library(tidyverse)

## -- Attaching packages --------

## 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 -----------------  
## 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)

## 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()  
## )

head(parole)

## # A tibble: 6 x 9  
## male race age state time.served max.sentence multiple.offens~ crime  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1 33.2 1 5.5 18 0 4  
## 2 0 1 39.7 1 5.4 12 0 3  
## 3 1 2 29.5 1 5.6 12 0 3  
## 4 1 1 22.4 1 5.7 18 0 1  
## 5 1 2 21.6 1 5.4 12 0 1  
## 6 1 2 46.7 1 6 18 0 4  
## # ... with 1 more variable: violator <dbl>

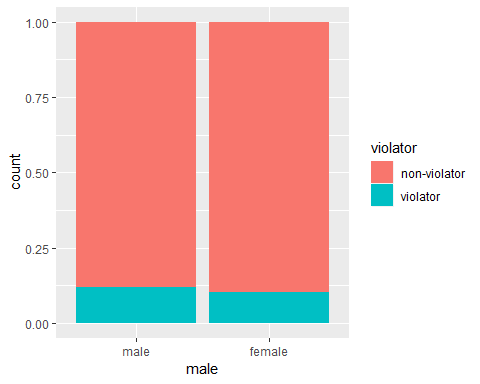
parole = parole %>% mutate(male = as\_factor(as.character(male))) %>%  
mutate(male = fct\_recode(male,  
"female" = "0",  
"male" = "1")) %>% mutate(race = as\_factor(as.character(race))) %>%  
mutate(race = fct\_recode(race,  
"white" = "1",  
"other" = "2")) %>% mutate(state = as\_factor(as.character(state))) %>%  
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-related" = "3",  
"driving-related" = "4",  
"other" = "1")) %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>%  
mutate(multiple.offenses = fct\_recode(multiple.offenses,  
"otherwise" = "0",  
"multiple offenses" = "1")) %>% mutate(violator = as\_factor(as.character(violator))) %>%  
mutate(violator = fct\_recode(violator,  
"non-violator" = "0",  
"violator" = "1"))

Task 1:

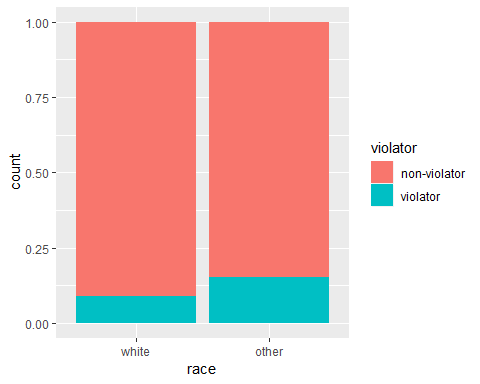
set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE)  
train = parole[train.rows,]   
test = parole[-train.rows,]

Task2:

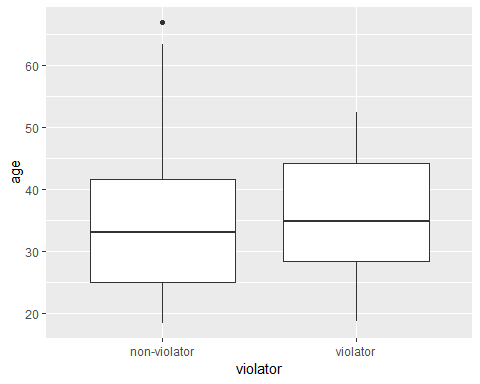
ggplot(train,aes(x=male, fill=violator)) + geom\_bar(position = "fill")



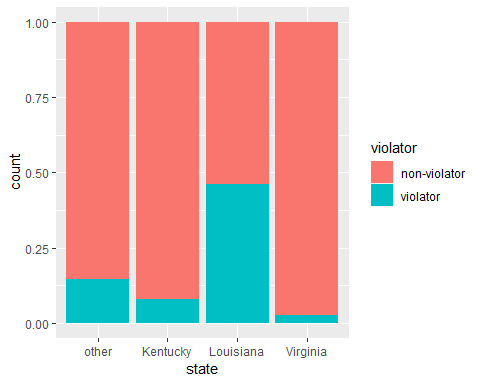
ggplot(train,aes(x=race, fill=violator)) + geom\_bar(position = "fill")



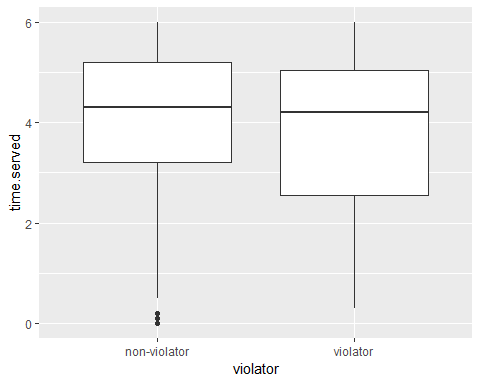
ggplot(train, aes(x=violator, y = age)) + geom\_boxplot()



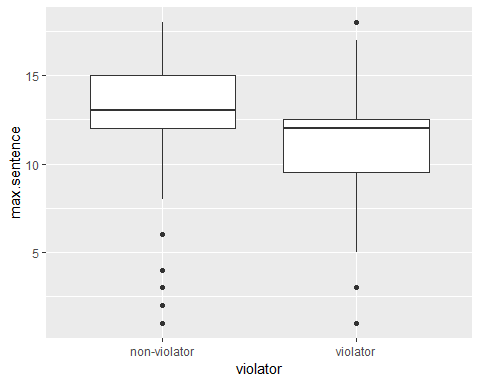
ggplot(train, aes(x=state, fill=violator)) + geom\_bar(position = "fill")



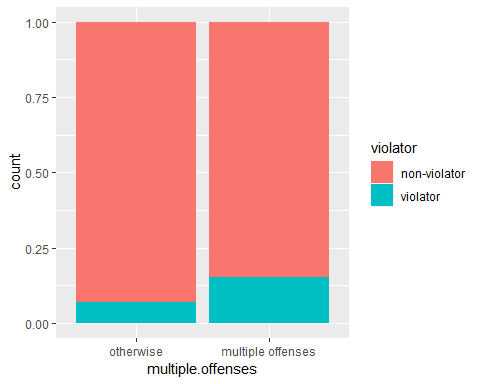
ggplot(train, aes(x=violator, y = time.served)) + geom\_boxplot()



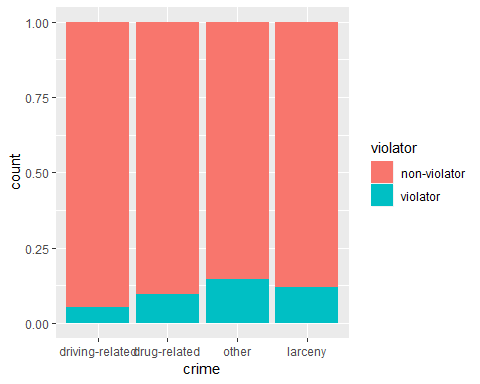
ggplot(train, aes(x=violator, y = max.sentence)) + geom\_boxplot()



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



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



Upon reviewing the visualizations in task 2, I have identied that the state, multiple.offenses, and crime variables are the ones who would be most predictive of the response variable violator. This is because in the three graphics, you are able to clearly see the differences between the categories and how the response variable is affected by them.

Task3:

logregmod <- glm(violator ~ multiple.offenses, train, family = "binomial")  
  
summary(logregmod)

##   
## 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 offenses 0.9018 0.3247 2.777 0.00549  
##   
## (Intercept) \*\*\*  
## multiple.offensesmultiple offenses \*\*   
## ---  
## 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

The positive coffecieint of the multiple.offenses variable suggests that the probability of an individual violating parole, given that they have multiple offenses, increases. The AIC value is low, 335.5. We will reference this number when determining the best model.

Task 4:

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

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  
## raceother 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 offenses 1.77974 0.41476 4.291 1.78e-05  
##   
## (Intercept) \*\*\*  
## raceother \*\*   
## age   
## stateKentucky   
## stateLouisiana .   
## stateVirginia \*\*\*  
## max.sentence   
## multiple.offensesmultiple offenses \*\*\*  
## ---  
## 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

Backward stepwise was utilized to determine the best model to predict whether an individal will violate their parole. The AIC value of the model was significantly lower than the model using just the multiple.offenses variable, the AIC value is 252.28. The model appears to be of good quality. The variables that are significant in thise model is raceother, stateLouisiana, and multiple offenses. An interpretation of this model is if someone is from either Kentucky or Virginia, the probabily of parole being violated goes down because of the negative value. This is due to the fact that the data favors those states who have less amount of parole violators. In my opinion, being from a certain state does not necessarily predict whether someone will violate their parole or not.

Task 5:

logregmod2 <- glm(violator ~ multiple.offenses + state + race, train, family = "binomial")  
  
summary(logregmod2)

##   
## Call:  
## glm(formula = violator ~ multiple.offenses + state + 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  
## multiple.offensesmultiple offenses 1.6596 0.3985 4.165 3.12e-05  
## 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  
## raceother 1.0024 0.3966 2.528 0.0115  
##   
## (Intercept) \*\*\*  
## multiple.offensesmultiple offenses \*\*\*  
## stateKentucky   
## stateLouisiana   
## stateVirginia \*\*\*  
## raceother \*   
## ---  
## 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

The second logistic regression model seems be about the same than the one created by the backward stepwise. The second logistic regression model has an AIC value of 252.42 compared to 252.28. I believe this is due to the fact of the couple of variables missing from the first model.

Task 6:

Parolee1 = data.frame(race = c("white","other"), multiple.offenses = c("multiple offenses","otherwise"), state = c("Louisiana","Kentucky"))  
  
predict(logregmod2, Parolee1, type="response")

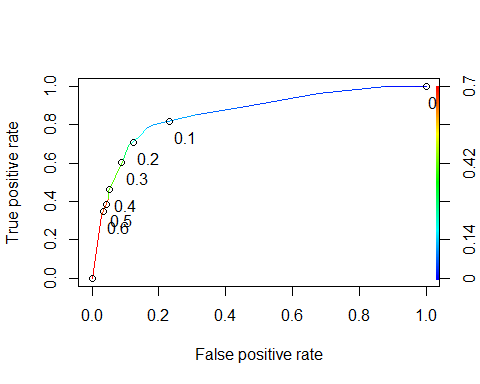
## 1 2   
## 0.4086820 0.1153326

Parolee 1 has about a 41% chance of violating their parole, and Parolee 2 has about a 12% chance of violating theirs.

TasK 7:

predictions = predict(logregmod2, train, type="response")

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))



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  
## non-violator 357 61  
## violator 14 41

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

## [1] 0.8414376

sensitivity: 0.7818182

specificity: 0.8373206

Accuracy: 0.8414376

Implcations of incorrectly classifying a parolee would be someone being misclassified as violating their parole would serve a longer sentnece, or someone who did actually violate their parole and not be classified could be enticed to do something they should not be doing.

Task 9:

t1 = table(train$violator,predictions > 0.25)  
t1

##   
## FALSE TRUE  
## non-violator 373 45  
## violator 17 38

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

## [1] 0.8689218

t1 = table(train$violator,predictions > 0.65)  
t1

##   
## FALSE TRUE  
## non-violator 406 12  
## violator 37 18

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

## [1] 0.8964059

t1 = table(train$violator,predictions > 1)  
t1

##   
## FALSE  
## non-violator 418  
## violator 55

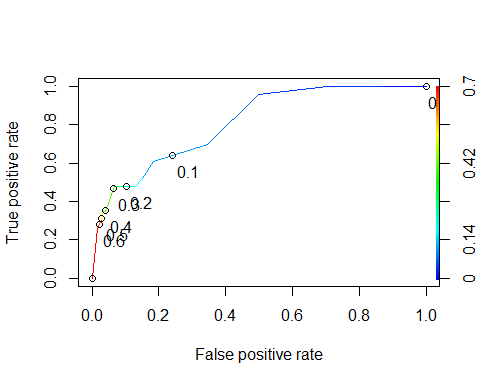
(t1[1])/nrow(train)

## [1] 0.8837209

Task 10:

predictionsTest = predict(logregmod2, test, type="response")

ROCRpred = prediction(predictionsTest, test$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))



t1 = table(test$violator,predictionsTest > 0.65)  
t1

##   
## FALSE TRUE  
## non-violator 176 3  
## violator 17 6

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

## [1] 0.9009901

The accuracy for the of the model, using the testing set, is .90.