Framingham Heart Study 4240 subset

Uohna Thiessen

February 23, 2017

Step 1. Introduction: Framingham Heart Study Project

The Ten Year Chronic Heart Disease (TenYearCHD) Risk Factor calculator is available as and software application online. It requires the input of four factors: "age", "gender", "Cholesterol" and "Blood Pressure". Recently, some forms of the application now include a diagnosis of "diabetes" response. The calculation of the probability of CHD is based on a model built from the Framingham Heart Study dataset. It is my contention that the measure of diabetes is not as good a risk factor as the actual measure of the blood glucose level. I have used a subset of the FHS dataset to build two models, one with and without "glucose" and evaluated the comparable quality of the two. I contend that including "glucose" specifically blood glucose levels, improve the accuracy of the model.

Step 2. Set up environment for Modelling

```
library(dplyr)
```

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

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

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

```
library(ggplot2)
library(caTools)
library(tidyr)
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
library(statisticalModeling)
library(ROCR)
library(rpart.plot)
```

```
## Loading required package: rpart
```

```
library(corrplot)
```

Step 3. Load the dataset into R

```
framingham = read.csv("framingham_edx.csv")
fhs = framingham
str(fhs)
```

```
## 'data.frame':    4240 obs. of  16 variables:
##  $ male           : int  1 0 1 0 0 0 0 0 1 1 ...
##  $ age            : int  39 46 48 61 46 43 63 45 52 43 ...
##  $ education      : int  4 2 1 3 3 2 1 2 1 1 ...
##  $ currentSmoker  : int  0 0 1 1 1 0 0 1 0 1 ...
##  $ cigsPerDay     : int  0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentStroke: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentHyp   : int  0 0 0 1 0 1 0 0 1 1 ...
##  $ diabetes       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ totChol        : int  195 250 245 225 285 228 205 313 260 225 ...
##  $ sysBP          : num  106 121 128 150 130 ...
##  $ diaBP          : num  70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI            : num  27 28.7 25.3 28.6 23.1 ...
##  $ heartRate      : int  80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose        : int  77 76 70 103 85 99 85 78 79 88 ...
##  $ TenYearCHD     : int  0 0 0 1 0 0 1 0 0 0 ...
```

```
summary(fhs)
```

```
##       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   : 9.006   Mean   :0.02962   Mean   :0.005896   Mean   :0.3106
##  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.   : 48.0
##  1st Qu.:0.00000   1st Qu.:206.0   1st Qu.:117.0   1st Qu.: 75.0
```

```
##  Median :0.00000   Median :234.0   Median :128.0   Median : 82.0
##  Mean   :0.02571   Mean   :236.7   Mean   :132.4   Mean   : 82.9
##  3rd Qu.:0.00000   3rd Qu.:263.0   3rd Qu.:144.0   3rd Qu.: 90.0
##  Max.   :1.00000   Max.   :696.0   Max.   :295.0   Max.   :142.5
##                    NA's   :50
##       BMI          heartRate         glucose         TenYearCHD
##  Min.   :15.54   Min.   : 44.00   Min.   : 40.00   Min.   :0.0000
##  1st Qu.:23.07   1st Qu.: 68.00   1st Qu.: 71.00   1st Qu.:0.0000
##  Median :25.40   Median : 75.00   Median : 78.00   Median :0.0000
##  Mean   :25.80   Mean   : 75.88   Mean   : 81.96   Mean   :0.1519
##  3rd Qu.:28.04   3rd Qu.: 83.00   3rd Qu.: 87.00   3rd Qu.:0.0000
##  Max.   :56.80   Max.   :143.00   Max.   :394.00   Max.   :1.0000
##  NA's   :19      NA's   :1        NA's   :388
```

Step 4. Preprocess dataset: Choose statistically non-relevent risk factors

```
summary(glm(TenYearCHD ~ ., data = fhs, family = binomial))
```

```
##
## Call:
## glm(formula = TenYearCHD ~ ., family = binomial, data = fhs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9582  -0.5939  -0.4264  -0.2829   2.8409
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -8.328186   0.715449 -11.641  < 2e-16 ***
## male             0.555279   0.109033   5.093 3.53e-07 ***
## age              0.063515   0.006679   9.509  < 2e-16 ***
## education       -0.047767   0.049395  -0.967  0.33353
## currentSmoker    0.071601   0.156752   0.457  0.64783
## cigsPerDay       0.017914   0.006238   2.872  0.00408 **
## BPMeds           0.162496   0.234326   0.693  0.48802
## prevalentStroke  0.693660   0.489569   1.417  0.15652
## prevalentHyp     0.234208   0.138026   1.697  0.08973 .
## diabetes         0.039167   0.315506   0.124  0.90120
## totChol          0.002332   0.001127   2.070  0.03850 *
## sysBP            0.015403   0.003808   4.044 5.24e-05 ***
## diaBP           -0.004159   0.006438  -0.646  0.51831
## BMI              0.006672   0.012758   0.523  0.60097
## heartRate       -0.003246   0.004211  -0.771  0.44082
## glucose          0.007127   0.002234   3.190  0.00142 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3121.2  on 3657  degrees of freedom
## Residual deviance: 2754.5  on 3642  degrees of freedom
##   (582 observations deleted due to missingness)
## AIC: 2786.5
##
## Number of Fisher Scoring iterations: 5
```

```
fhs_sub = fhs[c("TenYearCHD", "age", "male", "totChol",
                "sysBP", "cigsPerDay", "glucose")]
```

Step 5. Clean/Tidy the dataset

a. Remove missing values

```
fhs_sub = na.omit(fhs_sub)
```

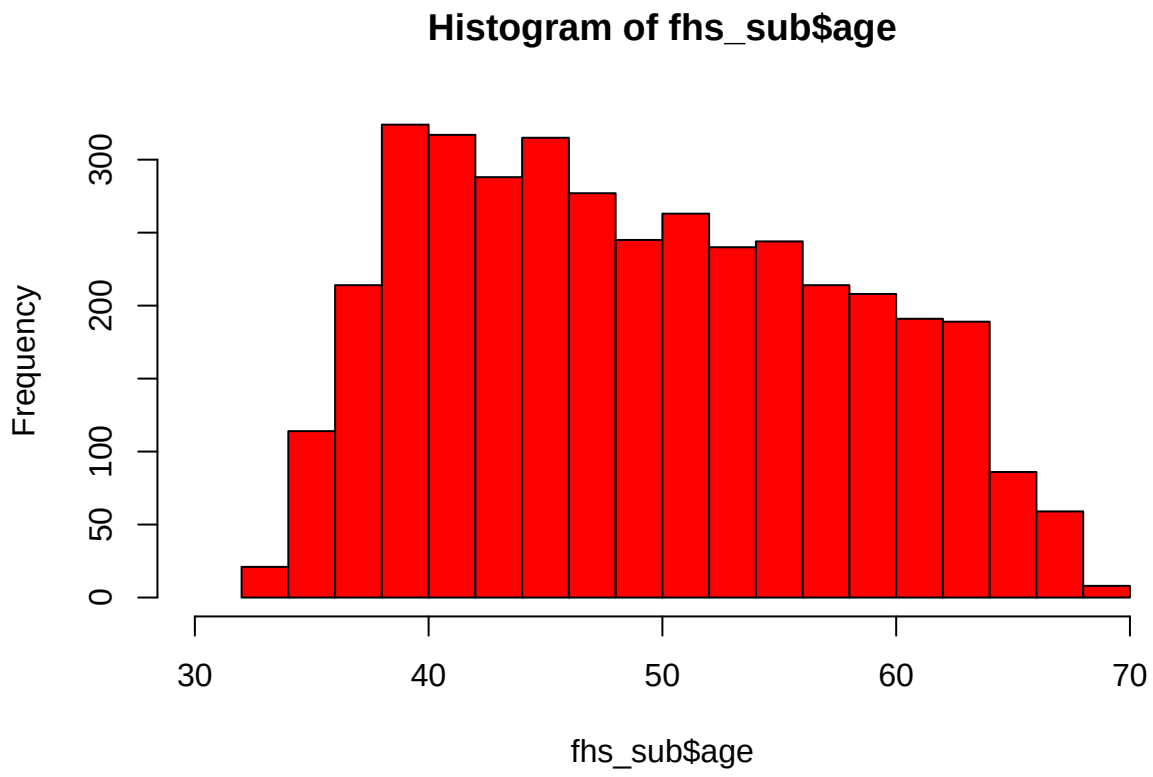
b. Factorize categorical values and rename

```
fhs_sub$male = as.factor(fhs_sub$male)
fhs_sub$male <- factor(fhs_sub$male, levels = c(0,1),
                       labels = c("female", "male"))
```

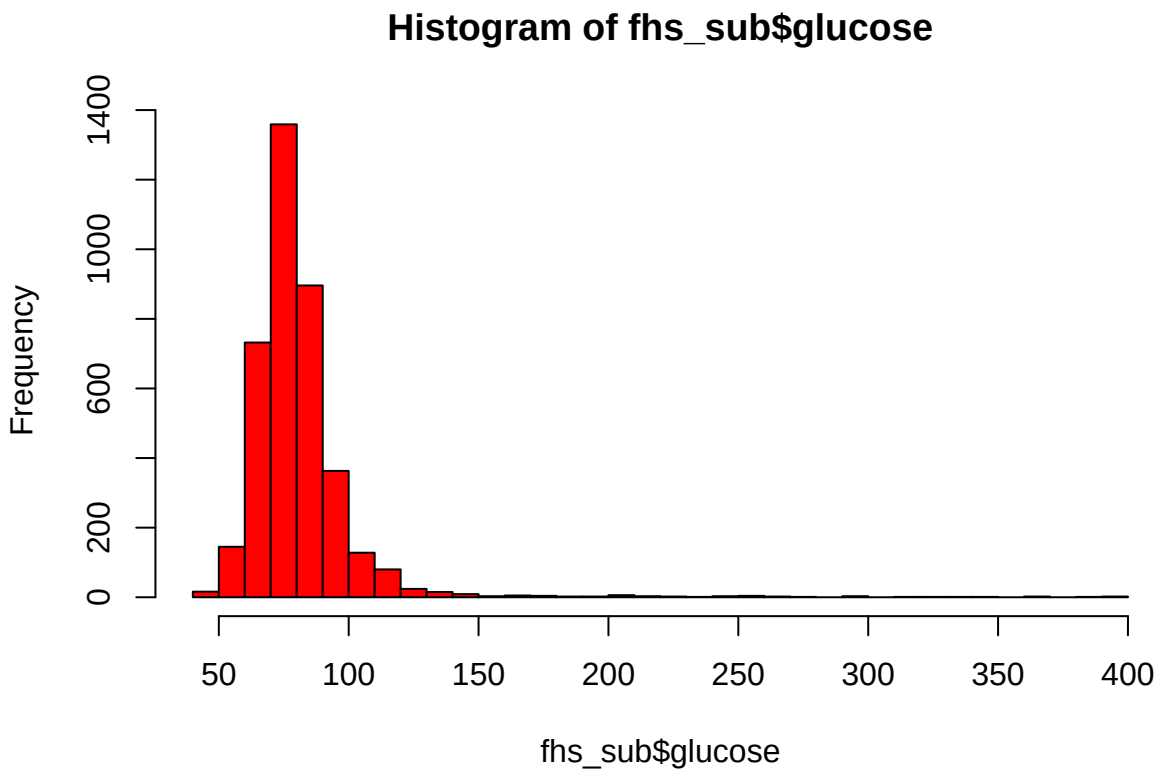
Step 5: Generate graphs for descriptive statistics

a. Histograms

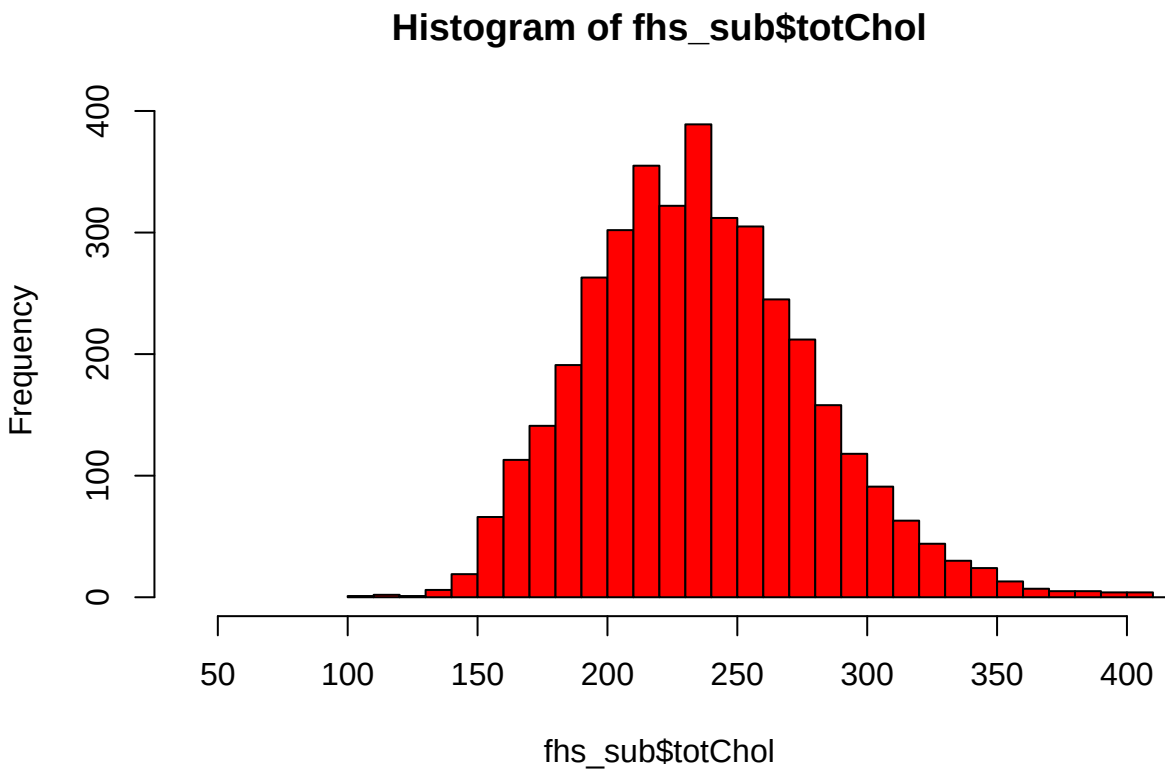
```
hist(fhs_sub$age, xlim = c(30,70), breaks = 25, col="Red")
```



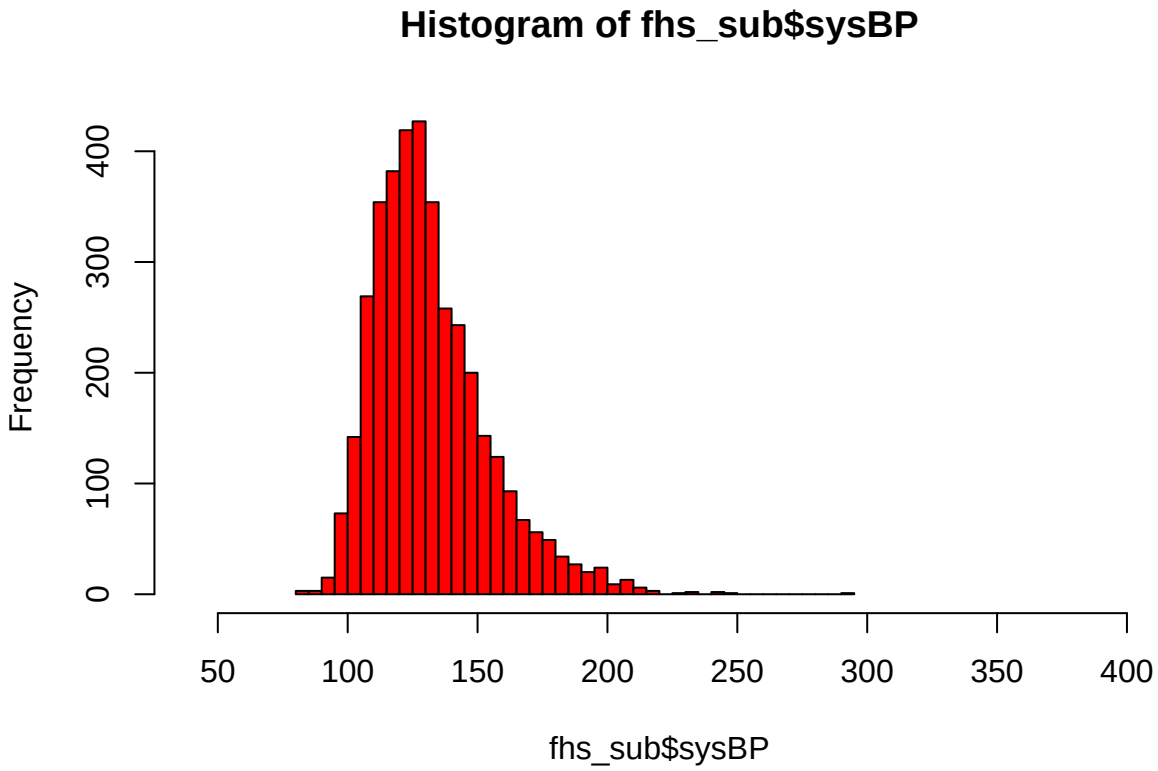

```
hist(fhs_sub$glucose, xlim = c(40,400), breaks=50,col="Red")
```




```
hist(fhs_sub$totChol, xlim = c(40,400), breaks=50,col="Red")
```

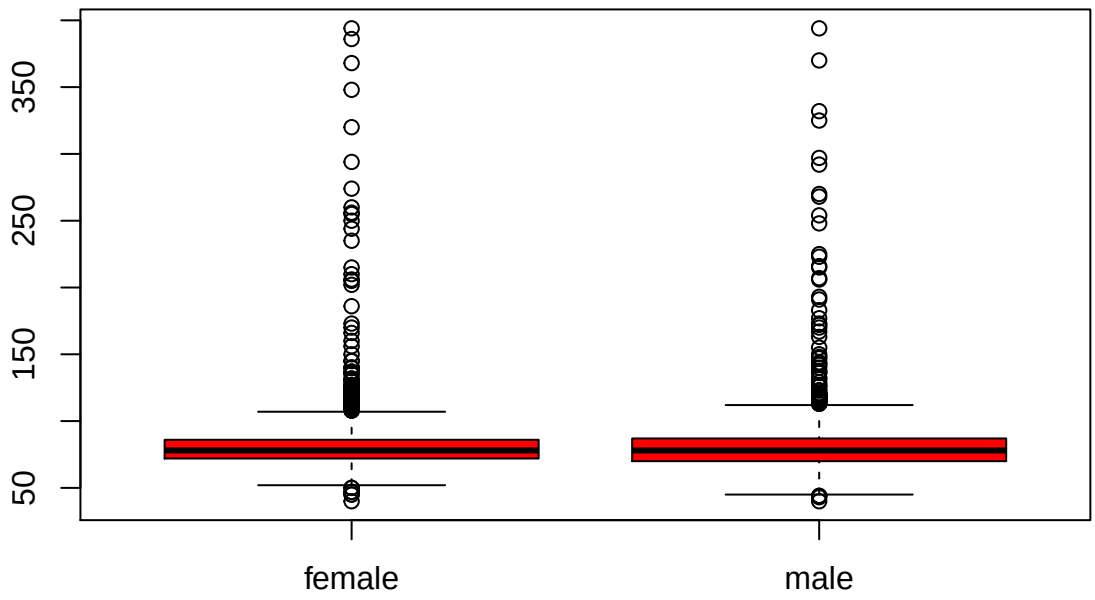



```
hist(fhs_sub$sysBP, xlim = c(40,400), breaks=50,col="Red")
```

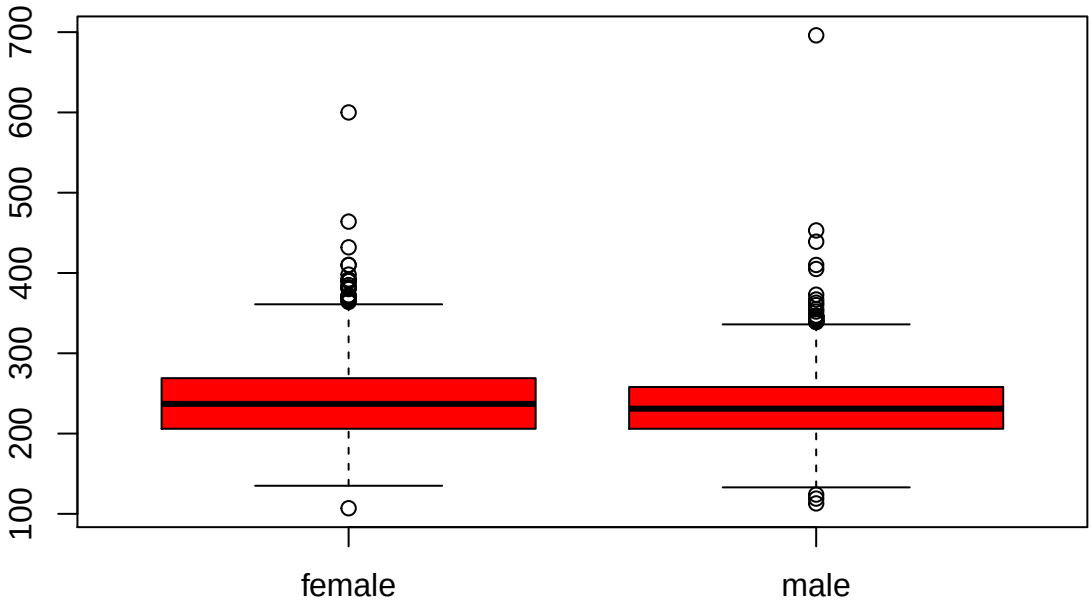


b. Boxplots

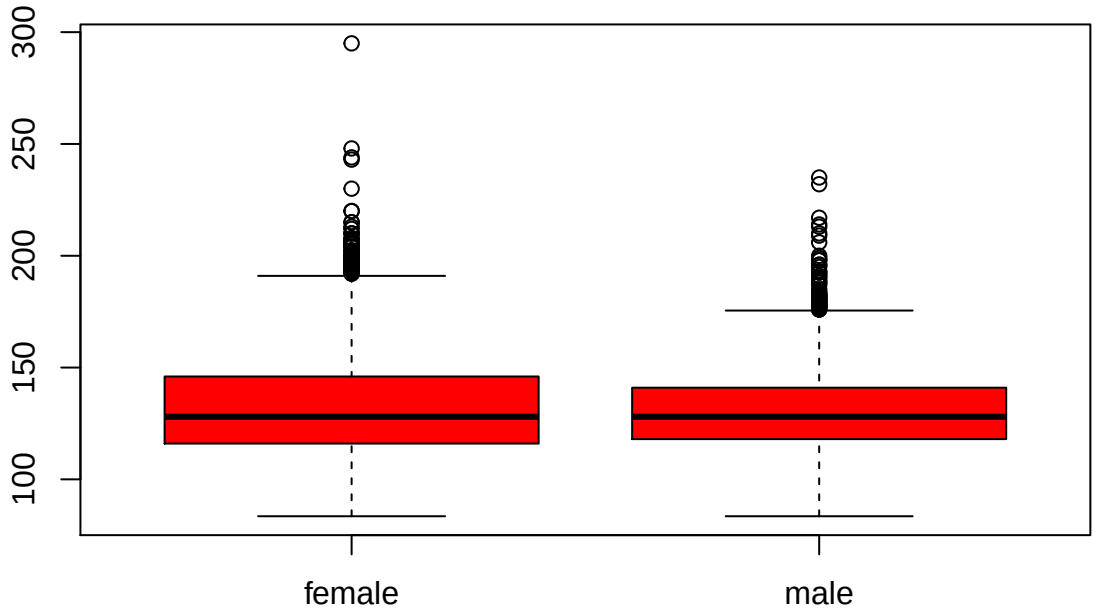
```
boxplot(glucose~male, data=fhs_sub, col="red")
```




```
boxplot(totChol~male, data=fhs_sub, col="red")
```



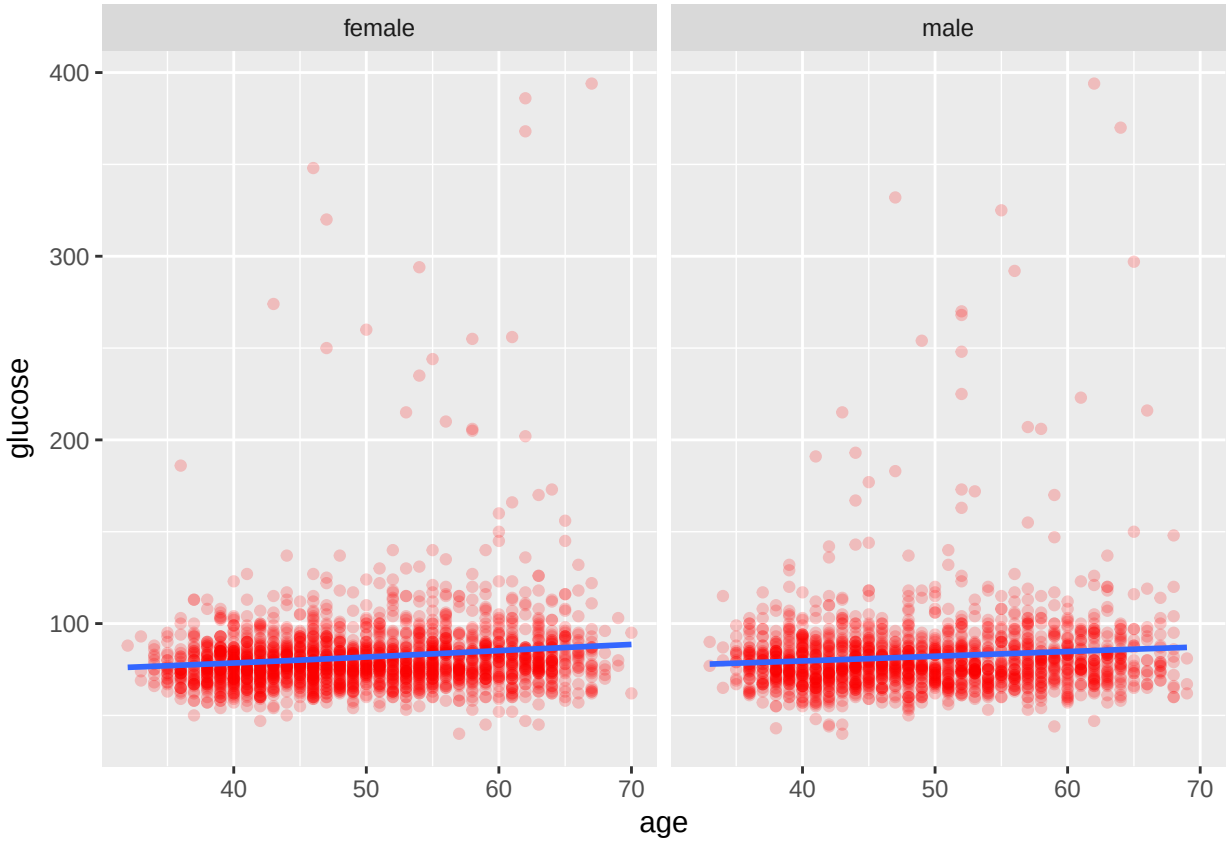

```
boxplot(sysBP~male, data=fhs_sub, col="red")
```



c. Scaterplots

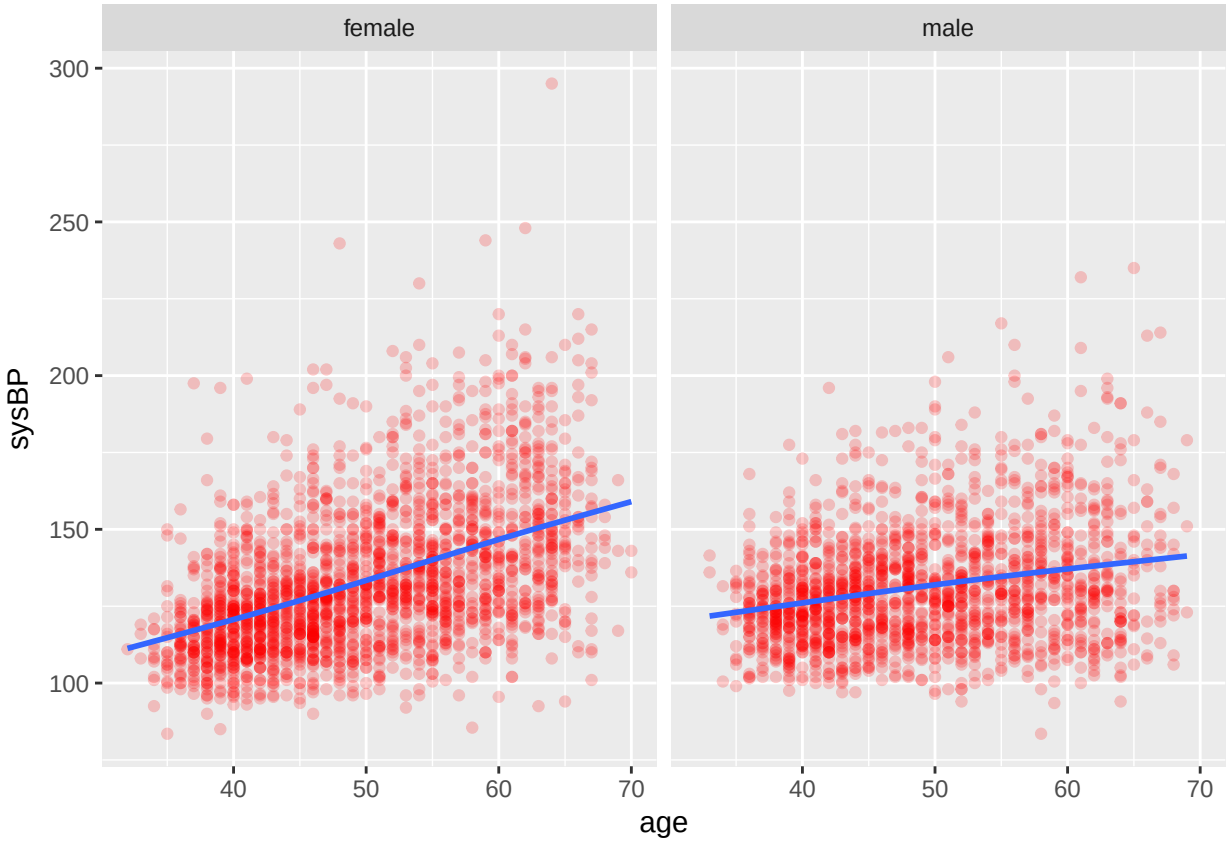
```
ggplot(fhs_sub, aes(age, glucose)) + geom_point(color = "red", alpha=0.2) +
                                    geom_smooth(se=FALSE) + facet_grid(.~male)
```

```
## `geom_smooth()` using method = 'gam'
```



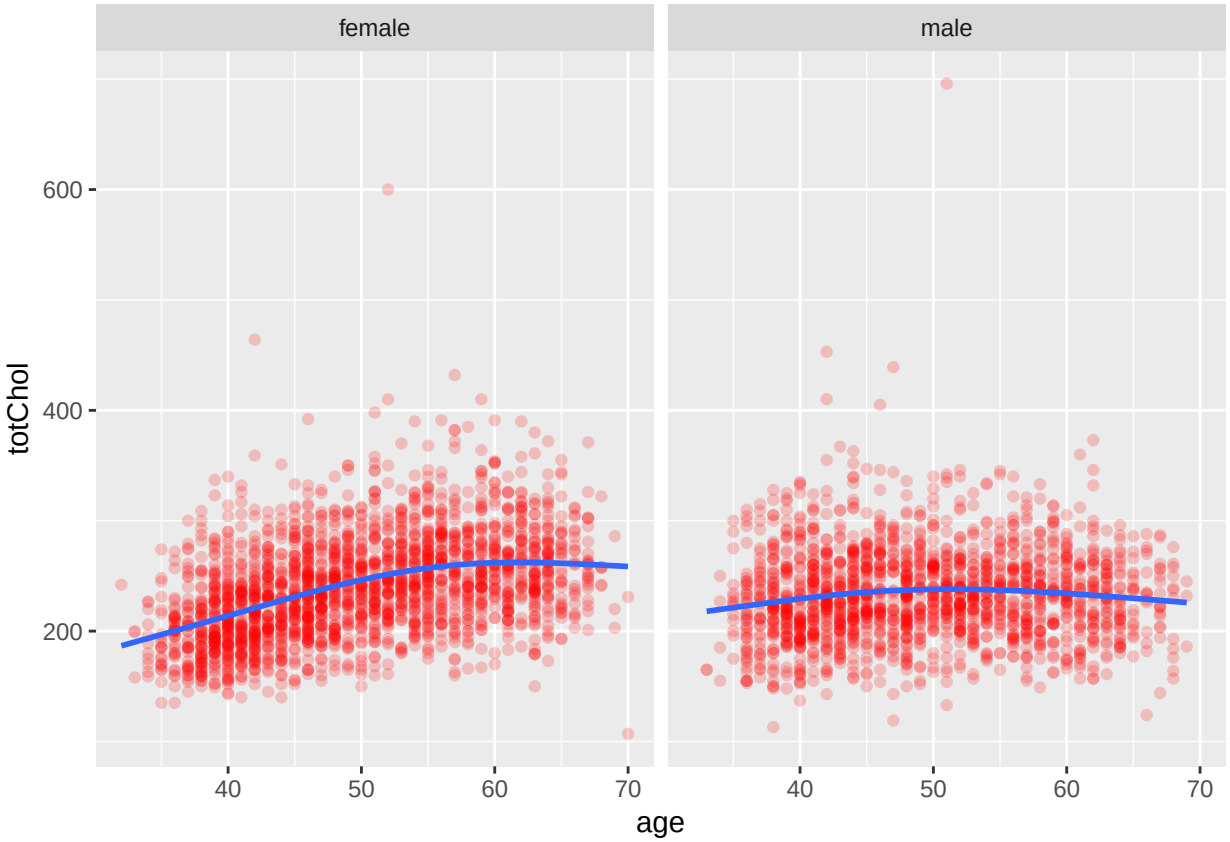

```
ggplot(fhs_sub, aes(age, sysBP)) + geom_point(color = "red", alpha=0.2) +
                                   geom_smooth(se=FALSE) + facet_grid(.~male)
```

```
## `geom_smooth()` using method = 'gam'
```



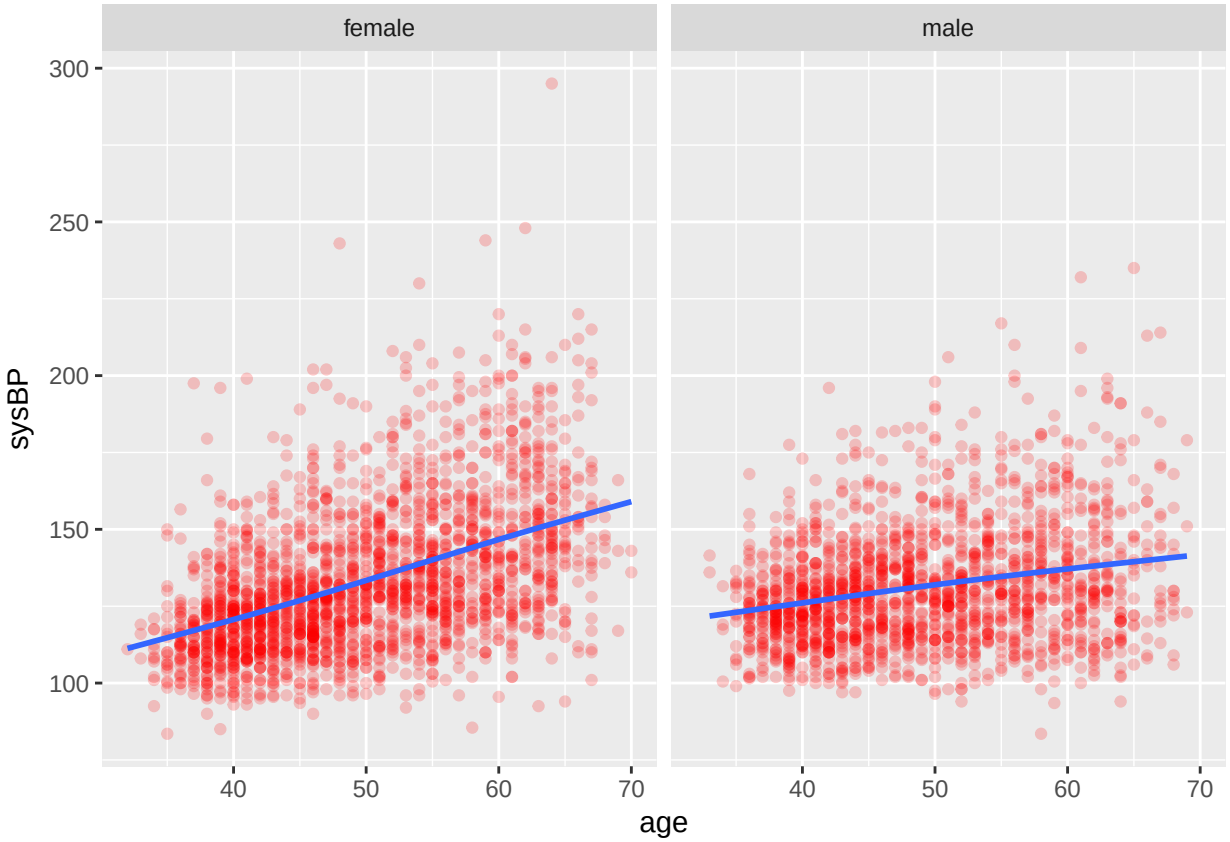

```
ggplot(fhs_sub, aes(age, totChol)) + geom_point(color = "red", alpha=0.2) +
                                    geom_smooth(se=FALSE) + facet_grid(.~male)
```

```
## `geom_smooth()` using method = 'gam'
```



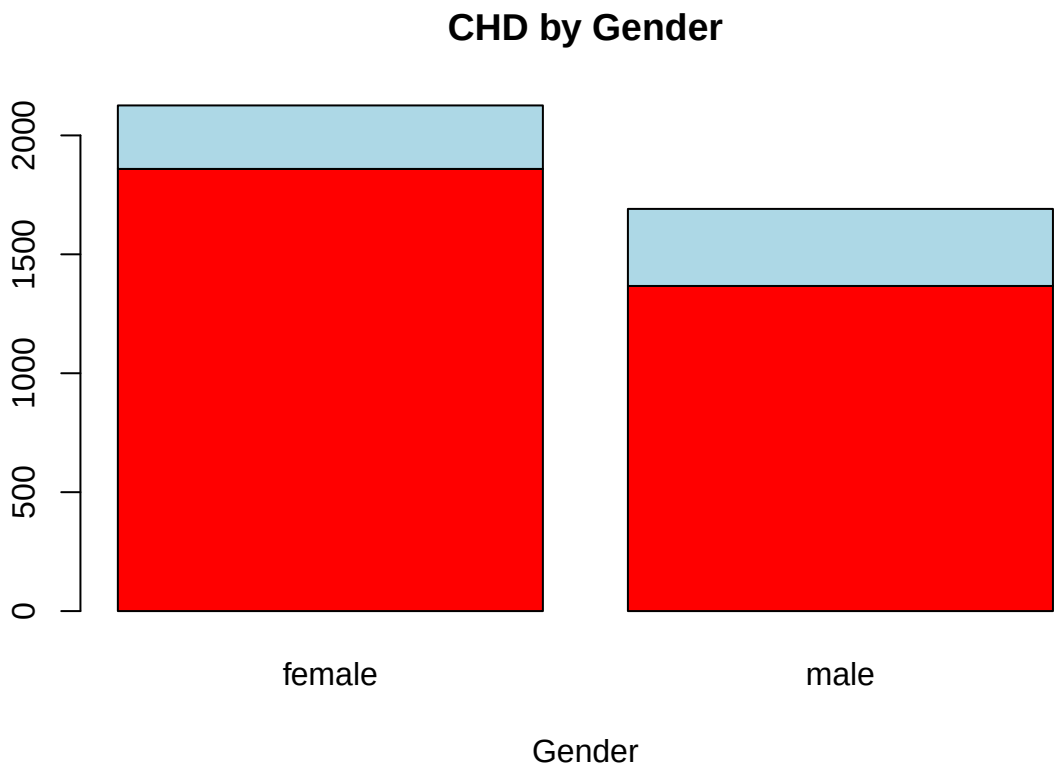

```
ggplot(fhs_sub, aes(age, sysBP)) + geom_point(color = "red", alpha=0.2) +
                                   geom_smooth(se=FALSE) + facet_grid(.~male)
```

```
## `geom_smooth()` using method = 'gam'
```



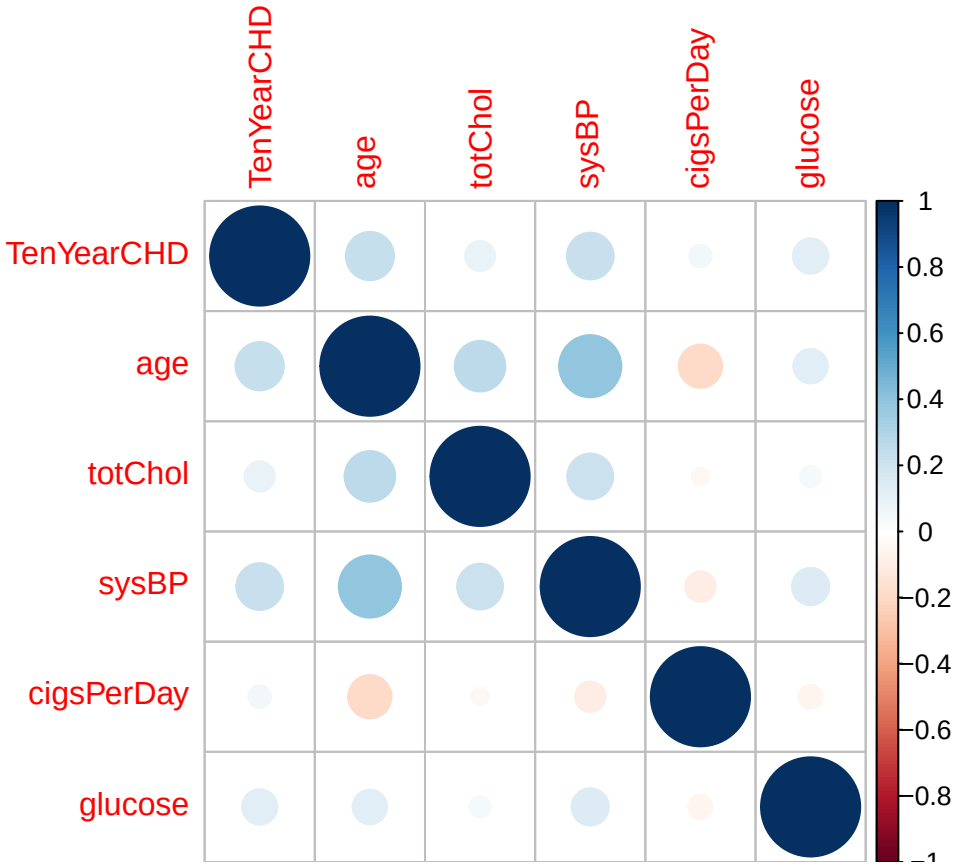
d. Bargraphs

```
counts <- table(fhs_sub$TenYearCHD, fhs_sub$male)
barplot(counts, main="CHD by Gender", xlab="Gender", col=c("red","lightblue"))
```



e. Generate Correlation matrix for numeric data

```
fhs_sub_num = fhs_sub[c("TenYearCHD", "age", "totChol", "sysBP", "cigsPerDay", "glucose")]
p = cor(fhs_sub_num)
corrplot(p)
```



Step 6: Build the Logistic Regress Model

a. Split data set into 'train' and 'test'

```
set.seed(1000)
split = sample.split(fhs_sub$TenYearCHD, SplitRatio = 0.65)
fhs_train = subset(fhs_sub, split==TRUE)
fhs_test = subset(fhs_sub, split!=TRUE)
```

b.Build two models (with 'glucose' and w/o 'glucose')

```
base_mod = glm(TenYearCHD ~ age + male + sysBP + totChol,
               data = fhs_train, family = binomial)
summary(base_mod)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ age + male + sysBP + totChol, family = binomial,
##     data = fhs_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6620  -0.6004  -0.4286  -0.2874   2.8405
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.665728   0.540091 -16.045  < 2e-16 ***
## age          0.064460   0.007553   8.535  < 2e-16 ***
```

```
## malemale     0.712110   0.121423   5.865  4.5e-09 ***
## sysBP        0.020137   0.002610   7.716  1.2e-14 ***
## totChol      0.002220   0.001292   1.718   0.0858 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2138.2  on 2480  degrees of freedom
## Residual deviance: 1893.8  on 2476  degrees of freedom
## AIC: 1903.8
##
## Number of Fisher Scoring iterations: 5
```

```
aug_mod = glm(TenYearCHD ~ age + male + sysBP + totChol + glucose,
                data = fhs_train, family = binomial)
summary(aug_mod)
```

```
##
## Call:
## glm(formula = TenYearCHD ~ age + male + sysBP + totChol + glucose,
##     family = binomial, data = fhs_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9904  -0.5953  -0.4254  -0.2865   2.8467
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.995799   0.556085 -16.177  < 2e-16 ***
## age          0.063237   0.007576   8.347  < 2e-16 ***
## malemale     0.716720   0.121803   5.884  4.0e-09 ***
## sysBP        0.019054   0.002640   7.217  5.3e-13 ***
## totChol      0.002326   0.001294   1.798  0.07225 .
## glucose      0.006100   0.001928   3.163  0.00156 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2138.2  on 2480  degrees of freedom
## Residual deviance: 1883.8  on 2475  degrees of freedom
## AIC: 1895.8
##
## Number of Fisher Scoring iterations: 5
```

Step 7. Cross Evaluation of the models

a. Generate first confusion matrix and calculate accuracy

```
base_mod_pred = predict(base_mod, type = "response", newdata = fhs_test)
table(fhs_test$TenYearCHD, base_mod_pred > 0.5)
```

```
##
##     FALSE TRUE
##   0  1116   13
```

```
##    1    197    10
```

Accuracy = $(1116 + 10)/(1116 + 197 + 13 + 10) = 0.843$

b. Generate second (aug_model) and calculate accuracy

```
aug_mod_pred = predict(aug_mod, type = "response", newdata = fhs_test)
table(fhs_test$TenYearCHD, aug_mod_pred > 0.5)
```

```
##
##      FALSE TRUE
##    0  1120    9
##    1   194   13
```

Accuracy = $(1120 + 13)/(1120 + 13 + 9 + 194) = 0.848$

8. Calculate AUC values

```
ROCRbase_mod = prediction(base_mod_pred, fhs_test$TenYearCHD)
as.numeric(performance(ROCRbase_mod, "auc")@ y.values)
```

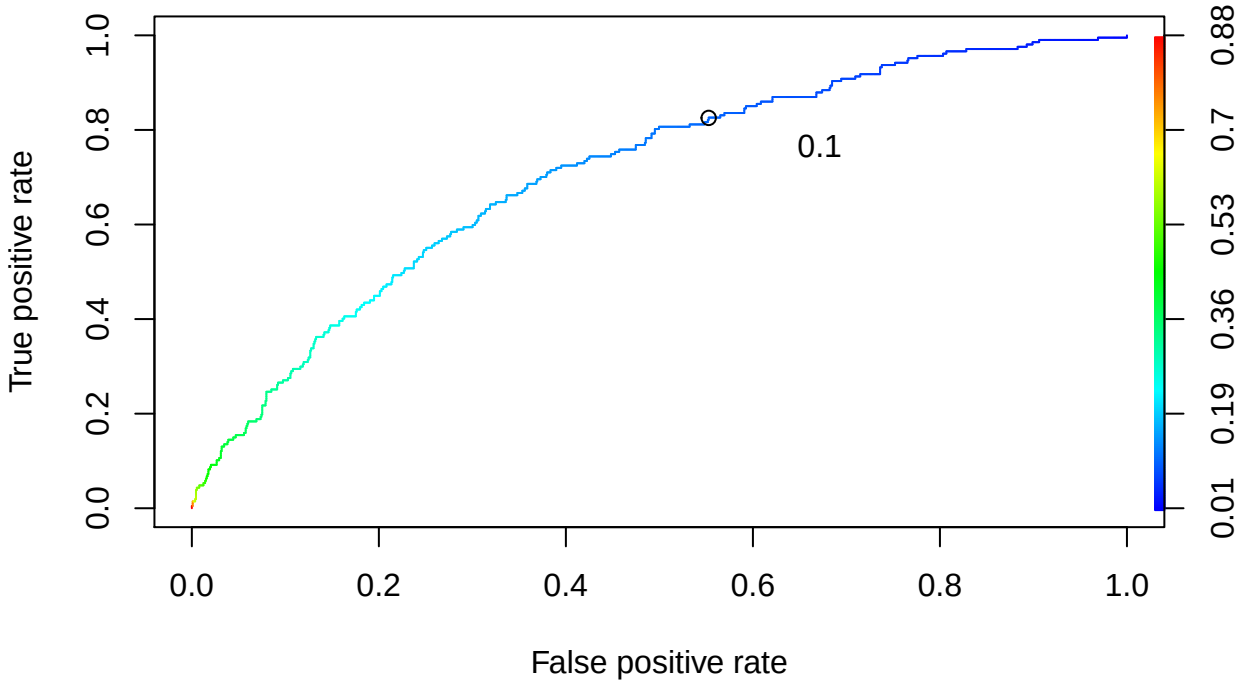
```
## [1] 0.7077658
```

```
ROCRaug_mod = prediction(aug_mod_pred, fhs_test$TenYearCHD)
as.numeric(performance(ROCRaug_mod, "auc")@ y.values)
```

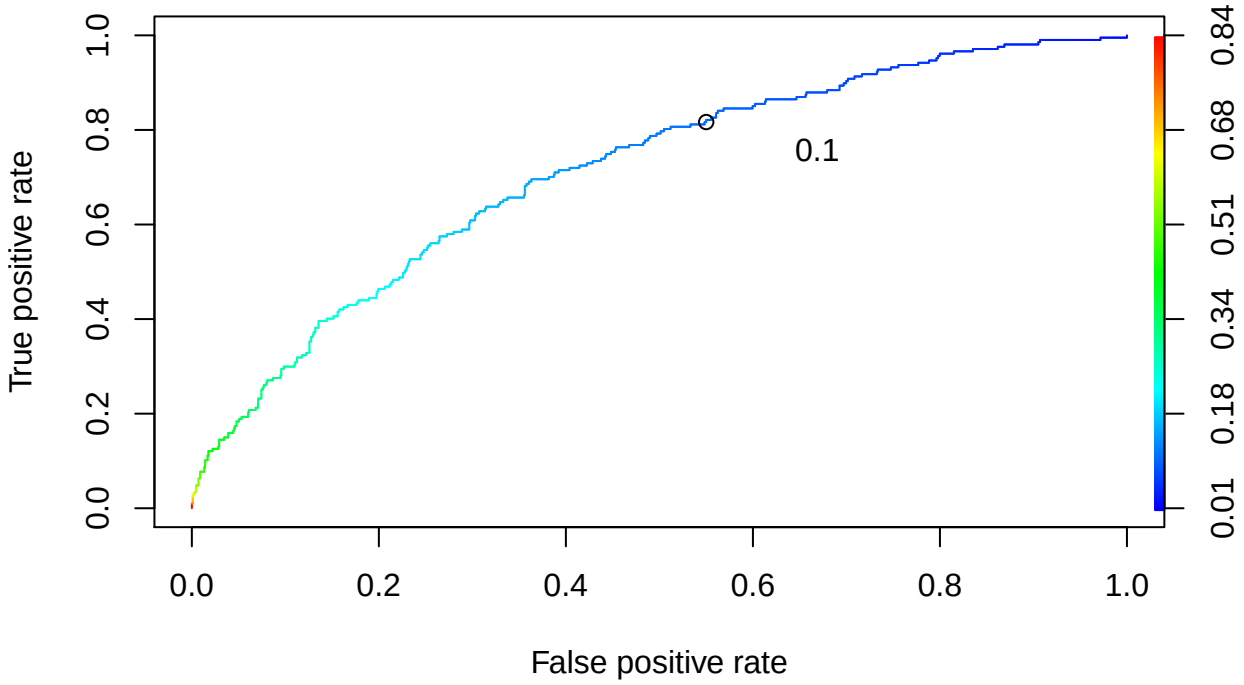
```
## [1] 0.7112703
```

Step 9. Plot ROC Curves

```
ROCRperf_base_mod= performance(ROCRbase_mod, "tpr", "fpr")
plot(ROCRperf_base_mod, colorize = TRUE, print.cutoffs.at=seq(0.1,0.1),
                                                  text.adj=c(-02, 1.7))
```

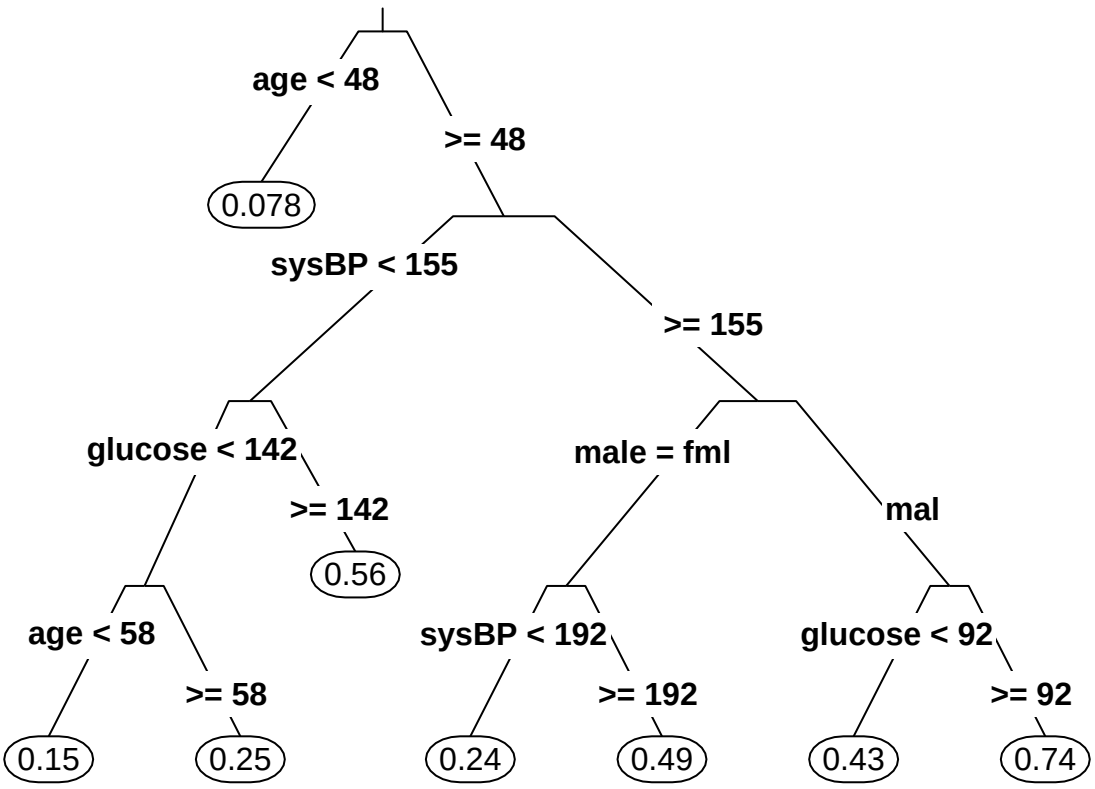



```
ROCRperf_aug_mod= performance(ROCRaug_mod, "tpr", "fpr")
plot(ROCRperf_aug_mod, colorize = TRUE, print.cutoffs.at=seq(0.1,0.1),
                                                    text.adj=c(-02, 1.7))
```



Step 10. Generate Recursive Partioning and Plot Regression Tree

```
rpart_mod = rpart(TenYearCHD ~ ., cp = 0.005, data = fhs_sub)
prp(rpart_mod, type = 3)
```



The regression tree indicates that a person, who is over 48 years, whose blood pressure is above 155, is male and has a blood sugar reading above 92, has a 74% chance of developing Chronic Heart Disease (CHD) in the next ten years.