FinalProjectPartII

Javier Mencia

2024-03-20

Data import

```
accidents <- read.csv('Accidents.csv')
bikers <- read.csv('Bikers.csv')
data <- merge(accidents, bikers, by = "Accident_Index")


# Set seed for reproducibility
set.seed(1008476114)

# Generate a vector of row indices for the training set

train_indices <- sample(nrow(data), 0.5 * nrow(data))  # 50% for training, adjust as needed

# Create the testing set (remaining rows not in the training set)

test_set <- data[-train_indices, ]

# Create the training set

data <- data[train_indices, ]
```

Data cleaning

```
cols_to_exclude <- setdiff(names(data), "Date")

# Replace "Unknown" with NA in columns except for "Date"
data[, cols_to_exclude][data[, cols_to_exclude] == "Unknown" | data[, cols_to_exclude]=="Other" | data[

# Check the missing values for each column
initialrow<- nrow(data)
sapply(data, function(x)sum(is.na(x)))
```

```
##       Accident_Index   Number_of_Vehicles Number_of_Casualties
##                    0                    0                    0
##                 Date                 Time          Speed_limit
##                    0                    0                    0
##      Road_conditions   Weather_conditions                  Day
##                  814                17995                    0
##            Road_type     Light_conditions               Gender
##                15436                    0                   63
```

```
##             Severity              Age_Grp
##                    0                    0
```

```
#Delete missing data
data<- na.omit(data)
completerows<- nrow(data)
#See proportion of data that was removed
(initialrow-completerows)/completerows
```

```
## [1] 0.07687145
```

```
initialrow-completerows
```

```
## [1] 29548
```

Data Wrangling

Change format of some of the data

```
data<- data %>%
  mutate(Severity = ifelse(Severity %in% c("Fatal"), 1, 0))%>%
  mutate(Age_Grp = ifelse(Age_Grp == "6 to 10", "06 to 10", Age_Grp))%>%
  mutate(Gender = ifelse(Gender=="Male", 1, 0))%>%filter(Speed_limit != 660)

head(data)
```

```
##   Accident_Index Number_of_Vehicles Number_of_Casualties       Date  Time
## 1  200001EK01069                  2                    1 2000-11-08 19:37
## 2  2018320335651                  2                    1 2018-07-16 05:00
## 3  199737G000705                  2                    1 1997-04-15 19:25
## 4  199117K101476                  2                    1 1991-08-31 19:20
## 5  198797LC65305                  2                    1 1987-07-05 20:15
## 6  199601XH00726                  2                    2 1996-07-30 11:40
##   Speed_limit Road_conditions Weather_conditions      Day          Road_type
## 1          30             Dry              Clear   Friday Single carriageway
## 2          30             Dry              Clear   Monday          Slip road
## 3          30             Dry              Clear  Tuesday Single carriageway
## 4          30             Dry              Clear Saturday Single carriageway
## 5          60             Dry              Clear Thursday Single carriageway
## 6          30             Dry              Clear  Tuesday Single carriageway
##   Light_conditions Gender Severity  Age_Grp
## 1         Daylight      1        0 06 to 10
## 2         Daylight      0        0 46 to 55
## 3         Daylight      1        0 06 to 10
## 4         Daylight      1        0 36 to 45
## 5         Daylight      1        0 11 to 15
## 6         Daylight      1        1 11 to 15
```

Now create a new variable for time of day:

```r
time_to_seconds <- function(time_str) {
  time_parts <- strsplit(time_str, ":")[[1]]  # Split time string into hours and minutes
  hours <- as.numeric(time_parts[1])  # Extract hours
  minutes <- as.numeric(time_parts[2])  # Extract minutes

  seconds <- hours * 3600 + minutes * 60  # Convert hours and minutes to seconds
  return(seconds)
}

# Apply the function to the 'Time' column and create a new column 'time_data' with the number of second
time_data <- sapply(data$Time, time_to_seconds)


# Use case_when() to categorize time into intervals
data$TimeDay <- case_when(
  time_data < 6 * 3600 | time_data >22 * 3600~ "Night",
  time_data < 9 * 3600 ~ "Morning",
  time_data < 14 * 3600 ~ "Midday",
  time_data < 18 * 3600 ~ "Afternoon",
TRUE ~ "Evening"
)
data <- data %>%
  mutate(Weekend = ifelse(Day %in% c("Saturday", "Sunday"), 1, 0))
```

Make data usable:

```r
#cleandata <- data%>% #Select variables
#  dplyr::select(Severity, Age_Grp,  Number_of_Casualties, Number_of_Vehicles, Weather_conditions, , Ro

data<- subset(data, select = -c(Date, Accident_Index, Time))
#Removing accident index, because it does not contain any information on the accident
#We created a variable for Time of Day, so we can remove time
#We coded a variable for weekend or not so we can remove day
#Date should also be removed as it is in date format
head(data)
```

```
##   Number_of_Vehicles Number_of_Casualties Speed_limit Road_conditions
## 1                  2                    1          30             Dry
## 2                  2                    1          30             Dry
## 3                  2                    1          30             Dry
## 4                  2                    1          30             Dry
## 5                  2                    1          60             Dry
## 6                  2                    2          30             Dry
##   Weather_conditions      Day          Road_type Light_conditions Gender
## 1              Clear   Friday Single carriageway         Daylight      1
## 2              Clear   Monday          Slip road         Daylight      0
## 3              Clear  Tuesday Single carriageway         Daylight      1
## 4              Clear Saturday Single carriageway         Daylight      1
## 5              Clear Thursday Single carriageway         Daylight      1
## 6              Clear  Tuesday Single carriageway         Daylight      1
##   Severity  Age_Grp TimeDay Weekend
## 1        0 06 to 10 Evening       0
## 2        0 46 to 55   Night       0
```

```
## 3        0 06 to 10 Evening       0
## 4        0 36 to 45 Evening       1
## 5        0 11 to 15 Evening       0
## 6        1 11 to 15  Midday       0
```

```r
data$Road_conditions <- factor(data$Road_conditions)
data$Weather_conditions <- factor(data$Weather_conditions)
data$Light_conditions <- factor(data$Light_conditions)
data$Road_type <- factor(data$Road_type)
data$TimeDay <- factor(data$TimeDay)
data$Age_Grp <- factor(data$Age_Grp)

head(data)
```

```
##   Number_of_Vehicles Number_of_Casualties Speed_limit Road_conditions
## 1                  2                    1          30             Dry
## 2                  2                    1          30             Dry
## 3                  2                    1          30             Dry
## 4                  2                    1          30             Dry
## 5                  2                    1          60             Dry
## 6                  2                    2          30             Dry
##   Weather_conditions      Day          Road_type Light_conditions Gender
## 1              Clear   Friday Single carriageway         Daylight      1
## 2              Clear   Monday          Slip road         Daylight      0
## 3              Clear  Tuesday Single carriageway         Daylight      1
## 4              Clear Saturday Single carriageway         Daylight      1
## 5              Clear Thursday Single carriageway         Daylight      1
## 6              Clear  Tuesday Single carriageway         Daylight      1
##   Severity  Age_Grp TimeDay Weekend
## 1        0 06 to 10 Evening       0
## 2        0 46 to 55   Night       0
## 3        0 06 to 10 Evening       0
## 4        0 36 to 45 Evening       1
## 5        0 11 to 15 Evening       0
## 6        1 11 to 15  Midday       0
```

```r
#Combine Age Groups into less groups:
data <- data %>%
  mutate(Age_Grp = case_when(
    Age_Grp %in% c("06 to 10", "11 to 15", "16 to 20") ~ "under 21",
    Age_Grp %in% c("21 to 25", "26 to 35", "36 to 45", "46 to 55") ~ "21 to 55",
    Age_Grp %in% c("56 to 65", "66 to 75") ~ "over 55",
    TRUE ~ as.character(Age_Grp)

  )) %>%
  mutate(DualCarriageway=ifelse(Road_type == "Dual carriageway", 1, 0))

encoded_data <- data.frame(data)
encoded_data <- subset(encoded_data, select = -c(Road_conditions, Weather_conditions, Road_type, Number

summary(encoded_data)
```

```
##   Speed_limit               Light_conditions      Gender
```

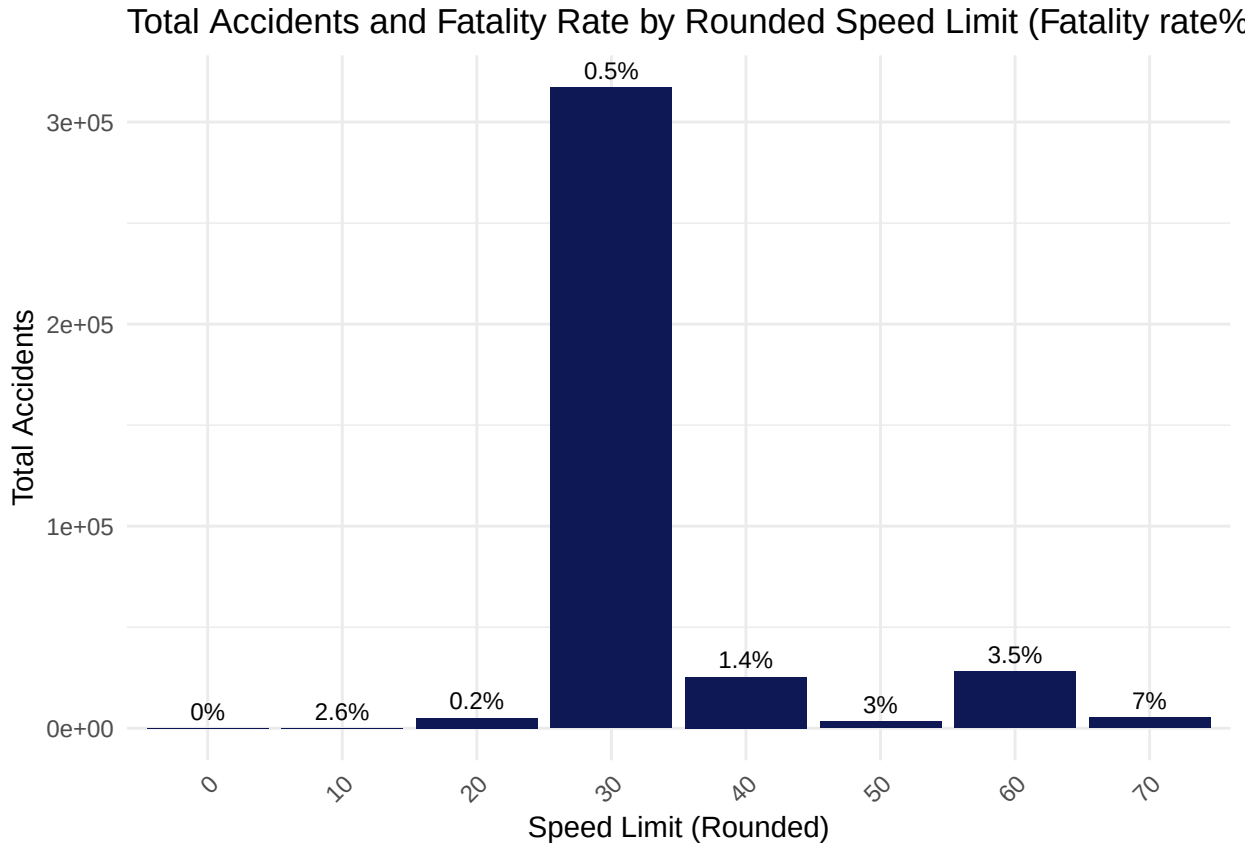
```
##  Min.   : 0.00   Darkness lights lit: 65456   Min.   :0.000
##  1st Qu.:30.00   Darkness no lights : 11134   1st Qu.:1.000
##  Median :30.00   Daylight           :307792   Median :1.000
##  Mean   :33.45                                Mean   :0.798
##  3rd Qu.:30.00                                3rd Qu.:1.000
##  Max.   :70.00                                Max.   :1.000
##     Severity          Age_Grp               TimeDay          Weekend
##  Min.   :0.000000   Length:384382      Afternoon:130720   Min.   :0.0000
##  1st Qu.:0.000000   Class :character   Evening  : 80653   1st Qu.:0.0000
##  Median :0.000000   Mode  :character   Midday   : 89028   Median :0.0000
##  Mean   :0.008481                      Morning  : 66984   Mean   :0.1951
##  3rd Qu.:0.000000                      Night    : 16997   3rd Qu.:0.0000
##  Max.   :1.000000                                         Max.   :1.0000
##  DualCarriageway
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.07412
##  3rd Qu.:0.00000
##  Max.   :1.00000
```

```r
data_summary <- data %>%
  group_by(Age_Grp) %>%
  summarise(
    count = n(),  # Count of observations in each Age Group
    relative_frequency = n() / nrow(data),  # Relative frequency of each Age Group
    fatality_rate = mean(Severity)  # Fatality rate of each Age Group
  )

# Print the summary table
print(data_summary)
```

```
## # A tibble: 3 x 4
##   Age_Grp   count relative_frequency fatality_rate
##   <chr>     <int>              <dbl>         <dbl>
## 1 21 to 55 189907             0.494        0.00763
## 2 over 55   26729             0.0695       0.0264
## 3 under 21 167746             0.436        0.00659
```

```r
data$Speed_limit_rounded <- round(data$Speed_limit, -1)

# Calculate the total number of accidents and fatality rate for each rounded speed limit
speed_summary <- data %>%
  group_by(Speed_limit_rounded) %>%
  summarise(total_accidents = n(),
            fatality_rate = mean(Severity) * 100) %>%  # Fatality rate in percentage
  arrange(Speed_limit_rounded)  # Arrange by rounded speed limit

# Create the bar chart
ggplot(speed_summary, aes(x = factor(Speed_limit_rounded), y = total_accidents, fill = NULL)) +
  geom_bar(stat = "identity", fill ="#0E1854") +
  geom_text(aes(label = paste0(round(fatality_rate, 1), "%")),
            vjust = -0.5, size = 3, color = "black") +  # Add labels for fatality rate
```

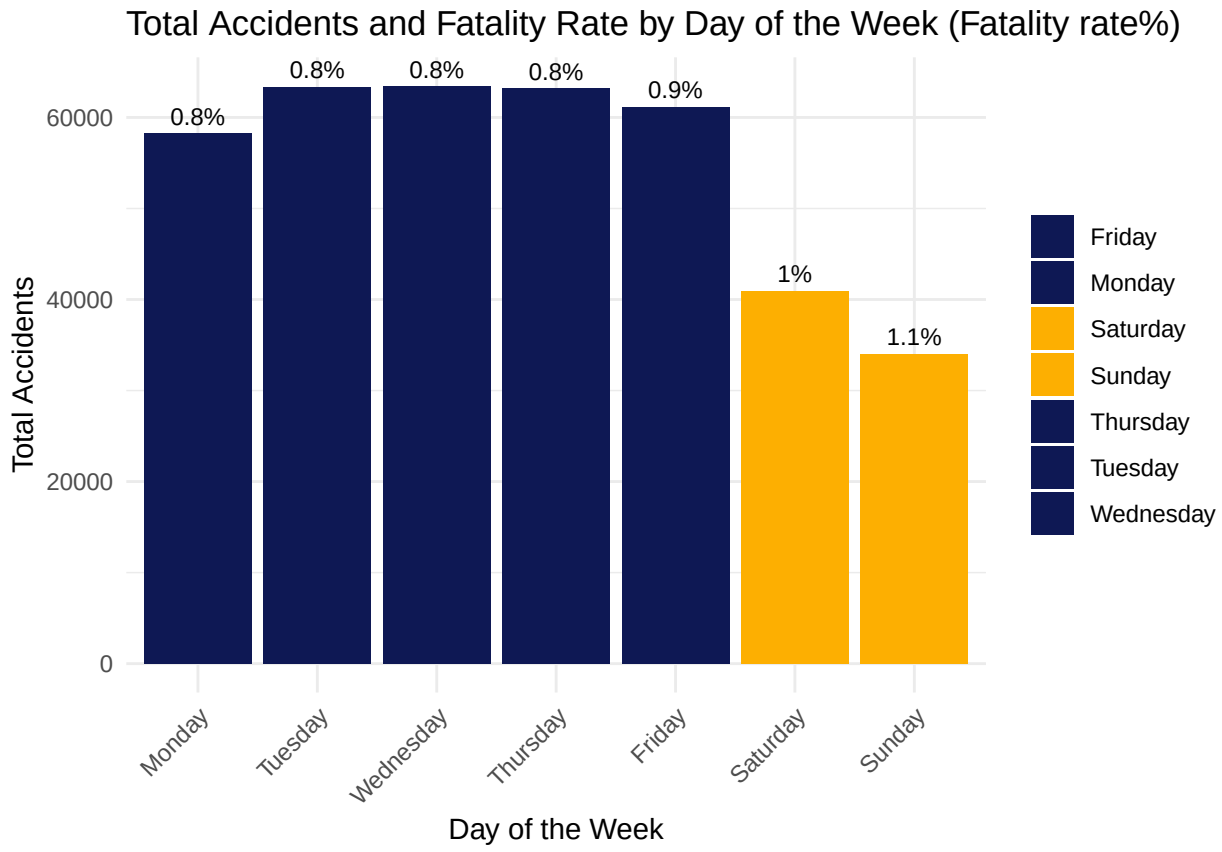
```
  labs(x = "Speed Limit (Rounded)", y = "Total Accidents", fill = NULL,
       title = "Total Accidents and Fatality Rate by Rounded Speed Limit (Fatality rate%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
data<-data[, !(names(data) %in% c("Speed_limit_rounded"))]
```

```
day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")

# Calculate the total number of accidents and fatality rate for each day of the week
accidents_summary <- data %>%
  group_by(Day) %>%
  summarise(total_accidents = n(),
            fatality_rate = mean(Severity) * 100) %>%  # Fatality rate in percentage
  arrange(factor(Day, levels = day_order))  # Arrange by the specified order of days

# Create the bar chart
ggplot(accidents_summary, aes(x = factor(Day, levels = day_order), y = total_accidents,
    fill = Day)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(fatality_rate, 1), "%")),
            vjust = -0.5, size = 3, color = "black") +  # Add labels for fatality rate
  labs(x = "Day of the Week", y = "Total Accidents", fill = NULL,
       title = "Total Accidents and Fatality Rate by Day of the Week (Fatality rate%)") +
  theme_minimal() +
```

```
  scale_fill_manual(values = c(
    '#0E1854',
    '#0E1854',
    '#FDAF01',
    '#FDAF01',
    '#0E1854',
    '#0E1854',
    '#0E1854'
  )) +  # Set the same color for all bars
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Fit a full model

```
logit.mod1 <- glm(Severity ~ ., family = binomial(link = logit), data = encoded_data)
summary(logit.mod1)
```

```
##
## Call:
## glm(formula = Severity ~ ., family = binomial(link = logit),
##     data = encoded_data)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -7.325719   0.085177 -86.006  < 2e-16 ***
## Speed_limit                    0.060397   0.001246  48.490  < 2e-16 ***
```

```
## Light_conditionsDarkness no lights  0.566819   0.072271   7.843  4.4e-15 ***
## Light_conditionsDaylight           -0.047307   0.059096  -0.801   0.4234
## Gender                             -0.032199   0.046739  -0.689   0.4909
## Age_Grpover 55                      1.206072   0.048054  25.098  < 2e-16 ***
## Age_Grpunder 21                    -0.005989   0.041323  -0.145   0.8848
## TimeDayEvening                      0.104897   0.053063   1.977   0.0481 *
## TimeDayMidday                      -0.005010   0.050325  -0.100   0.9207
## TimeDayMorning                     -0.114260   0.057615  -1.983   0.0473 *
## TimeDayNight                        0.768628   0.075462  10.186  < 2e-16 ***
## Weekend                             0.080159   0.043188   1.856   0.0634 .
## DualCarriageway                     0.498155   0.048781  10.212  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 37592  on 384381  degrees of freedom
## Residual deviance: 33529  on 384369  degrees of freedom
## AIC: 33555
##
## Number of Fisher Scoring iterations: 8
```

We see that both our predictors of interest are significant, as well as three others: Age over 55, the accident taking place with no light and the accident taking place at night.

For more specific variable selection use AIC and BIC with stepwise selection and also use Elastic net and compare them.

Use AIC first:

```
## Stepwise elimination based on AIC ##
sel.var.aic <- step(logit.mod1, trace = 0, k = 2, direction = "both")
sel.var.aic
```

```
##
## Call:  glm(formula = Severity ~ Speed_limit + Light_conditions + Age_Grp +
##     TimeDay + Weekend + DualCarriageway, family = binomial(link = logit),
##     data = encoded_data)
##
## Coefficients:
##                          (Intercept)                          Speed_limit
##                            -7.349294                             0.060345
## Light_conditionsDarkness no lights             Light_conditionsDaylight
##                             0.566149                            -0.046383
##                       Age_Grpover 55                       Age_Grpunder 21
##                             1.205502                            -0.007141
##                         TimeDayEvening                         TimeDayMidday
##                             0.104233                            -0.004088
##                       TimeDayMorning                          TimeDayNight
##                            -0.114733                             0.766967
##                              Weekend                       DualCarriageway
##                             0.079367                             0.496911
##
## Degrees of Freedom: 384381 Total (i.e. Null);  384370 Residual
## Null Deviance:       37590
```

```
## Residual Deviance: 33530     AIC: 33550
```

```
select_var_aic<-attr(terms(sel.var.aic), "term.labels")
select_var_aic
```

```
## [1] "Speed_limit"      "Light_conditions" "Age_Grp"          "TimeDay"
## [5] "Weekend"          "DualCarriageway"
```

Now BIC

```
## Stepwise elimination based on BIC ##
sel.var.bic <- step(logit.mod1, trace = 0, k = log(nrow(encoded_data)), direction = "both")
sel.var.bic
```

```
##
## Call:  glm(formula = Severity ~ Speed_limit + Light_conditions + Age_Grp +
##     TimeDay + DualCarriageway, family = binomial(link = logit),
##     data = encoded_data)
##
## Coefficients:
##                         (Intercept)                         Speed_limit
##                           -7.344789                            0.060553
## Light_conditionsDarkness no lights           Light_conditionsDaylight
##                            0.564125                           -0.043094
##                       Age_Grpover 55                      Age_Grpunder 21
##                            1.205213                           -0.003627
##                       TimeDayEvening                        TimeDayMidday
##                            0.102505                            0.005954
##                       TimeDayMorning                         TimeDayNight
##                           -0.124912                            0.775832
##                     DualCarriageway
##                            0.495233
##
## Degrees of Freedom: 384381 Total (i.e. Null);  384371 Residual
## Null Deviance:       37590
## Residual Deviance: 33530     AIC: 33550
```

```
select_var_bic<-attr(terms(sel.var.bic), "term.labels")
select_var_bic
```

```
## [1] "Speed_limit"      "Light_conditions" "Age_Grp"          "TimeDay"
## [5] "DualCarriageway"
```

COmpare them with LRT:

```
library(openintro)
```

```
## Loading required package: airports
```

```
## Loading required package: cherryblossom
```

```
## Loading required package: usdata

##
## Attaching package: 'openintro'

## The following objects are masked from 'package:MASS':
##
##     housing, mammals

## The following object is masked from 'package:survival':
##
##     transplant
```

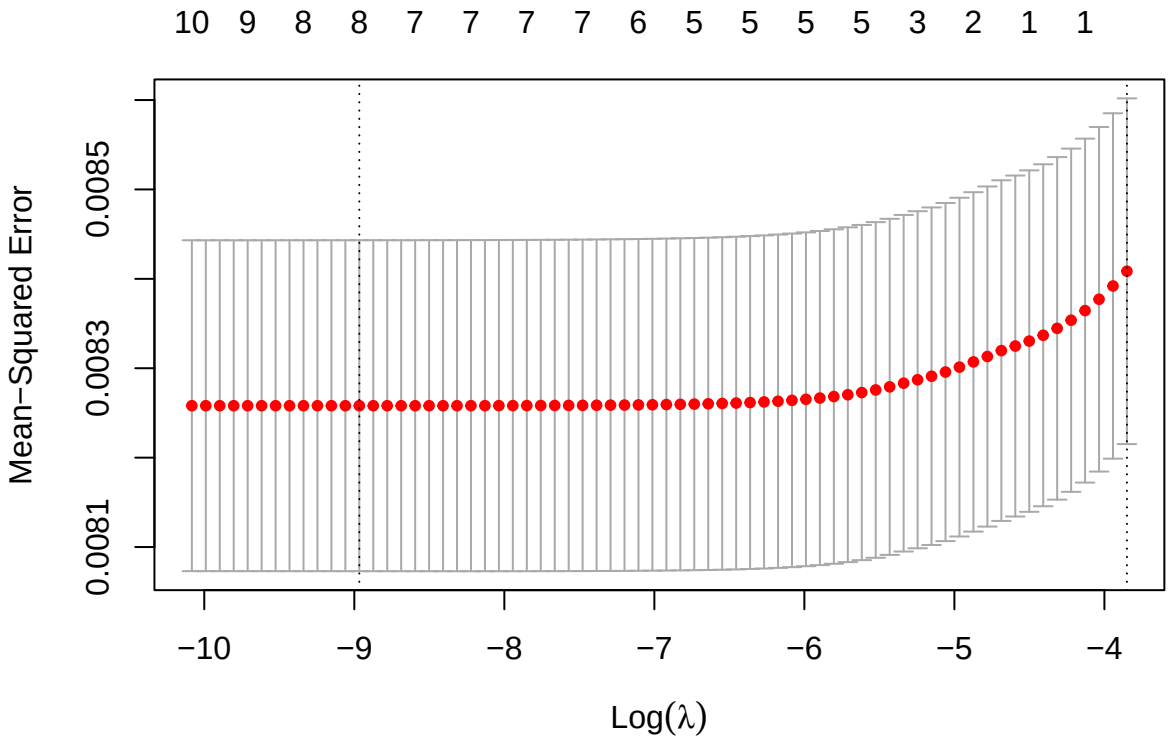
```
modAIC <- glm(Severity ~ ., data = encoded_data[,which(colnames(encoded_data) %in% c(select_var_aic, "S
modBIC <- glm(Severity ~ ., data = encoded_data[,which(colnames(encoded_data) %in% c(select_var_bic, "S

lrtest(modAIC, modBIC)
```

```
## Likelihood ratio test for MLE method
## Chi-squared 1 d.f. =  3.340058 , P value =  0.06761224
```

This p-value tells us that we should NOT reject the null hypothesis in the Likelihood ratio test (greater than the 0.05 cutoff) and conclude that the simpler model selected by BIC provides a better fit for the data than the more complex model from AIC that includes Weekend.

Now select AIC model and compare it with Elastic Net's selection

```
selected_data <- encoded_data[, c("Severity", select_var_bic)]

# Specify the columns containing categorical variables
categorical_columns <- c("Light_conditions", "Age_Grp", "TimeDay")

# Convert categorical variables to factors
selected_data[categorical_columns] <- lapply(selected_data[categorical_columns], factor)

# Convert the subsetted dataframe to matrix format with dummy variables
X <- model.matrix(~ . - Severity, data = selected_data)

# Extract the response variable (y)
y <- selected_data$Severity


# Fit Lasso model with cross-validation
cv.out <- cv.glmnet(x = X, y = y, alpha = 0.5)

plot(cv.out)
```




```
best.lambda <- cv.out$lambda.1se
best.lambda
```

```
## [1] 0.02128582
```

```
co <- coef(cv.out, s = "lambda.1se")

# Selection of the significant features (predictors)
## threshold for variable selection ##
thresh <- 0.00
# select variables #
inds <- which(abs(co) > thresh)
variables <- row.names(co)[inds]
sel.var.lasso <- variables[!(variables %in% '(Intercept)')]
sel.var.lasso
```

```
## character(0)
```

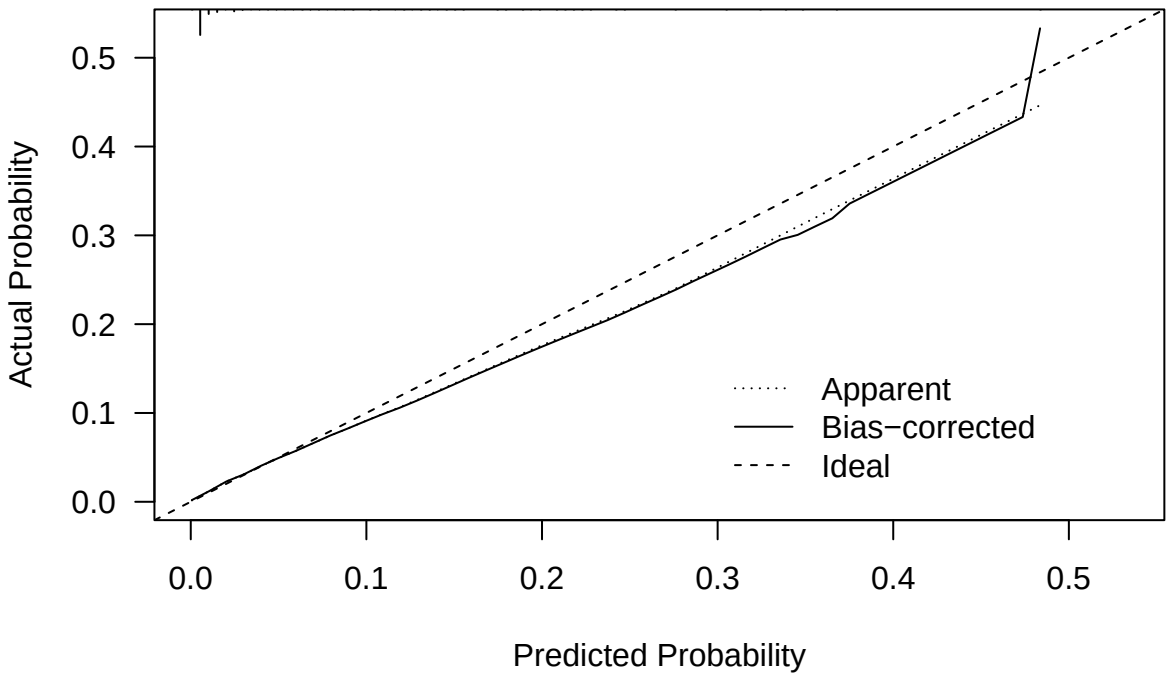
We notice that elastic net only chooses one variable, namely Speed limit. This was also chosen by AIC and BIC which shows that it is a highly significant predictor

Now do some model calibration:

```r
library(rms)
```

```
## Loading required package: Hmisc

##
## Attaching package: 'Hmisc'

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

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

## Registered S3 method overwritten by 'rms':
##   method       from
##   print.lrtest epiDisplay

##
## Attaching package: 'rms'

## The following object is masked from 'package:epiDisplay':
##
##     lrtest
```

```r
## Fit the final model with lrm from rms package using variables from BIC ##
lrm.final <- lrm(Severity ~ Speed_limit+Light_conditions+Age_Grp+TimeDay+DualCarriageway, data = encode

#Fit also model from AIC selection and Elastic Net
lrm.aic <- lrm(Severity ~ Speed_limit+Light_conditions+Age_Grp+TimeDay+DualCarriageway+Weekend, data =

lrm.en <- lrm(Severity ~ Speed_limit, data = encoded_data, x =TRUE, y = TRUE, model= T)




cross.calib <- calibrate(lrm.final, method="crossvalidation", B=8) # model calibration
plot(cross.calib, las=1, xlab = "Predicted Probability")
```




```
##
## n=384382   Mean absolute error=0.002   Mean squared error=0
## 0.9 Quantile of absolute error=0.002
```

Now plot the AUC curve for the model

```
library(pROC)
```

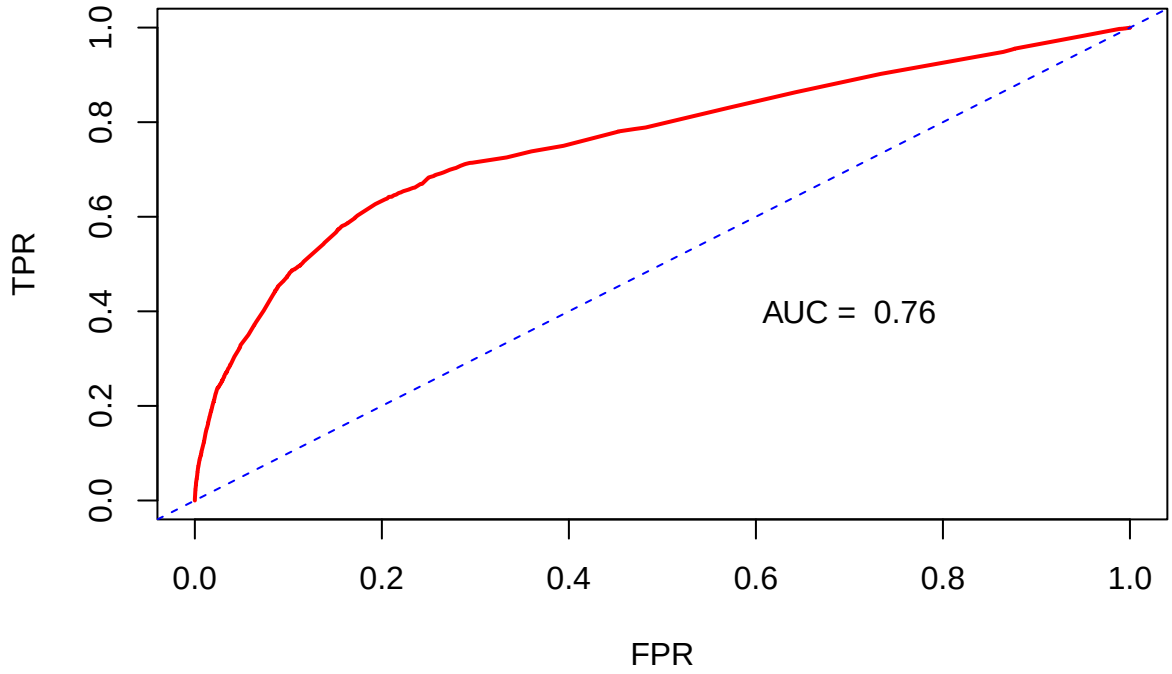
```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following object is masked from 'package:epiDisplay':
##
##     ci

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

```
p <- predict(lrm.final, type = "fitted")

roc_logit <- roc(selected_data$Severity ~ p)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## The True Positive Rate ##
TPR <- roc_logit$sensitivities
## The False Positive Rate ##
FPR <- 1 - roc_logit$specificities

plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col = 'red')
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit),2)))
```




```
auc(roc_logit)
```

```
## Area under the curve: 0.7581
```

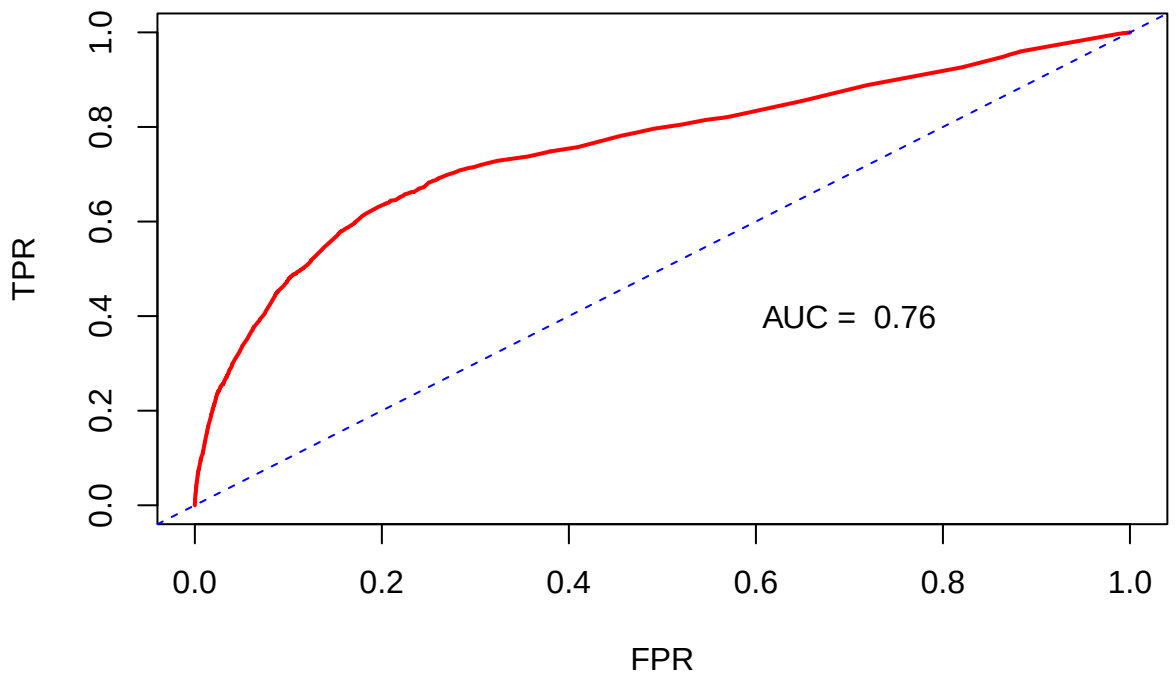
We obtained an AUC of 0.76 which suggests the model has a good discrimination ability

Look at AUC of modAIC model

```
p <- predict(lrm.aic, type = "fitted")

roc_logit <- roc(selected_data$Severity ~ p)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```
## The True Positive Rate ##
TPR <- roc_logit$sensitivities
## The False Positive Rate ##
FPR <- 1 - roc_logit$specificities

plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col = 'red')
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit),2)))
```



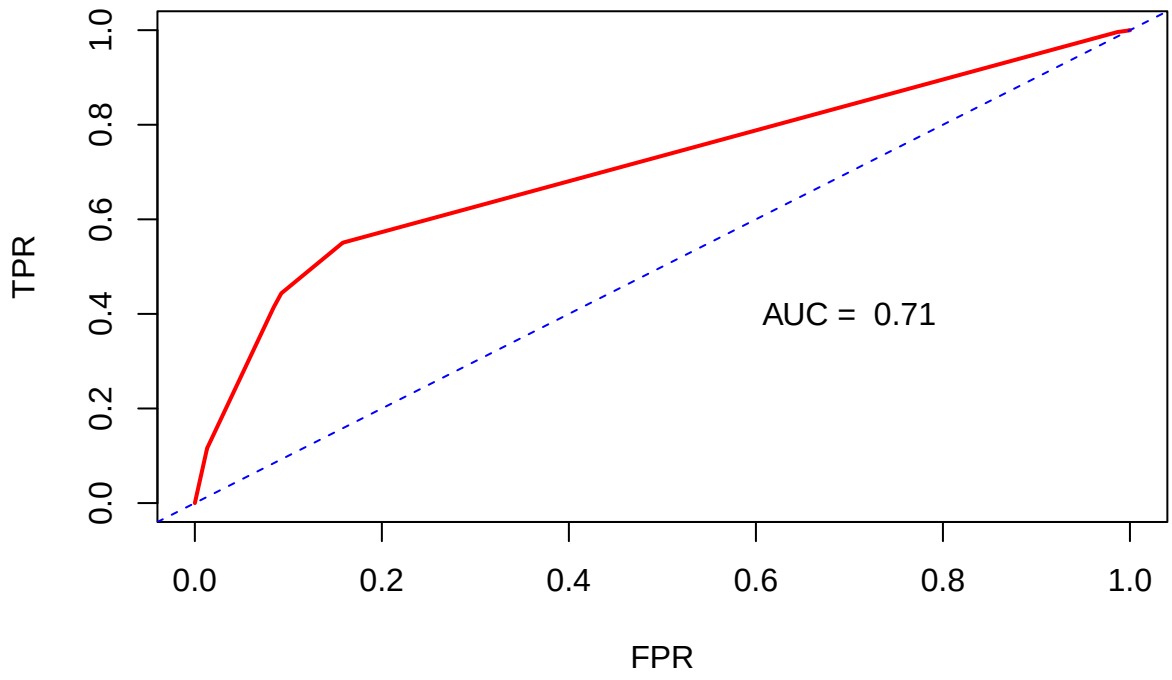

```
auc(roc_logit)
```

```
## Area under the curve: 0.7563
```

```
p <- predict(lrm.en, type = "fitted")

roc_logit <- roc(selected_data$Severity ~ p)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
## The True Positive Rate ##
TPR <- roc_logit$sensitivities
## The False Positive Rate ##
FPR <- 1 - roc_logit$specificities

plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col = 'red')
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit),2)))
```




```r
auc(roc_logit)
```

```
## Area under the curve: 0.7097
```

Also AIC has the highest AUC out of all three models, so we will choose it.

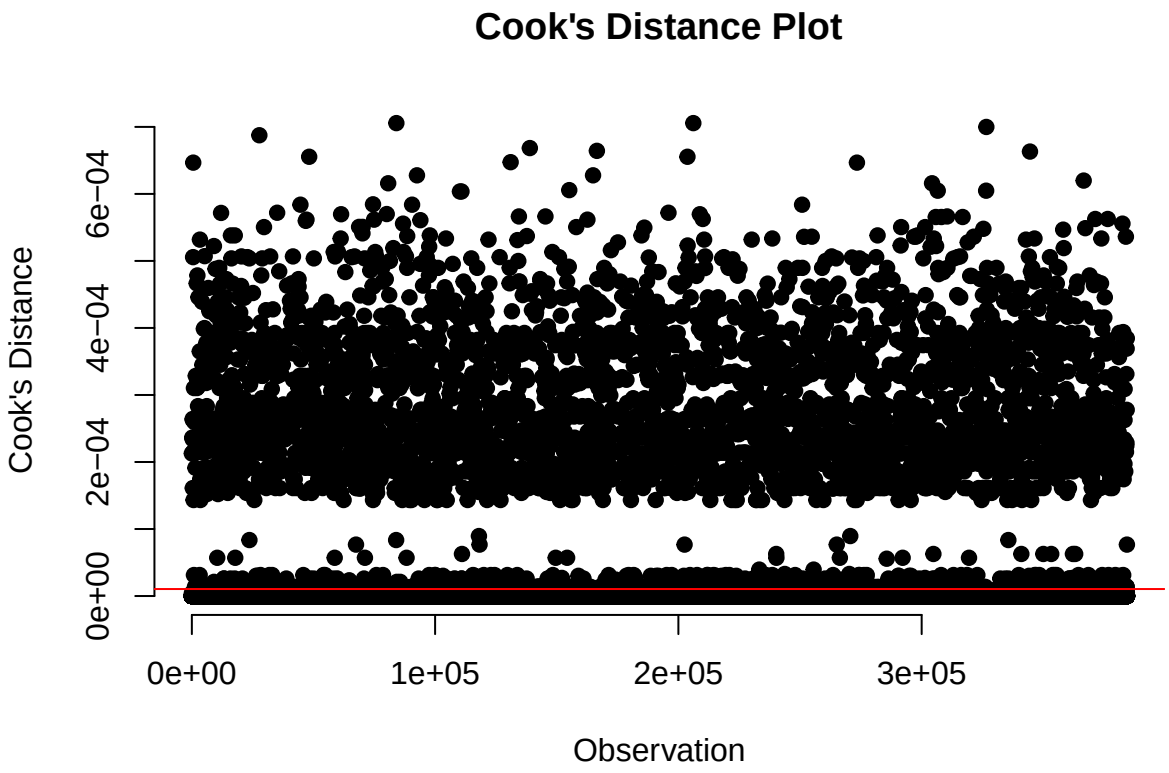
Now look for influential observations using Cook's Distance

```r
# Calculate Cook's distance
cooksd <- cooks.distance(modBIC)



# Find influential observations (those with Cook's distance > threshold)
threshold <- 4 / (nrow(encoded_data) - length(coef(lrm.final)) - 1)
influential_obs <- which(cooksd > threshold)
```

```r
print(length(influential_obs))
```

```
## [1] 3890
```

```r
# Plot Cook's distance
plot(cooksd, pch = 19, frame = FALSE, xlab = "Observation", ylab = "Cook's Distance", main = "Cook's Dis
abline(h = threshold, col = "red")  # Add threshold line
```

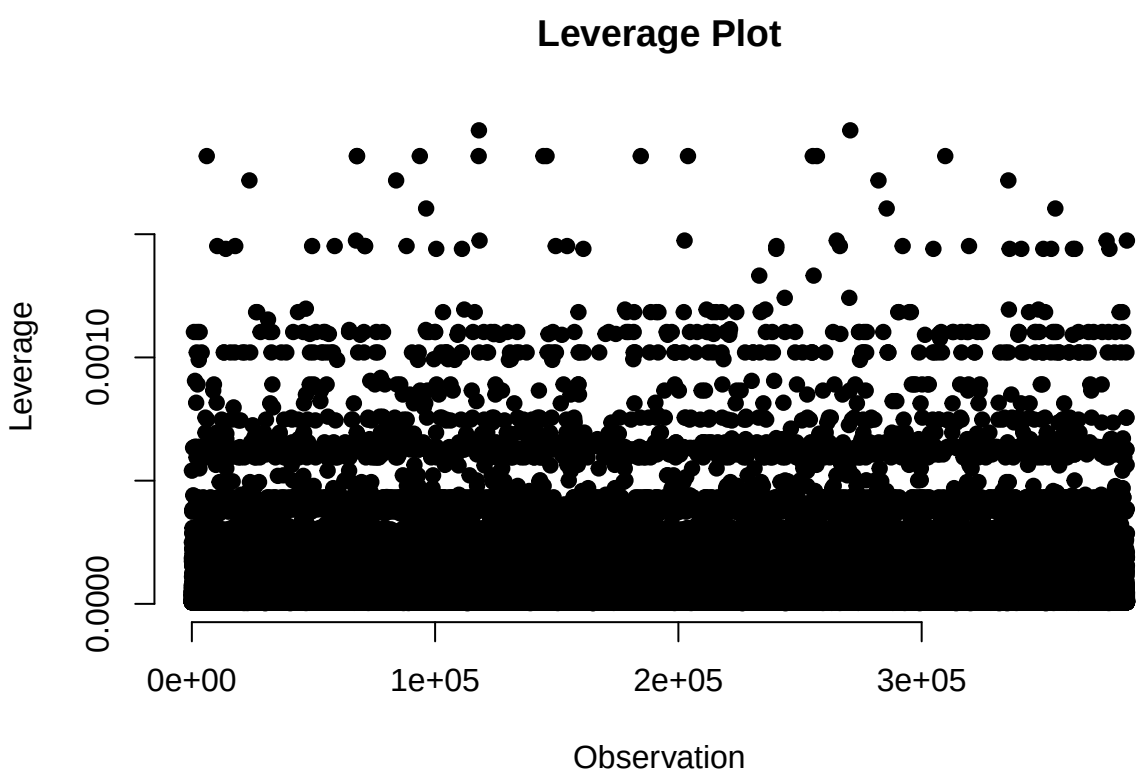


We see that almost 4000 observations were marked as influential, but we have almost 400000 observations so this is less than 1%, for the context of this investigation, we will keep them to ensure the results are more generalizable

```r
nrow(encoded_data)
```

```
## [1] 384382
```

```r
# Calculate leverage values
leverage <- hatvalues(modBIC)

# Plot leverage values
plot(leverage, pch = 19, frame = FALSE, xlab = "Observation", ylab = "Leverage", main = "Leverage Plot"
```


Leverage Plot
Leverage
0.0000
0.0010
0e+00
1e+05
2e+05
3e+05
Observation