Exercise 11.2: Machine Learning

Madeleine Sharp

2020-11-12

Introduction to Machine Learning

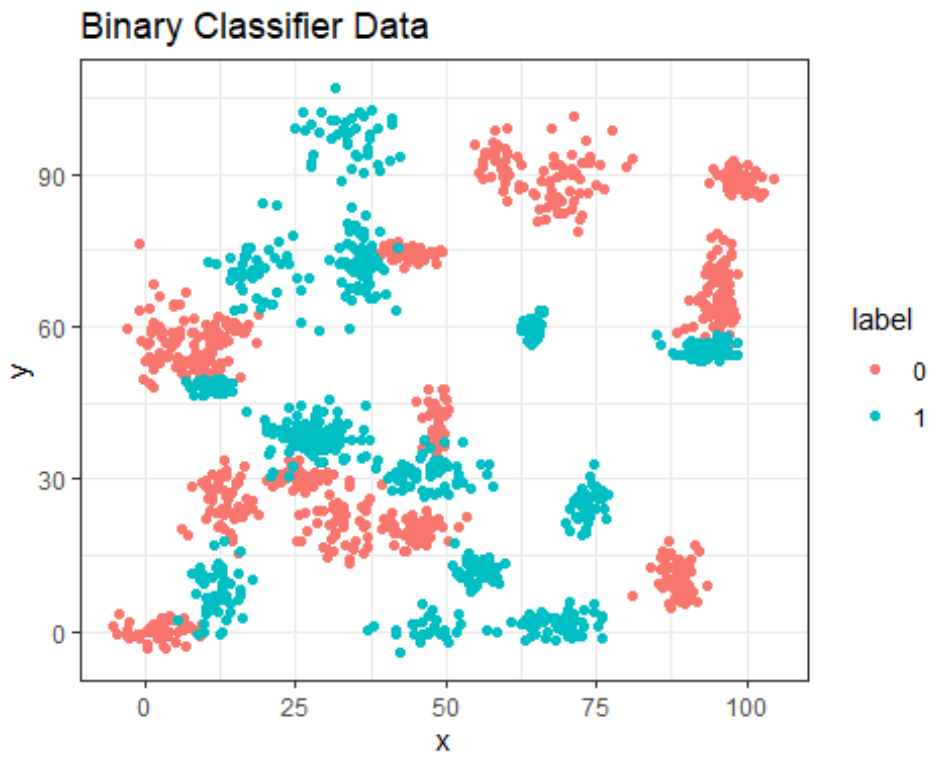
a. These assignments are here to provide you with an introduction to the "Data Science" use for these tools. This is your future. It may seem confusing and weird right now but it hopefully seems far less so than earlier in the semester. Attempt these homework assignments. You will not be graded on your answer but on your approach. This should be a, "Where am I on learning this stuff" check. If you can't get it done, please explain why.
b. Include all of your answers in a R Markdown report.
c. Regression algorithms are used to predict numeric quantity while classification algorithms predict categorical outcomes. A spam filter is an example use case for a classification algorithm. The input dataset is email labeled as either spam (i.e. junk emails) or ham (i.e. good emails). The classification algorithm uses features extracted from the emails to learn which emails fall into which category.
d. In this problem, you will use the nearest neighbors algorithm to fit a model on two simplified datasets. The first dataset (found in binary-classifier-data.csv) contains three variables; label, x, and y. The label variable is either 0 or 1 and is the output we want to predict using the x and y variables (You worked with this dataset last week!). The second dataset (found in trinary-classifier-data.csv) is similar to the first dataset except that the label variable can be 0, 1, or 2.
e. Note that in real-world datasets, your labels are usually not numbers, but text-based descriptions of the categories (e.g. spam or ham). In practice, you will encode categorical variables into numeric values.

•

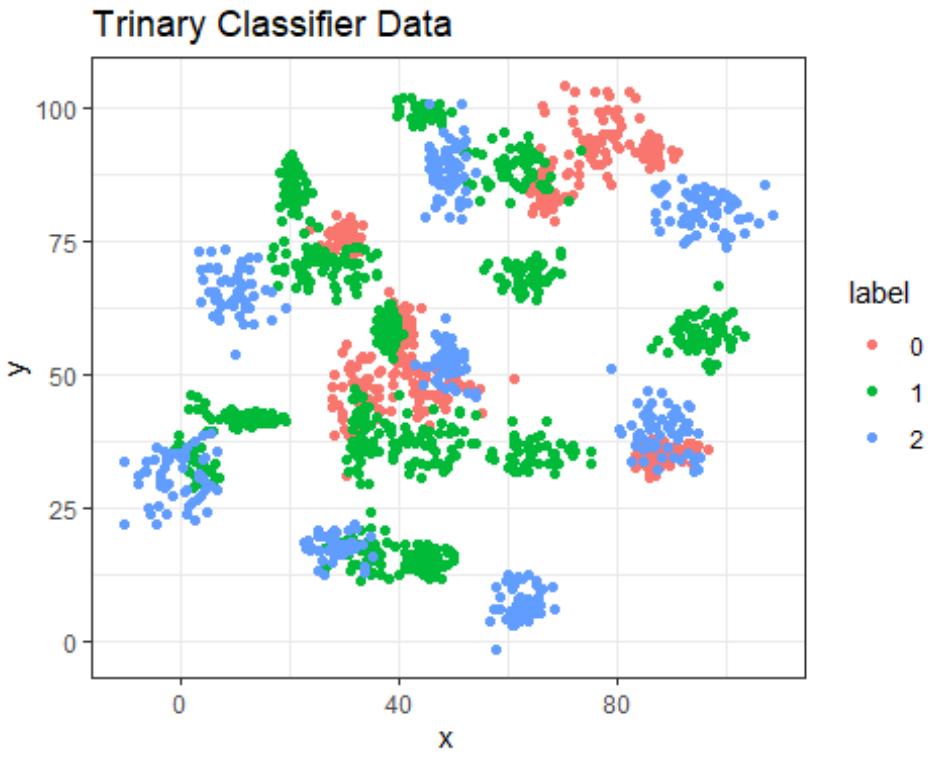
i. Plot the data from each dataset using a scatter plot.

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

## Set the working directory to the root of your DSC 520 directory
setwd("/Users/Madeleine's PC/Documents/Madeleine/Documents/Bellevue
University Courses/Masters in DS/BU DSC520/dsc520")

binary_df <- read.csv("data/binary-classifier-data.csv")
trinary_df <- read.csv("data/trinary-classifier-data.csv")

binary_df$label <- as.factor(binary_df$label)
```

```
trinary_df$label <- as.factor(trinary_df$label)

ggplot(data = binary_df, aes(x = x, y = y, color = label)) + geom_point() +
ggtitle("Binary Classifier Data") +
theme_bw()
```




```
ggplot(data = trinary_df, aes(x = x, y = y, color = label)) +
geom_point() +
ggtitle("Trinary Classifier Data") +
theme_bw()
```



-

  ii. The k nearest neighbors algorithm categorizes an input value by looking at the labels for the k nearest points and assigning a category based on the most common label. In this problem, you will determine which points are nearest by calculating the Euclidean distance between two points. As a refresher, the Euclidean distance between two points:

See image below (embed image)

-

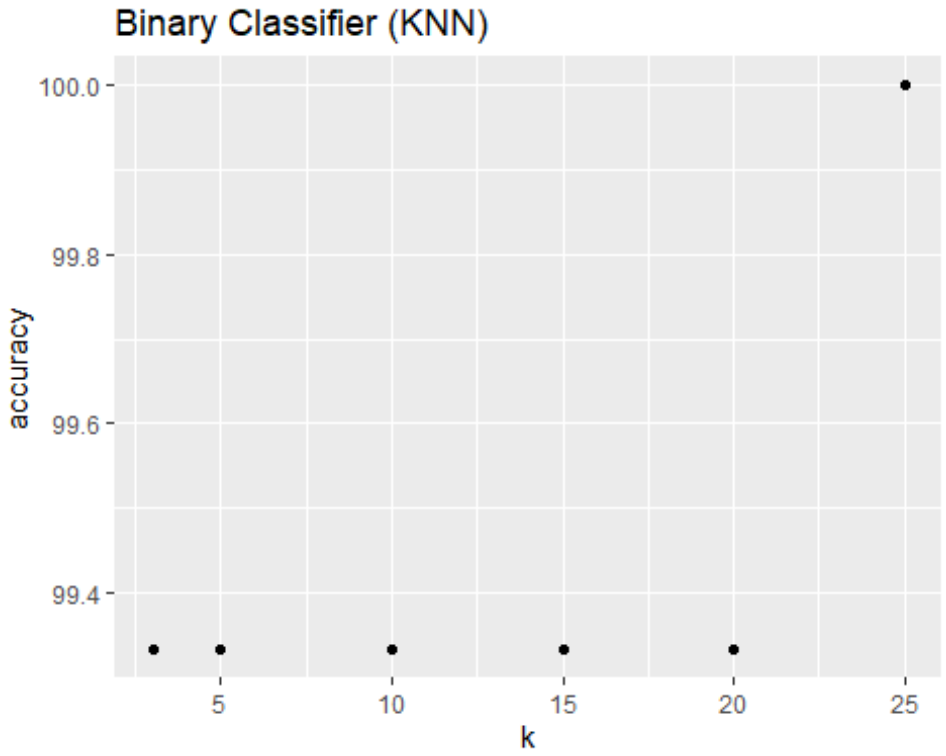
  i. Fitting a model is when you use the input data to create a predictive model. There are various metrics you can use to determine how well your model fits the data. For this problem, you will focus on a single metric, accuracy. Accuracy is simply the percentage of how often the model predicts the correct result. If the model always predicts the correct result, it is 100% accurate. If the model always predicts the incorrect result, it is 0% accurate.

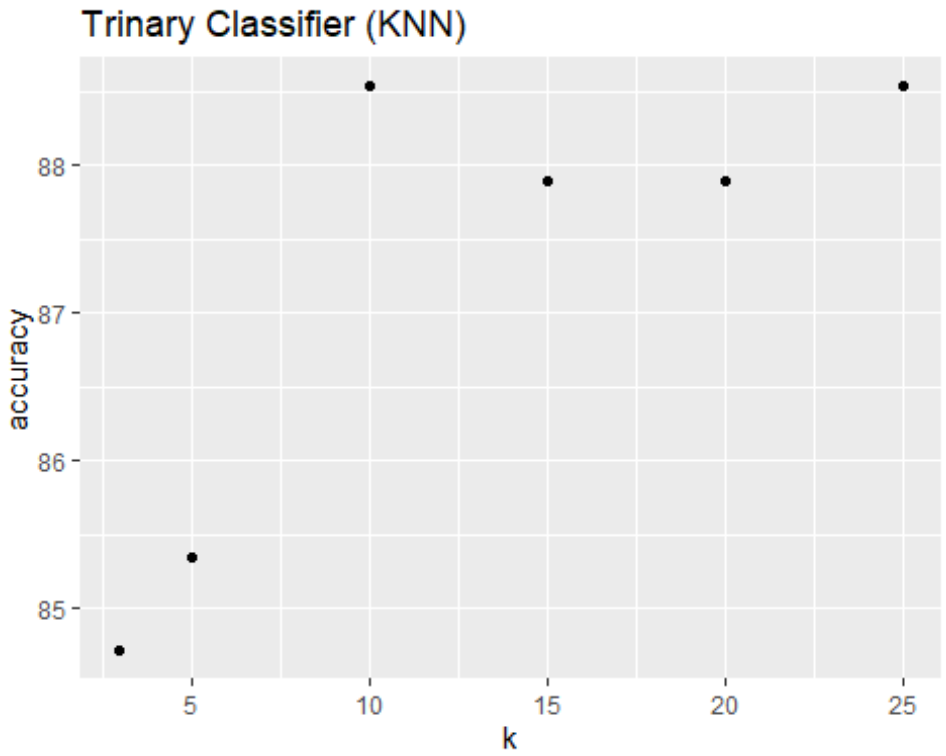
-

  ii. Fit a k nearest neighbors' model for each dataset for k=3, k=5, k=10, k=15, k=20, and k=25. Compute the accuracy of the resulting models for each value of k. Plot the results in a graph where the x-axis is the different values of k and the y-axis is the accuracy of the model.

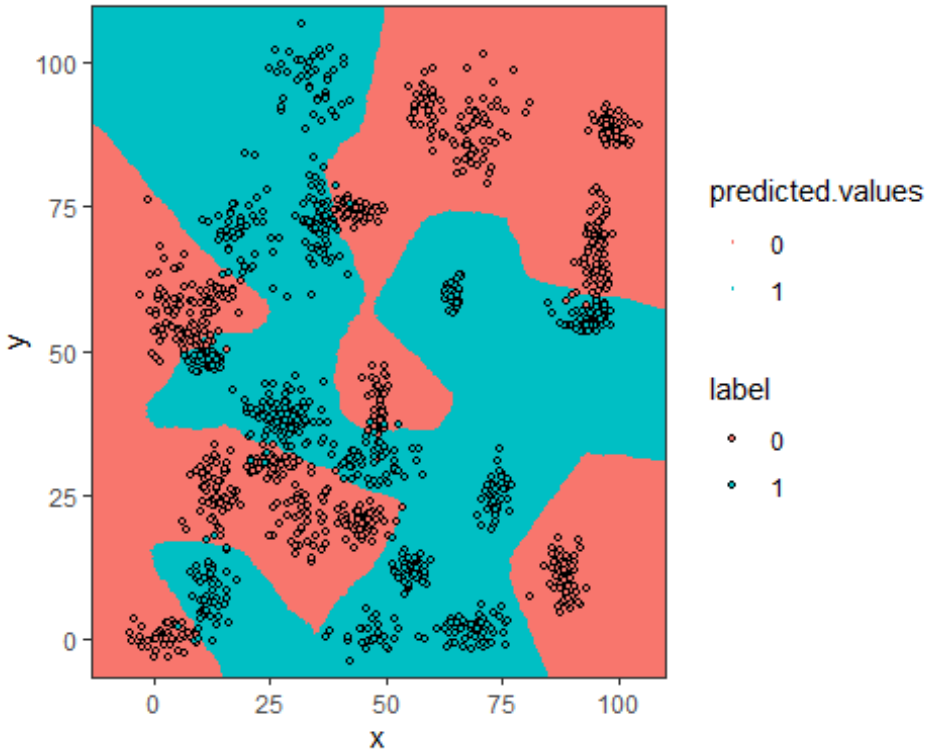
I decided to try this a couple of ways based on my research and reviewing multiple sources - please see the below.

Method 1

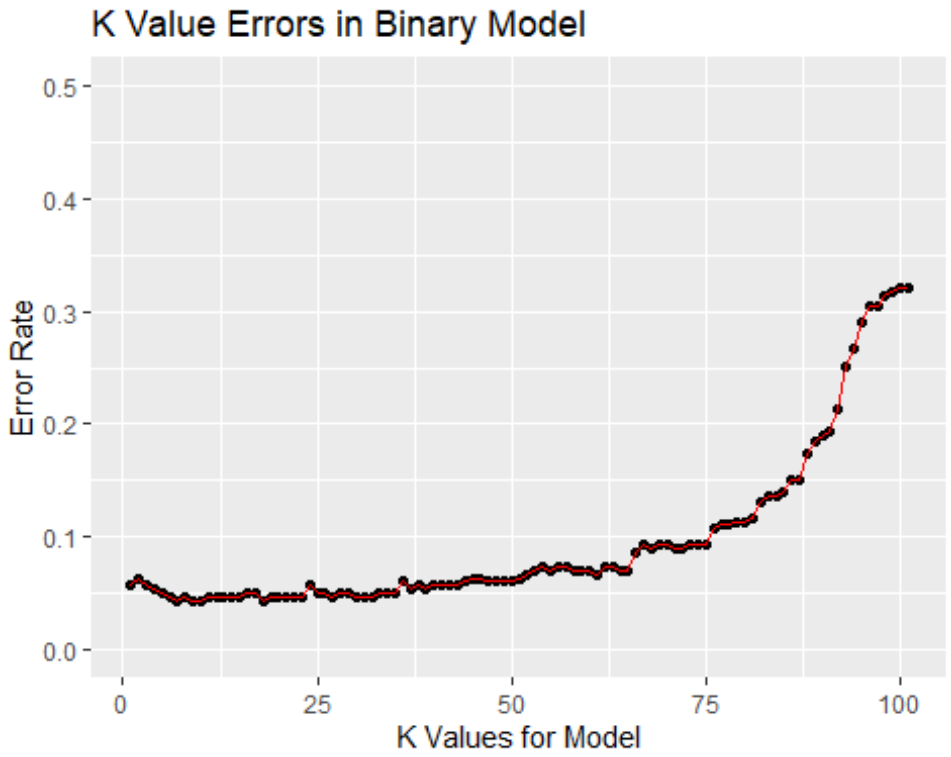

Binary Classifier (KNN)
100.0
99.8
99.6
99.4
accuracy
5
10
15
20
25
k

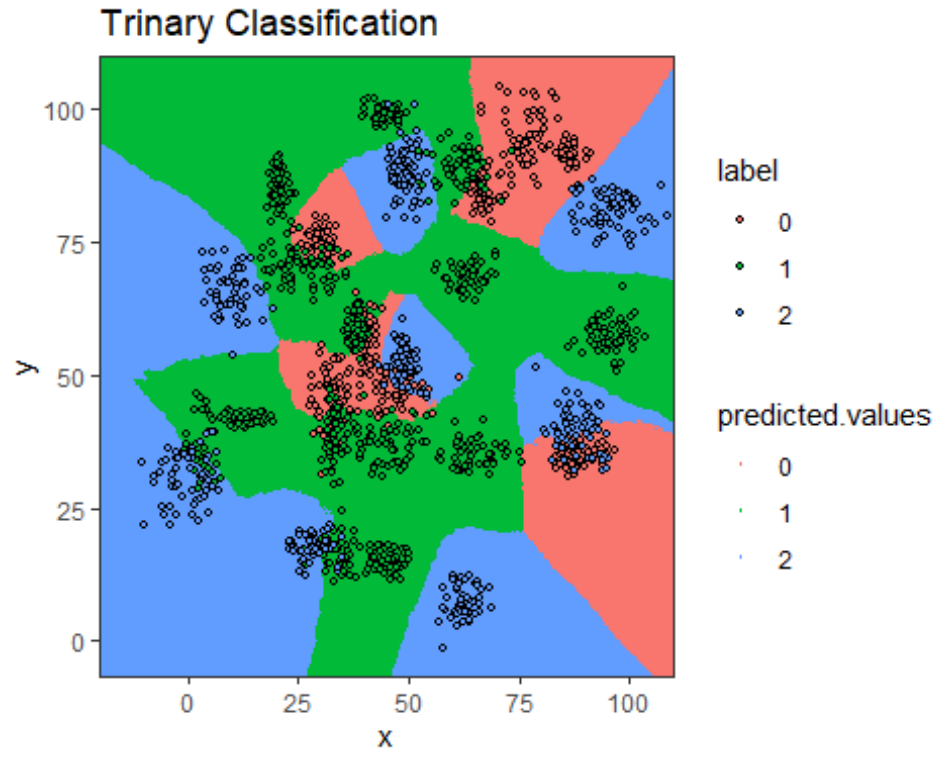

Trinary Classifier (KNN)
88
87
86
85
accuracy
5
10
15
20
25
k

Method 2



k.values	error.rate
3	0.05686
5	0.05017
10	0.04348
15	0.04682
20	0.04682
25	0.05017

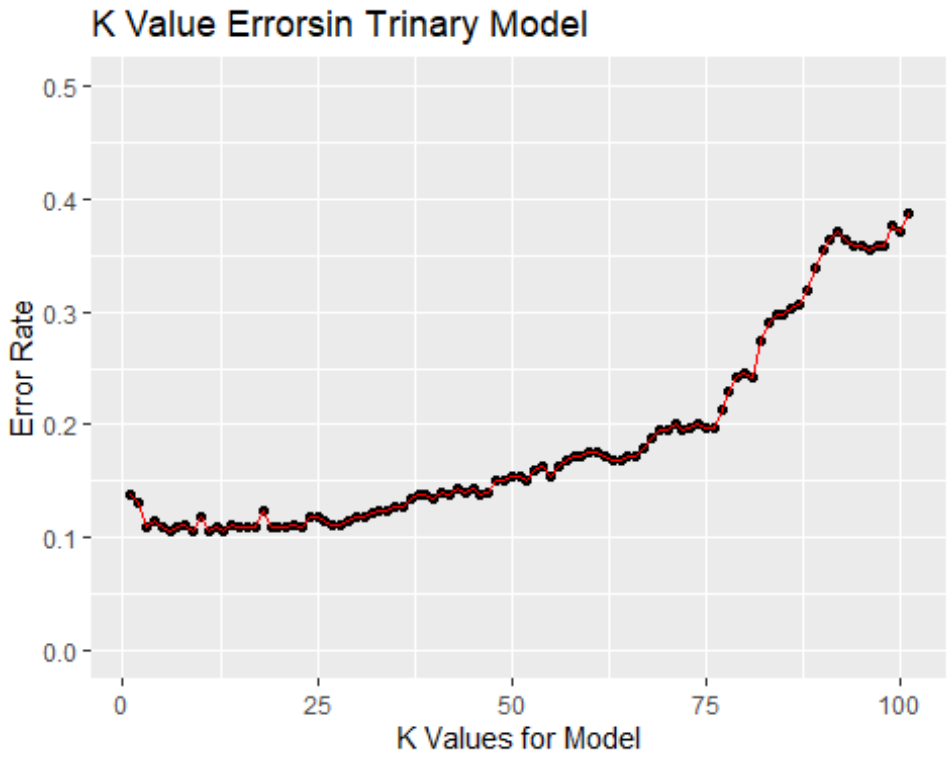




k.values	error.rate
3	0.1086

```
5          0.1086
```

```
10        0.1182

15        0.1086

20        0.1086

25        0.1182
```



* i. Looking back at the plots of the data, do you think a linear classifier would work well on these datasets?

No, I do not believe that a linear model would be a good fit for this dataset and these data points. Rather, it appears as though the data is less linear and more cluster-like. Not only that, but this data appears to be too complex to be linear, and thus no line would be best-of-fit or give a good decision boundary. Overall, because this dataset is "a bit all over," a linear classifier/function would not reasonably be able to partition the data.

- ii. How does the accuracy of your logistic regression classifier from last week compare? Why is the accuracy different between these two methods?

The previous model appeared to be more accurate, at approximately 81%. The difference likely has to do with the ways in which the models are trained and tested.

Clustering

a. These assignments are here to provide you with an introduction to the "Data Science" use for these tools. This is your future. It may seem confusing and weird right now but it hopefully seems far less so than earlier in the semester. Attempt these homework assignments. You will not be graded on your answer but on your approach. This should be a, "Where am I on learning this stuff" check. If you can't get it done, please explain why.
b. Remember to submit this assignment in an R Markdown report.
c. 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.
d. 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.

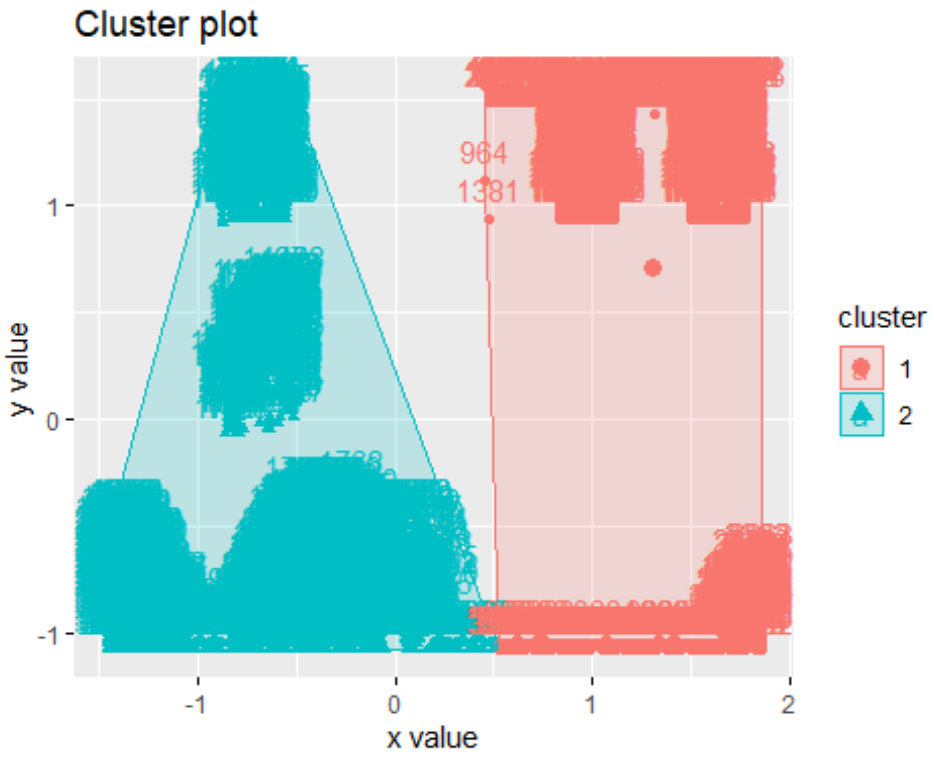
- i. Plot the dataset using a scatter plot.

```
## '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 ...

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

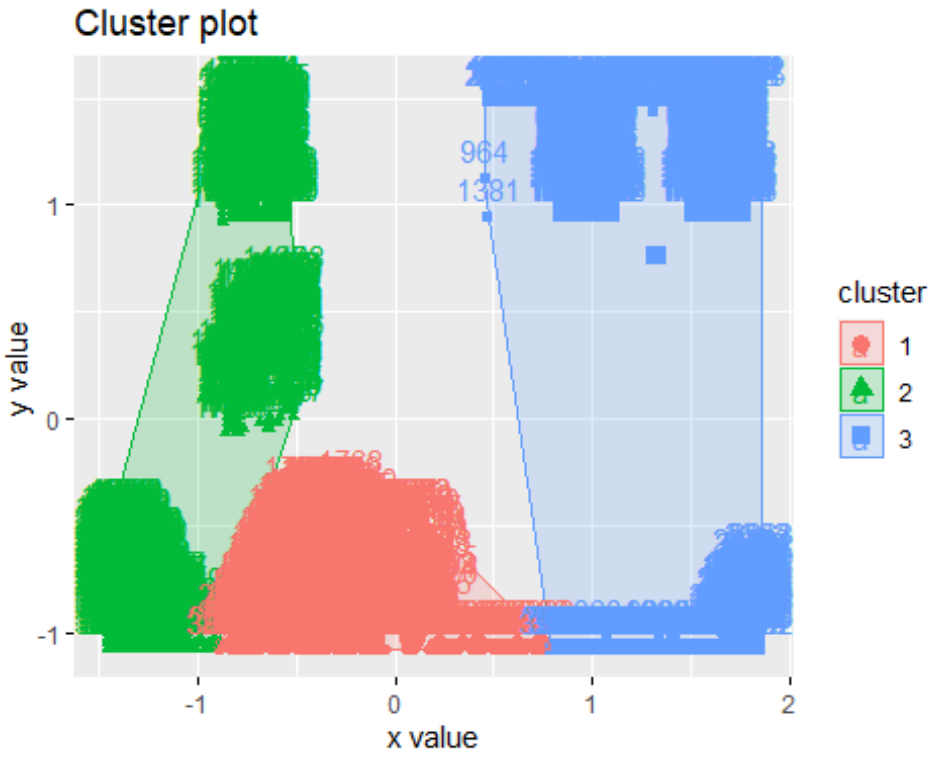
- ii. Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

- iii. 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.

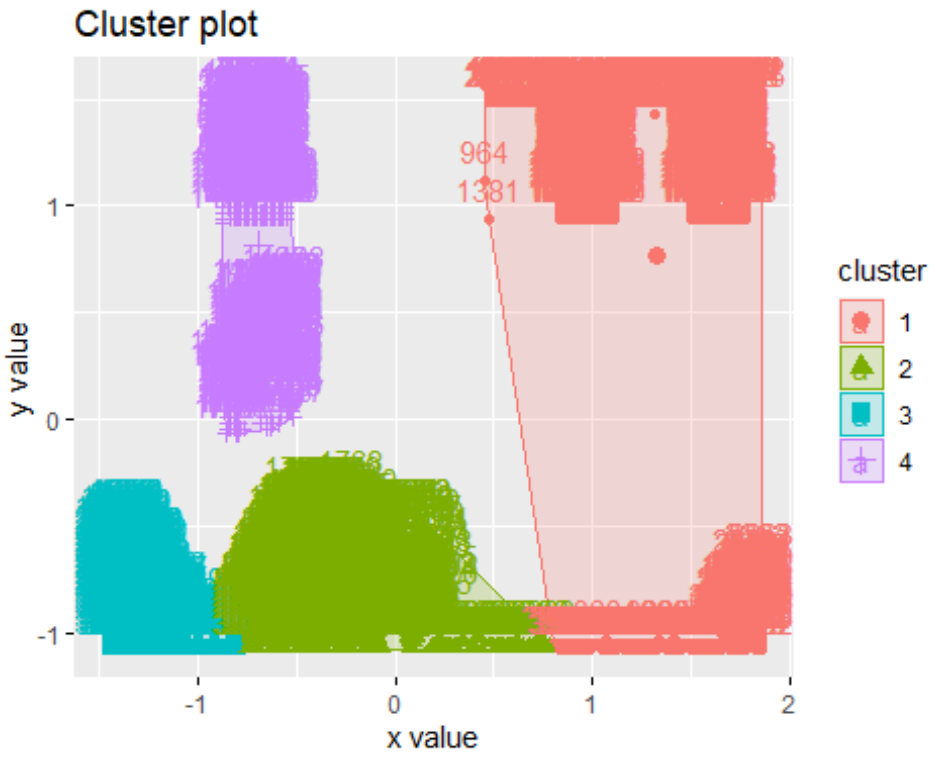
Cluster k=2



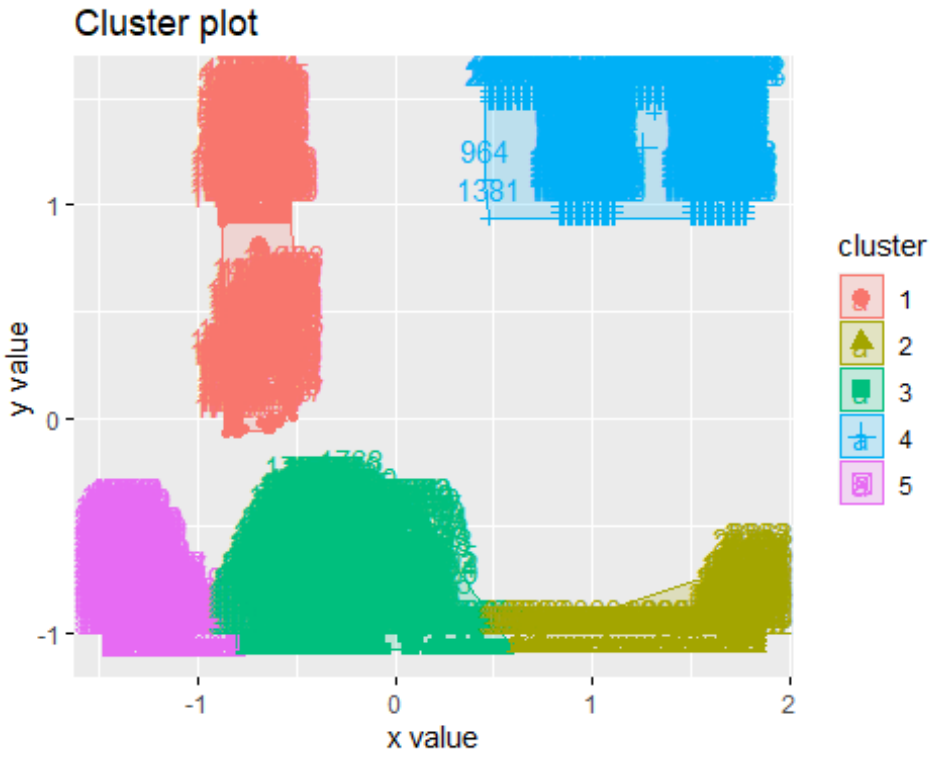
Cluster k=3



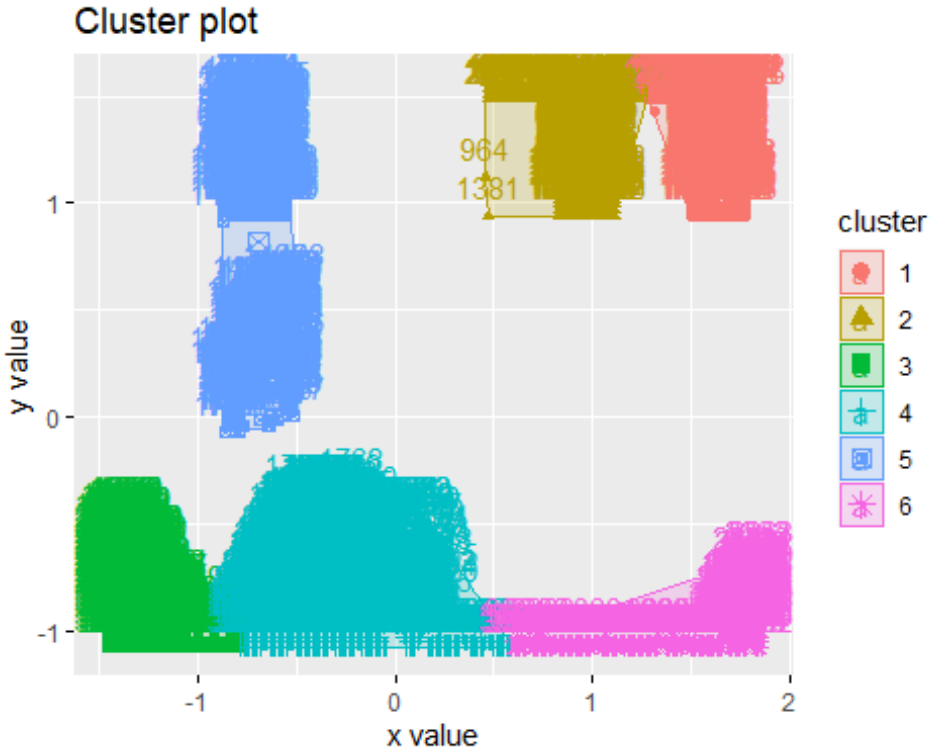
Cluster k=4


Cluster plot
964
1381
y value
x value
cluster
1
2
3
4

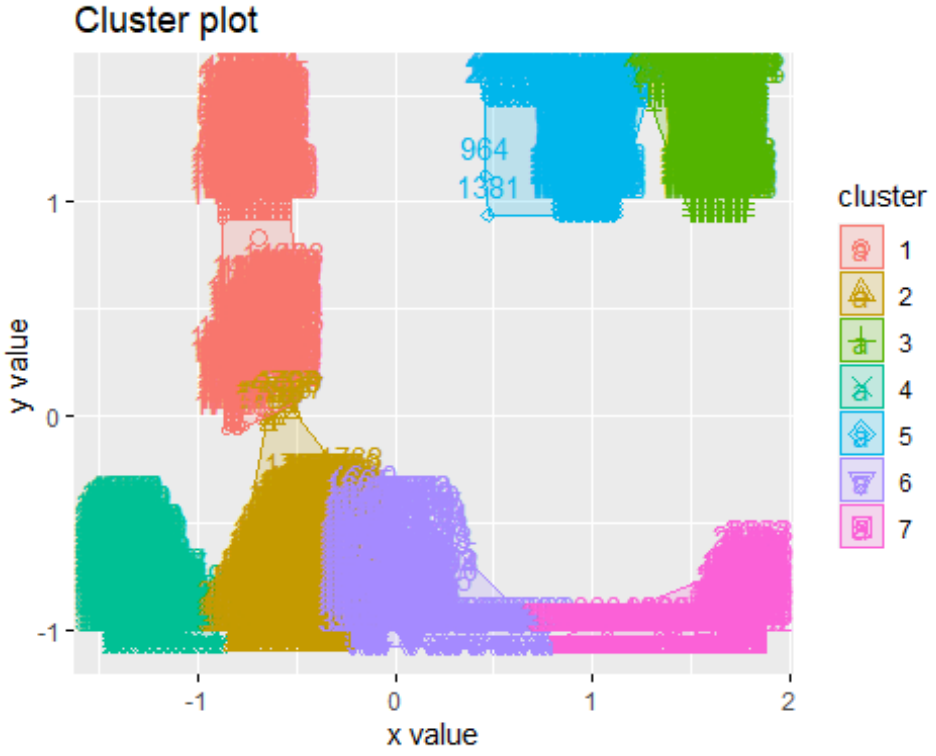
Cluster k=5


Cluster plot
964
1381
y value
x value
cluster
1
2
3
4
5

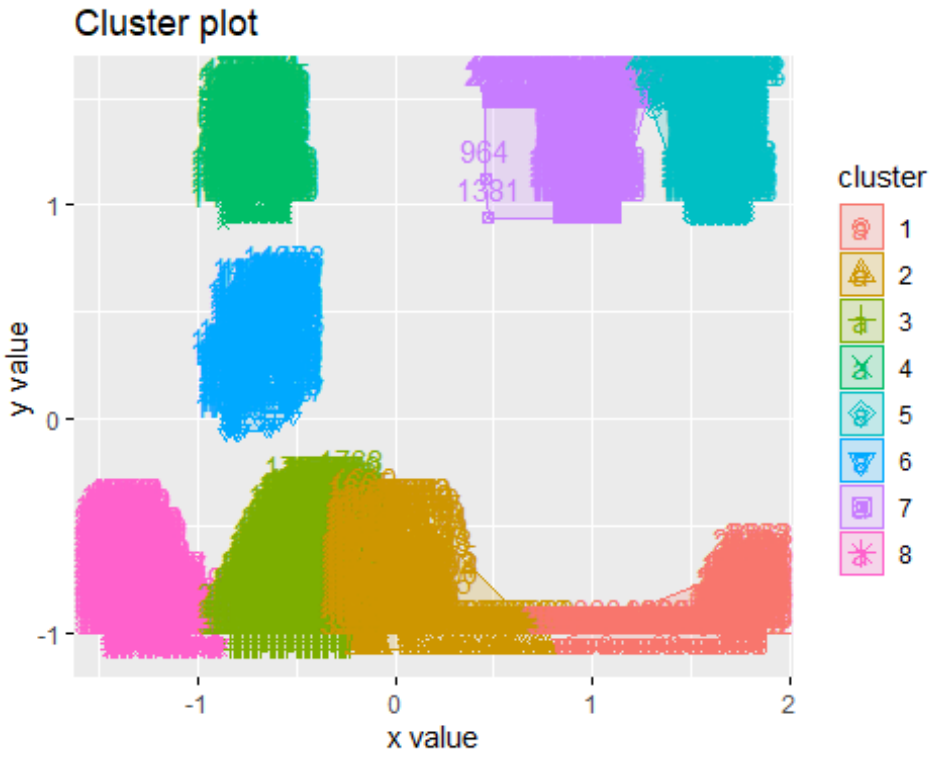
Cluster k=6



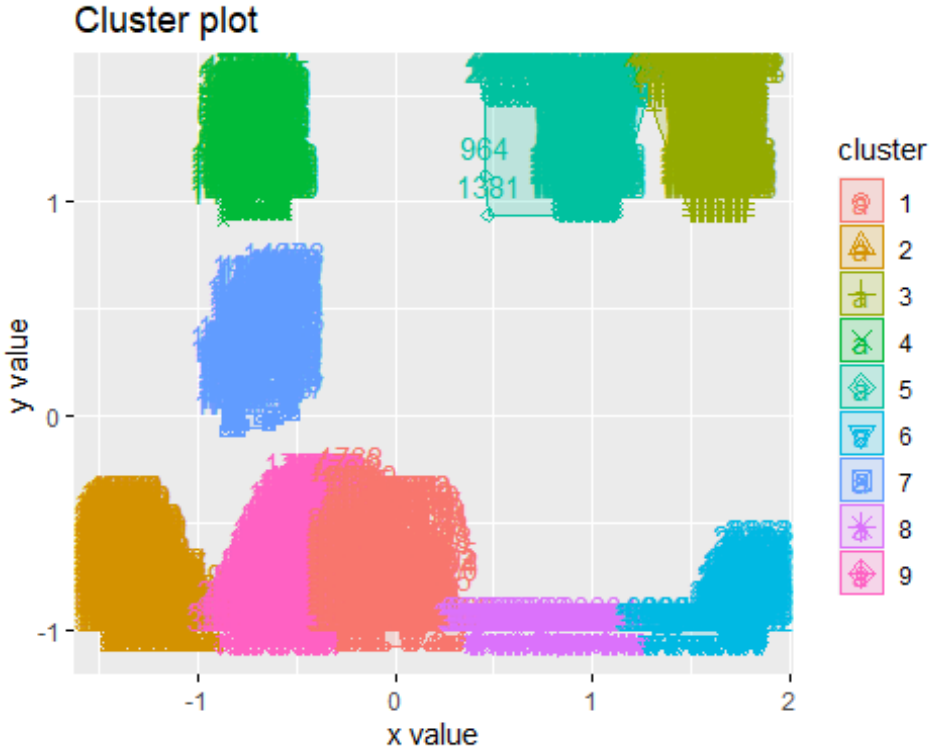
Cluster k=7



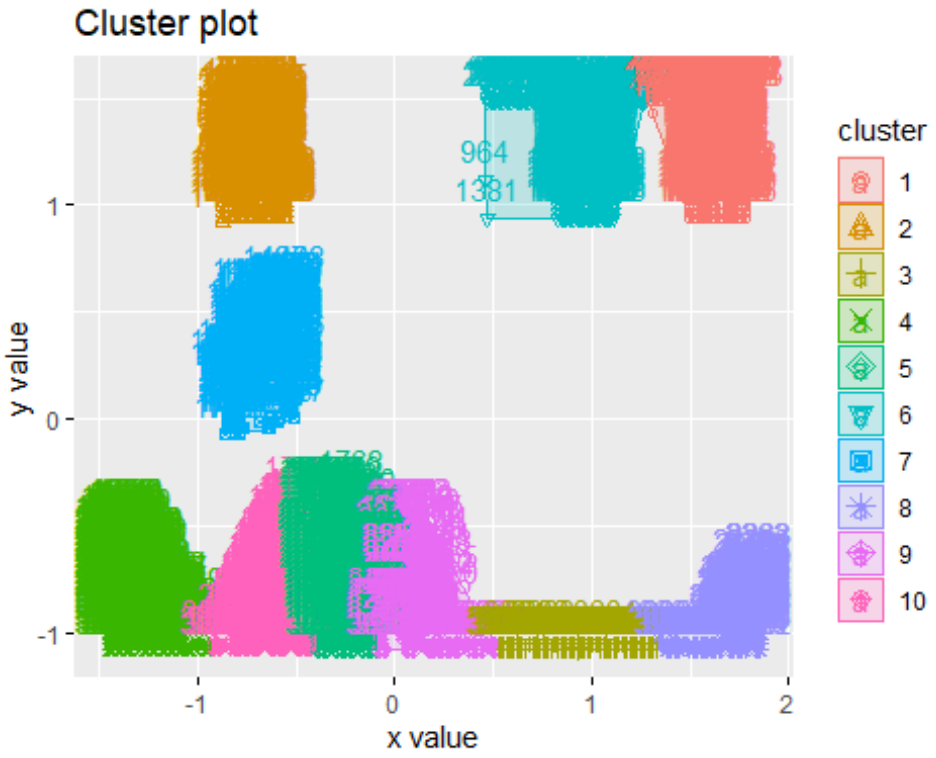
Cluster k=8



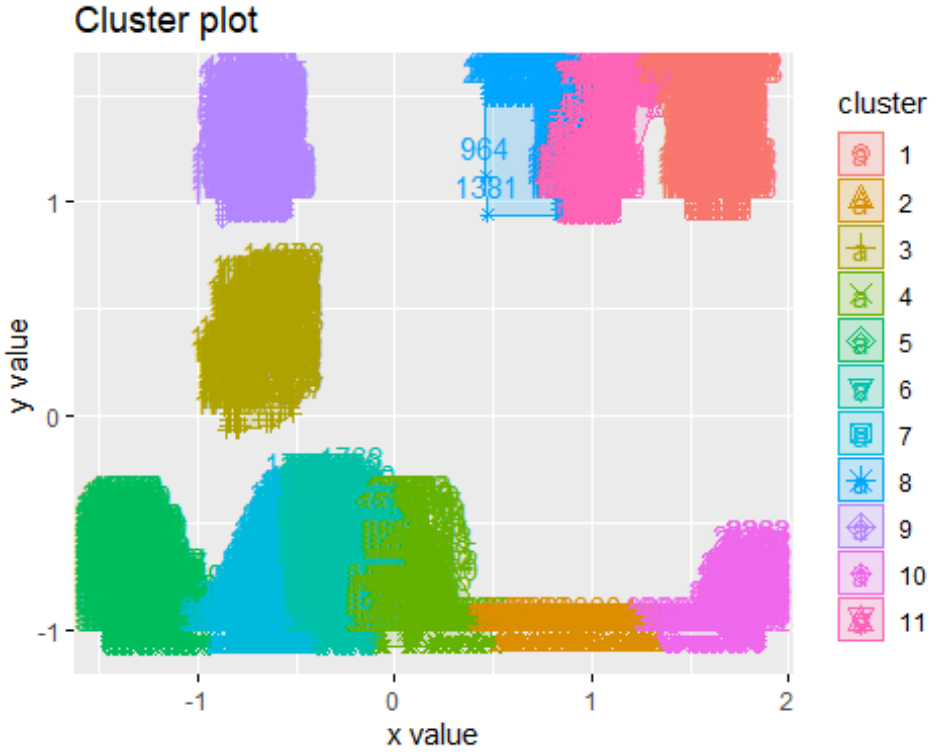
Cluster k=9



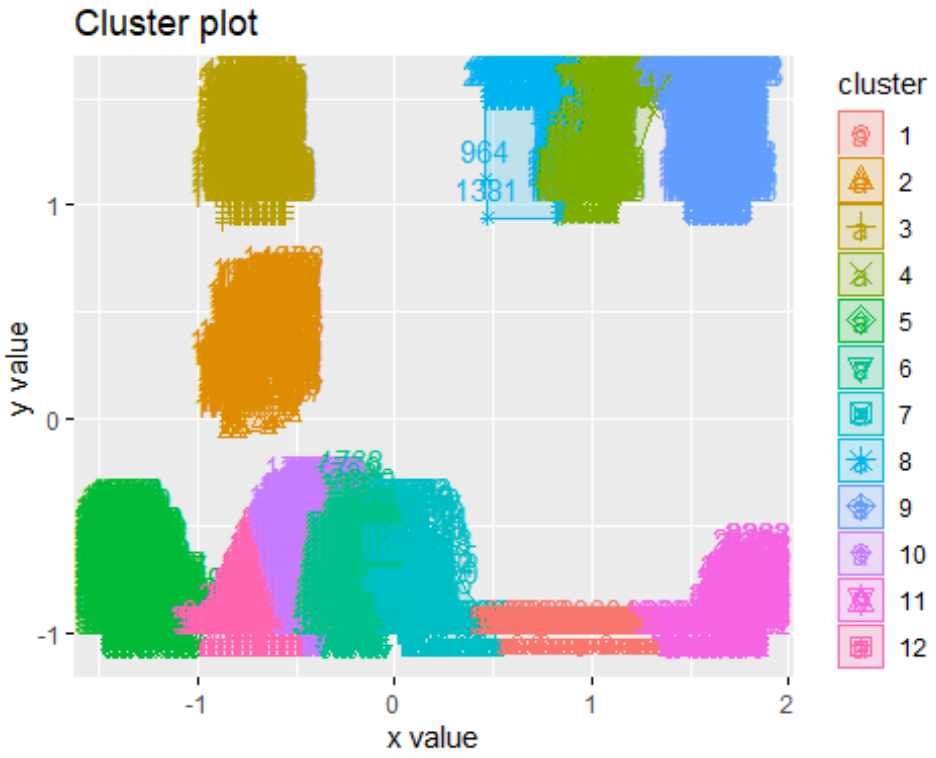
Cluster k=10



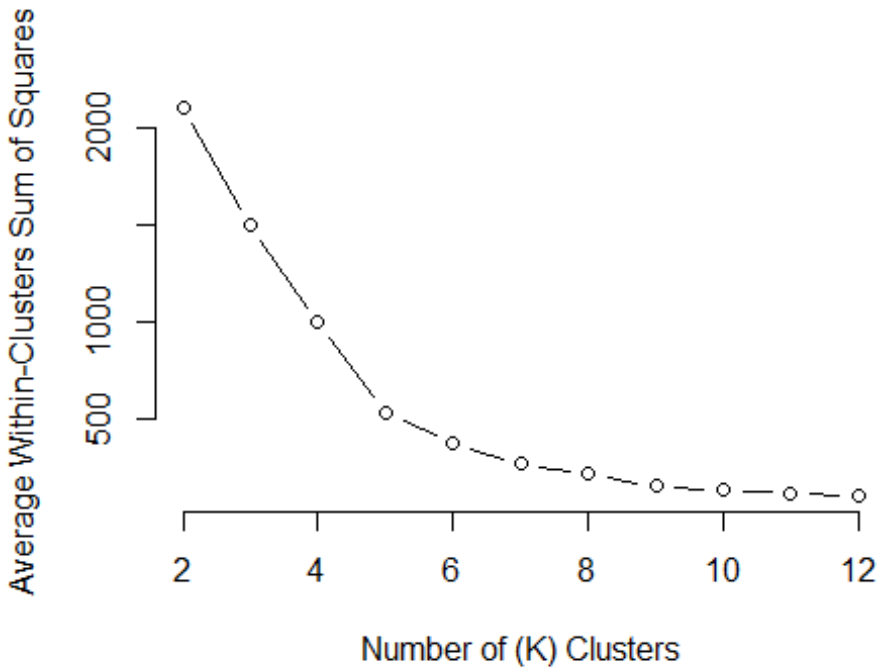
Cluster k=11



Cluster k=12



e. 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.



> f. 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?

Based on the line graph above, the elbow point appears to be k=5.