# Final4063-Shantal-Cruz

```r
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=80), tidy=TRUE)

library(readr)
library(ROCR)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(caTools)
library(class)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(factoextra)
```

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

```r
data <- readr::read_csv("DatasetforFINAL.csv")
```

```
## New names:
## * `` -> `...1`

## Rows: 5000 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Fname, Lname, gender, City, boughtelectronics
## dbl (5): ...1, ID, FamilyIncome, EdYears, FamilySize
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# City Toronto only
myCity <- data[data$City == "Vancouver", ]

# View(myCity)

############################################
# 1) Divide the data set to Train and Test Data Sets
# 70% of the data for training and 30% for testing

set.seed(4063)
spl <- sample.split(myCity$boughtelectronics, SplitRatio = 0.7)

train <- myCity[spl == TRUE, ]
test <- myCity[spl == FALSE, ]
train
```

## A tibble: 429 x 10

```
...1    ID Fname   Lname        gender City  FamilyIncome EdYears FamilySize
```

1 21 21 Maleah Kettler F Vanc~ 34133 11 2 2 22 22 Cody Russell M Vanc~ 30157 15 2 3 30 30 Rachel Hollingswor~ F Vanc~ 24902 12 4 4 37 37 Brecken Cobb F Vanc~ 30669 10 5 5 46 46 Rowan Dodson M Vanc~ 33854 10 3 6 72 72 Lance Dixon M Vanc~ 16211 12 5 7 78 78 Zachary Guida M Vanc~ 39088 11 4 8 79 79 Rachel Gibbons F Vanc~ 29304 10 4 9 80 80 Cody Knapp M Vanc~ 60816 11 3 10 108 108 Rheanna Patten F Vanc~ 27940 12 3 # i 419 more rows # i 1 more variable: boughtelectronics

```r
############################################
# 2a) Use the train data and create a Naive Bayes Classifier for predicting whether the customer will b

library(e1071)
nb <- e1071::naiveBayes(boughtelectronics ~ FamilyIncome + FamilySize, data = train)

# 2b) Use the test data and your model and make predictions regarding whether the customer will buy ele

nb_pred <- predict(nb, newdata = test)
nb_pred
```

[1] YES NO NO NO NO NO NO YES NO NO NO NO YES YES NO NO YES NO [19] NO NO NO NO NO YES NO YES NO YES NO NO NO NO NO NO NO NO [37] NO YES NO YES NO NO NO NO NO NO YES YES YES NO NO YES NO NO [55] NO NO NO NO NO NO NO NO NO NO NO NO NO YES NO NO NO NO [73] NO NO NO NO NO YES NO NO NO NO NO YES YES NO NO NO NO NO [91] NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO [109] NO NO NO NO NO NO NO NO NO NO YES NO NO NO NO NO NO NO [127] NO NO NO NO YES YES NO NO NO YES YES NO NO NO NO NO NO NO [145] NO NO NO YES NO NO YES NO NO NO NO YES NO NO NO NO NO NO [163] NO NO NO YES NO NO NO NO NO NO NO NO NO NO YES YES NO NO [181] NO NO NO Levels: NO YES

```r
############################################
# 3a) Use the train data and K-Nearest Neighbor Classifier for predicting whether the customer will buy

# determine best k first
```

```r
best_accuracy <- 0
best_k <- 0
for (i in 1:10) {
    knn_model <- caret::knn3(x = train[, c("EdYears", "FamilySize")], y = factor(train$boughtelectronics

    knn_pred <- predict(knn_model, newdata = test[, c("EdYears", "FamilySize")], type = "class")

    accuracy <- mean(knn_pred == test$boughtelectronics)
    # print(paste("Accuracy is", accuracy, "for k =", i))
    if (accuracy > best_accuracy) {
        best_accuracy <- accuracy
        best_k <- i
    }
}
# plot(test$EdYears, test$FamilySize, col = factor(test$boughtelectronics), pch = 19, cex = 5, main = p
# print(paste("Best accuracy is k =", best_k))

knn_model <- caret::knn3(x = train[, c("EdYears", "FamilySize")], y = factor(train$boughtelectronics), k
knn_model
```

5-nearest neighbor model Training set outcome distribution:

NO YES 330 99

```r
# 3b) Use the test data and your model and make predictions regarding whether the customer will buy ele

knn_pred <- predict(knn_model, newdata = test[, c("EdYears", "FamilySize")], type = "class")
knn_pred
```

[1] NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO [19] NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO [37] NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO YES NO NO [55] NO NO NO NO NO NO NO NO NO NO NO NO YES NO NO NO NO NO [73] NO NO NO NO YES YES NO NO NO NO NO NO NO NO NO NO NO NO [91] NO NO NO NO NO NO NO NO NO NO NO NO NO YES NO NO NO NO [109] NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO YES [127] NO NO NO YES NO NO NO NO NO YES NO NO NO NO NO NO YES NO [145] NO NO YES NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO [163] NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO NO YES [181] NO NO NO Levels: NO YES

```r
############################################
# 4a) Compose the confusion matrix of   Naive Bayes Classifier
nb_cm <- caret::confusionMatrix(nb_pred, as.factor(test$boughtelectronics))
nb_cm
```

Confusion Matrix and Statistics

```
      Reference
```

Prediction NO YES NO 137 17 YES 4 25

```
           Accuracy : 0.8852
             95% CI : (0.8299, 0.9275)
No Information Rate : 0.7705
```

```
P-Value [Acc > NIR] : 5.657e-05

              Kappa : 0.636
```

Mcnemar's Test P-Value : 0.008829

```
        Sensitivity : 0.9716
        Specificity : 0.5952
     Pos Pred Value : 0.8896
     Neg Pred Value : 0.8621
         Prevalence : 0.7705
     Detection Rate : 0.7486
```

Detection Prevalence : 0.8415
Balanced Accuracy : 0.7834

```
   'Positive' Class : NO
```

```r
# Compose the confusion matrix of K-Nearest Neighbor
knn_cm <- caret::confusionMatrix(knn_pred, as.factor(test$boughtelectronics))
knn_cm
```

Confusion Matrix and Statistics

```
      Reference
```

Prediction NO YES NO 137 35 YES 4 7

```
           Accuracy : 0.7869
             95% CI : (0.7204, 0.8438)
No Information Rate : 0.7705
P-Value [Acc > NIR] : 0.3349

              Kappa : 0.1867
```

Mcnemar's Test P-Value : 1.556e-06

```
        Sensitivity : 0.9716
        Specificity : 0.1667
     Pos Pred Value : 0.7965
     Neg Pred Value : 0.6364
         Prevalence : 0.7705
     Detection Rate : 0.7486
```

Detection Prevalence : 0.9399
Balanced Accuracy : 0.5691

```
   'Positive' Class : NO
```

```r
# Decide which model has better accuracy.
print(paste("Naive Bayes accuracy:", nb_cm$overall["Accuracy"]))
```

[1] "Naive Bayes accuracy: 0.885245901639344"

```r
print(paste("KNN accuracy:", knn_cm$overall["Accuracy"]))
```

[1] "KNN accuracy: 0.786885245901639"

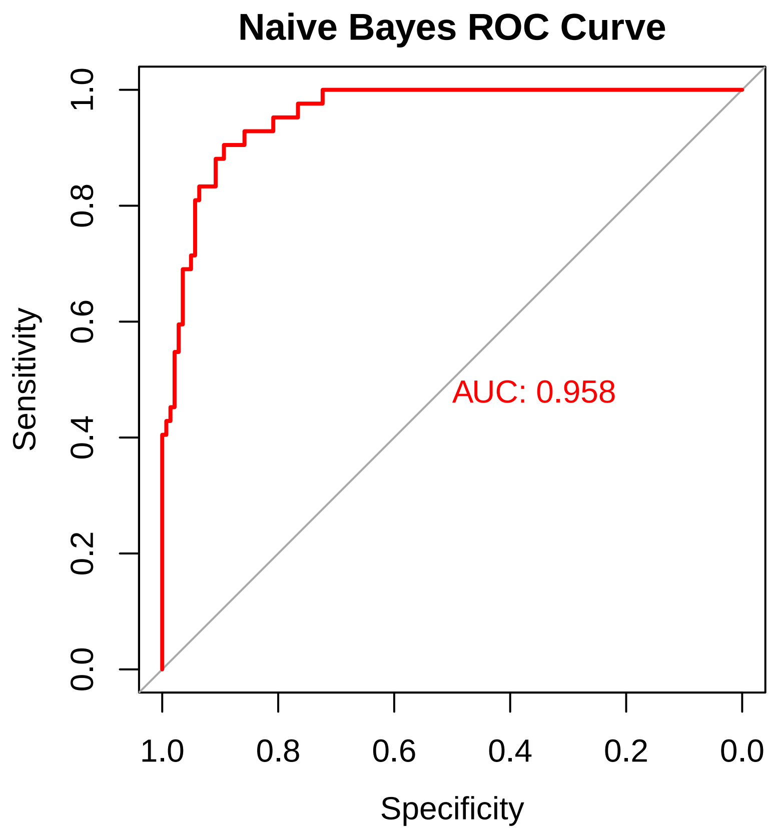
```r
print("Naive Bayes has better accuracy than K-Nearest Neighbor.")
```

[1] "Naive Bayes has better accuracy than K-Nearest Neighbor."

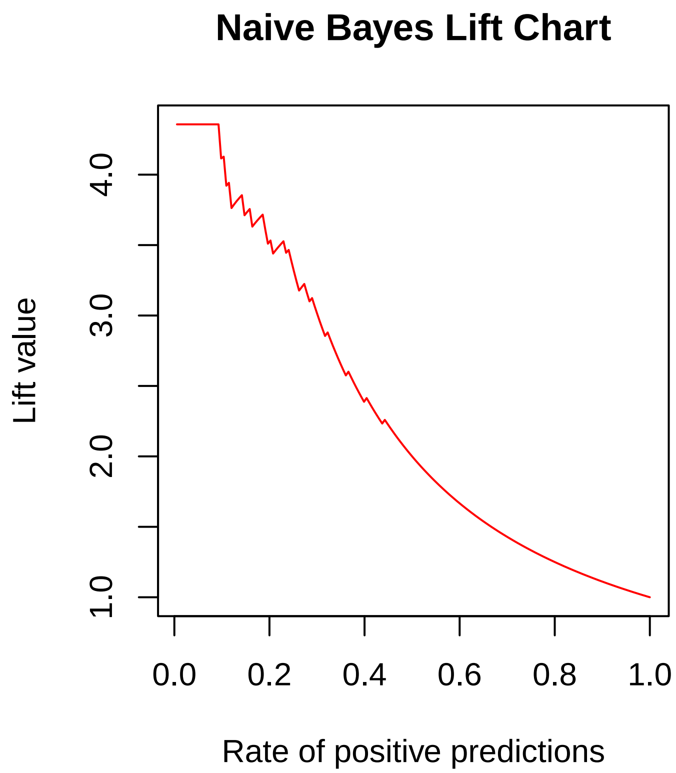
```r
############################################
# 4b) Compose the ROC, gain and lift charts of the Naive Bayes model

# get probability of buying electronics
nb_pred_prob <- predict(nb, newdata = test, type = "raw")[, 2]

par(pty = "s")

nb_roc <- pROC::roc(test$boughtelectronics, nb_pred_prob)
```

```
## Setting levels: control = NO, case = YES
## Setting direction: controls < cases
```

```r
plot(nb_roc, col = "red", main = "Naive Bayes ROC Curve", print.auc = TRUE)
```



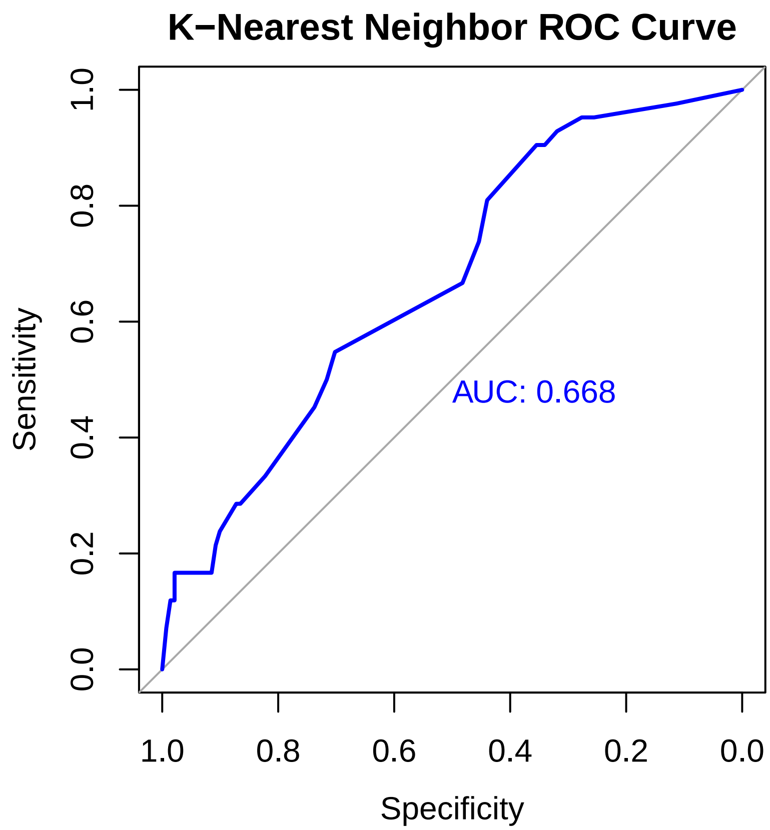

```
nb_prediction <- prediction(nb_pred_prob, test$boughtelectronics)
nb_perf <- performance(nb_prediction, "lift", x.measure = "rpp")
plot(nb_perf, col = "red", main = "Naive Bayes Lift Chart")
```



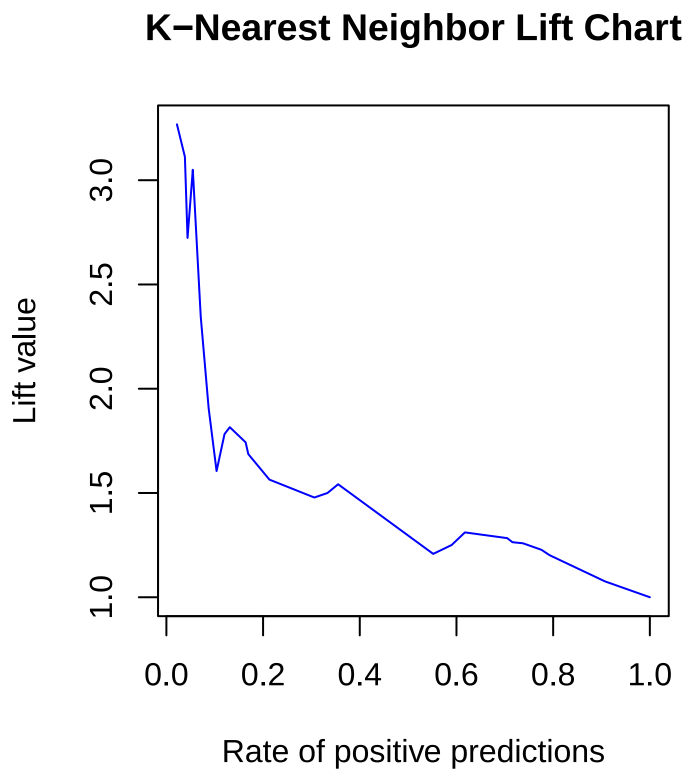

```r
# Compose the ROC, gain and lift charts of the K-Nearest Neighbor model
knn_pred_prob <- predict(knn_model, newdata = test[, c("EdYears", "FamilySize")])[, 2]

knn_roc <- pROC::roc(test$boughtelectronics, knn_pred_prob)
```

```
## Setting levels: control = NO, case = YES
## Setting direction: controls < cases
```

```r
plot(knn_roc, col = "blue", main = "K-Nearest Neighbor ROC Curve", print.auc = TRUE)
```




```
knn_prediction <- prediction(knn_pred_prob, test$boughtelectronics)
knn_perf <- performance(knn_prediction, "lift", x.measure = "rpp")
plot(knn_perf, col = "blue", main = "K-Nearest Neighbor Lift Chart")
```



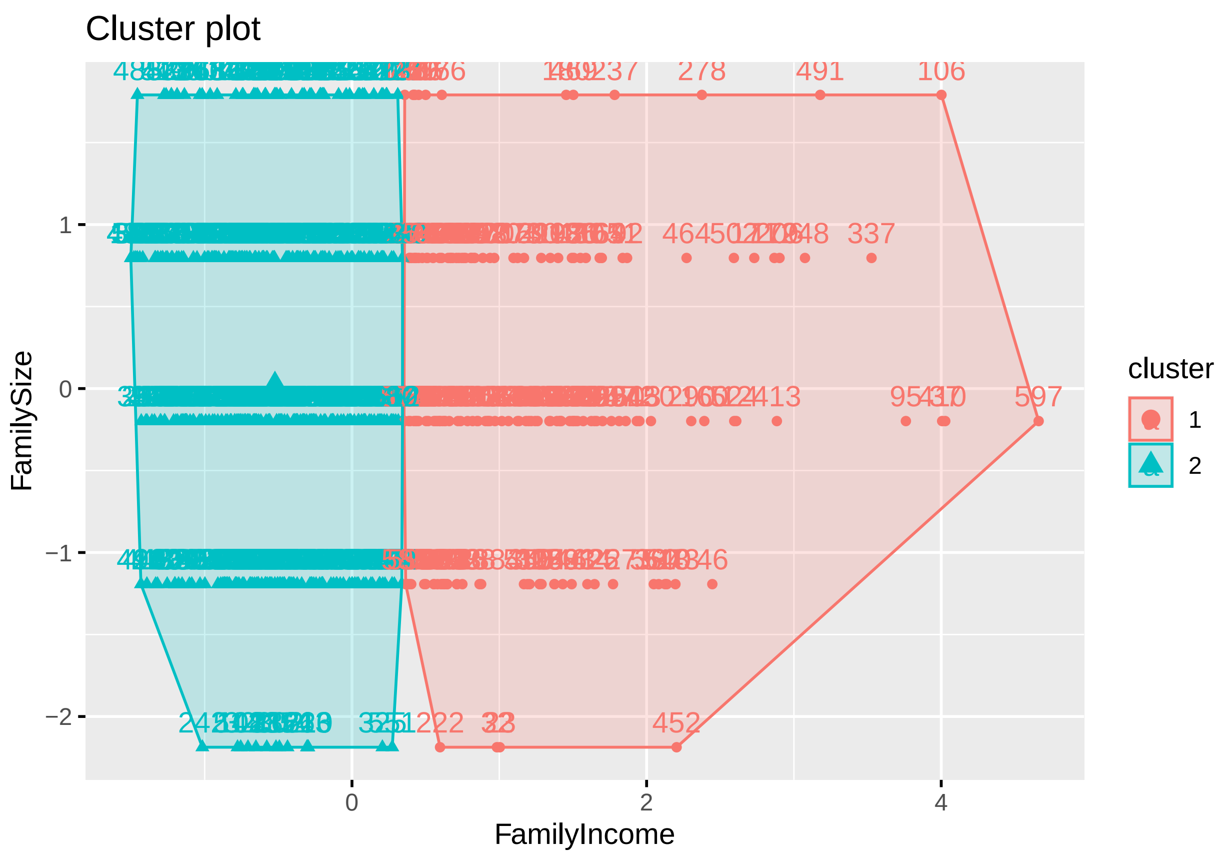

```
# Which model is better? Argue why?
print("Naive Bayes is better because it has a higher AUC and higher lift than the K-Nearest Neighbor mod
```

[1] "Naive Bayes is better because it has a higher AUC and higher lift than the K-Nearest Neighbor model. The Naive Bayes model also has a higher accuracy than the K-Nearest Neighbor model. A higher AUC means that the model is better at distinguishing between the two classes. A higher lift means that the model is better at predicting the positive class."

```
############################################
# 5) Use k-means clustering, what would be an optimum model that would cluster the buyers based on fami


# silhouette method
km_s <- factoextra::fviz_nbclust(myCity[, c("FamilyIncome", "FamilySize")], FUNcluster = kmeans, method
km_s <- km_s$data
km_optimal_k <- as.numeric(km_s$clusters[which.max(km_s$y)])
print(paste("Optimum number of k-means clusters is", km_optimal_k))
```

[1] "Optimum number of k-means clusters is 2"

```
# K-Means Clustering
km <- stats::kmeans(myCity[, c("FamilyIncome", "FamilySize")], centers = km_optimal_k)

# plot kmeans nicely
km_cluster <- factoextra::fviz_cluster(list(data = myCity[, c("FamilyIncome", "FamilySize")], cluster =
print(km_cluster)
```



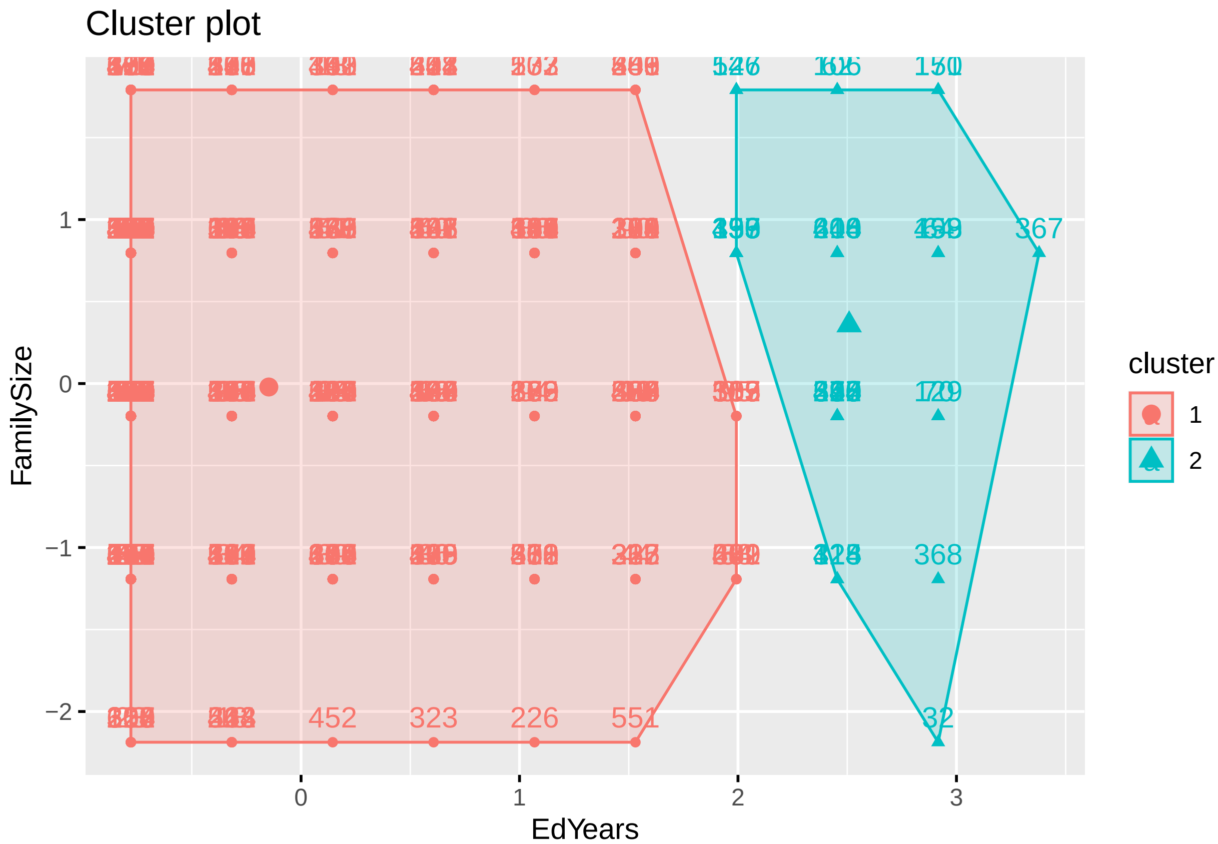

```
##############################################
# 6) Using Hierarchical Clustering what is optimum number of  clusters of customers do you detect based
hc_s <- factoextra::fviz_nbclust(myCity[, c("EdYears", "FamilySize")], FUNcluster = hcut, method = "sil
hc_s <- hc_s$data
hc_optimal_k <- as.numeric(hc_s$clusters[which.max(hc_s$y)])

print(paste("Optimum number of heirarchical clusters is", hc_optimal_k))
```

[1] "Optimum number of heirarchical clusters is 2"

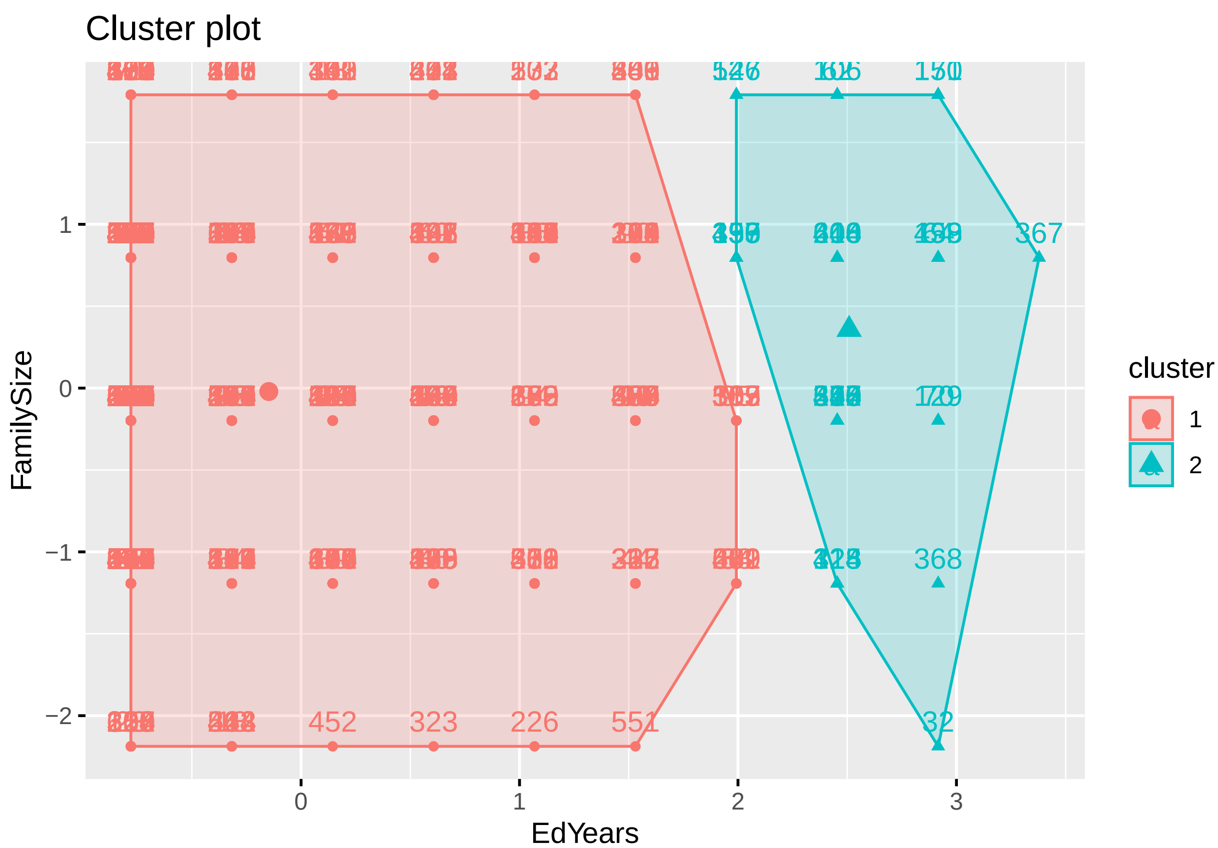
```
# Hierarchical Clustering
hc <- stats::hclust(dist(myCity[, c("EdYears", "FamilySize")]))
cut_tree <- cutree(hc, k = hc_optimal_k)

# Visualize the dendrogram
dend <- as.dendrogram(hc)
print(dend)
```

'dendrogram' with 2 branches and 612 members total, at height 9.486833

```
# Create a clustering object using cut_tree
hc_cluster <- factoextra::fviz_cluster(list(data = myCity[, c("EdYears", "FamilySize")], cluster = cut_
hc_cluster
```




```r
# Plot the clustering object
print(hc_cluster)
```



Cluster plot
FamilySize
EdYears
cluster
1
2