Emily Murray

Feb. 11, 2019

BAN502 - Mod. 3, Asmt 2

### Libraries & Data

library(tidyverse)

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

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

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.8  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.3.1 v forcats 0.3.0

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

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

## -- 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)

## 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)

## 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

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()  
## )

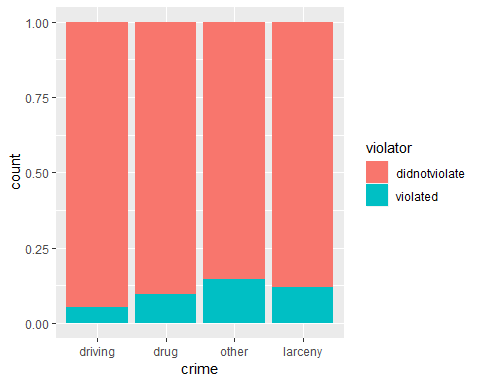
parole = parole %>%  
 mutate(male = as\_factor(as.character(male))) %>%  
 mutate(male = fct\_recode(male, "male" = "1", "female" = "0")) %>%  
 mutate(race = as\_factor(as.character(race))) %>%  
 mutate(race = fct\_recode(race, "white" = "1", "otherwise" = "2")) %>%  
 mutate(state = as\_factor(as.character(state))) %>%  
 mutate(state = fct\_recode(state, "kentucky" = "2", "louisiana" = "3", "virginia" = "4", "other" = "1")) %>%  
 mutate(crime = as\_factor(as.character(crime))) %>%  
 mutate(crime = fct\_recode(crime, "larceny" = "2", "drug" = "3", "driving" = "4", "other" = "1")) %>%  
 mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "multiple" = "1", "single" = "0")) %>%  
 mutate(violator = as\_factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator, "violated" = "1", "didnotviolate" = "0"))

### Task 1

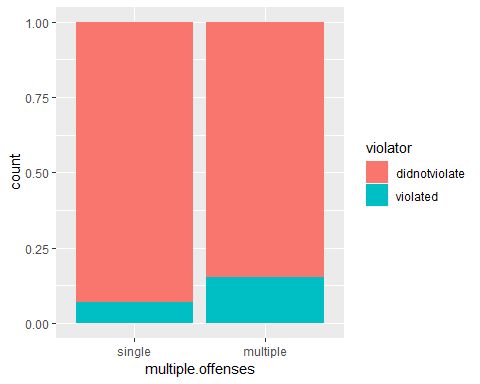
set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p = 0.7, list = FALSE)  
  
train = parole[train.rows,]  
test = parole[-train.rows,]

### Task 2

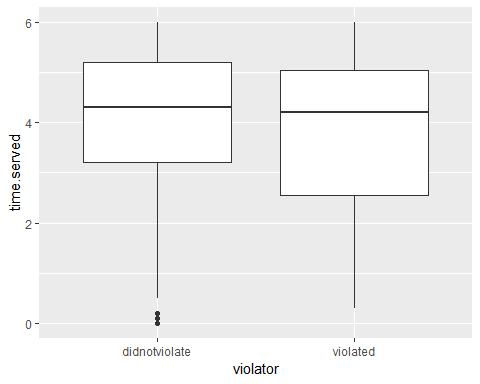
ggplot(train, aes(x=crime, fill = violator)) + geom\_bar(position = "fill")



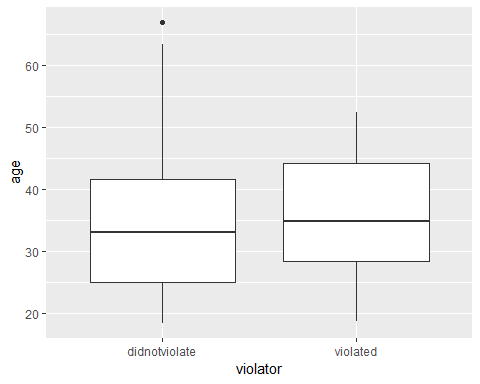
#Crime: no huge indicator here, drug & larceny are about the same, with the other category showing the most, which seems too general to show any significance  
  
ggplot(train, aes(x=multiple.offenses, fill = violator)) + geom\_bar(position = "fill")



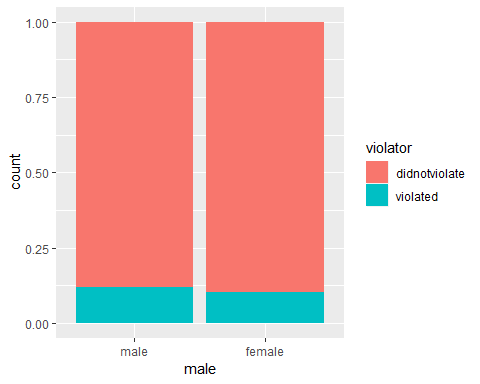
#Multiple offenses: those with multiple offenses seem to have a higher rate of violating their parole, this would make sense as violating parole could result in more jail time (this relationship may not be the best indicator then)  
  
ggplot(train, aes(x=violator, y = time.served)) + geom\_boxplot()



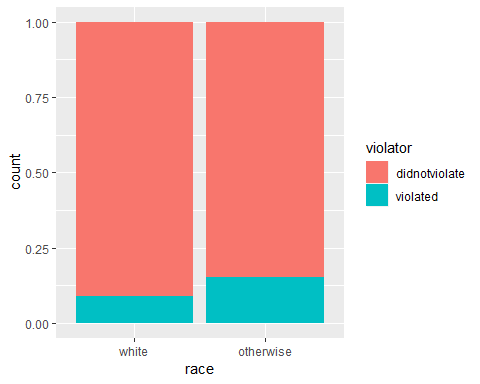
#Time served: the median and max time served of violators vs. non-violators are nearly identical, with those who violated having a slightly lower time served  
  
ggplot(train, aes(x=violator, y = age)) + geom\_boxplot()



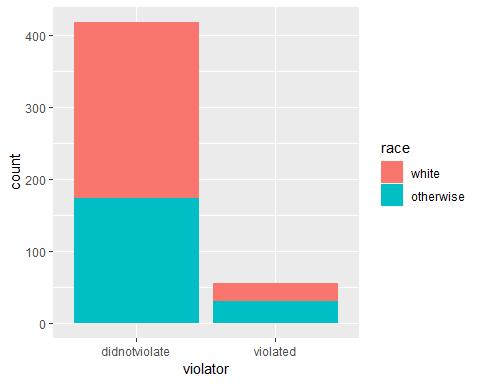
#Age: similarly, the median age for violators is about the same as those who do not violate; violators tend to generally be a bit older though = this could be due to the data available (more people at the median age in general and age can skew higher)  
  
ggplot(train, aes(x=male, fill=violator)) + geom\_bar(position = "fill")



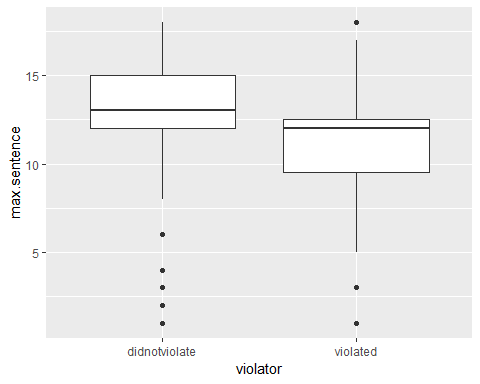
ggplot(train, aes(x=race, fill=violator)) + geom\_bar(position = "fill")



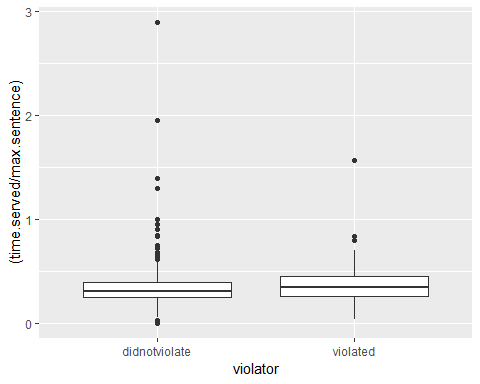
ggplot(train, aes(x=violator, fill=race)) + geom\_bar()



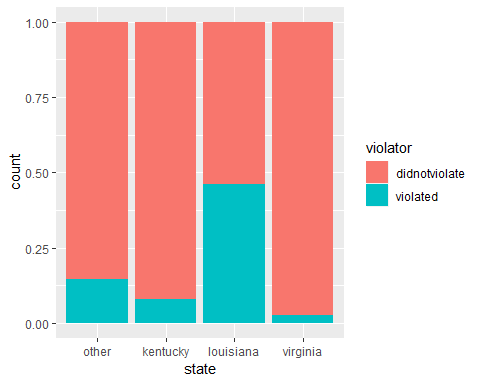
#Sex & Race: Sex doesn't appear significant; non-whites appear to have a bigher rate of violating parole  
  
ggplot(train, aes(x=violator, y = max.sentence)) + geom\_boxplot()



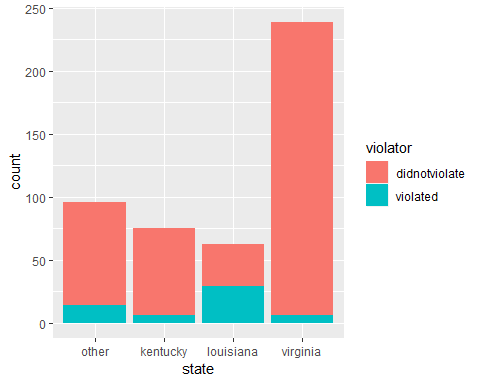
#Those with a lower max sentence seem to violate parole significantly less - this could be because max sentence = higher crime = more rigid reform in prison? or that the threat of a long sentence keeps prisoners   
  
ggplot(train, aes(x=violator, y=(time.served/max.sentence))) + geom\_boxplot()



#Ratio of sentence time served: my guess here is that if an inmate serves the full sentence or more, they will be averse to violating their parole. If they are let off early, they may not feel as threatened by being incarcerated again. This could be a huge stretch, the boxplot looks interesting but probably not relevant for this assignment  
  
ggplot(train, aes(x=state, fill = violator)) + geom\_bar(position = "fill")



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



#State: Louisiana has a way higher rate of parole violators. Even though they have fewer inmates represented in the data, there are still enough to show if might be significant.

*Variables that seem the most significant include: max.sentence, state, multiple.offenses, race*

### Task 3

model1 = glm(violator ~ max.sentence, train, family = "binomial")  
summary(model1)

##   
## Call:  
## glm(formula = violator ~ max.sentence, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.1970 -0.5121 -0.4699 -0.3613 2.4932   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.22883 0.51865 0.441 0.659   
## max.sentence -0.18284 0.04237 -4.315 1.6e-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: 321.79 on 471 degrees of freedom  
## AIC: 325.79  
##   
## Number of Fisher Scoring iterations: 5

*Model quality seems okay, p-value is significant and shows that as max sentence increases, likelihood of parole violation goes down.*

### Task 4

mod2 = glm(violator ~ max.sentence + state, train, family = "binomial")  
summary(mod2)

##   
## Call:  
## glm(formula = violator ~ max.sentence + state, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.161 -0.423 -0.227 -0.224 2.715   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.95103 0.68990 -2.828 0.00468 \*\*   
## max.sentence 0.01363 0.04639 0.294 0.76884   
## statekentucky -0.66318 0.51621 -1.285 0.19889   
## statelouisiana 1.66695 0.43335 3.847 0.00012 \*\*\*  
## statevirginia -1.89943 0.50521 -3.760 0.00017 \*\*\*  
## ---  
## 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: 264.49 on 468 degrees of freedom  
## AIC: 274.49  
##   
## Number of Fisher Scoring iterations: 6

#This looks better, AIC is down but p values are iffy for some variables.  
   
mod3 = glm(violator ~ max.sentence + state + multiple.offenses, train, family = "binomial")  
summary(mod3)

##   
## Call:  
## glm(formula = violator ~ max.sentence + state + multiple.offenses,   
## family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5382 -0.3615 -0.2549 -0.2339 2.6455   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.22107 0.82313 -3.913 9.11e-05 \*\*\*  
## max.sentence 0.05839 0.05144 1.135 0.25628   
## statekentucky -0.31701 0.53967 -0.587 0.55692   
## statelouisiana 1.31270 0.46155 2.844 0.00445 \*\*   
## statevirginia -2.73971 0.54493 -5.028 4.97e-07 \*\*\*  
## multiple.offensesmultiple 1.67458 0.39997 4.187 2.83e-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: 245.58 on 467 degrees of freedom  
## AIC: 257.58  
##   
## Number of Fisher Scoring iterations: 6

#AIC is down again and looks like multiple offenses is significant.  
  
mod4 = glm(violator ~ max.sentence + state + multiple.offenses +race, train, family = "binomial")  
summary(mod4)

##   
## Call:  
## glm(formula = violator ~ max.sentence + state + multiple.offenses +   
## race, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5622 -0.3615 -0.2850 -0.1651 2.8178   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.50690 0.85551 -4.099 4.15e-05 \*\*\*  
## max.sentence 0.06763 0.05360 1.262 0.20708   
## statekentucky -0.43515 0.54530 -0.798 0.42487   
## statelouisiana 0.76022 0.51347 1.481 0.13872   
## statevirginia -3.36095 0.61695 -5.448 5.10e-08 \*\*\*  
## multiple.offensesmultiple 1.76711 0.41035 4.306 1.66e-05 \*\*\*  
## raceotherwise 1.03845 0.40232 2.581 0.00985 \*\*   
## ---  
## 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: 238.81 on 466 degrees of freedom  
## AIC: 252.81  
##   
## Number of Fisher Scoring iterations: 6

mod5 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial")  
summary(mod5)

##   
## 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   
## statelouisiana 0.5292 0.4769 1.110 0.2672   
## statevirginia -3.2301 0.6028 -5.358 8.39e-08 \*\*\*  
## multiple.offensesmultiple 1.6596 0.3985 4.165 3.12e-05 \*\*\*  
## raceotherwise 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

#AIC is about the same with race added and with max.sentence removed. (252)  
  
  
#Backward Stepwise  
allmod = glm(violator ~., train, family = "binomial")   
summary(allmod)

##   
## Call:  
## glm(formula = violator ~ ., 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) -5.33220 1.39750 -3.816 0.000136 \*\*\*  
## malefemale -0.53377 0.49107 -1.087 0.277051   
## raceotherwise 1.06698 0.41324 2.582 0.009824 \*\*   
## age 0.03361 0.01696 1.982 0.047493 \*   
## statekentucky -0.30132 0.56939 -0.529 0.596665   
## statelouisiana 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   
## max.sentence 0.08458 0.05644 1.499 0.133963   
## multiple.offensesmultiple 1.72841 0.41857 4.129 3.64e-05 \*\*\*  
## crimedrug 0.11232 0.71712 0.157 0.875535   
## crimeother 0.87795 0.62271 1.410 0.158571   
## crimelarceny 1.06304 0.73146 1.453 0.146139   
## ---  
## 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

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

## Start: AIC=256.16  
## violator ~ male + race + age + state + time.served + max.sentence +   
## multiple.offenses + crime  
##   
## 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 + race + age + state + max.sentence + multiple.offenses +   
## crime  
##   
## 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 + race + age + 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 ~ race + age + 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 ~ race + age + 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 \*\*\*  
## raceotherwise 1.07386 0.40527 2.650 0.00806 \*\*   
## age 0.02636 0.01660 1.588 0.11224   
## statekentucky -0.41360 0.54930 -0.753 0.45147   
## statelouisiana 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.offensesmultiple 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

*The best model has race + age + state + max.sentence + multiple.offenses - the same as my model 4 plus age; but the AIC is also around 252 here. Age and max sentence don’t look hugely infulential in the summary data.*

### Task 5

mod5 = glm(violator ~ state + multiple.offenses + race, train, family = "binomial")  
summary(mod5)

##   
## 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   
## statelouisiana 0.5292 0.4769 1.110 0.2672   
## statevirginia -3.2301 0.6028 -5.358 8.39e-08 \*\*\*  
## multiple.offensesmultiple 1.6596 0.3985 4.165 3.12e-05 \*\*\*  
## raceotherwise 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 was the last model I tried manually before checking sstepwise. The quality is just about as good as the stepwise model. Louisiana and Kentucky have high p values, but Virginia’s is very small. All other variables look significant.

# Task 6

newdata = data.frame(state = "louisiana", multiple.offenses = "multiple", race = "white")  
predict(mod5, newdata, type="response")

## 1   
## 0.408682

#0.408682  
  
newdata = data.frame(state = "kentucky", multiple.offenses = "single", race = "otherwise")  
predict(mod5, newdata, type="response")

## 1   
## 0.1153326

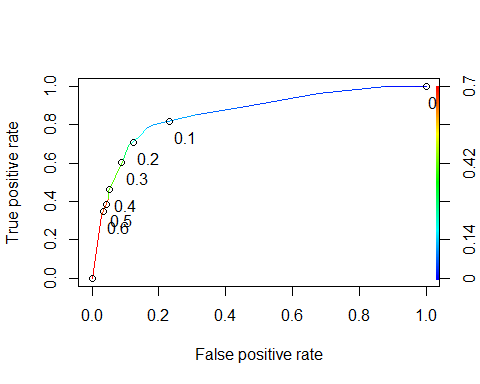
#0.1153326

# Task 7

predictions = predict(mod5, type="response")  
head(predictions)

## 1 2 3 4 5 6   
## 0.07187555 0.17425270 0.07187555 0.17425270 0.17425270 0.07187555

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

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 0.7818182  
## specificity 0.8373206  
## cutoff 0.1161882

#sensitivity 0.7818182  
#specificity 0.8373206  
#cutoff 0.1161882  
  
t1 = table(train$violator,predictions > 0.1161882)  
t1

##   
## FALSE TRUE  
## didnotviolate 357 61  
## violated 14 41

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

## [1] 0.8414376

#accuracy 0.8414376

*There are very few false positives with this threshold, meaning it looks unlikely toincorrectly predict/assume the parolee will violate. Alternatively, the number predicted not to violate who DID (false negative) is pretty high. If we are not looking out for these people, there is a higher chance they slip up and violate parole. So it would make sense to seek a model predicting fewer false negatives (increase the threshold).*

# Task 9

t1 = table(train$violator,predictions > 0.2)  
t1

##   
## FALSE TRUE  
## didnotviolate 367 51  
## violated 16 39

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

## [1] 0.858351

#Accuracy goes up slightly (~2.5-3%)  
  
t1 = table(train$violator,predictions > 0.3)  
t1

##   
## FALSE TRUE  
## didnotviolate 397 21  
## violated 30 25

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

## [1] 0.8921776

#Accuracy increases another ~4%  
  
t1 = table(train$violator,predictions > 0.4)  
t1

##   
## FALSE TRUE  
## didnotviolate 397 21  
## violated 30 25

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

## [1] 0.8921776

#Accuracy is not affected with another .1 increase  
  
t1 = table(train$violator,predictions > 0.1)  
t1

##   
## FALSE TRUE  
## didnotviolate 340 78  
## violated 11 44

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

## [1] 0.8118393

#Decreasing the threshold by even .016 decreases the accuracy from 84 to 81

# Task 10

testpred = predict(mod5, test, type="response")  
  
  
t2 = table(test$violator,testpred > 0.1161882)  
t2

##   
## FALSE TRUE  
## didnotviolate 156 23  
## violated 12 11

(t2[1,1]+t2[2,2])/nrow(test)

## [1] 0.8267327

t2 = table(test$violator,testpred > 0.3)  
t2

##   
## FALSE TRUE  
## didnotviolate 172 7  
## violated 15 8

(t2[1,1]+t2[2,2])/nrow(test)

## [1] 0.8910891