Weeks 11 & 12 Exercises

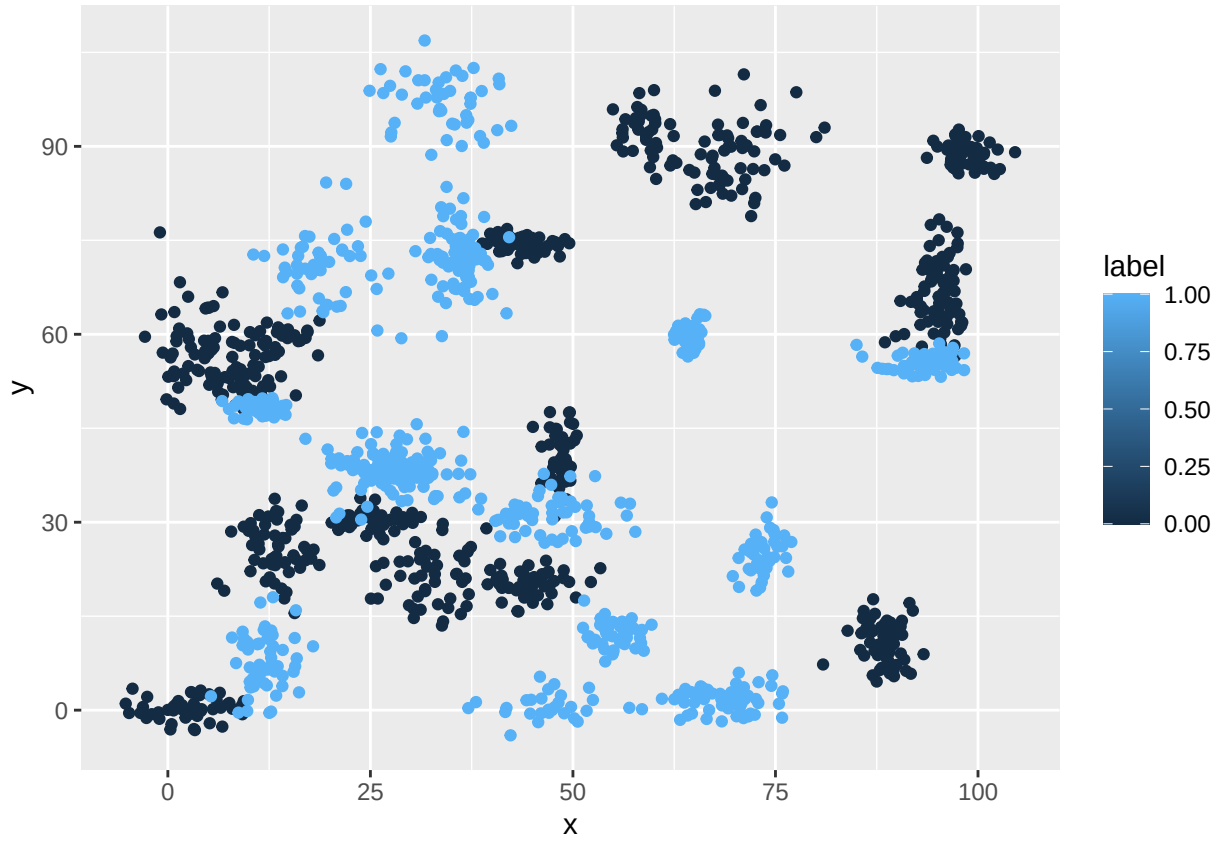
Stewart Wilson

2022-06-04

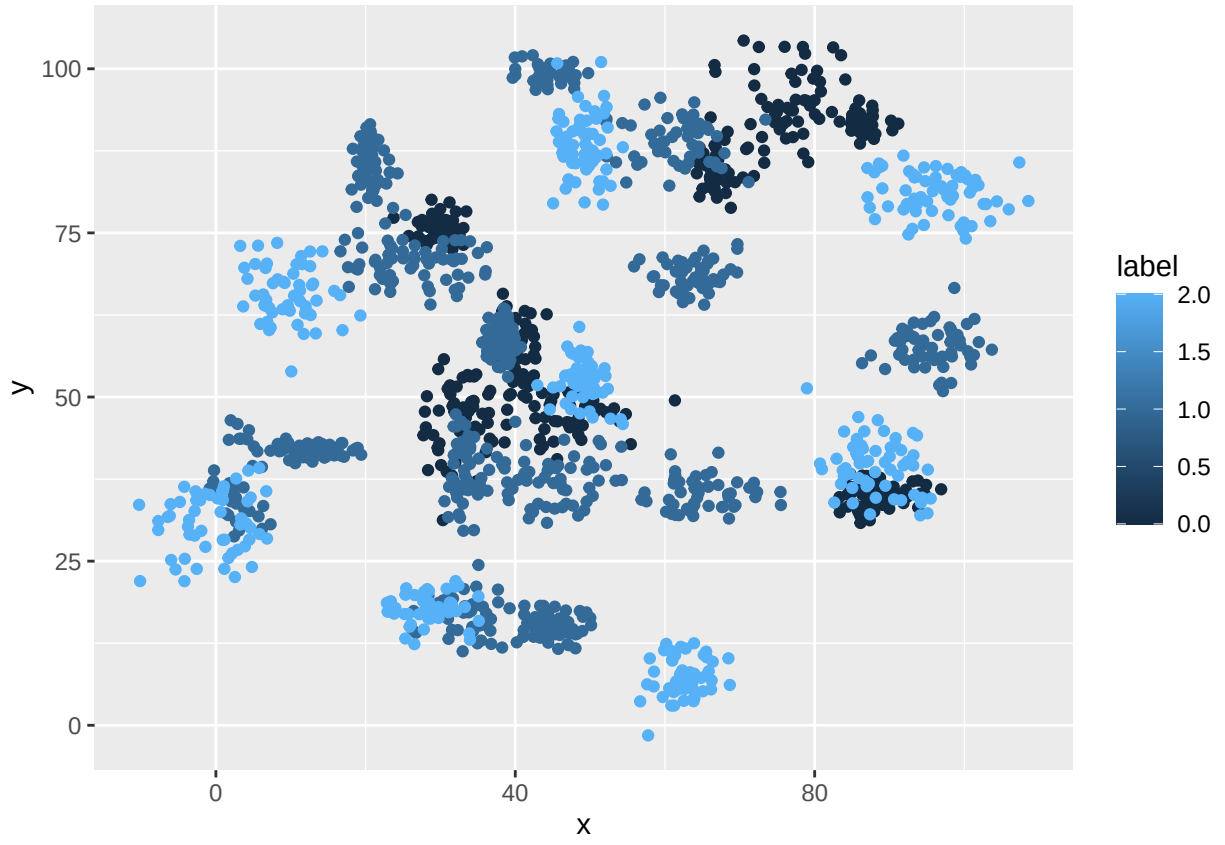
Introduction to Machine Learning

Scatter Plot of Data

```
binary <- read.csv("C:/Users/Stewart/Documents/GitHub/dsc520/data/binary-classifier-data.csv")
trinary <- read.csv("C:/Users/Stewart/Documents/GitHub/dsc520/data/trinary-classifier-data.csv")
# scatterplot of binary and trinary data
binary_plot <- ggplot(binary, aes(x=x, y=y, colour=label)) + geom_point()
trinary_plot <- ggplot(trinary, aes(x=x, y=y, colour=label)) + geom_point()
binary_plot
```



```
trinary_plot
```

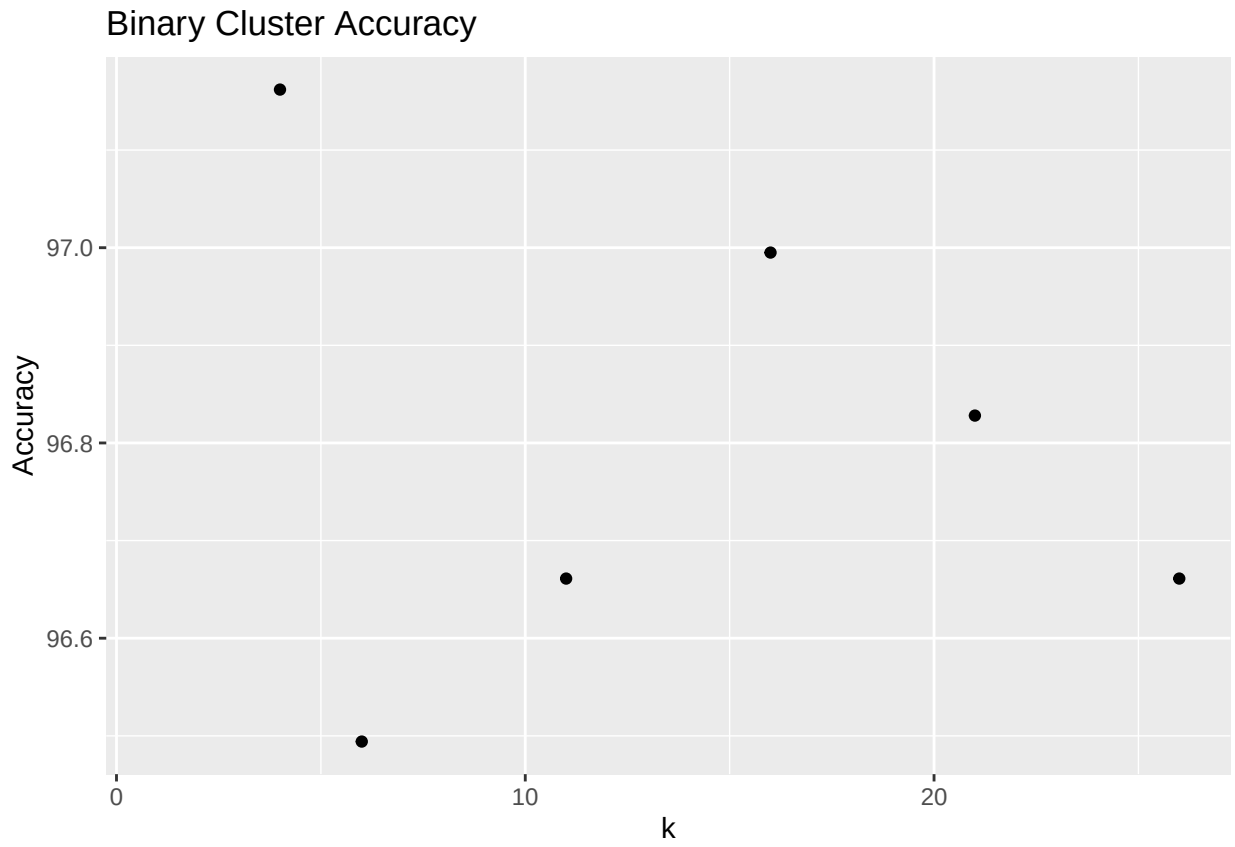


Training Binary Data Sets

```
# split binary into test and training set
train_index <- createDataPartition(binary$label, p=.6)$Resample1
training_binary <- binary[train_index, ]
test_binary <- binary[-train_index, ]
# matrix for num of clusters
kmatrix <- c(3, 5, 10, 15, 20, 25)
# will collect accuracy count for clusters
accBin <- c()
# runs nearest neighbor for every given k in kmatrix
# calculates accuracy for each loops
index <- 0
for(i in kmatrix){
  index = i + 1
  kModBin <- knn(train=training_binary, test=test_binary, cl=training_binary$label, k=i)
  accBin[index] <- 100*sum(test_binary$label == kModBin)/NROW(test_binary$label)
}
# plotting accuracy vs k
accuracy_df <- data.frame(accBin)
accuracy_df$k <- seq.int(nrow(accuracy_df))
```

```
accuracy_plot <- ggplot(data = accuracy_df, aes(x = k, y = accBin)) + geom_point() + xlab("k") + ylab("
accuracy_plot
```

```
## Warning: Removed 20 rows containing missing values (geom_point).
```



Training Trinary Data Sets

```
## Warning: Removed 20 rows containing missing values (geom_point).
```

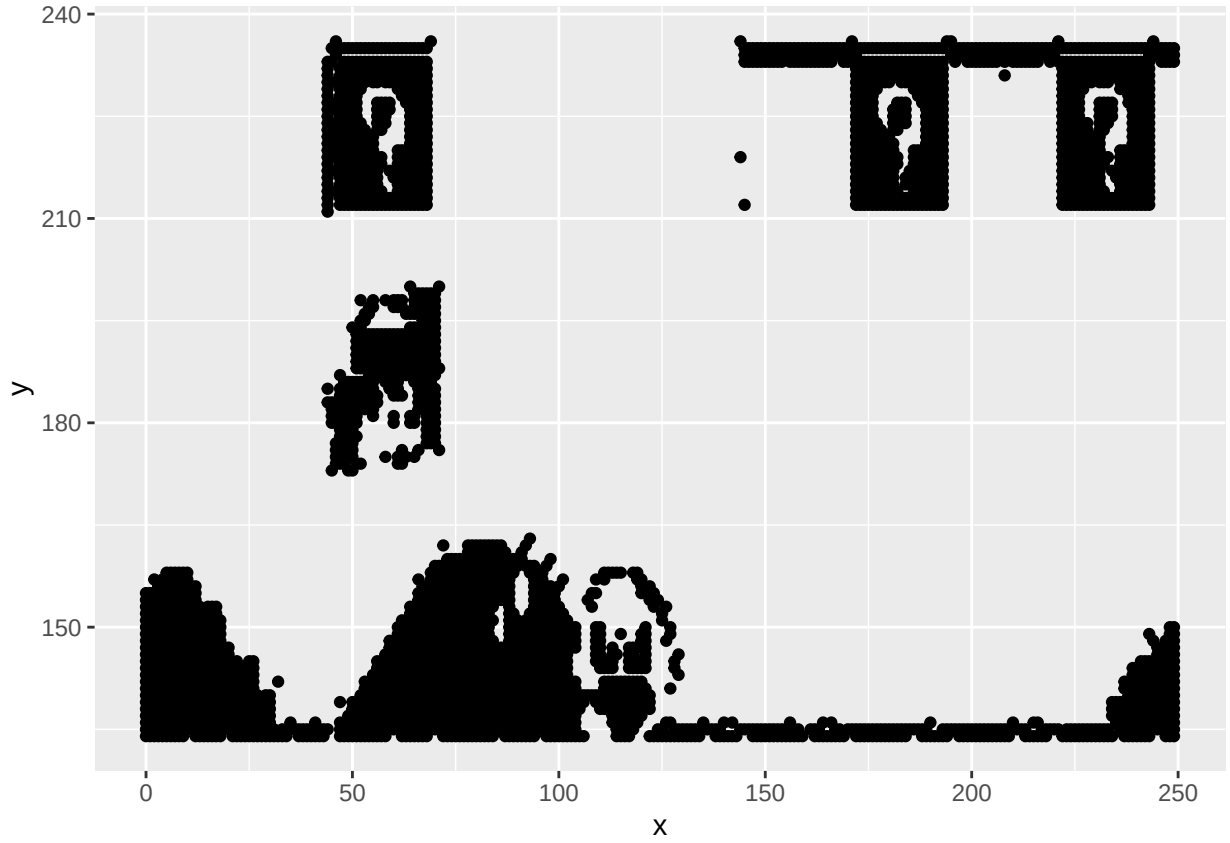
Linear Model?

Looking back at the original graphs of the data, I do not think a linear classifier would work well on these datasets. There is no clear straight line that could split either dataset nicely so the model would most likely not be very accurate.

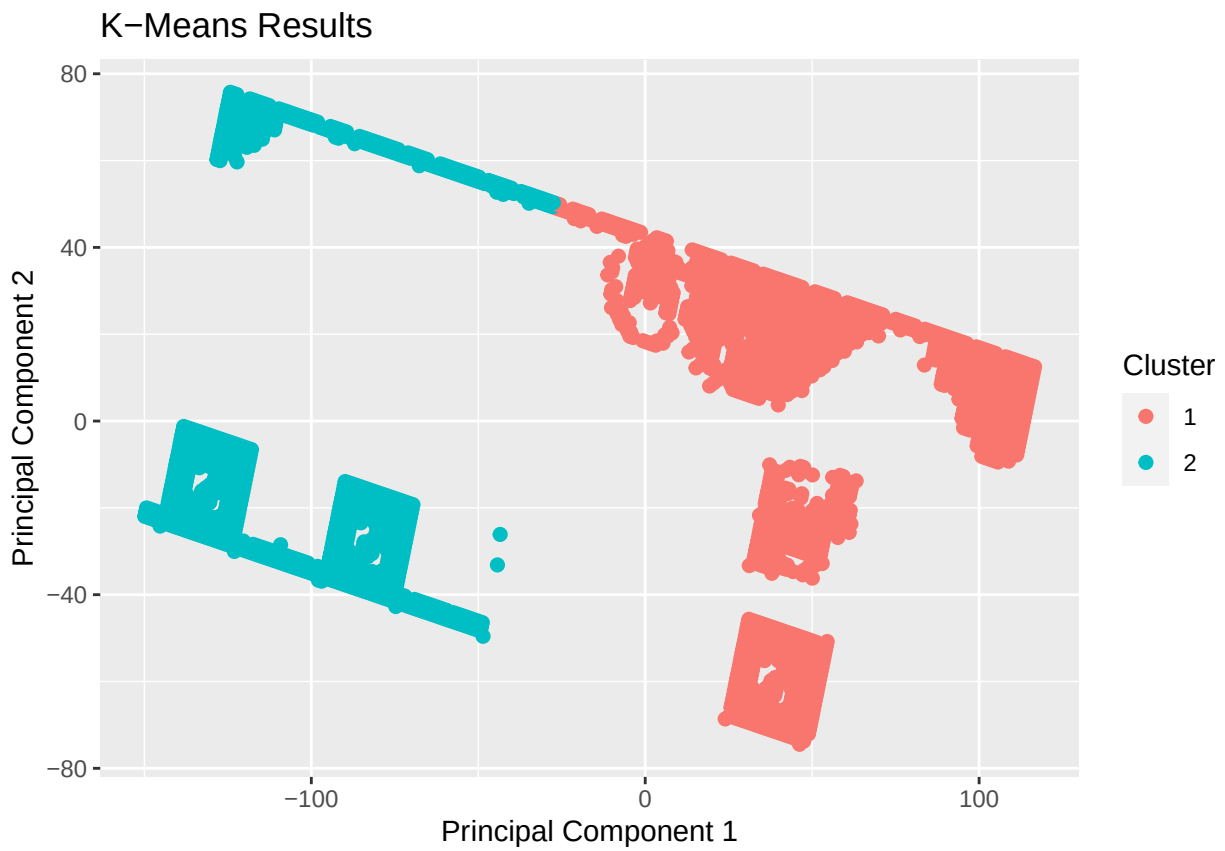
Looking back at the accuracy from last week's exercises (58%), it is clear that clustering vastly improved the accuracy of the model.

Clustering

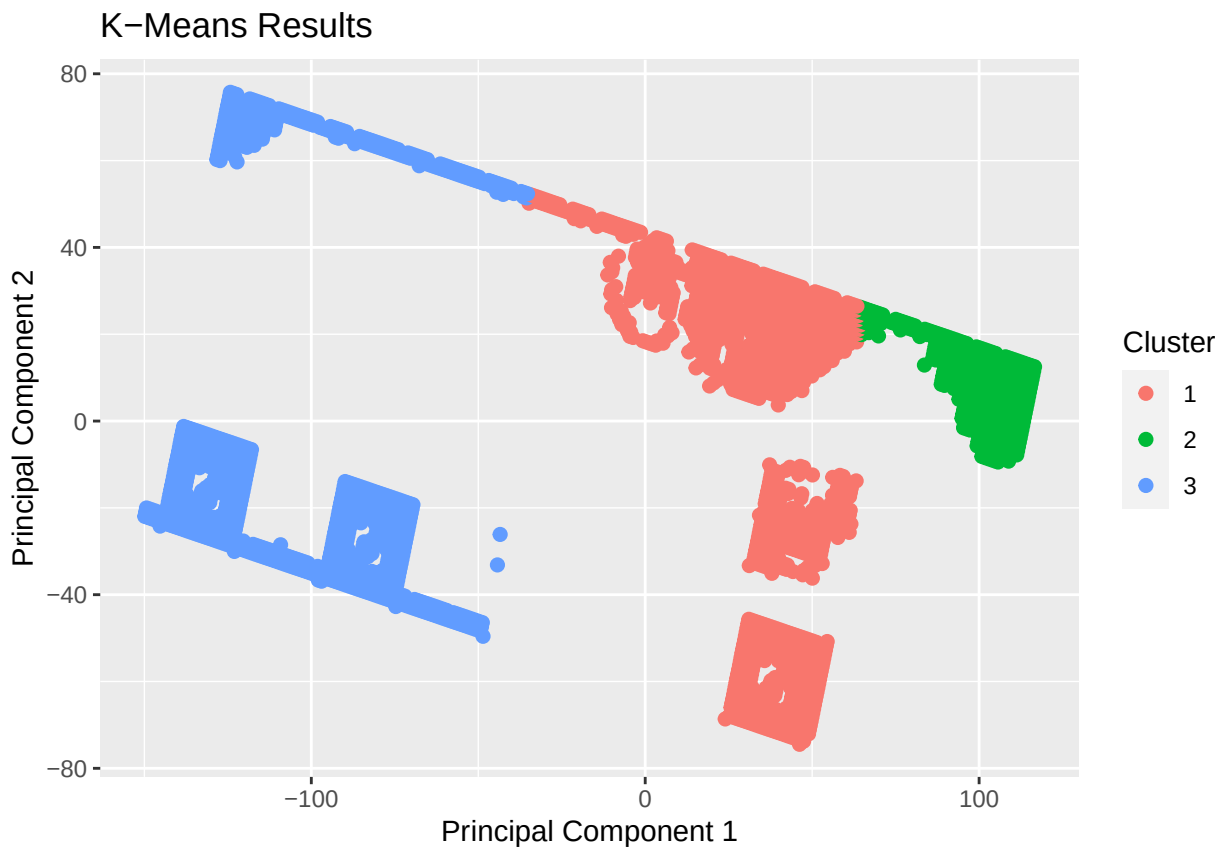
```
clustering <- read.csv("C:/Users/Stewart/Documents/GitHub/dsc520/data/clustering-data.csv")
# scatterplot of the data
clust_scatter <- ggplot(data=clustering, aes(x=x, y=y)) + geom_point()
clust_scatter
```



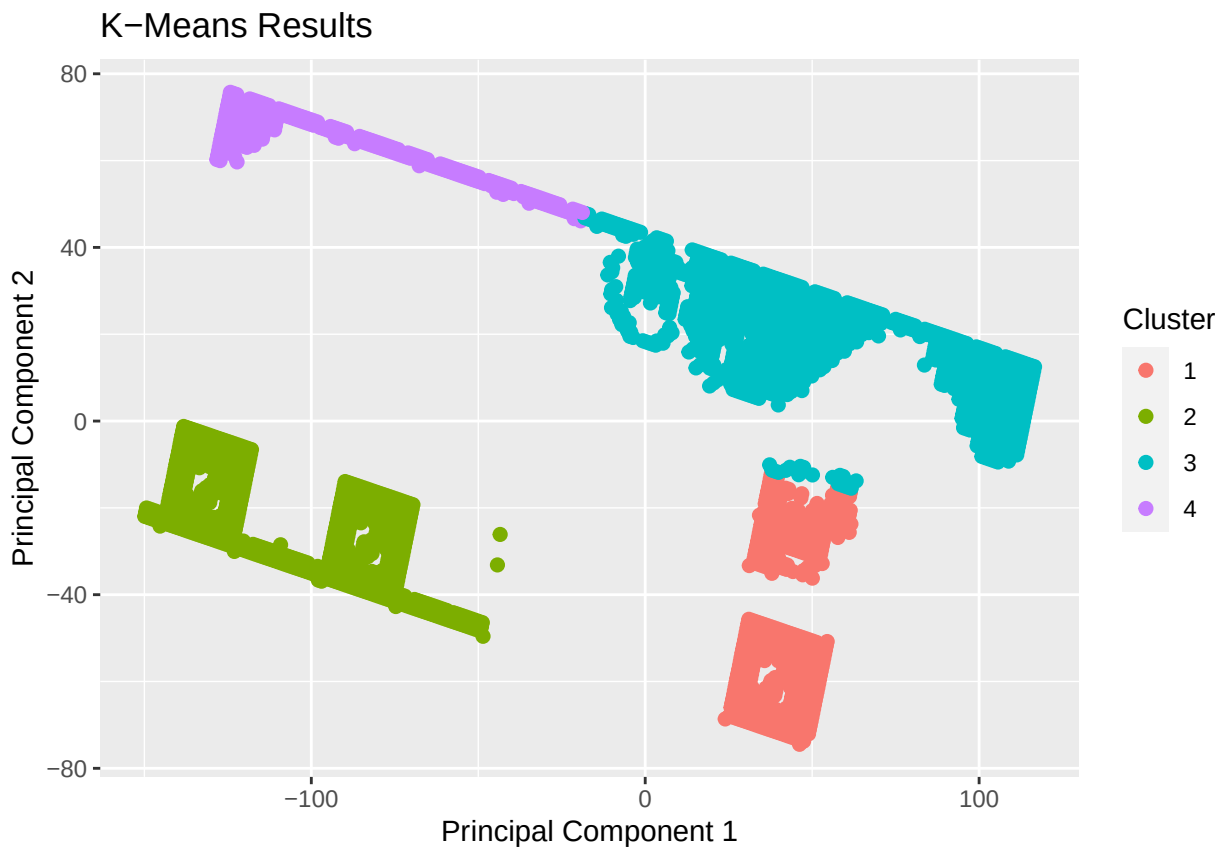

```
# set seeds and plots each cluster result for every k between 2 and 12
set.seed(278613)
km2 <- kmeans(x=clustering, centers = 2)
plot(km2, data=clustering)
```



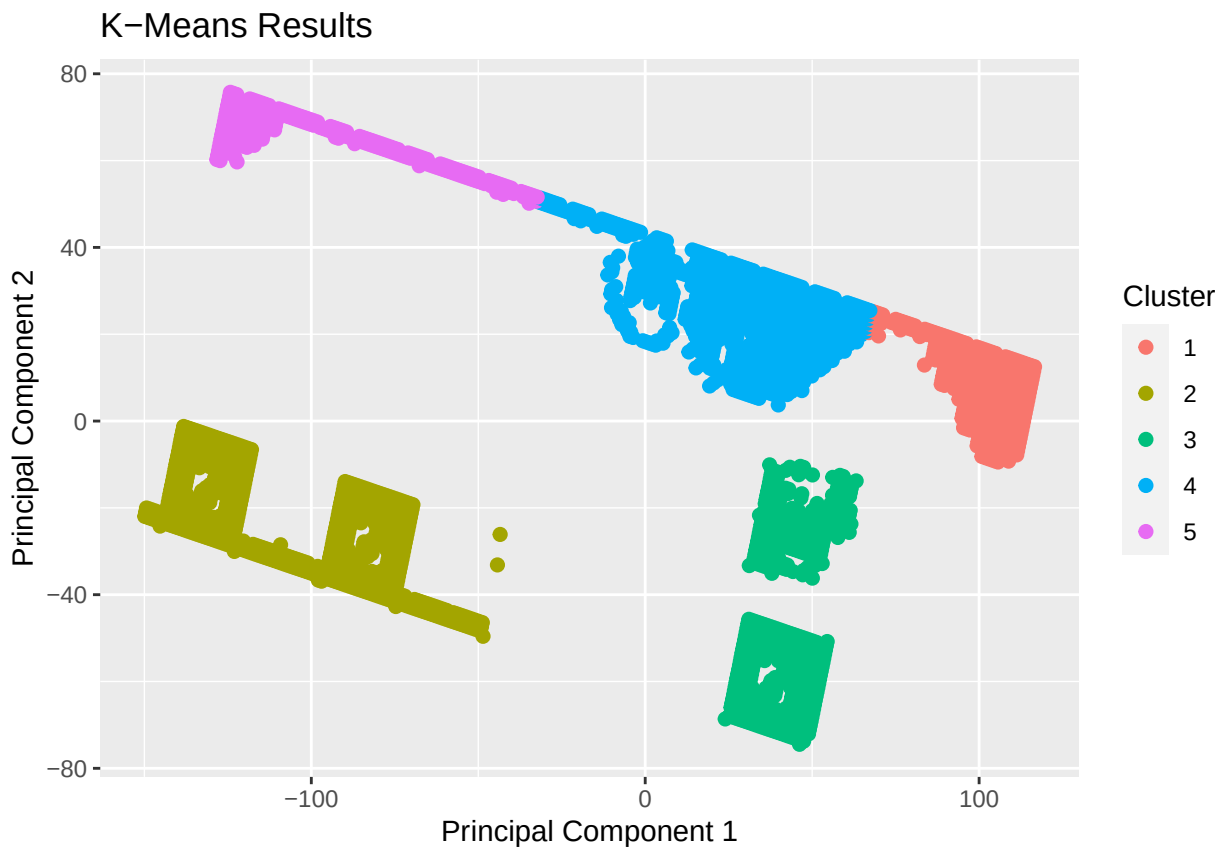

```
km3 <-  kmeans(x=clustering, centers = 3)
plot(km3, data=clustering)
```



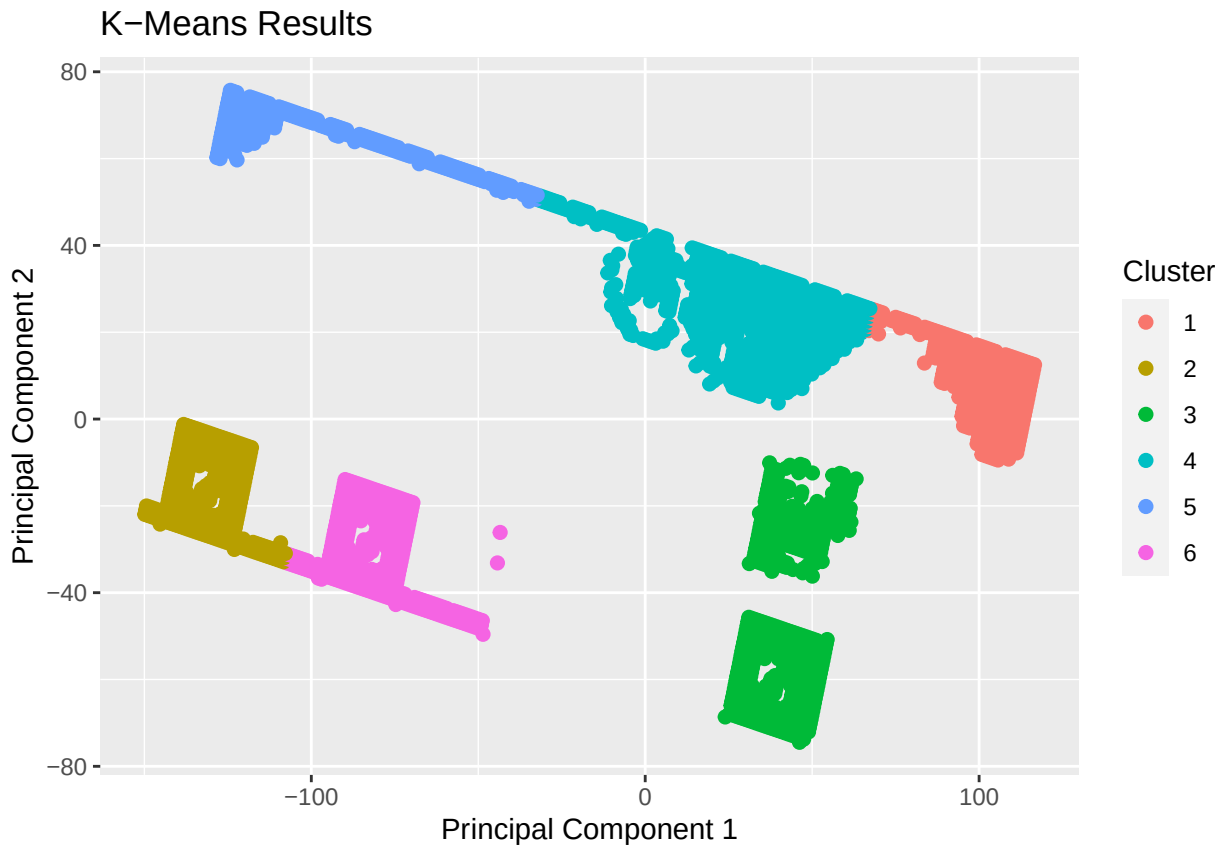

```
km4 <-  kmeans(x=clustering, centers = 4)
plot(km4, data=clustering)
```



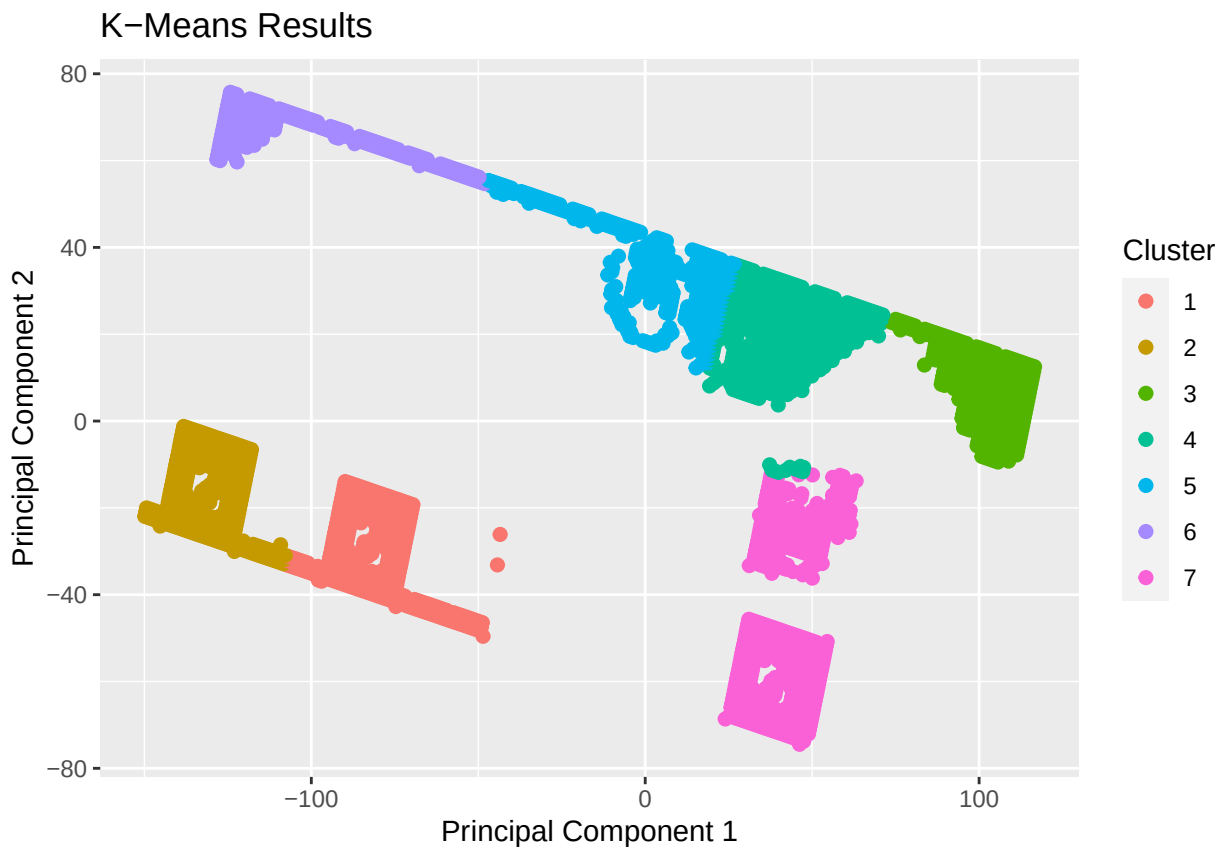

```
km5 <-  kmeans(x=clustering, centers = 5)
plot(km5, data=clustering)
```



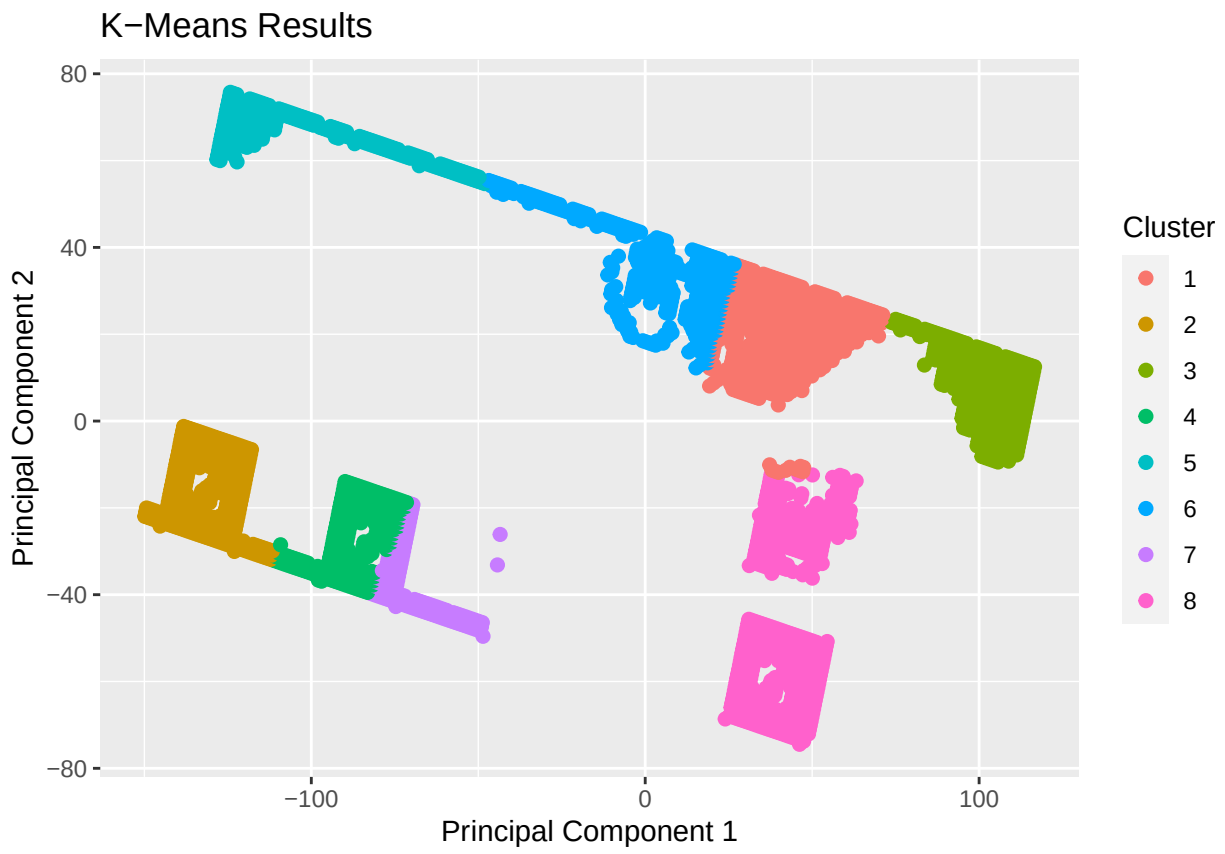

```
km6 <-  kmeans(x=clustering, centers = 6)
plot(km6, data=clustering)
```



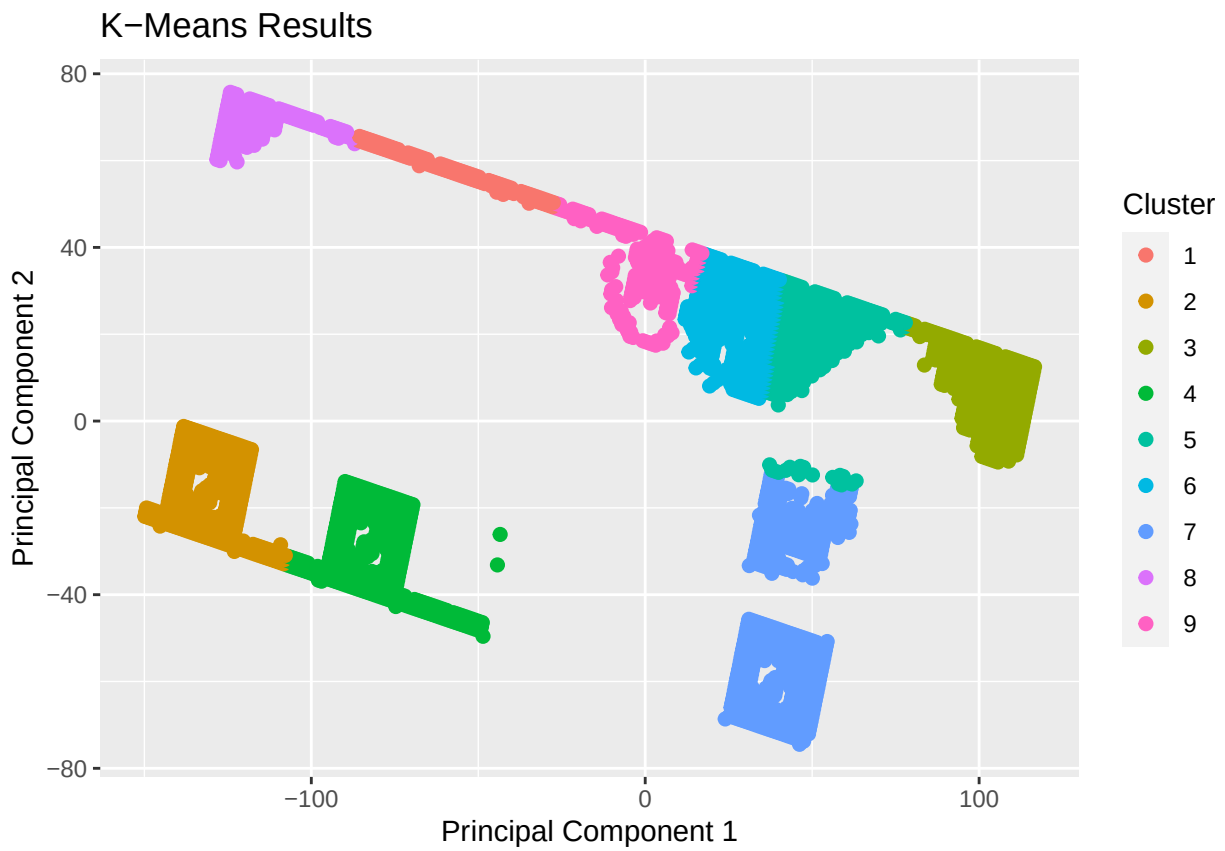

```
km7 <-  kmeans(x=clustering, centers = 7)
plot(km7, data=clustering)
```



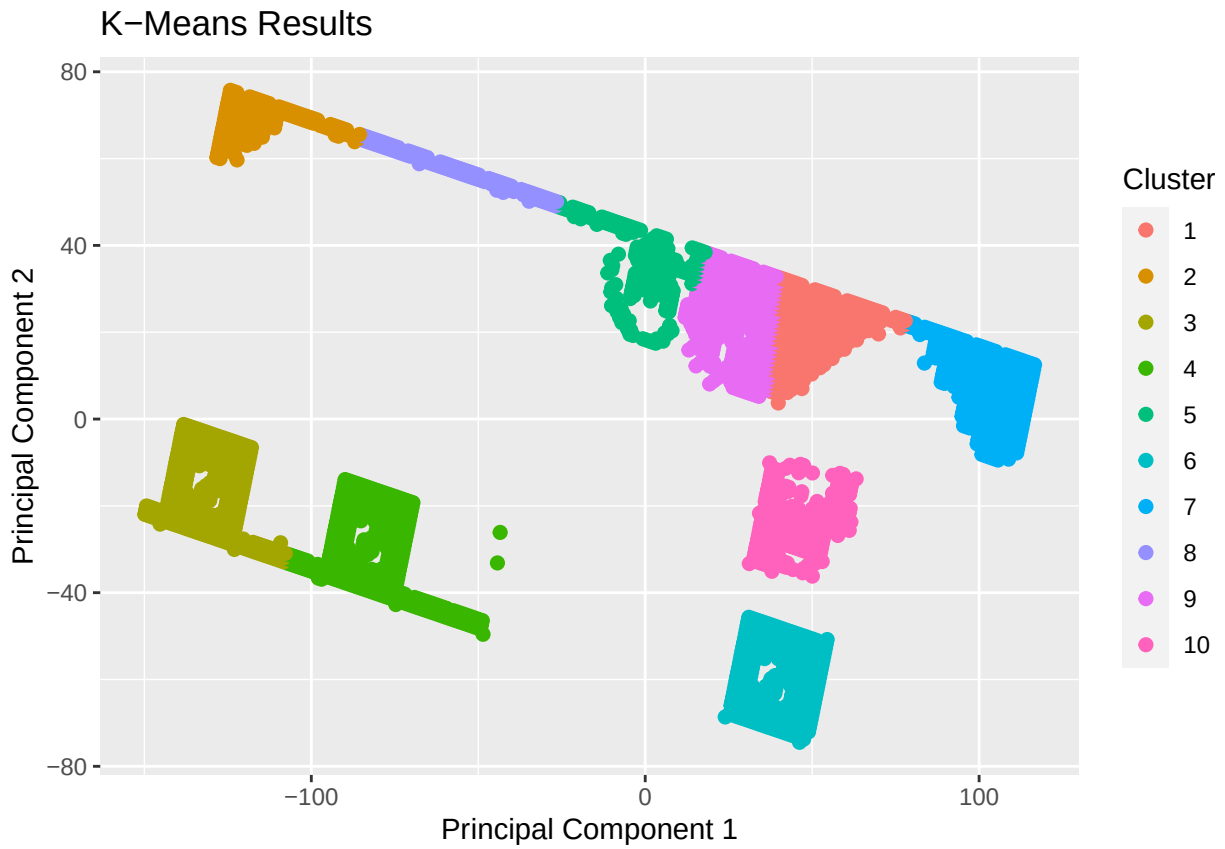

```
km8 <-  kmeans(x=clustering, centers = 8)
plot(km8, data=clustering)
```



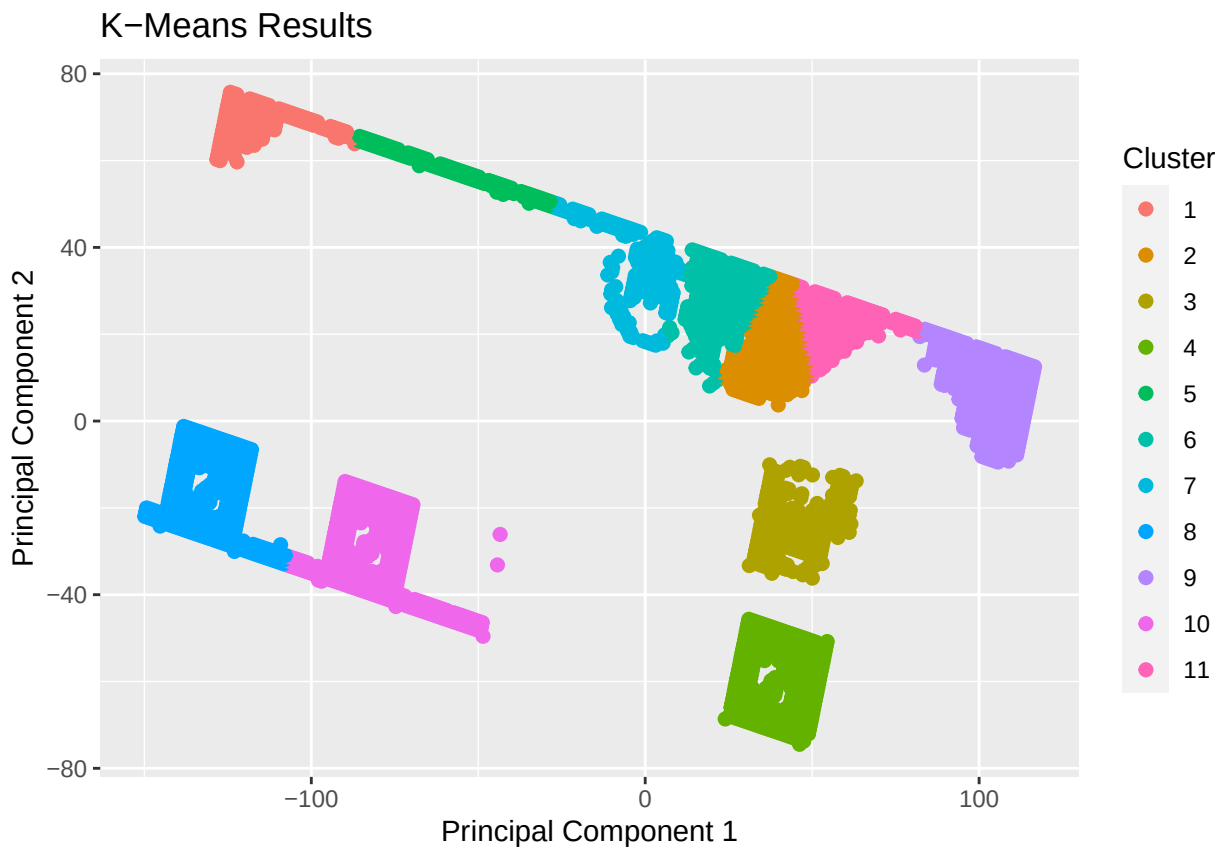

```
km9 <-  kmeans(x=clustering, centers = 9)
plot(km9, data=clustering)
```



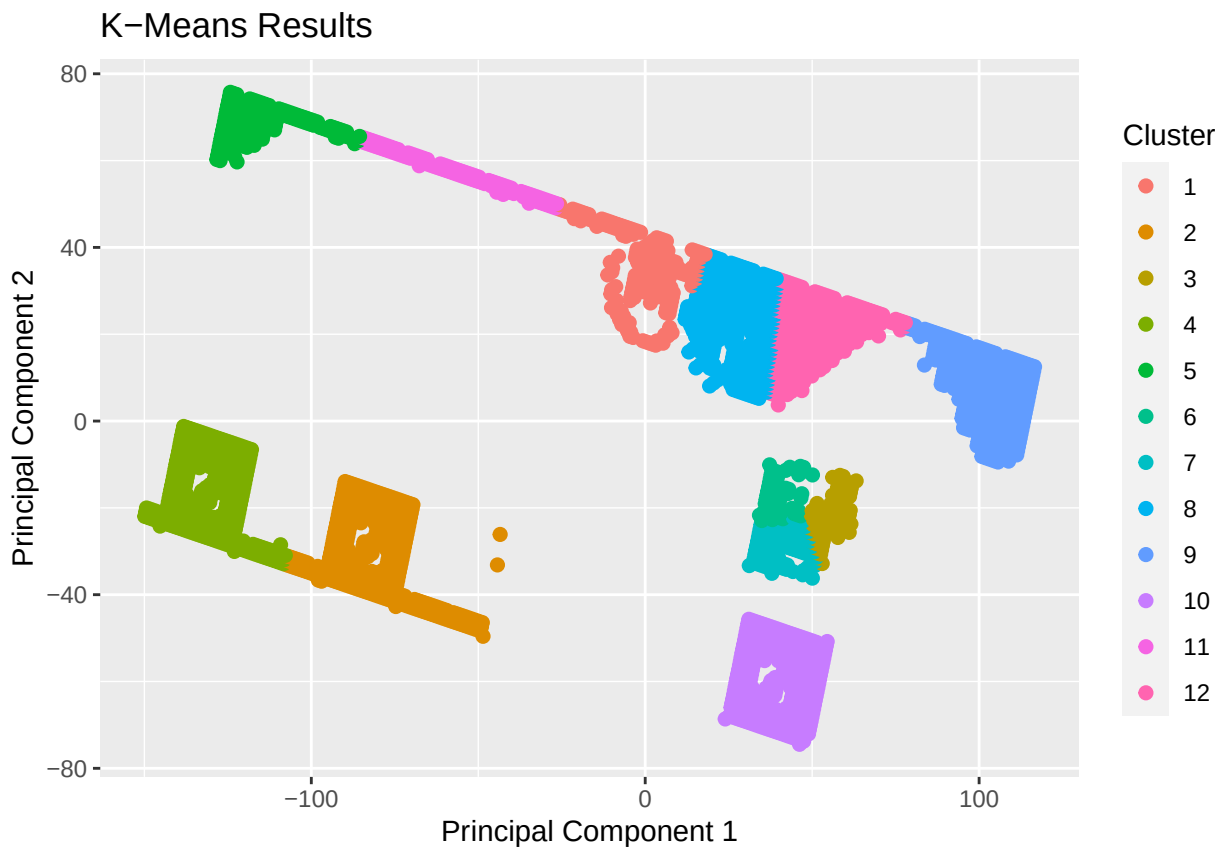

```
km10 <- kmeans(x=clustering, centers =10)
plot(km10, data=clustering)
```




```
km11 <-  kmeans(x=clustering, centers = 11)
plot(km11, data=clustering)
```

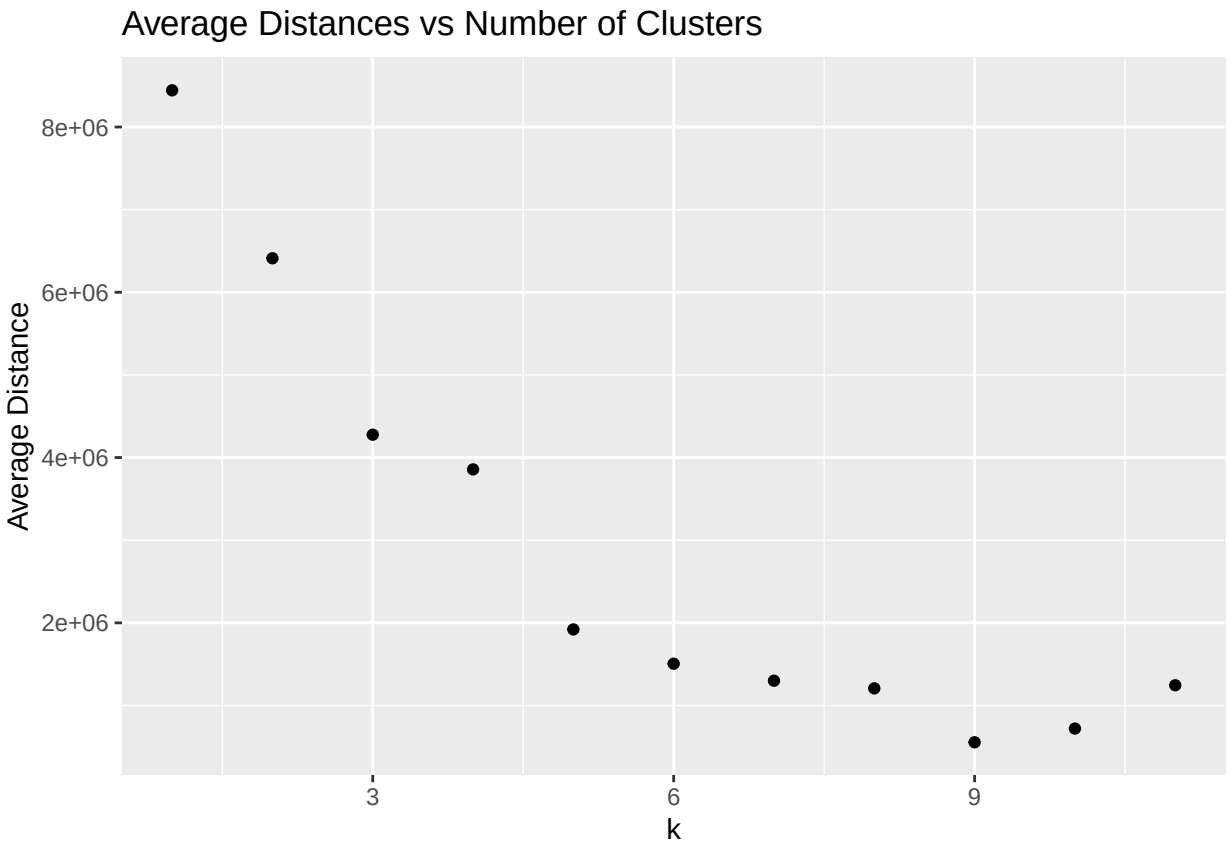



```
km12 <-  kmeans(x=clustering, centers = 12)
plot(km12, data=clustering)
```



Average Distance

```
# list of distances for each k
distances <- 2:12 %>% map(function(k) kmeans(x=clustering, k)$tot.withinss)
# distances currently list of lists this turns it into list
distances <- unlist(distances, recursive=FALSE)
# make distances a data frame and then plot k vs distance
distance_df <- data.frame(distances)
distance_df$ID <- seq.int(nrow(distance_df))
ggplot(distance_df, aes(x=ID, y=distances)) + geom_point() + xlab("k") + ylab("Average Distance") + ggt
```



Based on the above graph, the elbow point is at k=5, meaning 5 clusters is an ideal amount for the dataset.