library("tidyverse")

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

## ✓ ggplot2 3.2.1 ✓ purrr 0.3.3  
## ✓ tibble 2.1.3 ✓ dplyr 0.8.4  
## ✓ tidyr 1.0.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ 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

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, "female" = "0", "male" = "1"))

parole = parole %>% mutate(race = as.factor(as.character(race))) %>%  
 mutate(race = fct\_recode(race, "white" = "1", "nonwhite" = "2"))

parole = parole %>% mutate(state = as.factor(as.character(state))) %>%  
 mutate(state = fct\_recode(state, "OtherState" = "1", "Kentucky" = "2", "Louisiana" = "3", "Virginia" = "4"))

parole = parole %>% mutate(crime = as.factor(as.character(crime))) %>%  
 mutate(crime = fct\_recode(crime, "OtherCrime" = "1", "larceny" = "2", "drugs" = "3", "driving" = "4"))

parole = parole %>% mutate(multiple.offenses = as.factor(as.character(multiple.offenses))) %>%  
 mutate(multiple.offenses = fct\_recode(multiple.offenses, "SingleOffense" = "0", "MultiOffenses" = "1"))

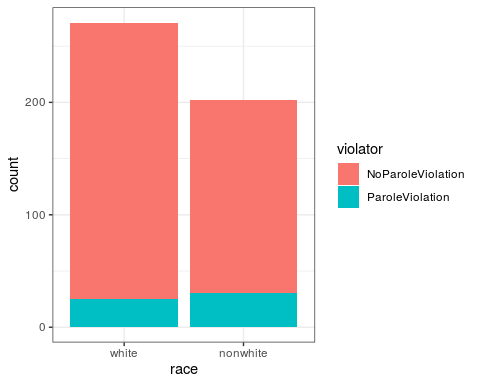
parole = parole %>% mutate(violator = as.factor(as.character(violator))) %>%  
 mutate(violator = fct\_recode(violator, "NoParoleViolation" = "0", "ParoleViolation" = "1"))

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

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



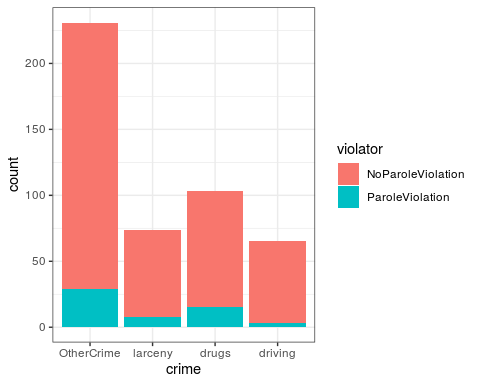
ggplot(train, aes(x=race, fill = violator))+ geom\_bar() + theme\_bw()



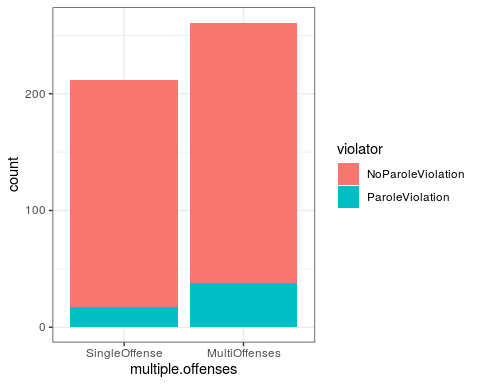
ggplot(train, aes(x=state, fill = violator))+ geom\_bar() + theme\_bw()



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



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



Task 2 - i decided based on the lecture to use the bar plot from the beginning it seemed to show the best correlation between two variables.

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   
## -0.5610 -0.5610 -0.4089 -0.4089 2.2465   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.4398 0.2529 -9.648 <2e-16 \*\*\*  
## multiple.offensesMultiOffenses 0.6702 0.3078 2.177 0.0295 \*   
## ---  
## 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: 335.02 on 471 degrees of freedom  
## AIC: 339.02  
##   
## Number of Fisher Scoring iterations: 5

Task 3 - I chose multiple.offenses as the variable that would be the most predictive of “violator”. The multiple offenses would be most likely to violate parole as shown in the estimate being over 0.5. Like it was spoken about in the lecture, I cannot use AIC on it’s own but as a comparison.

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

##   
## Call:  
## glm(formula = violator ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6055 -0.3932 -0.2643 -0.1384 2.9470   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.885777 1.197993 -2.409 0.01600 \*   
## malemale -0.137577 0.411340 -0.334 0.73803   
## racenonwhite 1.143719 0.403890 2.832 0.00463 \*\*   
## age 0.005279 0.016910 0.312 0.75490   
## stateKentucky 0.124282 0.492370 0.252 0.80072   
## stateLouisiana 0.217202 0.556154 0.391 0.69614   
## stateVirginia -3.801561 0.666733 -5.702 1.19e-08 \*\*\*  
## time.served -0.109344 0.118901 -0.920 0.35777   
## max.sentence 0.065956 0.054593 1.208 0.22700   
## multiple.offensesMultiOffenses 1.711032 0.396463 4.316 1.59e-05 \*\*\*  
## crimelarceny 0.392910 0.514075 0.764 0.44469   
## crimedrugs -0.210563 0.413351 -0.509 0.61047   
## crimedriving -0.727043 0.690775 -1.053 0.29257   
## ---  
## 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: 242.09 on 460 degrees of freedom  
## AIC: 268.09  
##   
## 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

forwardmod = stepAIC(emptymod, direction = "forward", scope = list(upper=allmod, lower=emptymod), trace = TRUE)

## Start: AIC=342.04  
## violator ~ 1  
##   
## Df Deviance AIC  
## + state 3 275.18 283.18  
## + max.sentence 1 331.01 335.01  
## + multiple.offenses 1 335.02 339.02  
## + race 1 336.51 340.51  
## + time.served 1 336.61 340.61  
## <none> 340.04 342.04  
## + crime 3 335.07 343.07  
## + male 1 339.72 343.72  
## + age 1 339.95 343.95  
##   
## Step: AIC=283.18  
## violator ~ state  
##   
## Df Deviance AIC  
## + multiple.offenses 1 254.96 264.96  
## + race 1 267.66 277.66  
## <none> 275.18 283.18  
## + max.sentence 1 274.27 284.27  
## + time.served 1 274.44 284.44  
## + age 1 275.11 285.11  
## + male 1 275.13 285.13  
## + crime 3 271.72 285.72  
##   
## Step: AIC=264.96  
## violator ~ state + multiple.offenses  
##   
## Df Deviance AIC  
## + race 1 246.98 258.98  
## <none> 254.96 264.96  
## + max.sentence 1 253.11 265.11  
## + time.served 1 254.47 266.47  
## + male 1 254.91 266.91  
## + age 1 254.94 266.94  
## + crime 3 252.75 268.75  
##   
## Step: AIC=258.98  
## violator ~ state + multiple.offenses + race  
##   
## Df Deviance AIC  
## <none> 246.98 258.98  
## + max.sentence 1 245.31 259.31  
## + time.served 1 246.33 260.33  
## + male 1 246.78 260.78  
## + age 1 246.98 260.98  
## + crime 3 244.78 262.79

summary(forwardmod)

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesMultiOffenses 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## racenonwhite 1.09382 0.38974 2.807 0.00501 \*\*   
## ---  
## 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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

Task 4 - based on the forward stepwise this is the best model i can use to predict “violator”. The variables that seem to be significant are multiple.offenses, race and state. However the best variable is multiple.offenses as a predictor for whether an offender will violate parole or not. The AICs are very similar but multiple.offenses as the smallest based on the forward stepwise.

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

##   
## Call:  
## glm(formula = violator ~ state + multiple.offenses + race, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3609 -0.4094 -0.2705 -0.1575 2.9653   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.51087 0.36354 -6.907 4.96e-12 \*\*\*  
## stateKentucky 0.07372 0.46051 0.160 0.87282   
## stateLouisiana 0.10381 0.50018 0.208 0.83559   
## stateVirginia -3.60795 0.63788 -5.656 1.55e-08 \*\*\*  
## multiple.offensesMultiOffenses 1.73482 0.39421 4.401 1.08e-05 \*\*\*  
## racenonwhite 1.09382 0.38974 2.807 0.00501 \*\*   
## ---  
## 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: 246.98 on 467 degrees of freedom  
## AIC: 258.98  
##   
## Number of Fisher Scoring iterations: 6

Task 5 - this model shows if you are from Kentucky and Virginia your chances of violating parole are negative meaning you have a less chance. Also if you are from Viriginia your chances of violating parole are higher than in Kentucky and Virginia based on the estimate being positive for Louisiana and negative for Kentucky and Virginia. Also you have a higher chance to violate parole if you have multiple offenses as well as a slightly higher chance if your race is nonwhite based on the pr value being smaller than .05.

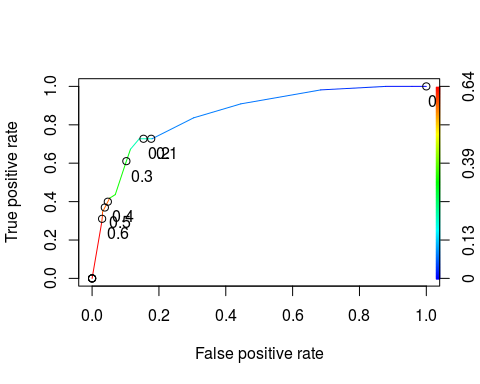
The AIC number has gotten lower than previous models so this means we are getting better at the models.

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

## 1 2 3 4 5 6   
## 0.075099778 0.195125037 0.195125037 0.075099778 0.075099778 0.195125037   
## 7 8 9 10 11 12   
## 0.075099778 0.075099778 0.075099778 0.075099778 0.195125037 0.075099778   
## 13 14 15 16 17 18   
## 0.075099778 0.075099778 0.075099778 0.195125037 0.075099778 0.195125037   
## 19 20 21 22 23 24   
## 0.075099778 0.195125037 0.075099778 0.075099778 0.075099778 0.075099778   
## 25 26 27 28 29 30   
## 0.075099778 0.075099778 0.075099778 0.075099778 0.315173008 0.315173008   
## 31 32 33 34 35 36   
## 0.315173008 0.578782484 0.315173008 0.315173008 0.315173008 0.315173008   
## 37 38 39 40 41 42   
## 0.315173008 0.315173008 0.315173008 0.315173008 0.315173008 0.315173008   
## 43 44 45 46 47 48   
## 0.315173008 0.315173008 0.315173008 0.080383167 0.080383167 0.080383167   
## 49 50 51 52 53 54   
## 0.080383167 0.206962905 0.596642134 0.080383167 0.080383167 0.080383167   
## 55 56 57 58 59 60   
## 0.080383167 0.206962905 0.206962905 0.080383167 0.080383167 0.080383167   
## 61 62 63 64 65 66   
## 0.206962905 0.331296156 0.331296156 0.080383167 0.080383167 0.331296156   
## 67 68 69 70 71 72   
## 0.080383167 0.080383167 0.206962905 0.080383167 0.080383167 0.080383167   
## 73 74 75 76 77 78   
## 0.080383167 0.080383167 0.206962905 0.080383167 0.080383167 0.080383167   
## 79 80 81 82 83 84   
## 0.080383167 0.080383167 0.080383167 0.080383167 0.080383167 0.596642134   
## 85 86 87 88 89 90   
## 0.331296156 0.080383167 0.080383167 0.206962905 0.206962905 0.206962905   
## 91 92 93 94 95 96   
## 0.206962905 0.080383167 0.080383167 0.080383167 0.596642134 0.331296156   
## 97 98 99 100 101 102   
## 0.080383167 0.331296156 0.080383167 0.080383167 0.080383167 0.206962905   
## 103 104 105 106 107 108   
## 0.080383167 0.080383167 0.080383167 0.080383167 0.206962905 0.080383167   
## 109 110 111 112 113 114   
## 0.331296156 0.080383167 0.080383167 0.080383167 0.080383167 0.080383167   
## 115 116 117 118 119 120   
## 0.331296156 0.080383167 0.080383167 0.080383167 0.080383167 0.080383167   
## 121 122 123 124 125 126   
## 0.080383167 0.206962905 0.206962905 0.080383167 0.080383167 0.080383167   
## 127 128 129 130 131 132   
## 0.080383167 0.080383167 0.082635845 0.603862386 0.337996072 0.603862386   
## 133 134 135 136 137 138   
## 0.603862386 0.603862386 0.337996072 0.603862386 0.603862386 0.603862386   
## 139 140 141 142 143 144   
## 0.603862386 0.603862386 0.337996072 0.603862386 0.603862386 0.337996072   
## 145 146 147 148 149 150   
## 0.603862386 0.337996072 0.603862386 0.603862386 0.337996072 0.603862386   
## 151 152 153 154 155 156   
## 0.603862386 0.337996072 0.211945335 0.337996072 0.337996072 0.603862386   
## 157 158 159 160 161 162   
## 0.603862386 0.337996072 0.211945335 0.603862386 0.337996072 0.337996072   
## 163 164 165 166 167 168   
## 0.603862386 0.337996072 0.211945335 0.603862386 0.211945335 0.603862386   
## 169 170 171 172 173 174   
## 0.603862386 0.603862386 0.603862386 0.211945335 0.603862386 0.603862386   
## 175 176 177 178 179 180   
## 0.082635845 0.603862386 0.082635845 0.211945335 0.211945335 0.082635845   
## 181 182 183 184 185 186   
## 0.211945335 0.082635845 0.082635845 0.211945335 0.211945335 0.082635845   
## 187 188 189 190 191 192   
## 0.075099778 0.075099778 0.195125037 0.075099778 0.075099778 0.075099778   
## 193 194 195 196 197 198   
## 0.075099778 0.075099778 0.315173008 0.195125037 0.315173008 0.315173008   
## 199 200 201 202 203 204   
## 0.075099778 0.075099778 0.075099778 0.075099778 0.315173008 0.075099778   
## 205 206 207 208 209 210   
## 0.075099778 0.315173008 0.075099778 0.075099778 0.075099778 0.075099778   
## 211 212 213 214 215 216   
## 0.075099778 0.075099778 0.075099778 0.075099778 0.075099778 0.075099778   
## 217 218 219 220 221 222   
## 0.075099778 0.075099778 0.075099778 0.315173008 0.075099778 0.075099778   
## 223 224 225 226 227 228   
## 0.075099778 0.075099778 0.075099778 0.075099778 0.075099778 0.075099778   
## 229 230 231 232 233 234   
## 0.075099778 0.075099778 0.075099778 0.075099778 0.075099778 0.035909972   
## 235 236 237 238 239 240   
## 0.012321743 0.012321743 0.012321743 0.002196228 0.012321743 0.002196228   
## 241 242 243 244 245 246   
## 0.012321743 0.035909972 0.035909972 0.012321743 0.035909972 0.002196228   
## 247 248 249 250 251 252   
## 0.035909972 0.035909972 0.006528726 0.012321743 0.035909972 0.002196228   
## 253 254 255 256 257 258   
## 0.035909972 0.035909972 0.035909972 0.012321743 0.035909972 0.035909972   
## 259 260 261 262 263 264   
## 0.012321743 0.035909972 0.012321743 0.035909972 0.012321743 0.035909972   
## 265 266 267 268 269 270   
## 0.006528726 0.012321743 0.012321743 0.006528726 0.035909972 0.006528726   
## 271 272 273 274 275 276   
## 0.006528726 0.012321743 0.002196228 0.035909972 0.035909972 0.035909972   
## 277 278 279 280 281 282   
## 0.035909972 0.006528726 0.012321743 0.035909972 0.035909972 0.035909972   
## 283 284 285 286 287 288   
## 0.006528726 0.035909972 0.035909972 0.006528726 0.035909972 0.012321743   
## 289 290 291 292 293 294   
## 0.012321743 0.035909972 0.002196228 0.035909972 0.006528726 0.012321743   
## 295 296 297 298 299 300   
## 0.035909972 0.006528726 0.035909972 0.012321743 0.012321743 0.012321743   
## 301 302 303 304 305 306   
## 0.012321743 0.002196228 0.012321743 0.035909972 0.006528726 0.012321743   
## 307 308 309 310 311 312   
## 0.006528726 0.012321743 0.006528726 0.035909972 0.035909972 0.035909972   
## 313 314 315 316 317 318   
## 0.006528726 0.012321743 0.012321743 0.012321743 0.012321743 0.002196228   
## 319 320 321 322 323 324   
## 0.012321743 0.012321743 0.035909972 0.012321743 0.035909972 0.012321743   
## 325 326 327 328 329 330   
## 0.012321743 0.012321743 0.002196228 0.006528726 0.035909972 0.012321743   
## 331 332 333 334 335 336   
## 0.012321743 0.035909972 0.002196228 0.012321743 0.006528726 0.035909972   
## 337 338 339 340 341 342   
## 0.035909972 0.006528726 0.035909972 0.012321743 0.012321743 0.012321743   
## 343 344 345 346 347 348   
## 0.006528726 0.012321743 0.035909972 0.035909972 0.012321743 0.035909972   
## 349 350 351 352 353 354   
## 0.006528726 0.035909972 0.035909972 0.035909972 0.002196228 0.035909972   
## 355 356 357 358 359 360   
## 0.012321743 0.012321743 0.035909972 0.035909972 0.035909972 0.035909972   
## 361 362 363 364 365 366   
## 0.035909972 0.035909972 0.012321743 0.006528726 0.035909972 0.012321743   
## 367 368 369 370 371 372   
## 0.002196228 0.012321743 0.012321743 0.035909972 0.035909972 0.035909972   
## 373 374 375 376 377 378   
## 0.035909972 0.035909972 0.035909972 0.012321743 0.035909972 0.035909972   
## 379 380 381 382 383 384   
## 0.035909972 0.012321743 0.012321743 0.012321743 0.012321743 0.035909972   
## 385 386 387 388 389 390   
## 0.012321743 0.006528726 0.012321743 0.002196228 0.035909972 0.012321743   
## 391 392 393 394 395 396   
## 0.006528726 0.012321743 0.012321743 0.012321743 0.035909972 0.006528726   
## 397 398 399 400 401 402   
## 0.006528726 0.006528726 0.035909972 0.035909972 0.035909972 0.035909972   
## 403 404 405 406 407 408   
## 0.006528726 0.035909972 0.035909972 0.012321743 0.012321743 0.006528726   
## 409 410 411 412 413 414   
## 0.012321743 0.035909972 0.035909972 0.035909972 0.012321743 0.035909972   
## 415 416 417 418 419 420   
## 0.035909972 0.035909972 0.035909972 0.035909972 0.035909972 0.006528726   
## 421 422 423 424 425 426   
## 0.035909972 0.012321743 0.035909972 0.035909972 0.012321743 0.012321743   
## 427 428 429 430 431 432   
## 0.012321743 0.012321743 0.035909972 0.012321743 0.012321743 0.006528726   
## 433 434 435 436 437 438   
## 0.035909972 0.035909972 0.035909972 0.035909972 0.035909972 0.002196228   
## 439 440 441 442 443 444   
## 0.012321743 0.035909972 0.006528726 0.006528726 0.035909972 0.012321743   
## 445 446 447 448 449 450   
## 0.012321743 0.012321743 0.035909972 0.035909972 0.012321743 0.035909972   
## 451 452 453 454 455 456   
## 0.012321743 0.035909972 0.002196228 0.035909972 0.035909972 0.002196228   
## 457 458 459 460 461 462   
## 0.006528726 0.035909972 0.006528726 0.002196228 0.012321743 0.035909972   
## 463 464 465 466 467 468   
## 0.012321743 0.012321743 0.035909972 0.002196228 0.035909972 0.012321743   
## 469 470 471 472 473   
## 0.035909972 0.012321743 0.075099778 0.075099778 0.075099778

Task 6 - based on the data above i used the train table and found examples of parolee 1 and parolee 2. The predicted probability of parole violation of the Parolee1 is around 33.8% and Parolee2 is around 20.7%

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] 0.8524576

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]], specificty=1-x[[ind]], cutoff = p[[ind]])  
 },perf@x.values, perf@y.values, pred@cutoffs)   
}  
print(opt.cut(ROCRperf, ROCRpred))

## [,1]  
## sensitivity 0.7272727  
## specificty 0.8588517  
## cutoff 0.2069629

Task 8 see above sensitivity is 0.73 specificity is 0.86 and cutoff is 0.21 The implications of incorrectly classifying a parolee could cause a repeat violator to be set free instead of denied parole and would cost more money to apprehend them once they violated parole.

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

##   
## FALSE TRUE  
## NoParoleViolation 405 13  
## ParoleViolation 36 19

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

## [1] 0.8964059

t1 = table(train$violator,predictions > 0.4)  
t1

##   
## FALSE TRUE  
## NoParoleViolation 405 13  
## ParoleViolation 36 19

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

## [1] 0.8964059

Task 10 - The probability threshhold to use would be 0.5 or 0.4 to determine the accuracy of the model which would give us 90% accuracy on the model created above based on the information in the tables above.