ComS 573: Homework #2

Due on February 28, 2014

Professor De Brabanter at 10am

Josh Davis

Problem 1

From ISLR: Chapter 3, Problem 14.

Using a created simulated data, answer the questions regarding simple linear regression.

```
# Ensure consistent values
set.seed(1)

# Create uniform distribution for first input
x1 <- runif(100)

# Normal distribution for second input
x2 <- 0.5 * x1 + rnorm(100)/10

# Our Linear Model
y <- 2 + (2 * x1) + (0.3 * x2) + rnorm(100)
```

Part A

Write out the form of the linear model. What are the regression coefficients?

Solution

The model that we created is just $Y = 2 + 2X_1 + 0.3X_2 + \epsilon$. Thus given we have two predictors, our model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$. This gives us:

$$\beta_0 = 2, \quad \beta_1 = 2, \quad \beta_2 = 0.3$$

Part B

What is the correlation between X_1 and X_2? Create a scatterplot displaying the relationship between the variables.

Solution

We can measure the correlation between the two variables by calculating the covariance. Using our linear model, we can calculate the covariance as follows:

$$\text{Cov}(X_1, X_2) = \text{E}[X_1 X_2] - \text{E}[X_1]\text{E}[X_2]$$

We can calculate the expected value of our random variables by using the **mean()** function in R. This gives us:

```
mean(x1 * x2) - mean(x1) * mean(x2)

## [1] 0.03766
```

Or we can use the **cov()** function in R:

```
cov(x1, x2)

## [1] 0.03804
```

We can scale this value to be dimensionless if we didn't know the magnitude of our values. This value is ρ and is calculated as:

$$\rho = \frac{\text{Cov}(X_1, X_2)}{(\text{Std}X)(\text{Std}Y)}$$

Using R, we can determine that our Std$X$ and the Std$Y$ are:

```
sd(x1)

## [1] 0.2676

sd(x2)

## [1] 0.1702
```

Substituting our values gives us:

$$\begin{aligned}\rho &= \frac{0.0380}{(0.2676)(0.1702)} \\ &= .8343\end{aligned}$$

R can do this as well automatically with the **cor()** function:

```
cor(x1, x2)

## [1] 0.8351
```

Which checks out with our value. This suggests that X_1 and X_2 are pretty colinear. Let's visualize this by using a scatter plot:

```
plot(x1, x2, main = "Correlation of X1 and X2", xlab = "X1", ylab = "X2")
```

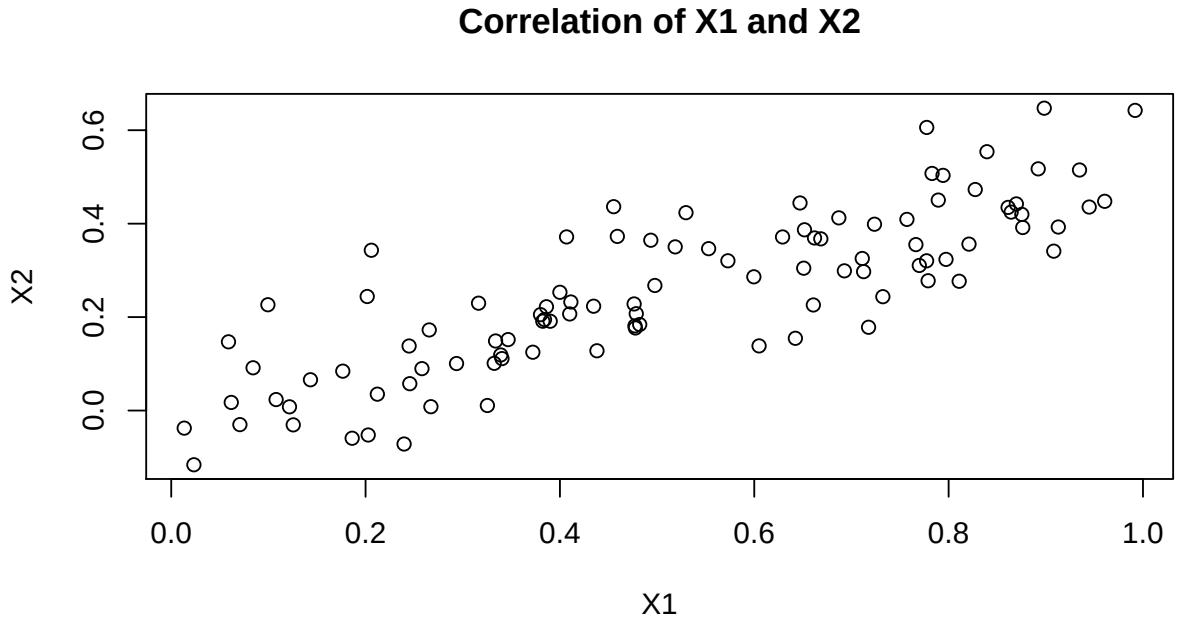


Figure 1: Correlation of given predictors.

Part C

Using the data, fit a least squares regression to predict Y using $x1$ and $x2$. Describe the results obtained. What are $\hat{\beta}_0$ and $\hat{\beta}_1$, and $\hat{\beta}_2$? How do these relate to the true β_0, β_1, β_2? Can you reject the null hypothesis $H_0 : \beta_1 = 0$? How about $H_0 : \beta_2 = 0$?

Solution

First let's fit a linear model to it using R's **lm()** function.

```
data <- data.frame(x1 = x1, x2 = x2, y = y)
fit <- lm(y ~ x1 + x2, data = data)
```

This gives the following estimators:

$$\hat{\beta}_0 = 2.1305, \quad \hat{\beta}_1 = 1.4396, \quad \hat{\beta}_2 = 1.0097$$

The values of our estimators are off by a wee bit but pretty close.

If we look at the significance of our estimated values, we get the following:

```
summary.lm(fit)

##
## Call:
## lm(formula = y ~ x1 + x2, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.130      0.232    9.19  7.6e-15 ***
## x1             1.440      0.721    2.00    0.049 *
## x2             1.010      1.134    0.89    0.375
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.06 on 97 degrees of freedom
## Multiple R-squared:  0.209,Adjusted R-squared:  0.193
## F-statistic: 12.8 on 2 and 97 DF,  p-value: 1.16e-05
```

Taking this into account, our p-value for X_1 is then 0.049. This is just barely under the standard of 0.05. Thus we can consider it significant.

This means we can reject $H_0 : \beta_1 = 0$ and thus can accept the alternative hypothesis. Our conclusion is that X_1 does have an effect on our predictor.

However the p-value for X_2 is 0.375. This means we can't reject the $H_0 : \beta_2 = 0$ and it isn't apparent that X_2 has a significant effect on our predictor.

Problem 1 continued on next page...

Part D

Now fit a least squares regression to predict Y using only $x1$. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

Solution

Fitting only $x1$ gives us:

```
fit.x1 <- lm(y ~ x1, data = data)
```

This gives us the estimator $\hat{\beta}_1 = 1.9759$. Looking at the significance of this estimator:

```
summary.lm(fit.x1)

##
## Call:
## lm(formula = y ~ x1, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8950 -0.6687 -0.0779  0.5922  2.4556
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.112      0.231    9.15  8.3e-15 ***
## x1             1.976      0.396    4.99  2.7e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.06 on 98 degrees of freedom
## Multiple R-squared:  0.202,Adjusted R-squared:  0.194
## F-statistic: 24.9 on 1 and 98 DF,  p-value: 2.66e-06
```

Since our t-statisticis large enough, our accompanying p-value is small (< 0.05) and thus it means we can reject the $H_0 : \beta_1 = 0$ for this model.

Part E

Now fit a least squares regression to predict Y using only $x2$. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

Solution

Fitting only $x2$ gives us:

```
fit.x2 <- lm(y ~ x2, data = data)
```

This gives us the estimator $\hat{\beta}_1 = 2.8996$. Looking at the significance of this estimator:

```
summary.lm(fit.x2)

##
## Call:
```

```
## lm(formula = y ~ x2, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.627 -0.752 -0.036  0.724  2.449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.390      0.195   12.26  < 2e-16 ***
## x2             2.900      0.633    4.58  1.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.07 on 98 degrees of freedom
## Multiple R-squared:  0.176,Adjusted R-squared:  0.168
## F-statistic:   21 on 1 and 98 DF,  p-value: 1.37e-05
```

Just like our previous linear model, the t-statisticis large and thus our p-value is small enough (< 0.05) and we can reject the $H_0 : \beta_1 = 0$.

Part F

Do the results obtained in (c-e) contradict each other? Explain your answer.

Solution

No, they don't contradict each other. In this case, we know from part (b) that our two predictors are correlated. The combined model, part (c), can't take this into account. Each regression coefficient β_j, estimates the expected change in Y per unit change in X_j with all other predictors held fixed.

Since changing X_1 affects X_2, this changes our combined model. Thus our models from part (d-e) are a better reflection of how they affect our response variable, Y.

This is one of the "woes" of interpreting regression coefficients according to *Data Analysis & Regression* by Mosteller and Tukey, 1977.

Part G

Now suppose we obtain one additional observation, which was unfortunately mismeasured.

```
new.x1 <- c(x1, 0.1)
new.x2 <- c(x2, 0.8)
new.y <- c(y, 6)
new.data <- data.frame(x1 = new.x1, x2 = new.x2, y = new.y)
```

Re-fit the lienar models from (c-e) using this new data. What effects does this new observation have on each of the models? In each model, is this observation an outlier? A high leverage point? Both? Explain your answers and make suitable plots.

Solution

Fitting a combined linear model to it from part (c) gives us:

```
new.fit <- lm(y ~ x1 + x2, data = new.data)
```

This gives the following estimators:

$$\hat{\beta}_0 = 2.2267, \quad \hat{\beta}_1 = 0.5394, \quad \hat{\beta}_2 = 2.5146$$

The significance of these estimators is then:

```
summary.lm(new.fit)

##
## Call:
## lm(formula = y ~ x1 + x2, data = new.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7335 -0.6932 -0.0526  0.6638  2.3062
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.227      0.231    9.62  7.9e-16 ***
## x1             0.539      0.592    0.91   0.3646
## x2             2.515      0.898    2.80   0.0061 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.07 on 98 degrees of freedom
## Multiple R-squared:  0.219,Adjusted R-squared:  0.203
## F-statistic: 13.7 on 2 and 98 DF,  p-value: 5.56e-06
```

This changed from our previous model because only our X_1 estimate was significant but now it isn't anymore. Instead the X_2 estimator is significant.

Fitting a single linear model for x_1 like from part (d) we get:

```
new.fit.x1 <- lm(y ~ x1, data = new.data)
```

This gives us the estimator $\hat{\beta}_1 = 1.7657$. Looking at the significance of this estimator:

```
summary.lm(new.fit.x1)

##
## Call:
## lm(formula = y ~ x1, data = new.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.890 -0.656 -0.091  0.568  3.567
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.257      0.239    9.44  1.8e-15 ***
## x1             1.766      0.412    4.28  4.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.11 on 99 degrees of freedom
## Multiple R-squared:  0.156,Adjusted R-squared:  0.148
## F-statistic: 18.3 on 1 and 99 DF,  p-value: 4.29e-05
```

This hasn't changed our single regression model for X_1. The estimator was significant before and after the added data point.

Now fitting a single linear model for x_2 like from part (e) we get:

```
new.fit.x2 <- lm(y ~ x2, data = new.data)
```

This gives us the estimator $\hat{\beta}_1 = 3.119$. Looking at the significance of this estimator:

```
summary.lm(new.fit.x2)

##
## Call:
## lm(formula = y ~ x2, data = new.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.647 -0.710 -0.069  0.727  2.381
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.345      0.191   12.26  < 2e-16 ***
## x2             3.119      0.604    5.16  1.3e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.07 on 99 degrees of freedom
## Multiple R-squared:  0.212,Adjusted R-squared:  0.204
## F-statistic: 26.7 on 1 and 99 DF,  p-value: 1.25e-06
```

This hasn't changed our single regression model for X_2. The estimator was significant before and after the added data point.

To determine if the added points are outliers or leverage points, let's take a look at graphs for each new model:

X_1 and X_2 Model

```
par(mfrow = c(2, 2))
plot(new.fit)
```

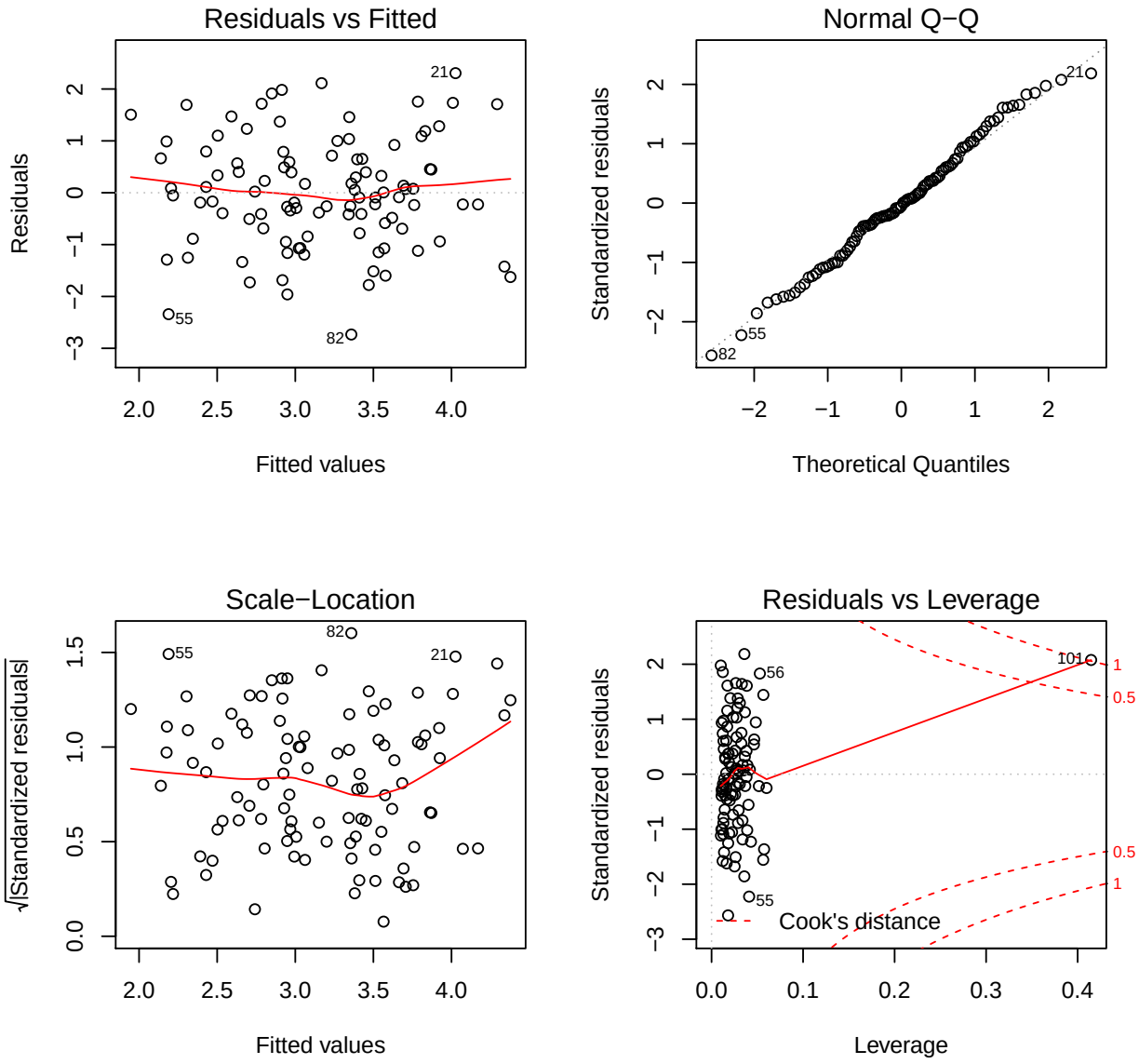


Figure 2: X_1 and X_2 Plots.

Interpretation

Based on these plots, the point doesn't look like an outlier but it does look like a leverage point. This is because it reaches high levels of Cook's Distance on the last plot.

X_1 Model

```
par(mfrow = c(2, 2))
plot(new.fit.x1)
```

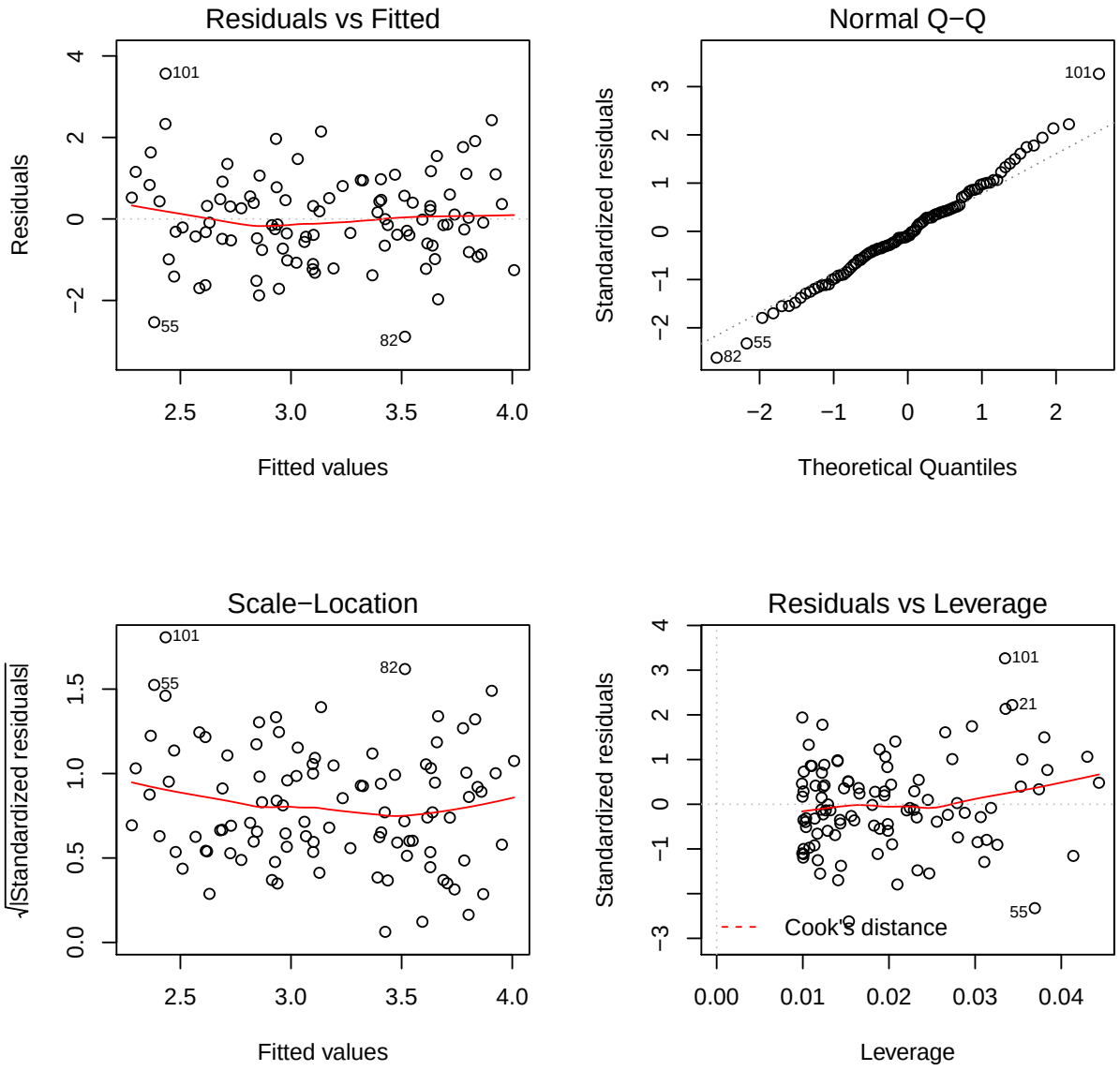


Figure 3: X_1 Plots.

Interpretation

Based on these plots, the value of the standardized residual in the Normal Q-Q plot makes it appear that the point that was added makes it an outlier but not a leverage point because of the distance from the line.

X_2 Model

```
par(mfrow = c(2, 2))
plot(new.fit.x2)
```

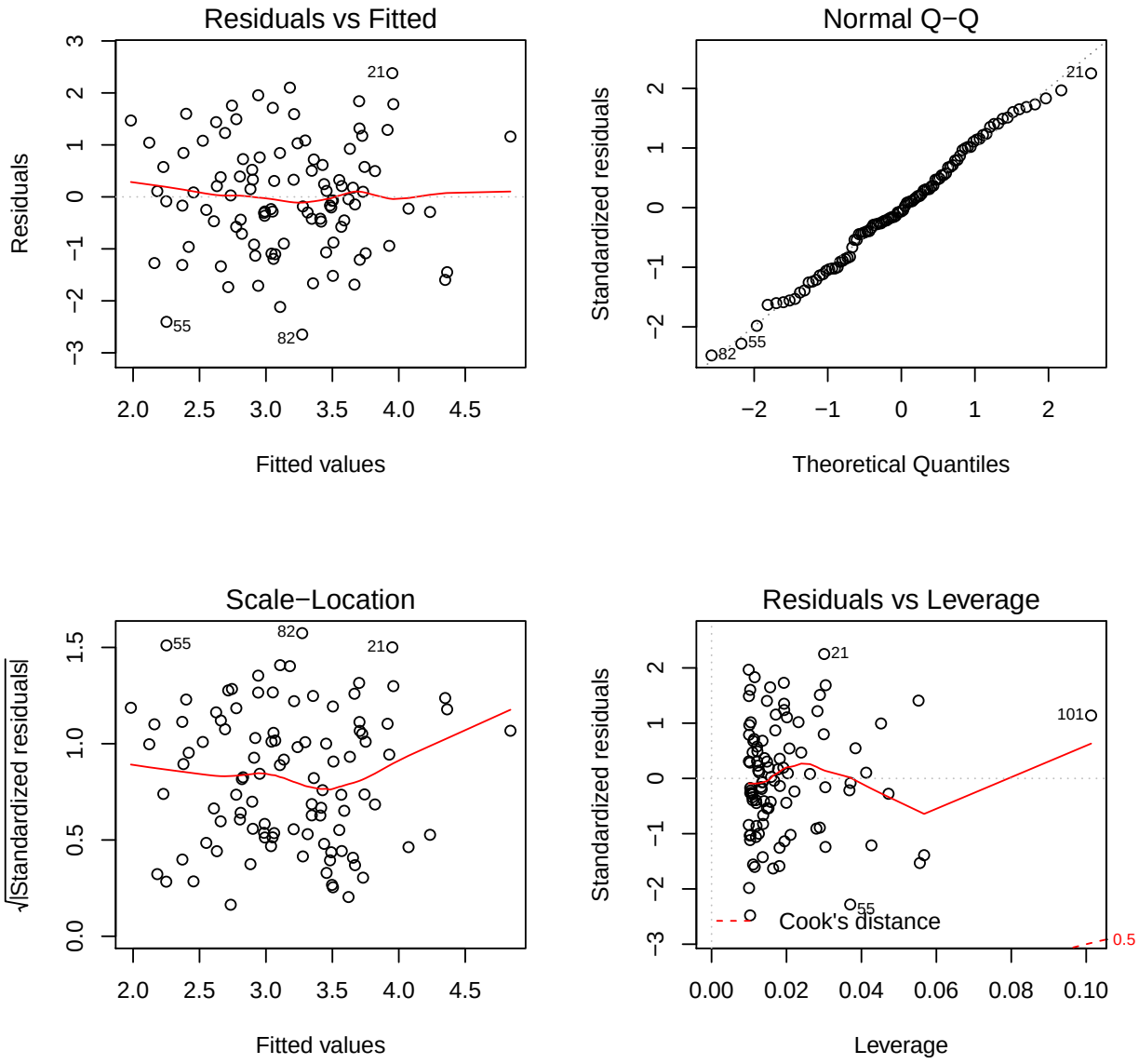


Figure 4: X_2 Plots.

Interpretation

Based on these plots, the point doesn't look like an outlier but it does look like it has a little bit of leverage although not as much as our first model.

Problem 2

From ISLR: Chapter 4, Problem 3 (Conceptual).

This problem relates to the QDA model, in which the obesrvations within each class are drawn from a normal distribution with a class specific mean vector and a class specific covariance matrix. We consider the simple case where $p = 1$, there is only one feature. Suppose that we have K classes, and if an observation belongs to the kth class then X comes from a one-dimensional normal distribution, $X \sim \mathrm{N}(\mu_k, \sigma_k^2)$. Recall that the density function for the one-dimensional normal distribution is given in Eq. 4.11 in the text. Prove that in this case, the Bayes classifier is not linear. Argue that it is in fact quadratic.

Hint: For this problem, you should follow the arguments laid out in Section 4.4.2, but without making the assumption that $\sigma_1^2 = \ldots = \sigma_K^2$.

Solution

Proof. We want to show that when taking a single dimension classification problem and don't assume that $\sigma_1^2 = \ldots = \sigma_K^2$, that we will end up with a quadratic function, not a linear one as we do with LDA.

If we have K classes and we want to see if an observation comes from the kth class, then we can use *Bayes' Theorem* for this which is (Equation 4.10 from ISLR):

$$\Pr(Y = K \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Since we are given that X comes from a one-dimensional normal distribution, we know that the following is the density for a given x and the k class:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

where μ_k and σ_k^2 are the mean and variance parameters for the kth class.

Given these two formulas, we can now create a function, $p_k(x)$ which will represent the posterior probability of being in the kth class:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} \exp\left(-\frac{1}{2\sigma_l^2}(x - \mu_l)^2\right)}$$

By taking the log of both sides of the equation, we let $\delta_k(x) = \log(p_k(x))$ we get:

$$\delta_k(x) = \log\left[\frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} \exp\left(-\frac{1}{2\sigma_l^2}(x - \mu_l)^2\right)}\right]$$

Since log is monotonic and we are trying to maximize the posterior probability, our denominator doesn't depend on the class we want, k. Therefore we want to maximize our numerator (also known as maximum a

posteriori estimation) and we get:

$$\begin{aligned}
\delta_k(x) &= \log\left[\frac{\pi_k}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)\right] \\
&= \log\left[\frac{\pi_k}{\sqrt{2\pi}\sigma_k}\right] + \log\left[\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)\right] \\
&= \log(\pi_k) - \log(\sqrt{2\pi}\sigma_k) + \log\left[\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)\right] \\
&= \log(\pi_k) - \log(\sqrt{2\pi}\sigma_k) - \frac{1}{2\sigma_k^2}(x-\mu_k)^2 \\
&= \log(\pi_k) - \frac{1}{2}\log(2\pi) + \log(\sigma_k) - \frac{1}{2\sigma_k^2}(x^2 - \mu_k - \mu_k + \mu_k^2) \\
&= \log(\pi_k) - \frac{1}{2}\log(2\pi) + \log(\sigma_k) - \frac{1}{2\sigma_k^2}(x^2 - 2\mu_k + \mu_k^2)
\end{aligned}$$

By further maximizing and only considering the values that allow us to maximize our value based on k, we get:

$$\delta_k(x) = -\frac{x^2}{2\sigma_k^2} + \frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + \log(\pi_k) + \log(\sigma_k)$$

Thus we can see that by not assuming that $\sigma_1^2 = \ldots = \sigma_K^2$, our discriminant function isn't linear in terms of x, instead it is quadratic. Therefore when neglecting the second assumption of LDA with a single dimensional problem, we end up with a quadratic discrimination function. □

Problem 3

From ISLR: Chapter 4, Problem 7 (Conceptual).

Suppose that you wish to predict whether a given stock will issue a dividend this year based on X, last year's percent profit. We examine a large number of companies and discover that the mean value of X for companies is issued a dividend was $\overline{X} = 10$, while the mean for those that didn't was $\overline{X} = 0$. In addition, the variance of X for these two sets of companies was $\sigma^2 = 36$. Finally, 80% of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was $X = 4$ last year.

Hint: Recall that the density function of a normal random variable is below. You will need to use Bayes' Theorem.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

Solution

We want to predict whether or not that a company will issue a dividend this year given that its percentage profit was 4.

This gives us two classes, or $K = 2$. Where the classes are whether a company issued a dividend or not. We can assign values to these such that $k_1 =$ issued a dividend, and $k_2 =$ did not issue a dividend.

By taking the values that we get out of the problem statement, we know we are given $\sigma = 6$, $\mu_1 = 10$ when issued a dividend, and $\mu_2 = 0$ when a dividend isn't given. We also know that the total probability of companies that issued dividends, or $\pi_1 = 0.80$ and thus $\pi_2 = 0.20$.

By using Bayes' Theorem, this gives us:

$$\begin{aligned}
\Pr(Y = 1 \mid X = 4) &= \frac{\pi_1 f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} \\
&= \frac{\pi_1 f_1(4)}{\sum_{l=1}^{2} \pi_l f_l(4)} \\
&= \frac{\pi_1 f_1(4)}{\pi_1 f_1(4) + \pi_2 f_2(4)}
\end{aligned}$$

by substituting in our values and formulas, we get:

$$\begin{aligned}
\Pr(Y = 1 \mid X = 4) &= \frac{\pi_1 f_1(4)}{\pi_1 f_1(4) + \pi_2 f_2(4)} \\
&= \frac{(0.80) f_1(4)}{(0.80) f_1(4) + (.20) f_2(4)} \\
&= \frac{\frac{0.80}{\sqrt{2\pi(36)}} e^{-(4-10)^2/2(36)}}{\frac{0.80}{\sqrt{2\pi(36)}} e^{-(4-10)^2/2(36)} + \frac{0.20}{\sqrt{2\pi(36)}} e^{-(4-0)^2/2(36)}} \\
&= \frac{(0.80)(0.0403)}{(0.80)(0.0403) + (0.20)(0.0532)} \\
&= .7519
\end{aligned}$$

Thus there is a 75.2% chance that the company will issue a dividend with a percentage profit of 4.

Problem 4

From ISLR: Chapter 4, Problem 10 (Applied).

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.

Part A

Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

Solution

A basic summary of our data:

```
summary(Weekly)
```

```
##       Year           Lag1               Lag2               Lag3          
##  Min.   :1990   Min.   :-18.195   Min.   :-18.195   Min.   :-18.195  
##  1st Qu.:1995   1st Qu.: -1.154   1st Qu.: -1.154   1st Qu.: -1.158  
##  Median :2000   Median :  0.241   Median :  0.241   Median :  0.241  
##  Mean   :2000   Mean   :  0.151   Mean   :  0.151   Mean   :  0.147  
##  3rd Qu.:2005   3rd Qu.:  1.405   3rd Qu.:  1.409   3rd Qu.:  1.409  
##  Max.   :2010   Max.   : 12.026   Max.   : 12.026   Max.   : 12.026  
##       Lag4               Lag5              Volume          Today        
##  Min.   :-18.195   Min.   :-18.195   Min.   :0.087   Min.   :-18.195  
##  1st Qu.: -1.158   1st Qu.: -1.166   1st Qu.:0.332   1st Qu.: -1.154  
##  Median :  0.238   Median :  0.234   Median :1.003   Median :  0.241  
##  Mean   :  0.146   Mean   :  0.140   Mean   :1.575   Mean   :  0.150  
##  3rd Qu.:  1.409   3rd Qu.:  1.405   3rd Qu.:2.054   3rd Qu.:  1.405  
##  Max.   : 12.026   Max.   : 12.026   Max.   :9.328   Max.   : 12.026  
##  Direction 
##  Down:484  
##  Up  :605  
##            
##            
##            
##
```

Looking at the data might be helpful:

```
pairs(Weekly)
```

Problem 4 continued on next page...

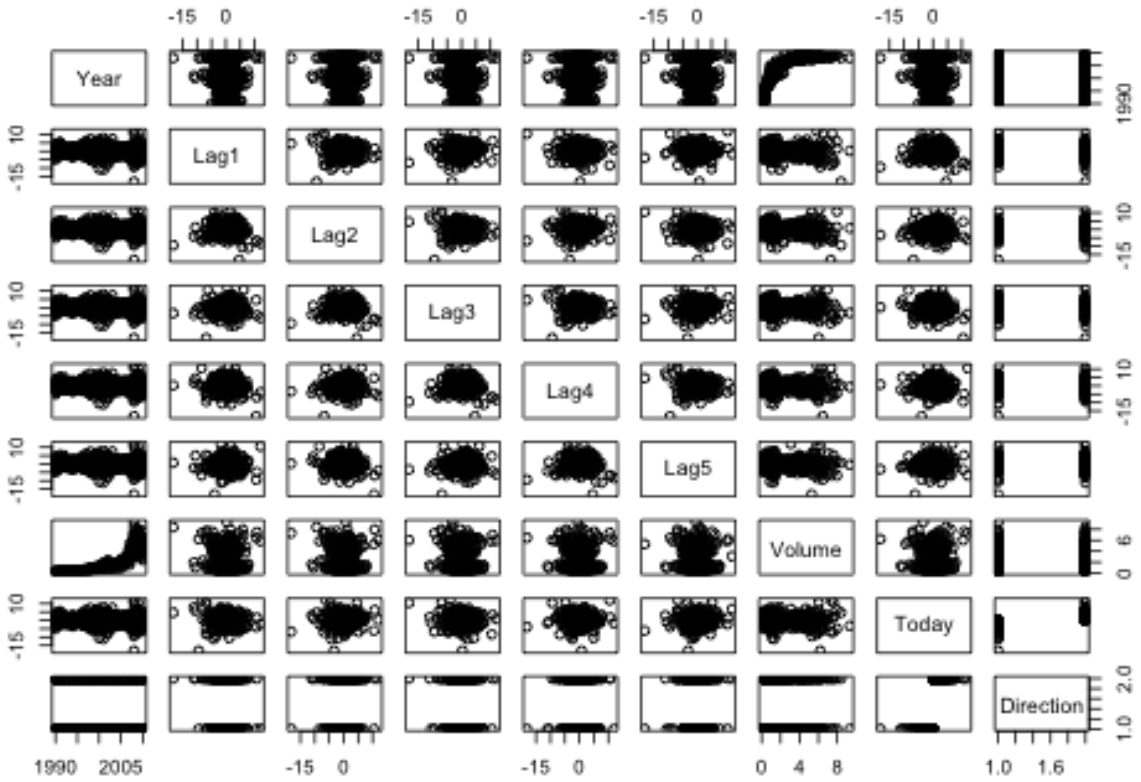


Figure 5: Pairs of Weekly data.

Lastly, let's look at some of the correlation coefficients to see if there's anything interesting:

```
cor(Weekly[1:8])
```

```
##              Year       Lag1      Lag2      Lag3       Lag4       Lag5    Volume
## Year     1.00000 -0.032289 -0.03339 -0.03001 -0.031128 -0.030519  0.84194
## Lag1    -0.03229  1.000000 -0.07485  0.05864 -0.071274 -0.008183 -0.06495
## Lag2    -0.03339 -0.074853  1.00000 -0.07572  0.058382 -0.072499 -0.08551
## Lag3    -0.03001  0.058636 -0.07572  1.00000 -0.075396  0.060657 -0.06929
## Lag4    -0.03113 -0.071274  0.05838 -0.07540  1.000000 -0.075675 -0.06107
## Lag5    -0.03052 -0.008183 -0.07250  0.06066 -0.075675  1.000000 -0.05852
## Volume   0.84194 -0.064951 -0.08551 -0.06929 -0.061075 -0.058517  1.00000
## Today   -0.03246 -0.075032  0.05917 -0.07124 -0.007826  0.011013 -0.03308
##              Today
## Year    -0.032460
## Lag1    -0.075032
## Lag2     0.059167
## Lag3    -0.071244
## Lag4    -0.007826
## Lag5     0.011013
## Volume -0.033078
```

```
## Today   1.000000
```

Most of them are small with the exception of **Volume** and **Year** which suggests that they might be correlated.

Part B

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?

Solution

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

##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial(), data = Weekly)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.695  -1.256   0.991   1.085   1.458
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2669     0.0859    3.11   0.0019 **
## Lag1         -0.0413     0.0264   -1.56   0.1181
## Lag2          0.0584     0.0269    2.18   0.0296 *
## Lag3         -0.0161     0.0267   -0.60   0.5469
## Lag4         -0.0278     0.0265   -1.05   0.2937
## Lag5         -0.0145     0.0264   -0.55   0.5833
## Volume       -0.0227     0.0369   -0.62   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
##
## Number of Fisher Scoring iterations: 4
```

According to the results of the summary, the t-statistic for **Lag2** is large and is considered significant based on the p-value.

Part C

Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix

is telling you about the types of mistakes made by logistic regression.

Solution

To calculate our confusion matrix, we need to assign our predictions to classes. We can do this using the following code:

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

##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial(), data = Weekly)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.695  -1.256   0.991   1.085   1.458
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2669     0.0859    3.11   0.0019 **
## Lag1         -0.0413     0.0264   -1.56   0.1181
## Lag2          0.0584     0.0269    2.18   0.0296 *
## Lag3         -0.0161     0.0267   -0.60   0.5469
## Lag4         -0.0278     0.0265   -1.05   0.2937
## Lag5         -0.0145     0.0264   -0.55   0.5833
## Volume       -0.0227     0.0369   -0.62   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
##
## Number of Fisher Scoring iterations: 4
```

```
# Compute our prediction from our logistic model
pred.glm <- predict.glm(fit.glm, Weekly, type = "response")

# Predicted classes
classes <- rep("Down", dim(Weekly)[1])
classes[pred.glm >= 0.5] <- "Up"
```

And our confusion matrix is thus:

```
# Make matrix
table(Weekly$Direction, classes, dnn = c("Predicted:", ""))

##
## Predicted: Down  Up
##       Down   54 430
##       Up     48 557

m <- mean(classes == Weekly$Direction)
m

## [1] 0.5611
```

Giving us the fraction of: $(54 + 557)/(54 + 430 + 48 + 557) = 0.561$ and in R: 0.5611.

The confusion matrix is telling us how many predicted classes were right and how many were wrong. Therefore when the classes match, it means it was correctly classified. If they don't match, then they were classified incorrectly.

Part D

Now fit the logistic regression model using a training data period from 1990 to 2008, with *Lag2* as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

Solution

Using a subset of the data, we generate our model like so:

```
# Select the subset of the data we want
Weekly.train <- subset(Weekly, Year <= 2008)
Weekly.test <- subset(Weekly, Year > 2008)

# Create our new logistic model
fit.glm.lag2 <- glm(Direction ~ Lag2, data = Weekly.train, family = binomial())

# Let's take a gander at it...
summary.glm(fit.glm.lag2)

##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial(), data = Weekly.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -1.54   -1.26    1.02    1.09    1.37
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2033     0.0643    3.16   0.0016 **
## Lag2          0.0581     0.0287    2.02   0.0430 *
## ---
```

```
## 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: 1355
##
## Number of Fisher Scoring iterations: 4
```

Then we can use this model for predictions:

```
# Compute our prediction from our subset logistic model
pred.glm.lag2 <- predict.glm(fit.glm.lag2, Weekly.test, type = "response")

# Predicted classes for the subset
classes <- rep("Down", dim(Weekly.test)[1])
classes[pred.glm.lag2 >= 0.5] <- "Up"
```

And our confusion matrix is thus:

```
# Make matrix
table(Weekly.test$Direction, classes, dnn = c("Predicted:", ""))

##
## Predicted: Down Up
##       Down    9 34
##       Up      5 56

m.glm <- mean(classes == Weekly.test$Direction)
m.glm

## [1] 0.625
```

Giving us the fraction of: $(9+56)/(9+34+5+56) = 0.625$ and in R: 0.625.

Part E

Repeat (d) using LDA.

Solution

We'll use the subset of the Weekly data and run LDA on it. This can be done with the following code:

```
# Create our new logistic model
fit.lda <- lda(formula = Direction ~ Lag2, data = Weekly.train)
```

Then we can use this model for predictions:

```
# Compute our prediction from our subset logistic model
pred.lda <- predict(fit.lda, Weekly.test, type = "response")
```

And our confusion matrix is thus:

```
# Make matrix
table(Weekly.test$Direction, pred.lda$class, dnn = c("Predicted:", ""))

##
## Predicted: Down Up
##       Down    9 34
##       Up      5 56

m.lda <- mean(pred.lda$class == Weekly.test$Direction)
m.lda

## [1] 0.625
```

Giving us the fraction of: $(9+56)/(9+34+5+56) = 0.625$ and in R: 0.625.

Part F

Repeat (d) using QDA.

Solution

We'll use the subset of the Weekly data and run QDA on it. This can be done with the following code:

```
# Create our new logistic model
fit.qda <- qda(formula = Direction ~ Lag2, data = Weekly.train)
```

Then we can use this model for predictions:

```
# Compute our prediction from our subset logistic model
pred.qda <- predict(fit.qda, Weekly.test, type = "response")
```

And our confusion matrix is thus:

```
# Make matrix
table(Weekly.test$Direction, pred.qda$class, dnn = c("Predicted:", ""))

##
## Predicted: Down Up
##       Down    0 43
##       Up      0 61

m.qda <- mean(pred.qda$class == Weekly.test$Direction)
m.qda

## [1] 0.5865
```

Giving us the fraction of: $(0+61)/(0+43+0+61) = 0.587$ and in R: 0.5865.

Part G

Is it justified to use QDA? Use appropriate hypothesis test(s) we've seen in class.

Solution

QDA and LDA share the same first assumption. And that assumption is that all the classes come from a multivariate normal.

Class Mean Difference

First we should check to make sure that the class means are significantly different. Since we are only using a single class, we are fine.

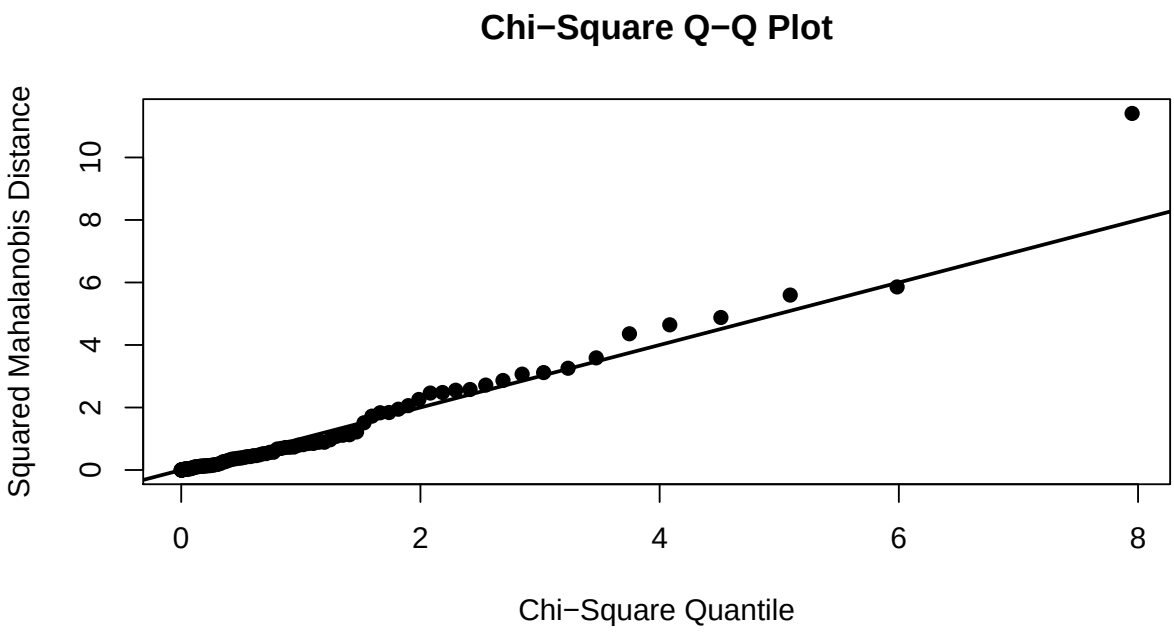
Size of the Training Set

Another thing to consider with using QDA is how big our training set is. If there isn't a lot of data to use for training, it might be a better idea to not use QDA. The number of data points we have is 985. Thus we should have enough to do QDA.

Assumption 1

The first assumption is that each class comes from a multivariate normal. To ensure that this is the case, we can use the **HZ.test()** function like so:

```
# Check to ensure MVN for each variable of our classifier
HZ.test(Weekly.test$Lag2, cov = TRUE, plot = TRUE)
```




```
##
##  Henze-Zirkler's Multivariate Normality Test
##
## data:  Weekly.test$Lag2
## HZ = 0.6329, p-value = 0.109
```

Based on these results, we can see that we can't reject that our class comes from multivariate normal. Thus it must be multivariate normal.

Taking all of these into account, we should be justified in using QDA.

Problem 4 continued on next page...

Part H

Repeat (d) using KNN with $K = 1$.

Solution

We need to create some matrices for our test and training data:

```
train.knn <- as.matrix(Weekly.train$Lag2)
test.knn <- as.matrix(Weekly.test$Lag2)
```

Now we can use our data as well a KNN to predict our test data:

```
pred.knn <- knn(train.knn, test.knn, Weekly.train$Direction, k = 1)
```

This gives us the confusion matrix of:

```
table(pred.knn, Weekly.test$Direction)

##
## pred.knn Down Up
##     Down   21 30
##     Up     22 31

m.knn <- mean(pred.knn == Weekly.test$Direction)
m.knn

## [1] 0.5
```

Giving us the fraction of: $(21 + 31)/(21 + 30 + 22 + 31) = 0.500$ and in R: 0.5.

Part I

Which of these methods appears to provide the best results on this data?

Solution

Based on all the methods that we ran, we got the following values:

1. Logistic Regression: 0.625
2. LDA: 0.625
3. QDA: 0.5865
4. KNN: 0.5

Thus Logistic Regression and LDA had the highest correct classifications.

Part J

Could you create a better classifier? How would you do this?

Solution

One way that might improve the classifier is to use Naive Bayes. Although whether or not Naive Bayes will improve it is dependent on many factors, as we discussed in class, the robustness of Naive Bayes might do pretty well in a surprising fashion.