Homework 3

Modern Applied Statistics II

Exercises (ISLR)

1. Question 4.7.1 pg 168

Using a little bit of algebra, prove that (4.2) is equivalent to (4.3). In other words, the logistic function representation and logit representation for the logistic regression model are equivalent. (page 132)

The logistic function representation for logistic regression is the following:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

The logit representation for the logistic regression model is the following:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$$

Through algebra we will prove these two representations are equivalent.

By solving for $p(X)$ in the logit representation we find that

$$p(X) = (1-p(X))(e^{\beta_0+\beta_1 X})$$

Then by substituting the value for $p(X)$ given to us by the logistic function we find that

$$p(X) = e^{\beta_0+\beta_1 X}(1 - \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}})$$

therefore,

$$p(X) = e^{\beta_0+\beta_1 X}(\frac{1+e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} - \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}) = e^{\beta_0+\beta_1 X}\frac{1}{1+e^{\beta_0+\beta_1 X}} = p(X)$$

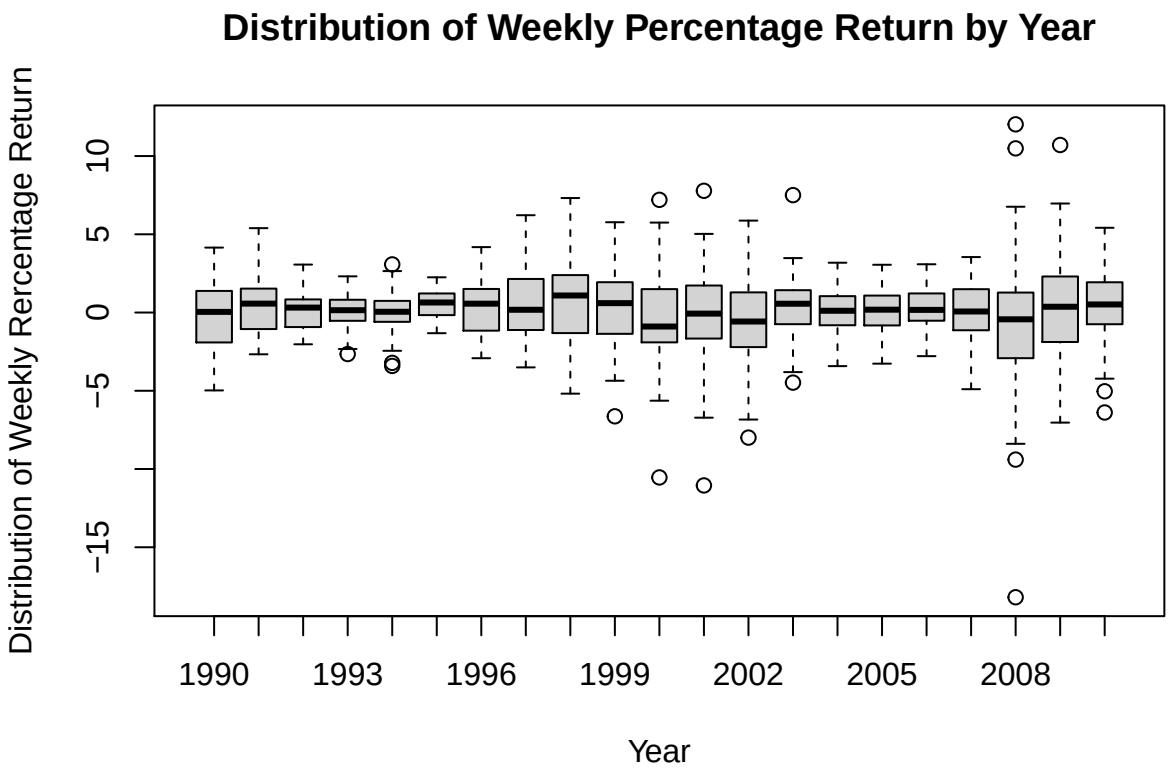
2. Question 4.7.10(a-d) pg 171

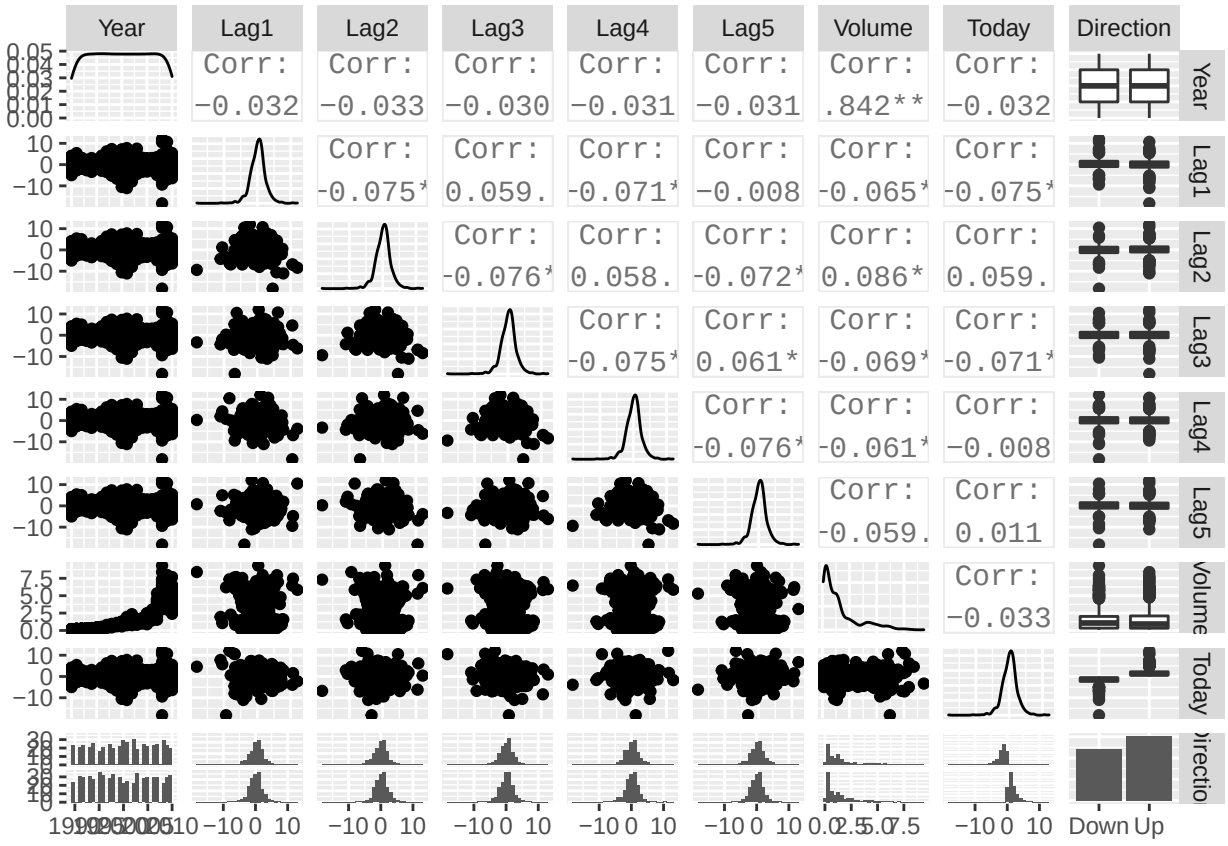
This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1, 089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

Below I printed a numerical summary of the data set Weekly, I plotted the distribution of the Weekly percentage returns for each year, and I plotted a pair plot showing the interaction between the variables. From the different summaries there are a few patterns that emerge. One interesting pattern to note is that despite the weekly returns fluctuating throughout each year, the mean percentage return is extremely close to 0 across the board for each year in the data set. Another pattern is that each lag variable has a similar distribution and yet none are correlated with each other.

```
##       Year           Lag1               Lag2               Lag3         
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
##       Lag4               Lag5              Volume            Today        
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540  
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410  
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499  
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050  
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260  
##  Direction 
##  Down:484  
##  Up  :605  
##            
##            
##            
##
```


Distribution of Weekly Percentage Return by Year
Distribution of Weekly Rercentage Return
10
5
0
−5
−15
1990
1993
1996
1999
2002
2005
2008
Year



(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

By referencing the summary of the linear model used to perform logistic regression on the weekly data set, we can see there is one predictor that appears to be statistically significant which is "Lag2". "Lag2" has a p-value with 0.01 level of significance suggesting we can reject the null hypothesis for this predictor, in this model.

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial(), data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
```

```
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

According to the confusion matrix printed below, we can see the overall percentage of correctly predicted values is 56%. The sensitivity of the model, however, is only around 11% and the specificity of the model is at 92%. What this tells us is that the model has a high false positive rate. The model is going to tell us the market will go up more than is likely which isn't the best for making solid predictions. When taking a look at the ROC curve of a model, we want it hug the top left meaning the model has a high true positive and a low false negative rate (Gareth, 148). The specificity of the model is quite high meaning it predicted most of the correct instances where the market did see an increase. To improve this model we would need to help it better determine when the market is going to decrease.

```
## $`Confusion Matrix`
##       pred
## valCol Down  Up
##   Down   54 430
##   Up     48 557
##
## $`Overall Percentage Correct`
## [1] 56.10652
##
## $Sensitivity
## [1] 11.15702
##
## $Specificity
## [1] 92.06612
```

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

After fitting a model using "lag2" as the only predictor we can see this model performs much better than the previous one. The overall percentage of correct predictions increased to 62%, the sensitivity of the model increased to 20%, and the specificity stayed relatively the same at around 91%. This means this second model was better at determining when the market would decrease and still maintained its ability to accurately detect when the market would increase.

```
## $`Confusion Matrix`
##       pred
```

```
## valCol Down Up
##   Down    9 34
##   Up      5 56
##
## $`Overall Percentage Correct`
## [1] 62.5
##
## $Sensitivity
## [1] 20.93023
##
## $Specificity
## [1] 91.80328
```

3. Question 4.7.11(a,b,c,f) pg 172

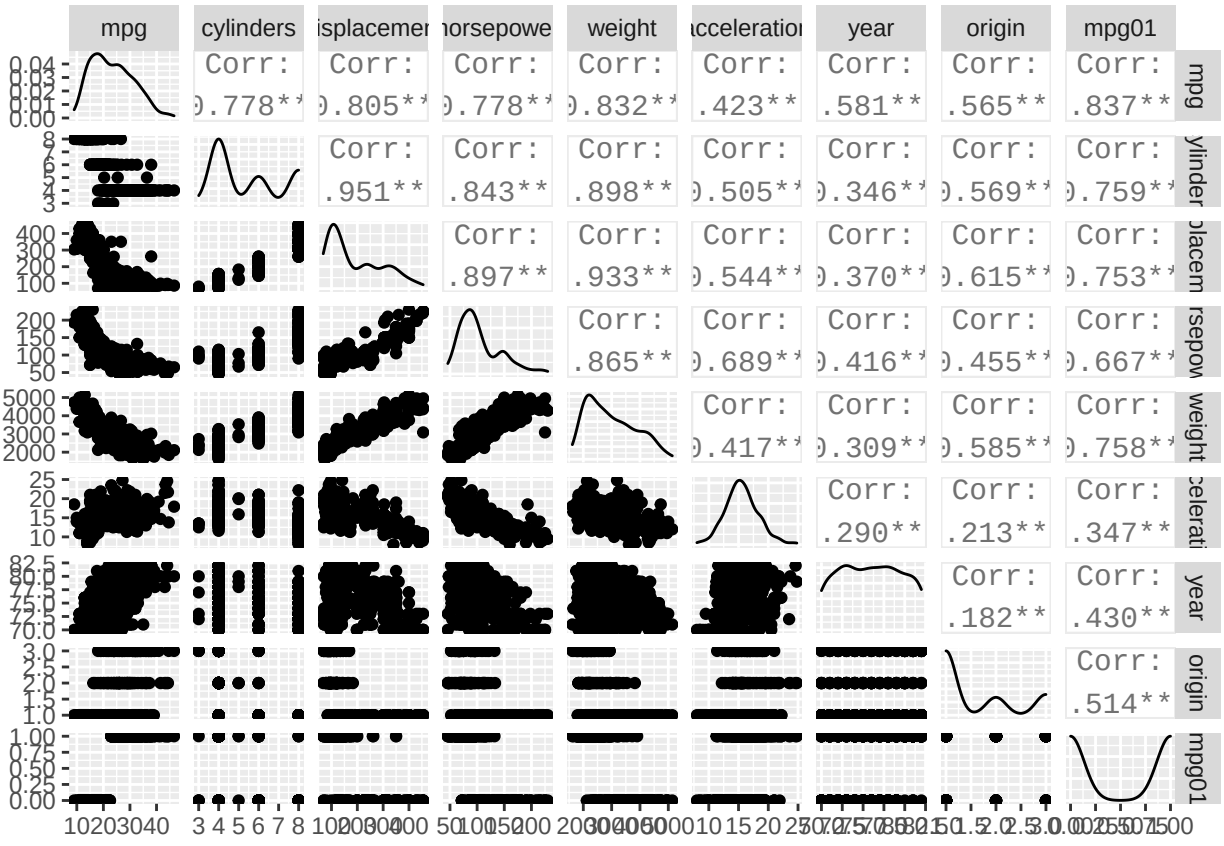
In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.

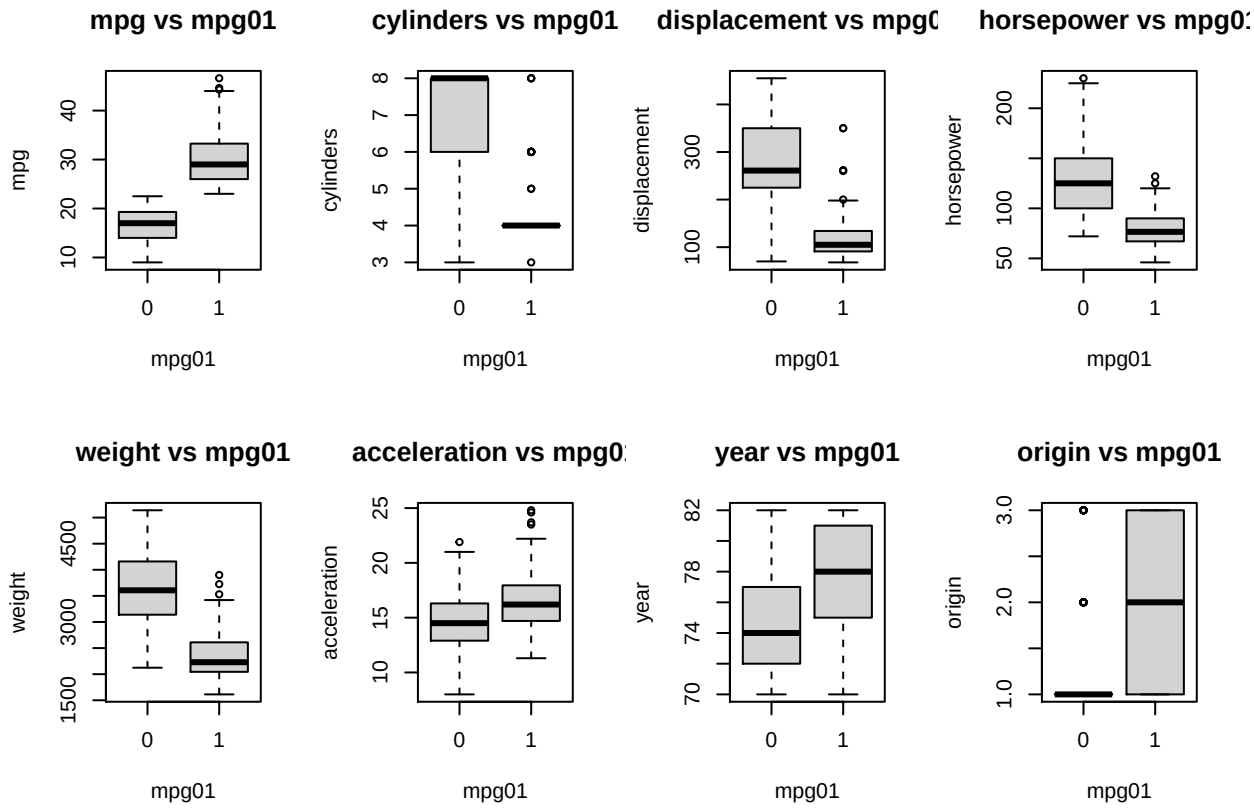
(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

```
## Variable 'mpg01' created successfully
```

(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

Below I printed a pair plot of the continuous variables to see how they interact with each other. Also, I printed boxplots of the continuous variables against the response "mpg01" in order to see if there was a significant difference in the two groups for each variable. As we can see, there is a strong relationship between the predictors mpg, cylinders, displacement, horsepower, and weight. However, we do not want to use mpg because mpg01 is derived from it and therefore would lead to collinearity within the data. So, we go with the other four variables to fit our model with.


mpg
cylinders
isplacemer
horsepowe
weight
cceleratio
year
origin
mpg01
Corr:
.778**
.805**
.778**
.832**
.423**
.581**
.565**
.837**
.951**
.843**
.898**
.505**
.346**
.569**
.759**
.897**
.933**
.544**
.370**
.615**
.753**
.865**
.689**
.416**
.455**
.667**
.417**
.309**
.585**
.758**
.290**
.213**
.347**
.182**
.430**
.514**



(c) Split the data into a training set and a test set.

Below I split the data into a training and test set with a 70/30 split using randomly generated samples.

(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

Below I fit a glm() with a binomial family function on the training set in order to make predictions on the test set. I then was able to calculate the test error and accuracy of the model by creating a confusion matrix and taking the properly corresponding values from there. The test error can be calculated by taking the number of predictions that were made and dividing that number by the number of incorrect predictions and then multiplying that by 100 to give a percentage. As we can see, the test error for this model was around 13%.

```
##
## Call:
## glm(formula = mpg01 ~ cylinders + displacement + horsepower +
##     weight, family = binomial(), data = train.auto)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6044  -0.1309   0.0363   0.2844   3.2258
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  13.928774   2.392608   5.822 5.83e-09 ***
## cylinders     0.029701   0.458499   0.065  0.94835
## displacement -0.014736   0.010888  -1.353  0.17594
## horsepower   -0.050746   0.018871  -2.689  0.00716 **
## weight       -0.002383   0.000911  -2.616  0.00889 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 379.84  on 273  degrees of freedom
## Residual deviance: 126.40  on 269  degrees of freedom
## AIC: 136.4
##
## Number of Fisher Scoring iterations: 7
```

```
## Test Error: 13.55932
```

```
## Test Accuracy: 86.44068
```

4. Write a reusable function in RMD that calculates the misclassification rate, sensitivity, and specificity, and return a table similar to `Table 4.7`. Call this function `misclass.fun.*`, replacing `*` with your initials. The arguments for this function are a threshold, predicted probabilities, and original binary response data. Test your function using the data and model from 4.7.10 b) with threshold values of `c(0.25, 0.5, 0.75)`.

The formulas used for calculating specificity and sensitivity are the following: Sensitivity = TP / (TP + P), and Specificity = TN / (FP + TN).

```
misclass.fun.RF <- function(thresh, pred_prob, resData){
  # vector of 0's
  pred <- rep(0, length(pred_prob))
  # predictions based upon threshold
  pred[pred_prob > thresh] = 1

  # confusion matrix table
  tab <- table(resData, pred)
  # misclassification rate
  misclass <- (1-sum(diag(tab))/sum(tab))*100
  # Sensitivity calculations
  sens <- (tab[4]/(tab[4] + tab[2]))*100
  # Specificity calculations
  spec <- (tab[3]/(tab[3] + tab[1]))*100

  # dataframe used to return a table like Table 4.7 in ISLR pg 149

  lab <- c("Misclassification Rate", "Sensitivity","Specificity" )
  vals <- c(misclass, sens, spec)
  df.dat <- data.frame(Metric=lab, Value=vals)

  return(kable(df.dat, caption = paste("Model with Threshold", thresh)))
}
```

Here I am testing my function with the threshold values of c(0.25, 0.5, 0.75). We can see with 0.25 we get NA values. This is because all of the predictions were turned into 1's with the threshold that low meaning it is too low. For the threshold value of 0.5 we can see we get a model that has a misclassification rate of 43% but a sensitivity rate of 92% and specificity of 88% making this the best performing model of the three. Lastly, we can see the metrics drastically decrease with the threshold of 0.75 meaning it is too high for this problem.

```
##
##
## Table: Model with Threshold 0.25
##
## |Metric                 |    Value|
## |:----------------------|--------:|
## |Misclassification Rate | 55.55556|
## |Sensitivity            |       NA|
## |Specificity            |       NA|
##
##
## Table: Model with Threshold 0.5
##
## |Metric                 |    Value|
## |:----------------------|--------:|
## |Misclassification Rate | 43.89348|
## |Sensitivity            | 92.06612|
## |Specificity            | 88.84298|
##
##
## Table: Model with Threshold 0.75
##
## |Metric                 |      Value|
## |:----------------------|----------:|
## |Misclassification Rate | 55.4637282|
## |Sensitivity            |  0.3305785|
## |Specificity            |  0.2066116|
```

sources

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. (2013). An introduction to statistical learning : with applications in R. New York :Springer,