MSDS6372 Project 2 - Osteoporosis in Women

Caroll Rodriguez, Rajat Chandna, Randall Hendrickson, Lokesh Maganti

August 18, 2018

Contents

Objective 1 - EDA and logistic regression model

TODO: remove below from project description Display the ability to perform EDA and build a logistic regression model

• Perform your logistic regression analysis and provide interpretation of the regression coefficients including hypothesis testing, and confidence intervals. For simplicity sake, you do not need to include interactions with this model. Comment on the practical vs statistical significance of the deemed important factors..

Logistical Considerations.

• Just like last time, this does not have to be extremely fancy in terms of the model building approach, let EDA, feature selection, and overall intuition guide you.

Introduction

TODO:

Data Description

The data set provided is about predicting whether a woman with osteoporosis will have another bone fracture. Of course getting a bone fracture is somewhat circumstantial, but with this disease every day life could trigger a break if the progression of the disease is strong.

The dataset included a total of 14 variables: 3 ID variables which tell us the subject, doctor and physical location of each record, 4 continuous variables (BMI, Weight, Height, and Age), 6 categorical variables (PRIORFRAC, PREMENO, MOMFRAC, PREMEO, MOMFRAC, ARMASSIST, SMOKE, RATERISK), and the response (FRACTURE). We were unable to find a mapping the subjects with their location to understand the mix of countries represented.

We have 500 subjects in the dataset of which 33% of the subjects have/had fractures.

Missing values were not detected in dataset. Special characters were removed from column headings. What we know/don't know about the sample (500)

Exploratory Analysis

Assumptions

This is a prospective study which means it's a study over time of a group of similar individuals who differ with respect to certain factors under a study and how these factors affect rates of a certain outcome (Fracture vs No-Fracture) Linearity - Independence of errors - Based on SUB_ID(Subject ID) we confirm each record is an independent sample. Multicollinearity - Weight and BMI are highly correlated but we will remove one from the analysis.

```
GLOW dataset:

Variable Name Type    #Unique

    SUB_ID  integer  500 - Identification Code (1 - n)
   SITE_ID  integer    6 - Study Site (1 - 6)
    PHY_ID  integer  127 - Physician ID code (128 unique codes)
 PRIORFRAC*  factor    2 - History of Prior Fracture (1: No, 2: Yes)
       AGE  integer   36 - Age at Enrollment (Years)
```

```
    WEIGHT   numeric   128 - Weight at enrollment (Kilograms)
    HEIGHT   integer    34 - Height at enrollment (Centimeters)
       BMI   numeric   409 - Body Mass Index (Kg/m^2)
   PREMENO*  factor      2 - Menopause before age 45 (1: No, 2: Yes)
   MOMFRAC*  factor      2 - Mother had hip fracture (1: No, 2: Yes)
 ARMASSIST*  factor      2 - Arms are needed to stand from a chair (1: No, 2: Yes)
     SMOKE*  factor      2 - Former or current smoker (1: No, 2: Yes)
   RATERISK* factor      3 - Self-reported risk of fracture (1: Less, 2: Same, 3: Greater)
  FRACSCORE  integer    12 - Fracture Risk Score (Composite Risk Score)
  FRACTURE*  factor      2 - Any fracture in first year (1: No, 2: Yes)
```


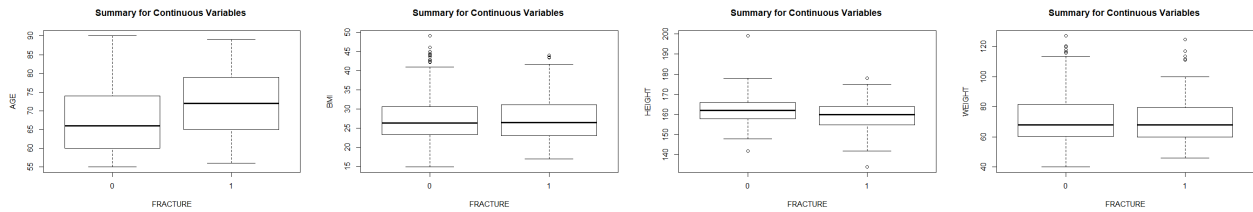**Figure 1 - Boxplots for Continuous Variables AGE, BMI, HEIGHT, WEIGHT**

In Figure 1, we see the boxplots for the continous variables AGE, WEIGHT, HEIGHT, BMI.
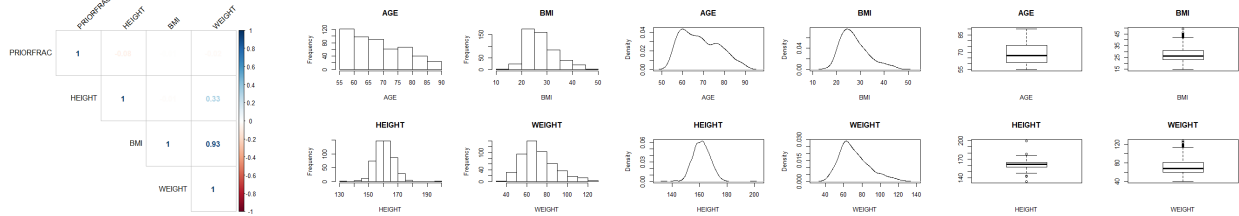

Figure 2 - Correlation, and Density plots for Continuous Variables AGE, BMI, HEIGHT, WEIGHT
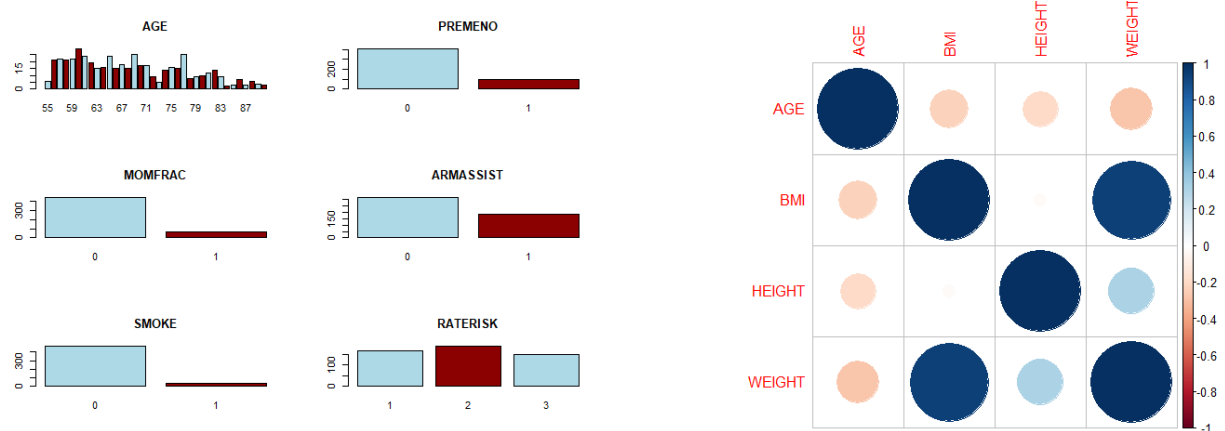

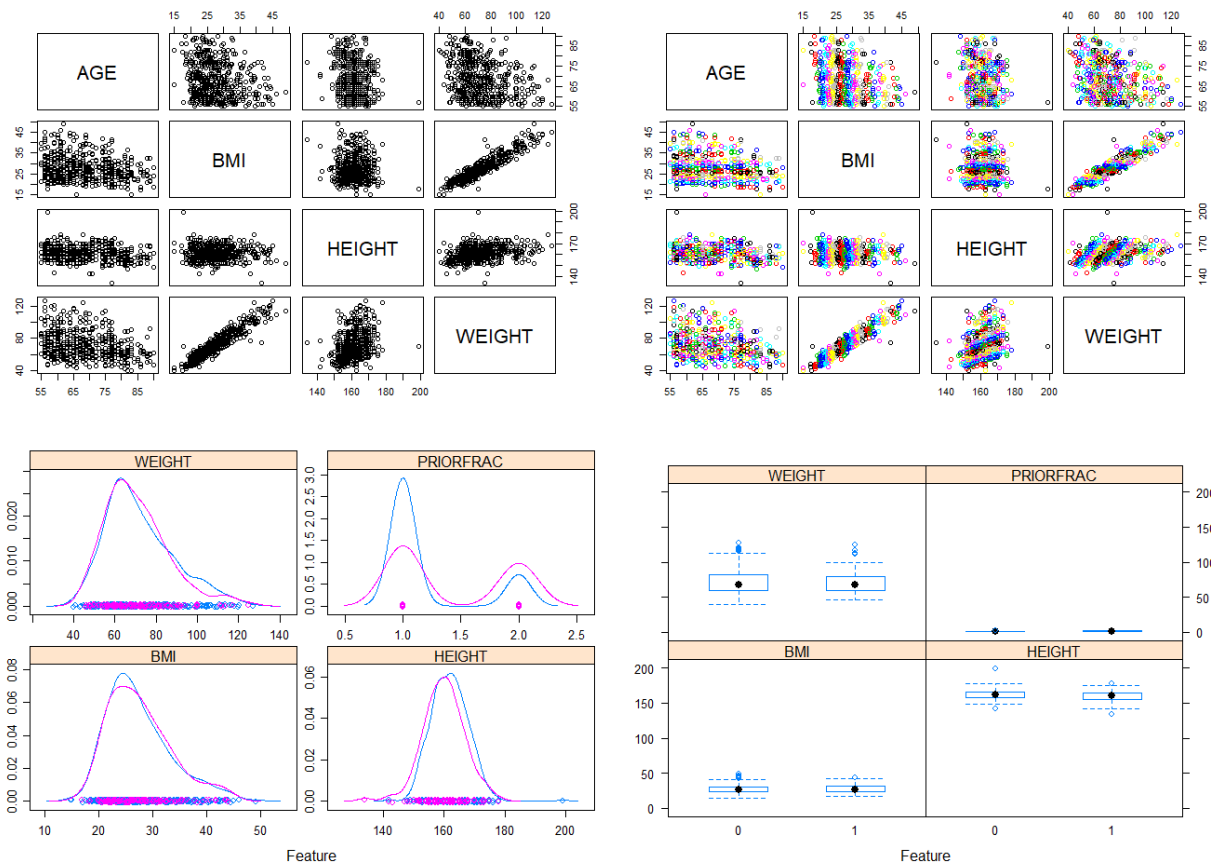**Figure 3 - Barplot (occurrances) and Multivariate Plots for Categorical and Continuous Variables**

Figure 4 - Scatterplots
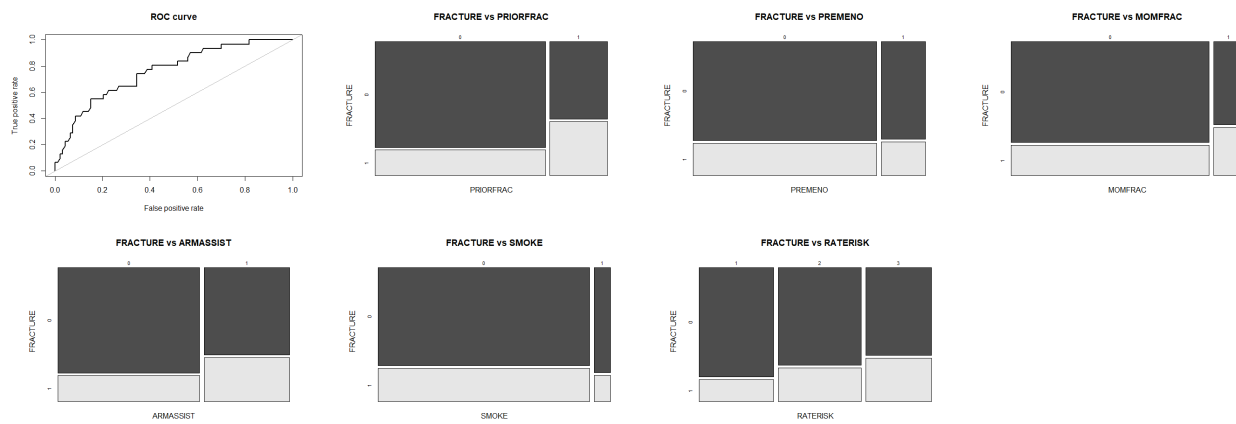
Figure 5 - ROC and 2-way Tables
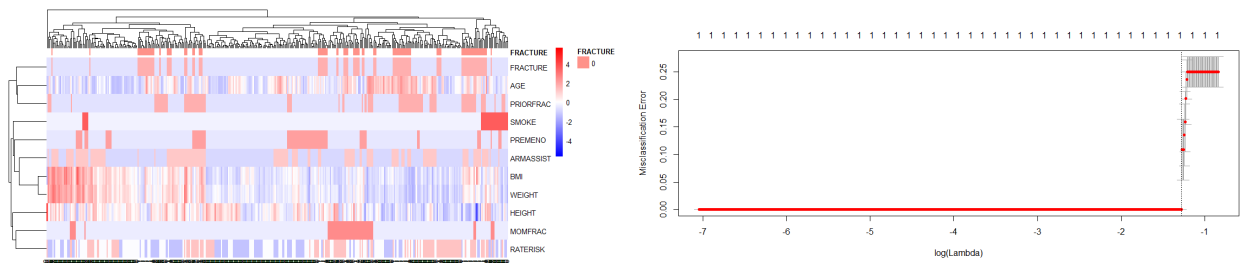
Figure 6 - Clustering

Restatement of Problem and the overall approach to solve it.

TODO:

Model Selection

TODO: add a description for the below selection

Simple Logistic Model Fit:

```
Call:
glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
    ARMASSIST + RATERISK, family = binomial(link = "logit"),
    data = trainingData)

Deviance Residuals:
    Min      1Q    Median      3Q      Max
-1.5491  -0.7377  -0.5763   0.2298   2.2214

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.33365    3.80104   0.877  0.38047
AGE          0.04347    0.01578   2.755  0.00587 **
HEIGHT      -0.04881    0.02165  -2.254  0.02418 *
PRIORFRAC1   0.22281    0.30097   0.740  0.45912
MOMFRAC1     0.33522    0.38263   0.876  0.38097
ARMASSIST1   0.68418    0.27861   2.456  0.01406 *
RATERISK.L   0.50762    0.24656   2.059  0.03951 *
RATERISK.Q  -0.06727    0.22219  -0.303  0.76209
```

Model Equation

$log(\pi) = 3.33365 + 0.04347 * \text{AGE} - 0.04881 * \text{HEIGHT} + 0.22281 * \text{PRIORFRAC} + 0.33522 * \text{MOMFRAC} + 0.68418 * \text{ARMASSIST} + 0.50762 * \text{RATERISK\_2} - 0.06727 * \text{RATERISK\_3}$

```
Coefficients and 95% Confidence Intervals converted to Normal Scale
             ODDs_Ratio      2.5 %        97.5 %
(Intercept) 28.0403767  0.0172620  5.378401e+04
AGE          1.0444253  1.0128322  1.077662e+00
HEIGHT       0.9523628  0.9118527  9.929059e-01
PRIORFRAC1   1.2495774  0.6857875  2.237992e+00
```

6

```
MOMFRAC1     1.3982464 0.6449924 2.917750e+00
ARMASSIST1   1.9821443 1.1469582 3.428352e+00
RATERISK.L   1.6613370 1.0283916 2.713240e+00
RATERISK.Q   0.9349462 0.6060378 1.451499e+00
```

Checking Assumptions

TODO: Lokesh please add assumptions

Interpretation of the Coefficients:

For AGE: All the other variables being constant, for every one-year increase in age of women, the odds of being getting a fracture in first year (versus not getting a fracture in first year) increases by a factor of 1.04(4% increase). The 95% Confidence Interval for this multiplicative factor is from 1.01(1% increase) to 1.08(8% increase).

For HEIGHT: All the other variables being constant, for every one unit increase in height of women, the odds of being getting a fracture in first year (versus not getting a fracture in first year) decreases by a factor of 0.95(5% decrease). The 95% Confidence Interval for this multiplicative factor is from 0.91(9% decrease) to 0.99(1% decrease).

** All the below coefficients are of format: The estimated odds for Person X with/without characteristic are M times the odds (of developing fracture in first year), for another Person Y without/with that characteristic. **

For PRIORFRAC: All the other variables being constant, the estimated odds for a woman, who has history of prior fracture, are 1.25 (25% more) times the odds of having fracture again in first year, for a woman who didn't have prior history of fracture. The 95% Confidence Interval for this estimated odds ratio is from 0.69 times to 2.23 times.

For MOMFRAC: All the other variables being constant, the estimated odds for a woman, whose mother had hip fracture, are 1.40 (40% more) times the odds of having fracture in first year, for a woman whose mother didn't have hip fracture. The 95% Confidence Interval for this estimated odds ratio is from 0.64 times to 2.91 times.

For ARMASSIST: All the other variables being constant, the estimated odds for a woman, who needed arms to stand from a chair, are 1.98 (98% more) times the odds of having fracture in first year, for a woman who didn't need arms to stand from a chair. The 95% Confidence Interval for this estimated odds ratio is from 1.15 times to 3.43 times.

For RATERISK.2: All the other variables being constant, the estimated odds for a woman, who self-reported that her risk for developing fracture is same as others of the same age, are 1.66 (66% more) times the odds of having fracture in first year, for a woman who self-reported that her risk for developing fracture is less than others of the same age. The 95% Confidence Interval for this estimated odds ratio is from 1.03 times to 2.71 times.

For RATERISK.3: All the other variables being constant, the estimated odds for a woman, who self-reported that her risk for developing fracture is greater than others of the same age, are 0.93 (7% less) times the odds of having fracture in first year, for a woman who self-reported that her risk for developing fracture is less than others of the same age. The 95% Confidence Interval for this estimated odds ratio is from 0.61 times to 1.45 times.

Final conclusions from the analysis of Objective 1

TODO:

take writeup from EDA and place here with reference to figures, etc

Objective 2 - additional competing models

- Record the predictive performance metrics from your simple, highly interpretable model from Objective 1.

- You must include one additional model which is also a more complicated logistic regression model than in Objective 1. By complicated, I do not mean that you include more predictors (that will be somewhat sorted out in Objective 1), but rather model complexity through interaction terms, new variables created by the group, or transformations.

- Create another competing model using just the continuous predictors and use LDA or QDA.

- Use a nonparameteric model approach as a competing model. Random forest for predictors that are both categorical and continuous or a k-nearest neighbors approach if just working with continuous predictors.

- Provide a summary table of the performance across the competing methods. Summarize the overall findings. A really great report will also give insight as to why the "best" model won out. This is where a thorough EDA will always help.

Record predictive performance from objective 1

Another competing model (interaction terms)

Another competing model (just continuous) LDA

nonparametric model (random forest)

another nonparametric model (conditional random forest)

Summary table of performance

Model	Predictors	Accuracy	95% CI	Sensitivity	Specificity	AUC
Logistic Regression (logit)	7	74.5%	(66.7%, 81.3%)	94.6%	13.5%	67.8%
Logistic Regression w/Interactions (logit)	7 + 3 interactions	75.2%	(67.4%, 81.9%)	92.9%	21.6%	72.2%
Random Forest*	10	74.5%	(66.7%, 81.3%)	99.1%	0.0%	74.2%
*RF Lower Cutoff (decreasing the probability from 50% to 30%)	10	73.8%	(66%, 80.7%)	91.1%	21.6%	
Conditional Random Forest	10	75.2%	(67.4%, 81.9%)	94.6%	16.2%	69.3%
LDA	4	73.2%	(65.3%, 80.1%)	94.6%	8.1%	60.2%

Conclusion/Discussion

```
to do:
```

```
take writeup from EDA and place here with reference to figures, etc
```

Appendix ==

*** *Appendix A:* EDA - Analysis ===========================

Data Set 1: Osteoporosis in Women

From Hosmer, Lemeshow, and Sturdivant (2013), Applied Logistic Regression, 3rd Edition. The Global Longitudinal Study of Osteoporosis in Women (GLOW) is an international study of osteoporosis in women aged 55 years and over. The major goals of the study are to examine prevention and treatment of fractures and distribution of risk factors among older women. Complete details on the study as well as a list of GLOW publications may be found at the Center for Outcomes Research web site, http://www.outcomes-umassmed. org/glow. There are over 60K observations in the original data set. This data set contains a sample of 500 of them. The link below is to a website with the data set and description of the variables. The data set in question is called "glow500".

https://www.umass.edu/statdata/statdata/data/glow/index.html Note: If you choose this data set, you MAY NOT use the Hosmer, Lemeshow, and Sturdivant text to help you in your analysis. You may only use Chapter 1 in order to obtain a description of the data.

Of course if you dont have the book

https://www.umass.edu/statdata/statdata/data/glow/glow.pdf provides definitions to the variables.

The Global Longitudinal Study of Osteoporosis in Women (GLOW) (2005-2014) was a prospective cohort study of physician practices in the provision of prophylaxis and treatment against osteoporotic fractures. The goal of this research was to improve understanding of the risk and prevention of osteoporosis-related fractures among female residents of 10 countries who were 55 years of age and older. GLOW enrolled over 60,000 women through over 700 physicians in 10 countries, and conducted annual follow-up for up to 5 years through annual patient questionnaires.

Setup:

Data Import and Cleaning

Missing values were not detected in dataset. Special characters were removed from column headings. What we know/don't know about the sample (500): 1. We do not know if the subjects are distributed equally around the world. We will assume that the same percentage from each region was selected for the sample in this dataset. 2. Based on the Sub_ID(Subject ID), we can assume that the datat is independent sample of participants.

```
glow_data_file <- here::here("data", "glow500.csv")
dataset_loc <-
dataset <- read.csv(glow_data_file, sep=",", stringsAsFactors = TRUE, header=TRUE,na.strings=c(""))

# List rows of data that have missing values
Missing_values <- dataset[!complete.cases(dataset),]

# Create new dataset without missing data
dataset <- na.omit(dataset)

#remove FRACSCORE feature per professor Turner
drops <- c("FRACSCORE")
dataset <- dataset[ , !(names(dataset) %in% drops)]
```

```r
#Cleanup column names
colnames(dataset)[colnames(dataset)=="ï..SUB_ID"] <- "SUB_ID"

#set categorical variables as factors
dataset$PRIORFRAC <- factor(dataset$PRIORFRAC,labels=c("0","1"))
dataset$PREMENO <- factor(dataset$PREMENO,labels=c("0","1"))
dataset$MOMFRAC <- factor(dataset$MOMFRAC,labels=c("0","1"))
dataset$ARMASSIST <- factor(dataset$ARMASSIST,labels=c("0","1"))
dataset$SMOKE <- factor(dataset$SMOKE,labels=c("0","1"))
dataset$RATERISK <- factor(dataset$RATERISK,labels=c("1","2","3"))
dataset$FRACTURE <- factor(dataset$FRACTURE,labels=c("0","1"))

#rearrange columns
dataset <- dataset[c("SUB_ID","SITE_ID","PHY_ID","AGE","BMI","HEIGHT","WEIGHT","PRIORFRAC","PREMENO","MO

str(dataset)
```

```
## 'data.frame':    500 obs. of  14 variables:
##  $ SUB_ID   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SITE_ID  : int  1 4 6 6 1 5 5 1 1 4 ...
##  $ PHY_ID   : int  14 284 305 309 37 299 302 36 8 282 ...
##  $ AGE      : int  62 65 88 82 61 67 84 82 86 58 ...
##  $ BMI      : num  28.2 34 20.6 24.3 29.4 ...
##  $ HEIGHT   : int  158 160 157 160 152 161 150 153 156 166 ...
##  $ WEIGHT   : num  70.3 87.1 50.8 62.1 68 68 50.8 40.8 62.6 63.5 ...
##  $ PRIORFRAC: Factor w/ 2 levels "0","1": 1 1 2 1 1 2 1 2 2 1 ...
##  $ PREMENO  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MOMFRAC  : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ...
##  $ ARMASSIST: Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ...
##  $ SMOKE    : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ RATERISK : Factor w/ 3 levels "1","2","3": 2 2 1 1 2 2 1 2 2 1 ...
##  $ FRACTURE : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Exploratory Data Analysis

Grouping Variables as Continuous, Categorical, and ID

```r
numericVar <- dataset[,4:7]
ID_var <- dataset[,c(1:3)]
set_noID <- dataset[4:14]
categoricalVar <- dataset[8:14]
```

Create Train and Validation Datasets

```r
validation_index = createDataPartition(dataset$FRACTURE, p=0.70, list=FALSE)
validationData = dataset[-validation_index,c(4:14)]
trainingData = dataset[validation_index,c(4:14)]
```

Summary Statistics

Assumptions This is a prospective study which means its a study over time of a group of similar individuals who differ with respect to certain factors under a study and how these factors affect rates of a certain outcome (Fracture vs No-Fracture) Linearity - Independence of errors - Based on SUB_ID(Subject ID) we confirm each record is an independent sample. Multicollinearity - Weight and BMI are highly correlated but we will remove one from the

```r
#Summary stats by groups for continous predictors
t(aggregate(AGE~FRACTURE,data=dataset,summary))
```

```
##              [,1]        [,2]
## FRACTURE     "0"         "1"
## AGE.Min.     "55.00000"  "56.00000"
## AGE.1st Qu.  "60.00000"  "65.00000"
## AGE.Median   "66.00000"  "72.00000"
## AGE.Mean     "67.48533"  "71.79200"
## AGE.3rd Qu.  "74.00000"  "79.00000"
## AGE.Max.     "90.00000"  "89.00000"
```

```r
t(aggregate(BMI~FRACTURE,data=dataset,summary))
```

```
##              [,1]        [,2]
## FRACTURE     "0"         "1"
## BMI.Min.     "14.87637"  "17.04223"
## BMI.1st Qu.  "23.32087"  "23.04688"
## BMI.Median   "26.36709"  "26.43080"
## BMI.Mean     "27.50140"  "27.70793"
## BMI.3rd Qu.  "30.61756"  "31.09282"
## BMI.Max.     "49.08241"  "44.03628"
```

```r
t(aggregate(WEIGHT~FRACTURE,data=dataset,summary))
```

```
##                 [,1]          [,2]
## FRACTURE        "0"           "1"
## WEIGHT.Min.     " 39.90000"   " 45.80000"
## WEIGHT.1st Qu.  " 60.30000"   " 59.90000"
## WEIGHT.Median   " 68.00000"   " 68.00000"
## WEIGHT.Mean     " 72.16693"   " 70.79200"
## WEIGHT.3rd Qu.  " 81.60000"   " 79.40000"
## WEIGHT.Max.     "127.00000"   "124.70000"
```

```r
t(aggregate(HEIGHT~FRACTURE,data=dataset,summary))
```

```
##                 [,1]        [,2]
## FRACTURE        "0"         "1"
## HEIGHT.Min.     "142.000"   "134.000"
## HEIGHT.1st Qu.  "158.000"   "155.000"
## HEIGHT.Median   "162.000"   "160.000"
## HEIGHT.Mean     "161.864"   "159.864"
## HEIGHT.3rd Qu.  "166.000"   "164.000"
## HEIGHT.Max.     "199.000"   "178.000"
```

```
#par(mfrow=c(2,2)) # put four figures in a row (2*4)
for (i in 4:7) {
  boxplot(dataset[,i] ~ dataset$FRACTURE,ylab=names(dataset)[i],xlab="FRACTURE", main="Summary for Conti
}
```
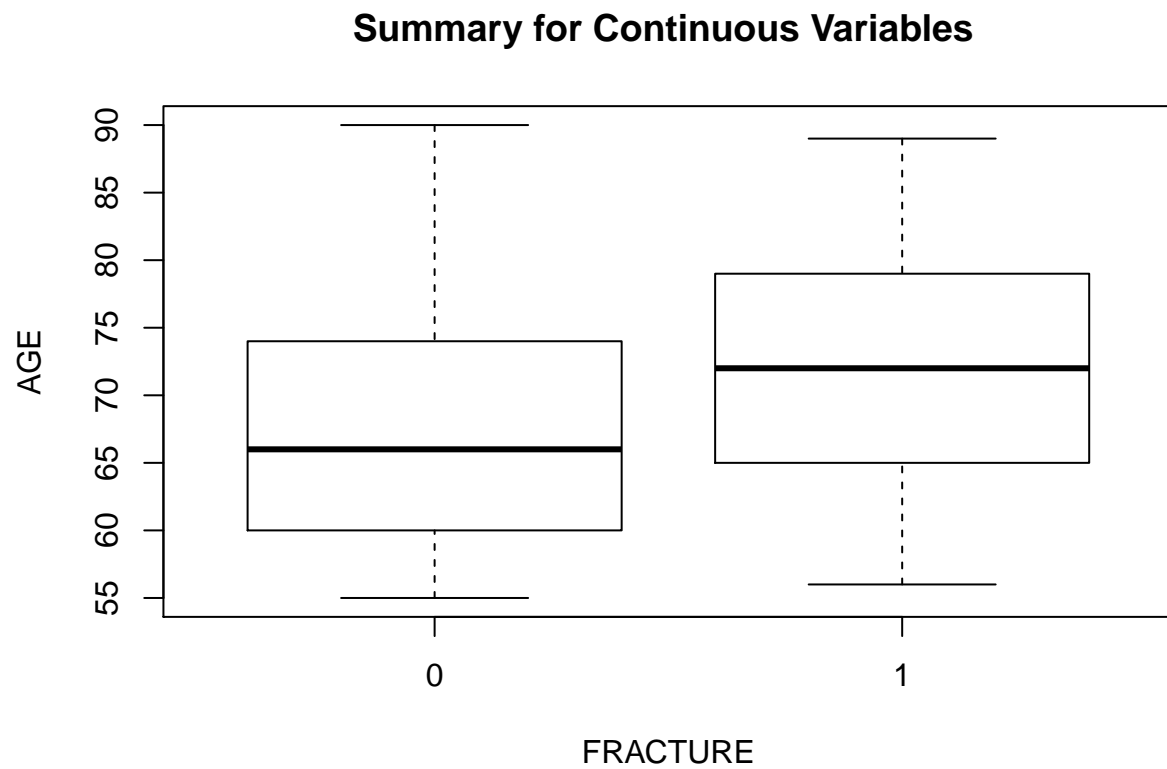
Summary for Continuous Variables

Summary for Continuous Variables
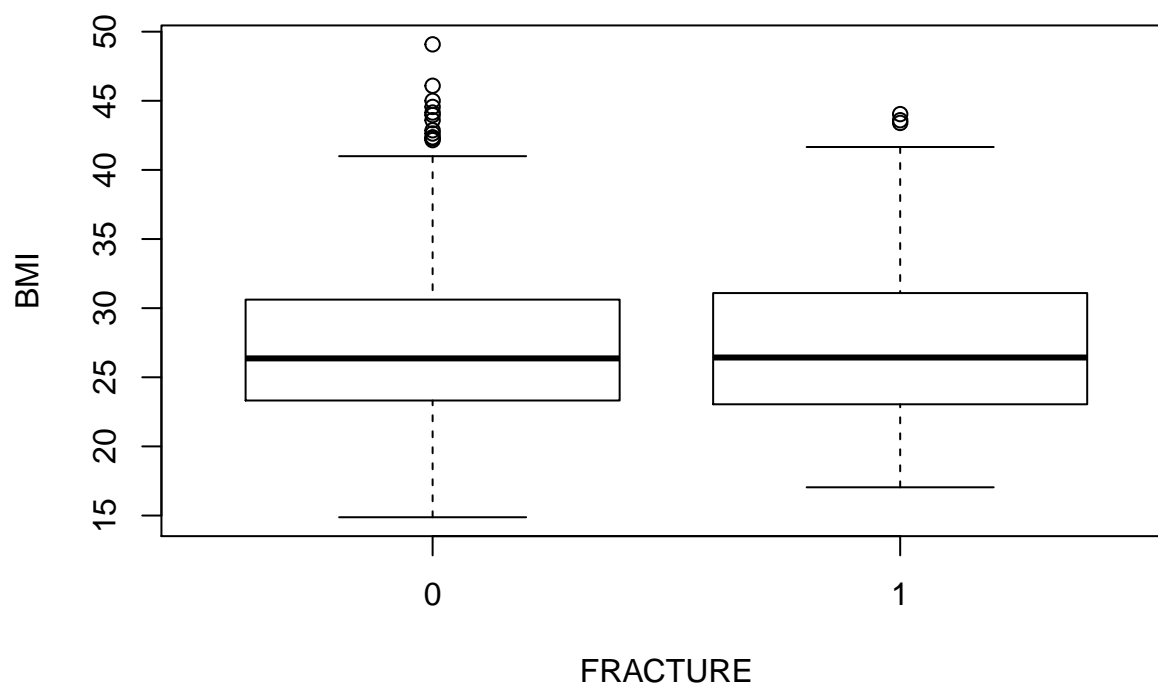
Summary for Continuous Variables
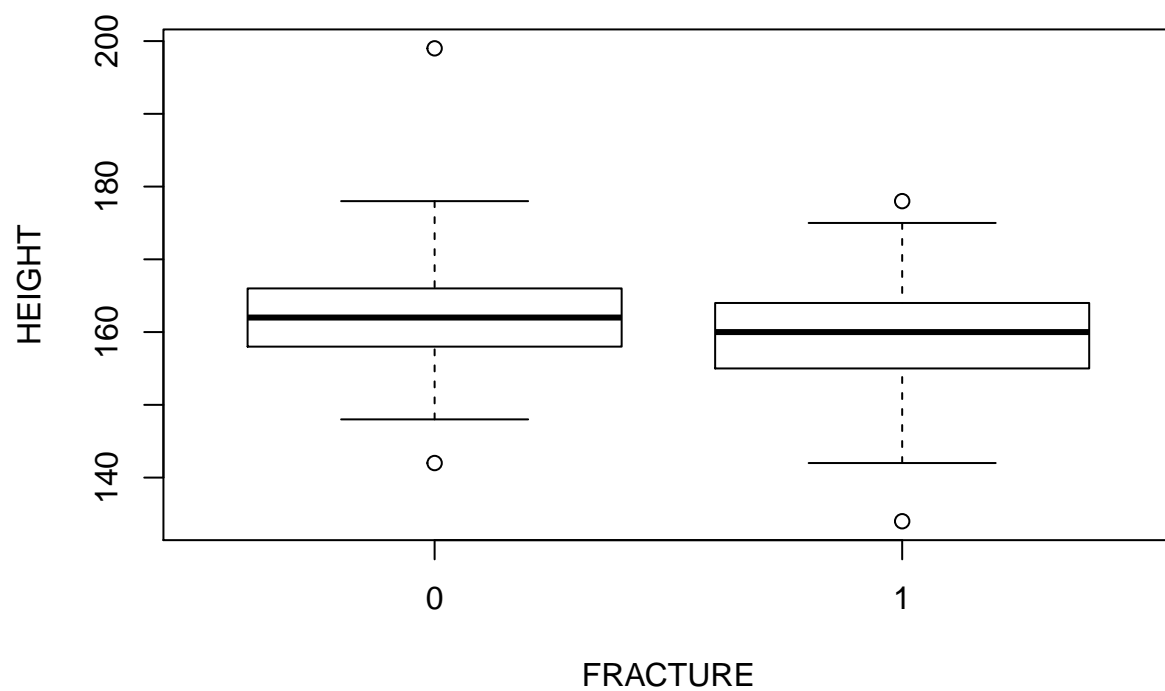
Summary for Continuous Variables
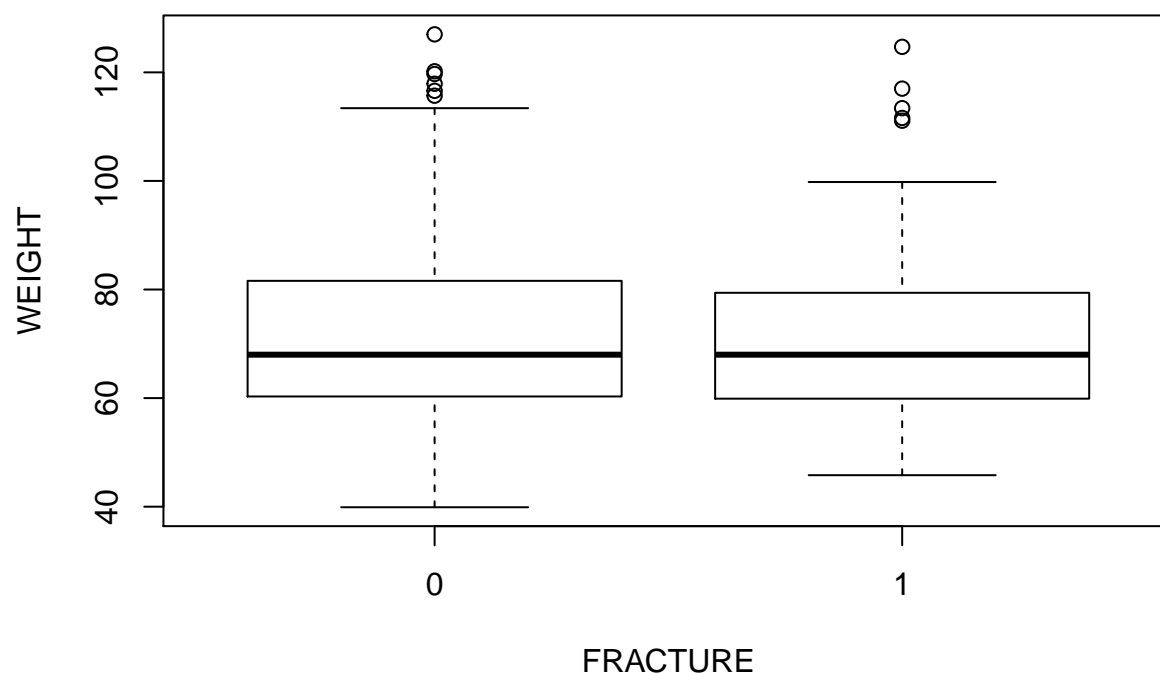

```r
#create an nicer summary table
index<-which(sapply(dataset,is.numeric))
tab.cont<-c()
for (i in index){
  tab.cont<-rbind(tab.cont,summary(dataset[,i]))
}
rownames(tab.cont)<-names(dataset)[index]
View(tab.cont)
tab.cont
```

```
##               Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## SUB_ID    1.00000 125.75000 250.50000 250.50000 375.25000 500.00000
## SITE_ID   1.00000   2.00000   3.00000   3.43600   5.00000   6.00000
## PHY_ID    1.00000  57.75000 182.50000 178.55000 298.00000 325.00000
## AGE      55.00000  61.00000  67.00000  68.56200  76.00000  90.00000
## BMI      14.87637  23.26889  26.41898  27.55303  30.79205  49.08241
## HEIGHT  134.00000 157.00000 161.50000 161.36400 165.00000 199.00000
## WEIGHT   39.90000  59.90000  68.00000  71.82320  81.30000 127.00000
```

```r
# display the first 20 rows
print(head(dataset, n=20))
```

```
##    SUB_ID SITE_ID PHY_ID AGE      BMI HEIGHT WEIGHT PRIORFRAC PREMENO
## 1       1       1      1  14   62 28.16055    158   70.3         0       0
## 2       2       4    284  65 34.02344    160   87.1         0       0
```

```
## 3      3        6   305  88 20.60936    157    50.8        1        0
## 4      4        6   309  82 24.25781    160    62.1        0        0
## 5      5        1    37  61 29.43213    152    68.0        0        0
## 6      6        5   299  67 26.23356    161    68.0        1        0
## 7      7        5   302  84 22.57778    150    50.8        0        0
## 8      8        1    36  82 17.42919    153    40.8        1        0
## 9      9        1     8  86 25.72321    156    62.6        1        0
## 10    10        4   282  58 23.04398    166    63.5        0        0
## 11    11        6   315  67 28.87778    153    67.6        0        0
## 12    12        1    34  56 42.27473    167   117.9        0        0
## 13    13        6   315  59 25.56775    162    67.1        0        0
## 14    14        1    33  72 21.15702    165    57.6        0        0
## 15    15        1    23  64 23.90625    160    61.2        0        1
## 16    16        3   179  68 30.09143    161    78.0        0        0
## 17    17        4   284  67 38.82461    165   105.7        0        0
## 18    18        4   283  69 25.07240    162    65.8        0        0
## 19    19        3   179  78 31.09282    162    81.6        1        0
## 20    20        6   313  60 23.00296    157    56.7        0        0
##     MOMFRAC ARMASSIST SMOKE RATERISK FRACTURE
## 1         0         0     0        2        0
## 2         0         0     0        2        0
## 3         1         1     0        1        0
## 4         0         0     0        1        0
## 5         0         0     0        2        0
## 6         0         0     1        2        0
## 7         0         0     0        1        0
## 8         0         0     0        2        0
## 9         0         0     0        2        0
## 10        0         0     0        1        0
## 11        1         0     1        1        0
## 12        0         1     1        2        0
## 13        0         0     1        1        0
## 14        0         1     0        1        0
## 15        0         0     0        2        0
## 16        0         1     0        1        0
## 17        0         0     0        1        0
## 18        0         0     0        2        0
## 19        0         1     0        3        0
## 20        0         0     0        2        0
```

```r
# display the dimensions of the dataset
print(dim(dataset))
```

```
## [1] 500  14
```

```r
# list types for each attribute
print(sapply(dataset,class))
```

```
##     SUB_ID   SITE_ID    PHY_ID       AGE       BMI    HEIGHT    WEIGHT
## "integer" "integer" "integer" "integer" "numeric" "integer" "numeric"
## PRIORFRAC   PREMENO   MOMFRAC ARMASSIST     SMOKE  RATERISK  FRACTURE
##  "factor"  "factor"  "factor"  "factor"  "factor"  "factor"  "factor"
```

```
# Standard Deviations for the non-categorical columns
std=sapply(set_noID,sd)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling var(x)
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling var(x)
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling var(x)
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling var(x)
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling var(x)
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling var(x)
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): Calling var(x)
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.
```

```
print('The standard deviations are:')
```

```
## [1] "The standard deviations are:"
```

```
print(std)
```

```
##          AGE        BMI     HEIGHT     WEIGHT  PRIORFRAC     PREMENO
## 8.9895372  5.9739583  6.3554928 16.4359918  0.4345961  0.3958249
##      MOMFRAC   ARMASSIST      SMOKE   RATERISK   FRACTURE
## 0.3366402  0.4848651  0.2554025  0.7922470  0.4334464
```

Correlations

BMI and Weight show to be highly correlation which makes sense since weight is a factor in calculation of BMI. We will remove Weight from models in order to meet assumptions.

```
#Training dataset without ID columns, convert PRIORFRAC to numeric for corrplot
train_df <- trainingData[2:5]
train_df$PRIORFRAC <- as.numeric(train_df$PRIORFRAC)
corrplot(cor(train_df), method = "number", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```
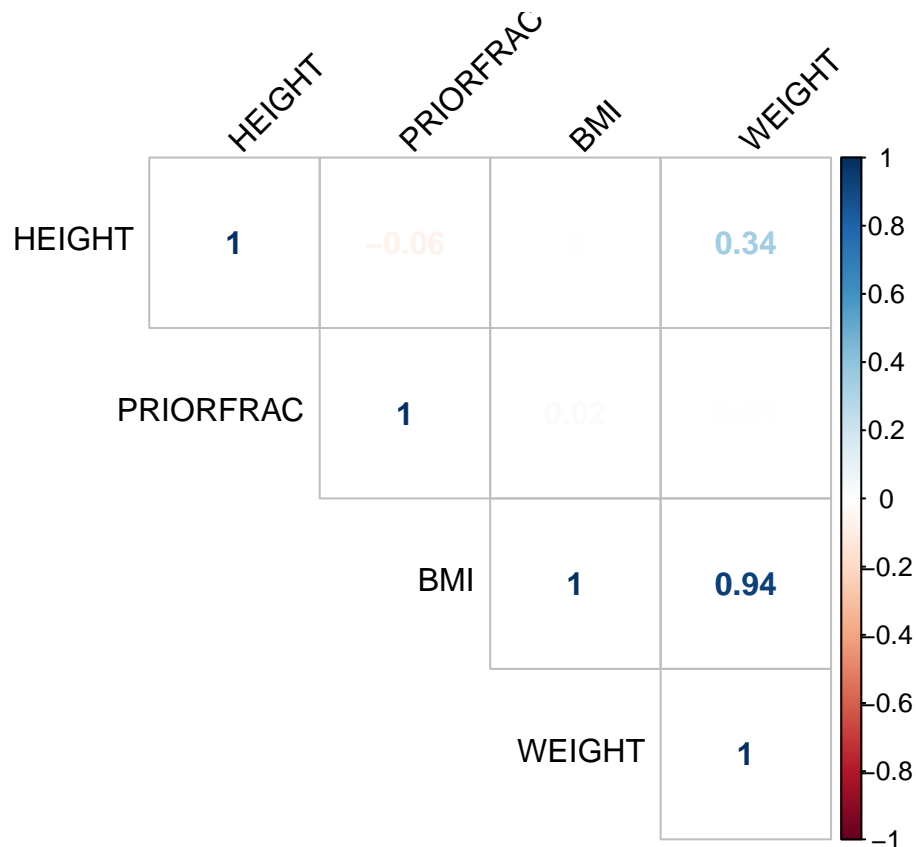
Visualization of Continuous Variables

For the categorical variables, we show an unbalanced dataset of subjects with majority false PRIORFRAC, PREMENO, MOMFRAC, ARMASSIST, and SMOKE. There was a good balance of subjects in the 3 levels of RATERISK. An unbalanced dataset will cause a model to favor the skewed numbers.

For the continous variables, we can see that BMI and Weight are highly correlated and weight and height are also correlated. When building the model, we will remove Weight as to meet the assumptions of logistic regression.

```r
# Data visualizations
dataset_numeric = numericVar
#Histograms
par(mfrow=c(2,2)) # put four figures in a row (2*4)
for (i in 1:4) {
  hist(dataset_numeric[,i],xlab=names(dataset_numeric)[i],main=names(dataset_numeric)[i])
}
```

AGE

BMI

HEIGHT

WEIGHT
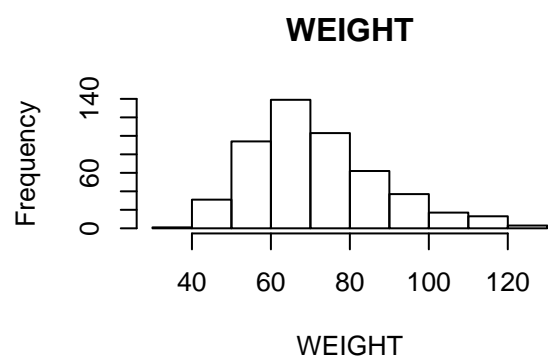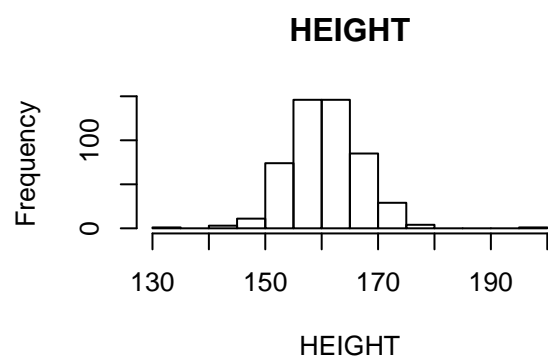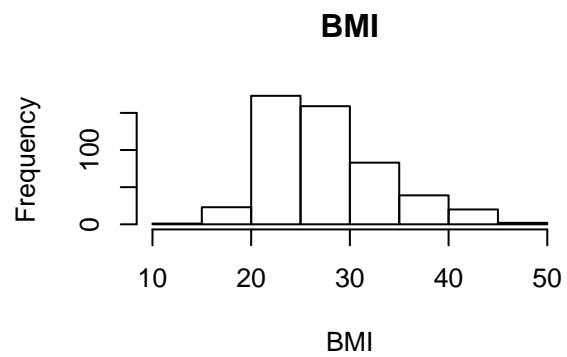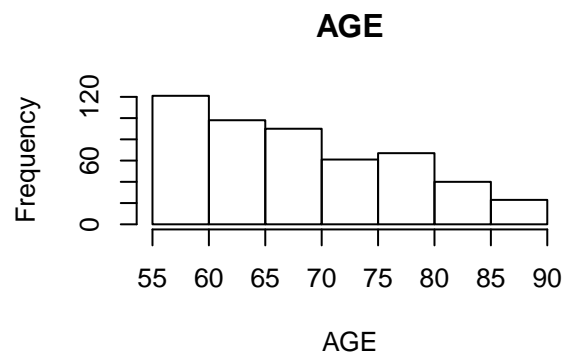
```r
#Density Plots
par(mfrow=c(2,2))
for(i in 1:4) {
  plot(density(dataset_numeric[,i]), xlab=names(dataset_numeric)[i], main=names(dataset_numeric)[i])
}
```

AGE

BMI
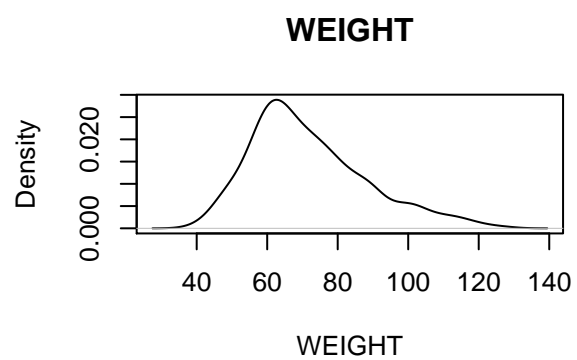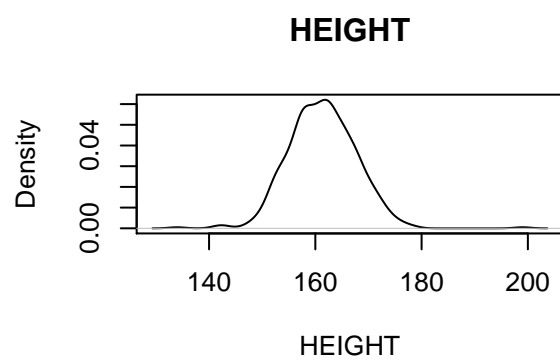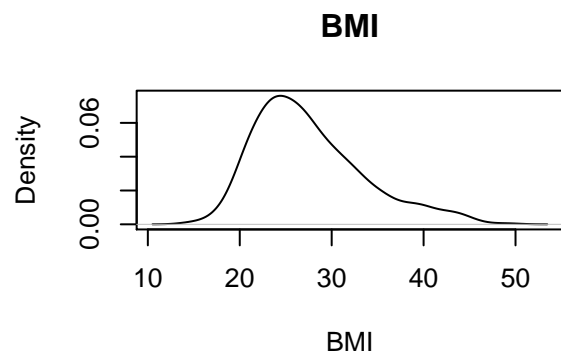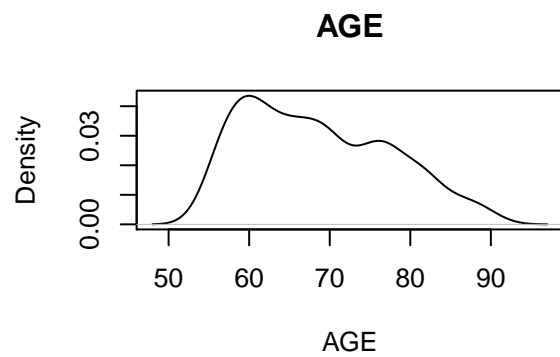
HEIGHT

WEIGHT


```r
#Box And Whisker Plots
par(mfrow=c(2,2))
for(i in 1:4) {
  boxplot(dataset_numeric[,i], xlab=names(dataset_numeric)[i], main=names(dataset_numeric)[i])
}
```

AGE

BMI

HEIGHT

WEIGHT

AGE

BMI
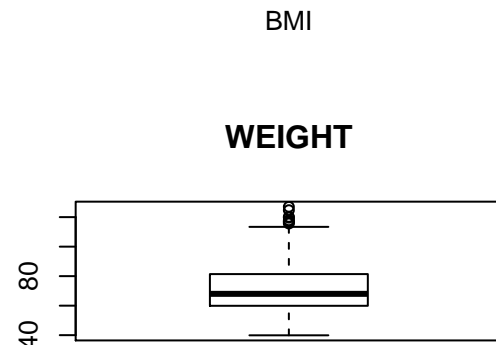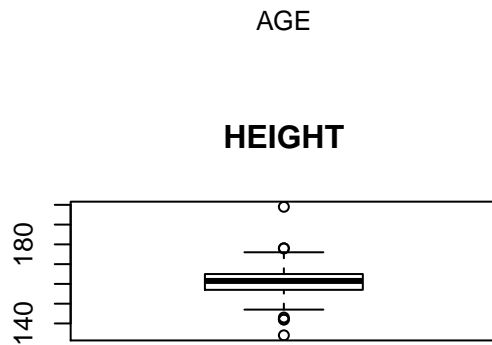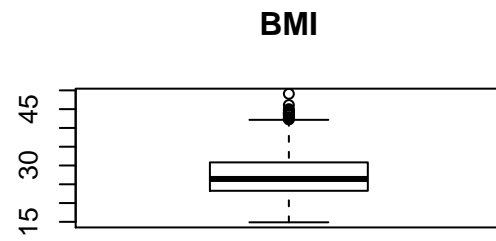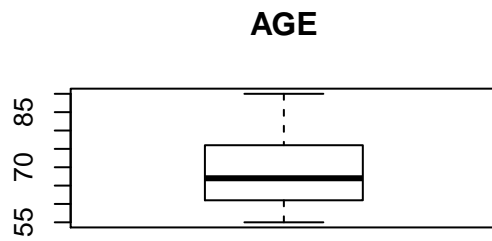
HEIGHT

WEIGHT

```
#Barplots, which is used to count the accurances for categorical attributes
dataset_categorical = set_noID[,-c(2:5,11)]

par(mfrow=c(3,2))
for(i in 1:6) {
  counts <- table(dataset_categorical[,i]) # get the count for each categorical value
  name <- names(dataset_categorical)[i]
  barplot(counts, main=name,col=c("lightblue","darkred"))
}
```

AGE

PREMENO

MOMFRAC

ARMASSIST

SMOKE

RATERISK
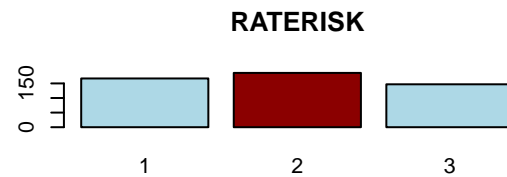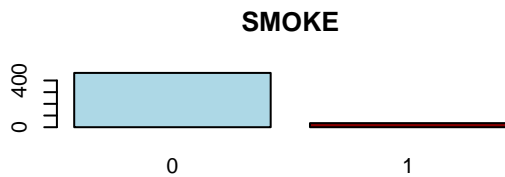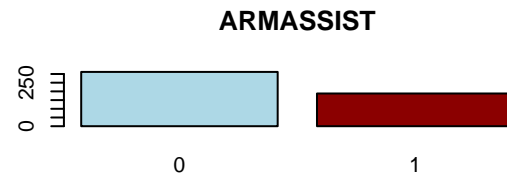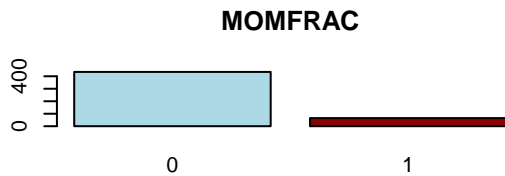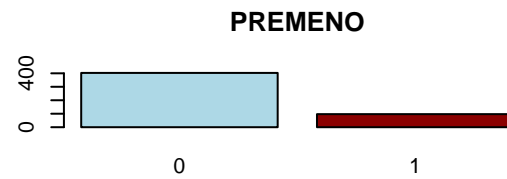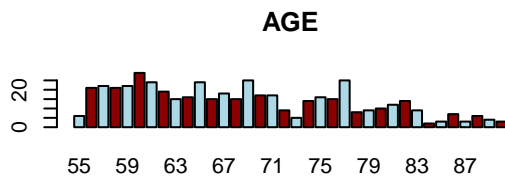
```r
#Multivariate Visualization
correlations1=cor(dataset_numeric)
print(correlations1)
```

```
##              AGE         BMI       HEIGHT      WEIGHT
## AGE     1.0000000 -0.22125651 -0.19264861 -0.2715964
## BMI    -0.2212565  1.00000000 -0.02437689  0.9373360
## HEIGHT -0.1926486 -0.02437689  1.00000000  0.3159691
## WEIGHT -0.2715964  0.93733603  0.31596915  1.0000000
```

```r
par(mfrow=c(1,1))
corrplot(correlations1, methods="circle")
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt
## = tl.srt, : "methods" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col
## = tl.col, : "methods" is not a graphical parameter
```

```
## Warning in title(title, ...): "methods" is not a graphical parameter
```
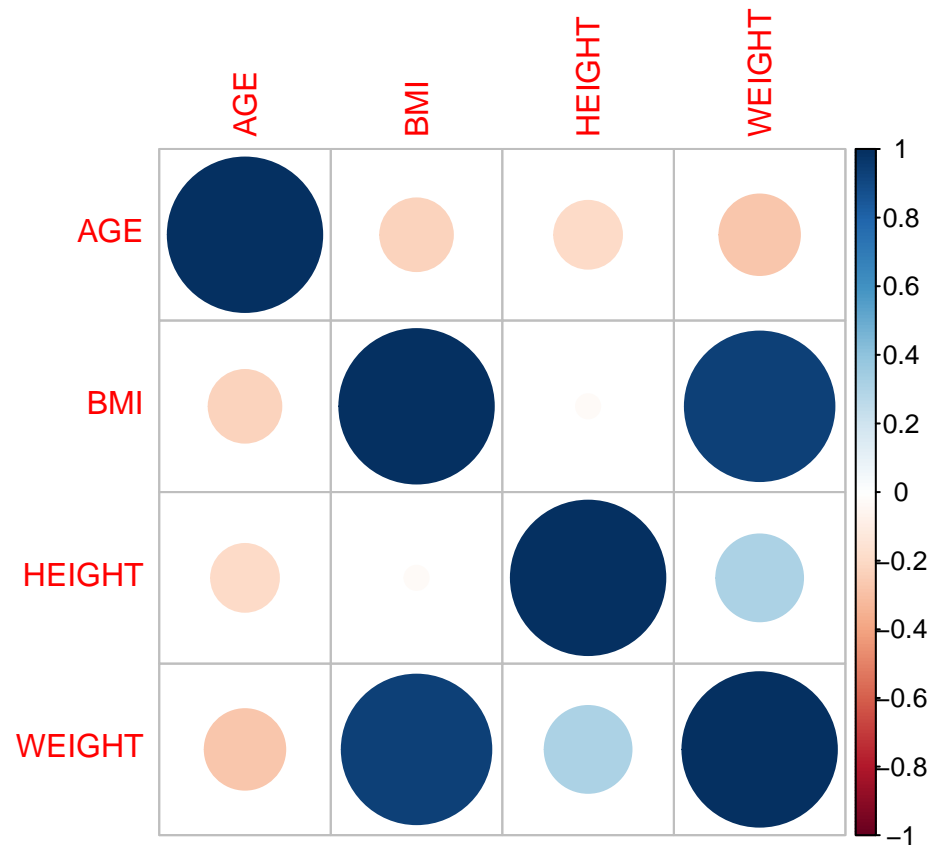
```
# pair-wise scatterplots of the numeric attributes
par(mfrow=c(1,1))
pairs(dataset_numeric)
```
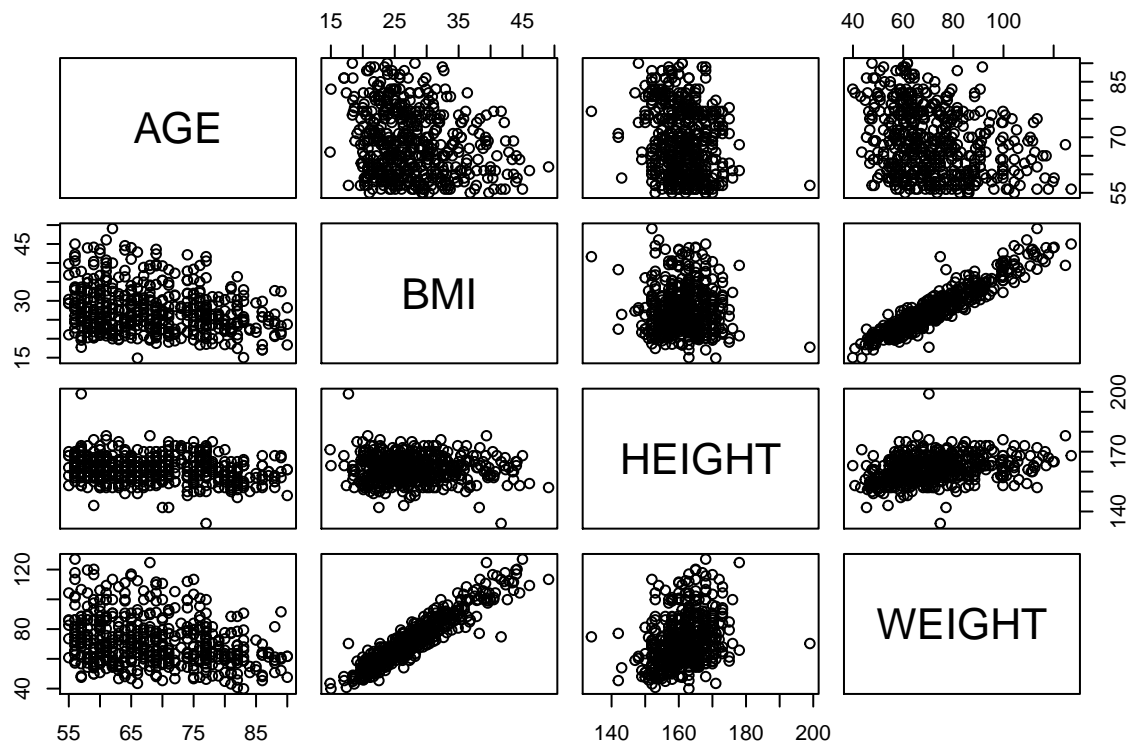
```r
#Scatterplot Matrix By Class (use different color to distinguish different class)
par(mfrow=c(1,1))
pairs(dataset_numeric, col=dataset[,5])
```
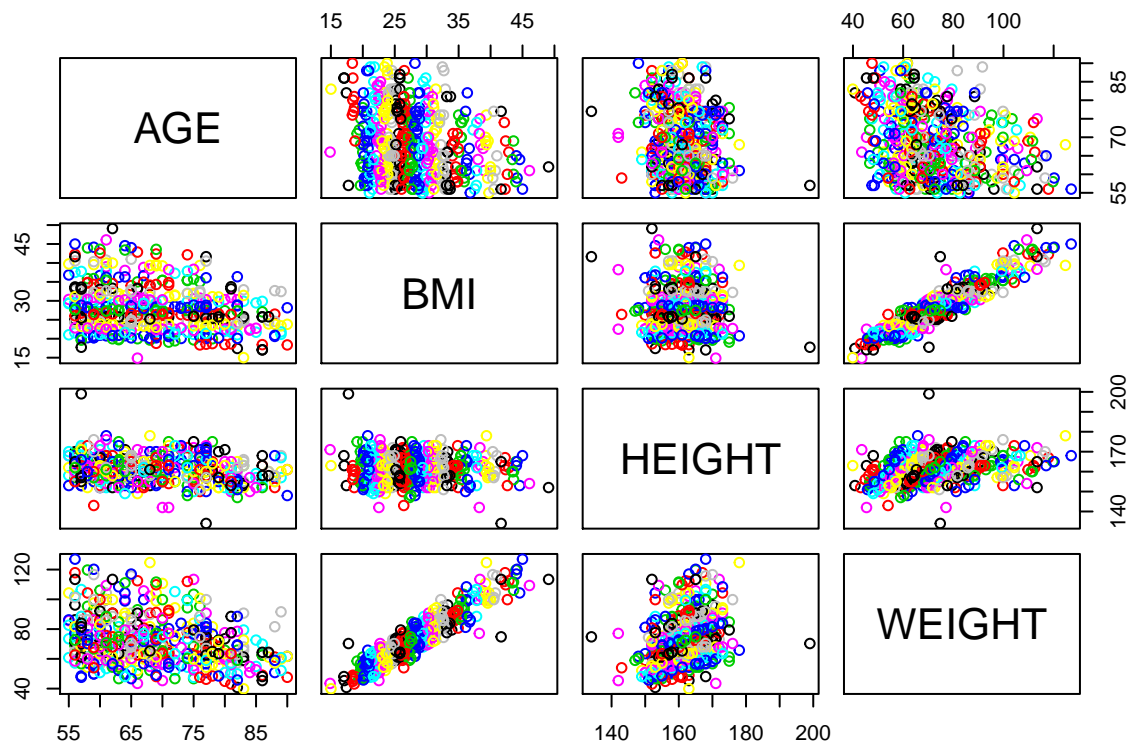
```
# density plots for each attribute by class value
X <- set_noID[2:5]
Y <- set_noID$FRACTURE
X$PRIORFRAC <- as.numeric(X$PRIORFRAC)
scales <- list(x=list(relation="free"), y=list(relation="free"))
par(mfrow=c(1,1))
featurePlot(x=X, y=set_noID$FRACTURE, plot="density", scales=scales)
```
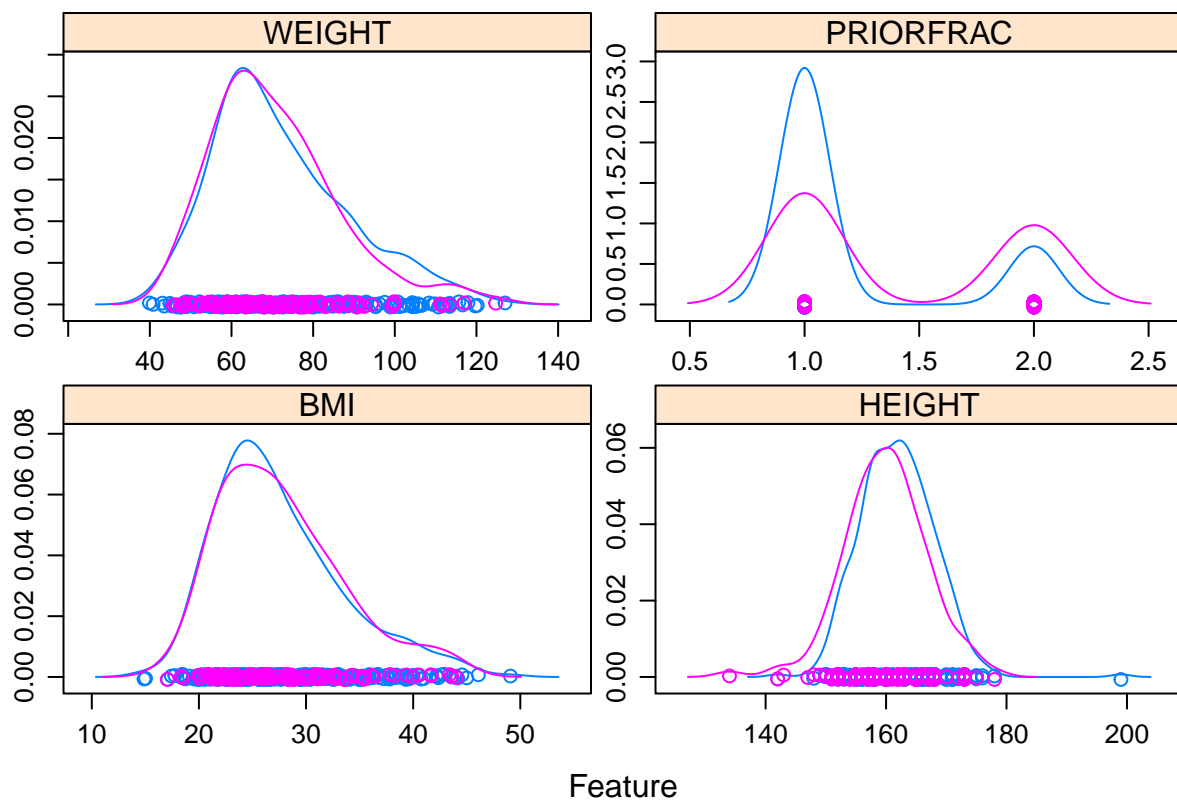
```
#Box And Whisker Plots By Class
par(mfrow=c(1,1))
featurePlot(x=X, y=set_noID$FRACTURE, plot="box")
```
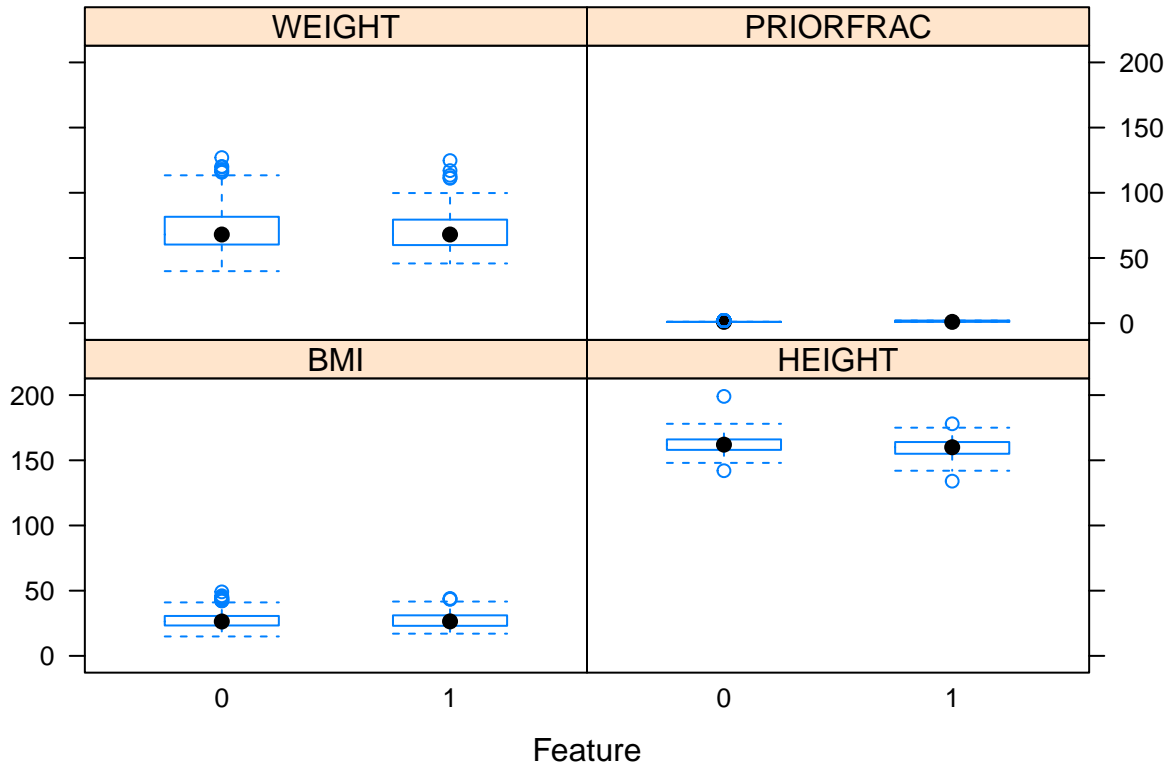
Checking the Balance of the Full dataset

The current sample dataset containes a larger propotion of subjects that did not develop fracture. Building a model against this dataset could produce bias towards the majority class. Below you will see how many subjects with(1)/without(0) Fractures as well as the proportion percentage for each. After splitting the dataset into training and validation(test) sets, we noticed the proportion of the training and test was not any better.

We fit a logistic model on the unbalanced training dataset with a threshold of .05. It shows a Precision of 1 which says there are no false positives. Recall equals 0.20 is low and indicates that we have higher number of false negatives. The F equals 0.20 is also low and suggests weak accuracy of this model.

We also plotted a ROC curve to visualize the model. The AUC equals 0.764 which is low and shows the data is not balanced.

We will attempt to balance the dataset in order to create a more balanced distribution of and a better prediction.

```
table(dataset$FRACTURE)
```

```
##
##   0   1
## 375 125
```

```
prop.table(table(dataset$FRACTURE))
```

```
##
##    0    1
## 0.75 0.25
```

```r
# split the data into training and validation sets
set.seed(84)
validation_index = createDataPartition(dataset$FRACTURE, p=0.75, list=FALSE)
validationData = dataset[-validation_index,c(4:14)]
trainingData = dataset[validation_index,c(4:14)]
prop.table(table(validationData$FRACTURE))
```

```
##
##    0    1
## 0.75 0.25
```

```r
prop.table(table(trainingData$FRACTURE))
```

```
##
##    0    1
## 0.75 0.25
```

```r
#fit a logistic regressio to unblanced training set
fit.dataset <- glm(formula=FRACTURE~ ., data = trainingData, family="binomial")
pred.fit.dataset <- predict(fit.dataset, newdata = validationData, type="response")
#Check Accuracy of fitted model.
accuracy.meas(validationData$FRACTURE,pred.fit.dataset, threshold=.05)
```

```
##
## Call:
## accuracy.meas(response = validationData$FRACTURE, predicted = pred.fit.dataset,
##      threshold = 0.05)
##
## Examples are labelled as positive when predicted is greater than 0.05
##
## precision: 0.250
## recall: 1.000
## F: 0.200
```

```r
#Check Accuracy of Test dataset using ROC curve
roc.curve(validationData$FRACTURE, pred.fit.dataset, plotit = TRUE)
```

ROC curve
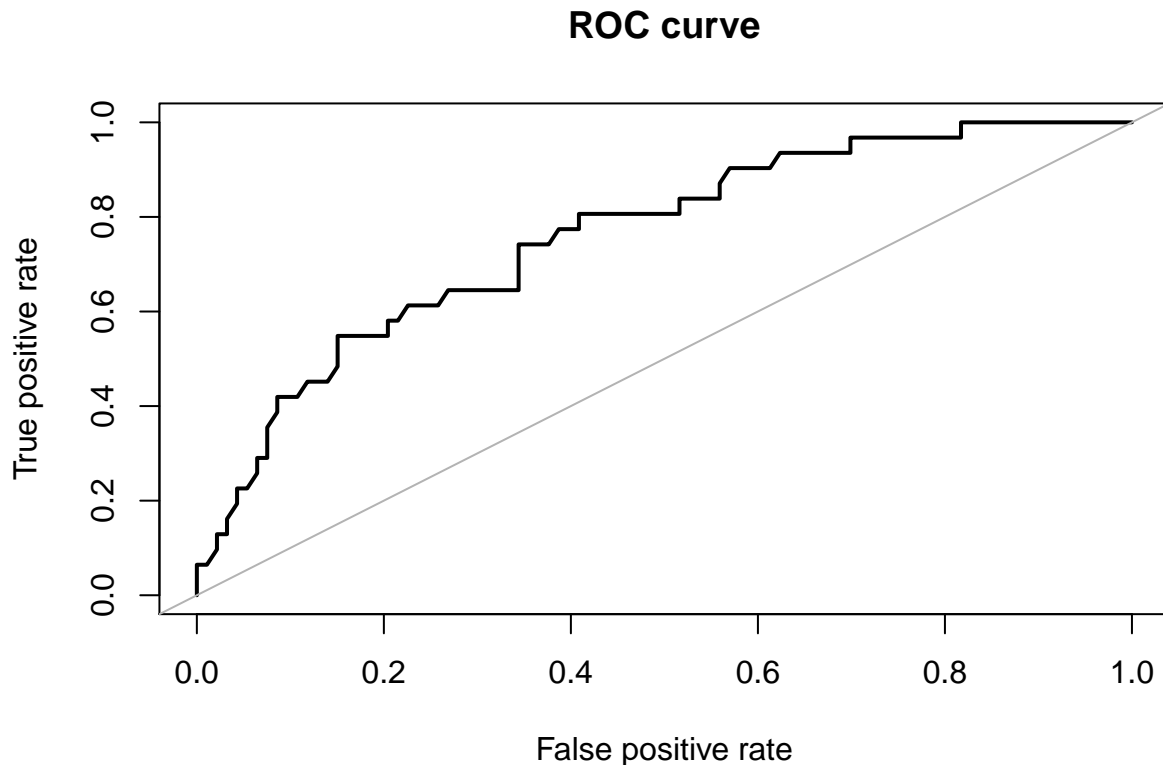

```
## Area under the curve (AUC): 0.760
```

##Create a vector of all categorical variables and run frequency 2X2s with Mosaic plots.

Chi-Square Test For the 2-way tables the chisq test independence will show if 2 categorical variables are related in some population. Null Hypothesis: The two categorical variables are independent. Alternative Hypothesis: The two categorical variables are dependent

Variable: PRIORFRAC 41% of subjects with Prior Franctures also had current Fractures but only make up 25% of the overall subjects in the sample that had prior fractures. The Chi-squared p-value favors overwhemingly the alternative hypothesis that the PRIORFRAC variable is dependent on Fracture variable.

Variable: PREMENO 80% of the sample subjects are not in Pre-Menopausehad of which 24% had fractures. The same frequency of 25% Premenopausal women had fractures. The Chi-squared p-value favors the null hypothesis that the PREMENO variable is independent on Fracture variable.

Variable: MOMFRAC 13% of subjects have Mothers with a history of fractures. Out of those 13%, 36% of subjects also had fractures. The Chi-squared p-value favors the alternative hypothesis that the MOMFRAC variable is probably dependent on Fracture variable.

Variable: ARMASSIST 62% (312/500) subjects do not have Armassist of which 20% had fractures. Of those with Armassist, 33% had fractures. The Chi-squared p-value favors the alternative hypothesis that the ARMASSIST variable is most likely dependent on Fracture variable.

Variable: SMOKE In the dataset, 93% of subjects are non-smokers of which 26% had fractures. 7% of the subjects who were smokers of which 26% had no fractures. Although the subjects are not balance in smoker vs non-smoker category, the p-value for Chi-squared test shows .47 we favor the alternative hypothesis that the Smoke variable is dependent on the Fracture.

Variable: RATERISK Raterisk shows the frequency of subjects in each Raterisk level is between 29%-33%. This is pretty even in terms of how many subjects are within each Raterisk. For those that did have Fractures, their probability of a fracture increased with the level of Raterisk. This makes sense.

```
categoricalVarVec  <- c("PRIORFRAC","PREMENO","MOMFRAC","ARMASSIST","SMOKE","RATERISK")
for(categoricalVar in categoricalVarVec){
  CrossTable(dataset[,categoricalVar], dataset$FRACTURE, chisq = TRUE , expected = TRUE, dnn=c(categori
  mosaicplot(CrossTable(dataset[ ,categoricalVar], dataset$FRACTURE)$t, main=paste("FRACTURE vs",catego
}
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##              | FRACTURE
##    PRIORFRAC |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##           0 |       301 |        73 |       374 |
##             |   280.500 |    93.500 |           |
##             |     1.498 |     4.495 |           |
##             |     0.805 |     0.195 |     0.748 |
##             |     0.803 |     0.584 |           |
##             |     0.602 |     0.146 |           |
## -------------|-----------|-----------|-----------|
##           1 |        74 |        52 |       126 |
##             |    94.500 |    31.500 |           |
##             |     4.447 |    13.341 |           |
##             |     0.587 |     0.413 |     0.252 |
##             |     0.197 |     0.416 |           |
##             |     0.148 |     0.104 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       375 |       125 |       500 |
##             |     0.750 |     0.250 |           |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  23.78123      d.f. =  1      p =  1.079299e-06
```

31

```
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  22.63532     d.f. =  1     p =  1.958512e-06
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:   500
##
##
##                          | dataset$FRACTURE
## dataset[, categoricalVar] |         0 |         1 | Row Total |
## -------------------------|-----------|-----------|-----------|
##                        0 |       301 |        73 |       374 |
##                          |     1.498 |     4.495 |           |
##                          |     0.805 |     0.195 |     0.748 |
##                          |     0.803 |     0.584 |           |
##                          |     0.602 |     0.146 |           |
## -------------------------|-----------|-----------|-----------|
##                        1 |        74 |        52 |       126 |
##                          |     4.447 |    13.341 |           |
##                          |     0.587 |     0.413 |     0.252 |
##                          |     0.197 |     0.416 |           |
##                          |     0.148 |     0.104 |           |
## -------------------------|-----------|-----------|-----------|
##             Column Total |       375 |       125 |       500 |
##                          |     0.750 |     0.250 |           |
## -------------------------|-----------|-----------|-----------|
##
##
```

FRACTURE vs PRIORFRAC
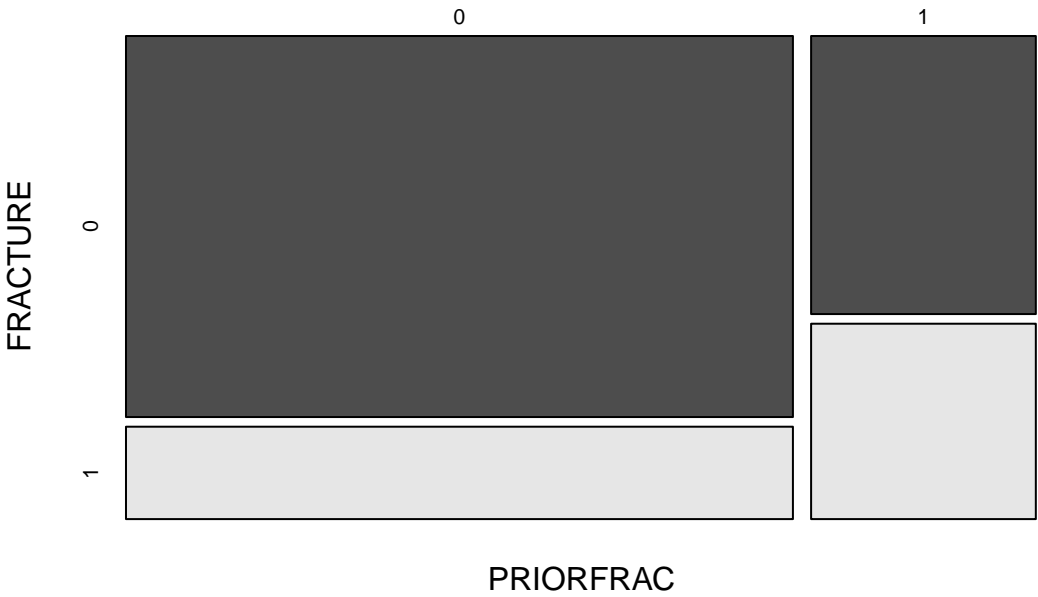

```
##
##
##      Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##             | FRACTURE
##     PREMENO |         0 |         1 | Row Total |
## ------------|-----------|-----------|-----------|
##           0 |       303 |       100 |       403 |
##             |   302.250 |   100.750 |           |
##             |     0.002 |     0.006 |           |
##             |     0.752 |     0.248 |     0.806 |
##             |     0.808 |     0.800 |           |
##             |     0.606 |     0.200 |           |
## ------------|-----------|-----------|-----------|
```

```
##              1 |        72 |        25 |        97 |
##                | 72.750 |   24.250 |           |
##                |   0.008 |    0.023 |           |
##                |   0.742 |    0.258 |    0.194 |
##                |   0.192 |    0.200 |           |
##                |   0.144 |    0.050 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       375 |       125 |       500 |
##                |   0.750 |    0.250 |           |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  0.038372      d.f. =  1      p =  0.844698
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  0.004263556      d.f. =  1      p = 0.9479384
##
##
##
##
##     Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |             N / Row Total |
## |             N / Col Total |
## |           N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:   500
##
##
##                               | dataset$FRACTURE
## dataset[, categoricalVar] |         0 |         1 | Row Total |
## -------------------------|-----------|-----------|-----------|
##                        0 |       303 |       100 |       403 |
##                          |     0.002 |     0.006 |           |
##                          |     0.752 |     0.248 |     0.806 |
##                          |     0.808 |     0.800 |           |
##                          |     0.606 |     0.200 |           |
## -------------------------|-----------|-----------|-----------|
##                        1 |        72 |        25 |        97 |
##                          |     0.008 |     0.023 |           |
##                          |     0.742 |     0.258 |     0.194 |
##                          |     0.192 |     0.200 |           |
##                          |     0.144 |     0.050 |           |
## -------------------------|-----------|-----------|-----------|
```

```
##               Column Total |       375 |       125 |       500 |
##                            |     0.750 |     0.250 |           |
## ---------------------------|-----------|-----------|-----------|
##
##
```

FRACTURE vs PREMENO

PREMENO

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##            | FRACTURE
##    MOMFRAC |        0 |        1 | Row Total |
## -----------|----------|----------|-----------|
##         0 |      334 |      101 |       435 |
```

```
##                    |    326.250 |    108.750 |            |
##                    |      0.184 |      0.552 |            |
##                    |      0.768 |      0.232 |      0.870 |
##                    |      0.891 |      0.808 |            |
##                    |      0.668 |      0.202 |            |
## -------------|-----------|-----------|-----------|
##            1 |        41 |        24 |        65 |
##                    |     48.750 |     16.250 |            |
##                    |      1.232 |      3.696 |            |
##                    |      0.631 |      0.369 |      0.130 |
##                    |      0.109 |      0.192 |            |
##                    |      0.082 |      0.048 |            |
## -------------|-----------|-----------|-----------|
## Column Total |       375 |       125 |       500 |
##                    |      0.750 |      0.250 |            |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  5.664604     d.f. =  1      p =  0.01731063
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  4.957265     d.f. =  1      p =  0.02598127
##
##
##
##
##     Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##                          | dataset$FRACTURE
## dataset[, categoricalVar] |         0 |         1 | Row Total |
## -------------------------|-----------|-----------|-----------|
##                        0 |       334 |       101 |       435 |
##                          |      0.184 |      0.552 |            |
##                          |      0.768 |      0.232 |      0.870 |
##                          |      0.891 |      0.808 |            |
##                          |      0.668 |      0.202 |            |
## -------------------------|-----------|-----------|-----------|
```
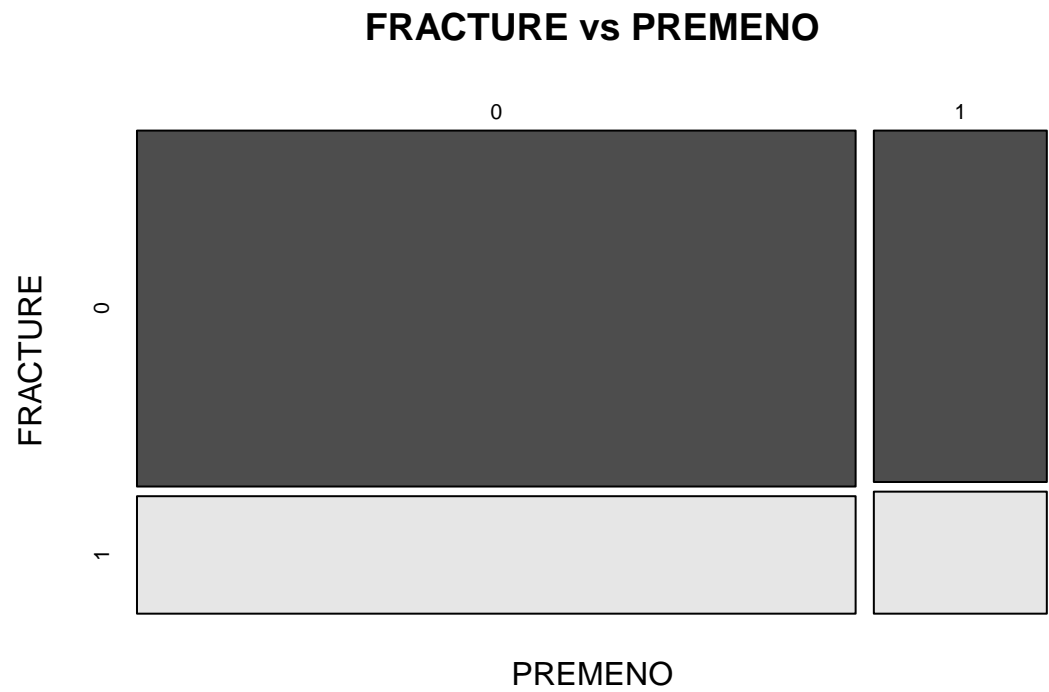
36

```
##                           1 |          41 |          24 |          65 |
##                             |       1.232 |       3.696 |             |
##                             |       0.631 |       0.369 |       0.130 |
##                             |       0.109 |       0.192 |             |
##                             |       0.082 |       0.048 |             |
## -------------------------|-----------|-----------|-----------|
##              Column Total |         375 |         125 |         500 |
##                             |       0.750 |       0.250 |             |
## -------------------------|-----------|-----------|-----------|
##
##
```

FRACTURE vs MOMFRAC


```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |            N / Row Total |
## |            N / Col Total |
## |          N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
```

```
##
##
##              | FRACTURE
##     ARMASSIST |          0 |          1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |        250 |         62 |        312 |
##              |    234.000 |     78.000 |            |
##              |      1.094 |      3.282 |            |
##              |      0.801 |      0.199 |      0.624 |
##              |      0.667 |      0.496 |            |
##              |      0.500 |      0.124 |            |
## -------------|-----------|-----------|-----------|
##            1 |        125 |         63 |        188 |
##              |    141.000 |     47.000 |            |
##              |      1.816 |      5.447 |            |
##              |      0.665 |      0.335 |      0.376 |
##              |      0.333 |      0.504 |            |
##              |      0.250 |      0.126 |            |
## -------------|-----------|-----------|-----------|
## Column Total |        375 |        125 |        500 |
##              |      0.750 |      0.250 |            |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  11.63848     d.f. =  1      p =  0.0006460138
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  10.92244     d.f. =  1      p =  0.0009500637
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##                              | dataset$FRACTURE
## dataset[, categoricalVar] |          0 |          1 | Row Total |
## -------------------------|-----------|-----------|-----------|
```

```
##                         0 |        250 |         62 |        312 |
##                           |      1.094 |      3.282 |            |
##                           |      0.801 |      0.199 |      0.624 |
##                           |      0.667 |      0.496 |            |
##                           |      0.500 |      0.124 |            |
## -------------------------|-----------|-----------|-----------|
##                         1 |        125 |         63 |        188 |
##                           |      1.816 |      5.447 |            |
##                           |      0.665 |      0.335 |      0.376 |
##                           |      0.333 |      0.504 |            |
##                           |      0.250 |      0.126 |            |
## -------------------------|-----------|-----------|-----------|
##              Column Total |        375 |        125 |        500 |
##                           |      0.750 |      0.250 |            |
## -------------------------|-----------|-----------|-----------|
##
##
```

FRACTURE vs ARMASSIST
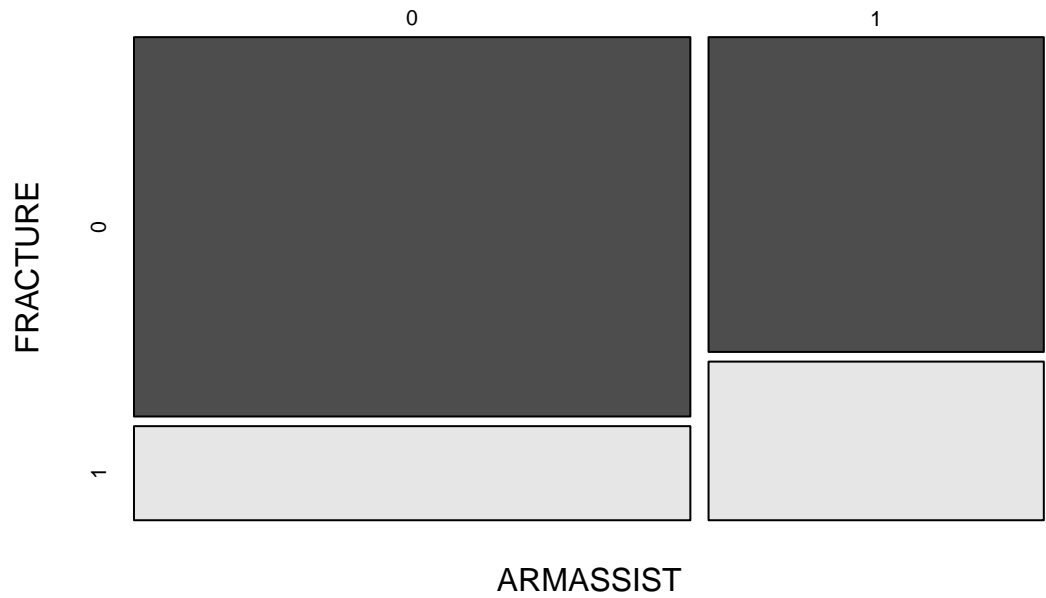

```
##
##
##     Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |          N / Row Total  |
```

39

```
## |             N / Col Total |
## |           N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
## | FRACTURE
## SMOKE |           0 |           1 | Row Total |
## -------------|-----------|-----------|-----------|
## 0 |         347 |         118 |         465 |
## |     348.750 |     116.250 |             |
## |       0.009 |       0.026 |             |
## |       0.746 |       0.254 |       0.930 |
## |       0.925 |       0.944 |             |
## |       0.694 |       0.236 |             |
## -------------|-----------|-----------|-----------|
## 1 |          28 |           7 |          35 |
## |      26.250 |       8.750 |             |
## |       0.117 |       0.350 |             |
## |       0.800 |       0.200 |       0.070 |
## |       0.075 |       0.056 |             |
## |       0.056 |       0.014 |             |
## -------------|-----------|-----------|-----------|
## Column Total |         375 |         125 |         500 |
## |       0.750 |       0.250 |             |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  0.5017921    d.f. =  1     p =  0.4787137
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  0.2560164    d.f. =  1     p =  0.6128703
##
##
##
##
## Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
```
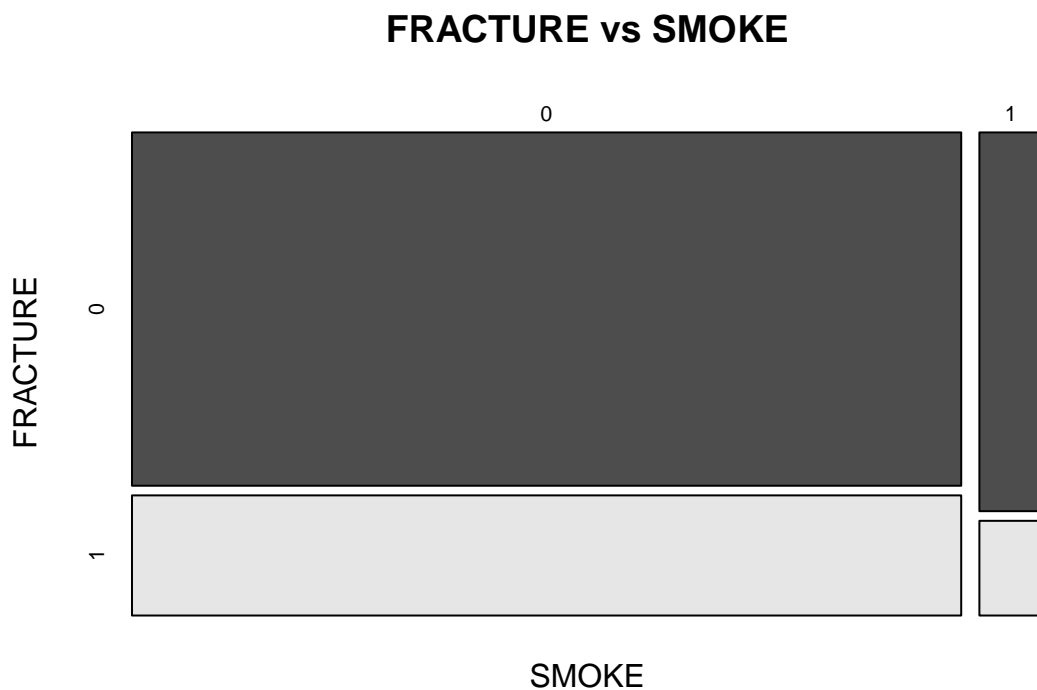
```
## Total Observations in Table:   500
##
##
##                           | dataset$FRACTURE
## dataset[, categoricalVar] |          0 |          1 | Row Total |
## -------------------------|-----------|-----------|-----------|
##                        0 |        347 |        118 |        465 |
##                          |      0.009 |      0.026 |           |
##                          |      0.746 |      0.254 |      0.930 |
##                          |      0.925 |      0.944 |           |
##                          |      0.694 |      0.236 |           |
## -------------------------|-----------|-----------|-----------|
##                        1 |         28 |          7 |         35 |
##                          |      0.117 |      0.350 |           |
##                          |      0.800 |      0.200 |      0.070 |
##                          |      0.075 |      0.056 |           |
##                          |      0.056 |      0.014 |           |
## -------------------------|-----------|-----------|-----------|
##             Column Total |        375 |        125 |        500 |
##                          |      0.750 |      0.250 |           |
## -------------------------|-----------|-----------|-----------|
##
##
```

FRACTURE vs SMOKE


```
##
##
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##              | FRACTURE
##    RATERISK  |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##           1  |       139 |        28 |       167 |
##              |   125.250 |    41.750 |           |
##              |     1.509 |     4.528 |           |
##              |     0.832 |     0.168 |     0.334 |
##              |     0.371 |     0.224 |           |
##              |     0.278 |     0.056 |           |
## -------------|-----------|-----------|-----------|
##           2  |       138 |        48 |       186 |
##              |   139.500 |    46.500 |           |
##              |     0.016 |     0.048 |           |
##              |     0.742 |     0.258 |     0.372 |
##              |     0.368 |     0.384 |           |
##              |     0.276 |     0.096 |           |
## -------------|-----------|-----------|-----------|
##           3  |        98 |        49 |       147 |
##              |   110.250 |    36.750 |           |
##              |     1.361 |     4.083 |           |
##              |     0.667 |     0.333 |     0.294 |
##              |     0.261 |     0.392 |           |
##              |     0.196 |     0.098 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       375 |       125 |       500 |
##              |     0.750 |     0.250 |           |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  11.54688     d.f. =  2     p =  0.003109037
##
##
##
##
##
```

```
##    Cell Contents
## |-----------------------|
## |                     N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-----------------------|
##
##
## Total Observations in Table:  500
##
##
##                          | dataset$FRACTURE
## dataset[, categoricalVar] |          0 |          1 | Row Total |
## ------------------------|-----------|-----------|-----------|
##                       1 |        139 |         28 |        167 |
##                         |      1.509 |      4.528 |            |
##                         |      0.832 |      0.168 |      0.334 |
##                         |      0.371 |      0.224 |            |
##                         |      0.278 |      0.056 |            |
## ------------------------|-----------|-----------|-----------|
##                       2 |        138 |         48 |        186 |
##                         |      0.016 |      0.048 |            |
##                         |      0.742 |      0.258 |      0.372 |
##                         |      0.368 |      0.384 |            |
##                         |      0.276 |      0.096 |            |
## ------------------------|-----------|-----------|-----------|
##                       3 |         98 |         49 |        147 |
##                         |      1.361 |      4.083 |            |
##                         |      0.667 |      0.333 |      0.294 |
##                         |      0.261 |      0.392 |            |
##                         |      0.196 |      0.098 |            |
## ------------------------|-----------|-----------|-----------|
##            Column Total |        375 |        125 |        500 |
##                         |      0.750 |      0.250 |            |
## ------------------------|-----------|-----------|-----------|
##
##
```
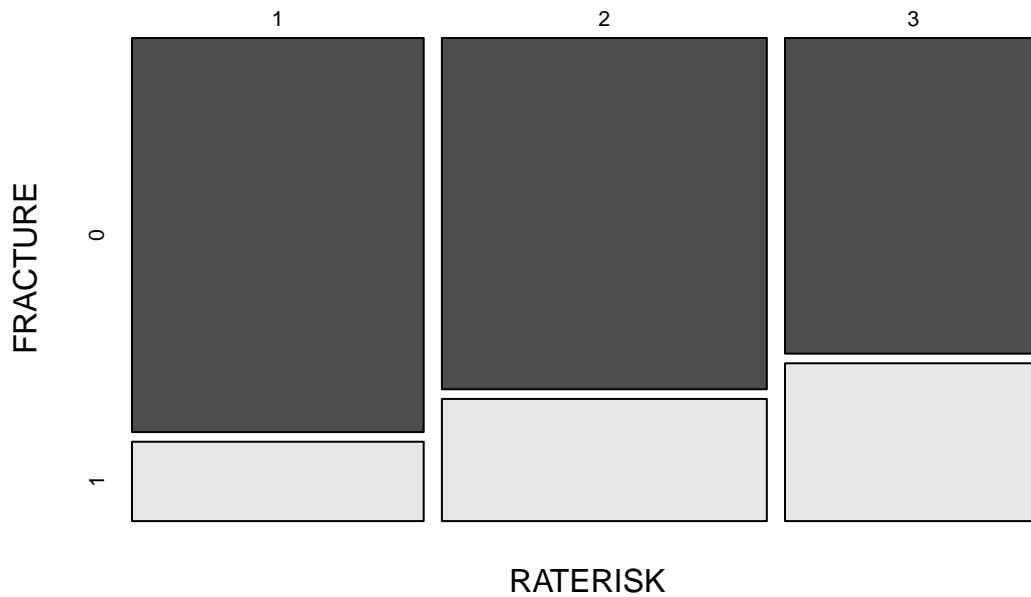
FRACTURE vs RATERISK

#Logistic Regression

Training set will be 70% of dataset and Test set will be remaining 30%

Build Model using Training Data

Question of Interest? What are the odds of getting a fracture, given certain conditions?

```
set.seed(84)
model <- glm(FRACTURE ~ AGE + WEIGHT + HEIGHT + BMI + PRIORFRAC + PREMENO + MOMFRAC + ARMASSIST + SMOKE
model
```

```
##
## Call:  glm(formula = FRACTURE ~ AGE + WEIGHT + HEIGHT + BMI + PRIORFRAC +
##     PREMENO + MOMFRAC + ARMASSIST + SMOKE + RATERISK, family = "binomial",
##     data = trainingData)
##
## Coefficients:
## (Intercept)          AGE       WEIGHT       HEIGHT          BMI
##    -12.04673      0.03168     -0.10711      0.04735      0.29193
##   PRIORFRAC1      PREMENO1      MOMFRAC1    ARMASSIST1       SMOKE1
##      0.73265      0.04114      0.35482      0.30067     -0.08005
##    RATERISK2    RATERISK3
##      0.38692      0.57786
##
## Degrees of Freedom: 375 Total (i.e. Null);  364 Residual
```

```
## Null Deviance:        422.9
## Residual Deviance: 385.4      AIC: 409.4
```

```
summary(model)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + WEIGHT + HEIGHT + BMI + PRIORFRAC +
##     PREMENO + MOMFRAC + ARMASSIST + SMOKE + RATERISK, family = "binomial",
##     data = trainingData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4739  -0.7388  -0.5757  -0.1189   2.1597
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.04673   13.81668  -0.872  0.38326
## AGE           0.03168    0.01715   1.847  0.06472 .
## WEIGHT       -0.10711    0.09271  -1.155  0.24793
## HEIGHT        0.04735    0.08516   0.556  0.57823
## BMI           0.29193    0.23882   1.222  0.22157
## PRIORFRAC1    0.73265    0.28371   2.582  0.00981 **
## PREMENO1      0.04114    0.32545   0.126  0.89940
## MOMFRAC1      0.35482    0.36197   0.980  0.32697
## ARMASSIST1    0.30067    0.29666   1.014  0.31080
## SMOKE1       -0.08005    0.50041  -0.160  0.87290
## RATERISK2     0.38692    0.32506   1.190  0.23393
## RATERISK3     0.57786    0.34936   1.654  0.09812 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 422.88  on 375  degrees of freedom
## Residual deviance: 385.45  on 364  degrees of freedom
## AIC: 409.45
##
## Number of Fisher Scoring iterations: 4
```

```
h1 <- hoslem.test(model$y, fitted(model), g = 10) #number of groups to divide dataset into is 10
h1
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model$y, fitted(model)
## X-squared = 7.8006, df = 8, p-value = 0.4532
```

Interpretation of logistic regression model: Weight, height, BMI, Premeno, Armassist, and Smoke are not statistically significant variables. Priorfrac and Age are statistically significant variables and have the lowest p-value indicating a strong association with having a Fracture.

Clustering

```
#Lets look at a heatmap using hierarchical clustering to see if the
#response naturually clusters out using the predictors

#Transposting the predictor matrix and giving the response categories its
#row names.
#Get Training Set

# convert factors to numeric for pheatmap
temp <- trainingData
indx <- sapply(temp, is.factor)
temp[indx] <- lapply(temp[indx], function(x) as.numeric(as.character(x)))

dat.train <- temp

dat.train.x <- dat.train[,1:ncol(dat.train)]
dat.train.y <- dat.train$FRACTURE

dat.train.y <- as.factor(as.character(dat.train.y))

#Heatmap
x<-t(dat.train.x)
colnames(x)<-dat.train.y
pheatmap(x,annotation_col=data.frame(FRACTURE=dat.train.y),scale="row",legend=T,color=colorRampPalette(
```
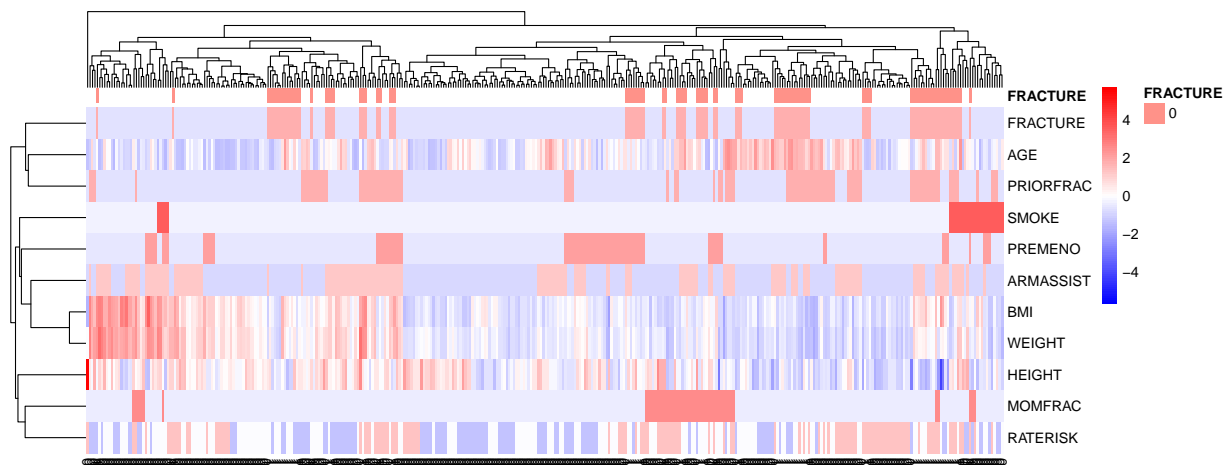


```
##logistic regression
dat.train.x <- as.matrix(dat.train.x)

cvfit <- cv.glmnet(dat.train.x, dat.train.y, family = "binomial", type.measure = "class", nlambda = 1000
plot(cvfit)
```

```r
coef(cvfit, s = "lambda.min")
```
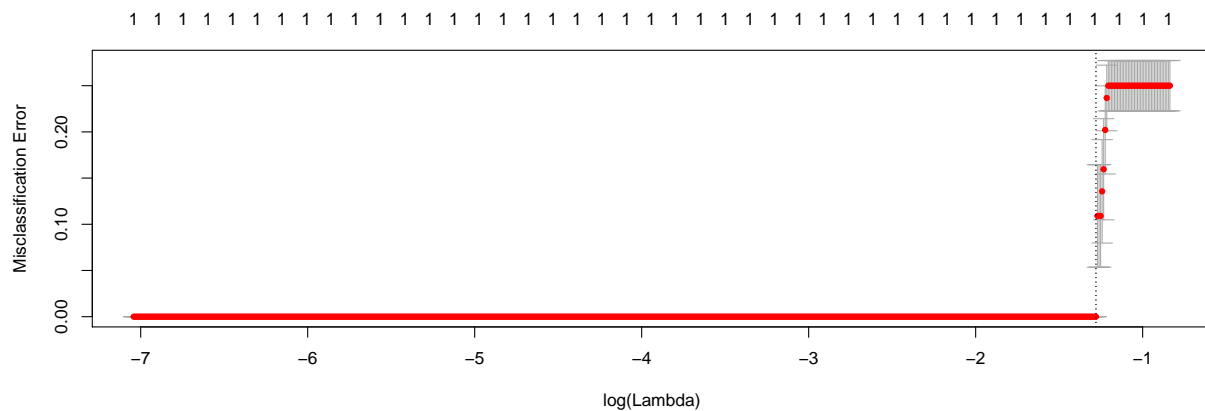
```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) -1.653758
## AGE          .
## BMI          .
## HEIGHT       .
## WEIGHT       .
## PRIORFRAC    .
## PREMENO      .
## MOMFRAC      .
## ARMASSIST    .
## SMOKE        .
## RATERISK     .
## FRACTURE     1.726571
```

*** *Appendix B:* Model Comparison - Analysis ===========

```
## 'data.frame':    500 obs. of  14 variables:
##  $ SUB_ID   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SITE_ID  : int  1 4 6 6 1 5 5 1 1 4 ...
##  $ PHY_ID   : int  14 284 305 309 37 299 302 36 8 282 ...
##  $ PRIORFRAC: Factor w/ 2 levels "0","1": 1 1 2 1 1 2 1 2 2 1 ...
##  $ AGE      : int  62 65 88 82 61 67 84 82 86 58 ...
##  $ WEIGHT   : num  70.3 87.1 50.8 62.1 68 68 50.8 40.8 62.6 63.5 ...
##  $ HEIGHT   : int  158 160 157 160 152 161 150 153 156 166 ...
##  $ BMI      : num  28.2 34 20.6 24.3 29.4 ...
##  $ PREMENO  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MOMFRAC  : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ...
##  $ ARMASSIST: Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ...
##  $ SMOKE    : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ RATERISK : Factor w/ 3 levels "1","2","3": 2 2 1 1 2 2 1 2 2 1 ...
##  $ FRACTURE : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Create Train and Validation Datasets

```r
set.seed(999)
validation_index = createDataPartition(dataset$FRACTURE, p=0.70, list=FALSE)
validationData = dataset[-validation_index,c(4:14)]
trainingData = dataset[validation_index,c(4:14)]

table(dataset$FRACTURE)
```

```
##
##   0   1
## 375 125
```

```r
table(trainingData$FRACTURE)
```

```
##
##   0   1
## 263  88
```

```r
table(validationData$FRACTURE)
```

```
##
##   0   1
## 112  37
```

```r
#BarPlots of Fracture counts between full, training and validation datasets.
par(mfrow=c(1,3))
#par(mar=c(5,8,4,2)) # increase y-axis margin.
count_full <- table(dataset$FRACTURE)
count_trn <- table(trainingData$FRACTURE)
count_test <- table(validationData$FRACTURE)


barplot(count_full,main="Full Dataset", ylab="Count", col=c("orange","blue"),names.arg=c("0 No Fracture

barplot(count_trn,main="Training Dataset", ylab="Count", col=c("orange","blue"),names.arg=c("0 No Fract

barplot(count_test,main="Validation Dataset", ylab="Count", col=c("orange","blue"),names.arg=c("0 No Fra
```

**Full Dataset**   **Training Dataset**   **Validation Dataset**
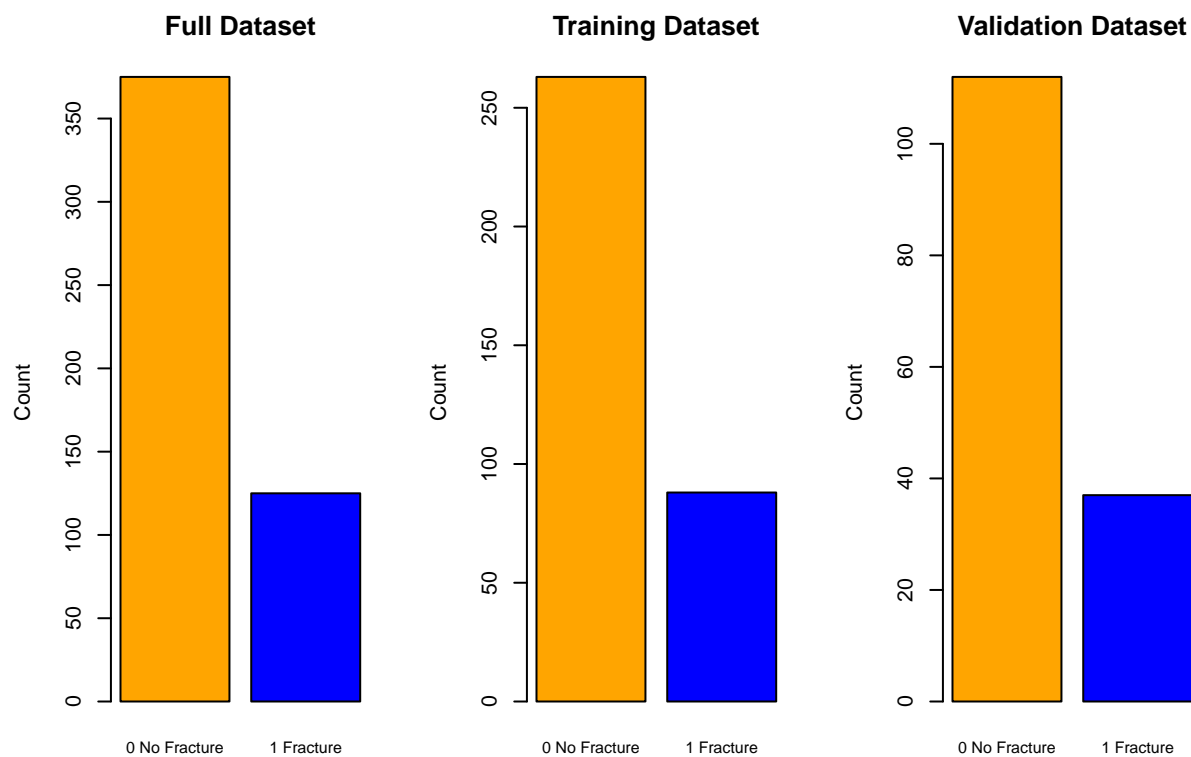
```
set.seed(999)

## Formatting Test Data Set
# Recode Rate Risk Variable since its ordinal and we donot want to loose its info if it gets
# coded as nominal variable before running the Model
validationData$RATERISK <- factor(validationData$RATERISK, levels = c(1,2,3), ordered = T)

xfactors_test <- model.matrix(validationData$FRACTURE ~ validationData$PRIORFRAC + validationData$PREMEN
x_test <- as.matrix(data.frame(validationData$AGE, validationData$WEIGHT, validationData$HEIGHT, validat

## Formatting Training Data Set
trainingData$RATERISK <- factor(trainingData$RATERISK, levels = c(1,2,3), ordered = T)
xfactors_train <- model.matrix(trainingData$FRACTURE ~ trainingData$PRIORFRAC + trainingData$PREMENO + 
x_train <- as.matrix(data.frame(trainingData$AGE, trainingData$WEIGHT, trainingData$HEIGHT, trainingData

# Doing Cross validation to find the best fitting model based upon Lasso
cvfit <- cv.glmnet(x_train, y=trainingData$FRACTURE, family = "binomial", type.measure = "class", nlambd
plot(cvfit)
```
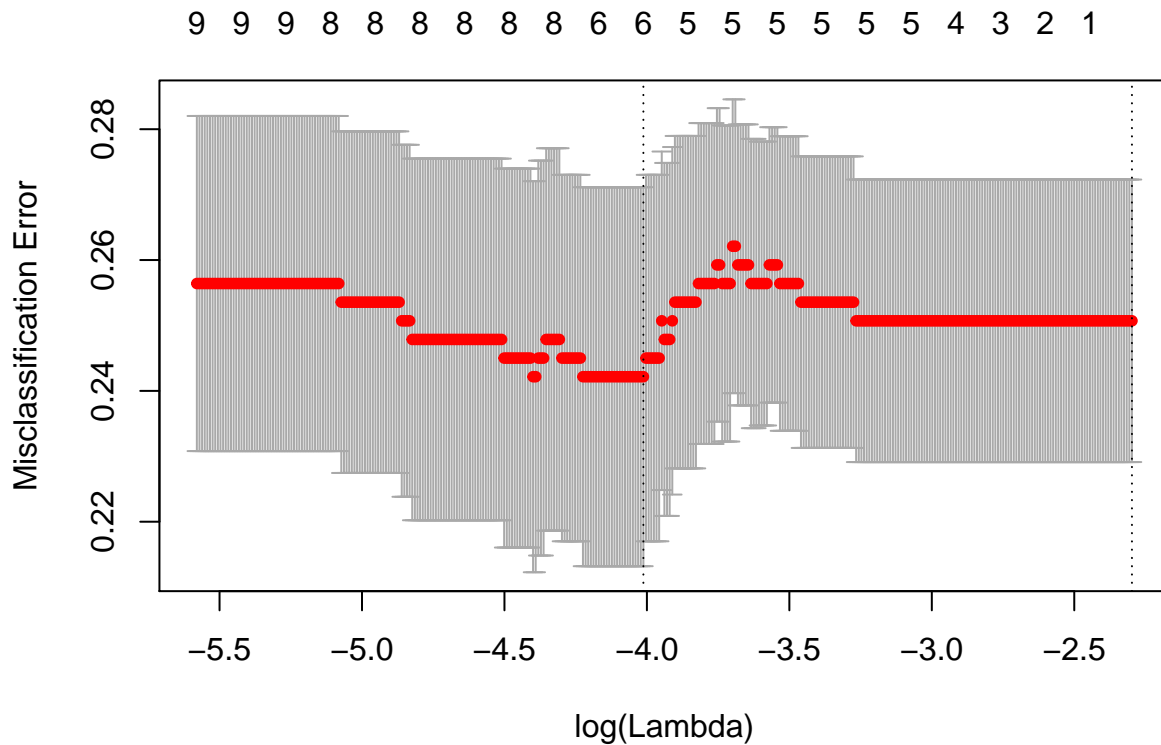
```
# Model with Lowest Lambda is shrinking all the coefficients, hence selecting lambda based upon
# Test Set AUC and EDA Results
#cvfit$glmnet.fit
coef(cvfit, s="lambda.min")
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                 1
## (Intercept)             1.52902669
## trainingData.AGE        0.03598544
## trainingData.WEIGHT     .
## trainingData.HEIGHT    -0.03340871
## trainingData.BMI        .
## trainingData.PRIORFRAC1 0.15180349
## trainingData.PREMENO1   .
## trainingData.MOMFRAC1   0.04035537
## trainingData.ARMASSIST1 0.52512963
## trainingData.SMOKE1     .
## trainingData.RATERISK.L 0.33991586
## trainingData.RATERISK.Q .
```

```
# Fitting the best model based upon selected lambda
fit <- glmnet(x_train, y=trainingData$FRACTURE, family="binomial", alpha = 1, lambda = cvfit$lambda.min)

# First Predicting the responses on training data set itself
fit.pred <- predict(fit, newx = x_train, type = "response")
```
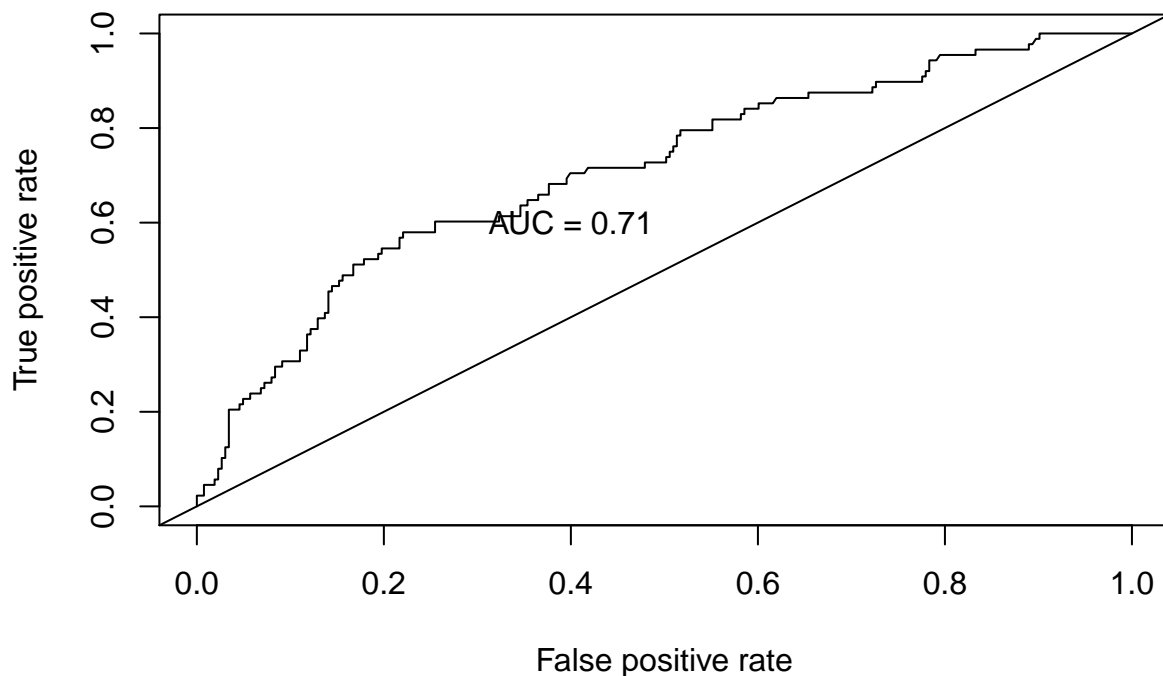
```r
#Create ROC curves for training Data Set
pred <- prediction(fit.pred[,1], trainingData$FRACTURE)
roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred, measure = "auc")
auc.train <- auc.train@y.values

##Plot ROC for training Set
plot(roc.perf)
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```
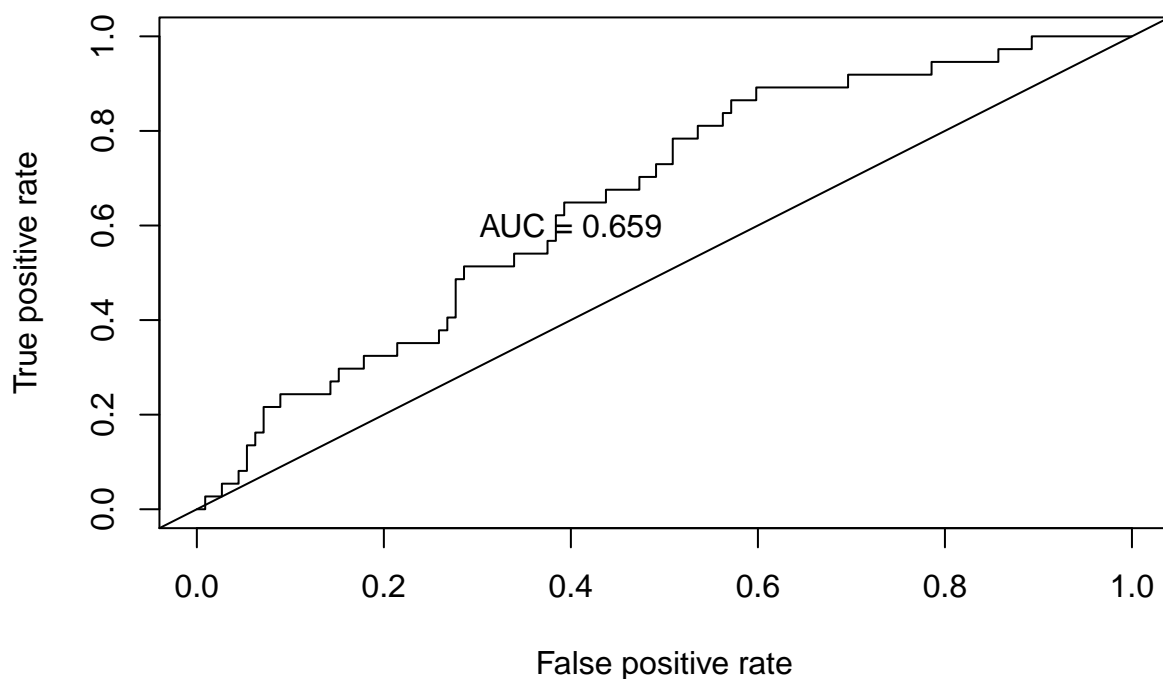


```r
#Run model from training set on validation Set
fit.pred1 <- predict(fit, newx = x_test, type = "response")

#ROC curves
pred1 <- prediction(fit.pred1[,1], validationData$FRACTURE)
roc.perf1 = performance(pred1, measure = "tpr", x.measure = "fpr")
auc.val1 <- performance(pred1, measure = "auc")
auc.val1 <- auc.val1@y.values
plot(roc.perf1)
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(auc.val1[[1]],3), sep = ""))
```

```
#confusion matrix
pdata <- predict(fit, newx = x_test, type = "response")
pdata_logical <- pdata[, 1] > 0.5
confusionMatrix(data = as.factor(as.numeric(pdata_logical)), reference = as.factor(as.numeric(validation
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 108  35
##          1   4   2
##
##                Accuracy : 0.7383
##                  95% CI : (0.66, 0.8068)
##     No Information Rate : 0.7517
##     P-Value [Acc > NIR] : 0.6866
##
##                   Kappa : 0.0255
##  Mcnemar's Test P-Value : 1.556e-06
##
##             Sensitivity : 0.96429
##             Specificity : 0.05405
##          Pos Pred Value : 0.75524
##          Neg Pred Value : 0.33333
##              Prevalence : 0.75168
##          Detection Rate : 0.72483
```

```
##      Detection Prevalence : 0.95973
##         Balanced Accuracy : 0.50917
##
##          'Positive' Class : 0
##
```

```
#mydata <- dataset[, c(4:14)] %>% dplyr::select_if(is.numeric)
#predictors <- colnames(mydata)
#mydata <- mydata %>%
#  mutate(logit = log(probabilities/(1-probabilities))) %>%
#  gather(key = "predictors", value = "predictor.value", -logit)
```

Run Normal Logit Model with Identified Predictors

```
set.seed(999)

logit.fit <- glm(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK , data = trainingD
summary(logit.fit)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK, family = binomial(link = "logit"),
##     data = trainingData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5491  -0.7377  -0.5763   0.2298   2.2214
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.33365    3.80104   0.877  0.38047
## AGE          0.04347    0.01578   2.755  0.00587 **
## HEIGHT      -0.04881    0.02165  -2.254  0.02418 *
## PRIORFRAC1   0.22281    0.30097   0.740  0.45912
## MOMFRAC1     0.33522    0.38263   0.876  0.38097
## ARMASSIST1   0.68418    0.27861   2.456  0.01406 *
## RATERISK.L   0.50762    0.24656   2.059  0.03951 *
## RATERISK.Q  -0.06727    0.22219  -0.303  0.76209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.31  on 350  degrees of freedom
## Residual deviance: 355.55  on 343  degrees of freedom
## AIC: 371.55
##
## Number of Fisher Scoring iterations: 4
```

```r
# To exponentiate the log ODDS to make it ODDS Ratio and also get corresponding 95% CIs
exp(cbind(ODDs_Ratio = coef(logit.fit), confint(logit.fit)))
```

```
## Waiting for profiling to be done...
```

```
##              ODDs_Ratio      2.5 %        97.5 %
## (Intercept) 28.0403767 0.0172620 5.378401e+04
## AGE          1.0444253 1.0128322 1.077662e+00
## HEIGHT       0.9523628 0.9118527 9.929059e-01
## PRIORFRAC1   1.2495774 0.6857875 2.237992e+00
## MOMFRAC1     1.3982464 0.6449924 2.917750e+00
## ARMASSIST1   1.9821443 1.1469582 3.428352e+00
## RATERISK.L   1.6613370 1.0283916 2.713240e+00
## RATERISK.Q   0.9349462 0.6060378 1.451499e+00
```

```r
# First Predicting the responses on training data set itself
logistic.fit.pred.train <- predict(logit.fit, newdata=trainingData, type = "response")

#Create ROC curves for training Data Set
pred.train <- prediction(logistic.fit.pred.train, trainingData$FRACTURE)
roc.perf = performance(pred.train, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.train, measure = "auc")
auc.train <- auc.train@y.values

##Plot ROC for training Set
plot(roc.perf, main="Logistic Reg Training Data Set")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```
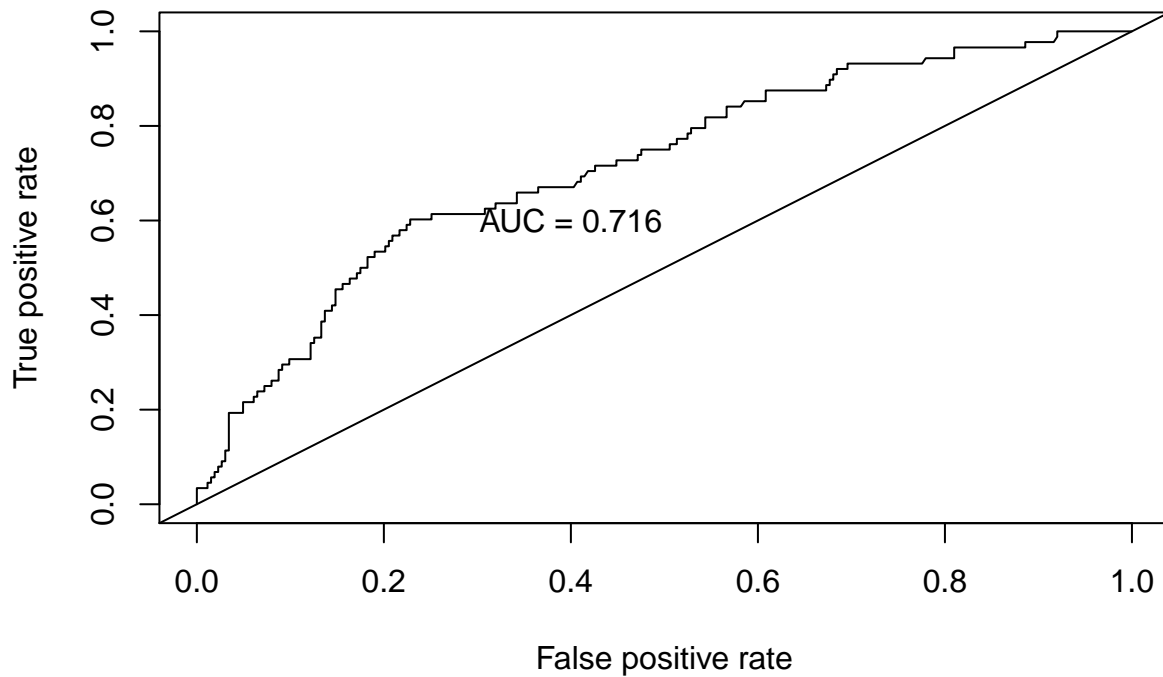
Logistic Reg Training Data Set


```r
#Run model from training set on validation Set
logistic.fit.pred.test <- predict(logit.fit, newdata=validationData, type = "response")

#ROC curves
pred.test <- prediction(logistic.fit.pred.test, validationData$FRACTURE)
roc.perf1 = performance(pred.test, measure = "tpr", x.measure = "fpr")
auc.val1 <- performance(pred.test, measure = "auc")
auc.val1 <- auc.val1@y.values
plot(roc.perf1, main="Logistic Reg Validation Data Set")
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(auc.val1[[1]],3), sep = ""))
```

Logistic Reg Validation Data Set
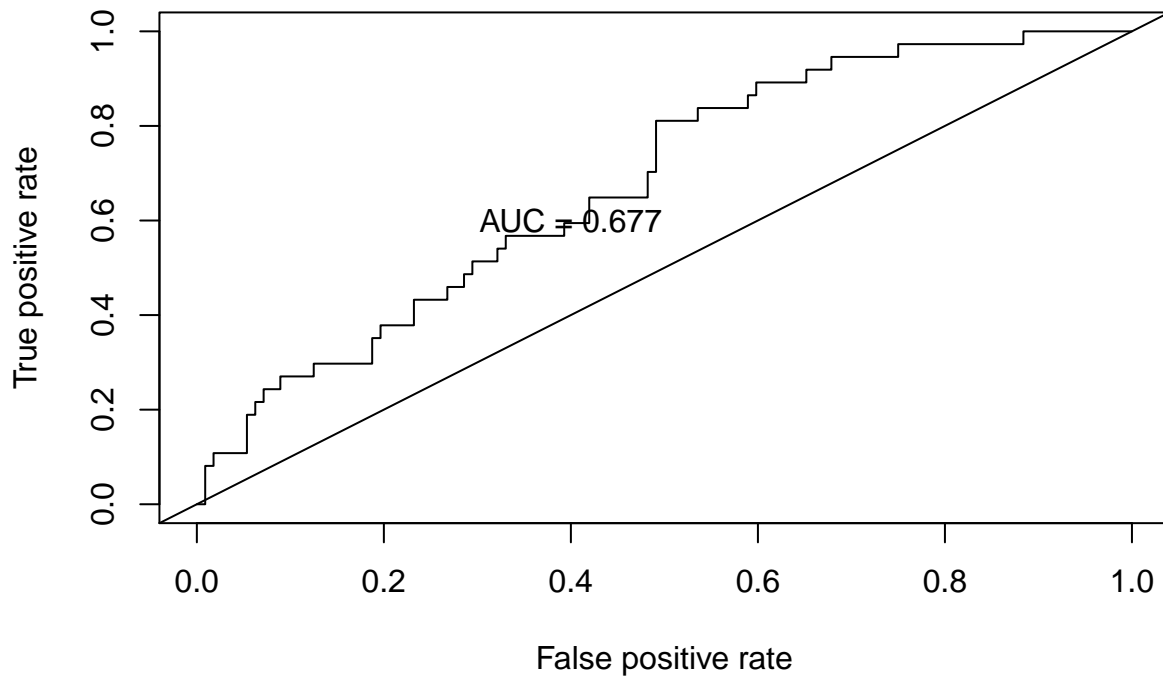

```
#confusion matrix
pdata_logical <-  logistic.fit.pred.test > 0.5
confusionMatrix(data = as.factor(as.numeric(pdata_logical)), reference = as.factor(as.numeric(validatio
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##         0 106  32
##         1   6   5
##
##              Accuracy : 0.745
##                95% CI : (0.6672, 0.8128)
##    No Information Rate : 0.7517
##    P-Value [Acc > NIR] : 0.6175
##
##                 Kappa : 0.1067
##  Mcnemar's Test P-Value : 5.002e-05
##
##           Sensitivity : 0.9464
##           Specificity : 0.1351
##        Pos Pred Value : 0.7681
##        Neg Pred Value : 0.4545
##            Prevalence : 0.7517
##        Detection Rate : 0.7114
##  Detection Prevalence : 0.9262
```

```
##         Balanced Accuracy : 0.5408
##
##          'Positive' Class : 0
##
```

Add Interactions to Normal logit

```
set.seed(999)
# Since top 3 predictors are Age, PriorFrac and RISK, adding model complexity
# via interactions
logit.fit.interactions <- glm(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK + AGI
summary(logit.fit.interactions)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:PRIORFRAC + RATERISK:AGE + MOMFRAC:ARMASSIST,
##     family = binomial(link = "logit"), data = trainingData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5521  -0.7592  -0.5543   0.2845   2.3802
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.65488    3.98301   0.415  0.67779
## AGE               0.06735    0.02058   3.273  0.00107 **
## HEIGHT           -0.04913    0.02191  -2.242  0.02495 *
## PRIORFRAC1        3.83397    2.26015   1.696  0.08982 .
## MOMFRAC1          0.74167    0.50795   1.460  0.14426
## ARMASSIST1        0.80585    0.29990   2.687  0.00721 **
## RATERISK.L        1.64141    2.00121   0.820  0.41210
## RATERISK.Q       -1.25322    1.78216  -0.703  0.48193
## AGE:PRIORFRAC1   -0.05038    0.03124  -1.612  0.10688
## AGE:RATERISK.L   -0.01548    0.02788  -0.555  0.57869
## AGE:RATERISK.Q    0.01632    0.02498   0.653  0.51355
## MOMFRAC1:ARMASSIST1 -0.74229  0.76220  -0.974  0.33011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.31  on 350  degrees of freedom
## Residual deviance: 351.13  on 339  degrees of freedom
## AIC: 375.13
##
## Number of Fisher Scoring iterations: 5
```

```
# First Predicting the responses on training data set itself
logistic.fit.pred.train.interaction <- predict(logit.fit.interactions, newdata=trainingData, type = "re:
```

```
#Create ROC curves for training Data Set
pred.train.interaction <- prediction(logistic.fit.pred.train.interaction, trainingData$FRACTURE)
roc.perf = performance(pred.train.interaction, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.train.interaction, measure = "auc")
auc.train <- auc.train@y.values

##Plot ROC for training Set
plot(roc.perf, main="Logistic Reg With Interactions Training Data Set")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```

## Logistic Reg With Interactions Training Data Set
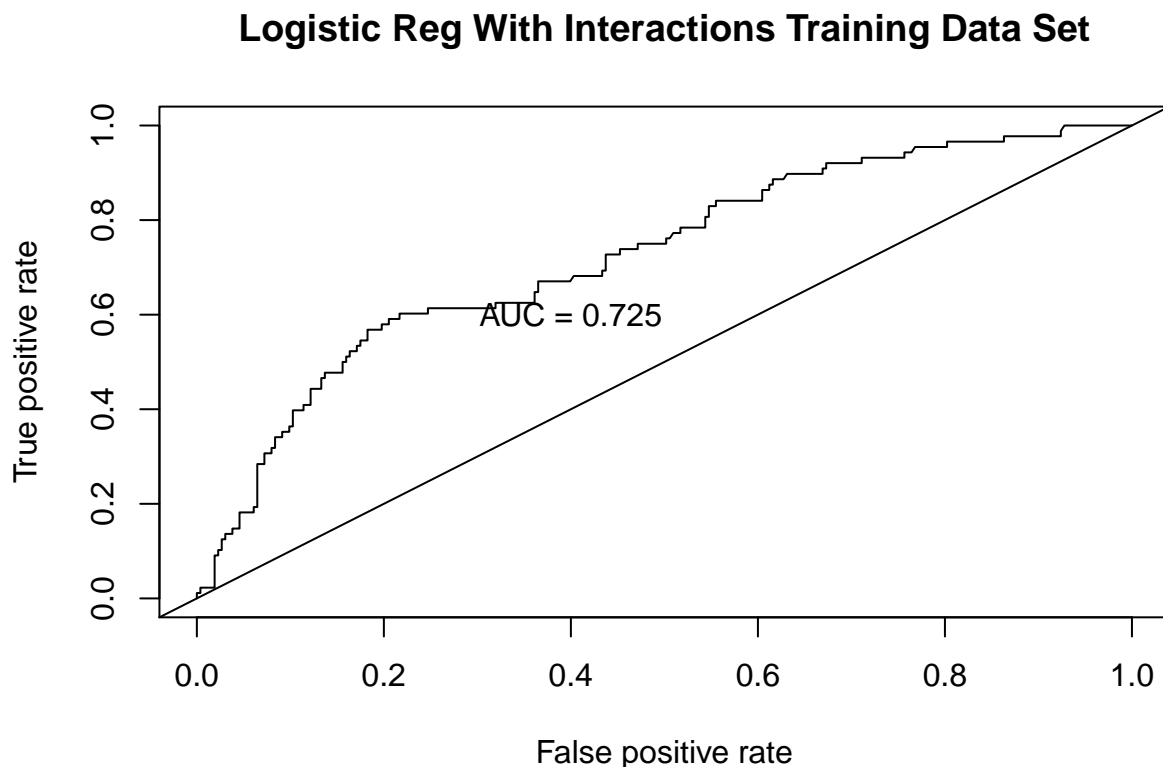

```
#Run model from training set on validation Set
logistic.fit.pred.test.interaction <- predict(logit.fit.interactions, newdata=validationData, type = "r

#ROC curves
pred.test.interaction <- prediction(logistic.fit.pred.test.interaction, validationData$FRACTURE)
roc.perf1 = performance(pred.test.interaction, measure = "tpr", x.measure = "fpr")
auc.val1 <- performance(pred.test.interaction, measure = "auc")
auc.val1 <- auc.val1@y.values
plot(roc.perf1, main="Logistic Reg With Interactions Validations Data Set")
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(auc.val1[[1]],3), sep = ""))
```

Logistic Reg With Interactions Validations Data Set
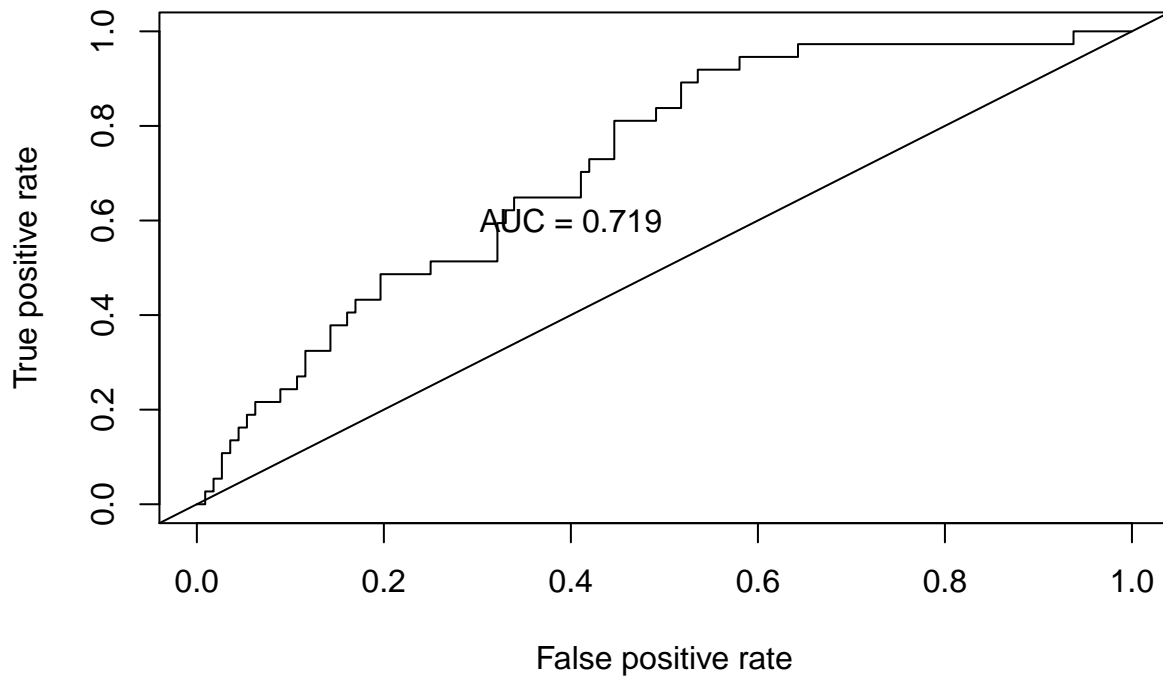

```r
#confusion matrix
pdata_logical <-  logistic.fit.pred.test.interaction > 0.5
confusionMatrix(data = as.factor(as.numeric(pdata_logical)), reference = as.factor(as.numeric(validatio
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 105  30
##          1   7   7
##
##                Accuracy : 0.7517
##                  95% CI : (0.6743, 0.8187)
##     No Information Rate : 0.7517
##     P-Value [Acc > NIR] : 0.5440183
##
##                   Kappa : 0.16
##  Mcnemar's Test P-Value : 0.0002983
##
##             Sensitivity : 0.9375
##             Specificity : 0.1892
##          Pos Pred Value : 0.7778
##          Neg Pred Value : 0.5000
##              Prevalence : 0.7517
##          Detection Rate : 0.7047
##    Detection Prevalence : 0.9060
```

```
##       Balanced Accuracy : 0.5633
##
##          'Positive' Class : 0
##
```

```r
# Checking the assumptions
probabilities <- predict(logit.fit.interactions, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")
head(predicted.classes)
```

```
##     1     3     4     5     6     7
## "neg" "neg" "neg" "neg" "neg" "neg"
```

```r
# Linearity assumption
subNumericPred <- trainingData %>% dplyr::select(AGE, HEIGHT)
predictors <- colnames(subNumericPred)
subNumericPred <- subNumericPred %>%
                  mutate(logit = log(probabilities/(1-probabilities))) %>%
                  gather(key = "predictors", value = "predictor.value", -logit)

ggplot(subNumericPred, aes(logit, predictor.value)) +
                geom_point(size = 0.5, alpha = 0.5) +
                geom_smooth(method = "loess") +
                theme_bw() +
                facet_wrap(~predictors, scales = "free_y")
```
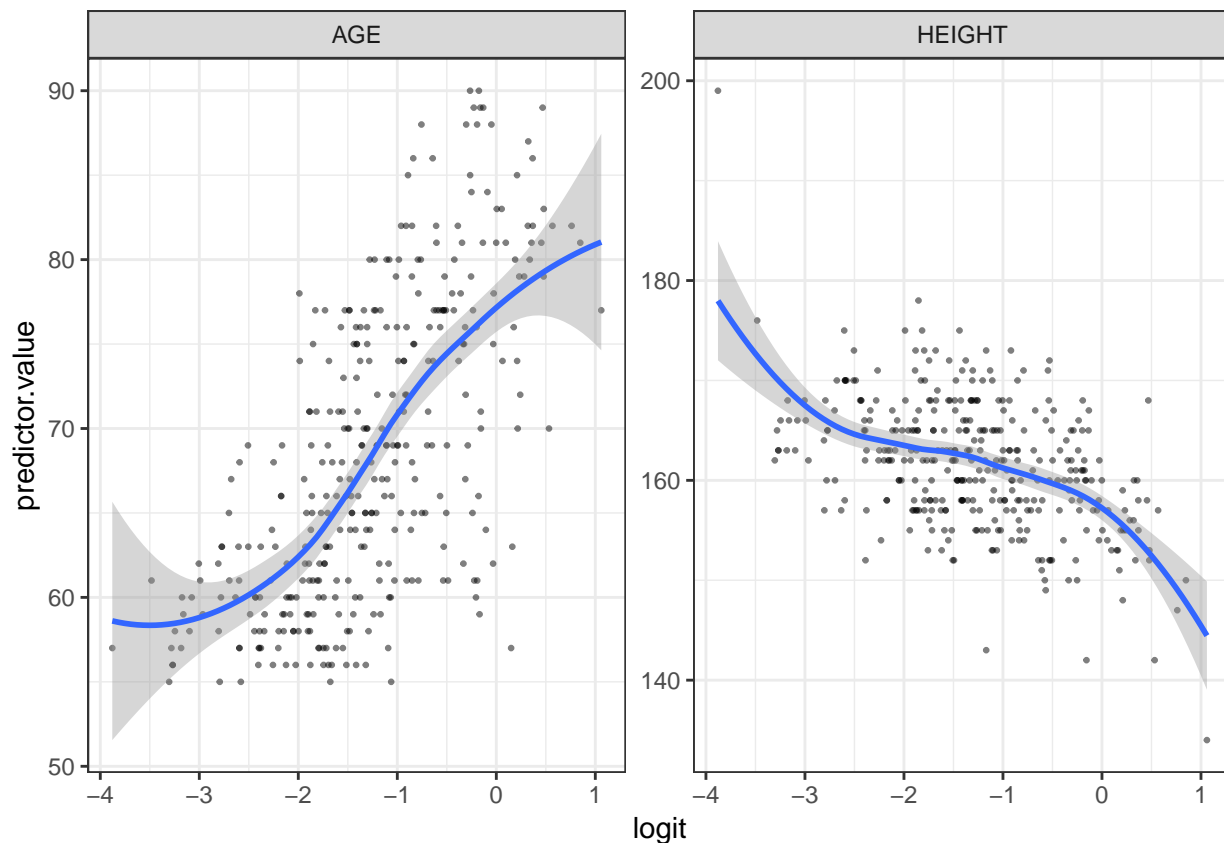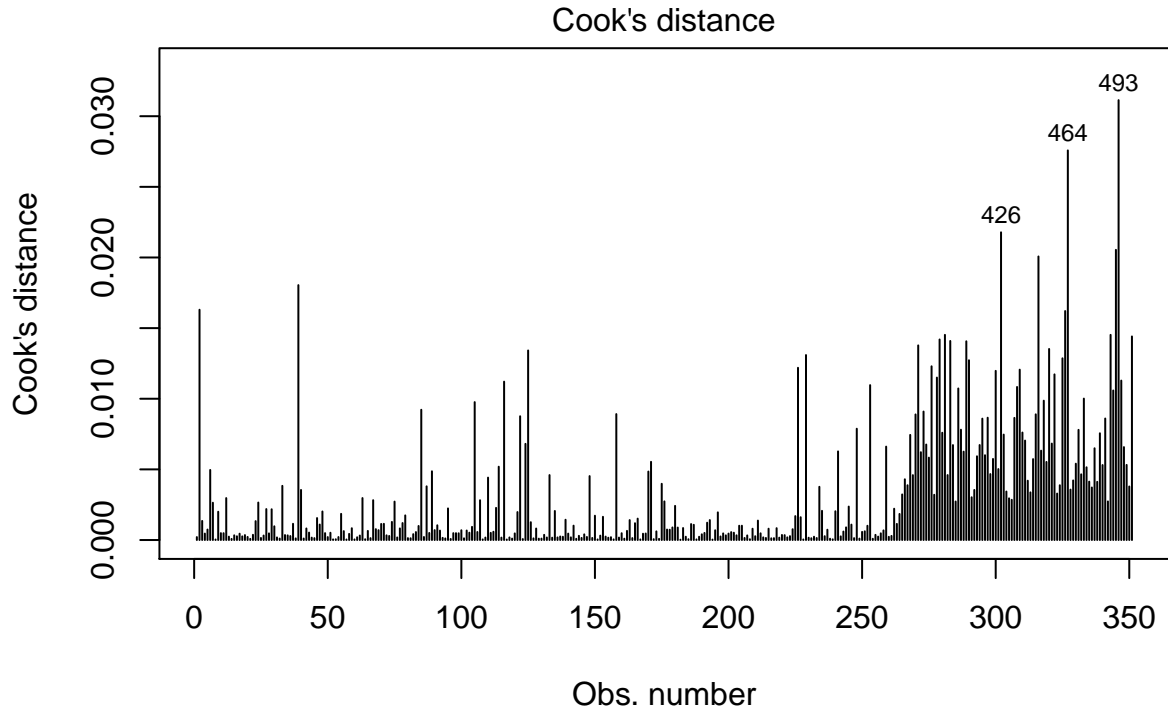
```
plot(logit.fit.interactions, which = 4, id.n =3)
```

Cook's distance

Obs. number

ı(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERI

## Running Random Forest Fit

```
set.seed(999)
```

```
str(trainingData)
```

```
## 'data.frame':    351 obs. of  11 variables:
##  $ PRIORFRAC: Factor w/ 2 levels "0","1": 1 2 1 1 2 1 2 1 1 1 ...
##  $ AGE      : int   62 88 82 61 67 84 86 58 67 56 ...
##  $ WEIGHT   : num   70.3 50.8 62.1 68 68 ...
##  $ HEIGHT   : int   158 157 160 152 161 150 156 166 153 167 ...
##  $ BMI      : num   28.2 20.6 24.3 29.4 26.2 ...
##  $ PREMENO  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MOMFRAC  : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 2 1 ...
##  $ ARMASSIST: Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 2 ...
##  $ SMOKE    : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 2 2 ...
##  $ RATERISK : Ord.factor w/ 3 levels "1"<"2"<"3": 2 1 1 2 2 1 2 1 1 2 ...
##  $ FRACTURE : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 1 ...
```

```
rf.fit <- randomForest(FRACTURE ~ ., data=trainingData, mtry=4, ntree=500, maxnodes = 12, importance=T)
rf.fit
```

```
##
## Call:
##  randomForest(formula = FRACTURE ~ ., data = trainingData, mtry = 4,      ntree = 500, maxnodes = 12
##                 Type of random forest: classification
##                       Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 24.79%
## Confusion matrix:
##     0 1 class.error
## 0 260 3  0.01140684
## 1  84 4  0.95454545
```

```
rf.fit.pred.train <- predict(rf.fit, newdata=trainingData, type="prob")
pred.train.rf <- prediction(rf.fit.pred.train[,2], trainingData$FRACTURE)
roc.perf = performance(pred.train.rf, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.train.rf, measure = "auc")
auc.train <- auc.train@y.values
plot(roc.perf, main="Random Forest Training Data Set")
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```

Random Forest Training Data Set
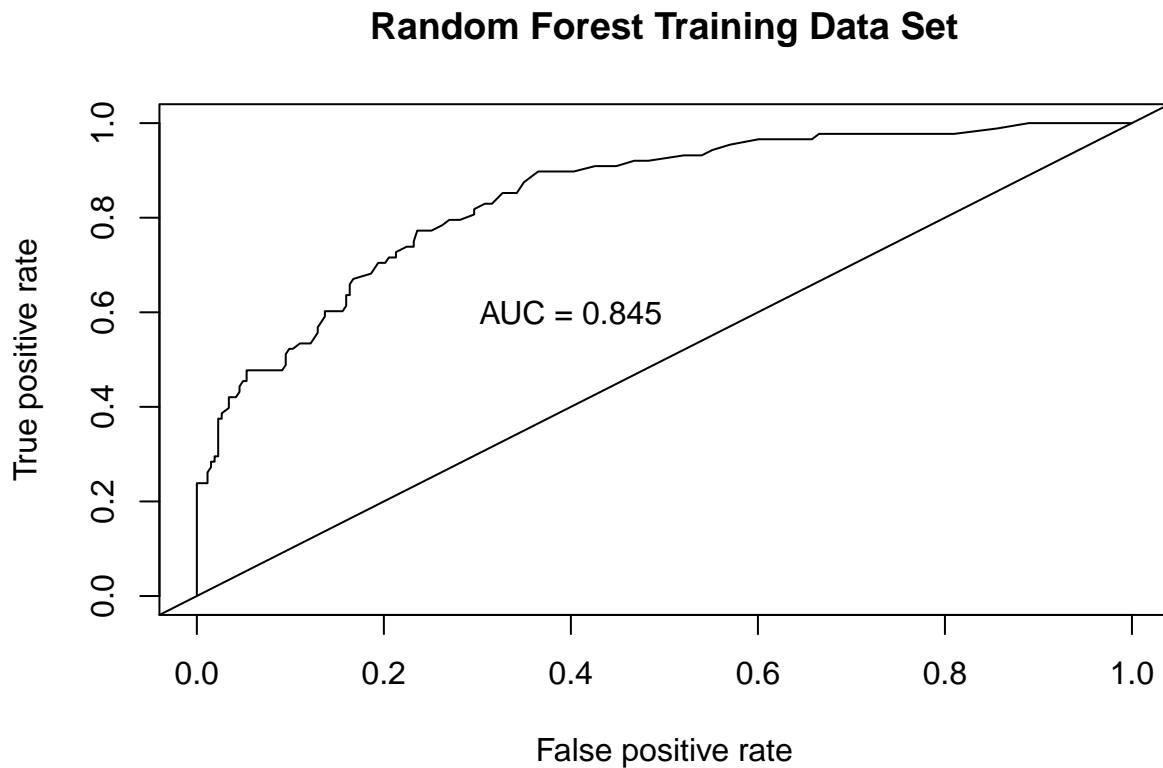
```
#confusion matrix Training
pdata_logical_train <-  (rf.fit.pred.train[,2] >= 0.5)
confusionMatrix(data = as.factor(as.numeric(pdata_logical_train)), reference = as.factor(as.numeric(tra
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 263  76
##          1   0  12
##
##                Accuracy : 0.7835
##                  95% CI : (0.7367, 0.8254)
##     No Information Rate : 0.7493
##     P-Value [Acc > NIR] : 0.07672
##
##                   Kappa : 0.1913
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.1364
##          Pos Pred Value : 0.7758
##          Neg Pred Value : 1.0000
##              Prevalence : 0.7493
##          Detection Rate : 0.7493
##    Detection Prevalence : 0.9658
##       Balanced Accuracy : 0.5682
##
##        'Positive' Class : 0
##
```

```
rf.fit.pred.test <- predict(rf.fit, newdata=validationData, type="prob")
pred.test.rf <- prediction(rf.fit.pred.test[,2], validationData$FRACTURE)
roc.perf = performance(pred.test.rf, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.test.rf, measure = "auc")
auc.train <- auc.train@y.values
plot(roc.perf, main="Random Forest Validation Data Set")
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```
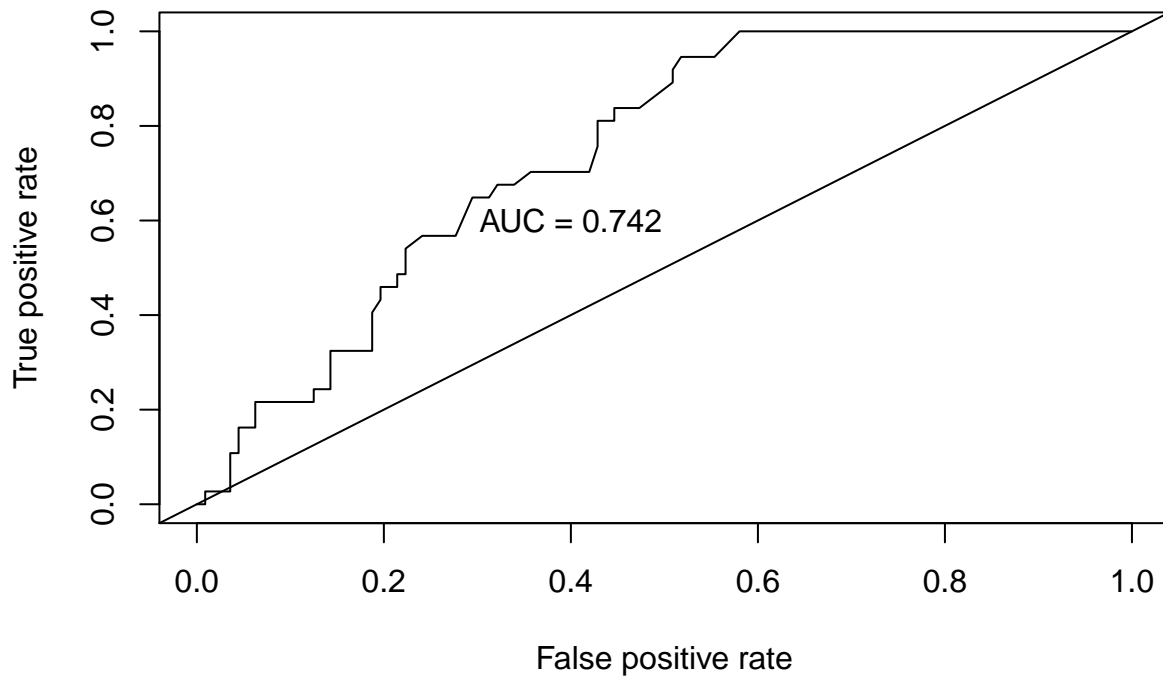
Random Forest Validation Data Set


```r
#confusion matrix Test
pdata_logical <-  rf.fit.pred.test[,2] > 0.5
confusionMatrix(data = as.factor(as.numeric(pdata_logical)), reference = as.factor(as.numeric(validation
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 111  37
##          1   1   0
##
##                Accuracy : 0.745
##                  95% CI : (0.6672, 0.8128)
##     No Information Rate : 0.7517
##     P-Value [Acc > NIR] : 0.6175
##
##                   Kappa : -0.0132
##  Mcnemar's Test P-Value : 1.365e-08
##
##             Sensitivity : 0.9911
##             Specificity : 0.0000
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.0000
##              Prevalence : 0.7517
##          Detection Rate : 0.7450
##    Detection Prevalence : 0.9933
```

```
##       Balanced Accuracy : 0.4955
##
##         'Positive' Class : 0
##
```

```
#confusion matrix Test Lower Cutoff
pdata_logical_lowercf <-  rf.fit.pred.test[,2] >= 0.3
confusionMatrix(data = as.factor(as.numeric(pdata_logical_lowercf)), reference = as.factor(as.numeric(va
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 102  29
##          1  10   8
##
##                Accuracy : 0.7383
##                  95% CI : (0.66, 0.8068)
##     No Information Rate : 0.7517
##     P-Value [Acc > NIR] : 0.686582
##
##                   Kappa : 0.1533
##  Mcnemar's Test P-Value : 0.003948
##
##             Sensitivity : 0.9107
##             Specificity : 0.2162
##          Pos Pred Value : 0.7786
##          Neg Pred Value : 0.4444
##              Prevalence : 0.7517
##          Detection Rate : 0.6846
##    Detection Prevalence : 0.8792
##       Balanced Accuracy : 0.5635
##
##         'Positive' Class : 0
##
```

```
varImpPlot(rf.fit)
```

rf.fit
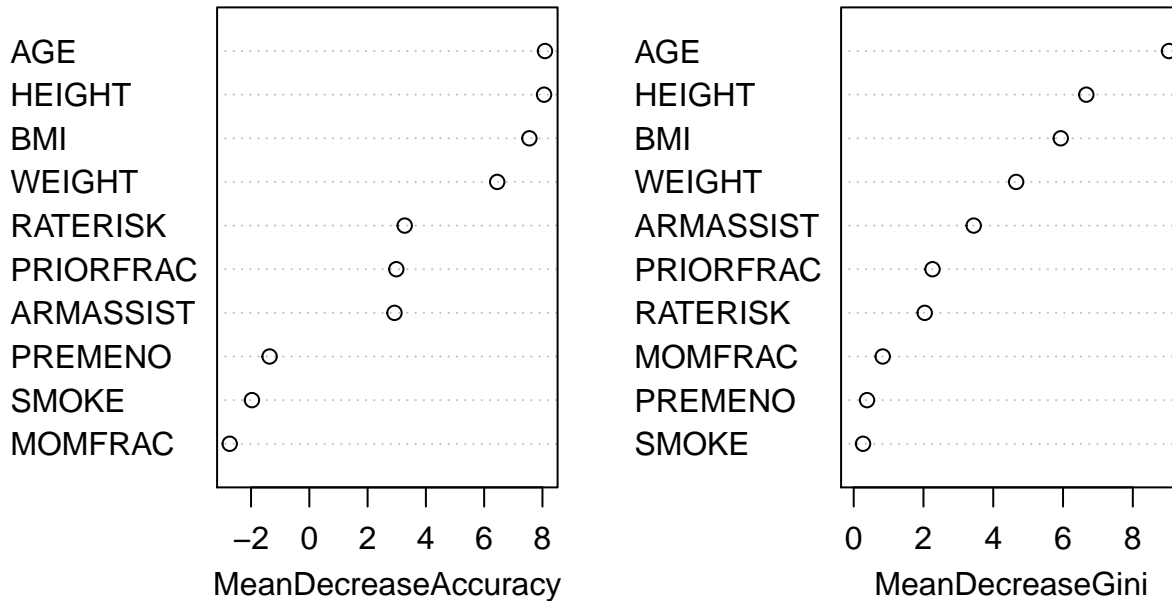
Running Conditional Random Forest Fit

```
set.seed(999)

crf.fit <- cforest(FRACTURE ~ ., data=trainingData, control=cforest_unbiased(ntree=500))
crf.fit
```

```
##
##   Random Forest using Conditional Inference Trees
##
## Number of trees:  500
##
## Response:  FRACTURE
## Inputs:  PRIORFRAC, AGE, WEIGHT, HEIGHT, BMI, PREMENO, MOMFRAC, ARMASSIST, SMOKE, RATERISK
## Number of observations:  351
```

```
crf.fit.pred.train <- predict(crf.fit, newdata=trainingData, OOB = TRUE, type="prob")
unlist.Pred.train <- matrix(unlist(crf.fit.pred.train), ncol=2,  byrow = TRUE)
pred.train.crf <- prediction(unlist.Pred.train[,2], trainingData$FRACTURE)
roc.perf = performance(pred.train.crf, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.train.crf, measure = "auc")
auc.train <- auc.train@y.values
plot(roc.perf, main="Conditional Random Forest Training Data Set")
```

```r
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```

Conditional Random Forest Training Data Set
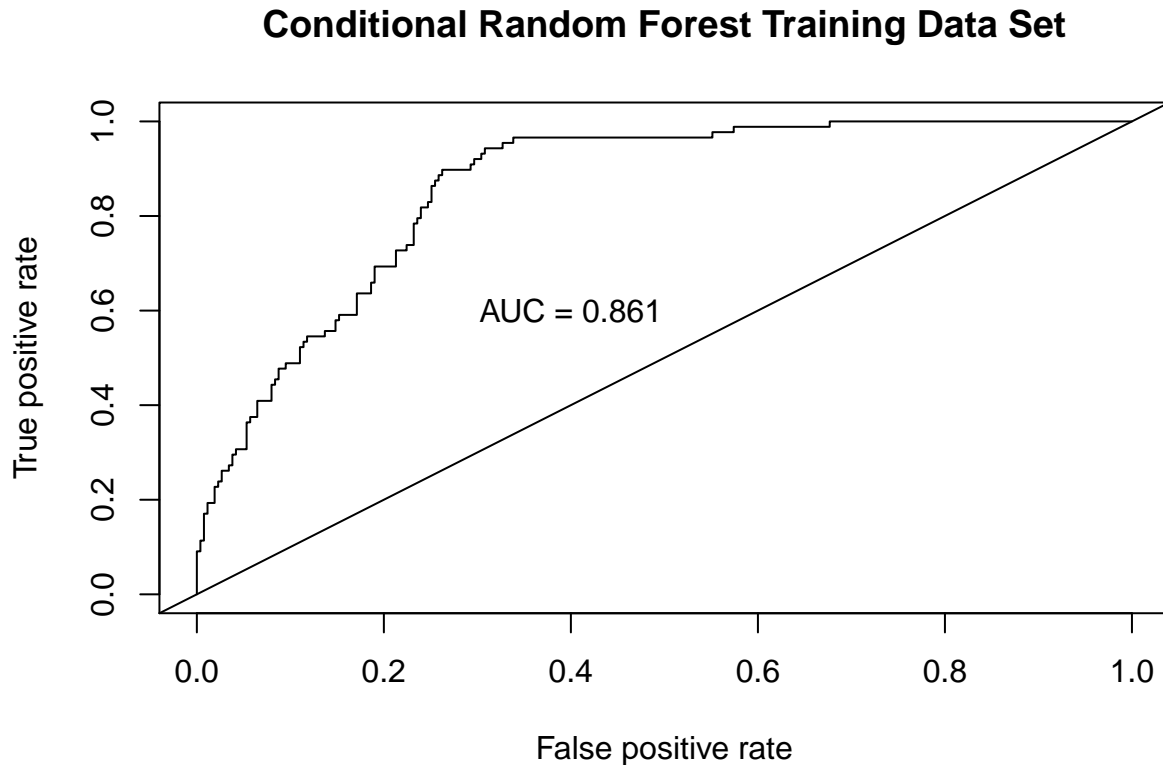

```r
#confusion matrix Training
pdata_logical_train <-  (unlist.Pred.train[,2] >= 0.5)
confusionMatrix(data = as.factor(as.numeric(pdata_logical_train)), reference = as.factor(as.numeric(trai
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 258  70
##          1   5  18
##
##                Accuracy : 0.7863
##                  95% CI : (0.7397, 0.8281)
##     No Information Rate : 0.7493
##     P-Value [Acc > NIR] : 0.06
##
##                   Kappa : 0.246
##  Mcnemar's Test P-Value : 1.467e-13
##
##             Sensitivity : 0.9810
##             Specificity : 0.2045
##          Pos Pred Value : 0.7866
```

```
##          Neg Pred Value : 0.7826
##               Prevalence : 0.7493
##          Detection Rate : 0.7350
##    Detection Prevalence : 0.9345
##       Balanced Accuracy : 0.5928
##
##          'Positive' Class : 0
##
```

```r
crf.fit.pred.test <- predict(crf.fit, newdata=validationData, OOB = T, type="prob")
unlist.Pred.test <- matrix(unlist(crf.fit.pred.test), ncol=2,  byrow = TRUE)
pred.test.crf <- prediction(unlist.Pred.test[,2], validationData$FRACTURE)
roc.perf = performance(pred.test.crf, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.test.crf, measure = "auc")
auc.train <- auc.train@y.values
plot(roc.perf, main="Conditional Random Forest Validation Data Set")
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```

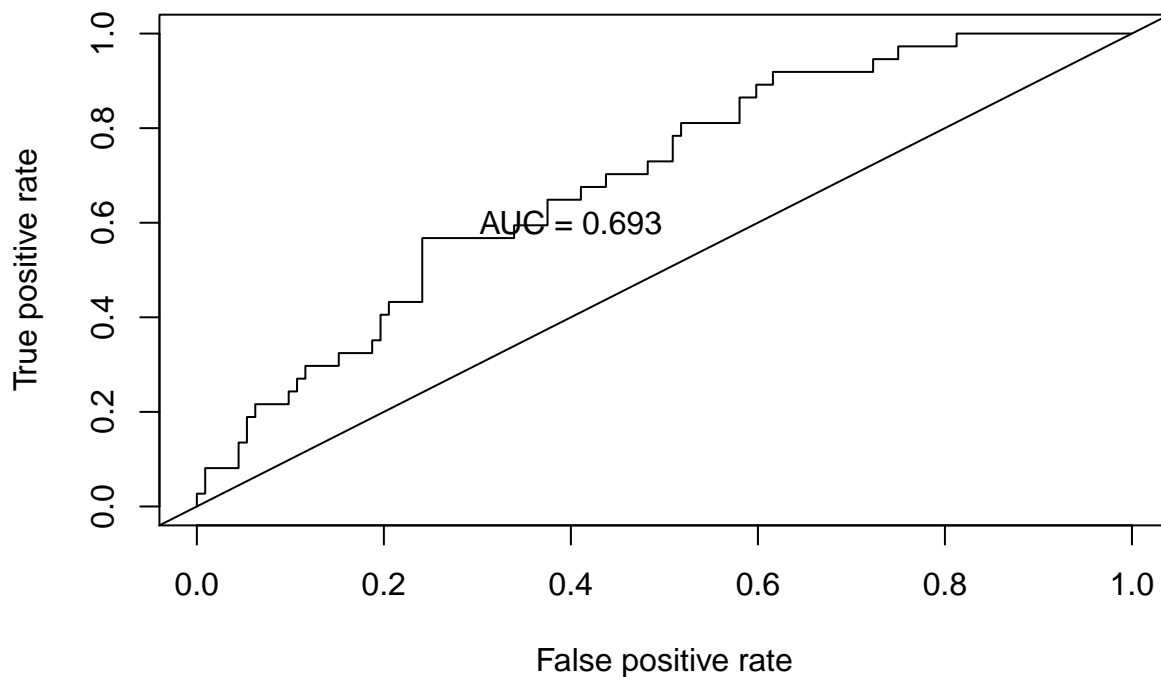## Conditional Random Forest Validation Data Set


```r
#confusion matrix
pdata_logical <-   unlist.Pred.test[,2] > 0.5
confusionMatrix(data = as.factor(as.numeric(pdata_logical)), reference = as.factor(as.numeric(validation
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction   0   1
##          0 106  31
##          1   6   6
##
##                 Accuracy : 0.7517
##                   95% CI : (0.6743, 0.8187)
##      No Information Rate : 0.7517
##      P-Value [Acc > NIR] : 0.544
##
##                    Kappa : 0.1403
##   Mcnemar's Test P-Value : 7.961e-05
##
##              Sensitivity : 0.9464
##              Specificity : 0.1622
##           Pos Pred Value : 0.7737
##           Neg Pred Value : 0.5000
##               Prevalence : 0.7517
##           Detection Rate : 0.7114
##     Detection Prevalence : 0.9195
##        Balanced Accuracy : 0.5543
##
##         'Positive' Class : 0
##
```

```
relativeImp <- varimp(crf.fit)
sort(relativeImp, decreasing = T)
```

```
##          AGE        HEIGHT     ARMASSIST           BMI     PRIORFRAC
##  8.372093e-03  7.581395e-03  6.124031e-03  1.674419e-03  4.496124e-04
##       WEIGHT      RATERISK         SMOKE       PREMENO       MOMFRAC
##  2.790698e-04 -7.751938e-05 -7.751938e-05 -3.255814e-04 -7.751938e-04
```

LDA AND QDA Model fit

```
library(MASS)
library(gridExtra)

## Assumption of Eq Variance / CoVariance
box.AGE <- ggplot(dataset, aes(x = FRACTURE, y = AGE, col = FRACTURE, fill = FRACTURE)) +
  geom_boxplot(alpha = 0.2) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("blue", "red")) +
  scale_fill_manual(values = c("blue", "red"))

box.HEIGHT <- ggplot(dataset, aes(x = FRACTURE, y = HEIGHT, col = FRACTURE, fill = FRACTURE)) +
  geom_boxplot(alpha = 0.2) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("blue", "red")) +
  scale_fill_manual(values = c("blue", "red"))
```

```
box.WEIGHT <- ggplot(dataset, aes(x = FRACTURE, y = WEIGHT, col = FRACTURE, fill = FRACTURE)) +
  geom_boxplot(alpha = 0.2) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("blue", "red")) +
  scale_fill_manual(values = c("blue", "red"))

box.BMI <- ggplot(dataset, aes(x = FRACTURE, y = BMI, col = FRACTURE, fill = FRACTURE)) +
  geom_boxplot(alpha = 0.2) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("blue", "red")) +
  scale_fill_manual(values = c("blue", "red"))

grid.arrange(box.AGE, box.HEIGHT, box.WEIGHT, box.BMI, nrow = 2, ncol = 2)
```
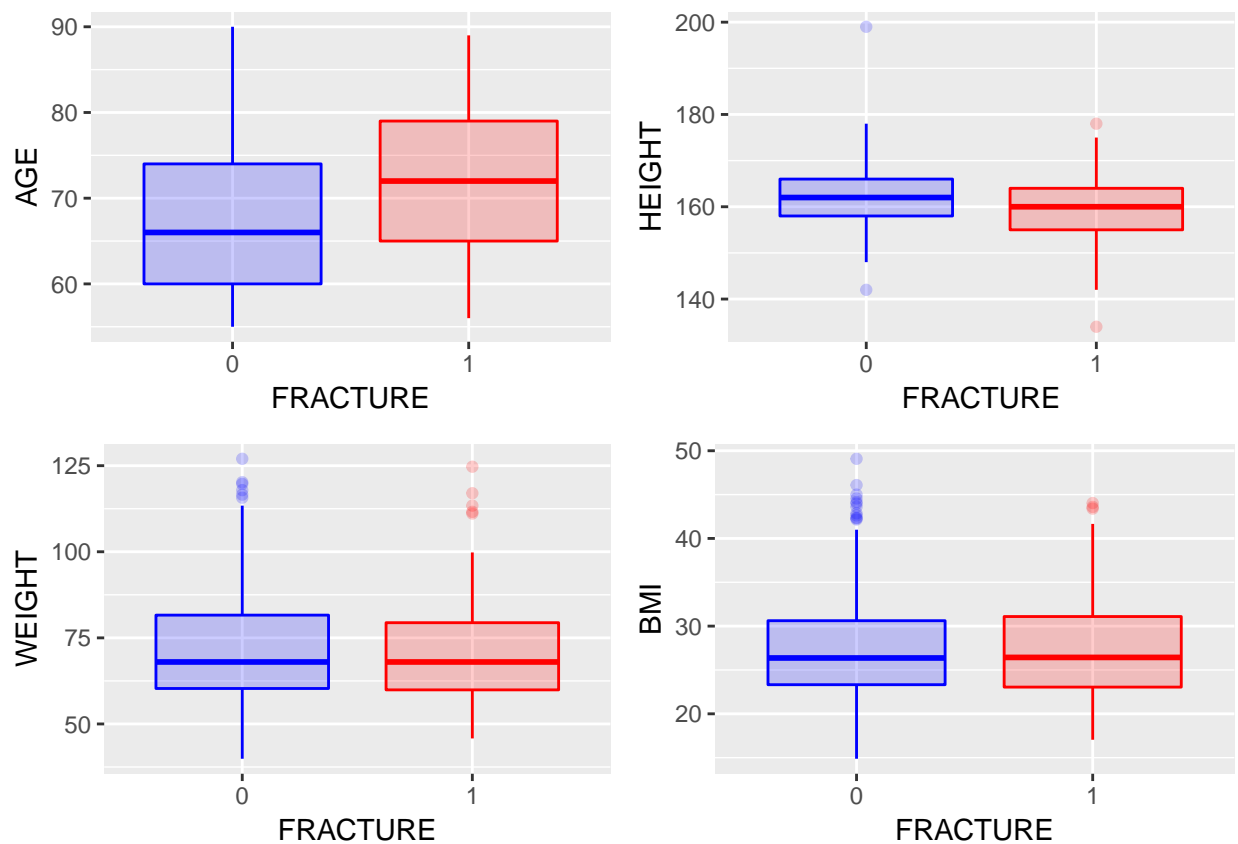


```
covEllipses(dataset[,c(5:8)], dataset$FRACTURE, fill = TRUE, pooled = FALSE,  col = c("blue", "red"), va
```
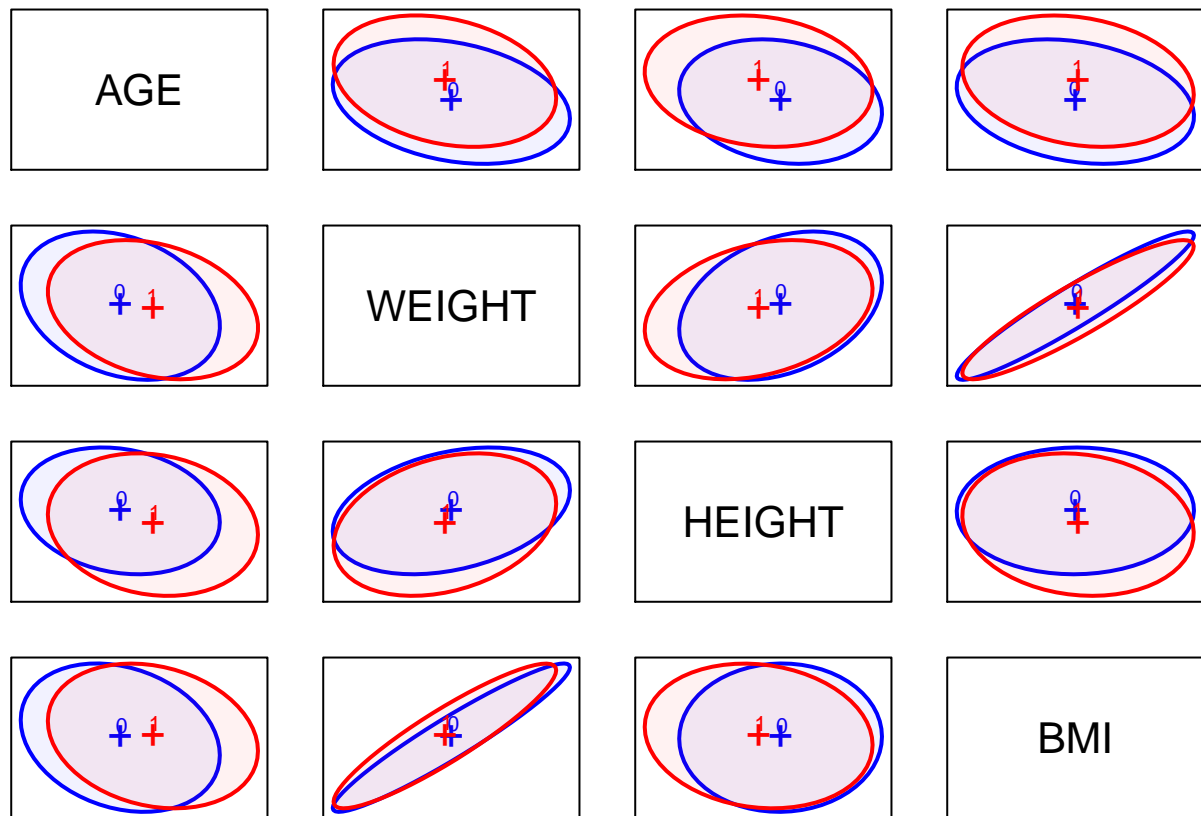
```
#
# Conducting Levene Test
leveneTest(AGE ~ FRACTURE, dataset) # Came Back Not Significant
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1   1.522 0.2179
##       498
```

```
leveneTest(HEIGHT ~ FRACTURE, dataset)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.9566 0.3285
##       498
```

```
leveneTest(WEIGHT ~ FRACTURE, dataset)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.9475 0.3308
##       498
```

```r
leveneTest(BMI ~ FRACTURE, dataset)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.0188 0.8911
##        498
```

```r
# Came Back Not Significant, Confirms findings from previous plots

density.AGE <- ggplot(dataset, aes(x = AGE, y = ..density.., col = FRACTURE)) +
  geom_density(aes(y = ..density..)) +
  scale_color_manual(values = c("blue", "red")) +
  theme(legend.position = "none")

density.HEIGHT <- ggplot(dataset, aes(x = HEIGHT, y = ..density.., col = FRACTURE)) +
  geom_density(aes(y = ..density..)) +
  scale_color_manual(values = c("blue", "red")) +
  theme(legend.position = "none")

density.WEIGHT <- ggplot(dataset, aes(x = WEIGHT, y = ..density.., col = FRACTURE)) +
  geom_density(aes(y = ..density..)) +
  scale_color_manual(values = c("blue", "red")) +
  theme(legend.position = "none")

density.BMI <- ggplot(dataset, aes(x = BMI, y = ..density.., col = FRACTURE)) +
  geom_density(aes(y = ..density..)) +
  scale_color_manual(values = c("blue", "red")) +
  theme(legend.position = "none")

grid.arrange(density.AGE, density.HEIGHT, density.WEIGHT, density.BMI, nrow = 2, ncol = 2)
```
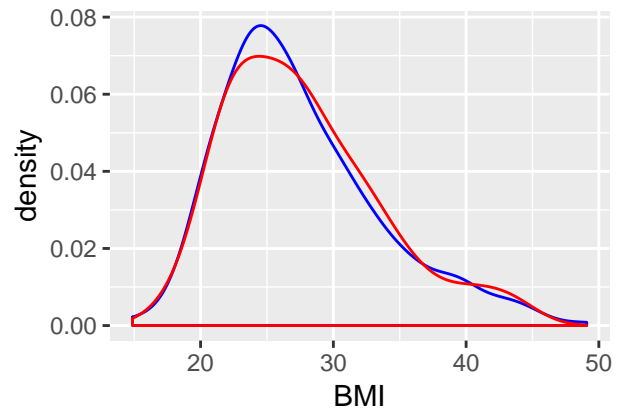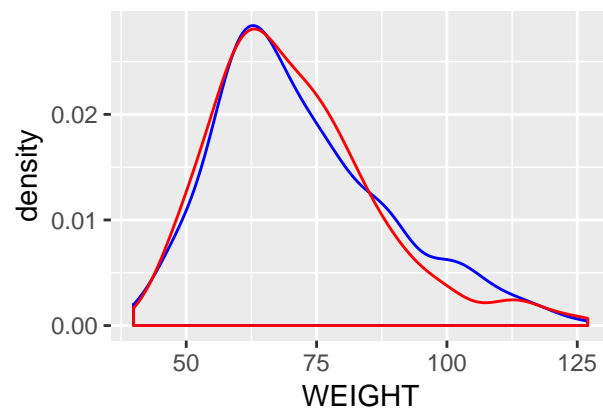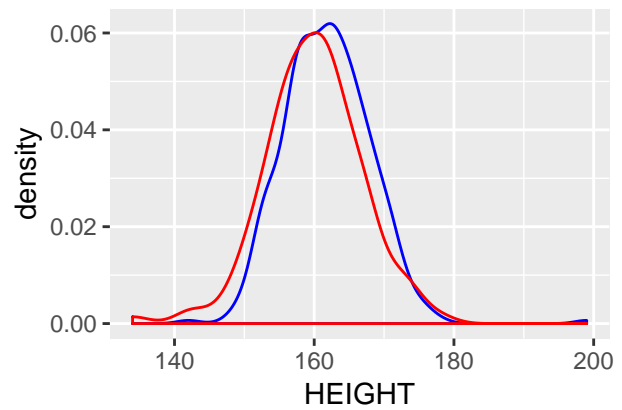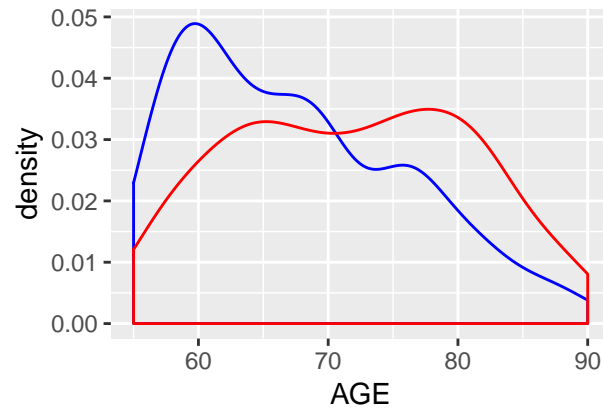
```r
# Check QQ Plot for AGE to ascertain Normality in BOTH Groups
frac.yes <- subset(dataset, FRACTURE == 1)
frac.no <- subset(dataset, FRACTURE == 0)
# Plot
qqnorm(frac.yes$AGE, main = "Distribution of AGE in Fracture=Yes Group"); qqline(frac.yes$AGE, col = 2)
```

Distribution of AGE in Fracture=Yes Group
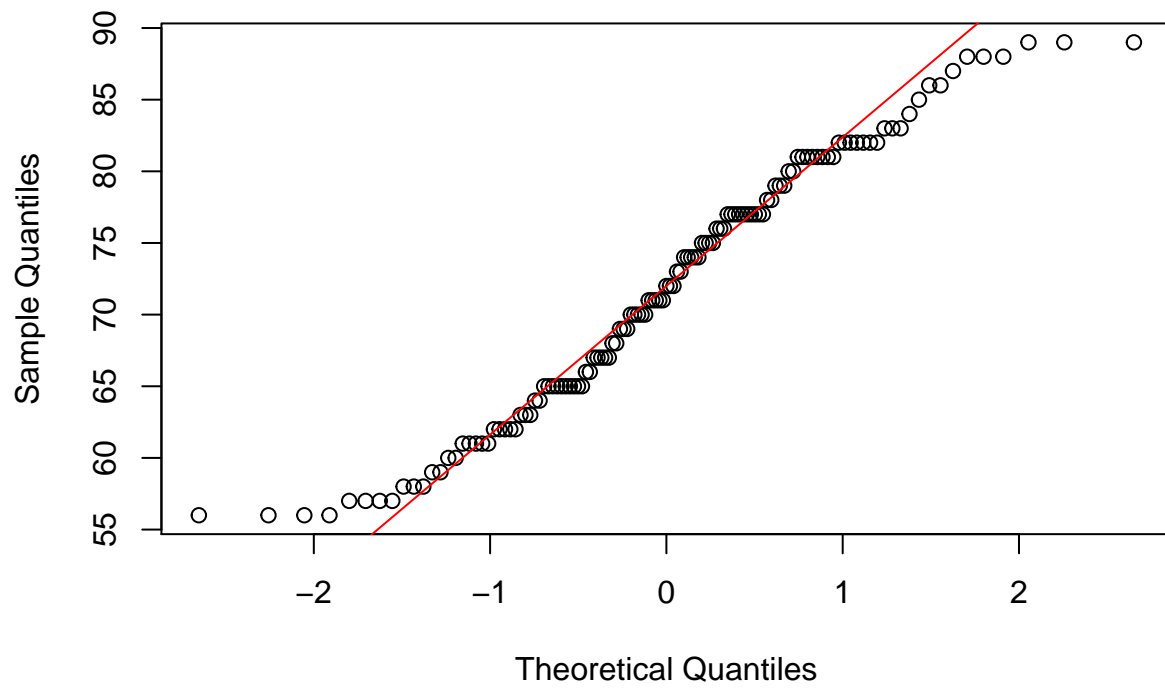

```
qqnorm(frac.no$AGE, main = "Distribution of AGE in Fracture=No Group"); qqline(frac.no$AGE, col = 2)
```

Distribution of AGE in Fracture=No Group
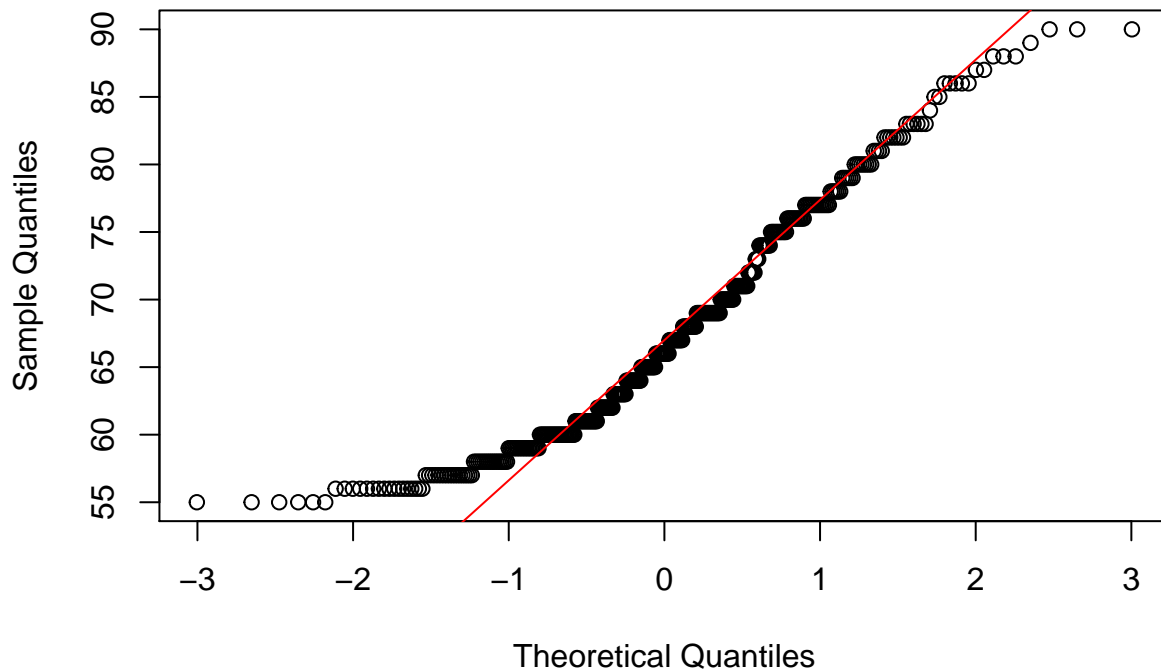

```
## Assumptions for Normality and of Equal Variance-Coavariance matrices Are Successfully Met.
## Run the LDA Now

set.seed(999)

lda.fit <- lda(FRACTURE ~ AGE + HEIGHT + WEIGHT + BMI, data = trainingData)
lda.fit
```

```
## Call:
## lda(FRACTURE ~ AGE + HEIGHT + WEIGHT + BMI, data = trainingData)
##
## Prior probabilities of groups:
##         0         1
## 0.7492877 0.2507123
##
## Group means:
##        AGE   HEIGHT   WEIGHT      BMI
## 0 67.16730 162.2129 72.01559 27.31879
## 1 71.95455 159.7614 70.53409 27.69421
##
## Coefficients of linear discriminants:
##              LD1
## AGE     0.08790497
## HEIGHT  0.20784982
## WEIGHT -0.31576637
## BMI     0.86125425
```

```
#ROC on training data set
ldaprd <- predict(lda.fit, newdata = trainingData)$posterior
ldaprd <- ldaprd[,2]
pred.train <- prediction(ldaprd, trainingData$FRACTURE)
roc.perf = performance(pred.train, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.train, measure = "auc")
auc.train <- auc.train@y.values

#Plot ROC on Training Data
plot(roc.perf,main="LDA Training Data Set")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```

LDA Training Data Set


```
prd <- predict(lda.fit, newdata = trainingData)$class
confusionMatrix(data = prd, reference = trainingData$FRACTURE)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 258  76
##          1   5  12
##
##               Accuracy : 0.7692
##                 95% CI : (0.7216, 0.8123)
```

```
##       No Information Rate : 0.7493
##       P-Value [Acc > NIR] : 0.2128
##
##                     Kappa : 0.1604
##  Mcnemar's Test P-Value : 7.381e-15
##
##               Sensitivity : 0.9810
##               Specificity : 0.1364
##            Pos Pred Value : 0.7725
##            Neg Pred Value : 0.7059
##                Prevalence : 0.7493
##            Detection Rate : 0.7350
##      Detection Prevalence : 0.9516
##         Balanced Accuracy : 0.5587
##
##          'Positive' Class : 0
##
```

```r
#ROC on test data set
ldaprd.test <- predict(lda.fit, newdata = validationData)$posterior
ldaprd.test <- ldaprd.test[,2]
pred.test <- prediction(ldaprd.test, validationData$FRACTURE)
roc.perf = performance(pred.test, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.test, measure = "auc")
auc.train <- auc.train@y.values

#Plot ROC on Training Data
plot(roc.perf,main="LDA Validation Data Set")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```

LDA Validation Data Set
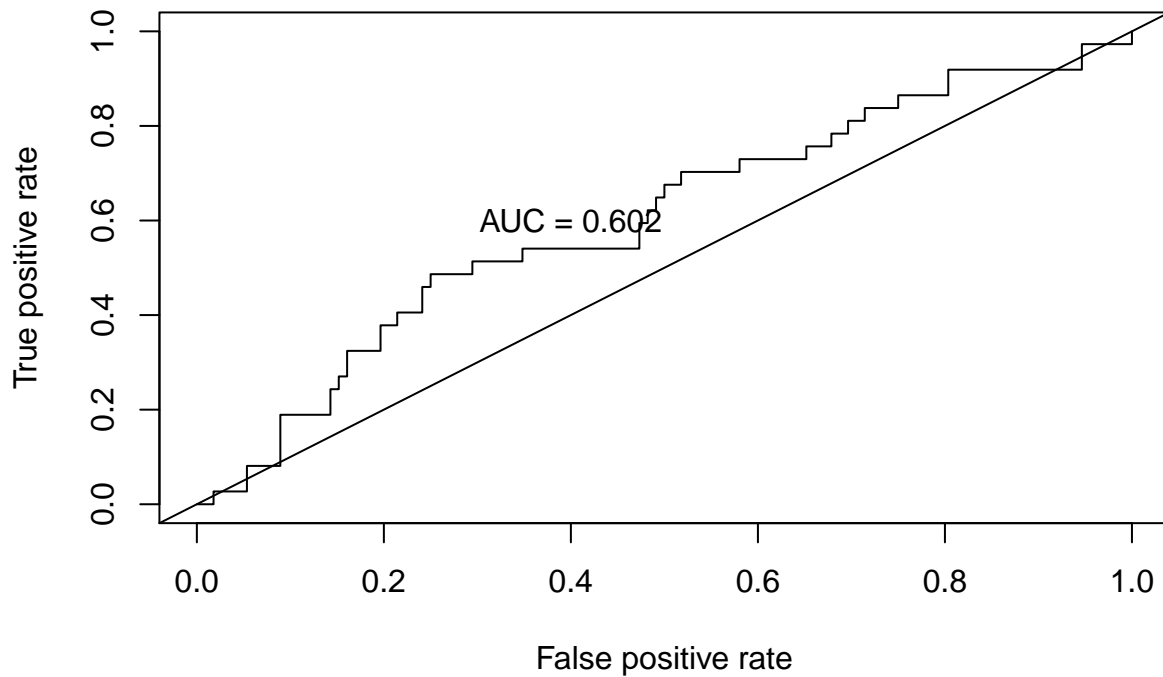

```
prd.test <- predict(lda.fit, newdata = validationData)$class
confusionMatrix(data = prd.test, reference = validationData$FRACTURE)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 106  34
##          1   6   3
##
##                Accuracy : 0.7315
##                  95% CI : (0.6529, 0.8008)
##     No Information Rate : 0.7517
##     P-Value [Acc > NIR] : 0.7493
##
##                   Kappa : 0.0368
##  Mcnemar's Test P-Value : 1.963e-05
##
##             Sensitivity : 0.94643
##             Specificity : 0.08108
##          Pos Pred Value : 0.75714
##          Neg Pred Value : 0.33333
##              Prevalence : 0.75168
##          Detection Rate : 0.71141
##    Detection Prevalence : 0.93960
```

```
##      Balanced Accuracy : 0.51375
##
##      'Positive' Class : 0
##
```

```
## Running QDA to see if it improves AUC

qda.fit <- qda(FRACTURE ~ AGE + HEIGHT + WEIGHT + BMI, data = trainingData)

#ROC on training data set
qdaprd <- predict(qda.fit, newdata = trainingData)$posterior
qdaprd <- qdaprd[,2]
pred.train <- prediction(qdaprd, trainingData$FRACTURE)
roc.perf = performance(pred.train, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.train, measure = "auc")
auc.train <- auc.train@y.values

#Plot ROC on Training Data
plot(roc.perf,main="QDA Training Data Set")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```
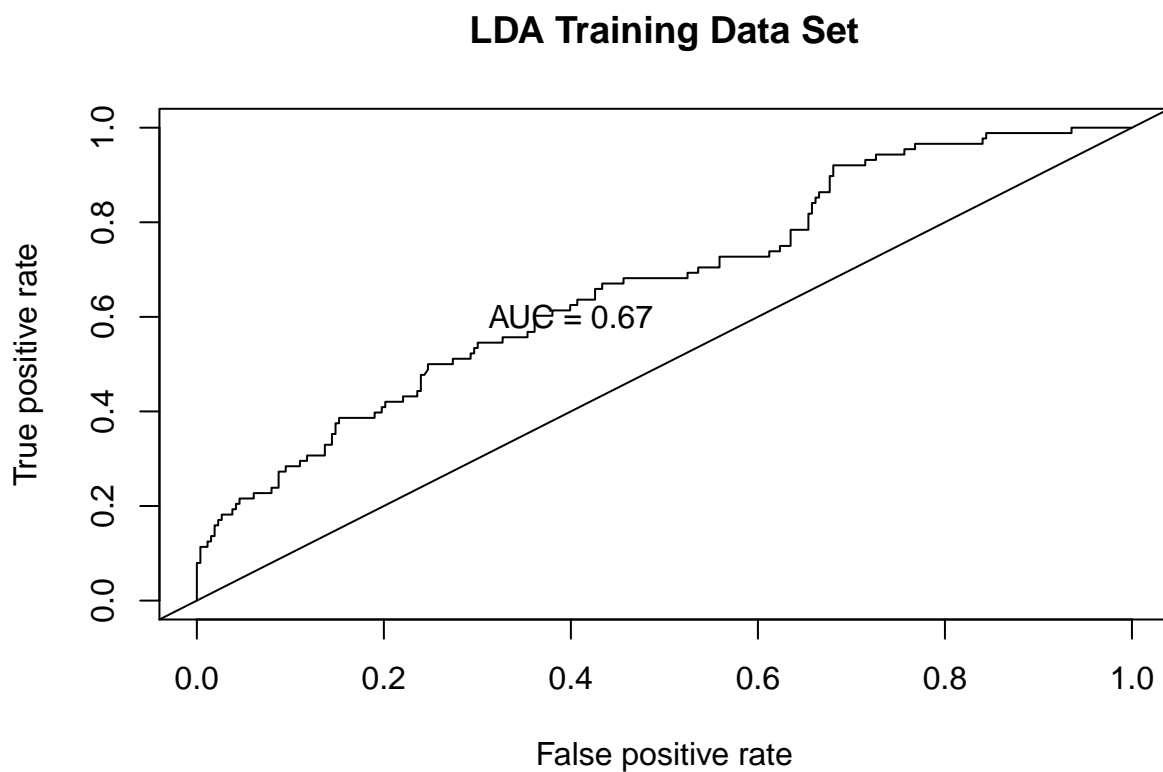
QDA Training Data Set


```
prd <- predict(qda.fit, newdata = trainingData)$class
confusionMatrix(data = prd, reference = trainingData$FRACTURE)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction   0   1
##          0 249  71
##          1  14  17
##
##                 Accuracy : 0.7578
##                   95% CI : (0.7095, 0.8017)
##      No Information Rate : 0.7493
##      P-Value [Acc > NIR] : 0.3827
##
##                    Kappa : 0.1784
##  Mcnemar's Test P-Value : 1.247e-09
##
##              Sensitivity : 0.9468
##              Specificity : 0.1932
##           Pos Pred Value : 0.7781
##           Neg Pred Value : 0.5484
##               Prevalence : 0.7493
##           Detection Rate : 0.7094
##     Detection Prevalence : 0.9117
##        Balanced Accuracy : 0.5700
##
##         'Positive' Class : 0
##
```

```r
#ROC on test data set
qdaprd.test <- predict(qda.fit, newdata = validationData)$posterior
qdaprd.test <- qdaprd.test[,2]
pred.test <- prediction(qdaprd.test, validationData$FRACTURE)
roc.perf = performance(pred.test, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred.test, measure = "auc")
auc.train <- auc.train@y.values

#Plot ROC on Training Data
plot(roc.perf,main="QDA Validation Data Set")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```

QDA Validation Data Set
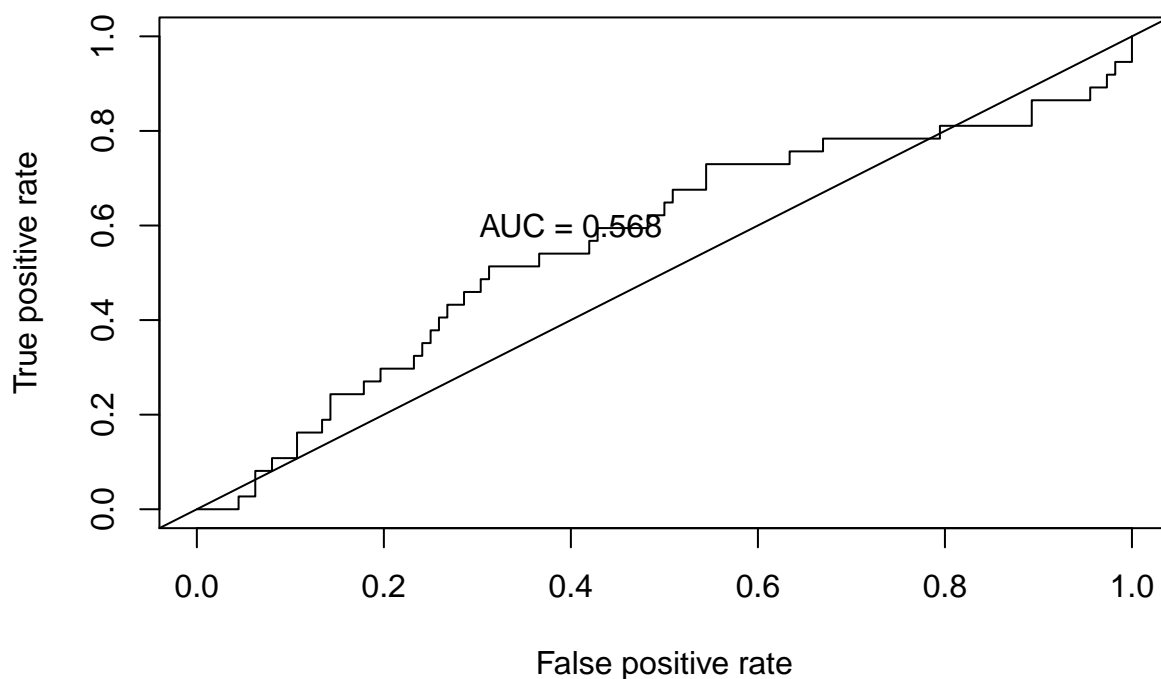

```
prd.test <- predict(qda.fit, newdata = validationData)$class
confusionMatrix(data = prd.test, reference = validationData$FRACTURE)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 103  33
##          1   9   4
##
##                Accuracy : 0.7181
##                  95% CI : (0.6387, 0.7887)
##     No Information Rate : 0.7517
##     P-Value [Acc > NIR] : 0.8512971
##
##                   Kappa : 0.0355
##  Mcnemar's Test P-Value : 0.0003867
##
##             Sensitivity : 0.9196
##             Specificity : 0.1081
##          Pos Pred Value : 0.7574
##          Neg Pred Value : 0.3077
##              Prevalence : 0.7517
##          Detection Rate : 0.6913
##    Detection Prevalence : 0.9128
```
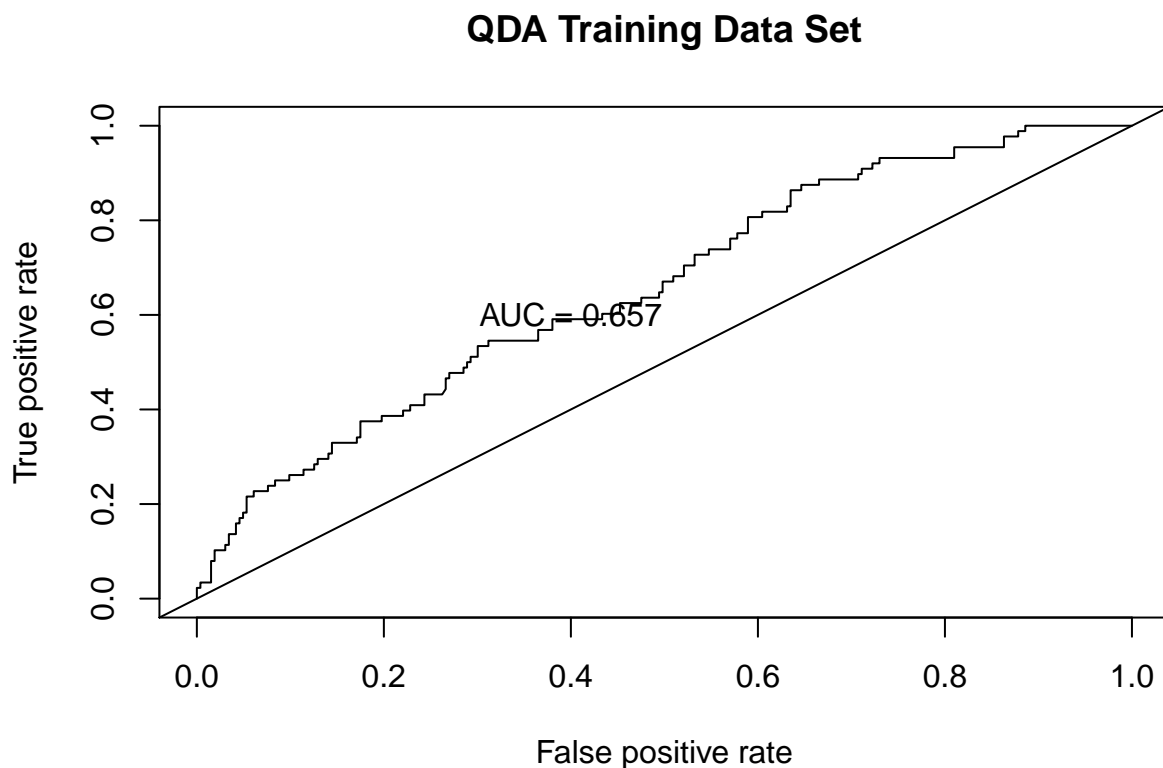
```
##          Balanced Accuracy : 0.5139
##
##          'Positive' Class : 0
##
```

```
frac.yes <- subset(dataset, FRACTURE == 1)
frac.no <- subset(dataset, FRACTURE == 0)

box.Prior.Age.Frac.Yes <- ggplot(frac.yes, aes(x = PRIORFRAC, y = AGE, col = PRIORFRAC, fill = PRIORFRAC
  geom_boxplot(alpha = 0.2) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("blue", "red")) +
  scale_fill_manual(values = c("blue", "red")) +
  ggtitle("FRACTURE = YES GROUP")

box.Prior.Age.Frac.No <- ggplot(frac.no, aes(x = PRIORFRAC, y = AGE, col = PRIORFRAC, fill = PRIORFRAC )
  geom_boxplot(alpha = 0.2) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("blue", "red")) +
  scale_fill_manual(values = c("blue", "red"))+
   ggtitle("FRACTURE = NO GROUP")

grid.arrange(box.Prior.Age.Frac.No, box.Prior.Age.Frac.Yes, nrow = 1, ncol = 2)
```
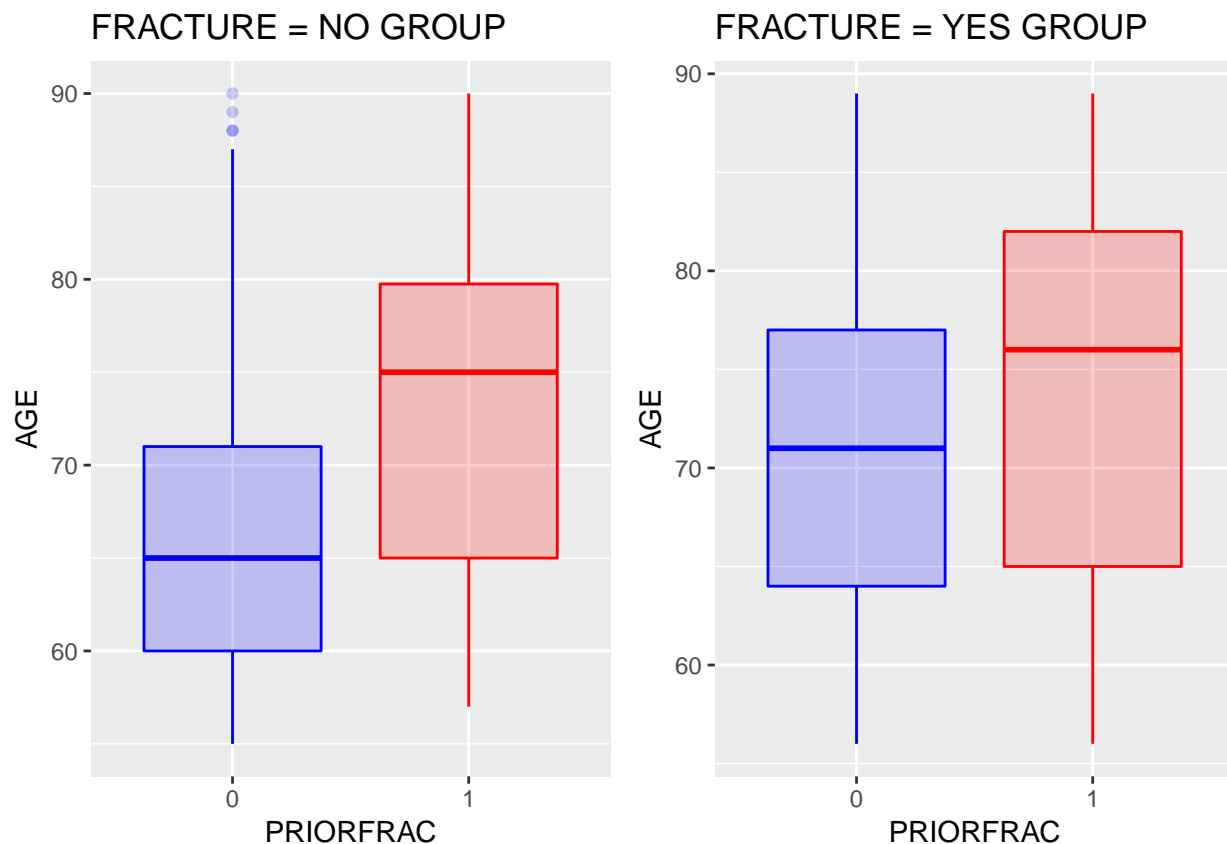
```
par(mfrow = c(1, 2))
mosplot.Frac.No <- mosaicplot(CrossTable(frac.no$MOMFRAC, frac.no$ARMASSIST)$t, main = "FRACTURE = NO GR
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  375
##
##
##                 | frac.no$ARMASSIST
## frac.no$MOMFRAC |         0 |         1 | Row Total |
## ----------------|-----------|-----------|-----------|
##               0 |       225 |       109 |       334 |
##                 |     0.024 |     0.049 |           |
##                 |     0.674 |     0.326 |     0.891 |
##                 |     0.900 |     0.872 |           |
##                 |     0.600 |     0.291 |           |
## ----------------|-----------|-----------|-----------|
##               1 |        25 |        16 |        41 |
##                 |     0.199 |     0.398 |           |
##                 |     0.610 |     0.390 |     0.109 |
##                 |     0.100 |     0.128 |           |
##                 |     0.067 |     0.043 |           |
## ----------------|-----------|-----------|-----------|
##    Column Total |       250 |       125 |       375 |
##                 |     0.667 |     0.333 |           |
## ----------------|-----------|-----------|-----------|
##
##
```

```
mosplot.Frac.Yes <- mosaicplot(CrossTable(frac.yes$MOMFRAC, frac.yes$ARMASSIST)$t, main = "FRACTURE = Y
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
```

```
## |-------------------------|
##
##
## Total Observations in Table:   125
##
##
##                 | frac.yes$ARMASSIST
## frac.yes$MOMFRAC |          0 |          1 | Row Total |
## ----------------|-----------|-----------|-----------|
##               0 |        47 |        54 |       101 |
##                 |     0.191 |     0.188 |           |
##                 |     0.465 |     0.535 |     0.808 |
##                 |     0.758 |     0.857 |           |
##                 |     0.376 |     0.432 |           |
## ----------------|-----------|-----------|-----------|
##               1 |        15 |         9 |        24 |
##                 |     0.805 |     0.792 |           |
##                 |     0.625 |     0.375 |     0.192 |
##                 |     0.242 |     0.143 |           |
##                 |     0.120 |     0.072 |           |
## ----------------|-----------|-----------|-----------|
##    Column Total |        62 |        63 |       125 |
##                 |     0.496 |     0.504 |           |
## ----------------|-----------|-----------|-----------|
##
##
```

FRACTURE = NO GROUP

FRACTURE = YES GROUP

*** *Appendix C:* Test interaction - LDA ================

Get Glow dataset

```
glow <- read_glow_dataset()
```

model interactions - main effects

```
model_z1 <- glm(FRACTURE ~ AGE, family = binomial, data = glow)
model_z2 <- glm(FRACTURE ~ WEIGHT, family = binomial, data = glow)
model_z3 <- glm(FRACTURE ~ HEIGHT, family = binomial, data = glow)
model_z4 <- glm(FRACTURE ~ BMI, family = binomial, data = glow)
model_z5 <- glm(FRACTURE ~ PRIORFRAC, family = binomial, data = glow)
model_z6 <- glm(FRACTURE ~ PREMENO, family = binomial, data = glow)
model_z7 <- glm(FRACTURE ~ MOMFRAC, family = binomial, data = glow)
model_z8 <- glm(FRACTURE ~ ARMASSIST, family = binomial, data = glow)
model_z9 <- glm(FRACTURE ~ SMOKE, family = binomial, data = glow)
model_z10 <- glm(FRACTURE ~ RATERISK, family = binomial, data = glow)
```

```
## AGE            0.05289    0.01163   4.548 5.42e-06 ***
## WEIGHT        -0.005197   0.006415 -0.810    0.418
## HEIGHT        -0.05167    0.01709  -3.022  0.00251 **
## BMI            0.005758   0.017185  0.335  0.73760
## PRIORFRACYes 1.0638      0.2231     4.769 1.85e-06 ***
## PREMENOYes    0.05077    0.25921   0.196    0.845
## MOMFRACYes    0.6605     0.2810     2.351   0.0187 *
## ARMASSISTYes 0.7091      0.2098     3.381 0.000723 ***
## SMOKEYes     -0.3077     0.4358    -0.706    0.48
## RATERISKSame    0.5462      0.2664   2.050   0.0404 *
## RATERISKGreater  0.9091      0.2711   3.353   0.0008 ***
```

```
> code below:
```

This leads us to consider the covariates above that are significant in the univariate results above at the 25% level.

AGE, HEIGHT, PRIORFRAC, MOMFRAC, ARMASSIST, RATERISK {SAME, GREATER}

```
# fit a univariate logistic regression model for each covariate
# continuous - AGE WEIGHT HEIGHT BMI
# categorical - PRIORFRAC PREMENO MOMFRAC ARMASSIST SMOKE RATERISK

# model0
#model_z1 <- glm(FRACTURE ~ AGE, family = binomial, data = glow)
#model_z2 <- glm(FRACTURE ~ WEIGHT, family = binomial, data = glow)
#model_z3 <- glm(FRACTURE ~ HEIGHT, family = binomial, data = glow)
#model_z4 <- glm(FRACTURE ~ BMI, family = binomial, data = glow)
#model_z5 <- glm(FRACTURE ~ PRIORFRAC, family = binomial, data = glow)
#model_z6 <- glm(FRACTURE ~ PREMENO, family = binomial, data = glow)
#model_z7 <- glm(FRACTURE ~ MOMFRAC, family = binomial, data = glow)
#model_z8 <- glm(FRACTURE ~ ARMASSIST, family = binomial, data = glow)
#model_z9 <- glm(FRACTURE ~ SMOKE, family = binomial, data = glow)
```

```
#model_z10 <- glm(FRACTURE ~ RATERISK, family = binomial, data = glow)

#summary(model_z1)
#summary(model_z2)
#summary(model_z3)
#summary(model_z4)
#summary(model_z5)
#summary(model_z6)
#summary(model_z7)
#summary(model_z8)
#summary(model_z9)
#summary(model_z10)

# not interesting due to all variables (i.e. SUB_ID, SITE_ID, PHY_ID)
# model00 <- glm(FRACTURE ~ ., family = binomial, data = glow)
# summary(model00)

# full model, order by continuous, then factor
model0 <- glm(FRACTURE ~ AGE + WEIGHT + HEIGHT + BMI + PRIORFRAC + PREMENO + MOMFRAC + ARMASSIST + SMOKI
summary(model0)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + WEIGHT + HEIGHT + BMI + PRIORFRAC +
##     PREMENO + MOMFRAC + ARMASSIST + SMOKE + RATERISK, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6811  -0.7228  -0.5639  -0.1008   2.2182
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -15.74709   12.67053  -1.243  0.21394
## AGE              0.03895    0.01476   2.640  0.00829 **
## WEIGHT          -0.12189    0.08664  -1.407  0.15949
## HEIGHT           0.06620    0.07825   0.846  0.39755
## BMI              0.33181    0.22339   1.485  0.13745
## PRIORFRACYes     0.67577    0.25012   2.702  0.00690 **
## PREMENOYes       0.10080    0.28540   0.353  0.72395
## MOMFRACYes       0.63438    0.30784   2.061  0.03933 *
## ARMASSISTYes     0.36102    0.25647   1.408  0.15924
## SMOKEYes        -0.31228    0.46216  -0.676  0.49923
## RATERISKSame     0.42256    0.28144   1.501  0.13324
## RATERISKGreater  0.75645    0.29944   2.526  0.01153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 503.84  on 488  degrees of freedom
## AIC: 527.84
```

86

```
##
## Number of Fisher Scoring iterations: 4
```

```
# fit model # note - should remove below model1
model0_fitted <- update(model0, . ~ . - WEIGHT - BMI - PREMENO - SMOKE)
summary(model0_fitted)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK, family = binomial, data = glow)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.66692  -0.72502  -0.56338  -0.03841   2.22148
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.70935    3.22992   0.839  0.40157
## AGE               0.03434    0.01305   2.632  0.00848 **
## HEIGHT           -0.04383    0.01827  -2.400  0.01640 *
## PRIORFRACYes      0.64526    0.24606   2.622  0.00873 **
## MOMFRACYes        0.62122    0.30698   2.024  0.04300 *
## ARMASSISTYes      0.44579    0.23281   1.915  0.05551 .
## RATERISKSame      0.42202    0.27925   1.511  0.13071
## RATERISKGreater   0.70692    0.29342   2.409  0.01599 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 507.50  on 492  degrees of freedom
## AIC: 523.5
##
## Number of Fisher Scoring iterations: 4
```

```
# build model with following covariates (drop WEIGHT, BMI, PREMENO, SMOKE)
# AGE, HEIGHT, PRIORFRAC, MOMFRAC, ARMASSIST, RATERISK {SAME, GREATER}
```

```
model1 <- glm(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK, family = binomial,
summary(model1)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK, family = binomial, data = glow)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.66692  -0.72502  -0.56338  -0.03841   2.22148
##
## Coefficients:
```

```
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.70935    3.22992   0.839  0.40157
## AGE              0.03434    0.01305   2.632  0.00848 **
## HEIGHT          -0.04383    0.01827  -2.400  0.01640 *
## PRIORFRACYes      0.64526    0.24606   2.622  0.00873 **
## MOMFRACYes        0.62122    0.30698   2.024  0.04300 *
## ARMASSISTYes      0.44579    0.23281   1.915  0.05551 .
## RATERISKSame      0.42202    0.27925   1.511  0.13071
## RATERISKGreater   0.70692    0.29342   2.409  0.01599 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 507.50  on 492  degrees of freedom
## AIC: 523.5
##
## Number of Fisher Scoring iterations: 4
```

```
# from above result, adding back the removed covariates we see they are not needed to keep the remaining
# this becomes the model, adding back removed covariates WEIGHT, BMI, PREMENO, SMOKE the coefficients d
# this becomes the main effects model


# need to check scale of logit for remaining continous variables AGE HEIGHT
# assume HEIGHT is linearin logit
```

The main effects model

model1 <- glm(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RA-TERISK, family = binomial, data = glow)

use lrtest from package lmtest

test interactions for the following:

5. AGE: [HEIGHT, PRIORFRAC, MOMFRAC, ARMASSIST, RATERISK]
6. HEIGHT: [PRIORFRAC, MOMFRAC, ARMASSIST, RATERISK]
7. PRIORFRAC: [MOMFRAC, ARMASSIST, RATERISK]
8. MOMFRAC: [ARMASSIST, RATERISK]
9. ARMASSIST: RATERISK

total 15 interactions

```
library(lmtest)

# model AGE* , HEIGHT* , PRIORFRAC*

model_effects <- glm(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK, family = bin
lrtest(model_effects)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    8 -253.75
## 2    1 -281.17 -7 54.835  1.608e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# (5) AGE: [HEIGHT, PRIORFRAC, MOMFRAC, ARMASSIST, RATERISK]

test <- model_effects
test <- update(test, . ~ . + AGE:HEIGHT)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:HEIGHT, family = binomial, data = glow)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.66848  -0.73323  -0.56252   0.02069   2.23640
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     14.749125  23.931667   0.616   0.5377
## AGE             -0.135869   0.335087  -0.405   0.6851
## HEIGHT          -0.119095   0.149402  -0.797   0.4254
## PRIORFRACYes     0.634947   0.246751   2.573   0.0101 *
## MOMFRACYes       0.623682   0.307316   2.029   0.0424 *
## ARMASSISTYes     0.447271   0.232895   1.920   0.0548 .
## RATERISKSame     0.435127   0.280319   1.552   0.1206
## RATERISKGreater  0.707865   0.293394   2.413   0.0158 *
## AGE:HEIGHT       0.001065   0.002095   0.508   0.6113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 507.24  on 491  degrees of freedom
## AIC: 525.24
##
## Number of Fisher Scoring iterations: 4
```

```r
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     AGE:HEIGHT
```

```
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    9 -253.62
## 2    1 -281.17 -8 55.096   4.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + AGE:PRIORFRAC)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:PRIORFRAC, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48423  -0.74080  -0.53895  -0.00078  2.26588
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.63708    3.35881   0.190 0.849565
## AGE               0.05669    0.01649   3.437 0.000589 ***
## HEIGHT           -0.04058    0.01828  -2.220 0.026406 *
## PRIORFRACYes      4.85428    1.86766   2.599 0.009346 **
## MOMFRACYes        0.66973    0.30857   2.170 0.029972 *
## ARMASSISTYes      0.41887    0.23395   1.790 0.073391 .
## RATERISKSame      0.43496    0.28053   1.551 0.121014
## RATERISKGreater   0.72044    0.29561   2.437 0.014804 *
## AGE:PRIORFRACYes -0.05864    0.02583  -2.270 0.023188 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 502.34  on 491  degrees of freedom
## AIC: 520.34
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     AGE:PRIORFRAC
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    9 -251.17
## 2    1 -281.17 -8 59.991   4.679e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + AGE:MOMFRAC)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:MOMFRAC, family = binomial, data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.58376  -0.72859  -0.56182  -0.02562   2.22962
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.59055    3.24368   0.799  0.42450
## AGE              0.03633    0.01393   2.609  0.00908 **
## HEIGHT          -0.04402    0.01827  -2.409  0.01601 *
## PRIORFRACYes     0.65010    0.24630   2.639  0.00830 **
## MOMFRACYes       1.57119    2.31121   0.680  0.49662
## ARMASSISTYes     0.45447    0.23374   1.944  0.05185 .
## RATERISKSame     0.42505    0.27940   1.521  0.12819
## RATERISKGreater  0.71044    0.29363   2.420  0.01554 *
## AGE:MOMFRACYes  -0.01353    0.03264  -0.414  0.67854
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499   degrees of freedom
## Residual deviance: 507.33  on 491   degrees of freedom
## AIC: 525.33
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     AGE:MOMFRAC
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    9 -253.66
## 2    1 -281.17 -8 55.005  4.406e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + AGE:ARMASSIST)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:ARMASSIST, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min      1Q   Median       3Q      Max
## -1.6352  -0.7272  -0.5646  -0.0295   2.2329
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.33972    3.33003   0.703  0.48230
## AGE                0.03990    0.01785   2.235  0.02542 *
## HEIGHT            -0.04395    0.01827  -2.406  0.01614 *
## PRIORFRACYes       0.64031    0.24609   2.602  0.00927 **
## MOMFRACYes         0.63376    0.30795   2.058  0.03959 *
## ARMASSISTYes       1.24419    1.76410   0.705  0.48063
## RATERISKSame       0.42815    0.27964   1.531  0.12575
## RATERISKGreater    0.71996    0.29494   2.441  0.01464 *
## AGE:ARMASSISTYes  -0.01132    0.02479  -0.457  0.64802
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 507.29  on 491  degrees of freedom
## AIC: 525.29
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     AGE:ARMASSIST
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -253.65
## 2   1 -281.17 -8 55.043  4.331e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + AGE:RATERISK)
summary(test)
```

```
## 
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:RATERISK, family = binomial, data = glow)
## 
## Deviance Residuals:
##       Min        1Q    Median        3Q       Max
## -1.68668  -0.74463  -0.56590  -0.02638   2.34976
## 
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.53632    3.53444   0.152  0.87939
## AGE                0.06673    0.02473   2.698  0.00697 **
## HEIGHT            -0.04496    0.01840  -2.443  0.01456 *
## PRIORFRACYes       0.65827    0.24598   2.676  0.00745 **
## MOMFRACYes         0.65241    0.30765   2.121  0.03395 *
## ARMASSISTYes       0.48569    0.23443   2.072  0.03828 *
## RATERISKSame       3.28427    2.27575   1.443  0.14898
## RATERISKGreater    4.25804    2.28873   1.860  0.06282 .
## AGE:RATERISKSame  -0.03999    0.03151  -1.269  0.20438
## AGE:RATERISKGreater -0.05021   0.03202  -1.568  0.11690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 504.79  on 490  degrees of freedom
## AIC: 524.79
## 
## Number of Fisher Scoring iterations: 5
```

```
lrtest(test)
```

```
## Likelihood ratio test
## 
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     AGE:RATERISK
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1  10 -252.40
## 2   1 -281.17 -9 57.54  3.982e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# (4) HEIGHT: [PRIORFRAC, MOMFRAC, ARMASSIST, RATERISK]

test <- model_effects
test <- update(test, . ~ . + HEIGHT:PRIORFRAC)
summary(test)
```

```
## 
## Call:
```

```
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + HEIGHT:PRIORFRAC, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6670  -0.7274  -0.5615  -0.0037   2.2377
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        3.79297    3.89138   0.975  0.32970
## AGE                0.03395    0.01307   2.597  0.00941 **
## HEIGHT            -0.05041    0.02253  -2.238  0.02524 *
## PRIORFRACYes      -2.41864    6.03699  -0.401  0.68869
## MOMFRACYes         0.63692    0.30850   2.065  0.03896 *
## ARMASSISTYes       0.43526    0.23394   1.861  0.06281 .
## RATERISKSame       0.42634    0.27946   1.526  0.12711
## RATERISKGreater    0.70410    0.29356   2.399  0.01646 *
## HEIGHT:PRIORFRACYes 0.01915   0.03770   0.508  0.61146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 507.24  on 491  degrees of freedom
## AIC: 525.24
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     HEIGHT:PRIORFRAC
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -253.62
## 2   1 -281.17 -8 55.092  4.236e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + HEIGHT:MOMFRAC)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + HEIGHT:MOMFRAC, family = binomial,
##     data = glow)
##
```

```
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -1.62068   -0.74163  -0.55649    0.06604   2.26717
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       4.73834    3.51132   1.349  0.17719
## AGE               0.03385    0.01307   2.589  0.00961 **
## HEIGHT           -0.05646    0.02021  -2.794  0.00521 **
## PRIORFRACYes      0.68102    0.24763   2.750  0.00596 **
## MOMFRACYes      -11.35526    7.64959  -1.484  0.13770
## ARMASSISTYes      0.47848    0.23444   2.041  0.04126 *
## RATERISKSame      0.42455    0.28002   1.516  0.12949
## RATERISKGreater   0.70475    0.29372   2.399  0.01642 *
## HEIGHT:MOMFRACYes 0.07401    0.04718   1.569  0.11675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 505.08  on 491  degrees of freedom
## AIC: 523.08
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     HEIGHT:MOMFRAC
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -252.54
## 2   1 -281.17 -8 57.258  1.603e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + HEIGHT:ARMASSIST)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + HEIGHT:ARMASSIST, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6742  -0.7177  -0.5638  -0.1472   2.1734
##
```

```
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.57428    4.12234  -0.139  0.88920
## AGE                 0.03401    0.01308   2.601  0.00931 **
## HEIGHT             -0.02318    0.02432  -0.953  0.34051
## PRIORFRACYes        0.67913    0.24841   2.734  0.00626 **
## MOMFRACYes          0.58729    0.30807   1.906  0.05660 .
## ARMASSISTYes        7.53985    5.77628   1.305  0.19179
## RATERISKSame        0.41583    0.27981   1.486  0.13725
## RATERISKGreater     0.70729    0.29369   2.408  0.01603 *
## HEIGHT:ARMASSISTYes -0.04419    0.03594  -1.229  0.21890
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 505.98  on 491  degrees of freedom
## AIC: 523.98
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     HEIGHT:ARMASSIST
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -252.99
## 2   1 -281.17 -8 56.352  2.409e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + HEIGHT:RATERISK)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + HEIGHT:RATERISK, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.64936  -0.72375  -0.57251  -0.05841  2.22612
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         3.25641    5.81516   0.560  0.57549
## AGE                 0.03321    0.01310   2.536  0.01122 *
```

```
## HEIGHT                    -0.04674    0.03532  -1.323  0.18573
## PRIORFRACYes               0.64451    0.24655   2.614  0.00895 **
## MOMFRACYes                 0.62504    0.30650   2.039  0.04142 *
## ARMASSISTYes               0.44610    0.23290   1.915  0.05544 .
## RATERISKSame               2.93823    7.29965   0.403  0.68730
## RATERISKGreater           -3.15056    7.29448  -0.432  0.66581
## HEIGHT:RATERISKSame       -0.01577    0.04550  -0.347  0.72890
## HEIGHT:RATERISKGreater     0.02394    0.04528   0.529  0.59695
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 506.55  on 490  degrees of freedom
## AIC: 526.55
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     HEIGHT:RATERISK
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  10 -253.28
## 2   1 -281.17 -9 55.786  8.624e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# (3) PRIORFRAC: [MOMFRAC, ARMASSIST, RATERISK]

test <- model_effects
test <- update(test, . ~ . + PRIORFRAC:MOMFRAC)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + PRIORFRAC:MOMFRAC, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.52616  -0.73215  -0.54992   0.02399   2.25279
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.97592    3.23781   0.919  0.35804
## AGE               0.03598    0.01313   2.741  0.00612 **
## HEIGHT           -0.04652    0.01837  -2.533  0.01130 *
```

```
## PRIORFRACYes              0.80102    0.26285   3.047  0.00231 **
## MOMFRACYes                0.95902    0.35985   2.665  0.00770 **
## ARMASSISTYes              0.43294    0.23384   1.851  0.06411 .
## RATERISKSame              0.41959    0.28027   1.497  0.13437
## RATERISKGreater           0.71282    0.29401   2.425  0.01533 *
## PRIORFRACYes:MOMFRACYes  -1.07823    0.65021  -1.658  0.09726 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 504.75  on 491  degrees of freedom
## AIC: 522.75
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
## 	PRIORFRAC:MOMFRAC
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1   9 -252.37
## 2   1 -281.17 -8 57.59  1.382e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + PRIORFRAC:ARMASSIST)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
## 	ARMASSIST + RATERISK + PRIORFRAC:ARMASSIST, family = binomial,
## 	data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.69860  -0.71874  -0.56691  -0.04199  2.21033
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           2.90711    3.25923   0.892  0.37241
## AGE                   0.03434    0.01306   2.630  0.00854 **
## HEIGHT               -0.04486    0.01842  -2.436  0.01487 *
## PRIORFRACYes          0.52412    0.34418   1.523  0.12780
## MOMFRACYes            0.63247    0.30798   2.054  0.04001 *
## ARMASSISTYes          0.36456    0.28322   1.287  0.19803
## RATERISKSame          0.42507    0.27929   1.522  0.12802
```

```
## RATERISKGreater             0.68837    0.29591    2.326   0.02000 *
## PRIORFRACYes:ARMASSISTYes    0.24587    0.48467    0.507   0.61194
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 507.24  on 491  degrees of freedom
## AIC: 525.24
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     PRIORFRAC:ARMASSIST
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -253.62
## 2   1 -281.17 -8 55.093  4.235e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test <- model_effects
test <- update(test, . ~ . + PRIORFRAC:RATERISK)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + PRIORFRAC:RATERISK, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min       1Q    Median        3Q       Max
## -1.69776  -0.71989  -0.56384  -0.03822   2.21130
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                2.733523   3.235459   0.845  0.39819
## AGE                        0.034508   0.013060   2.642  0.00823 **
## HEIGHT                    -0.043896   0.018313  -2.397  0.01653 *
## PRIORFRACYes               0.564292   0.497212   1.135  0.25641
## MOMFRACYes                 0.623104   0.307302   2.028  0.04260 *
## ARMASSISTYes               0.429891   0.236033   1.821  0.06856 .
## RATERISKSame               0.426181   0.324504   1.313  0.18907
## RATERISKGreater            0.632806   0.355571   1.780  0.07513 .
## PRIORFRACYes:RATERISKSame  0.001597   0.625563   0.003  0.99796
## PRIORFRACYes:RATERISKGreater 0.208811 0.624586   0.334  0.73814
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 507.32  on 490  degrees of freedom
## AIC: 527.32
##
## Number of Fisher Scoring iterations: 4
```

```r
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     PRIORFRAC:RATERISK
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  10 -253.66
## 2   1 -281.17 -9 55.015   1.21e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# (2) MOMFRAC: [ARMASSIST, RATERISK]

test <- model_effects
test <- update(test, . ~ . + MOMFRAC:ARMASSIST)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + MOMFRAC:ARMASSIST, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.65273  -0.72683  -0.55140   0.03367   2.27218
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.96640    3.25148   0.912  0.36160
## AGE                  0.03760    0.01323   2.842  0.00448 **
## HEIGHT              -0.04738    0.01846  -2.567  0.01025 *
## PRIORFRACYes         0.61633    0.24770   2.488  0.01284 *
## MOMFRACYes           1.17111    0.38940   3.007  0.00263 **
## ARMASSISTYes         0.65026    0.25220   2.578  0.00993 **
## RATERISKSame         0.41386    0.28032   1.476  0.13985
## RATERISKGreater      0.71051    0.29445   2.413  0.01582 *
## MOMFRACYes:ARMASSISTYes -1.33817  0.62405  -2.144  0.03201 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 502.83  on 491  degrees of freedom
## AIC: 520.83
##
## Number of Fisher Scoring iterations: 4
```

```r
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     MOMFRAC:ARMASSIST
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    9 -251.41
## 2    1 -281.17 -8 59.509  5.818e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
test <- model_effects
test <- update(test, . ~ . + MOMFRAC:RATERISK)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + MOMFRAC:RATERISK, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##      Min       1Q    Median       3Q       Max
## -1.73530  -0.73156  -0.56262  -0.02886   2.20217
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              2.76974    3.23715   0.856  0.39221
## AGE                      0.03436    0.01308   2.627  0.00861 **
## HEIGHT                  -0.04393    0.01832  -2.398  0.01649 *
## PRIORFRACYes             0.64526    0.24663   2.616  0.00889 **
## MOMFRACYes               0.02648    0.83795   0.032  0.97479
## ARMASSISTYes             0.44890    0.23340   1.923  0.05444 .
## RATERISKSame             0.29742    0.29700   1.001  0.31663
## RATERISKGreater          0.70206    0.31167   2.253  0.02428 *
## MOMFRACYes:RATERISKSame  1.04615    0.95957   1.090  0.27561
## MOMFRACYes:RATERISKGreater 0.36775  0.96207   0.382  0.70227
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 562.34  on 499  degrees of freedom
```

```
## Residual deviance: 505.79  on 490  degrees of freedom
## AIC: 525.79
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##     MOMFRAC:RATERISK
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  10 -252.90
## 2   1 -281.17 -9 56.542  6.183e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# (1) ARMASSIST: RATERISK
```

```
test <- model_effects
test <- update(test, . ~ . + ARMASSIST:RATERISK)
summary(test)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + ARMASSIST:RATERISK, family = binomial,
##     data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6586  -0.7419  -0.5544  -0.0470   2.2531
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 2.38363    3.22529   0.739  0.45988
## AGE                         0.03534    0.01314   2.691  0.00713 **
## HEIGHT                     -0.04274    0.01819  -2.349  0.01883 *
## PRIORFRACYes                0.70856    0.25028   2.831  0.00464 **
## MOMFRACYes                  0.61378    0.30782   1.994  0.04616 *
## ARMASSISTYes                0.60776    0.44193   1.375  0.16906
## RATERISKSame                0.36244    0.36906   0.982  0.32607
## RATERISKGreater             0.98400    0.38373   2.564  0.01034 *
## ARMASSISTYes:RATERISKSame   0.10760    0.56723   0.190  0.84956
## ARMASSISTYes:RATERISKGreater -0.60953    0.58200  -1.047  0.29496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 505.42  on 490  degrees of freedom
```

```
## AIC: 525.42
##
## Number of Fisher Scoring iterations: 4
```

```
lrtest(test)
```

```
## Likelihood ratio test
##
## Model 1: FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
##      ARMASSIST:RATERISK
## Model 2: FRACTURE ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  10 -252.71
## 2   1 -281.17 -9 56.912  5.253e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Results from interactions

```
## AGE:HEIGHT              0.001065   0.002095  0.508  0.6113
   AGE:PRIORFRACYes       -0.05864    0.02583  -2.270  0.023188 *
## AGE:MOMFRACYes         -0.01353    0.03264  -0.414  0.67854
## AGE:ARMASSISTYes       -0.01132    0.02479  -0.457  0.64802
## AGE:RATERISKSame       -0.03999    0.03151  -1.269  0.20438
## AGE:RATERISKGreater    -0.05021    0.03202  -1.568  0.11690
## HEIGHT:PRIORFRACYes     0.01915    0.03770   0.508  0.61146
## HEIGHT:MOMFRACYes       0.07401    0.04718   1.569  0.11675
## HEIGHT:ARMASSISTYes    -0.04419    0.03594  -1.229  0.21890
## HEIGHT:RATERISKSame    -0.01577    0.04550  -0.347  0.72890
## HEIGHT:RATERISKGreater  0.02394    0.04528   0.529  0.59695
   PRIORFRACYes:MOMFRACYes    -1.07823    0.65021  -1.658  0.09726 .
## PRIORFRACYes:ARMASSISTYes  0.24587    0.48467   0.507  0.61194
## PRIORFRACYes:RATERISKSame      0.001597   0.625563   0.003  0.99796
## PRIORFRACYes:RATERISKGreater  0.208811   0.624586   0.334  0.73814
   MOMFRACYes:ARMASSISTYes    -1.33817    0.62405  -2.144  0.03201 *
## MOMFRACYes:RATERISKSame     1.04615    0.95957   1.090  0.27561
## MOMFRACYes:RATERISKGreater  0.36775    0.96207   0.382  0.70227
## ARMASSISTYes:RATERISKSame      0.10760    0.56723   0.190  0.84956
## ARMASSISTYes:RATERISKGreater -0.60953    0.58200  -1.047  0.29496
```

Add to main effects model

we find three interactions, **AGE:PRIORFRACYes, PRIORFRACYes:MOMFRACYes, MOM-FRACYes:ARMASSISTYes**

```
model_effects_new <- glm(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
                         AGE:PRIORFRAC + PRIORFRAC:MOMFRAC + MOMFRAC:ARMASSIST, family = binomial, data =
summary(model_effects_new)
```

```
##
```

```
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:PRIORFRAC + PRIORFRAC:MOMFRAC +
##     MOMFRAC:ARMASSIST, family = binomial, data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.47664  -0.74929  -0.51571   0.07753   2.33224
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.20626    3.38765   0.356 0.721785
## AGE                      0.05949    0.01677   3.547 0.000389 ***
## HEIGHT                  -0.04610    0.01854  -2.487 0.012886 *
## PRIORFRACYes             4.63031    1.88158   2.461 0.013860 *
## MOMFRACYes               1.42093    0.42468   3.346 0.000820 ***
## ARMASSISTYes             0.59571    0.25545   2.332 0.019701 *
## RATERISKSame             0.42125    0.28217   1.493 0.135462
## RATERISKGreater          0.72341    0.29695   2.436 0.014847 *
## AGE:PRIORFRACYes        -0.05408    0.02602  -2.079 0.037662 *
## PRIORFRACYes:MOMFRACYes -0.83184    0.64852  -1.283 0.199606
## MOMFRACYes:ARMASSISTYes -1.15254    0.61838  -1.864 0.062350 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 496.53  on 489  degrees of freedom
## AIC: 518.53
##
## Number of Fisher Scoring iterations: 4

# create final model with interactions terms AGE:PRIORFRAC + MOMFRAC:ARMASSIST
model_effects_final <- glm(FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK +
                    AGE:PRIORFRAC + MOMFRAC:ARMASSIST, family = binomial, data = glow)
summary(model_effects_final)


##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:PRIORFRAC + MOMFRAC:ARMASSIST,
##     family = binomial, data = glow)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6995  -0.7459  -0.5238   0.0620   2.3123
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.96955    3.38252   0.287 0.774392
## AGE              0.05890    0.01666   3.535 0.000408 ***
## HEIGHT          -0.04413    0.01848  -2.388 0.016949 *
## PRIORFRACYes     4.65073    1.88342   2.469 0.013538 *
```

```
## MOMFRACYes                  1.19902    0.39487   3.036 0.002393 **
## ARMASSISTYes                0.61423    0.25358   2.422 0.015426 *
## RATERISKSame                0.42626    0.28154   1.514 0.130012
## RATERISKGreater             0.72116    0.29660   2.431 0.015040 *
## AGE:PRIORFRACYes           -0.05610    0.02600  -2.158 0.030950 *
## MOMFRACYes:ARMASSISTYes    -1.26534    0.62377  -2.029 0.042507 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 498.17  on 490  degrees of freedom
## AIC: 518.17
##
## Number of Fisher Scoring iterations: 4
```

Final Interaction Model

```
(Intercept)
AGE                       0.05890    0.01666   3.535 0.000408 ***
HEIGHT                   -0.04413    0.01848  -2.388 0.016949 *
PRIORFRACYes              4.65073    1.88342   2.469 0.013538 *
MOMFRACYes                1.19902    0.39487   3.036 0.002393 **
ARMASSISTYes              0.61423    0.25358   2.422 0.015426 *
RATERISKGreater           0.72116    0.29660   2.431 0.015040 *
AGE:PRIORFRACYes         -0.05610    0.02600  -2.158 0.030950 *
MOMFRACYes:ARMASSISTYes  -1.26534    0.62377  -2.029 0.042507 *

FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC + ARMASSIST + RATERISK + AGE:PRIORFRAC + MOMFRAC:ARMASSIS
```

```r
library(pROC)
```

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

```
##
## Attaching package: 'pROC'
```

```
## The following object is masked from 'package:gmodels':
##
##     ci
```

```
## The following object is masked from 'package:glmnet':
##
##     auc
```

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

```
library(vcdExtra)
```

```
## Loading required package: gnm

##
## Attaching package: 'gnm'

## The following object is masked from 'package:modeltools':
##
##     parameters

## The following object is masked from 'package:lattice':
##
##     barley

##
## Attaching package: 'vcdExtra'

## The following object is masked from 'package:carData':
##
##     Burt

## The following object is masked from 'package:plyr':
##
##     summarise

## The following object is masked from 'package:dplyr':
##
##     summarise
```

```
# vcov(model_effects_final)

HLtest(model_effects_final)
```

```
## Hosmer and Lemeshow Goodness-of-Fit Test
##
## Call:
## glm(formula = FRACTURE ~ AGE + HEIGHT + PRIORFRAC + MOMFRAC +
##     ARMASSIST + RATERISK + AGE:PRIORFRAC + MOMFRAC:ARMASSIST,
##     family = binomial, data = glow)
##  ChiSquare df   P_value
##   7.268011  8 0.5080118
```

```
glow$predict_mfinal <- predict(model_effects_final, type = "response")
with(glow, addmargins(table(glow$predict_mfinal > 0.5, glow$FRACTURE)))
```

```
##
##          No Yes Sum
##   FALSE 354  97 451
##   TRUE   21  28  49
##   Sum   375 125 500
```

```
(roc_final_model <- roc(glow$FRACTURE ~ glow$predict_mfinal, data = glow))
```

```
##
## Call:
## roc.formula(formula = glow$FRACTURE ~ glow$predict_mfinal, data = glow)
##
## Data: glow$predict_mfinal in 375 controls (glow$FRACTURE No) < 125 cases (glow$FRACTURE Yes).
## Area under the curve: 0.7331
```

```
plot(roc_final_model, main = "ROC: Final Effects Model")
```

ROC: Final Effects Model
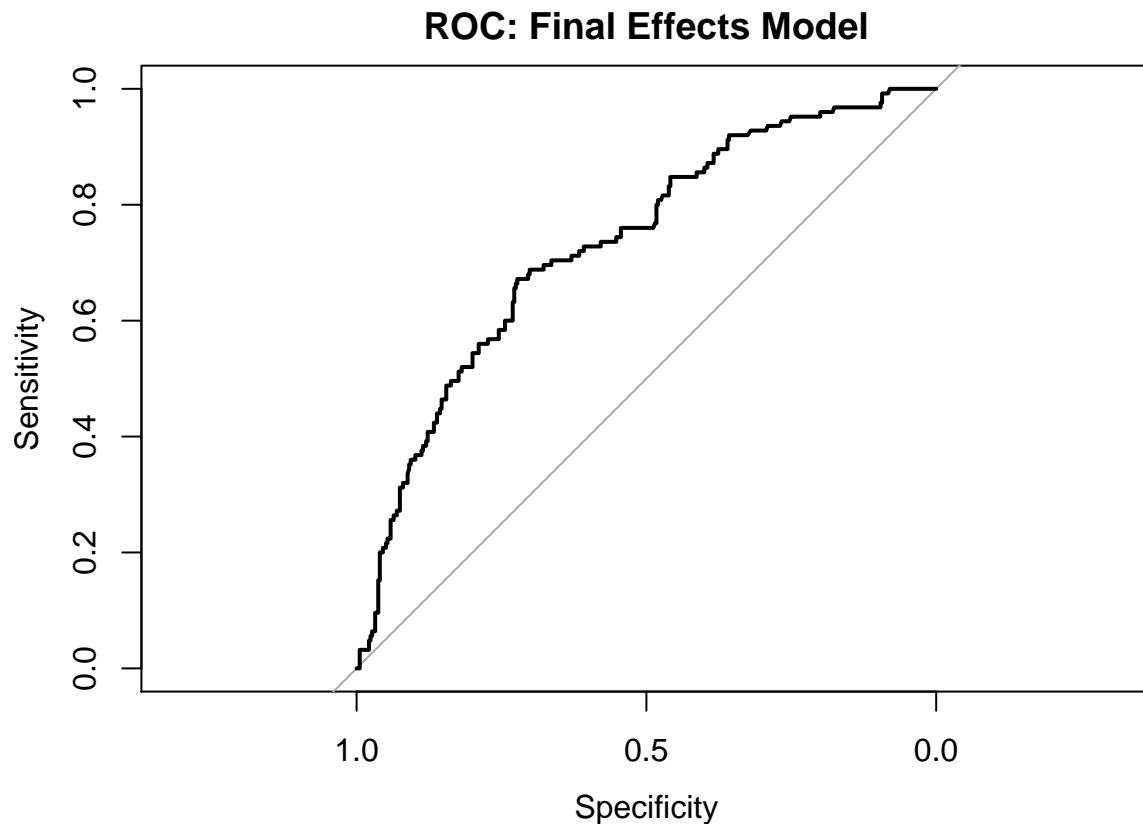
misc

```
## restart with clean data
glow <- read_glow_dataset()

model_last <- glm(FRACTURE ~ AGE:PRIORFRAC + HEIGHT + MOMFRAC:ARMASSIST + I(as.integer(RATERISK) == 3),
HLtest(model_last)
```

```
## Hosmer and Lemeshow Goodness-of-Fit Test
##
## Call:
## glm(formula = FRACTURE ~ AGE:PRIORFRAC + HEIGHT + MOMFRAC:ARMASSIST +
##      I(as.integer(RATERISK) == 3), family = binomial, data = glow)
##   ChiSquare df   P_value
##    3.10152  8 0.9278259
```

```r
summary(HLtest(model_last))
```

```
## Partition for Hosmer and Lemeshow Goodness-of-Fit Test
##
##                cut total obs      exp          chi
## 1  [0.0243,0.0967]    50  47 45.97075  0.1518032
## 2   (0.0967,0.123]    50  46 44.35914  0.2463653
## 3    (0.123,0.152]    50  42 43.19969 -0.1825284
## 4     (0.152,0.18]    50  41 41.81265 -0.1256745
## 5     (0.18,0.213]    50  42 40.34124  0.2611609
## 6    (0.213,0.251]    50  36 38.55936 -0.4121599
## 7    (0.251,0.292]    50  38 36.55362  0.2392312
## 8    (0.292,0.372]    50  32 33.67421 -0.2885110
## 9     (0.372,0.47]    50  28 29.25398 -0.2318447
## 10    (0.47,0.724]    50  23 21.27536  0.3739034
## Hosmer and Lemeshow Goodness-of-Fit Test
##
## Call:
## glm(formula = FRACTURE ~ AGE:PRIORFRAC + HEIGHT + MOMFRAC:ARMASSIST +
##     I(as.integer(RATERISK) == 3), family = binomial, data = glow)
##   ChiSquare df   P_value
##     3.10152  8 0.9278259
```

```r
# classification table
glow$predict_last <- predict(model_last, type = "response")
with(glow, addmargins(table(predict_last > 0.5, FRACTURE)))
```

```
##          FRACTURE
##           No Yes Sum
##   FALSE  355 103 458
##   TRUE    20  22  42
##   Sum    375 125 500
```

```r
# Sensitivy, specificity, ROC (using pROC)
roc_model_last <- roc(glow$FRACTURE ~ glow$predict_last, data = glow)
plot(roc_model_last, main = "ROC: Last model")
```

ROC: Last model
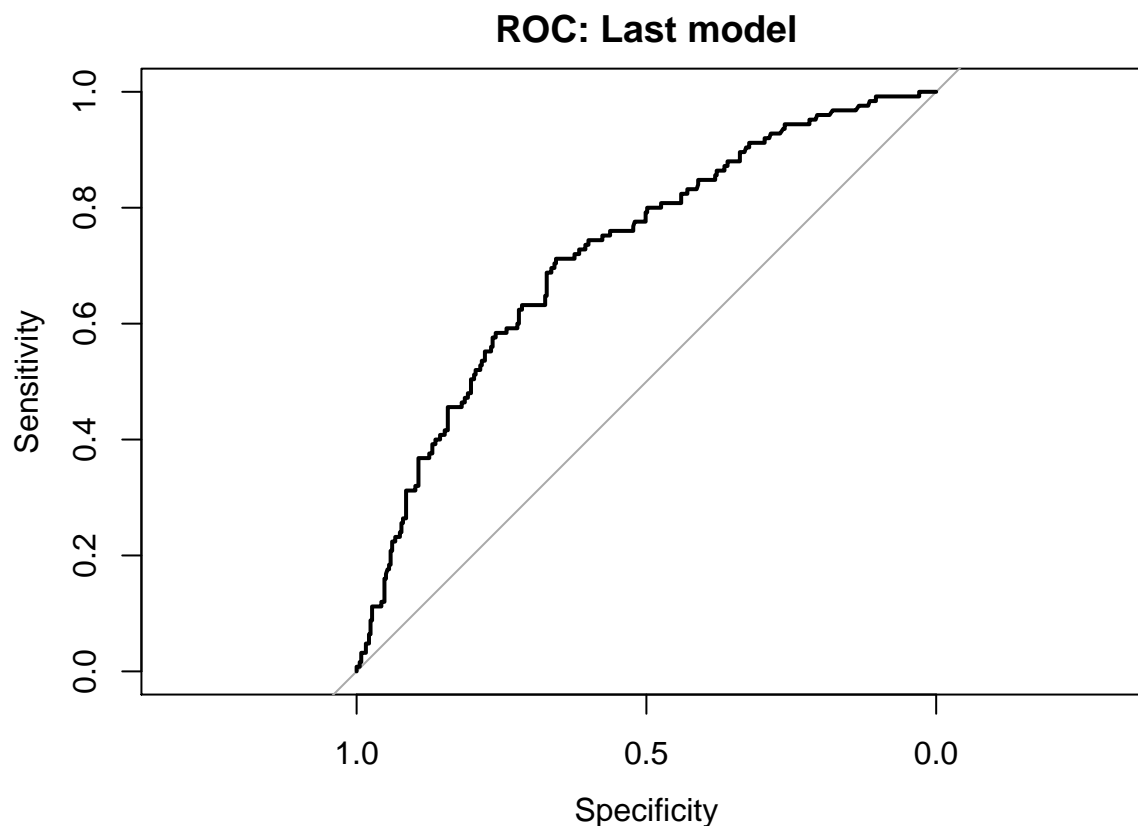

```r
# create table
vars <- c("thresholds","sensitivities","specificities")
model_table <- data.frame(roc_model_last[vars])

findIndex <- function(x, y) which.min( (x-y)^2 )
cutPoints <- seq(0.05, 0.75, by = 0.05)

tableIndex <- mapply(findIndex, y = cutPoints, MoreArgs = list(x = roc_model_last$thresholds))

model_table[tableIndex, ]
```

```
##       thresholds sensitivities specificities
## 3    0.05165803         1.000   0.005333333
## 43   0.09905744         0.976   0.128000000
## 120  0.15054070         0.880   0.349333333
## 202  0.20014367         0.760   0.549333333
## 259  0.25035362         0.640   0.674666667
## 316  0.29918726         0.520   0.789333333
## 349  0.34952747         0.416   0.842666667
## 379  0.40012793         0.320   0.893333333
## 401  0.44487716         0.240   0.925333333
## 416  0.49400138         0.176   0.944000000
## 426  0.55045683         0.120   0.954666667
## 443  0.59996390         0.056   0.978666667
## 452  0.65801860         0.024   0.992000000
## 455  0.69417409         0.008   0.994666667
```

```
## 457 0.71832154          0.008    1.000000000
```

```r
# plot
plot(specificities ~ thresholds, xlim = c(0, 1), type = "l",
xlab = "probability cutoff", ylab = "sensitivity / specificity",
ylim = c(0, 1), data = model_table, main = "probability sensitivity")
with(model_table, lines(thresholds, sensitivities, col = "red"))
legend(x = 0.75, y = 0.55, legend = c("Sensitivity", "Specificity"),
lty = 1, col = c("red","black"))
abline(h = c(0, 1), col = "grey80", lty = "dotted")
```

probability sensitivity