parole <- read\_csv("~/Ban502/Module 3/parole.csv")

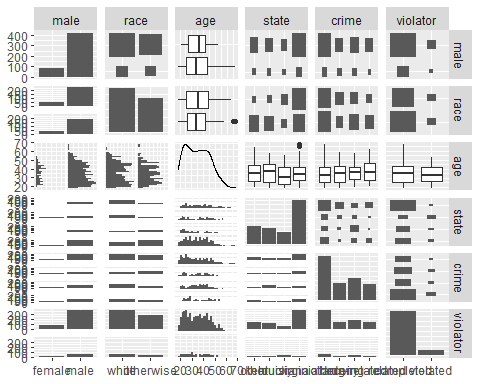
##   
## -- Column specification --------------------------------------------------------  
## cols(  
## male = col\_double(),  
## race = col\_double(),  
## age = col\_double(),  
## state = col\_double(),  
## time.served = col\_double(),  
## max.sentence = col\_double(),  
## multiple.offenses = col\_double(),  
## crime = col\_double(),  
## violator = col\_double()  
## )

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

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

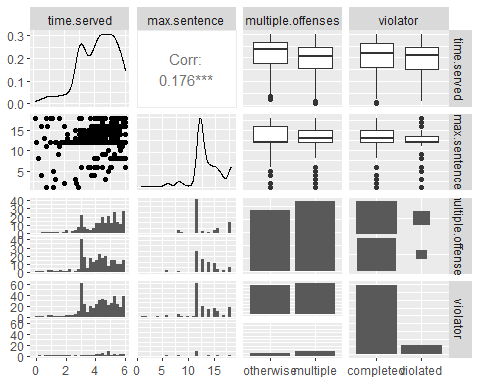
ggpairs(train, columns = c("male", "race","age", "state","crime", "violator"))

## `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`.

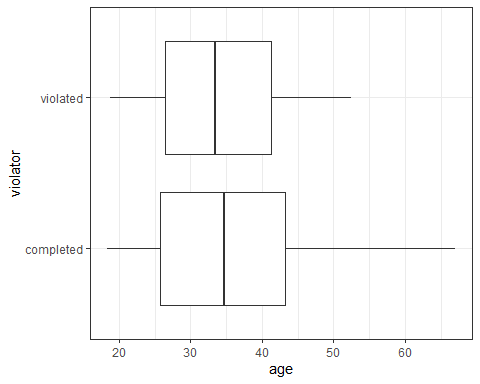


ggpairs(train, columns = c("time.served", "max.sentence","multiple.offenses", "violator"))

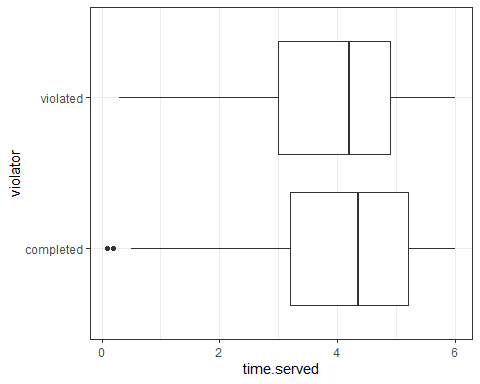
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



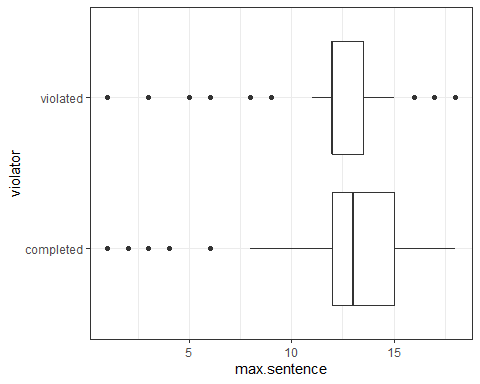
ggplot(train,aes(x=age,y=violator)) + geom\_boxplot() + theme\_bw()



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



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



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

##   
## female male  
## completed 0.8850575 0.8833333  
## violated 0.1149425 0.1166667

t1 = table(train$violator, train$race)  
prop.table(t1, margin = 2 )

##   
## white otherwise  
## completed 0.8951049 0.8687783  
## violated 0.1048951 0.1312217

t1 = table(train$violator, train$state)  
prop.table(t1, margin = 2 )

##   
## other kentucky louisiana virginia  
## completed 0.8640777 0.8522727 0.6060606 0.9760000  
## violated 0.1359223 0.1477273 0.3939394 0.0240000

t1 = table(train$violator, train$multiple.offenses)  
prop.table(t1, margin = 2 )

##   
## otherwise multiple  
## completed 0.91983122 0.85185185  
## violated 0.08016878 0.14814815

t1 = table(train$violator, train$crime)  
prop.table(t1, margin = 2 )

##   
## other larceny drug-related drving-related  
## completed 0.87280702 0.87356322 0.87610619 0.93670886  
## violated 0.12719298 0.12643678 0.12389381 0.06329114

## Task 2 response:

The box plots do now show any major differences in medians. The tabular data shows large differences between levels of state and multiple.offenses.

parole\_model =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
parole\_recipe = recipe(violator ~ state, train) %>%   
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe) %>%   
 add\_model(parole\_model)  
  
parole\_fit = fit(logreg\_wf, train)

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

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0008 -0.5405 -0.2204 -0.2204 2.7312   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.84958 0.28751 -6.433 1.25e-10 \*\*\*  
## state\_kentucky 0.09704 0.41584 0.233 0.815481   
## state\_louisiana 1.41880 0.38226 3.712 0.000206 \*\*\*  
## state\_virginia -1.85583 0.50341 -3.686 0.000227 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 300.70 on 503 degrees of freedom  
## AIC: 308.7  
##   
## Number of Fisher Scoring iterations: 6

## Task 3 Response:

Although the AIC appears acceptable, it doesn’t tell us much at this stage, as there is nothing to compare it to yet. We see that state appears significant though.

parole\_model2 =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
parole\_recipe2 = recipe(violator ~ crime, train) %>%   
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe2) %>%   
 add\_model(parole\_model2)  
  
parole\_fit = fit(logreg\_wf, train)

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

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5216 -0.5216 -0.5200 -0.3616 2.3495   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.926009 0.198766 -9.690 <2e-16 \*\*\*  
## crime\_larceny -0.006829 0.378913 -0.018 0.986   
## crime\_drug.related -0.030054 0.347904 -0.086 0.931   
## crime\_drving.related -0.768618 0.503010 -1.528 0.127   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 361.73 on 503 degrees of freedom  
## AIC: 369.73  
##   
## Number of Fisher Scoring iterations: 5

## Task 4 Response:

After running all variables through the log regression, we get the following AIC scores/significance levels:  
male: 368.66 – not significant  
race: 367.83 – not significant  
age: 368.28 – not significant  
state: 308.7 – significant  
time.served: 364.55 – somewhat significant  
max.sentence: 358.29 – significant  
multiple.offenses: 362.85 – somewhat significant  
crime: 369.73 – not significant

According to this model, State is the best indicator of whether or not parole will be violated. Other significant variables include max sentence, time served, and multiple offenses.

parole\_model3 =   
 logistic\_reg() %>%   
 set\_engine("glm")   
  
parole\_recipe3 = recipe(violator ~ state + multiple.offenses + race, train) %>%   
 step\_dummy(all\_nominal(), -all\_outcomes())   
  
logreg\_wf = workflow() %>%  
 add\_recipe(parole\_recipe3) %>%   
 add\_model(parole\_model3)  
  
parole\_fit3 = fit(logreg\_wf, train)

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

##   
## Call:  
## stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5216 -0.5216 -0.5200 -0.3616 2.3495   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.926009 0.198766 -9.690 <2e-16 \*\*\*  
## crime\_larceny -0.006829 0.378913 -0.018 0.986   
## crime\_drug.related -0.030054 0.347904 -0.086 0.931   
## crime\_drving.related -0.768618 0.503010 -1.528 0.127   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 364.67 on 506 degrees of freedom  
## Residual deviance: 361.73 on 503 degrees of freedom  
## AIC: 369.73  
##   
## Number of Fisher Scoring iterations: 5

## Task 5 response:

With this combination of variables, we see the AIC has decreased, therefore, this model would be considered an improvement to prior models. State and multiple offenses are both showing as significant while race is not.

newdata = data.frame(state = "louisiana", multiple.offenses = "multiple", race = "white")  
predict(parole\_fit3, newdata, type="prob")

## # A tibble: 1 x 2  
## .pred\_completed .pred\_violated  
## <dbl> <dbl>  
## 1 0.557 0.443

newdata2 = data.frame(state = "kentucky", multiple.offenses = "otherwise", race = "otherwise")  
predict(parole\_fit3, newdata2, type="prob")

## # A tibble: 1 x 2  
## .pred\_completed .pred\_violated  
## <dbl> <dbl>  
## 1 0.848 0.152

## Task 6 Response:

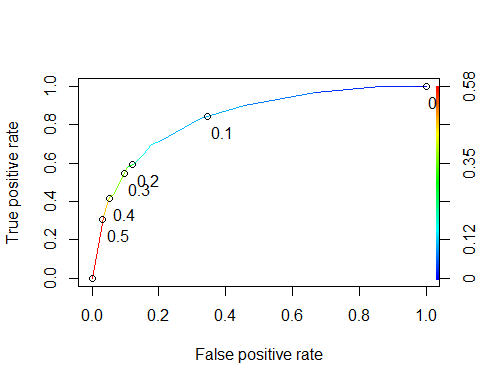
Parole 1’s prediction to violate is 44.28%

Parole 2’s prediction to violate is 15.2%

predictions = predict(parole\_fit3, train, type="prob")[2]  
head(predictions)

## # A tibble: 6 x 1  
## .pred\_violated  
## <dbl>  
## 1 0.0729  
## 2 0.0729  
## 3 0.0729  
## 4 0.107   
## 5 0.107   
## 6 0.0729

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))



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

## [1] 0.834916

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 0.7118644  
## specificity 0.7968750  
## cutoff 0.1070172

## Task 8 Response:

IT = table(train$violator,predictions > 0.1070172)  
IT

##   
## FALSE TRUE  
## completed 368 80  
## violated 18 41

Ideal cut off: 0.1070172  
sensitivity 0.7118644  
specificity 0.7968750  
accuracy:

(368+41)/(368+80+18+41)

## [1] 0.8067061

The implication of incorrectly classifying a parolee means risk of improper levels of supervision, which can either mean an under-supervised likely-violator, which can be unsafe, or an over-supervised unlikely-violator, which is inefficient.

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

##   
## FALSE TRUE  
## completed 434 14  
## violated 42 17

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

## [1] 0.8895464

## Task 10: applying to test data:

fitted.results <- predict(parole\_fit3,test)  
fitted.results <- ifelse(fitted.results > 0.5,1,0)

## Warning in Ops.factor(left, right): '>' not meaningful for factors

misClasificError <- mean(fitted.results != test$violator)  
print(paste('Accuracy',1-misClasificError))

## [1] "Accuracy NA"

"Accuracy 0.842696629213483"

## [1] "Accuracy 0.842696629213483"