Assignment 11.2

2022-02-28

Load Libraries

```
library(ggplot2)
library(pander)
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(class)
library(caTools)
```

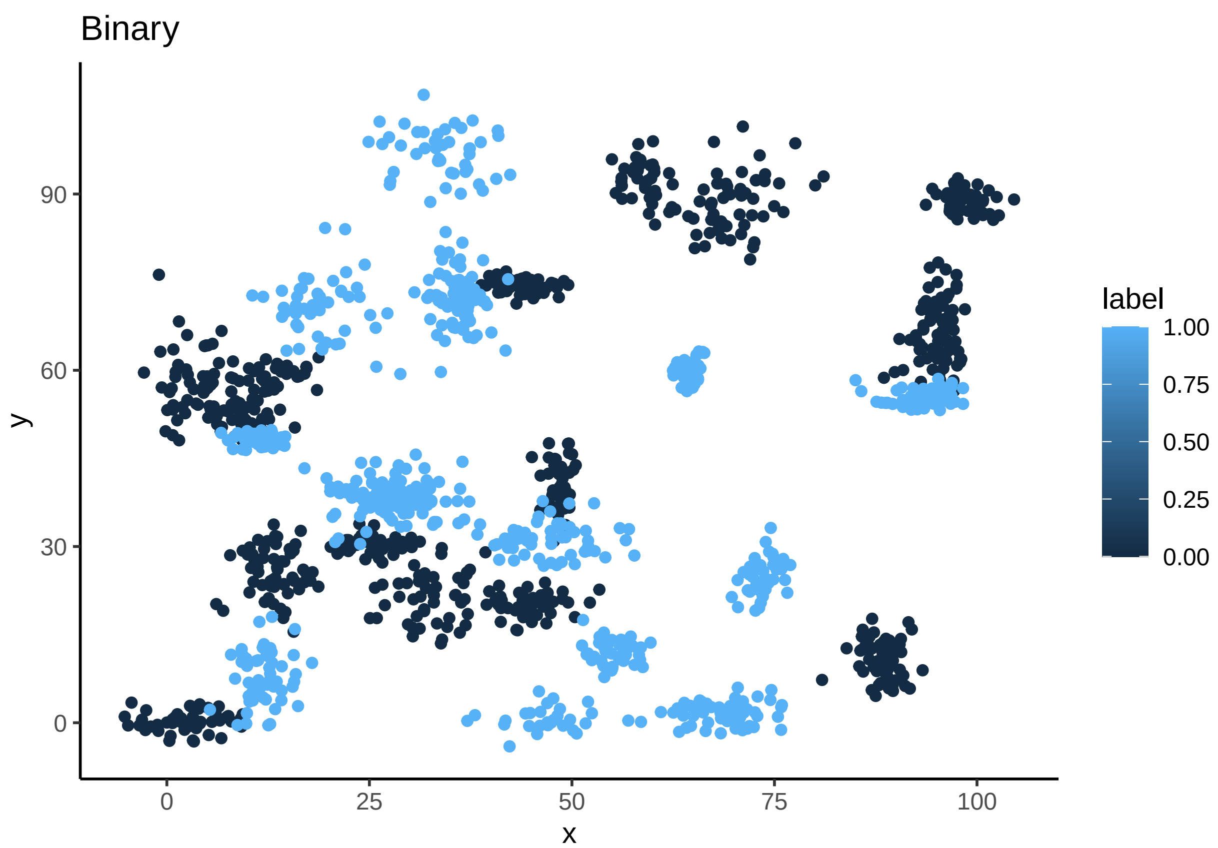
Binary Dataset

```
setwd('/Users/logan/Documents/GitHub/dsc520clone')

binary_class_df <- read.csv('data/binary-classifier-data.csv')

summary(binary_class_df)
```

I.) Plot the data from each dataset using a scatter plot.

```
##      label              x                 y
##  Min.   :0.000   Min.   : -5.20   Min.   : -4.019
##  1st Qu.:0.000   1st Qu.: 19.77   1st Qu.: 21.207
##  Median :0.000   Median : 41.76   Median : 44.632
##  Mean   :0.488   Mean   : 45.07   Mean   : 45.011
##  3rd Qu.:1.000   3rd Qu.: 66.39   3rd Qu.: 68.698
##  Max.   :1.000   Max.   :104.58   Max.   :106.896
```

```
binary_scatter <- ggplot(data=binary_class_df, aes(x=x,y=y, color=label)) + ggtitle('Binary') + geom_poi
binary_scatter
```



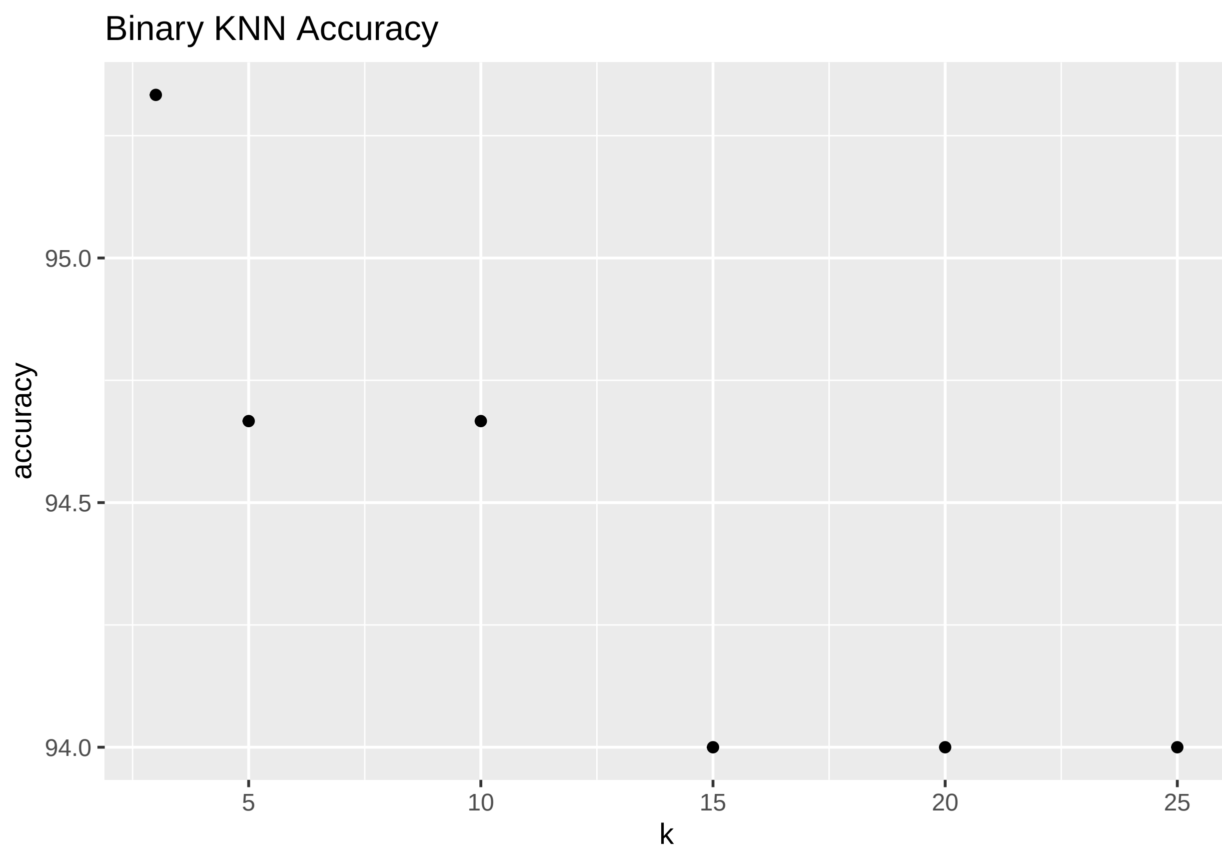

```
ran <- sample(1:nrow(binary_class_df), 0.9 * nrow(binary_class_df))
nor <- function(x) { (x-min(x)/max(x)-min(x))}

binary_norm <- as.data.frame(lapply(binary_class_df[,c(2,3)],nor))

summary(binary_norm)
```

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

```
##        x                  y
##  Min.   :  0.04973   Min.   :  0.0376
##  1st Qu.: 25.01908   1st Qu.: 25.2644
##  Median : 47.00983   Median : 48.6891
##  Mean   : 50.32345   Mean   : 49.0677
##  3rd Qu.: 71.64102   3rd Qu.: 72.7545
##  Max.   :109.82598   Max.   :110.9526
```

```
binary_train <- binary_norm[ran,]
binary_test <- binary_norm[-ran,]

binary_target <- binary_class_df[ran,1]
binary_test_cat <- binary_class_df[-ran,1]

accuracy <- function(x){sum(diag(x)/sum(rowSums(x))) * 100}
```

```r
k_value <- c(3, 5, 10, 15, 20, 25)

bpr3 <- knn(binary_train, binary_test, cl=binary_target, k=3)
bpr5 <- knn(binary_train, binary_test, cl=binary_target, k=5)
bpr10 <- knn(binary_train, binary_test, cl=binary_target, k=10)
bpr15 <- knn(binary_train, binary_test, cl=binary_target, k=15)
bpr20 <- knn(binary_train, binary_test, cl=binary_target, k=20)
bpr25 <- knn(binary_train, binary_test, cl=binary_target, k=25)

btab3 <- table(bpr3, binary_test_cat)
btab5 <- table(bpr5, binary_test_cat)
btab10 <- table(bpr10, binary_test_cat)
btab15 <- table(bpr15, binary_test_cat)
btab20 <- table(bpr20, binary_test_cat)
btab25 <- table(bpr25, binary_test_cat)

bacc3 <- accuracy(btab3)
bacc5 <- accuracy(btab5)
bacc10 <- accuracy(btab10)
bacc15 <- accuracy(btab15)
bacc20 <- accuracy(btab20)
bacc25 <- accuracy(btab25)

bacc_df <- data.frame(k_value, c(bacc3, bacc5, bacc10, bacc15, bacc20, bacc25))
names(bacc_df) <- c('k', 'accuracy')

ggplot(bacc_df, aes(x = k,y = accuracy)) + geom_point() + ggtitle('Binary KNN Accuracy')
```



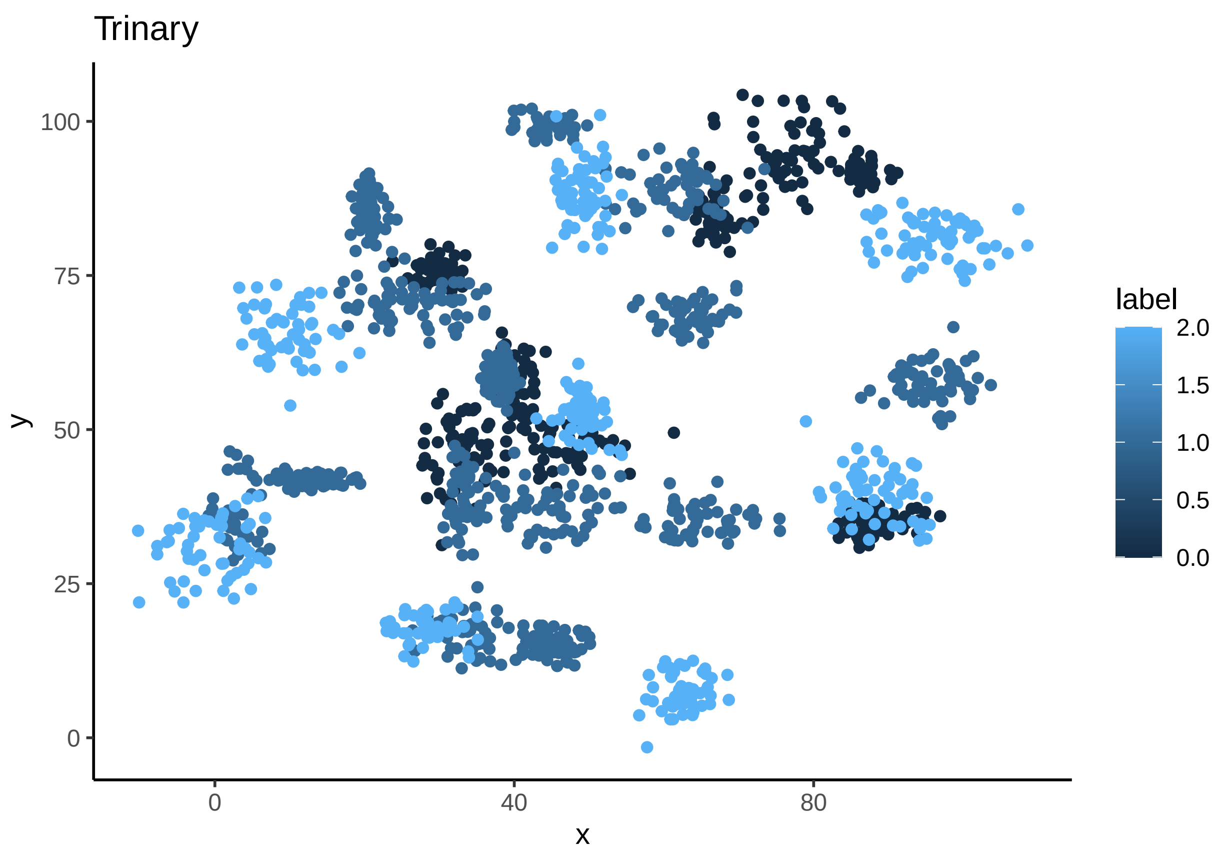
Trinary Dataset

```
trinary_class_df <- read.csv('data/trinary-classifier-data.csv')
summary(trinary_class_df)
```

I.) Plot the data from each dataset using a scatter plot.

```
##      label              x                 y
##  Min.   :0.000   Min.   :-10.26   Min.   : -1.541
##  1st Qu.:0.000   1st Qu.: 31.15   1st Qu.: 35.906
##  Median :1.000   Median : 45.59   Median : 55.073
##  Mean   :1.037   Mean   : 48.86   Mean   : 55.282
##  3rd Qu.:2.000   3rd Qu.: 66.27   3rd Qu.: 77.403
##  Max.   :2.000   Max.   :108.56   Max.   :104.293
```

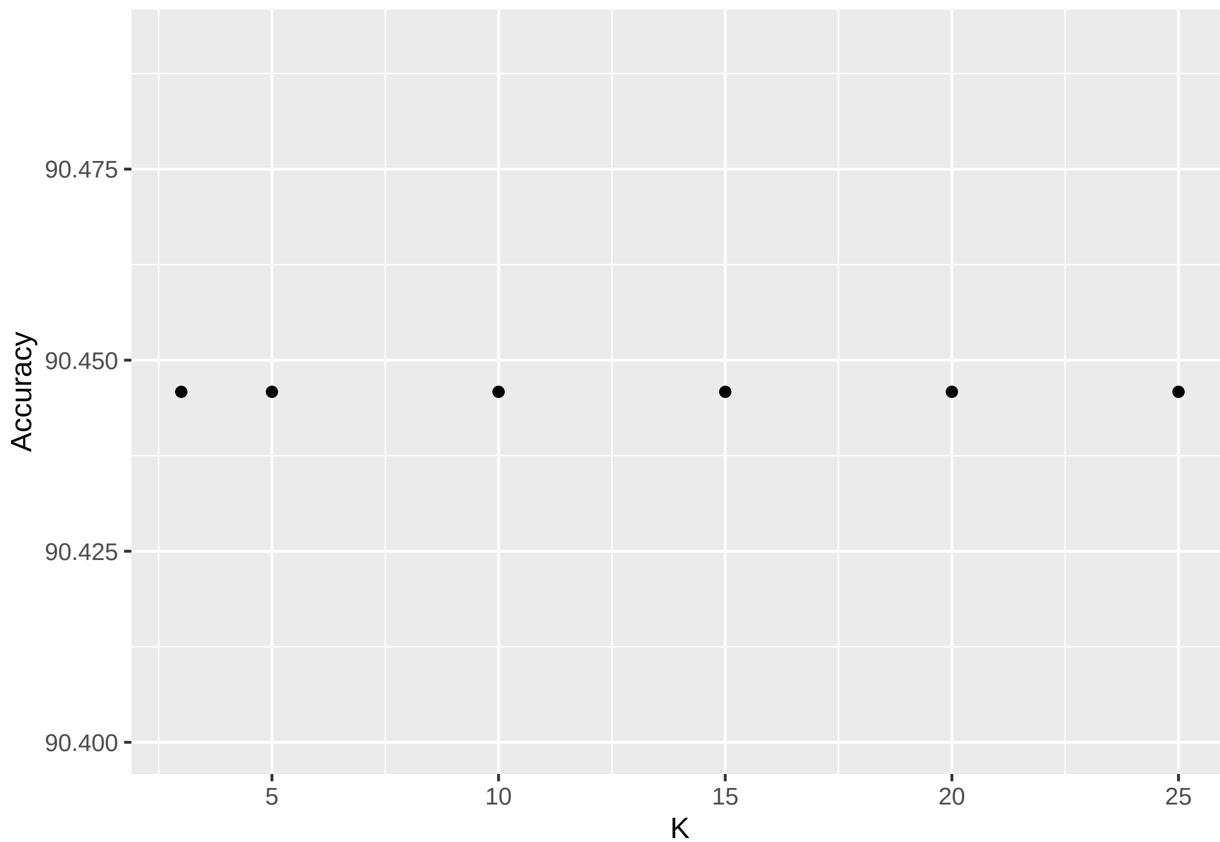
```
trinary_scatter <- ggplot(data=trinary_class_df, aes(x=x, y=y, color=label)) + ggtitle('Trinary') + geor
trinary_scatter
```




```
ran <- sample(1:nrow(trinary_class_df), 0.9 * nrow(trinary_class_df))
nor <- function(x) { (x-min(x)/max(x)-min(x))}
trinary_norm <- as.data.frame(lapply(trinary_class_df[,c(2,3)],nor))

trinary_train <- trinary_norm[ran,]
trinary_test <- trinary_norm[-ran,]

trinary_target <- trinary_class_df[ran,1]
trinary_test_cat <- trinary_class_df[-ran,1]
k_value = c(3, 5, 10, 15, 20, 25)

tpr3 <- knn(trinary_train, trinary_test, cl=trinary_target, k=3)
tpr5 <- knn(trinary_train, trinary_test, cl=trinary_target, k=5)
tpr10 <- knn(trinary_train, trinary_test, cl=trinary_target, k=10)
tpr15 <- knn(trinary_train, trinary_test, cl=trinary_target, k=15)
tpr20 <- knn(trinary_train, trinary_test, cl=trinary_target, k=20)
tpr25 <- knn(trinary_train, trinary_test, cl=trinary_target, k=25)

ttab3 <- table(tpr3, trinary_test_cat)
ttab5 <- table(tpr3, trinary_test_cat)
ttab10 <- table(tpr3, trinary_test_cat)
ttab15 <- table(tpr3, trinary_test_cat)
ttab20 <- table(tpr3, trinary_test_cat)
ttab25 <- table(tpr3, trinary_test_cat)
```

```r
tacc3 <- accuracy(ttab3)
tacc5 <- accuracy(ttab5)
tacc10 <- accuracy(ttab10)
tacc15 <- accuracy(ttab15)
tacc20 <- accuracy(ttab20)
tacc25 <- accuracy(ttab25)


trin_df <- data.frame(k_value, c(tacc3, tacc5, tacc10, tacc15, tacc20, tacc25))
names(trin_df) <- c('K',"Accuracy")
ggplot(trin_df, aes(x = K, y=Accuracy)) + geom_point()
```

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 do not believe that a linear classifer would work well on this data set. this dataset based off the scatterplots.

Last week my model's accuracy was roughly 56.27% accurate these models are 90-96% accurate which is a great improvement.

Clustering

Based off the gap statistic and the optimal number of clusters graph it appears that k=10 is the elbow point and optimal number of clusters.

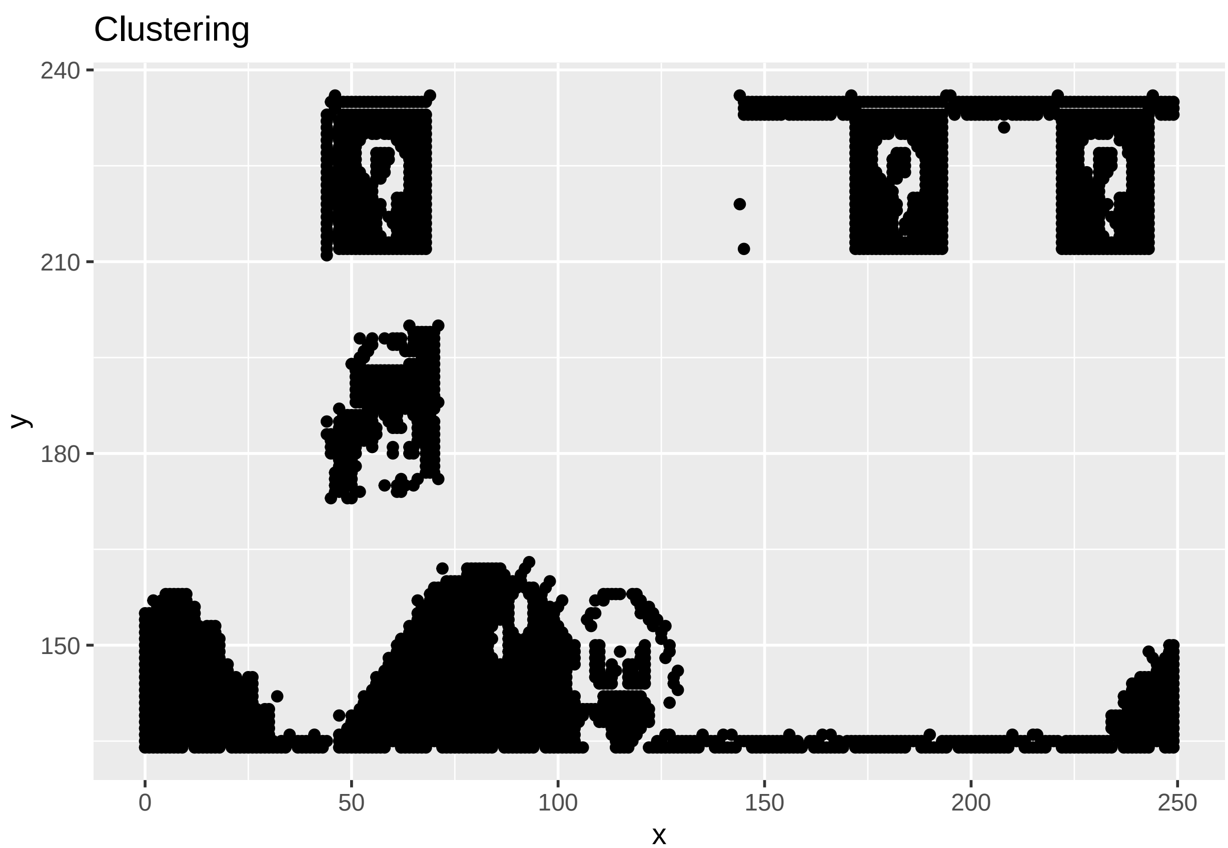
```r
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
clustering_df <- read.csv('data/clustering-data.csv')
summary(clustering_df)
```

```
##        x                y        
##  Min.   :  0.0   Min.   :134.0  
##  1st Qu.: 56.0   1st Qu.:141.0  
##  Median : 82.0   Median :154.0  
##  Mean   :109.6   Mean   :175.7  
##  3rd Qu.:180.0   3rd Qu.:218.0  
##  Max.   :249.0   Max.   :236.0
```

```r
ggplot(clustering_df, aes(x=x, y=y)) + geom_point() + ggtitle('Clustering')
```



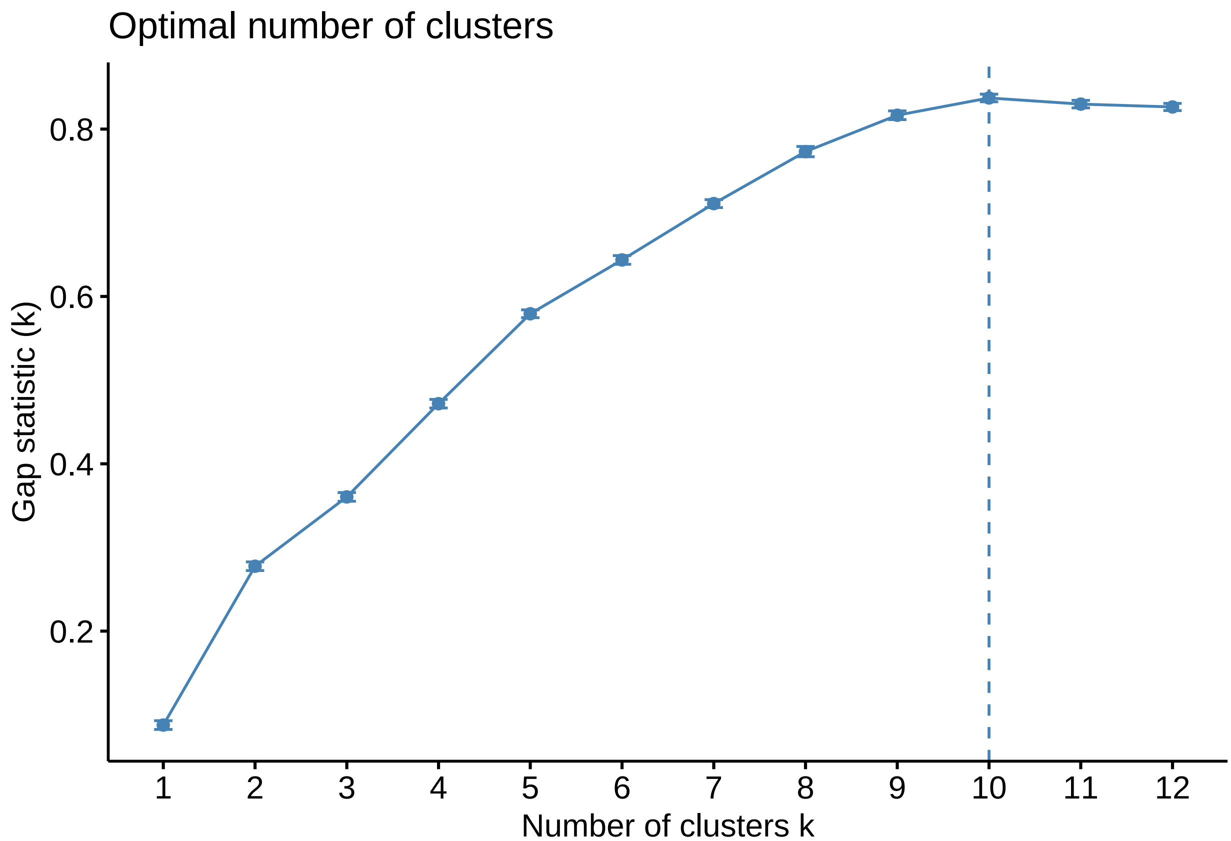

```r
k2 <- kmeans(clustering_df,centers=2, nstart=25)


k3 <- kmeans(clustering_df,centers=3, nstart=25)


k4 <- kmeans(clustering_df,centers=4, nstart=25)
```

```r
k5 <- kmeans(clustering_df,centers=5, nstart=25)

k6 <- kmeans(clustering_df,centers=6, nstart=25)

k7 <- kmeans(clustering_df,centers=7, nstart=25)

k8 <- kmeans(clustering_df,centers=8, nstart=25)

k9 <- kmeans(clustering_df,centers=9, nstart=25)

k10 <- kmeans(clustering_df,centers=10, nstart=25)

k11 <- kmeans(clustering_df,centers=11, nstart=25)

k12 <- kmeans(clustering_df,centers=12, nstart=25)
gap_stat <- clusGap(clustering_df, FUN=kmeans, nstart=25, K.max=12, B=50)
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

```
## Warning: did not converge in 10 iterations
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 201100)
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 201100)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 201100)
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

```
fviz_gap_stat(gap_stat)
```


Optimal number of clusters
Gap statistic (k)
0.8
0.6
0.4
0.2
1
2
3
4
5
6
7
8
9
10
11
12
Number of clusters k