Team 1: Framingham Heart Study CHD Predictions

Kaiyu Wang, Chinar Boolchandani

I. Setup

```
library(readr)
library(data.table)
library(ggplot2)
library(dplyr)
library(reshape2)
library(glmnet)
library(ROCR)
library(pROC)
library(PRROC)
library(lattice)
library(caret)
library(e1071)
library(randomForest)
library(corrplot)
library(xgboost)
library(stringr)
library(magrittr)

CHD <-fread("framingham.csv")
```

II. Clean Data

1. Summary

Summary statistics for all features.

```
summary(CHD)
```

```
##       male             age          education     currentSmoker
##  Min.   :0.0000   Min.   :32.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:42.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :49.00   Median :2.000   Median :0.0000
##  Mean   :0.4292   Mean   :49.58   Mean   :1.979   Mean   :0.4941
##  3rd Qu.:1.0000   3rd Qu.:56.00   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :70.00   Max.   :4.000   Max.   :1.0000
##                                   NA's   :105
##    cigsPerDay         BPMeds        prevalentStroke     prevalentHyp
##  Min.   : 0.000   Min.   :0.00000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.: 0.000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.0000
##  Median : 0.000   Median :0.00000   Median :0.000000   Median :0.0000
##  Mean   : 9.003   Mean   :0.02963   Mean   :0.005899   Mean   :0.3105
##  3rd Qu.:20.000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:1.0000
##  Max.   :70.000   Max.   :1.00000   Max.   :1.000000   Max.   :1.0000
##  NA's   :29       NA's   :53
##     diabetes          totChol          sysBP           diaBP
##  Min.   :0.00000   Min.   :107.0   Min.   : 83.5   Min.   : 48.00
##  1st Qu.:0.00000   1st Qu.:206.0   1st Qu.:117.0   1st Qu.: 75.00
##  Median :0.00000   Median :234.0   Median :128.0   Median : 82.00
##  Mean   :0.02572   Mean   :236.7   Mean   :132.4   Mean   : 82.89
##  3rd Qu.:0.00000   3rd Qu.:263.0   3rd Qu.:144.0   3rd Qu.: 89.88
##  Max.   :1.00000   Max.   :696.0   Max.   :295.0   Max.   :142.50
##                    NA's   :50
##       BMI          heartRate         glucose         TenYearCHD
##  Min.   :15.54   Min.   : 44.00   Min.   : 40.00   Min.   :0.000
##  1st Qu.:23.07   1st Qu.: 68.00   1st Qu.: 71.00   1st Qu.:0.000
##  Median :25.40   Median : 75.00   Median : 78.00   Median :0.000
##  Mean   :25.80   Mean   : 75.88   Mean   : 81.97   Mean   :0.152
##  3rd Qu.:28.04   3rd Qu.: 83.00   3rd Qu.: 87.00   3rd Qu.:0.000
##  Max.   :56.80   Max.   :143.00   Max.   :394.00   Max.   :1.000
##  NA's   :19      NA's   :1        NA's   :388
```

2. Replace NA

Missing data is replaced with median since all of them all numerical medical data.

```
education_median<-median(CHD$education,na.rm=TRUE)
CHD[is.na(education),education:=education_median]

cigsPerDay_median<-median(CHD$cigsPerDay,na.rm=TRUE)
CHD[is.na(cigsPerDay),cigsPerDay:=cigsPerDay_median]

BPMeds_median<-median(CHD$BPMeds,na.rm=TRUE)
CHD[is.na(BPMeds),BPMeds:=BPMeds_median]

totChol_median<-median(CHD$totChol,na.rm=TRUE)
CHD[is.na(totChol),totChol:=totChol_median]

glucose_median<-median(CHD$glucose,na.rm=TRUE)
CHD[is.na(glucose),glucose:=glucose_median]

heartRate_median<-median(CHD$heartRate,na.rm=TRUE)
CHD[is.na(heartRate),heartRate:=heartRate_median]

BMI_median<-median(CHD$BMI,na.rm=TRUE)
CHD[is.na(BMI),BMI:=BMI_median]
```

3. Rename Column male

```
colnames(CHD)[1] <- 'is_male'
```

III. Descriptive Data Analysis

Data Transformation for better data visualization

1. Convert categorical data to dummy variables.
2. Create new columns as feature engineering.
3. Drop unnecessary columns.

```
CHD2 <- CHD %>%
  mutate(is_male = if_else (is_male ==1,"Male","Female"),
         currentSmoker = if_else (currentSmoker ==1,"Smoker","Not a smoker"),
         BPMeds = if_else (BPMeds ==1,"BP meds","No BP meds"),
         prevalentStroke = if_else (prevalentStroke ==1,"Stroke","No Stroke"),
         prevalentHyp = if_else (prevalentHyp ==1,"Hypertensive Yes","Hypertensive No
"),
         diabetes = if_else (diabetes ==1,"Has diabetes","No diabetes"),
         TenYearCHD = if_else (TenYearCHD ==1,"Has CHD","No CHD"),
         education = as.factor(education)) %>%
  mutate_if(is.character,as.factor) %>%
  dplyr::select(TenYearCHD,is_male,currentSmoker,BPMeds,prevalentStroke,prevalentHyp,
diabetes,everything())

#new columns creation
CHD2$BP <- CHD2$sysBP + CHD2$diaBP

#dropping cols
CHD2$sysBP = NULL
CHD2$diaBP = NULL
```

1. Distribution of Ten Year Risk of CHD

Labeled data shows only 15% risk of ten year CHD.

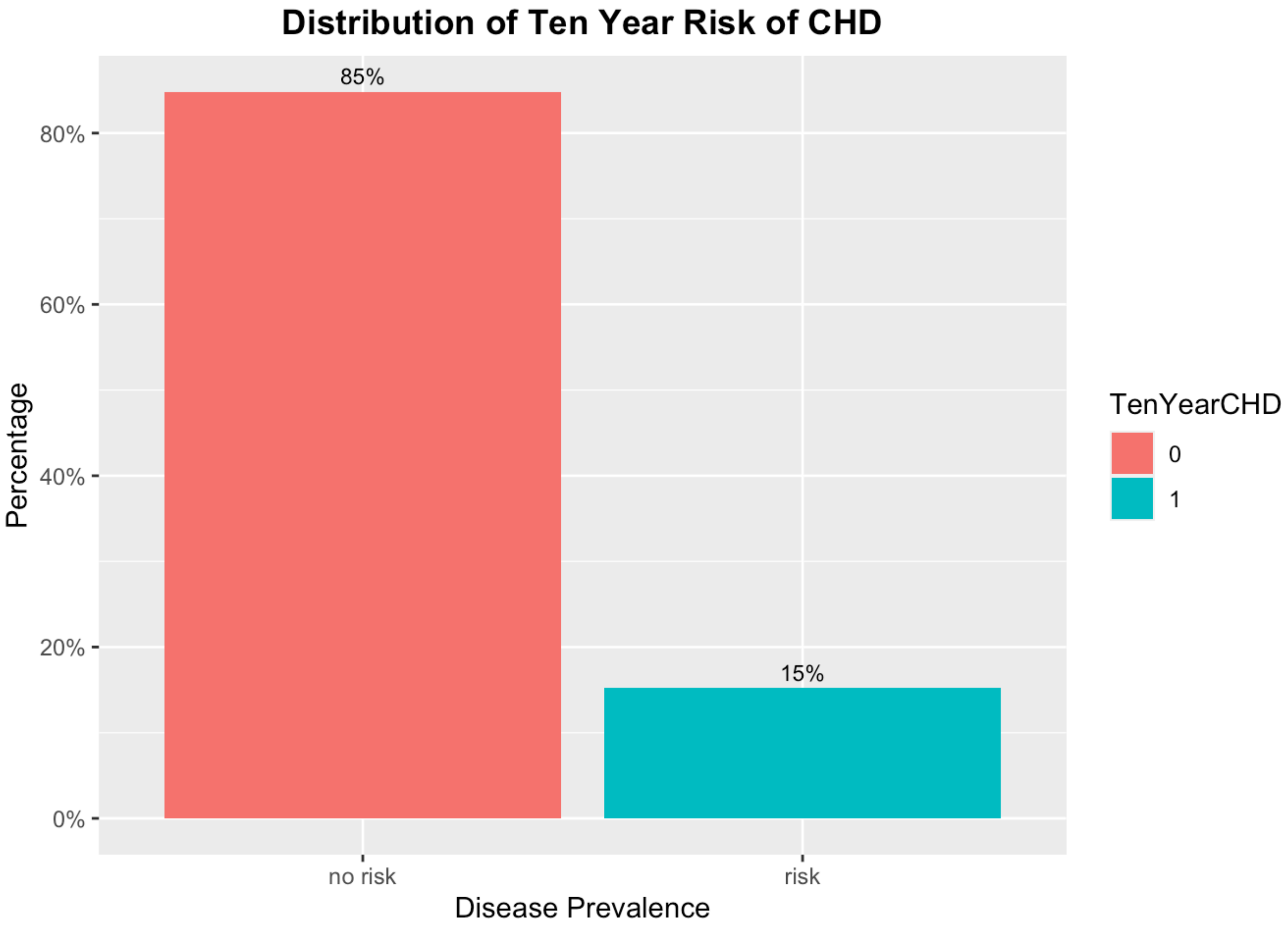
```
count1 <- length(which(CHD$TenYearCHD == 1))
cat(count1, "people have a 10 year risk of CHD\n")
```

```
## 644 people have a 10 year risk of CHD
```

```
count2 <- length(which(CHD$TenYearCHD == 0))
cat(count2, "people DO NOT have a 10 year risk of CHD")
```

```
## 3594 people DO NOT have a 10 year risk of CHD
```

```
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))

ggplot(data = CHD, aes(x = factor(TenYearCHD),
                          y = prop.table(stat(count)),
                          fill = factor(TenYearCHD),
                          label = scales::percent(prop.table(stat(count))))) +
    geom_bar(position = "dodge") +
    geom_text(stat = 'count',
              position = position_dodge(.9),
              vjust = -0.5,
              size =3) +
    scale_x_discrete(labels = c("no risk", "risk")) +
    scale_y_continuous(labels = scales::percent) +
    labs(x = 'Disease Prevalence', y = 'Percentage',fill='TenYearCHD') +
    ggtitle("Distribution of Ten Year Risk of CHD") +
    common_theme
```



##There is some imbalance in the dataset as seen from plot above, 15% of records belong to peope with CHD in Ten years whereas 85% people belong to the class of No CHD in Ten years, we will deal with imbalance while splitting our data

2. Distribution of Percentage of CHD with Age

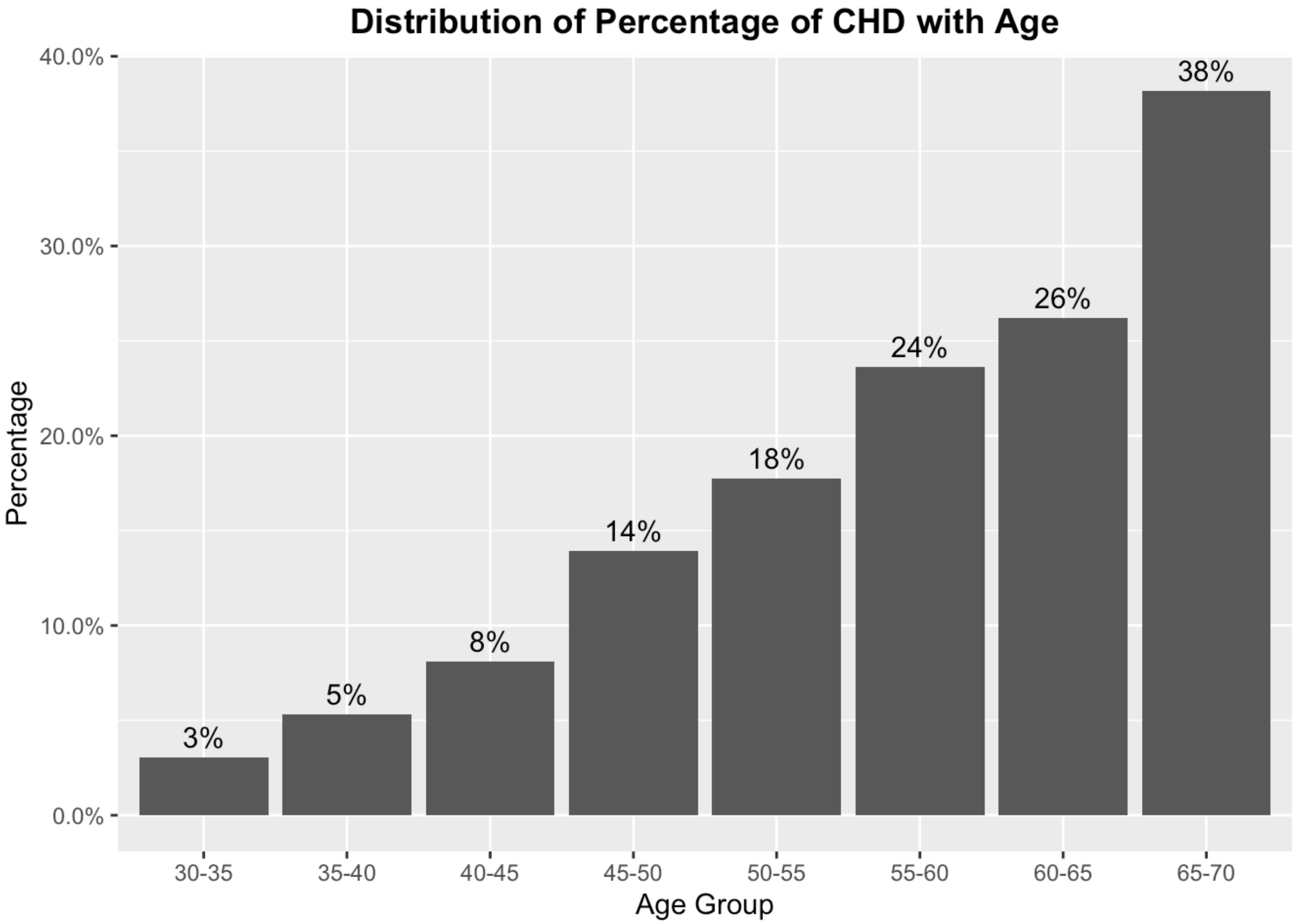
As age increases, the percentage of CHD increases.

```
CHD_a<-data.frame(CHD)
CHD_a$agec <-
  cut(CHD_a$age, breaks = c(30,35,40,45,50,55,60,65,70),
      labels = c("30-35","35-40","40-45","45-50","50-55","55-60","60-65","65-70"))

d <- CHD_a %>% group_by(agec) %>% summarise(perc = mean(TenYearCHD=='1'))
d$perc_r <- round(d$perc,2)*100
d$perc_r <- interaction(d$perc_r, "%", sep = "")
d
```

agec	**perc**	**perc_r**
<fct>	<dbl>	<fct>
30-35	0.03030303	3%
35-40	0.05294118	5%
40-45	0.08085612	8%
45-50	0.13932292	14%
50-55	0.17721519	18%
55-60	0.23608769	24%
60-65	0.26226013	26%
65-70	0.38181818	38%

8 rows

```
ggplot(d, aes(x = agec, y = perc)) +
  geom_col() +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = perc_r), vjust = -0.5) +
  labs(x = 'Age Group', y = 'Percentage') +
  ggtitle("Distribution of Percentage of CHD with Age") +
  common_theme
```

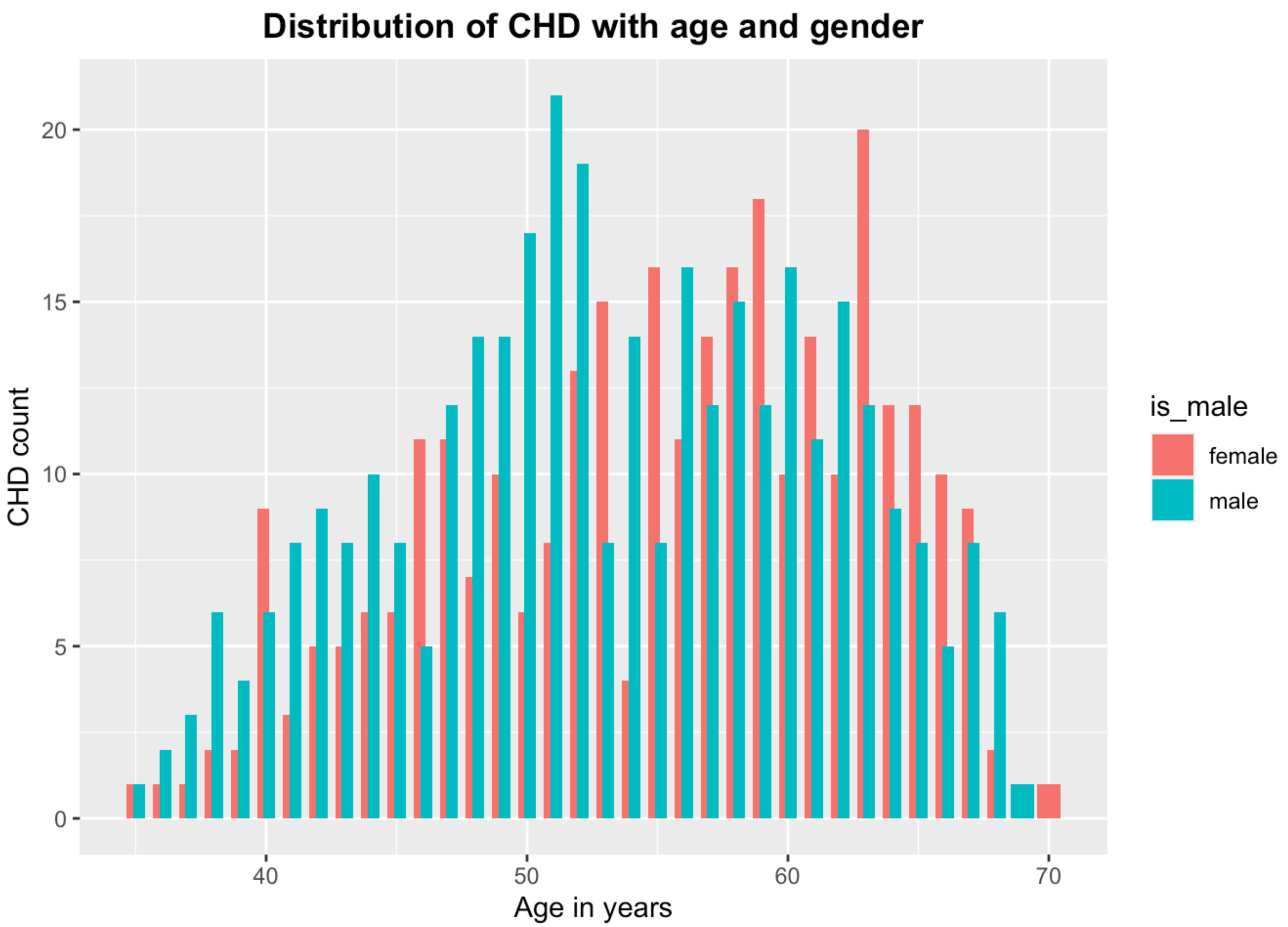


## Above plot shows people in age groups 65 and above are more likely to have CHD in Ten Years

3. Histogram of CHD with Age and Gender

Males are more likely to have CHD than females at earlier ages.

```
CHD_1 <- CHD[ CHD$TenYearCHD=='1',]
CHD_1$is_male[CHD_1$is_male == 0] <- "female"
CHD_1$is_male[CHD_1$is_male == 1] <- "male"
ggplot(data=CHD_1, aes(age,fill=is_male)) +
  geom_bar(position = position_dodge(width = 0.5)) +
  labs(x = "Age in years", y = "CHD count") +
  ggtitle("Distribution of CHD with age and gender") +
  common_theme
```



##From above distribution plot we can see Females have more tendency to have CHD after 60 years whereas in males it is more prevalent after 50 years.

4. Probability of Disease in Smokers

Smoking or not doesn't have significant impact on risk of ten year CHD.

```
d2 <- CHD %>% group_by(currentSmoker) %>% summarise(perc = mean(TenYearCHD=='1'))
d2
```

currentSmoker <int>	perc <dbl>
0	0.1450560
1	0.1590258

2 rows

From above table we see people being a current smoker does not show strong positive relationship with TenYearCHD

5. Line Chart of Percentage of CHD with Age and Gender

```
d3 <- CHD_a %>% group_by(agec,factor(is_male)) %>% summarise(perc = mean(TenYearCHD==
'1'))
```

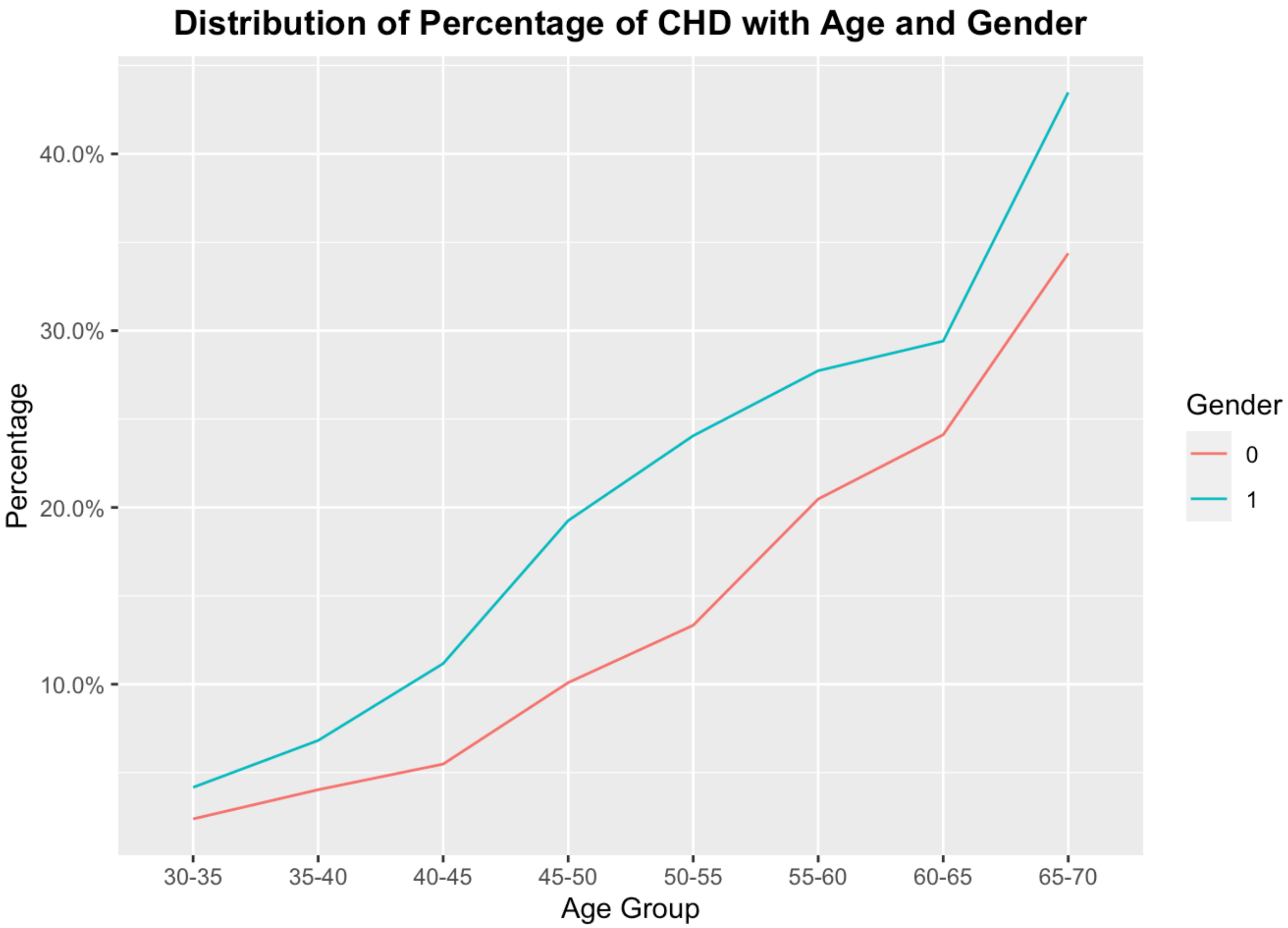
```
## `summarise()` has grouped output by 'agec'. You can override using the `.groups` a
rgument.
```

```
d3
```

agec <fct>	factor(is_male) <fct>	perc <dbl>
30-35	0	0.02380952
30-35	1	0.04166667
35-40	0	0.04032258
35-40	1	0.06818182
40-45	0	0.05482456
40-45	1	0.11168831
45-50	0	0.10089686
45-50	1	0.19254658
50-55	0	0.13333333
50-55	1	0.24054983

1-10 of 16 rows Previous **1** 2 Next

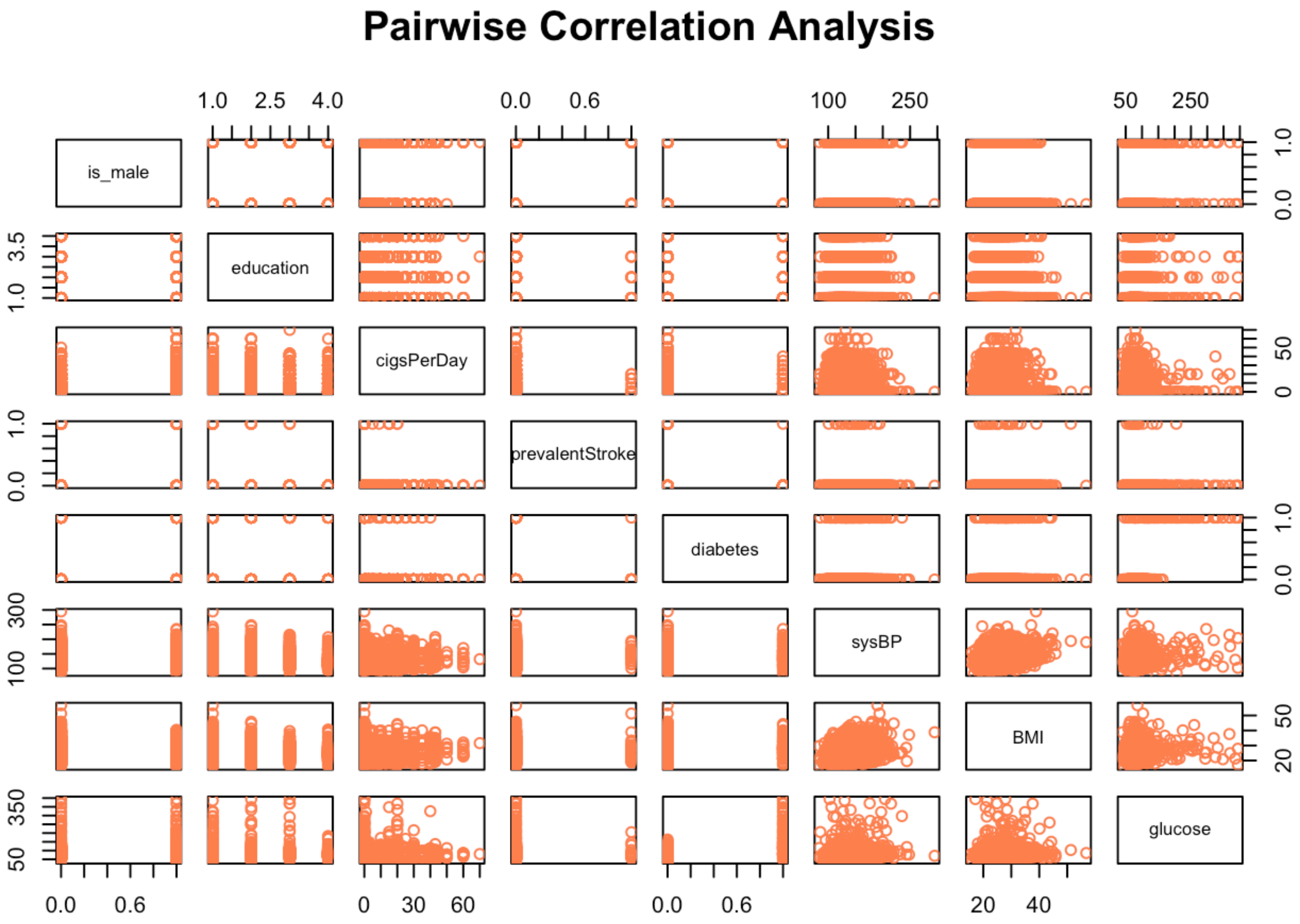
```
ggplot() +
  geom_line(data = d3,
            aes(agec, perc, group = `factor(is_male)`, color = `factor(is_male)`)) +
  scale_y_continuous(labels = scales::percent) +
  labs(x = 'Age Group', y = 'Percentage', color = 'Gender') +
  ggtitle("Distribution of Percentage of CHD with Age and Gender") +
  common_theme
```



###Above plot shows for females tend to have more CHD and the prevalence incrases in higher age groups compared to males

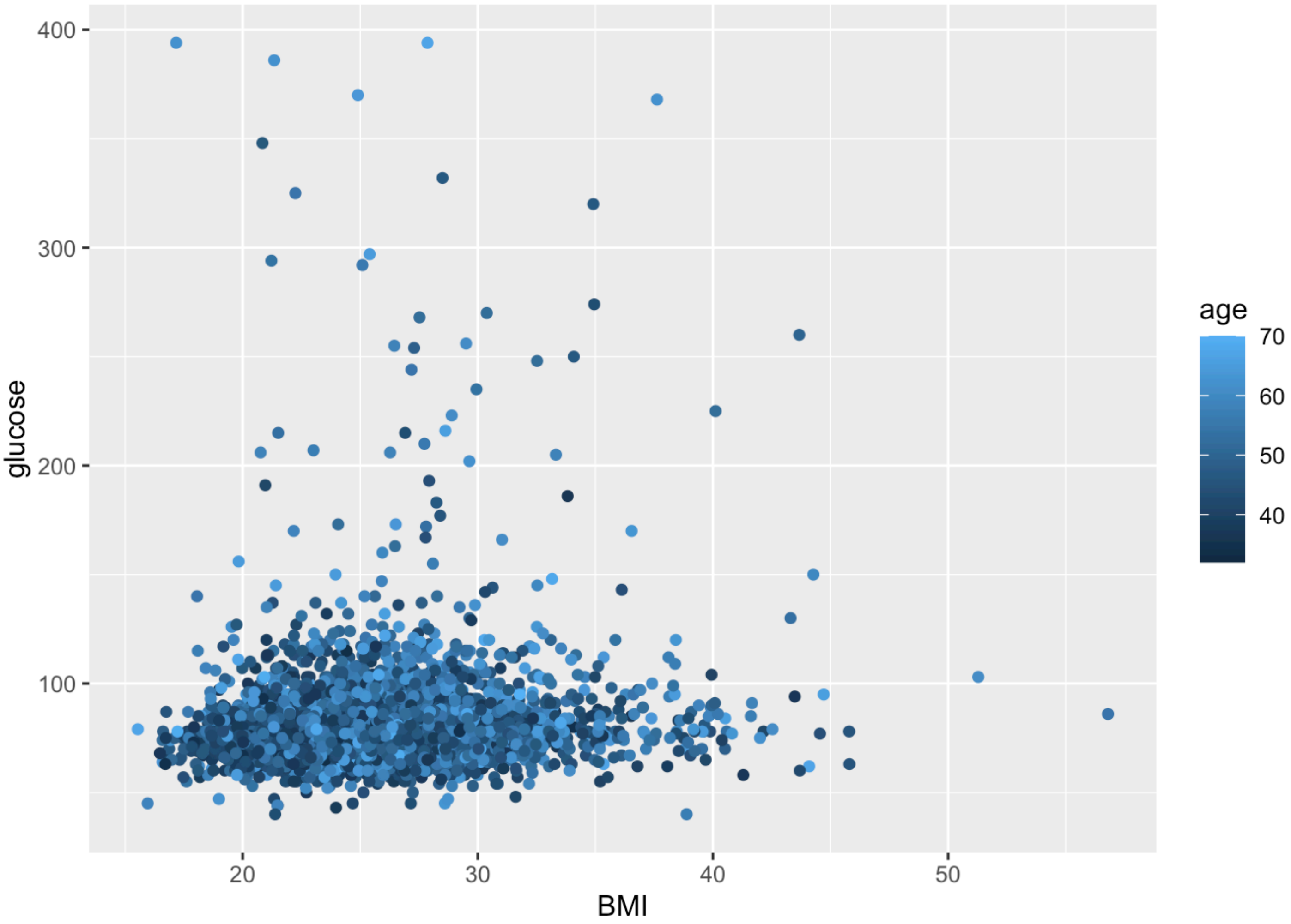
6. Pairwise Correlation Analysis

```
a <- CHD[,c(1,3,5,7,9,11,13,15)]
pairs(a, col = "coral",main = "Pairwise Correlation Analysis")
```



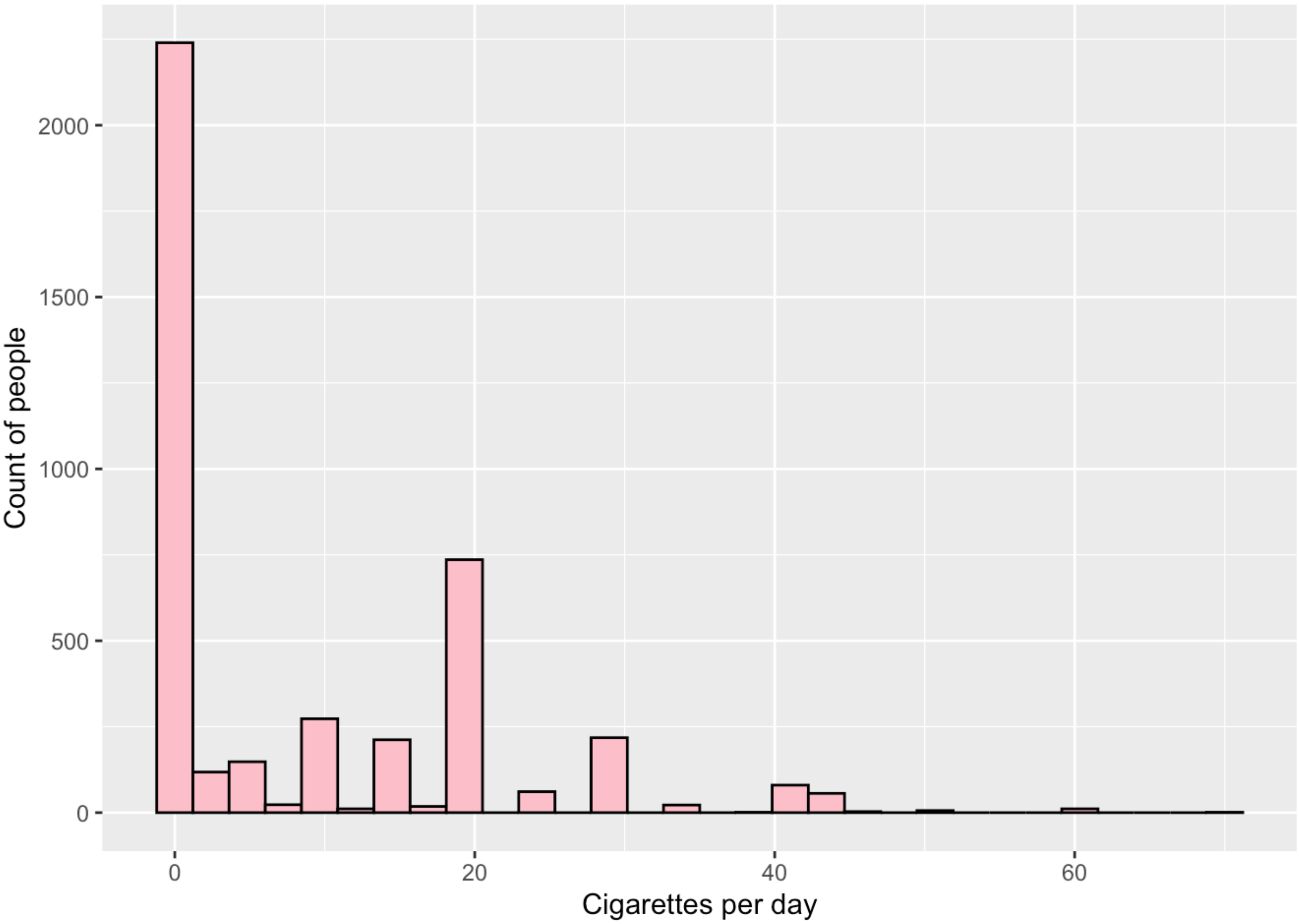
7. Distribution of BMI and Cigarettes per day

```
ggplot(data = CHD, aes(BMI, glucose, color = age)) +
  geom_point(fill = "blue")
```



```
ggplot(data = CHD, aes(x = cigsPerDay)) +
  geom_histogram(color = "black", fill = "pink") +
  labs(x = 'Cigarettes per day', y = 'Count of people')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

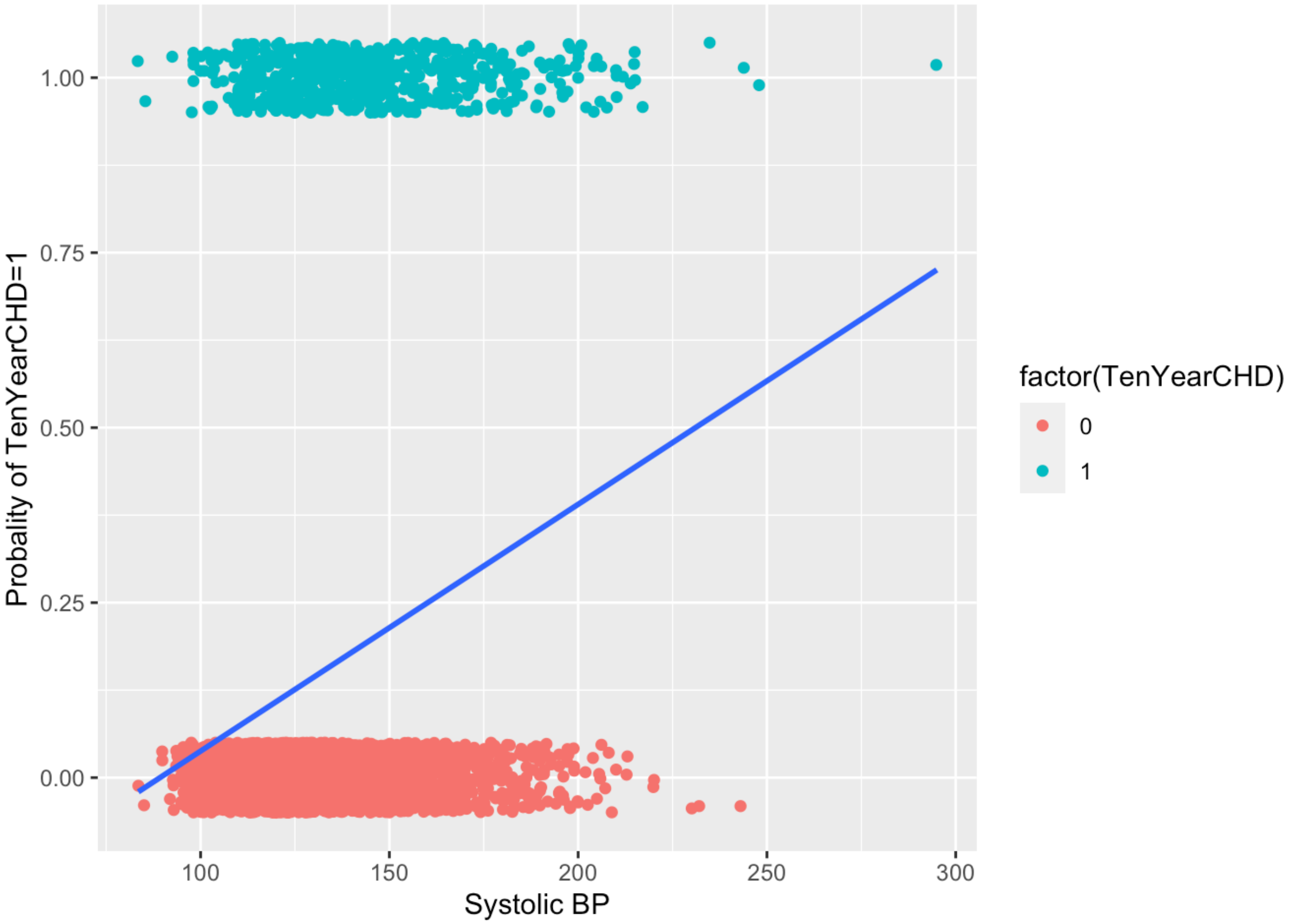


## Many people are in our dataset dont smoke and very few smoke

8.Probability distribution of TenYearCHD=1 with systolic BP

```
CHD %>% mutate(TenYearCHD = as.numeric(TenYearCHD)) %>%
        ggplot(aes(x=sysBP, y=TenYearCHD)) +
        geom_jitter(height = .05, aes(color = factor(TenYearCHD))) +
        geom_smooth(method = "lm",  se = FALSE) +
        ylab("Probality of TenYearCHD=1") + xlab("Systolic BP")
```

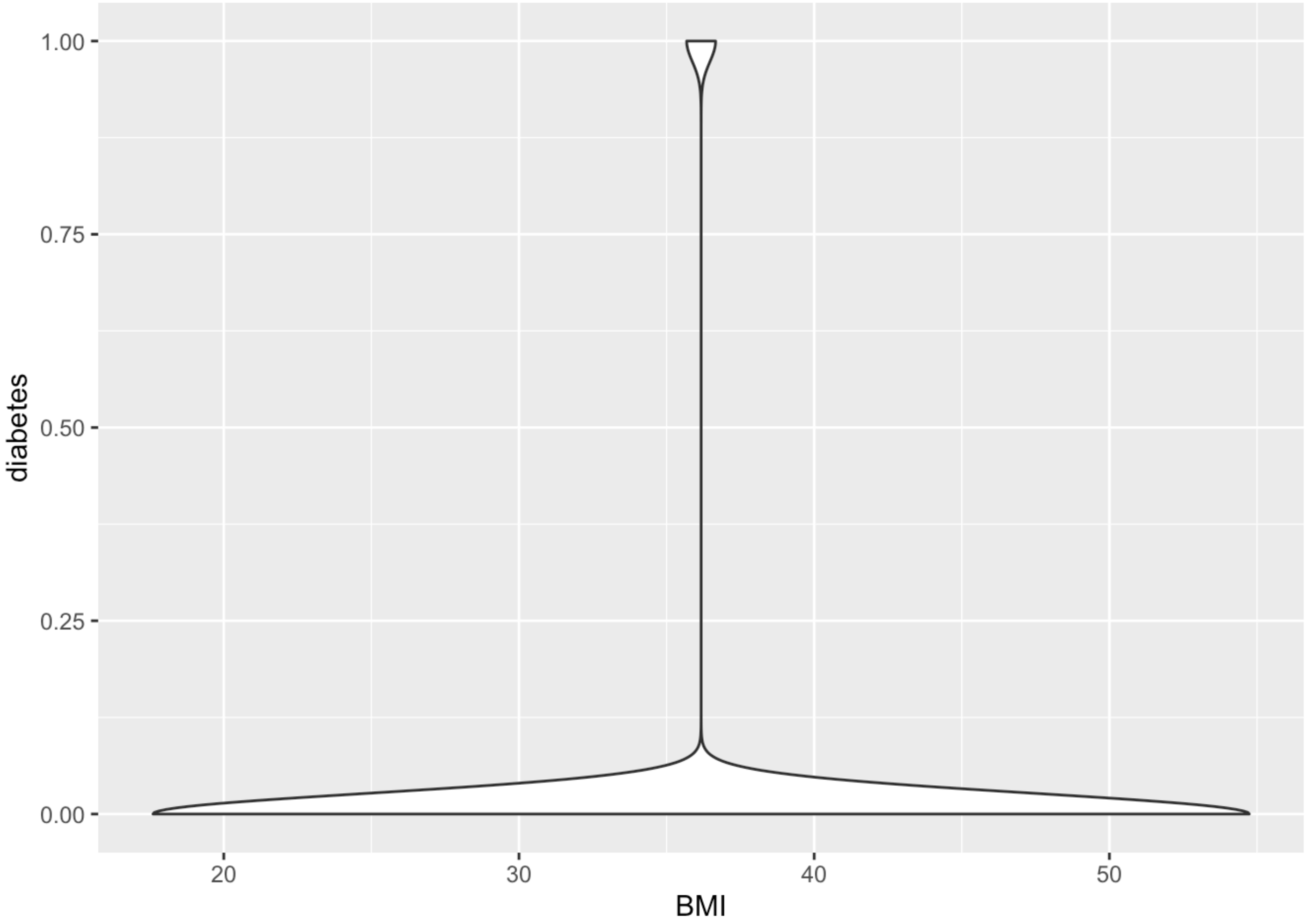
```
## `geom_smooth()` using formula 'y ~ x'
```



### Higher BPs are a lot more prevalent in people with higher systolic BPs in our data set; Additionally, a linear fit of probability of hearth disease over systolic BP shows a much higher probability of disease as the BP increases

9.Probability distribution of TenYearCHD=1 with systolic BP

```
ggplot(data=CHD, aes(x = BMI, y = diabetes, hue = TenYearCHD)) +
  geom_violin(data = CHD)
```



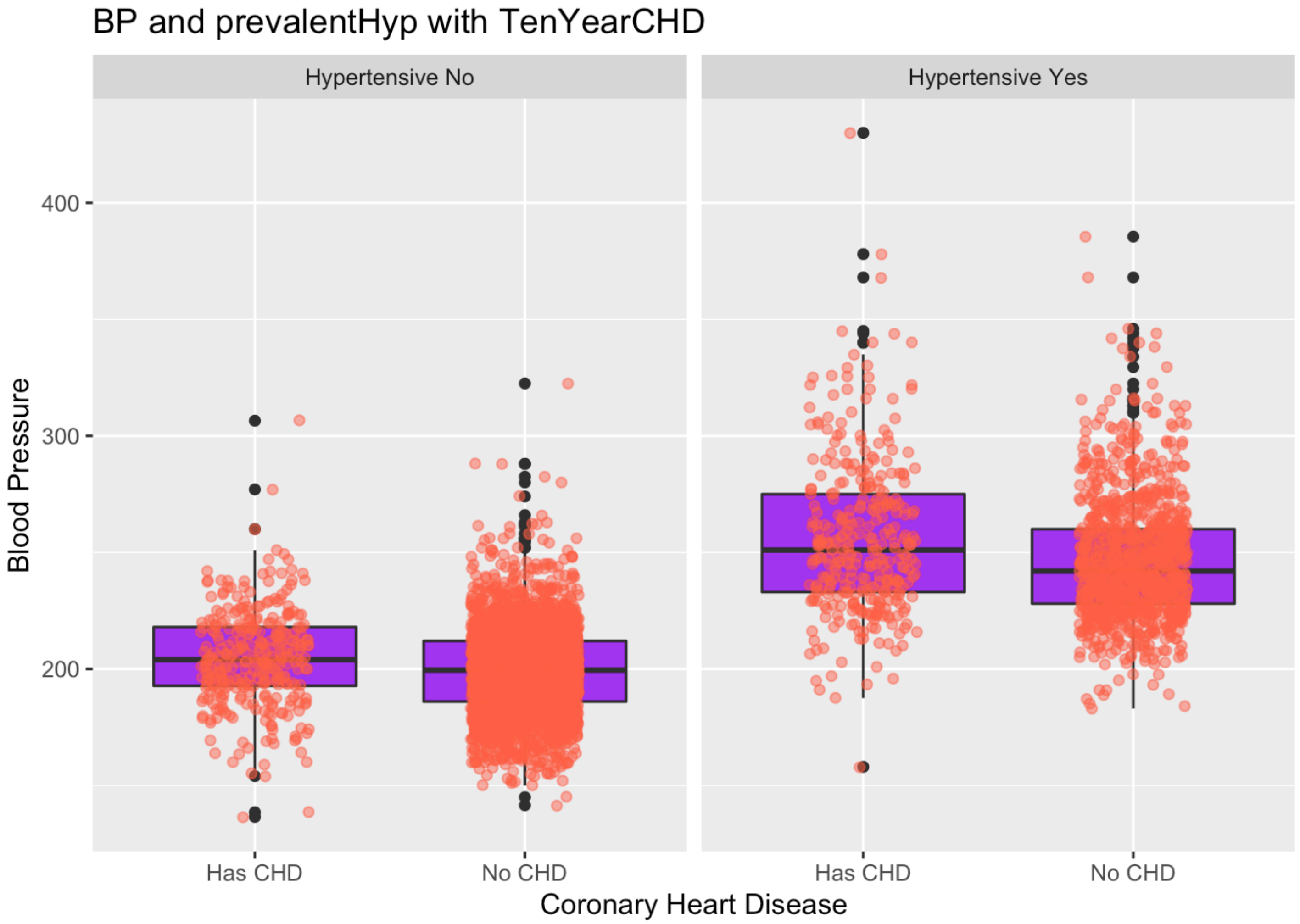
```
geom_histogram(binwidth=1,position="identity", alpha=0.5)
```

```
## geom_bar: na.rm = FALSE, orientation = NA
## stat_bin: binwidth = 1, bins = NULL, na.rm = FALSE, orientation = NA, pad = FALSE
## position_identity
```

##While our data has BMI ranging from <20 to >50, everyone with diabetes seems to have a BMI in the range of 35-40 which is in the obesity range, showing a clear correlation between diabetes prevalence and obesity

10. Relationship of BP and Prevalent Hypertension with TenYearCHD

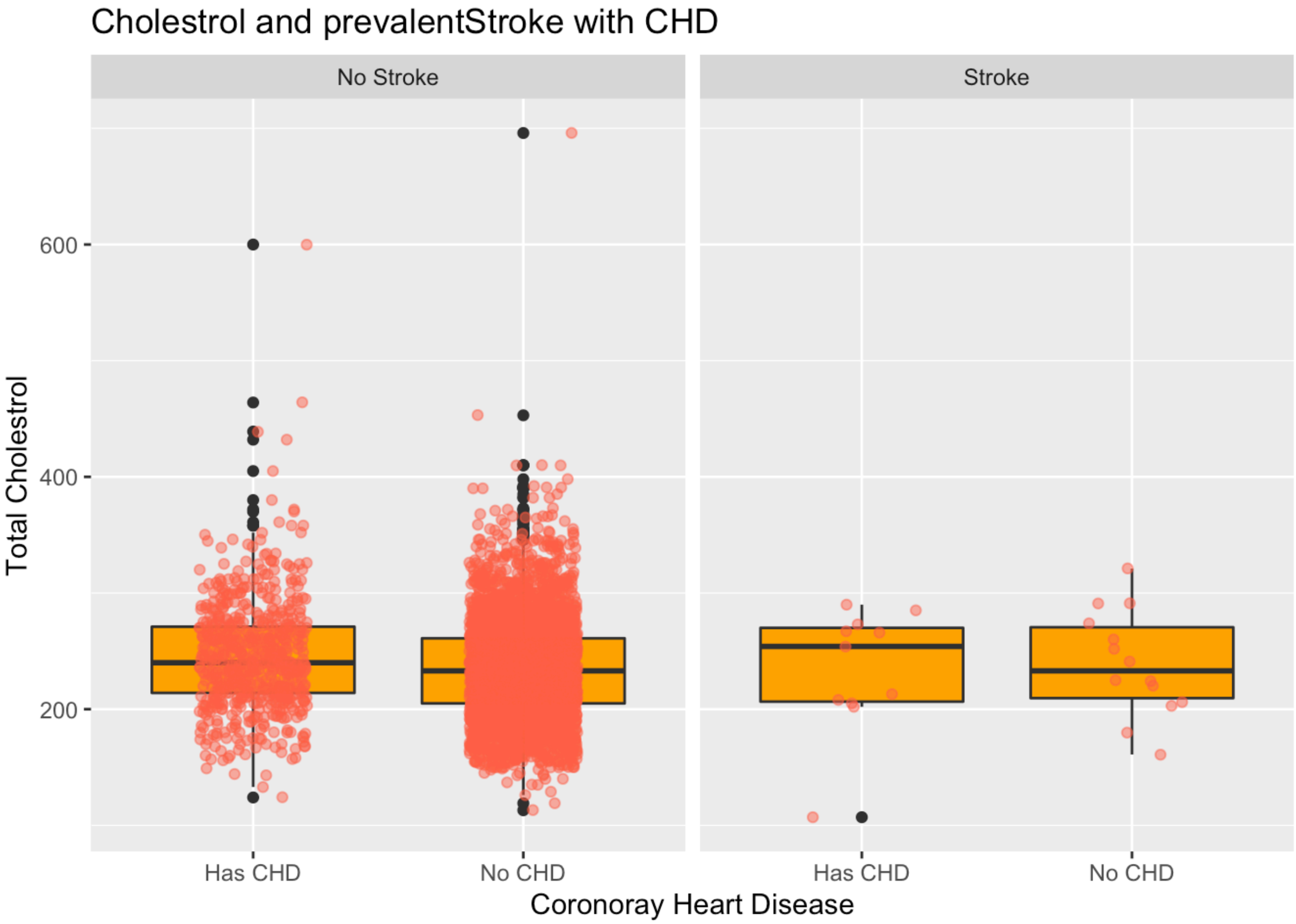
```
CHD2 %>%
  ggplot(aes(x = TenYearCHD, y = BP)) +
  geom_boxplot(fill = 'purple') +
  xlab("Coronary Heart Disease") +
  ylab("Blood Pressure") +
  facet_grid( ~ prevalentHyp) +
  ggtitle("BP and prevalentHyp with TenYearCHD") +
  geom_jitter(
    alpha = 0.5,
    width = 0.2,
    height = 0.2,
    color = "tomato"
  )
```



## People with prevalent Hypertension and high median BP are most likely to have CHD in Ten Years

11. Relationship of TotChol and Prevalent Stroke with TenYearCHD

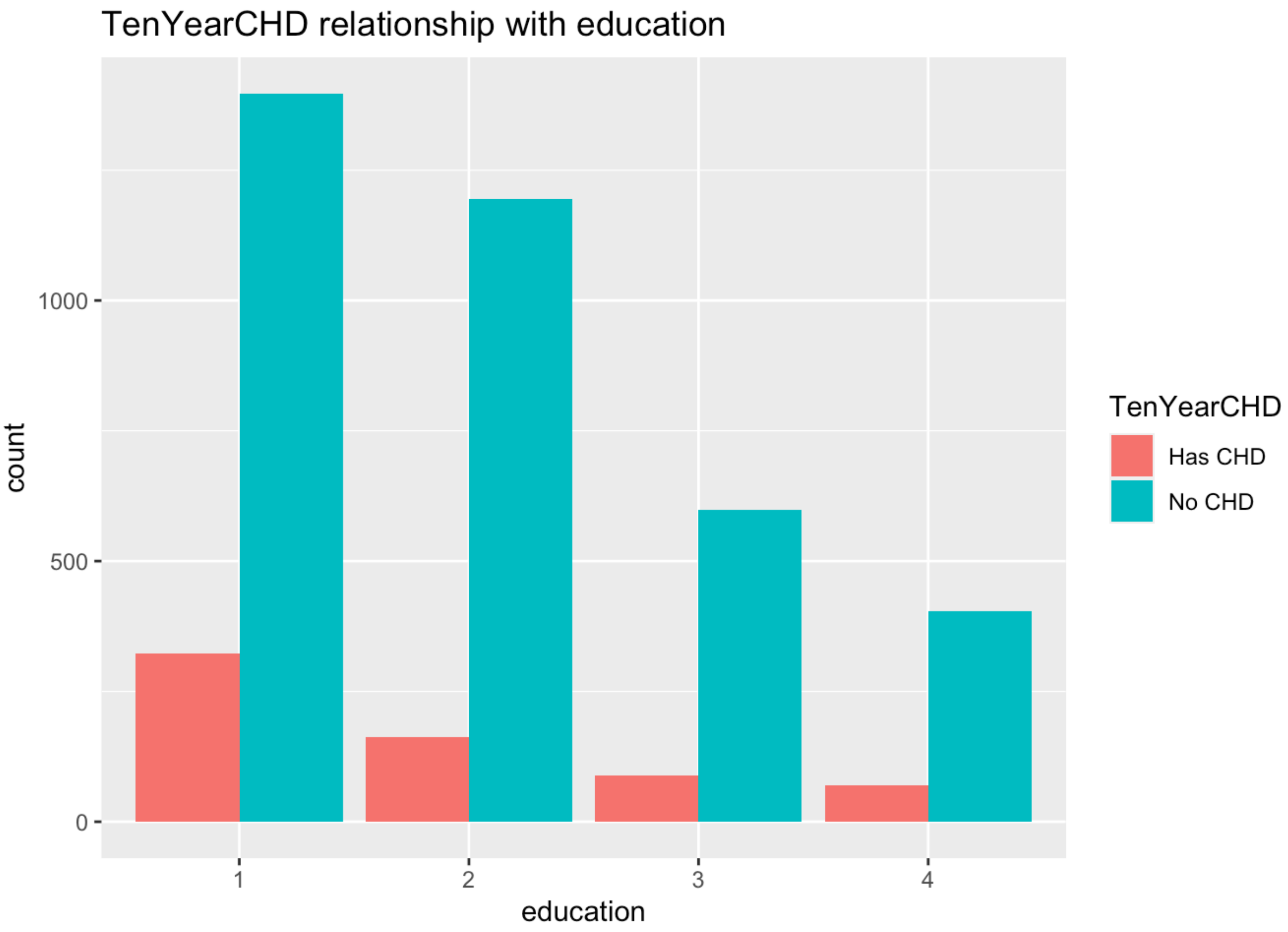
```
CHD2 %>%
  ggplot(aes(x = TenYearCHD, y = totChol)) +
  geom_boxplot(fill = 'orange') +
  xlab("Coronoray Heart Disease") +
  ylab("Total Cholestrol") +
  facet_grid( ~ prevalentStroke) +
  geom_jitter(
    alpha = 0.5,
    width = 0.2,
    height = 0.2,
    color = "tomato"
  ) +
  ggtitle("Cholestrol and prevalentStroke with CHD")
```



## Irrespective of prevalent Stroke yes or no, people who in general have higher median TotChol levels tend me be more probable to have CHD in Ten years

12. Distribution of TotChol within gender and Prevalent Stroke

```
ggplot(data = CHD2 ,aes(x = education, fill = TenYearCHD)) +
    geom_bar(position = "dodge")+
  ggtitle("TenYearCHD relationship with education")
```

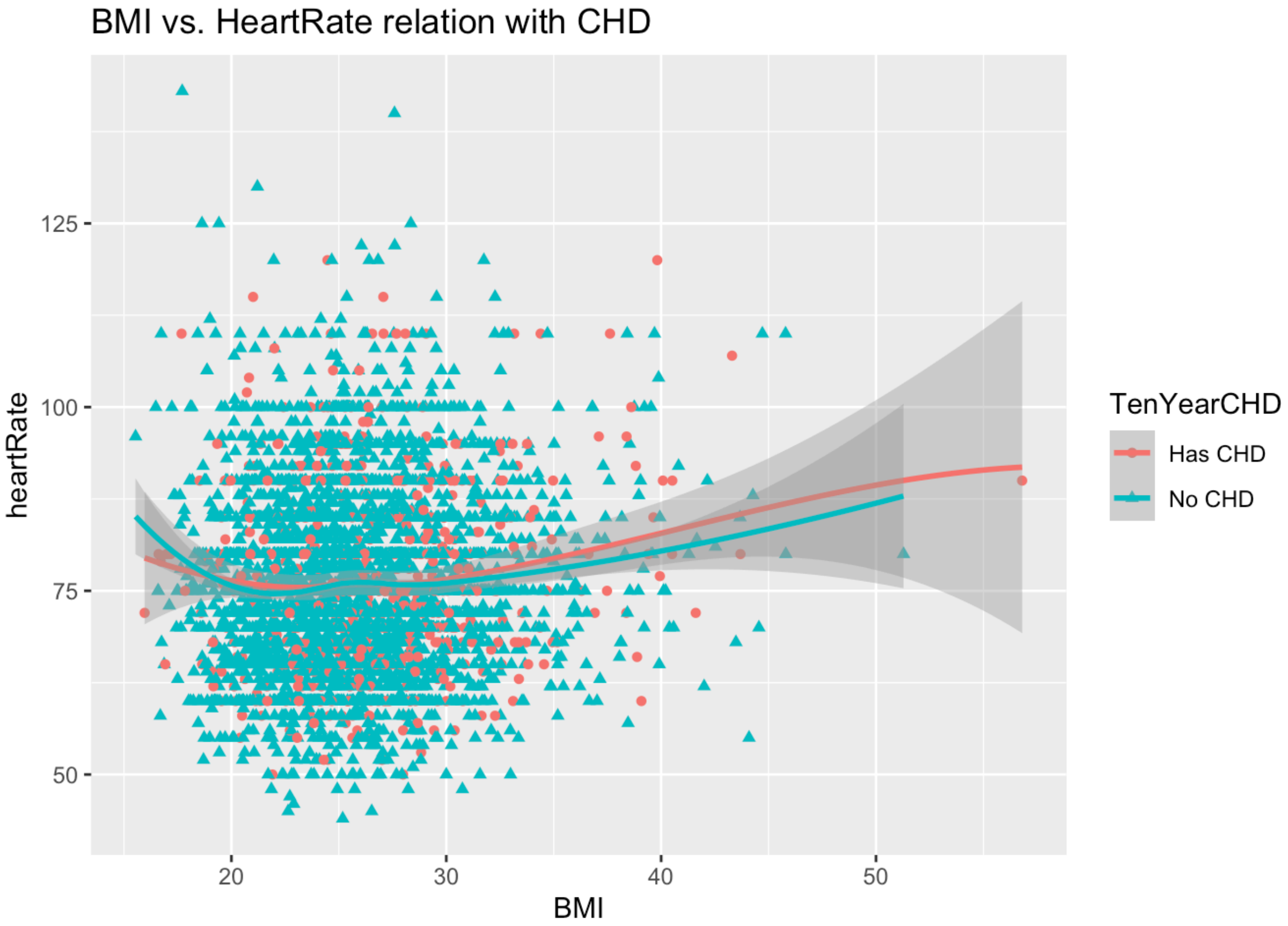


## education level 1 belongs to lowest and 4 is the highest. Above plot shows that TenYearCHD is more probable in people with lower levels of education.

13. TenYearCHD relationship with BMI and heartrate

```
ggplot(data = CHD2,
       aes(x = BMI, y = heartRate,
           color = TenYearCHD, shape = TenYearCHD)) +
    geom_point() +
    geom_smooth(method = "loess") +
    labs(title = "BMI vs. HeartRate relation with CHD")
```

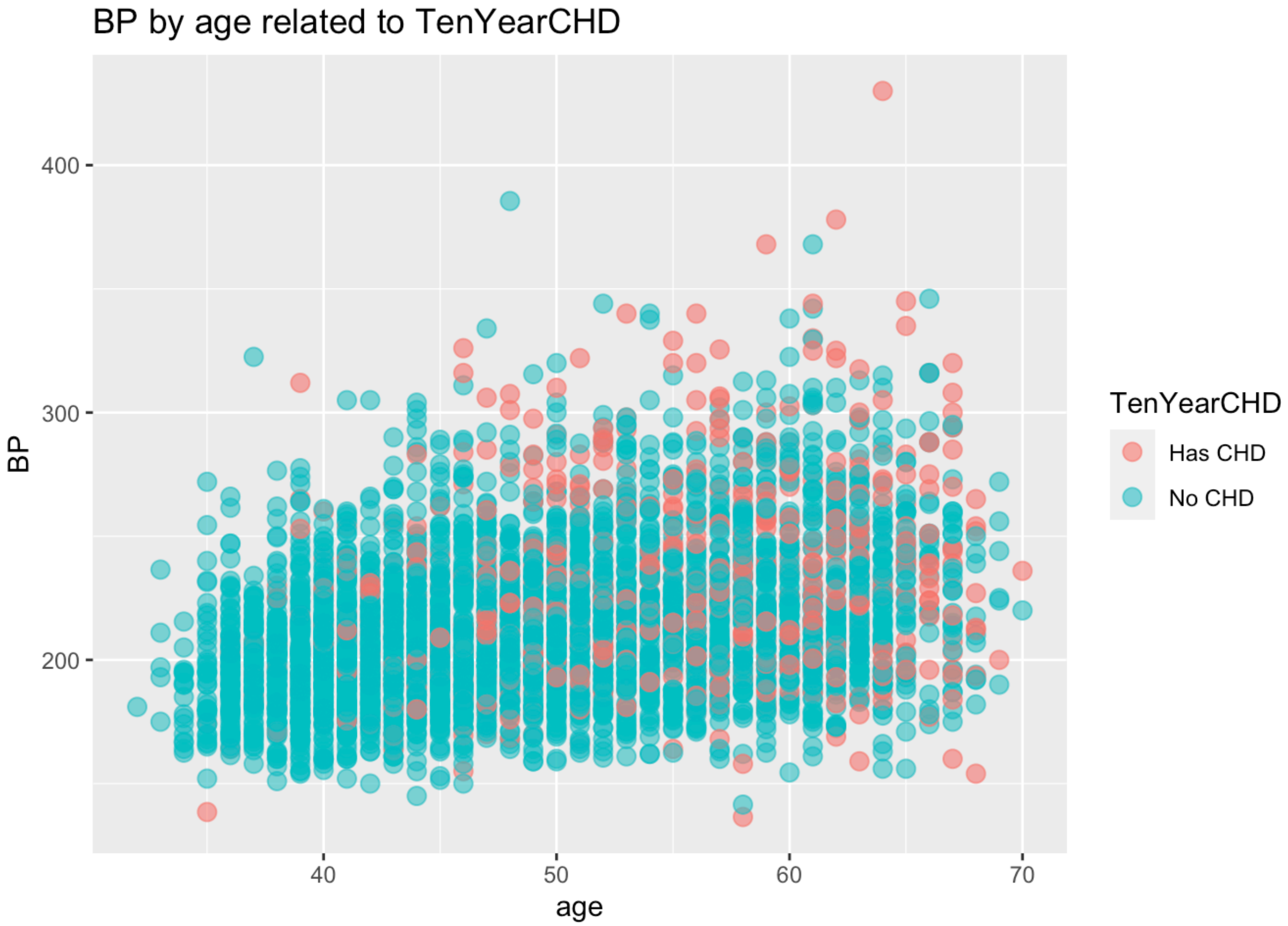
```
## `geom_smooth()` using formula 'y ~ x'
```



## Most people have BMI level between 20-40 and normal hearRate ranges between 50-100 with some outliers. People are more likely to have TenYearCHD if the HeartRate is above 75 but there is not a very strong relationship

14. TenYearCHD relationship with age and BP

```
ggplot(CHD2,
       aes(x = age,
           y = BP,
           color = TenYearCHD)) +
  geom_point(size = 3,
             alpha = .6) +
  labs(title = "BP by age related to TenYearCHD")
```



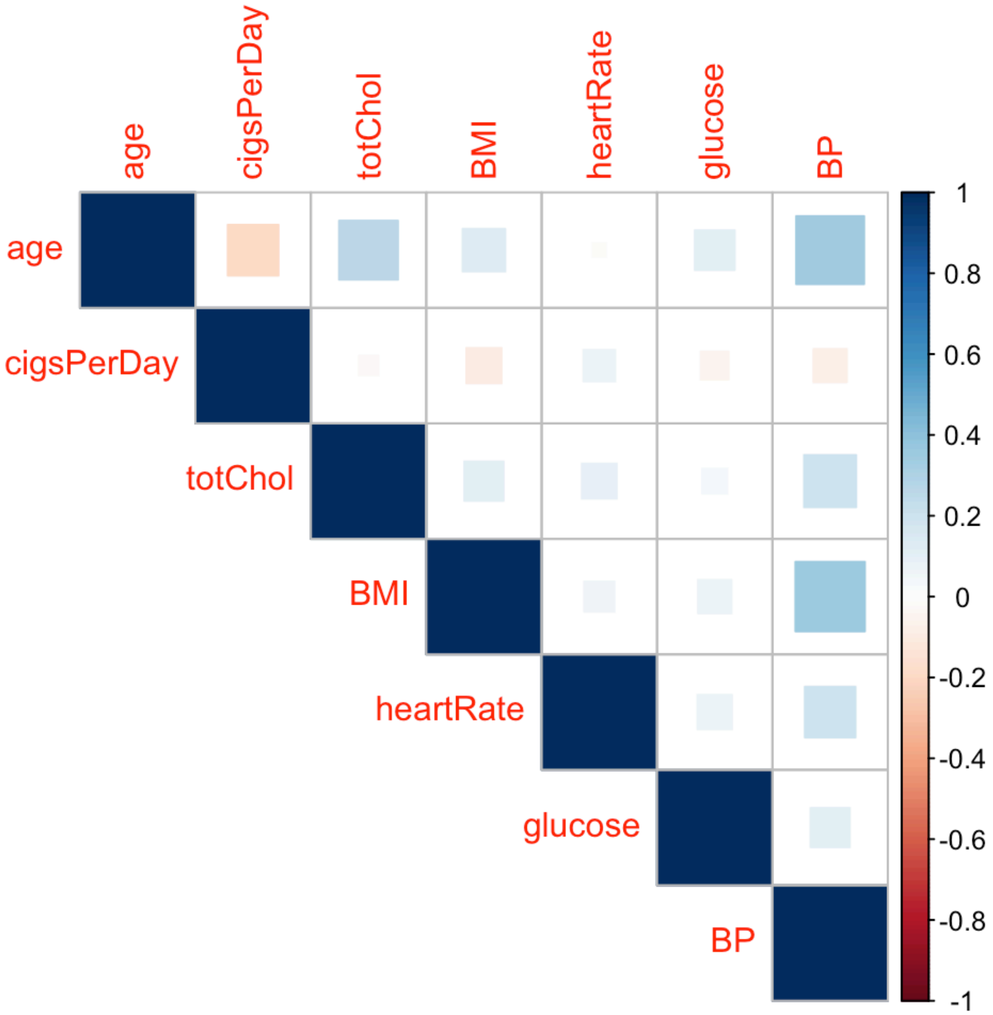
## For older people the BP levels tend to be higher than people below 50 years and so is the probability of TenYearCHD

15. Correlation plot with significant features

```
cor_heart <- cor(CHD2[,c(8,10:15)])
cor_heart
```

```
##                    age  cigsPerDay     totChol         BMI   heartRate
## age         1.00000000 -0.19184667  0.26010450  0.13517428 -0.01284772
## cigsPerDay -0.19184667  1.00000000 -0.02697620 -0.09221079  0.07385272
## totChol     0.26010450 -0.02697620  1.00000000  0.11481074  0.09053715
## BMI         0.13517428 -0.09221079  0.11481074  1.00000000  0.06751977
## heartRate  -0.01284772  0.07385272  0.09053715  0.06751977  1.00000000
## glucose     0.11778820 -0.05686326  0.04555918  0.08221939  0.08731520
## BP          0.34572093 -0.08136402  0.20242932  0.36145811  0.19149633
##                glucose          BP
## age         0.11778820  0.34572093
## cigsPerDay -0.05686326 -0.08136402
## totChol     0.04555918  0.20242932
## BMI         0.08221939  0.36145811
## heartRate   0.08731520  0.19149633
## glucose     1.00000000  0.11420233
## BP          0.11420233  1.00000000
```

```
corrplot(cor_heart,method = 'square',type='upper')
```



##BP seems highly correlated to totChol, age and BMI. cigsPerDay is highly negatively correlated with age which makes sens.

IV. Machine Learning

1. Split Dataset

```
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 3.6.2
```

```
CHD_o<-data.frame(CHD)
CHD <-
  dummy_cols(
    CHD,
    select_columns = c(
      'is_male',
      'education',
      'currentSmoker',
      'BPMeds',
      'prevalentStroke',
      'prevalentHyp',
      'diabetes',
      'TenYearCHD'
    ),
    remove_first_dummy = TRUE,
    remove_selected_columns = TRUE
  )
```

```
names(CHD)[names(CHD)=='TenYearCHD_1'] <- 'TenYearCHD'
names(CHD)
```

```
##  [1] "age"               "cigsPerDay"        "totChol"
##  [4] "sysBP"             "diaBP"             "BMI"
##  [7] "heartRate"         "glucose"           "is_male_1"
## [10] "education_2"       "education_3"       "education_4"
## [13] "currentSmoker_1"   "BPMeds_1"          "prevalentStroke_1"
## [16] "prevalentHyp_1"    "diabetes_1"        "TenYearCHD"
```

```
# we decided to do both under-sampling and oversampling on our imbalanced data using
ROSE(Random Over Sampling Examples) package which deals with imbalanced classes in ca
se of binary classification problems
library(ROSE)
```

```
## Warning: package 'ROSE' was built under R version 3.6.2
```

```
## Loaded ROSE 0.0-4
```

```
##
## Attaching package: 'ROSE'
## The following object is masked from 'package:PRROC':
##
##     roc.curve
```

```
CHD <- ovun.sample(TenYearCHD ~ ., data = CHD, method = "both", p=0.5,N=2000, seed =
1)$data
CHD_o<-ovun.sample(TenYearCHD ~ ., data = CHD_o, method = "both", p=0.5,N=2000, seed
= 1)$data
```

```
set.seed(1)
#train-test split ratio 0.8
id <- createDataPartition(CHD$TenYearCHD, p = 0.8, list = FALSE)
train<-CHD[id, ]
test<-CHD[-id, ]

id_o <- createDataPartition(CHD_o$TenYearCHD, p = 0.8, list = FALSE)
train_o<-CHD_o[id_o, ]
test_o<-CHD_o[-id_o, ]
```

2a. Linear Regression Classification

```
train_y <- train_o$TenYearCHD
test_y <- test_o$TenYearCHD
train_x <- train_o[, -16]
test_x <- test_o[, -16]

linearModel <- lm(train_o$TenYearCHD ~ ., train_o)
result <- data.table(predict(linearModel, test_x))
linear_results <- result[,round(V1)]
linearModel
```

```
##
## Call:
## lm(formula = train_o$TenYearCHD ~ ., data = train_o)
##
## Coefficients:
##     (Intercept)          is_male              age        education
##      -1.0041060        0.0809102        0.0135336        0.0123658
##   currentSmoker       cigsPerDay           BPMeds  prevalentStroke
##       0.0229427        0.0040153        0.0811487        0.2864912
##    prevalentHyp         diabetes          totChol            sysBP
##      -0.0011981        0.1634090        0.0005773        0.0047403
##           diaBP              BMI        heartRate          glucose
##      -0.0023145        0.0043572       -0.0006090        0.0001805
```

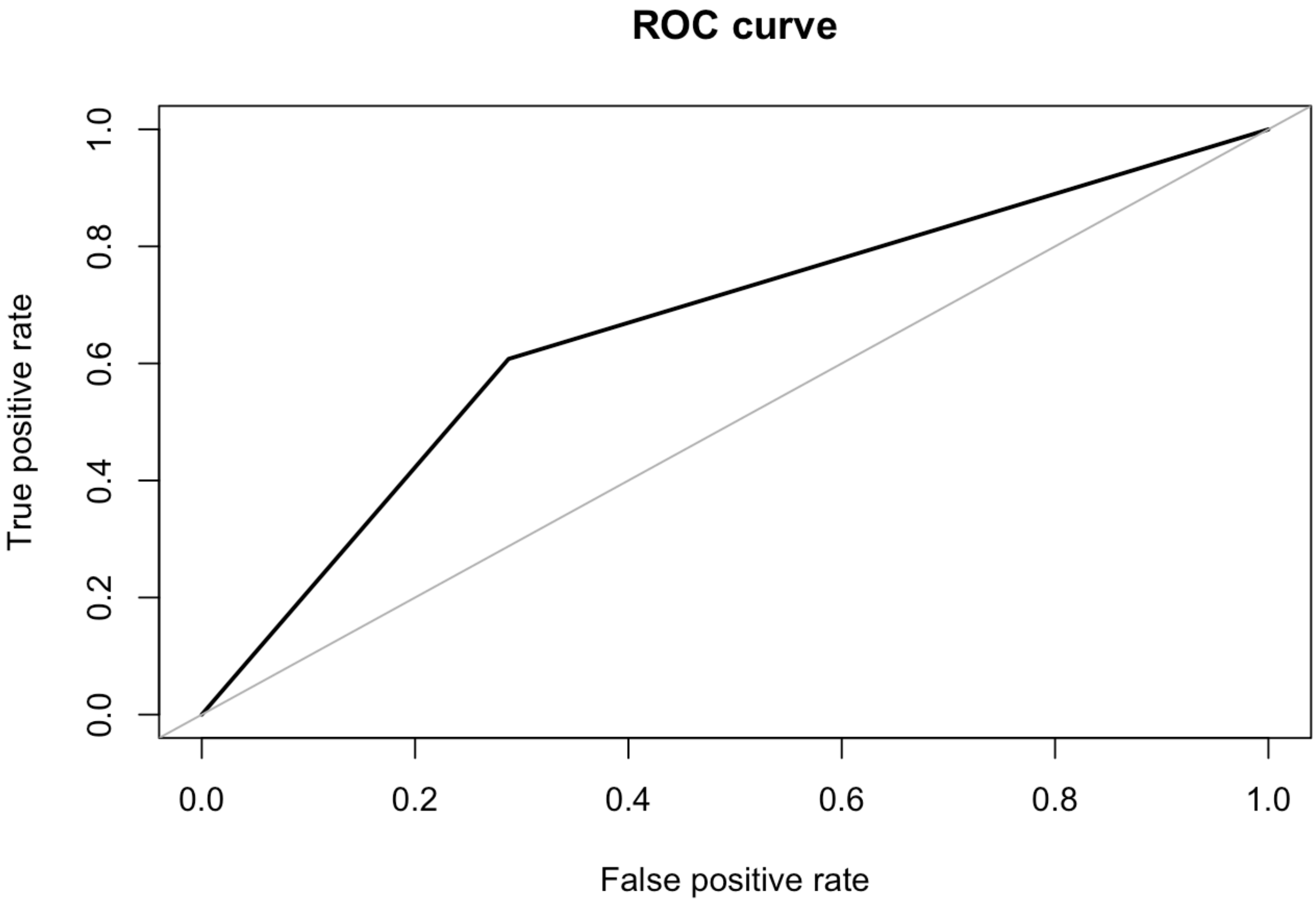
```
accuracy_lm <- linear_results + test_y #0 = True negative, #2 = True positive
accuracy <- 1 - (sum(accuracy_lm == 1)/length(accuracy_lm))

cat("The linear regression model accuracy is", accuracy)
```

```
## The linear regression model accuracy is 0.665
```

```
cmat <- confusionMatrix(as.factor(linear_results), as.factor(test_y), positive = "1")
cmat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 156  71
##          1  63 110
##
##                Accuracy : 0.665
##                  95% CI : (0.6164, 0.7111)
##     No Information Rate : 0.5475
##     P-Value [Acc > NIR] : 1.142e-06
##
##                   Kappa : 0.3213
##
##  Mcnemar's Test P-Value : 0.5454
##
##             Sensitivity : 0.6077
##             Specificity : 0.7123
##          Pos Pred Value : 0.6358
##          Neg Pred Value : 0.6872
##              Prevalence : 0.4525
##          Detection Rate : 0.2750
##    Detection Prevalence : 0.4325
##       Balanced Accuracy : 0.6600
##
##        'Positive' Class : 1
##
```

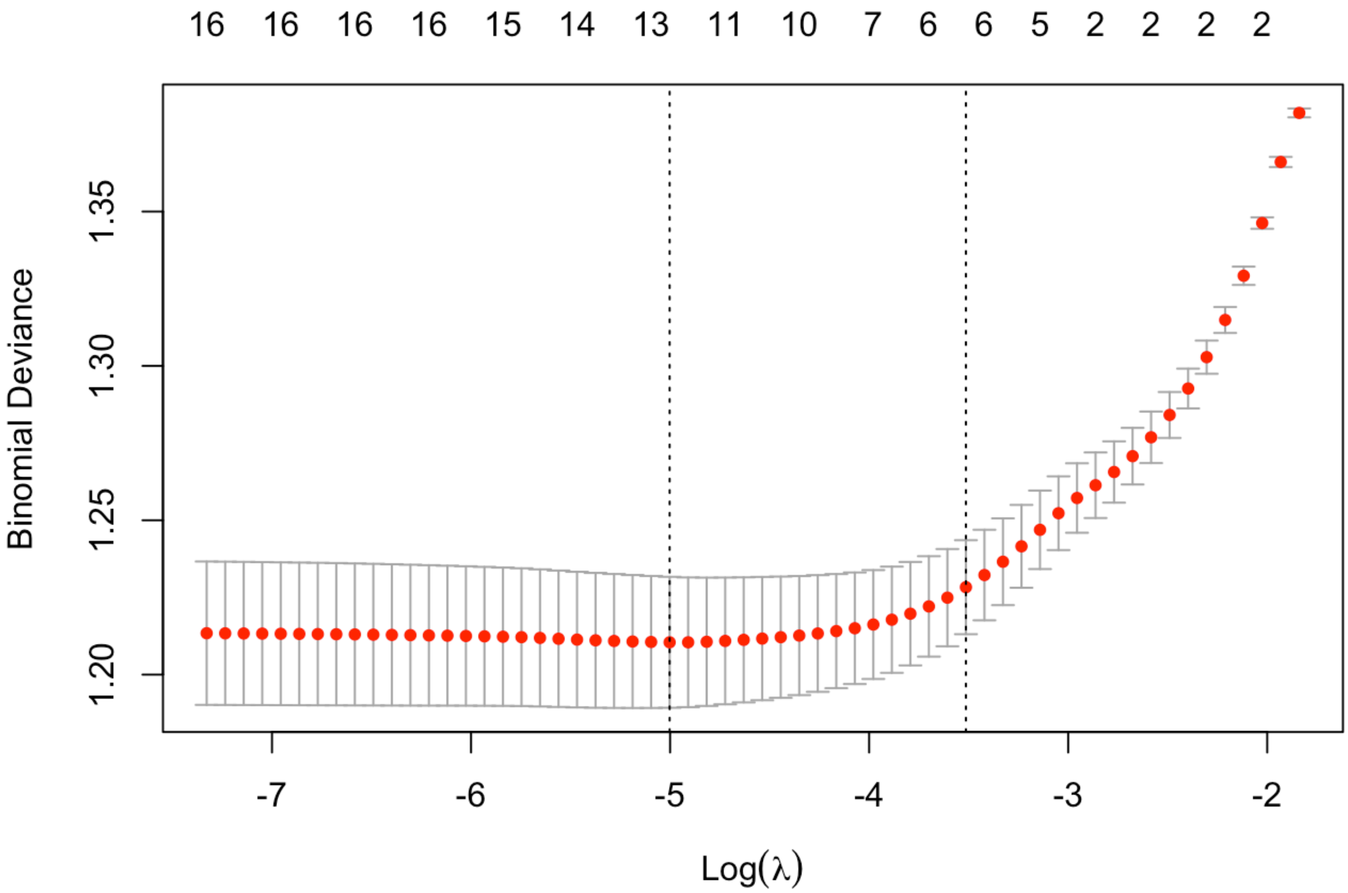
```
#F1 score
roc.curve(as.numeric(test_y),as.numeric(linear_results))
```



```
## Area under the curve (AUC): 0.660
```

2b. Linear Regression with Lasso Classification

```
# Create formula
formula <- as.formula(TenYearCHD ~ .)

# Training set modeling
train.matrix <- model.matrix(formula, train)[, -1]
train_y <- train$TenYearCHD
fit <- cv.glmnet(train.matrix, train_y, family = "binomial", alpha = 1, nfolds = 10)

# plot
plot(fit)
```



```
# Create testing matrices
test.matrix <- model.matrix(formula, test) [, -1]
```

```
coef(fit, s=fit$lambda.min)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)       -6.1894203984
## age                0.0595707759
## cigsPerDay         0.0155286282
## totChol            0.0006113751
## sysBP              0.0174537939
## diaBP              .
## BMI                .
## heartRate          .
## glucose            0.0014680481
## is_male_1          0.3593981467
## education_2       -0.0684953323
## education_3        0.1398760993
## education_4        .
## currentSmoker_1    0.0142719129
## BPMeds_1           0.6010763469
## prevalentStroke_1  0.3090095083
## prevalentHyp_1     .
## diabetes_1         0.8941309306
```

```
# Predicting test data

test.predictions <- predict(fit, test.matrix, s = fit$lambda.min, type = "response")

##F1 score, select cutoff which makes the F1 score largest
Fmeasure <- c()
cutoffs <- seq(0.05, 0.85, 0.01)
for(cutoff in cutoffs) {
  predicted.CHD <- ifelse(test.predictions > cutoff, 1, 0)
  cmat <-
    confusionMatrix(as.factor(predicted.CHD),
                    as.factor(test$TenYearCHD),
                    positive = "1")
  Fmeasure <- c(Fmeasure,  cmat$byClass[7])
}
```

```
## Warning in confusionMatrix.default(as.factor(predicted.CHD),
## as.factor(test$TenYearCHD), : Levels are not in the same order for
## reference and data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(predicted.CHD),
## as.factor(test$TenYearCHD), : Levels are not in the same order for
## reference and data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(predicted.CHD),
## as.factor(test$TenYearCHD), : Levels are not in the same order for
## reference and data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(predicted.CHD),
## as.factor(test$TenYearCHD), : Levels are not in the same order for
## reference and data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(predicted.CHD),
## as.factor(test$TenYearCHD), : Levels are not in the same order for
## reference and data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(predicted.CHD),
## as.factor(test$TenYearCHD), : Levels are not in the same order for
## reference and data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(predicted.CHD),
## as.factor(test$TenYearCHD), : Levels are not in the same order for
## reference and data. Refactoring data to match.
```

```
cutoffs[which.max(Fmeasure)]
```

```
## [1] 0.34
```

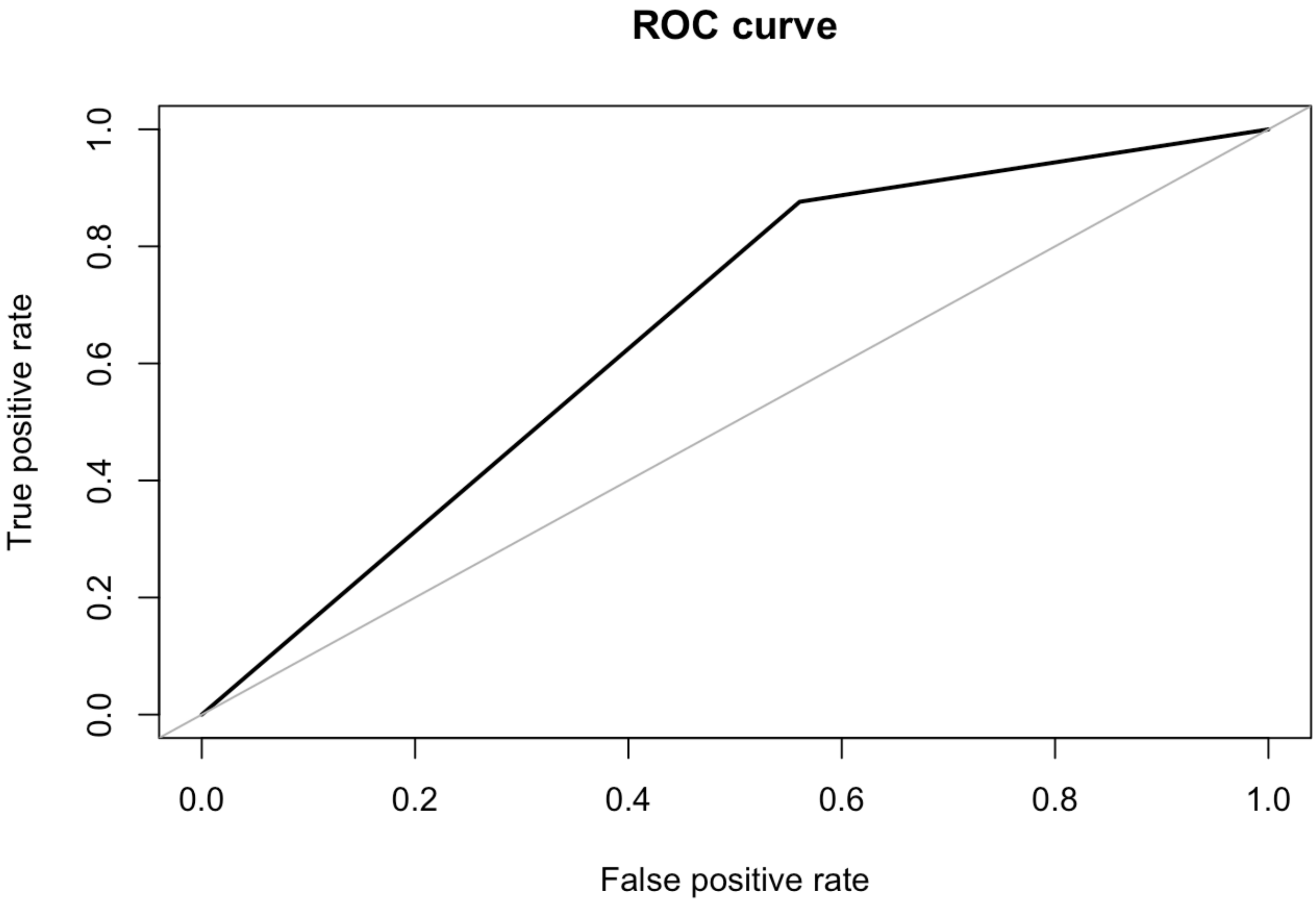
```
#0.15

predicted.CHD <- ifelse(test.predictions > cutoffs[which.max(Fmeasure)], 1, 0)
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  87  25
##          1 111 177
## 
##                Accuracy : 0.66            
##                  95% CI : (0.6113, 0.7063)
##     No Information Rate : 0.505           
##     P-Value [Acc > NIR] : 2.723e-10       
## 
##                   Kappa : 0.317           
## 
##  Mcnemar's Test P-Value : 3.130e-13       
## 
##             Sensitivity : 0.8762          
##             Specificity : 0.4394          
##          Pos Pred Value : 0.6146          
##          Neg Pred Value : 0.7768          
##              Prevalence : 0.5050          
##          Detection Rate : 0.4425          
##    Detection Prevalence : 0.7200          
##       Balanced Accuracy : 0.6578          
## 
##        'Positive' Class : 1               
## 
```

```
#F1 score
cmat$byClass[7]
```

```
##       F1 
## 0.722449
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```



```
## Area under the curve (AUC): 0.658
```

3. Linear Regression with Ridge Classification

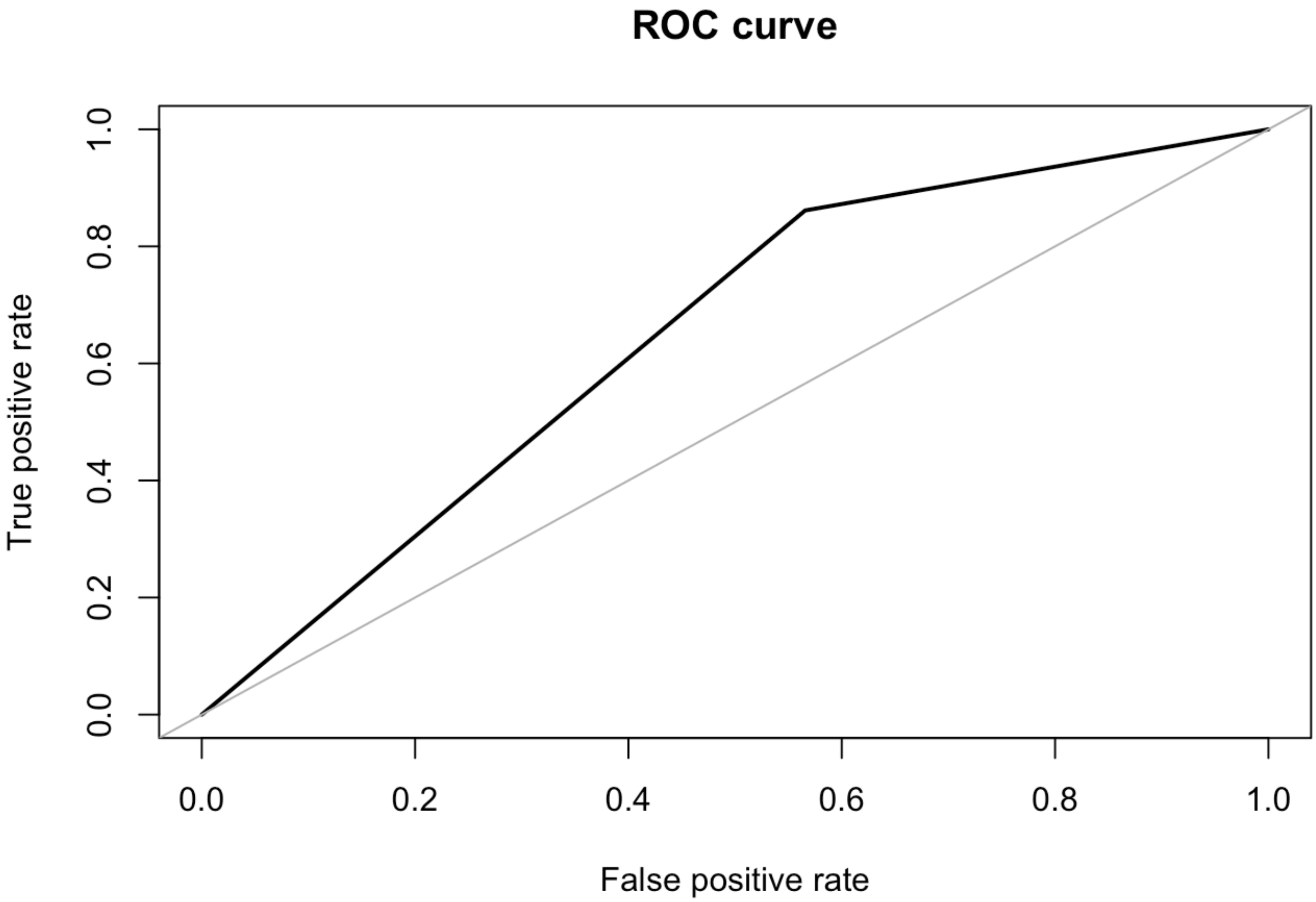
```
# Create formula
formula <- as.formula(TenYearCHD ~ .)

# Training set modeling
train.matrix <- model.matrix(formula, train)[, -1]
train_y <- train$TenYearCHD
fit <- cv.glmnet(train.matrix, train_y, family = "binomial", alpha = 0, nfolds = 10)
coef(fit, s=fit$lambda.min)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                             s1
## (Intercept)       -6.5420398727
## age                0.0573290662
## cigsPerDay         0.0139625849
## totChol            0.0013269638
## sysBP              0.0171451405
## diaBP              0.0005396473
## BMI                0.0082805487
## heartRate         -0.0007131508
## glucose            0.0029178871
## is_male_1          0.3771405780
## education_2       -0.1301164786
## education_3        0.1903054311
## education_4       -0.0315162402
## currentSmoker_1    0.1044638271
## BPMeds_1           0.7325007815
## prevalentStroke_1  0.6836984833
## prevalentHyp_1    -0.0664713886
## diabetes_1         0.8504983562
```

```
test.matrix <- model.matrix(formula, test) [, -1]
test.predictions <- predict(fit, test.matrix, s = fit$lambda.min, type = "response")
predicted.CHD <- ifelse(test.predictions > cutoffs[which.max(Fmeasure)], 1, 0)
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  86  28
##          1 112 174
##                                          
##                Accuracy : 0.65           
##                  95% CI : (0.601, 0.6967)
##     No Information Rate : 0.505          
##     P-Value [Acc > NIR] : 3.406e-09      
##                                          
##                   Kappa : 0.297          
##                                          
##  Mcnemar's Test P-Value : 2.303e-12      
##                                          
##             Sensitivity : 0.8614         
##             Specificity : 0.4343         
##          Pos Pred Value : 0.6084         
##          Neg Pred Value : 0.7544         
##              Prevalence : 0.5050         
##          Detection Rate : 0.4350         
##    Detection Prevalence : 0.7150         
##       Balanced Accuracy : 0.6479         
##                                          
##        'Positive' Class : 1              
## 
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```



```
## Area under the curve (AUC): 0.648
```

4. Logistic Classification

```
#use variables selected by lasso
coefs <- coef(fit,s=fit$lambda.min)
variables <- which(coefs !=0)

selectvariables <- names(coefs[variables,])[-1]
selectvariables
```

```
##  [1] "age"              "cigsPerDay"       "totChol"
##  [4] "sysBP"            "diaBP"            "BMI"
##  [7] "heartRate"        "glucose"          "is_male_1"
## [10] "education_2"      "education_3"      "education_4"
## [13] "currentSmoker_1"  "BPMeds_1"         "prevalentStroke_1"
## [16] "prevalentHyp_1"   "diabetes_1"
```

```
train2 <- train.matrix[,selectvariables]
test2 <- test.matrix[,selectvariables]

newtrain <- data.frame(train2, TenYearCHD = train$TenYearCHD)
newtest <- data.frame(test2, TenYearCHD = test$TenYearCHD)

fit2 <- glm(TenYearCHD ~ ., data = newtrain, family = binomial(link = "logit"))
summary(fit2)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ ., family = binomial(link = "logit"),
##     data = newtrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4799  -0.9632  -0.5564   1.0335   2.2817
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.1131157  0.8069368  -8.815  < 2e-16 ***
## age                0.0621682  0.0078699   7.900 2.80e-15 ***
## cigsPerDay         0.0164485  0.0071774   2.292 0.021922 *
## totChol            0.0012394  0.0012468   0.994 0.320190
## sysBP              0.0244001  0.0049093   4.970 6.69e-07 ***
## diaBP             -0.0068231  0.0077387  -0.882 0.377942
## BMI                0.0100027  0.0151039   0.662 0.507806
## heartRate         -0.0009857  0.0046374  -0.213 0.831670
## glucose            0.0023498  0.0030201   0.778 0.436543
## is_male_1          0.4215720  0.1224082   3.444 0.000573 ***
## education_2       -0.1098346  0.1370945  -0.801 0.423039
## education_3        0.2485368  0.1667643   1.490 0.136133
## education_4        0.0022558  0.1907696   0.012 0.990565
## currentSmoker_1    0.0842569  0.1802165   0.468 0.640120
## BPMeds_1           0.8388460  0.3243689   2.586 0.009707 **
## prevalentStroke_1  0.7974914  0.7409248   1.076 0.281773
## prevalentHyp_1    -0.2191342  0.1640958  -1.335 0.181744
## diabetes_1         1.0143197  0.4785873   2.119 0.034056 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2213.9  on 1599  degrees of freedom
## Residual deviance: 1903.4  on 1582  degrees of freedom
## AIC: 1939.4
##
## Number of Fisher Scoring iterations: 4
```

```
# Predicting test data

test.predictions <- predict(fit2, newtest,  type = "response")

predicted.CHD <- ifelse(test.predictions > cutoffs[which.max(Fmeasure)], 1, 0)
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  87  32
##          1 111 170
## 
##                Accuracy : 0.6425
##                  95% CI : (0.5934, 0.6895)
##     No Information Rate : 0.505
##     P-Value [Acc > NIR] : 2.024e-08
## 
##                   Kappa : 0.2821
## 
##  Mcnemar's Test P-Value : 6.906e-11
## 
##             Sensitivity : 0.8416
##             Specificity : 0.4394
##          Pos Pred Value : 0.6050
##          Neg Pred Value : 0.7311
##              Prevalence : 0.5050
##          Detection Rate : 0.4250
##    Detection Prevalence : 0.7025
##       Balanced Accuracy : 0.6405
## 
##        'Positive' Class : 1
## 
```

```
#F1 score
cmat$byClass[7]
```

```
##        F1
## 0.7039337
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```

```
## Area under the curve (AUC): 0.640
```

```
#use full data
fit3 <- glm(TenYearCHD ~ ., data = train, family = binomial(link = "logit"))
summary(fit3)
```

```
## 
## Call:
## glm(formula = TenYearCHD ~ ., family = binomial(link = "logit"), 
##     data = train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4799  -0.9632  -0.5564   1.0335   2.2817  
## 
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       -7.1131157  0.8069368  -8.815  < 2e-16 ***
## age                0.0621682  0.0078699   7.900 2.80e-15 ***
## cigsPerDay         0.0164485  0.0071774   2.292 0.021922 *  
## totChol            0.0012394  0.0012468   0.994 0.320190    
## sysBP              0.0244001  0.0049093   4.970 6.69e-07 ***
## diaBP             -0.0068231  0.0077387  -0.882 0.377942    
## BMI                0.0100027  0.0151039   0.662 0.507806    
## heartRate         -0.0009857  0.0046374  -0.213 0.831670    
## glucose            0.0023498  0.0030201   0.778 0.436543    
## is_male_1          0.4215720  0.1224082   3.444 0.000573 ***
## education_2       -0.1098346  0.1370945  -0.801 0.423039    
## education_3        0.2485368  0.1667643   1.490 0.136133    
## education_4        0.0022558  0.1907696   0.012 0.990565    
## currentSmoker_1    0.0842569  0.1802165   0.468 0.640120    
## BPMeds_1           0.8388460  0.3243689   2.586 0.009707 ** 
## prevalentStroke_1  0.7974914  0.7409248   1.076 0.281773    
## prevalentHyp_1    -0.2191342  0.1640958  -1.335 0.181744    
## diabetes_1         1.0143197  0.4785873   2.119 0.034056 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2213.9  on 1599  degrees of freedom
## Residual deviance: 1903.4  on 1582  degrees of freedom
## AIC: 1939.4
## 
## Number of Fisher Scoring iterations: 4
```

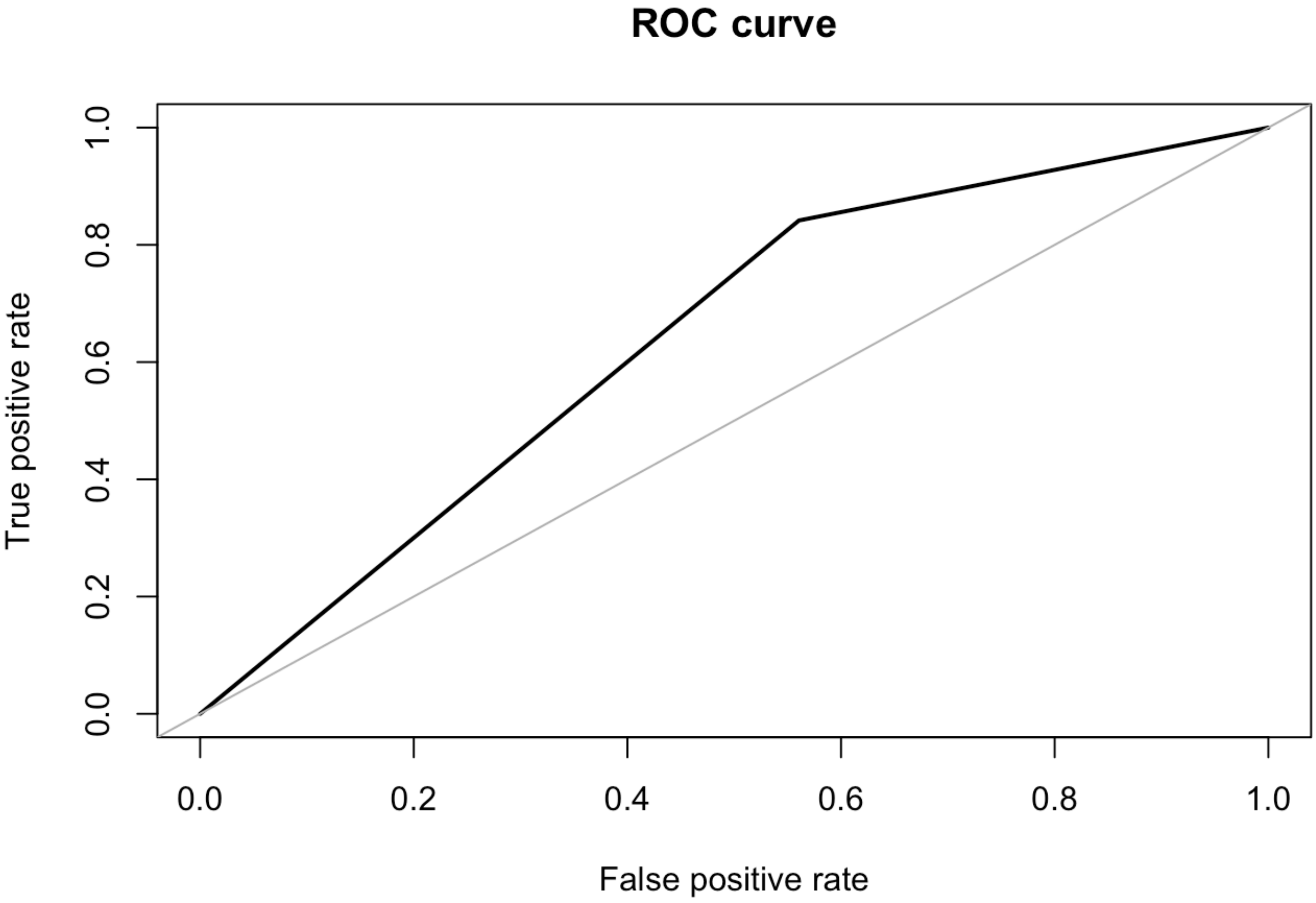
```
#Predicting test data

test.predictions <- predict(fit3, test,  type = "response")


predicted.CHD <- ifelse(test.predictions > cutoffs[which.max(Fmeasure)], 1, 0)
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  87  32
##          1 111 170
##                                           
##                Accuracy : 0.6425          
##                  95% CI : (0.5934, 0.6895)
##     No Information Rate : 0.505           
##     P-Value [Acc > NIR] : 2.024e-08       
##                                           
##                   Kappa : 0.2821          
##                                           
##  Mcnemar's Test P-Value : 6.906e-11       
##                                           
##             Sensitivity : 0.8416          
##             Specificity : 0.4394          
##          Pos Pred Value : 0.6050          
##          Neg Pred Value : 0.7311          
##              Prevalence : 0.5050          
##          Detection Rate : 0.4250          
##    Detection Prevalence : 0.7025          
##       Balanced Accuracy : 0.6405          
##                                           
##        'Positive' Class : 1               
## 
```

```
#F1 score
cmat$byClass[7]
```

```
##        F1 
## 0.7039337
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```



```
## Area under the curve (AUC): 0.640
```

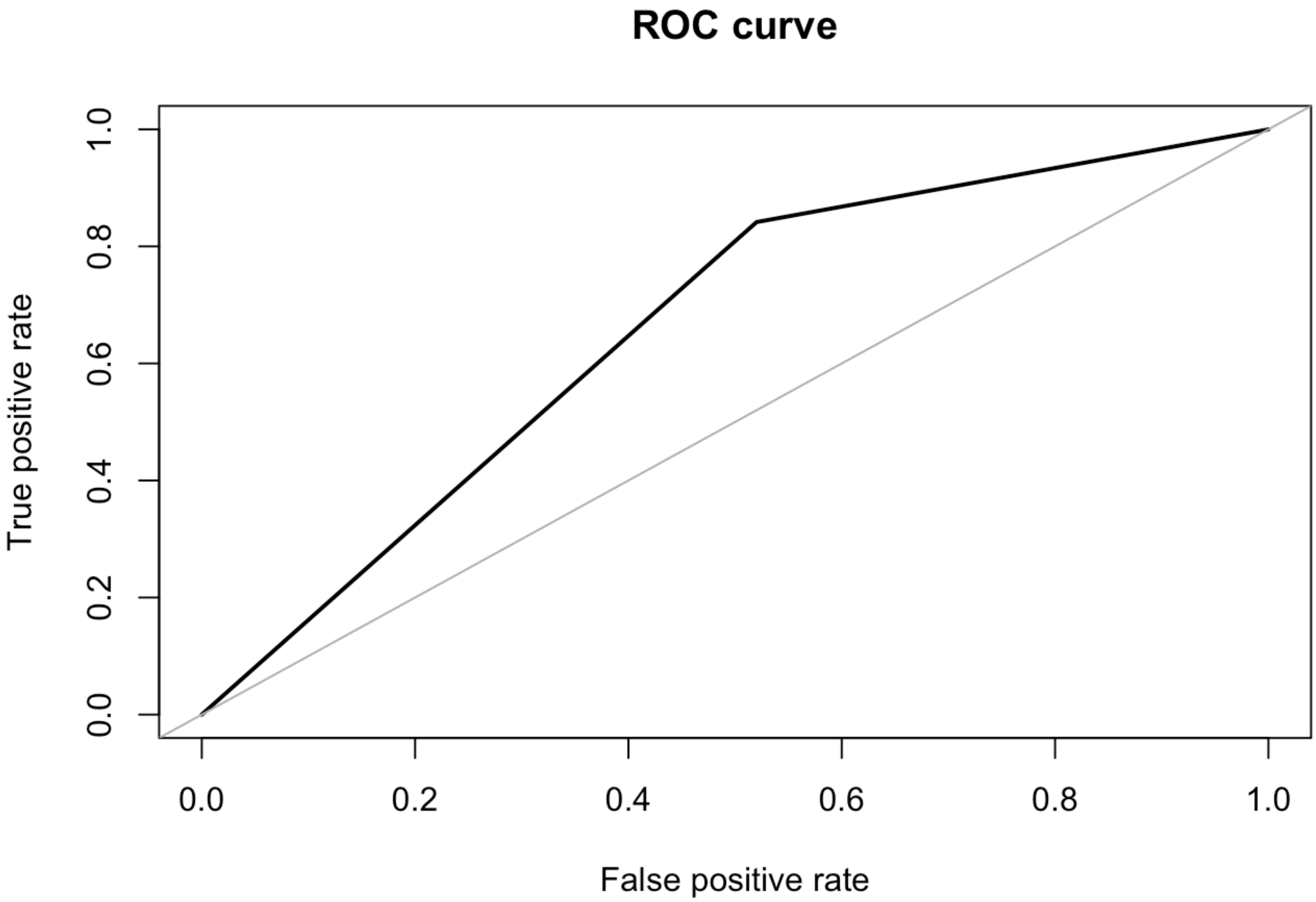
```
#use backward selection with AIC criterion

fit4 <- step(fit3,trace = F)
summary(fit4)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ age + cigsPerDay + sysBP + is_male_1 +
##     education_3 + BPMeds_1 + diabetes_1, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4130  -0.9697  -0.5619   1.0394   2.1887
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.513268   0.460445 -14.146  < 2e-16 ***
## age          0.065276   0.007388   8.836  < 2e-16 ***
## cigsPerDay   0.018742   0.004745   3.950 7.81e-05 ***
## sysBP        0.018827   0.002633   7.151 8.61e-13 ***
## is_male_1    0.426315   0.117524   3.627 0.000286 ***
## education_3  0.278821   0.152794   1.825 0.068029 .
## BPMeds_1     0.814684   0.316228   2.576 0.009988 **
## diabetes_1   1.311779   0.346545   3.785 0.000154 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2213.9  on 1599  degrees of freedom
## Residual deviance: 1909.9  on 1592  degrees of freedom
## AIC: 1925.9
##
## Number of Fisher Scoring iterations: 4
```

```
#Predicting test data

test.predictions <- predict(fit4, test, type = "response")

predicted.CHD <- ifelse(test.predictions > cutoffs[which.max(Fmeasure)], 1, 0)
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  95  32
##          1 103 170
##                                           
##                Accuracy : 0.6625          
##                  95% CI : (0.6138, 0.7087)
##     No Information Rate : 0.505           
##     P-Value [Acc > NIR] : 1.409e-10       
##                                           
##                   Kappa : 0.3225          
##                                           
##  Mcnemar's Test P-Value : 1.695e-09       
##                                           
##             Sensitivity : 0.8416          
##             Specificity : 0.4798          
##          Pos Pred Value : 0.6227          
##          Neg Pred Value : 0.7480          
##              Prevalence : 0.5050          
##          Detection Rate : 0.4250          
##    Detection Prevalence : 0.6825          
##       Balanced Accuracy : 0.6607          
##                                           
##        'Positive' Class : 1               
## 
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```



```
## Area under the curve (AUC): 0.661
```

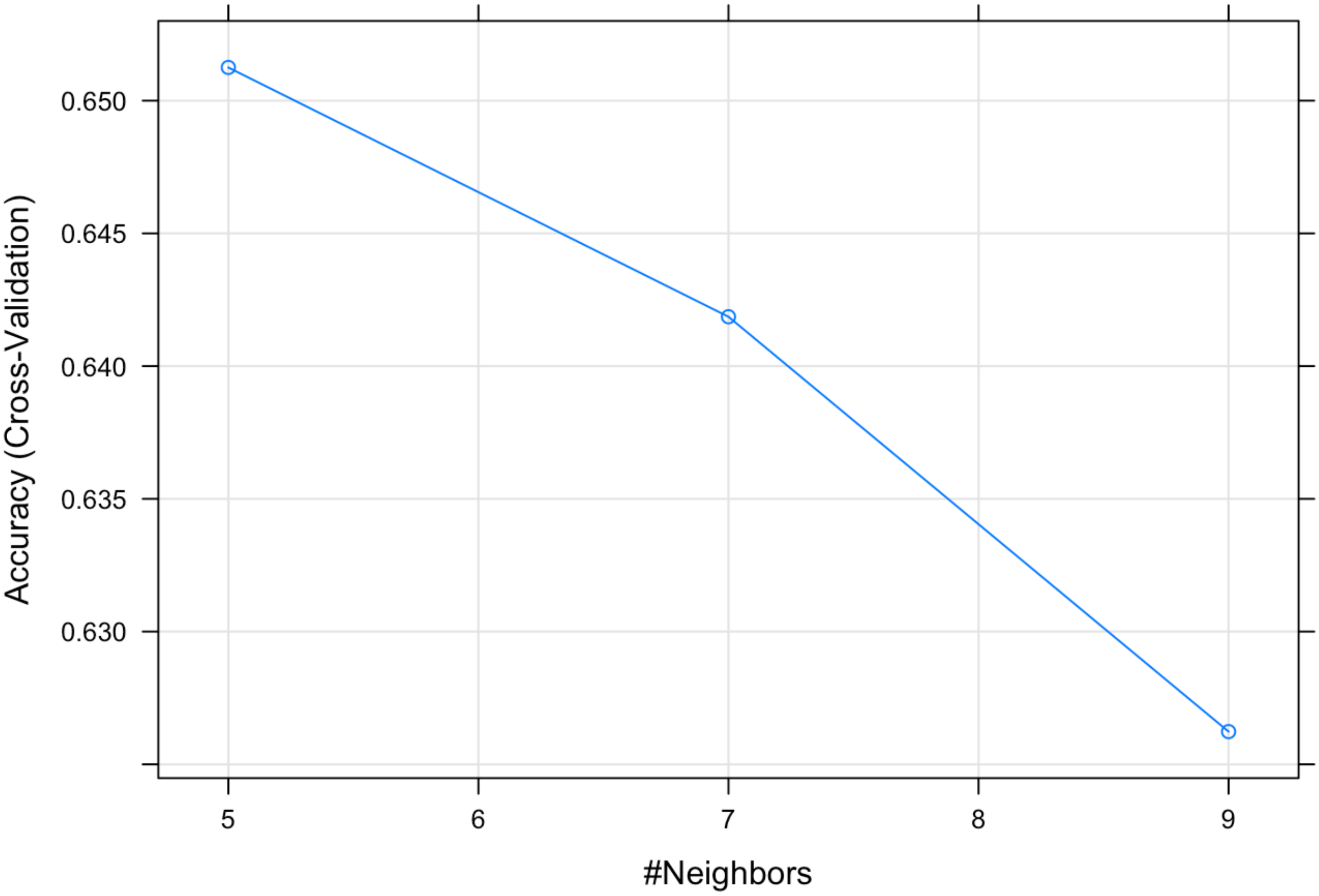
```
#all logistic models are similar
```

5. KNN - K-Nearest Neighbors

```
set.seed(1)
#set 10-folds cross validation
ctrl <- trainControl(method = "cv",
                     number = 10)
#KNN for k-nearest neighbors

#check parameters tuning results
m <- train(factor(TenYearCHD) ~ ., data = train,
           method = "knn",
           trControl = ctrl)

m
```

```
## k-Nearest Neighbors
##
## 1600 samples
##   17 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1440, 1440, 1440, 1439, 1441, 1440, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.6512512  0.3034580
##   7  0.6418604  0.2826862
##   9  0.6262314  0.2510460
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

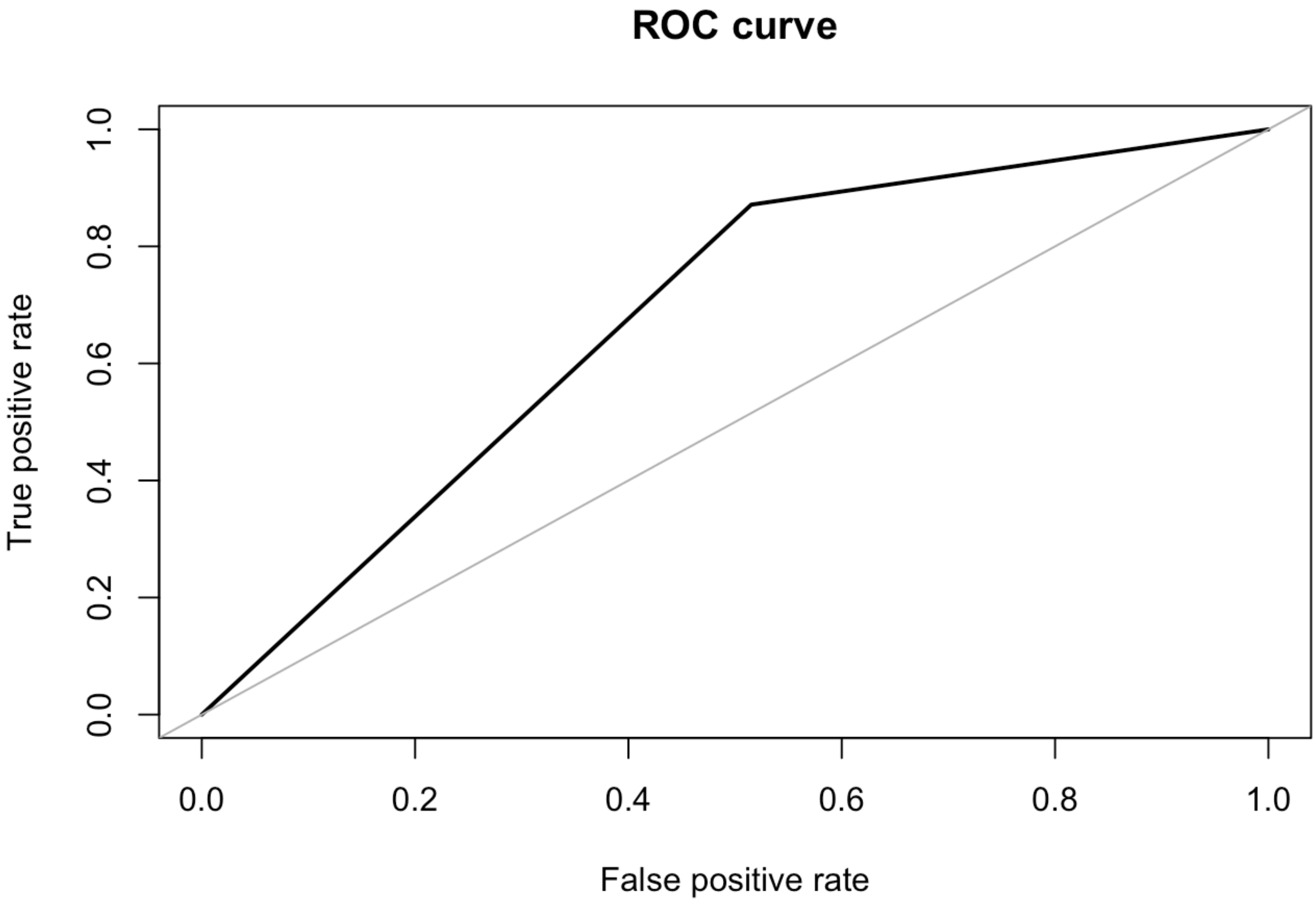
```
plot(m)
```



```
test.predictions <- predict(m, test,  type = "prob")[,2]

predicted.CHD <- ifelse(test.predictions > cutoffs[which.max(Fmeasure)], 1, 0)
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  96  26
##          1 102 176
## 
##                Accuracy : 0.68
##                  95% CI : (0.6318, 0.7255)
##     No Information Rate : 0.505
##     P-Value [Acc > NIR] : 1.029e-12
## 
##                   Kappa : 0.3575
## 
##  Mcnemar's Test P-Value : 3.377e-11
## 
##             Sensitivity : 0.8713
##             Specificity : 0.4848
##          Pos Pred Value : 0.6331
##          Neg Pred Value : 0.7869
##              Prevalence : 0.5050
##          Detection Rate : 0.4400
##    Detection Prevalence : 0.6950
##       Balanced Accuracy : 0.6781
## 
##        'Positive' Class : 1
## 
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```



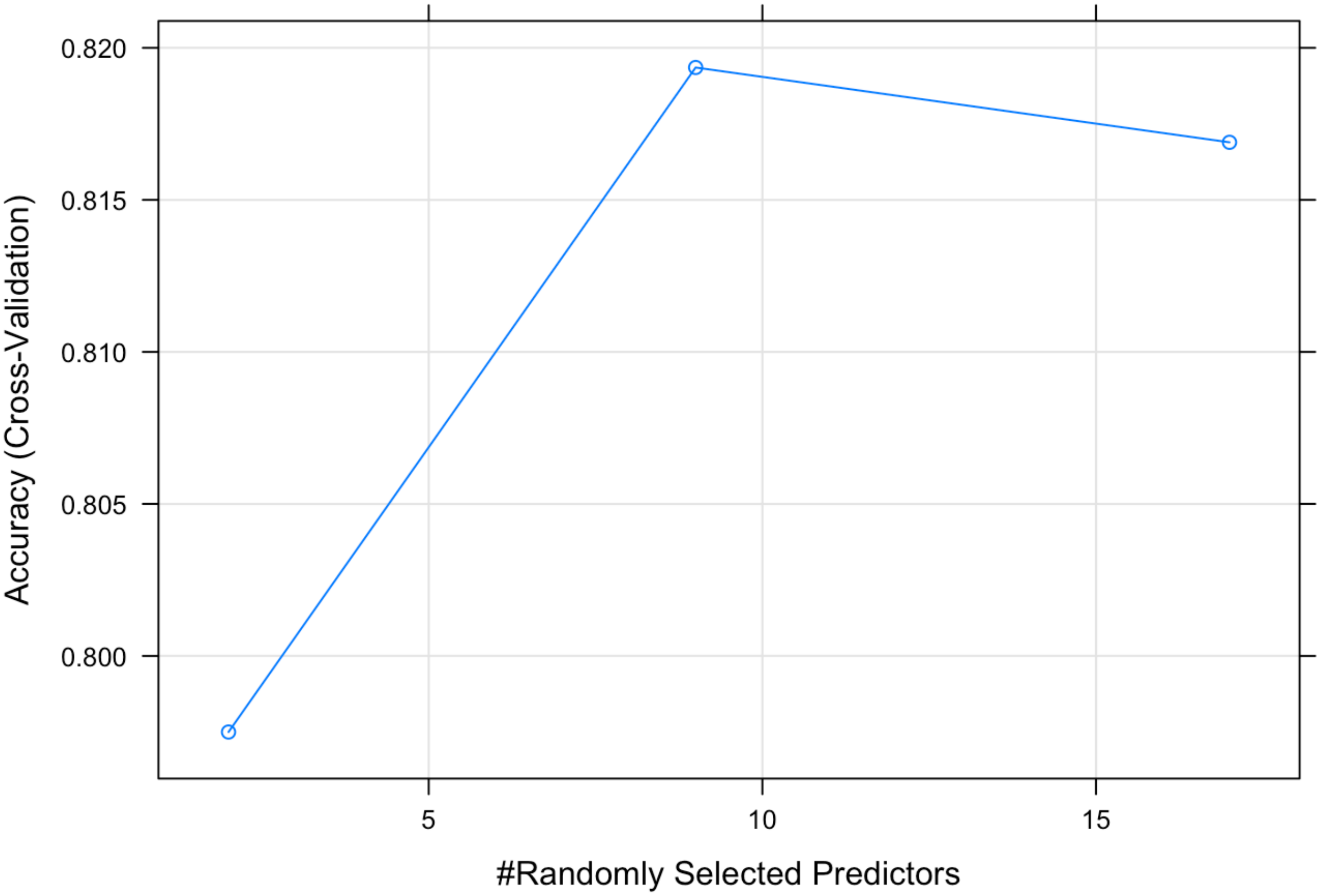
```
## Area under the curve (AUC): 0.678
```

6. Random Forest

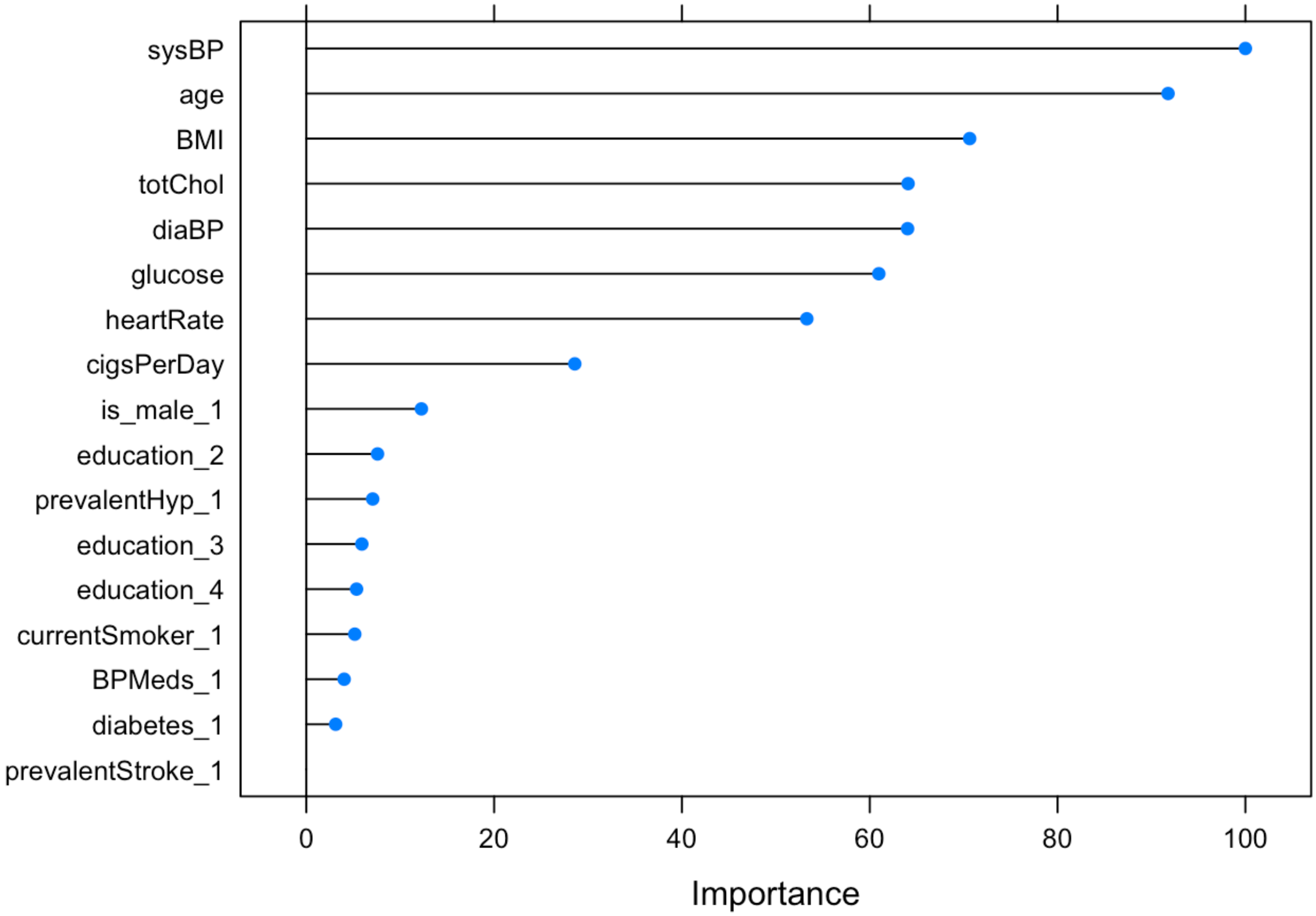
```
set.seed(1)
#set  10-folds cross validation
ctrl <- trainControl(method = "cv",
                     number = 10)
#rf for random forest

#check parameters tuning results
m <- train(factor(TenYearCHD) ~ ., data = train,
           method = "rf",
           trControl = ctrl)
m
```

```
## Random Forest
##
## 1600 samples
##   17 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1440, 1440, 1440, 1439, 1441, 1440, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.7974983  0.5936187
##    9    0.8193539  0.6382111
##   17    0.8168930  0.6332630
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 9.
```

```
plot(m)
```



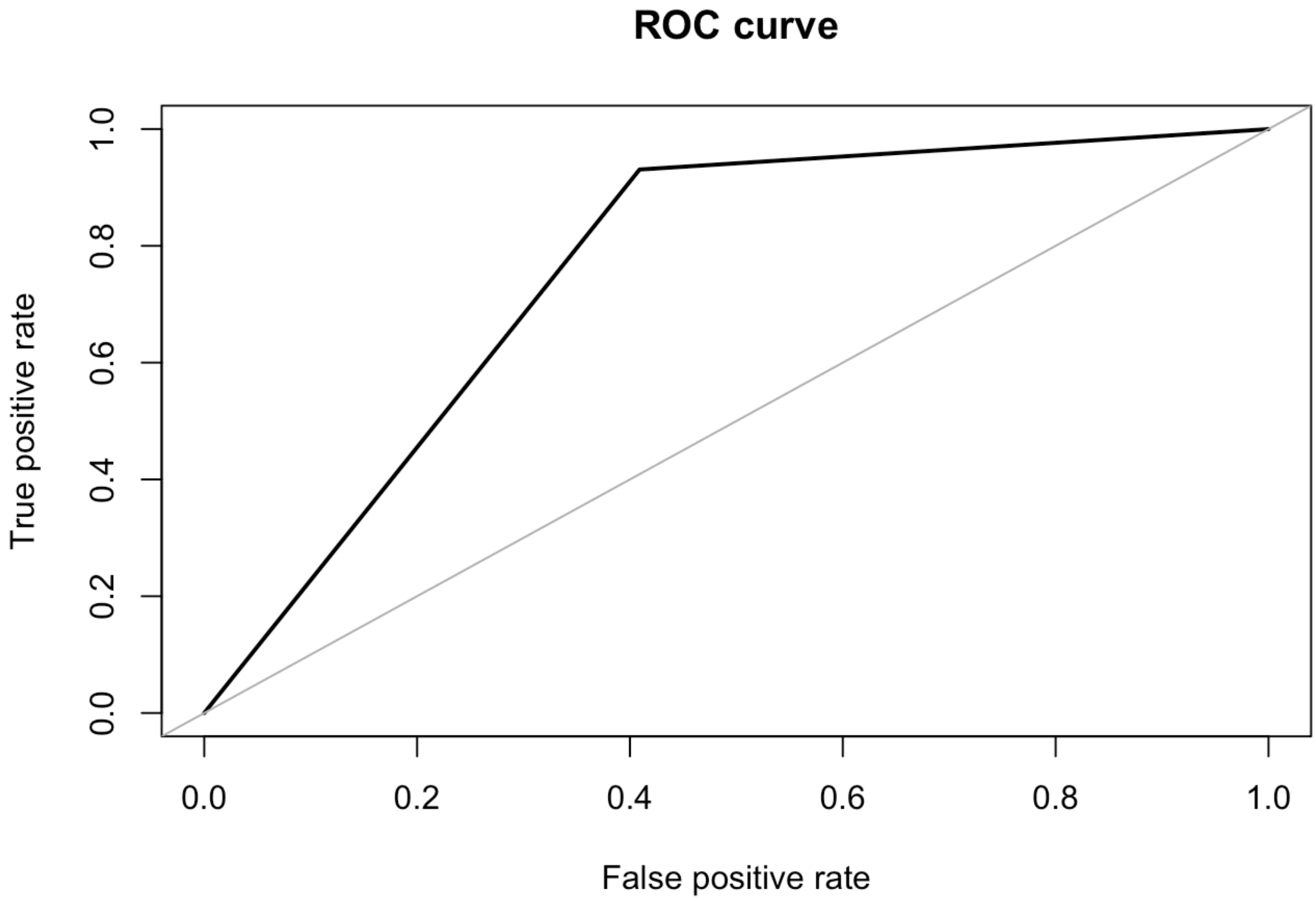
```
#variable important plot
plot(varImp(m))
```



```
test.predictions <- predict(m, test,  type = "prob")[,2]

predicted.CHD <- ifelse(test.predictions > cutoffs[which.max(Fmeasure)], 1, 0)
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 117  14
##          1  81 188
##                                           
##                Accuracy : 0.7625          
##                  95% CI : (0.7177, 0.8034)
##     No Information Rate : 0.505           
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.5234          
##                                           
##  Mcnemar's Test P-Value : 1.275e-11       
##                                           
##             Sensitivity : 0.9307          
##             Specificity : 0.5909          
##          Pos Pred Value : 0.6989          
##          Neg Pred Value : 0.8931          
##              Prevalence : 0.5050          
##          Detection Rate : 0.4700          
##    Detection Prevalence : 0.6725          
##       Balanced Accuracy : 0.7608          
##                                           
##        'Positive' Class : 1               
## 
```

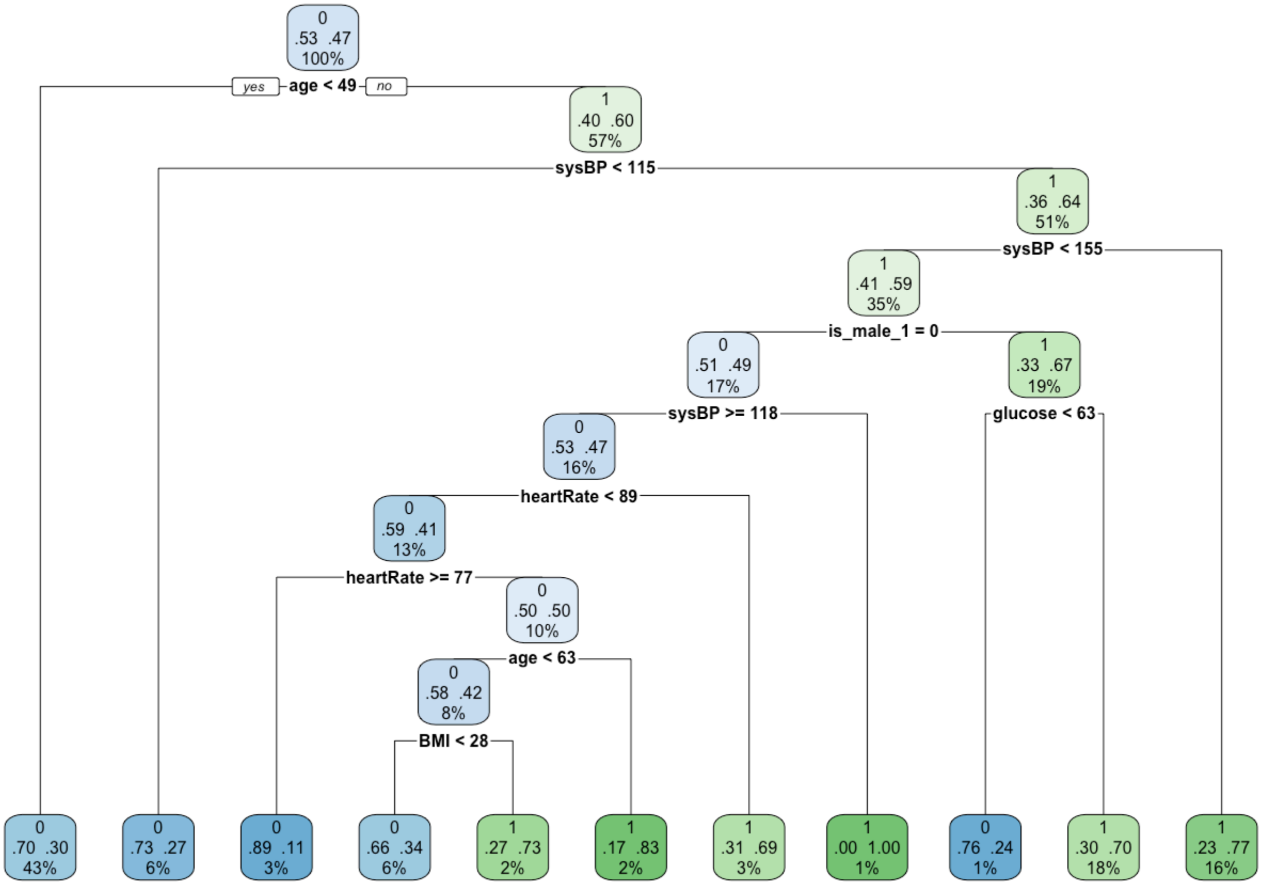
```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```



```
## Area under the curve (AUC): 0.761
```

7. Decision Tree

```
library(rpart)
library(rpart.plot)

train$TenYearCHD <- as.factor(train$TenYearCHD)

fit5 <- rpart(TenYearCHD~., data = train, method = 'class')
rpart.plot(fit5, extra = 104)
```

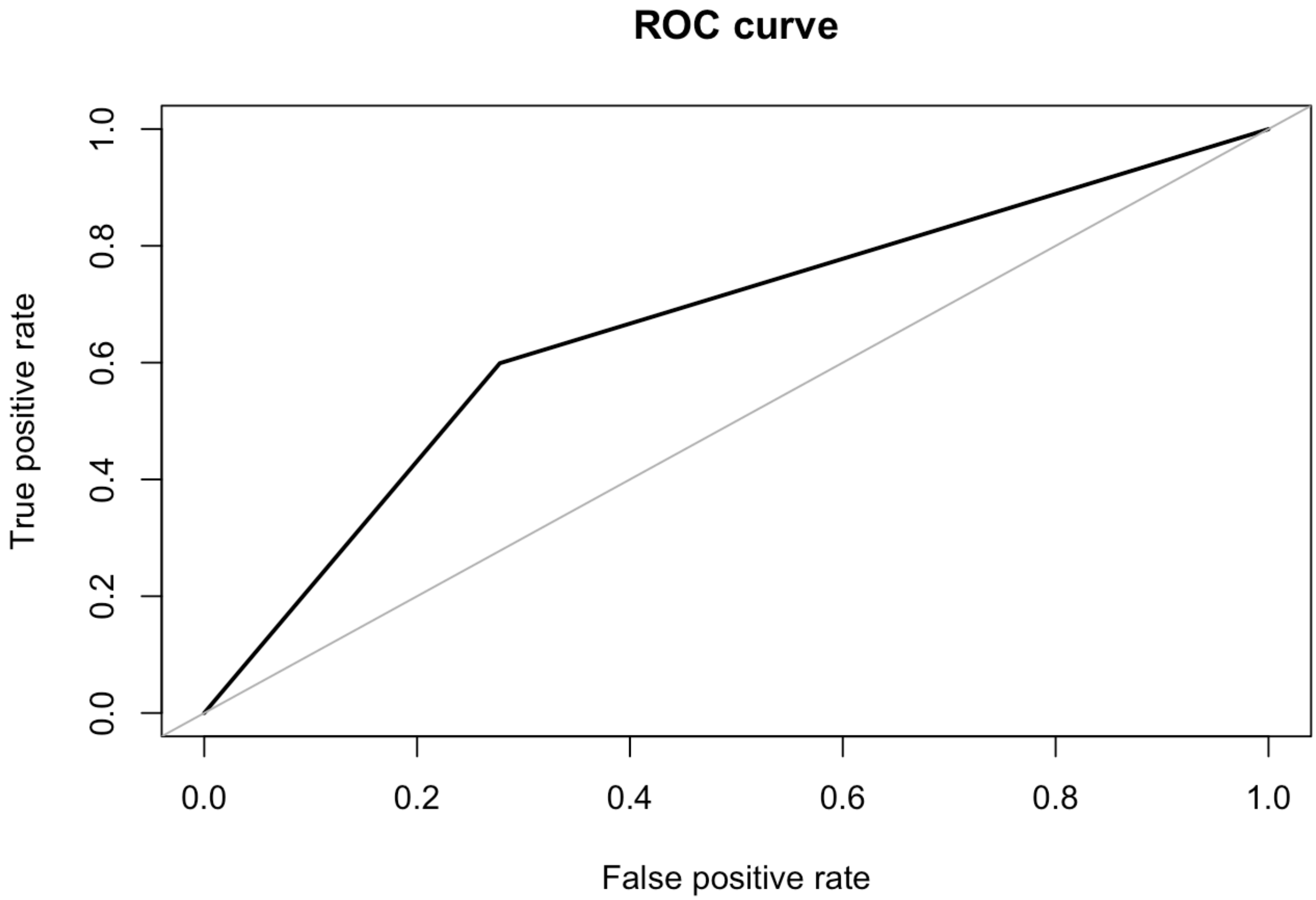


```
predicted.CHD <- predict(fit5, test, type = 'class')
```

```
cmat <- confusionMatrix(as.factor(predicted.CHD), as.factor(test$TenYearCHD), positiv
e = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 143  81
##          1  55 121
##                                           
##                Accuracy : 0.66            
##                  95% CI : (0.6113, 0.7063)
##     No Information Rate : 0.505           
##     P-Value [Acc > NIR] : 2.723e-10       
##                                           
##                   Kappa : 0.3208          
##                                           
##  Mcnemar's Test P-Value : 0.03205         
##                                           
##             Sensitivity : 0.5990          
##             Specificity : 0.7222          
##          Pos Pred Value : 0.6875          
##          Neg Pred Value : 0.6384          
##              Prevalence : 0.5050          
##          Detection Rate : 0.3025          
##    Detection Prevalence : 0.4400          
##       Balanced Accuracy : 0.6606          
##                                           
##        'Positive' Class : 1               
## 
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(predicted.CHD))
```



```
## Area under the curve (AUC): 0.661
```

8. Extreme Gradient Boosting Model

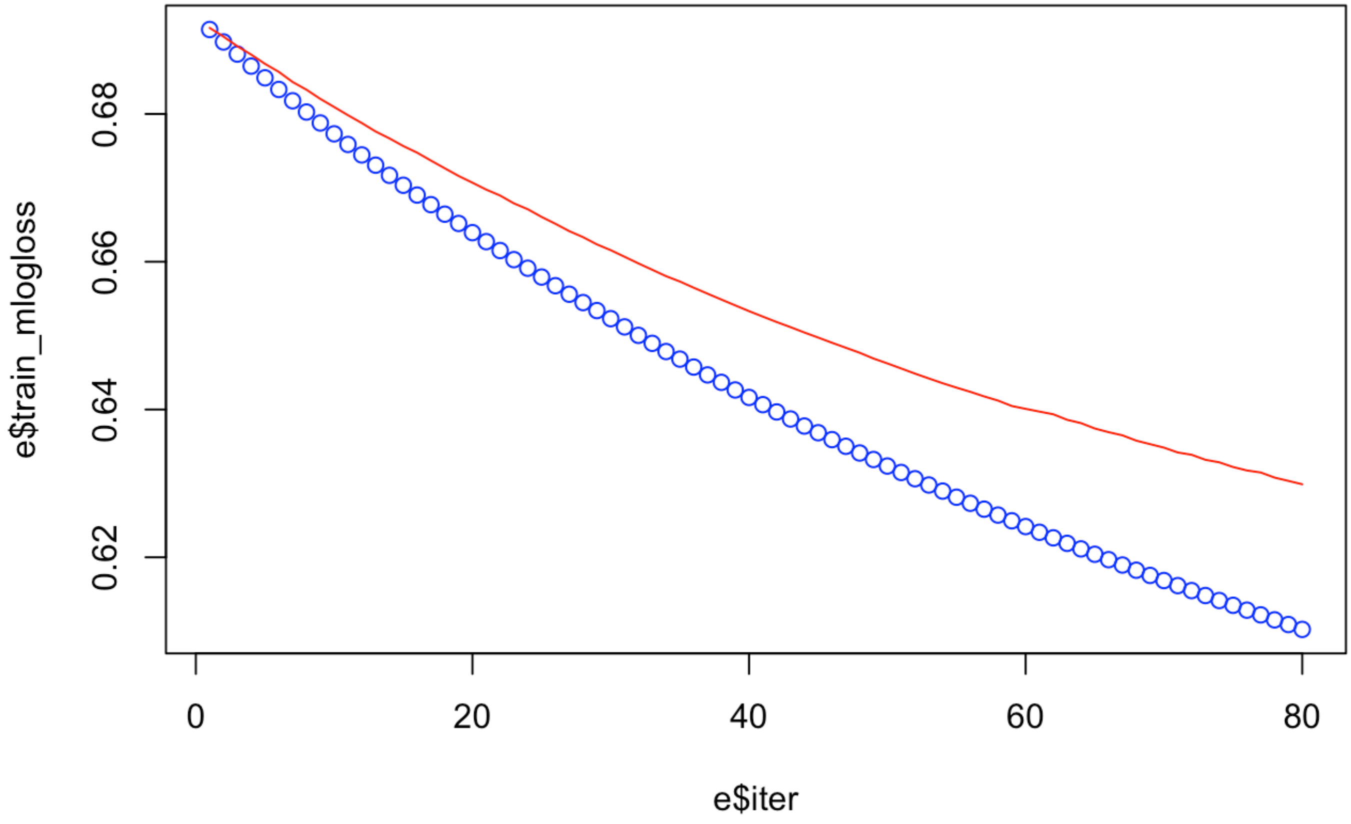
```
#Creating a matrix, one-hot encoding for factor variable
training <- sparse.model.matrix(TenYearCHD ~ .-1, data = train)     #independent vari
able
train_label <-  as.numeric(levels(train$TenYearCHD))[train$TenYearCHD] #dependent var
iable
train_matrix <- xgb.DMatrix(data = as.matrix(training), label = train_label)

testing <- sparse.model.matrix(TenYearCHD~.-1, data = test)
test_label <- test[,"TenYearCHD"]
test_matrix <- xgb.DMatrix(data = as.matrix(testing), label = test_label)
```

```
#Defining parameters
nc <- length(unique(train_label))
xgb_params <- list("objective" = "multi:softprob",
                   "eval_metric" = "mlogloss",
                   "num_class" = nc)
watchlist <- list(train = train_matrix, test = test_matrix)
```

```
#XGBoost Model
set.seed(333)
best_model <- xgb.train(params = xgb_params,
                       data = train_matrix,
                       nrounds = 80,
                       watchlist = watchlist,
                       eta = 0.01,
                       max.depth = 3,
                       gamma = 0,
                       subsample = 1,
                       colsample_bytree = 1,
                       missing = NA)
```

```
## [12:22:47] WARNING: amalgamation/../src/learner.cc:573:
## Parameters: { "missing" } might not be used.
##
##   This may not be accurate due to some parameters are only used in language bindin
gs but
##   passed down to XGBoost core.  Or some parameters are not used but slip through t
his
##   verification. Please open an issue if you find above cases.
##
##
## [1]  train-mlogloss:0.691429 test-mlogloss:0.691648
## [2]  train-mlogloss:0.689744 test-mlogloss:0.690469
## [3]  train-mlogloss:0.688102 test-mlogloss:0.689167
## [4]  train-mlogloss:0.686474 test-mlogloss:0.688035
## [5]  train-mlogloss:0.684889 test-mlogloss:0.686782
## [6]  train-mlogloss:0.683316 test-mlogloss:0.685696
## [7]  train-mlogloss:0.681783 test-mlogloss:0.684325
## [8]  train-mlogloss:0.680262 test-mlogloss:0.683282
## [9]  train-mlogloss:0.678780 test-mlogloss:0.682042
## [10] train-mlogloss:0.677311 test-mlogloss:0.680943
## [11] train-mlogloss:0.675877 test-mlogloss:0.679831
## [12] train-mlogloss:0.674457 test-mlogloss:0.678773
## [13] train-mlogloss:0.673070 test-mlogloss:0.677626
## [14] train-mlogloss:0.671695 test-mlogloss:0.676692
## [15] train-mlogloss:0.670353 test-mlogloss:0.675658
## [16] train-mlogloss:0.669025 test-mlogloss:0.674761
## [17] train-mlogloss:0.667726 test-mlogloss:0.673689
## [18] train-mlogloss:0.666440 test-mlogloss:0.672647
## [19] train-mlogloss:0.665182 test-mlogloss:0.671610
## [20] train-mlogloss:0.663938 test-mlogloss:0.670698
## [21] train-mlogloss:0.662720 test-mlogloss:0.669758
## [22] train-mlogloss:0.661516 test-mlogloss:0.668962
## [23] train-mlogloss:0.660284 test-mlogloss:0.667898
## [24] train-mlogloss:0.659123 test-mlogloss:0.667108
## [25] train-mlogloss:0.657930 test-mlogloss:0.666067
## [26] train-mlogloss:0.656758 test-mlogloss:0.665136
## [27] train-mlogloss:0.655606 test-mlogloss:0.664143
## [28] train-mlogloss:0.654474 test-mlogloss:0.663327
## [29] train-mlogloss:0.653392 test-mlogloss:0.662348
## [30] train-mlogloss:0.652296 test-mlogloss:0.661565
## [31] train-mlogloss:0.651183 test-mlogloss:0.660673
## [32] train-mlogloss:0.650037 test-mlogloss:0.659776
## [33] train-mlogloss:0.648957 test-mlogloss:0.658914
## [34] train-mlogloss:0.647846 test-mlogloss:0.658047
## [35] train-mlogloss:0.646824 test-mlogloss:0.657326
## [36] train-mlogloss:0.645745 test-mlogloss:0.656485
## [37] train-mlogloss:0.644684 test-mlogloss:0.655679
## [38] train-mlogloss:0.643678 test-mlogloss:0.654882
## [39] train-mlogloss:0.642647 test-mlogloss:0.654095
## [40] train-mlogloss:0.641635 test-mlogloss:0.653314
## [41] train-mlogloss:0.640657 test-mlogloss:0.652571
## [42] train-mlogloss:0.639672 test-mlogloss:0.651840
## [43] train-mlogloss:0.638720 test-mlogloss:0.651148
## [44] train-mlogloss:0.637762 test-mlogloss:0.650424
## [45] train-mlogloss:0.636859 test-mlogloss:0.649738
## [46] train-mlogloss:0.635927 test-mlogloss:0.649036
## [47] train-mlogloss:0.635022 test-mlogloss:0.648357
## [48] train-mlogloss:0.634115 test-mlogloss:0.647682
## [49] train-mlogloss:0.633235 test-mlogloss:0.646900
## [50] train-mlogloss:0.632351 test-mlogloss:0.646241
## [51] train-mlogloss:0.631482 test-mlogloss:0.645555
## [52] train-mlogloss:0.630632 test-mlogloss:0.644860
## [53] train-mlogloss:0.629785 test-mlogloss:0.644204
## [54] train-mlogloss:0.628958 test-mlogloss:0.643558
## [55] train-mlogloss:0.628132 test-mlogloss:0.642963
## [56] train-mlogloss:0.627311 test-mlogloss:0.642394
## [57] train-mlogloss:0.626513 test-mlogloss:0.641773
## [58] train-mlogloss:0.625714 test-mlogloss:0.641212
## [59] train-mlogloss:0.624941 test-mlogloss:0.640475
## [60] train-mlogloss:0.624157 test-mlogloss:0.640091
## [61] train-mlogloss:0.623387 test-mlogloss:0.639718
## [62] train-mlogloss:0.622632 test-mlogloss:0.639356
## [63] train-mlogloss:0.621886 test-mlogloss:0.638608
## [64] train-mlogloss:0.621136 test-mlogloss:0.638161
## [65] train-mlogloss:0.620409 test-mlogloss:0.637432
## [66] train-mlogloss:0.619678 test-mlogloss:0.636934
## [67] train-mlogloss:0.618953 test-mlogloss:0.636507
## [68] train-mlogloss:0.618251 test-mlogloss:0.635788
## [69] train-mlogloss:0.617546 test-mlogloss:0.635311
## [70] train-mlogloss:0.616854 test-mlogloss:0.634843
## [71] train-mlogloss:0.616166 test-mlogloss:0.634186
## [72] train-mlogloss:0.615490 test-mlogloss:0.633877
## [73] train-mlogloss:0.614824 test-mlogloss:0.633193
## [74] train-mlogloss:0.614150 test-mlogloss:0.632855
## [75] train-mlogloss:0.613501 test-mlogloss:0.632218
## [76] train-mlogloss:0.612846 test-mlogloss:0.631752
## [77] train-mlogloss:0.612207 test-mlogloss:0.631475
## [78] train-mlogloss:0.611533 test-mlogloss:0.630788
## [79] train-mlogloss:0.610900 test-mlogloss:0.630339
## [80] train-mlogloss:0.610242 test-mlogloss:0.629884
```

```
e <- data.frame(best_model$evaluation_log)
plot(e$iter, e$train_mlogloss, col = 'blue')
lines(e$iter, e$test_mlogloss, col = 'red')
```

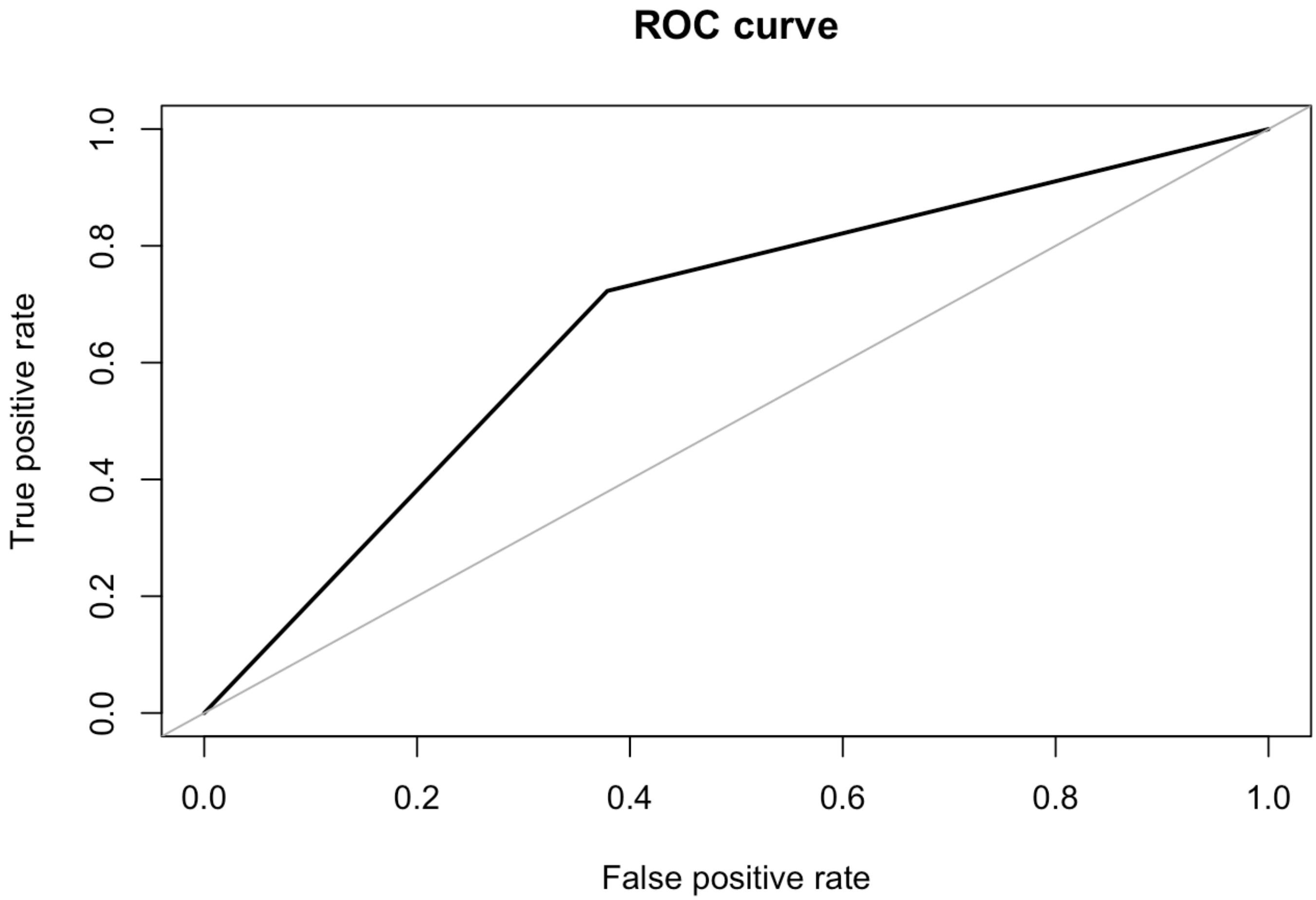


```
pred <- predict(best_model, newdata = test_matrix)
prediction <- matrix(pred, nrow = nc, ncol = length(pred)/nc) %>%
t() %>%
data.frame() %>%
mutate(label = test_label, max_prob = max.col(., "last")-1)
```

```
cmat <- confusionMatrix(as.factor(prediction$max_prob), as.factor(test$TenYearCHD), p
ositive = "1")
cmat
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 123  56
##          1  75 146
##                                           
##                Accuracy : 0.6725          
##                  95% CI : (0.6241, 0.7183)
##     No Information Rate : 0.505           
##     P-Value [Acc > NIR] : 9.059e-12       
##                                           
##                   Kappa : 0.3443          
##                                           
##  Mcnemar's Test P-Value : 0.1158          
##                                           
##             Sensitivity : 0.7228          
##             Specificity : 0.6212          
##          Pos Pred Value : 0.6606          
##          Neg Pred Value : 0.6872          
##              Prevalence : 0.5050          
##          Detection Rate : 0.3650          
##    Detection Prevalence : 0.5525          
##       Balanced Accuracy : 0.6720          
##                                           
##        'Positive' Class : 1               
## 
```

```
roc.curve(as.numeric(test$TenYearCHD), as.numeric(prediction$max_prob))
```



```
## Area under the curve (AUC): 0.672
```

Conclusions:

Our best performing model for this dataset , for predicting if a person will have Coronary Heart Disease in next ten years , was Random Forest with Accuracy of 76.25% and AUC of 0.76 We also observed some of the features that are significant for predicting Ten Year CHD were: 1. gender 2. age 3. systolic BP 4. BMI 5. diabetes and glucose Future scope: We believe if we have more relevant data we can make better predictions, for eg. currently in our dataset not many are current Smokers or not many people have diabetes. Collecting more varied population data can help capture more trends in data and make better predictions. Also more work on complex feature engineering and hyperparameter tuning can be done to improve the performance of our models. We hope projects and analyses like these can help medical research and make people’s lives better.