## Logistic Regression

# 

library(tidyverse)

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

## v ggplot2 3.1.0 v purrr 0.3.2   
## v tibble 2.1.1 v dplyr 0.8.0.1  
## v tidyr 0.8.3 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)

##   
## 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

#install.packages("ROCR")  
library(ROCR)

## Loading required package: gplots

##   
## 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()  
## )

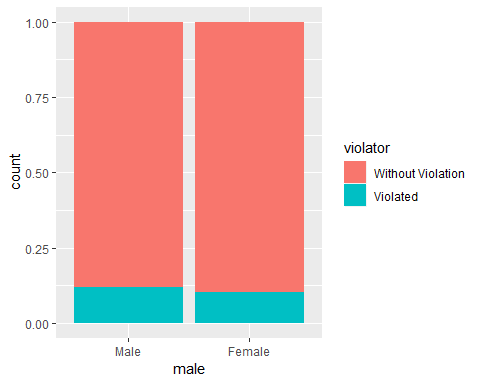
parole = parole %>% mutate( male = as\_factor(as.character(male))) %>%  
 mutate(male = fct\_recode(male,  
 "Male" = "1",  
 "Female" = "0"))  
  
  
parole = parole %>% mutate(race = as\_factor(as.character(race))) %>%  
 mutate(race = fct\_recode(race,  
 "White" = "1",  
 "Other" = "2"))  
  
parole = parole %>% mutate(state = as\_factor(as.character(state))) %>%  
 mutate(state = fct\_recode(state,  
 "Other" = "1",  
 "Kentucky" = "2",  
 "Louisiana" = "3",  
 "Virginia" = "4"))  
  
parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>%  
 mutate(crime = fct\_recode(crime,  
 "Other" = "1",  
 "Larceny" = "2",  
 "Drug-related" = "3",  
 "Driving-related" = "4"))  
  
parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses,  
 "Multiple" = "1",  
 "Other" = "0"))  
  
parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator,  
 "Violated" = "1",  
 "Without Violation" = "0"))

### Task 1

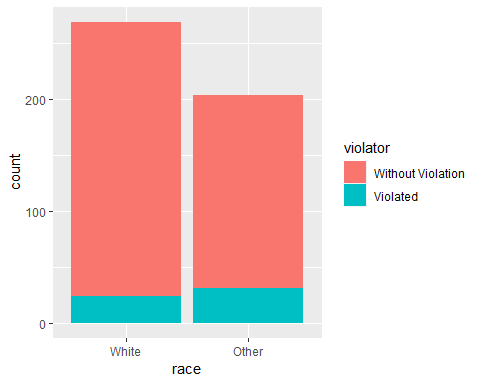
set.seed(12345)  
train.rows = createDataPartition(y=parole$violator, p=0.7, list=FALSE)  
train = parole[train.rows,]  
test = parole[-train.rows,]

### Task 2

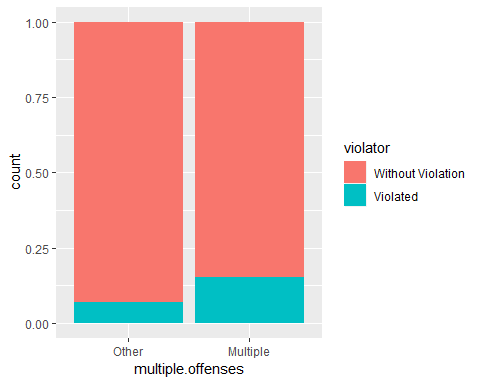
ggplot(train, aes(x=male, fill = violator)) + geom\_bar(position="fill")



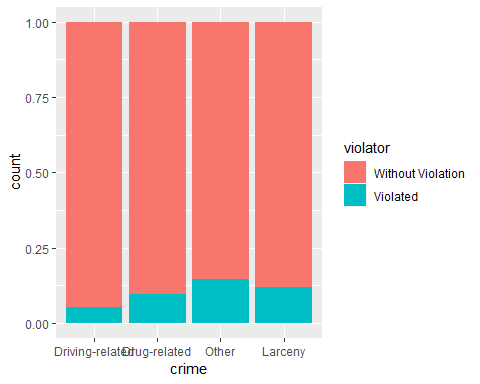
ggplot(train, aes(x=race, fill = violator)) + geom\_bar()



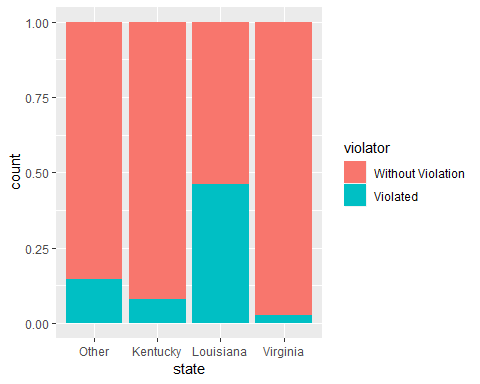
ggplot(train, aes(x=multiple.offenses, fill = violator)) +geom\_bar(position="fill")



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



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



Using the position=“fill” in the bar graph makes each bar the same height, and in some cases makes it easier to identify the proprotion of the variable by the legend. Based of the data visualizations, Louisiana seems to have the most potential parole violators compared to the other locations identified. The one variable I was specifically targeting was whether multople offenses had any impact on whether parole was violated. According to the graph, multiple offenses does have a high probability of violating parole, but not as high as I imagined it would be.

### 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   
## -0.5763 -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.offensesMultiple 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

Using the variable “multiple.offenses” to build a logistic regression model, the quality of the model is good. The p-value for multiple.offenses shows to be a significant predictor of whether parole is violated or not. Also, the AIC is fairly low which indicates that this variable is a good fit in predicting the response variable.

### Task 4

allmod = glm(violator ~., train, family = "binomial")  
emptymod = glm(violator ~1, train, family = "binomial")  
  
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   
## stateLouisiana 0.86000 0.51900 1.657 0.09751 .   
## stateVirginia -3.34208 0.62057 -5.386 7.22e-08 \*\*\*  
## multiple.offensesMultiple 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

Using the forward stepwise method, the AIC for this model is low which indicates a good model. The varibales that are significant include state=Virginia and multiple offenses, as well as other race. The state of Virignia being one of the significant variables was surprising since previously the bar chart had indicated that Louisiana was the highest state of likely parole violations.

### Task 5

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

##   
## 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 \*\*\*  
## 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

A logistic regression with only variables state, multiple offeneses and race came out with almost the exact AIC value as the forward stepwise method did. However, the forward stepwise method included additional variables such as age and max sentence. In this model multiple offesnses and the state of Virignia are the significant variables.

### Task 6

parolee1 = data.frame(state = "Louisiana", multiple.offenses = "Multiple", race = "White")  
predict(mod2, parolee1, type = "response")

## 1   
## 0.408682

parolee2 = data.frame(state = "Kentucky", multiple.offenses = "Other", race = "Other")  
predict(mod2, parolee2, type = "response")

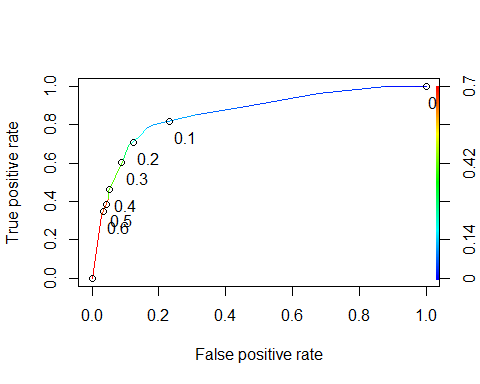
## 1   
## 0.1153326

### Task 7

predictions = predict(mod2, 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,label.ordering = c("Without Violation", "Violated"))  
  
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))



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

## [1] 0.8586124

### 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

t1 = table(train$violator,predictions > 0.1161882)  
t1

##   
## FALSE TRUE  
## Without Violation 357 61  
## Violated 14 41

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

## [1] 0.8414376

Accuracy = 0.84

If a parolee gets incorrectly classified as not violating parole when they really did, there could be a possibility that they don’t receive the consequences that they would if it were known that they violated parole. If a parolee gets classified as violating parolee, but they really didn’t, the parolee could receive punishment for violating when they really did not vioalte.

### Task 9

t1 = table(train$violator,predictions > 0.5)  
t1

##   
## FALSE TRUE  
## Without Violation 406 12  
## Violated 37 18

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

## [1] 0.8964059

### Task 10

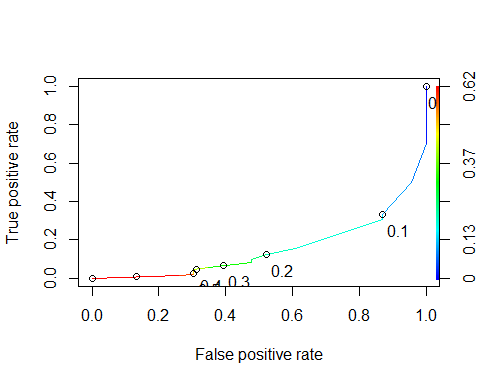
mod3= glm(violator ~ state + multiple.offenses + race, test, family = "binomial")  
summary(mod3)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = test)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3191 -0.5607 -0.1746 -0.1529 2.8950   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4666 0.5282 -4.670 3.01e-06 \*\*\*  
## stateKentucky 0.6960 0.6275 1.109 0.26731   
## stateLouisiana 1.0817 0.7241 1.494 0.13522   
## stateVirginia -3.4208 1.1513 -2.971 0.00297 \*\*   
## multiple.offensesMultiple 1.4436 0.5854 2.466 0.01367 \*   
## raceOther 0.2683 0.5792 0.463 0.64318   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 143.22 on 201 degrees of freedom  
## Residual deviance: 107.85 on 196 degrees of freedom  
## AIC: 119.85  
##   
## Number of Fisher Scoring iterations: 7

predictions = predict(mod3, type="response")  
head(predictions)

## 1 2 3 4 5 6   
## 0.07823664 0.31981733 0.07823664 0.07823664 0.07823664 0.07823664

ROCRpred = prediction(predictions, test$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))



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

## [1] 0.1541171

t1 = table(test$violator,predictions > 0.5)  
t1

##   
## FALSE TRUE  
## Without Violation 175 4  
## Violated 16 7

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

## [1] 0.9009901