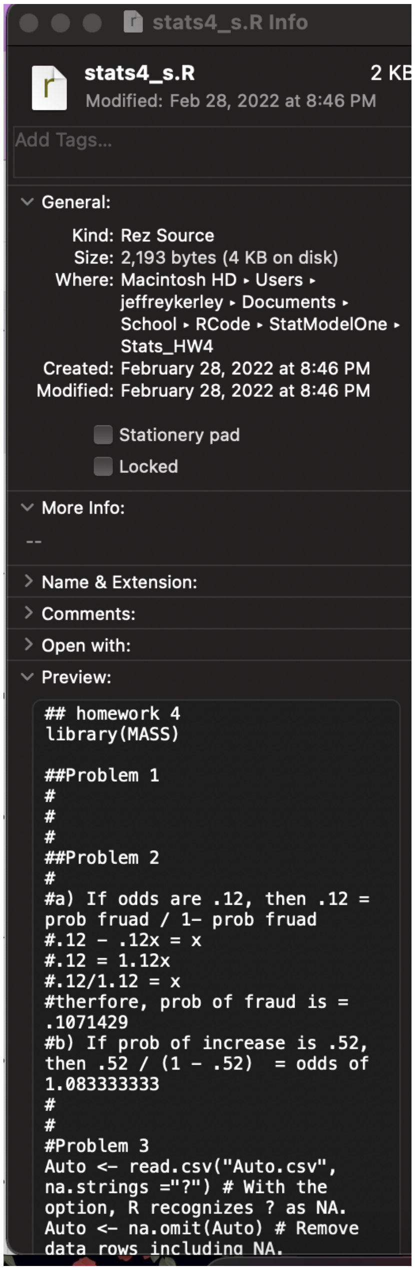
Jeffrey Kerley
Jakcqc
Jakcqc@umsystem.edu
March 1st, 2022

I FORGOT TO SUBMIT BUT DID NOT EDIT PAST DUE DATE!!!!!!



Problem 1: The discriminant function serves to represent this bayes maximum at each point in the two classes. It is just a rewrite and extension of the distance we calculate between two points either in Euclid or mahalanobis.

Problem 2:

a) If odds are .12, then .12 = prob fruad / 1- prob fruad
.12 - .12x = x
.12 = 1.12x
.12/1.12 = x
therfore, prob of fraud is = .1071429

b) If prob of increase is .52, then .52 / (1 - .52) = odds of 1.083333333

Problem 3:

a) Mpg, cylinders, horsepower, as well as year, appear to be statistically significant.

```
> #a)
> newLogRR = glm(Auto.class$origin ~ ., data = Auto.class, family = binomial)
> summary(newLogRR)

Call:
glm(formula = Auto.class$origin ~ ., family = binomial, data = Auto.class)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.79504  -0.34097  -0.14154  -0.04997   2.71356

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  10.966194   4.700567   2.333  0.01965 *
mpg           0.241707   0.051999   4.648 3.35e-06 ***
cylinders    -1.195342   0.289233  -4.133 3.58e-05 ***
horsepower    0.028470   0.015685   1.815  0.06952 .
acceleration -0.003536   0.102960  -0.034  0.97260
year         -0.198193   0.061297  -3.233  0.00122 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 359.93  on 323  degrees of freedom
Residual deviance: 207.19  on 318  degrees of freedom
AIC: 219.19

Number of Fisher Scoring iterations: 7
```

b) The estimated coefficient for year is -0.198, which means that for each increase in year, the log odds for being japanese car will go down that amount. Cylinders has -1.195 cooefficiant, so again, the log odds go down being japanese that amount for each increase in Cylinders.

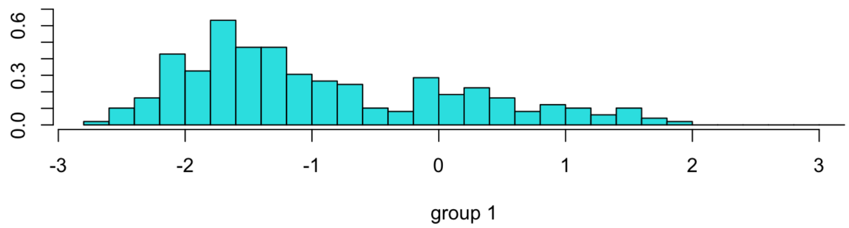
c)

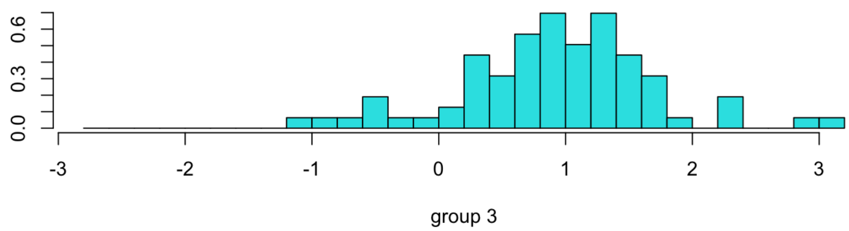
```
> makeConProb = predict(newLogRR, type='response')
> makeConPred =rep("American", length(makeConProb))
> makeConPred[makeConProb > 0.5] = "Japanese"
> newOT = table(makeConPred, Auto.class$origin)
> newOT

makeConPred   1   3
  American 219  22
  Japanese  26  57
> (newOT[1] + newOT[4]) / (newOT[1]+newOT[2]+ newOT[3]+newOT[4])
[1] 0.8518519
```

d)

```
> #d
> autoLda = lda(Auto.class$origin ~., data = Auto.class)
> autoLda
Call:
lda(Auto.class$origin ~ ., data = Auto.class)

Prior probabilities of groups:
        1         3
0.7561728 0.2438272

Group means:
       mpg cylinders horsepower acceleration     year
1 20.03347  6.277551  119.04898     14.99020 75.59184
3 30.45063  4.101266   79.83544     16.17215 77.44304

Coefficients of linear discriminants:
                     LD1
mpg           0.166780032
cylinders    -0.351573318
horsepower    0.010948130
acceleration -0.007765445
year         -0.113510801
> plot(autoLda)
```

e) We can see from the plots, that





f)

```
> autoLdaPred = predict(autoLda, newdata = Auto.class , type = "response")
> autoLdaPredT = table(autoLdaPred$class, Auto.class$origin)
> autoLdaPredT

      1   3
  1 220  23
  3  25  56
```

g)

```
autoQda = qda(Auto.class$origin ~., data = Auto.class)
> autoQda
Call:
qda(Auto.class$origin ~ ., data = Auto.class)

Prior probabilities of groups:
        1         3 
0.7561728 0.2438272 

Group means:
       mpg cylinders horsepower acceleration     year
1 20.03347  6.277551  119.04898     14.99020 75.59184
3 30.45063  4.101266   79.83544     16.17215 77.44304
```

h)

```
autoQdaPred = predict(autoQda, newdata = Auto.class , type = "response")
```

```
> autoQdaPredT = table(autoQdaPred$class, Auto.class$origin)
> autoQdaPredT

     1   3
  1 194  10
  3  51  69
```

i) Below are scores for lda then qda for each quantified performance. Lda first, and then Qda. We see better accuracy for lda, better sensitivity for lda, but better precision and specificity for Qda.

```
 #accuracy
> (autoLdaPredT[1] + autoLdaPredT[4]) / summerT(autoLdaPredT)
[1] 0.8518519
> (autoQdaPredT[1] + autoQdaPredT[4]) / summerT(autoQdaPredT)
[1] 0.8117284
> #sensitivity
> autoLdaPredT[4] / (autoLdaPredT[2] + autoLdaPredT[4])
[1] 0.691358
> autoQdaPredT[4] / (autoQdaPredT[2] + autoQdaPredT[4])
[1] 0.575
> #specificity
> autoLdaPredT[1] / (autoLdaPredT[3] + autoLdaPredT[1])
[1] 0.9053498
> autoQdaPredT[1] / (autoQdaPredT[3] + autoQdaPredT[1])
[1] 0.9509804
> #precision
> autoLdaPredT[4] / (autoLdaPredT[4] + autoLdaPredT[3])
[1] 0.7088608
> autoQdaPredT[4] / (autoQdaPredT[4] + autoQdaPredT[3])
[1] 0.8734177
```