ProblemSet6

David Aarhus

3/22/2020

#Question a)

```
library(MASS)
data(Boston)

# a binary outcome for pricey home
Boston$PriceyHome <- ifelse(Boston$medv > 35, 1, 0)
# converting chas into a factor

Boston$chas <- factor(Boston$chas)
set.seed(2020)
trainSize <- 0.75
train_idx <- sample(1:nrow(Boston), size = nrow(Boston) * trainSize, replace=FALSE)
housing_train <- Boston[train_idx,]
housing_test <- Boston[-train_idx,]
head(housing_train)
```

```
##         crim zn indus chas   nox    rm   age    dis rad tax ptratio  black
## 412 14.05070  0 18.10    0 0.597 6.657 100.0 1.5275  24 666    20.2  35.05
## 236  0.33045  0  6.20    0 0.507 6.086  61.5 3.6519   8 307    17.4 376.75
## 87   0.05188  0  4.49    0 0.449 6.015  45.1 4.4272   3 247    18.5 395.99
## 22   0.85204  0  8.14    0 0.538 5.965  89.2 4.0123   4 307    21.0 392.53
## 216  0.19802  0 10.59    0 0.489 6.182  42.4 3.9454   4 277    18.6 393.63
## 321  0.16760  0  7.38    0 0.493 6.426  52.3 4.5404   5 287    19.6 396.90
##     lstat medv PriceyHome
## 412 21.22 17.2          0
## 236 10.88 24.0          0
## 87  12.86 22.5          0
## 22  13.83 19.6          0
## 216  9.47 25.0          0
## 321  7.20 23.8          0
```

#Question b)

```
library("doBy")
summaryBy(. ~ PriceyHome, housing_train, FUN=mean)
```

```
##   PriceyHome crim.mean  zn.mean indus.mean  nox.mean  rm.mean age.mean dis.mean
## 1          0 3.9723238 10.18421  11.526316 0.5553070 6.157643 68.32076 3.842112
## 2          1 0.9861332 23.77027   6.517838 0.5178108 7.541730 61.67568 3.687368
##   rad.mean tax.mean ptratio.mean black.mean lstat.mean medv.mean
## 1 9.672515 416.0322     18.63947   356.1615  13.300058  20.35702
## 2 6.135135 306.3514     16.15676   387.2238   4.701892  43.58108
```

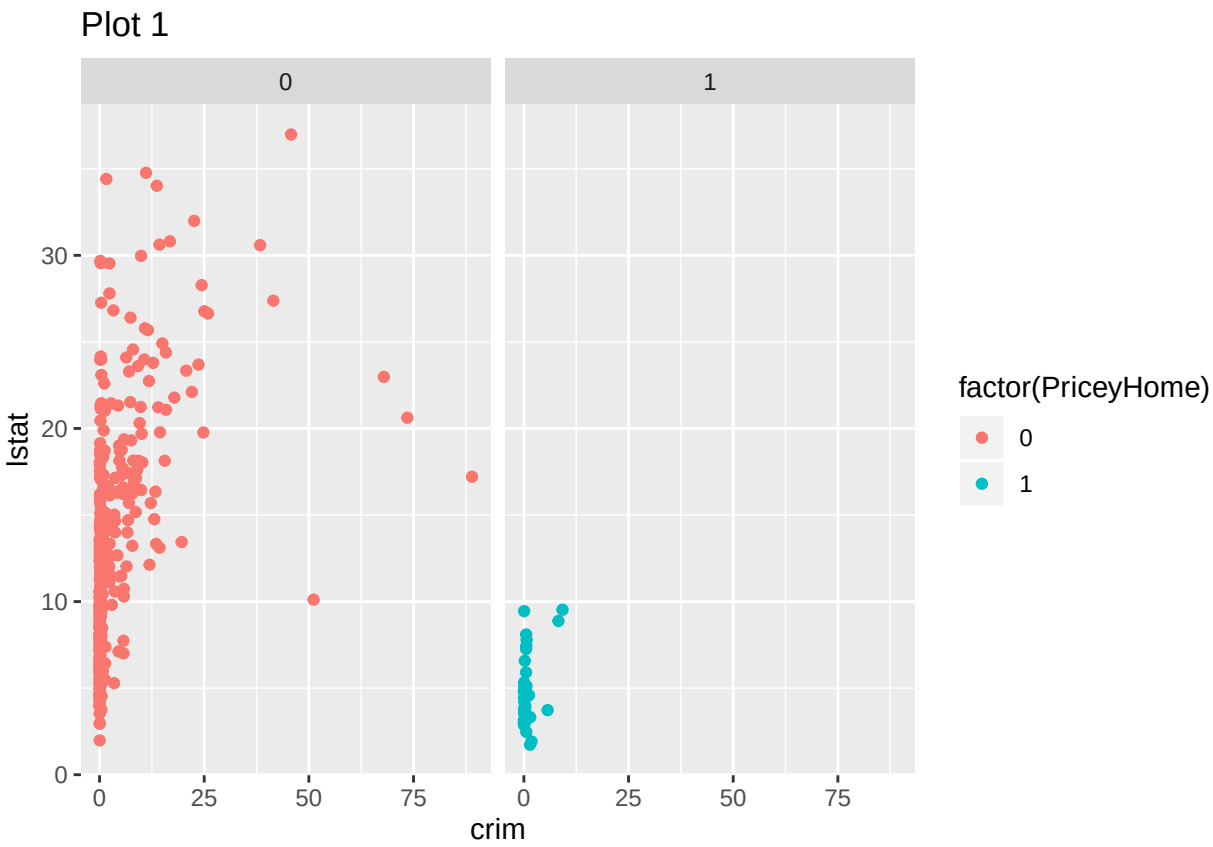
Non-Pricey and Pricey Homes differ the most with per capita crime rate by town (crim), proportion of

residential land zoned for lots over 25,000 sq.ft (zn), proportion of non-retail business acres per town (indus), full-value property-tax rate per $10,000 (tax), lower status of the population (lstat), and median value of owner-occupied homes (medv).

#Question c)

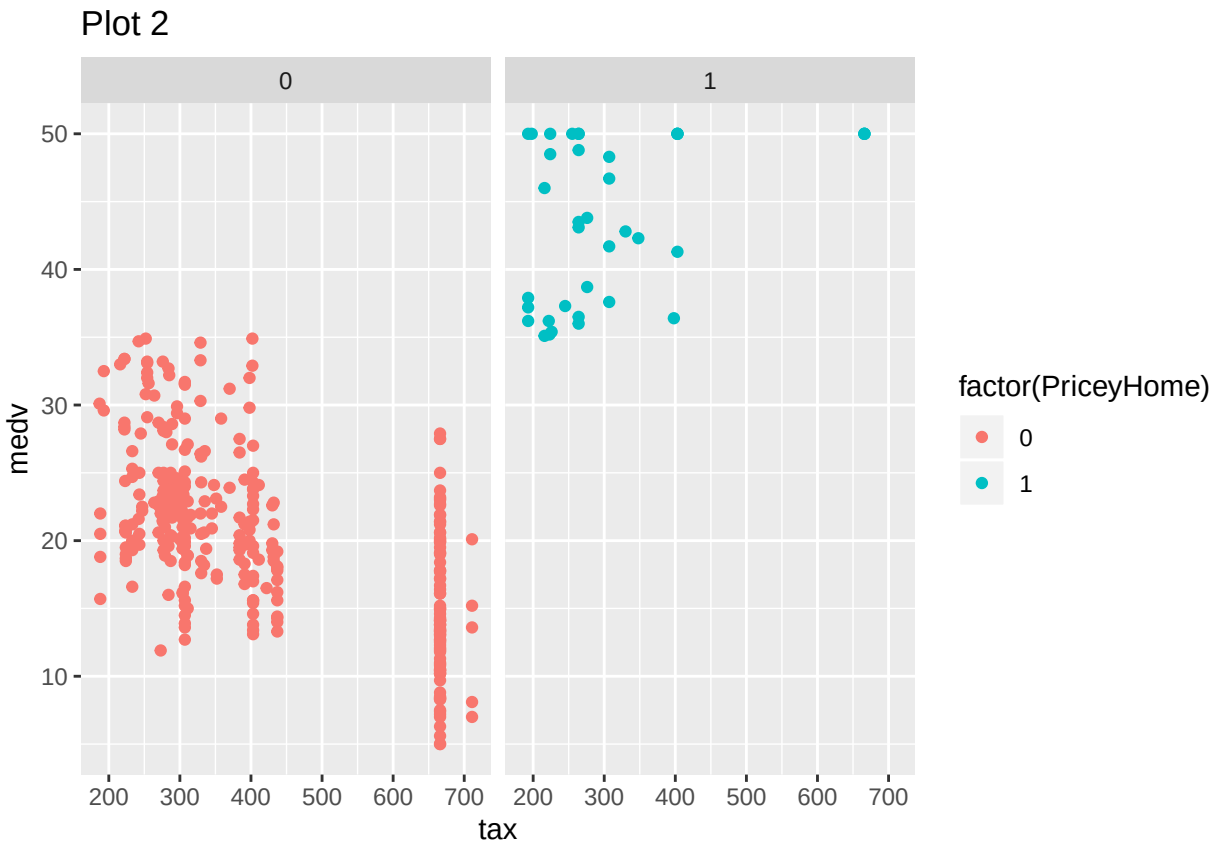
```
library(ggplot2)

ggplot(housing_train, aes(crim, lstat, color = factor(PriceyHome))) +
         geom_point() +
         facet_wrap('~PriceyHome') +
         labs(title = "Plot 1")
```



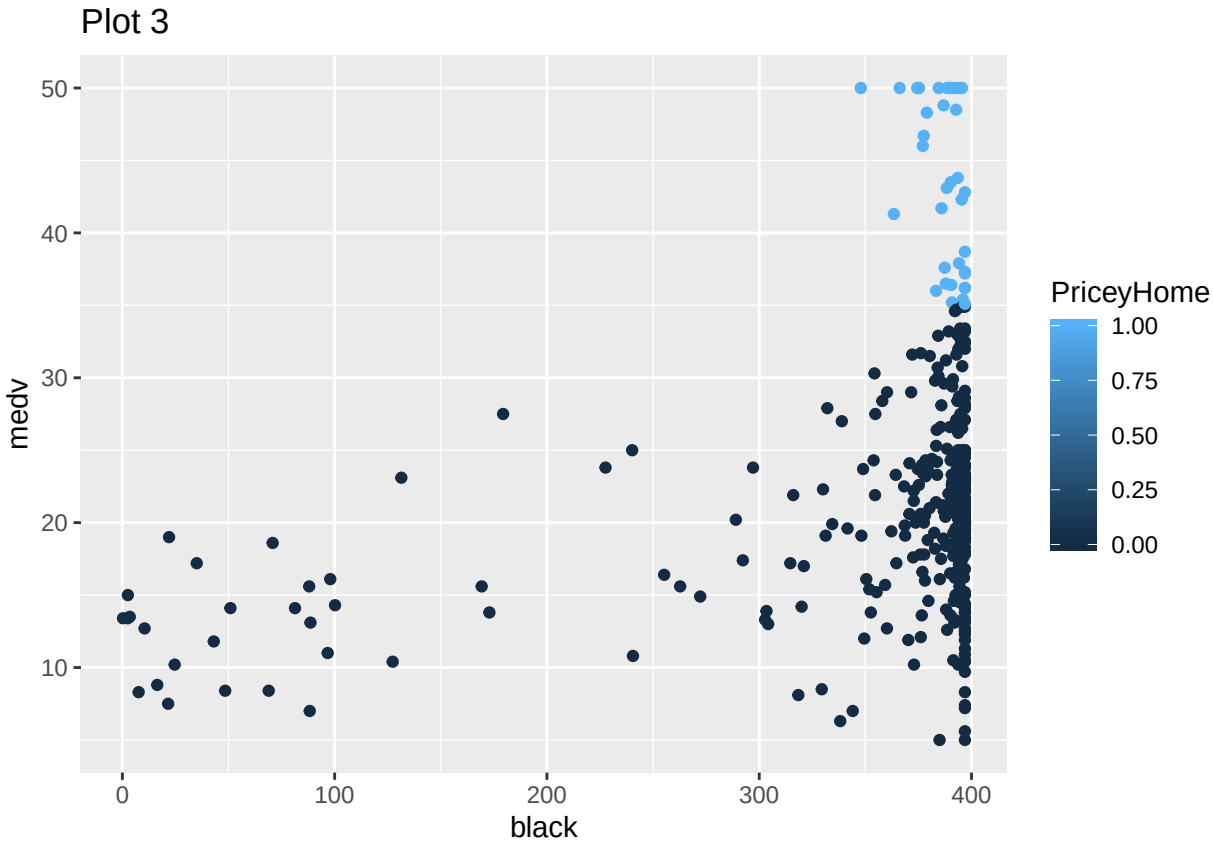
Plot 1 shows how most non-Pricey Homes have more crime and contain the lower status of the population

```
ggplot(housing_train, aes(tax, medv, color = factor(PriceyHome))) +
         geom_point() +
         facet_wrap('~PriceyHome') +
         labs(title = "Plot 2")
```



Plot 2 simply describes how as the value of a house reaches about $35,000, the house enters the category of a 'PriceyHome"

```
ggplot(housing_train, aes(black, medv, color = PriceyHome)) +
  geom_point() +
  labs(title = "Plot 3")
```



Plot 3 shows how most of the proportion of the black community do not live in "PriceyHomes".

#Question d)

```
logit_fit <- glm(PriceyHome ~ chas,
                 data = housing_train,
                 family = binomial)
exp(logit_fit$coefficients[2])
```

```
##    chas1
## 3.393939
```

From this coefficent we can conclude that living on the Charles River makes the home 339.4% more likely to be a PriceyHome

#Question e)

```
logit_fit2 <- glm(PriceyHome ~ chas + crim + lstat + ptratio + zn + rm + tax + rad + nox,
                 data = housing_train,
                 family = binomial)
summary(logit_fit2)
```

```
##
## Call:
## glm(formula = PriceyHome ~ chas + crim + lstat + ptratio + zn +
##     rm + tax + rad + nox, family = binomial, data = housing_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.11405  -0.13644  -0.04046  -0.00842   3.01389
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.600700   7.493998  -1.014 0.310469
## chas1        0.117263   0.865149   0.136 0.892184
## crim         0.040950   0.044915   0.912 0.361913
## lstat       -0.529936   0.159084  -3.331 0.000865 ***
## ptratio     -0.386204   0.203410  -1.899 0.057611 .
## zn          -0.008074   0.012029  -0.671 0.502062
## rm           2.145226   0.614049   3.494 0.000477 ***
## tax         -0.012840   0.005341  -2.404 0.016222 *
## rad          0.319324   0.127432   2.506 0.012216 *
## nox          6.739855   5.022217   1.342 0.179593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 242.434  on 378  degrees of freedom
## Residual deviance:  84.638  on 369  degrees of freedom
## AIC: 104.64
##
## Number of Fisher Scoring iterations: 9
```

```
#Need to exponentiate to interpret
exp(logit_fit2$coefficients)
```

```
##  (Intercept)        chas1         crim        lstat      ptratio           zn
## 5.001011e-04 1.124415e+00 1.041801e+00 5.886424e-01 6.796318e-01 9.919583e-01
##           rm          tax          rad          nox
## 8.543972e+00 9.872421e-01 1.376198e+00 8.454384e+02
```

From our model we can see that living on the River makes a house about 112.4 % more likely to be a PriceyHome. This still shows how much of an impact living on the River is, however it is significanly lower when you consider all the variables in the model.

#Question f)

```
preds_train <- data.frame(scores = predict(logit_fit2, type = "response"), housing_train)
preds_train <- data.frame(class_preds05 = ifelse(preds_train$scores > 0.5, 1, 0), preds_train)

preds_test <- data.frame(scores = predict(logit_fit2, newdata = housing_test, type = "response"), housi
preds_test <- data.frame(class_preds05 = ifelse(preds_test$scores > 0.5, 1, 0), preds_test)
```

#Question g)

```
#Train confusion matrix
table(preds_train$class_preds05,preds_train$PriceyHome)
```

```
##
##       0   1
##   0 339  11
##   1   3  26
```

Train Accuracy: 365/379 = 0.963 Train True Positive: 26 Train True Negative: 339 Sensitivity: 26/37 = 0.703 Specificity: 339/342 = 0.991 False positive rate: 3/342 = 0.0089

```
#Test confusion matrix
table(preds_test$PriceyHome, preds_test$class_preds05)
```

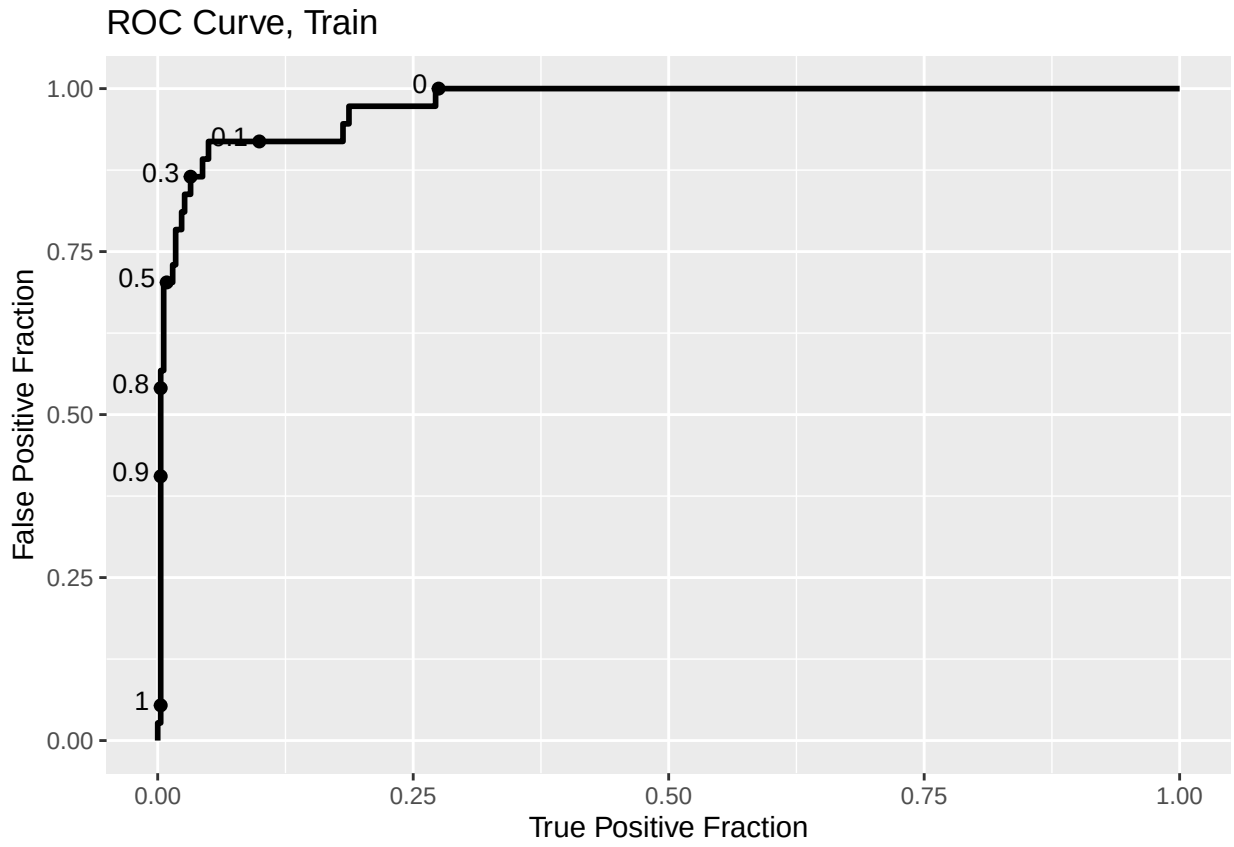
```
##
##        0   1
##   0 113   3
##   1   4   7
```

Test Accuracy: 120/127 = 0.945 Train True Positive: 7 Train True Negative: 113 Sensitivity: 7/10 = 0.70 Specificity: 113/117 = 0.966 False positive rate: 4/117 = 0.034

#Question h) Typically, we would like to have a specificity and sensitivity close to 1, however in this case it is not extremely important to have a high sensitivity due to the fact the stakes are not high, like if we were trying to predict bomb locations in the military. However, if you are a real estate agent and are trying to create an extremely accurate model to help determine price on a house, you would want to raise the sensitivity so customers would see how close you are to the market. Also it would give you the highest amout of profit. The cutoff is at 0.5 right now so in order to raise sensitivity, I would lower the cutoff to about 0.45 to help raise the sensitivity rating by allowing for more positive predictions.

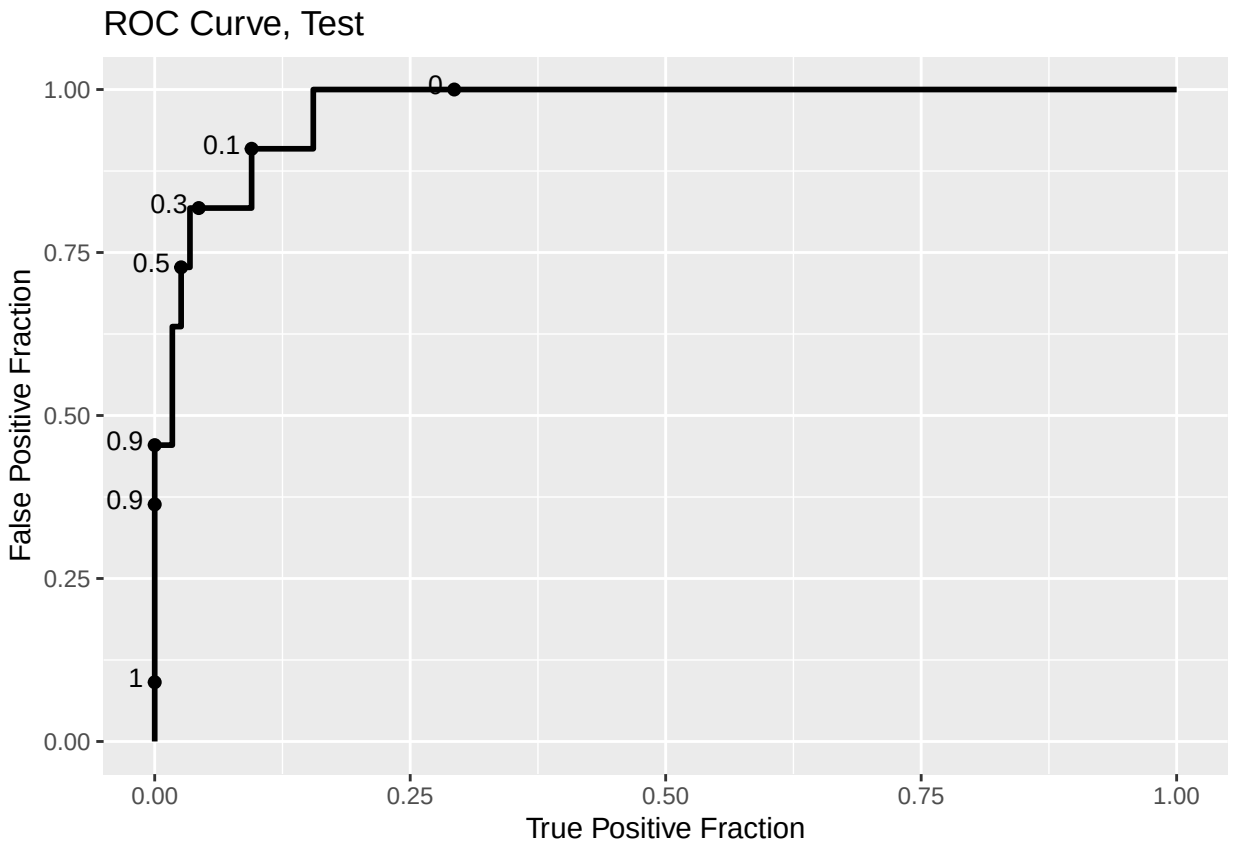
#Question i)

```
library(plotROC)
train_ROC <- ggplot(preds_train, aes(m = scores, d = PriceyHome)) +
geom_roc(labelsize = 3.5, cutoffs.at = c(.99,.9,.8,.5,.3,.1,.01)) + labs(x = "True Positive Fraction",
                                                                         y = "False Positive Fraction",
                                                                         title = "ROC Curve, Train")
train_ROC
```




```
test_ROC <- ggplot(preds_test, aes(m = scores, d = PriceyHome)) +
geom_roc(labelsize = 3.5, cutoffs.at = c(.99,.9,.8,.5,.3,.1,.01)) + labs(x = "True Positive Fraction",
                                                                         y = "False Positive Fraction",
                                                                         title = "ROC Curve, Test")
```

```
test_ROC
```



#Question j)

```
calc_auc(train_ROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9742374
```

```
calc_auc(test_ROC)
```

```
##   PANEL group      AUC
## 1     1    -1 0.968652
```

Our AUC's are very close to 1 which means they are very accurate. This also means they could be overfit, which is not what we want. To lessen this we can take out some variables in our model. This would allow the model to not be as fit, allowing for more degrees of freedom. Resulting in a lower AUC.

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.