FE590. Assignment #2.

John-Craig Borman

2018-10-08

Instructions

In this assignment, you should use R markdown to answer the questions below. Simply type your R code into embedded chunks as shown above. When you have completed the assignment, knit the document into a PDF file, and upload both the .pdf and .Rmd files to Canvas.

```
CWID = 10402229 #Place here your Campus wide ID number, this will personalize
#your results, but still maintain the reproduceable nature of using seeds.
#If you ever need to reset the seed in this assignment, use this as your seed
#Papers that use -1 as this CWID variable will earn 0's so make sure you
change
#this value before you submit your work.
personal = CWID %% 10000
set.seed(personal)#You can reset the seed at any time in your code, but
please always set it to this seed.

library(magrittr) # Piping operators
```

Question 1

Use the Auto data set from the textbook's website. When reading the data, use the options as.is = TRUE and na.strings="?". Remove the unavailable data using the na.omit() function.

```
df = read.table("Auto.data", as.is = T, na.strings = "?", header = T)
df = na.omit(df)
```

1. List the names of the variables in the data set.

```
names(df)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "year"         "origin"
## [9] "name"
```

2. The columns origin and name are unimportant variables. Create a new data frame called cars that contains none of these unimportant variables

```
cars = df[, !(colnames(df) %in% c("origin", "name"))]
names(cars)

## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"       "acceleration" "year"
```

3. What is the range of each quantitative variable? Answer this question using the range() function with the sapply() function e.g., sapply(cars, range). Print a simple table of the ranges of the variables. The rows should correspond to the variables. The first column should be the lowest value of the corresponding variable, and the second column should be the maximum value of the variable. The columns should be suitably labeled.

```
sapply(cars, range) %>%
  t() %>%
  set_colnames(c("Min", "Max"))

##                Min    Max
## mpg              9   46.6
## cylinders        3    8.0
## displacement    68  455.0
## horsepower      46  230.0
## weight        1613 5140.0
## acceleration     8   24.8
## year            70   82.0
```

4. What is the mean and standard deviation of each variable? Create a simple table of the means and standard deviations.

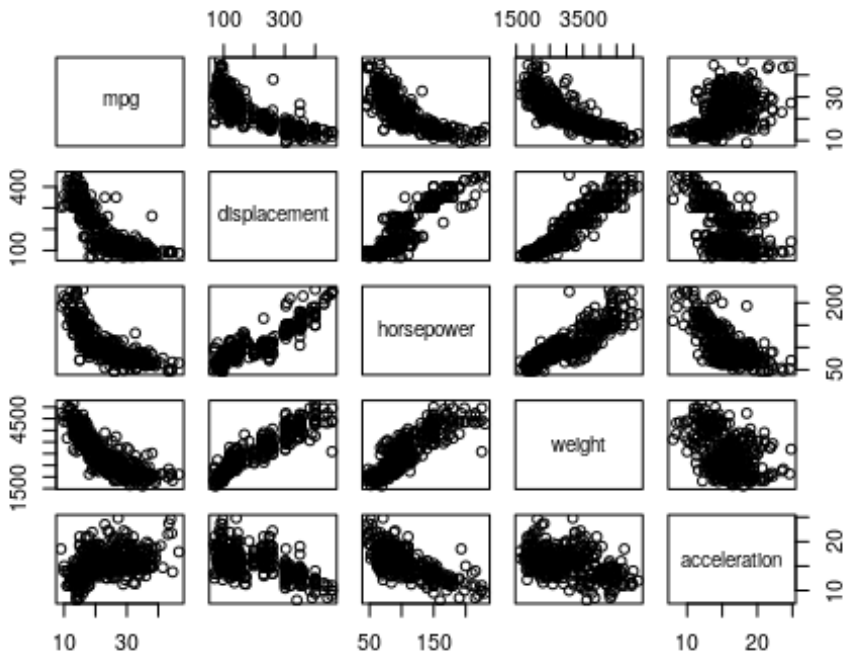
```
means <-
  cars %>%
  sapply(mean)

sds <-
  cars %>%
  sapply(sd)

tbl <-
  rbind(means, sds) %>%
  t() %>%
  set_colnames(c("Mean", "Std Dev"))
```

```
tbl
```

```
##                       Mean     Std Dev
## mpg              23.445918   7.805007
## cylinders         5.471939   1.705783
## displacement   194.411990 104.644004
## horsepower     104.469388  38.491160
## weight        2977.584184 849.402560
## acceleration    15.541327   2.758864
## year            75.979592   3.683737
```

5. Create a scatterplot matrix that includes the variables mpg, displacement, horsepower, weight, and acceleration using the pairs() function.

```
pairs(cars[,c("mpg", "displacement", "horsepower", "weight",
"acceleration")])
```



6. Using the regsubsets function in the leaps library, regress mpg onto

```
library(leaps)

model <- leaps::regsubsets(data = cars, mpg ~ displacement + horsepower +
weight + acceleration)
```

7. Print a table showing what variables would be selected using best subset selection for all predictors (displacement, horsepower, weight, acceleration) up to order 2 (i.e. weight and weight^2).

```
model <- leaps::regsubsets(data = cars,
                       mpg ~ displacement + horsepower + weight +
acceleration + poly(displacement, 2) + poly(horsepower, 2) + poly(weight, 2)
+ poly(acceleration, 2))

## Reordering variables and trying again:

summary(model)$which
```

```
##   (Intercept) displacement horsepower weight acceleration
## 1        TRUE        FALSE      FALSE  FALSE        FALSE
## 2        TRUE        FALSE      FALSE  FALSE        FALSE
## 3        TRUE        FALSE       TRUE   TRUE        FALSE
## 4        TRUE        FALSE       TRUE  FALSE        FALSE
## 5        TRUE         TRUE      FALSE  FALSE         TRUE
## 6        TRUE         TRUE       TRUE   TRUE         TRUE
## 7        TRUE        FALSE       TRUE   TRUE        FALSE
## 8        TRUE         TRUE       TRUE  FALSE         TRUE
##   poly(displacement, 2)1 poly(displacement, 2)2 poly(horsepower, 2)1
## 1                  FALSE                  FALSE                FALSE
## 2                  FALSE                  FALSE                FALSE
## 3                  FALSE                   TRUE                FALSE
## 4                  FALSE                   TRUE                FALSE
## 5                  FALSE                   TRUE                 TRUE
## 6                  FALSE                   TRUE                FALSE
## 7                   TRUE                   TRUE                FALSE
## 8                  FALSE                   TRUE                FALSE
##   poly(horsepower, 2)2 poly(weight, 2)1 poly(weight, 2)2
## 1                FALSE             TRUE            FALSE
## 2                FALSE             TRUE             TRUE
## 3                FALSE            FALSE            FALSE
## 4                 TRUE             TRUE            FALSE
## 5                 TRUE            FALSE            FALSE
## 6                 TRUE            FALSE            FALSE
## 7                 TRUE            FALSE            FALSE
## 8                 TRUE             TRUE             TRUE
##   poly(acceleration, 2)1 poly(acceleration, 2)2
## 1                  FALSE                  FALSE
## 2                  FALSE                  FALSE
## 3                  FALSE                  FALSE
## 4                  FALSE                  FALSE
## 5                  FALSE                  FALSE
## 6                  FALSE                  FALSE
## 7                   TRUE                   TRUE
## 8                  FALSE                   TRUE
```

a. What is the most important variable affecting fuel consumption?

```
summary(model)$which %>% apply(2, sum)
```

```
##             (Intercept)             displacement               horsepower
##                       8                        3                        5
##                  weight             acceleration poly(displacement, 2)1
##                       3                        3                        1
## poly(displacement, 2)2   poly(horsepower, 2)1   poly(horsepower, 2)2
##                       6                        1                        5
##       poly(weight, 2)1       poly(weight, 2)2 poly(acceleration, 2)1
##                       4                        2                        1
## poly(acceleration, 2)2
##                       2
```

Out of all the models constructed, the displacement^2 was included the most often therefore one could argue that the importance of this variable is reflected in its frequency.

b. What is the second most important variable affecting fuel consumption?

Based on the previous notion of importance, horsepower and horsepower^2 are tied for the second most important variable.

c. What is the third most important variable affecting fuel consumption?

The next most important variable is weight^2. Technically this would be the fourth most important variable, given that two variables are tied for second.

Question 2

This exercise involves the Boston housing data set.

1. Load in the Boston data set, which is part of the MASS library in R. The data set is contained in the object Boston. Read about the data set using the command ?Boston. How many rows are in this data set? How many columns? What do the rows and columns represent?

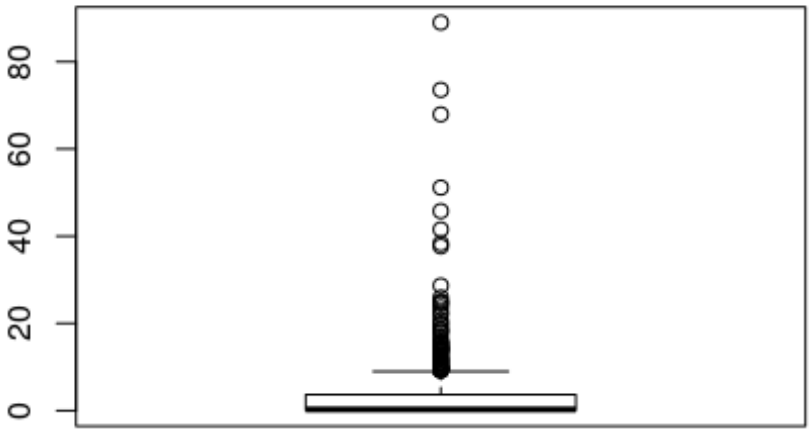
```
library(MASS)
dim(Boston)
```

```
## [1] 506  14
```

The dimensions of the Boston data set are 506 (rows) by 14 (cols) representing the observations (suburbs) and variables, respectively.

2. Do any of the suburbs of Boston appear to have particularly high crime rates?

```
boxplot(Boston$crim)
```



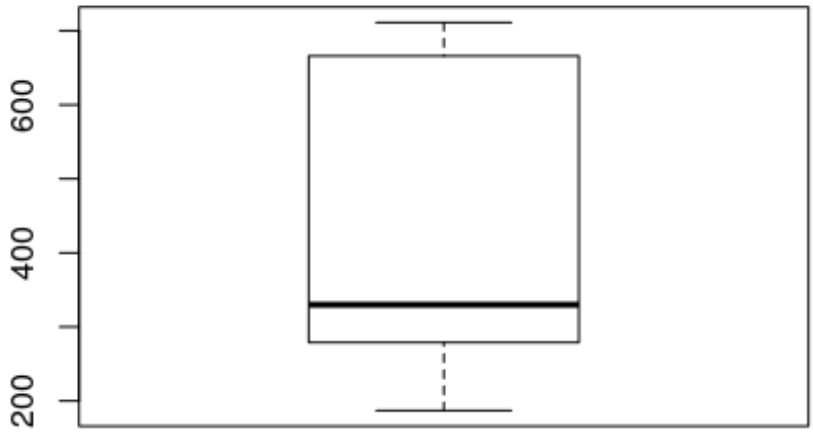

```
Boston$crim %>% is_greater_than(50) %>% sum
```

```
## [1] 4
```

The boxplot above shows a visual representation of the `summary()` function applied to the `crim` variable. There are a number of suburbs (observations) that have crime rates above 50 crimes per capita, specifically 4 suburbs. Note that the number of 50 crimes per capita is an arbitrary number, but to me the rate of 50 crimes per person living in a suburb is a *significantly* high number.

Tax rates?

```
boxplot(Boston$tax)
```



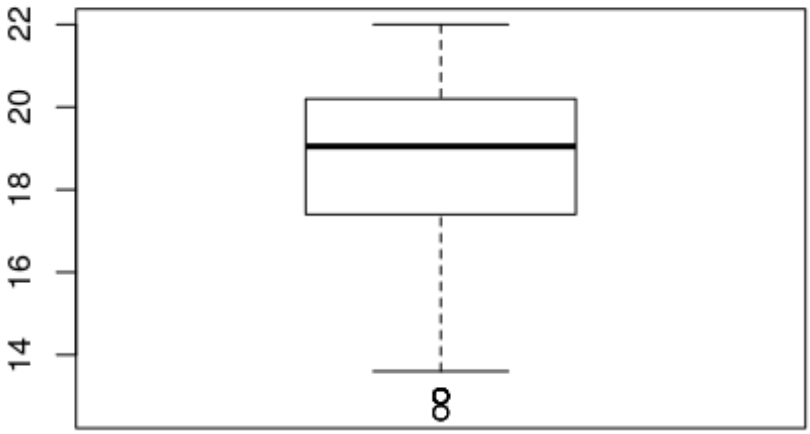

```
Boston$tax %>% is_greater_than(650) %>% sum()
```

```
## [1] 137
```

There are 137 suburbs with tax rates above what is, roughly, the third quantile rate (~650) that can be considered particularly high tax rates

Pupil-teacher ratios?

```
boxplot(Boston$ptratio)
```




```
Boston$ptratio %>% is_greater_than(20) %>% sum
```

```
## [1] 201
```

Using a similar methodology, there are 201 suburbs with a pupil-teacher ratio above 20 (roughly the third quantile). However, as can be seen in the boxplot, there are no extraordinary values in regards to the variable. In my opinion, these pupil-teacher ratios are relatively normal/small.

Comment on the range of each predictor.

Crime Rate - The crime rate has a very sporadic distribution across a range of values from 0 to over 100.

Tax Rate - The tax rate has a very wide distribution ranging from 200 to 700.

Pupil-Teacher Ratio - Like the tax rate, the Pupil-Teacher ration has a wide distribution and ranges from ~12 to 22.

3. How many of the suburbs in this data set bound the Charles river?

```
Boston$chas %>% sum()
```

```
## [1] 35
```

There are 35 along the Charles river.

4. What is the median pupil-teacher ratio among the towns in this data set?

```
Boston$ptratio %>% median
```

```
## [1] 19.05
```

The median pupil-teacher ratio is 19.05

5. In this data set, how many of the suburbs average more than seven rooms per dwelling?

```
Boston %>% subset(rm > 7) %>% nrow()
```

```
## [1] 64
```

There are 64 suburbs that average more than 7 rooms per dwelling.

More than eight rooms per dwelling?

```
Boston %>% subset(rm > 8) %>% nrow()
```

```
## [1] 13
```

There are 13 that average more than 8

Question 3

This question should be answered using the Weekly data set, which is part of the ISLR package. This data contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

1. What does the data represent?

```
library(ISLR)
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       
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202  
##  Median :  0.2380   Median :  0.2340   Median :1.00268  
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462  
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373  
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
```

```
##       Today          Direction
##  Min.   :-18.1950   Down:484
##  1st Qu.: -1.1540   Up  :605
##  Median :  0.2410
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
```

The data represents a time series of financial percentage returns and volume also containing variables indicating a k-day lag, that is the value of the time series k-days ago. Additionally, it also contains a categorical variable that indicates the direction.

2. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

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

```
##
## 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
```

At the 99% level, the second lag is statistically significant

3. Fit a logistic regression model using a training data period from 1990 to 2008, using the predictors from the previous problem that you determined were statistically significant. Test your model on the held out data (that is, the data from 2009 and 2010) and express its accuracy.

```
test <- Weekly[Weekly$Year %in% 2009:2010,]
train <- Weekly[Weekly$Year %in% 1990:2008,]

model.bin <- glm(Direction ~ Lag2, data = train, family = "binomial")
summary(model.bin)

##
## Call:
## glm(formula = Direction ~ Lag2, family = "binomial", data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4

pred <- predict(model.bin, test, type = "response") > 0.50 # Greater than 50%
indicates up
pred.vals <-  rep("Up", length(pred))
pred.vals[!pred] <-  rep("Down", sum(!pred))

act.vals <- test$Direction
tbl <- table(act.vals, pred.vals)

# Confusion Matrix
tbl
```

```
##          pred.vals
## act.vals Down Up
##     Down    9 34
##     Up      5 56
```

```
# CM as %
tbl %>% divide_by(sum(.))
```

```
##          pred.vals
## act.vals       Down         Up
##     Down 0.08653846 0.32692308
##     Up   0.04807692 0.53846154
```

In sum, the model did decent on the test data falling short with a relatively high False Positive rate of 37.78%, meaning that when this model predicted up, 34/90 predictions were wrong. The accuracy ($= \frac{TP+TN}{Total}$) is 62.5%, a bit bettern than predicting a fair coin flip (assuming that directional movements of stocks follow a similar distribution). The misclassification or error rate ($= 1 - Accuracy$) is 37.5%.

```
table(pred.vals) %>% divide_by(sum(.))
```

```
## pred.vals
##      Down        Up
## 0.1346154 0.8653846
```

```
table(act.vals) %>% divide_by(sum(.))
```

```
## act.vals
##      Down        Up
## 0.4134615 0.5865385
```

The binomial model seems to heavily favor upward predictions on the test data relative to downward predictions, whereas the distribution of true values in the test data are somewhat closer a fair coin flip but still skewed towards more "Up"s than "Down"s. This statistic combined with the high Type I Error signifies that this model's accuracy might look better than it actually is simply due to the distribution of the actual outcomes. If we chose "Up" every tome, we would have 58.6% accuracy so the binomial model itself adds about 4% of accuracy to this baseline. Which is not bad, but not very robust at predicting downward movements as only 13.4% of predictions were in this direction.

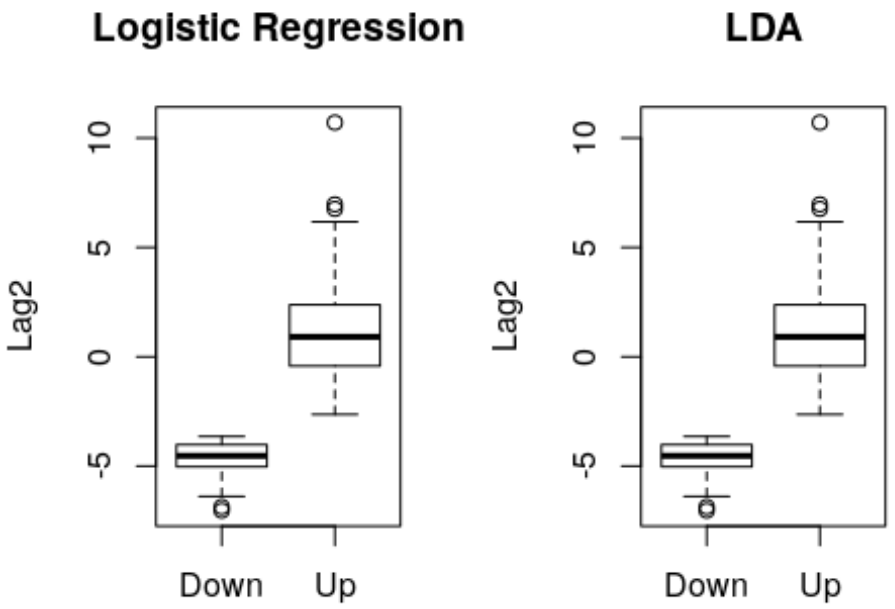
4. Repeat Part 3 using LDA.

```
model.lda <- lda(Direction ~ Lag2, data = train)

lda.vals <- predict(model.lda, test)$class

tbl <- table(act.vals, lda.vals)

# Confusion Matrix
tbl
```

```
##         lda.vals
## act.vals Down Up
##     Down    9 34
##     Up      5 56

# CM as %
tbl %>% divide_by(sum(.))

##         lda.vals
## act.vals       Down         Up
##     Down 0.08653846 0.32692308
##     Up   0.04807692 0.53846154
```

Oddly enough, the LDA model produces the same output as the logistic model. I suspect that this may be the case due to Lag2 being the only predictor and thus both models find the same (or roughly similar) threshold/separation boundary used for classifying as up/down.

```
par(mfrow=c(1,2))
plot(as.factor(pred.vals), test$Lag2, main = "Logistic Regression", ylab =
"Lag2") %>% print

## $stats
##        [,1]   [,2]
## [1,] -6.388 -2.640
## [2,] -5.032 -0.422
## [3,] -4.529  0.911
## [4,] -4.021  2.374
## [5,] -3.646  6.168
##
## $n
## [1] 14 90
##
## $conf
##           [,1]      [,2]
## [1,] -4.955918 0.4453356
## [2,] -4.102082 1.3766644
##
## $out
## [1] -6.868 -7.035  6.760 10.707  6.967
##
## $group
## [1] 1 1 2 2 2
##
## $names
## [1] "Down" "Up"

plot(as.factor(lda.vals), test$Lag2, main = "LDA", ylab = "Lag2") %>% print
```




```
## $stats
##        [,1]   [,2]
## [1,] -6.388 -2.640
## [2,] -5.032 -0.422
## [3,] -4.529  0.911
## [4,] -4.021  2.374
## [5,] -3.646  6.168
##
## $n
## [1] 14 90
##
## $conf
##           [,1]      [,2]
## [1,] -4.955918 0.4453356
## [2,] -4.102082 1.3766644
##
## $out
## [1] -6.868 -7.035  6.760 10.707  6.967
##
## $group
## [1] 1 1 2 2 2
##
## $names
## [1] "Down" "Up"
```

The plots above show that both models separate up/down predictions by a Lag2 of roughly -3.

5. Repeat Part 3 using QDA.

```
model.qda <- qda(Direction ~ Lag2, data = train)

qda.vals <- predict(model.qda, test)$class

tbl <- table(act.vals, qda.vals)

# Confusion Matrix
tbl
```

```
##          qda.vals
## act.vals Down Up
##     Down    0 43
##     Up      0 61
```

```
# CM as %
tbl %>% divide_by(sum(.))
```

```
##          qda.vals
## act.vals      Down        Up
##     Down 0.0000000 0.4134615
##     Up   0.0000000 0.5865385
```

Here, the QDA model produces "Up"-only predictions resulting in a 58.7% accuracy but an even worse off False Positive rate of 41.3%. QDA is suffering here because there is one predictor in a model that should be used on multivariate data.

6. Repeat Part 3 using KNN with K = 1, 2, 3.

```
library(class)

train$Direction %<>% as.numeric()
test$Direction %<>% as.numeric()

knn.pred.1 <- knn(train = train, test = test, train$Direction, k = 1)
knn.pred.2 <- knn(train = train, test = test, train$Direction, k = 2)
knn.pred.3 <- knn(train = train, test = test, train$Direction, k = 3)

tbl1 <- table(act.vals, knn.pred.1)
tbl2 <- table(act.vals, knn.pred.2)
tbl3 <- table(act.vals, knn.pred.3)

tbl1
```

```
##          knn.pred.1
## act.vals  1  2
##     Down 37  6
##     Up   13 48
```

```
tbl2
```

```
##          knn.pred.2
## act.vals  1  2
##     Down 36  7
##     Up   18 43
```

```
tbl3
```

```
##          knn.pred.3
## act.vals  1  2
##     Down 37  6
##     Up    4 57
```

```
tbl1 %>% divide_by(sum(.))
```

```
##          knn.pred.1
## act.vals          1          2
##     Down 0.35576923 0.05769231
##     Up   0.12500000 0.46153846
```

```
tbl2 %>% divide_by(sum(.))
```

```
##          knn.pred.2
## act.vals          1          2
##     Down 0.34615385 0.06730769
##     Up   0.17307692 0.41346154
```

```
tbl3 %>% divide_by(sum(.))
```

```
##          knn.pred.3
## act.vals          1          2
##     Down 0.35576923 0.05769231
##     Up   0.03846154 0.54807692
```

The KNN models appear to produce predictions that are less biased towards choosing up over down or vice versa. All three models have high accuracies of 81.7%, 83.7% and 90.4%, respectively. The KNN models are subject to lower False Positive and False Negative rates relative to the prevous Logistic, LDA and QDA models previously discussed.

7. Which of these methods in Parts 3, 4, 5, and 6 appears to provide the best results on this data?

I would suggest that the KNN model of k=3 provides the best results on the test data. As mentioned previously, there was only one statistically significant predictor (Lag2) so each of the linear models in Parts 3, 4 and 5 did not have much flexibility in terms of information available for predictions and final model fits. Given the accuracy and error rates discussed in each part, the KNN models appear to produce the best results given high accuracy, low error rates and low bias towards any one outcome.

Question 4

Write a function that works in R to give you the parameters from a linear regression on a data set between two sets of values (in other words you only have to do the 2-D case and your output will be the coefficients beta_0 and beta_1). Include in the output the standard error of your variables. You cannot use the lm command in this function or any of the other built in regression models. For example your output could be a 2x2 matrix with the parameters in the first column and the standard errors in the second column. For up to 5 bonus points, format your output so that it displays and operates similar in function to the output of the lm command.(i.e. in a data frame that includes all potentially useful outputs)

A linear regression has the form $y = \hat{\beta}\,\mathbf{X}$ where $\hat{\beta}$ is what we intend to solve for, that is the vector of coefficients corresponding to the predictors in $\mathbf{X}$ plus an extra column to represent the intercept $\hat{\beta}_0$.

To estimate $\hat{\beta}$ we can use the closed form solution (as seen in Week 4 Slides):

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$$

```
estimate_lm <- function(y, X, debug = FALSE){
  # A function that estimates a linear model of betas and std devs
corresponding to the coefficients in a linear regression model

  X %<>% as.matrix()
  y %<>% as.numeric()

  if(!is.vector(y) || !is.matrix(X) || !is.numeric(y) || !is.numeric(X)){
    stop("Arguments 'y' or 'X' are not numeric vectors/matrices")
  }

  X <- cbind(rep(1, nrow(X)), X) # Prepend column of 1's to X

  if(debug) print(head(X)); print(head(y))

  b <- solve(t(X) %*% X) %*% t(X) %*% y

  y.fit <- X %*% b

  sigma_sq <- (1 / (nrow(X) - ncol(X) - 1)) * sum( (y - y.fit)^2)
```

```
  var.b <- solve(t(X) %*% X) * sigma_sq

  std_err.b <- sqrt(diag(var.b))

  Coefficients <- data.frame(Estimate = b, `Std. Error` = std_err.b)

  mdl <- list(Coefficients = Coefficients,
              var.b = var.b)

  return(mdl)
}
```

Compare the output of your function to that of the lm command in R.

```
mdl <- estimate_lm(train$Today, train[, c("Lag1", "Lag2")], TRUE)
```

```
##       Lag1   Lag2
## 1 1  0.816  1.572
## 2 1 -0.270  0.816
## 3 1 -2.576 -0.270
## 4 1  3.514 -2.576
## 5 1  0.712  3.514
## 6 1  1.178  0.712
## [1] -0.270 -2.576  3.514  0.712  1.178 -1.372
```

```
print(mdl$Coefficients)
```

```
##         Estimate Std..Error
##       0.13284367 0.07252842
## Lag1 -0.08472386 0.03199193
## Lag2  0.06446175 0.03199584
```

```
lm(Today ~ Lag1 + Lag2, data = train) %>% summary()
```

```
##
## Call:
## lm(formula = Today ~ Lag1 + Lag2, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.9094  -1.2792   0.1328   1.2124  11.5819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.13284    0.07249   1.833  0.06717 .
## Lag1        -0.08472    0.03198  -2.650  0.00819 **
## Lag2         0.06446    0.03198   2.016  0.04410 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.267 on 982 degrees of freedom
```

```
## Multiple R-squared:  0.01219,    Adjusted R-squared:  0.01017
## F-statistic: 6.058 on 2 and 982 DF,  p-value: 0.002428
```

My `estimate_lm()` function outputs a list containing 2 components:

1. Coefficients - a data frame containing the estimated value and the standard deviation
2. var.b - The covariance matrix of the estimated coefficients

The Coefficients data frame matches the output of the `lm` function in the Coefficients section.

Question 5

Using the Advertising data set (Sales, TV, Radio, Newspaper), do the following:

1. Randomly split the data into two different pieces of roughly equal size.

```
df <- read.csv("Advertising.csv", stringsAsFactors = F, header = T)
df <- df[, -1] # Drop the row numbers
df %>% head()
```

```
##      TV radio newspaper sales
## 1 230.1  37.8      69.2  22.1
## 2  44.5  39.3      45.1  10.4
## 3  17.2  45.9      69.3   9.3
## 4 151.5  41.3      58.5  18.5
## 5 180.8  10.8      58.4  12.9
## 6   8.7  48.9      75.0   7.2
```

```
obs <- 1:nrow(df)

idx1 <- sample(obs, nrow(df)/2, replace = F)
idx2 <- (obs)[!(obs %in% idx1)]

train <- df[idx1,]
test <- df[idx2,]

dim(df)
```

```
## [1] 200   4
```

```
dim(train)
```

```
## [1] 100   4
```

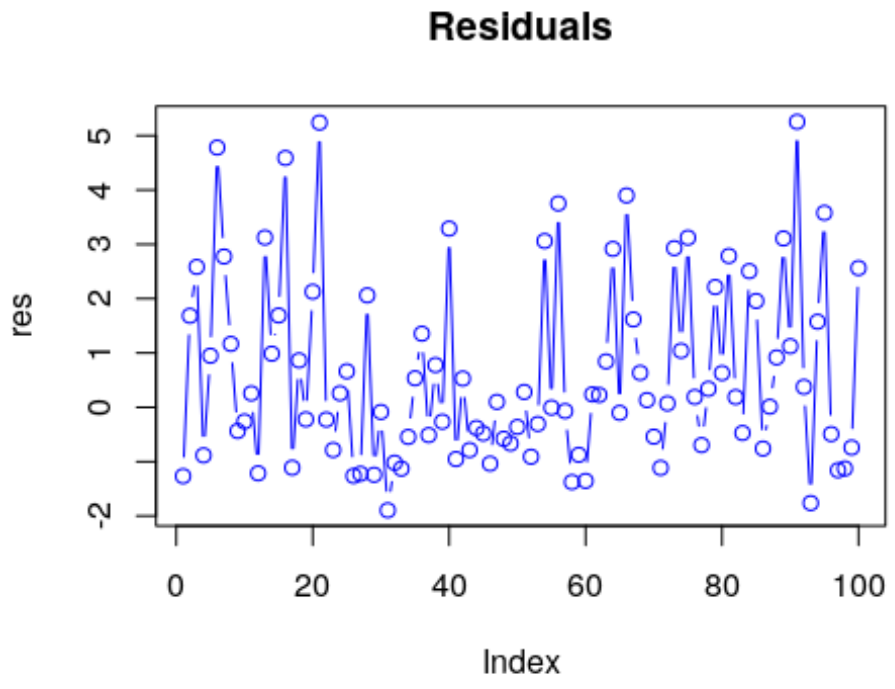
```
dim(test)
```

```
## [1] 100   4
```

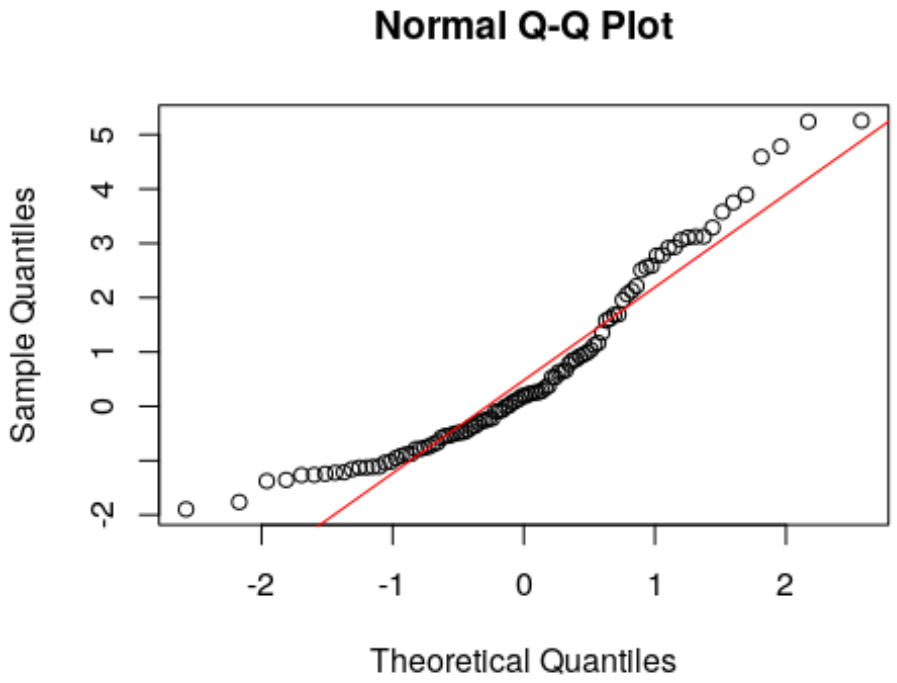
`train` and `test` are randomly split data frames containing observations from `df` (Advertising)

2. Pick one set to run a linear regression to predict sales based on all TV and Radio, and then test your accuracy using the other set

```
model <- lm(sales ~ TV + radio, data = train)
pred.vals <- predict(model, newdata = test)

# Calculate MSE
mean((pred.vals - test$sales)^2)

## [1] 3.195009

# Calculate RMSE
sqrt(mean((pred.vals - test$sales)^2))

## [1] 1.787459

res <- (pred.vals - test$sales)

# Plot Residuals
plot(res, col="blue", type="b", main = "Residuals")
```



```r
# QQ Plot
{qqnorm(res)
qqline(res, col="red")}
```



Overall, the model does work fairly well out of sample and produces residuals that are approximately normal (following the red line), however there are instances where the deviation is large towards the distant quantiles.

3. Repeat the previous problem using all three predictors (including newspaper). What do you determine from this result?

```r
model <- lm(sales ~ TV + radio + newspaper, data = train)
pred.vals <- predict(model, newdata = test)

# Calculate MSE
mean((pred.vals - test$sales)^2)
```

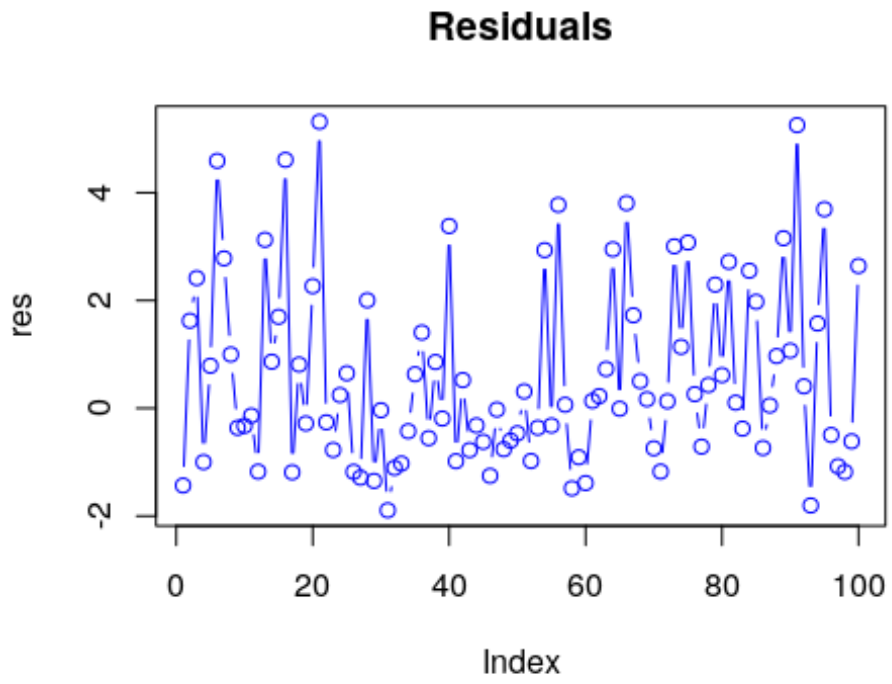
```
## [1] 3.219382
```

```r
# Calculate RMSE
sqrt(mean((pred.vals - test$sales)^2))
```

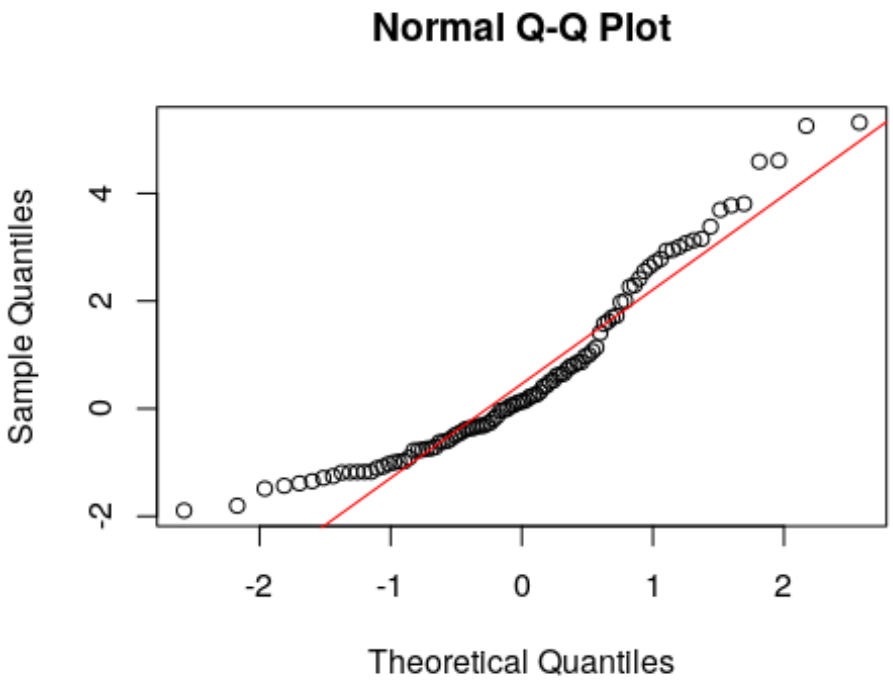
```
## [1] 1.794264
```

```r
res <- (pred.vals - test$sales)
```

```r
# Plot Residuals
plot(res, col="blue", type="b", main = "Residuals")
```




```r
# QQ Plot
{qqnorm(res)
qqline(res, col="red")}
```



The results between the previous two models are quite similar, however the addition of the `newspaper` variable really does not add any additional information that is valuable to the model. In fact, the MSE and RMSE increase as a result. We may want to investigate leaving this variable out as it is not helpful in this out of sample test.

4. Determine the LOOCV error for the linear regression using all three predictors.

```
model <- lm(sales ~ TV + radio + newspaper, data = df)
numer <- sum(resid(model)^2/(1-influence(model)$hat)^2)
denom <- nrow(df)
numer/denom

## [1] 2.9469
```

The formula above comes from the the Generalized Cross Validation expression located in the Week 5 lecture.