Mod3Assign3Parole

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.2

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

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- 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(ROCR) #best subset selection

## Warning: package 'ROCR' was built under R version 3.6.2

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.6.2

##   
## Attaching package: 'gplots'

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

library(caret)

## Warning: package 'caret' was built under R version 3.6.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

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, "Other" = "2", "White" = "1" ))  
parole = parole %>% mutate(state = as.factor(state)) %>%   
 mutate(state = fct\_recode(state, "Other" = "1", "Kentucky" = "2", "Louisianna" = "3", "Virginia" = "4" ))  
parole = parole %>% mutate(crime = as.factor(crime)) %>%   
 mutate(crime = fct\_recode(crime, "Other" = "1", "Larceny" = "2", "Drugs" = "3", "Driving" = "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, "No" = "0", "Yes" = "1" ))  
  
str(parole)

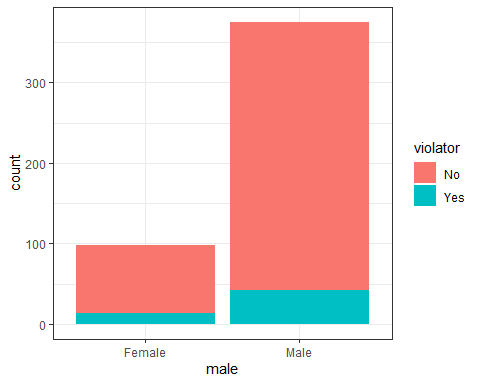
## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 675 obs. of 9 variables:  
## $ male : Factor w/ 2 levels "Female","Male": 2 1 2 2 2 2 2 1 1 2 ...  
## $ race : Factor w/ 2 levels "White","Other": 1 1 2 1 2 2 1 1 1 2 ...  
## $ age : num 33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...  
## $ 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 "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...

**Task 1**

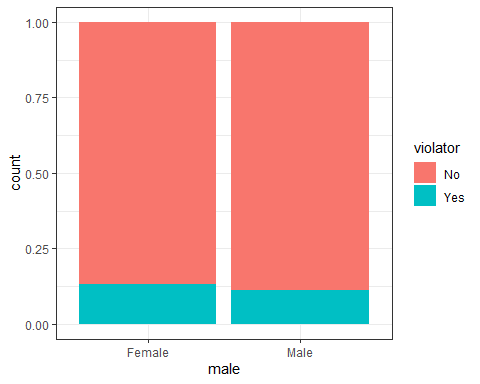
set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE)  
train = parole[train.rows,]   
test = parole[-train.rows,]

**Task 2** **Male**

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



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

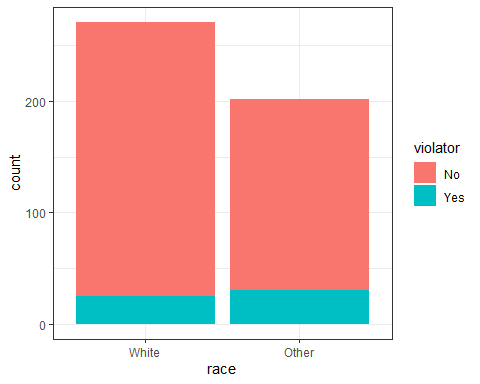


t1 = table(train$violator, train$male) #create a table object  
prop.table(t1, margin = 2 )

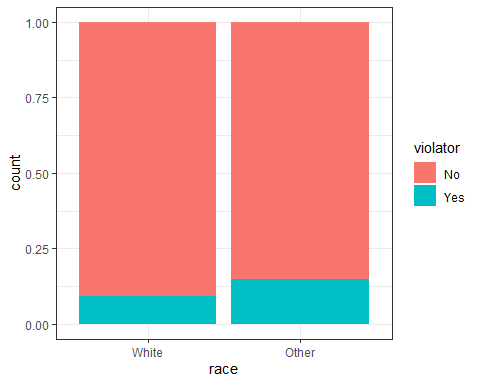
##   
## Female Male  
## No 0.8673469 0.8880000  
## Yes 0.1326531 0.1120000

**Race**

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



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



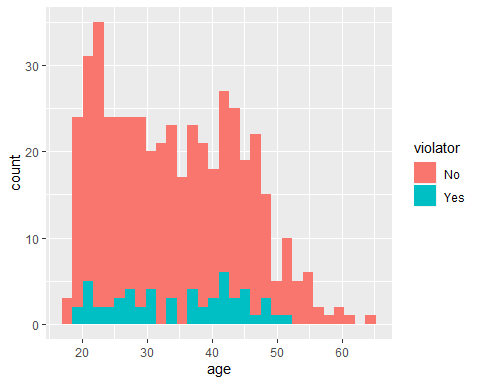
t2 = table(train$violator, train$race) #create a table object  
prop.table(t2, margin = 2 )

##   
## White Other  
## No 0.90774908 0.85148515  
## Yes 0.09225092 0.14851485

**Age**

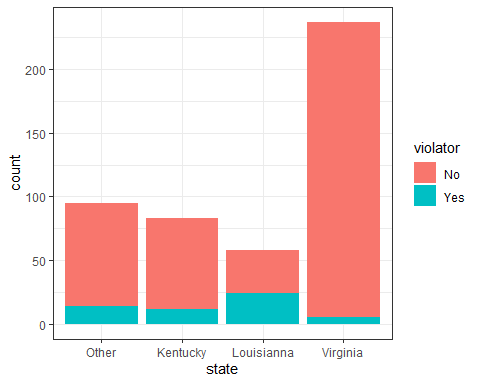
ggplot(train, aes(x=age)) + geom\_histogram(aes(fill = violator))

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

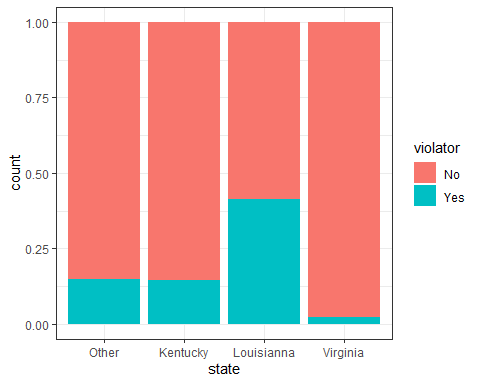


**State**

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



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



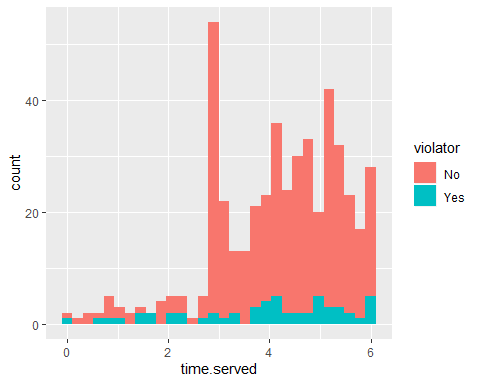
t3 = table(train$violator, train$state) #create a table object  
prop.table(t3, margin = 2 )

##   
## Other Kentucky Louisianna Virginia  
## No 0.85263158 0.85542169 0.58620690 0.97890295  
## Yes 0.14736842 0.14457831 0.41379310 0.02109705

**Time Served**

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

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



**Max Sentence**

ggplot(train, aes(x=max.sentence)) + geom\_histogram(breaks=seq(1,18, by=1), aes(fill = violator))

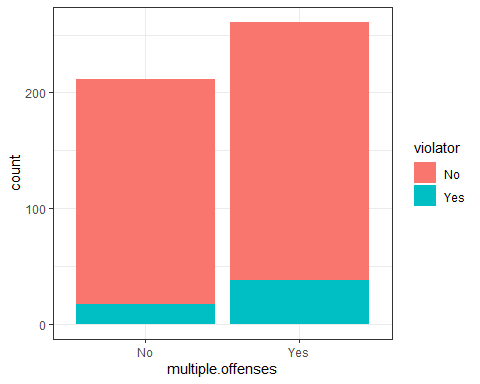


t4 = table(train$violator, train$max.sentence) #create a table object  
prop.table(t4, margin = 2 )

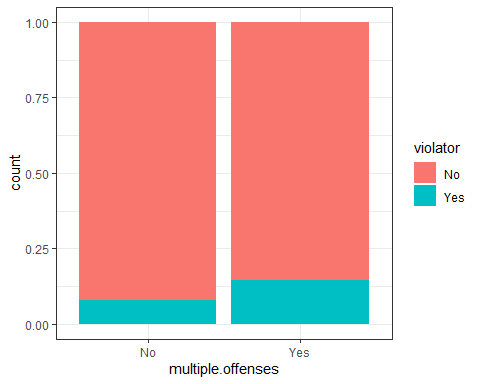
##   
## 1 2 3 4 6 8  
## No 1.00000000 1.00000000 0.33333333 1.00000000 0.50000000 0.92307692  
## Yes 0.00000000 0.00000000 0.66666667 0.00000000 0.50000000 0.07692308  
##   
## 9 10 11 12 13 14  
## No 0.66666667 0.50000000 0.50000000 0.85204082 0.96363636 0.94000000  
## Yes 0.33333333 0.50000000 0.50000000 0.14795918 0.03636364 0.06000000  
##   
## 15 16 17 18  
## No 0.97619048 0.81818182 1.00000000 0.92307692  
## Yes 0.02380952 0.18181818 0.00000000 0.07692308

**Multiple Offenses**

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



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

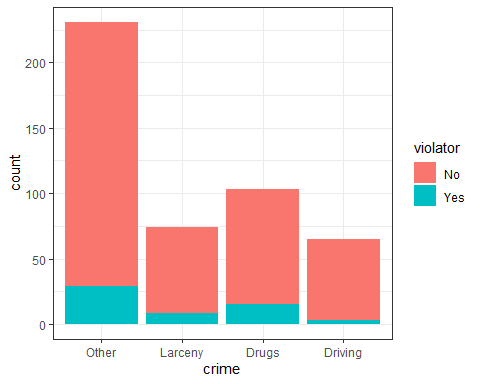


t5 = table(train$violator, train$multiple.offenses) #create a table object  
prop.table(t5, margin = 2 )

##   
## No Yes  
## No 0.91981132 0.85440613  
## Yes 0.08018868 0.14559387

**Crime**

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



t6 = table(train$violator, train$crime) #create a table object  
prop.table(t6, margin = 2 )

##   
## Other Larceny Drugs Driving  
## No 0.87445887 0.89189189 0.85436893 0.95384615  
## Yes 0.12554113 0.10810811 0.14563107 0.04615385

ANSWER: Male vs. Female - Doesn’t look like much difference  
Race - There’s a higher frequency for non-white.  
Age - There looks to be less of a tendency for those under the age of 25, and then there are almost none after the age of 50.  
State - Louisianna with a 42% Violator rate? Viginia with 2% ? That’s what we need to look into.  
Time Served - Can’t really see anything  
Max Sentence - Maybe something for 6 months and 12 months, but nothing is jumping out.  
Multiple Offenses - There is a 5% higher tendency for those with multiple offenses to have parole violation.  
Crime - Parole violation occurs less for driving offenses. For the rest it’s about the same.

**Task 3** I’ll be using the “state” variable

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   
## -1.0335 -0.5589 -0.2065 -0.2065 2.7780   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.75539 0.28944 -6.065 1.32e-09 \*\*\*  
## stateKentucky -0.02238 0.42567 -0.053 0.958067   
## stateLouisianna 1.40709 0.39351 3.576 0.000349 \*\*\*  
## stateVirginia -2.08191 0.53672 -3.879 0.000105 \*\*\*  
## ---  
## 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: 275.18 on 469 degrees of freedom  
## AIC: 283.18  
##   
## Number of Fisher Scoring iterations: 6

ANSWER: With the state of “other” being the baseline, the estimate for Kentucky is very close to it. Virginia has a high negative, indicating that the chance of being a parole violator is lower for that state. Louisianna has a high positive, showing that they have an increased chance of violating parole. AOC is 283.18 . We’ll see if we need that later on.

**Task 4** **Empty model**

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

**Task 4** **All model**

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   
## stateLouisianna 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   
## crimeDrugs -0.210563 0.413351 -0.509 0.61047   
## crimeDriving -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

**Task 4** **Forward stepwise**

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   
## stateLouisianna 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

ANSWER: AIC with only state - 283.18  
AIC for empty model - 342.04  
AIC for All - 268.09  
AIC with forward stepwise - 258.98 . The variables used are state, multiple offenses, and race. This jives with what we saw when we looked at the data as it correlated against “violator”.

**Task 5**

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

##   
## 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   
## stateLouisianna 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

ANSWER: We get the same model we got with forward stepwise regression. From looking at the p-values, all variables seem to be significant.

**Task 6** What is the predicted probability of parole violation of the two following parolees? Parolee1: Louisiana with multiple oﬀenses and white race

newdata = data.frame(state = "Louisianna", multiple.offenses = "Yes", race = "White")  
predict(threemod, newdata, type="response")

## 1   
## 0.3379961

ANSWER: The predicted probability is 0.3379961, or 34%

Parolee2: Kentucky with no multiple oﬀenses and other race

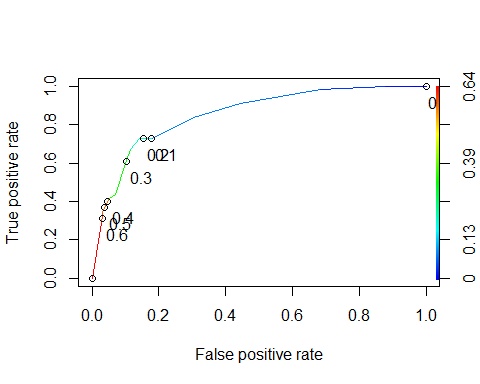
newdata = data.frame(state = "Kentucky", multiple.offenses = "No", race = "Other")  
predict(threemod, newdata, type="response")

## 1   
## 0.2069629

ANSWER: The predicted probability is 0.2069629, or 20.7%

**Task 7**

predictions = predict(threemod, type="response")  
#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predictions, train$violator)   
  
###You shouldn't need to ever change the next two lines:  
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))



#Determine threshold to balance sensitivity and specificity  
#DO NOT modify this code  
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 0.7272727  
## specificity 0.8588517  
## cutoff 0.2069629

ANSWER: The probability threshold that is the best balance is 0.2069629. This looks accurate when eyeballing the ROC graph.

**Task 8**

t1 = table(train$violator,predictions > 0.2069629)  
t1

##   
## FALSE TRUE  
## No 359 59  
## Yes 15 40

calculate Accuracy

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

## [1] 0.8435518

calculate sensitivity

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

## [1] 0.4040404

calculate specificity

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

## [1] 0.959893

ANSWER: Accuracy = 0.8435518 Sensitivity = 0.4040404 Specificity = 0.959893

Let’s assume this model is being used to target those parolees who are most likely to violate their parole, and this data will be used to perform some type of intervention. If extra intervention is given to someone who is not likely to violate their parole, that is a waste of resources (FN). This must be balanced against the cost of not intervening with a parolee who ends up violating their parole (FP) and they end up back in the system.

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

t1 = table(train$violator,predictions > 0.1)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8245243

t1

##   
## FALSE TRUE  
## No 350 68  
## Yes 15 40

t1 = table(train$violator,predictions > 0.2)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8435518

t1

##   
## FALSE TRUE  
## No 359 59  
## Yes 15 40

t1 = table(train$violator,predictions > 0.3)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8646934

t1

##   
## FALSE TRUE  
## No 376 42  
## Yes 22 33

t1 = table(train$violator,predictions > 0.4)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8964059

t1

##   
## FALSE TRUE  
## No 405 13  
## Yes 36 19

t1 = table(train$violator,predictions > 0.5)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8964059

t1

##   
## FALSE TRUE  
## No 405 13  
## Yes 36 19

t1 = table(train$violator,predictions > 0.6)  
(t1[1,1]+t1[2,2])/nrow(train)

## [1] 0.8921776

t1

##   
## FALSE TRUE  
## No 406 12  
## Yes 39 16

ANSWER: A probability threshold of 0.4 gives an accuracy of 0.8964059

predictions\_test = predict(threemod, test, type="response")  
t1 = table(test$violator,predictions\_test > 0.4)  
t1

##   
## FALSE TRUE  
## No 174 5  
## Yes 15 8

#accuracy  
(t1[1,1]+t1[2,2])/nrow(test)

## [1] 0.9009901

ANSWER: Using a probability threshold of 0.4 against the testing data gives an accuracy of 0.9