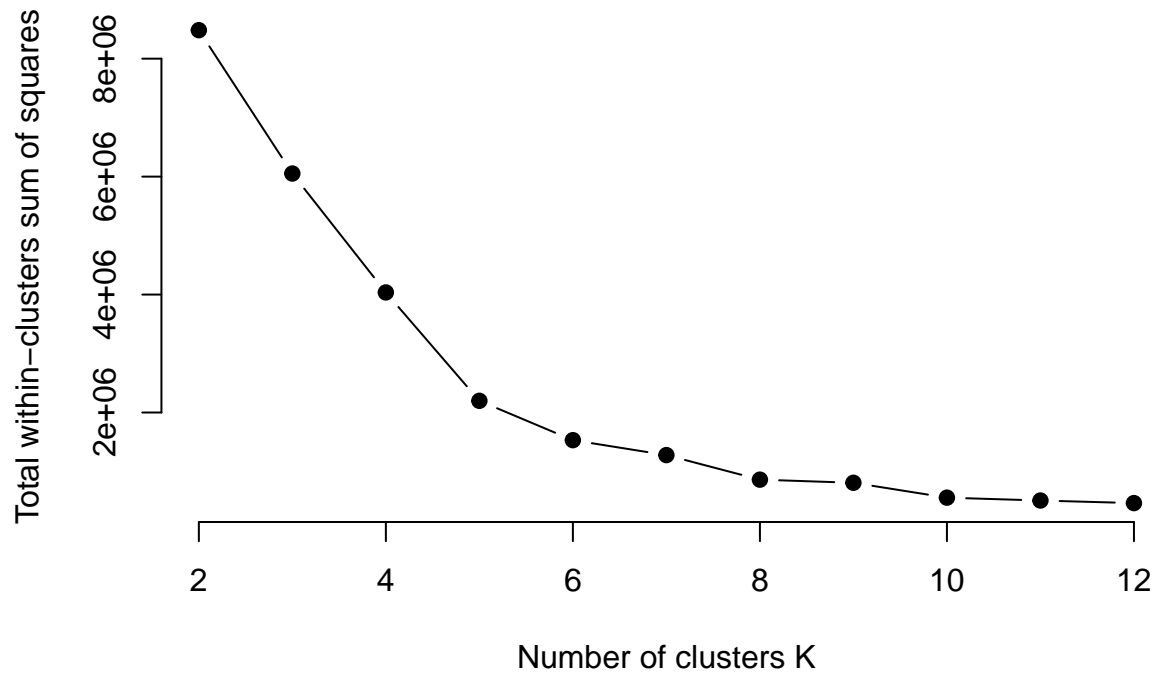
Calculate this average distance from the center of each cluster for each value of k and plot it as a line chart where k is the x-axis and the average distance is the y-axis.

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?

```
plot(k.values, wss_values,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
```



```
ylab="Total within-clusters sum of squares")
```



Looking

at result **6** looks to be optimal number of clusters as there is kind of bend in elbow at 6.