STAT 5532– FINAL EXCERCISES

Thanh Doan – Student ID 0159701

1. Exercise 7.1

```
> x = c(1,1.7,1.25,1.2,1.45,1.85,1.6,1.5,1.95,2)
> x2 = x^2;
> x2
 [1] 1.0000 2.8900 1.5625 1.4400 2.1025 3.4225 2.5600 2.2500 3.8025 4.0000
> cor(x,x2)
[1] 0.9954134
```

- The correlation between **x** and $\mathbf{x}^2$ is 0.9954
- From the correlation between x and $\mathbf{x}^2$ we can see the potential difficulty in fitting a second-order model because of the multi-co-linearity problem.

- To show the multi-co-linearity problem, I regress $\mathbf{x}^2$ on **x** and compute the VIF

```
>
> summary(lm(x2 ~ x));

Call:
lm(formula = x2 ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.10057 -0.08914 -0.01098  0.06131  0.17368

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.2222     0.1639  -13.56 8.41e-07 ***
x             3.0485     0.1036   29.43 1.93e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1038 on 8 degrees of freedom
Multiple R-squared: 0.9908, Adjusted R-squared: 0.9897
F-statistic: 866.1 on 1 and 8 DF,  p-value: 1.926e-09

> VIF = 1/(1-0.9908)
> print(VIF)
[1] 108.6957
```

- I also created a response vector, **y**, and regress **y** on **x** and $\mathbf{x}^2$ then use the **vif**() function to compute the variance inflation factors.

```
> y = 5 + 2*x + 3*x2
> y
 [1] 10.0000 17.0700 12.1875 11.7200 14.2075 18.9675 15.8800 14.7500 20.3075 21.0000
> gaussian_noise = rnorm(10, mean = 0, sd = 0.2)
> y = y + gaussian_noise
> y
 [1] 10.05138 17.09622 12.51313 11.89314 14.10679 18.89610 15.86919 14.89125 20.14788 21.00353
> my.quadratic.model = lm(y ~ x + x2);
> my.quadratic.model


Call:
lm(formula = y ~ x + x2)

Coefficients:
(Intercept)            x           x2
      5.009        2.335        2.804

> vif(my.quadratic.model)
       x       x2
109.2629 109.2629
```

- The VIF values in this step are slightly different than the ones in previous steps because of a round-off error, computed, in the previous step.

- The VIF values of 109.26 are high and suggest serious multi-co-linearity problem.

2. **Exercise 7.2**

a. Fit a second-order polynomial that expresses weight loss as a function of the number of months since production

```
Console C:/Users/th/git/mva/regression/
> weightloss.quad.lm = lm(y ~ X1 + X2, data= rocket.data);
summary(weightloss.quad.lm);

Call:
lm(formula = y ~ X1 + X2, data = rocket.data)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.005364 -0.002727  0.001045  0.002409  0.003273 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.633000   0.004196   389.2  < 2e-16 ***
X1          -1.232182   0.007010  -175.8 5.09e-14 ***
X2           1.494545   0.002484   601.6  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.003568 on 7 degrees of freedom
Multiple R-squared:     1,	Adjusted R-squared:     1 
F-statistic: 1.859e+06 on 2 and 7 DF,  p-value: < 2.2e-16
```

The fitted model is

$$\hat{y} = 1.633 - 1.232x + 1.49x^2$$

b. Test for significance of the regression

The statistic $\mathrm{F} = 1.86 \times 10^6$, with $\mathrm{p-value} \approx 0.0000$. Thus the regression is significant

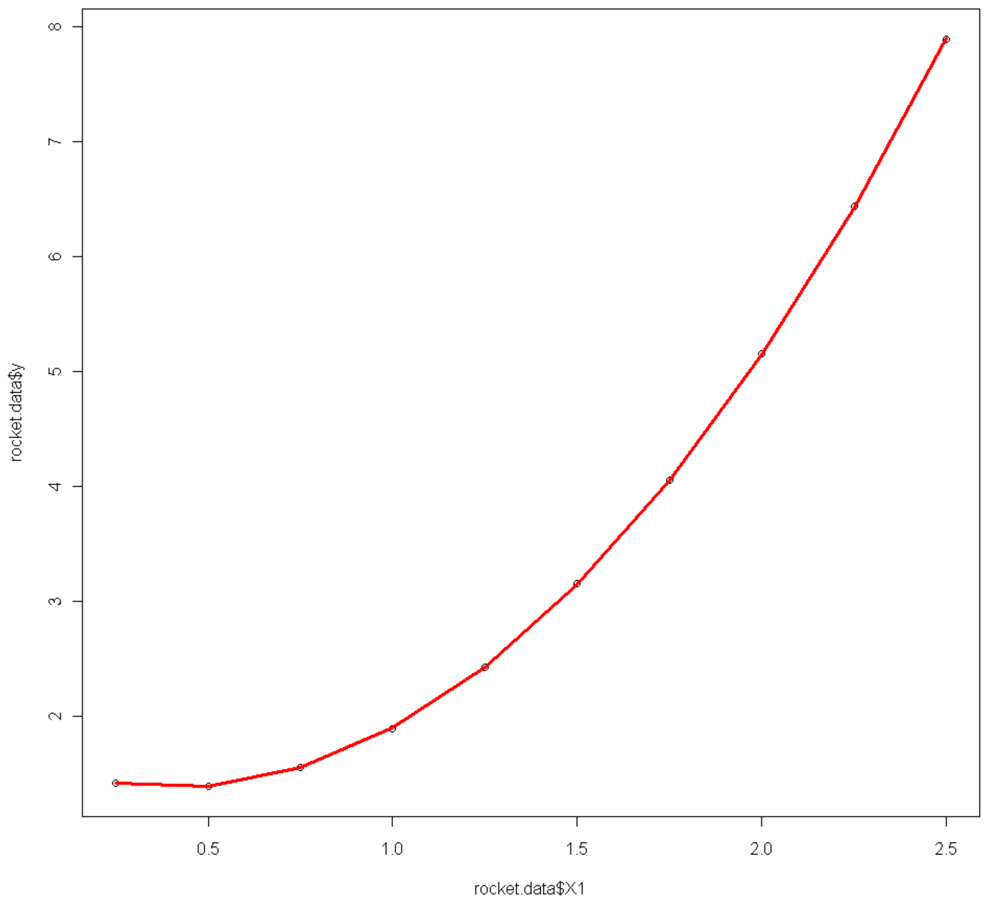
c. Test the hypothesis $H_0 : \beta_2 = 0$

```
> # F test to compare Full model and Reduce model
f.test.lm = function(R.lm, F.lm) {
   SSE.Reduce.Model = sum(resid(R.lm)^2);
   SSE.Full.Model = sum(resid(F.lm)^2);
   Extra.SumSquare = SSE.Reduce.Model - SSE.Full.Model;
   df.num = R.lm$df - F.lm$df
   df.den = F.lm$df;
   F = ( Extra.SumSquare / df.num) / (SSE.Full.Model / df.den);
   p.value = 1 - pf(F, df.num, df.den);

   SSE.data =  data.frame(SSE.Full.Model, SSE.Reduce.Model, Extra.SumSquare);
   F.data = data.frame(F, df.num, df.den, p.value);
   test_result = list(Method="extra-sum-of-squares",SS.Residuals=SSE.data,F.statistic=F.data);
   return(test_result);
}

f.test.lm(weightloss.linear.lm , weightloss.quad.lm);

$Method
[1] "extra-sum-of-squares"

$SS.Residuals
  SSE.Full.Model SSE.Reduce.Model Extra.SumSquare
1   8.909091e-05         4.607025        4.606936

$F.statistic
         F df.num df.den p.value
1 361973.6      1      7       0
```

Using extra-sum-of-squares method to compare the full model (with x^2 term in) with reduce model we have F = 361973.6 and p-value less than 0.0001

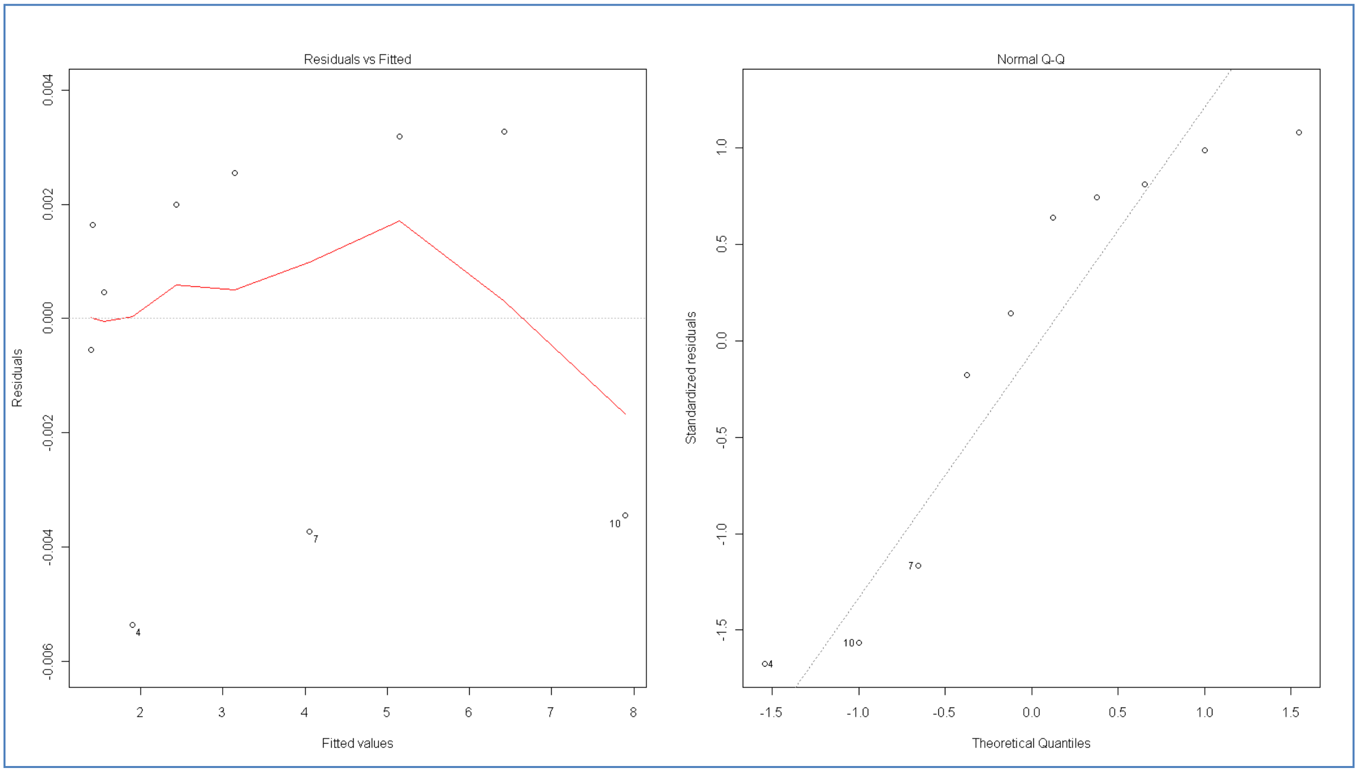
We can reject the hypothesis $H_0 : \beta_2 = 0$

d. Yes. This second-order polynomial model, while fitting the given data below very well, can be potential hazards in extrapolating as most quadratic models fitting in a small value range of x.



3. Exercise 7.3

```
> residuals = resid(weightloss.quad.lm)
> standardized.residuals = rstandard(weightloss.quad.lm)
> studentized.residuals = rstudent(weightloss.quad.lm);
> residuals = data.frame(residuals, standardized.residuals, studentized.residuals);
> residuals
       residuals standardized.residuals studentized.residuals
1   0.0016363636              0.7423075             0.7160016
2  -0.0005454545             -0.1800360            -0.1670682
3   0.0004545455              0.1409896             0.1307168
4  -0.0053636364             -1.6761634            -2.0056738
5   0.0020000000              0.6365013             0.6071164
6   0.0025454545              0.8100926             0.7878386
7  -0.0037272727             -1.1647915            -1.2010436
8   0.0031818182              0.9869275             0.9847982
9   0.0032727273              1.0802161             1.0955578
10 -0.0034545455             -1.5670936            -1.8006985
>
```



- From the computed residuals above… there is no suggestion of outlier.
- There is a problem of normality, but the size of the sample is small so there is no need to read in too much
- The residuals seem to indicate that the quadratic model is adequate.

4. Exercise 7.14

Data, x2 is square of x1.

```
Console C:/Users/th/git/mva/regression/
> X1 = c(0.25,0.5,0.75,1.0,1.25,1.50,1.75,2.0,2.25,2.50);
X2 = X1^2;
y = c(1.42, 1.39, 1.55, 1.89, 2.43, 3.15, 4.05, 5.15, 6.43, 7.89);
rocket.data = data.frame(X1,X2,y);
print(rocket.data);

     X1     X2    y
1  0.25 0.0625 1.42
2  0.50 0.2500 1.39
3  0.75 0.5625 1.55
4  1.00 1.0000 1.89
5  1.25 1.5625 2.43
6  1.50 2.2500 3.15
7  1.75 3.0625 4.05
8  2.00 4.0000 5.15
9  2.25 5.0625 6.43
10 2.50 6.2500 7.89
>
```

a. Fit a second-order model to the data and evaluate the VIF.

```
>
> second.order.model = lm(y ~ X1 + X2);
summary(second.order.model);
vif(second.order.model);


Call:
lm(formula = y ~ X1 + X2)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.005364 -0.002727  0.001045  0.002409  0.003273 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.633000   0.004196   389.2  < 2e-16 ***
X1          -1.232182   0.007010  -175.8 5.09e-14 ***
X2           1.494545   0.002484   601.6  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.003568 on 7 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:      1 
F-statistic: 1.859e+06 on 2 and 7 DF,  p-value: < 2.2e-16

      X1       X2 
19.90625 19.90625 
>
```

- VIF = 19.9 is rather high.

b. Fit a second-order model $y = \beta_0 + \beta_1(x - \bar{x}) + \beta_{11}(x - \bar{x})^2 + \in$ to the data and evaluate the VIF.

```
> rocket.data$centered_X = rocket.data$X1 - mean(rocket.data$X1);
rocket.data$centered_X_square = rocket.data$centered_X^2;
center.x.quadratic.model = lm(y ~ centered_X + centered_X_square, data=rocket.data);
summary(center.x.quadratic.model)
vif(center.x.quadratic.model);


Call:
lm(formula = y ~ centered_X + centered_X_square, data = rocket.data)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.005364 -0.002727  0.001045  0.002409  0.003273 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2.764375   0.001707  1619.6   <2e-16 ***
centered_X        2.877818   0.001571  1831.7   <2e-16 ***
centered_X_square 1.494545   0.002484   601.6   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.003568 on 7 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:      1 
F-statistic: 1.859e+06 on 2 and 7 DF,  p-value: < 2.2e-16

       centered_X centered_X_square 
                1                 1 
>
```

- The fitted model is $\hat{y} = 2.76 + 2.87(x - 1.375) + 1.49(x - 1.375)^2$
- The VIF values are 1.

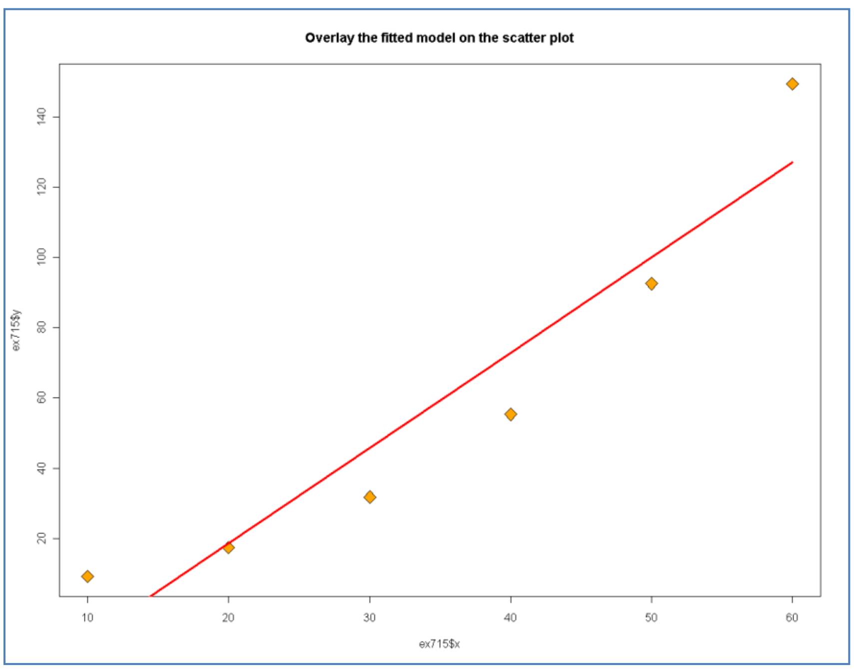
c. The impact of centering the x's in a polynomial model on multicolinearity is good. By centering the x's the VIF values reduce from 19 to 1 and this technique remove the ill-conditioning of the X'X matrix.

5. Exercise 7.15

```
>
> ex715
      y  x
1   9.2 10
2  17.5 20
3  31.8 30
4  55.3 40
5  92.5 50
6 149.4 60
>
```

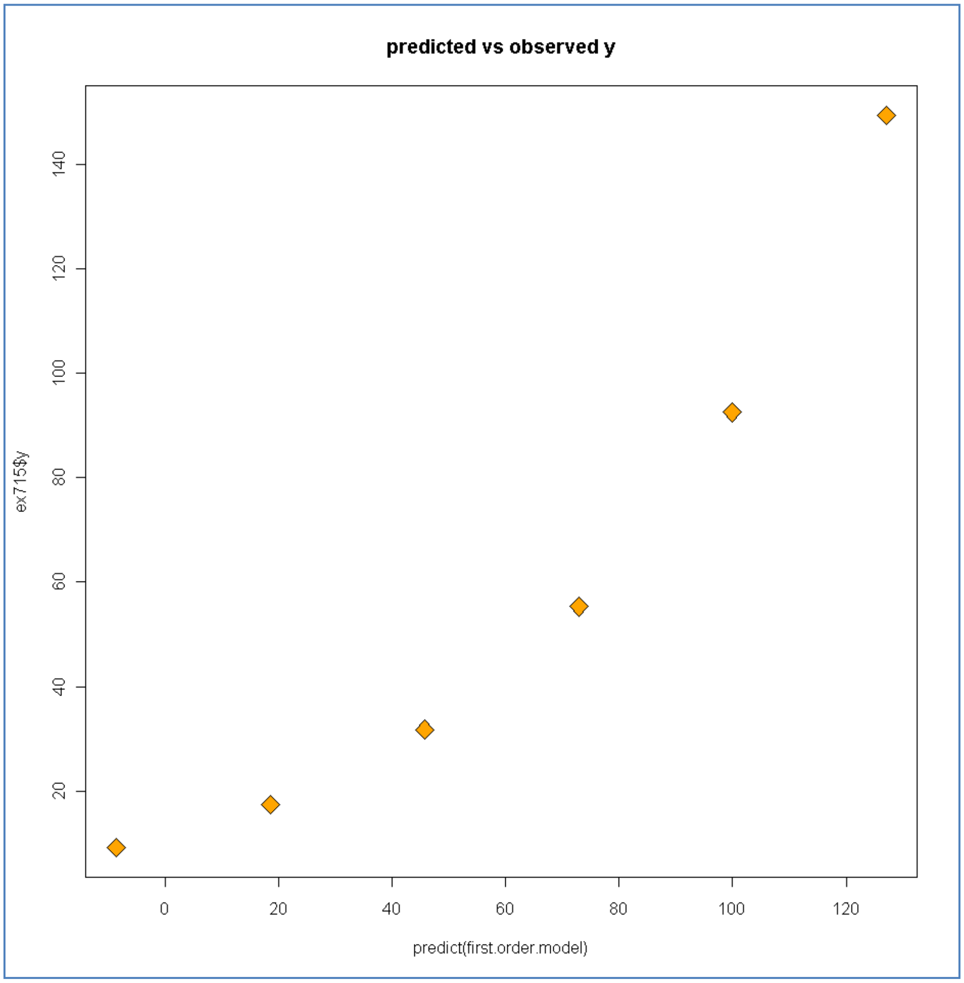
a. Fit a first-order model to the data

```
> first.order.model = lm(y ~ x, data=ex715);
summary(first.order.model);

Call:
lm(formula = y ~ x, data = ex715)

Residuals:
      1       2       3       4       5       6 
 17.738  -1.090 -13.919 -17.548  -7.476  22.295 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept) -35.6667    17.2317  -2.070  0.10725   
x             2.7129     0.4425   6.131  0.00359 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.51 on 4 degrees of freedom
Multiple R-squared: 0.9038, Adjusted R-squared: 0.8798 
F-statistic: 37.59 on 1 and 4 DF,  p-value: 0.003586
```



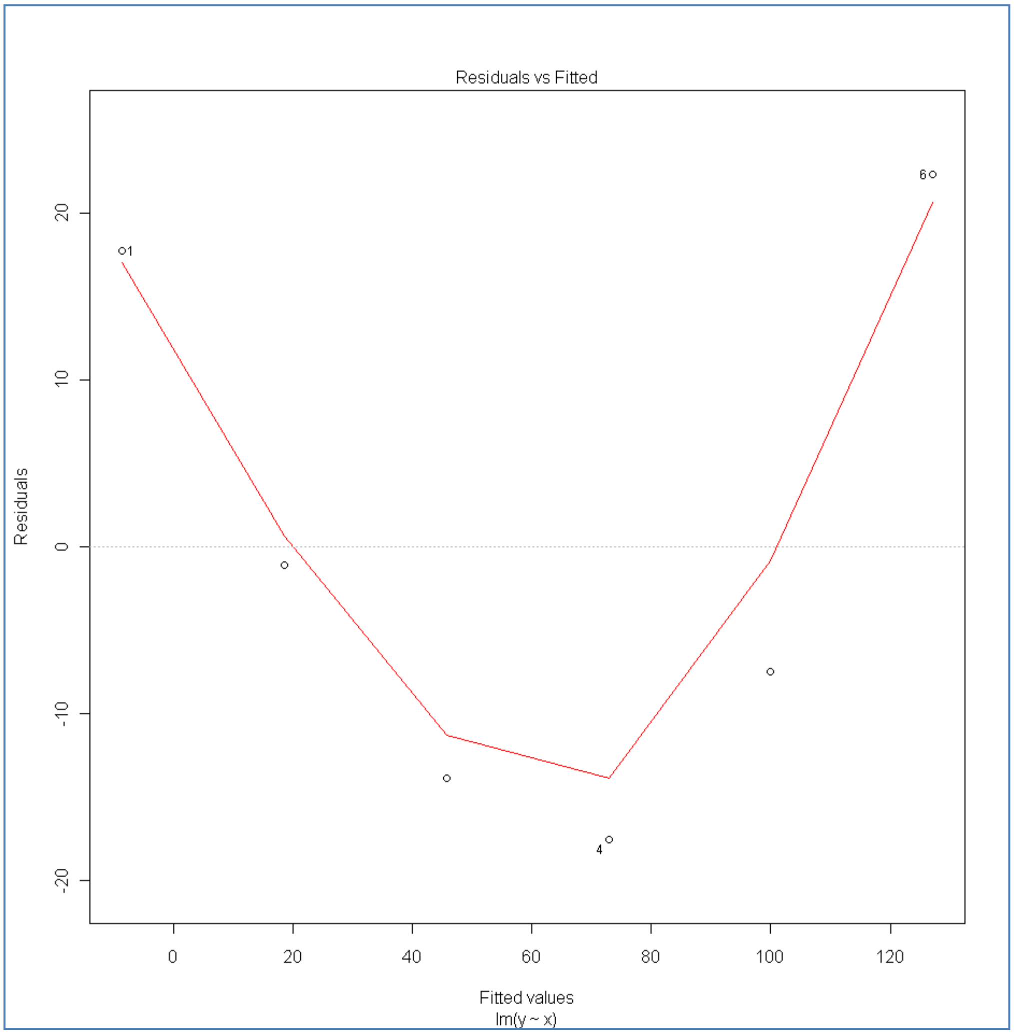
- The fitted model is $\hat{y} = -35.67 + 2.71x$ has high adjusted R square and the test of regression (F.statistic = 37.59) is significant
- However the overlay of fitted model does not appear to follow the data well. The data appear to be nonlinear.

b. Scatter plot of predicted y versus observed y.



- The scatter plot of predicted y.hat vs. observed y suggests the first order model does not fit the data very well.

c. Plot the residuals vs. the fitted y.



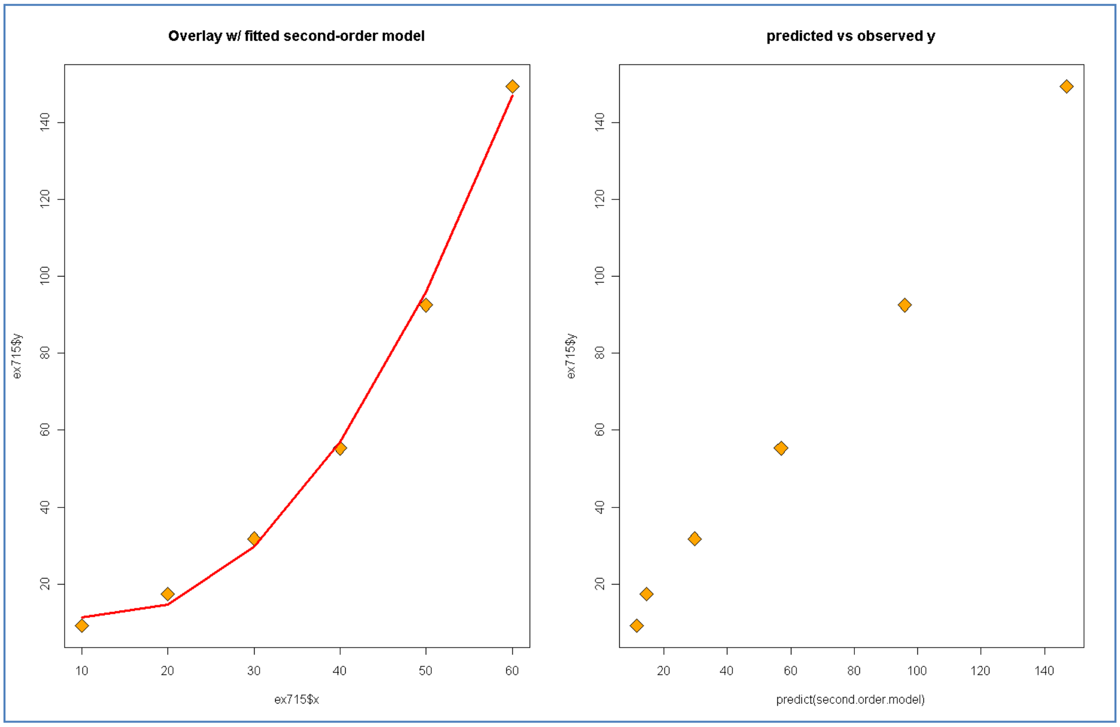
- The residuals vs. fitted y plot suggest the model is inadequate.

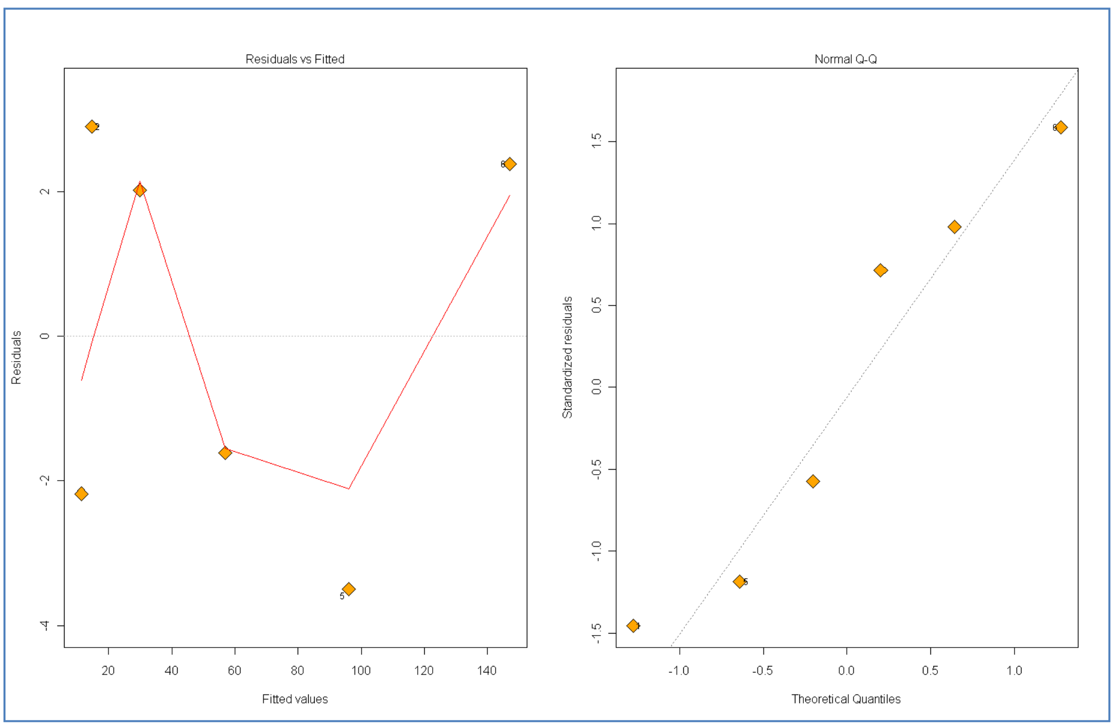
d. Fit a second-order model to the data

```
Console C:/Users/th/git/mva/regression/
> ex715$x2 = ex715$x^2;
second.order.model = lm(y ~ x + x2, data=ex715);
summary(second.order.model);


Call:
lm(formula = y ~ x + x2, data = ex715)

Residuals:
     1      2      3      4      5      6 
-2.179  2.893  2.014 -1.614 -3.493  2.379 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 20.100000   6.335989   3.172  0.05039 . 
x           -1.469643   0.414518  -3.545  0.03822 * 
x2           0.059750   0.005797  10.307  0.00195 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.542 on 3 degrees of freedom
Multiple R-squared: 0.9974, Adjusted R-squared: 0.9956 
F-statistic: 566.4 on 2 and 3 DF,  p-value: 0.0001357 

> anova(first.order.model, second.order.model);
Analysis of Variance Table

Model 1: y ~ x
Model 2: y ~ x + x2
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)   
1      4 1370.46                                
2      3   37.64  1    1332.8 106.24 0.001948 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> 
```

- The fitted second-order model is $\hat{y} = 20 - 1.47x + 0.06x^2$
- Using anova (similar to extra-sum-of-squares method) to compare the two models and the result suggest the quadratic term is significant (F=106.24, p-value=0.0019).

e. Overlay second-order model on scatter plot. Plot predicted vs. observed y and residuals vs. fitted y.





- The overlay of fitted second-order model on the scatter plot suggest this model fit the data well
- The predicted vs. observed y suggest predicted values do not deviate much from the observed ones.
- The residuals vs. fitted values plot suggest the residuals spread more evenly comparing to the first model. This plot, together with the probability plot suggests the there is no serious problem with model adequacy.

6. Exercise 8.3

Dataset:

- city=1 (San Diego)
- city=2 (Boston)
- city=3 (Austin)
- city=4 (Minneapolis)

```
>
> ex83 <- read.csv("C:/Users/th/git/mva/regression/ex83.csv");
ex83$city <- factor(ex83$city);
ex83
       y x1   x2 city
1  16.68  7  560    1
2  11.50  3  220    1
3  12.03  3  340    1
4  14.88  4   80    1
5  13.75  6  150    1
6  18.11  7  330    1
7   8.00  2  110    1
8  17.83  7  210    2
9  79.24 30 1460    2
10 21.50  5  605    2
11 40.33 16  688    2
12 21.00 10  215    2
13 13.50  4  255    2
14 19.75  6  462    2
15 24.00  9  448    2
16 29.00 10  776    2
17 15.35  6  200    2
18 19.00  7  132    3
19  9.50  3   36    3
20 35.10 17  770    3
21 17.90 10  140    3
22 52.32 26  810    3
23 18.75  9  450    3
24 19.83  8  635    4
25 10.75  4  150    4
```

a. A model that relate delivery time y to cases x1, distance x2 and city

```
> delivery.site.model = lm(y ~ x1 + x2 + city, data=ex83);
summary(delivery.site.model)
model.matrix(delivery.site.model);

Call:
lm(formula = y ~ x1 + x2 + city, data = ex83)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.4800 -1.5922 -0.5583  1.1045  6.1611 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.701347   1.240109   2.178  0.04218 *  
x1           1.770277   0.186790   9.477 1.24e-08 ***
x2           0.010833   0.003786   2.862  0.00999 ** 
city2        1.452538   1.583004   0.918  0.37034    
city3       -2.737737   1.936269  -1.414  0.17356    
city4       -2.285101   2.416243  -0.946  0.35616    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.986 on 19 degrees of freedom
Multiple R-squared: 0.9707, Adjusted R-squared: 0.963 
F-statistic: 125.9 on 5 and 19 DF,  p-value: 6.919e-14
```

- Since city is a factor/categorical variable… R internally created the design matrix as follow

```
> model.matrix(delivery.site.model);
   (Intercept) x1   x2 city2 city3 city4
1            1  7  560     0     0     0
2            1  3  220     0     0     0
3            1  3  340     0     0     0
4            1  4   80     0     0     0
5            1  6  150     0     0     0
6            1  7  330     0     0     0
7            1  2  110     0     0     0
8            1  7  210     1     0     0
9            1 30 1460     1     0     0
10           1  5  605     1     0     0
11           1 16  688     1     0     0
12           1 10  215     1     0     0
13           1  4  255     1     0     0
14           1  6  462     1     0     0
15           1  9  448     1     0     0
16           1 10  776     1     0     0
17           1  6  200     1     0     0
18           1  7  132     0     1     0
19           1  3   36     0     1     0
20           1 17  770     0     1     0
21           1 10  140     0     1     0
22           1 26  810     0     1     0
23           1  9  450     0     1     0
24           1  8  635     0     0     1
25           1  4  150     0     0     1
attr(,"assign")
[1] 0 1 2 3 3 3
attr(,"contrasts")
attr(,"contrasts")$city
[1] "contr.treatment"
```

Indicator coded variables			
city2	**city3**	**city4**	**Interpretation**
0	0	0	Observation from San Diego
1	0	0	Observation from Boston
0	1	0	Observation from Austin
0	0	1	Observation from Minneapolis

- The estimated parameters of the model is

$$\hat{y} = 2.7 + 1.77x_1 + 0.01x_2 + 1.45city_2 - 2.74city_3 - 2.28city_4$$

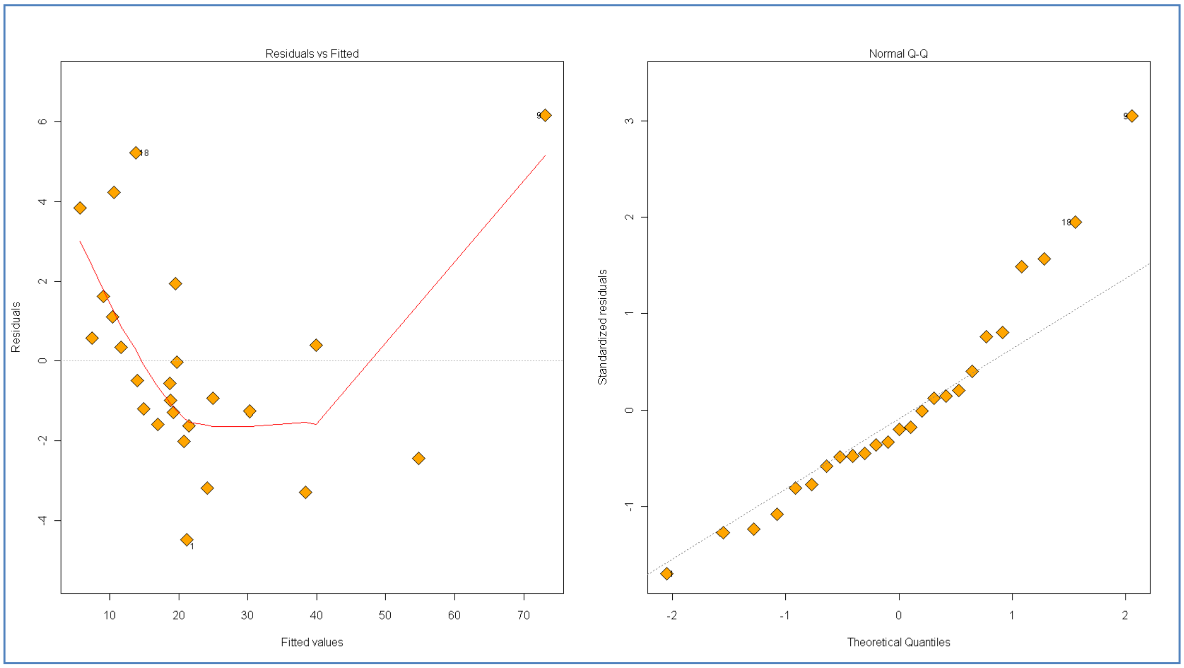
b. Is delivery site (city) is an important variable?

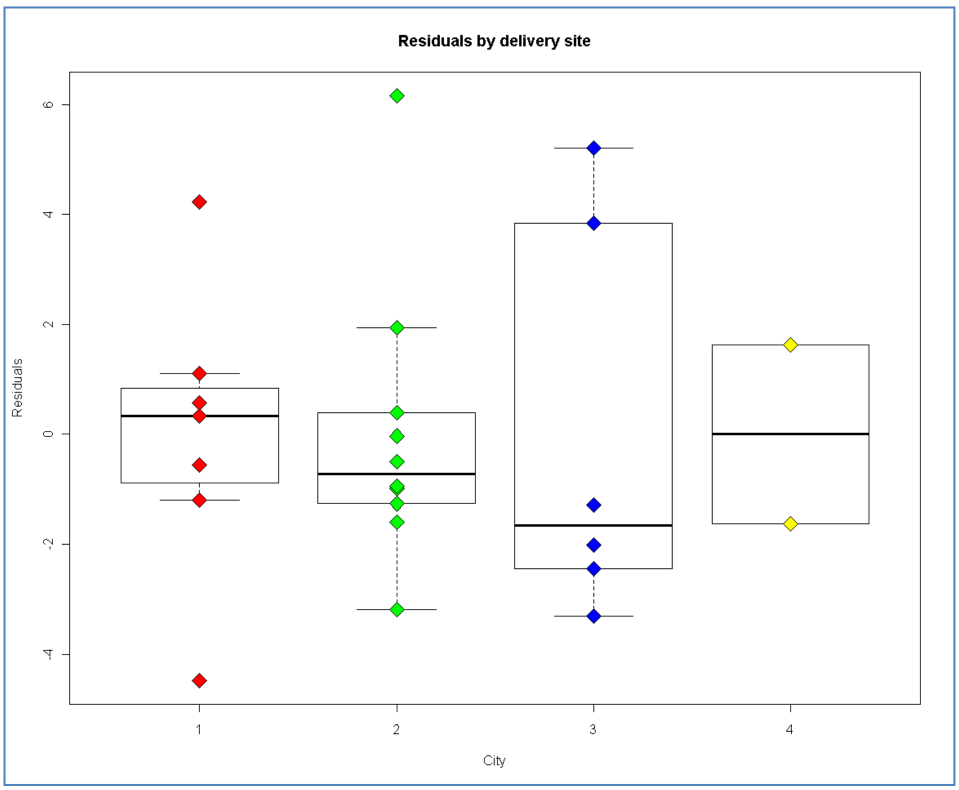
Use extra-sum-of-squares method to compare ***delivery.site.model*** model with a reduced model that removes delivery site as an explained variable.

```
> without.delivery.site.model = lm(y ~ x1 + x2, data=ex83);
f.test.lm(without.delivery.site.model, delivery.site.model);

$Method
[1] "extra-sum-of-squares"

$SS.Residuals
  SSE.Full.Model SSE.Reduce.Model Extra.SumSquare
1       169.4511         233.7317        64.28055

$F.statistic
         F df.num df.den    p.value
1 2.402522      3     19 0.09946447
```

- F = 2.4, p-value = 0.099 suggest delivery site is not an important variable

c. Analyze the residuals





- Observation 9 has large residual

7. Exercise 8.4

a. Fit a linear model relating y to x1 and x11

```
>
> B3.table$x11 <- factor(B3.table$x11);
transmission.type.model = lm(y ~ x1 + x11, data=B3.table);
summary(transmission.type.model);


Call:
lm(formula = y ~ x1 + x11, data = B3.table)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.9153 -1.8882  0.1106  1.7706  6.7829 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 33.618408   1.539505  21.837  < 2e-16 ***
x1          -0.045736   0.008682  -5.268 1.20e-05 ***
x111        -0.498689   2.228198  -0.224    0.824    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.115 on 29 degrees of freedom
Multiple R-squared: 0.7727, Adjusted R-squared: 0.757 
F-statistic: 49.28 on 2 and 29 DF,  p-value: 4.696e-10
```

- The fitted model is $\hat{y} = 33.6 - 0.0457x_1 - 0.5x_{11}$
- The ***t*** statistic = -0.22 with p-value = 0.824 suggest the type of transmission does NOT significantly effect the mileage performance.

b. Fit an interaction model

```
> interaction.model = lm(y ~ x1 * x11, data=B3.table);
no.interaction.model = lm(y ~ x1 + x11, data=B3.table);
summary(interaction.model);


Call:
lm(formula = y ~ x1 * x11, data = B3.table)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.2712 -1.2660  0.1342  1.5181  4.6599 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  42.91963    2.73493  15.693 2.10e-15 ***
x1           -0.11677    0.01984  -5.886 2.49e-06 ***
x111        -13.46371    3.84413  -3.502 0.001567 ** 
x1:x111       0.08165    0.02127   3.839 0.000647 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.566 on 28 degrees of freedom
Multiple R-squared: 0.851, Adjusted R-squared: 0.8351 
F-statistic: 53.33 on 3 and 28 DF,  p-value: 1.064e-11

> anova(no.interaction.model, interaction.model);

Analysis of Variance Table

Model 1: y ~ x1 + x11
Model 2: y ~ x1 * x11
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1     29 281.34                                  
2     28 184.34  1    97.003 14.734 0.0006468 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> |
```

- The fitted model is $\hat{y} = 42.92 - 0.117x_1 - 13.46x_{11} + 0.082x_1x_{11}$
- There is significant interaction between engine displacement and the type of transmission
- When transmission is automatic (x11=1),
  - $\hat{y} = (42.92 - 13.46) + (-0.117 + 0.082)x_1 = 29.46 - 0.035\,x_1$

  This suggest that… on average… for one cubic inch increase in engine displacement, miles per gallon decreases by 0.035

- When transmission is automatic (x11=0),
  - $\hat{y} = 42.91 - 0.117\,x_1$

  This suggest that… on average… for one cubic inch increase in engine displacement, miles per gallon decreases by 0.117

8. Exercise 8.5

a. Fit a linear model relating y to x10 and x11

```
> x10.and.transmission.type.model = lm(y ~ x10 + x11, data=B3.table);
summary(x10.and.transmission.type.model);


Call:
lm(formula = y ~ x10 + x11, data = B3.table)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.0711 -1.8726 -0.0827  2.3078  6.8477 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 39.1919052  2.5570509  15.327 1.92e-15 ***
x10         -0.0047484  0.0009544  -4.975 2.72e-05 ***
x111        -2.6958431  1.9805597  -1.361    0.184    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.2 on 29 degrees of freedom
Multiple R-squared:  0.76, Adjusted R-squared: 0.7434 
F-statistic: 45.91 on 2 and 29 DF,  p-value: 1.032e-09
```

- The fitted model is $\hat{y} = 39.19 - 0.047x_{10} - 2.69x_{11}$
- The ***t*** statistic = -1.36 with p-value = 0.184 suggest the type of transmission does NOT significantly effect the mileage performance.

b. Fit an interaction model

```
> interaction.model = lm(y ~ x10 * x11, data=B3.table);
no.interaction.model = lm(y ~ x10 + x11, data=B3.table);
summary(interaction.model);


Call:
lm(formula = y ~ x10 * x11, data = B3.table)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5063 -1.7205  0.2403  1.3557  4.8855 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  58.108420   5.077985  11.443 4.53e-12 ***
x10          -0.012517   0.002055  -6.090 1.44e-06 ***
x111        -26.724910   6.107349  -4.376 0.000152 ***
x10:x111      0.009035   0.002217   4.076 0.000342 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.58 on 28 degrees of freedom
Multiple R-squared: 0.8494, Adjusted R-squared: 0.8332 
F-statistic: 52.63 on 3 and 28 DF,  p-value: 1.244e-11

> anova(no.interaction.model, interaction.model);
Analysis of Variance Table

Model 1: y ~ x10 + x11
Model 2: y ~ x10 * x11
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1     29 297.04                                  
2     28 186.42  1    110.62 16.616 0.0003424 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

- The fitted model is $\hat{y} = 58.10 - 0.0125x_{10} - 26.2x_{11} + 0.009x_{10}x_{11}$
- There is significant interaction between vehicle weight and the type of transmission
- When transmission is automatic (x11=1),
    - $\hat{y} = (58.10 - 26.2) + (-0.0125 + 0.009)x_{10} = 31.9 - 0.0035\, x_{10}$

  Which indicate that… on average… for one lb increase in vehicle weight, miles per gallon decreases by 0.0035.

- When transmission is automatic (x11=0),
    - $\hat{y} = 58.1 - 0.0125\, x_{10}$

  This suggest that… on average… for one lb increase in vehicle weight, miles per gallon decreases by 0.0125

9. Exercise 8.9

HW - Ex 8.9

One-way ANOVA, 4 treatments

$n_1 = 3, n_2 = 2, n_3 = 4, n_4 = 3.$

- The y vector is

$$\begin{bmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \\ y_{33} \\ y_{34} \\ y_{41} \\ y_{42} \\ y_{43} \end{bmatrix}$$

- No complication introduced by the unbalanced nature of these data.

- The estimated parameters for this multiple regression model is

$$\boxed{\begin{aligned} \hat{\beta}_0 &= \bar{Y}_{..} - \bar{Y}_{1.} - \bar{Y}_{2.} - \bar{Y}_{3.} = \bar{Y}_{4.} \\ \hat{\beta}_1 &= \bar{Y}_{1.} - \bar{Y}_{4.};\ \hat{\beta}_2 = \bar{Y}_{2.} - \bar{Y}_{4.} \\ \hat{\beta}_3 &= \bar{Y}_{3.} - \bar{Y}_{4.} \end{aligned}}$$

- The X matrix is

$$X = \begin{array}{c} \begin{array}{cccc} & x_1 & x_2 & x_3 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

10. Exercise 8.10

Exercise 8.10 :

Eq 8.18

$y_{ij} = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \varepsilon_{ij}$, $i = 1,2,3$
$j = 1,2,\dots n.$

represents the regression model corresponding to the ANOVA with 3 treatments and n observations

a) show that

$$\beta_0 = \frac{\mu_1 + \mu_2 + \mu_3}{3} = \bar{\mu}$$

$$\beta_1 = \mu_1 - \bar{\mu}, \quad \beta_2 = \mu_2 - \bar{\mu}.$$

we have:

$$\left.\begin{aligned} y_{1j} &= \beta_0 + \beta_1 \underbrace{x_{1j}}_{1} + \beta_2 \underbrace{x_{2j}}_{0} + \varepsilon_{1j} \\ &= \beta_0 + \beta_1 + \varepsilon_{1j} \end{aligned}\right\} \Rightarrow \mu_1 = \beta_0 + \beta_1$$

$$y_{2j} = \beta_0 + \beta_2 + \varepsilon_{2j} \quad \} \Rightarrow \mu_2 = \beta_0 + \beta_2$$

$$\left.\begin{aligned} y_{3j} &= \beta_0 + \beta_1 \underbrace{x_{1j}}_{-1} + \beta_2 \underbrace{x_{2j}}_{-1} + \varepsilon_{3j} \\ &= \beta_0 - \beta_1 - \beta_2 \end{aligned}\right\} \Rightarrow \mu_3 = \beta_0 - \beta_1 - \beta_2$$

$$\Rightarrow \mu_1 + \mu_2 + \mu_3 = 3\beta_0 + \beta_1 + \beta_2 - \beta_1 - \beta_2 = 3\beta_0$$

$$\Rightarrow \boxed{\beta_0 = \frac{\mu_1 + \mu_2 + \mu_3}{3} = \frac{\mu_1 + \mu_2 + \mu_3}{3}}$$

Exercise 8.10 – Cont.

a) $\mu_1 = \dfrac{\mu_1 + \mu_2 + \mu_3}{3} + \beta_1 = \bar{\mu} + \beta_1$

$\Rightarrow \boxed{\beta_1 = \mu_1 - \beta_0 = \mu_1 - \bar{\mu}}$

$\mu_2 = \beta_0 + \beta_2 \Rightarrow \boxed{\beta_2 = \mu_2 - \beta_0 = \mu_2 - \bar{\mu}}$

b) Write down y vector and X matrix

$$Y = \begin{bmatrix} y_{11} \\ y_{12} \\ \vdots \\ y_{1n} \\ y_{21} \\ y_{22} \\ \vdots \\ y_{2n} \\ y_{31} \\ y_{32} \\ \vdots \\ y_{3n} \end{bmatrix} \qquad X = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ \vdots & \vdots & \vdots \\ 1 & -1 & -1 \end{bmatrix}$$

c) $SSR(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2) = \hat{\beta}' X' Y$

$$= \left[\bar{y}_{..},\ \bar{y}_{1.} - \bar{y}_{..},\ \bar{y}_{2.} - \bar{y}_{..}\right] \begin{bmatrix} y_{..} \\ y_{1.} - y_{3.} \\ y_{2.} - y_{3.} \end{bmatrix}$$

$$= y_{..}\bar{y}_{..} + (y_{1.} - y_{3.})(\bar{y}_{1.} - \bar{y}_{..}) + (y_{2.} - y_{3.})(\bar{y}_{2.} - \bar{y}_{..})$$

$$= (y_{1.} + y_{2.} + y_{3.})\bar{y}_{..} + y_{1.}(\bar{y}_{1.} - \bar{y}_{..}) + y_{2.}(\bar{y}_{2.} - \bar{y}_{..}) - y_{3.}(\bar{y}_{1.} + \bar{y}_{2.} - 2\bar{y}_{..})$$

$$= y_{1.}\bar{y}_{1.} + y_{2.}\bar{y}_{2.} + y_{3.}(3\bar{y}_{..} - \bar{y}_{1.} - \bar{y}_{2.})$$

$= y_{1.}\bar{y}_{1.} + y_{2.}\bar{y}_{2.} + y_{3.}\bar{y}_{3.}$ The same as the usuall SS.

11. Exercise 8.11

a. Write down y vector and X matrix for the corresponding regression model

```
> # y vector
y = ex811.table$y;
print(y);

 [1]  7  7 15 11  9 12 17 12 18 18 14 18 18 19 19 19 25 22 19 23  7 10 11 15 11
> X = model.matrix(anova.model);
print(X);
   (Intercept) x1 x2 x3 x4
1            1  1  0  0  0
2            1  1  0  0  0
3            1  1  0  0  0
4            1  1  0  0  0
5            1  1  0  0  0
6            1  0  1  0  0
7            1  0  1  0  0
8            1  0  1  0  0
9            1  0  1  0  0
10           1  0  1  0  0
11           1  0  0  1  0
12           1  0  0  1  0
13           1  0  0  1  0
14           1  0  0  1  0
15           1  0  0  1  0
16           1  0  0  0  1
17           1  0  0  0  1
18           1  0  0  0  1
19           1  0  0  0  1
20           1  0  0  0  1
21           1  0  0  0  0
22           1  0  0  0  0
23           1  0  0  0  0
24           1  0  0  0  0
25           1  0  0  0  0
```

b. Find the estimates of the model parameters

```
> summary(anova.model)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4, data = ex811.table)

Residuals:
   Min     1Q Median     3Q    Max 
  -3.8   -2.6    0.4    1.4    5.2 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   10.800      1.270   8.506 4.46e-08 ***
x1            -1.000      1.796  -0.557  0.58375    
x2             4.600      1.796   2.562  0.01859 *  
x3             6.800      1.796   3.787  0.00116 ** 
x4            10.800      1.796   6.015 7.01e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.839 on 20 degrees of freedom
Multiple R-squared: 0.7469, Adjusted R-squared: 0.6963 
F-statistic: 14.76 on 4 and 20 DF,  p-value: 9.128e-06
```

- The estimated $\widehat{\beta_0} = 10.8$
- The estimated $\widehat{\beta_1} = -1$
- The estimated $\widehat{\beta_2} = 4.6$
- The estimated $\widehat{\beta_3} = 6.8$
- The estimated $\widehat{\beta_4} = 10.8$

c. Find point estimate of the difference in mean strength between 15% and 25% cotton

- The estimated difference is $\bar{y}_{1.} - \bar{y}_{3.} = -7.8$
- The estimated difference is $\widehat{\beta_1} - \widehat{\beta_3} = -1 - 6.8 = -7.8$

d. The F statistic F=14.76 with p-value < 0.0001 indicates that the mean tensile strength is not the same for all cotton percentages

12. Exercise 10.9

a. Sample 8 random rows and call them as **test.set.** The rest of is put into a **training.set**

```
> B3.table <- read.csv("C:/Users/th/git/mva/regression/B3.table.csv");
B3.table$x11 <- factor(B3.table$x11);

all.rows = 1:32;
test.rows = sample(all.rows, 8, replace=F);
print(all.rows);
print(test.rows);

 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
[1] 14 11  7  1 13 32 22 10
> train.rows = all.rows[!(all.rows %in% test.rows)];
print(train.rows);

 [1]  2  3  4  5  6  8  9 12 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31
>
```

```
> test.set = B3.table[test.rows,];
print(test.set);

       y    x1  x2  x3  x4   x5 x6 x7    x8   x9  x10 x11
14 19.70 258.0 110 195 8.0 3.08  1  3 171.5 77.0 3375   1
11 16.50 350.0 155 250 8.5 3.08  4  3 195.4 74.4 3885   1
7  22.12 231.0 110 175 8.0 2.56  2  3 179.3 65.4 3020   1
1  18.90 350.0 165 260 8.0 2.56  4  3 200.3 69.9 3910   1
13 21.50 171.0 109 146 8.2 3.22  2  4 170.4 66.9 2655   0
32 16.50 360.0 165 255 8.5 2.73  4  3 185.2 69.0 3660   1
22 21.47 360.0 180 290 8.4 2.45  2  3 214.2 76.3 4250   1
10 30.40  96.9  75  83 9.0 4.30  2  5 165.2 65.0 2320   0
>
```

```
> train.set = B3.table[train.rows,];
print(train.set);

       y    x1  x2  x3   x4   x5 x6 x7    x8   x9  x10 x11
2  17.00 350.0 170 275 8.50 2.56  4  3 199.6 72.9 3860   1
3  20.00 250.0 105 185 8.25 2.73  1  3 196.7 72.2 3510   1
4  18.25 351.0 143 255 8.00 3.00  2  3 199.9 74.0 3890   1
5  20.07 225.0  95 170 8.40 2.76  1  3 194.1 71.8 3365   0
6  11.20 440.0 215 330 8.20 2.88  4  3 184.5 69.0 4215   1
8  21.47 262.0 110 200 8.50 2.56  2  3 179.3 65.4 3180   1
9  34.70  89.7  70  81 8.20 3.90  2  4 155.7 64.0 1905   0
12 36.50  85.3  80  83 8.50 3.89  2  4 160.6 62.2 2009   0
15 20.30 140.0  83 109 8.40 3.40  2  4 168.8 69.4 2700   0
16 17.80 302.0 129 220 8.00 3.00  2  3 199.9 74.0 3890   1
17 14.39 500.0 190 360 8.50 2.73  4  3 224.1 79.8 5290   1
18 14.89 440.0 215 330 8.20 2.71  4  3 231.0 79.7 5185   1
19 17.80 350.0 155 250 8.50 3.08  4  3 196.7 72.2 3910   1
20 16.41 318.0 145 255 8.50 2.45  2  3 197.6 71.0 3660   1
21 23.54 231.0 110 175 8.00 2.56  2  3 179.3 65.4 3050   1
23 16.59 400.0 185  NA 7.60 3.08  4  3 196.0 73.0 3850   1
24 31.90  96.9  75  83 9.00 4.30  2  5 165.2 61.8 2275   0
25 29.40 140.0  86  NA 8.00 2.92  2  4 176.4 65.4 2150   0
26 13.27 460.0 223 366 8.00 3.00  4  3 228.0 79.8 5430   1
27 23.90 133.6  96 120 8.40 3.91  2  5 171.5 63.4 2535   0
28 19.73 318.0 140 255 8.50 2.71  2  3 215.3 76.3 4370   1
29 13.90 351.0 148 243 8.00 3.25  2  3 215.5 78.5 4540   1
30 13.27 351.0 148 243 8.00 3.26  2  3 216.1 78.5 4715   1
31 13.77 360.0 195 295 8.25 3.15  4  3 209.3 77.4 4215   1
```

b. Find a appropriate regression model
   i. Fit initial (full) model
   ii. Use **All Possible Regressions** (leaps package in R) with **adjusted R^2** and **Cp** criteria to find the best subset of explanatory variables

```
> x3.subset = !is.na(B3.table$x3);
full.model = lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10, data=B3.table, subset=x3.subset);
X = model.matrix(full.model)[,-1];
y = B3.table$y[x3.subset];

# leaps try to find best model out of all possible models for a given criterion (adjr2 or Cp)
adjr2_models = leaps(X, y, nbest=3, method='adjr2');
#plot(adjr2_models$size, adjr2_models$adjr2, pch=23, bg='orange', cex=2);
best.model.adjr2 = adjr2_models$which[which((adjr2_models$adjr2 == max(adjr2_models$adjr2))),];
print(best.model.adjr2);

Cp_models = leaps(X, y, nbest=3, method='Cp');
#plot(Cp_models$size, Cp_models$Cp, pch=23, bg='orange', cex=2);
best.model.Cp = Cp_models$which[which((Cp_models$Cp == min(Cp_models$Cp))),]
print(best.model.Cp);

    1     2     3     4     5     6     7     8     9     A
FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
    1     2     3     4     5     6     7     8     9     A
FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
>
```

- The best subset in terms of **adjusted R^2** is $y \sim x_5 + x_8 + x_{10} + \in$
- The best subset in terms of Mallows **Cp statistic** is $y \sim x_5 + x_8 + x_{10} + \epsilon$

iii. Use **stepwise** regression to select explanatory variables

```
Step:  AIC=69.6
y ~ x5 + x8 + x9 + x10

       Df Sum of Sq    RSS    AIC
- x9    1     5.097 223.82 68.290
<none>              218.73 69.599
+ x7    1     3.713 215.01 71.085
+ x4    1     1.326 217.40 71.416
+ x1    1     0.765 217.96 71.494
+ x2    1     0.684 218.04 71.505
+ x6    1     0.547 218.18 71.524
+ x3    1     0.316 218.41 71.555
- x5    1    40.404 259.13 72.684
- x8    1    57.407 276.13 74.591
- x10   1   135.105 353.83 82.029

Step:  AIC=68.29
y ~ x5 + x8 + x10

       Df Sum of Sq    RSS    AIC
<none>              223.82 68.290
+ x9    1     5.097 218.73 69.599
+ x4    1     2.730 221.09 69.922
+ x3    1     1.642 222.18 70.069
+ x7    1     0.610 223.21 70.208
+ x6    1     0.137 223.69 70.272
+ x2    1     0.017 223.81 70.288
+ x1    1     0.000 223.82 70.290
- x5    1    36.314 260.14 70.800
- x8    1    52.960 276.78 72.661
- x10   1   194.838 418.66 85.076
>
```

- **stepwise also suggest** $y \sim x_5 + x_8 + x_{10} + \in$

c. Fit the model $y \sim x_5 + x_8 + x_{10} + \in$ to the training data (train.set) to estimate the parameters

```
> # this model is suggested by both stepwise and "all-possible-regression" method
choosen.model = lm(y ~ x5 + x8 + x10, data=train.set);
summary(choosen.model);


Call:
lm(formula = y ~ x5 + x8 + x10, data = train.set)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.4346 -1.9712 -0.5516  2.5694  5.6762 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  7.063328  15.185213   0.465  0.64685    
x5           3.037542   1.551171   1.958  0.06430 .  
x8           0.185148   0.104215   1.777  0.09085 .  
x10         -0.008855   0.002144  -4.130  0.00052 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.064 on 20 degrees of freedom
Multiple R-squared: 0.8287, Adjusted R-squared: 0.803 
F-statistic: 32.25 on 3 and 20 DF,  p-value: 7.4e-08

> #compute the PRESS statistic
PRESS(choosen.model);

[1] 275.4704
>
```

- The fitted model is $\hat{y} \sim 7.06 + 3.037x_5 + 0.185x_8 - 0.0088\, x_{10}$
- The PRESS statistic of this model is 275.47

d. Use the fitted model $\hat{y} \sim 7.06 + 3.037x_5 + 0.185x_8 - 0.0088\, x_{10}$ to predict 8 withheld observations in the test.set

i. Use the training model to predict the unseen data in the test.set
ii. Compute the prediction error
iii. Compute the average prediction error (Root Means Square Error) as a way to assess the model predictive power

```
>
> #compute predicted y for the test.set
y.predicted = predict(choosen.model, test.set);
y.observed = test.set$y;
predicted.error = y.observed - y.predicted
predicted.set = data.frame(y.observed, y.predicted, predicted.error);

# print root mean square error (RMSE)
RMSE = sqrt(sum(predicted.error^2)/nrow(test.set));

predicted.performance = list(prediction.data=predicted.set, RMSE=RMSE);
print(predicted.performance);

$prediction.data
   y.observed y.predicted predicted.error
14      19.70    18.28510       1.4148952
11      16.50    18.19392      -1.6939239
7       22.12    21.29338       0.8266161
1       18.90    17.30024       1.5997554
13      21.50    24.88254      -3.3825414
32      16.50    17.23472      -0.7347238
22      21.47    16.52886       4.9411403
10      30.40    30.16685       0.2331453

$RMSE
[1] 2.360203
```

- The model is $\hat{y} \sim 7.06 + 3.037x_5 + 0.185x_8 - 0.0088\, x_{10}$ is predicting pretty well

13. Exercise 11.12

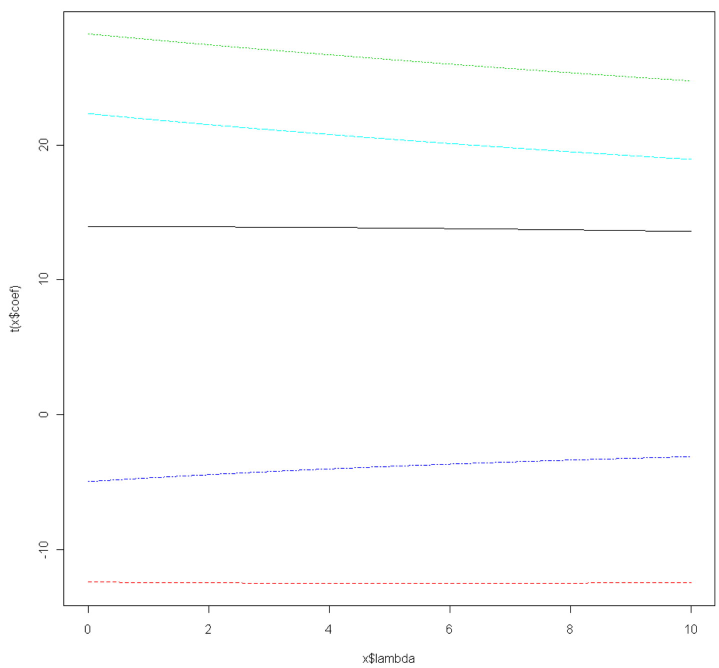
- There is one large condition index ($\eta_9 = 208.285 > 30$). Thus there is one dependence in the column of the design matrix X
- The variance decomposition proportions $\pi_T,\ \pi_C,\ \pi_{TC},\ \pi_{T2}, and\ \pi_{C2}$ all exceed 0.5, indicating that the following regressors are involved in multicolinearity relationsip

14. Exercise 11.19

a. There is no evidence of multicolinearity in this dataset
    i. Why: Compute the VIFs and see all of these values are under 2.0

```
> #Ex 11.19
B15.table <- read.csv("C:/Users/th/git/mva/regression/B15.csv");
ols.model = lm(MORT ~ PRECIP +  EDUC + NONWHITE + NOX + SO2, data=B15.table);
vif(ols.model);

  PRECIP      EDUC  NONWHITE       NOX       SO2
2.030523 1.513351 1.315836 1.681246 1.425845
```

b. Perform the ridge trace on the data



- The ridge trace shows flat lines

c. The ridge trace suggest k=0, hence the estimates of the coefficients for ridge and ordinary Least Square are the same.

d. Principal component regression gives

```
> print(pcr.model$coefficients);
, , 1 comps

               MORT
PRECIP    16.134258
EDUC     -12.331239
NONWHITE   8.700157
NOX      -11.851176
SO2       -2.391118

, , 2 comps

               MORT
PRECIP    16.345937
EDUC     -21.212931
NONWHITE  20.642662
NOX        2.267678
SO2       18.316435

, , 3 comps

               MORT
PRECIP    17.081439
EDUC     -16.845349
NONWHITE  26.595799
NOX        3.593442
SO2       15.844978

, , 4 comps

               MORT
PRECIP    17.510034
EDUC     -11.045776
NONWHITE  26.718602
NOX       -3.186863
SO2       22.880231

, , 5 comps

               MORT
PRECIP    14.051840
EDUC     -12.511607
NONWHITE  28.474830
NOX       -5.006647
SO2       22.514784
```

```
> print(pcr.model$loadings);

Loadings:
         Comp 1 Comp 2 Comp 3 Comp 4 Comp 5
PRECIP    0.641                       0.761
EDUC     -0.490  0.305  0.551 -0.510  0.323
NONWHITE  0.345 -0.410  0.750        -0.387
NOX      -0.471 -0.484  0.167  0.596  0.401
SO2             -0.710 -0.312 -0.619

               Comp 1 Comp 2 Comp 3 Comp 4 Comp 5
SS loadings       1.0    1.0    1.0    1.0    1.0
Proportion Var    0.2    0.2    0.2    0.2    0.2
Cumulative Var    0.2    0.4    0.6    0.8    1.0
>
```

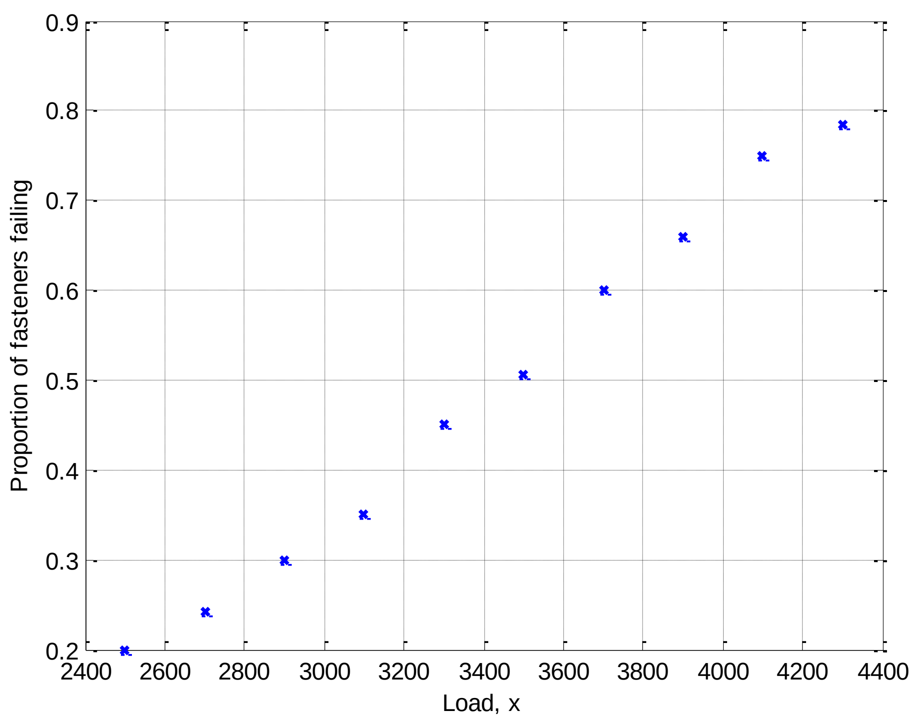
```
> library(pls);
pcr.model = pcr(MORT ~ PRECIP +  EDUC  + NONWHITE +  NOX + SO2, ncomp = 5, scale = TRUE,  data=B15.table);
summary(pcr.model);

Data:  X dimension: 60 5
 Y dimension: 60 1
Fit method: svdpc
Number of components considered: 5
TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps  5 comps
X       39.30    68.77    85.46    93.59   100.00
MORT    32.21    64.57    65.93    67.29    67.46
>
```

- The principal components regression account for 85.46% of the variation with 3 variables (components)
- While the ordinary least square accounts for only 67.5% of the variation in the model with all 5 variables

15. Exercise 14.3

Data:

```
>> disp(data);
plot(x,r./n,'x','LineWidth',2);
grid on;
xlabel('Load, x');
ylabel('Proportion of fasteners failing');
        2500          50          10
        2700          70          17
        2900         100          30
        3100          60          21
        3300          40          18
        3500          85          43
        3700          90          54
        3900          50          33
        4100          80          60
        4300          65          51
```



a. Fit a logistic model

```
>> % MATLAB: Use the glmfit function to carry out the associated regression:
[b,dev,stats] = glmfit(x,[r n],'binomial','link','logit');
>> % print estimated coeeficients
disp(stats.beta)
   -5.3397
    0.0015

>> %print deviance
disp(dev);
    0.3719

>> % print the t statistics
disp(stats.t);
   -9.7852
    9.8290

>> % print the p-values
disp(stats.p);
  1.0e-021 *

    0.1304
    0.0845
```

- Thus, the fitted model is $\hat{p} = \frac{1}{1+e^{(5.3397-0.0015x)}}$

b. Check the adequacy of the model $\hat{p} = \frac{1}{1+e^{(5.3397-0.0015x)}}$

- The deviance = 0.3719
- The model is adequate

c. The difference in deviances is 0.372 – 0.284 = 0.088

```
>> % Fit a quadratic
x2 = x.^2;
X = cat(2,x,x2);
[b2,dev2,stats2] = glmfit(X,[r n],'binomial','link','logit');
>> % print dev2
disp(dev2);
    0.2837

>> % print difference in deviance
disp(dev-dev2);
    0.0882
```

- This small difference in deviances, when comparing to Chi-square 1 d.f. indicate that there is no need for the quadratic term.

d. Find Wald statistics for each individual parameters for the quadratic model
   i. For H_0: β_1=0, the Wald statistic Z = 0.42 which is not significant
   ii. For H_0: β_2=0, the Wald statistic Z = 0.30 which is not significant

```
>>
>> %Ho: beta1=0. Get Wald statistic
Z1 = stats2.beta(2)/stats2.se(2);
disp(Z1);
    0.4179

>> %Ho: beta2=0. Get Wald statistic
Z2 = stats2.beta(3)/stats2.se(3);
disp(Z2);
    0.2970
```

e. Find approximate 95% CIs on the model parameters for the model in part C
   i. For β_1:, the CI is [-0.0033, 0.0052]
   ii. For β_2:, the CI is [-0.00000053, 0.00000071]

```
>>
>> %  get CI for beta1
ci.low = stats2.beta(2) - 1.96*stats2.se(2);
ci.high = stats2.beta(2) + 1.96*stats2.se(2);
fprintf('95 percent confidence interval for beta1: (%6.4f, %6.4f) \n',ci.low, ci.high);

95 percent confidence interval for beta1: (-0.0033, 0.0052)
>> %  get CI for beta2
beta2.ci.low = stats2.beta(3) - 1.96*stats2.se(3);
beta2.ci.high = stats2.beta(3) + 1.96*stats2.se(3);
fprintf('95 percent confidence interval for beta1: (%10.8f, %10.8f) \n',beta2.ci.low, beta2.ci.high);
95 percent confidence interval for beta1: (-0.00000053, 0.00000071)
>>
>>
>>
```

16. Exercise 14.6

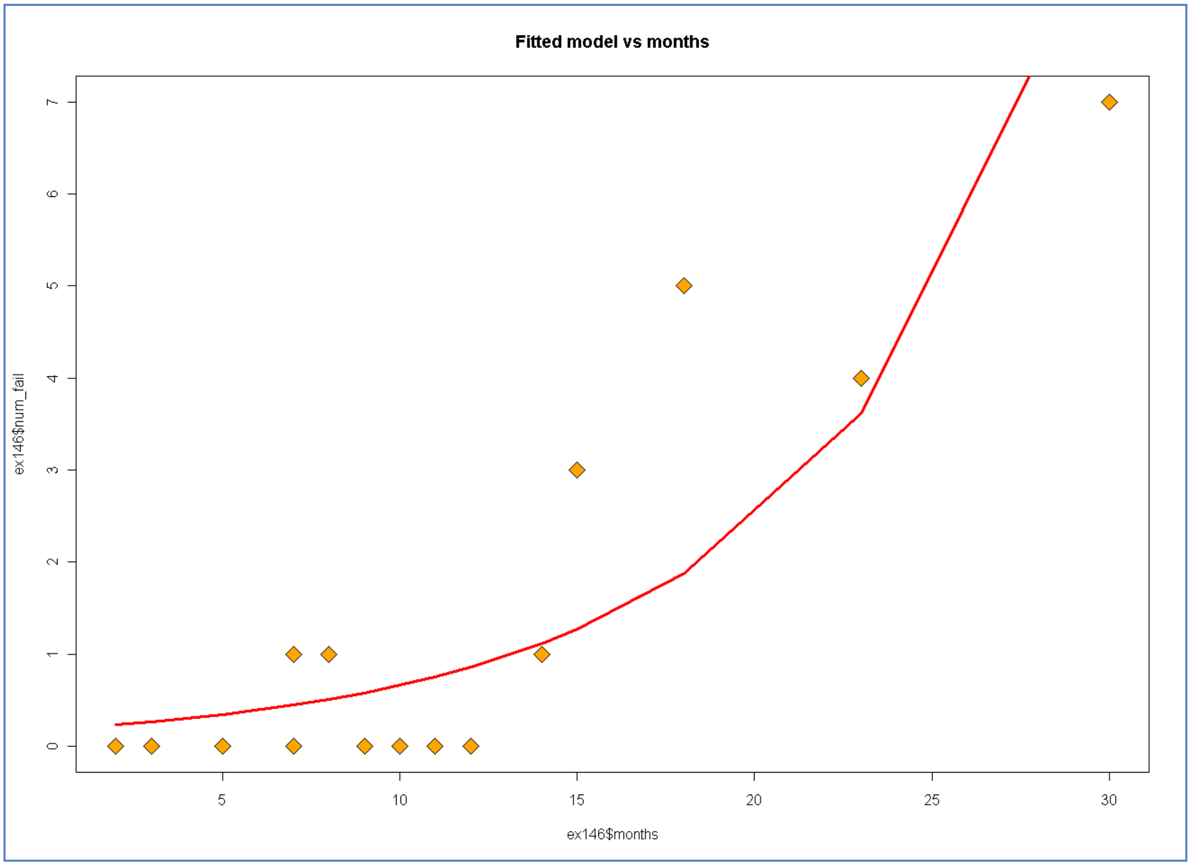
a. Fit a logistic model

```
> poisson.model = glm(num_fail ~ months, family=poisson(), data=ex146);
summary(poisson.model);


Call:
glm(formula = num_fail ~ months, family = poisson(), data = ex146)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3106  -1.0114  -0.7003   0.4031   1.8813

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.71995    0.55770  -3.084  0.00204 **
months       0.13065    0.02433   5.370 7.88e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 44.167  on 14  degrees of freedom
Residual deviance: 14.935  on 13  degrees of freedom
AIC: 38.481

Number of Fisher Scoring iterations: 5

> deviance = poisson.model$deviance;
p.value = 1-pchisq(deviance, poisson.model$df.residual);

Goodness.Of.Fit = data.frame(Method='Deviance', ChiSquare=deviance, DF=poisson.model$df.residual, P=p.value);
Goodness.Of.Fit.Test = list(Test='Goodness Of Fit', Result=Goodness.Of.Fit);
print(Goodness.Of.Fit.Test);

$Test
[1] "Goodness Of Fit"

$Result
    Method ChiSquare DF         P
1 Deviance  14.93496 13 0.3114308
```

- The fitted model is $\hat{y} = \ e^{-1.72+0.13*\text{months}}$

b. The deviance = 14.935, with p-value =0.311 indicate the model is adequate.

c. Construct graph overlay by fitted model



d. Expand the model in part a to include a quadratic term

```
> ex146$x2 = ex146$months^2;
expand.poisson.model = glm(num_fail ~ months + x2, family=poisson(), data=ex146);
summary(expand.poisson.model);


Call:
glm(formula = num_fail ~ months + x2, family = poisson(), data = ex146)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3308  -0.8141  -0.3901   0.4821   1.2854

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.436107   1.705741  -2.601   0.0093 **
months       0.458657   0.179552   2.554   0.0106 *
x2          -0.008259   0.004350  -1.899   0.0576 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 44.167  on 14  degrees of freedom
Residual deviance: 10.769  on 12  degrees of freedom
AIC: 36.315

Number of Fisher Scoring iterations: 5

> # partial deviance test of full vs. reduced
anova(poisson.model, expand.poisson.model);

Analysis of Deviance Table

Model 1: num_fail ~ months
Model 2: num_fail ~ months + x2
  Resid. Df Resid. Dev Df Deviance
1        13     14.935
2        12     10.769  1   4.1655
> deviance = anova(poisson.model, expand.poisson.model)$Deviance[2];
print(deviance);

[1] 4.165524
> df = anova(poisson.model, expand.poisson.model)$Df[2];
p.value = 1-pchisq(deviance, df);
print(p.value);

[1] 0.04125466
> |
```

- The expanded model is $\hat{y} = \ e^{-4.44+0.46*\text{months}+0.46*\text{month}^2}$
- The deviance difference is 4.165 and p-value = 0.04. Thus the quadratic term is significant at the level of alpha = 0.05.

e. Find the Wald statistics for each individual parameter in the model developed in part A

- For beta1, the Wald statistic Z = 5.37

```
> # Compute Wald statistic for beta1
beta1 = coef(poisson.model)['months'];
SE = sqrt(vcov(poisson.model)['months', 'months'])
Z = beta1/SE;

#print Wald statistic for beta 1
print(Z);


  months
5.369807
```

f. Find approximate 95% confidence interval on beta1

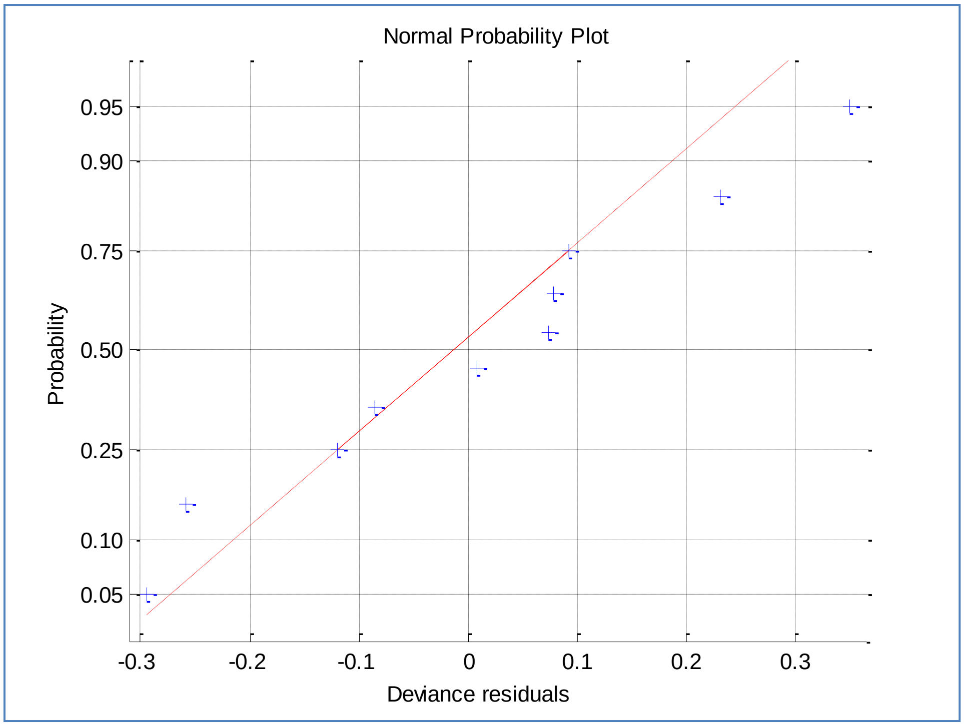
- The 95% confidence interval for β_1 is (0.083, 0.178)

```
> # Compute CI for beta1
beta1.hat = coef(poisson.model)['months'];
SE = sqrt(vcov(poisson.model)['months', 'months']);
ci.upr = beta1.hat + SE * qnorm(0.975);
ci.lwr = beta1.hat + SE * qnorm(0.025);
data.frame(center=beta1.hat, lower=ci.lwr, upper=ci.upr);
          center      lower     upper
months 0.1306460 0.08296058 0.1783314
>
```

17. Exercise 14.12

Reconsider the model for aircraft fastener from problem 14.3

- Construct plots of the deviance residuals from model



- Plot of the deviance residuals from the logistic model developed from problem 14.3
- The normal probability plot of deviance residuals indicate no severe problem with normality
- If time permit, should also plot the deviance residuals vs. $2sin^{-1}\sqrt{\pi\hat{\pi}_i}$

18. Exercise 14.21

Exercise 14.21

Let $\eta = x'\beta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2$

We have

$$\hat{\eta}(x_1+1) - \hat{\eta}(x_1)$$

$$= \hat{\beta}_0 + \hat{\beta}_1(x+1) + \hat{\beta}_{12}(x_1+1)x_2 - \left(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_{12} x_1 x_2\right)$$

$$= \hat{\beta}_1 + \hat{\beta}_{12} x_2$$

Hence, $\boxed{\widehat{O_R} = e^{\hat{\beta}_1 + \hat{\beta}_{12} x_2}}$

- Therefore this odds ratio does not have the same interpretation as in the case where the linear predictor does not have interaction term.
- In this case, the odds ratio $\widehat{O_R} = e^{\hat{\beta}_1 + \hat{\beta}_{12} x_2}$ include the estimated interaction coefficient and x_2 has to be fixed when interpret.