Assignment_11.2_Venkidusamy_KesavAdithya

Kesav Adithya Venkidusamy

11/13/2021

```
knitr::opts_chunk$set(echo = TRUE)

library(ggplot2)
library(class)
library(useful)
library(scales)
```

Binary Data

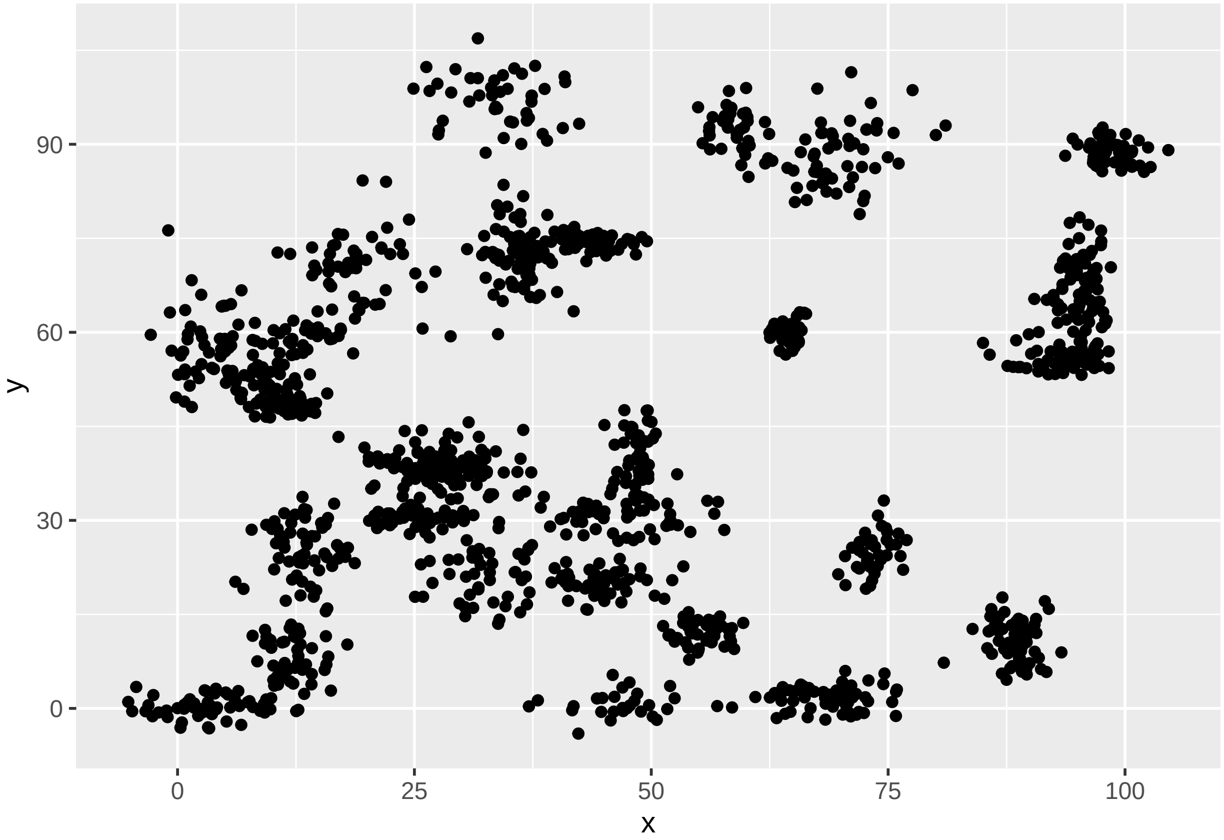
```
binary_df <- read.csv("E:/Personal/Bellevue University/Course/github/dsc520/data/binary-classifier-data
head(binary_df)
```

```
##   label        x        y
## 1     0 70.88469 83.17702
## 2     0 74.97176 87.92922
## 3     0 73.78333 92.20325
## 4     0 66.40747 81.10617
## 5     0 69.07399 84.53739
## 6     0 72.23616 86.38403
```

```
# Total number of records present in the data set
nrow(binary_df)
```

```
## [1] 1498
```

```
#Plot the data set using ggplot function
ggplot(binary_df, aes(x=x, y=y)) + geom_point()
```



Create Sample data

```
nrow_binary_df <- nrow(binary_df)

#Considering 80% for training sample
nrow_binary_sample_df <- round(0.8 * nrow_binary_df)

#Creating a vector which is an 80% random sample
set.seed(1)
binary_sample_indices <- sample(1:nrow_binary_df, nrow_binary_sample_df)

# Subset the data frame to training indices
binary_train_df <- binary_df[binary_sample_indices,]

#Creating test data
binary_test_df <- binary_df[-binary_sample_indices,]
```

Nearest neighbor algrithm

k=3

```
knn_3 <- knn(train=binary_train_df, test=binary_test_df, cl=binary_train_df$label, k=3)
cm_3 <- table(binary_test_df$label, knn_3)
cm_3
```

```
##    knn_3
##       0   1
##   0 149   4
##   1   4 143
```

```
mc_err_3 <- mean(knn_3 != binary_test_df$label)
acc_03 <- (1 - mc_err_3)
cat("Accuracy with k=3 is: ", percent(acc_03))
```

```
## Accuracy with k=3 is:  97%
```

k=5

```
knn_5 <- knn(train=binary_train_df, test=binary_test_df, cl=binary_train_df$label, k=5)
cm_5 <- table(binary_test_df$label, knn_5)
cm_5
```

```
##    knn_5
##       0   1
##   0 148   5
##   1   4 143
```

```
mc_err_5 <- mean(knn_5 != binary_test_df$label)
acc_05 <- (1 - mc_err_5)
cat("Accuracy with k=5 is: ", percent(acc_05))
```

```
## Accuracy with k=5 is:  97%
```

k=10

```
knn_10 <- knn(train=binary_train_df, test=binary_test_df, cl=binary_train_df$label, k=10)
cm_10 <- table(binary_test_df$label, knn_10)
cm_10
```

```
##    knn_10
##       0   1
##   0 146   7
##   1   3 144
```

```r
mc_err_10 <- mean(knn_10 != binary_test_df$label)
acc_10 <- (1 - mc_err_10)
cat("Accuracy with k=10 is: ", percent(acc_10))
```

```
## Accuracy with k=10 is:  97%
```

k=15

```r
knn_15 <- knn(train=binary_train_df, test=binary_test_df, cl=binary_train_df$label, k=15)
cm_15 <- table(binary_test_df$label, knn_15)
cm_15
```

```
##    knn_15
##       0   1
##   0 147   6
##   1   3 144
```

```r
mc_err_15 <- mean(knn_15 != binary_test_df$label)
acc_15 <- (1 - mc_err_15)
cat("Accuracy with k=15 is: ", percent(acc_15))
```

```
## Accuracy with k=15 is:  97%
```

k=20

```r
knn_20 <- knn(train=binary_train_df, test=binary_test_df, cl=binary_train_df$label, k=20)
cm_20 <- table(binary_test_df$label, knn_20)
cm_20
```

```
##    knn_20
##       0   1
##   0 147   6
##   1   2 145
```

```r
mc_err_20 <- mean(knn_20 != binary_test_df$label)
acc_20 <- (1 - mc_err_20)
cat("Accuracy with k=20 is: ", percent(acc_20))
```

```
## Accuracy with k=20 is:  97%
```

k=25

```r
knn_25 <- knn(train=binary_train_df, test=binary_test_df, cl=binary_train_df$label, k=25)
cm_25 <- table(binary_test_df$label, knn_25)
cm_25
```

```
##    knn_25
##       0   1
##   0 146   7
##   1   2 145
```

```
mc_err_25 <- mean(knn_25 != binary_test_df$label)
acc_25 <- (1 - mc_err_25)
cat("Accuracy with k=25 is: ", percent(acc_25))
```

```
## Accuracy with k=25 is:  97%
```

Plot the accuracy and k values

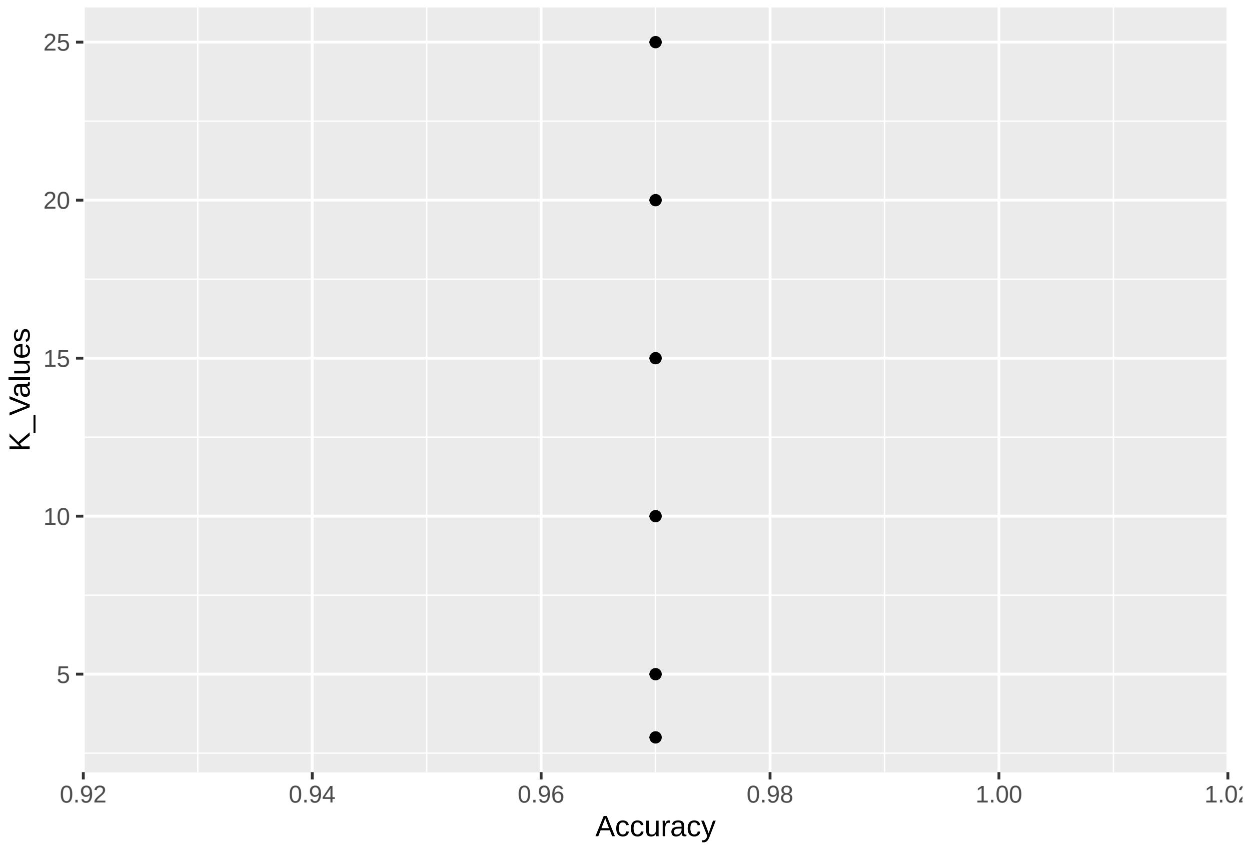
```
k_vals <- c(3,5,10,15,20,25)
acc_list <- ls(pattern="acc_\\d")
acc_vals <- sapply(acc_list, function(x) parse(text=x))
plot_vals <- as.data.frame(cbind(unlist(data.frame(as.list(acc_vals))), k_vals))
acc_vals
```

```
## expression(acc_03 = acc_03, acc_05 = acc_05, acc_10 = acc_10,
##     acc_15 = acc_15, acc_20 = acc_20, acc_25 = acc_25)
```

```
colnames(plot_vals) <- c("Accuracy", "K_Values")
plot_vals <- transform(plot_vals, Accuracy=as.numeric(Accuracy))
plot_vals <- transform(plot_vals, Accuracy=round(Accuracy, digits=2))
plot_vals
```

```
##        Accuracy K_Values
## acc_03     0.97        3
## acc_05     0.97        5
## acc_10     0.97       10
## acc_15     0.97       15
## acc_20     0.97       20
## acc_25     0.97       25
```

```
ggplot(plot_vals, aes(x=Accuracy, y =K_Values))+geom_point()
```



Trinary Data

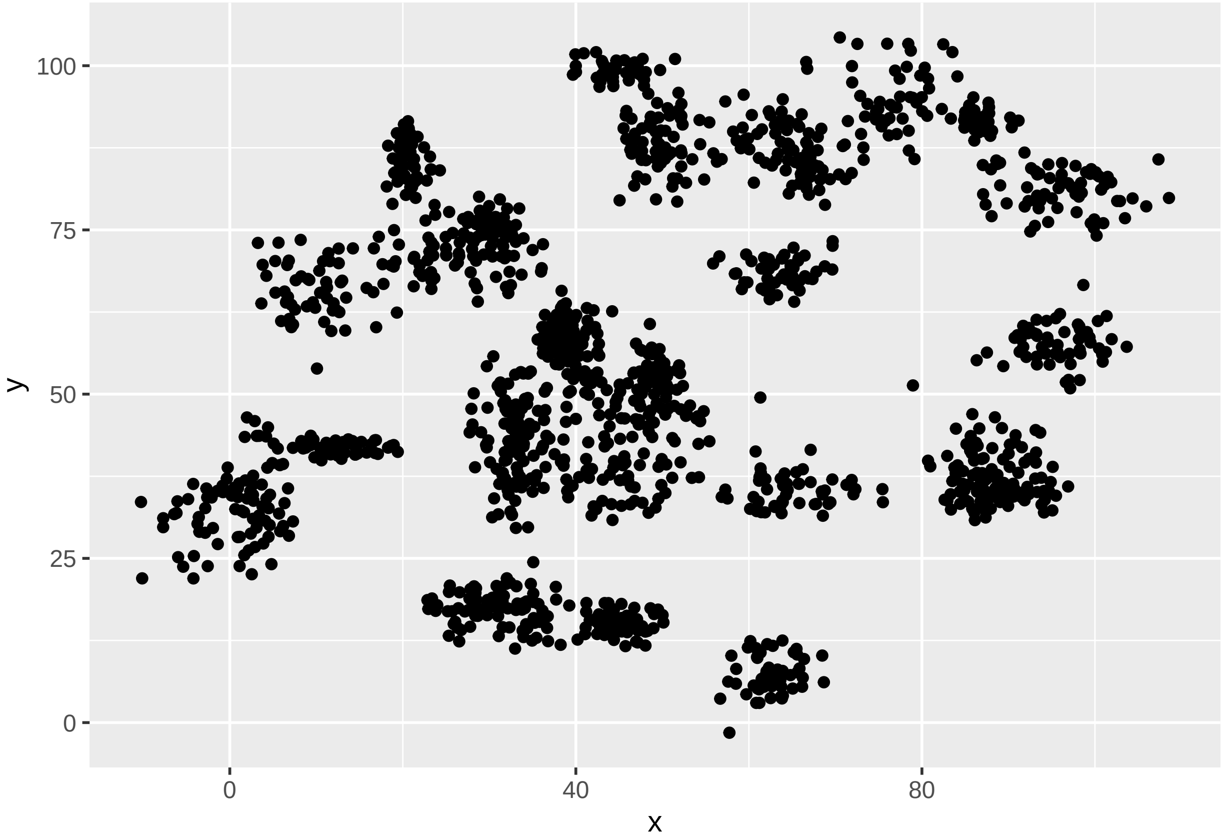
```
trinary_df <- read.csv("E:/Personal/Bellevue University/Course/github/dsc520/data/trinary-classifier-da
head(trinary_df)
```

```
##   label        x        y
## 1     0 30.08387 39.63094
## 2     0 31.27613 51.77511
## 3     0 34.12138 49.27575
## 4     0 32.58222 41.23300
## 5     0 34.65069 45.47956
## 6     0 33.80513 44.24656
```

```
# Total number of records present in the data set
nrow(trinary_df)
```

```
## [1] 1568
```

```
#Plot the data set using ggplot function
ggplot(trinary_df, aes(x=x, y=y)) + geom_point()
```



Create Sample data for Trinary data

```
nrow_trinary_df <- nrow(trinary_df)

#Considering 80% for training sample
nrow_trinary_sample_df <- round(0.8 * nrow_trinary_df)

#Creating a vector which is an 80% random sample
set.seed(1)
trinary_sample_indices <- sample(1:nrow_trinary_df, nrow_trinary_sample_df)

# Subset the data frame to training indices
trinary_train_df <- trinary_df[trinary_sample_indices,]

#Creating test data
trinary_test_df <- trinary_df[-trinary_sample_indices,]
```

Nearest neighbor algrithm

k=3

```
knn_3 <- knn(train=trinary_train_df, test=trinary_test_df, cl=trinary_train_df$label, k=3)
cm_3 <- table(trinary_test_df$label, knn_3)
cm_3
```

```
##    knn_3
##       0   1   2
##   0  71   4   0
##   1   4 132   2
##   2   5   4  92
```

```
mc_err_3 <- mean(knn_3 != trinary_test_df$label)
acc_03 <- (1 - mc_err_3)
cat("Accuracy with k=3 is: ", percent(acc_03))
```

```
## Accuracy with k=3 is:  94%
```

k=5

```
knn_5 <- knn(train=trinary_train_df, test=trinary_test_df, cl=trinary_train_df$label, k=5)
cm_5 <- table(trinary_test_df$label, knn_5)
cm_5
```

```
##    knn_5
##       0   1   2
##   0  71   4   0
##   1   3 135   0
##   2   6   1  94
```

```
mc_err_5 <- mean(knn_5 != trinary_test_df$label)
acc_05 <- (1 - mc_err_5)
cat("Accuracy with k=5 is: ", percent(acc_05))
```

```
## Accuracy with k=5 is:  96%
```

k=10

```
knn_10 <- knn(train=trinary_train_df, test=trinary_test_df, cl=trinary_train_df$label, k=10)
cm_10 <- table(trinary_test_df$label, knn_10)
cm_10
```

```
##    knn_10
##       0   1   2
##   0  67   7   1
##   1   4 134   0
##   2   7   2  92
```

```
mc_err_10 <- mean(knn_10 != trinary_test_df$label)
acc_10 <- (1 - mc_err_10)
cat("Accuracy with k=10 is: ", percent(acc_10))
```

```
## Accuracy with k=10 is:  93%
```

k=15

```
knn_15 <- knn(train=trinary_train_df, test=trinary_test_df, cl=trinary_train_df$label, k=15)
cm_15 <- table(trinary_test_df$label, knn_15)
cm_15
```

```
##    knn_15
##       0   1   2
##   0  65   9   1
##   1   6 130   2
##   2   9   3  89
```

```
mc_err_15 <- mean(knn_15 != trinary_test_df$label)
acc_15 <- (1 - mc_err_15)
cat("Accuracy with k=15 is: ", percent(acc_15))
```

```
## Accuracy with k=15 is:  90%
```

k=20

```
knn_20 <- knn(train=trinary_train_df, test=trinary_test_df, cl=trinary_train_df$label, k=20)
cm_20 <- table(trinary_test_df$label, knn_20)
cm_20
```

```
##    knn_20
##       0   1   2
##   0  64  11   0
##   1   5 132   1
##   2   8   3  90
```

```
mc_err_20 <- mean(knn_20 != trinary_test_df$label)
acc_20 <- (1 - mc_err_20)
cat("Accuracy with k=20 is: ", percent(acc_20))
```

```
## Accuracy with k=20 is:  91%
```

k=25

```
knn_25 <- knn(train=trinary_train_df, test=trinary_test_df, cl=trinary_train_df$label, k=25)
cm_25 <- table(trinary_test_df$label, knn_25)
cm_25
```

```
##    knn_25
##       0   1   2
##   0  64  10   1
##   1   7 131   0
##   2   9   3  89
```

```
mc_err_25 <- mean(knn_25 != trinary_test_df$label)
acc_25 <- (1 - mc_err_25)
cat("Accuracy with k=25 is: ", percent(acc_25))
```

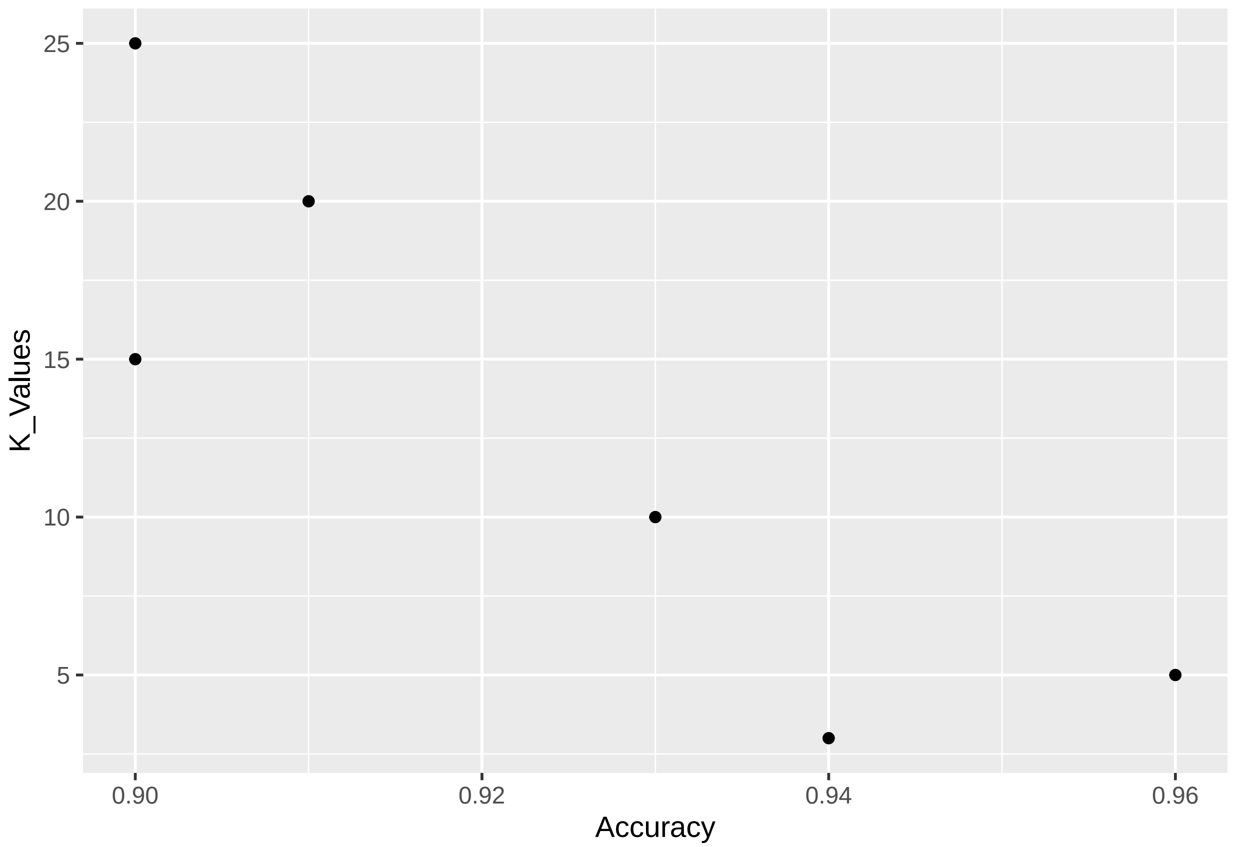
```
## Accuracy with k=25 is:  90%
```

Plot the accuracy and k values

```
k_vals <- c(3,5,10,15,20,25)
acc_list <- ls(pattern="acc_\\d")
acc_vals <- sapply(acc_list, function(x) parse(text=x))
plot_vals <- as.data.frame(cbind(unlist(data.frame(as.list(acc_vals))), k_vals))
colnames(plot_vals) <- c("Accuracy", "K_Values")
plot_vals <- transform(plot_vals, Accuracy=as.numeric(Accuracy))
plot_vals <- transform(plot_vals, Accuracy=round(Accuracy, digits=2))
plot_vals
```

```
##        Accuracy K_Values
## acc_03     0.94        3
## acc_05     0.96        5
## acc_10     0.93       10
## acc_15     0.90       15
## acc_20     0.91       20
## acc_25     0.90       25
```

```
ggplot(plot_vals, aes(x=Accuracy, y =K_Values))+geom_point()
```



Clustering

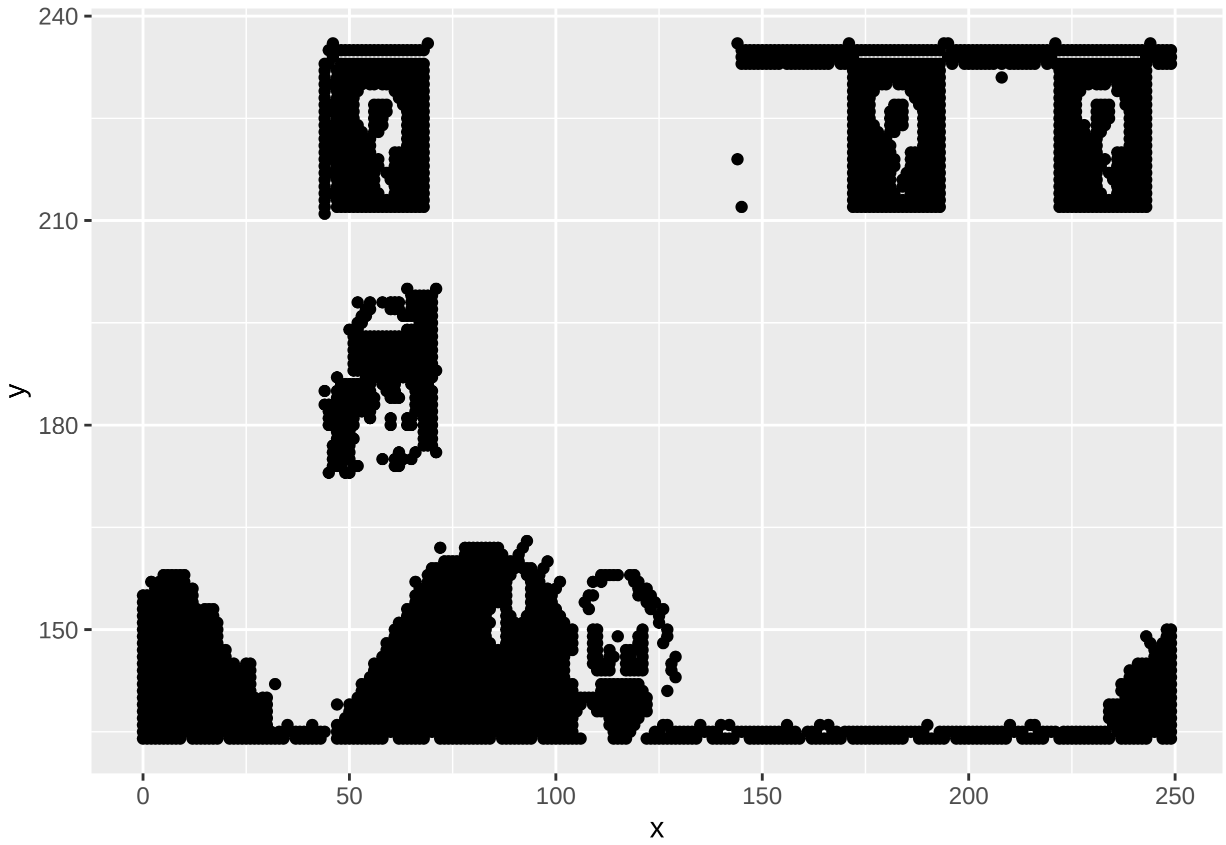
```
cluster_df <- read.csv("E:/Personal/Bellevue University/Course/github/dsc520/data/clustering-data.csv")

cat("Total number of records: ",nrow(cluster_df))
```

```
## Total number of records:  4022
```

```
head(cluster_df)
```

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

```
ggplot(cluster_df, aes(x=x,y=y)) + geom_point()
```



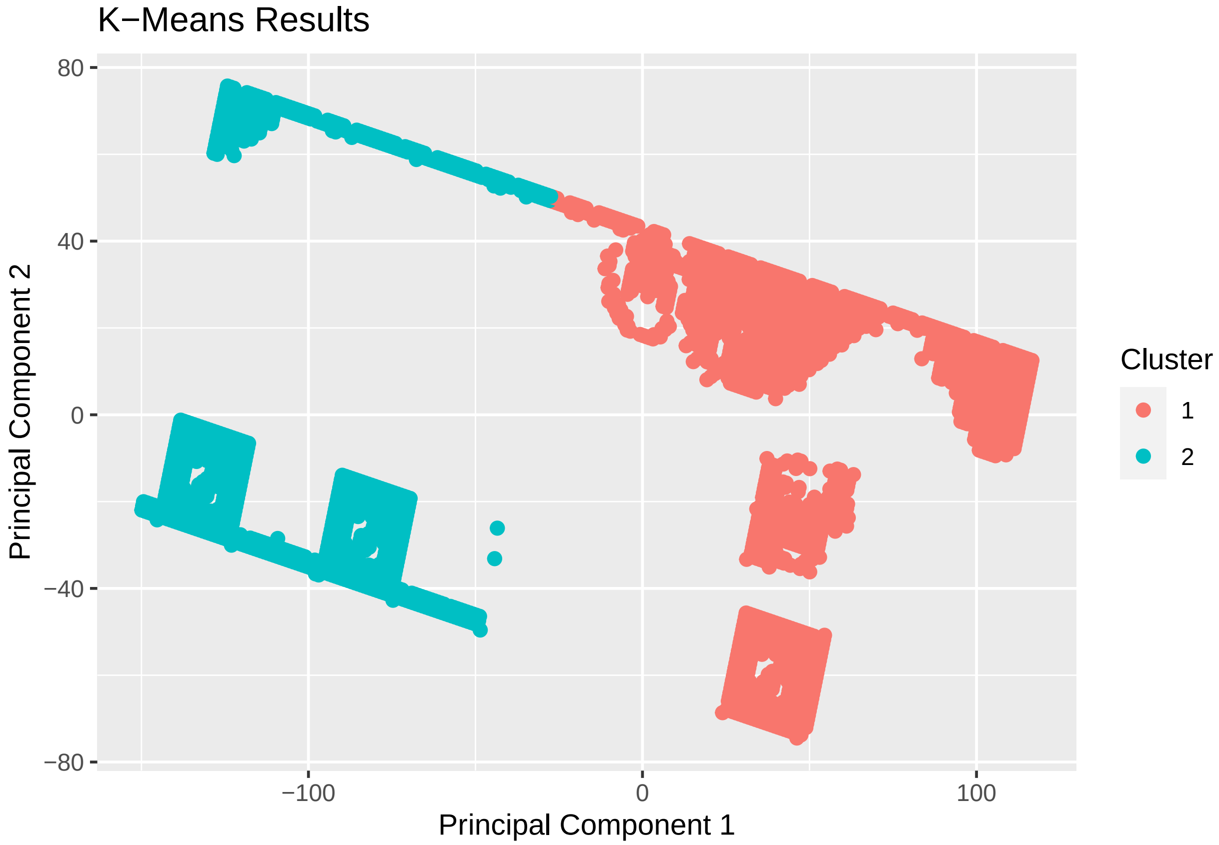
K-mean plot for K=2 to 12

```
for (x in 2:12){
  print(paste0("k-means algorithm creating variable k", x))
  assign(paste0("k",x),kmeans(cluster_df, centers=x))

}
```

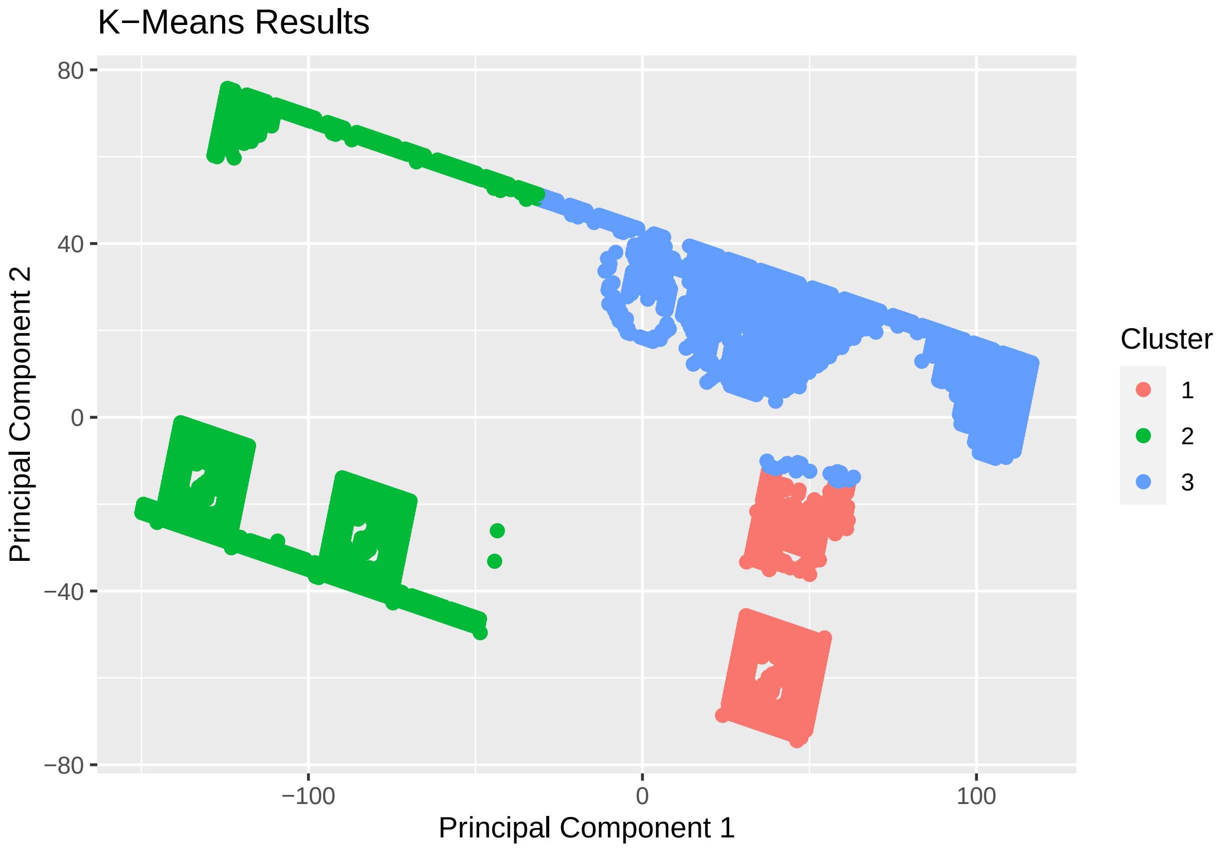
```
## [1] "k-means algorithm creating variable k2"
## [1] "k-means algorithm creating variable k3"
## [1] "k-means algorithm creating variable k4"
## [1] "k-means algorithm creating variable k5"
## [1] "k-means algorithm creating variable k6"
## [1] "k-means algorithm creating variable k7"
## [1] "k-means algorithm creating variable k8"
## [1] "k-means algorithm creating variable k9"
## [1] "k-means algorithm creating variable k10"
## [1] "k-means algorithm creating variable k11"
## [1] "k-means algorithm creating variable k12"
```

Display k-means cluster

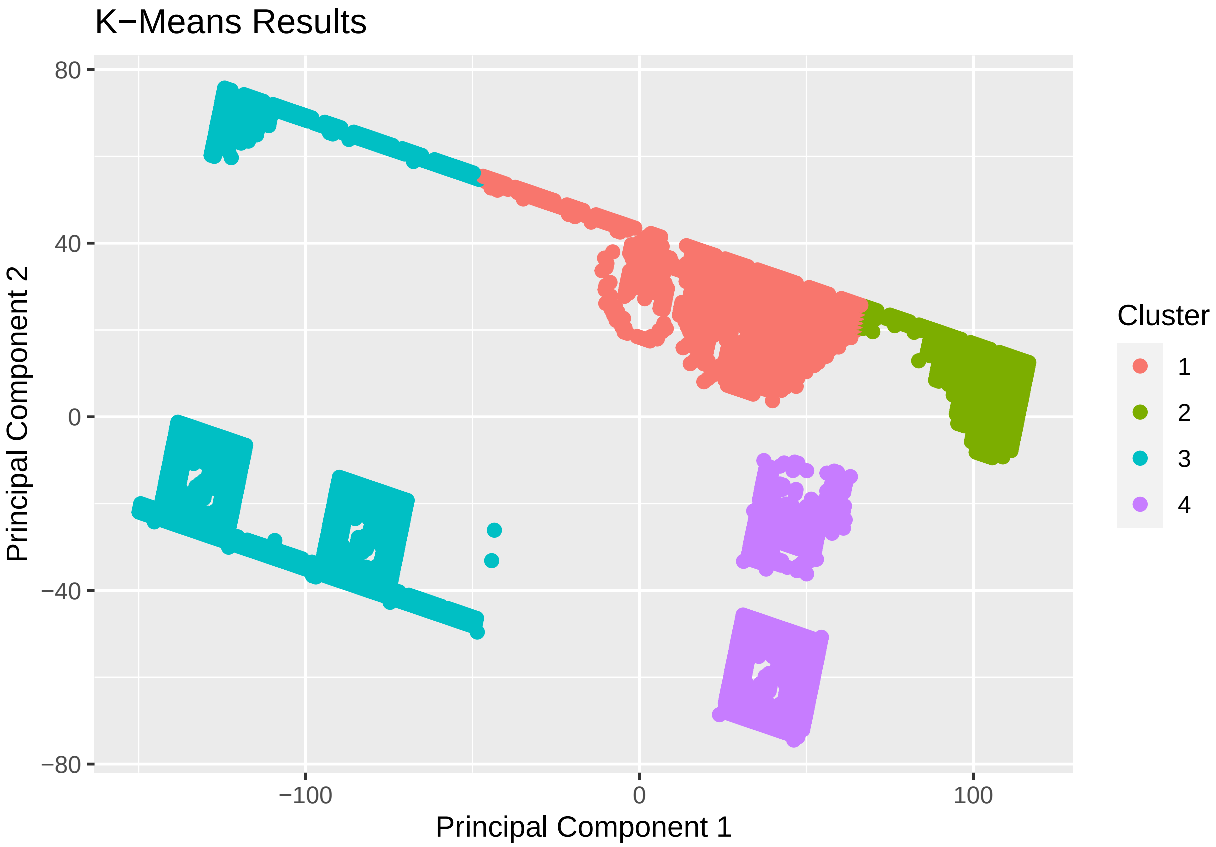
```
k2_cluster <- useful::plot.kmeans(k2, data=cluster_df)
k3_cluster <- useful::plot.kmeans(k3, data=cluster_df)
k4_cluster <- useful::plot.kmeans(k4, data=cluster_df)
k5_cluster <- useful::plot.kmeans(k5, data=cluster_df)
k6_cluster <- useful::plot.kmeans(k6, data=cluster_df)
k7_cluster <- useful::plot.kmeans(k7, data=cluster_df)
k8_cluster <- useful::plot.kmeans(k8, data=cluster_df)
k9_cluster <- useful::plot.kmeans(k9, data=cluster_df)
k10_cluster <- useful::plot.kmeans(k10, data=cluster_df)
k11_cluster <- useful::plot.kmeans(k11, data=cluster_df)
k12_cluster <- useful::plot.kmeans(k12, data=cluster_df)

print(k2_cluster)
```



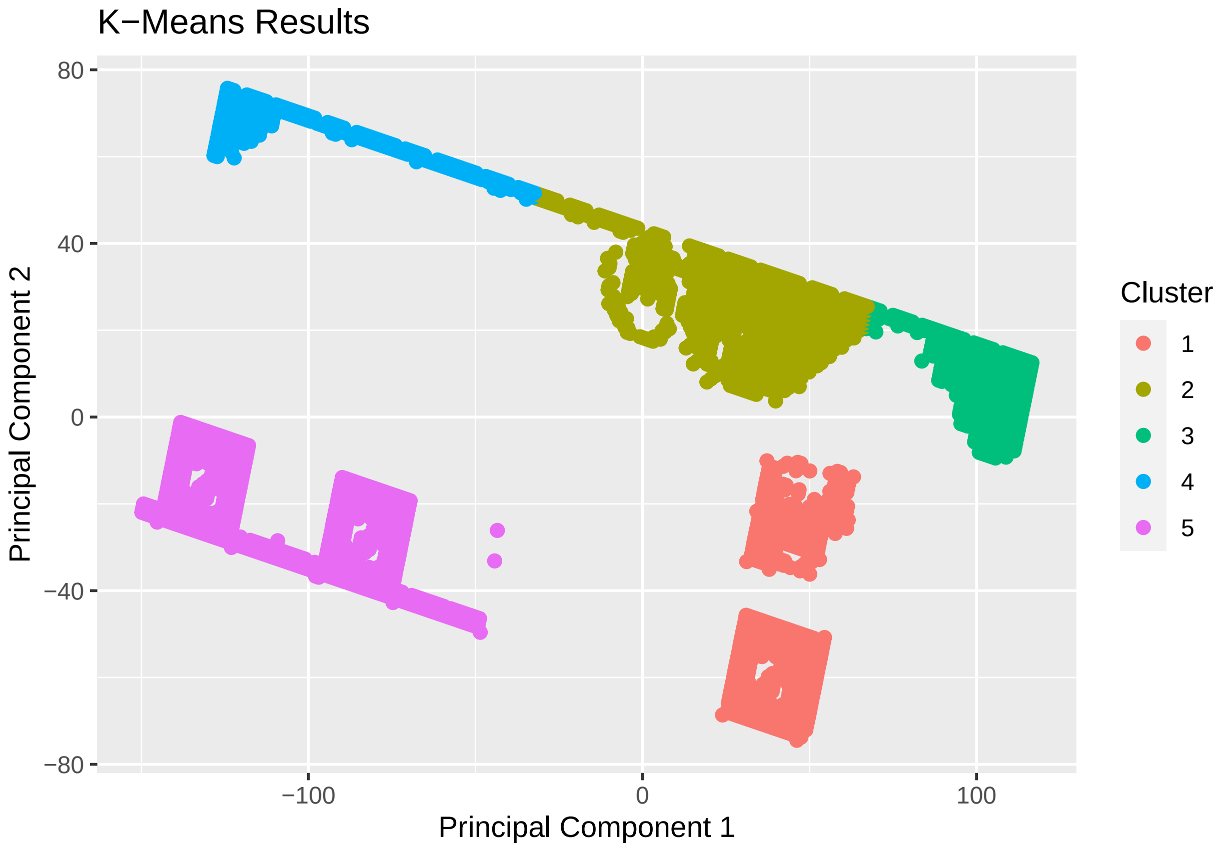
```
print(k3_cluster)
```



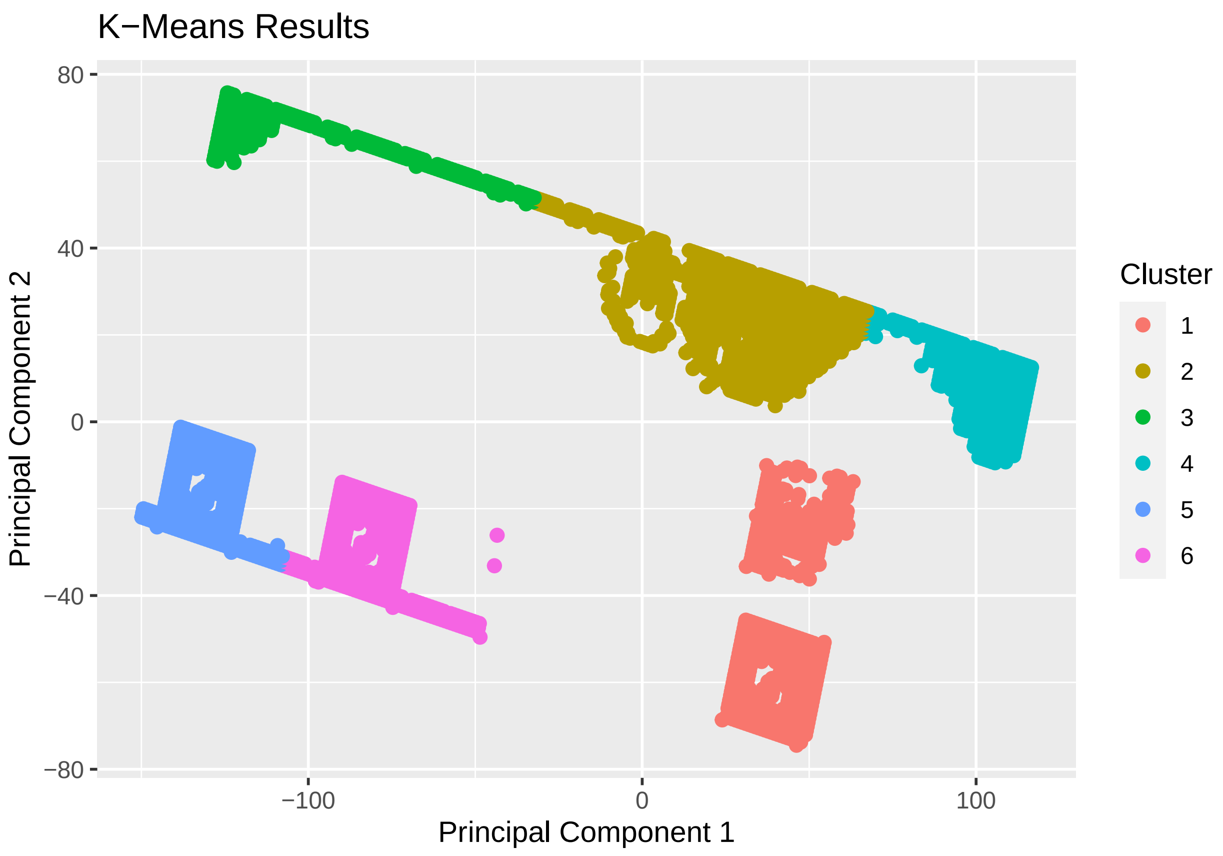
```
print(k4_cluster)
```



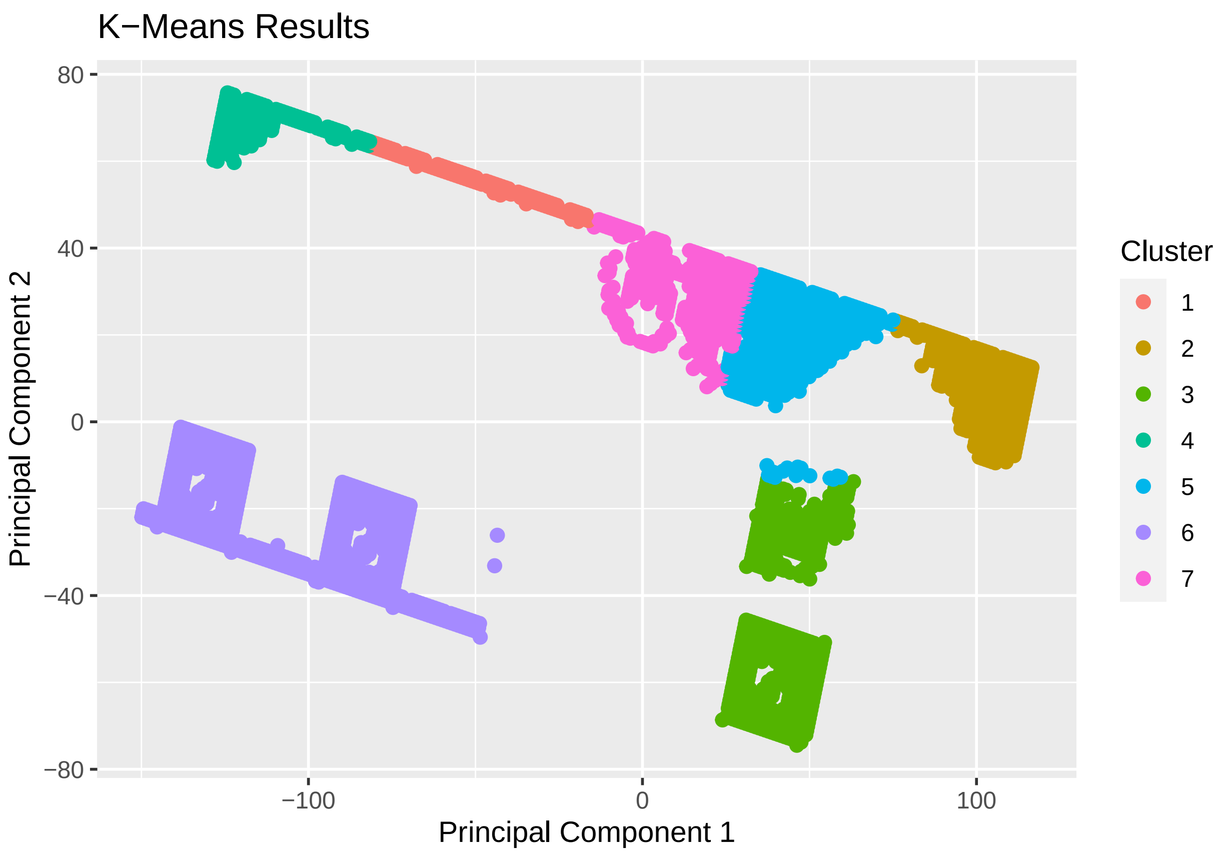
```
print(k5_cluster)
```



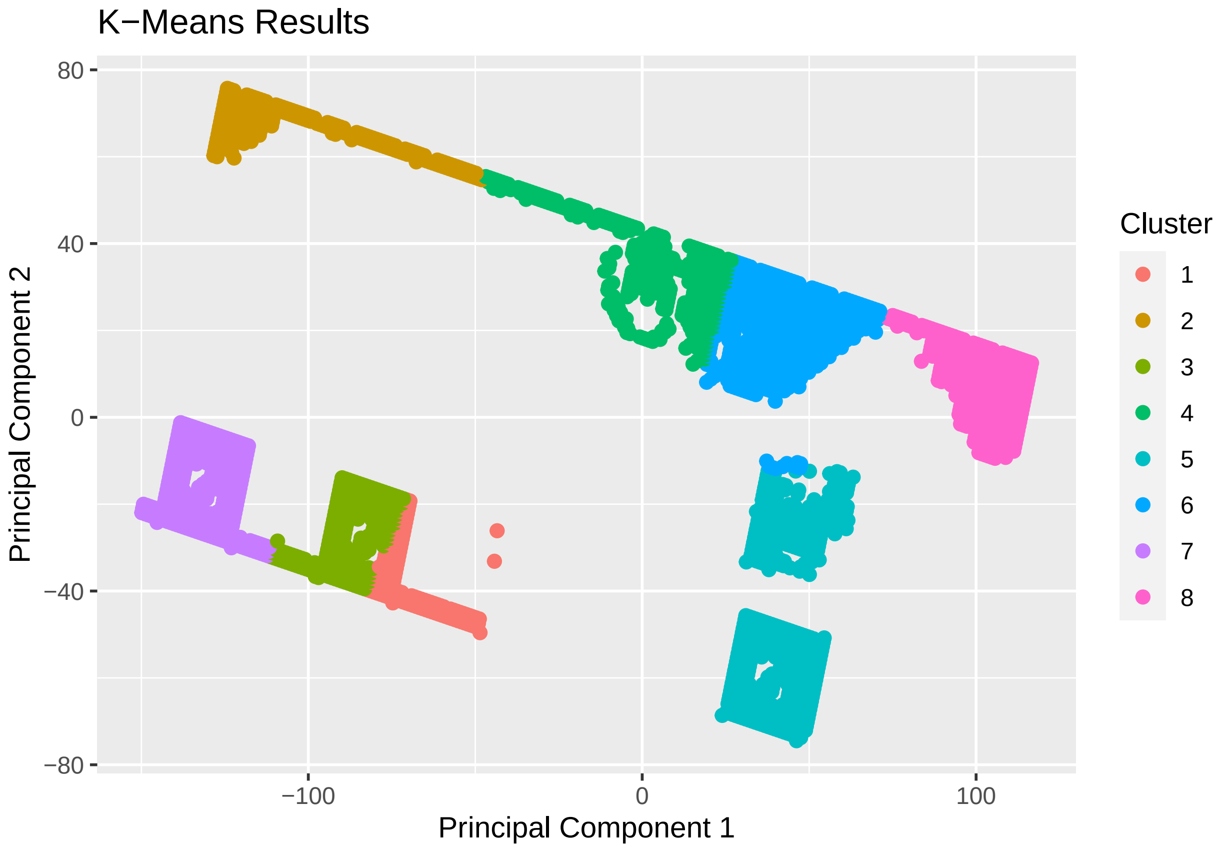
```
print(k6_cluster)
```



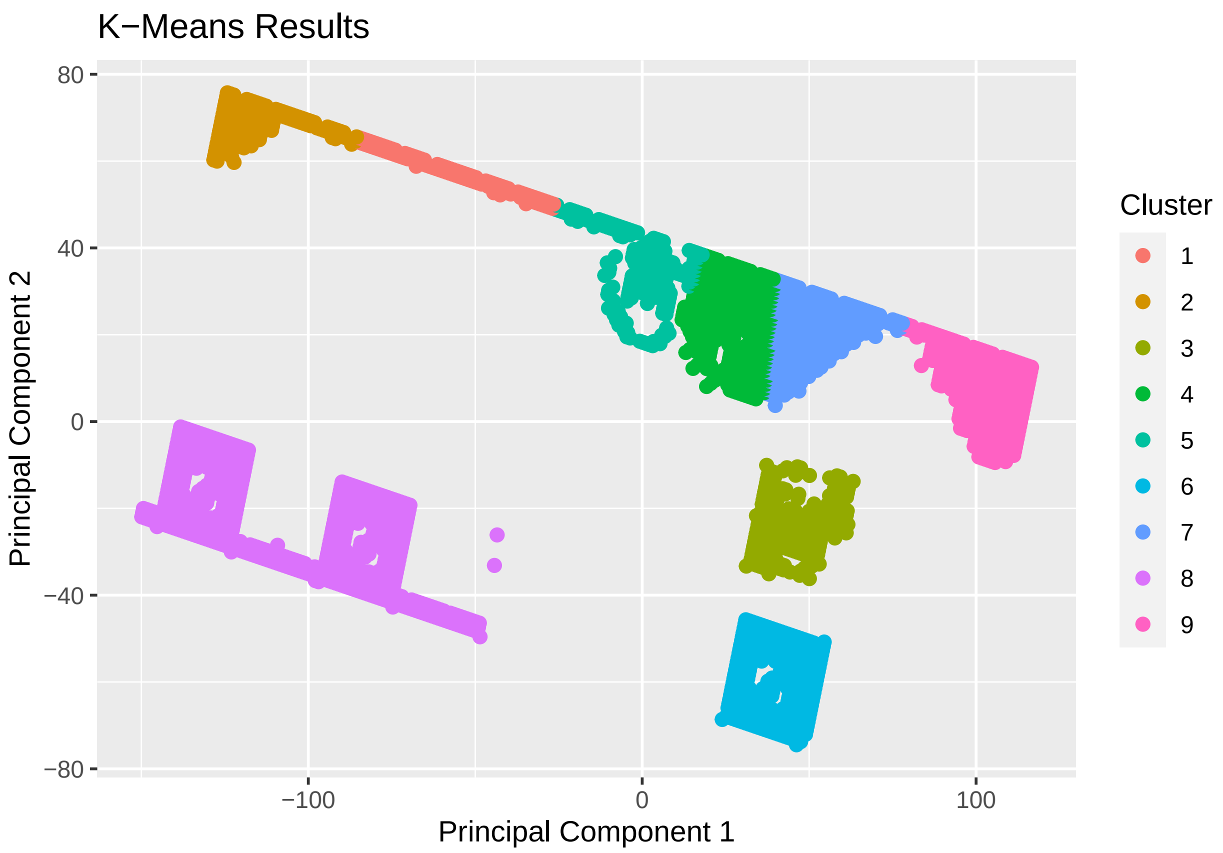
```
print(k7_cluster)
```



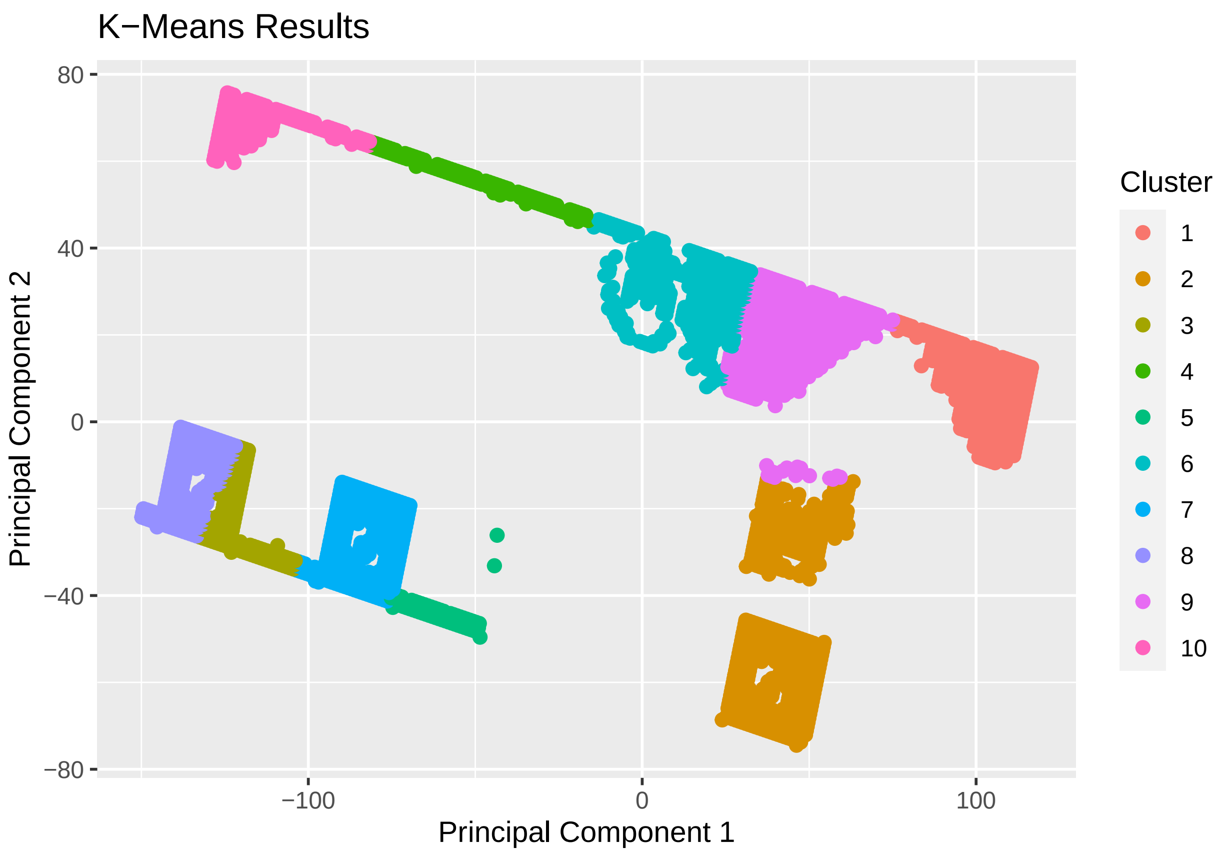
```
print(k8_cluster)
```



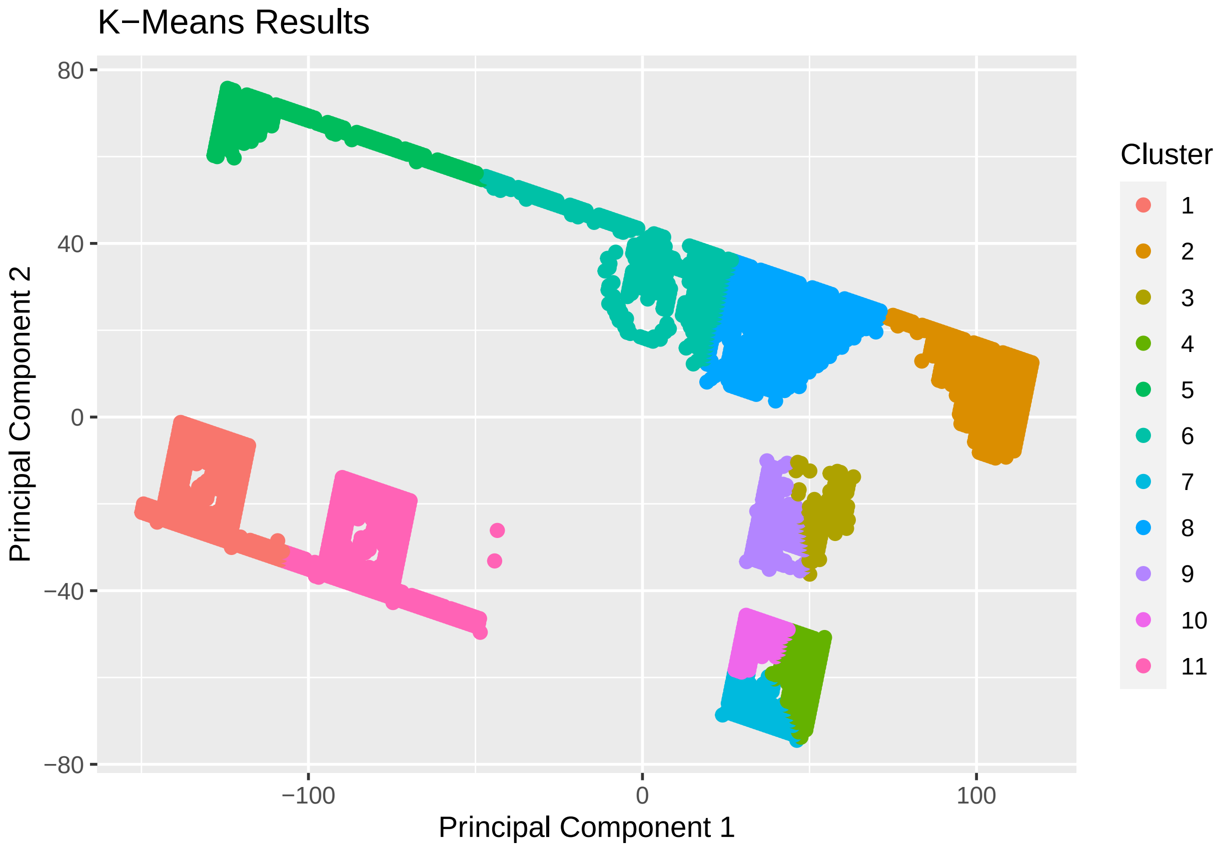
```
print(k9_cluster)
```



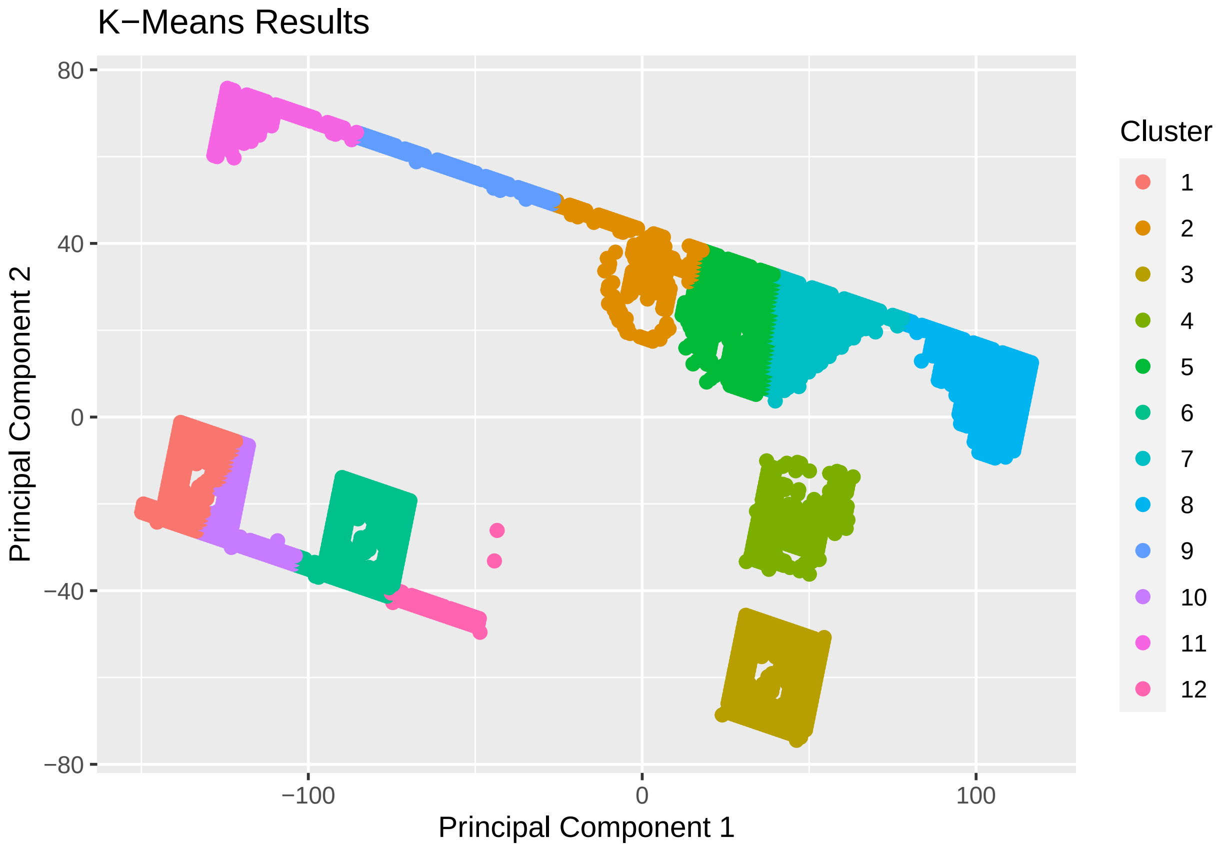
```
print(k10_cluster)
```



```
print(k11_cluster)
```



```
print(k12_cluster)
```



Accuracy for k-mean values

```
for (x in 2:12){
  temp_k <- eval(parse(text=paste0("k",x)), .GlobalEnv)
  print("Accuracy of")
  print(paste0("k",x))
  print(mean(temp_k$centers))
  cat("\n")
}
```

```
## [1] "Accuracy of"
## [1] "k2"
## [1] 158.9452
##
## [1] "Accuracy of"
## [1] "k3"
## [1] 147.7754
##
## [1] "Accuracy of"
## [1] "k4"
## [1] 133.6578
##
## [1] "Accuracy of"
## [1] "k5"
## [1] 144.1224
##
## [1] "Accuracy of"
## [1] "k6"
## [1] 156.056
##
## [1] "Accuracy of"
## [1] "k7"
## [1] 142.6891
##
## [1] "Accuracy of"
## [1] "k8"
## [1] 157.7385
##
## [1] "Accuracy of"
## [1] "k9"
## [1] 139.2011
##
## [1] "Accuracy of"
## [1] "k10"
## [1] 163.7221
##
## [1] "Accuracy of"
## [1] "k11"
## [1] 144.853
##
## [1] "Accuracy of"
## [1] "k12"
## [1] 157.6006
```

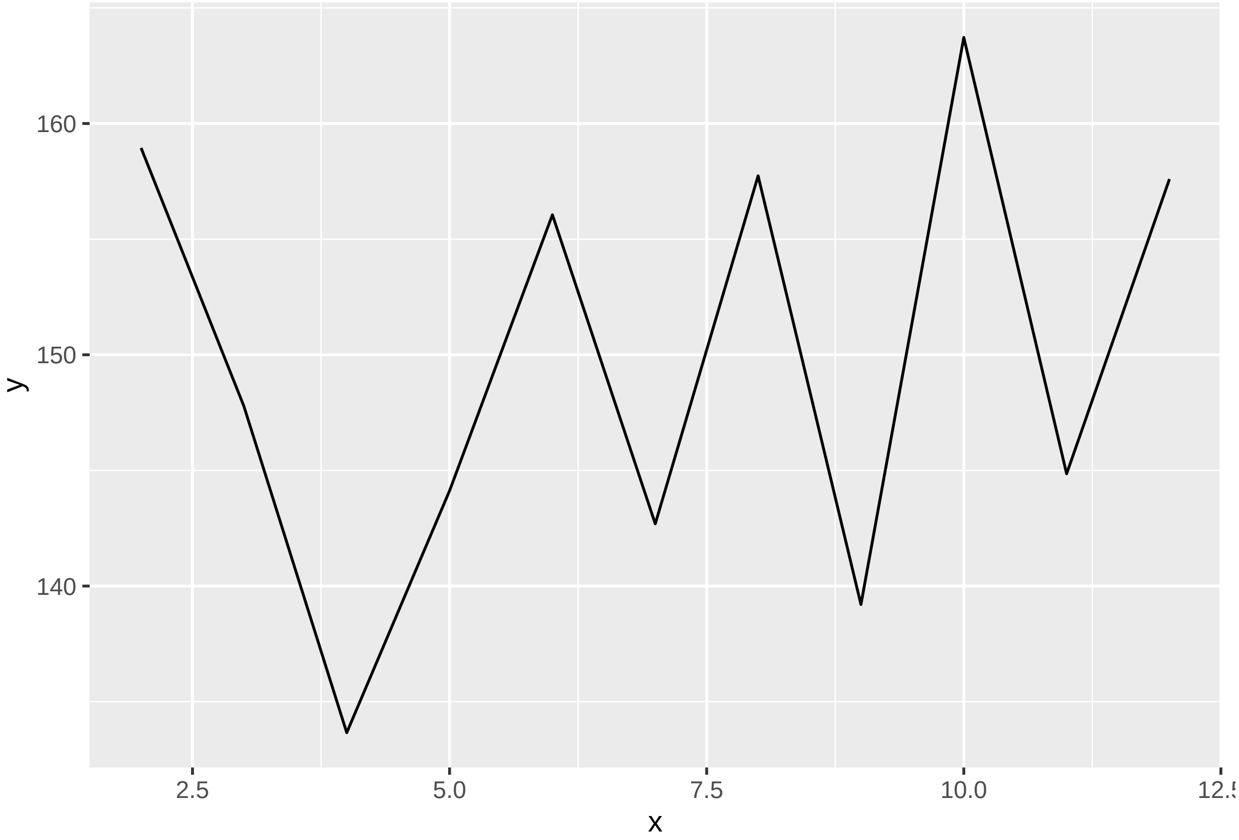
```
k_clusers <- list(k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12)

k_dists <- sapply(k_clusers, function(x) mean(x$centers))
k_dists
```

```
##  [1] 158.9452 147.7754 133.6578 144.1224 156.0560 142.6891 157.7385 139.2011
##  [9] 163.7221 144.8530 157.6006
```

```
dist_data <- cbind(2:12, k_dists)
colnames(dist_data) <- c("x", "y")
dist_data <- data.frame(dist_data)
dist_data
```

```
##     x        y
## 1   2 158.9452
## 2   3 147.7754
## 3   4 133.6578
## 4   5 144.1224
## 5   6 156.0560
## 6   7 142.6891
## 7   8 157.7385
## 8   9 139.2011
## 9  10 163.7221
## 10 11 144.8530
## 11 12 157.6006
```

```
ggplot(dist_data, aes(x=x,y=y)) + geom_line()
```



Observation: The elbow of this plot is present between 7.5 to 8