## Module 3 Assignment 2 Logistic Regression

### Nicholas Barrett, 02/10/19

library(tidyverse)

## -- Attaching packages -------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ----------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(MASS) #access to forward and backward selection algorithms

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(caret) #for splitting functions

## Warning: package 'caret' was built under R version 3.5.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(ROCR) #for threshold selction

## Warning: package 'ROCR' was built under R version 3.5.2

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.5.2

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library(GGally)

## Warning: package 'GGally' was built under R version 3.5.2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':  
##   
## nasa

library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 3.5.2

Load data from the parole.csv file.

parole = read\_csv("parole.csv")

## Parsed with column specification:  
## cols(  
## male = col\_integer(),  
## race = col\_integer(),  
## age = col\_double(),  
## state = col\_integer(),  
## time.served = col\_double(),  
## max.sentence = col\_integer(),  
## multiple.offenses = col\_integer(),  
## crime = col\_integer(),  
## violator = col\_integer()  
## )

Structure and summary

str(parole)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 675 obs. of 9 variables:  
## $ male : int 1 0 1 1 1 1 1 0 0 1 ...  
## $ race : int 1 1 2 1 2 2 1 1 1 2 ...  
## $ age : num 33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...  
## $ state : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ time.served : num 5.5 5.4 5.6 5.7 5.4 6 6 4.8 4.5 4.7 ...  
## $ max.sentence : int 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ crime : int 4 3 3 1 1 4 3 1 3 2 ...  
## $ violator : int 0 0 0 0 0 0 0 0 0 0 ...  
## - attr(\*, "spec")=List of 2  
## ..$ cols :List of 9  
## .. ..$ male : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ race : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ age : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_double" "collector"  
## .. ..$ state : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ time.served : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_double" "collector"  
## .. ..$ max.sentence : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ multiple.offenses: list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ crime : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## .. ..$ violator : list()  
## .. .. ..- attr(\*, "class")= chr "collector\_integer" "collector"  
## ..$ default: list()  
## .. ..- attr(\*, "class")= chr "collector\_guess" "collector"  
## ..- attr(\*, "class")= chr "col\_spec"

summary(parole)

## male race age state   
## Min. :0.0000 Min. :1.000 Min. :18.40 Min. :1.000   
## 1st Qu.:1.0000 1st Qu.:1.000 1st Qu.:25.35 1st Qu.:2.000   
## Median :1.0000 Median :1.000 Median :33.70 Median :3.000   
## Mean :0.8074 Mean :1.424 Mean :34.51 Mean :2.887   
## 3rd Qu.:1.0000 3rd Qu.:2.000 3rd Qu.:42.55 3rd Qu.:4.000   
## Max. :1.0000 Max. :2.000 Max. :67.00 Max. :4.000   
## time.served max.sentence multiple.offenses crime   
## Min. :0.000 Min. : 1.00 Min. :0.0000 Min. :1.000   
## 1st Qu.:3.250 1st Qu.:12.00 1st Qu.:0.0000 1st Qu.:1.000   
## Median :4.400 Median :12.00 Median :1.0000 Median :2.000   
## Mean :4.198 Mean :13.06 Mean :0.5363 Mean :2.059   
## 3rd Qu.:5.200 3rd Qu.:15.00 3rd Qu.:1.0000 3rd Qu.:3.000   
## Max. :6.000 Max. :18.00 Max. :1.0000 Max. :4.000   
## violator   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.1156   
## 3rd Qu.:0.0000   
## Max. :1.0000

Factor conversion. Convert the response variable.

parole = parole %>% mutate(male = as.factor(male)) %>%   
 mutate(male = fct\_recode(male, "Female" = "0", "Male" = "1" ))

parole = parole %>%mutate(race = as.factor(race)) %>%  
 mutate(race = fct\_recode(race, "Other" = "2", "White" = "1" ))

parole = parole %>% mutate(state = as.factor(state))%>%  
 mutate(state = fct\_recode(state, "Kentucky"= "2", "Lousiana"= "3", "Virginia"= "4", "Other"= "1" ))

parole = parole %>% mutate(crime = as.factor(crime))%>%  
 mutate(crime = fct\_recode(crime, "Larceny"= "2", "Drug-Related"= "3", "Driving-Related"= "4", "Other"= "1" ))

parole = parole %>% mutate(multiple.offenses = as.factor(multiple.offenses))%>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "Incarcerated" = "1", "Other" = "0" ))

parole = parole %>% mutate(violator = as.factor(violator))%>%  
 mutate(violator = fct\_recode(violator, "Violated" = "1", "No Violation" = "0" ))

str(parole)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 675 obs. of 9 variables:  
## $ male : Factor w/ 2 levels "Female","Male": 2 1 2 2 2 2 2 1 1 2 ...  
## $ race : Factor w/ 2 levels "White","Other": 1 1 2 1 2 2 1 1 1 2 ...  
## $ age : num 33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...  
## $ state : Factor w/ 4 levels "Other","Kentucky",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ time.served : num 5.5 5.4 5.6 5.7 5.4 6 6 4.8 4.5 4.7 ...  
## $ max.sentence : int 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: Factor w/ 2 levels "Other","Incarcerated": 1 1 1 1 1 1 1 1 1 1 ...  
## $ crime : Factor w/ 4 levels "Other","Larceny",..: 4 3 3 1 1 4 3 1 3 2 ...  
## $ violator : Factor w/ 2 levels "No Violation",..: 1 1 1 1 1 1 1 1 1 1 ...

### Task 1 - Split the data into training and testing sets with training set having 70% of data. use random number of 12345

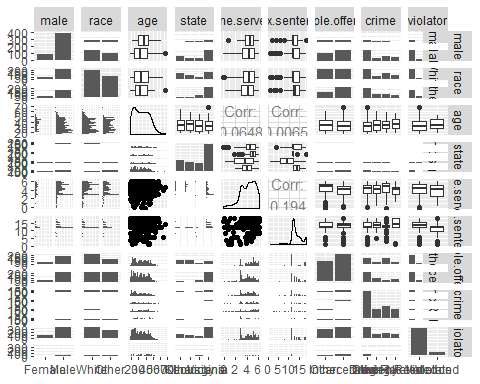
set.seed(12345)  
  
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE)   
train = parole[train.rows,]   
test = parole[-train.rows,]

### Task 2 - Identify the variables in the training set that are most predictive of the response variable “violator” using data visualizations and/or tables. Explain the thought process.

I initially used ggpairs correlation visualization but realized that the variables with several levels created too many variables to get a good feel for the variable with the best correlation to “violators”. I also tried another ggpairs plot with fewer variables to try to determine if it would be helpful. Ultimately, I decided to use bar plots and tables (like the Titantic expample) comparing the eight variables individually to violators.

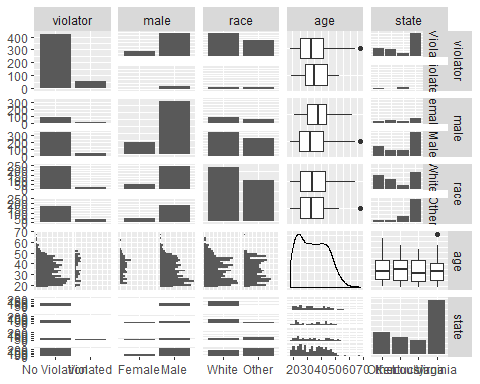
ggpairs(train)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



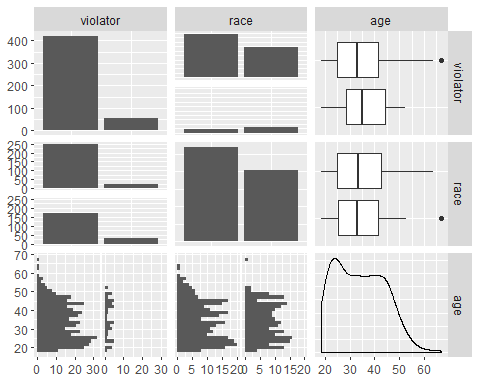
ggpairs(train, columns = c("violator", "male", "race", "age", "state"))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



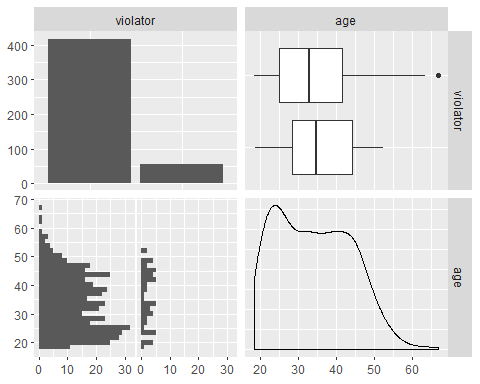
ggpairs(train, columns = c("violator", "race", "age"))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

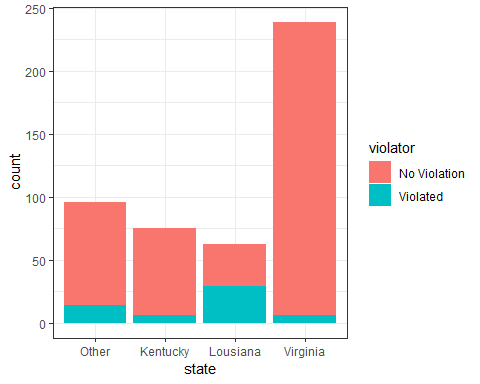


ggpairs(train, columns = c("violator", "age"))

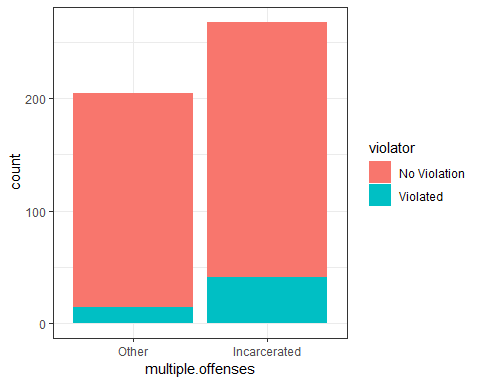
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



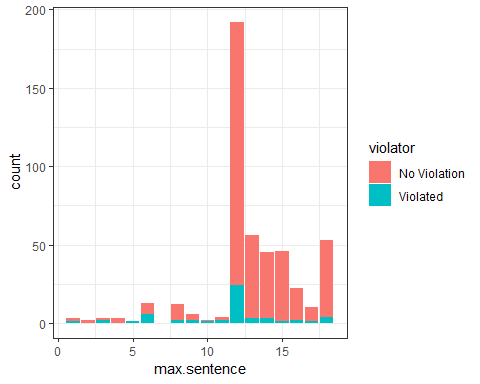
ggplot(train, aes(x=state, fill = violator)) + geom\_bar() + theme\_bw()



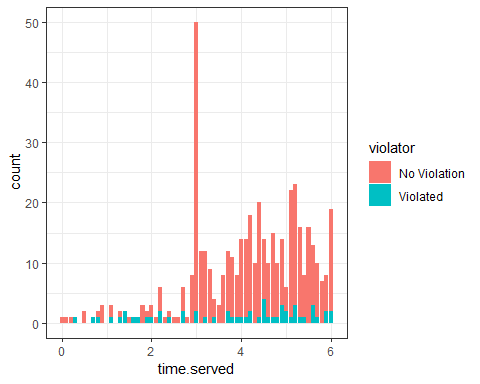
ggplot(train, aes(x=multiple.offenses, fill = violator)) + geom\_bar() + theme\_bw()



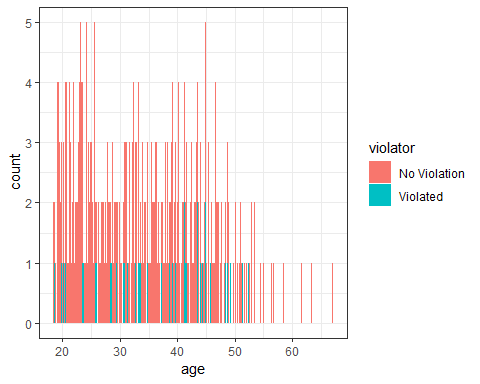
ggplot(train, aes(x=max.sentence, fill = violator)) + geom\_bar() + theme\_bw()



ggplot(train, aes(x=time.served, fill = violator)) + geom\_bar() + theme\_bw()



ggplot(train, aes(x=age, fill = violator)) + geom\_bar() + theme\_bw()



t1 = table(parole$violator, parole$race)  
prop.table(t1, margin = 2)

##   
## White Other  
## No Violation 0.90488432 0.85664336  
## Violated 0.09511568 0.14335664

t2 = table(parole$violator, parole$state)  
prop.table(t2, margin = 2)

##   
## Other Kentucky Lousiana Virginia  
## No Violation 0.86013986 0.88333333 0.54878049 0.97878788  
## Violated 0.13986014 0.11666667 0.45121951 0.02121212

t3 = table(parole$violator, parole$male)  
prop.table(t3, margin = 2)

##   
## Female Male  
## No Violation 0.8923077 0.8825688  
## Violated 0.1076923 0.1174312

t4 = table(parole$violator, parole$multiple.offenses)  
prop.table(t4, margin = 2)

##   
## Other Incarcerated  
## No Violation 0.9201278 0.8535912  
## Violated 0.0798722 0.1464088

t5 = table(parole$violator, parole$crime)  
prop.table(t5, margin = 2)

##   
## Other Larceny Drug-Related Driving-Related  
## No Violation 0.87619048 0.87735849 0.87581699 0.93069307  
## Violated 0.12380952 0.12264151 0.12418301 0.06930693

t6 = table(parole$violator, parole$max.sentence)  
prop.table(t6, margin = 2)

##   
## 1 2 3 4 5  
## No Violation 0.75000000 1.00000000 0.33333333 1.00000000 0.00000000  
## Violated 0.25000000 0.00000000 0.66666667 0.00000000 1.00000000  
##   
## 6 8 9 10 11  
## No Violation 0.56250000 0.85000000 0.66666667 0.66666667 0.60000000  
## Violated 0.43750000 0.15000000 0.33333333 0.33333333 0.40000000  
##   
## 12 13 14 15 16  
## No Violation 0.86496350 0.96103896 0.95454545 0.98333333 0.88888889  
## Violated 0.13503650 0.03896104 0.04545455 0.01666667 0.11111111  
##   
## 17 18  
## No Violation 0.93333333 0.89743590  
## Violated 0.06666667 0.10256410

### Task 3 - Identify the variable that is most predictive based on observation from Task 2.

The variable that is most predicative to violator is the parolees state with Lousiana’s 45% violator percentage of those incarcerated violating parole. The state/violator model showed significance in all states except Kentucky and highlighted the likilook of a violation clearly as the highest in Lousiana. The AIC value was also very low in comparison to the multiple.offenses/violator model (272.58 to 335.5, respectfully) I considered mulitple offenses at the most predicative with 14.64% of those incarcerated violating but upon comparing the two in linear models and AIC, it appears state is the stronger predictor.

mod1 = glm(violator ~ state, train, family = "binomial")  
summary(mod1)

##   
## Call:  
## glm(formula = violator ~ state, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.1106 -0.4084 -0.2255 -0.2255 2.7147   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.7677 0.2892 -6.113 9.79e-10 \*\*\*  
## stateKentucky -0.6747 0.5146 -1.311 0.189803   
## stateLousiana 1.6086 0.3841 4.188 2.81e-05 \*\*\*  
## stateVirginia -1.8916 0.5046 -3.749 0.000178 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 264.58 on 469 degrees of freedom  
## AIC: 272.58  
##   
## Number of Fisher Scoring iterations: 6

mod2 = glm(violator ~ multiple.offenses, train, family = "binomial")  
summary(mod2)

##   
## Call:  
## glm(formula = violator ~ multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5763 -0.5763 -0.3761 -0.3761 2.3169   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.6132 0.2769 -9.438 < 2e-16 \*\*\*  
## multiple.offensesIncarcerated 0.9018 0.3247 2.777 0.00549 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 331.50 on 471 degrees of freedom  
## AIC: 335.5  
##   
## Number of Fisher Scoring iterations: 5

### Task 4 - Use forward or backward stepwise to determine the best model for predicting violators using AIC to evaluate goodness.

I used backward and forward stepwise to find the best model. The backward stepwise model is a good model and best predicts violators with the combination of age, race, state, max.entence and multiple offenses. This combination produces a AIC of 252.28. This result is somewhat intuitive when considering state and multiple sentences. The plots I used to compare variables led me to believe state and multiple sentences were good predictores. What wasnt intuitive is the insignificance of the P values of Louisiana. Louisana parolees appeared to have the highest correlation to violations but in this model the P value was insignificant. Model significance: race, Virginia, Max.Sentence and multiple offences. Model insignificance: age, kentucky, Lousiana and max. sentence.

The cooefficients make sense.

In my forward stepwise model the AIC was the same as backward model at 252.28

allmod = glm(violator ~ male + age + race + state + time.served + crime + max.sentence + multiple.offenses, train, family = "binomial")   
summary(allmod)

##   
## Call:  
## glm(formula = violator ~ male + age + race + state + time.served +   
## crime + max.sentence + multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9635 -0.3638 -0.2354 -0.1449 2.9869   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.98802 1.36340 -3.659 0.000254 \*\*\*  
## maleMale 0.53377 0.49107 1.087 0.277051   
## age 0.03361 0.01696 1.982 0.047493 \*   
## raceOther 1.06698 0.41324 2.582 0.009824 \*\*   
## stateKentucky -0.30132 0.56939 -0.529 0.596665   
## stateLousiana 0.87804 0.52428 1.675 0.093984 .   
## stateVirginia -3.46523 0.63742 -5.436 5.44e-08 \*\*\*  
## time.served -0.03009 0.12159 -0.247 0.804537   
## crimeLarceny 0.18508 0.50343 0.368 0.713139   
## crimeDrug-Related -0.76563 0.46946 -1.631 0.102918   
## crimeDriving-Related -0.87795 0.62271 -1.410 0.158571   
## max.sentence 0.08458 0.05644 1.499 0.133963   
## multiple.offensesIncarcerated 1.72841 0.41857 4.129 3.64e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 230.16 on 460 degrees of freedom  
## AIC: 256.16  
##   
## Number of Fisher Scoring iterations: 6

emptymod = glm(violator ~1, train, family = "binomial")   
summary(emptymod)

##   
## Call:  
## glm(formula = violator ~ 1, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.4972 -0.4972 -0.4972 -0.4972 2.0745   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.0281 0.1434 -14.14 <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: 340.04 on 472 degrees of freedom  
## Residual deviance: 340.04 on 472 degrees of freedom  
## AIC: 342.04  
##   
## Number of Fisher Scoring iterations: 4

backmod = stepAIC(allmod, direction = "backward", trace = TRUE)

## Start: AIC=256.16  
## violator ~ male + age + race + state + time.served + crime +   
## max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - time.served 1 230.22 254.22  
## - crime 3 235.30 255.30  
## - male 1 231.41 255.41  
## <none> 230.16 256.16  
## - max.sentence 1 232.46 256.46  
## - age 1 234.09 258.09  
## - race 1 236.97 260.97  
## - multiple.offenses 1 248.67 272.67  
## - state 3 304.40 324.40  
##   
## Step: AIC=254.22  
## violator ~ male + age + race + state + crime + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - crime 3 235.38 253.38  
## - male 1 231.56 253.56  
## <none> 230.22 254.22  
## - max.sentence 1 232.50 254.50  
## - age 1 234.09 256.09  
## - race 1 236.97 258.98  
## - multiple.offenses 1 249.39 271.39  
## - state 3 304.94 322.95  
##   
## Step: AIC=253.38  
## violator ~ male + age + race + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 236.28 252.28  
## <none> 235.38 253.38  
## - max.sentence 1 237.41 253.41  
## - age 1 238.26 254.26  
## - race 1 242.32 258.32  
## - multiple.offenses 1 255.31 271.31  
## - state 3 309.30 321.30  
##   
## Step: AIC=252.28  
## violator ~ age + race + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 236.28 252.28  
## - max.sentence 1 238.31 252.31  
## - age 1 238.81 252.81  
## - race 1 243.44 257.44  
## - multiple.offenses 1 256.39 270.39  
## - state 3 309.81 319.80

summary(backmod)

##   
## Call:  
## glm(formula = violator ~ age + race + state + max.sentence +   
## multiple.offenses, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7414 -0.3643 -0.2668 -0.1502 2.7714   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.60426 1.13428 -4.059 4.92e-05 \*\*\*  
## age 0.02636 0.01660 1.588 0.11224   
## raceOther 1.07386 0.40527 2.650 0.00806 \*\*   
## stateKentucky -0.41360 0.54930 -0.753 0.45147   
## stateLousiana 0.86000 0.51900 1.657 0.09751 .   
## stateVirginia -3.34208 0.62057 -5.386 7.22e-08 \*\*\*  
## max.sentence 0.07733 0.05475 1.412 0.15788   
## multiple.offensesIncarcerated 1.77974 0.41476 4.291 1.78e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 236.28 on 465 degrees of freedom  
## AIC: 252.28  
##   
## Number of Fisher Scoring iterations: 6

forwardmod = stepAIC(emptymod, direction = "forward", scope=list(upper=allmod,lower=emptymod),  
 trace = TRUE)

## Start: AIC=342.04  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 264.58 272.58  
## + max.sentence 1 321.79 325.79  
## + multiple.offenses 1 331.50 335.50  
## + race 1 335.64 339.64  
## + time.served 1 336.02 340.02  
## <none> 340.04 342.04  
## + age 1 338.27 342.27  
## + crime 3 334.34 342.34  
## + male 1 339.78 343.78  
##   
## Step: AIC=272.58  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 246.88 256.88  
## + race 1 259.14 269.14  
## + age 1 262.48 272.48  
## <none> 264.58 272.58  
## + crime 3 259.43 273.43  
## + male 1 263.58 273.58  
## + time.served 1 264.29 274.29  
## + max.sentence 1 264.49 274.49  
##   
## Step: AIC=256.88  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 240.42 252.42  
## <none> 246.88 256.88  
## + age 1 245.01 257.01  
## + max.sentence 1 245.58 257.58  
## + male 1 246.13 258.13  
## + time.served 1 246.88 258.88  
## + crime 3 242.93 258.93  
##   
## Step: AIC=252.42  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## + age 1 238.31 252.31  
## <none> 240.42 252.42  
## + max.sentence 1 238.81 252.81  
## + male 1 239.85 253.85  
## + time.served 1 240.37 254.37  
## + crime 3 236.69 254.69  
##   
## Step: AIC=252.31  
## violator ~ state + multiple.offenses + race + age  
##   
## Df Deviance AIC  
## + max.sentence 1 236.28 252.28  
## <none> 238.31 252.31  
## + male 1 237.41 253.41  
## + crime 3 233.88 253.88  
## + time.served 1 238.18 254.18  
##   
## Step: AIC=252.28  
## violator ~ state + multiple.offenses + race + age + max.sentence  
##   
## Df Deviance AIC  
## <none> 236.28 252.28  
## + male 1 235.38 253.38  
## + crime 3 231.56 253.56  
## + time.served 1 236.12 254.12

summary(forwardmod)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race + age +   
## max.sentence, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7414 -0.3643 -0.2668 -0.1502 2.7714   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.60426 1.13428 -4.059 4.92e-05 \*\*\*  
## stateKentucky -0.41360 0.54930 -0.753 0.45147   
## stateLousiana 0.86000 0.51900 1.657 0.09751 .   
## stateVirginia -3.34208 0.62057 -5.386 7.22e-08 \*\*\*  
## multiple.offensesIncarcerated 1.77974 0.41476 4.291 1.78e-05 \*\*\*  
## raceOther 1.07386 0.40527 2.650 0.00806 \*\*   
## age 0.02636 0.01660 1.588 0.11224   
## max.sentence 0.07733 0.05475 1.412 0.15788   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 236.28 on 465 degrees of freedom  
## AIC: 252.28  
##   
## Number of Fisher Scoring iterations: 6

### Task 5 - Create a logistic regression model using the training set to predict “violators” using state, mulitple.offenses and race.

mod3 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial")  
summary(mod3)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4553 -0.3862 -0.2931 -0.1787 2.8791   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.5582 0.3709 -6.898 5.28e-12 \*\*\*  
## stateKentucky -0.4816 0.5417 -0.889 0.3740   
## stateLousiana 0.5292 0.4769 1.110 0.2672   
## stateVirginia -3.2301 0.6028 -5.358 8.39e-08 \*\*\*  
## multiple.offensesIncarcerated 1.6596 0.3985 4.165 3.12e-05 \*\*\*  
## raceOther 1.0024 0.3966 2.528 0.0115 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 340.04 on 472 degrees of freedom  
## Residual deviance: 240.42 on 467 degrees of freedom  
## AIC: 252.42  
##   
## Number of Fisher Scoring iterations: 6

This model looks good with an AIC of 252.42 which is very similar to the best model. Considering the coefficients, Kentucky and Lousiana were insignificant; whereas, race, all other states, Virginia and multiple.offenses were significant.

### Task 6 - Predict the probability of a parole violation of two parolees. Parolee1: Lousiana with multiple offenses and white race. Parolee2: Kentucky with no mulitple offenses and other race.

The predicted probability of Parolee 1 to violate is 40.86% that they will violate parole.

The predicated probability of Parolee 2 to violate is 11.53% that they will violate parole.

mod4 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial")

Parolee1Pred <- data.frame(state = "Lousiana", multiple.offenses = "Incarcerated", race = "White")

predict(mod4, Parolee1Pred, type = "response")

## 1   
## 0.408682

Parolee2Pred <- data.frame(state = "Kentucky", multiple.offenses = "Other", race = "Other")  
predict(mod4, Parolee2Pred, type = "response")

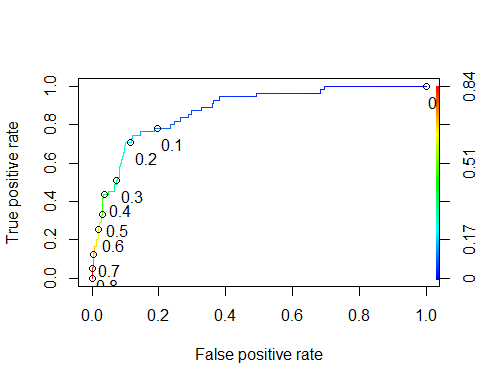
## 1   
## 0.1153326

### Task 7 - Develop a ROC curve and determien the probably threshold that best balances specificity and sensitivity on the training set. Use type = “response” and do not use .2.

predictions <- predict(backmod, train, type = "response")  
head(predict)

##   
## 1 function (object, ...)   
## 2 UseMethod("predict")

ROCRpred = prediction(predictions, train$violator)   
  
ROCRperf = performance(ROCRpred, "tpr", "fpr")  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



### Task 8 - What is the accuracy, sensitivity and specificity of the model on the training set given the cutoff from Task 7? What are the implications of incorrectly classifying a parolee?

The accuracy of the training set 0.8414376

The sensitivity of the training set is 0.7636364

The specificity of the training set is 0.8540670

The implications of incorrectly classifying a parolee to balance accuracy against sensitivy and specificity is assuming that some parolees will be violators when they will not be and conversely assuming some parolees will not be violators when they will be violators of parole.

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.8755763

opt.cut = function(perf, pred){  
 cut.ind = mapply(FUN=function(x, y, p){  
 d = (x - 0)^2 + (y-1)^2  
 ind = which(d == min(d))  
 c(sensitivity = y[[ind]], specificity = 1-x[[ind]],   
 cutoff = p[[ind]])  
 }, perf@x.values, perf@y.values, pred@cutoffs)  
}  
print(opt.cut(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.7636364  
## specificity 0.8540670  
## cutoff 0.1455707

t1 = table(train$violator,predictions > 0.1455707)  
t1

##   
## FALSE TRUE  
## No Violation 357 61  
## Violated 14 41

(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8414376

### Task 9 - Identify a probability threshold (via trial ad error) that best maximizes accuracy on the training set.

By using trial and error, the probability threshold that best maximizes accuracy is 0.68.

t1 = table(train$violator,predictions > .68)  
t1

##   
## FALSE TRUE  
## No Violation 415 3  
## Violated 46 9

(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8964059

### Task 10 - Use your probability threshold from Task 9 to determine accuracy of the model on the testing set.

The accuracy of the predicative model is 84.1% when balancing sensitivity and specificity. Through trial and error calculations in probability thresholds. The best accuracy generated through trial and error is 89.64%