INFO-Assignment2

Daniel Prvanov

2022-08-14

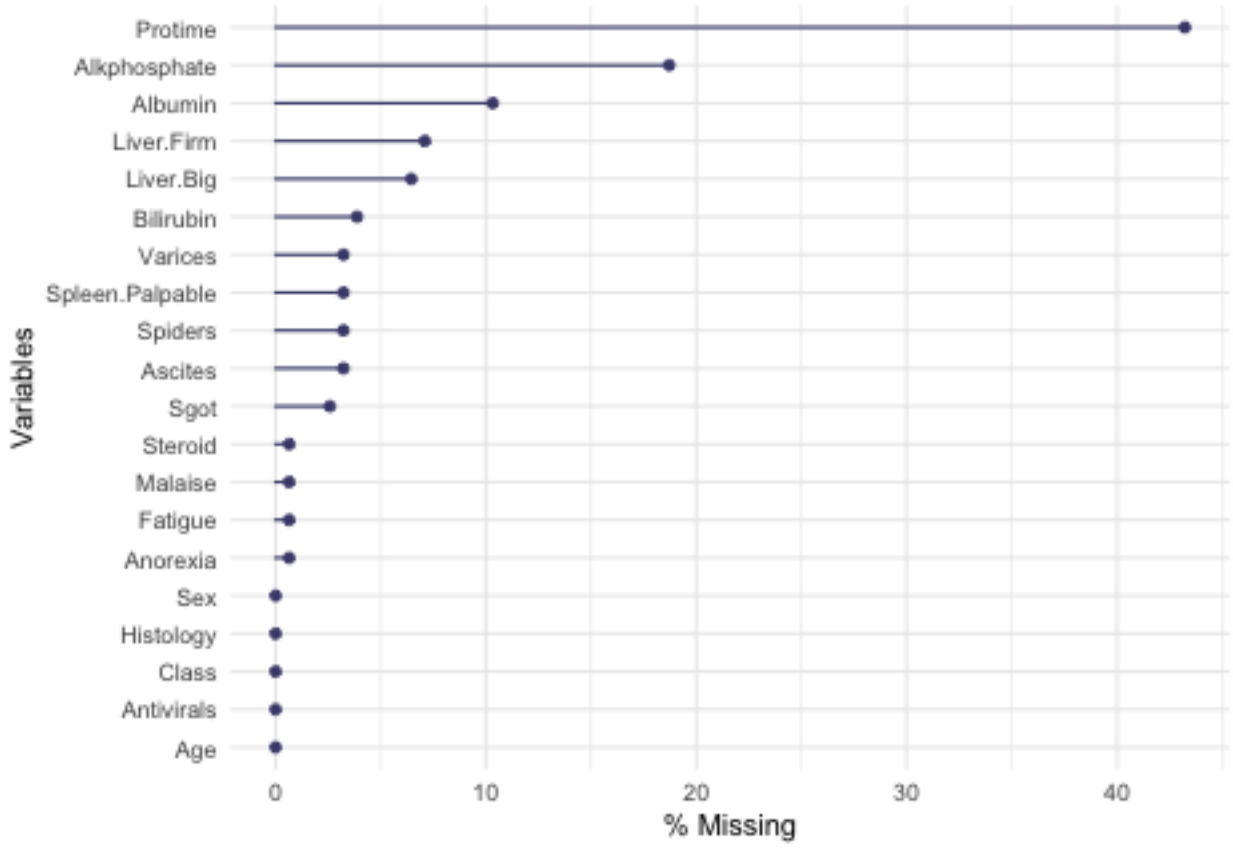
R Markdown

```
library(naniar)  # install.packages("naniar") if you don't have it.
library(ggplot2) # package naniar produces ggplots
```

```
hepatitis <-read.csv("hepatitis.data")
hepatitis <- data.factorise(hepatitis, factor.cols = c(1,3:14,20))
```

Question One (10 marks)

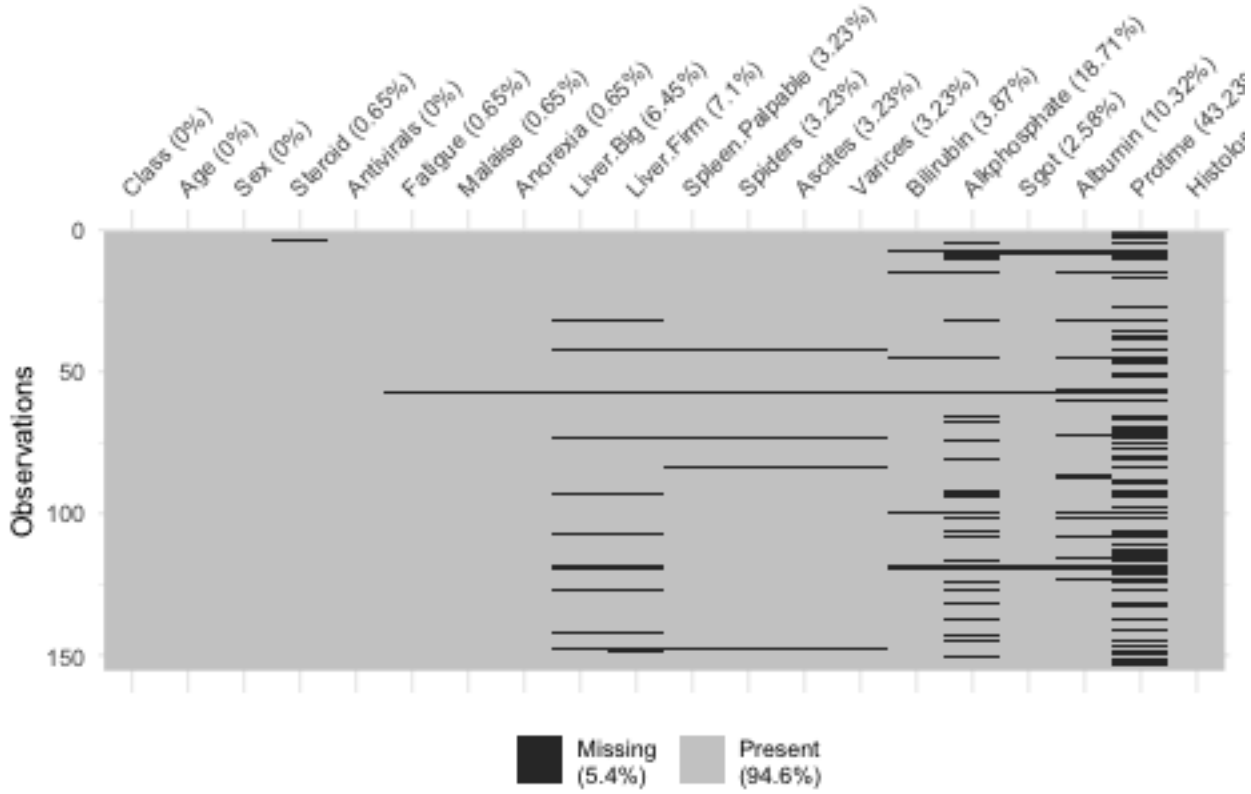
```
gg_miss_var(hepatitis,show_pct=TRUE)
```



```
print(paste("Number of NA's in Hepatitis:",length(which(is.na(hepatitis)==TRUE))))
```

```
## [1] "Number of NA's in Hepatitis: 167"
```

```
vis_miss(hepatitis)
```



From this we can see that a large percentage of variables of "Protime" are missing (43.23%) and a total of 167 variables (5.4%) are missing.. We can also see that that a few columns are missing several rows worth of data, there is a row around the 50th obsevation missing all variables apart "Class", "Age", "Sex" and "Steroid" it probably would be smart to remove this observation as it does not contain much information. The variables "Age", "Antivirals", "Class", "Histology", and "Sex" are all missing zero variables.

Question Two (10 marks)

The explanatory variable "Sex" needs to be removed because all rows that Sex is female (two), the class also equal to Survived (two) Therefore they are directly correlated and the model would predict that if someone is female that they will also survive, which in the real world is not correct and is only correct here to due to the small sample size of females. That is why "Sex" needs to be removed.

```
hepatitis <- subset(hepatitis, select = -c(Sex))
```

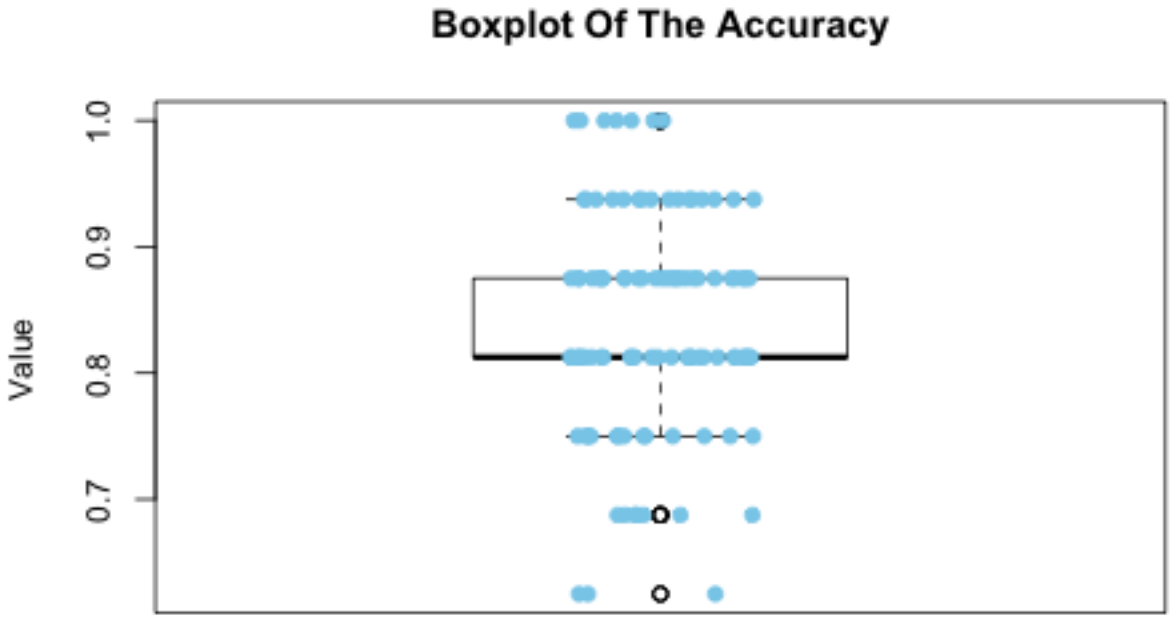
Question Three (45 marks) - Logistic Regression

I Have already turned variables to factors and removed the explanatory variable "Sex".

```
library(mice)
imps <- make.imputations(hepatitis) #3.3 Create an imputed data set for the hepatitis data set
x = replicate(100,test.logistic(imps, Class ~ .)) #3.4 Estimate the accuracy of a logistic model using
```

Comment On The Accuracy Of The Model

```
boxplot(x, ylab = "Value", main = "Boxplot Of The Accuracy", col = "white") #boxplot
stripchart(x, method = "jitter", vertical = TRUE, add = TRUE, pch = 19, col = "skyblue")
```




```
print(paste("the mean of the accuracy is",mean(x))) #mean accuracy
```

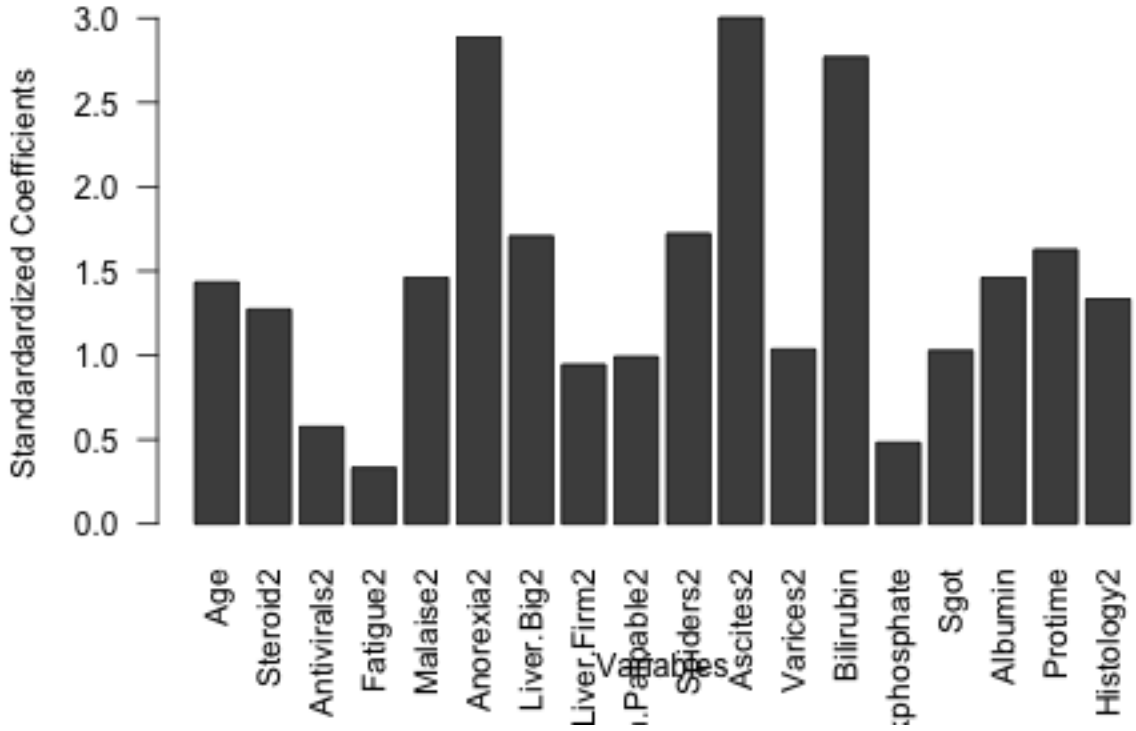
```
## [1] "the mean of the accuracy is 0.839375"
```

When running the function 100 times I got a mean accuracy of 0.85875 (Note: the accuracy is not always 0.85875, as if rerun make.imputations I get similar but different results). In the Breiman and Diaconis/Efron paper they had a predictive error rate of 17.4, or an accuracy of 0.826. From this we can see that my results have a slightly better accuracy then the paper, this could be due to the removing of the "Sex" variables, as well as imputing the data. The paper does not state what they did to deal with NA values, they might have imputed the data as well not I am not sure.

3.5 Create a Barplot Showing Variable Importance For Each Explanatory

```
glm100 = replicate(100,glm.coefficients(imps, Class ~ .))
glm100 = abs(glm100)
sd = apply(glm100, 1, sd)
mean = apply(glm100, 1, mean)
barplot.values = mean/sd
```

```
barplot.values <- barplot.values[c(-1)]
barplot.values_ordered <- sort(barplot.values, decreasing = F)
barplot(t(barplot.values), las = 2, xlab = "Variables", ylab = "Standardardized Coefficients")
```



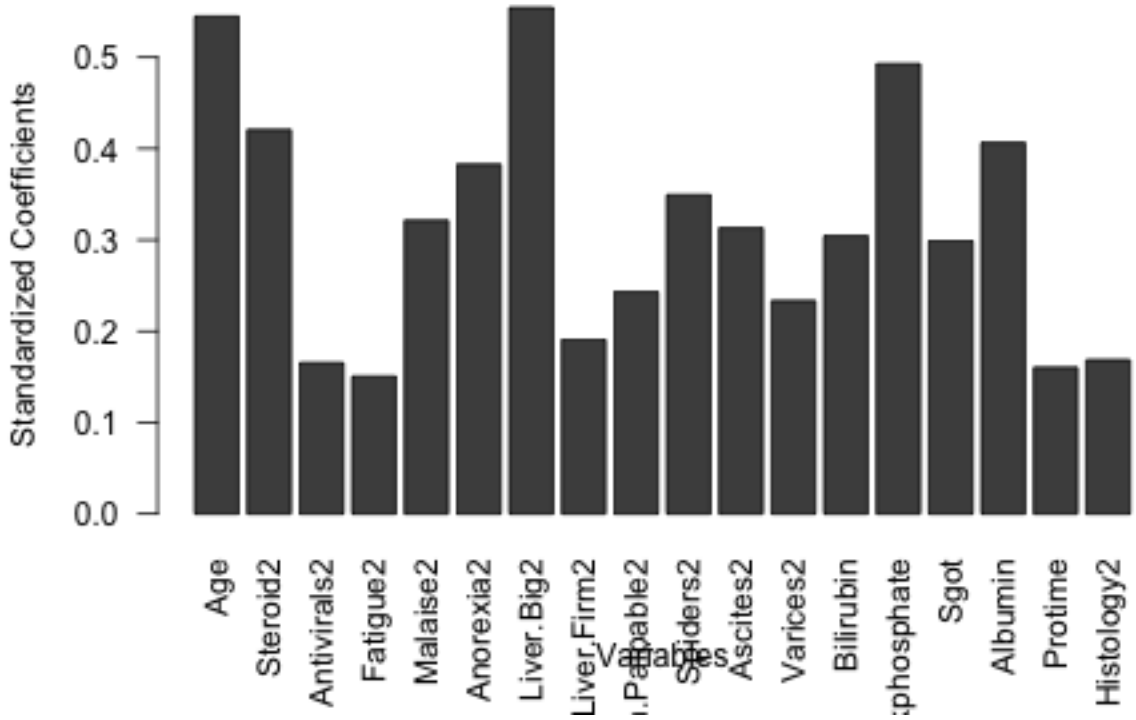
In my first graph that the three most important covariates are Anorexia, Ascites, and Bilrubin. In the Beriman paper it was stated variable 7 and 11 are the most important variables (they should relate to Anorexia and Spiders) In my model Anerexia is also one of the two most important covariates but Spiders is only my models four most important variable. The biggest difference between the two graphs is that my model thinks Ascites is a very important covariates while Beriman does not think it is. Note: all graphs that I describe will be differnet when viewed later after rerunning

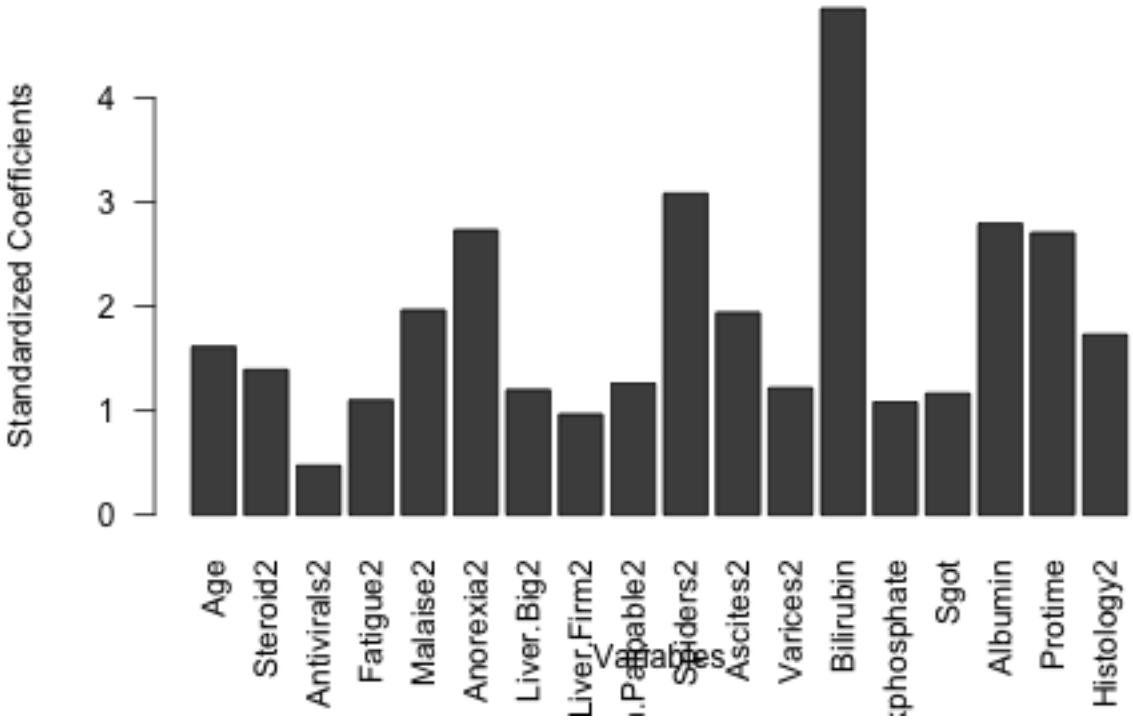
If we create five more different graphs we will get different results.

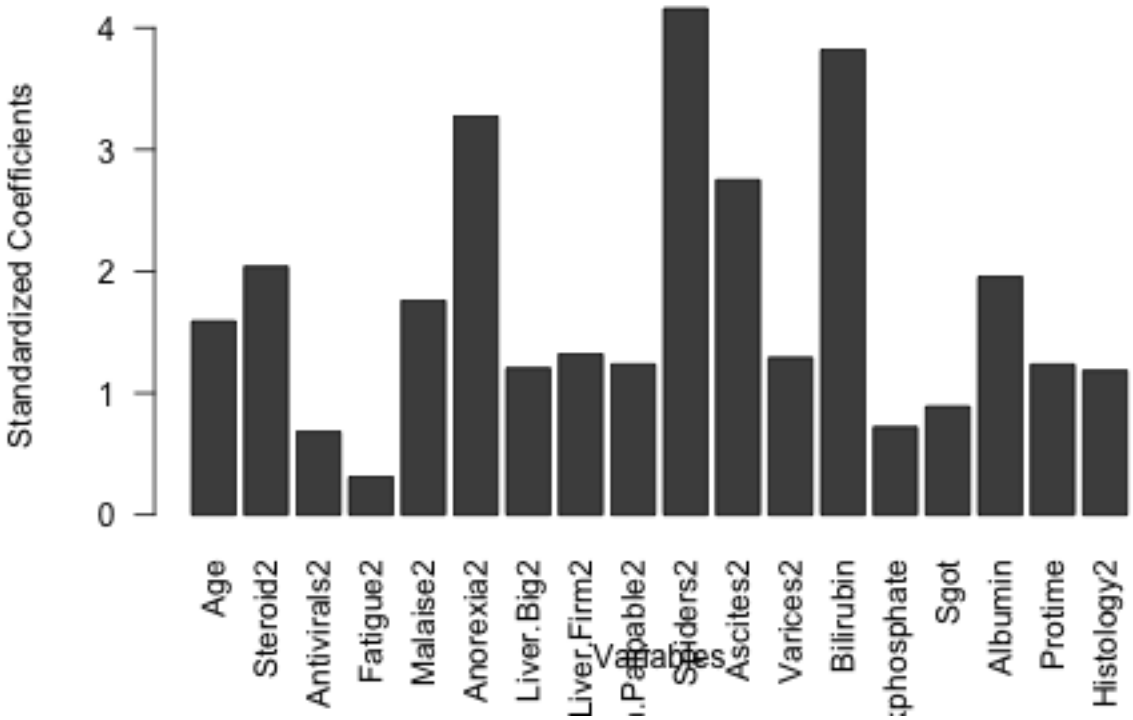
```
build.graph.values <- function(dataset, formula){
  imps <- make.imputations(dataset)
  glm100 <- replicate(100,glm.coefficients(imps, formula))
  glm100 <- abs(glm100)
```

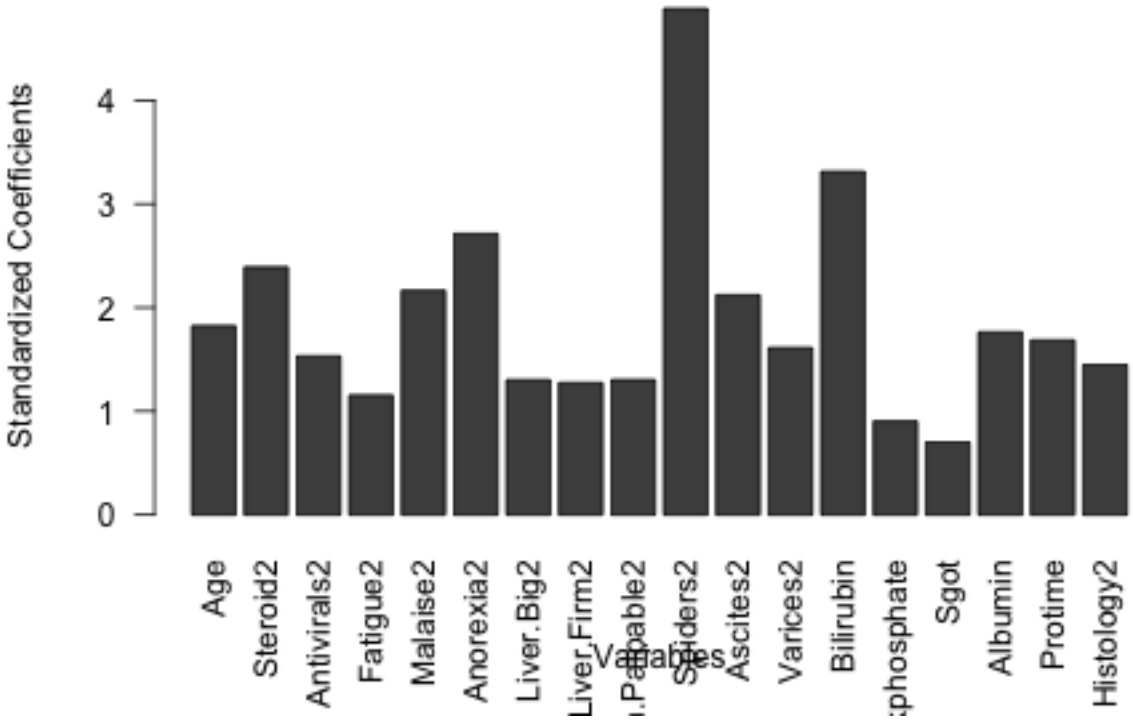
```
  sd <- apply(glm100,1,sd)
  mean <- apply(glm100,1,mean)
  barplot.values <- mean/sd
  barplot.values <- barplot.values[c(-1)]
  barplot(t(barplot.values), las =2, xlab = "Variables", ylab = "Standardized Coefficients")
}
```

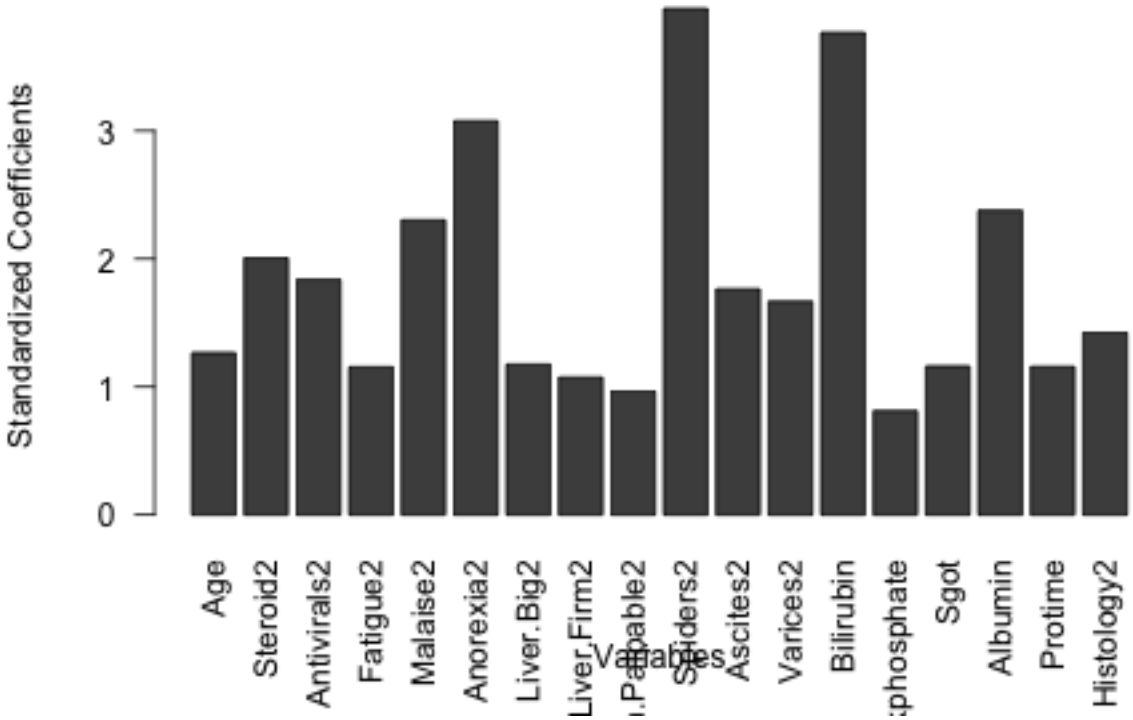
```
replicate(5,build.graph.values(hepatitis, Class ~ .))
```




Standardized Coefficients
0
1
2
3
4
Age
Steroid2
Antivirals2
Fatigue2
Malaise2
Anorexia2
Liver.Big2
Liver.Firm2
Spiders2
Ascites2
Varices2
Bilirubin
Sgot
Albumin
Protime
Histology2


Standardized Coefficients
0
1
2
3
4
Age
Steroid2
Antivirals2
Fatigue2
Malaise2
Anorexia2
Liver.Big2
Liver.Firm2
Spleen.Palpable2
Spiders2
Ascites2
Varices2
Bilirubin
Sgot
Albumin
Protime
Histology2
Variables


Standardized Coefficients
0
1
2
3
4
Age
Steroid2
Antivirals2
Fatigue2
Malaise2
Anorexia2
Liver.Big2
Liver.Firm2
Spleen.Palpable2
Spiders2
Ascites2
Varices2
Bilirubin
Sgot
Albumin
Protime
Histology2
Variables




```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]  0.7  0.7  0.7  0.7  0.7
##  [2,]  1.9  1.9  1.9  1.9  1.9
##  [3,]  3.1  3.1  3.1  3.1  3.1
##  [4,]  4.3  4.3  4.3  4.3  4.3
##  [5,]  5.5  5.5  5.5  5.5  5.5
##  [6,]  6.7  6.7  6.7  6.7  6.7
##  [7,]  7.9  7.9  7.9  7.9  7.9
##  [8,]  9.1  9.1  9.1  9.1  9.1
##  [9,] 10.3 10.3 10.3 10.3 10.3
## [10,] 11.5 11.5 11.5 11.5 11.5
## [11,] 12.7 12.7 12.7 12.7 12.7
## [12,] 13.9 13.9 13.9 13.9 13.9
## [13,] 15.1 15.1 15.1 15.1 15.1
## [14,] 16.3 16.3 16.3 16.3 16.3
## [15,] 17.5 17.5 17.5 17.5 17.5
## [16,] 18.7 18.7 18.7 18.7 18.7
## [17,] 19.9 19.9 19.9 19.9 19.9
## [18,] 21.1 21.1 21.1 21.1 21.1
```

When rerunning the model again, we get different results with differnet variables becoming the most important.

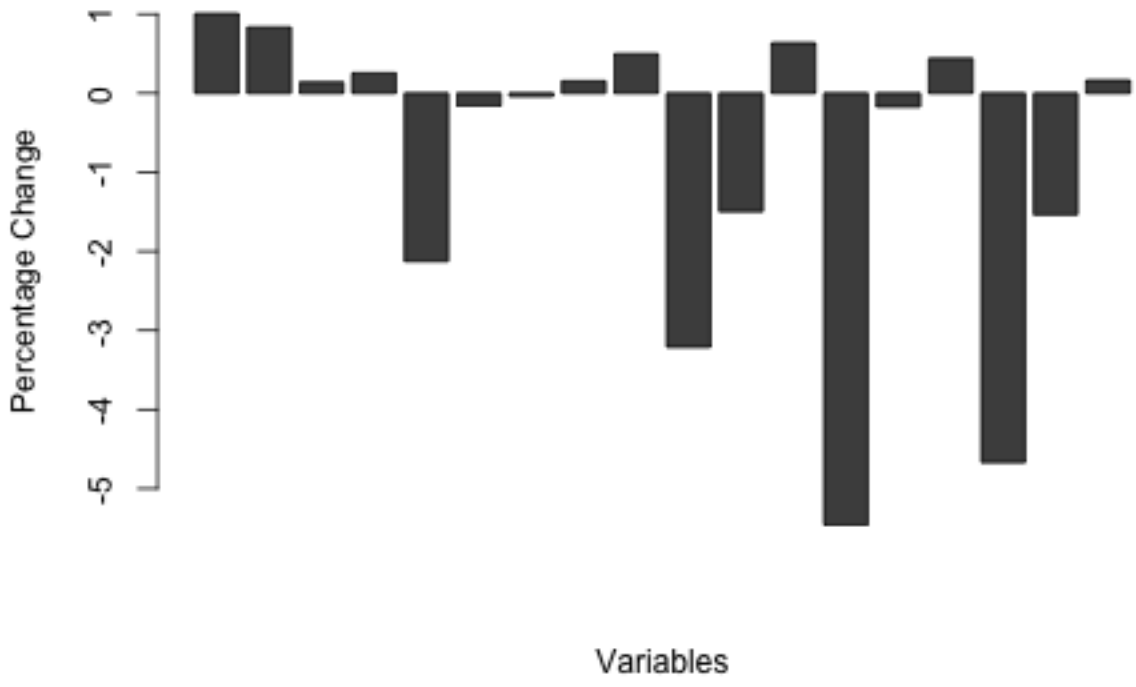
When running the model again we get different resuslts, this means that the stability of logistic regression for the hepatitis dataset isn't the best. I wouldn't be confident using this appraoch to assess variable importance due to the randomness of it. But it could be used to give a rough ball park of the variables importance. For

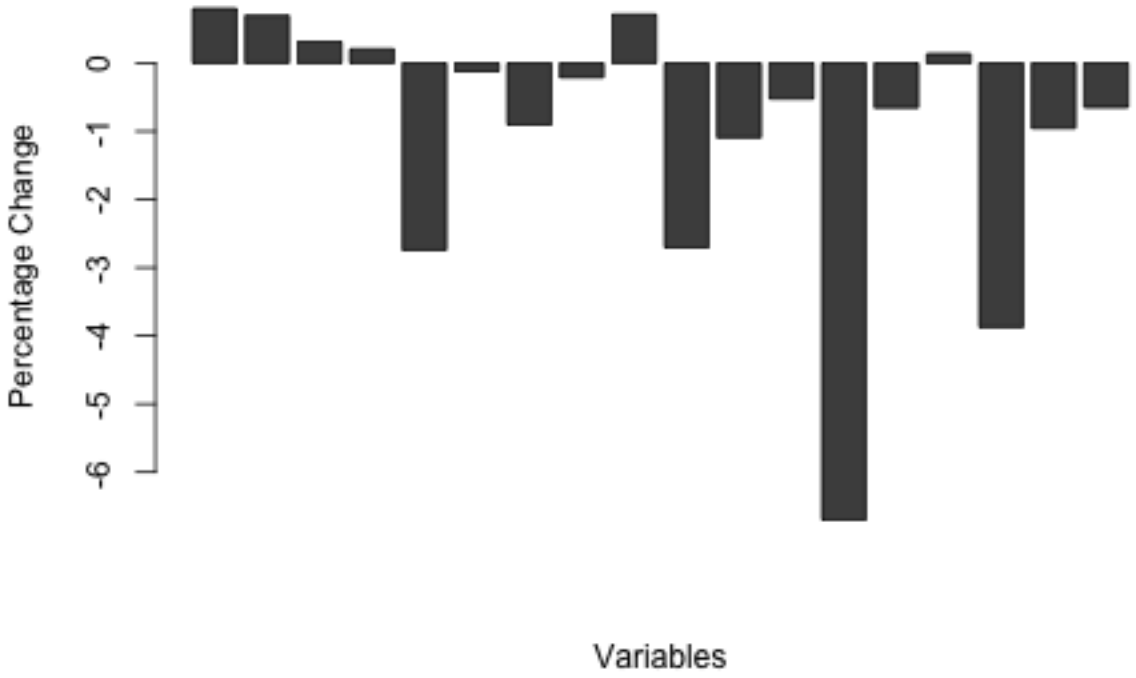
example we might not know how important the "Spiders" variable is but both times it has appeared near the top of the graph so it most likely is important even if we don't know how important it is and the same would be true for variables that have a low score.

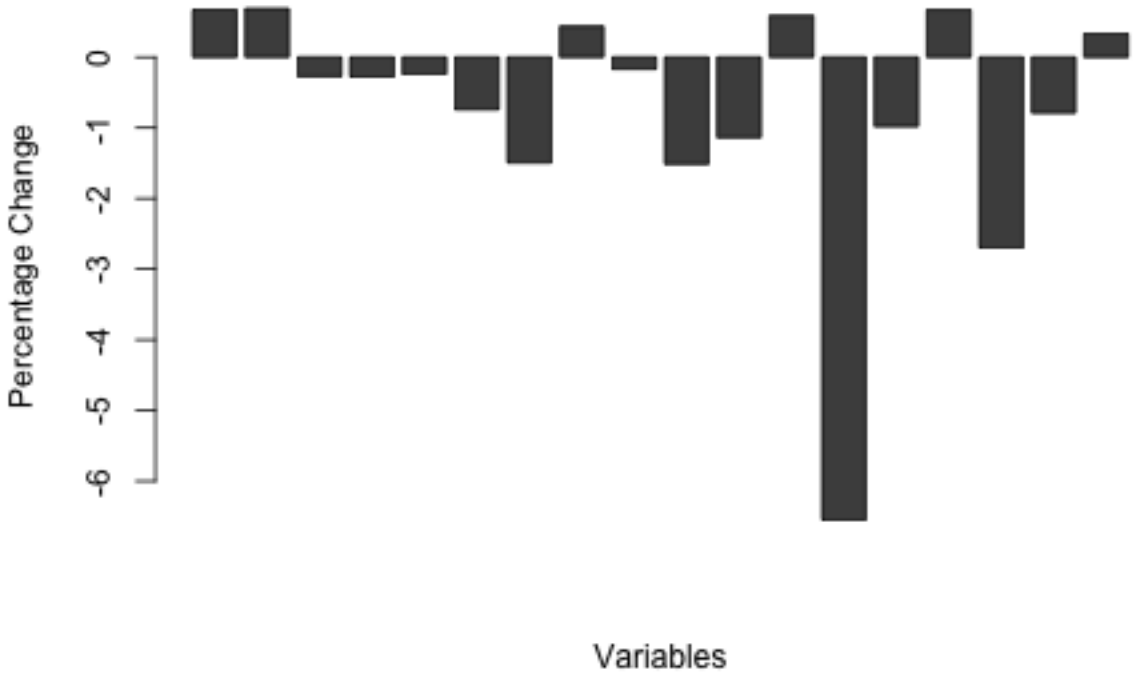
Question 3.6 (25 marks) - Implement Variable Importance Using Permutation

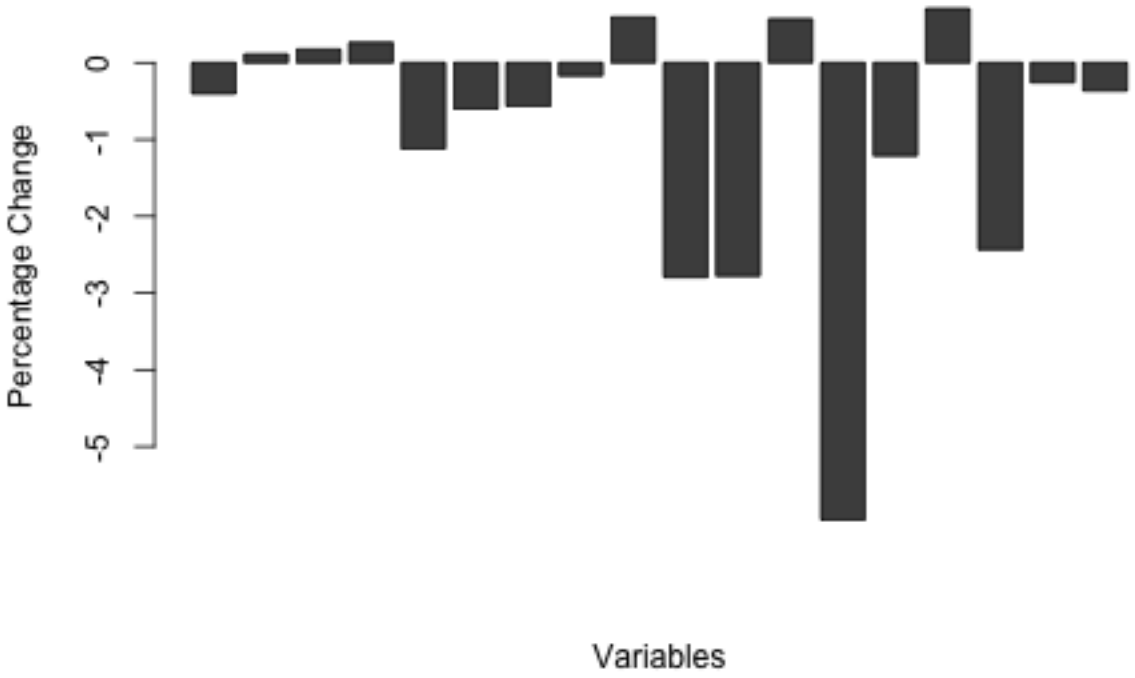
36.20 mins in

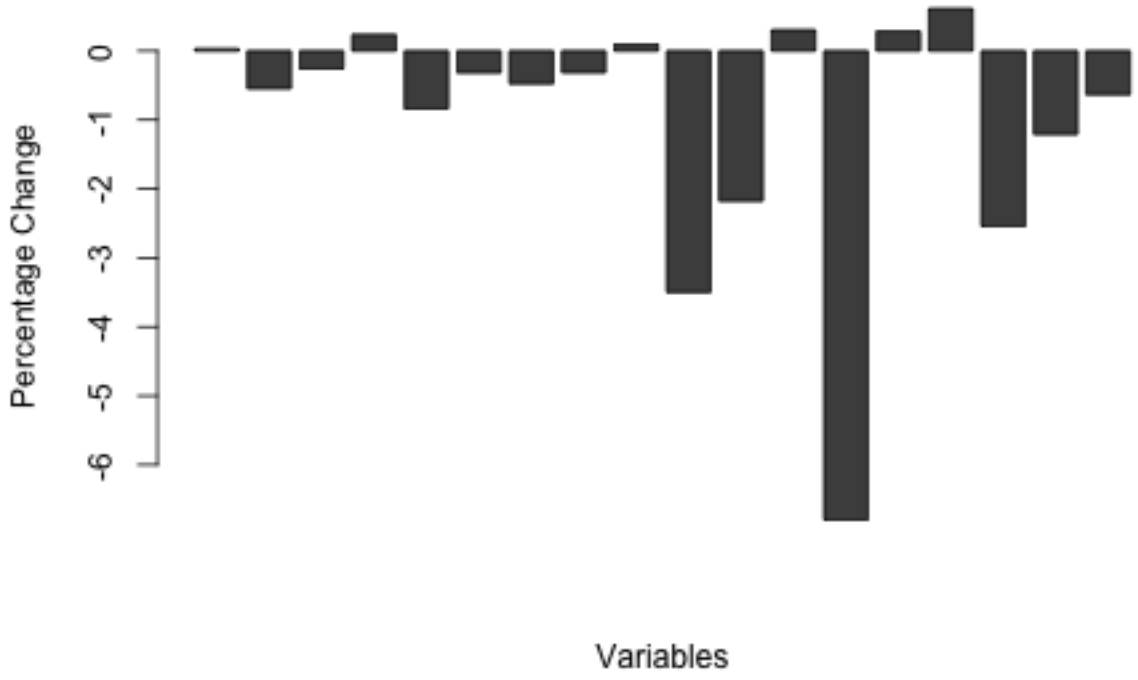
```
build.permuted.graph <- function(imp, formula){
  var.imps <- collect.var.imp(imp, formula)
  col.means <- colMeans(var.imps)
  barplot(t(col.means),xlab = "Variables", ylab = "Percentage Change")
}

replicate(5,build.permuted.graph(imps,Class ~ .))
```




Percentage Change
0
-1
-2
-3
-4
-5
-6
Variables


Percentage Change
0
-1
-2
-3
-4
-5
-6
Variables


Percentage Change
0
-1
-2
-3
-4
-5
Variables




```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]  0.7  0.7  0.7  0.7  0.7
##  [2,]  1.9  1.9  1.9  1.9  1.9
##  [3,]  3.1  3.1  3.1  3.1  3.1
##  [4,]  4.3  4.3  4.3  4.3  4.3
##  [5,]  5.5  5.5  5.5  5.5  5.5
##  [6,]  6.7  6.7  6.7  6.7  6.7
##  [7,]  7.9  7.9  7.9  7.9  7.9
##  [8,]  9.1  9.1  9.1  9.1  9.1
##  [9,] 10.3 10.3 10.3 10.3 10.3
## [10,] 11.5 11.5 11.5 11.5 11.5
## [11,] 12.7 12.7 12.7 12.7 12.7
## [12,] 13.9 13.9 13.9 13.9 13.9
## [13,] 15.1 15.1 15.1 15.1 15.1
## [14,] 16.3 16.3 16.3 16.3 16.3
## [15,] 17.5 17.5 17.5 17.5 17.5
## [16,] 18.7 18.7 18.7 18.7 18.7
## [17,] 19.9 19.9 19.9 19.9 19.9
## [18,] 21.1 21.1 21.1 21.1 21.1
```

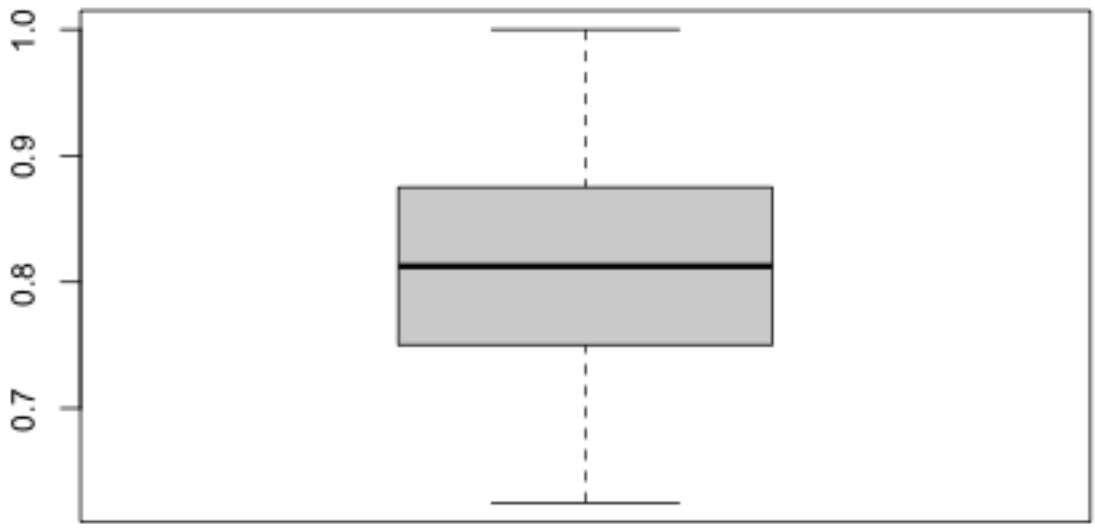
Using permutation to check for variable importance we can see that the same variables from before are still considered important and these graphs are more similar meaning the stability is greater, for example in my run the five graphs that I created show that the 13th variable (Bilrubin) has the greatest percentage change ### compare to other literature tomorrow

Question Four (20 marks) - Decision Trees and Random Forest

4.1

```
test.rpart <- function(imp, formula, perc.train=0.9){
  ttdata <- get.imp.train.test(imp, perc.train = perc.train)
  train <- ttdata[[1]]
  test <- ttdata[[2]]
  ##run model on the training data
  ##get the prediction on the test data
  ##do the confusion matrix
  ##calculate the accuracy
}
```

```
rf100 = replicate(100,test.rf(imps, Class~.)) #this shit wrong yo.
boxplot(rf100)
```




```
mean(rf100)
```

```
## [1] 0.829375
```

4.2

```
tree.one <- replicate(100,test.rf(imps, Class~., ntrees = 1))
tree.five <- replicate(100,test.rf(imps, Class~., ntrees = 5))
tree.ten <- replicate(100,test.rf(imps, Class~., ntrees = 10))
tree.twenty <- replicate(100,test.rf(imps, Class~., ntrees = 20))
```

```
mean(tree.one)
```

```
## [1] 0.781875
```

```
mean(tree.five)
```

```
## [1] 0.811875
```

```
mean(tree.ten)
```

```
## [1] 0.824375
```

```
mean(tree.twenty)
```

```
## [1] 0.821875
```

Question Five (15 marks) - Explain The Steps That You Would Take To Understand The Dataset, Prior To Building A Model For Prediction