## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

## ── Attaching packages ────────────────────────────────────── tidymodels 0.1.2 ──

## ✓ broom 0.7.3 ✓ recipes 0.1.15  
## ✓ dials 0.0.9 ✓ rsample 0.0.8   
## ✓ infer 0.5.4 ✓ tune 0.1.2   
## ✓ modeldata 0.1.0 ✓ workflows 0.2.1   
## ✓ parsnip 0.1.5 ✓ yardstick 0.0.7

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

##   
## Attaching package: 'e1071'

## The following object is masked from 'package:tune':  
##   
## tune

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.1

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

parole <- read\_csv("~/Documents/MBA\_Predictive/Week 3/parole.csv")

##   
## ── 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,"white" = "1" , "Other" = "2"))  
parole = parole %>% mutate(age= as\_factor(age))  
parole = parole %>% mutate(state = as\_factor(state)) %>%   
 mutate(state = fct\_recode(state,"Kentucky" = "2", "Louisiana"= "3", "Virginia" = "4", "other"= "1"))  
parole = parole %>% mutate(crime = as\_factor(crime))%>%  
 mutate(crime = fct\_recode(crime, "other"= "1", "larceny" = "2", "drug-related" = "3","driving-related"= "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" ))

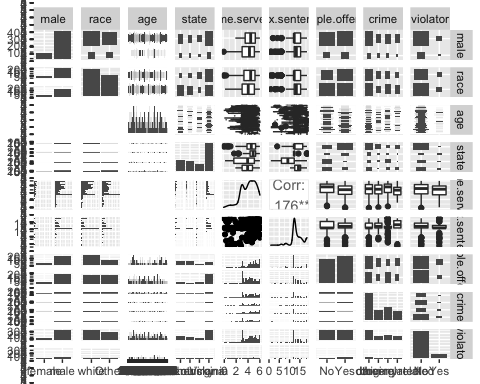
# Task 1

set.seed(12345)  
parole\_split = initial\_split(parole, prob = 0.70, strata = violator)  
train = training(parole\_split)  
test = testing(parole\_split)

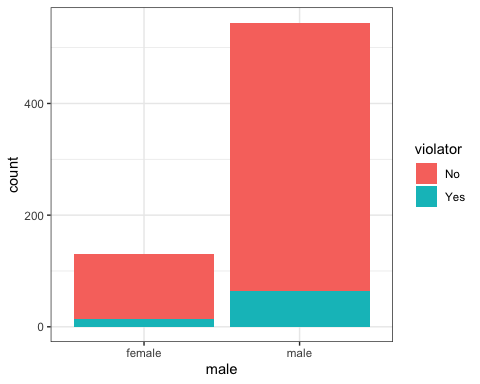
# Task 2

ggpairs(train, cardinality\_threshold = 302)

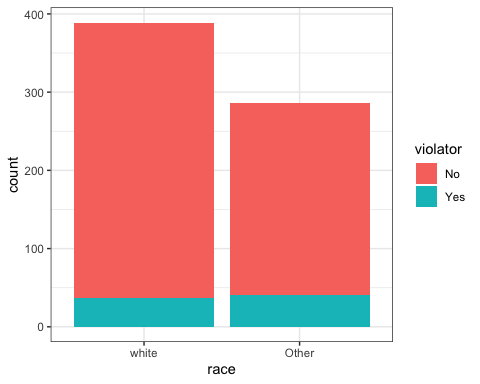
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()`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



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



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



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

##   
## female male  
## No 0.8923077 0.8825688  
## Yes 0.1076923 0.1174312

I started with ggpairs to find the highest correlation to “violators”. I came accross the error of the cardinality threshold being too small. I had to use the code “cardinality\_threshold = 302)” to fix the error. After reviewing the ggpairs matrix, I found that “male” and “violator” was most correlated. I then created a box plot to visualize the data. I found that males are more likely to break their parole, and is the greatest predictor. I also created ti which shows the data in a tabulated form.

# Task 3

parole\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
parole\_recipe = recipe(violator ~ male, parole)   
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe) %>%   
 add\_model(parole\_model)  
  
parole\_fit = fit(logreg\_wf, parole)

summary(parole\_fit$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.4998 -0.4998 -0.4998 -0.4774 2.1111   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.11453 0.28292 -7.474 7.78e-14 \*\*\*  
## malemale 0.09755 0.31265 0.312 0.755   
## ---  
## 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: 483.17 on 673 degrees of freedom  
## AIC: 487.17  
##   
## Number of Fisher Scoring iterations: 4

Male has two levels representing each of the two classes (Male and Female). Note that the female coefficeint is negative. This suggests that probability of a female violating parole is lower than a male.

The AIC is 487.17. I will use this value to compare this model to others. Smaller AIC is better.

# Task 4

parole\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
parole\_recipe = recipe(violator ~ male+race+state+multiple.offenses, parole)  
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe) %>%   
 add\_model(parole\_model)  
  
parole\_fit = fit(logreg\_wf, parole)

summary(parole\_fit$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4252 -0.4121 -0.2655 -0.1844 2.8576   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.72593 0.43253 -6.302 2.93e-10 \*\*\*  
## malemale 0.25945 0.35599 0.729 0.4661   
## raceOther 0.73836 0.31819 2.320 0.0203 \*   
## stateKentucky 0.08509 0.39807 0.214 0.8307   
## stateLouisiana 0.77045 0.39291 1.961 0.0499 \*   
## stateVirginia -3.12308 0.51096 -6.112 9.83e-10 \*\*\*  
## multiple.offensesYes 1.52366 0.32132 4.742 2.12e-06 \*\*\*  
## ---  
## 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: 352.71 on 668 degrees of freedom  
## AIC: 366.71  
##   
## Number of Fisher Scoring iterations: 6

First I added race, and the AIC went down to 485. I added state and the AIC went down to 388.16. I added crime but this time the AIC went up, which means that the model was getting worse. I removed crime from the model. I added age and the model went to 730.66, so I took age back out. I added multiple offenses, and the AIC went down to 366.71, which is better. The variables that are significant are multiple.offenses, Louisiana, Virginia, and race\_other.

# Task 5

parole\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
parole\_recipe = recipe(violator ~ race+state+multiple.offenses, parole)  
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe) %>%   
 add\_model(parole\_model)  
  
parole\_fit = fit(logreg\_wf, parole)

summary(parole\_fit$fit$fit$fit)

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4012 -0.4051 -0.2604 -0.1801 2.8739   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.50359 0.30055 -8.330 < 2e-16 \*\*\*  
## raceOther 0.74594 0.31828 2.344 0.0191 \*   
## stateKentucky 0.04449 0.39449 0.113 0.9102   
## stateLouisiana 0.75016 0.39147 1.916 0.0553 .   
## stateVirginia -3.12945 0.51147 -6.119 9.44e-10 \*\*\*  
## multiple.offensesYes 1.51964 0.32027 4.745 2.09e-06 \*\*\*  
## ---  
## 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: 353.26 on 669 degrees of freedom  
## AIC: 365.26  
##   
## Number of Fisher Scoring iterations: 6

The AIC of this model is 365.26, which is the lowest it has been through all of the models. This suggests that this is a good model. All variables are significant except for Kentucky. Louisiana is at .0553, which could still be considered significant.

# Task 6

predictions = predict(parole\_fit, parole, type="prob")  
head(predictions)

## # A tibble: 6 x 2  
## .pred\_No .pred\_Yes  
## <dbl> <dbl>  
## 1 0.924 0.0756  
## 2 0.924 0.0756  
## 3 0.853 0.147   
## 4 0.924 0.0756  
## 5 0.853 0.147   
## 6 0.853 0.147

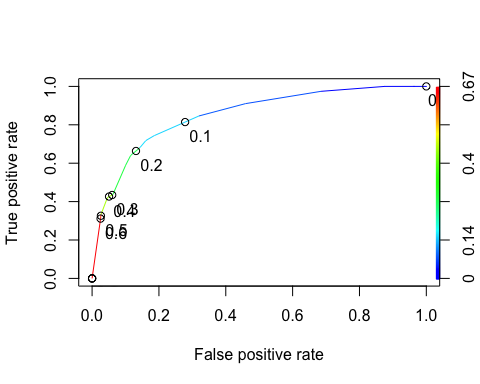
predictions = predict(parole\_fit, parole, type="prob")[2]  
head(predictions)

## # A tibble: 6 x 1  
## .pred\_Yes  
## <dbl>  
## 1 0.0756  
## 2 0.0756  
## 3 0.147   
## 4 0.0756  
## 5 0.147   
## 6 0.147

The prediction rate is 51.68% that Parolee1: Louisiana with multiple offenses and white race Parolee2: Kentucky with no multiple offenses and other race will have a parole violation.

# Task 7

ROCRpred = prediction(predictions, parole$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] 0.8506206

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 0.7435897  
## specificity 0.8140704  
## cutoff 0.1470858

# Task 8

t1 = table(parole$violator,predictions > 0.1470858)  
t1

##   
## FALSE TRUE  
## No 501 96  
## Yes 22 56

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

## [1] 0.8251852

56/(22+56) #Sensitivity

## [1] 0.7179487

501/(501+96) #Specificity

## [1] 0.839196

One implication of incorrectly classifying a parolee is that someone who you assume will not violate parole could have less restraint and do something to bad to harm someone or themselves. If someone is incorrectly classified that they will likely violate parole, but they do not, the parole officer could spend too much time supervising them when they could be supervising someone who actually needs it.

# Task 9

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

##   
## FALSE TRUE  
## No 582 15  
## Yes 54 24

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

## [1] 0.8977778

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

##   
## FALSE TRUE  
## No 582 15  
## Yes 54 24

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

## [1] 0.8977778

There is not a difference between 0.5 and 0.6.0.6 best maximizes the accuracy on the training set. I also tried 0.4 and 0.3, but the accuracy was worse.

# Task 10

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

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

(t1[1])/nrow(parole)

## [1] 0.8844444

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

## [1] 0.8844444