MAT3375_Assignment-4

Rahul Atre

2023-11-22

Exercise 8.3

Consider the delivery time data in Example 3.1. In Section 4.2.5 it is noted that these observations were collected in four cities: San Diego, Boston, Austin, and Minneapolis.

a. Develop a model that relates delivery time y to cases x_1, distance x_2, and the city in which the delivery was made. Estimate the parameters of the model.

Ans: From reading Section 4.2, the text mentions that from the 25 observations in Table 3.2 that were collected, obs. 1-7 were collected in San Diego, 8-17 in Boston, 18-23 in Austin, and 24-25 in Minneapolis. We will introduce 4-1=3 dummy variables.

- We let $x_3 = 1$ and rep. if the obs. was collected in San Diego, 0 otherwise
- We let $x_4 = 1$ and rep. if the obs. was collected in Boston, 0 otherwise
- We let $x_5 = 1$ and rep. if the obs. was collected in Austin, 0 otherwise
- Also, if $x_3 = x_4 = x_5 = 0$, then the obs. was collected in Minneapolis

```
delivery_df = p8.3

san_data = rep(0, 25)
boston_data = rep(0, 25)
austin_data = rep(0, 25)

delivery_df$x3 = san_data
delivery_df$x4 = boston_data
delivery_df$x5 = austin_data

for (i in 1:7){
  delivery_df[i, ]$x3 = 1
}

for (i in 8:17){
  delivery_df[i, ]$x4 = 1
}

for (i in 18:23){
  delivery_df[i, ]$x5 = 1
}
```

Also, we know that y rep. the delivery time, x_1 rep. the # of cases, and x_2 rep. the distance.

Let us now calculate the parameters using the lm() function to fit the linear model:

```
full_model = lm(y ~ x1 + x2 + x3 + x4 + x5, data = delivery_df)
full_model
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = delivery_df)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##     0.41625      1.77028      0.01083      2.28510      3.73764     -0.45264
```

From the above function call, we obtain $b_0 = 0.41625$, $b_1 = 1.77028$, $b_2 = 0.01083$, $b_3 = 2.28510$, $b_4 = 3.73764$, and $b_5 = -0.45264$.

Therefore, the linear regression model will be $\hat{y} = 0.41625 + 1.77028x_1 + 0.01083x_2 + 2.28510x_3 + 3.73764x_4 - 0.45264x_5$.

b. Is there an indication that delivery site is an important variable?

We can perform a partial F-test to check if there is an indication of delivery site being an important variable.

- Full Model: $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \epsilon_i$
- Reduced Model: $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon_i$

We already have the full model from part a.

```
reduced_model = lm(y ~ x1 + x2, data = delivery_df)
anova(reduced_model, full_model)
```

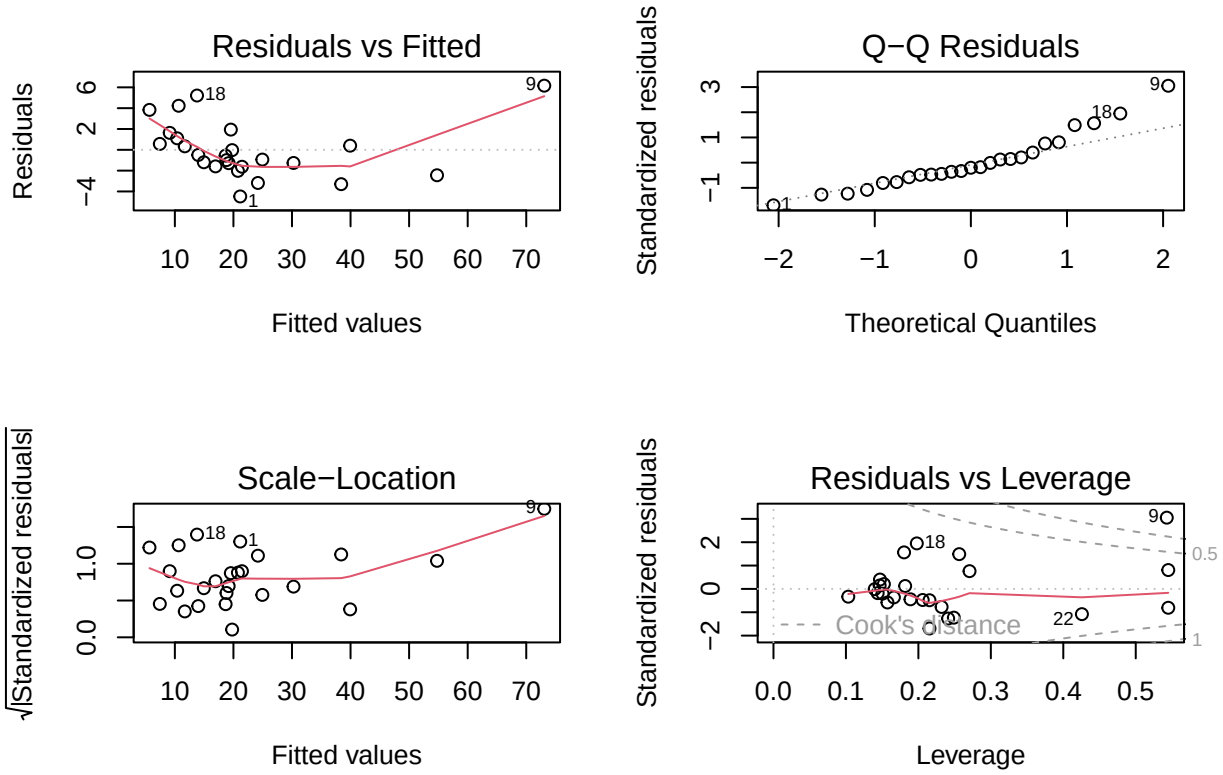
```
## Analysis of Variance Table
##
## Model 1: y ~ x1 + x2
## Model 2: y ~ x1 + x2 + x3 + x4 + x5
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     22 233.73
## 2     19 169.45  3    64.281 2.4025 0.09946 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let the null hypothesis be $H_0 : \beta_3 = \beta_4 = \beta_5 = 0$. From the above anova table, we can see that the partial F-statistic is 2.4025 and the p-value is 0.09946. Since the p-value is quite high, this statistic is not significant. Therefore, we do not reject the null hypothesis and there is not enough evidence to suggest that the delivery site is an important variable.

c. Analyze the residuals from this model. What conclusions can you draw regarding model adequacy?

To analyze the residuals, we need to generate the residual vs. fitted, normal qq-plot, and residuals vs. leverage

```
par(mfrow=c(2,2))
plot(full_model)
```



From the QQ-plot, we can see that most of the residual points towards the end don't follow the line of best fit, indicating that the normality assumption is not met (Residuals are not normally distributed). From the residuals vs. fitted, we can see that the points follow a u-shaped curve, meaning that the constancy of variance assumption is not met and that the linear model is not ideal for this dataset. Overall, the quality of this model is not satisfied.

Exercise 9.17

Apply ridge regression to the Hald cement data in Table B.21.

a. Use the ridge trace to select an appropriate value of k. Is the final model a good one?

```
hald_df <- read.table("C://Users/User/OneDrive/Documents/Rahul/uOttawa/2023(9) - Fall/MAT3375 - Regressi
y = hald_df$Y
x1 = hald_df$X1
x2 = hald_df$X2
x3 = hald_df$X3
x4 = hald_df$X4
```

From this data, we can perform cross-validation and obtain the minimum λ value, k, for ridge regression:

```
combinedX = data.matrix(hald_df[, c("X1", "X2", "X3", "X4")])
multipleModels = cv.glmnet(combinedX, y, alpha=0)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
min_lambda = multipleModels$lambda.min
ridge_model = glmnet(combinedX, y, alpha = 0, lambda = min_lambda)

coef(ridge_model)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept) 86.3899965
## X1           1.1298726
## X2           0.2906186
## X3          -0.2546536
## X4          -0.3464944
```

From the above function call, we obtain $b_0 = 86.3899965$, $b_1 = 1.1298726$, $b_2 = 0.2906186$, $b_3 = -0.2546536$, $b_4 = -0.3464944$.

Therefore, the ridge regression model will be $\hat{y} = 86.3899965 + 1.1298726x_1 + 0.2906186x_2 - 0.2546536x_3 - 0.3464944x_4$.

To know if it is a good model, we must check the R^2 value:

```
y_predict= predict(ridge_model, combinedX)
SSE = sum((y_predict - y)^2)
SST = sum((y - mean(y))^2)
r_sqr = 1 - SSE/SST

r_sqr
```

```
## [1] 0.9793364
```

From the above function call, $R^2 = 0.9793364 \approx 97.93\%$. Therefore, about 97.93% of the total variability in y is explained by this model, which is extremely good. Although there are more ways to check if this is a good model, a high R^2 value is a decent enough indicator.

Therefore, this final model is exceptionally good at fitting the provided data.

b. How much inflation in the residual sum of squares has resulted from the use of ridge regression?

If we want to know the inflation in SSE from ridge in comparison to the least-squares model, we need to examine the SSE of it:

```
least_sqr_model = lm(Y ~ X1 + X2 + X3 + X4, data = hald_df)
anova(least_sqr_model)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## X1         1 1450.08 1450.08 242.3679 2.888e-07 ***
```

```
## X2          1 1207.78 1207.78 201.8705 5.863e-07 ***
## X3          1    9.79    9.79   1.6370    0.2366
## X4          1    0.25    0.25   0.0413    0.8441
## Residuals   8   47.86    5.98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
SSE #Ridge Regression residual sum of squares
```

```
## [1] 56.11751
```

From the above function call, we can see that the residual sum of squares is 47.86, whereas for the ridge regression it is 56.11751. Calculating the overall inflation:

```
inflation = (56.11751 - 47.86)/47.86 * 100
```

Therefore, the inflation in the residual sum of squares has resulted from the use of ridge regression is 17.25%.

c. Compare the ridge regression model with the two-regressor model involving x_1 and x_2 developed by all possible regressions in Example 9.1.

From the example given in the textbook, if we take a look at the two-regressor model involving x_1 and x_2, we can see that the $R^2 = 0.97868 = 97.868\%$, and the $SSE = 57.9045$. In comparison, the ridge regression model gave us $R^2 = 0.97868 = 97.93\%$, and $SSE = 56.11751$.

Both models appear to have very similar values for the two. Both models fit the data exceptionally well, with very high R^2 values.

Exercise 10.4

Consider the solar thermal energy test data in Table B.2.

a. Use forward selection to specify a subset regression model.

First, let us specify what the predictors and response represent. y rep. the total heat flux (kwatts) x_1 rep. the Insolation (watts/m^2) x_2 rep. the position of focal point in east direction (inches) x_3 rep. the position of focal point in south direction (inches) x_4 rep. the position of focal point in north direction (inches) x_5 rep. the time of day

We can use the olsrr package to perform forward regression on the model:

```
lin_model = lm(y ~ ., data = table.b2)
ols_step_forward_p(lin_model)
```

```
##
##                            Selection Summary
## --------------------------------------------------------------------------
##         Variable                    Adj.
## Step    Entered     R-Square    R-Square     C(p)        AIC        RMSE
## --------------------------------------------------------------------------
##    1    x4            0.7205      0.7102    38.4923    231.9133    12.3277
```

```
##    2    x3           0.8587      0.8478     9.0975    214.1313     8.9321
##    3    x2           0.8741      0.8590     7.5963    212.7817     8.5978
##    4    x1           0.8909      0.8727     5.7873    210.6363     8.1698
##    5    x5           0.8988      0.8768     6.0000    210.4660     8.0390
## ----------------------------------------------------------------------------
```

From the above function call, we can see that forward selection has chosen all predictors x_1, x_2, x_3, x_4, x_5. Checking the performance of the model:

```
summary(lin_model)
```

```
##
## Call:
## lm(formula = y ~ ., data = table.b2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6848  -2.7688   0.6273   3.9166  17.3962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 325.43612   96.12721   3.385  0.00255 **
## x1            0.06753    0.02899   2.329  0.02900 *
## x2            2.55198    1.24824   2.044  0.05252 .
## x3            3.80019    1.46114   2.601  0.01598 *
## x4          -22.94947    2.70360  -8.488 1.53e-08 ***
## x5            2.41748    1.80829   1.337  0.19433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.039 on 23 degrees of freedom
## Multiple R-squared:  0.8988, Adjusted R-squared:  0.8768
## F-statistic: 40.84 on 5 and 23 DF,  p-value: 1.077e-10
```

The linear model when all predictors are included is $\hat{y} = 325.43612 + 0.06753x_1 + 2.55198x_2 + 3.80019x_3 - 22.94947x_4 + 2.41748x_5$.

b. Use backward elimination to specify a subset regression model.

```
ols_step_backward_p(lin_model)
```

```
## [1] "No variables have been removed from the model."
```

Since backward elimination did not remove any variables from the model, similar to forward selection, it has chosen all predictors x_1, x_2, x_3, x_4, x_5. As such, the linear model for backward will be the same, which is $\hat{y} = 325.43612 + 0.06753x_1 + 2.55198x_2 + 3.80019x_3 - 22.94947x_4 + 2.41748x_5$.

c. Use stepwise regression to specify a subset regression model.

```
ols_step_both_p(lin_model)
```

```
##
##                               Stepwise Selection Summary
## ----------------------------------------------------------------------------------------
##                      Added/                   Adj.
## Step    Variable    Removed     R-Square    R-Square     C(p)        AIC        RMSE
## ----------------------------------------------------------------------------------------
##    1       x4       addition       0.721       0.710    38.4920    231.9133    12.3277
##    2       x3       addition       0.859       0.848     9.0980    214.1313     8.9321
##    3       x2       addition       0.874       0.859     7.5960    212.7817     8.5978
##    4       x1       addition       0.891       0.873     5.7870    210.6363     8.1698
## ----------------------------------------------------------------------------------------
```

From the above function call, the stepwise regression has chosen the predictors x_1, x_2, x_3, x_4 and omitted x_5. Let us fit this stepwise model:

```
stepwise_model = lm(y ~ x1 + x2 + x3 + x4, data = table.b2)
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = table.b2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.322  -2.639   0.025   4.786  16.003
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 270.21013   88.21060   3.063  0.00534 **
## x1            0.05156    0.02685   1.920  0.06676 .
## x2            2.95141    1.23167   2.396  0.02471 *
## x3            5.33861    0.91506   5.834 5.13e-06 ***
## x4          -21.11940    2.36936  -8.914 4.42e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.17 on 24 degrees of freedom
## Multiple R-squared:  0.8909, Adjusted R-squared:  0.8727
## F-statistic: 48.99 on 4 and 24 DF,  p-value: 3.327e-11
```

From the above function call, we get the model $\hat{y} = 270.21013 + 0.05156x_1 + 2.95141x_2 + 5.33861x_3 - 21.11940x_4$.

d. Apply all possible regressions to the data. Evaluate R_p^2, C_p and MS_{Res} for each model. Which subset model do you recommend?

```
ols_step_all_possible(lin_model)
```

```
##    Index N     Predictors   R-Square Adj. R-Square Mallow's Cp
## 4      1 1             x4 0.72052420    0.71017324   38.492277
## 1      2 1             x1 0.39393873    0.37149202  112.687090
## 5      3 1             x5 0.12328631    0.09081543  174.174842
## 3      4 1             x3 0.01256447   -0.02400721  199.329010
## 2      5 1             x2 0.01047594   -0.02617310  199.803490
## 13     6 2          x3 x4 0.85871542    0.84784738    9.097518
## 15     7 2          x4 x5 0.82020641    0.80637613   17.846129
## 8      8 2          x1 x4 0.73386067    0.71338841   37.462451
## 11     9 2          x2 x4 0.72053206    0.69903453   40.490491
## 6     10 2          x1 x2 0.44922541    0.40685814  102.126871
## 7     11 2          x1 x3 0.42636430    0.38223847  107.320539
## 9     12 2          x1 x5 0.39429011    0.34769704  114.607262
## 14    13 2          x3 x5 0.37034618    0.32191127  120.046927
## 12    14 2          x2 x5 0.12963616    0.06268509  174.732260
## 10    15 2          x2 x3 0.03427915   -0.04000706  196.395794
## 22    16 3       x2 x3 x4 0.87412678    0.85902200    7.596312
## 19    17 3       x1 x3 x4 0.86478901    0.84856370    9.717698
## 25    18 3       x3 x4 x5 0.86376234    0.84741382    9.950942
## 21    19 3       x1 x4 x5 0.86288638    0.84643274   10.149946
## 24    20 3       x2 x4 x5 0.82022133    0.79864789   19.842738
## 17    21 3       x1 x2 x4 0.73615340    0.70449181   38.941581
## 16    22 3       x1 x2 x3 0.52969885    0.47326272   85.844637
## 20    23 3       x1 x3 x5 0.46570209    0.40158634  100.383642
## 23    24 3       x2 x3 x5 0.46085816    0.39616114  101.484104
## 18    25 3       x1 x2 x5 0.45487844    0.38946385  102.842597
## 26    26 4    x1 x2 x3 x4 0.89089317    0.87270870    5.787266
## 29    27 4    x1 x3 x4 x5 0.88036169    0.86042197    8.179844
## 30    28 4    x2 x3 x4 x5 0.87488338    0.85403061    9.424426
## 28    29 4    x1 x2 x4 x5 0.86898542    0.84714966   10.764344
## 27    30 4    x1 x2 x3 x5 0.58159740    0.51186363   76.054147
## 31    31 5 x1 x2 x3 x4 x5 0.89876023    0.87675159    6.000000
```

Applying all possible regressions, the recommended subset model that we obtain is the one that includes all predictors x_1, x_2, x_3, x_4, x_5. This is the same as forward selection and backwards elimination. This has the best R^2 value, which is $0.89876023 = 89.87\%$, and a Mallow's Cp at 6, which is exactly p+1. It is closest to the actual number of predictors.

e. Compare and contrast the models produced by the variable selection strategies in parts a - d.

From the previous parts, we have obtained two models, one with all predictors, and one with x_5 removed. Let us compare these:

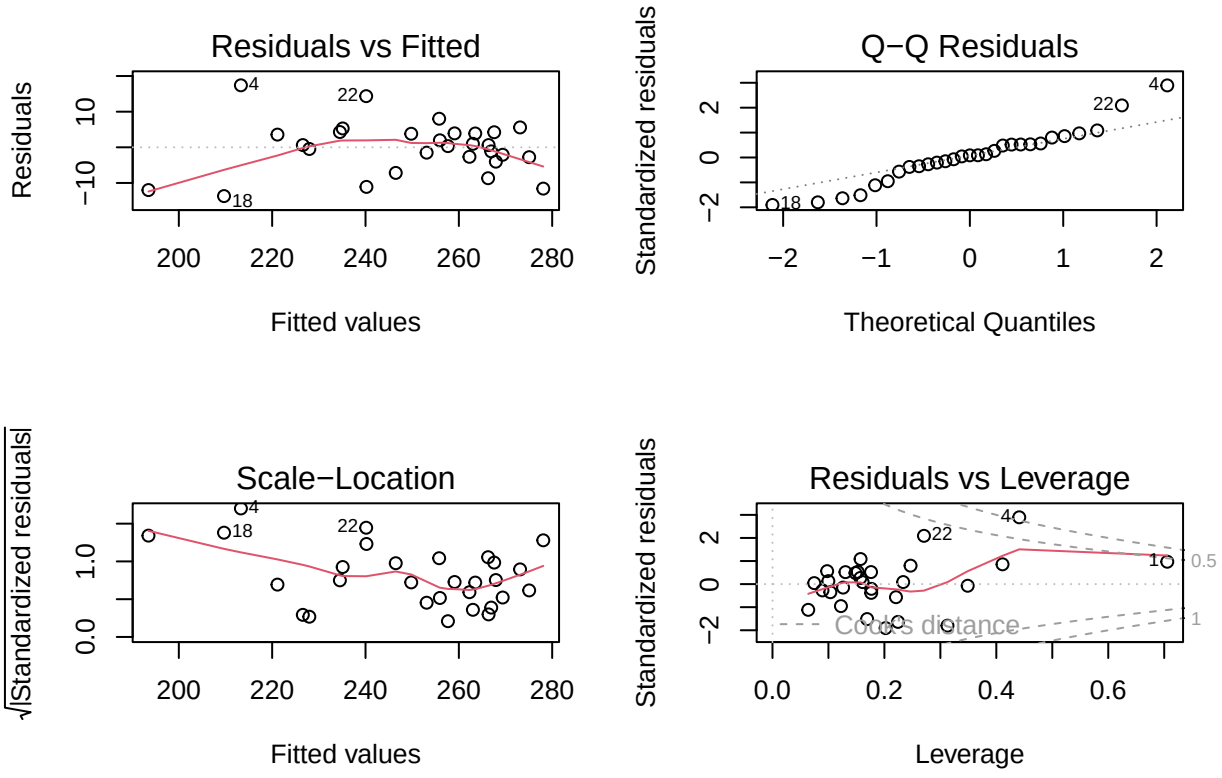
i) Full Model

```
summary(lin_model) #Overall summary
```

```
##
## Call:
## lm(formula = y ~ ., data = table.b2)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -13.6848  -2.7688   0.6273   3.9166  17.3962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 325.43612   96.12721   3.385  0.00255 **
## x1            0.06753    0.02899   2.329  0.02900 *
## x2            2.55198    1.24824   2.044  0.05252 .
## x3            3.80019    1.46114   2.601  0.01598 *
## x4          -22.94947    2.70360  -8.488 1.53e-08 ***
## x5            2.41748    1.80829   1.337  0.19433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.039 on 23 degrees of freedom
## Multiple R-squared:  0.8988, Adjusted R-squared:  0.8768
## F-statistic: 40.84 on 5 and 23 DF,  p-value: 1.077e-10
```

```
par(mfrow=c(2,2)) #Residuals
plot(lin_model)
```




```
ols_coll_diag(lin_model) #Variance factor and tolerance
```

```
## Tolerance and Variance Inflation Factor
## ---------------------------------------
```

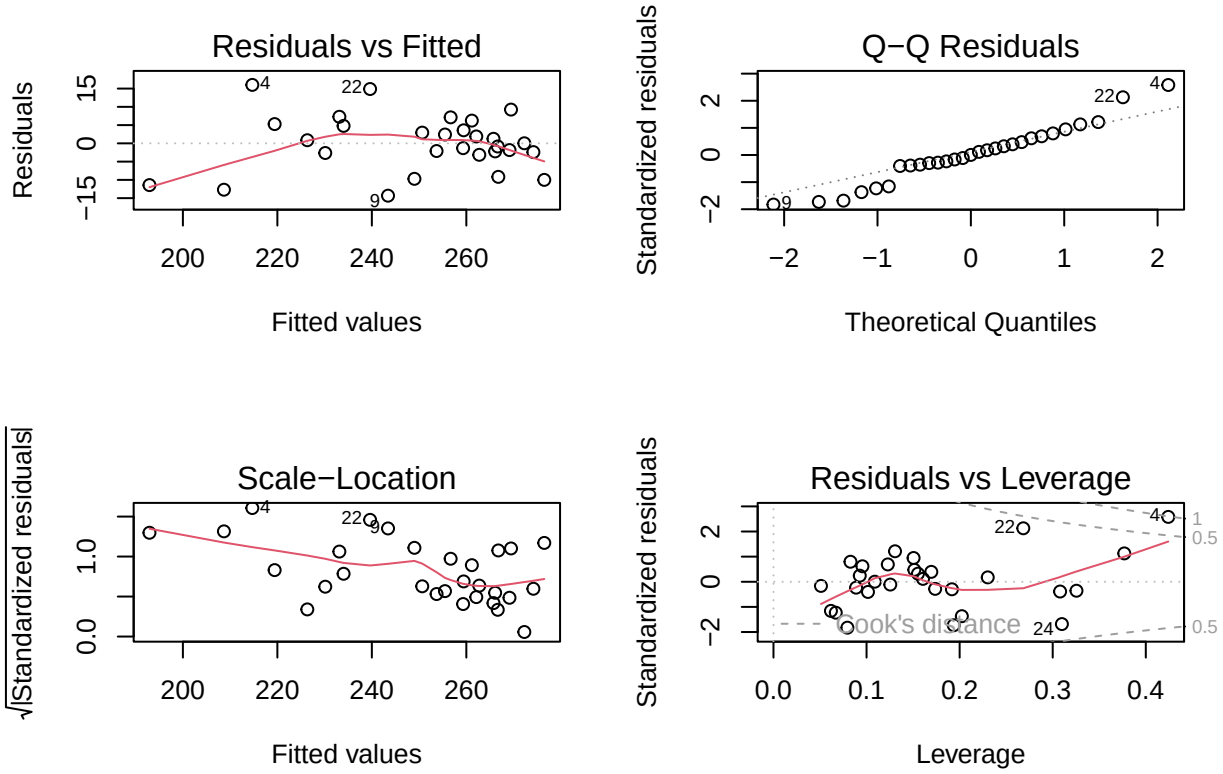
```
##   Variables Tolerance      VIF
## 1        x1 0.4312139 2.319035
## 2        x2 0.7377635 1.355448
## 3        x3 0.3148644 3.175970
## 4        x4 0.3828387 2.612066
## 5        x5 0.1862177 5.370059
##
##
## Eigenvalue and Condition Index
## ------------------------------
##     Eigenvalue Condition Index    intercept           x1           x2
## 1 5.9668908760         1.00000 6.767793e-06 0.0001277963 3.247665e-05
## 2 0.0254639495        15.30774 1.421387e-04 0.0781510580 6.706206e-04
## 3 0.0048957271        34.91125 3.114582e-03 0.2626084537 7.714844e-02
## 4 0.0015969065        61.12717 5.777198e-04 0.0020179978 2.721602e-01
## 5 0.0009778935        78.11389 8.735064e-03 0.5618754807 2.816480e-02
## 6 0.0001746475       184.83871 9.874237e-01 0.0952192134 6.218234e-01
##             x3           x4           x5
## 1 2.313969e-05 0.0000305478 0.0001056611
## 2 7.705420e-05 0.0007613186 0.0776507788
## 3 1.065040e-02 0.0260365246 0.1182825202
## 4 4.500398e-04 0.3935994846 0.1026553078
## 5 5.716067e-01 0.0384692395 0.4264493421
## 6 4.171927e-01 0.5411028849 0.2748563901
```

ii) Stepwise Model

```r
summary(stepwise_model) #Overall summary
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = table.b2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.322  -2.639   0.025   4.786  16.003
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 270.21013   88.21060   3.063  0.00534 **
## x1            0.05156    0.02685   1.920  0.06676 .
## x2            2.95141    1.23167   2.396  0.02471 *
## x3            5.33861    0.91506   5.834 5.13e-06 ***
## x4          -21.11940    2.36936  -8.914 4.42e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.17 on 24 degrees of freedom
## Multiple R-squared:  0.8909, Adjusted R-squared:  0.8727
## F-statistic: 48.99 on 4 and 24 DF,  p-value: 3.327e-11
```

```r
par(mfrow=c(2,2)) #Residuals
plot(stepwise_model)
```




```r
ols_coll_diag(stepwise_model) #Variance factor and tolerance
```

```
## Tolerance and Variance Inflation Factor
## ---------------------------------------
##   Variables Tolerance      VIF
## 1        x1 0.5192672 1.925791
## 2        x2 0.7825999 1.277792
## 3        x3 0.8291484 1.206057
## 4        x4 0.5148215 1.942421
##
##
## Eigenvalue and Condition Index
## ------------------------------
##     Eigenvalue Condition Index    intercept           x1           x2
## 1 4.9838045250         1.00000 1.190499e-05 0.0002214391 4.941464e-05
## 2 0.0117219692        20.61960 1.565581e-04 0.3478225865 2.004681e-03
## 3 0.0028524425        41.79959 2.043824e-03 0.0132812661 2.301331e-01
## 4 0.0013918322        59.83938 5.964702e-05 0.1576982611 1.097120e-01
## 5 0.0002292311       147.44964 9.977281e-01 0.4809764471 6.581008e-01
##             x3           x4
## 1 8.725762e-05 5.881296e-05
## 2 6.236713e-03 2.577810e-02
```

```
## 3 3.196264e-01 1.020974e-02
## 4 5.140050e-01 5.333921e-01
## 5 1.600446e-01 4.305613e-01
```

Looking at the summary call for the full model, it is visually apparent that the full models predictors are less statistically significant than the ones for stepwise. For stepwise, 2/5 predictors have a p-value less than 0.001 (***). However, for the full model, only x_4, 1/5 predictor has a p-value less than 0.001.

Also, the R^2 value of the stepwise is $0.8909 = 89.09\%$, whereas the R^2 for the full model is $0.8988 = 89.88\%$. Both models have a similar R^2, which is quite high. Therefore, it can be said that both models explain the constance of variance well enough.

Let's look at the residuals now. For the full model's residuals vs. fitted, we can see that the points follow a downward u-shape curve, meaning that the constancy of variance assumption is not met. Although towards the end the points do follow a balanced distribution on both sides, it is not to the standard we would like. The points at the beginning of QQ-plot do not follow the line of best fit, so normality assumption is not met either.

In comparison to the full model, the stepwise model is not that different. The normality and constancy of variance assumptions are not properly met. Overall, the quality of both models for residuals is not as reasonable as we would like.

Lastly, for the variance inflation factor, the values for both models are quite low (less than 10). Therefore, there is little multicolinearity for the full model (forwards, backwards, all possible) and stepwise model.

Exercise 13.3

The compressive strength of an alloy fastener used in aircraft construction is being studied. Ten loads were selected over the range 2500 – 4300 psi and a number of fasteners were tested at those loads. The numbers of fasteners failing at each load were recorded. The complete test data are shown below.

```
p13.3
```

```
##       x   n  r
## 1  2500  50 10
## 2  2700  70 17
## 3  2900 100 30
## 4  3100  60 21
## 5  3300  40 18
## 6  3500  85 43
## 7  3700  90 54
## 8  3900  50 33
## 9  4100  80 60
## 10 4300  65 51
```

a. Fit a logistic regression model to the data. Use a simple linear regression model as the structure for the linear predictor.

```
x_load = p13.3$x
n_sampleSize = p13.3$n
rNumberFailing = p13.3$r

fail_per_sample = rNumberFailing / n_sampleSize
```

```
logistic_model = glm(fail_per_sample ~ x_load, family = "binomial", weights = n_sampleSize)
logistic_model
```

```
##
## Call:  glm(formula = fail_per_sample ~ x_load, family = "binomial",
##     weights = n_sampleSize)
##
## Coefficients:
## (Intercept)       x_load
##   -5.339712     0.001548
##
## Degrees of Freedom: 9 Total (i.e. Null);  8 Residual
## Null Deviance:       112.8
## Residual Deviance: 0.3719    AIC: 49.09
```

From the above glm() function call, the logistic regression model is $\hat{\pi}_i = \frac{e^{-5.34+0.0015x_i}}{1+e^{-5.34+0.0015x_i}}$, where $\hat{\pi}_i$ is the probability of the alloy fastener failing, and x_i representing the load given.

b. Does the model deviance indicate that the logistic regression model from part a is adequate?

If we check the above function call, we can see that the deviance is 0.3719, which is extremely low. Therefore the logistic regression model from part a is very adequate.

c. Expand the linear predictor to include a quadratic term. Is there any evidence that this quadratic term is required in the model?

```
quad_term = x_load^2
quadratic_logistic_model = glm(fail_per_sample ~ x_load + quad_term, family = "binomial", weights = n_s
quadratic_logistic_model
```

```
##
## Call:  glm(formula = fail_per_sample ~ x_load + quad_term, family = "binomial",
##     weights = n_sampleSize)
##
## Coefficients:
## (Intercept)       x_load    quad_term
##  -4.269e+00    9.059e-04    9.408e-08
##
## Degrees of Freedom: 9 Total (i.e. Null);  7 Residual
## Null Deviance:       112.8
## Residual Deviance: 0.2837    AIC: 51
```

After including the quadratic term, we obtain a logistic linear model that is $\hat{\pi}_i = \frac{e^{-4.269+0.0009x}}{1+e^{-5.34+0.0015x_i}}$, where $\hat{\pi}_i$ is the probability of the alloy fastener failing, and x_i representing the load given.

The residual deviance with the quadratic term included is 0.2837. The difference in deviance between the two is $Deviance_x$ - $Deviance_{x,x^2} = 0.3719 - 0.2837 = 0.0882$. Since the difference in deviances is so low, it does not warrant an increase in complexity of the model.

Therefore, there is not enough evidence to indicate that the quadratic term is required in the model.

d. For the quadratic model in part c, find Wald statistics for each individual model parameter.

```r
summary(quadratic_logistic_model)
```

```
##
## Call:
## glm(formula = fail_per_sample ~ x_load + quad_term, family = "binomial",
##     weights = n_sampleSize)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.269e+00  3.643e+00  -1.172    0.241
## x_load       9.059e-04  2.168e-03   0.418    0.676
## quad_term    9.408e-08  3.167e-07   0.297    0.766
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 112.83207  on 9  degrees of freedom
## Residual deviance:   0.28371  on 7  degrees of freedom
## AIC: 51
##
## Number of Fisher Scoring iterations: 3
```

From the above function call, we can obtain the Z-values, which will tell us the Wald Statistics.

H_0: For β_0, the Wald Statistic is z = -1.172, which is not significant. H_1: For β_1, the Wald Statistic is z = 0.418, which is not significant. H_2: For β_2, the Wald Statistic is z = 0.297, which is not significant.

e. Find approximate 95% confidence intervals on the model parameters for the quadratic model from part c.

We can obtain a 95% confidence interval using the following function:

```r
confint(quadratic_logistic_model)
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %       97.5 %
## (Intercept) -1.147650e+01 2.825783e+00
## x_load      -3.332064e-03 5.177857e-03
## quad_term   -5.277476e-07 7.155743e-07
```

From the above function call, we have obtained the following 95% confidence interval for the three parameters of our quadratic model:

β_0: [-11.477, 2.826]

β_1: [-3.33 x 10^{-3}, 5.178 x 10^{-3}]

β_2: [-5.277 x 10^{-7}, 7.156 x 10^{-7}]