## BAN 502

## Mandy Hawkins

## Module 3: Clafication with Logistic Regression

library(tidyverse)

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

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- 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(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

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

library(caret)

## 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\_integer(),  
## race = col\_integer(),  
## age = col\_double(),  
## state = col\_integer(),  
## time.served = col\_double(),  
## max.sentence = col\_integer(),  
## multiple.offenses = col\_integer(),  
## crime = col\_integer(),  
## violator = col\_integer()  
## )

View(parole)

Drop empty rows

parole = parole %>% drop\_na() #delete any row with an NA value   
str(parole) #check structure after the drop

## Classes 'tbl\_df', 'tbl' and 'data.frame': 675 obs. of 9 variables:  
## $ male : int 1 0 1 1 1 1 1 0 0 1 ...  
## $ race : int 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 : int 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 : int 18 12 12 18 12 18 18 12 13 12 ...  
## $ multiple.offenses: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ crime : int 4 3 3 1 1 4 3 1 3 2 ...  
## $ violator : int 0 0 0 0 0 0 0 0 0 0 ...

Convert the male, race, state, crime, multiple.oﬀenses, and violator variables to factors.

parole = parole %>% mutate(male = as\_factor(as.character(male))) %>% mutate(male = fct\_recode(male, "Female" = "0", "Male" = "1"))

parole = parole %>% mutate(race = as\_factor(as.character(race))) %>% mutate(race = fct\_recode(race, "NotWhite" = "2", "White" = "1"))

parole = parole %>% mutate(state = as\_factor(as.character(state))) %>% mutate(state = fct\_recode(state, "Kentucky"= "2", "Louisiana"="3", "Virginia"= "4", "Not\_KY\_LA\_or\_VA" ="1"))

parole = parole %>% mutate(crime = as\_factor(as.character(crime))) %>% mutate(crime = fct\_recode(crime, "Larceny" ="2", "Drug" = "3", "Driving" ="4", "Other" = "1"))

parole = parole %>% mutate(violator = as\_factor(as.character(violator))) %>% mutate(violator = fct\_recode(violator, "No" = "0", "Yes" = "1"))

parole = parole %>% mutate(multiple.offenses = as\_factor(as.character(multiple.offenses))) %>% mutate(multiple.offenses = fct\_recode(multiple.offenses, "No" = "0", "Yes" = "1"))

## Task 1

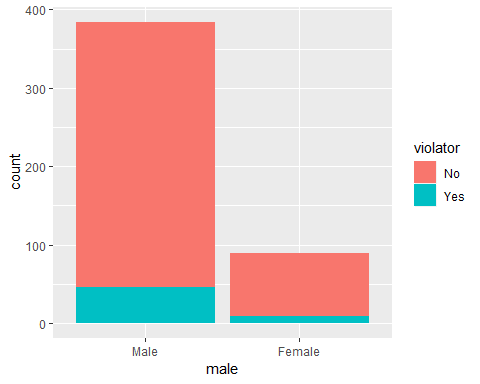
Split the data into training and testing sets. Your training set should have 70% of the data. Use a random number (set.seed) of 12345.

set.seed(12345)  
train.rows = createDataPartition(y = parole$violator, p=0.7, list = FALSE) #70% in training  
train = parole[train.rows,]   
test = parole[-train.rows,]

## Task 2

Our objective is to predict whether or not a parolee will violate his/her parole. Use appropriate data visualizations and/or tables to identify which variables in the training set appear to be most predictive of the response variable “violator”. Provide a brief explanation of your thought process. Hint: When plotting two categorical variables against each other, consider using a barplot with geom\_bar(position=“ﬁll”). See what you get when you try this

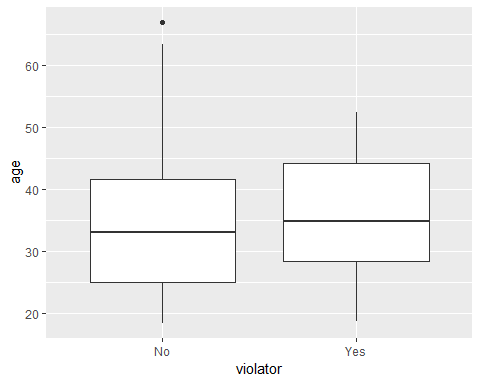
ggplot(train,aes(x=male, fill = violator)) + geom\_bar()

 Gender does appear to affect whether or not the person will violate parole.

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

 While white race violators are lower, there isn’t much sigificance between the two groups.

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

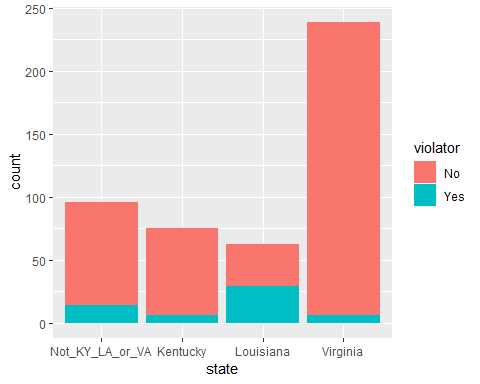
 Will look at table.

t1 = table(parole$violator,parole$age)  
prop.table(t1, margin = 2)

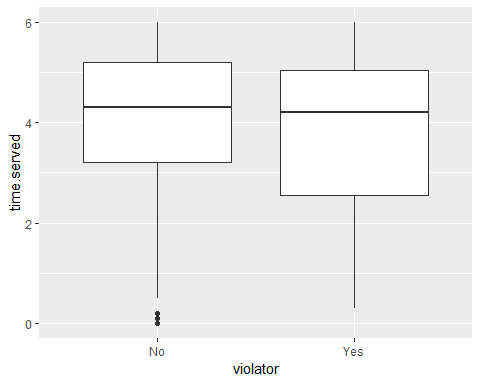
##   
## 18.4 18.5 18.7 18.8 19 19.1  
## No 1.0000000 1.0000000 0.5000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.5000000 0.0000000 0.0000000 0.0000000  
##   
## 19.2 19.3 19.4 19.5 19.6 19.7  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 19.9 20 20.2 20.3 20.4 20.5  
## No 0.6666667 1.0000000 0.7500000 0.5000000 1.0000000 0.7500000  
## Yes 0.3333333 0.0000000 0.2500000 0.5000000 0.0000000 0.2500000  
##   
## 20.6 20.7 20.8 20.9 21 21.1  
## No 0.6666667 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.3333333 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 21.2 21.3 21.4 21.5 21.6 21.7  
## No 1.0000000 1.0000000 1.0000000 1.0000000 0.5000000 0.5000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.5000000 0.5000000  
##   
## 21.8 21.9 22 22.1 22.2 22.3  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 22.4 22.5 22.6 22.8 22.9 23  
## No 0.6666667 1.0000000 1.0000000 0.6666667 1.0000000 1.0000000  
## Yes 0.3333333 0.0000000 0.0000000 0.3333333 0.0000000 0.0000000  
##   
## 23.1 23.2 23.3 23.4 23.6 23.7  
## No 1.0000000 1.0000000 0.8333333 1.0000000 0.8000000 0.6666667  
## Yes 0.0000000 0.0000000 0.1666667 0.0000000 0.2000000 0.3333333  
##   
## 23.8 24 24.2 24.3 24.4 24.5  
## No 1.0000000 1.0000000 0.8333333 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.1666667 0.0000000 0.0000000 0.0000000  
##   
## 24.6 24.7 24.8 24.9 25 25.1  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 25.2 25.3 25.4 25.5 25.6 25.7  
## No 1.0000000 0.7500000 1.0000000 1.0000000 0.8571429 1.0000000  
## Yes 0.0000000 0.2500000 0.0000000 0.0000000 0.1428571 0.0000000  
##   
## 25.8 25.9 26 26.3 26.4 26.5  
## No 0.5000000 1.0000000 0.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.5000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000  
##   
## 26.6 26.8 26.9 27 27.1 27.2  
## No 1.0000000 0.6666667 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.3333333 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 27.3 27.4 27.5 27.6 27.7 27.8  
## No 1.0000000 0.0000000 0.5000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 1.0000000 0.5000000 0.0000000 0.0000000 0.0000000  
##   
## 27.9 28 28.1 28.2 28.3 28.4  
## No 1.0000000 1.0000000 0.6666667 1.0000000 1.0000000 0.6666667  
## Yes 0.0000000 0.0000000 0.3333333 0.0000000 0.0000000 0.3333333  
##   
## 28.5 28.7 28.8 28.9 29 29.1  
## No 0.5000000 1.0000000 0.8000000 0.6666667 1.0000000 1.0000000  
## Yes 0.5000000 0.0000000 0.2000000 0.3333333 0.0000000 0.0000000  
##   
## 29.2 29.5 29.6 29.7 29.9 30  
## No 1.0000000 0.6666667 1.0000000 1.0000000 0.2500000 1.0000000  
## Yes 0.0000000 0.3333333 0.0000000 0.0000000 0.7500000 0.0000000  
##   
## 30.1 30.2 30.3 30.4 30.7 30.8  
## No 1.0000000 1.0000000 1.0000000 1.0000000 0.5000000 0.6666667  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.5000000 0.3333333  
##   
## 31 31.1 31.2 31.3 31.4 31.5  
## No 0.8333333 1.0000000 1.0000000 1.0000000 0.5000000 0.5000000  
## Yes 0.1666667 0.0000000 0.0000000 0.0000000 0.5000000 0.5000000  
##   
## 31.6 31.7 31.8 32 32.1 32.2  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.3333333  
##   
## 32.3 32.4 32.5 32.6 32.7 32.8  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.7500000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.2500000  
##   
## 32.9 33 33.2 33.3 33.4 33.5  
## No 1.0000000 1.0000000 1.0000000 0.0000000 1.0000000 0.5000000  
## Yes 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.5000000  
##   
## 33.6 33.7 33.8 33.9 34 34.1  
## No 1.0000000 0.5000000 1.0000000 1.0000000 1.0000000 0.0000000  
## Yes 0.0000000 0.5000000 0.0000000 0.0000000 0.0000000 1.0000000  
##   
## 34.2 34.3 34.4 34.5 34.6 34.7  
## No 0.6666667 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.3333333 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 34.8 34.9 35 35.1 35.2 35.3  
## No 1.0000000 0.6666667 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.3333333 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 35.4 35.5 35.6 35.8 35.9 36  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 36.1 36.2 36.3 36.4 36.5 36.6  
## No 1.0000000 1.0000000 1.0000000 1.0000000 0.8000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.2000000 0.0000000  
##   
## 36.7 36.8 37 37.2 37.3 37.4  
## No 1.0000000 1.0000000 1.0000000 0.6666667 0.5000000 0.5000000  
## Yes 0.0000000 0.0000000 0.0000000 0.3333333 0.5000000 0.5000000  
##   
## 37.5 37.6 37.8 38 38.1 38.2  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 38.3 38.4 38.5 38.6 38.7 38.8  
## No 0.5000000 1.0000000 1.0000000 1.0000000 0.5000000 1.0000000  
## Yes 0.5000000 0.0000000 0.0000000 0.0000000 0.5000000 0.0000000  
##   
## 38.9 39 39.1 39.2 39.4 39.5  
## No 1.0000000 1.0000000 1.0000000 0.7500000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.2500000 0.0000000 0.0000000  
##   
## 39.6 39.7 39.8 39.9 40 40.1  
## No 1.0000000 0.5000000 0.6666667 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.5000000 0.3333333 0.0000000 0.0000000 0.0000000  
##   
## 40.3 40.4 40.6 40.8 40.9 41  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 41.1 41.2 41.3 41.4 41.6 41.7  
## No 0.6666667 1.0000000 0.6000000 0.5000000 1.0000000 0.6666667  
## Yes 0.3333333 0.0000000 0.4000000 0.5000000 0.0000000 0.3333333  
##   
## 41.9 42 42.1 42.3 42.4 42.5  
## No 1.0000000 1.0000000 0.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000  
##   
## 42.6 42.8 43 43.1 43.2 43.3  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 43.4 43.5 43.6 43.7 43.8 44  
## No 1.0000000 1.0000000 0.5000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.5000000 0.0000000 0.0000000 0.0000000  
##   
## 44.1 44.2 44.3 44.4 44.5 44.6  
## No 0.7500000 1.0000000 1.0000000 0.0000000 1.0000000 1.0000000  
## Yes 0.2500000 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000  
##   
## 44.7 44.8 44.9 45 45.1 45.4  
## No 0.5000000 1.0000000 0.6666667 0.6000000 1.0000000 1.0000000  
## Yes 0.5000000 0.0000000 0.3333333 0.4000000 0.0000000 0.0000000  
##   
## 45.5 45.6 45.8 45.9 46 46.1  
## No 1.0000000 1.0000000 0.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000  
##   
## 46.2 46.3 46.4 46.5 46.6 46.7  
## No 1.0000000 1.0000000 1.0000000 0.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000  
##   
## 46.8 46.9 47 47.1 47.2 47.3  
## No 1.0000000 1.0000000 0.5000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.5000000 0.0000000 0.0000000 0.0000000  
##   
## 47.5 47.7 47.8 48 48.2 48.4  
## No 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667 0.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.3333333 1.0000000  
##   
## 48.5 48.7 48.8 48.9 49 49.3  
## No 0.6666667 1.0000000 0.5000000 1.0000000 1.0000000 0.0000000  
## Yes 0.3333333 0.0000000 0.5000000 0.0000000 0.0000000 1.0000000  
##   
## 49.9 50.1 50.2 50.5 50.6 50.9  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 51 51.1 51.2 51.3 51.4 51.7  
## No 1.0000000 1.0000000 1.0000000 1.0000000 0.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000  
##   
## 51.8 52.1 52.5 52.6 53 53.5  
## No 1.0000000 1.0000000 0.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000  
##   
## 53.8 53.9 54.1 54.4 54.5 54.8  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 54.9 55 55.7 56.4 56.5 56.8  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 57.5 58.5 59.4 61.4 61.6 63.4  
## No 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  
##   
## 65.1 67  
## No 1.0000000 1.0000000  
## Yes 0.0000000 0.0000000

Age doesn’t appear to affect whether or not the person will violate parole.

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

 Louisiana has higher violators.

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

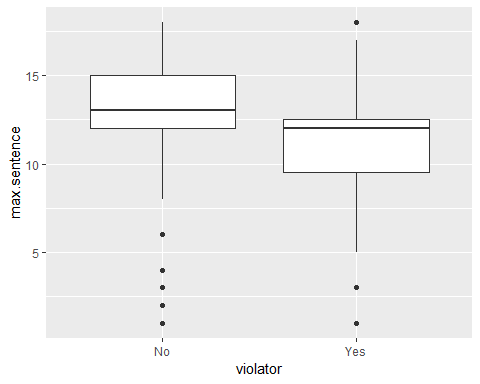
 Hard to tell. Will look at a table.

t1 = table(parole$violator,parole$time.served)  
prop.table(t1, margin = 2)

##   
## 0 0.1 0.2 0.3 0.5 0.7  
## No 1.00000000 0.50000000 1.00000000 0.00000000 1.00000000 0.50000000  
## Yes 0.00000000 0.50000000 0.00000000 1.00000000 0.00000000 0.50000000  
##   
## 0.8 0.9 1.1 1.2 1.3 1.4  
## No 0.33333333 1.00000000 0.75000000 1.00000000 0.50000000 0.00000000  
## Yes 0.66666667 0.00000000 0.25000000 0.00000000 0.50000000 1.00000000  
##   
## 1.5 1.6 1.7 1.8 1.9 2  
## No 1.00000000 0.00000000 0.33333333 1.00000000 0.50000000 0.66666667  
## Yes 0.00000000 1.00000000 0.66666667 0.00000000 0.50000000 0.33333333  
##   
## 2.1 2.2 2.3 2.4 2.5 2.6  
## No 1.00000000 0.66666667 1.00000000 0.66666667 1.00000000 1.00000000  
## Yes 0.00000000 0.33333333 0.00000000 0.33333333 0.00000000 0.00000000  
##   
## 2.7 2.8 2.9 3 3.1 3.2  
## No 0.66666667 1.00000000 1.00000000 0.96721311 1.00000000 0.89473684  
## Yes 0.33333333 0.00000000 0.00000000 0.03278689 0.00000000 0.10526316  
##   
## 3.3 3.4 3.5 3.6 3.7 3.8  
## No 1.00000000 0.75000000 1.00000000 1.00000000 0.82352941 0.93333333  
## Yes 0.00000000 0.25000000 0.00000000 0.00000000 0.17647059 0.06666667  
##   
## 3.9 4 4.1 4.2 4.3 4.4  
## No 0.83333333 0.84210526 0.94736842 0.85714286 1.00000000 0.89655172  
## Yes 0.16666667 0.15789474 0.05263158 0.14285714 0.00000000 0.10344828  
##   
## 4.5 4.6 4.7 4.8 4.9 5  
## No 0.85185185 0.93333333 0.95454545 0.93750000 0.68181818 0.71428571  
## Yes 0.14814815 0.06666667 0.04545455 0.06250000 0.31818182 0.28571429  
##   
## 5.1 5.2 5.3 5.4 5.5 5.6  
## No 0.96666667 0.87500000 0.92307692 0.92307692 1.00000000 0.81250000  
## Yes 0.03333333 0.12500000 0.07692308 0.07692308 0.00000000 0.18750000  
##   
## 5.7 5.8 5.9 6  
## No 0.91666667 0.92307692 0.88888889 0.84615385  
## Yes 0.08333333 0.07692308 0.11111111 0.15384615

Time served does appear to have some affect on violator.

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

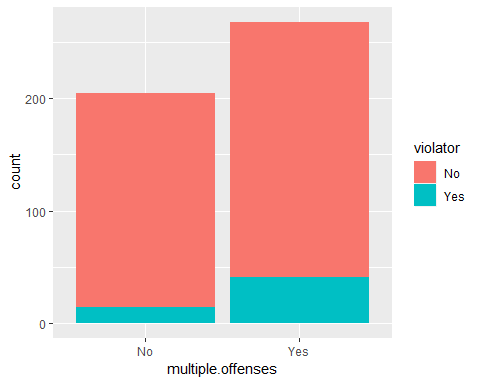
 I’m not a big fan of the box plot. To me it is hard to read.

t1 = table(parole$violator,parole$max.sentence)  
prop.table(t1, margin = 2)

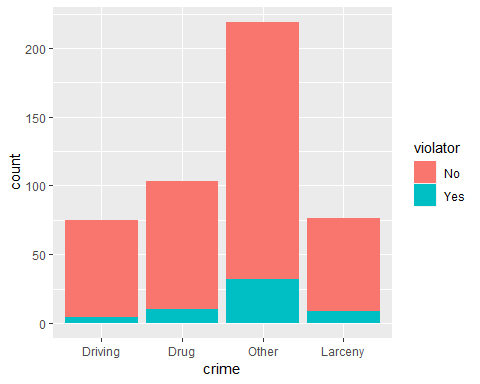
##   
## 1 2 3 4 5 6  
## No 0.75000000 1.00000000 0.33333333 1.00000000 0.00000000 0.56250000  
## Yes 0.25000000 0.00000000 0.66666667 0.00000000 1.00000000 0.43750000  
##   
## 8 9 10 11 12 13  
## No 0.85000000 0.66666667 0.66666667 0.60000000 0.86496350 0.96103896  
## Yes 0.15000000 0.33333333 0.33333333 0.40000000 0.13503650 0.03896104  
##   
## 14 15 16 17 18  
## No 0.95454545 0.98333333 0.88888889 0.93333333 0.89743590  
## Yes 0.04545455 0.01666667 0.11111111 0.06666667 0.10256410

So it appears, the longer max sentence, the less likely to violate parole.

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

 More violators with multiple offenses than not.

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

 If your crime is other more likely to violate parole.

## Task 3

Build a model around multiple offenses

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.offensesYes 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

Multiple offenses is significant (0.00549 is less than .05). The positive 0.9018 also implies that. The AIC is 335.5

## Task 4

Using forward stepwise, backward stepwise, or by manually building a model, create the best model you can to predict “violator”.

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   
## raceNotWhite 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.offensesYes 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

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

Race, Age, State = Virginia, and Multiple offenses are all significant predictors. AIC =256.16

Backward stepwise

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 \*\*\*  
## raceNotWhite 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.offensesYes 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

Age not significant in this model but race, stateVirginia and multiple offenses are. AIC=252.28 went down from 256.16.

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 3 264.58 272.58  
## + 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.offensesYes 1.77974 0.41476 4.291 1.78e-05 \*\*\*  
## raceNotWhite 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

Both models are the same so we will go with this model as our final model.

## Task 5

Create a logistic regression model using the training set to predict “violator” using the variables: state and race. (I predicted multiple offenseses in task #4) Comment on the quality of this model. Be sure to note which variables are signiﬁcant.

mod2 = glm(violator ~ state ,train, family = "binomial")  
summary(mod2)

##   
## Call:  
## glm(formula = violator ~ state, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.1106 -0.4084 -0.2255 -0.2255 2.7147   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.7677 0.2892 -6.113 9.79e-10 \*\*\*  
## stateKentucky -0.6747 0.5146 -1.311 0.189803   
## stateLouisiana 1.6086 0.3841 4.188 2.81e-05 \*\*\*  
## stateVirginia -1.8916 0.5046 -3.749 0.000178 \*\*\*  
## ---  
## 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: 264.58 on 469 degrees of freedom  
## AIC: 272.58  
##   
## Number of Fisher Scoring iterations: 6

StateVirginia is significant.

mod3 = glm(violator ~ race ,train, family = "binomial")  
summary(mod3)

##   
## Call:  
## glm(formula = violator ~ race, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5742 -0.5742 -0.4323 -0.4323 2.1985   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.3232 0.2139 -10.862 <2e-16 \*\*\*  
## raceNotWhite 0.6039 0.2895 2.086 0.037 \*   
## ---  
## 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: 335.64 on 471 degrees of freedom  
## AIC: 339.64  
##   
## Number of Fisher Scoring iterations: 5

raceNotwhite is significant.

## Task 6

What is the predicted probability of parole violation of the two following parolees? Parolee1: Louisiana with multiple oﬀenses and white race Parolee2: Kentucky with no multiple oﬀenses and other race # not sure what is wrong with code below

#newdata=data.frame(state = "Louisiana", multiple.offenses = "Yes", race = "White")  
#predict(fowardmod, newdata, type = "response")

## not sure what is wrong below

#newdata = data.frame(state="Kentucky", multiple.offenses = "No", race= "NotWhite")  
#predict(fowardmod, newdata, type = "response")

## Task 7

Develop an ROC curve and determine the probability threshold that best balances speciﬁcity and sensitivity (on the training set). Hint: In the predict function, use type = “response” and do not use the [,2] that we used in the logistic regression threshold lecture. We only had to include that code in that lecture because we used k-fold cross validation

(all variables seems to be the best).

library(e1071)  
ctrl = trainControl(method = "cv",number = 10) #set up caret 10 fold cross validation  
  
set.seed(123) #set random number seed for cross validation  
modkFold = train(violator ~., parole, method = "glm", trControl = ctrl)  
summary(modkFold)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6633 -0.4123 -0.2574 -0.1589 2.8738   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.182313 1.041740 -3.055 0.00225 \*\*   
## maleFemale -0.270624 0.370506 -0.730 0.46513   
## raceNotWhite 0.757252 0.324581 2.333 0.01965 \*   
## age 0.006554 0.013724 0.478 0.63297   
## stateKentucky 0.208399 0.417528 0.499 0.61769   
## stateLouisiana 0.893812 0.447042 1.999 0.04557 \*   
## stateVirginia -3.280842 0.526952 -6.226 4.78e-10 \*\*\*  
## time.served -0.076548 0.099531 -0.769 0.44184   
## max.sentence 0.053293 0.043824 1.216 0.22396   
## multiple.offensesYes 1.531547 0.325794 4.701 2.59e-06 \*\*\*  
## crimeDrug -0.123182 0.542514 -0.227 0.82038   
## crimeOther 0.157812 0.484705 0.326 0.74474   
## crimeLarceny 0.494705 0.584656 0.846 0.39747   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 483.27 on 674 degrees of freedom  
## Residual deviance: 348.68 on 662 degrees of freedom  
## AIC: 374.68  
##   
## Number of Fisher Scoring iterations: 6

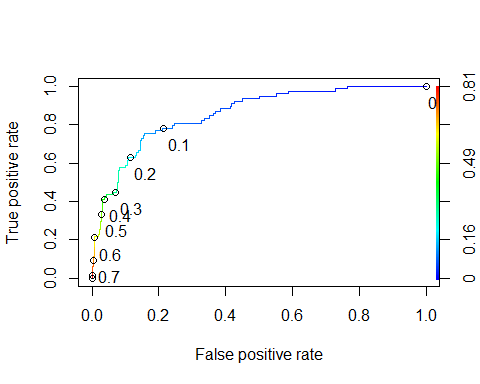
Develop predictions

predictions = predict(modkFold, type="prob")[,2] #develop predicted probabilities  
head(predictions)

## [1] 0.08117685 0.04353093 0.10488783 0.08668970 0.13016193 0.16534133

Threshold selection

#Change this next line to the names of your predictions and the response variable in the training data frame  
ROCRpred = prediction(predictions, parole$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 What is the accuracy, sensitivity, and speciﬁcity of the model on the training set given the cutoﬀ from Task 7? What are the implications of incorrectly classifying a parolee?

If choose threshold of 0.1 the sensitivity is around 0.79 and a 1 minus specivicity of around 0.2.

Area under the curve (AUC). AUC is a measure of the strength of the model. Values closer to 1 are better. Can be used to compare models.

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

## [1] 0.8572564

#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 0.7564103  
## specificity 0.8442211  
## cutoff 0.1302496

Test thresholds to evaluate accuracy

#confusion matrix  
t1 = table(parole$violator,predictions > 0.1302496)  
t1

##   
## FALSE TRUE  
## No 504 93  
## Yes 19 59

Calculate accuracy

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

## [1] 0.8340741

## Task 9

Apply trial and error to maximize accuracy (here trying 0.5 as threshold)

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

##   
## FALSE TRUE  
## No 581 16  
## Yes 52 26

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

## [1] 0.8992593

Threshold = 0.6

t1 = table(parole$violator,predictions > 0.6)  
t1

##   
## FALSE TRUE  
## No 592 5  
## Yes 62 16

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

## [1] 0.9007407

# Task 10

There are more people that do not violate parole than do. Below is a naive prediction (everyone does not violate parole)

t1 = table(parole$violator,predictions > 1) #set threshold to 1 so all are classifed as not delinquent  
t1

##   
## FALSE  
## No 597  
## Yes 78

(t1[1])/nrow(parole)

## [1] 0.8844444