FILIPPO RIVA
MT 3100 (Brandt)
Statistical Learning

Assignment 7

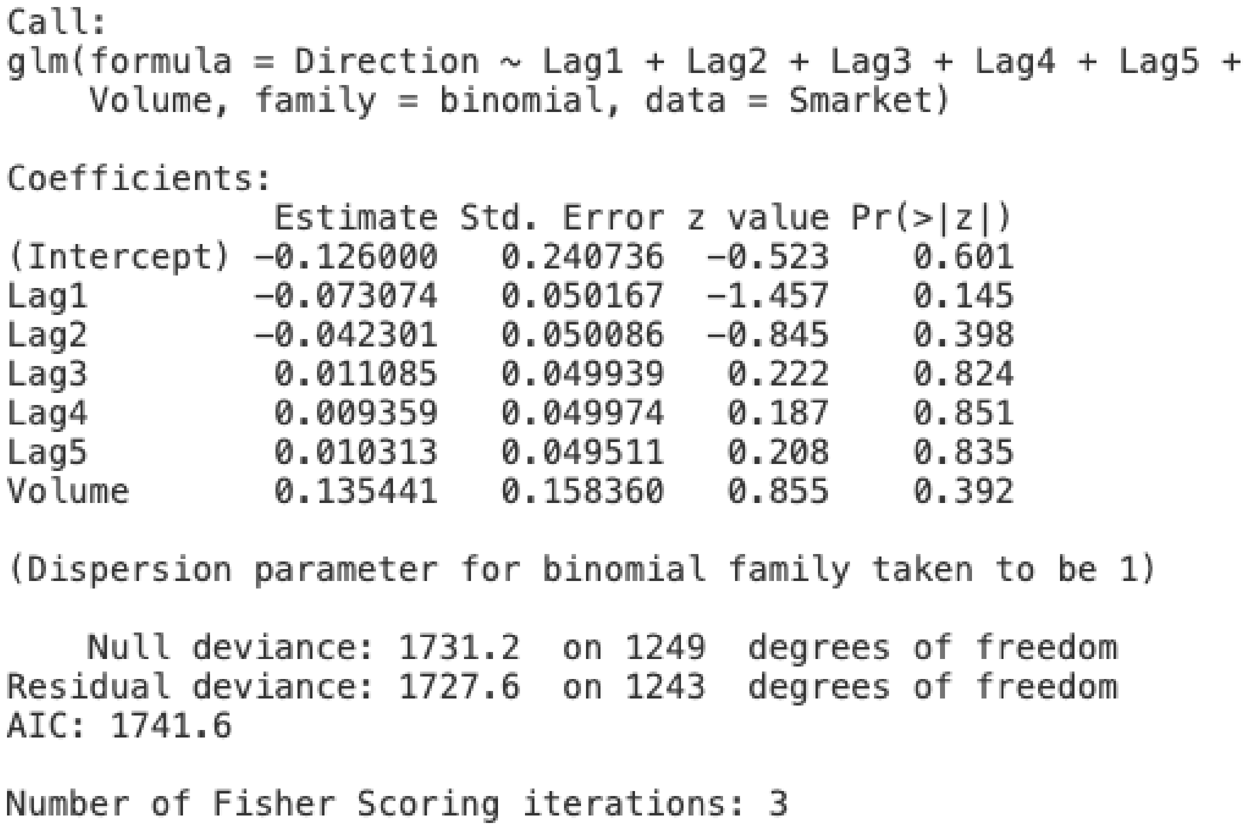
In this assignment we study logistic regression. Also, for the first time this semester, we will actually build a model one data set and test it on another.

For the work below, I recommend you stick with the book's names for the objects created (glm.fits, glm.probs, glm.pred, etc.).

a. Read in the Smarket file and find the logistic regression model for Direction as the book does (as a function of Lag1 through Lag5, and Volume). In your report, include output for the summary of your model, and make sure it matches the book's output on page 173.

CODE:

```
Smarket=read.csv("Smarket.csv",stringsAsFactors=T)
summary(Smarket)
glm.fits <- glm(Direction~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Smarket, family = binomial)
summary(glm.fits)
```

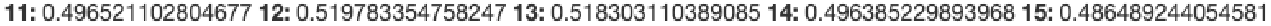


```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1731.2  on 1249  degrees of freedom
Residual deviance: 1727.6  on 1243  degrees of freedom
AIC: 1741.6

Number of Fisher Scoring iterations: 3
```

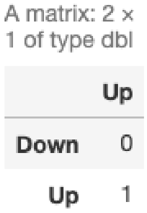
b. Follow the book's commands on p. 174 to set up glm.probs, the list of probabilities for Direction for the first 10 days. Make sure your numbers match the book's. In your report, give the probabilities for days 11 through 15. Continue to set up the list glm.pred as the book does. In your report, include entries 11 through 15 of glm.pred and make sure they agree with the probability values in glm.probs.

CODE 1:

```
glm.probs=predict(glm.fits, type= "response")
glm.probs[11:15]
Direction=Smarket$Direction
contrasts(Direction)
```



```
11: 0.496521102804677 12: 0.519783354758247 13: 0.518303110389085 14: 0.496385229893968 15: 0.486489244054581
```

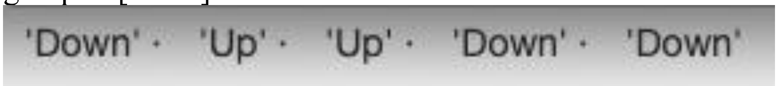


A matrix: 2 × 1 of type dbl

	Up
Down	0
Up	1

CODE 2:

```
glm.pred=rep("Down", 1250)
glm.pred[glm.probs > .5] = "Up"
glm.pred[11:15]
```

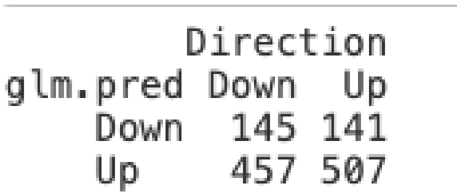


```
'Down' · 'Up' · 'Up' · 'Down' · 'Down'
```

c. Use the table function (p. 174) to create the confusion matrix. This matrix lets us examine the performance of the model. To get the table command to work, I had to use Smarket$Direction. Include the confusion matrix in your report, and make sure it agrees with the one in the book. The book explains the meaning of the values 145 and 507. In your report, explain the meaning of the value 141. What exactly does this number tell us?

CODE:

```
glm.pred=rep("Down", 1250)
glm.pred[glm.probs > .5] = "Up"
Direction=Smarket$Direction
table(glm.pred, Direction)
```



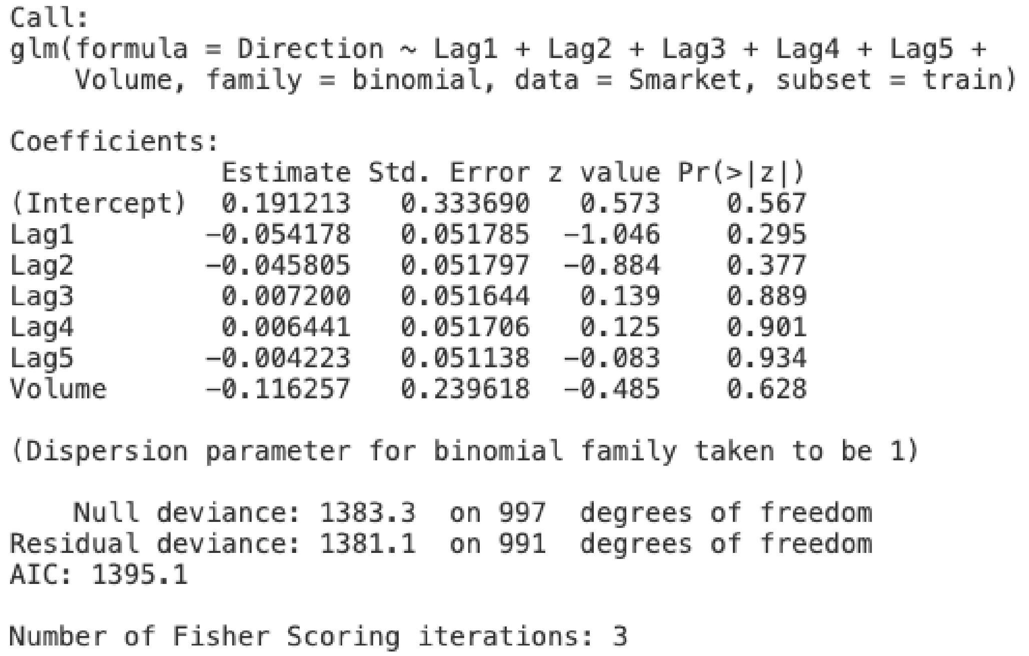
```
         Direction
glm.pred Down  Up
    Down  145 141
    Up    457 507
```

Probability = (145+507)/1250 = 0.5216
The number "141" is in the off-diagonal representing an incorrect prediction. The model that We developed incorrectly predicted that the market would go up 141 days when it went down.

d. Let's train a model on certain days and test it on others. Follow the book's commands to train the model years 2001 through 2004 and test it on 2005. In your report, include the summary command for your new model (the book does not include this!) and the confusion matrix. Make sure your confusion matrix matches the book's at the top of page 176. Note: to set up the training set, you may have to explain where Year comes from. That is, use this modification: train=(Smarket$Year2005). Finally, find and test the model that uses only Lag1 and Lag2. Make sure your confusion matrix matches the book's at the bottom of page 176

CODE 1:

```
Smarket=read.csv("Smarket.csv",stringsAsFactors=T)
Year=Smarket$Year
Lag1=Smarket$Lag1
Lag2=Smarket$Lag2
Lag3=Smarket$Lag3
Lag4=Smarket$Lag4
Lag5=Smarket$Lag5
Volume=Smarket$Volume
Direction=Smarket$Direction
train=(Year < 2005)
Smarket.2005=Smarket[!train,]
dim(Smarket.2005)
Direction.2005=Direction[!train]
glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, family=binomial, data=Smarket,
subset=train)
summary(glm.fits)
```

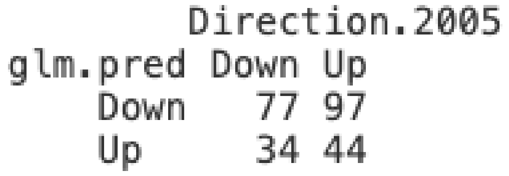


```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket, subset = train)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.191213   0.333690   0.573    0.567
Lag1        -0.054178   0.051785  -1.046    0.295
Lag2        -0.045805   0.051797  -0.884    0.377
Lag3         0.007200   0.051644   0.139    0.889
Lag4         0.006441   0.051706   0.125    0.901
Lag5        -0.004223   0.051138  -0.083    0.934
Volume      -0.116257   0.239618  -0.485    0.628

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1383.3  on 997  degrees of freedom
Residual deviance: 1381.1  on 991  degrees of freedom
AIC: 1395.1

Number of Fisher Scoring iterations: 3
```

CODE 2:

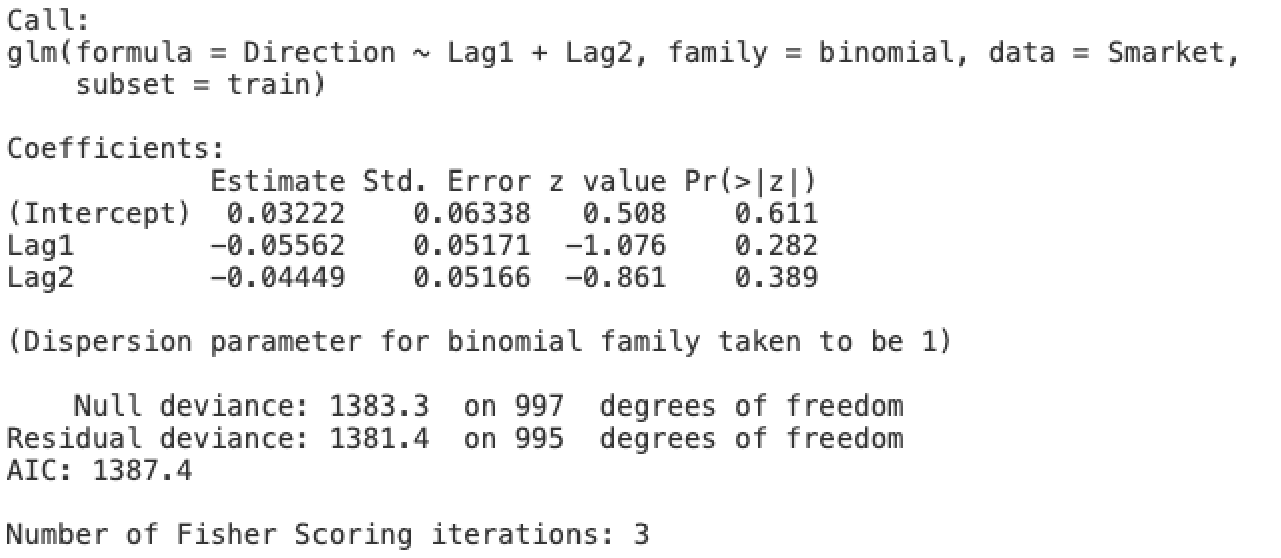
```
Year=Smarket$Year
train=(Year < 2005)
Smarket.2005 = Smarket[!train,]
dim(Smarket.2005)
Direction.2005 = Direction[!train]
glm.fits = glm( Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Smarket, family
= binomial, subset = train)
glm.probs = predict(glm.fits, Smarket.2005, type = "response")
glm.pred = rep("Down", 252)
glm.pred[glm.probs > .5] = "Up"
table(glm.pred, Direction.2005)
```

Confusion Matrix:

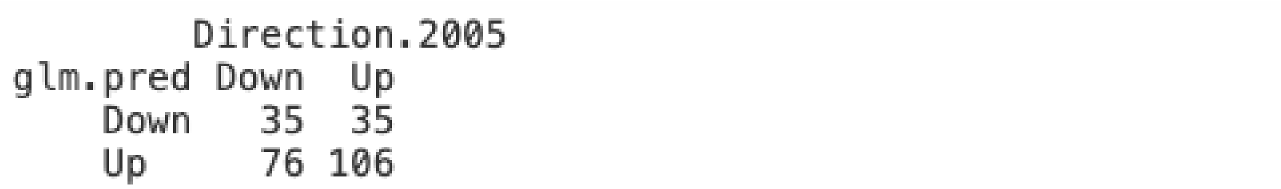


```
        Direction.2005
glm.pred Down Up
    Down   77 97
    Up     34 44
```

CODE 3:
glm.fits = glm(Direction ~ Lag1 + Lag2, data = Smarket, family = binomial, subset = train)
glm.probs = predict(glm.fits, Smarket.2005, type = "response")
glm.pred = rep("Down", 252)
glm.pred[glm.probs > .5] = "Up"
summary(glm.fits)
table(glm.pred, Direction.2005)



```
Call:
glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Smarket,
    subset = train)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.03222    0.06338   0.508    0.611
Lag1        -0.05562    0.05171  -1.076    0.282
Lag2        -0.04449    0.05166  -0.861    0.389

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1383.3  on 997  degrees of freedom
Residual deviance: 1381.4  on 995  degrees of freedom
AIC: 1387.4

Number of Fisher Scoring iterations: 3
```



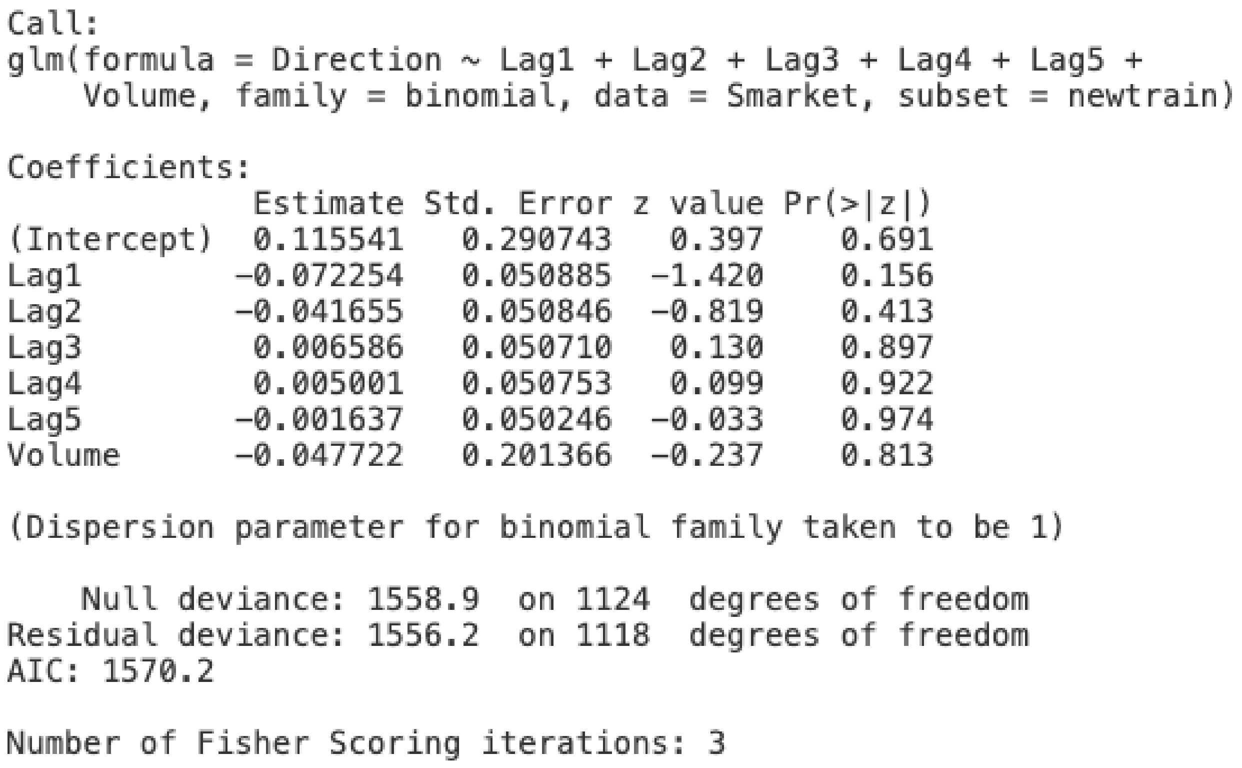
```
        Direction.2005
glm.pred Down  Up
    Down   35  35
    Up     76 106
```

e. Finally, let's try a different training set. Train your model (the one that uses Lag 1 through Lag 5 and Volume) on 2001 through 2004 and the first half of 2005, and then test in on the second half of 2005. That is, train your model on the first 1125 days and test on the last 125 days. Here is a tip to set up your training set: You need a vector that starts with TRUE 1125 times and ends with FALSE 125 times. These commands will set up your new training set.

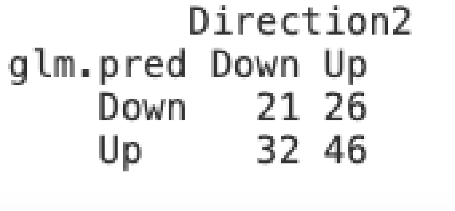
newtraint=rep(TRUE,1125) # Build the TRUE piece
newtrainf=rep(FALSE,125) # Build the FALSE piece
newtrain=c(newtraint,newtrainf) # Put them together

In your report, include the summary command for your model, its confusion matrix, and its success rate (percentage of correct predictions).

CODE:



```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket, subset = newtrain)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.115541   0.290743   0.397    0.691
Lag1        -0.072254   0.050885  -1.420    0.156
Lag2        -0.041655   0.050846  -0.819    0.413
Lag3         0.006586   0.050710   0.130    0.897
Lag4         0.005001   0.050753   0.099    0.922
Lag5        -0.001637   0.050246  -0.033    0.974
Volume      -0.047722   0.201366  -0.237    0.813

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1558.9  on 1124  degrees of freedom
Residual deviance: 1556.2  on 1118  degrees of freedom
AIC: 1570.2

Number of Fisher Scoring iterations: 3
```



```
        Direction2
glm.pred Down Up
    Down   21 26
    Up     32 46
```

Success Rate: (21+46)/125= 0.536

2. Use logistic regression to predict whether a car comes from the US. Before you begin, be sure you are working with the modified Auto data set that has rows containing missing hp entries removed. Make sure you get dim(Auto)=392 9.

First, we will create a new variable called FromUS and add it to our Auto data set:

FromUS=rep(0,392) # Set up a vector of 392 zeroes.
FromUS[Auto$origin==1]=1 # Switch value for US cars.
Auto=data.frame(Auto,FromUS) # Add new variable to Auto.
To set up the commands above, I followed the book's example (from the College data set) on page 56.

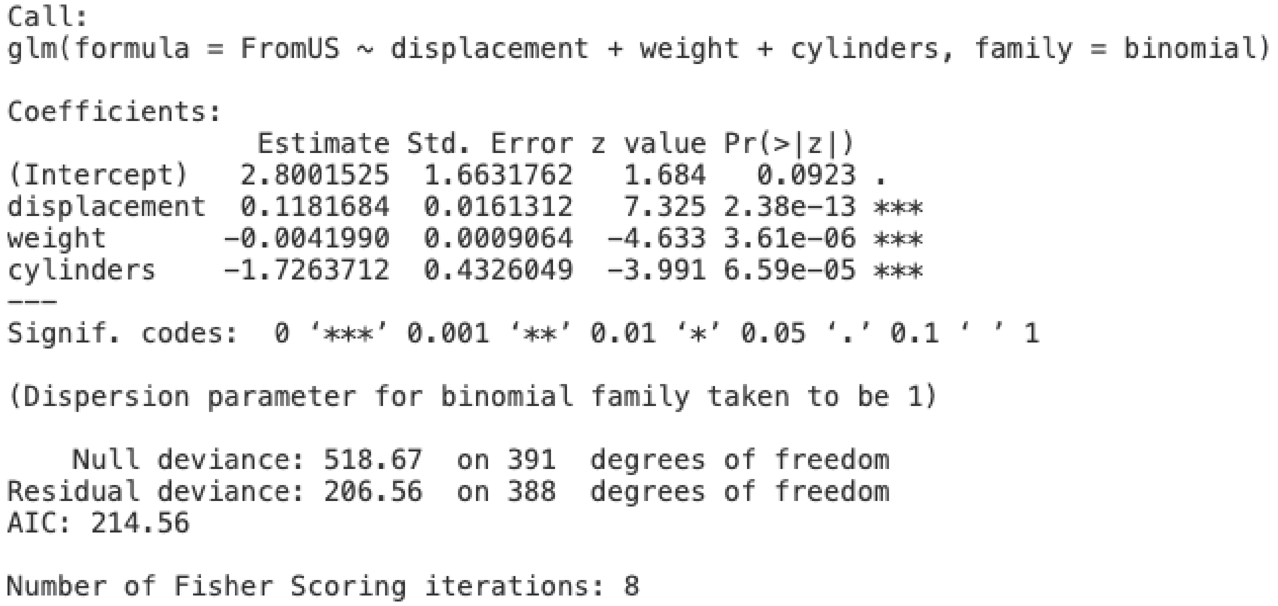
Use the glm command to find a logistic regression model for FromUS as a function of mpg, cylinders, displacement, horsepower, weight, and acceleration. You will see that some of the predictors are more significant than others. Remove insignificant predictors one at a time until all remaining predictors have three-star (***) significance. (My final model has 3 predictors.) Then create the confusion matrix to see how well the model works. In your report, include the summary of your final model, its confusion matrix, and the success rate.

Note that we are testing this model on its training set. This partially explains why the model does so well.

CODE 1:

```
Auto = read.csv(file="Auto.csv",na.strings="?", stringsAsFactors = T)
Auto= na.omit(Auto)
dim(Auto)
FromUS=rep(0,392)
FromUS[Auto$origin==1]=1
Auto=data.frame(Auto,FromUS)
displacement=Auto$displacement
acceleration=Auto$acceleration
horsepower=Auto$horsepower
weight=Auto$weight
mpg=Auto$mpg
cylinders=Auto$cylinders
LogMod=glm(FromUS ~ displacement + weight + cylinders, family=binomial)
summary(LogMod)
```

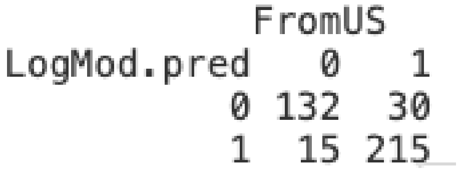
```
392 · 9
```



```
Call:
glm(formula = FromUS ~ displacement + weight + cylinders, family = binomial)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.8001525  1.6631762   1.684   0.0923 .
displacement  0.1181684  0.0161312   7.325 2.38e-13 ***
weight       -0.0041990  0.0009064  -4.633 3.61e-06 ***
cylinders    -1.7263712  0.4326049  -3.991 6.59e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 518.67  on 391  degrees of freedom
Residual deviance: 206.56  on 388  degrees of freedom
AIC: 214.56

Number of Fisher Scoring iterations: 8
```

CODE 2:

```
LogMod.probs=predict(LogMod,type="response")
LogMod.pred = rep(0,392)
LogMod.pred[LogMod.probs > 0.5]=1
table(LogMod.pred,FromUS)
```



```
           FromUS
LogMod.pred   0   1
          0 132  30
          1  15 215
```

Success Rate= (132+215)/392 = 0.885204081632653