DSC441 Project: Heart Attack Analysis

Erik Pak

2025-04-24

Overview

This is data set from kaggle for **Heart Attack Analysis & Prediction Dataset**

Data file used for Analysis : Heart_Attack_Data.csv

Link : https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

This dataset contains various features related to individuals, such as age, gender, cholesterol levels, blood pressure, and other health-related attributes. Based on these factors, the dataset aims to analyze and predict the likelihood of a heart attack.

As per of our analysis here we will try to find a models predictive power of patient having chances of having a heart attack or not using various techniques of machine learning.

Data Desription

The Data consists of below described variables

1. Age : Age of the patient
2. Sex : Sex of the patient
3. exng: exercise induced angina (1 = yes; 0 = no)
4. caa: number of major vessels (0-3)
5. cp : Chest Pain type
   Value 1: typical angina
   Value 2: atypical angina
   Value 3: non-anginal pain
   Value 4: asymptomatic
6. trtbps : resting blood pressure (in mm Hg)
7. chol : cholesterol in mg/dl fetched via BMI sensor
8. fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
9. restecg : resting electrocardiographic results
   Value 0: normal
   Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
   Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
10. thalachh : maximum heart rate achieved
11. oldpeak : Previous peak information
12. thall : Thall Rate

13. slp : slope details

14. output : 0= less chance of heart attack 1= more chance of heart attack

output is our Response/target variable

a) Data Gathering and Integration

The Data is loaded from the csv file.

The Data consists of 303 records and includes both numerical and categorical/ordinal variables(represented as 1/0 and scale).

```
# load the data
haData <- read.csv("./Data/Heart_Attack_Data.csv")

# count of records
nrow(haData)
```

```
## [1] 303
```

b) Data Exploration

In order to explore the data we first look at the description statistics with distributions.

From the data looking at the distribution of each variable we see few variables at a different scale like age, trtbps, chol, thalachh. The distribution shows few normal distribution like age, trtbps, chol, thalachh(approx) and some right skewed distribution like oldpeak.

We look at the consolidated pairs panel plot to somewhat understand the distribution of the data.

```
# summary of data
summary(haData)
```

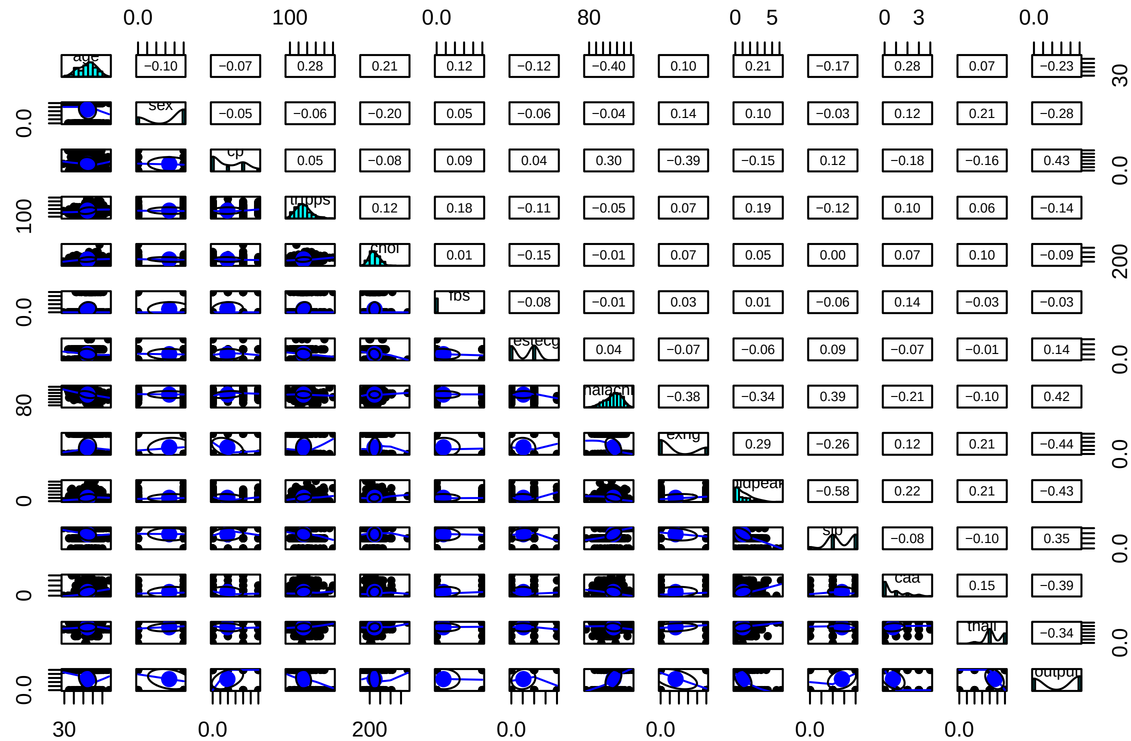
```
##       age             sex               cp            trtbps     
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0  
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0  
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0  
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6  
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0  
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0  
##       chol            fbs            restecg          thalachh    
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0  
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5  
##  Median :240.0   Median :0.0000   Median :1.0000   Median :153.0  
##  Mean   :246.3   Mean   :0.1485   Mean   :0.5281   Mean   :149.6  
##  3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0  
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0  
##       exng           oldpeak          slp             caa        
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000  
##  Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294  
```

```
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000  
##      thall           output      
##  Min.   :0.000   Min.   :0.0000  
##  1st Qu.:2.000   1st Qu.:0.0000  
##  Median :2.000   Median :1.0000  
##  Mean   :2.314   Mean   :0.5446  
##  3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :3.000   Max.   :1.0000
```

```r
# describe data
describe(haData)
```

```
##          vars   n   mean    sd median trimmed   mad min   max range  skew
## age         1 303  54.37  9.08   55.0   54.54 10.38  29  77.0  48.0 -0.20
## sex         2 303   0.68  0.47    1.0    0.73  0.00   0   1.0   1.0 -0.78
## cp          3 303   0.97  1.03    1.0    0.86  1.48   0   3.0   3.0  0.48
## trtbps      4 303 131.62 17.54  130.0  130.44 14.83  94 200.0 106.0  0.71
## chol        5 303 246.26 51.83  240.0  243.49 47.44 126 564.0 438.0  1.13
## fbs         6 303   0.15  0.36    0.0    0.06  0.00   0   1.0   1.0  1.97
## restecg     7 303   0.53  0.53    1.0    0.52  0.00   0   2.0   2.0  0.16
## thalachh    8 303 149.65 22.91  153.0  150.98 22.24  71 202.0 131.0 -0.53
## exng        9 303   0.33  0.47    0.0    0.28  0.00   0   1.0   1.0  0.74
## oldpeak    10 303   1.04  1.16    0.8    0.86  1.19   0   6.2   6.2  1.26
## slp        11 303   1.40  0.62    1.0    1.46  1.48   0   2.0   2.0 -0.50
## caa        12 303   0.73  1.02    0.0    0.54  0.00   0   4.0   4.0  1.30
## thall      13 303   2.31  0.61    2.0    2.36  0.00   0   3.0   3.0 -0.47
## output     14 303   0.54  0.50    1.0    0.56  0.00   0   1.0   1.0 -0.18
##          kurtosis   se
## age         -0.57 0.52
## sex         -1.39 0.03
## cp          -1.21 0.06
## trtbps       0.87 1.01
## chol         4.36 2.98
## fbs          1.88 0.02
## restecg     -1.37 0.03
## thalachh    -0.10 1.32
## exng        -1.46 0.03
## oldpeak      1.50 0.07
## slp         -0.65 0.04
## caa          0.78 0.06
## thall        0.25 0.04
## output      -1.97 0.03
```

```r
# plot of complete data
pairs.panels(haData)
```



b) (i) Visualization

In order to visualize the data we convert few of the binary and continuous variables to meaningful values .

From the visualization we see the male ratio is more in almost all variable comparisons and kind of gives us a visualization of Heart attack chances are more in Male , this is probably because of the proportion of male/female in the data , but we will not be looking at any regression technique here as we want to analyze and predict the likelihood of a heart attack.

From the bar plot of Output we see that is data is almost evenly balanced, so we do not need to perform any oversampling/under-sampling techniques.

The box plot of output vs age shows data of heart attack happening more in age range 44 ~ 58.

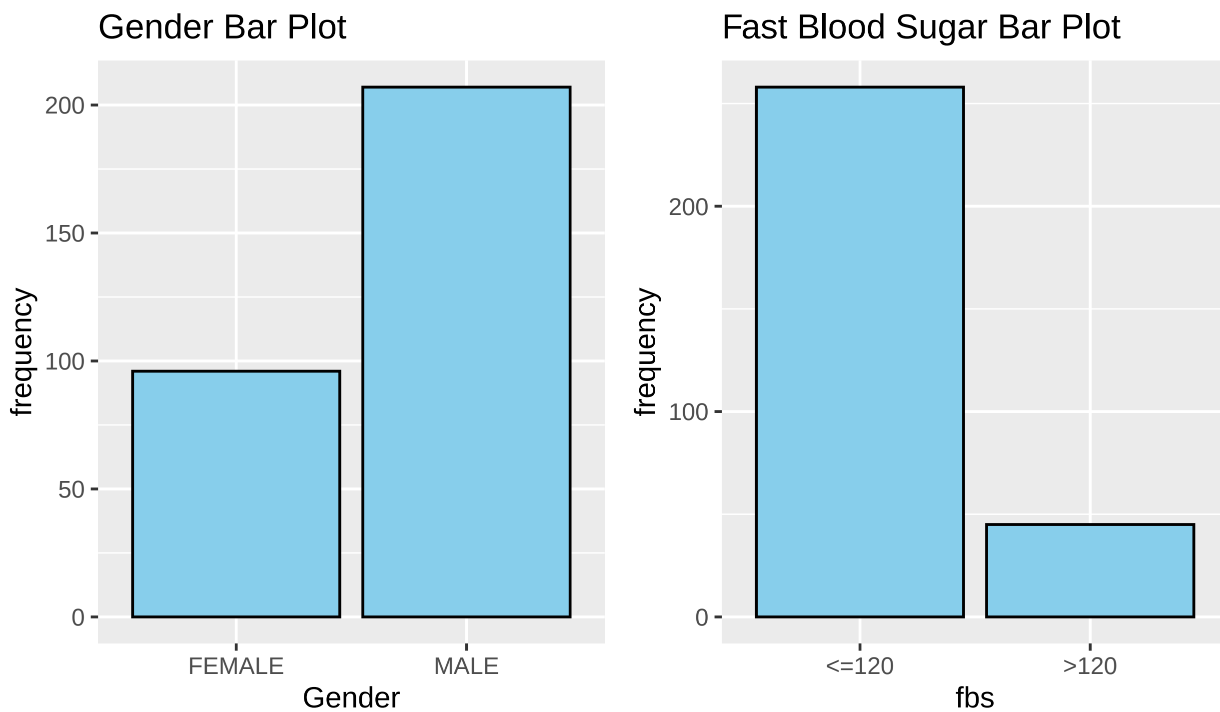
Factors like chest pain, fast blood sugar are all high in males.

```
# copy the data for visualization
haData_vis <- haData %>% mutate(fbs = if_else(fbs == 1, ">120", "<=120"),
                                sex = if_else(sex == 1, "MALE", "FEMALE"),
                                exng = if_else(exng == 1, "YES" ,"NO"),
                                cp = if_else(cp == 1, "ATYPICA_ANGINA",
                                             if_else(cp == 2, "NON-ANGINAL_PAIN"
                                                     , "ASYMPTOMATIC")),
                                restecg = if_else(restecg == 0, "NORMAL",
                                                  if_else(restecg == 1,
                                                          "ABNORMALITY",
                                                          "PROBABLE_OR_DEFINITE")),
```

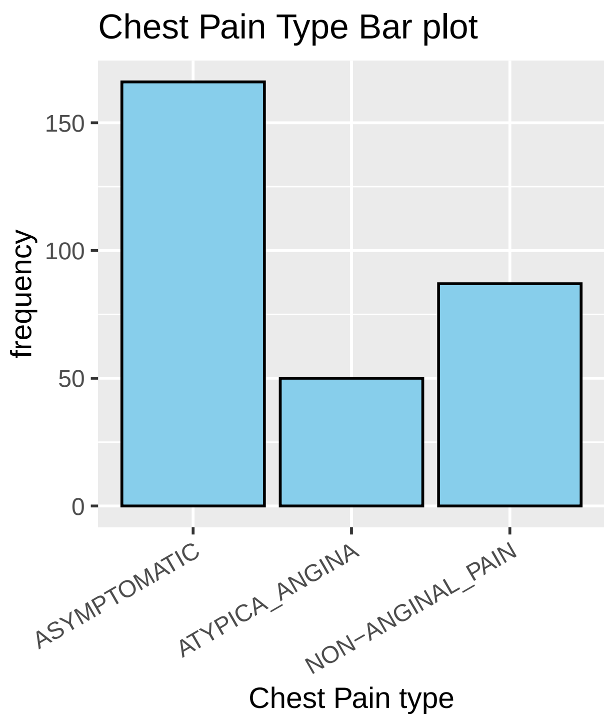
```r
                                  output = if_else(output == 1, "ATTACK", "NoATTACK"))


# summary of the data
summary(haData_vis)
```

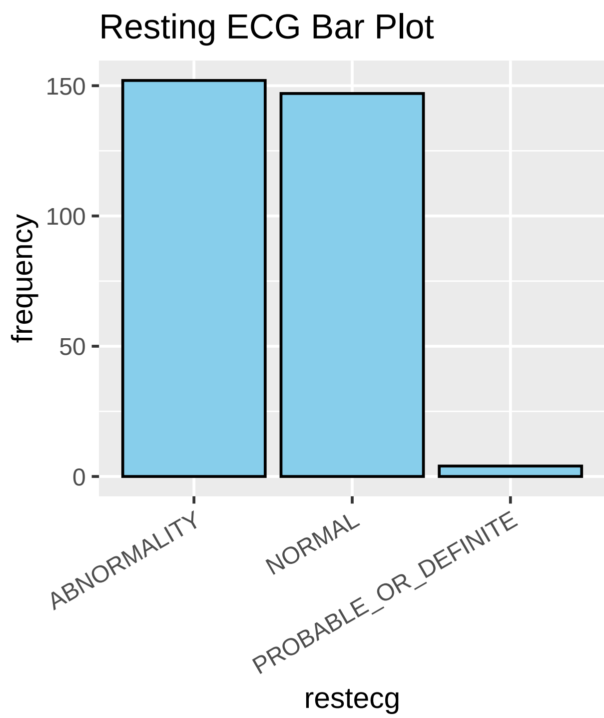
```
##       age            sex                  cp                trtbps     
##  Min.   :29.00   Length:303         Length:303         Min.   : 94.0  
##  1st Qu.:47.50   Class :character   Class :character   1st Qu.:120.0  
##  Median :55.00   Mode  :character   Mode  :character   Median :130.0  
##  Mean   :54.37                                         Mean   :131.6  
##  3rd Qu.:61.00                                         3rd Qu.:140.0  
##  Max.   :77.00                                         Max.   :200.0  
##       chol           fbs              restecg             thalachh    
##  Min.   :126.0   Length:303         Length:303         Min.   : 71.0  
##  1st Qu.:211.0   Class :character   Class :character   1st Qu.:133.5  
##  Median :240.0   Mode  :character   Mode  :character   Median :153.0  
##  Mean   :246.3                                         Mean   :149.6  
##  3rd Qu.:274.5                                         3rd Qu.:166.0  
##  Max.   :564.0                                         Max.   :202.0  
##      exng              oldpeak          slp             caa        
##  Length:303         Min.   :0.00   Min.   :0.000   Min.   :0.0000  
##  Class :character   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000  
##  Mode  :character   Median :0.80   Median :1.000   Median :0.0000  
##                     Mean   :1.04   Mean   :1.399   Mean   :0.7294  
##                     3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000  
##                     Max.   :6.20   Max.   :2.000   Max.   :4.0000  
##      thall          output         
##  Min.   :0.000   Length:303        
##  1st Qu.:2.000   Class :character  
##  Median :2.000   Mode  :character  
##  Mean   :2.314                     
##  3rd Qu.:3.000                     
##  Max.   :3.000
```

```r
# male female distribution
sex_bar <- ggplot(haData_vis, aes(x = factor(sex))) +
              geom_bar(color="black", fill="skyblue") +
              labs(title = "Gender Bar Plot", x = "Gender", y = 'frequency')

# fbs distribution
fbs_bar <- ggplot(haData_vis, aes(x = factor(fbs))) +
              geom_bar(color="black", fill="skyblue") +
              labs(title = "Fast Blood Sugar Bar Plot", x = "fbs", y = 'frequency')

# show the plot
ggarrange(sex_bar, fbs_bar)
```

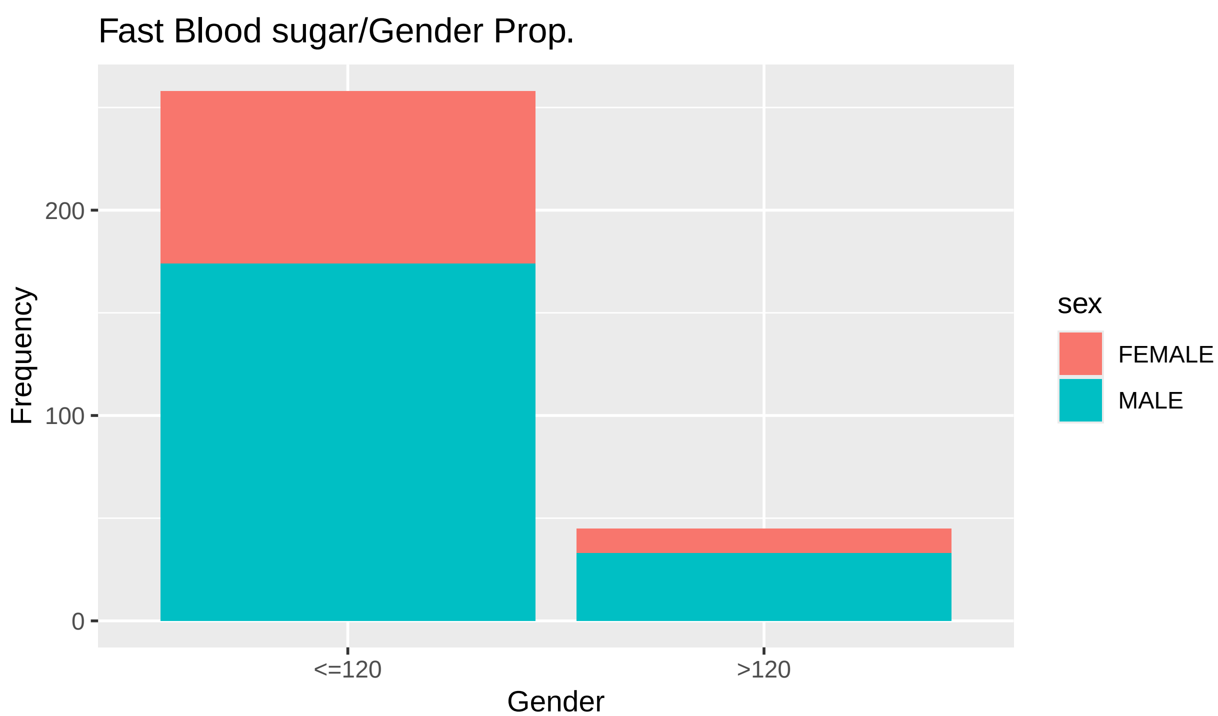



```
# cp bar plot
cp_bar <- ggplot(haData_vis, aes(x = factor(cp))) +
              geom_bar(color="black", fill="skyblue") +
              labs(title = "Chest Pain Type Bar plot", x = "Chest Pain type", y = 'frequency') +
              theme(axis.text.x = element_text(angle = 30,hjust = 1))

# rest ecg bar plot
restecg_bar <- ggplot(haData_vis, aes(x = factor(restecg))) +
                geom_bar(color="black", fill="skyblue") +
                labs(title = "Resting ECG Bar Plot", x = "restecg", y = 'frequency')+
                theme(axis.text.x = element_text(angle = 30,hjust = 1))

# show the plot
ggarrange(cp_bar, restecg_bar)
```

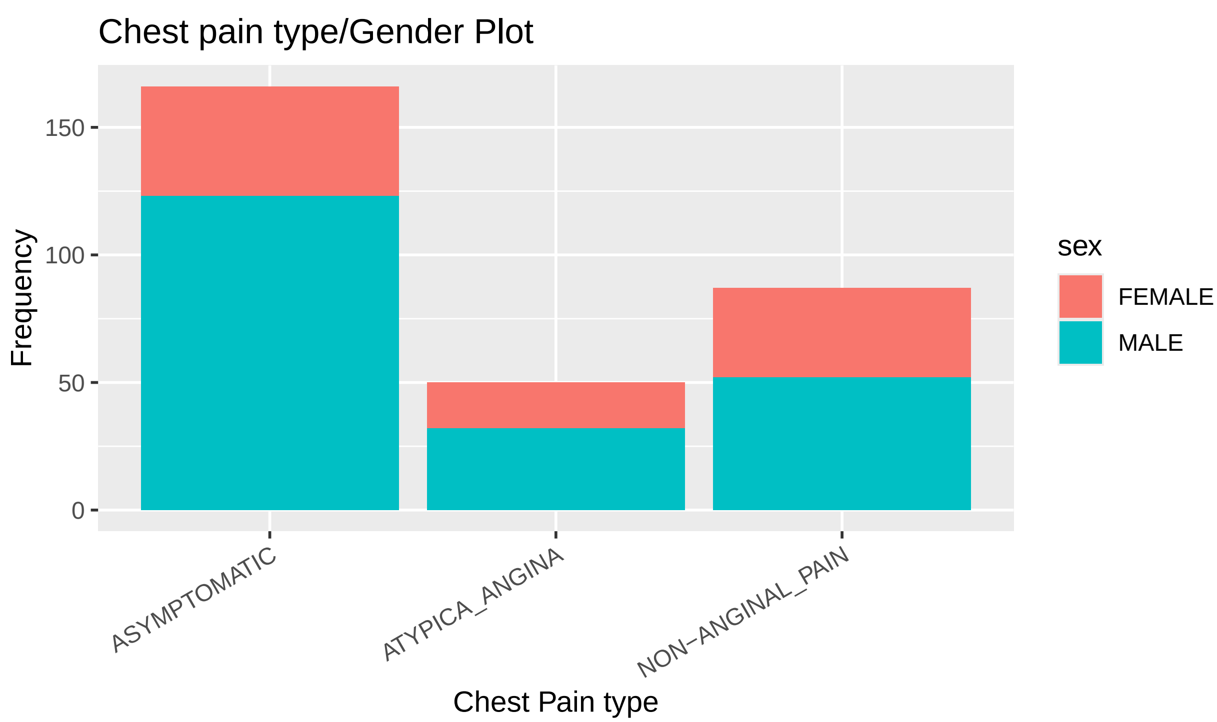




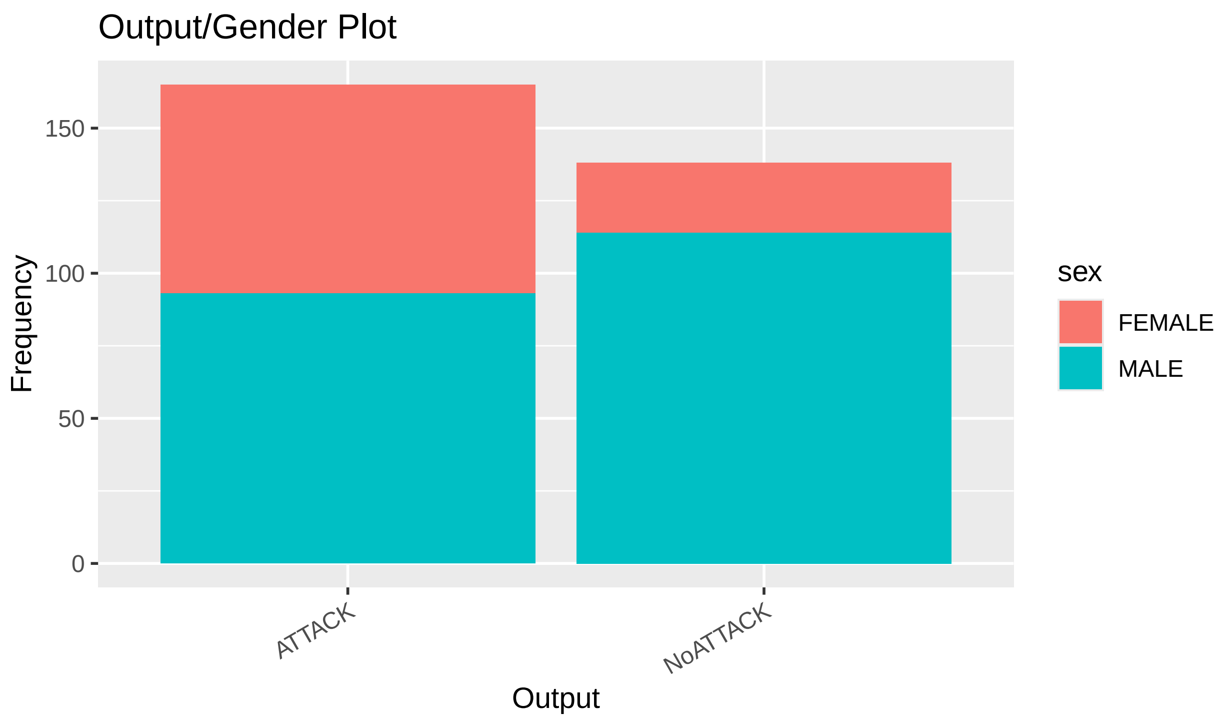

```r
# fasting blood sugar to gender
ggplot(haData_vis, aes(x=fbs, fill=sex)) +
        geom_bar(position="stack") +
        labs(title = "Fast Blood sugar/Gender Prop.", x = "Gender", y = 'Frequency')
```



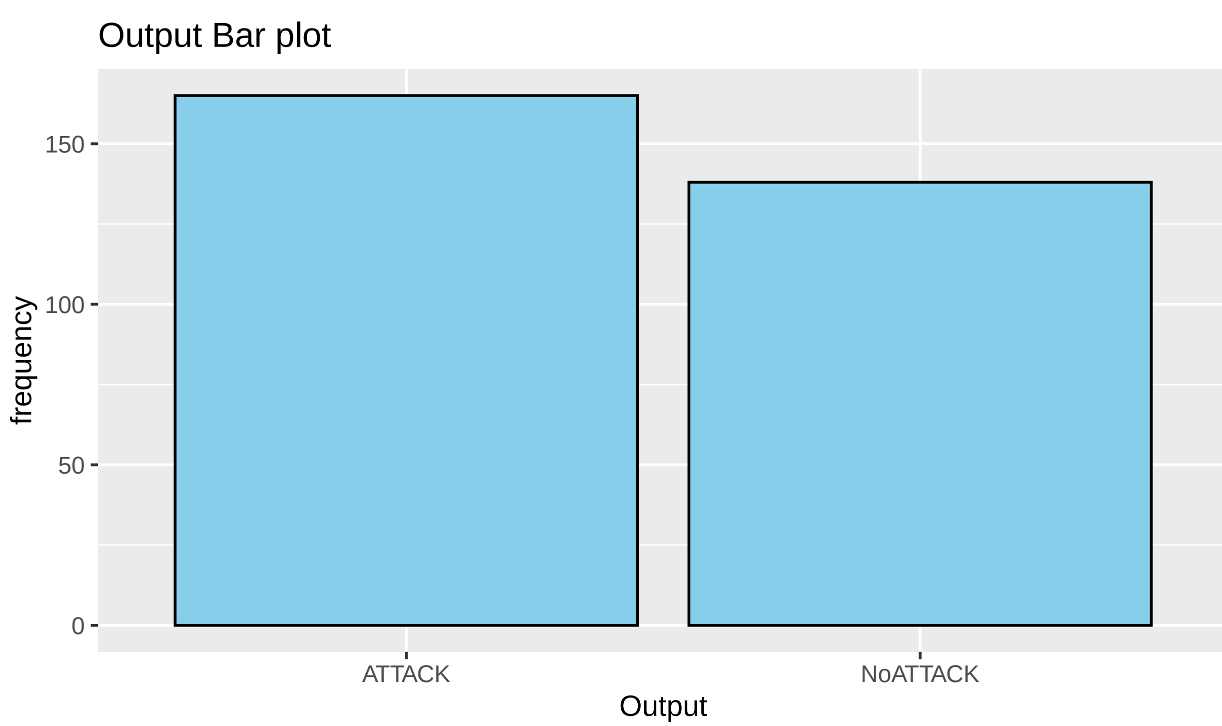
```r
# chest pain type to gender
ggplot(haData_vis, aes(x=cp, fill=sex)) +
          geom_bar(position="stack") +
          labs(title = "Chest pain type/Gender Plot", x = "Chest Pain type", y = 'Frequency') +
          theme(axis.text.x = element_text(angle = 30,hjust = 1))
```



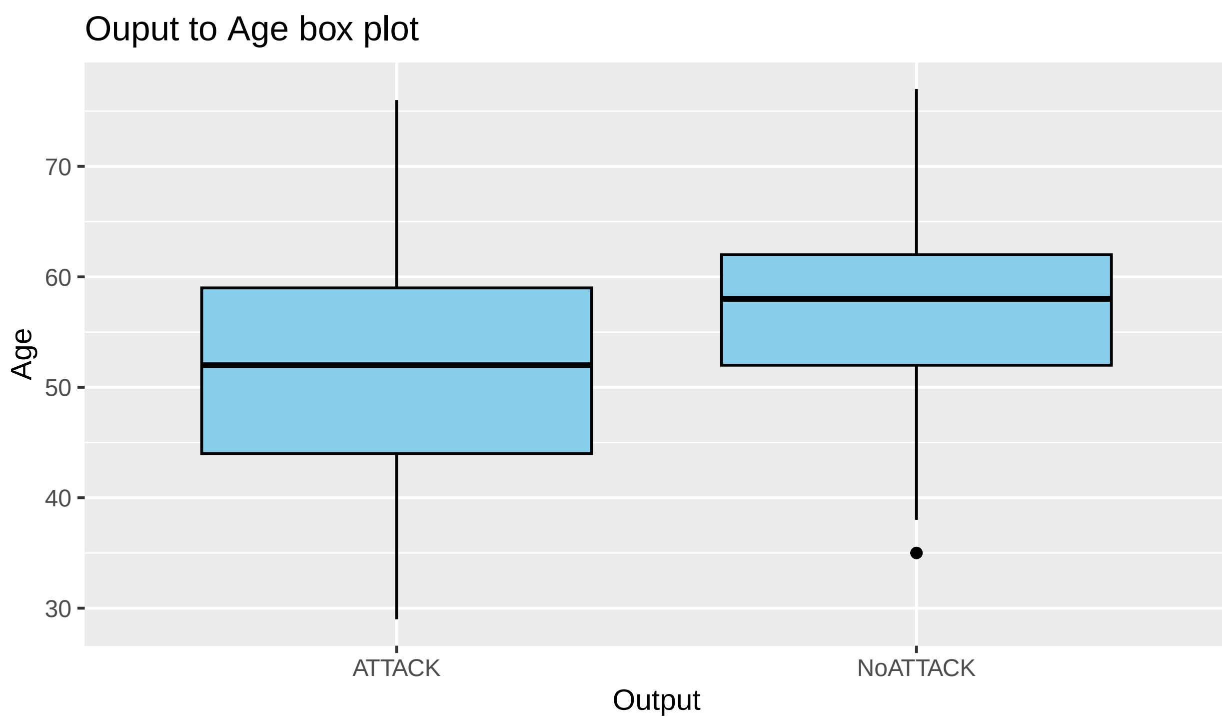

```r
# output with gender stack plot
ggplot(haData_vis, aes(x=output, fill=sex)) +
          geom_bar(position="stack") +
          labs(title = "Output/Gender Plot", x = "Output", y = 'Frequency') +
          theme(axis.text.x = element_text(angle = 30,hjust = 1))
```




```r
# output class bar plot
ggplot(haData_vis, aes(x = output)) +
    geom_bar(color="black", fill="skyblue") +
    labs(title = "Output Bar plot", x = "Output", y = 'frequency')
```



```r
# age to output
ggplot(haData_vis, aes(x= output, y = age)) +
  geom_boxplot(color="black", fill="skyblue") +
  labs(title = "Ouput to Age box plot", x="Output", y="Age")
```

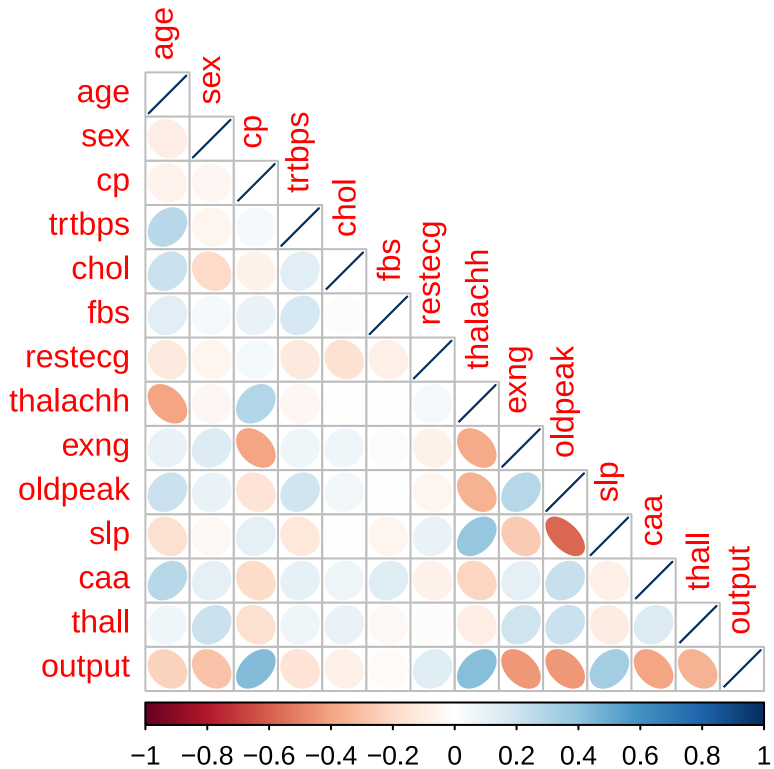


b) (ii) Correlation

In order to see the correlation we check on the data set with all everything numeric.

From the correlation plot we kind of see some +ve correlation of cp, thalachh with output, age having some +ve correlation with few descriptors. We don't have any variable non correlated with nothing neither we see any variable that is highly correlated to everyone.

```r
# correlation
corrplot(cor(haData), method = "ellipse", type="lower")
```



c) Data Cleaning

From the missing value analysis , we don't see any missing values that we need to take care of.

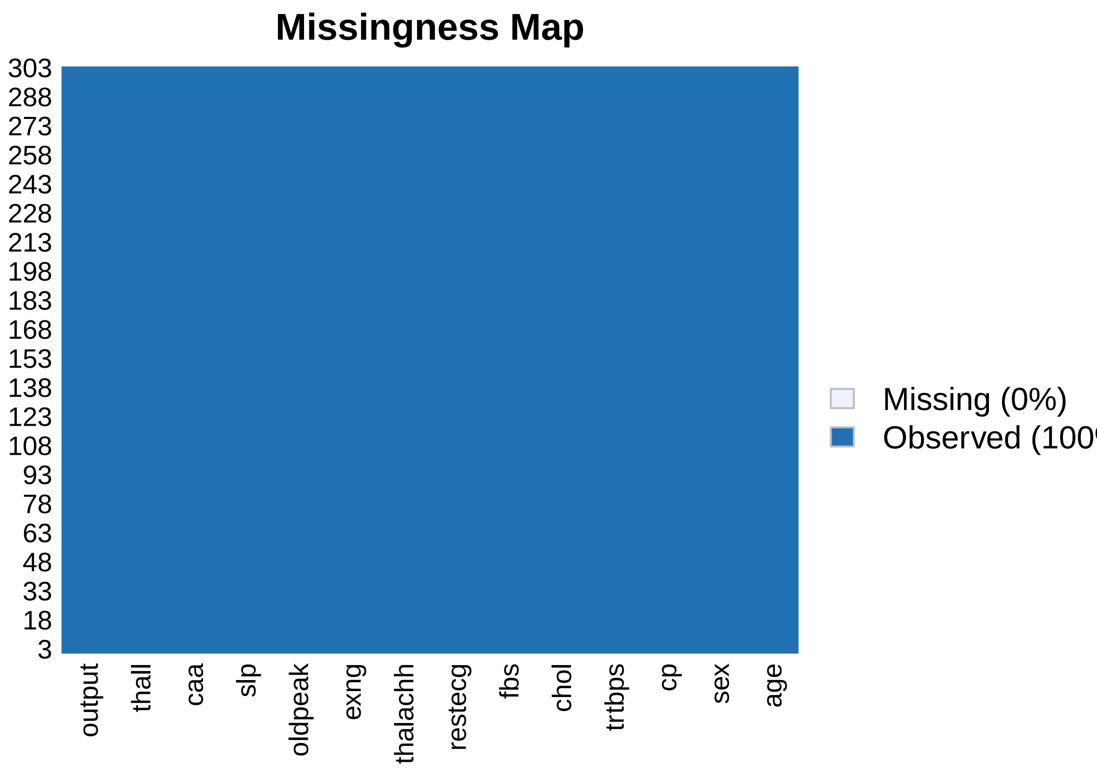
We mutated the age into young, old, Adult to see which age group is effected with more of heart attack and we plotted the data , and we see that adult age is more prone to heart attacks.

While visualizing the data we saw an outlier and investigated the record , but its not of much problem to us and we want to analyze every bit of the data and we leave it as is and see further how the model turns out.

```r
# check for NA's
haData_vis %>% map_int(~sum(is.na(.x)))
```

```
##      age      sex       cp  trtbps     chol      fbs  restecg thalachh
##        0        0        0       0        0        0        0        0
##     exng  oldpeak      slp      caa    thall   output
##        0        0        0       0        0        0
```

```r
# missing plot
missmap(haData_vis)
```




```r
# Possible Outlier record
haData_vis %>% filter(age==35,output=="NoATTACK")
```

```
##   age  sex           cp trtbps chol   fbs     restecg thalachh exng oldpeak slp
## 1  35 MALE ASYMPTOMATIC    120  198 <=120 ABNORMALITY      130  YES     1.6   1
## 2  35 MALE ASYMPTOMATIC    126  282 <=120      NORMAL      156  YES     0.0   2
##   caa thall   output
## 1   0     3 NoATTACK
## 2   0     3 NoATTACK
```

```r
# covert variable to factors
haData_vis$sex <- as.factor(haData_vis$sex)
haData_vis$cp <- as.factor(haData_vis$cp)
haData_vis$fbs <- as.factor(haData_vis$fbs)
haData_vis$restecg <- as.factor(haData_vis$restecg)
haData_vis$exng <- as.factor(haData_vis$exng)
haData_vis$output <- as.factor(haData_vis$output)

# summary of the data
summary(haData_vis)
```

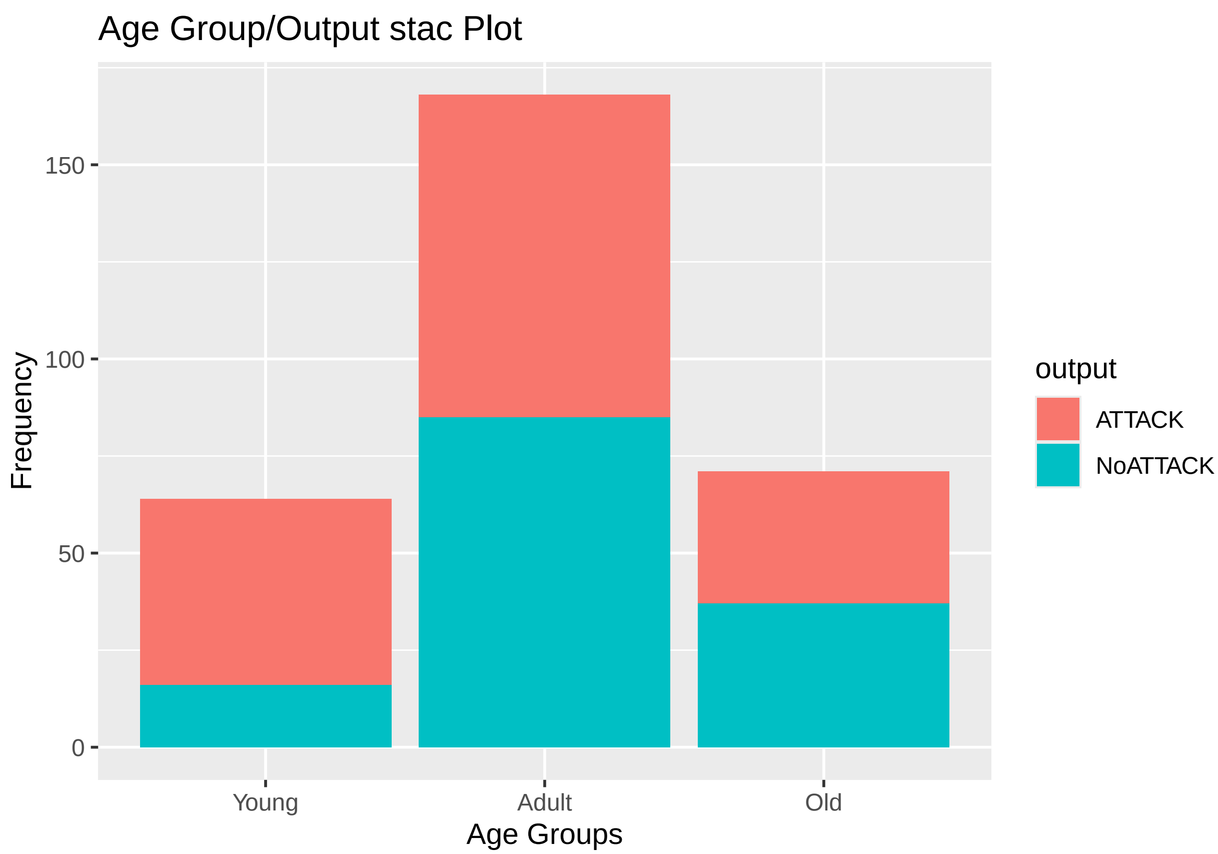
```
##       age            sex                      cp          trtbps
##  Min.   :29.00   FEMALE: 96   ASYMPTOMATIC    :166   Min.   : 94.0
##  1st Qu.:47.50   MALE  :207   ATYPICA_ANGINA  : 50   1st Qu.:120.0
##  Median :55.00                NON-ANGINAL_PAIN: 87   Median :130.0
```

```
##  Mean   :54.37                                      Mean   :131.6
##  3rd Qu.:61.00                                      3rd Qu.:140.0
##  Max.   :77.00                                      Max.   :200.0
##       chol           fbs                     restecg       thalachh
##  Min.   :126.0   <=120:258   ABNORMALITY         :152   Min.   : 71.0
##  1st Qu.:211.0   >120 : 45   NORMAL              :147   1st Qu.:133.5
##  Median :240.0               PROBABLE_OR_DEFINITE:  4   Median :153.0
##  Mean   :246.3                                          Mean   :149.6
##  3rd Qu.:274.5                                          3rd Qu.:166.0
##  Max.   :564.0                                          Max.   :202.0
##   exng        oldpeak          slp              caa              thall
##  NO :204   Min.   :0.00   Min.   :0.000   Min.   :0.0000   Min.   :0.000
##  YES: 99   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000
##            Median :0.80   Median :1.000   Median :0.0000   Median :2.000
##            Mean   :1.04   Mean   :1.399   Mean   :0.7294   Mean   :2.314
##            3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000
##            Max.   :6.20   Max.   :2.000   Max.   :4.0000   Max.   :3.000
##      output
##  ATTACK  :165
##  NoATTACK:138
##
##
##
##
```

```r
# mutate the range of ages
range<- haData_vis %>%
          mutate(age_bins = cut(age, breaks=3,
                                labels=c("Young","Adult","Old")))

# head of mutated data
head(range[,c(1,15)])
```

```
##   age age_bins
## 1  63      Old
## 2  37    Young
## 3  41    Young
## 4  56    Adult
## 5  57    Adult
## 6  57    Adult
```

```r
# visualize output with age bin
ggplot(range, aes(x=age_bins, fill=output)) +
          geom_bar(position="stack") +
          labs(title = "Age Group/Output stac Plot", x = "Age Groups", y = 'Frequency')
```




```r
# drop age from the data
range <- range[,-1]

#summary of the data
summary(range)
```

```
##      sex                     cp          trtbps           chol
##  FEMALE: 96   ASYMPTOMATIC    :166   Min.   : 94.0   Min.   :126.0
##  MALE  :207   ATYPICA_ANGINA  : 50   1st Qu.:120.0   1st Qu.:211.0
##               NON-ANGINAL_PAIN: 87   Median :130.0   Median :240.0
##                                      Mean   :131.6   Mean   :246.3
##                                      3rd Qu.:140.0   3rd Qu.:274.5
##                                      Max.   :200.0   Max.   :564.0
##     fbs                      restecg       thalachh       exng
##  <=120:258   ABNORMALITY          :152   Min.   : 71.0   NO :204
##  >120 : 45   NORMAL               :147   1st Qu.:133.5   YES: 99
##              PROBABLE_OR_DEFINITE:  4   Median :153.0
##                                         Mean   :149.6
##                                         3rd Qu.:166.0
##                                         Max.   :202.0
##     oldpeak          slp             caa              thall            output
##  Min.   :0.00   Min.   :0.000   Min.   :0.0000   Min.   :0.000   ATTACK  :165
##  1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000   NoATTACK:138
##  Median :0.80   Median :1.000   Median :0.0000   Median :2.000
##  Mean   :1.04   Mean   :1.399   Mean   :0.7294   Mean   :2.314
##  3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000
```

```
##  Max.   :6.20   Max.   :2.000   Max.   :4.0000   Max.   :3.000
##   age_bins
##  Young: 64
##  Adult:168
##  Old  : 71
##
##
##
```

d) Data Preprocessing

As part of data preprocessing we normalized the data with center scale and then created dummies for the data.

The normalized data will be used for various ML techniques and the dummies will be used for creating components as part of PCA analysis further down.

```r
# normalization with center scale
preproc1 <- preProcess(range, method=c("center", "scale"))
# We have to call predict to fit our data based on preprocessing
range_proc <- predict(preproc1, range)
# Here we can see the standardized version of our dataset
summary(range_proc)
```

```
##      sex                     cp          trtbps              chol
##  FEMALE: 96   ASYMPTOMATIC    :166   Min.   :-2.14525   Min.   :-2.3203
##  MALE  :207   ATYPICA_ANGINA  : 50   1st Qu.:-0.66277   1st Qu.:-0.6804
##               NON-ANGINAL_PAIN: 87   Median :-0.09258   Median :-0.1209
##                                      Mean   : 0.00000   Mean   : 0.0000
##                                      3rd Qu.: 0.47760   3rd Qu.: 0.5448
##                                      Max.   : 3.89872   Max.   : 6.1303
##     fbs                      restecg       thalachh          exng
##  <=120:258   ABNORMALITY         :152   Min.   :-3.4336   NO :204
##  >120 : 45   NORMAL              :147   1st Qu.:-0.7049   YES: 99
##              PROBABLE_OR_DEFINITE:  4   Median : 0.1464
##                                         Mean   : 0.0000
##                                         3rd Qu.: 0.7139
##                                         Max.   : 2.2856
##     oldpeak              slp                caa               thall
##  Min.   :-0.8954   Min.   :-2.2708   Min.   :-0.7132   Min.   :-3.7786
##  1st Qu.:-0.8954   1st Qu.:-0.6480   1st Qu.:-0.7132   1st Qu.:-0.5121
##  Median :-0.2064   Median :-0.6480   Median :-0.7132   Median :-0.5121
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.4827   3rd Qu.: 0.9747   3rd Qu.: 0.2646   3rd Qu.: 1.1212
##  Max.   : 4.4445   Max.   : 0.9747   Max.   : 3.1983   Max.   : 1.1212
##       output      age_bins
##  ATTACK  :165   Young: 64
##  NoATTACK:138   Adult:168
##                 Old  : 71
##
##
##
```

```r
# dummy variable for the categorical
dummyHa <- dummyVars(output ~., data = range_proc)
# transformation to dummy variables and a dataframe
dummiesHa <- as.data.frame(predict(dummyHa, newdata = range_proc))
```

```
## Warning in model.frame.default(Terms, newdata, na.action = na.action, xlev =
## object$lvls): variable 'output' is not a factor
```

```r
# head of data
head(dummiesHa)
```

```
##   sex.FEMALE sex.MALE cp.ASYMPTOMATIC cp.ATYPICA_ANGINA cp.NON-ANGINAL_PAIN
## 1          0        1               1                 0                   0
## 2          0        1               0                 0                   1
## 3          1        0               0                 1                   0
## 4          0        1               0                 1                   0
## 5          1        0               1                 0                   0
## 6          0        1               1                 0                   0
##        trtbps        chol fbs.<=120 fbs.>120 restecg.ABNORMALITY restecg.NORMAL
## 1  0.76269408 -0.25591036         0        1                   0              1
## 2 -0.09258463  0.07208025         1        0                   1              0
## 3 -0.09258463 -0.81542377         1        0                   0              1
## 4 -0.66277043 -0.19802967         1        0                   1              0
## 5 -0.66277043  2.07861109         1        0                   1              0
## 6  0.47760118 -1.04694656         1        0                   1              0
##   restecg.PROBABLE_OR_DEFINITE    thalachh exng.NO exng.YES    oldpeak
## 1                            0  0.01541728       1        0  1.0855423
## 2                            0  1.63077374       1        0  2.1190672
## 3                            0  0.97589950       1        0  0.3103986
## 4                            0  1.23784920       1        0 -0.2063639
## 5                            0  0.58297496       0        1 -0.3786180
## 6                            0 -0.07189928       1        0 -0.5508722
##          slp       caa      thall age_bins.Young age_bins.Adult age_bins.Old
## 1 -2.2708221 -0.713249 -2.1453238              0              0            1
## 2 -2.2708221 -0.713249 -0.5120748              1              0            0
## 3  0.9747397 -0.713249 -0.5120748              1              0            0
## 4  0.9747397 -0.713249 -0.5120748              0              1            0
## 5  0.9747397 -0.713249 -0.5120748              0              1            0
## 6 -0.6480412 -0.713249 -2.1453238              0              1            0
```

e) Clustering

As part of clustering we will be performing k mean clustering and HAC clustering and for visualization we will be performing PCA on the dummy variable data.

e) (i) Principal component analysis

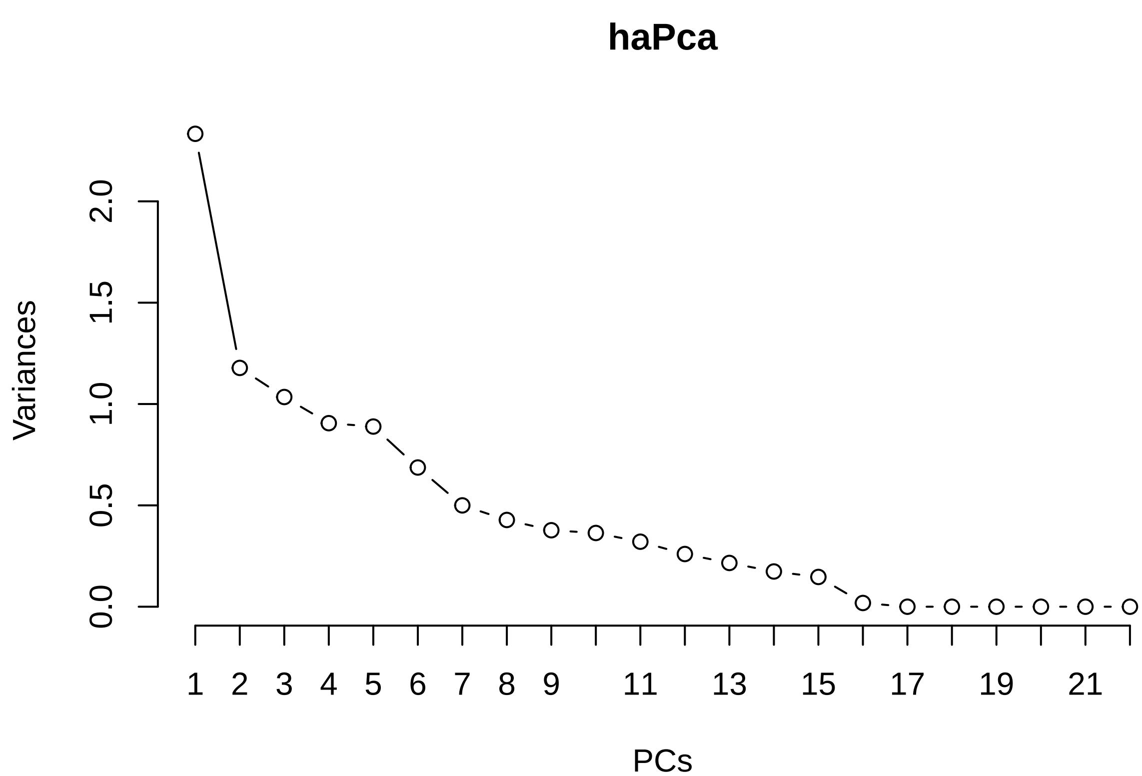
NOTE : Scaling is not required as we have already scaled the data

We perform PCA to get 2 dimensional data for visualization , as we have performed pca only for visualization we are not worried of selecting no.of components here. however PC11 captured variance ranging 91%.

```r
# pca on the data set
haPca <- prcomp(dummiesHa)
summary(haPca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.5274 1.0854 1.0171 0.95159 0.94277 0.82872 0.70708
## Proportion of Variance 0.2373 0.1198 0.1052 0.09211 0.09041 0.06986 0.05085
## Cumulative Proportion  0.2373 0.3572 0.4624 0.55449 0.64490 0.71475 0.76561
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.65420 0.61423 0.60307 0.56636 0.50980 0.46459 0.41688
## Proportion of Variance 0.04353 0.03838 0.03699 0.03263 0.02644 0.02196 0.01768
## Cumulative Proportion  0.80914 0.84752 0.88451 0.91714 0.94357 0.96553 0.98321
##                           PC15    PC16      PC17      PC18      PC19      PC20
## Standard deviation     0.38312 0.13535 2.088e-15 6.928e-16 2.878e-16 2.152e-16
## Proportion of Variance 0.01493 0.00186 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  0.99814 1.00000 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC21      PC22
## Standard deviation     1.248e-16 5.749e-17
## Proportion of Variance 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00
```

```r
# Visualize the scree plot
screeplot(haPca, npcs = 22, type="l") + title(xlab = "PCs")
```



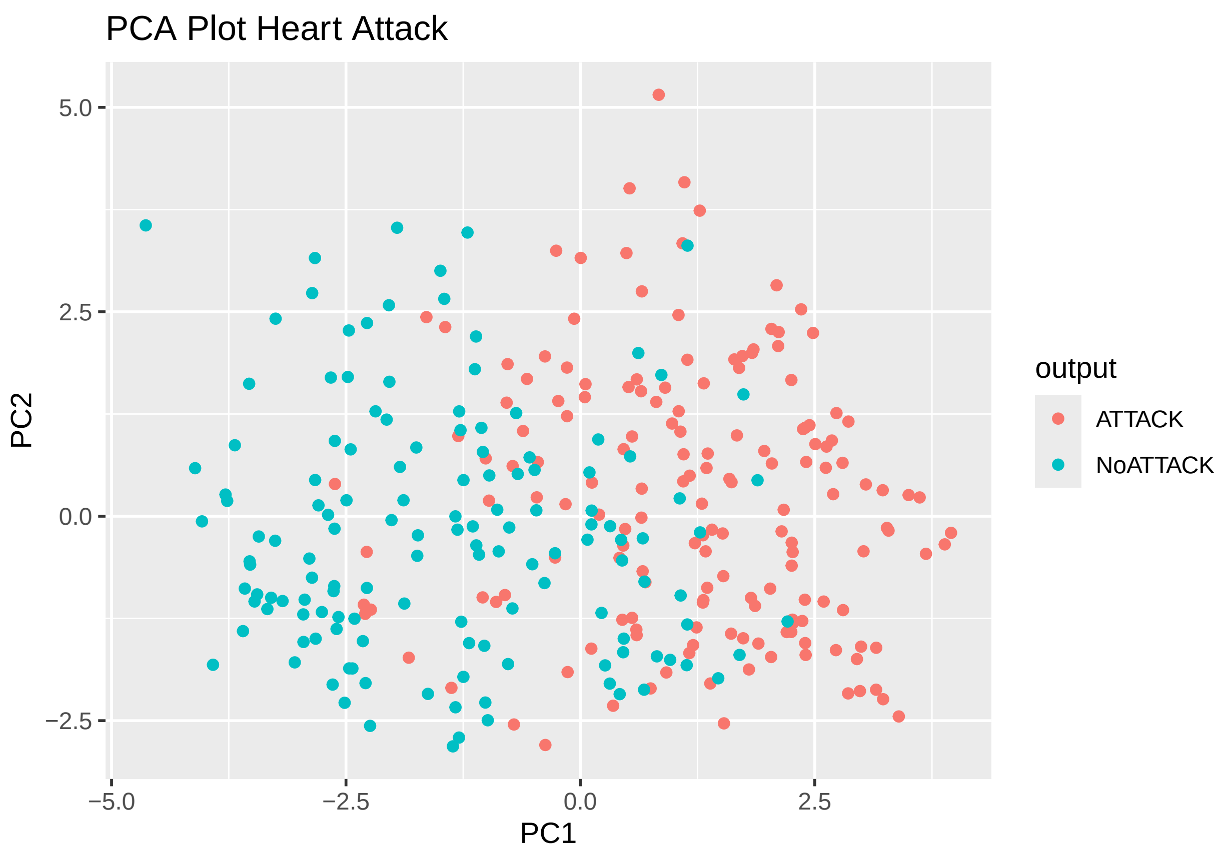
```
## integer(0)
```

```r
# reduced set of pc's
preProc <- preProcess(dummiesHa, method="pca", pcaComp=2)
haPcaReduced <- predict(preProc, dummiesHa)

# adding the output column back
haPcaReduced$output <- range$output

# head of reduced pca
head(haPcaReduced)
```

```
##          PC1        PC2 output
## 1 -1.6418543  2.4347551 ATTACK
## 2  1.3123886 -1.0262162 ATTACK
## 3  2.6255073  0.8513862 ATTACK
## 4  2.4031927 -1.6962768 ATTACK
## 5 -0.1579426  0.1475131 ATTACK
## 6  0.5518719 -1.2426848 ATTACK
```

The scatter plot for the 2D PCA data is displayed below with color category of output. From the scatter plot we don't see a prominent distinct grouping rather its overlapped for PC1 and PC2, this is may be because the data might in id a different dimension.

```r
# scatter plot for type of wine on the PC
ggplot(haPcaReduced, aes(x=PC1, y=PC2)) +
      geom_point(aes(col=output)) +
      labs(title = "PCA Plot Heart Attack ")
```



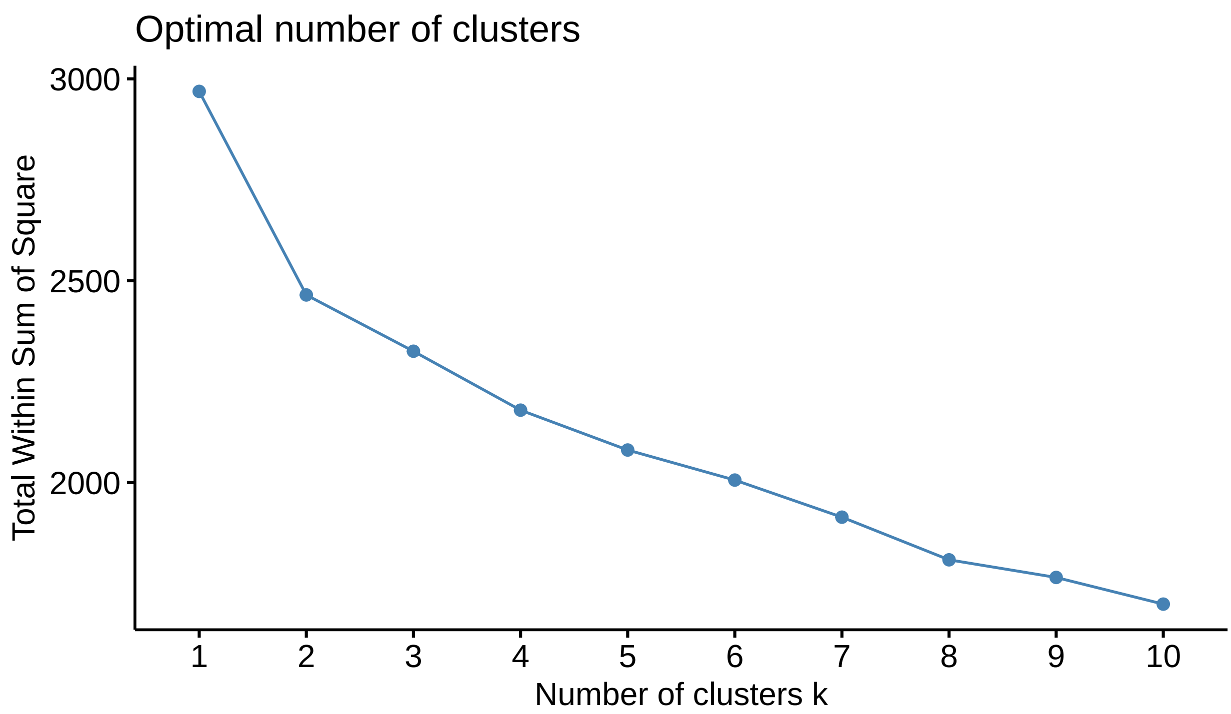
e) (ii) K mean Clustering

Looking at the above visualization we proceed with k mean clustering and try to see how it performs.

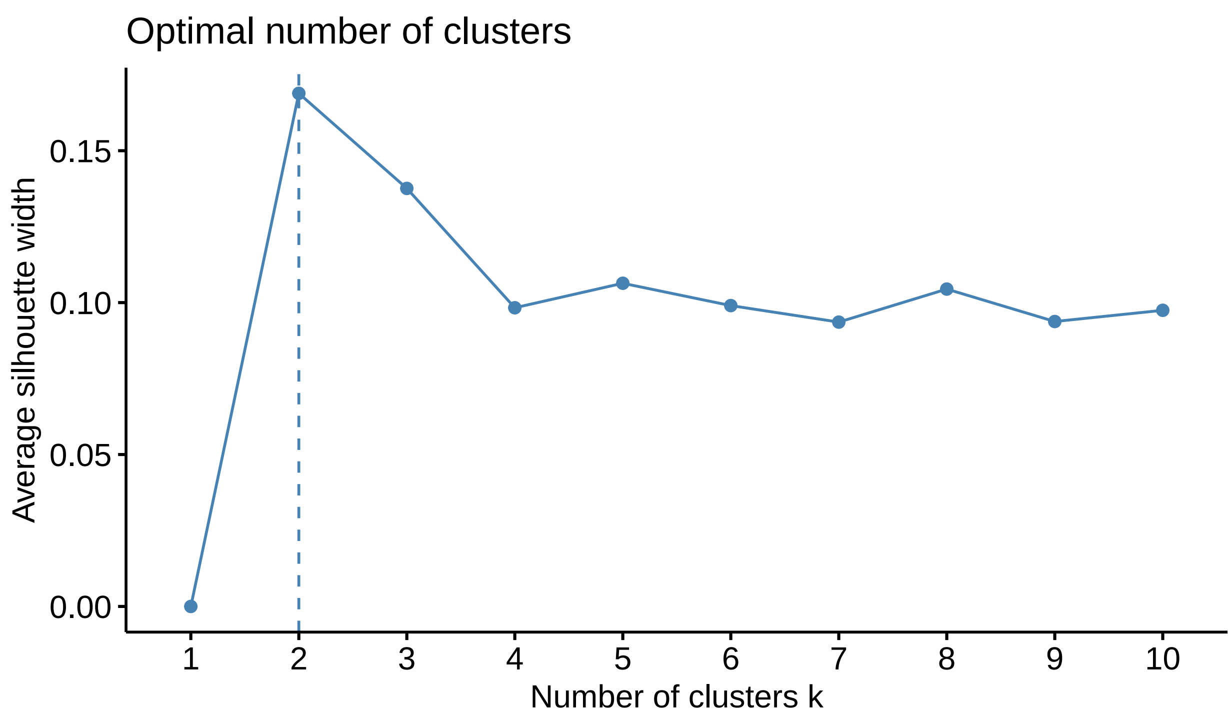
From the knee and shilhouete plots we get information to select the k value as 2.

We then perform the kmean with center as 2 and nstart 25 and displayed the model.

```
set.seed(3010)

# Find the knee
fviz_nbclust(dummiesHa, kmeans, method = "wss")
```




```r
# average silhouette
fviz_nbclust(dummiesHa, kmeans, method = "silhouette")
```




```r
# Fit the data with nstarts 25
fit <- kmeans(dummiesHa, centers = 2, nstart = 25)
# Display the kmeans object information
```

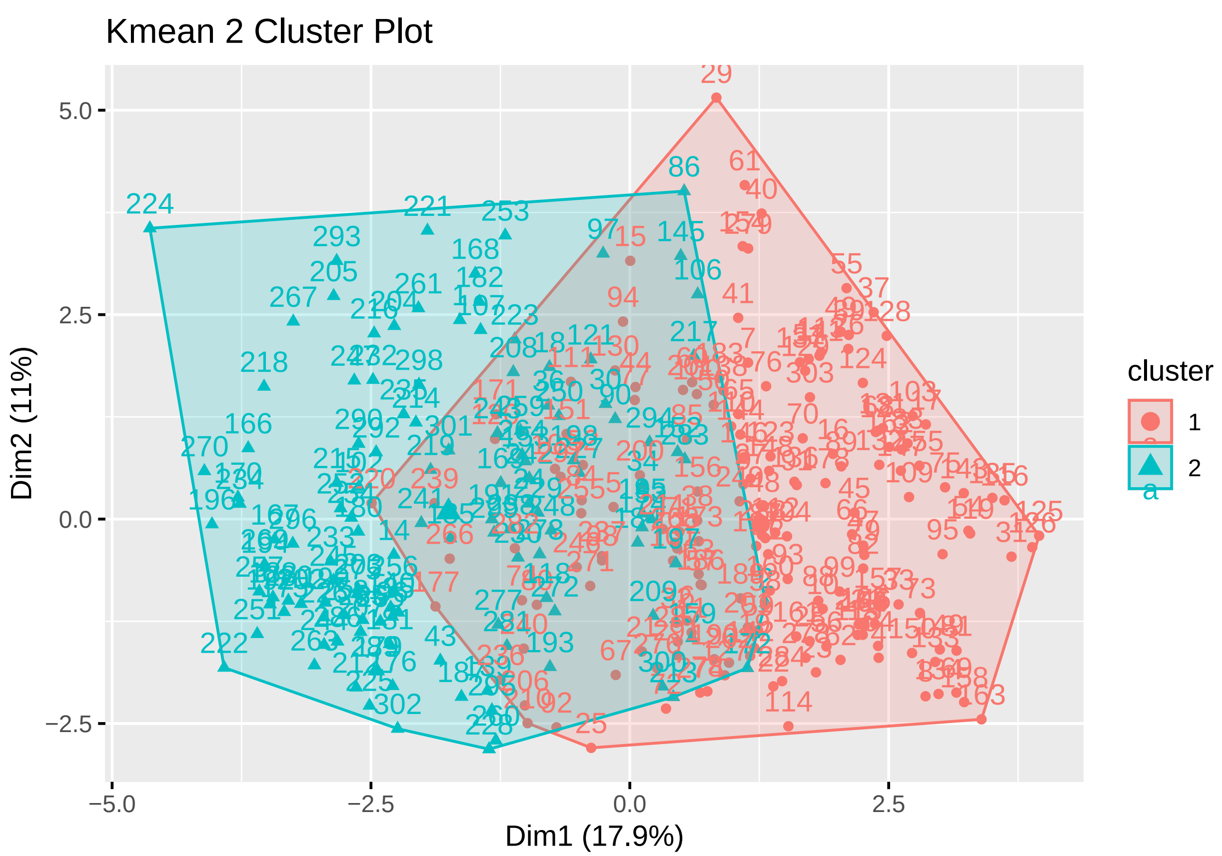
```r
fit
```

```
## K-means clustering with 2 clusters of sizes 176, 127
##
## Cluster means:
##   sex.FEMALE  sex.MALE cp.ASYMPTOMATIC cp.ATYPICA_ANGINA cp.NON-ANGINAL_PAIN
## 1  0.3806818 0.6193182       0.3636364        0.26136364           0.3750000
## 2  0.2283465 0.7716535       0.8031496        0.03149606           0.1653543
##       trtbps        chol fbs.<=120  fbs.>120 restecg.ABNORMALITY restecg.NORMAL
## 1 -0.1619140 -0.06812697 0.8636364 0.1363636           0.5852273      0.4147727
## 2  0.2243848  0.09441217 0.8346457 0.1653543           0.3858268      0.5826772
##   restecg.PROBABLE_OR_DEFINITE   thalachh   exng.NO  exng.YES    oldpeak
## 1                   0.00000000  0.5040924 0.8522727 0.1477273 -0.5567445
## 2                   0.03149606 -0.6985847 0.4251969 0.5748031  0.7715514
##          slp        caa      thall age_bins.Young age_bins.Adult age_bins.Old
## 1  0.5690444 -0.2631958 -0.2522397      0.2897727      0.5625000    0.1477273
## 2 -0.7885970  0.3647437  0.3495605      0.1023622      0.5433071    0.3543307
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   2   2   1   1   1   1   1   1   1   1   1   1   1   2   1   1   1   2   1   1
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   1   1   2   1   1   1   1   1   2   1   1   1   2   2   2   1   1   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   2   1   1   1   1   1   1   1   1   1   2   1   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   1   1   1   1   1   2   1   1   1   2   1   1   1   1   1   2   2   1   1   1
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   1   2   1   1   1   2   2   1   1   1   1   1   1   1   1   1   1   2   1   1
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   2
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   1   1   1   1   2   1   1   1   1   1   1   2   2   1   1   1   1   1   2   1
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   1   1   1   1   1   2   2   2   2   2   1   2   1   2   2   2   1   1   2   2
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   2   2   1   2   2   1   2   2   1   1   2   2   2   2   2   2   2   2   2   1
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   1   2   2   2   2   1   2   2   2   1   1   2   2   2   2   2   2   2   2   1
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   1   1   2   1   1
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   2   2   2   2   2   1   2   2   1   2   2   2   2   2   1   2   2   2   2   2
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   2   1   2   2   2   1   2   1   2   2   1   2   2   1   2   1   2   1   1   2
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   2   1   2   1   2   2   1   1   2   2   1   2   2   2   2   2   1   2   2   2
## 301 302 303
##   2   2   1
##
## Within cluster sum of squares by cluster:
## [1] 1237.178 1227.542
```

```
##  (between_SS / total_SS =  17.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

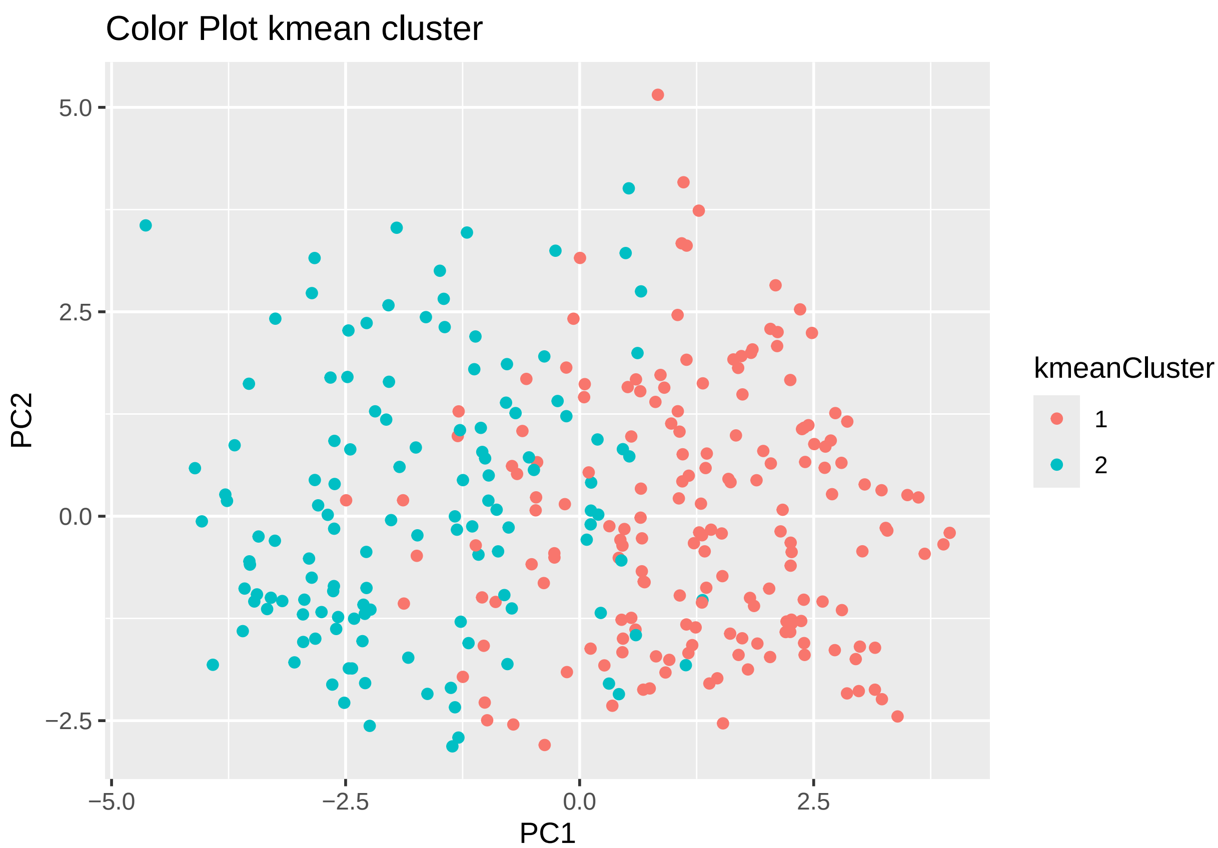
From the visualization we see 2 clusters with overlaps same as we saw in our PCA visualization in 2 dimension.

```r
# Display the cluster plot
fviz_cluster(fit, data = dummiesHa, main="Kmean 2 Cluster Plot")
```



For comparison we use the PCA to visualize the data for k mean cluster. From the comparison plot we kind of get similar result as the PCA visualization.

```r
# copy of the pca data
rotated_data <- haPcaReduced

# Assign kmean clusters labels as a new column
rotated_data$Kmean_clusters = as.factor(fit$cluster)

# Plot and color by kmean cluster label
ggplot(rotated_data, aes(x=PC1, y=PC2)) +
        geom_point(aes(col=Kmean_clusters)) +
        labs(title = "Color Plot kmean cluster", col = "kmeanCluster")
```

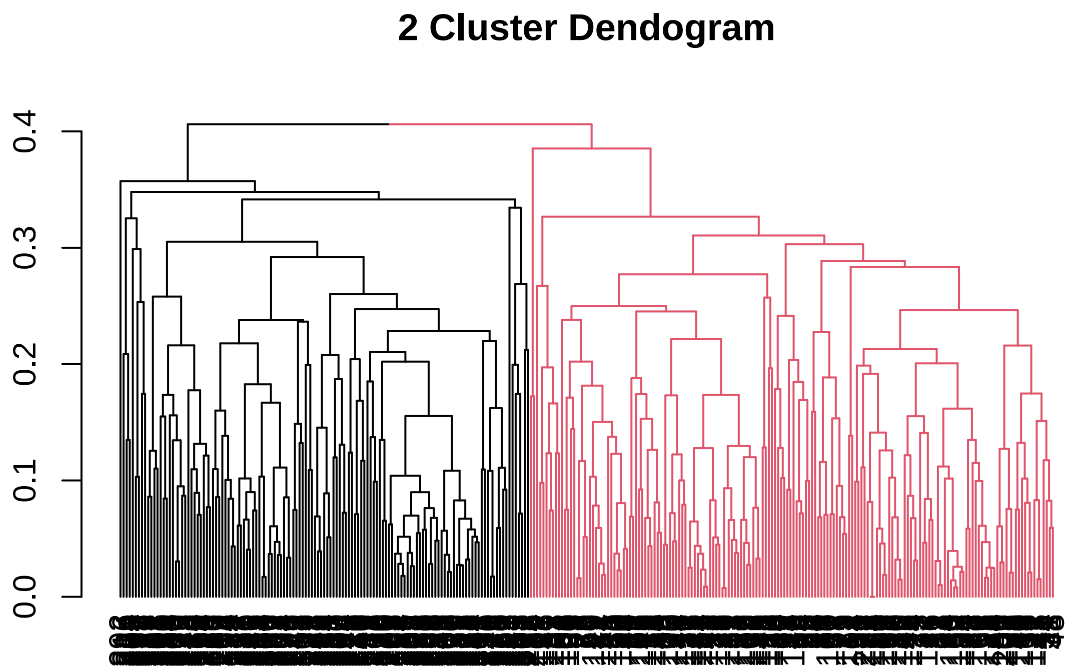


e) (iii) HAC Clustering

We also perform a HAC clustering for comparison and we see 2 good clusters from the dendogram as we selected k = 2 from the knee and shiloutte plot above.

In this step we used the gower matrix to compute the distance matrix as we have both factors and numeric in the data, we use the method as average for computing the clustering model.

```
# Pass dataframe directly with metric = gower
dist_mat_sw <- daisy(range_proc, metric = "gower")
# fit the model
hfit_sw <- hclust(dist_mat_sw, method = 'average')

# convert to dendogram
dend_sw <- as.dendrogram(hfit_sw)

# color the branches
coldend_sw <- color_branches(dend_sw, k = 2, col = c(1,2))
# plot the dendogram
plot(coldend_sw, main="2 Cluster Dendogram")
```



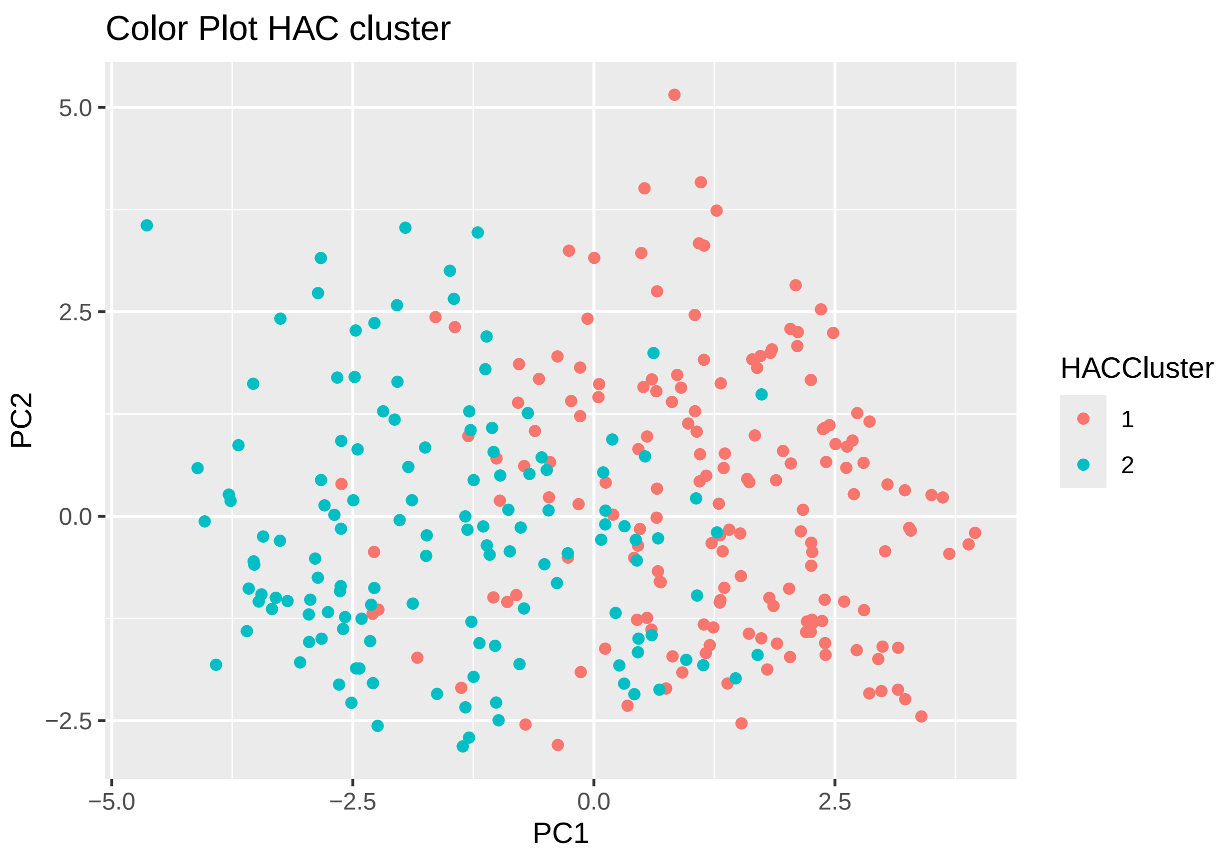
We build the model with 2 clusters here for HAC using cut-tree and the head of the data is displayed.

```
# Build the new model with k =2
h1_sw2 <- cutree(hfit_sw, k=2)

head(h1_sw2)
```

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

We performed the visualization for HAC with the pca components and we see a almost similar kind of visualization as we saw in our previous steps having overlaps in 2-D.

```
# Assign kmean clusters labels as a new column
rotated_data$hac_clusters = as.factor(h1_sw2)

# Plot and color by kmean cluster label
ggplot(rotated_data, aes(x=PC1, y=PC2)) +
        geom_point(aes(col=hac_clusters)) +
        labs(title = "Color Plot HAC cluster", col = "HACCluster")
```



We then compare the clusters with actual labels with cross tabulation for both kmean and HAC, and we see some decent result with True positives and True negatives for HAC technique. We see the models predictive power of getting most of true positives is over 98% which is pretty decent

So as a conclusion we see HAC performed better in terms of cross tabulation evaluation with predicting the most +ve's and -ve's accurately.

```
# comparison to actual label
result <- data.frame(Type = range$output, Kmeans = fit$cluster, hac = h1_sw2)
# View the first 6 cases one by one
head(result)
```

```
##     Type Kmeans hac
## 1 ATTACK      2   1
## 2 ATTACK      2   1
## 3 ATTACK      1   1
## 4 ATTACK      1   1
## 5 ATTACK      1   1
## 6 ATTACK      1   1
```

```
# Crosstab for K Means
result %>% group_by(Kmeans) %>% dplyr::select(Kmeans, Type) %>% table()
```

```
##       Type
## Kmeans ATTACK NoATTACK
##      1    139       37
##      2     26      101
```

```r
# Crosstab for hac
result %>% group_by(hac) %>% dplyr::select(hac, Type) %>% table()
```

```
##    Type
## hac ATTACK NoATTACK
##   1    163        7
##   2      2      131
```

f) Classification

For Classification we will be performing KNN and Decision Tree.

To proceed we take the preprocess data set(range) computed previously and then we divide the data to 70% training and 30% testing to proceed ahead.

```r
set.seed(1234)

# copy the data
clasData <- range

# Partition the data
index = createDataPartition(y=clasData$output, p=0.7, list=FALSE)

# get the train set as index
train_set = clasData[index,]

# get the test set
test_set = clasData[-index,]

# head of data
head(clasData)
```

```
##      sex                cp trtbps chol   fbs     restecg thalachh exng oldpeak
## 1   MALE      ASYMPTOMATIC    145  233  >120      NORMAL      150   NO     2.3
## 2   MALE NON-ANGINAL_PAIN    130  250 <=120 ABNORMALITY      187   NO     3.5
## 3 FEMALE    ATYPICA_ANGINA    130  204 <=120      NORMAL      172   NO     1.4
## 4   MALE    ATYPICA_ANGINA    120  236 <=120 ABNORMALITY      178   NO     0.8
## 5 FEMALE      ASYMPTOMATIC    120  354 <=120 ABNORMALITY      163  YES     0.6
## 6   MALE      ASYMPTOMATIC    140  192 <=120 ABNORMALITY      148   NO     0.4
##   slp caa thall output age_bins
## 1   0   0     1 ATTACK      Old
## 2   0   0     2 ATTACK    Young
## 3   2   0     2 ATTACK    Young
## 4   2   0     2 ATTACK    Adult
## 5   2   0     2 ATTACK    Adult
## 6   1   0     1 ATTACK    Adult
```

f) (i) KNN

We performed the KNN technique for classification using 10 fold cross validation with a grid search of rectangular and triangular with a euclidean and manhattan distance with a kmax of 3:9.

We see the model reported a highest **accuracy of 79% and a kappa of 58%** with kmax = 9, distance = 1 and rectangular kernel.

```r
set.seed(3010)

# Set number of folds
folds <- 10

# Generate stratified indices (per fold list of indices, which are the row numbers)
idx <- createFolds(train_set$output, folds, returnTrain = T)

# evaluation method as cv
ctrl <- trainControl(index = idx, method = 'cv', number = folds)

# tuneGrid with the tuning parameters
tuneGrid <- expand.grid(kmax = 3:9, kernel = c("rectangular","triangular"),
                        distance = 1:2)

# tune and fit the model with 10-fold cross validation,
# standardization, and our specialized tune grid
# preprocess is not required as we are using PC components
kknn_fit <- train(output ~ .,data = train_set,
                  method = 'kknn',
                  trControl = ctrl,
                  preProcess = c('center', 'scale'),
                  tuneGrid = tuneGrid)
```

```
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable ’.outcome’ is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
```

```
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
```

```
# Printing trained model provides report
kknn_fit
```

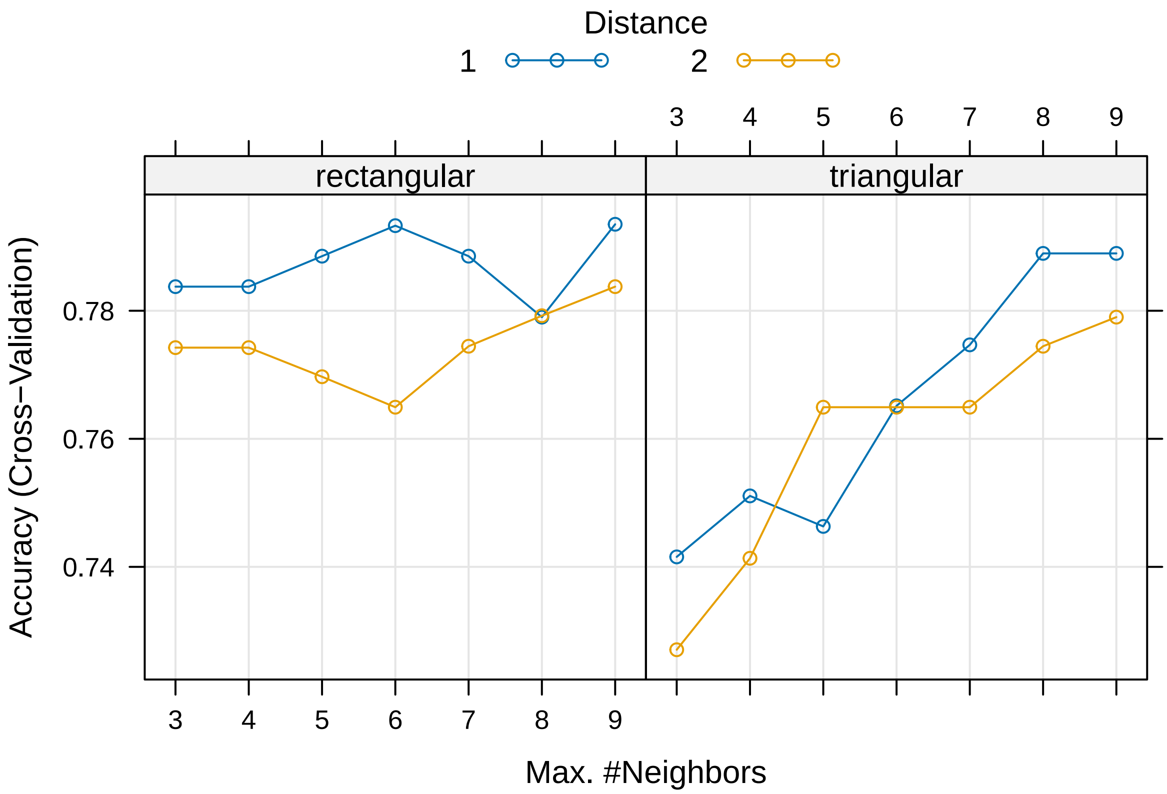
```
## k-Nearest Neighbors
##
## 213 samples
```

```
##  13 predictor
##   2 classes: 'ATTACK', 'NoATTACK'
##
## Pre-processing: centered (16), scaled (16)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 191, 191, 191, 192, 192, 192, ...
## Resampling results across tuning parameters:
##
##   kmax  kernel       distance  Accuracy   Kappa
##   3     rectangular  1         0.7837662  0.5613122
##   3     rectangular  2         0.7742424  0.5421024
##   3     triangular   1         0.7415584  0.4747625
##   3     triangular   2         0.7270563  0.4446110
##   4     rectangular  1         0.7837662  0.5613122
##   4     rectangular  2         0.7742424  0.5421024
##   4     triangular   1         0.7510823  0.4941353
##   4     triangular   2         0.7413420  0.4749212
##   5     rectangular  1         0.7885281  0.5711627
##   5     rectangular  2         0.7696970  0.5343499
##   5     triangular   1         0.7463203  0.4852578
##   5     triangular   2         0.7649351  0.5216942
##   6     rectangular  1         0.7932900  0.5806638
##   6     rectangular  2         0.7649351  0.5245120
##   6     triangular   1         0.7651515  0.5249177
##   6     triangular   2         0.7649351  0.5216942
##   7     rectangular  1         0.7885281  0.5693993
##   7     rectangular  2         0.7744589  0.5447486
##   7     triangular   1         0.7746753  0.5425661
##   7     triangular   2         0.7649351  0.5216942
##   8     rectangular  1         0.7790043  0.5506580
##   8     rectangular  2         0.7792208  0.5538193
##   8     triangular   1         0.7889610  0.5701524
##   8     triangular   2         0.7744589  0.5411415
##   9     rectangular  1         0.7935065  0.5807273
##   9     rectangular  2         0.7837662  0.5629960
##   9     triangular   1         0.7889610  0.5701524
##   9     triangular   2         0.7790043  0.5515356
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were kmax = 9, distance = 1 and kernel
##  = rectangular.
```

We plot the model accuracy and predicted the model with the test data and verified the Model performance.

The test performance was reported as **accuracy 84% with a kappa of 68%**. We see the test accuracy slightly higher than the test which suggests that the model is generalizing well to unseen data.

```r
# plot the model accuracy
plot(kknn_fit)
```




```r
# predict the model with test data
pred_knnTest <- predict(kknn_fit, test_set)
```

```
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
```

```r
# display the confusion matrix
confusionMatrix(as.factor(test_set$output), pred_knnTest)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ATTACK NoATTACK
##   ATTACK       47        2
##   NoATTACK     12       29
##
##                Accuracy : 0.8444
##                  95% CI : (0.7528, 0.9123)
##     No Information Rate : 0.6556
##     P-Value [Acc > NIR] : 5.172e-05
##
##                   Kappa : 0.68
##
##  Mcnemar's Test P-Value : 0.01616
##
```

```
##            Sensitivity : 0.7966
##            Specificity : 0.9355
##         Pos Pred Value : 0.9592
##         Neg Pred Value : 0.7073
##             Prevalence : 0.6556
##         Detection Rate : 0.5222
##   Detection Prevalence : 0.5444
##      Balanced Accuracy : 0.8660
##
##       'Positive' Class : ATTACK
##
```

f) (ii) Decision Tree

We perform another technique "Decision Tree" for classification and to do that we set a list of hyper parameters with the same train control and tried to capture the best model.

The dataframe having all the details fro this iteration is displayed.

And the best model was reported with 11 nodes having a **Test accuracy of 80% and a training accuracy of 84%** with minsplit =12 , max depth = 7 and maxbucket=12. Here we see the training accuracy is higher but there is not much significant difference which we should be worried about.

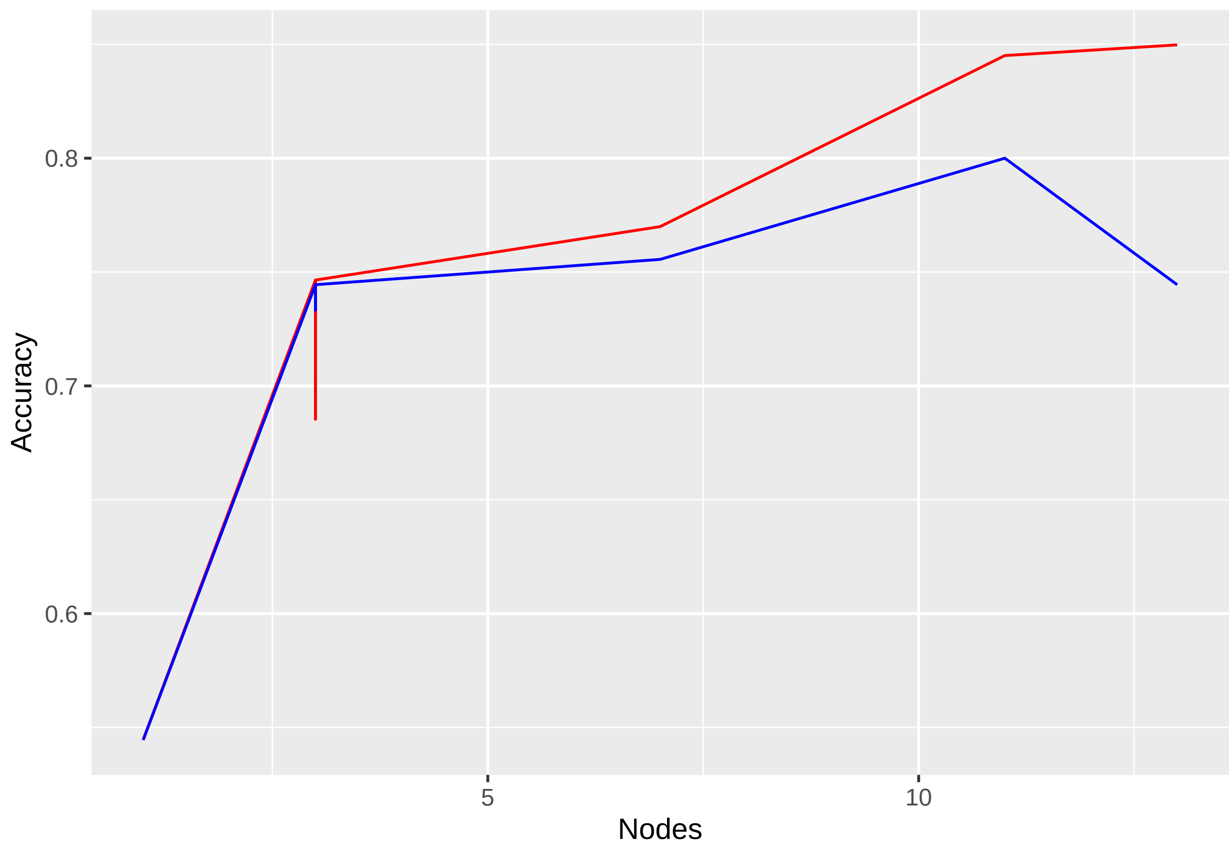
```r
# set the seed
set.seed(3010)

# initialize the set of hyper parameters
hyper_data <- list(c(2,1,2),c(2,2,2),c(5,2,3),c(5,3,3),c(50,3,50),c(100,4,100),c(50,5,50),
c(100,6,100),c(12,7,12),c(300,8,300),c(50,9,50),c(700,12,700),c(1000,25,1000))

# initialize the dataframe
comp_tbl <- data.frame()
# length of the set of parameters
len = length(hyper_data)

for(i in 1:len){
  # get the hyper data
  minSp = hyper_data[i][[1]][[1]]
  maxDp = hyper_data[i][[1]][[2]]
  minBk = hyper_data[i][[1]][[3]]

  # create hyper parameter
  hypers = rpart.control(minsplit = minSp, maxdepth = maxDp, minbucket = minBk)

  # build decision tree
  cmpTree <- train(output ~ .,data = train_set, control = hypers,
                   trControl = ctrl, method = "rpart1SE")

  # Train set confusion matrix
  pred_train <- predict(cmpTree, train_set)
  cfm_train <- confusionMatrix(train_set$output, pred_train)

  # Test set confusion matrix
  pred_test <- predict(cmpTree, test_set)
```

```r
  cfm_test <- confusionMatrix(test_set$output, pred_test)

  # training accuracy
  a_train <- cfm_train$overall[1]
  # testing accuracy
  a_test <- cfm_test$overall[1]
  # Get number of nodes
  nodes <- nrow(cmpTree$finalModel$frame)

  # Add rows to the table
  comp_tbl <- comp_tbl %>% rbind(list(nodes, a_train, a_test, minSp, maxDp, minBk))

}

# assign the column Names
names(comp_tbl) <- c("Nodes", "TrainAccuracy", "TestAccuracy", "MinSplit",
                     "MaxDepth", "MinBucket")

# display the data
comp_tbl
```

```
##    Nodes TrainAccuracy TestAccuracy MinSplit MaxDepth MinBucket
## 1      3     0.7464789    0.7444444        2        1         2
## 2      7     0.7699531    0.7555556        2        2         2
## 3      7     0.7699531    0.7555556        5        2         3
## 4     13     0.8497653    0.7444444        5        3         3
## 5      3     0.7464789    0.7444444       50        3        50
## 6      3     0.6854460    0.7333333      100        4       100
## 7      3     0.7464789    0.7444444       50        5        50
## 8      3     0.6854460    0.7333333      100        6       100
## 9     11     0.8450704    0.8000000       12        7        12
## 10     1     0.5446009    0.5444444      300        8       300
## 11     3     0.7464789    0.7444444       50        9        50
## 12     1     0.5446009    0.5444444      700       12       700
## 13     1     0.5446009    0.5444444     1000       25      1000
```

we then visualize the result to find a sweet spot for our model comparison and we see the train and test accuracy kind going hand in hand and then diverging towards the end which is kind of decent as per the predictive power of the model.

```r
# Visualize with line plot
ggplot(comp_tbl, aes(x=Nodes)) +
  geom_line(aes(y = TrainAccuracy), color = "red") +
  geom_line(aes(y = TestAccuracy), color="blue") +
  ylab("Accuracy")
```



In order to pull the model out and visualize the tree, we perform the decision tree with the same set of parameters as reported in the extensive testing above.

The accuracy of the model was reported as 84% in training and 80% in testing showing different measures of performance for the model.

From the feature importance we see features like oldpeak, exercise include angina, slope, number of major vessels are some of the important factors affecting heart attack.

```
set.seed(3010)

# create hyper parameter for 5 nodes
hypers = rpart.control(minsplit =12, maxdepth = 7, minbucket = 12)
# build decision tree
flTree <- train(output ~ .,data = train_set, control = hypers,
                    trControl = ctrl, method = "rpart1SE")

# train set confusion matrix
pred_train_fl <- predict(flTree, train_set)
cfm_train_fl <- confusionMatrix(train_set$output, pred_train_fl)
cfm_train_fl
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction ATTACK NoATTACK
##     ATTACK      101       15
```

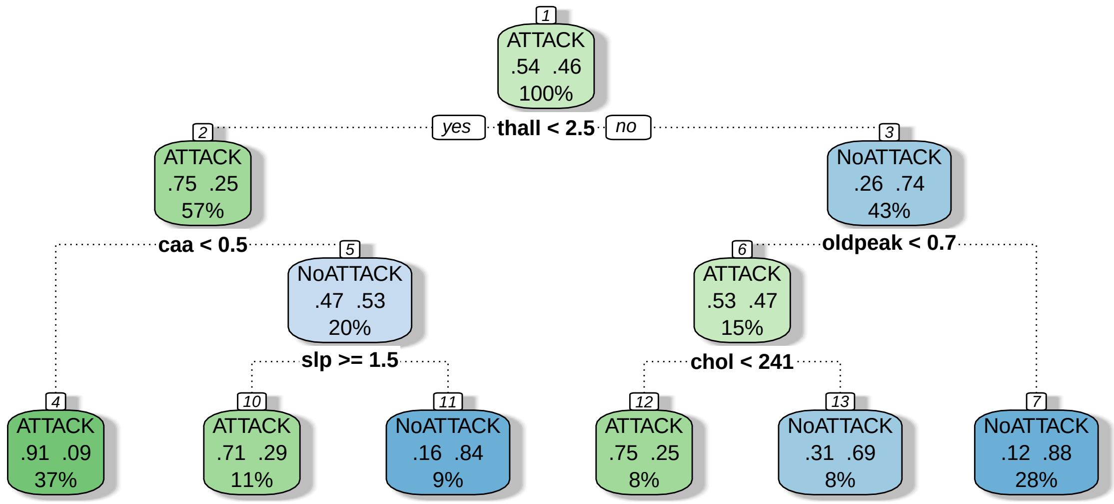
```
##   NoATTACK     18       79
##
##                Accuracy : 0.8451
##                  95% CI : (0.7894, 0.8909)
##     No Information Rate : 0.5587
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6869
##
##  Mcnemar's Test P-Value : 0.7277
##
##             Sensitivity : 0.8487
##             Specificity : 0.8404
##          Pos Pred Value : 0.8707
##          Neg Pred Value : 0.8144
##              Prevalence : 0.5587
##          Detection Rate : 0.4742
##    Detection Prevalence : 0.5446
##       Balanced Accuracy : 0.8446
##
##        'Positive' Class : ATTACK
##
```

```r
# test set confusion matrix
pred_test_fl <- predict(flTree, test_set)
cfm_test_fl <- confusionMatrix(test_set$output, pred_test_fl)
cfm_test_fl
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ATTACK NoATTACK
##   ATTACK       45        4
##   NoATTACK     14       27
##
##                Accuracy : 0.8
##                  95% CI : (0.7025, 0.8769)
##     No Information Rate : 0.6556
##     P-Value [Acc > NIR] : 0.001991
##
##                   Kappa : 0.5886
##
##  Mcnemar's Test P-Value : 0.033895
##
##             Sensitivity : 0.7627
##             Specificity : 0.8710
##          Pos Pred Value : 0.9184
##          Neg Pred Value : 0.6585
##              Prevalence : 0.6556
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5444
##       Balanced Accuracy : 0.8168
##
##        'Positive' Class : ATTACK
```

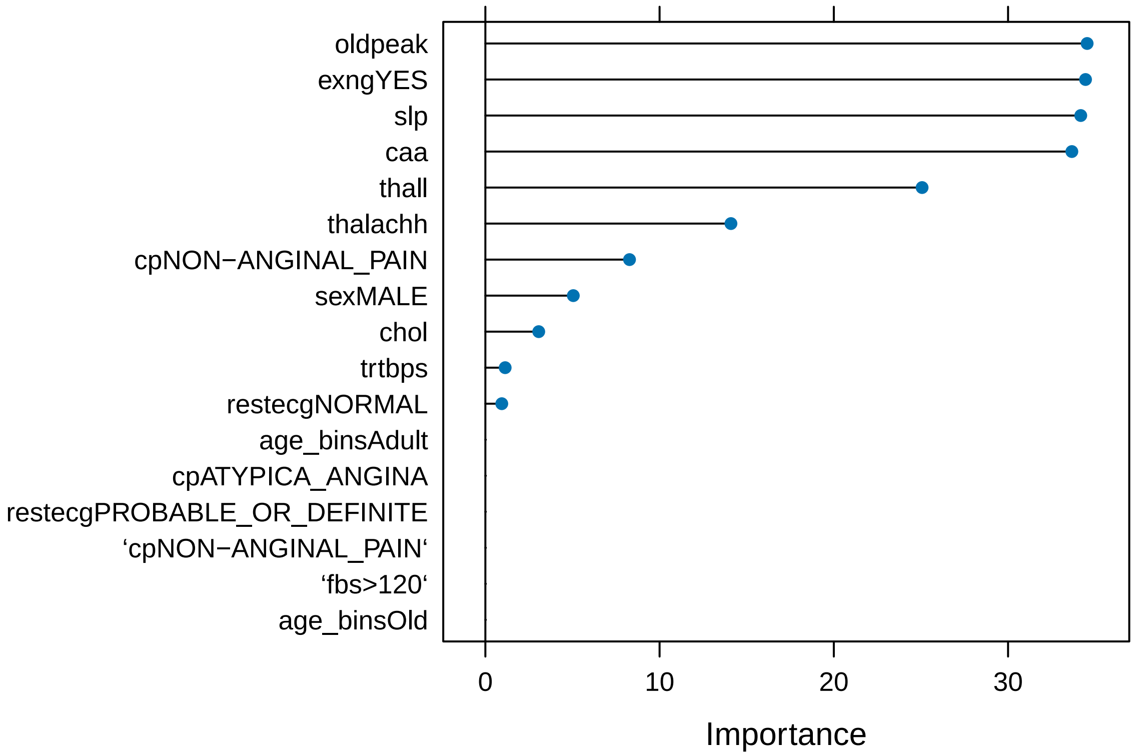
##

```r
# display the tree
fancyRpartPlot(flTree$finalModel, caption="Decision Tree")
```



Decision Tree

```r
# display the important features
plot(varImp(flTree, scale=FALSE))
```



g) Evaluation

From the above technique we see knn performing slightly better with test accuracy **84%** so we go ahead with computing other evaluation techniques for KNN

(1) 2X2 Confusion Matrix for KNN

As we had predicted the model with the test set in the previous steps we display the confusion matrix here.

From the confusion matrix we see the TP = 47 , TN = 29 , FN = 12, FP = 2.

```r
# Generate confusion matrix on the prediction
cmKnn = confusionMatrix(as.factor(test_set$output), pred_knnTest)
cmKnn
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ATTACK NoATTACK
##   ATTACK       47        2
##   NoATTACK     12       29
##
##                Accuracy : 0.8444
##                  95% CI : (0.7528, 0.9123)
##     No Information Rate : 0.6556
```

```
##     P-Value [Acc > NIR] : 5.172e-05
##
##                   Kappa : 0.68
##
##  Mcnemar's Test P-Value : 0.01616
##
##             Sensitivity : 0.7966
##             Specificity : 0.9355
##          Pos Pred Value : 0.9592
##          Neg Pred Value : 0.7073
##              Prevalence : 0.6556
##          Detection Rate : 0.5222
##    Detection Prevalence : 0.5444
##       Balanced Accuracy : 0.8660
##
##        'Positive' Class : ATTACK
##
```

```r
# confusion matrix of svm
m = cmKnn$table
```

(2) Precision and Recall for KNN

From the above confusion matrix we calculate the precision and recall manually and the result is same **96%** for precision and **80%** for Recall.(approx).

Precision is about predicting the positive prediction value which about calculating all TruePositives / (TruePositive + FalsePositives) and recall is same as sensitivity which is about calculating all TruePositives / (TruePositive + FalseNegatives).

```r
# precision TP/(TP+FP)
precision = m[1,1]/(m[1,1]+m[1,2])
precision # 0.9591837
```

```
## [1] 0.9591837
```

```r
# recall TP/(TP+FN) (recall)
recall = m[1,1]/(m[1,1]+m[2,1])
recall # 0.7966102
```

```
## [1] 0.7966102
```

(3) ROC Curve for KNN

As per the ROC Curve the AUC value is reported at **89%**.

A perfect classifier would have a ROC curve that passes through the top-left corner of the plot, indicating high sensitivity and low false positive rate across all thresholds. A random classifier, on the other hand, would produce a diagonal line from the bottom-left to the top-right of the plot.

And here we see a pretty decent curve if not perfect.

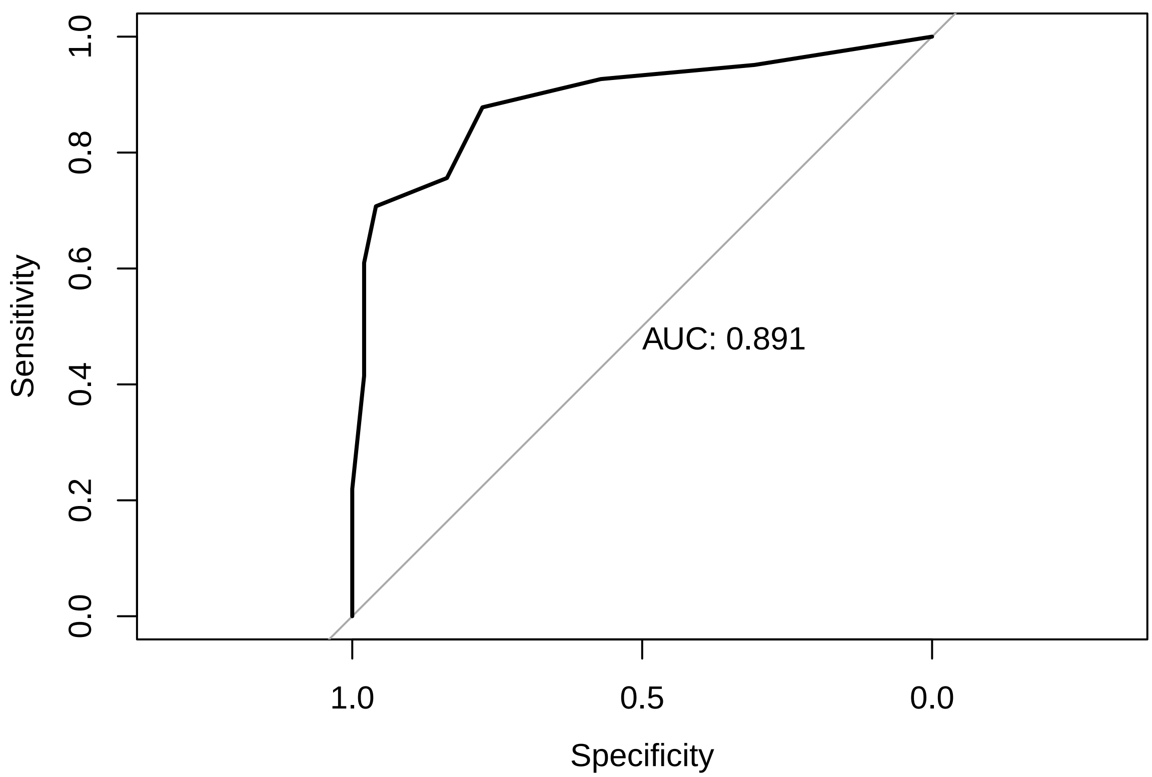
```r
# Get class probabilities for svm
pred_prob <- predict(kknn_fit, test_set, type = "prob")
```

```
## Warning in model.matrix.default(mt2, test, contrasts.arg = contrasts.arg):
## variable '.outcome' is absent, its contrast will be ignored
```

```r
head(pred_prob)
```

```
##      ATTACK  NoATTACK
## 1 0.5555556 0.4444444
## 2 0.5555556 0.4444444
## 3 0.7777778 0.2222222
## 4 1.0000000 0.0000000
## 5 0.8888889 0.1111111
## 6 0.7777778 0.2222222
```

```r
# plot the ROC
roc_obj <- roc((test_set$output), pred_prob[,1])
plot(roc_obj, print.auc=TRUE)
```



Conclusion: The performance metrics of a classifier can provide additional insights beyond just accuracy.

Precision: Precision is the proportion of correctly predicted positive instances (true positives) out of all the cases predicted as positive. A precision of 96% means that when the KNN classifier predicts a positive outcome, it is correct 96% of the time. This indicates that the KNN classifier has a low rate of false positives.

Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive instances that are correctly predicted by the classifier. With a recall of 80%, the KNN classifier correctly identifies 80% of the positive instances in the dataset. This indicates that the classifier has a moderate ability to avoid false negatives.

AUC: The Area Under the ROC Curve (AUC) measures a classifier's overall performance. An AUC of 86% indicates that the classifier has good discrimination power in distinguishing between positive and negative instances. The higher the AUC, the better the classifier correctly ranks positive instances higher than negative ones.

In comparison, accuracy measures the overall correctness of predictions, regardless of the class. An accuracy of 84% indicates that the classifier correctly predicted the class label for 84% of instances in the dataset.

Based on these performance metrics, the classifier performs well regarding precision, recall, and AUC. However, it's essential to consider a classification task's specific requirements and objectives. Depending on the application and the importance of false positives and false negatives, we may need to adjust the classification threshold or further optimize the model to achieve the desired balance between precision and recall.

h) Report

As part of the process of analyzing this data set, we performed the tasks below. Each task is labeled and explained sequentially in this markdown as you go through this file.

1. We performed **Data Gathering and Integration** by loading the data from the file. As we had just one data file, no integration step was involved.
2. We then performed **Data Exploration** and analyzed each variable, including distribution and correlations, supported by multiple visualization plots, to explain our initial hypothesis.
3. As part of **Data cleaning**, as we did not have any NAs, we performed additional visualization with some binning techniques and converted the required variable to factors.
4. As part of **Pre Processing**, we normalized the data and created dummy variables for further analysis.
5. Then we performed two **Clustering** techniques (kmean, hac) for which we also computed the PCA for visualizations.
6. As part of **Classification**, we performed two techniques(KNN, Decision tree) and accurately measured the model's performance over testing and training data.
7. As part of **Evaluation** we computed the *confusion matrix* with manual computation of *Precision and Recall* and also projected the *ROC Plot.* and compared various performance measures with accuracy reported for the model.

Overall, we implemented all the techniques for the data set we learned in the course.

Overall Takeaway

1. The dataset provided information about various variables related to heart health and potential risk factors for heart attacks.
2. Through data exploration, we gained insights into the distribution and correlations among the variables, which helped us better understand the dataset.
3. Different Clustering techniques, namely k-means and hierarchical agglomerative clustering (HAC), helped us identify potential patterns or groups within the data.
4. Classification techniques, including k-nearest neighbors (KNN) and decision tree, helped us predict heart attack risk based on the available variables.
5. With Model evaluation, we learned to measure performance metrics such as accuracy, precision, recall, and ROC plots to assess the effectiveness of the classification models.
6. Overall, the analysis provided insights into the dataset's variables, relationships, and potential predictive power in identifying individuals at risk of heart attacks.

The most interesting aspect of the analysis was how we used an ROC curve to summarize the model's discriminatory ability and how we used different techniques to achieve that.

i) Reflection

This course taught us different ML techniques for analyzing a data set. Techniques like clustering and classification were of great interest. We learned the Data science pipeline steps of studying, which we followed in this project. During the process, we learned that 80% of our time goes with data cleaning and preprocessing, and the rest is 20%. We also learned some ethics, which we must consider in our daily activities concerning data science.

After learning these techniques, it is clear that Data Science is not Black Magic, and it has to be done properly and with the correct methods. ;)