Week 12 Assignment

Myranda Swartzwelter

2/24/2022

Understanding Regression Algorithms

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 emails 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.

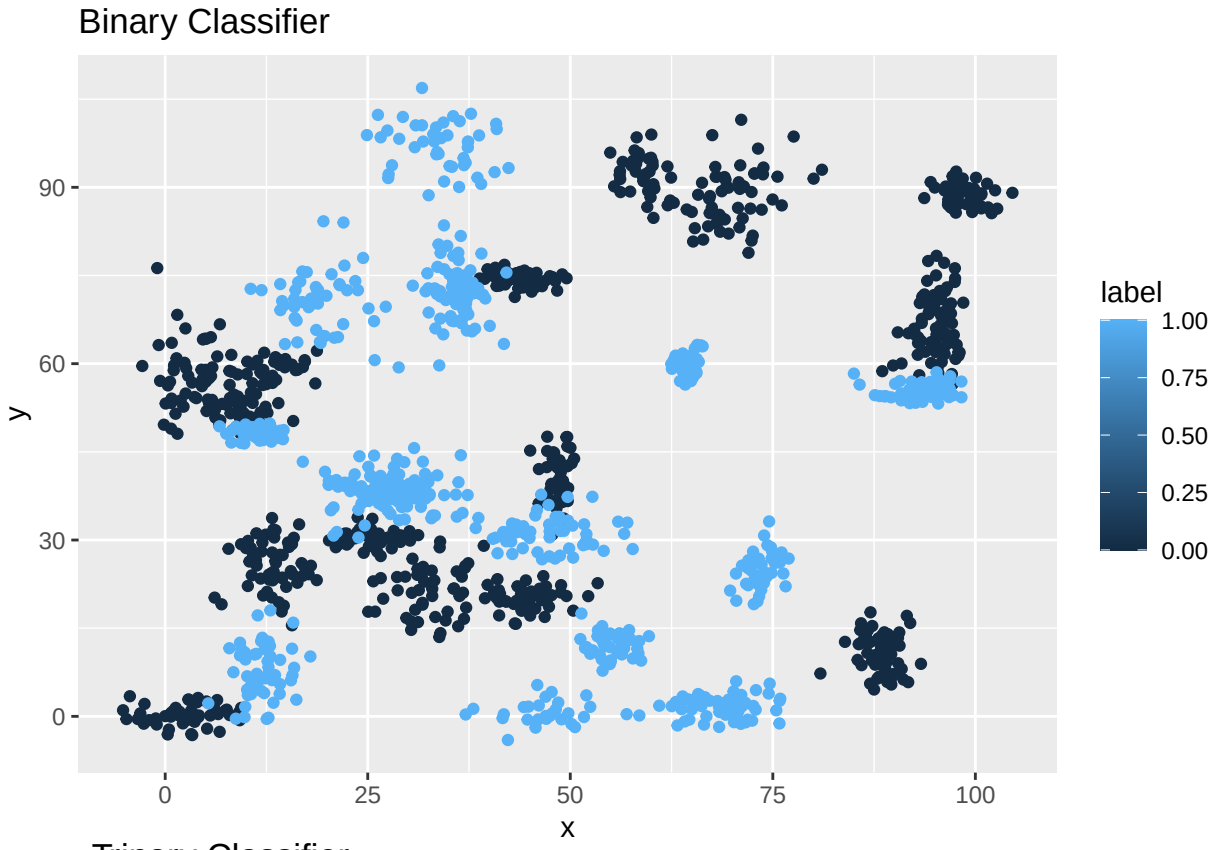
Setting up the problem

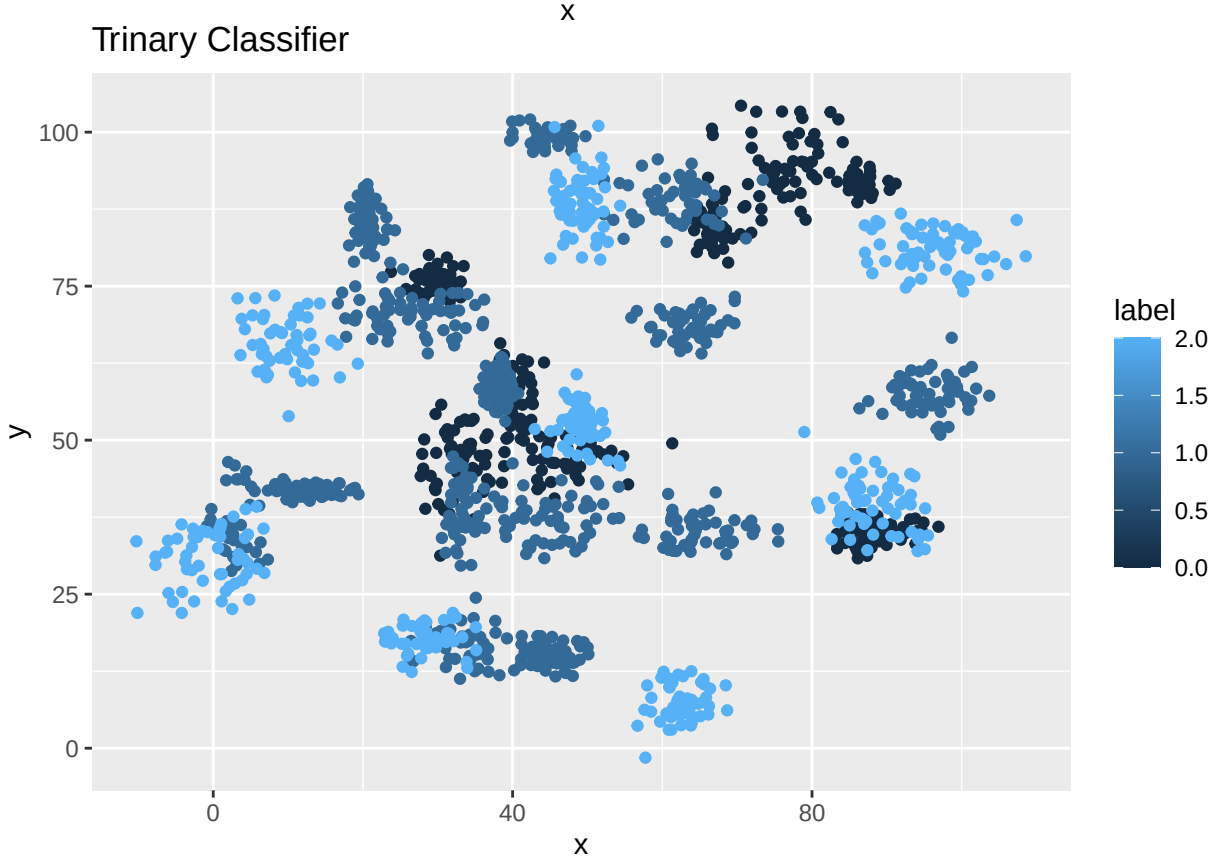
In this problem, you will use the nearest neighbors algorithm to fit a model on two simplified datasets. The first dataset (found in binary-classifier-data.csv) contains three variables; label, x, and y. The label variable is either 0 or 1 and is the output we want to predict using the x and y variables (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.

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.

Plot the Data from each dataset using a scatter plot

You can also embed plots, for example:


Binary Classifier
90
60
30
0
y
0
25
50
75
100
x
label
1.00
0.75
0.50
0.25
0.00


Trinary Classifier
100
75
50
25
0
y
0
40
80
x
label
2.0
1.5
1.0
0.5
0.0

K Nearest Neighbor

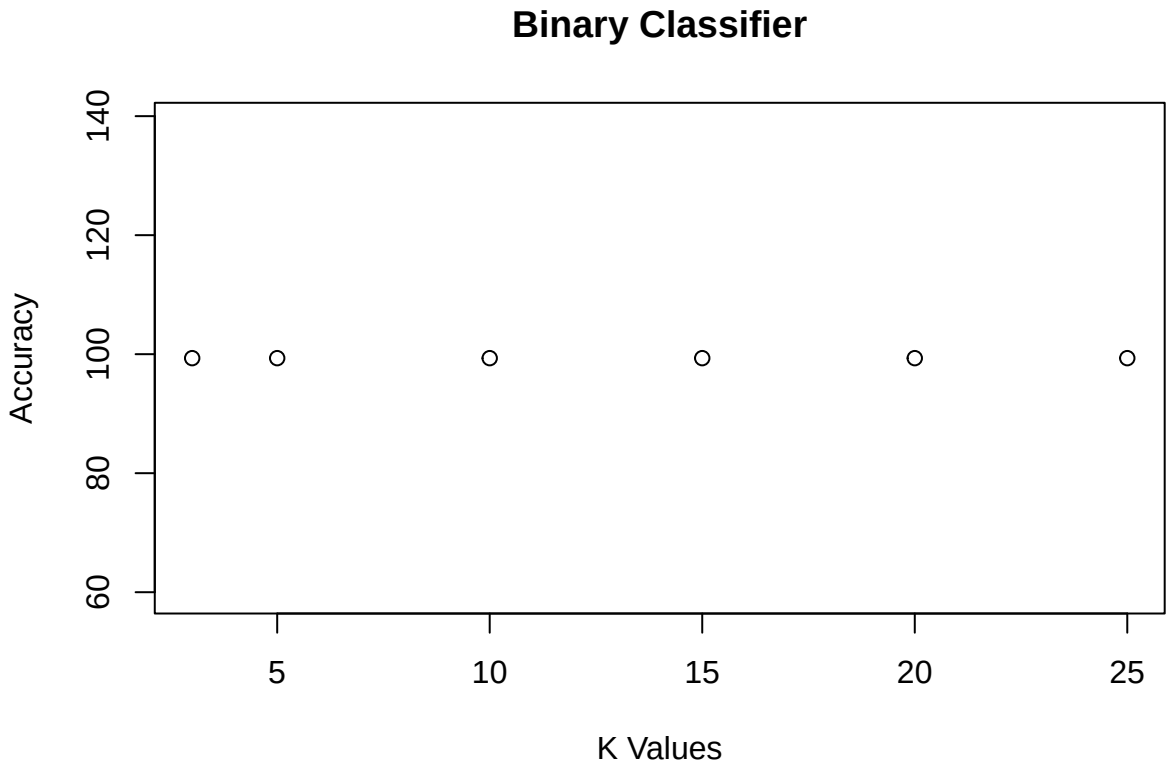
The k nearest neighbors algorithm categorizes an input value by looking at the labels for the k nearest points and assigning a category based on the most common label. In this problem, you will determine which points are nearest by calculating the Euclidean distance between two points. As a refresher,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

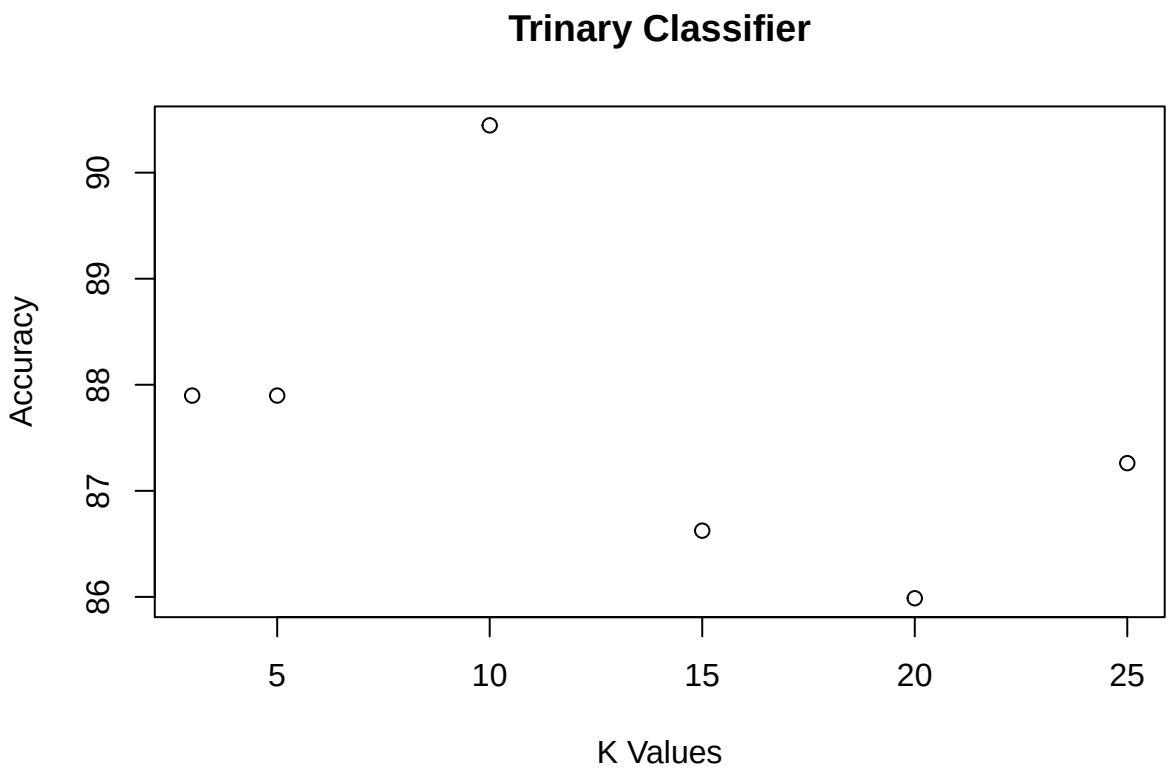
$$p_1 = (x_1, y_1)$$
$$p_2 = (x_2, y_2)$$
$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

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.

Fitting the Datasets using k Nearest Neighbor

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.





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

No I don't think a linear classifier would work well on these datasets, since the groups overlap and aren't easily spearated by a straight line.

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

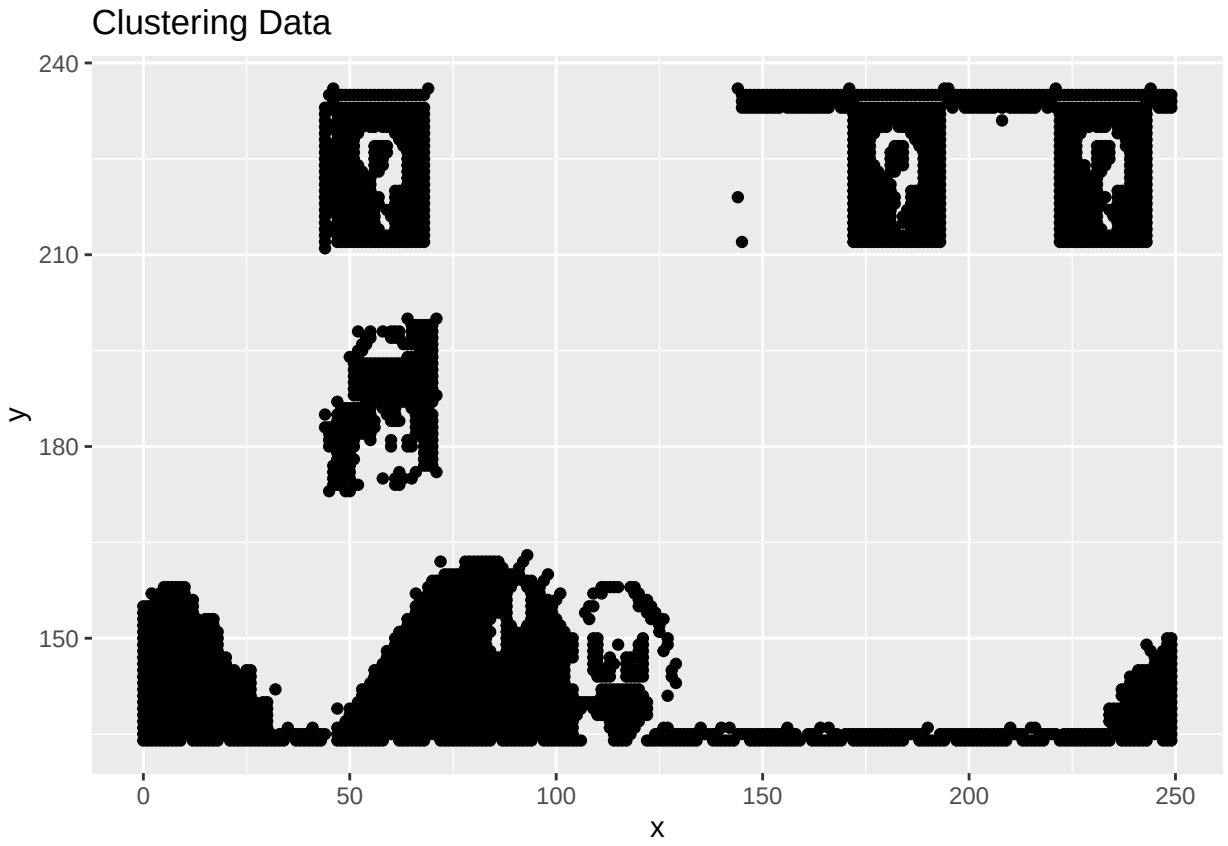
The K Nearest Neighbor is more accurate compared to the logistic regression classifier from last week. This makes sense because although a logisitic regression is likely better than a linear regression, the groups are not easily defined by a boundary line between the classes.

Clustering

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.

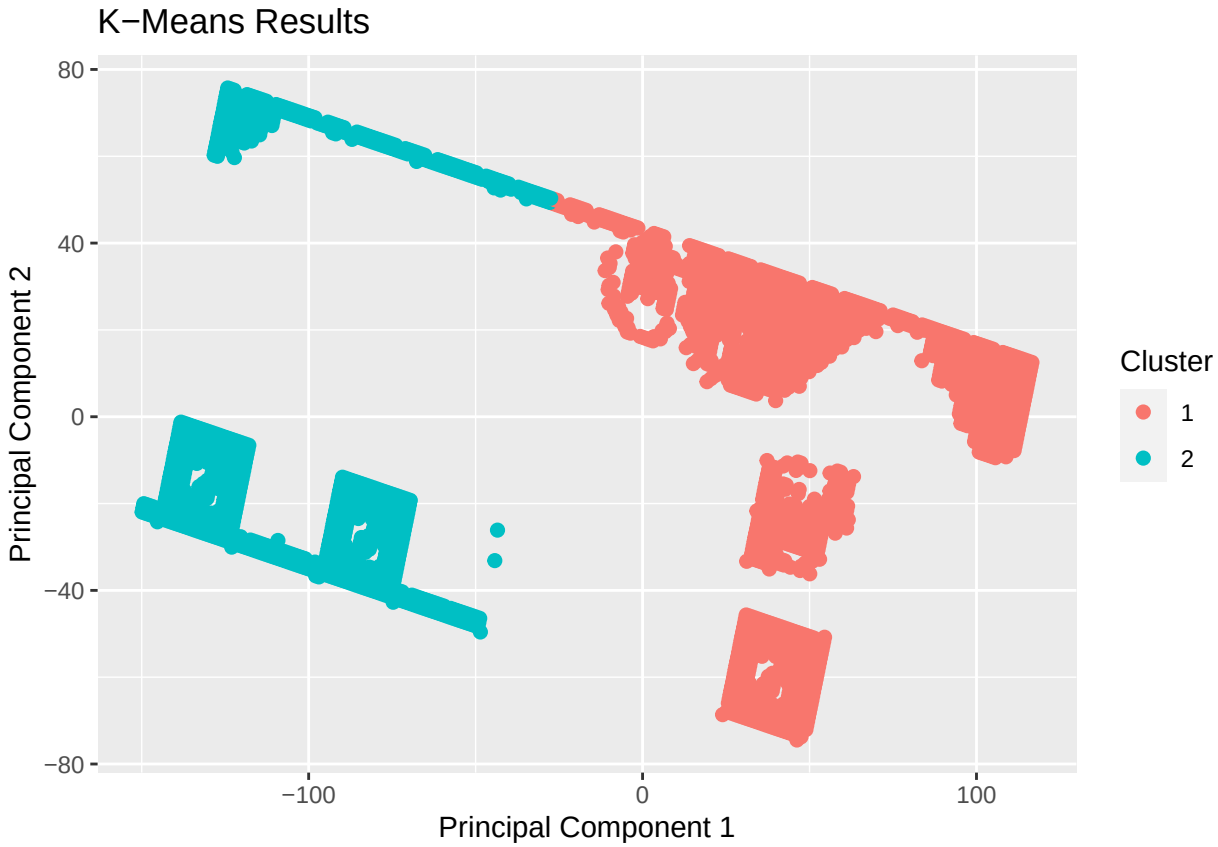
Plot the Data

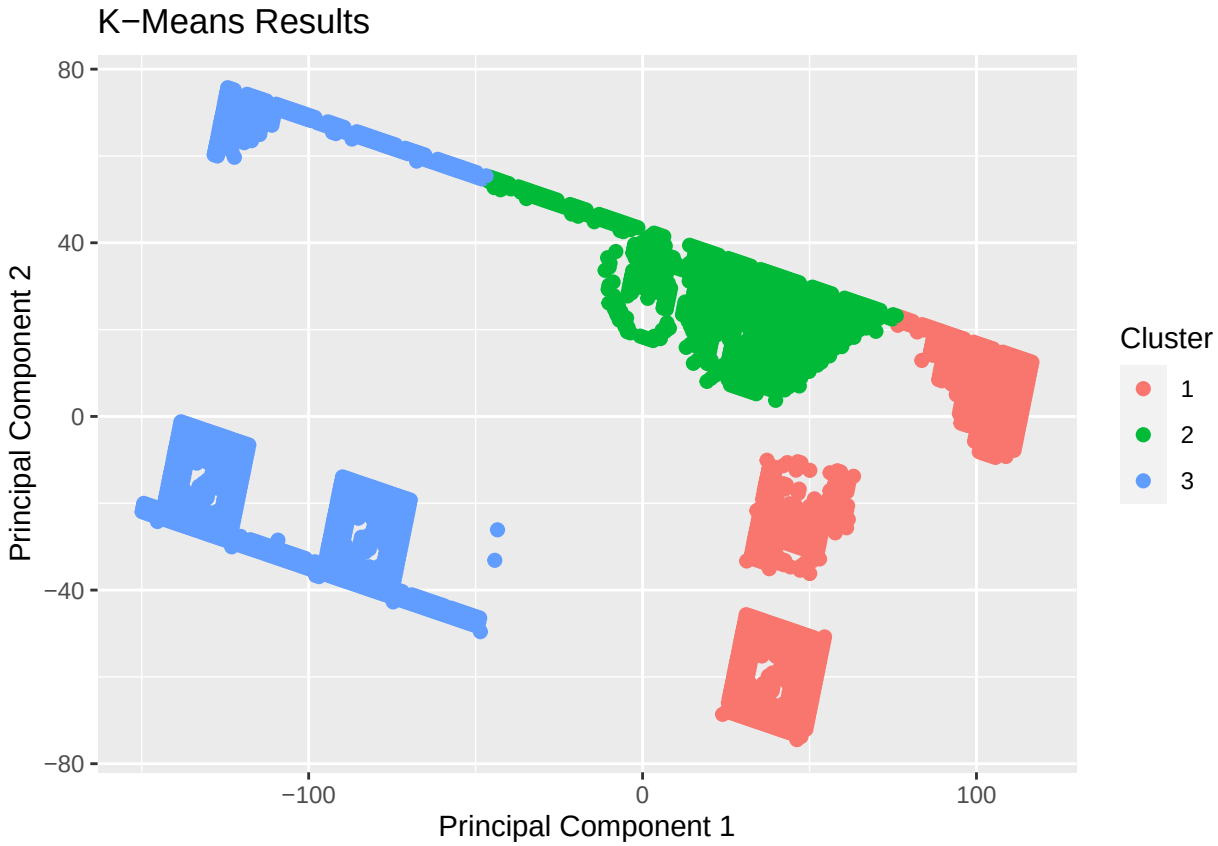


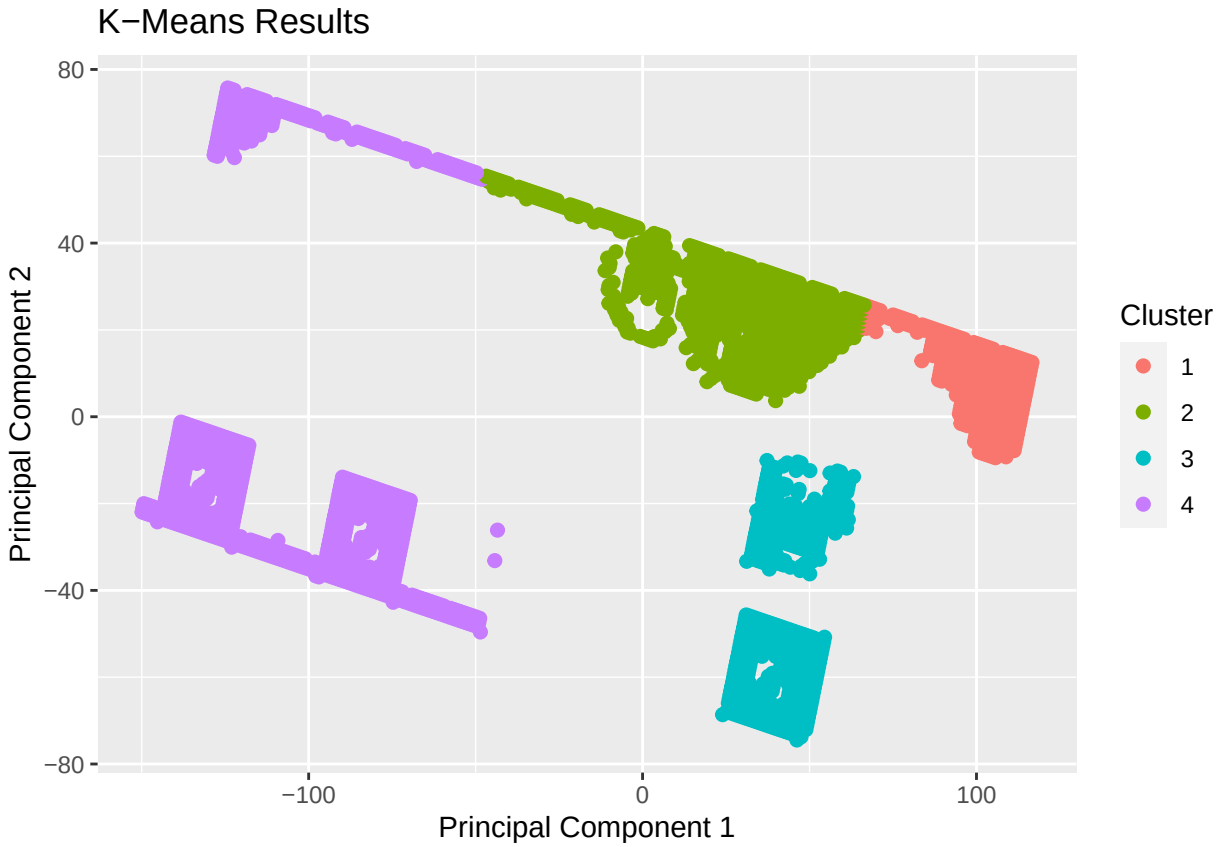
Fit the dataset using the k-means algorithm

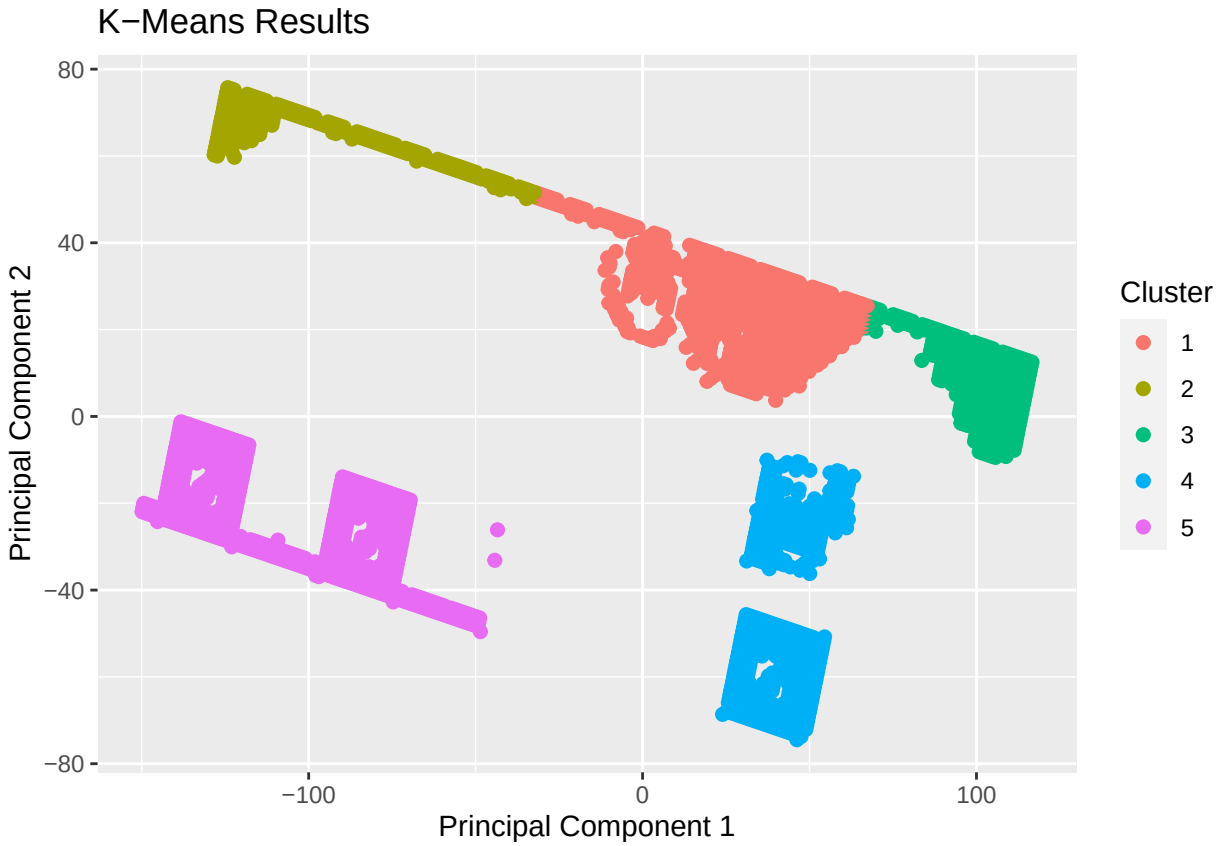
Use k values from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

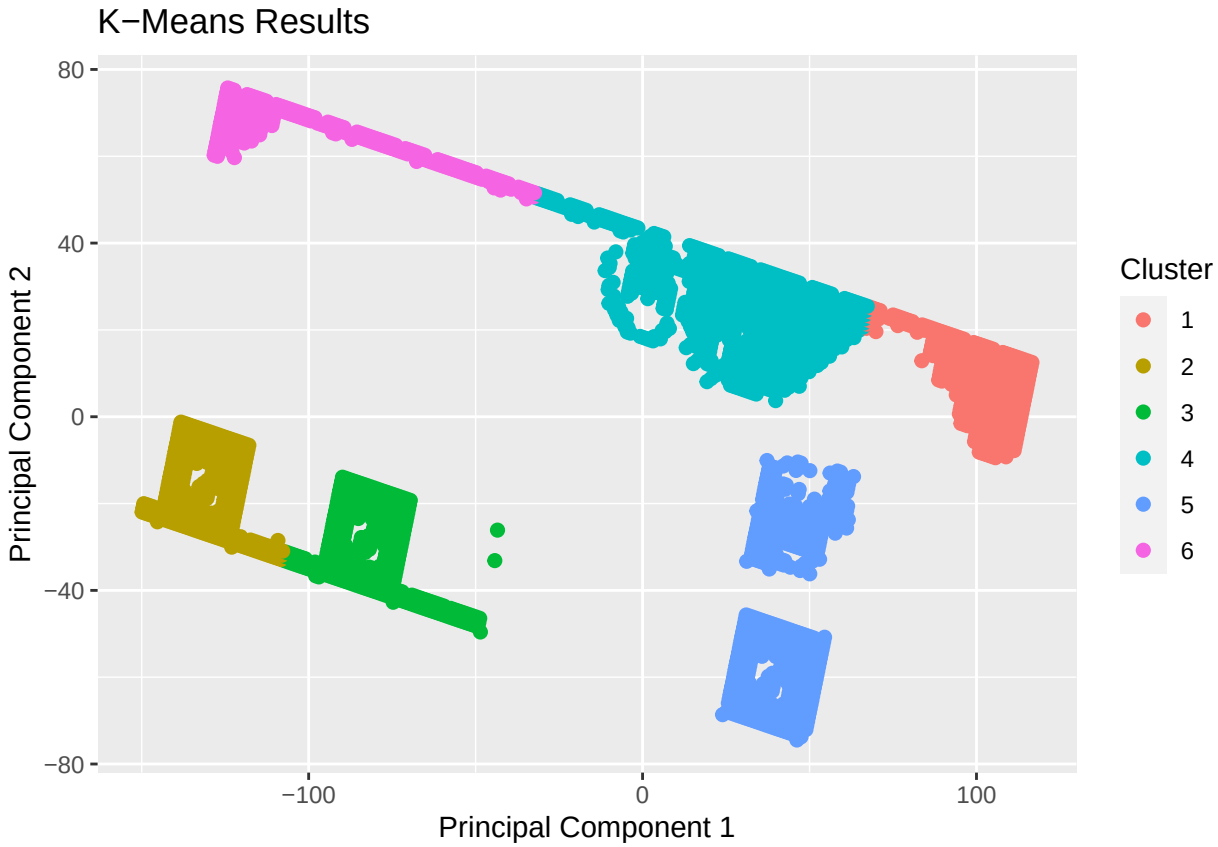
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

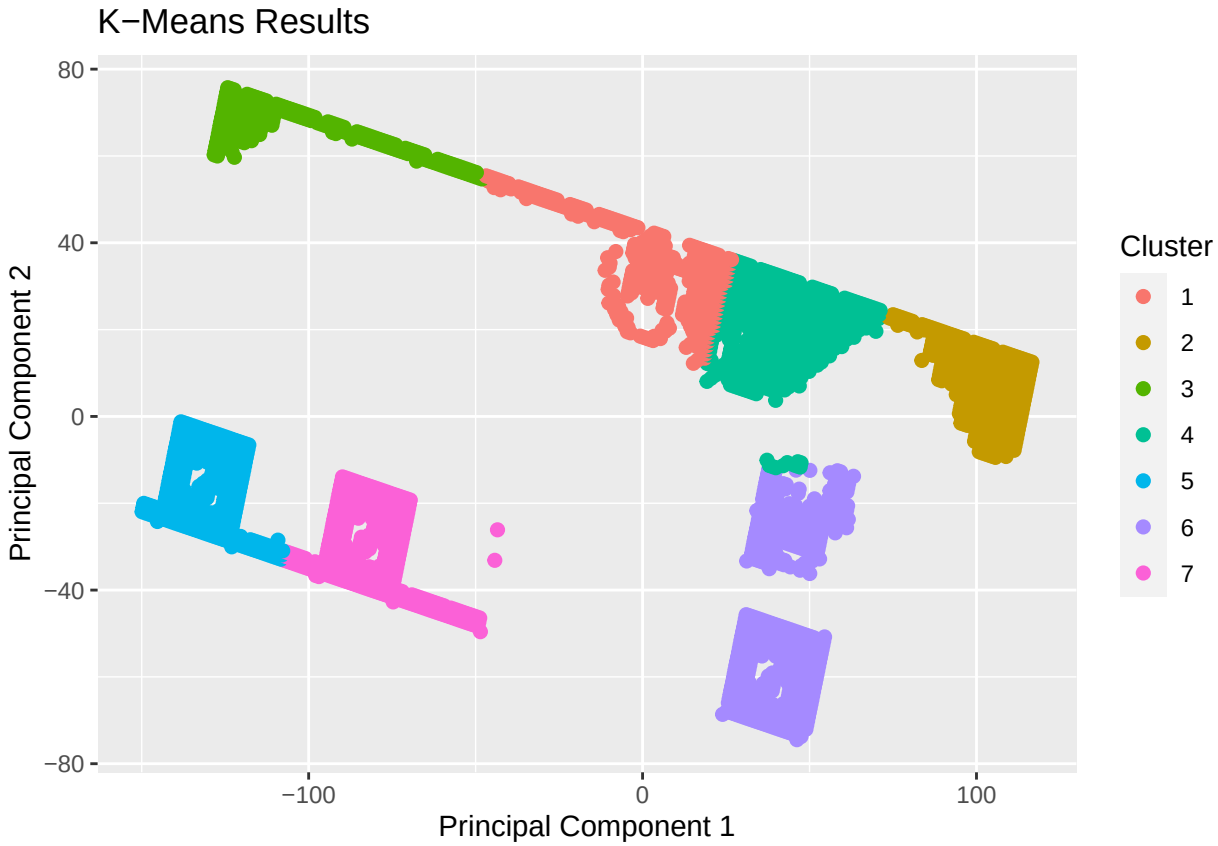

K−Means Results
80
40
0
−40
−80
Principal Component 2
−100
0
100
Principal Component 1
Cluster
1
2

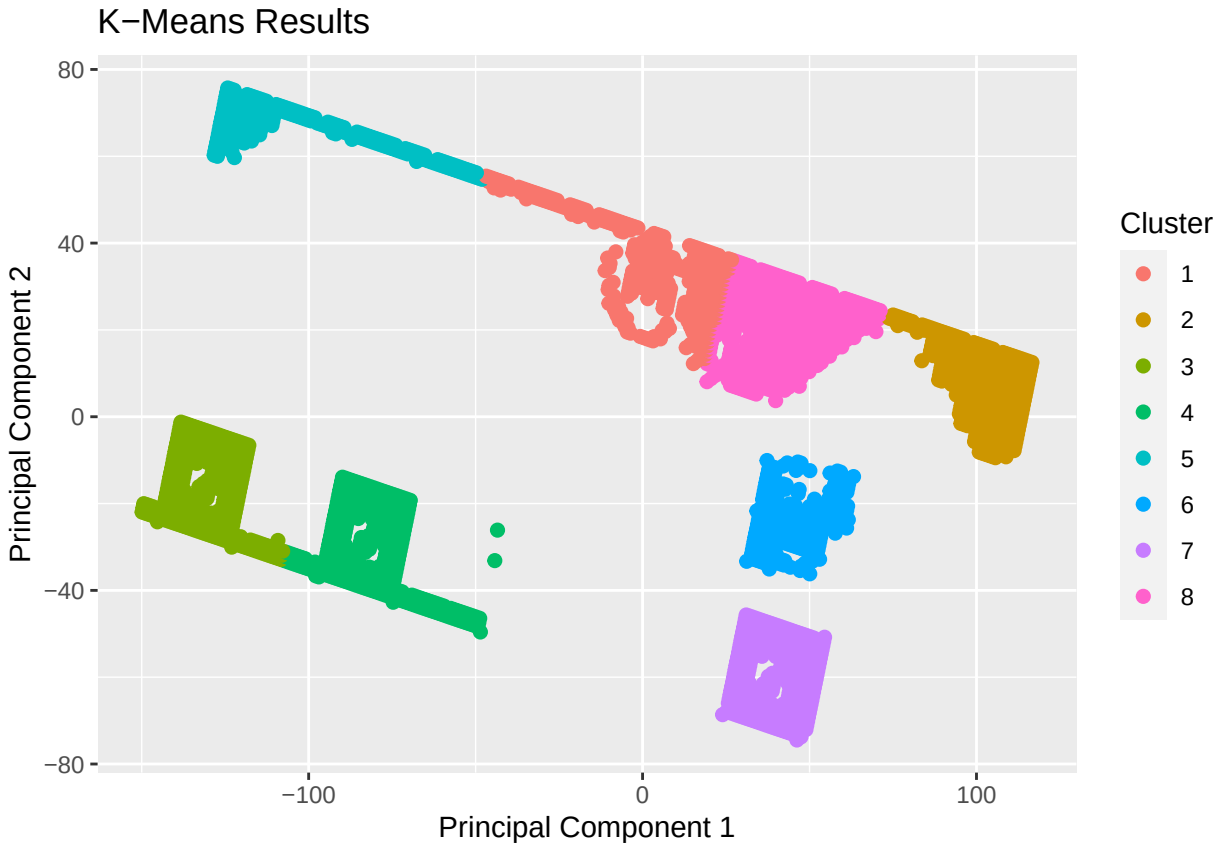

K−Means Results
80
40
0
−40
−80
Principal Component 2
−100
0
100
Principal Component 1
Cluster
1
2
3

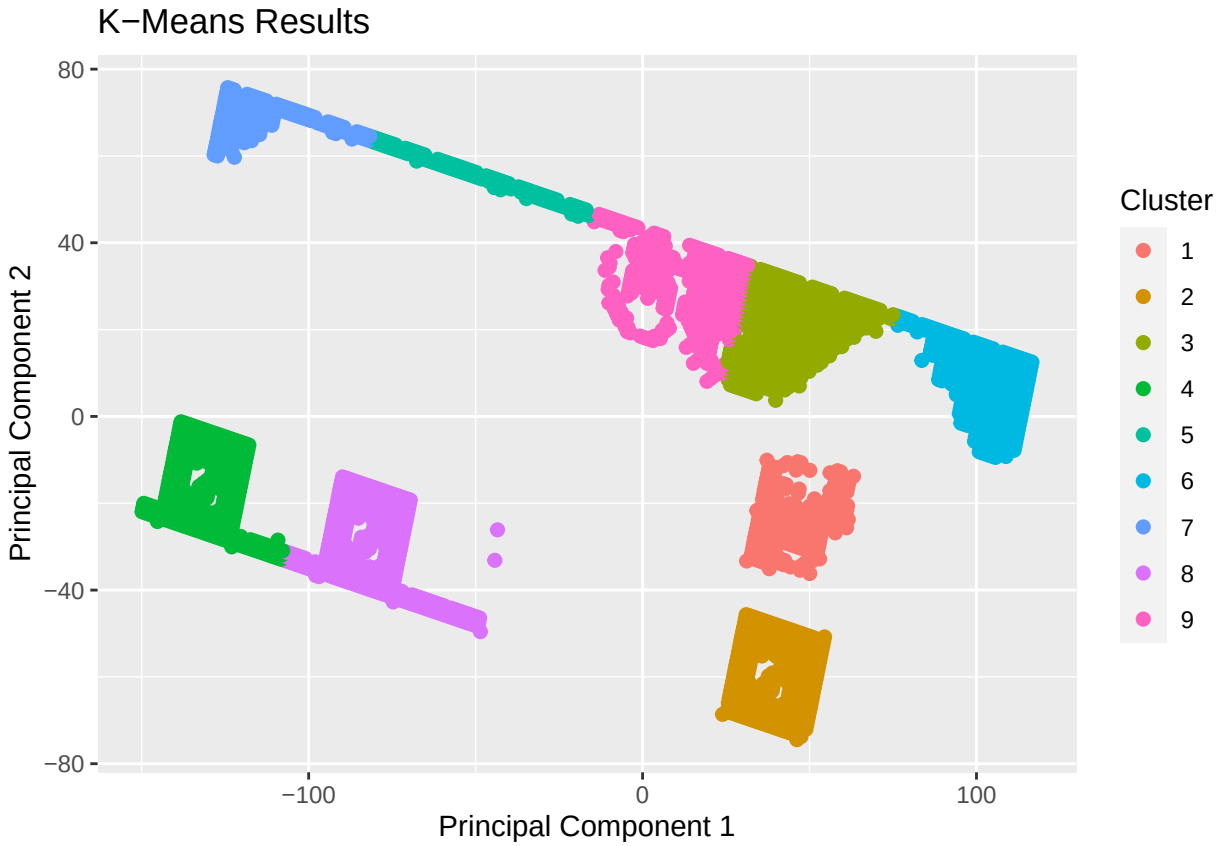

K−Means Results
80
40
0
−40
−80
Principal Component 2
−100
0
100
Principal Component 1
Cluster
1
2
3
4

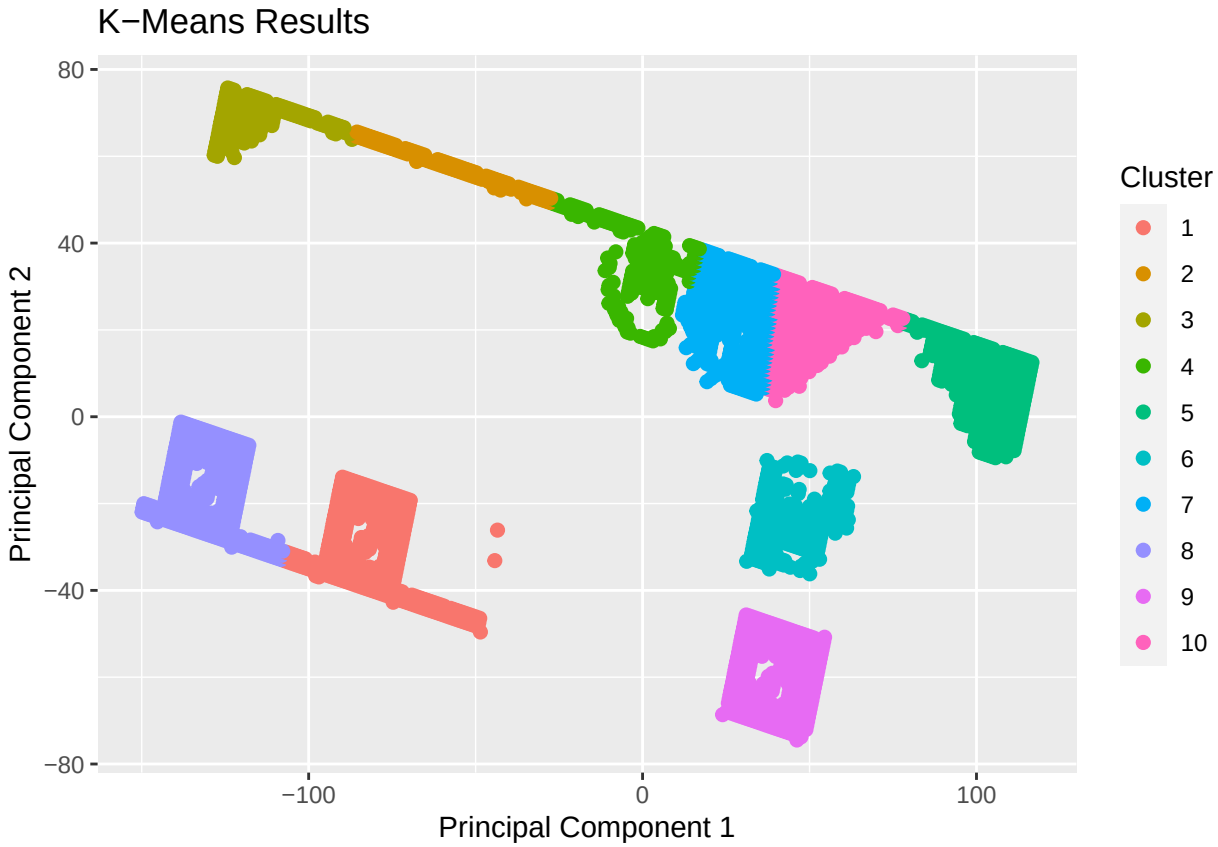

K−Means Results
80
40
0
−40
−80
Principal Component 2
−100
0
100
Principal Component 1
Cluster
1
2
3
4
5

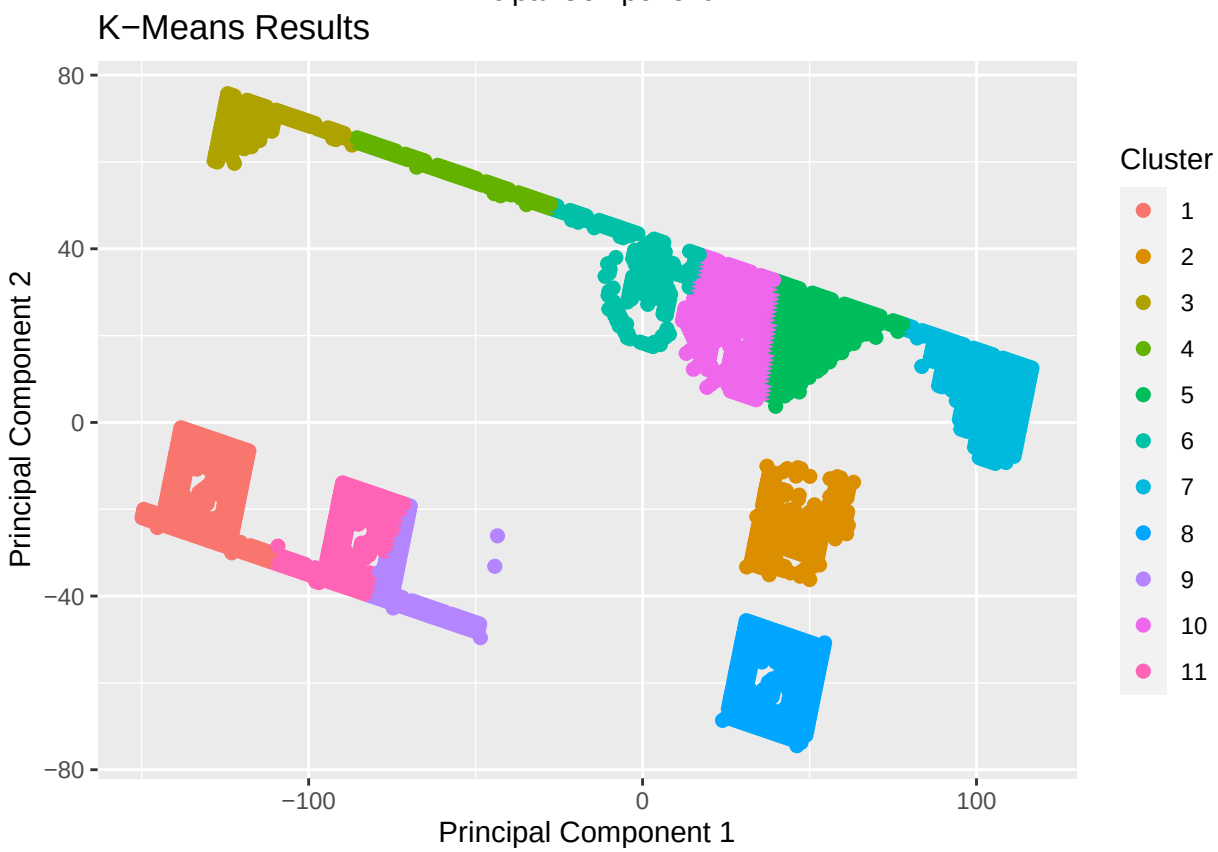

K−Means Results
80
40
0
−40
−80
Principal Component 2
−100
0
100
Principal Component 1
Cluster
1
2
3
4
5
6

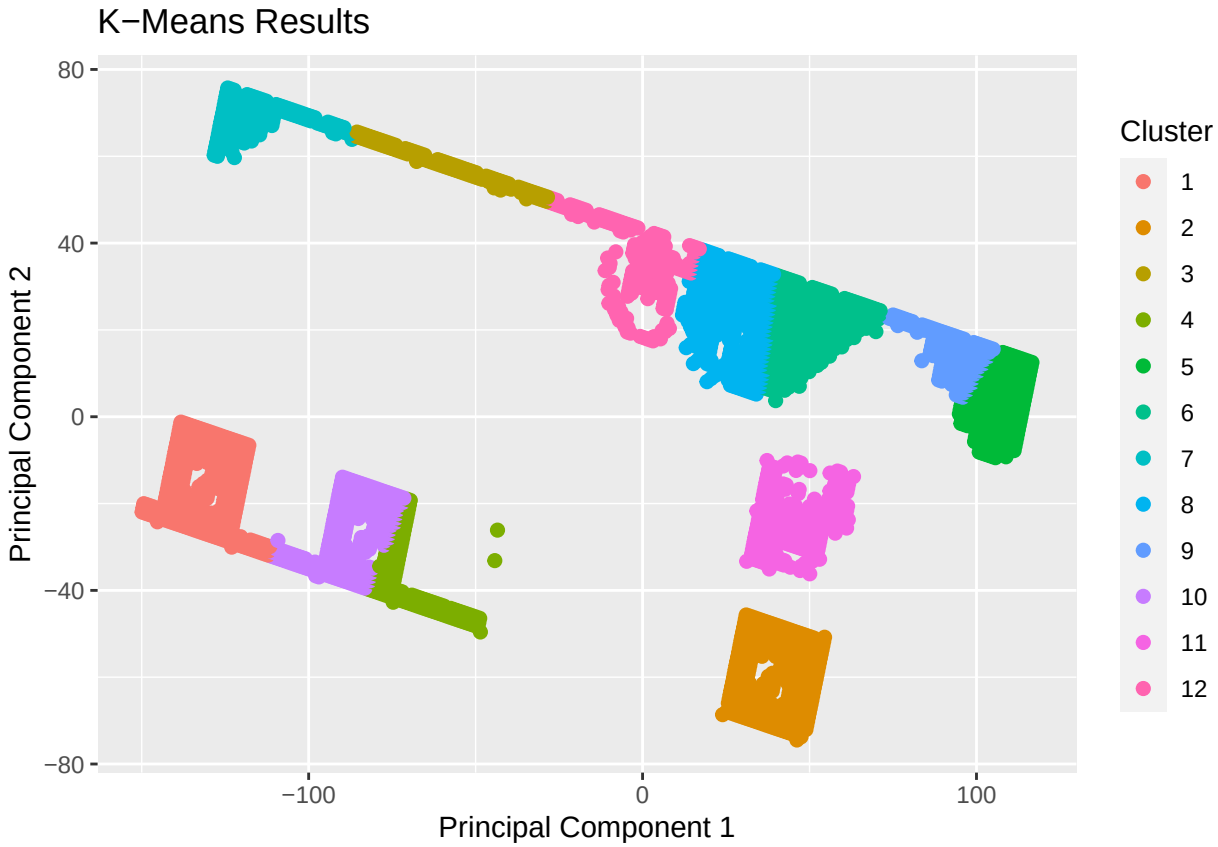

K−Means Results
80
40
0
−40
−80
Principal Component 2
−100
0
100
Principal Component 1
Cluster
1
2
3
4
5
6
7


K−Means Results
80
40
0
−40
−80
−100
0
100
Principal Component 1
Principal Component 2
Cluster
1
2
3
4
5
6
7
8


K−Means Results
80
40
0
−40
−80
−100
0
100
Principal Component 1
Principal Component 2
Cluster
1
2
3
4
5
6
7
8
9

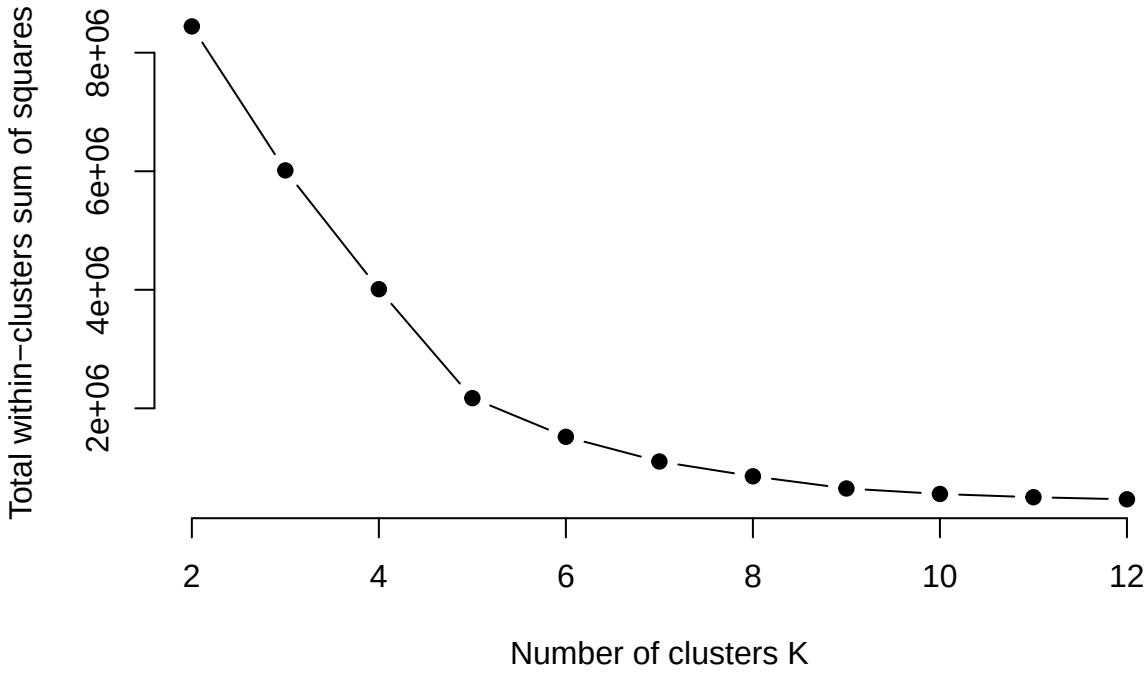

K−Means Results
Principal Component 2
Principal Component 1
80
40
0
−40
−80
−100
0
100
Cluster
1
2
3
4
5
6
7
8
9
10


K−Means Results
Principal Component 2
Principal Component 1
80
40
0
−40
−80
−100
0
100
Cluster
1
2
3
4
5
6
7
8
9
10
11



Within-cluster sum of squares

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.



The "Right" Number of Clusters

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?

Looking at the graph, we can see the elbow point is at 5 clusters for this data set, so we would say the k = 5 is the "right" number of clusters for this data set.