# Problem Set 5

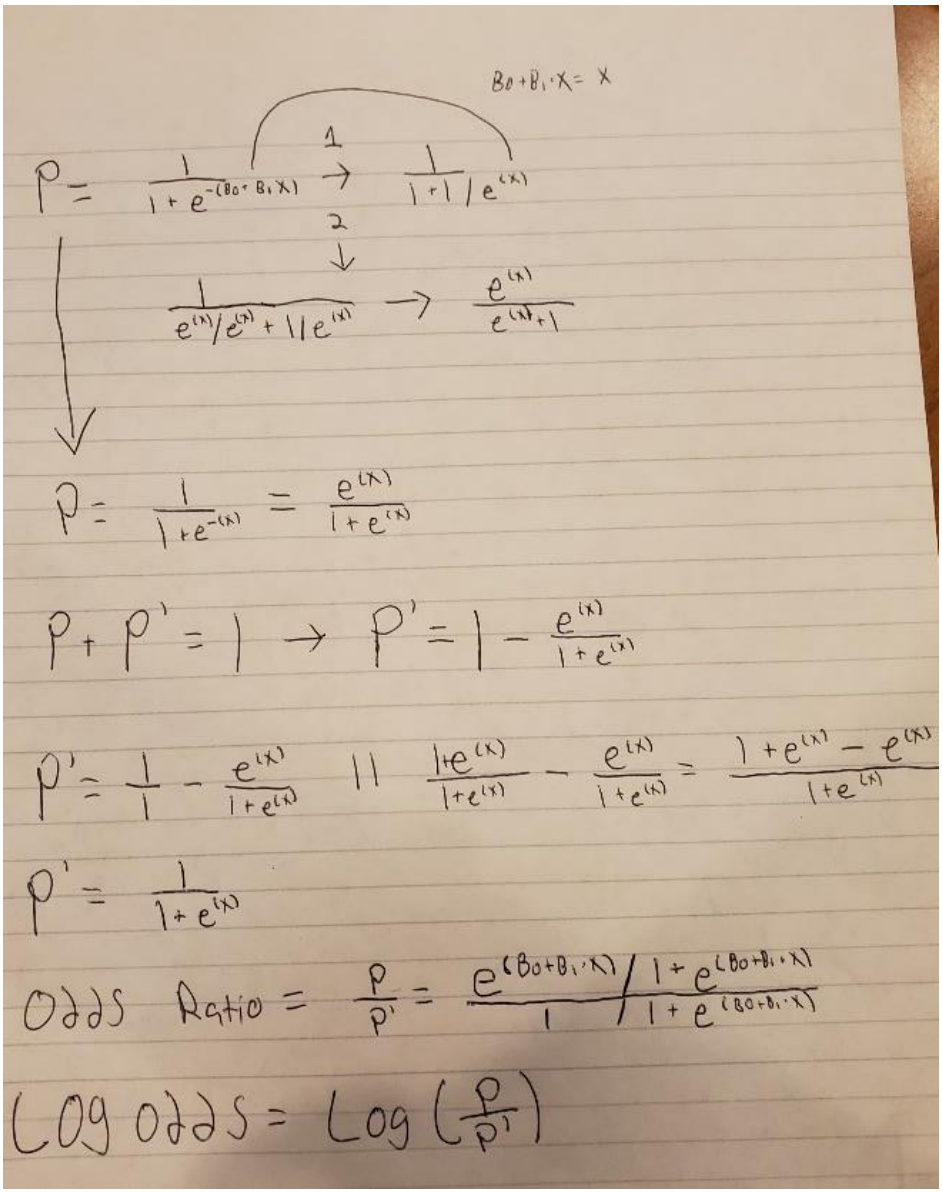
## MGSC 310, Fall 2019, Professor Hersh

Elmer Camargo + Nick Trella

## Libraries Needed

```
library("MASS")
library("tidyverse")
library("plotROC")
```

## Question 1 Derivation of Log Odds Ratio



## Question 2 Predicting Expensive Houses

### a - Preparing Data

```
data(Boston)
set.seed(1861)

trainSize <- 0.75
train_idx <-sample(1:nrow(Boston), size = floor(nrow(Boston)*trainSize))

housing <- Boston %>% mutate(PriceyHome = ifelse(medv > 40, 1,0), chas_factor
= factor(chas))

housing_train <- housing%>% slice(train_idx)
housing_test <- housing%>% slice(-train_idx)
```

### b - Where Pricey Homes Differ

```
housing_train <- housing_train %>% group_by(PriceyHome)
sum_train <- housing_train %>% summarise_all(list(mean = mean), na.rm = TRUE)

housing_test <- housing_test %>% group_by(PriceyHome)
sum_test <- housing_test %>% summarise_all(list(mean = mean), na.rm = TRUE)
```

Variables proportion of residential land zoned for lots over 25,000(zn), proximity to the Charles River (chas), and lower status of population (lstat) differ the most between pricey and non-pricey homes

## c - Plots! Plots! Plots!

```
ggplot(housing_train) + geom_point(aes(x = housing_train$zn, y = housing_train$PriceyHome, colour = housing_train$PriceyHome >0), shape = 36, size=6) +
  scale_colour_manual(name = 'PriceyHome', values = setNames(c('green','red'),c(T, F))) +
  xlab('proportion of residential land zoned for lots over 25,000 sq.ft.') +
ylab('PriceyHome')
```


Graph One seems to show that for larger homes there seems to be more homes classified as pricey that are either in small proportioned lands of residential lots or in large ones but not really in the 50th percentile. I believe this would be due to expensive homes being more likely to be in the heart of downtown areas or by itself. Pricey homes are not likely to appear in the high-density, “cookie cutter” suburbs.

```
ggplot(housing_train) + geom_point(aes(x = housing_train$age, y = housing_tra
in$PriceyHome, colour = housing_train$PriceyHome >0),size = 2) +
  scale_colour_manual(name = 'PriceyHome', values = setNames(c('green','red')
,c(T, F))) +
  xlab('Age') + ylab('PriceyHome')
```


Graph Two seems to show that there are more houses at the higher age range classified as Pricey homes. Though there are also greater homes classified as non-Pricey homes at the upper age range as well and not as many non-Pricey homes for young homes. This leads me to believe that older houses in particular are where the expensive houses and the cheap houses are. Once they age substantially they are either quite nice or awful.

```
ggplot(housing_train) + geom_point(aes(x = housing_train$lstat, y = housing_t
rain$PriceyHome, colour = housing_train$PriceyHome >0),size = 2) +
  scale_colour_manual(name = 'PriceyHome', values = setNames(c('green','red')
,c(T, F))) +
  xlab('lower status of the population (percent).') + ylab('PriceyHome')
```


Graph Three appears to show that PriceyHomes are not present often in areas where there is a greater percentage of lower status people. This leads me to the conclusion that in lower status areas (greater than approximately 10%) homes are valued less than they would be otherwise.

## d - Impact of the Charles River

```
log_train_mod = glm(PriceyHome ~ chas,
              data = housing_train,
              family = binomial)

log_test_mod = glm(PriceyHome ~ chas,
                   data = housing_test,
                   family = binomial)
```

The coefficient of 1.5614 on our log model means that if we take the exponential of that number exp(1.5614) = 4.765 we observe an approximately 376% increase in the chance of being a Priceyhome if the home is classifed as chas (if tract bounds river) (at a two star level of significance)

## e - Amenity Impact of the Charles River

```
log_train_mod_plus = glm(PriceyHome ~ chas + crim + lstat + ptratio + zn + rm
+ rad + nox,
              data = housing_train,
              family = binomial)

log_test_mod_plus = glm(PriceyHome ~ chas + crim + lstat + ptratio + zn + rm
+ rad + nox,
                        data = housing_test,
                        family = binomial)

summary(log_train_mod_plus)
##
## Call:
## glm(formula = PriceyHome ~ chas + crim + lstat + ptratio + zn +
##     rm + rad + nox, family = binomial, data = housing_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2426  -0.0538  -0.0053  -0.0003   2.5285
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.80950    8.97395   0.090 0.928124
## chas         0.29088    1.63620   0.178 0.858898
## crim         0.20146    0.08238   2.445 0.014472 *
## lstat       -1.06735    0.30497  -3.500 0.000465 ***
## ptratio     -0.74290    0.29048  -2.558 0.010542 *
## zn          -0.01266    0.01644  -0.770 0.441172
## rm           1.74403    0.59818   2.916 0.003551 **
## rad          0.27043    0.09777   2.766 0.005673 **
## nox          3.20584    6.30915   0.508 0.611366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 184.246  on 378  degrees of freedom
## Residual deviance:  51.163  on 370  degrees of freedom
## AIC: 69.163
##
## Number of Fisher Scoring iterations: 10
```

For chas, we can interpret is as again taking the exponetial of it (although we lost p-value significance,meaning that there is a decent chance this value would come about as per chance) but if we take the exponentialwe get exp(0.29088) = 1.3376 which would indicate a 33% higher chance of having Pricey home if chas=1 (if tract bounds river)

## f & g - Generating Probability Scores and Class Predictions / Classification + Confusion Matrices

```
preds_train_DF <- data.frame(
  scores_train = predict(log_train_mod_plus,
                         type = "response"),
  housing_train
)

preds_test_DF <- data.frame(
  scores_test = predict(log_test_mod_plus,
                        type = "response"),
  housing_test
)
```

## f & g - Generating Probability Scores and Class Predictions / Classification + Confusion Matrices

```
preds_train_DF <-  preds_train_DF %>% mutate(PosNeg05 =
                                               ifelse(scores_train > 0.5 & Pr
iceyHome == 1, "TP",
                                               ifelse(scores_train > 0.5 & Pr
iceyHome == 0, "FP",
                                               ifelse(scores_train <= 0.5 & P
riceyHome == 0, "TN",
                                               ifelse(scores_train <= 0.5 & P
riceyHome == 0, "FN",0)))))

preds_test_DF <-  preds_test_DF %>% mutate(PosNeg05 =
                                               ifelse(scores_test > 0.5 & Pri
ceyHome == 1, "TP",
                                               ifelse(scores_test > 0.5 & Pri
ceyHome == 0, "FP",
                                               ifelse(scores_test <= 0.5 & Pr
iceyHome == 0, "TN",
                                               ifelse(scores_test <= 0.5 & Pr
iceyHome == 1, "FN",0)))))
```

```
preds_train_DF <- data.frame(

  class_pred05 = ifelse(preds_train_DF$scores_train
                        > 0.5, 1, 0),
  preds_train_DF
)

preds_test_DF <- data.frame(

  class_pred05 = ifelse(preds_test_DF$scores_test
                        > 0.5, 1, 0),
  preds_test_DF
)

table(preds_train_DF$PosNeg05)
##
##   FN  FP  TN  TP
##   6   1 353  19
table(preds_train_DF$class_pred05,preds_train_DF$PriceyHome)
##
##       0   1
##   0 353   6
##   1   1  19

table(preds_test_DF$PosNeg05)
##
##  FN  FP  TN  TP
##   2   2 119   4
table(preds_test_DF$class_pred05,preds_test_DF$PriceyHome)
##
##       0   1
##   0 119   2
##   1   2   4
```

## h - Interpretation and Adjustments

Training: true positive rate/sensitivity is good = 76%, 19/25 - tp/tp+fn

Training: true negative rate/specificity = 99.7%, 353/354 - fn/tn+fp

Training: false positive rate/type 1 error = 0%, 1/354 - fp/fp+tn

Training: accuracy = 98% 372/379 - tp+tn/total

Testing: true positive rate/sensitivity is good = 67%, 4/6 - tp/tp+fn

Testing: true negative rate/specificity = 2%, 2/121 - fn/tn+fp

Testing: false positive rate/type 1 error = 2%, 2/121 - fp/fp+tn

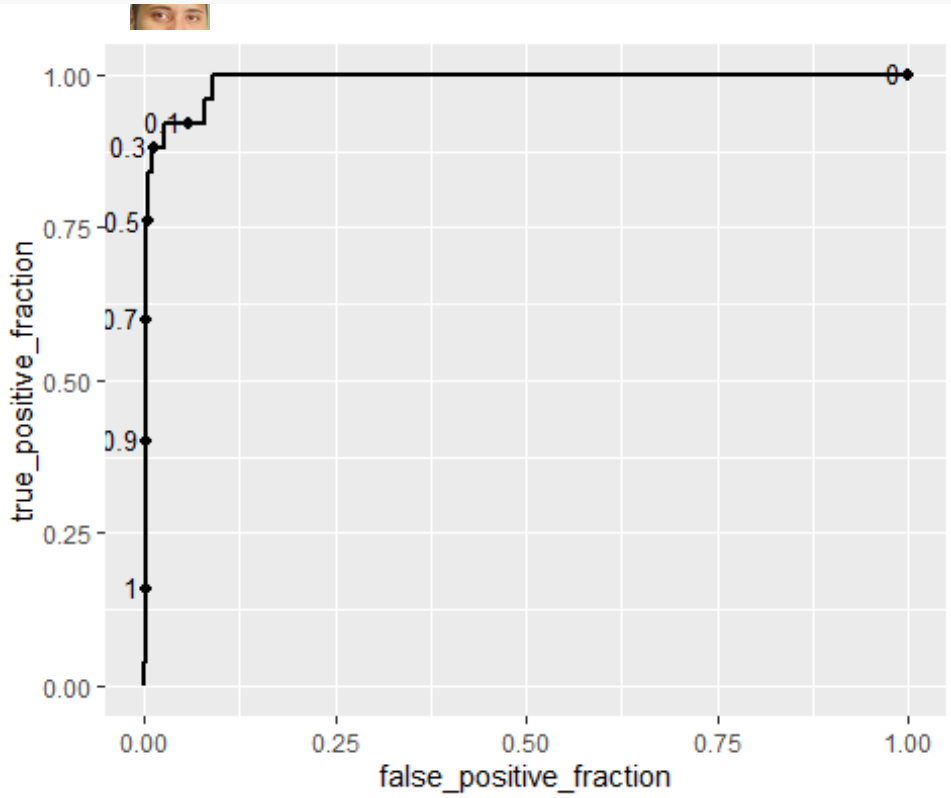
Training: accuracy = 97%, 123/127 - fp/fp+tn

We should increase the prob cut off (moviong closer to 0) because it should increase our true positive rate at the expense of our false positive rate. Other factors we should consider for changing the cutoff would be what the intention behind our model is If we are only concerned about predicting homes that are not PriceyHomes, we're doing good and could move the cutoff until we no longer are at that 99+ percentage prediction rate. However, if we are more concerned with predicting homes that are actually pricey, we would want to adjust out cutoff until we're I'd argue 90+ percent accurate on predicting them. The trade off is ever present and we should act in according to our best interest here. Our accuracy scores however are quite high for each set so we it appears minor adjustments only may be necessary (if we like how we did on positives and negatives)

## i - ROC Plots

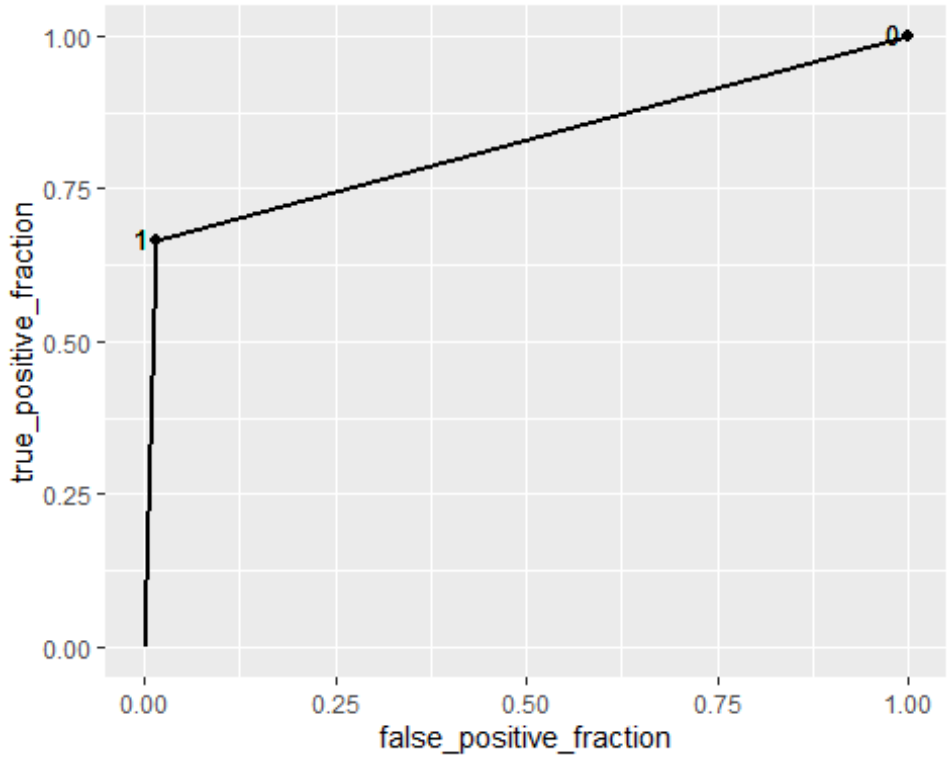
```
trainROC <- ggplot(data = preds_train_DF,
       aes(m = scores_train,
              d = PriceyHome)) +
       geom_roc(labelsize = 3.5,
                cutoffs.at = c(.99,.9,.7,.5,.3,.1,0))

trainROC
```



```
testROC <- ggplot(data = preds_test_DF,
       aes(m = scores_test,
              d = PriceyHome)) +
       geom_roc(labelsize = 3.5,
                cutoffs.at = c(.99,.9,.7,.5,.3,.1))

testROC
```




## j - Test of Fitness - Area Under the Curve

```
calc_auc(trainROC)
##   PANEL group       AUC
## 1     1    -1 0.9892655
calc_auc(testROC)
##   PANEL group       AUC
## 1     1    -1 0.8250689
```

The training model is most likely over fit as the area of the curve is nearly one. In order to reduce this overfitting, we would recommended taking out some variables and get more data.

The test model appears decently fit based on the AUC number calculated, leading us to believe that only minor changes would be neccesary to imporve model fit. More data would help though.