# Logistic Regression (Classification)

## May 28th, 2020

### Stephen Kiser

Libraries

library(tidyverse)

## -- Attaching packages ------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

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

library(MASS)

##   
## Attaching package: 'MASS'

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

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(ROCR)

Reading in dataset

parole <- read\_csv("parole.csv")

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

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,   
 "white" = "1",  
 "Other" = "2"))  
  
parole <- parole %>%  
 mutate(state = as.factor(state)) %>% mutate(state = fct\_recode(state,  
 "Other" = "1",   
 "Kentucky" = "2",  
 "Louisiana" = "3",  
 "Virginia" = "4"  
 ))  
  
parole <- parole %>%  
 mutate(crime = as.factor(crime)) %>% mutate(crime = fct\_recode(crime,   
 "Other" = "1",  
 "Larceny" = "2",  
 "Drug-related" = "3",  
 "Driving-related" = "4"))  
  
parole <- parole %>%  
 mutate(multiple.offenses = as.factor(multiple.offenses)) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses,  
 "No" = "0",  
 "Yes" = "1"))  
  
parole <- parole %>%  
 mutate(violator = as.factor(violator)) %>%  
 mutate(violator = fct\_recode(violator,  
 "Didnt Violate Parole" = "0",  
 "Violated Parole" = "1"))  
str(parole)

## Classes 'spec\_tbl\_df', '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 : num 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: Factor w/ 2 levels "No","Yes": 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 "Didnt Violate Parole",..: 1 1 1 1 1 1 1 1 1 1 ...

glimpse(parole)

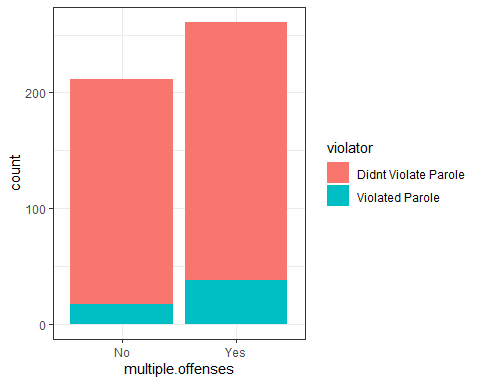
## Observations: 675  
## Variables: 9  
## $ male <fct> male, female, male, male, male, male, male, femal...  
## $ race <fct> white, white, Other, white, Other, Other, white, ...  
## $ age <dbl> 33.2, 39.7, 29.5, 22.4, 21.6, 46.7, 31.0, 24.6, 3...  
## $ state <fct> Other, Other, Other, Other, Other, Other, Other, ...  
## $ time.served <dbl> 5.5, 5.4, 5.6, 5.7, 5.4, 6.0, 6.0, 4.8, 4.5, 4.7,...  
## $ max.sentence <dbl> 18, 12, 12, 18, 12, 18, 18, 12, 13, 12, 12, 12, 1...  
## $ multiple.offenses <fct> No, No, No, No, No, No, No, No, No, No, Yes, No, ...  
## $ crime <fct> Driving-related, Drug-related, Drug-related, Othe...  
## $ violator <fct> Didnt Violate Parole, Didnt Violate Parole, Didnt...

### Task 1

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

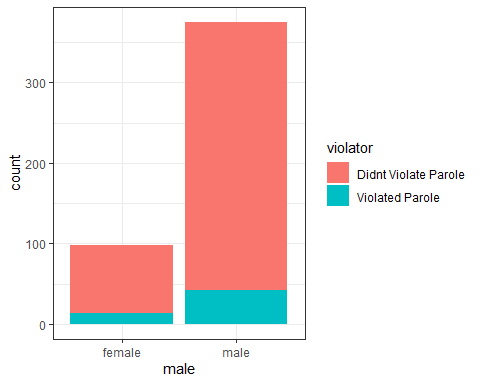
### Task 2

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



In this graph we see that those who had multiple offenses had a higher chance of violating their parole.

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

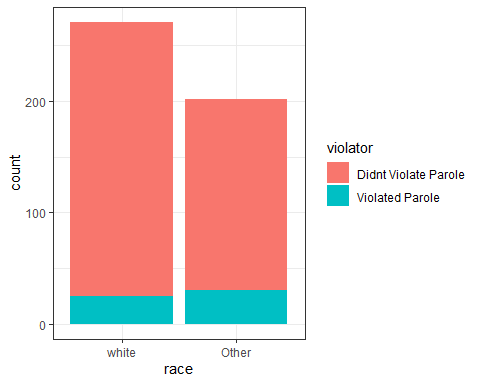


t1 <- table(train$violator, train$male)  
prop.table(t1, margin = 2)

##   
## female male  
## Didnt Violate Parole 0.8673469 0.8880000  
## Violated Parole 0.1326531 0.1120000

In this graph we see that males had a higher chance of violating parole, however there was a lot more male subjects then there were females. I made a table to see what the percentage of males and females who violated parole. As we can see females had about 13.27 % of violating their parole, and males had 11.20% chance of violating parole.

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

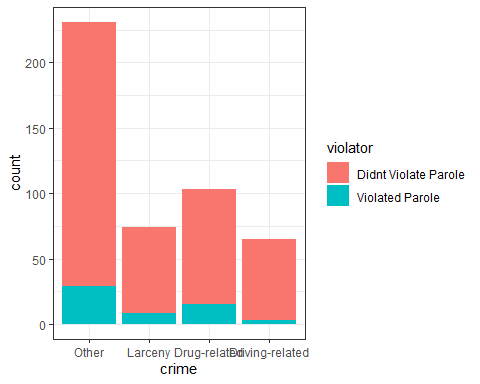


t2 <- table(train$violator, train$race)  
prop.table(t2, margin = 2)

##   
## white Other  
## Didnt Violate Parole 0.90774908 0.85148515  
## Violated Parole 0.09225092 0.14851485

In this graph we see how the race variable affected parole violation. Seeing from the graph there were more parolees who violated parole that were not white. We see this also in the table that other race had a 14.85% while white had 9.23%.

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

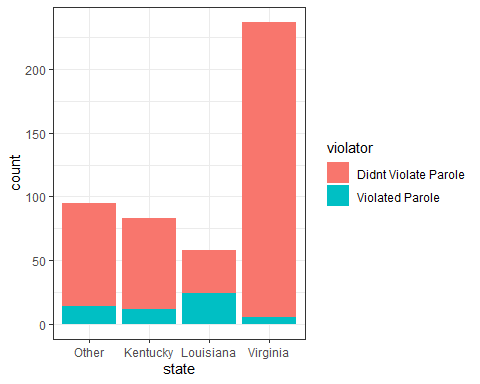


t3 <- table(train$violator, train$crime)  
prop.table(t3, margin = 2)

##   
## Other Larceny Drug-related Driving-related  
## Didnt Violate Parole 0.87445887 0.89189189 0.85436893 0.95384615  
## Violated Parole 0.12554113 0.10810811 0.14563107 0.04615385

From this graph we see the type of crime variable affects on parole violation. We see there was a higher chance of violation in the other bar, however this was almost double or triple amount of other crime compared to larceny, drug-related, and driving-related. I made a table to see the information better. We see that those who had drug-related crime had a higher chance of violating parole.

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

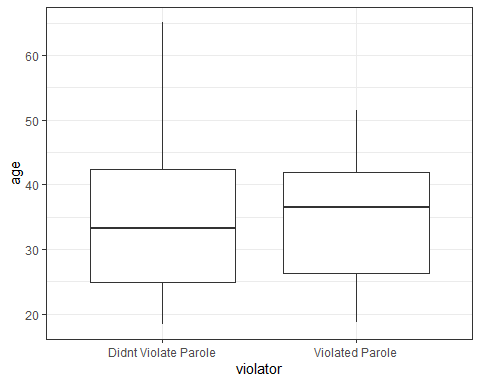


t4 <- table(train$violator, train$state)  
prop.table(t4, margin = 2)

##   
## Other Kentucky Louisiana Virginia  
## Didnt Violate Parole 0.85263158 0.85542169 0.58620690 0.97890295  
## Violated Parole 0.14736842 0.14457831 0.41379310 0.02109705

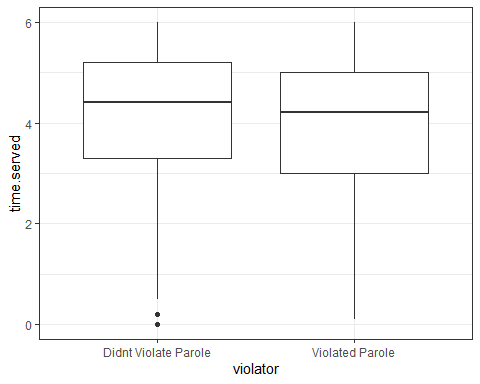
This graph compares which state had an effect to violating parole. We see that Virigina had the most parolees, and Louisiana almost had half of their parolees violate parole. Once again I made a graph and swa that Louisiana had 41.38% violate parole, and Virginia had the least with 2.11%.

ggplot(train, aes(x= violator, y= age)) +  
 geom\_boxplot()+  
 theme\_bw()



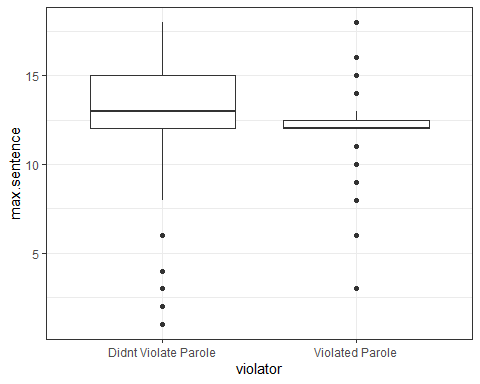
This graph sees how age variable affected parole violations. It seems that the average age of those who violated parole was older than those who did not violate parole. However, those who did not violate parole had a greater range.

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



This graph shows that the average time served was lower for those who violated parole while those who did not violate parole had a longer time served.

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



Finally, this is the last graph I did to see the the effects of variables on the violator variable. This graph shows almost the same trend that the previous graph did (violator variable vs. time served variable). Those who did not violate parole average max sentence was longer than those who violated parole. This graph does not show much information because both have a wide range.

### Task 3

mod1 = glm(violator ~ multiple.offenses, train, family = "binomial")  
summary(mod1)

##   
## Call:  
## glm(formula = violator ~ multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5610 -0.5610 -0.4089 -0.4089 2.2465   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4398 0.2529 -9.648 <2e-16 \*\*\*  
## multiple.offensesYes 0.6702 0.3078 2.177 0.0295 \*   
## ---  
## 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: 335.02 on 471 degrees of freedom  
## AIC: 339.02  
##   
## Number of Fisher Scoring iterations: 5

I chose mutlipe.offenses because it had the greater predictative ability of the violator variable. The AIC of this model is 339.02 which is seems small, but it seems a big large for a varialbe if it was the best predictor of violator variable. This means that there are more variables that affect the violator variable.

### Task 4

allmod = glm(violator ~., train, family = "binomial")  
summary(allmod)

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6055 -0.3932 -0.2643 -0.1384 2.9470   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.885777 1.197993 -2.409 0.01600 \*   
## malemale -0.137577 0.411340 -0.334 0.73803   
## raceOther 1.143719 0.403890 2.832 0.00463 \*\*   
## age 0.005279 0.016910 0.312 0.75490   
## stateKentucky 0.124282 0.492370 0.252 0.80072   
## stateLouisiana 0.217202 0.556154 0.391 0.69614   
## stateVirginia -3.801561 0.666733 -5.702 1.19e-08 \*\*\*  
## time.served -0.109344 0.118901 -0.920 0.35777   
## max.sentence 0.065956 0.054593 1.208 0.22700   
## multiple.offensesYes 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## crimeLarceny 0.392910 0.514075 0.764 0.44469   
## crimeDrug-related -0.210563 0.413351 -0.509 0.61047   
## crimeDriving-related -0.727043 0.690775 -1.053 0.29257   
## ---  
## 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: 242.09 on 460 degrees of freedom  
## AIC: 268.09  
##   
## 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

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

## Start: AIC=342.04  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 275.18 283.18  
## + max.sentence 1 331.01 335.01  
## + multiple.offenses 1 335.02 339.02  
## + race 1 336.51 340.51  
## + time.served 1 336.61 340.61  
## <none> 340.04 342.04  
## + crime 3 335.07 343.07  
## + male 1 339.72 343.72  
## + age 1 339.95 343.95  
##   
## Step: AIC=283.18  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 254.96 264.96  
## + race 1 267.66 277.66  
## <none> 275.18 283.18  
## + max.sentence 1 274.27 284.27  
## + time.served 1 274.44 284.44  
## + age 1 275.11 285.11  
## + male 1 275.13 285.13  
## + crime 3 271.72 285.72  
##   
## Step: AIC=264.96  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 246.98 258.98  
## <none> 254.96 264.96  
## + max.sentence 1 253.11 265.11  
## + time.served 1 254.47 266.47  
## + male 1 254.91 266.91  
## + age 1 254.94 266.94  
## + crime 3 252.75 268.75  
##   
## Step: AIC=258.98  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## + max.sentence 1 245.31 259.31  
## + time.served 1 246.33 260.33  
## + male 1 246.78 260.78  
## + age 1 246.98 260.98  
## + crime 3 244.78 262.79

summary(forwardmod)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesYes 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## raceOther 1.09382 0.38974 2.807 0.00501 \*\*   
## ---  
## 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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

Backward stepwise

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

## Start: AIC=268.09  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + crime  
##   
## Df Deviance AIC  
## - crime 3 244.47 264.47  
## - age 1 242.18 266.18  
## - male 1 242.20 266.20  
## - time.served 1 242.93 266.93  
## - max.sentence 1 243.57 267.57  
## <none> 242.09 268.09  
## - race 1 250.24 274.24  
## - multiple.offenses 1 261.96 285.96  
## - state 3 316.24 336.24  
##   
## Step: AIC=264.47  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - age 1 244.48 262.48  
## - male 1 244.85 262.85  
## - time.served 1 245.04 263.04  
## - max.sentence 1 246.00 264.00  
## <none> 244.47 264.47  
## - race 1 252.62 270.62  
## - multiple.offenses 1 265.46 283.46  
## - state 3 321.69 335.69  
##   
## Step: AIC=262.48  
## violator ~ male + race + state + time.served + max.sentence +   
## multiple.offenses  
##   
## Df Deviance AIC  
## - male 1 244.86 260.86  
## - time.served 1 245.04 261.04  
## - max.sentence 1 246.01 262.01  
## <none> 244.48 262.48  
## - race 1 252.65 268.65  
## - multiple.offenses 1 265.52 281.52  
## - state 3 322.14 334.14  
##   
## Step: AIC=260.86  
## violator ~ race + state + time.served + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - time.served 1 245.31 259.31  
## - max.sentence 1 246.33 260.33  
## <none> 244.86 260.86  
## - race 1 252.80 266.80  
## - multiple.offenses 1 265.93 279.93  
## - state 3 322.54 332.54  
##   
## Step: AIC=259.31  
## violator ~ race + state + max.sentence + multiple.offenses  
##   
## Df Deviance AIC  
## - max.sentence 1 246.98 258.98  
## <none> 245.31 259.31  
## - race 1 253.11 265.11  
## - multiple.offenses 1 266.89 278.89  
## - state 3 323.88 331.88  
##   
## Step: AIC=258.98  
## violator ~ race + state + multiple.offenses  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## - race 1 254.96 264.96  
## - multiple.offenses 1 267.66 277.66  
## - state 3 332.93 338.93

summary(backmod)

##   
## Call:  
## glm(formula = violator ~ race + state + multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## raceOther 1.09382 0.38974 2.807 0.00501 \*\*   
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesYes 1.73482 0.39421 4.401 1.08e-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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

After completing the forward stepwise and backward stepwise regressions we see that they give the same answers. Both stepwise regressions show that race, state, and multiple.offenses had an effect on the violator variable. The forward stepwise regression started with an AIC value of 342.04, and ended with an AIC value of 258.98. This is a difference of 80.04 from just multiple.offenses variable model. This means that having those three variables we have the smallest AIC value we can get with this data.

### Task 5

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

##   
## Call:  
## glm(formula = violator ~ race + state + multiple.offenses, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## raceOther 1.09382 0.38974 2.807 0.00501 \*\*   
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesYes 1.73482 0.39421 4.401 1.08e-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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

After making the logistic regression model with the race, state, and multiple.offenses variables we see that multiple.offenses(yes) and state(Virgina) were the significant variables. Also race(other) was significant, but not as significant as the multiple.offenses and state variables. The states Kentucky and Louisiana were not significant in this model.

### Task 6

parolee1 <- data.frame(state = "Louisiana", race = "white", multiple.offenses = "Yes")  
  
parolee2 <- data.frame(state = "Kentucky", race = "Other", multiple.offenses = "No")  
  
  
predicts <- predict(logmod, parolee1, type = "response")   
predicts2 <- predict(logmod, parolee2, type = "response")  
  
table(predicts)

## predicts  
## 0.337996071719129   
## 1

table(predicts2)

## predicts2  
## 0.206962905204274   
## 1

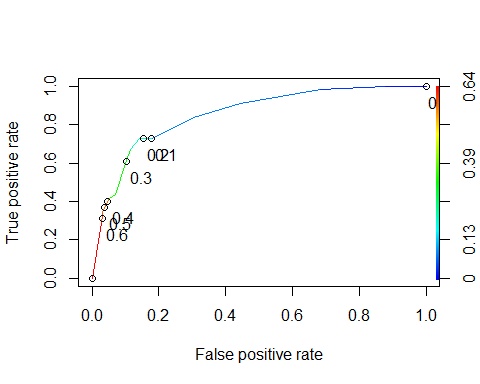
For the first parolee has about a 33.80% of violating parole, and the second parolee has about a 20.70% of violating parole.

### Task 7

pred <- predict(logmod, train, type = "response")  
head(pred)

## 1 2 3 4 5 6   
## 0.07509978 0.19512504 0.19512504 0.07509978 0.07509978 0.19512504

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



### Task 8

Area under the curve/Specificity and Sensitivity

#Area under curve  
as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.8524576

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(ROCRpref, ROCRpred))

## [,1]  
## sensitivity 0.7272727  
## specificity 0.8588517  
## cutoff 0.2069629

Accuracy

t5 = table(train$violator, pred > 0.2069629)  
t5

##   
## FALSE TRUE  
## Didnt Violate Parole 359 59  
## Violated Parole 15 40

#caluculating accuracy  
(t5[1,1]+t5[2,2])/nrow(train)

## [1] 0.8435518

The accuracy of this ROC curve is about 84.36%. The sensitivity is 0.7273 and the specificity is 0.8589. As we we can see in the table we incorrectly classified 74 parolees with this threshold of 0.2069629.

### Task 9 and Task 10

#Applying trial and error to maximize accuracy   
  
#(trying 0.3 as threshold)  
t6 = table(train$violator, pred > 0.3)  
t6

##   
## FALSE TRUE  
## Didnt Violate Parole 376 42  
## Violated Parole 22 33

(t6[1,1]+t6[2,2])/nrow(train)

## [1] 0.8646934

#(trying 0.2 as threshold)  
t7 = table(train$violator, pred >0.4)  
t7

##   
## FALSE TRUE  
## Didnt Violate Parole 405 13  
## Violated Parole 36 19

(t7[1,1]+t7[2,2])/nrow(train)

## [1] 0.8964059

#(trying 0.4 as threshold)  
t8 = table(train$violator, pred > 0.5)  
t8

##   
## FALSE TRUE  
## Didnt Violate Parole 405 13  
## Violated Parole 36 19

(t8[1,1]+t8[2,2])/nrow(train)

## [1] 0.8964059

#(trying 0.6)  
t9 = table(train$violator, pred > 0.6)  
t9

##   
## FALSE TRUE  
## Didnt Violate Parole 406 12  
## Violated Parole 39 16

(t9[1,1]+t9[2,2])/nrow(train)

## [1] 0.8921776

The threshold that best maximizes the accuracy of the training set it 0.5 because it gives us an accuracry of 0.8964. The threshold of 0.4 also gives us that value of accuracy, but it is not the maximum threshold we can have. As you can see though if we increase our threshold to 0.6 the accuracry of the training set starts to decrease. With an accuracy of 89.64% we can safely predict those who will violate parolee and those who do not with our model with a low chance of predicting wrong.