Matthew Reynolds
Oren Mangoubi
DS 4635
19 April 2023

Homework #3

Question 1 (Exercise 6):

Code

Exercise 6

Matthew Reynolds

2023-04-19

Question:
Suppose we collect data for a group of students in a statistics class with variables X1 = hours studied, X2 = undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, β^0 = –6, β^1 = 0.05, β^2 = 1.

a. Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

Answer:
Since we're given: $\hat{\beta_0}$ = –6, $\hat{\beta_1}$ = 0.05, $\hat{\beta_2}$ = 1, and X_1 = 40 and X_2 = 3.5. We simply plug the values in to our equation and solve.
$-6 + 0.05(40) + 1(3.5)$
Which results in:
-0.5
We plug that into our next equation: $\frac{e^k}{1+e^k}$
So:
$\frac{e^{-0.5}}{1+e^{-0.5}}$

Which results in:
37.75% chance of getting an A in the class given the 40 hours of studying and a undergrad GPA of 3.5

b. How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

Answer: We take our previous equation: $\frac{e^k}{1+e^k}$ and set it equal to 1/2. $\frac{e^k}{1+e^k} = \frac{1}{2}$

$1 + e^k = 2e^k$

$1 = e^k$

$ln(1) = ln(e^k)$

$0 = k$

So with our new value of k, we set our other equation for the Betas equal to it.
$-6 + 0.05(X_1) + 1(3.5) = 0$

$0 = k$

So with our new value of k, we set our other equation for the Betas equal to it.
$-6 + 0.05(X_1) + 1(3.5) = 0$

$-2.5 + 0.05(X_1 = 0$

$0.05(X_1) = 2.5$

$X_1 = \frac{2.5}{0.05}$

$X_1 = 50$

Therefore it would take our student from part A 50 hours to achieve a 50% probability of getting an A in their course.

Question 2 (Exercise 10):

Exercise 10

Matthew Reynolds

2023-04-19

Question:

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

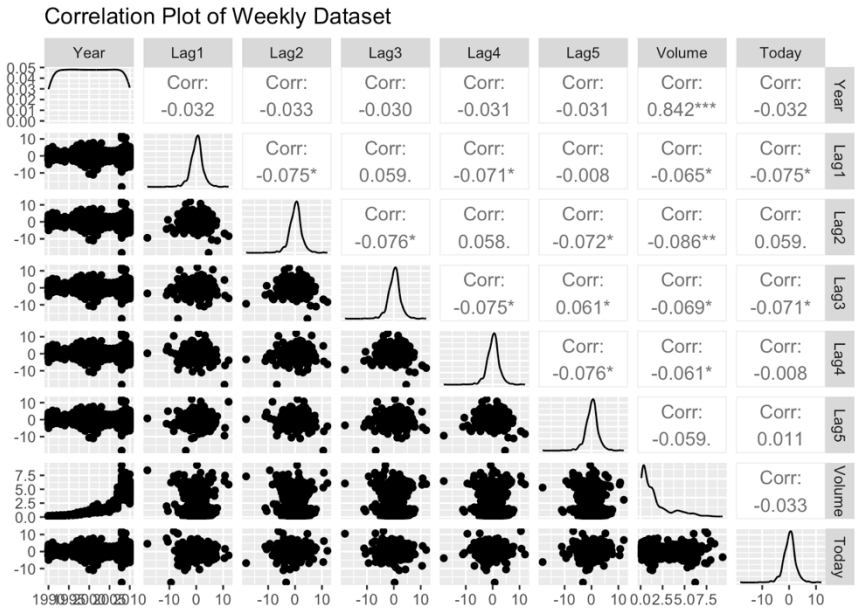
A)

```
summary(Weekly)
```

```
##       Year           Lag1               Lag2               Lag3         
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
##       Lag4               Lag5              Volume            Today         
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540  
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410  
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499  
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050  
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260  
##  Direction 
##  Down:484  
##  Up  :605  
##            
##            
##            
##
```

```
ggpairs(Weekly,
        columns = 1:8,
        title = "Correlation Plot of Weekly Dataset",
        upper = list(wrap(continuous = "cor", funcVal = cor, method = "pearson")),
```

```
ggpairs(Weekly,
        columns = 1:8,
        title = "Correlation Plot of Weekly Dataset",
        upper = list(wrap(continuous = "cor", funcVal = cor, method = "pearson")),
        lower = list(wrap(continuous = "smooth", funcVal = smooth, method = "lm")),
        diag = list(continuous = "density"),
        axisLabels = "show"
        )
```



I would say there seems to be a pattern between our Year and Volume which looks like it increases quadraticly. Otherwise, everything else seems rather Blob-like.

B)

```
fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

It would appear that Lag2 is the only one to have a P-Value lower than 0.05, so I would say it's the only significant one.

C)

```
predicted <- predict(fit, type = "response")
predicted_direction <- ifelse(predicted > 0.5, "Up", "Down")
confusion_df <- data.frame(Actual = Weekly$Direction, Predicted = predicted_direction)
confusion_matrix <- t(table(confusion_df))
print(confusion_matrix)
```

```
##          Actual
## Predicted Down  Up
##      Down   54  48
##      Up    430 557
```

```
overall_accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Overall Fraction of Correct Predictions:", overall_accuracy, "\n")
```

```
## Overall Fraction of Correct Predictions: 0.5610652
```

My True Positive was 557, meaning it predicted UP correctly. My False Positive was 430, meaning it predicted UP incorrectly. My True Negative was 54 meaning it predicted Down correctly. And My False Negative was 48 meaning it predicted Down incorrectly. My model was correct for ~56% of its predictions.

D)

```
train_data <- Weekly[Weekly$Year <= 2008, ]
test_data <- Weekly[Weekly$Year > 2008, ]
model <- glm(Direction ~ Lag2, data = train_data, family = binomial)
predictions <- predict(model, newdata = test_data, type = "response")
predicted_directions <- ifelse(predictions > 0.5, "Up", "Down")
confusion_matrix <- table(test_data$Direction, predicted_directions)
overall_accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
confusion_matrix <- t(confusion_matrix)
cat("Confusion Matrix:\n")
```

```
## Confusion Matrix:
```

```
print(confusion_matrix)
```

```
##
## predicted_directions Down Up
##                 Down    9  5
##                 Up     34 56
```

```
cat("\nOverall Fraction of Correct Predictions:", overall_accuracy)
```

```
##
## Overall Fraction of Correct Predictions: 0.625
```

The model correctly predicts the weekly trend only 62.5% of the time.

l)

```
train_data <- Weekly[Weekly$Year < 2009, ]
test_data <- Weekly[Weekly$Year >= 2009, ]

predictors <- c("Lag2", "Lag1", "Lag3", "Lag4", "Lag5", "Volume")

# Initialize best_model, best_variables, and best_accuracy
best_model <- NULL
best_variables <- NULL
best_accuracy <- 0

#brute force it
for (i in 1:length(predictors)) {
  for (j in 1:length(predictors)) {
    if (i == j) next

    predictors_current <- c(predictors[i], predictors[j])
    model <- glm(Direction ~ ., data = train_data[, c(predictors_current, "Direction")], family = "binomial")

    predicted_directions <- ifelse(predict(model, test_data) > 0.5, "Up", "Down")
    confusion_matrix <- table(predicted_directions, test_data$Direction)
    overall_accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)

    if (overall_accuracy > best_accuracy) {
      best_model <- model
      best_confusion_matrix <- t(confusion_matrix)
      best_accuracy <- overall_accuracy
      best_variables <- paste(predictors_current, collapse = ", ")
    }
  }
}

# output
cat("Best Model:\n")
```

```
## Best Model:
```

```
print(best_model)
```

```
print(best_model)
```

```
##
## Call:  glm(formula = Direction ~ ., family = "binomial", data = train_data[,
##     c(predictors_current, "Direction")])
##
## Coefficients:
## (Intercept)         Lag2         Lag5
##     0.20741      0.05652     -0.02973
##
## Degrees of Freedom: 984 Total (i.e. Null);  982 Residual
## Null Deviance:       1355
## Residual Deviance: 1349  AIC: 1355
```

```
cat("Best Confusion Matrix:\n")
```

```
## Best Confusion Matrix:
```

```
print(best_confusion_matrix)
```

```
##       predicted_directions
##        Down Up
##   Down   41  2
##   Up     55  6
```

```
cat("Variables used in the best-performing model:", best_variables, "\n")
```

```
## Variables used in the best-performing model: Lag2, Lag5
```

```
cat("Best Overall Fraction of Correct Predictions:", best_accuracy)
```

```
## Best Overall Fraction of Correct Predictions: 0.4519231
```

Question 3 (Exercise 2):

Exercise2

Matthew Reynolds

2023-04-19

Question:

We will now derive the probability that a given observation is part of a bootstrap sample. Suppose that we obtain a bootstrap sample from a set of n observations.

a. What is the probability that the first bootstrap observation is not the jth observation from the original sample? Justify your answer.

The probability that the first bootstrap observation is not the jth observation from the origional sample is found with the equation: $\frac{n-1}{n}$.

This is because each observation has an equal probability of being selected, and the jth observation has a probability of $\frac{1}{n}$ *of being selected.*

b. What is the probability that the second bootstrap observation is not the jth observation from the original sample?

The probability that the second bootstrap observation is not the jth observation is the same as above, $\frac{n-1}{n}$

c. Argue that the probability that the jth observation is not in the bootstrap sample is (1 – 1/n)n.

Since $\frac{n-1}{n} = 1 - \frac{1}{n}$ *and that we bootstrap with* n *draws, therefore the probability that it is not selected in a single bootstrap sample, is raised to the power of n. So our new equation is* $(1 - \frac{1}{n})^n$

d. When n = 5, what is the probability that the jth observation is in the bootstrap sample?

When $n = 5$ *the probability that the jth observation is in the bootstrap sample is* $(1 - \frac{1}{5})^5$ *which equals to 67.23%.*

e. When n = 100, what is the probability that the jth observation is in the bootstrap sample?

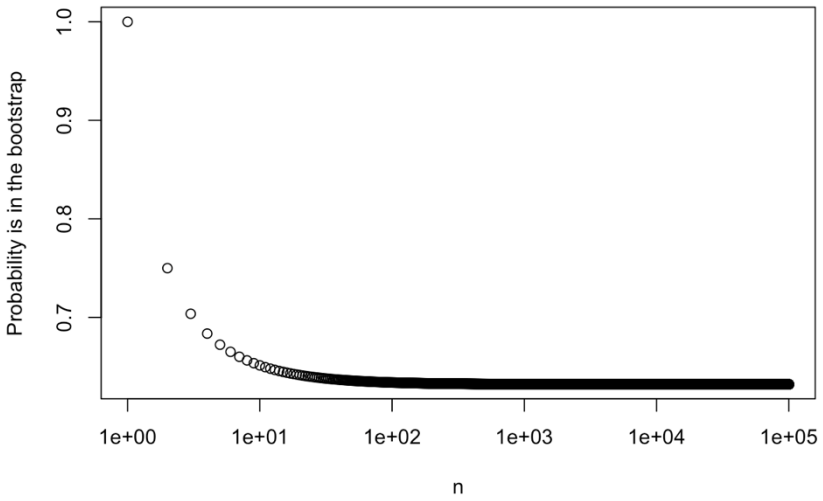
When $n = 100$ *the probability that the jth observation is in the bootstrap sample is* $(1 - \frac{1}{100})^{100}$ *which equals to 63.39%%.*

f. When n = 10, 000, what is the probability that the jth observa- tion is in the bootstrap sample?

When $n = 10,000$ *the probability that the jth observation is in the bootstrap sample is* $(1 - \frac{1}{10000})^{10000}$ *which equals to 63.21%%.*

g. Create a plot that displays, for each integer value of n from 1 to 100,000, the probability that the jth observation is in the bootstrap sample. Comment on what you observe.

```
x=seq(1,100000)
y=sapply(x,function(n){1-((1-(1/n))^n)})
plot(x,y,xlab="n",ylab="Probability is in the bootstrap",log="x")
```



The probability quickly drops to roughly 63% at a bit over 100 and plateaus there.

The probability quickly drops to roughly 63% at a bit over 100 and plateaus there.

h. We will now investigate numerically the probability that a boot- strap sample of size n = 100 contains the jth observation. Here j = 4. We repeatedly create bootstrap samples, and each time we record whether or not the fourth observation is contained in the bootstrap sample.

```
store=rep(NA, 10000)
for(i in 1:10000){
  store[i]=sum(sample(1:100, rep=TRUE)==4)>0
}
mean(store)
```

```
## [1] 0.6326
```

This made a list of 10,000. Then it sampled 100 digist from it and 64.44% of the time, j was in the sample.

Question 4 (Exercise 8):

Exercise8

Matthew Reynolds

2023-04-19

Question:

We will now perform cross-validation on a simulated data set. (a) Generate a simulated data set as follows:

```
set.seed(1)
x=rnorm(100)
y=x-2*x^2+rnorm(100)
```

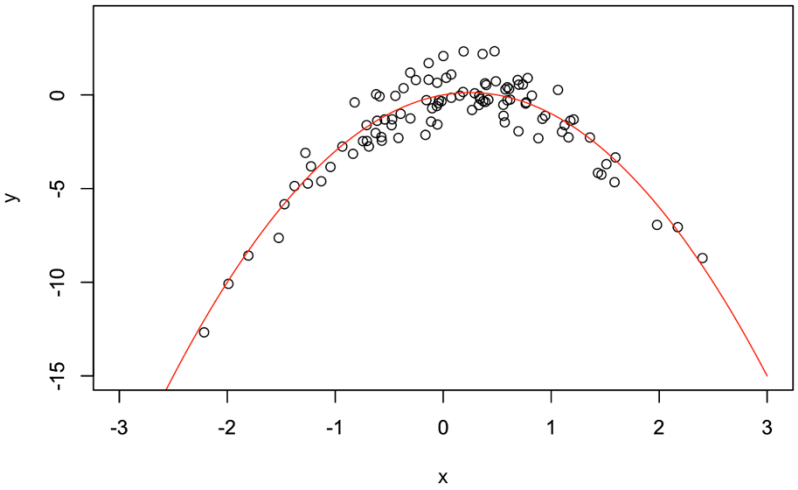
In this data set, what is n and what is p? Write out the model used to generate the data in equation form.

N is the number of observations, 100, and p is the number of predictor variables, 1. The equation for this is:

$$y = \beta_0 + \beta_1 * X + 2 * X^2 + \varepsilon$$

b. Create a scatterplot of X against Y . Comment on what you find.

```
plot(x, y, xlim = range(-3:3), ylim = range(-15:4))
par(new = TRUE)
curve(x - 2*x^2, from = -3, to = 3, xlim = range(-3:3), ylim = range(-15:4), xlab = "", ylab = "", col = "red")
```



X has a negative quadratic relationship to Y. Peaking around x = 0.2, y = 0.

c. Set a random seed, and then compute the LOOCV errors that result from fitting the following four models using least squares:

i. $Y = \beta0 + \beta1X + \varepsilon$
ii. $Y = \beta0 + \beta1X + \beta2X2 + \varepsilon$
iii. $Y = \beta0 + \beta1X + \beta2X2 + \beta3X3 + \varepsilon$
iv. $Y = \beta0 + \beta1X + \beta2X2 + \beta3X3 + \beta4X4 + \varepsilon$. Note you may find it helpful to use the data.frame() function to create a single data set containing both X and Y .

```
set.seed(499)
xy.data = data.frame("pred" = x, "resp" = y)

deg1.fit = glm(resp ~ pred, data = xy.data)
deg1.err = cv.glm(xy.data, deg1.fit)
cat("The LOOCV of Option I is: ", deg1.err$delta[1], "\n")
```

```
## The LOOCV of Option I is:  7.288162
```

```
deg2.fit = glm(resp ~ poly(pred, 2), data = xy.data)
deg2.err = cv.glm(xy.data, deg2.fit)
cat("The LOOCV of Option II is: ", deg2.err$delta[1], "\n")
```

```
## The LOOCV of Option II is:  0.9374236
```

```
deg3.fit = glm(resp ~ poly(pred, 3), data = xy.data)
deg3.err = cv.glm(xy.data, deg3.fit)
cat("The LOOCV of Option III is: ", deg3.err$delta[1], "\n")
```

```
## The LOOCV of Option III is:  0.9566218
```

```
deg4.fit = glm(resp ~ poly(pred, 4), data = xy.data)
deg4.err = cv.glm(xy.data, deg4.fit)
cat("The LOOCV of Option IV is: ", deg4.err$delta[1], "\n")
```

```
## The LOOCV of Option IV is:  0.9539049
```

d. Repeat (c) using another random seed, and report your results. Are your results the same as what you got in (c)? Why?

```
set.seed(3)
xy.data = data.frame("pred" = x, "resp" = y)

deg1.fit = glm(resp ~ pred, data = xy.data)
deg1.err = cv.glm(xy.data, deg1.fit)
cat("The LOOCV of Option I is: ", deg1.err$delta[1], "\n")
```

```
## The LOOCV of Option I is:  7.288162
```

```
deg2.fit = glm(resp ~ poly(pred, 2), data = xy.data)
deg2.err = cv.glm(xy.data, deg2.fit)
cat("The LOOCV of Option II is: ", deg2.err$delta[1], "\n")
```

```
## The LOOCV of Option II is:  0.9374236
```

```
deg3.fit = glm(resp ~ poly(pred, 3), data = xy.data)
deg3.err = cv.glm(xy.data, deg3.fit)
cat("The LOOCV of Option III is: ", deg3.err$delta[1], "\n")
```

```
## The LOOCV of Option III is:  0.9566218
```

```
deg4.fit = glm(resp ~ poly(pred, 4), data = xy.data)
deg4.err = cv.glm(xy.data, deg4.fit)
cat("The LOOCV of Option IV is: ", deg4.err$delta[1], "\n")
```

```
## The LOOCV of Option IV is:  0.9539049
```

The LOOCV are the exact same for both of my seeds. This is because LOOCV doesn't involve any randomness which could change the results.

e. Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

The $y = \beta_0 + \beta_1 * X + 2 * X^2 + \varepsilon$, aka the second option had the lowest LOOCV error of the 4. This didn't surprise me because this is what we used to compute the original data.

Question 5 (Exercise 1):

Exercise 1

Matthew Reynolds

2023-04-19

Question:

1. We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2, . . . , p predictors. Explain your answers:

a. Which of the three models with k predictors has the smallest training RSS?

The model with k predictors that has the smallest training RSS is the model that best fits the the predictors to the data. Meaning simply it contains all predictors with the greatest flexibility so that it could be fit extremely closely to the data.

b. Which of the three models with k predictors has the smallest test RSS?

We wouldn't exactly know which one of the three models would have the smallest test RSS without testing each of them. I'd say generally, medium flexibility/complexitiy is better, but its impossible to know without actually seeing data or their performance.

c. True or False:

i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.

TRUE. The model with (k+1) a variable is added only if it improves the performance. So this is taking the predictors for k models and just adding one more.

ii. The predictors in the k-variable model identified by back- ward stepwise are a subset of the predictors in the (k + 1)- variable model identified by backward stepwise selection.

TRUE. Opposite the previous, at each step a step is removed if it doesnt contribute to the performance. So the model with k-predictors is obtained by removing one of the predictors from the (k+1) model.

iii. The predictors in the k-variable model identified by back- ward stepwise are a subset of the predictors in the (k + 1)- variable model identified by forward stepwise selection.

FALSE. There is no direct link between the two because the criteria for adding/removing are different for the two.

iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by backward stepwise selection.

FALSE. Same as above there's no direct link between the two because the criteria for addding/removing are different for the two.

v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k + 1)-variable model identified by best subset selection.

FALSE. The predictors included in the k-variable model identified by the best subset selection isn't necessarily a subset of the predictors in the (k+1)-variable model identified by best subset selection. This is because the best subset considers all possible subsets and forward/backward only consider one at a time.

s