

MSDS 6372 Project 2

Contents

Data Set 1: Osteoporosis in Women	1
Setup:	1
Data Import and Cleaning	1
Exploratory Data Analysis	2
Create Train and Validation Datasets	3
Summary Statistics	3
Build Model using Training Data	33
Creating a balanced dataset by Oversampling and check accuracy.	42

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 data is independent sample of participants. 3.

```
glow_data_file <- 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)]

#Cleanup column names
colnames(dataset)[colnames(dataset)=="i.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$ARMASIST <- factor(dataset$ARMASIST, 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"))

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 ...
## $ 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 ...
## $ ARMASIST: 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

```
numericVar <- dataset[,5:8]
ID_var <- dataset[,c(1:3)]
set_noID <- dataset[4:14]
categoricalVar <- set_noID[,~c(2:5)]
```

Create Train and Validation Datasets

```
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

```
#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"
```

```
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"
```

```
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"
```

```
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"
```

```
#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
## WEIGHT     39.90000 59.90000 68.00000 71.82320 81.30000 127.00000
## HEIGHT     134.00000 157.00000 161.50000 161.36400 165.00000 199.00000
## BMI        14.87637 23.26889 26.41898 27.55303 30.79205 49.08241
```

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

```
##      SUB_ID SITE_ID PHY_ID PRIORFRAC AGE WEIGHT HEIGHT      BMI PREMENO
## 1         1       1     14         0  62   70.3   158 28.16055        0
## 2         2       4    284         0  65   87.1   160 34.02344        0
## 3         3       6    305         1  88   50.8   157 20.60936        0
## 4         4       6    309         0  82   62.1   160 24.25781        0
## 5         5       1     37         0  61   68.0   152 29.43213        0
## 6         6       5    299         1  67   68.0   161 26.23356        0
## 7         7       5    302         0  84   50.8   150 22.57778        0
## 8         8       1     36         1  82   40.8   153 17.42919        0
## 9         9       1      8         1  86   62.6   156 25.72321        0
## 10        10      4    282         0  58   63.5   166 23.04398        0
## 11        11      6    315         0  67   67.6   153 28.87778        0
## 12        12      1     34         0  56  117.9   167 42.27473        0
## 13        13      6    315         0  59   67.1   162 25.56775        0
```

```
## 14      14      1      33      0 72  57.6    165 21.15702      0
## 15      15      1      23      0 64  61.2    160 23.90625      1
## 16      16      3     179      0 68  78.0    161 30.09143      0
## 17      17      4     284      0 67 105.7    165 38.82461      0
## 18      18      4     283      0 69  65.8    162 25.07240      0
## 19      19      3     179      1 78  81.6    162 31.09282      0
## 20      20      6     313      0 60  56.7    157 23.00296      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
```

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

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

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

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

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

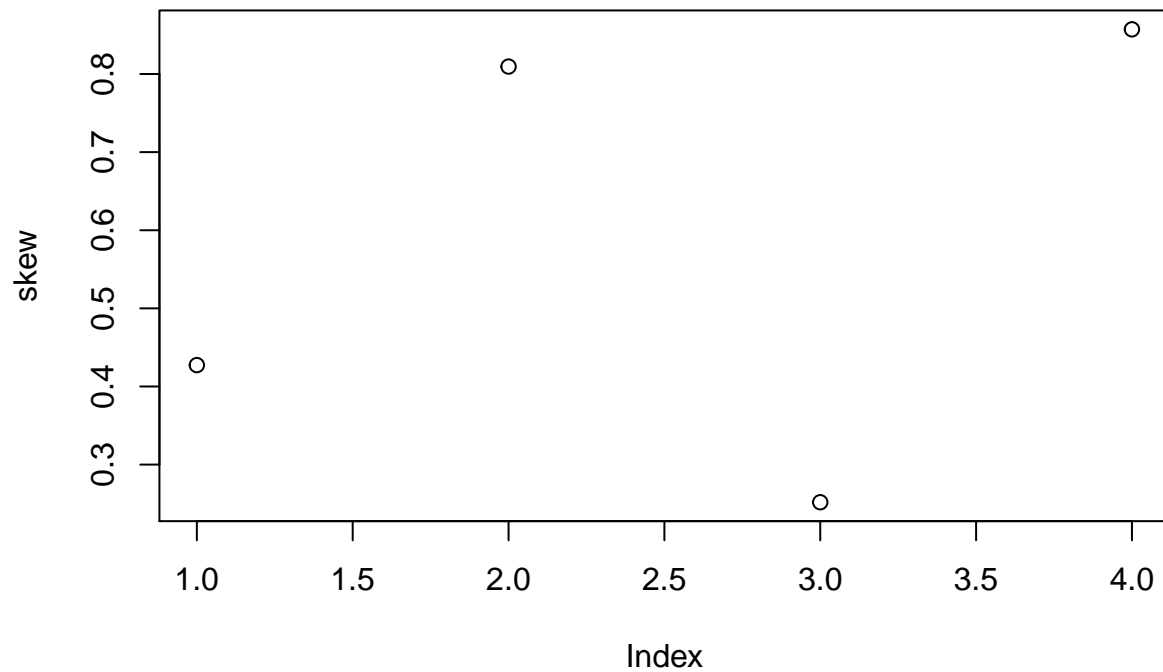
Skewness

The further the distribution of the skew value from zero, the larger the skew to the left (negative skew value) or right (positive skew value). The skewness results of the continuous variables show that Age, Weight, Height and BMI are positively skewed.

```
skew=apply(numericVar, 2, skewness)
print(skew)
```

```
##      AGE      WEIGHT      HEIGHT      BMI
## 0.4273768 0.8095144 0.2518129 0.8571709
```

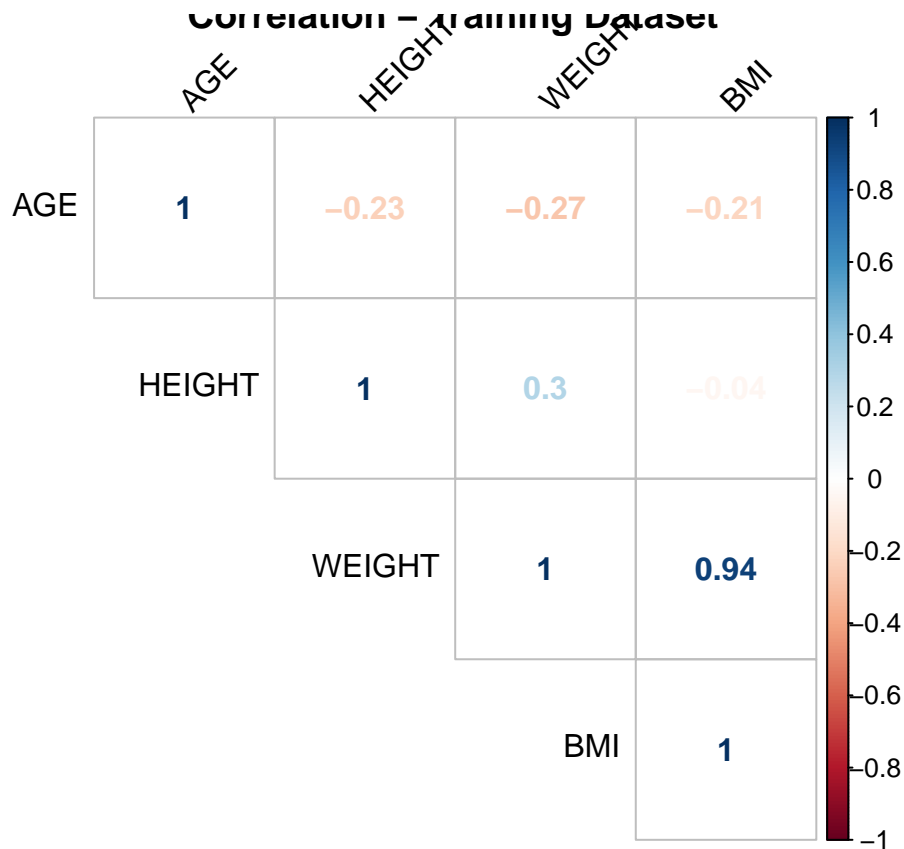
```
plot(skew)
```



###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.

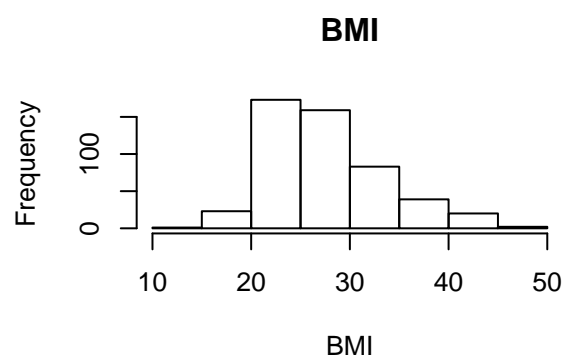
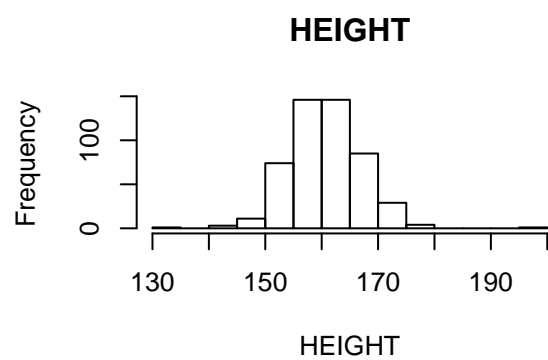
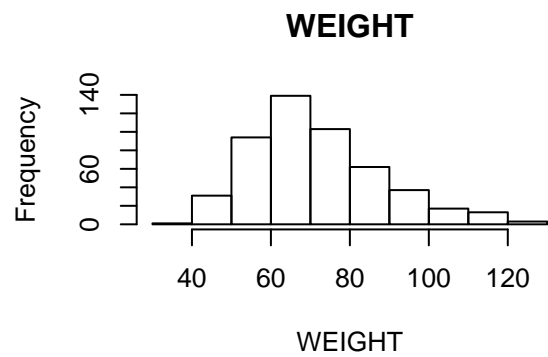
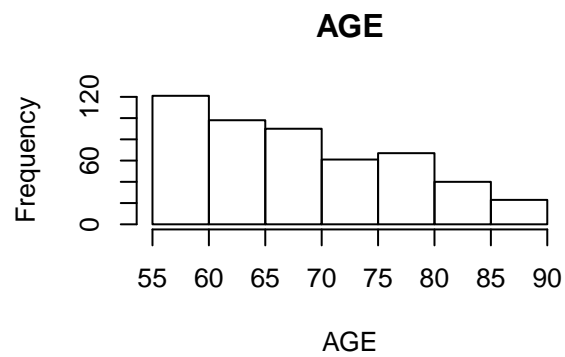
```
#Training dataset without ID columns  
corrplot(cor(trainingData[2:5]), method = "number", type = "upper", order = "hclust",  
          tl.col = "black", tl.srt = 45, main="Correlation - Training Dataset")
```



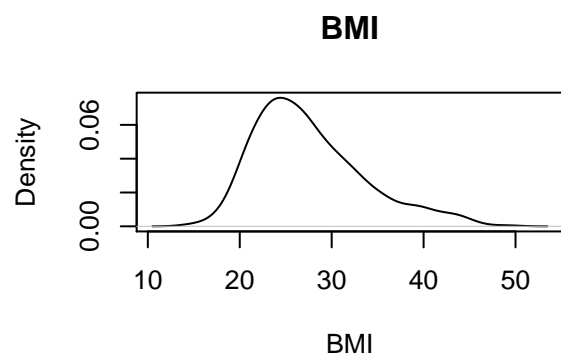
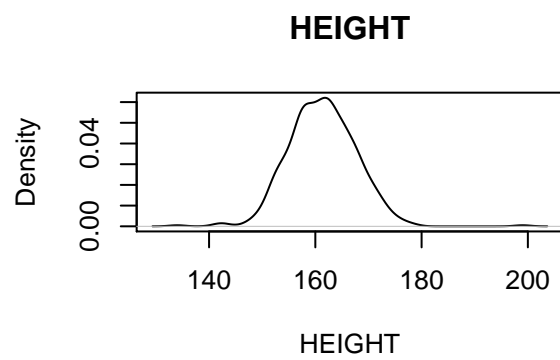
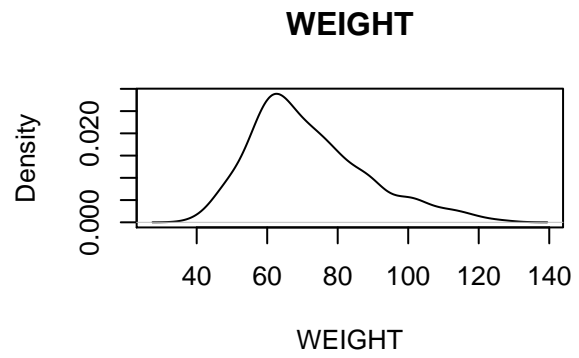
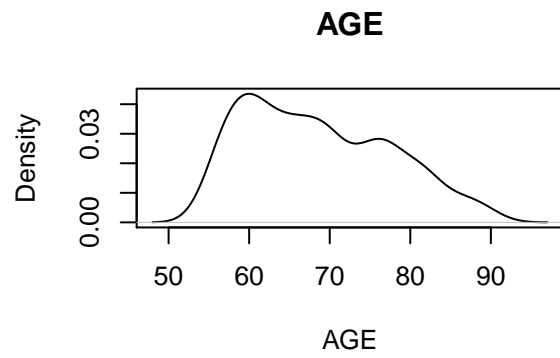
Visualization of Continuous Variables For the categorical variables, we show an unbalanced dataset of subjects with majority false PRIORFRAC, PREMENO, MOMFRAC, ARMASSIST, and SMOKE. There was a good balance of subjects in the 3 levels of RATERISK. An unbalanced dataset will cause a model to favor the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

For the continuous 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.

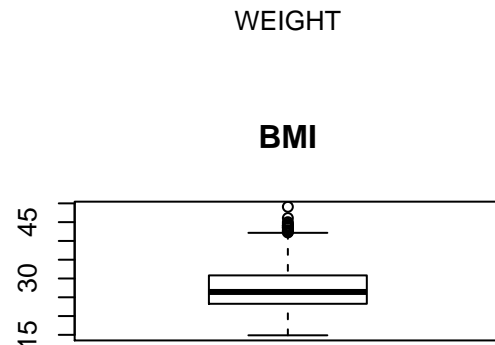
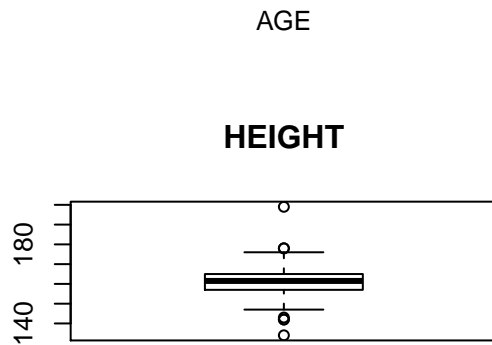
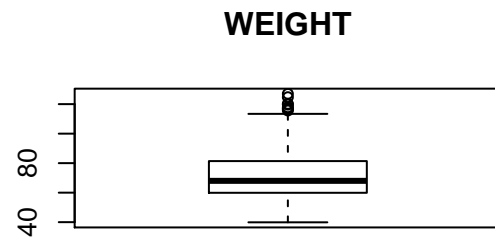
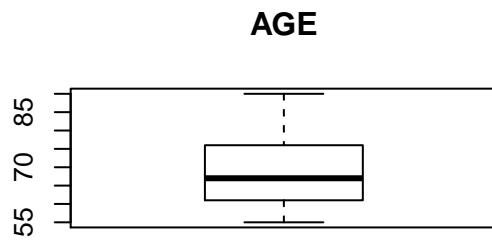
```
# Data visualizations
dataset_numeric = numericVar
#Histograms
par(mfrow=c(2,2)) # put four figures in a row (2*4)
for (i in 1:4) {
  hist(dataset_numeric[,i],xlab=names(dataset_numeric)[i],main=names(dataset_numeric)[i])
}
```

```
#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])
}
```



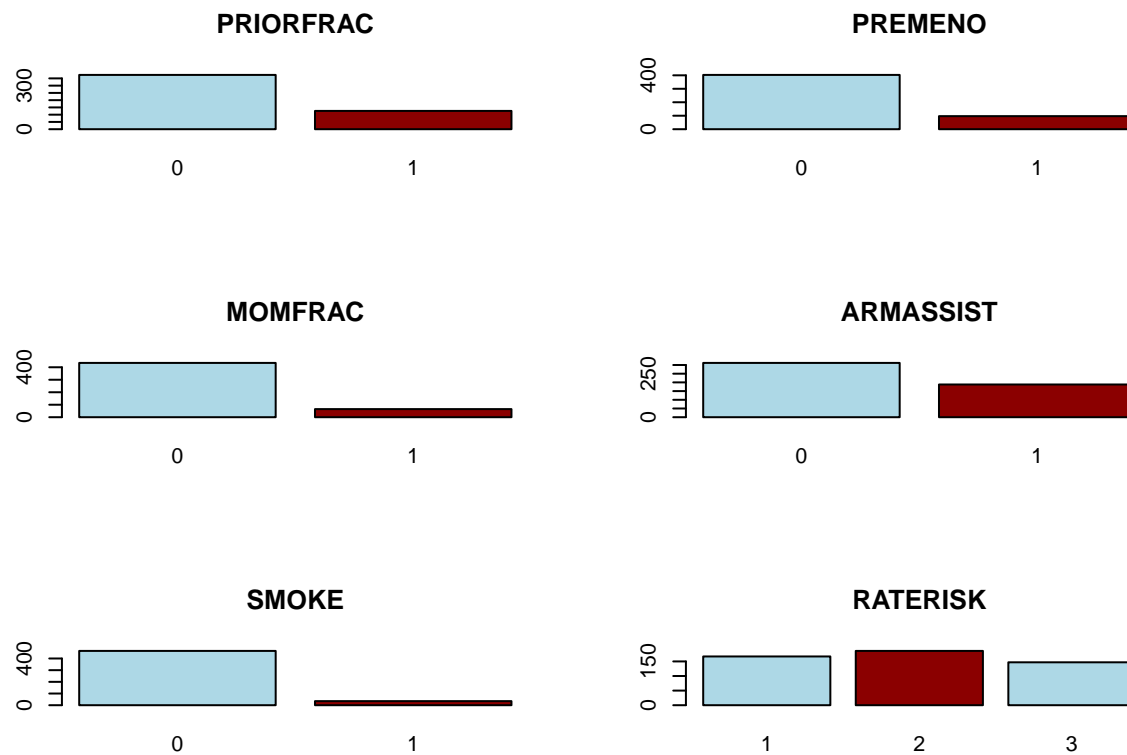
```
#Box And Whisker Plots
par(mfrow=c(2,2))
for(i in 1:4) {
  boxplot(dataset_numeric[,i], xlab=names(dataset_numeric)[i], main=names(dataset_numeric)[i])
}
```



HEIGHT

BMI

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



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

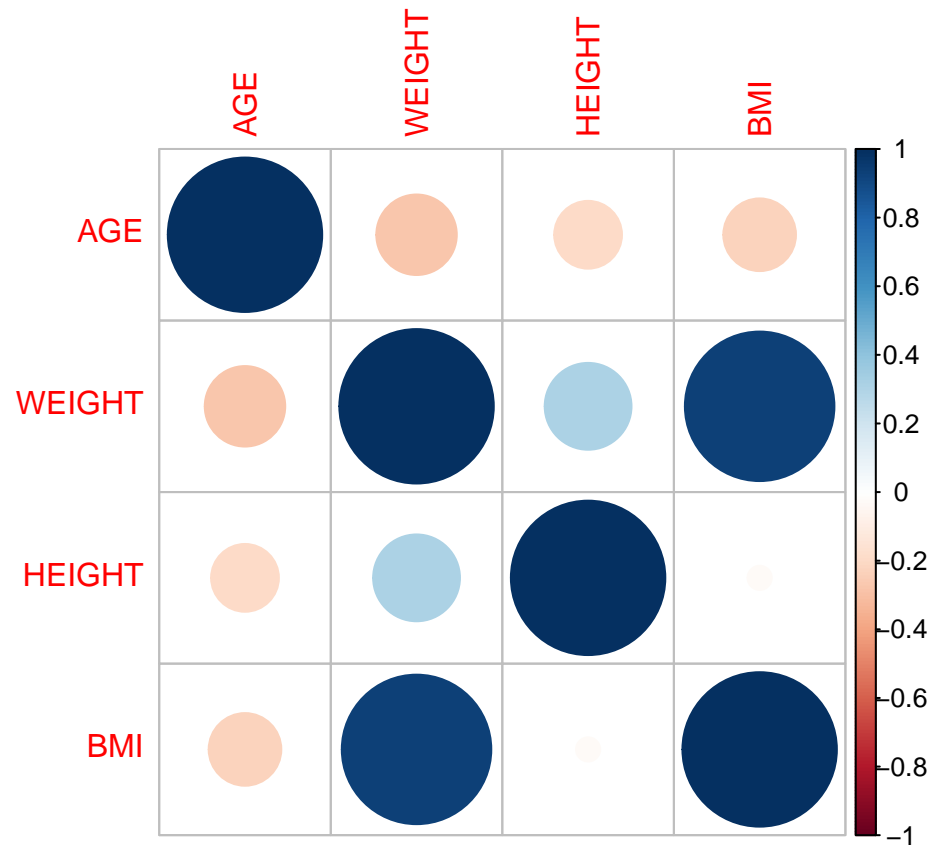
```
##          AGE      WEIGHT      HEIGHT      BMI
## AGE      1.0000000 -0.2715964 -0.19264861 -0.22125651
## WEIGHT -0.2715964  1.0000000  0.31596915  0.93733603
## HEIGHT -0.1926486  0.3159691  1.00000000 -0.02437689
## BMI     -0.2212565  0.9373360 -0.02437689  1.00000000
```

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

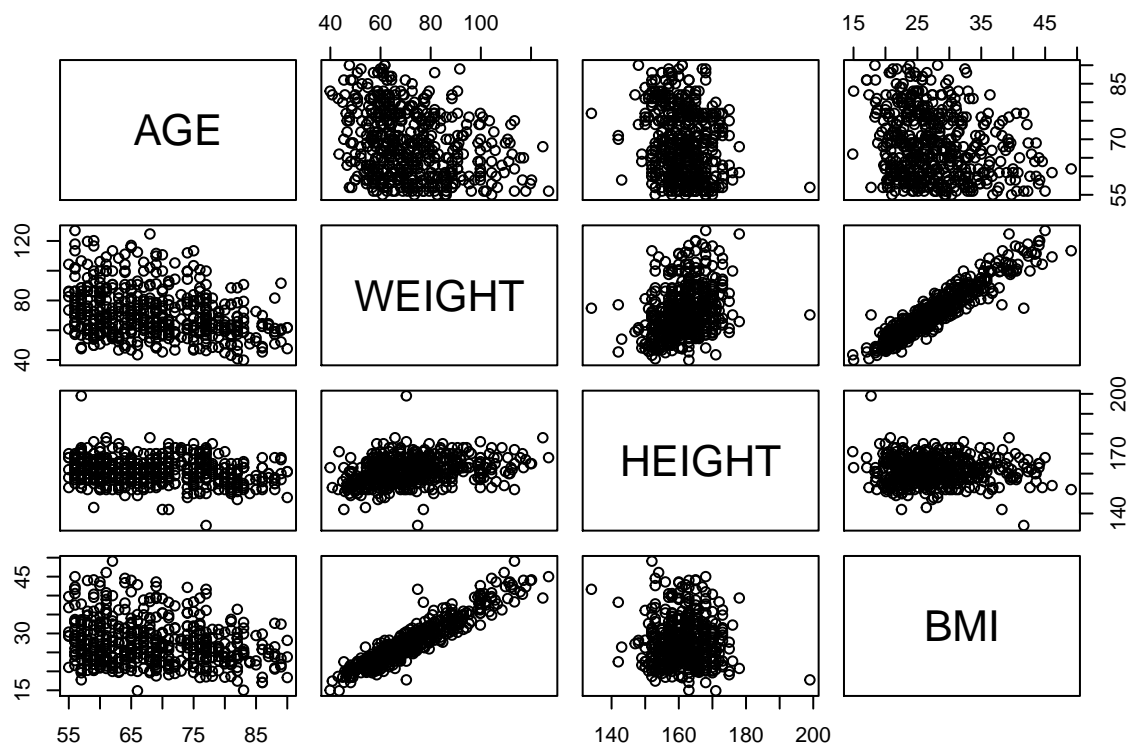
```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt
## = tl.srt, : "methods" is not a graphical parameter
```

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

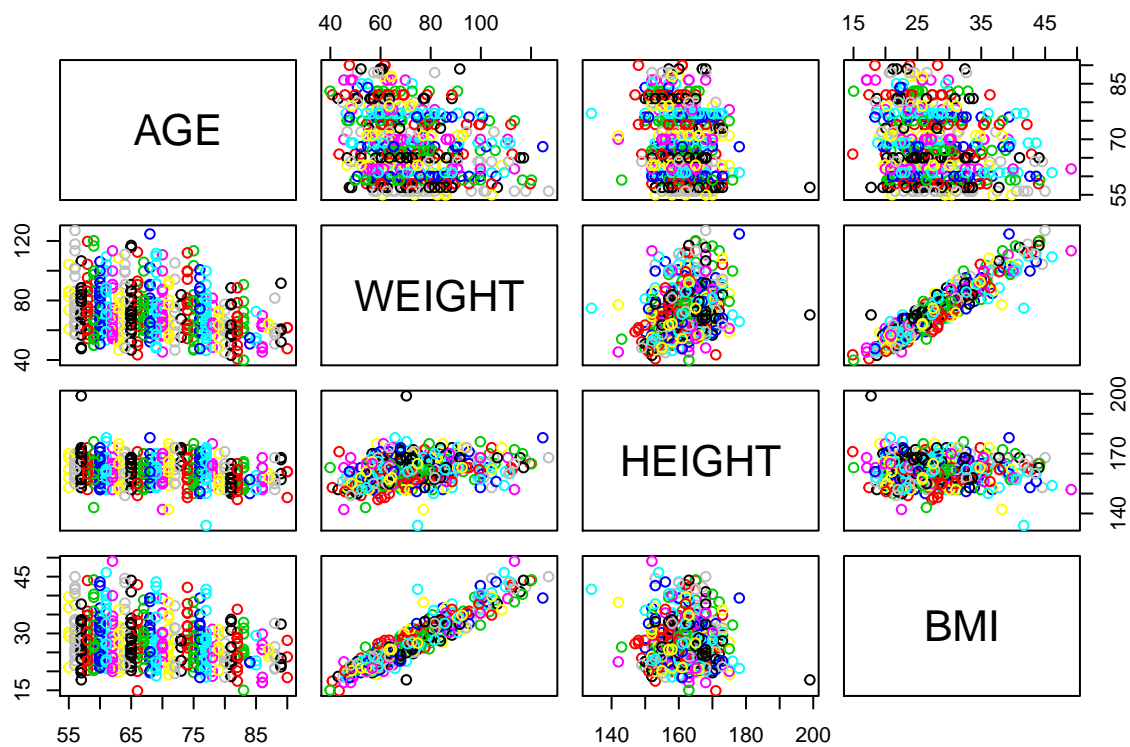
```
## Warning in title(title, ...): "methods" is not a graphical parameter
```



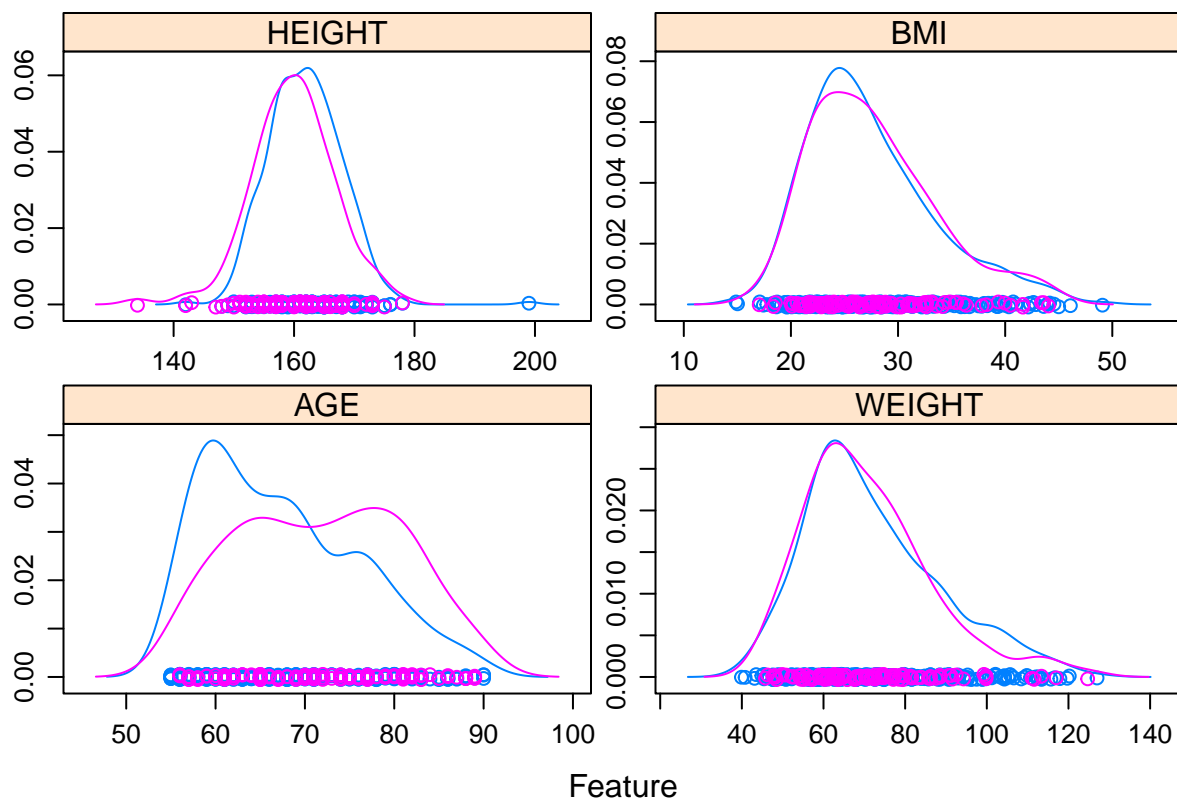
```
# pair-wise scatterplots of the numeric attributes  
par(mfrow=c(1,1))  
pairs(dataset_numeric)
```



```
#Scatterplot Matrix By Class (use different color to distinguish different class)
par(mfrow=c(1,1))
pairs(dataset_numeric, col=dataset[,5])
```



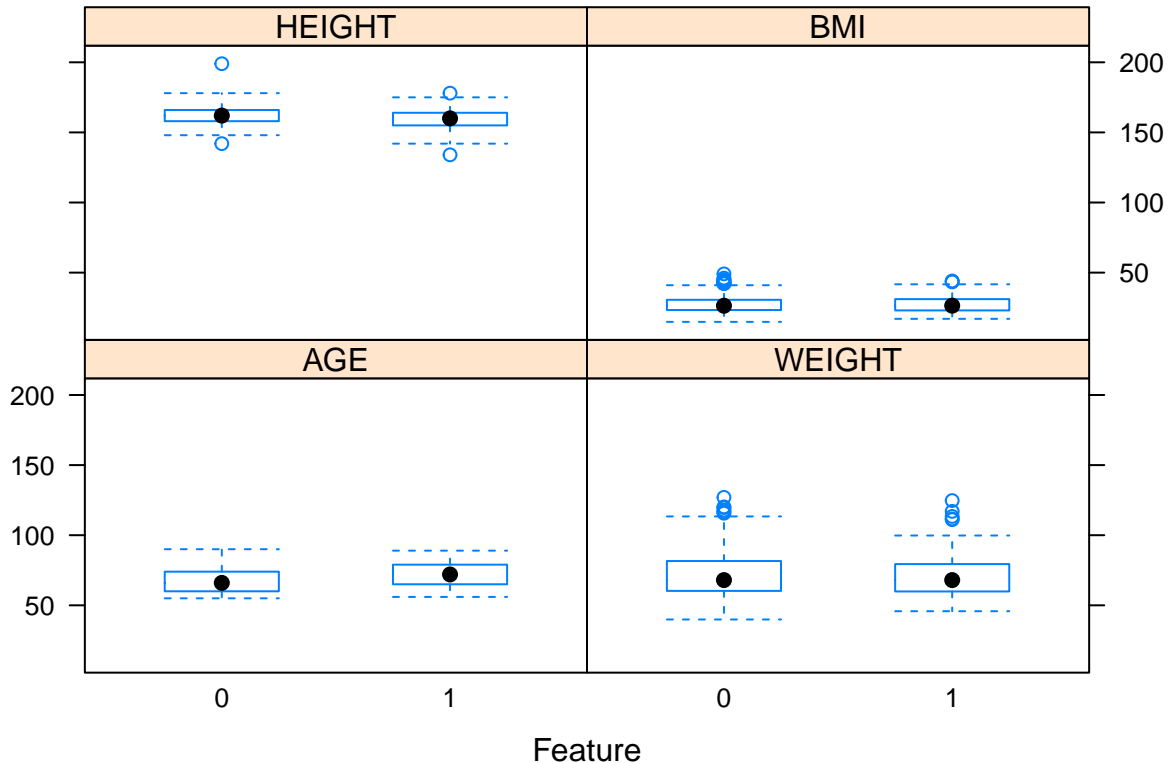
```
# density plots for each attribute by class value
X <- set_noID[2:5]
Y <- set_noID$FRACTURE
scales <- list(x=list(relation="free"), y=list(relation="free"))
par(mfrow=c(1,1))
featurePlot(x=set_noID[2:5], y=set_noID$FRACTURE, plot="density", scales=scales)
```



#Box And Whisker Plots By Class

```
par(mfrow=c(1,1))
```

```
featurePlot(x=set_noID[2:5], y=set_noID$FRACTURE, plot="box")
```

Checking the Balance of the Full dataset

The current sample dataset contains a larger proportion 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
```

```

# 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

prop.table(table(trainingData$FRACTURE))

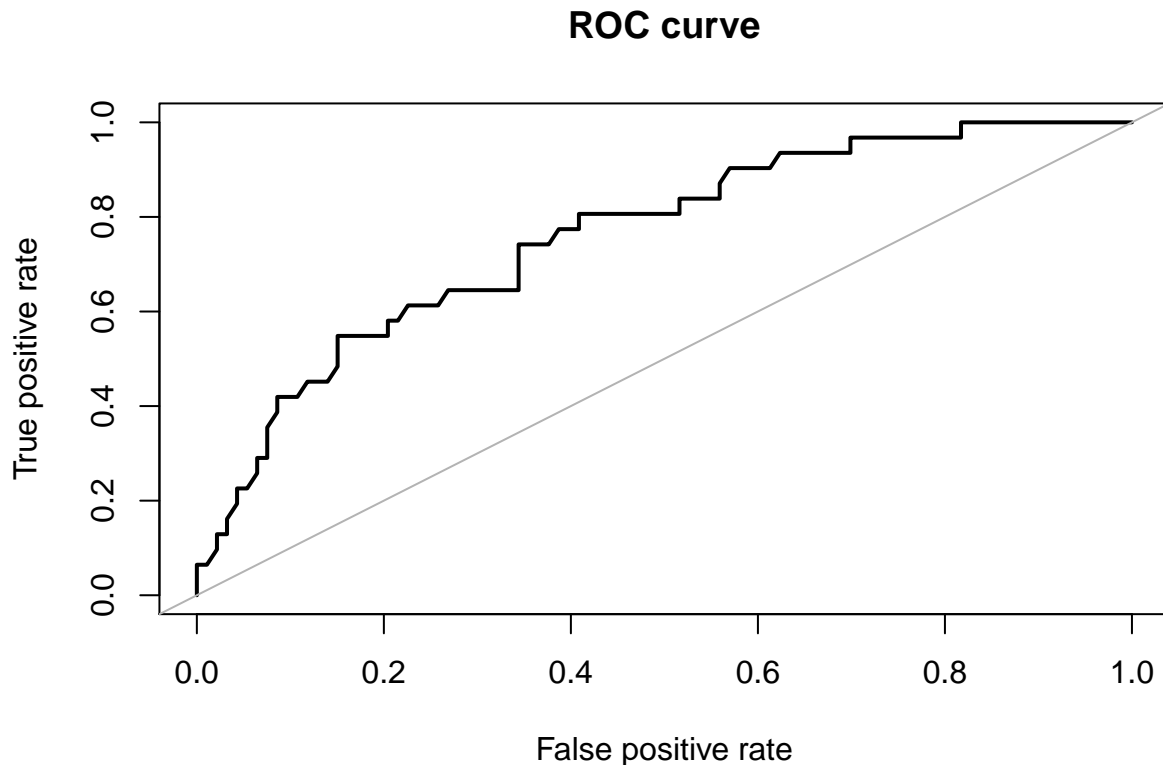
##
##      0      1
## 0.75 0.25

#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

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

```



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 Fractures 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 overwhelmingly the alternative hypothesis that the PRIORFRAC variable is dependent on Fracture variable.

Variable: PREMENO 80% of the sample subjects are not in Pre-Menopause had 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.

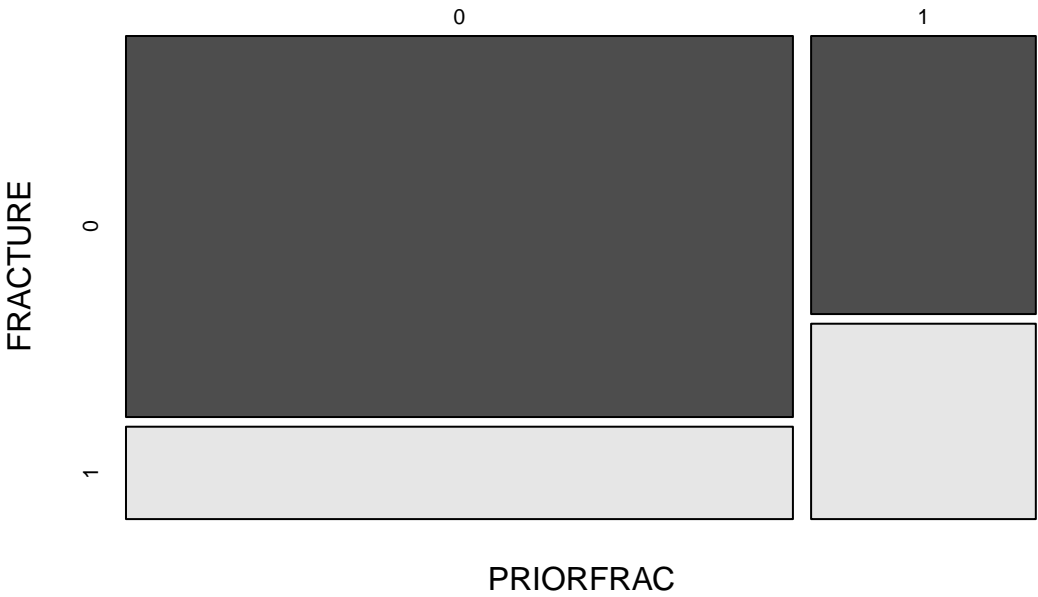
Variable: RATERISK Raterisk shows the frequency of subjects in each Raterisk level is between 29%-33%. This is pretty even in terms of how many subjects are within each Raterisk. For those that did have Fractures, their probability of a fracture increased with the level of Raterisk. This makes sense.

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

```
##
##
##      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
```

```
##
## 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 |          |
## -----|-----|-----|-----|

```

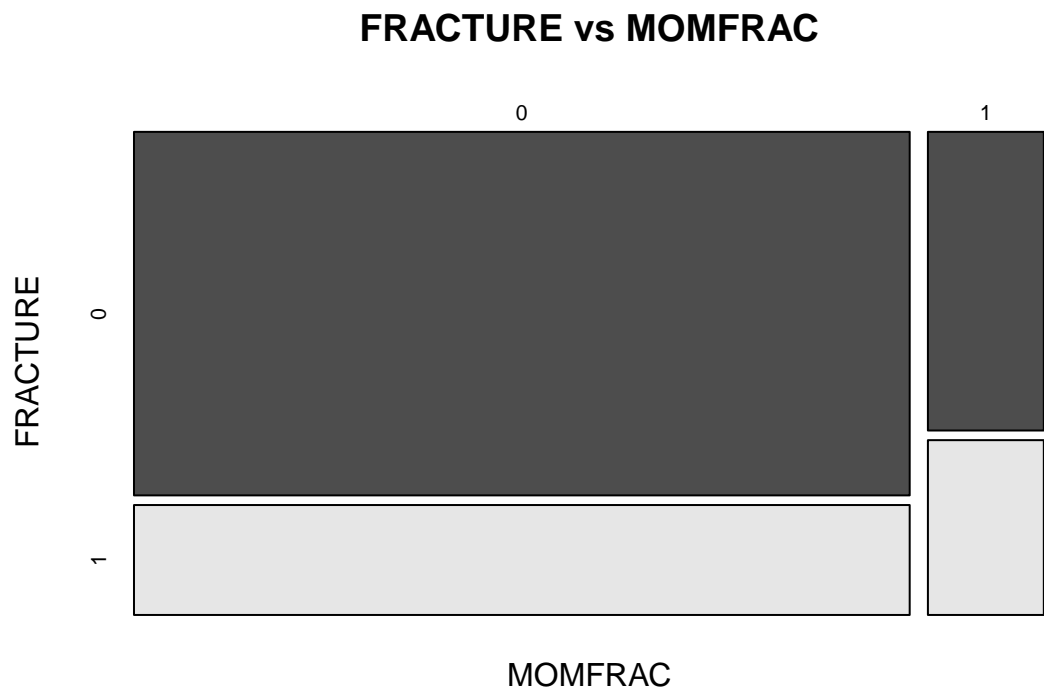


```

##          | 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 |          |
## -----|-----|-----|-----|

```

##		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	
##	-----		-----	-----	-----
##					
##					



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

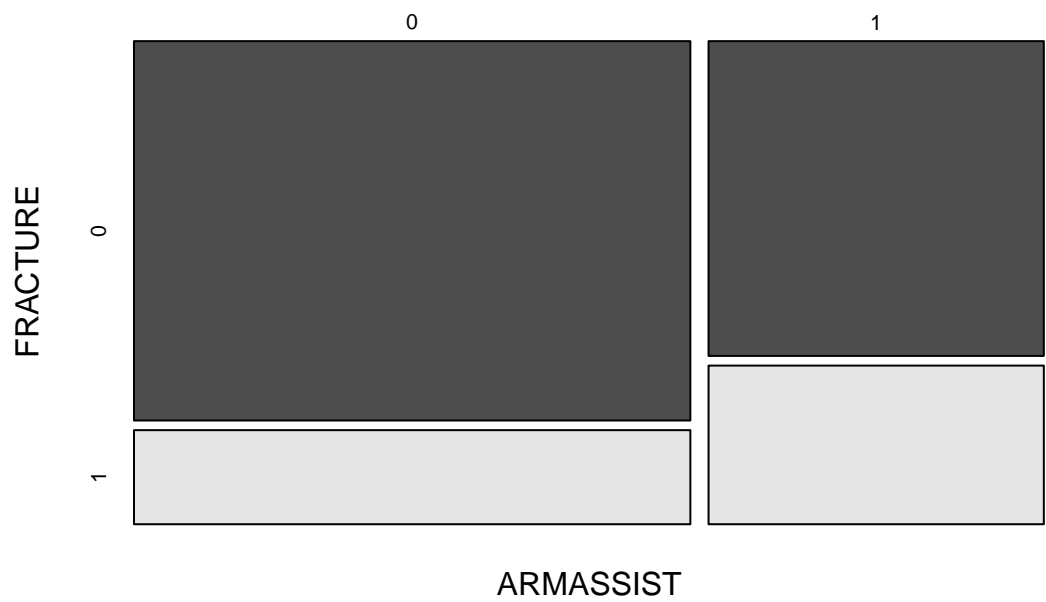
```

##
##
##      | 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



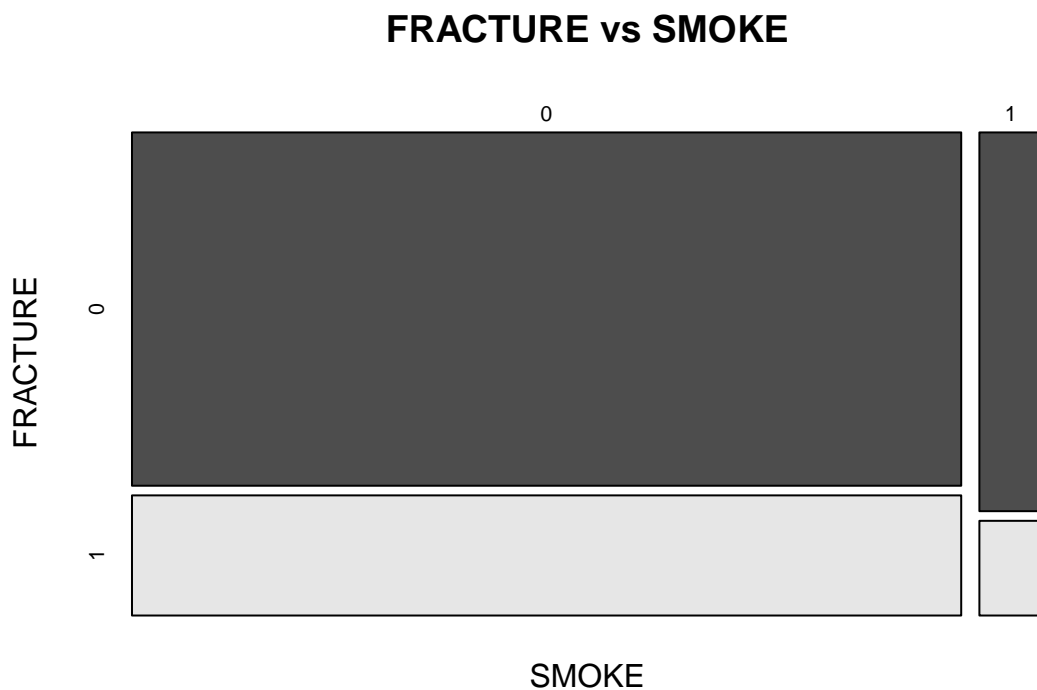
##	
##	
##	Cell Contents
##	-----
##	N
##	Expected N
##	Chi-square contribution
##	N / Row Total

```

## |          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[, categoricalVar] | 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 |    |
## -----|-----|-----|-----|
##
##
```



```
##
##
```

```

##      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
##
##
##
##

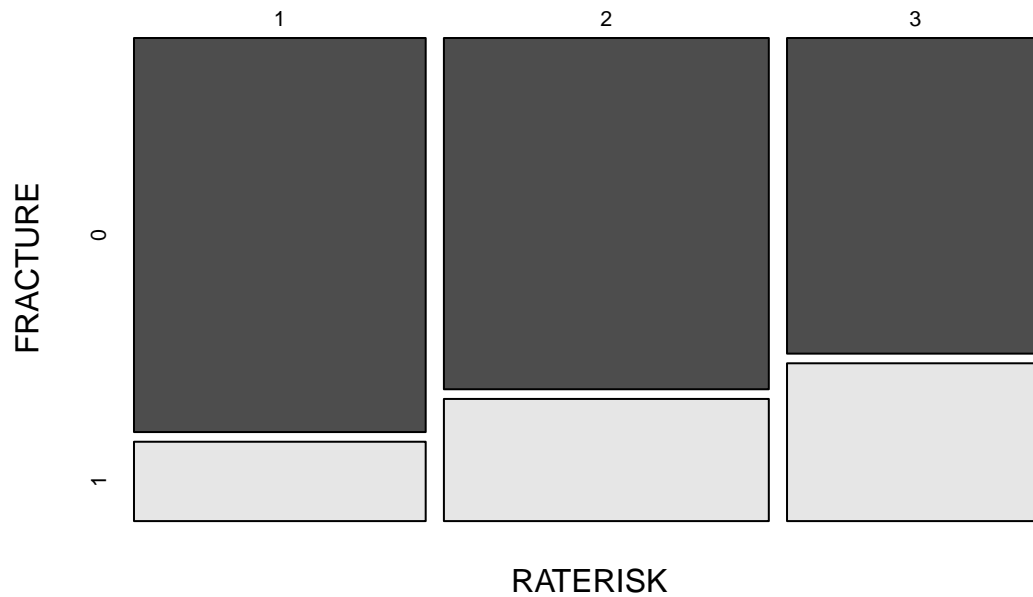
```

```

##      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. ###PCA and Scree Plot

```
new_my_data <- lapply(trainingData, function(x) {
  if(is.factor(x)) as.numeric(as.character(x)) else x
})
new_my_data <- data.frame(new_my_data)
str(new_my_data)
```

```
## 'data.frame': 376 obs. of 11 variables:
## $ PRIORFRAC: num 0 1 0 1 1 0 0 0 0 0 ...
## $ AGE : int 65 88 61 67 82 58 67 56 72 67 ...
## $ WEIGHT : num 87.1 50.8 68 68 40.8 ...
## $ HEIGHT : int 160 157 152 161 153 166 153 167 165 165 ...
## $ BMI : num 34 20.6 29.4 26.2 17.4 ...
## $ PREMENO : num 0 0 0 0 0 0 0 0 0 0 ...
## $ MOMFRAC : num 0 1 0 0 0 0 1 0 0 0 ...
## $ ARMASSIST: num 0 1 0 0 0 0 0 1 1 0 ...
## $ SMOKE : num 0 0 0 1 0 0 1 1 0 0 ...
## $ RATERISK : num 2 1 2 2 2 1 1 2 1 1 ...
## $ FRACTURE : num 0 0 0 0 0 0 0 0 0 0 ...
```

#Get Training Set

```
dat.train <- new_my_data
```

```
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))
```

#PCA

```
pc.result<-prcomp(dat.train.x,scale=TRUE)
```

```
pc.scores<-pc.result$x
```

```
pc.scores<-data.frame(pc.scores)
```

```
pc.scores$FRACTURE <- dat.train.y
```

#or

#principal component analysis

```
prin_comp <- prcomp(new_my_data, scale. = T)
```

```
names(prin_comp)
```

```
## [1] "sdev" "rotation" "center" "scale" "x"
```

```
prin_comp
```

```
## Standard deviations (1, ..., p=11):
```

```
## [1] 1.50941443 1.31400138 1.10196020 1.04887811 0.98934478 0.95744397
```

```
## [7] 0.92952730 0.87976521 0.84853075 0.65087348 0.05879005
```

```
##
```

```
## Rotation (n x k) = (11 x 11):
```

```
## PC1 PC2 PC3 PC4 PC5
```

```
## PRIORFRAC 0.07405715 -0.47973677 0.04196542 0.09391798 -0.13590836
```

```
## AGE 0.25108797 -0.44246880 -0.36503870 -0.23606187 -0.19052036
```

```
## WEIGHT -0.64071593 -0.07386956 -0.01492425 -0.09380628 0.04511049
```

```
## HEIGHT -0.22890271 0.24187107 0.22588400 -0.52730271 -0.16245665
```

```
## BMI      -0.59713909 -0.16251927 -0.09572198  0.09472088  0.11445051
## PREMENO  -0.11505234 -0.07693268  0.32458960  0.55307931  0.16142556
## MOMFRAC   0.11007196 -0.13173669  0.46323427 -0.46128883  0.06458940
## ARMASSIST -0.25734750 -0.47047462 -0.02257374 -0.19863859 -0.16959440
## SMOKE     -0.04972241  0.01391670  0.31483340  0.26442893 -0.86124378
## RATERISK  0.10918438 -0.20973926  0.61992312 -0.03166000  0.26572927
## FRACTURE  0.07750976 -0.44147408  0.02641809  0.10280064  0.17988806
##          PC6          PC7          PC8          PC9          PC10
## PRIORFRAC  0.009110347 -0.51653703  0.12726745 -0.65076852 -0.16932571
## AGE        0.245357588  0.13034942  0.07530809  0.08177107  0.65295027
## WEIGHT     -0.120183672 -0.03235101  0.02116421 -0.08190948  0.22942735
## HEIGHT     0.327444296 -0.34498645 -0.46918526 -0.08143823  0.14936479
## BMI        -0.245140543  0.09823153  0.18404370 -0.05154601  0.18874838
## PREMENO    0.622965173  0.25696963 -0.17636604 -0.16498653  0.17626188
## MOMFRAC    -0.148794845  0.59700737  0.10567492 -0.38334831 -0.04799385
## ARMASSIST  0.368115390  0.11545494  0.07410041  0.40014322 -0.57380527
## SMOKE      -0.230599843  0.09342985 -0.02981247  0.11155875  0.10632225
## RATERISK   -0.084062927 -0.37403086  0.29943887  0.43734477  0.24549790
## FRACTURE   -0.389848139  0.04558067 -0.76508915  0.12573228 -0.01463757
##          PC11
## PRIORFRAC -0.0073515954
## AGE       -0.0025546320
## WEIGHT    0.7054674906
## HEIGHT    -0.2415073104
## BMI       -0.6662434928
## PREMENO   0.0042646791
## MOMFRAC   0.0004639497
## ARMASSIST -0.0001841958
## SMOKE     -0.0025392898
## RATERISK  -0.0004029694
## FRACTURE  0.0048813762
```

#1. center and scale refers to respective mean and standard deviation of the variables that are used for

```
#outputs the mean of variables
prin_comp$center
```

```
##      PRIORFRAC      AGE      WEIGHT      HEIGHT      BMI
## 0.24734043  68.38297872  71.97367021  161.36968085  27.59879508
##      PREMENO      MOMFRAC      ARMASSIST      SMOKE      RATERISK
## 0.19148936  0.12765957  0.39095745  0.07180851  2.00000000
##      FRACTURE
## 0.25000000
```

```
#outputs the standard deviation of variables
prin_comp$scale
```

```
##      PRIORFRAC      AGE      WEIGHT      HEIGHT      BMI      PREMENO
## 0.4320412  8.8349082  17.0902771  6.6255293  6.1707189  0.3939975
##      MOMFRAC      ARMASSIST      SMOKE      RATERISK      FRACTURE
## 0.3341550  0.4886151  0.2585146  0.7797435  0.4335897
```

#2. The rotation measure provides the principal component loading. Each column of rotation matrix conta

```
prin_comp$rotation
```

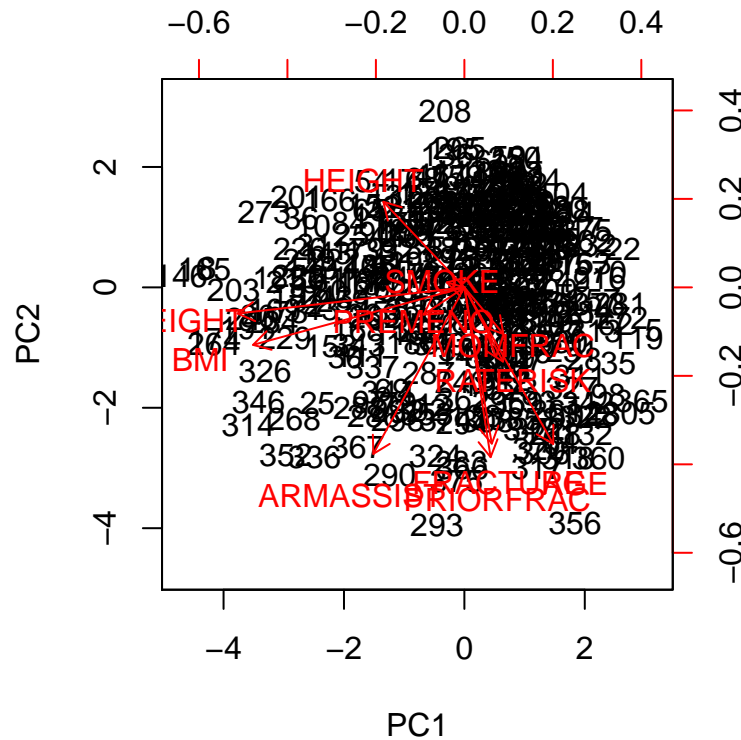
```
##          PC1          PC2          PC3          PC4          PC5
## PRIORFRAC  0.07405715 -0.47973677  0.04196542  0.09391798 -0.13590836
## AGE        0.25108797 -0.44246880 -0.36503870 -0.23606187 -0.19052036
## WEIGHT     -0.64071593 -0.07386956 -0.01492425 -0.09380628  0.04511049
## HEIGHT     -0.22890271  0.24187107  0.22588400 -0.52730271 -0.16245665
## BMI        -0.59713909 -0.16251927 -0.09572198  0.09472088  0.11445051
## PREMENO    -0.11505234 -0.07693268  0.32458960  0.55307931  0.16142556
## MOMFRAC     0.11007196 -0.13173669  0.46323427 -0.46128883  0.06458940
## ARMASSIST  -0.25734750 -0.47047462 -0.02257374 -0.19863859 -0.16959440
## SMOKE       -0.04972241  0.01391670  0.31483340  0.26442893 -0.86124378
## RATERISK    0.10918438 -0.20973926  0.61992312 -0.03166000  0.26572927
## FRACTURE    0.07750976 -0.44147408  0.02641809  0.10280064  0.17988806
##          PC6          PC7          PC8          PC9          PC10
## PRIORFRAC  0.009110347 -0.51653703  0.12726745 -0.65076852 -0.16932571
## AGE        0.245357588  0.13034942  0.07530809  0.08177107  0.65295027
## WEIGHT     -0.120183672 -0.03235101  0.02116421 -0.08190948  0.22942735
## HEIGHT     0.327444296 -0.34498645 -0.46918526 -0.08143823  0.14936479
## BMI        -0.245140543  0.09823153  0.18404370 -0.05154601  0.18874838
## PREMENO    0.622965173  0.25696963 -0.17636604 -0.16498653  0.17626188
## MOMFRAC    -0.148794845  0.59700737  0.10567492 -0.38334831 -0.04799385
## ARMASSIST  0.368115390  0.11545494  0.07410041  0.40014322 -0.57380527
## SMOKE      -0.230599843  0.09342985 -0.02981247  0.11155875  0.10632225
## RATERISK   -0.084062927 -0.37403086  0.29943887  0.43734477  0.24549790
## FRACTURE   -0.389848139  0.04558067 -0.76508915  0.12573228 -0.01463757
##          PC11
## PRIORFRAC -0.0073515954
## AGE       -0.0025546320
## WEIGHT    0.7054674906
## HEIGHT    -0.2415073104
## BMI       -0.6662434928
## PREMENO   0.0042646791
## MOMFRAC   0.0004639497
## ARMASSIST -0.0001841958
## SMOKE     -0.0025392898
## RATERISK  -0.0004029694
## FRACTURE  0.0048813762
```

```
prin_comp$rotation[1:5,1:4]
```

```
##          PC1          PC2          PC3          PC4
## PRIORFRAC  0.07405715 -0.47973677  0.04196542  0.09391798
## AGE        0.25108797 -0.44246880 -0.36503870 -0.23606187
## WEIGHT     -0.64071593 -0.07386956 -0.01492425 -0.09380628
## HEIGHT     -0.22890271  0.24187107  0.22588400 -0.52730271
## BMI        -0.59713909 -0.16251927 -0.09572198  0.09472088
```

```
dim(prin_comp$x)
```

```
#3 Plot result
biplot(prin_comp, scale = 0)
```



```
#compute standard deviation of each principal component
std_dev <- prin_comp$sdev
```

```
#check variance of first 10 components
pr_var[1:10]
```

#proportion of variance explained

```
prop_varex <- pr_var/sum(pr_var)
prop_varex[1:11]
```

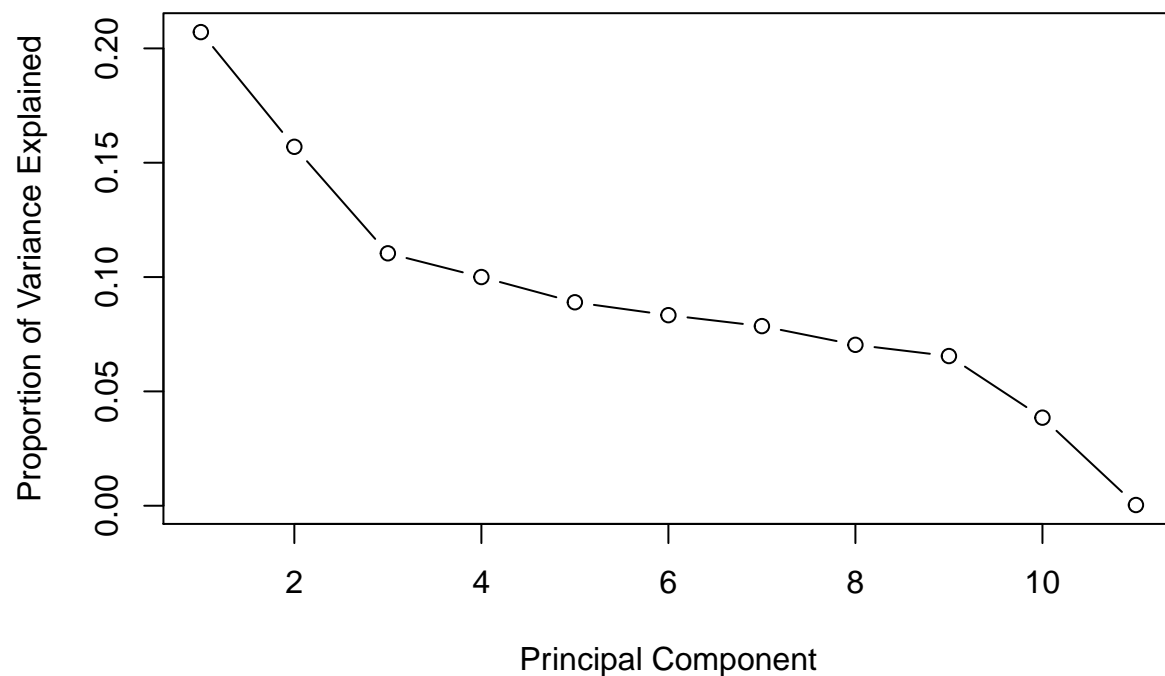
```
## [1] 0.2071210847 0.1569636036 0.1103923891 0.1000132085 0.0889820993
## [6] 0.0833362684 0.0785473632 0.0703624381 0.0654549491 0.0385123896
## [11] 0.0003142063
```

#This shows that first principal component explains 10.3% variance. Second component explains 7.3% vari.

#The answer to this question is provided by a scree plot. A scree plot is used to access components or .

#scree plot

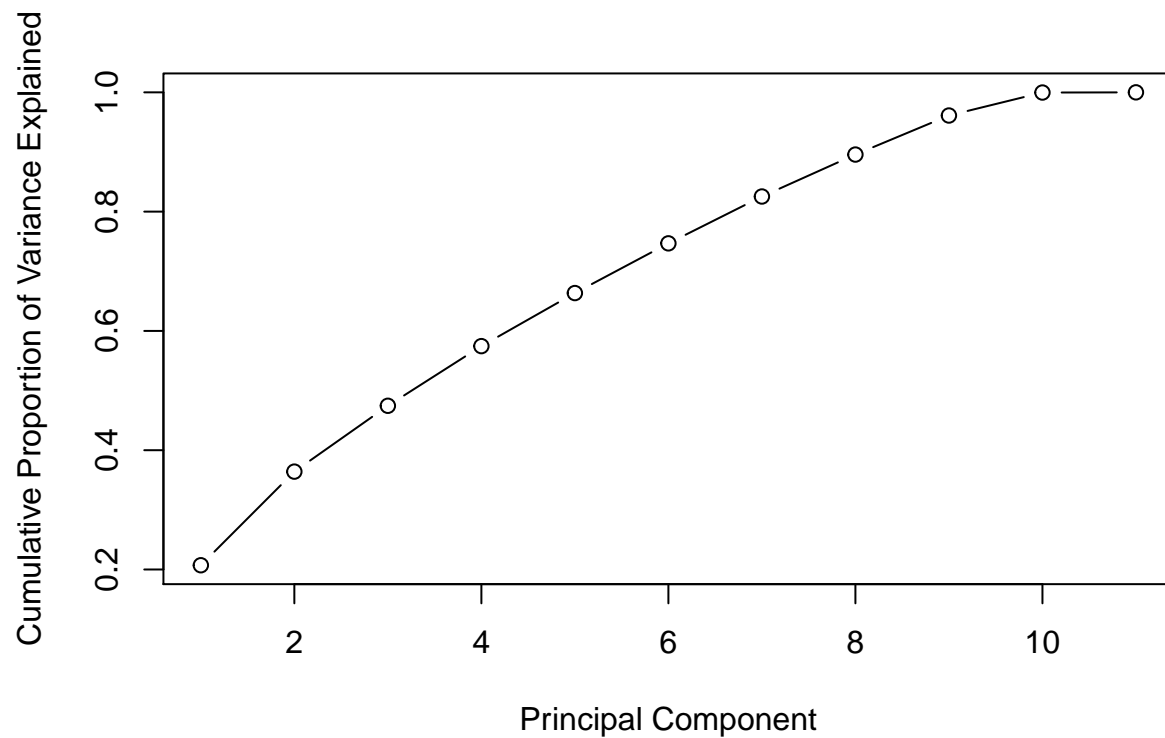
```
plot(prop_varex, xlab = "Principal Component",
      ylab = "Proportion of Variance Explained",
      type = "b")
```



#The plot above shows that ~ all 11 components explains around 99% variance in the data set. In order w

#cumulative scree plot

```
plot(cumsum(prop_varex), xlab = "Principal Component",
      ylab = "Cumulative Proportion of Variance Explained",
      type = "b")
```



##Clustering

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

*#Transposing the predictor matrix and giving the response categories its
#row names.*

#Get Training Set

```
dat.train <- new_my_data
```

```
dat.train.x <- dat.train[,1:ncol(dat.train)]
```

```
dat.train.y <- dat.train$FRACTURE
```

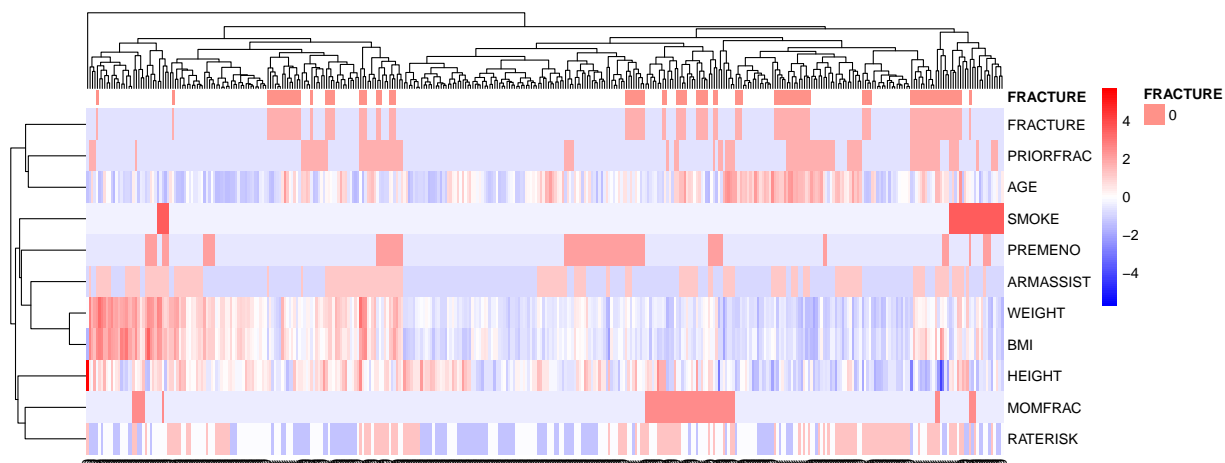
```
dat.train.y <- as.factor(as.character(dat.train.y))
```

#Heatmap

```
x<-t(dat.train.x)
```

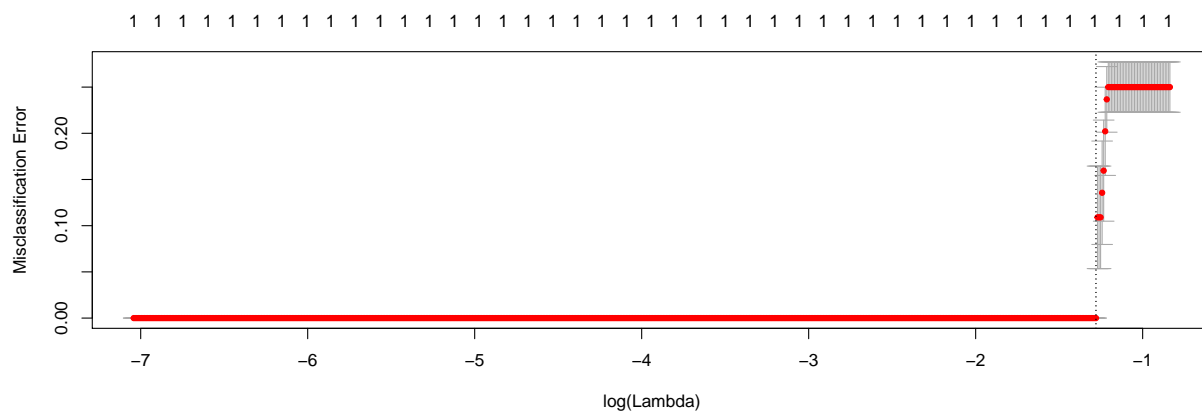
```
colnames(x)<-dat.train.y
```

```
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 = 100)
plot(cvfit)
```



```
coef(cvfit, s = "lambda.min")
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept) -1.653758
## PRIORFRAC    .
## AGE          .
## WEIGHT       .
## HEIGHT       .
## BMI          .
## PREMENO      .
## MOMFRAC      .
## ARMASSIST    .
## SMOKE        .
## RATERISK     .
## FRACTURE     1.726571
```

Creating a balanced dataset by Oversampling and check accuracy.

Below we created a balanced dataset using ROSE package. The idea is that by creating an over sampled dataset with randomly duplicated records, we produced a more balanced dataset of subjects.

```
#Look at the proportion of subjects with Fractures from the Full dataset.  
table(set_noID$FRACTURE)
```

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

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

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

```
#Create an oversampled dataset from Training dataset  
over <- ovun.sample(FRACTURE~., data=trainingData, method="over", N=564)$data  
table(over$FRACTURE)
```

```
##  
##    0    1  
## 282 282
```

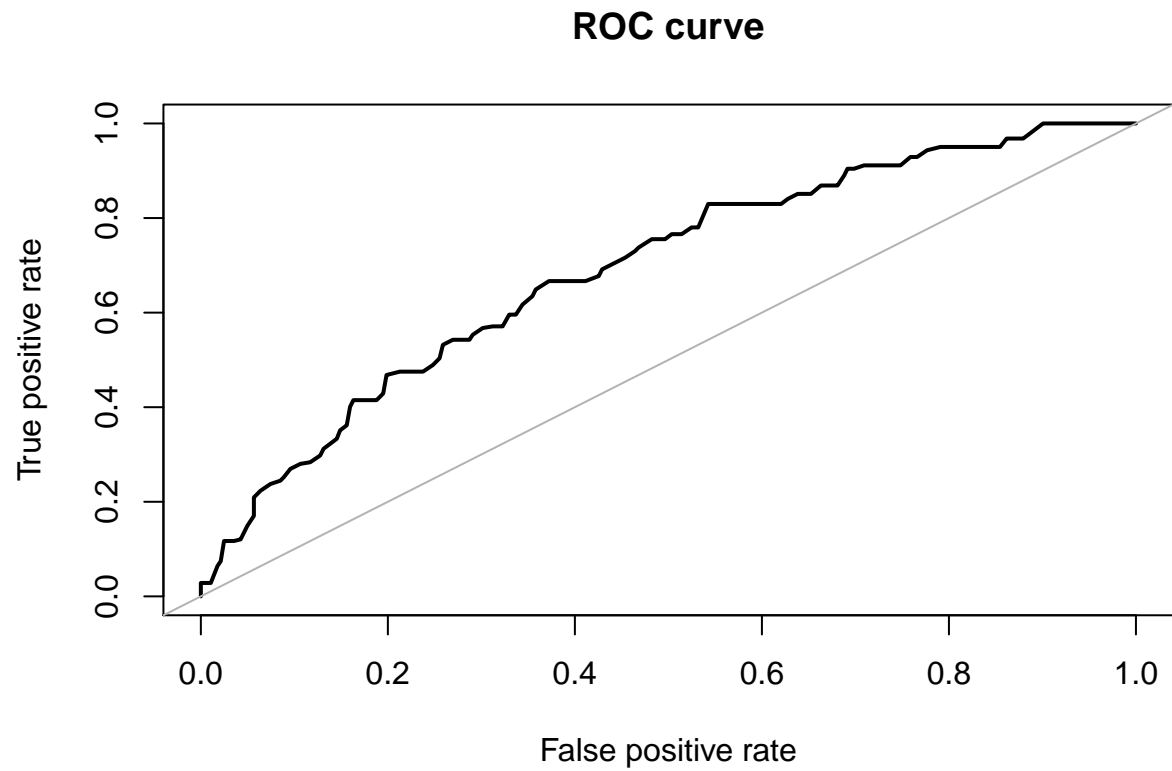
```
prop.table(table(over$FRACTURE))
```

```
##  
##    0    1  
## 0.5 0.5
```

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

```
##  
## Call:  
## accuracy.meas(response = over$FRACTURE, predicted = pred.fit.dataset,  
##      threshold = 0.05)  
##  
## Examples are labelled as positive when predicted is greater than 0.05  
##  
## precision: 0.500  
## recall: 1.000  
## F: 0.333
```

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



```
## Area under the curve (AUC): 0.691
```