CS-E-106: Data Modeling

Final Exam

Instructor: Hakan Gogtas
Submitted by: *Saurabh Kulkarni*

Due Date: *12/17/2019*

Import Libraries

Question 1 Use the PR1_Dataset data which contains 5 continuous variables (no categorical variables), the answer the questions below: (25 pts)

(a) Fit a regression model to predict Y by using all variables. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? (5pts)

```r
pr1_data = read.csv("PR1_Dataset.csv")
summary(pr1_data)
```

```
##        Y               X1              X2              X3
##  Min.   :218.0   Min.   :47.00   Min.   :64.00   Min.   :23.00
##  1st Qu.:256.0   1st Qu.:75.75   1st Qu.:76.75   1st Qu.:34.00
##  Median :261.0   Median :78.00   Median :82.00   Median :36.50
##  Mean   :260.9   Mean   :77.70   Mean   :80.40   Mean   :36.85
##  3rd Qu.:270.2   3rd Qu.:81.00   3rd Qu.:85.25   3rd Qu.:41.00
##  Max.   :281.0   Max.   :85.00   Max.   :90.00   Max.   :48.00
##        X4              X5
##  Min.   : 6.0   Min.   :14.00
##  1st Qu.: 9.0   1st Qu.:21.75
##  Median :13.5   Median :23.00
##  Mean   :14.0   Mean   :23.93
##  3rd Qu.:18.0   3rd Qu.:27.00
##  Max.   :33.0   Max.   :37.00
```

```r
lm_pr1 = lm(Y~X1+X2+X3+X4+X5, data=pr1_data)
summary(lm_pr1)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = pr1_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.0131  -2.9395   0.4694   2.5336   9.3248
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 155.0304    36.2383   4.278 0.000145 ***
## X1            0.3911     0.2571   1.521 0.137399
## X2            0.8639     0.1797   4.807 3.05e-05 ***
## X3            0.3616     0.2690   1.345 0.187679
## X4           -0.8467     0.3525  -2.402 0.021927 *
## X5            0.1923     0.2636   0.729 0.470718
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.268 on 34 degrees of freedom
## Multiple R-squared:  0.861,  Adjusted R-squared:  0.8406
## F-statistic: 42.13 on 5 and 34 DF,  p-value: 1.276e-13
```

Interpretation

R^2 is 86%. X2 and X4 are significant and X3 and X3 are not significant.

Multi-collinearity

```
vif(lm_pr1)
```

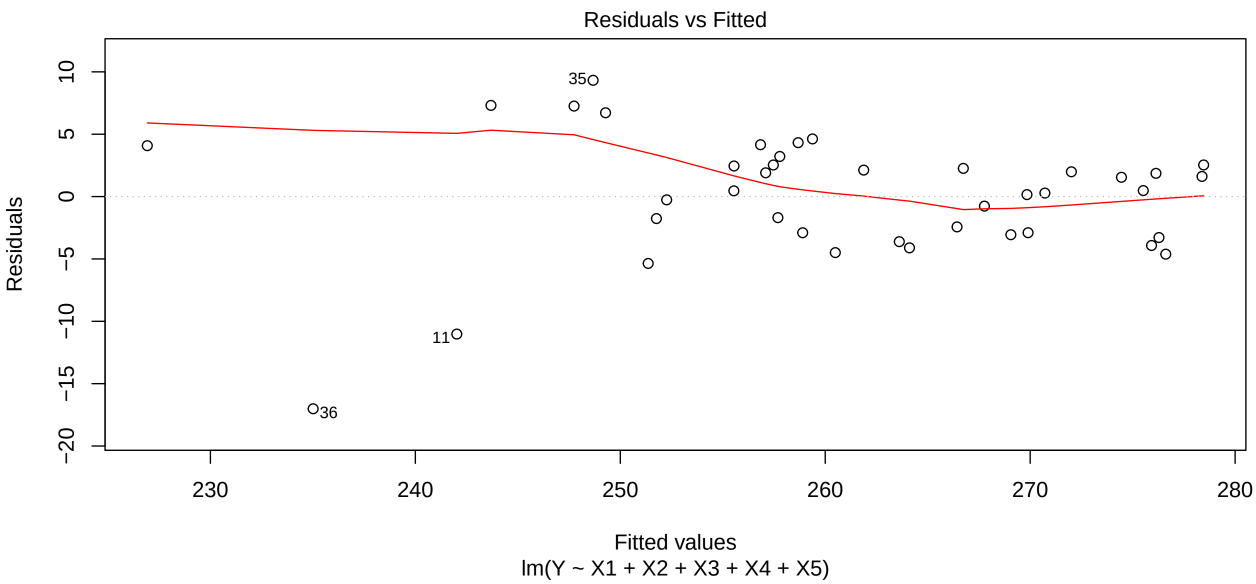
```
##       X1       X2       X3       X4       X5
## 3.916370 1.803353 2.812730 6.278713 1.624470
```

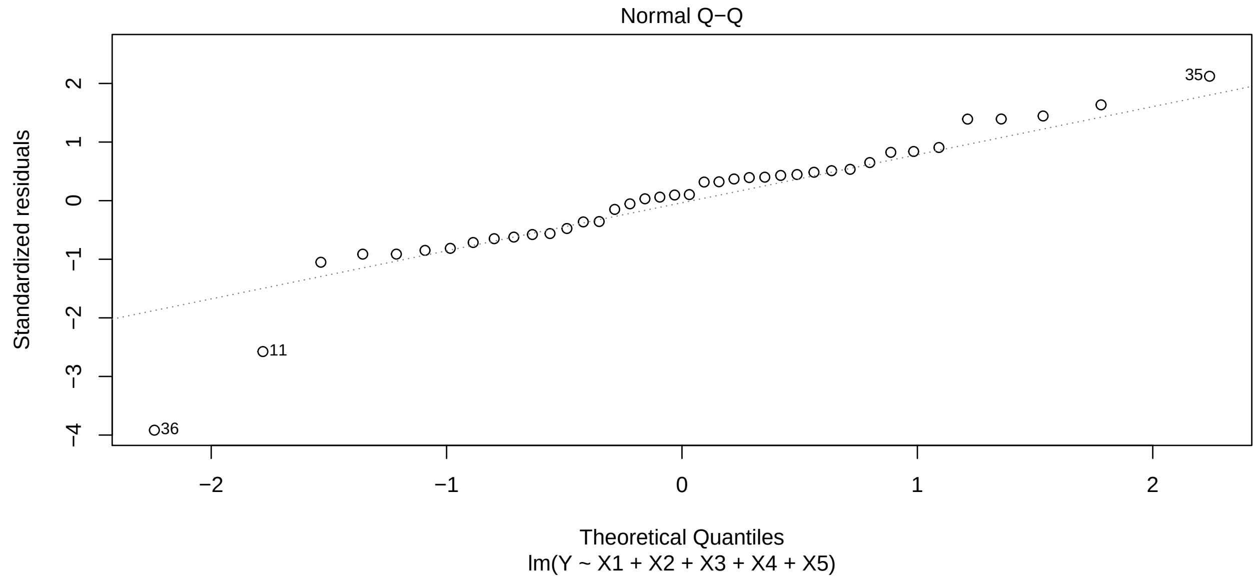
Interpretation

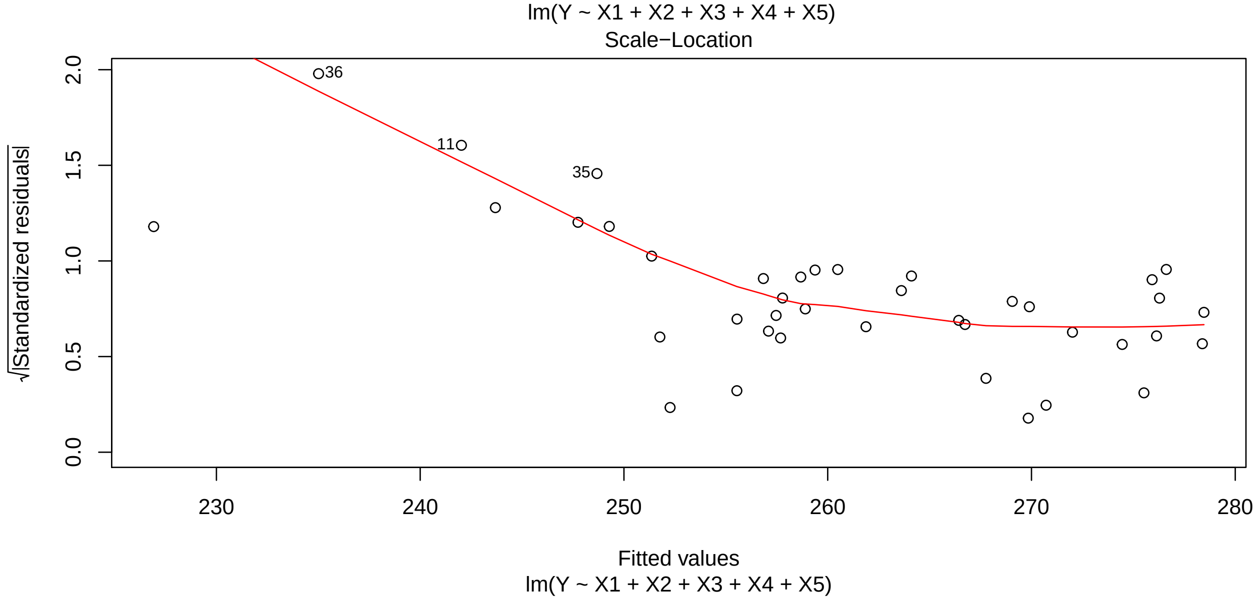
Since all the VIFs are <10, we can say that there is not any seriour multi-collinearity in the given data.

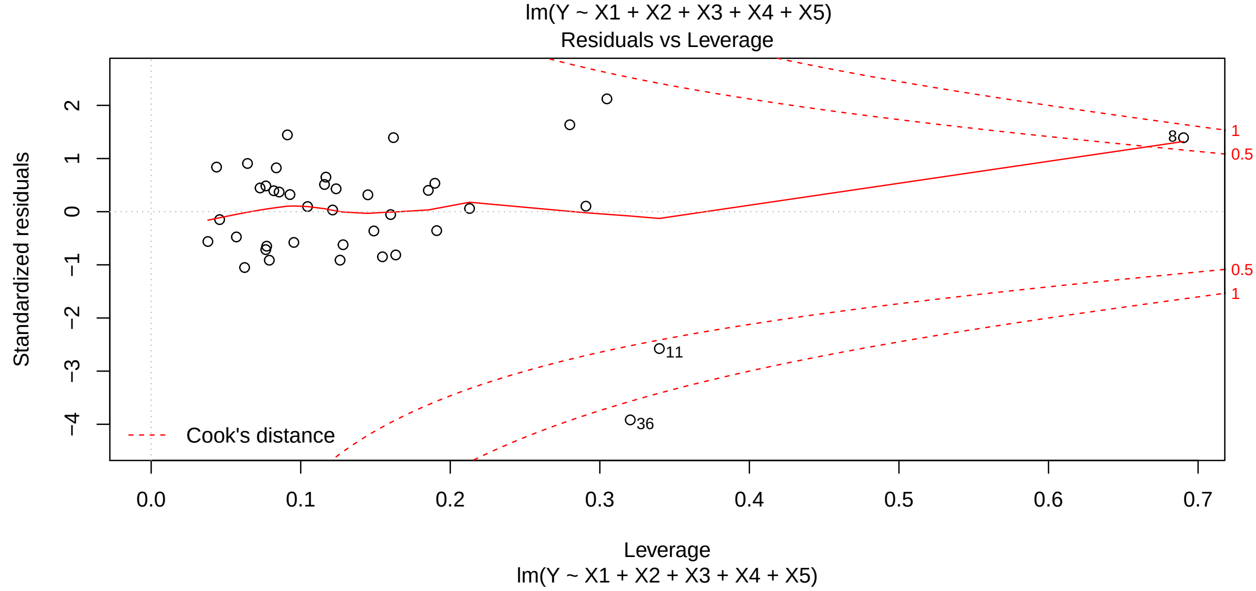
Normal Error & Constant Variance

```
plot(lm_pr1)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Y ~ X1 + X2 + X3 + X4 + X5)
35
11
36


Scale−Location
√|Standardized residuals|
Fitted values
lm(Y ~ X1 + X2 + X3 + X4 + X5)
36
11
35


Residuals vs Leverage
Standardized residuals
Leverage
lm(Y ~ X1 + X2 + X3 + X4 + X5)
Cook's distance
8
11
36
1
0.5

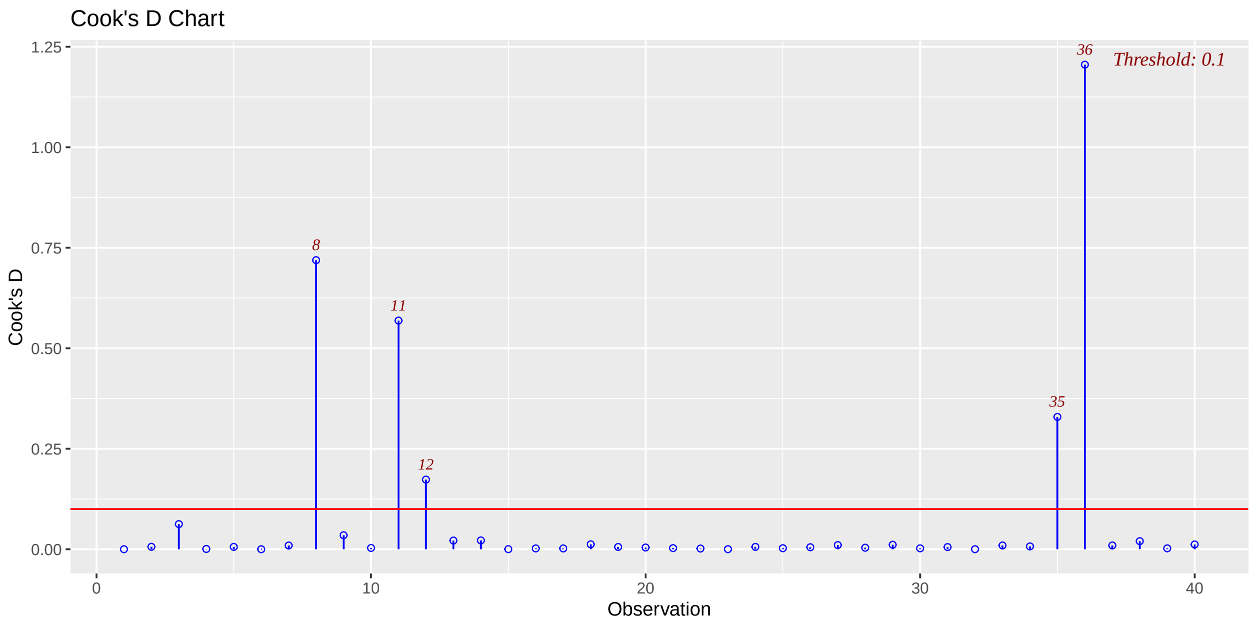
Interpretation

Normal Probability Plot: We can see that the this plot is mostly linear, so the error terms are in agreement with the normal distribution.

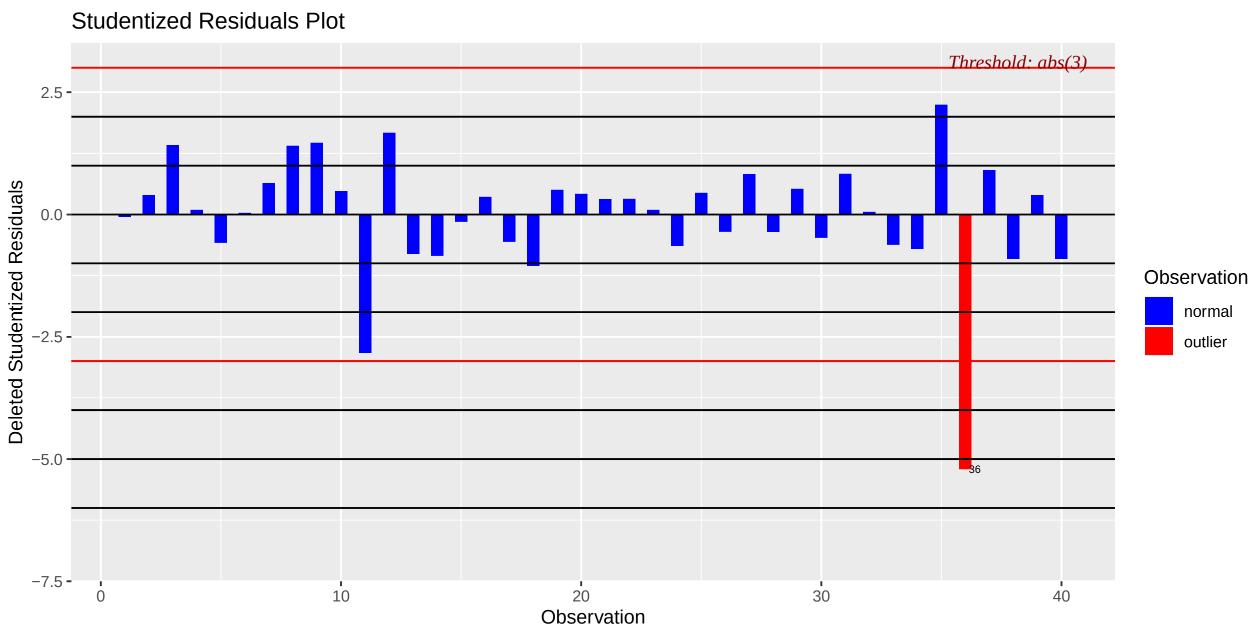
It also show that the error terms have a constant variance. However, we do see some outliers.

Outliers/Influential Points

```
ols_plot_cooksd_chart(lm_pr1)
```




```
ols_plot_resid_stud(lm_pr1)
```




```
#outliers in Xs
model = lm_pr1
df = pr1_data
n = nrow(df)
```

```r
p = length(model$coefficients)
hii = hatvalues(model)
index = hii>2*p/n
print("Hat values outliers")
```

```
## [1] "Hat values outliers"
```

```r
index[index]
```

```
##    8   11   35   36
## TRUE TRUE TRUE TRUE
```

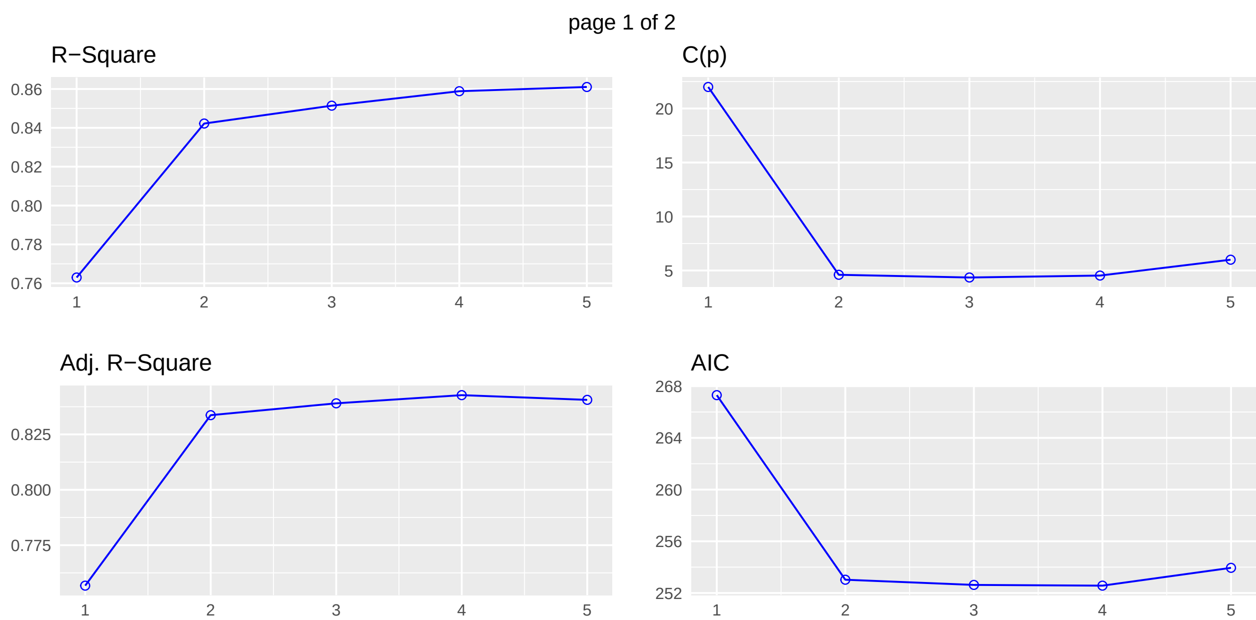
Interpretation

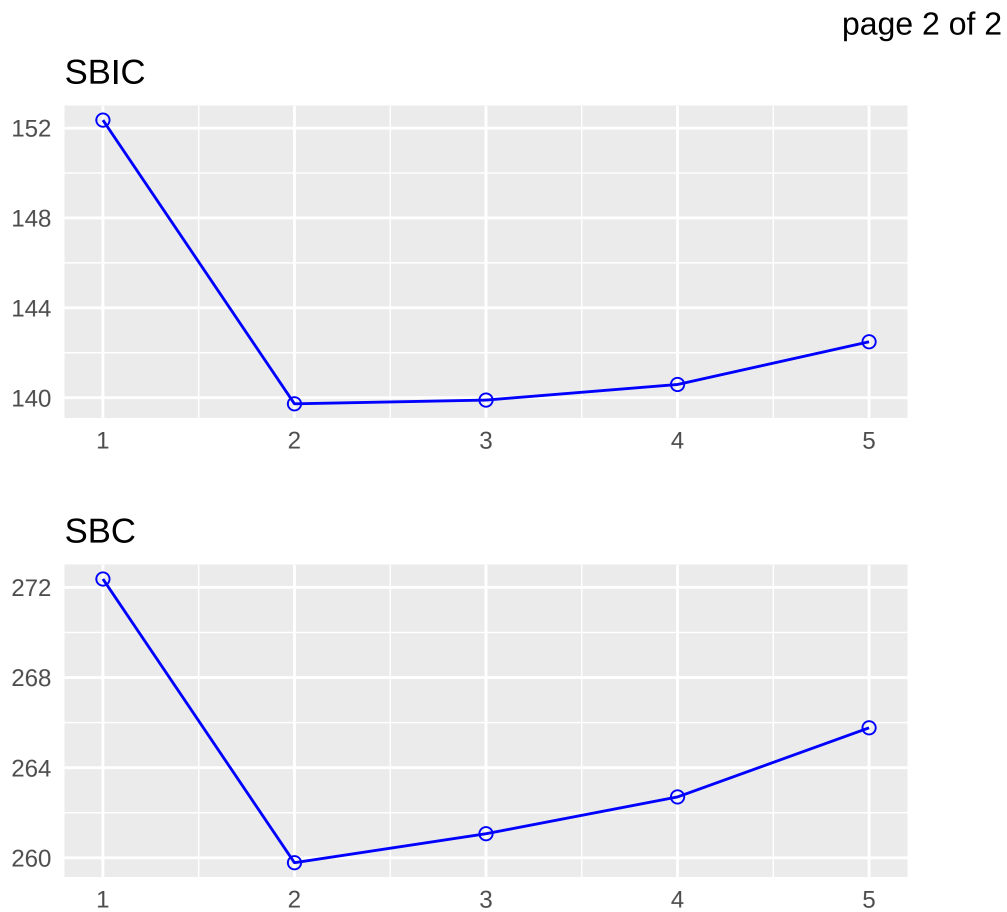
Both Cook's Distance and Hat values show that cases 8, 11, 35 and 36 are outliers. Case 36 is shown as outlier in the studentied residual plot as well.

Thus, 8,11 and 36 are clear outliers and 12 and 35 need further investigation.

(b) Use the stepwise variable selection procedure to find the best model. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? (5pts)

```r
k_pr1 = ols_step_best_subset(lm_pr1, prem=0.05, details=TRUE)
plot(k_pr1)
```

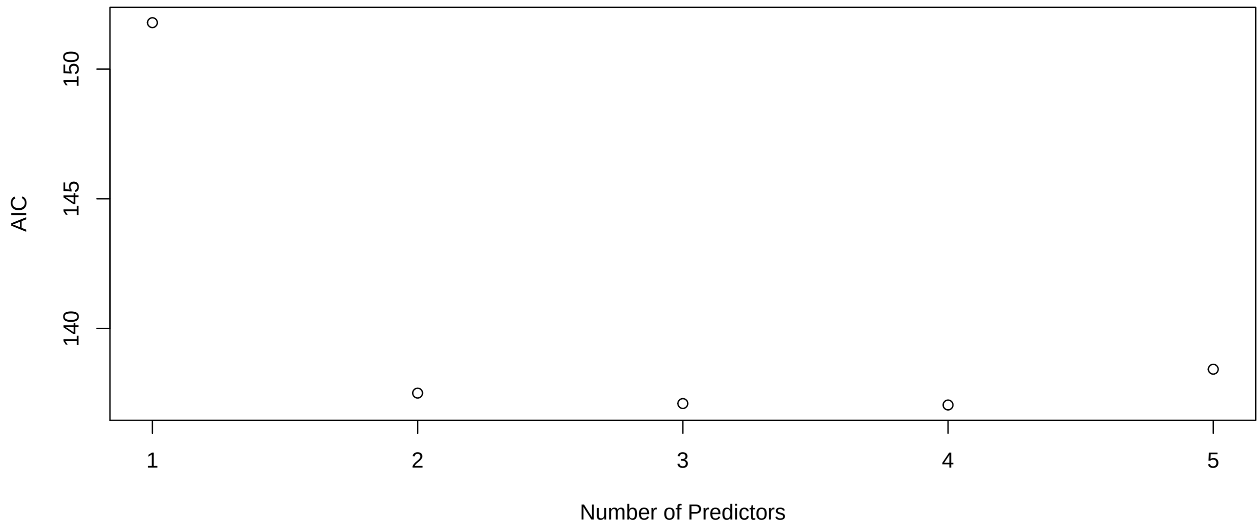




NOTE FOR GRADERS:

The above function was running into errors and after repeated troubleshooting was not resolving. Hence, I went with the R function `regsubsets()` from leaps library. See below.

```
library(olsrr)
k_pr1 = regsubsets(Y~X1+X2+X3+X4+X5, data=pr1_data)
rs = summary(k_pr1)
AIC <- nrow(pr1_data)*log(rs$rss/nrow(pr1_data)) + (2:6)*2
par(mfrow=c(1,1))
plot(AIC ~ I(1:5), ylab="AIC", xlab="Number of Predictors")
```




```r
rs$which
```

```
##   (Intercept)    X1    X2    X3   X4    X5
## 1        TRUE FALSE FALSE FALSE TRUE FALSE
## 2        TRUE FALSE  TRUE FALSE TRUE FALSE
## 3        TRUE FALSE  TRUE  TRUE TRUE FALSE
## 4        TRUE  TRUE  TRUE  TRUE TRUE FALSE
## 5        TRUE  TRUE  TRUE  TRUE TRUE  TRUE
```

```r
rs$adjr2
```

```
## [1] 0.7567300 0.8336868 0.8390299 0.8427276 0.8405966
```

Interpretation

We can see that model #3, containing X2 and X4 gives us the best asdjusted R^2 as we see the elbow at that model (based on the printed adj. R2 values above).

```r
pr1_best_lm = lm(Y~X2+X4, data=pr1_data)
summary(pr1_best_lm)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = pr1_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6799  -3.1931   0.4761   2.9719  11.5850
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 222.5896    15.2981  14.550  < 2e-16 ***
## X2            0.7323     0.1699   4.311 0.000116 ***
## X4           -1.4652     0.1786  -8.205 7.52e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.381 on 37 degrees of freedom
## Multiple R-squared:  0.8422, Adjusted R-squared:  0.8337
## F-statistic: 98.75 on 2 and 37 DF,  p-value: 1.459e-15
```

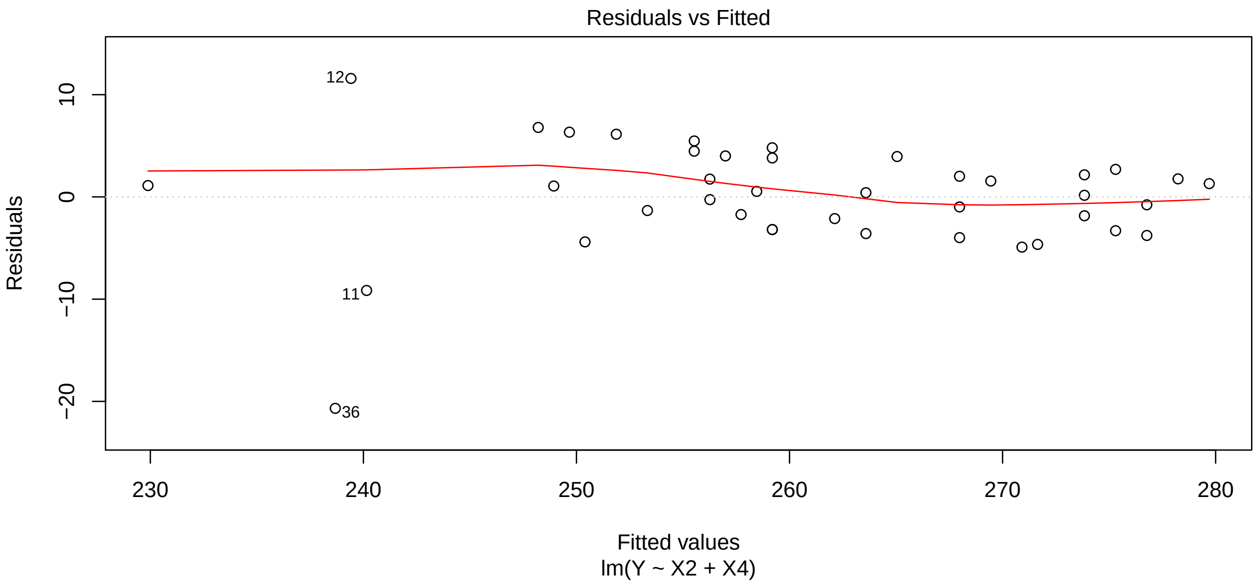
Multi-collinearity: No multi-collinearity exists.

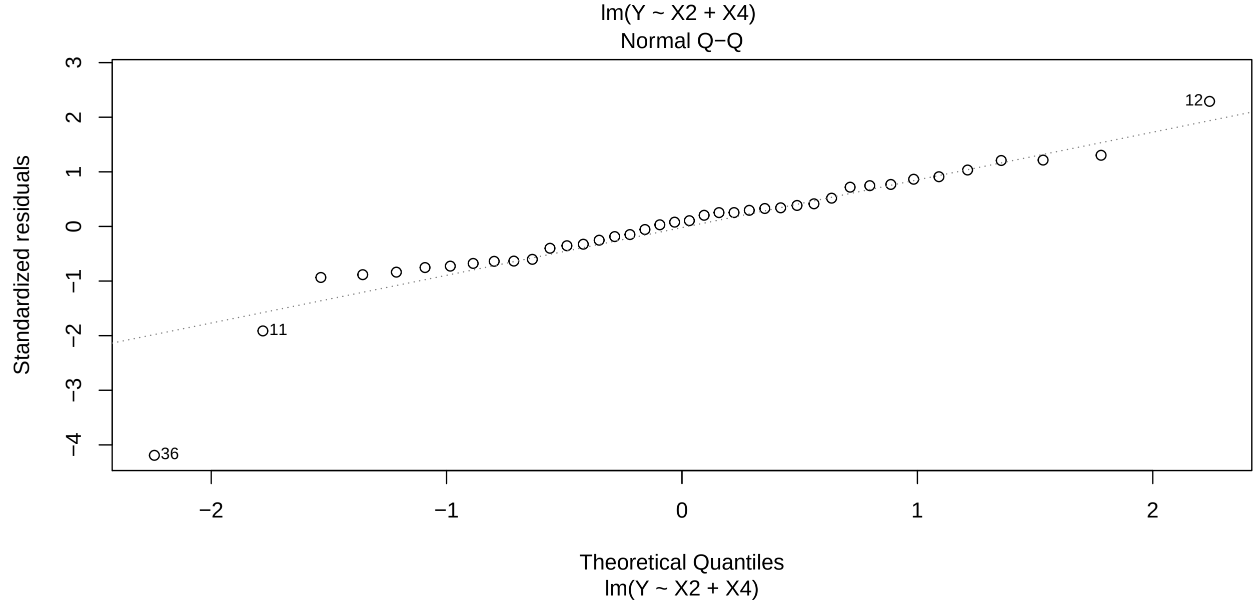
```r
vif(pr1_best_lm)
```

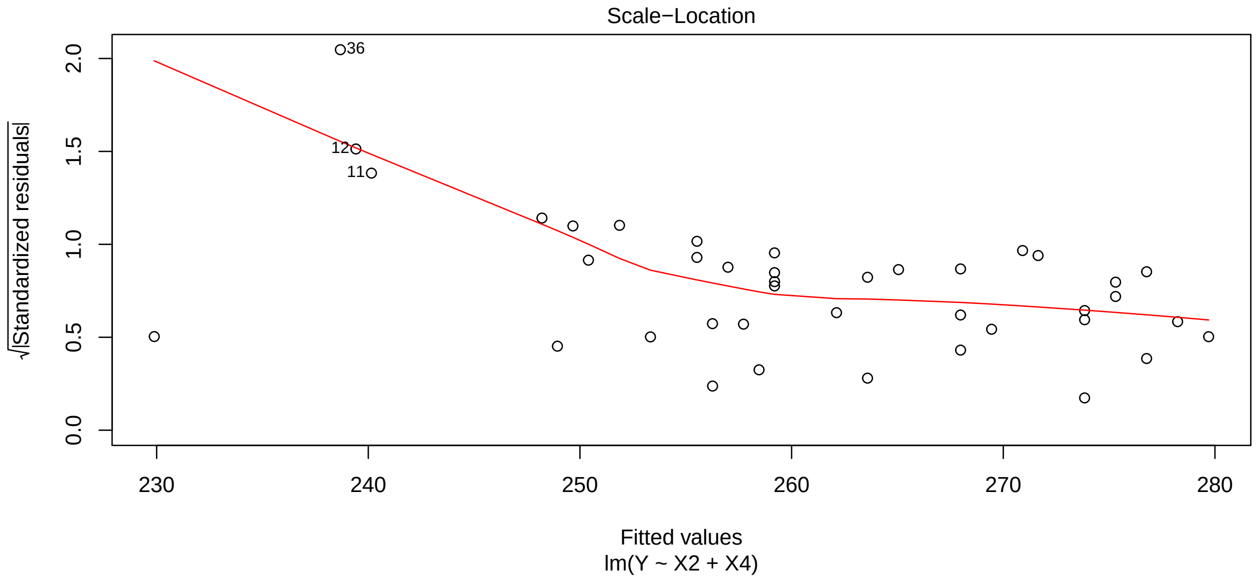
```
##       X2       X4
## 1.544187 1.544187
```

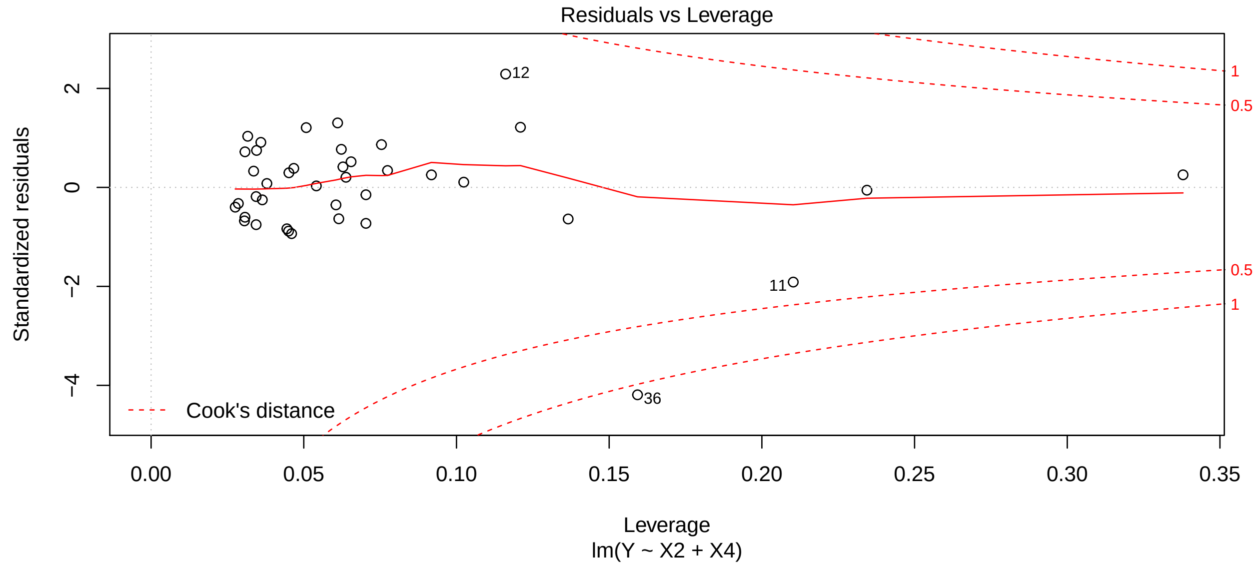
Normal Error & Constant Variance

```r
plot(pr1_best_lm)
```









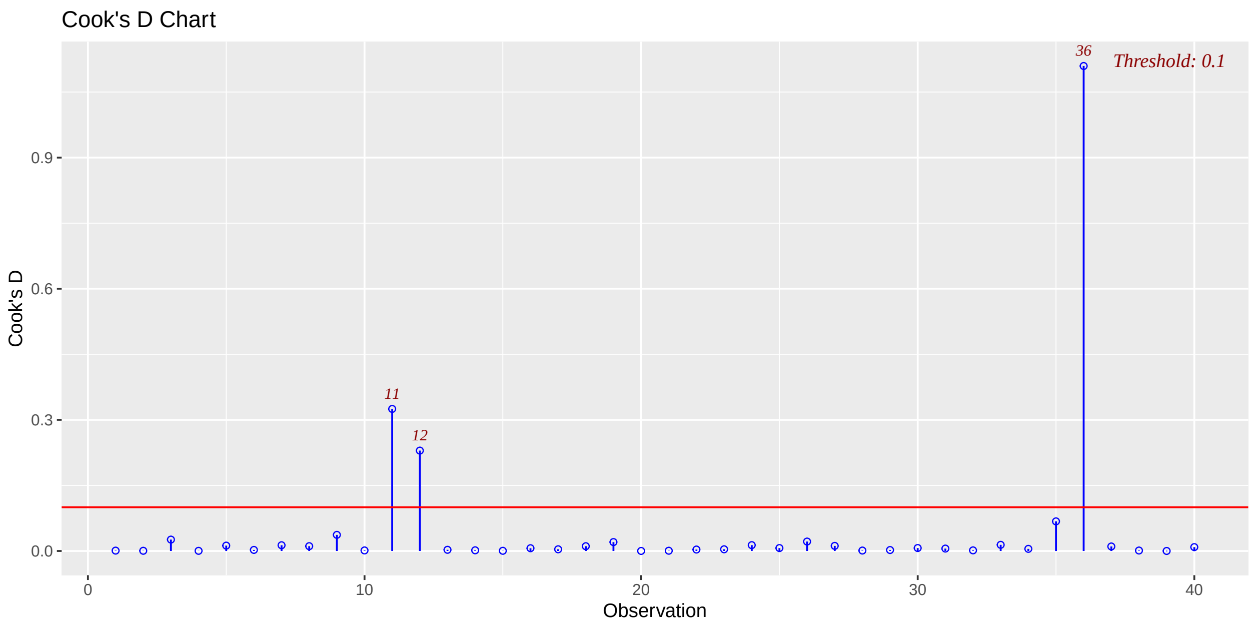
Interpretation

Normal Probability Plot: We can see that the this plot is mostly linear, so the error terms are in agreement with the normal distribution.

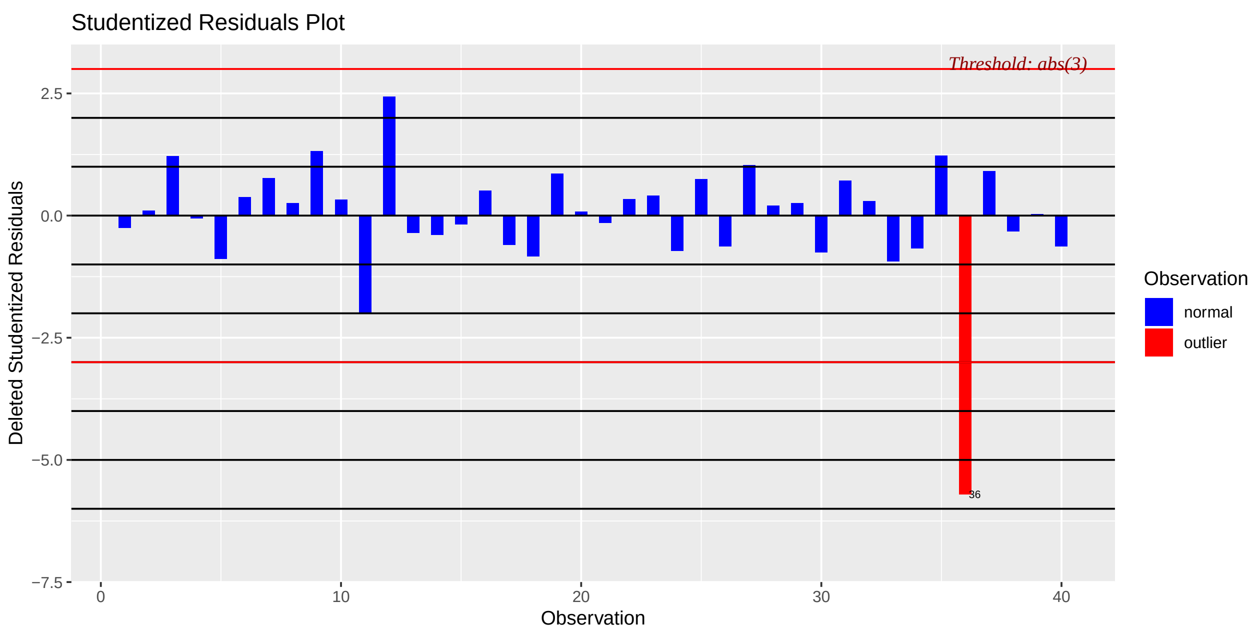
It also show that the error terms have a constant variance. However, we do see some outliers.

Outliers/Influential Points

```
ols_plot_cooksd_chart(pr1_best_lm)
```




```
ols_plot_resid_stud(pr1_best_lm)
```




```
#outliers in Xs
model = pr1_best_lm
df = pr1_data
n = nrow(df)
p = length(model$coefficients)
hii = hatvalues(model)
index = hii>2*p/n
print("Hat values outliers")
```

```
## [1] "Hat values outliers"
```

```
index[index]
```

```
##    4    8   11   36
```

```
## TRUE TRUE TRUE TRUE
```

Interpretation

Both Cook's Distance and Hat values show that only cases 11, 12 and 36 are outliers. Case 36 is shown as outlier in the studentied residual plot as well. 8 is no longer an outlier according to Cook's Distance

(c) Use the model built in part b, exclude the observation with the largest cook distance and refit the model and comment the model results (5pts)

```r
lm_pr1c = lm(Y~X2+X4, data=pr1_data[-36,])
summary(lm_pr1c)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = pr1_data[-36, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9810 -2.8786  0.3054  2.5058  9.4437
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 211.8978    11.3946  18.596  < 2e-16 ***
## X2            0.8233     0.1258   6.544 1.31e-07 ***
## X4           -1.1804     0.1404  -8.408 5.15e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.953 on 36 degrees of freedom
## Multiple R-squared:  0.8851, Adjusted R-squared:  0.8787
## F-statistic: 138.7 on 2 and 36 DF,  p-value: < 2.2e-16
```

Interpretation

R^2 is increased to 88% from 84% in part(b). We also see a decrease in the standard errors for both the coefficients suggesting a tighter fit, more confident fit. Thus, case #36 is truly an influential point.

(d) Use the model built in part b, fit the robust regression and compared it against the model in part c, comments on the model results. (5pts)

```r
lm_pr1d = rlm(Y~X2+X4, data=pr1_data)
summary(lm_pr1d)
```

```
##
## Call: rlm(formula = Y ~ X2 + X4, data = pr1_data)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -23.25840  -2.71474   0.09727   2.82577  10.01078
##
## Coefficients:
##             Value    Std. Error t value
## (Intercept) 217.2117  11.8652    18.3067
## X2            0.7732   0.1317     5.8689
## X4           -1.2858   0.1385    -9.2833
##
## Residual standard error: 4.073 on 37 degrees of freedom
```

Interpretation:

We see that the coefficients and residual standard error are not very different from those obtained in part (c) i.e. without largest cook's distance observation. However, we see greater difference compared to the coefficients and RSE values obtained in part (b). Which means that the robust regression probably gives much lower weight to case #36 during the model fitting process.

(e) Use the model built in part b, predict Y for X1=75, X2=78, X3=34, X4=18, X5=18 and calculate 95% confidence interval (5pts).

```
Xh = data.frame(cbind(X1=75,X2=78,X3=34, X4=18, X5=18))
pred = predict(pr1_best_lm, Xh, se.fit=TRUE, interval="confidence", level=1-0.05)
pred
```

```
## $fit
##        fit      lwr      upr
## 1 253.3316 251.2508 255.4124
##
## $se.fit
## [1] 1.026932
##
## $df
## [1] 37
##
## $residual.scale
## [1] 5.380997
```

Question 2 Use the PR2_Dataset data: X4, X5, X6, and X7 are the categorical variables, Y and remaining independent variables are continuous variables. X4 has two levels, X5 has 4, X6 has 5, and X7 has 3 levels (create dummy variables for the categorical variables). Answer the questions below: (30 pts)

```
pr2_data = read.csv("PR2_Dataset.csv")
pr2_data$X4 = as.factor(pr2_data$X4)
pr2_data$X5 = as.factor(pr2_data$X5)
pr2_data$X6 = as.factor(pr2_data$X6)
pr2_data$X7 = as.factor(pr2_data$X7)
summary(pr2_data)
```

```
##        Y                X1                X2              X3         X4
##  Min.   :   201   Min.   :  7716   Min.   :2.000   Min.   :19.00   1:27
##  1st Qu.:  1769   1st Qu.: 25717   1st Qu.:2.000   1st Qu.:24.00   2:94
##  Median :  8666   Median :113571   Median :2.000   Median :24.00
##  Mean   : 19438   Mean   :263428   Mean   :3.099   Mean   :31.12
##  3rd Qu.: 21535   3rd Qu.:459784   3rd Qu.:4.000   3rd Qu.:38.00
##  Max.   :155547   Max.   :941411   Max.   :8.000   Max.   :68.00
##  X5     X6     X7
##  1: 8   1:32   1:64
##  2:56   2:20   2:39
##  3:18   3: 1   3:18
##  4:39   4: 7
##         5:61
##
```

```
pr2_data$X4_1 = ifelse(pr2_data$X4==1, 1, 0)

pr2_data$X5_1 = ifelse(pr2_data$X5==1, 1, 0)
pr2_data$X5_2 = ifelse(pr2_data$X5==2, 1, 0)
pr2_data$X5_3 = ifelse(pr2_data$X5==3, 1, 0)
```

```r
pr2_data$X6_1 = ifelse(pr2_data$X6==1, 1, 0)
pr2_data$X6_2 = ifelse(pr2_data$X6==2, 1, 0)
pr2_data$X6_3 = ifelse(pr2_data$X6==3, 1, 0)
pr2_data$X6_4 = ifelse(pr2_data$X6==4, 1, 0)

pr2_data$X7_1 = ifelse(pr2_data$X7==1, 1, 0)
pr2_data$X7_2 = ifelse(pr2_data$X7==2, 1, 0)
```

(a) Fit a regression model to predict Y by using all variables. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? (10 pts)

```r
lm_pr2 = lm(Y~X1+X2+X3+X4_1+X5_1+X5_2+X5_3+X6_1+X6_2+X6_3+X6_4+X7_1+X7_2, data=pr2_data)
summary(lm_pr2)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4_1 + X5_1 + X5_2 + X5_3 + X6_1 +
##     X6_2 + X6_3 + X6_4 + X7_1 + X7_2, data = pr2_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -36890  -3898   1679   6180  58644
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.837e+04  8.340e+03  -3.402 0.000943 ***
## X1           2.771e-02  4.961e-03   5.585 1.79e-07 ***
## X2           9.661e+03  1.568e+03   6.159 1.30e-08 ***
## X3           1.282e+02  1.294e+02   0.991 0.324132
## X4_1         2.771e+04  1.457e+04   1.902 0.059893 .
## X5_1        -3.536e+04  1.830e+04  -1.933 0.055920 .
## X5_2        -6.664e+03  1.018e+04  -0.654 0.514195
## X5_3         1.111e+04  1.546e+04   0.719 0.473895
## X6_1        -2.215e+03  6.656e+03  -0.333 0.739917
## X6_2        -2.660e+03  3.985e+03  -0.667 0.505911
## X6_3        -1.800e+03  1.418e+04  -0.127 0.899233
## X6_4         5.194e+03  5.555e+03   0.935 0.351892
## X7_1         1.093e+04  1.113e+04   0.981 0.328566
## X7_2        -2.720e+03  4.527e+03  -0.601 0.549271
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13470 on 107 degrees of freedom
## Multiple R-squared:  0.8231, Adjusted R-squared:  0.8016
## F-statistic: 38.29 on 13 and 107 DF,  p-value: < 2.2e-16
```

Interpretation

We see that R^2 is 82%. X1, X2, X4, X5_1 seem to be significant.

Multi-collinearity

```r
vif(lm_pr2)
```

```
##        X1        X2        X3      X4_1      X5_1      X5_2      X5_3
##  1.767314  2.750584  1.473327 24.549139 13.790226 17.189914 20.199396
```

```
##      X6_1      X6_2      X6_3      X6_4      X7_1      X7_2
##  5.748744  1.461505  1.099084  1.121915 20.600710  2.986026
```

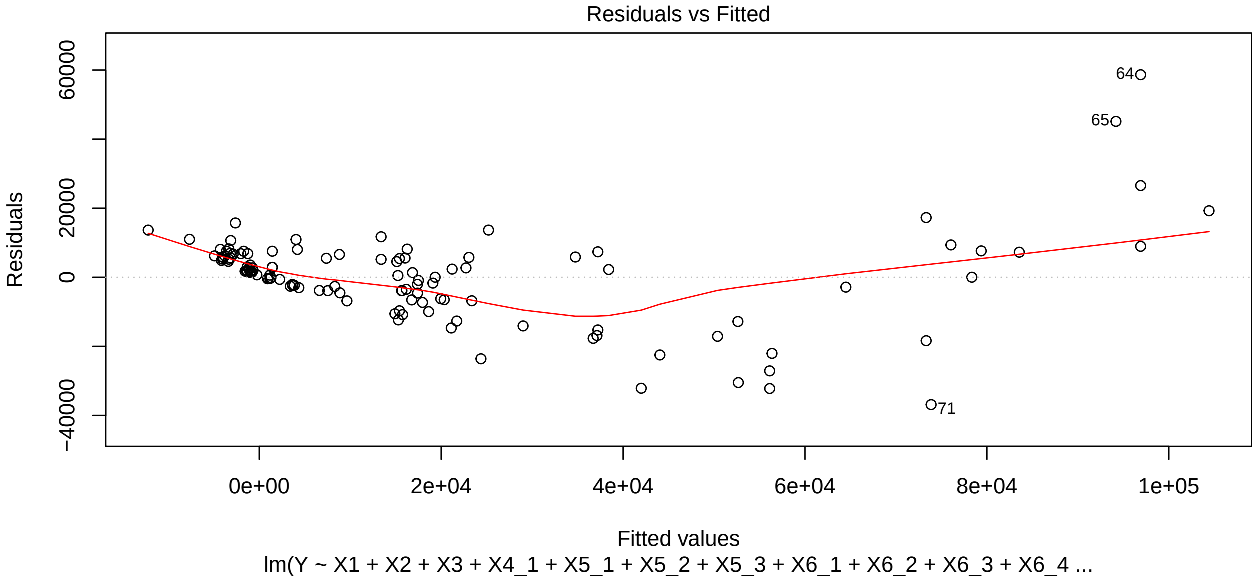
Interpretation

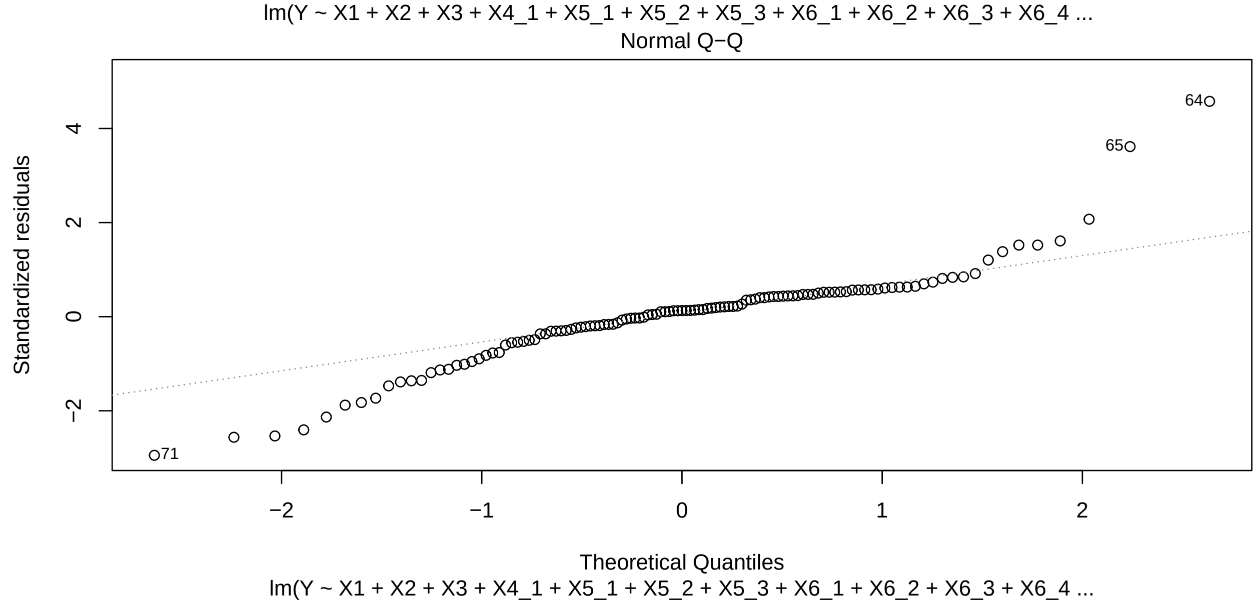
We do see some multi-collinearity, but nothing drastic.

Normal Error & Constant Variance

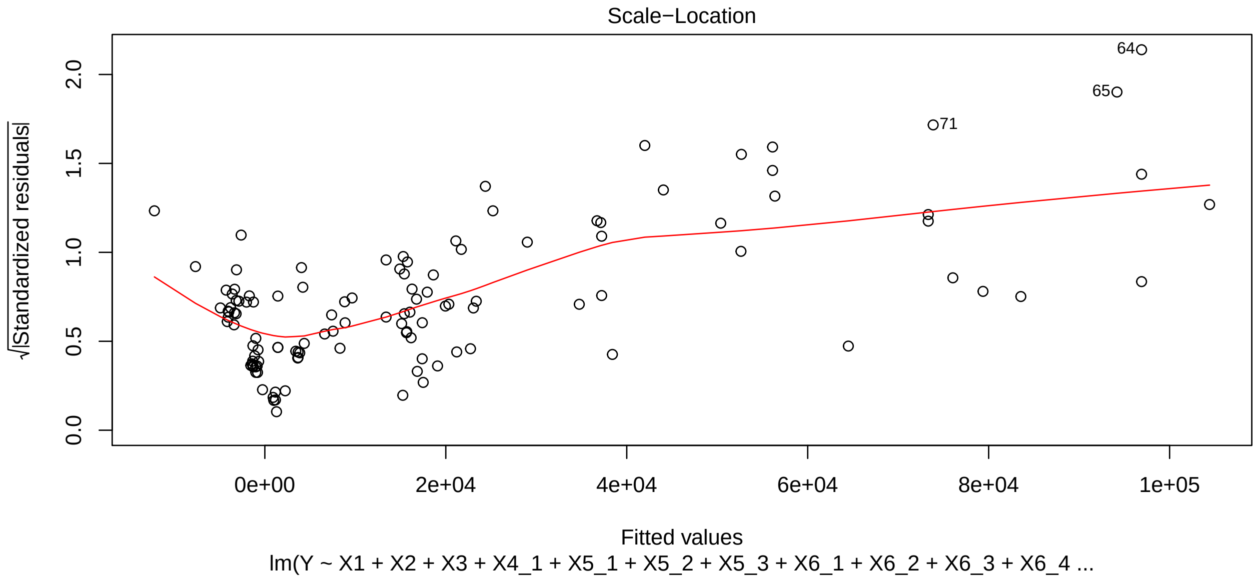
```r
plot(lm_pr2)
```

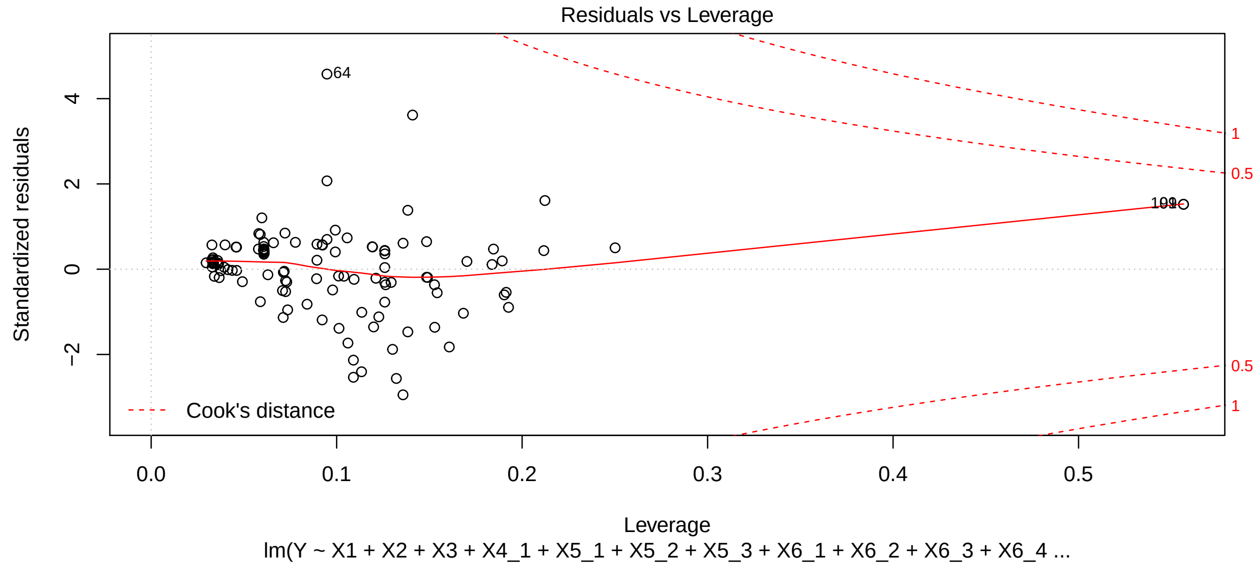
```
## Warning: not plotting observations with leverage one:
##   63, 79
```






```
## Warning: not plotting observations with leverage one:
##   63, 79
```



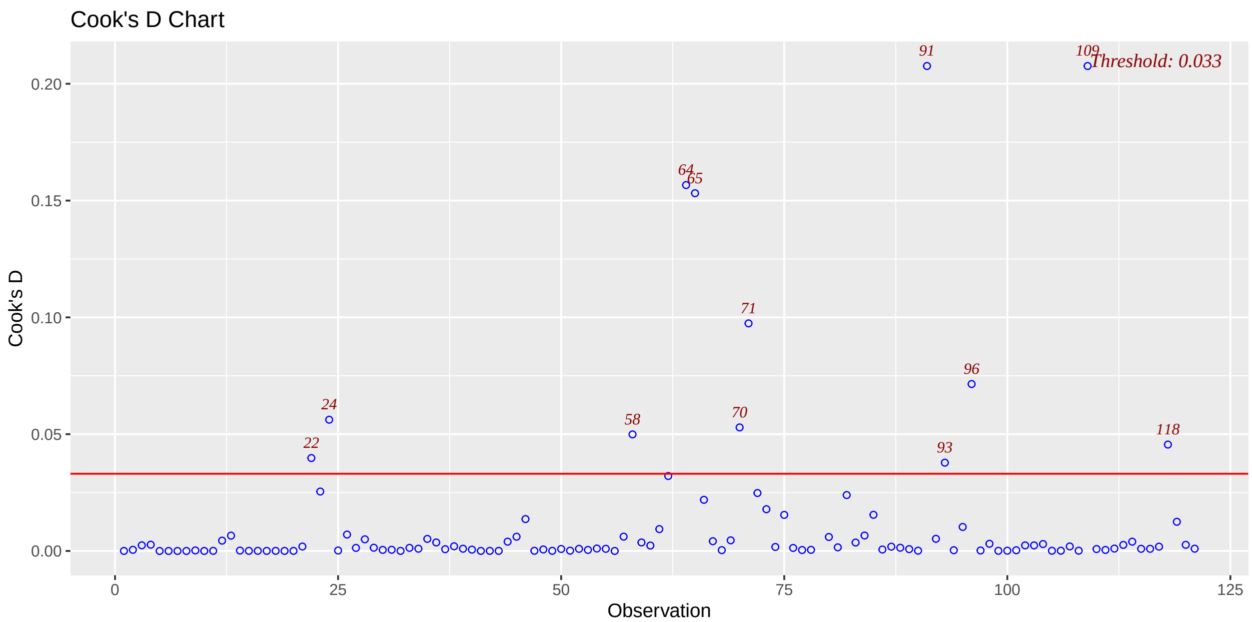


Interpretation

We see that the normal probability plot is not linear and the error variance is not constant.

Outliers

```
ols_plot_cooksd_chart(lm_pr2)
```




```
#outliers in Xs
model = lm_pr2
df = pr2_data
n = nrow(df)
p = length(model$coefficients)
hii = hatvalues(model)
index = hii>2*p/n
print("Hat values outliers")
```

```
## [1] "Hat values outliers"
```

```
index[index]
```

```
##   63   79   80   91  109
## TRUE TRUE TRUE TRUE TRUE
```

Interpretation

#109, 64, 65 and 91 are clear outliers according to the y-values. Outliers according to hat values printed above. 109 and 91 are common in both.

(b) Conduct the Breusch-Pagan for testing unequal variances and document your results (5pts).

Null Hypothesis: H_0: Error variance is constant

Alternate Hypothesis: H_1: Error variance is not constant

```
ei = lm_pr2$residuals
ei2 = ei^2
df = as.data.frame(cbind(pr2_data, ei, ei2))
f = lm(ei2~X1+X2+X3+X4_1+X5_1+X5_2+X5_3+X6_1+X6_2+X6_3+X6_4+X7_1+X7_2, data=df)
summary(f)
```

```
##
## Call:
## lm(formula = ei2 ~ X1 + X2 + X3 + X4_1 + X5_1 + X5_2 + X5_3 +
##     X6_1 + X6_2 + X6_3 + X6_4 + X7_1 + X7_2, data = df)
##
## Residuals:
```

```
##        Min         1Q     Median         3Q        Max
## -795671167  -67804865  -16663653   58261496 2563545746
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.121e+08  2.180e+08  -1.432   0.1550
## X1           2.422e+02  1.297e+02   1.868   0.0645 .
## X2           8.842e+07  4.100e+07   2.157   0.0333 *
## X3           1.341e+06  3.382e+06   0.396   0.6926
## X4_1        -4.062e+08  3.808e+08  -1.067   0.2886
## X5_1         3.900e+08  4.783e+08   0.815   0.4167
## X5_2         7.909e+07  2.661e+08   0.297   0.7669
## X5_3         8.957e+08  4.042e+08   2.216   0.0288 *
## X6_1        -1.085e+08  1.740e+08  -0.623   0.5344
## X6_2        -5.785e+07  1.042e+08  -0.555   0.5798
## X6_3        -2.038e+08  3.706e+08  -0.550   0.5836
## X6_4        -4.702e+07  1.452e+08  -0.324   0.7467
## X7_1         4.008e+07  2.910e+08   0.138   0.8907
## X7_2         2.938e+07  1.183e+08   0.248   0.8044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.52e+08 on 107 degrees of freedom
## Multiple R-squared:   0.36,  Adjusted R-squared:  0.2822
## F-statistic:  4.63 on 13 and 107 DF,  p-value: 2.879e-06
```

```r
#to find SSE(R) and SSR(R)
anova_R = as.data.frame(anova(f))
anova_R
```

```
##             Df       Sum Sq      Mean Sq      F value       Pr(>F)
## X1           1 3.046711e+18 3.046711e+18 24.585074551 2.685813e-06
## X2           1 2.067983e+18 2.067983e+18 16.687339871 8.535194e-05
## X3           1 1.266960e+15 1.266960e+15  0.010223581 9.196510e-01
## X4_1         1 4.234929e+17 4.234929e+17  3.417325212 6.727657e-02
## X5_1         1 1.151999e+18 1.151999e+18  9.295917316 2.894496e-03
## X5_2         1 5.934611e+16 5.934611e+16  0.478886390 4.904261e-01
## X5_3         1 5.970844e+17 5.970844e+17  4.818101246 3.032303e-02
## X6_1         1 2.564731e+16 2.564731e+16  0.206957947 6.500839e-01
## X6_2         1 2.958479e+16 2.958479e+16  0.238730932 6.261241e-01
## X6_3         1 3.508632e+16 3.508632e+16  0.283124839 5.957635e-01
## X6_4         1 1.253755e+16 1.253755e+16  0.101170259 7.510497e-01
## X7_1         1 2.231574e+14 2.231574e+14  0.001800742 9.662309e-01
## X7_2         1 7.641966e+15 7.641966e+15  0.061665936 8.043579e-01
## Residuals  107 1.326000e+19 1.239252e+17           NA           NA
```

```r
#to find SSE(F) and SSR(F)
anova_F =  as.data.frame(anova(lm_pr2))
anova_F
```

```
##             Df       Sum Sq      Mean Sq      F value       Pr(>F)
## X1           1 4.224117e+10 4.224117e+10 2.328643e+02 1.299298e-28
## X2           1 3.396621e+10 3.396621e+10 1.872466e+02 3.006786e-25
## X3           1 8.376367e+03 8.376367e+03 4.617667e-05 9.945908e-01
## X4_1         1 6.380212e+09 6.380212e+09 3.517241e+01 3.754627e-08
```

```
## X5_1        1 6.571736e+09 6.571736e+09 3.622822e+01 2.501226e-08
## X5_2        1 4.017518e+08 4.017518e+08 2.214750e+00 1.396385e-01
## X5_3        1 5.579148e+07 5.579148e+07 3.075635e-01 5.803365e-01
## X6_1        1 2.327260e+06 2.327260e+06 1.282956e-02 9.100306e-01
## X6_2        1 1.156949e+08 1.156949e+08 6.377953e-01 4.262794e-01
## X6_3        1 6.844262e+06 6.844262e+06 3.773059e-02 8.463534e-01
## X6_4        1 1.779593e+08 1.779593e+08 9.810423e-01 3.241762e-01
## X7_1        1 3.210473e+08 3.210473e+08 1.769848e+00 1.862294e-01
## X7_2        1 6.546866e+07 6.546866e+07 3.609112e-01 5.492711e-01
## Residuals 107 1.940961e+10 1.813982e+08           NA           NA
```

```r
nrow(anova_R)
```

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

```r
nrow(anova_F)
```

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

```r
SSR_R = sum(anova_R[1:13,2])
SSE_R = anova_R[14,2]

SSR_F = sum(anova_F[1:13,2])
SSE_F= anova_F[14,2]

n = nrow(pr2_data)

#chi-squared: [SSR(R)/2] / [SSE(F)/n]^2
chiTest = (SSR_R/2) / ((SSE_F/n))^2
print(chiTest)
```

```
## [1] 144.9321
```

```r
#p
chi = qchisq(1-0.05,1)
print(chi)
```

```
## [1] 3.841459
```

Decision Rule:

- If $chiTest \leq \chi^2(1-\alpha,1)$, conclude H_0: constant error variance
- If $chiTest > \chi^2(1-\alpha,1)$, conclude H_1: non-constant error variance

Result: Since $144.9321 > 3.841459$ i.e. $chiTest > \chi^2(1-\alpha,1)$, we conclude H_a. The error variance is not constant.

(c) Use weight least squares regression (perform only one iteration) document your results. (5 pts)

```r
ei_abs = abs(ei)
df1 = as.data.frame(cbind(pr2_data,ei_abs))
lm_ei_2c = lm(ei_abs~X1+X2+X3+X4_1+X5_1+X5_2+X5_3+X6_1+X6_2+X6_3+X6_4+X7_1+X7_2, data=df1)
summary(lm_ei_2c)
```

```
##
## Call:
## lm(formula = ei_abs ~ X1 + X2 + X3 + X4_1 + X5_1 + X5_2 + X5_3 +
##     X6_1 + X6_2 + X6_3 + X6_4 + X7_1 + X7_2, data = df1)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -16628  -3080   -173   1993  33075
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.312e+03  4.392e+03  -0.299  0.76576
## X1           6.283e-03  2.613e-03   2.405  0.01790 *
## X2           1.638e+03  8.261e+02   1.983  0.04990 *
## X3           1.954e+01  6.814e+01   0.287  0.77484
## X4_1        -1.350e+04  7.674e+03  -1.759  0.08144 .
## X5_1         1.116e+04  9.637e+03   1.158  0.24930
## X5_2         1.381e+03  5.362e+03   0.258  0.79723
## X5_3         2.294e+04  8.144e+03   2.816  0.00578 **
## X6_1         9.931e+02  3.506e+03   0.283  0.77750
## X6_2         1.064e+03  2.099e+03   0.507  0.61331
## X6_3        -9.982e+03  7.467e+03  -1.337  0.18415
## X6_4        -7.661e+02  2.926e+03  -0.262  0.79395
## X7_1        -6.194e+02  5.864e+03  -0.106  0.91607
## X7_2         2.269e+03  2.384e+03   0.952  0.34337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7093 on 107 degrees of freedom
## Multiple R-squared:  0.4932, Adjusted R-squared:  0.4316
## F-statistic: 8.009 on 13 and 107 DF,  p-value: 5.691e-11
```

```r
si = lm_ei_2c$fitted.values
wi = 1/(si^2)
```

```r
lm_2c = lm(Y~X1+X2+X3+X4_1+X5_1+X5_2+X5_3+X6_1+X6_2+X6_3+X6_4+X7_1+X7_2, weights=wi, data=pr2_data)
summary(lm_2c)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4_1 + X5_1 + X5_2 + X5_3 + X6_1 +
##     X6_2 + X6_3 + X6_4 + X7_1 + X7_2, data = pr2_data, weights = wi)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8869 -1.1530 -0.0745  0.3800  4.0710
##
## Coefficients: (5 not defined because of singularities)
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  3.999e+03  1.277e-11  3.131e+14   <2e-16 ***
## X1           7.897e-02  5.666e-17  1.394e+15   <2e-16 ***
## X2                  NA         NA         NA       NA
## X3                  NA         NA         NA       NA
## X4_1                NA         NA         NA       NA
## X5_1        -9.302e+03  1.203e+04 -7.730e-01   0.4409
## X5_2        -1.961e+04  1.152e+04 -1.702e+00   0.0915 .
## X5_3         1.718e+04  1.348e+04  1.274e+00   0.2053
## X6_1         3.923e+03  6.819e+03  5.750e-01   0.5663
## X6_2         8.663e+02  3.274e+03  2.650e-01   0.7918
## X6_3                NA         NA         NA       NA
```

```
## X6_4         1.385e+03  3.211e+03  4.310e-01   0.6671
## X7_1         1.388e+04  1.162e+04  1.195e+00   0.2346
## X7_2                NA         NA         NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.295 on 112 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.428e+29 on 8 and 112 DF,  p-value: < 2.2e-16
```

(d) Compare your model in part a against the regression tree and Neural Network Model, and calculate the SSE for each model, which method has the lowest SSE? And explain which model you will choose. (10 pts)

```r
yHat_lm = lm_pr2$fitted.values
yAct = pr2_data$Y
SSE_lm = sum((yHat_lm-yAct)^2)
SSE_lm
```

```
## [1] 19409611507
```

```r
tree_pr2d = rpart(Y~X1+X2+X3+X4_1+X5_1+X5_2+X5_3+X6_1+X6_2+X6_3+X6_4+X7_1+X7_2, data=pr2_data)
```

```r
yHat_tree = predict(tree_pr2d, pr2_data)
yAct = pr2_data$Y
SSE_tree = sum((yHat_tree-yAct)^2)
SSE_tree
```

```
## [1] 38120812350
```

```r
#Scale training data
pr2_num = pr2_data[,c("Y", "X1", "X2", "X3")]
max = apply(pr2_num, 2, max)
min = apply(pr2_num, 2, min)
scaled_pr2_data = as.data.frame(scale(pr2_num, center=min, scale=max-min))
new_df = pr2_data
new_df[,c("Y", "X1", "X2", "X3")] = scaled_pr2_data
summary(new_df)
```

```
##        Y                  X1                 X2               X3
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.01009   1st Qu.:0.01928   1st Qu.:0.0000   1st Qu.:0.1020
##  Median :0.05449   Median :0.11337   Median :0.0000   Median :0.1020
##  Mean   :0.12383   Mean   :0.27387   Mean   :0.1832   Mean   :0.2474
##  3rd Qu.:0.13733   3rd Qu.:0.48417   3rd Qu.:0.3333   3rd Qu.:0.3878
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
##  X4      X5      X6      X7          X4_1             X5_1
##  1:27    1: 8    1:32    1:64    Min.   :0.0000   Min.   :0.00000
##  2:94    2:56    2:20    2:39    1st Qu.:0.0000   1st Qu.:0.00000
##          3:18    3: 1    3:18    Median :0.0000   Median :0.00000
##          4:39    4: 7            Mean   :0.2231   Mean   :0.06612
##                  5:61            3rd Qu.:0.0000   3rd Qu.:0.00000
##                                  Max.   :1.0000   Max.   :1.00000
##       X5_2             X5_3             X6_1             X6_2
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.4628   Mean   :0.1488   Mean   :0.2645   Mean   :0.1653
```

```
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##       X6_3               X6_4              X7_1             X7_2
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000000   Median :0.00000   Median :1.0000   Median :0.0000
##  Mean   :0.008264   Mean   :0.05785   Mean   :0.5289   Mean   :0.3223
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
```

```
NN = neuralnet(Y~X1+X2+X3+X4_1+X5_1+X5_2+X5_3+X6_1+X6_2+X6_3+X6_4+X7_1+X7_2, data=new_df, hidden=14 , li
plot(NN)
```

```
maxY= max(pr2_data$Y)
minY = min(pr2_data$Y)
```

```
yHat_NN = predict(NN, new_df)*(maxY-minY)+minY
yAct = new_df$Y*(maxY-minY)+minY
SSE_NN = sum((yHat_NN-yAct)^2)
SSE_NN
```

```
## [1] 4790437985
```

```
cbind(SSE_lm, SSE_tree, SSE_NN)
```

```
##           SSE_lm    SSE_tree     SSE_NN
## [1,] 19409611507 38120812350 4790437985
```

Interpretation

We see that Neural network has the lowest SSE, however, we should use the linear model as it gives a good balance between predictability and interpretability.

Question 3 Use the PR3_Dataset data: Y is the outcome variable and indicates the number of awards earned by students at a high school in a year, X1 is a categorical predictor variable with three levels indicating the type of program in which the students were enrolled. It is coded as 1 = "General", 2 = "Academic" and 3 = "Social", and X2 is a continuous predictor variable and represents students' scores on their math final exam. Answer the following questions: (20pts)

(a) Build a model to predict the number of awards earned by students, is the model significant? (5pts)

```
pr3_data = read.csv("PR3_Dataset.csv")
pr3_data$X1 = as.factor(pr3_data$X1)
summary(pr3_data)
```

```
##        Y          X1            X2
##  Min.   :0.00   1: 45   Min.   :33.00
##  1st Qu.:0.00   2:105   1st Qu.:45.00
##  Median :0.00   3: 50   Median :52.00
##  Mean   :0.63           Mean   :52.65
##  3rd Qu.:1.00           3rd Qu.:59.00
##  Max.   :6.00           Max.   :75.00
```

```
pr3_data$X1_1 = ifelse(pr3_data$X1==1, 1,0)
pr3_data$X1_2 = ifelse(pr3_data$X1==2, 1,0)
```

```
pmod_pr3 = glm(Y~X1_1+X1_2+X2, data=pr3_data, family=poisson)
summary(pmod_pr3)
```

```
##
```

```
## Call:
## glm(formula = Y ~ X1_1 + X1_2 + X2, family = poisson, data = pr3_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.87732    0.62818  -7.764 8.21e-15 ***
## X1_1        -0.36981    0.44107  -0.838   0.4018
## X1_2         0.71405    0.32001   2.231   0.0257 *
## X2           0.07015    0.01060   6.619 3.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
```

```r
nothing = glm(Y~1, data=pr3_data)
```

```r
anova(pmod_pr3, nothing, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1_1 + X1_2 + X2
## Model 2: Y ~ 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       196     189.45
## 2       199     220.62 -3   -31.17 7.826e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation

We see tha the model is significant. And X2 and X1_2 are the significant variables.

(b) Find the predicted number awards earned by students given the independent variables below and calculate 99% confidence interval. (5pts)

```r
Xh = data.frame(cbind(X1_2=1,X1_1=0,X2=75))
predict(pmod_pr3, Xh, type="response", se.fit=TRUE)
```

```
## $fit
##        1
## 2.998657
##
## $se.fit
##         1
## 0.5099867
##
## $residual.scale
```

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

```r
pre1 = predict(pmod_pr3, Xh, type="link", se.fit=TRUE)
LowerCL = pre1$fit-qnorm(0.01,1)*pre1$se.fit
UpperCL = pre1$fit-qnorm(0.01,1)*pre1$se.fit
Prediction = pre1$fit
round(cbind(LowerCL,Prediction,UpperCL),3)
```

```
##   LowerCL Prediction UpperCL
## 1   1.324      1.098   1.324
```

(c) Fit the negative binomial model and compare it the model built in part a, which model is better? (10pts)

```r
lmod_pr3 = glm(Y~X1_1+X1_2+X2, data=pr3_data, family=negative.binomial(1))
summary(lmod_pr3)
```

```
##
## Call:
## glm(formula = Y ~ X1_1 + X1_2 + X2, family = negative.binomial(1),
##     data = pr3_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5791  -0.7761  -0.4828   0.1766   1.6930
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.98117    0.69415  -7.176 1.45e-11 ***
## X1_1        -0.36235    0.42169  -0.859   0.3912
## X1_2         0.68625    0.32773   2.094   0.0376 *
## X2           0.07226    0.01254   5.761 3.20e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.7114424)
##
##     Null deviance: 182.43  on 199  degrees of freedom
## Residual deviance: 123.41  on 196  degrees of freedom
## AIC: 383.97
##
## Number of Fisher Scoring iterations: 4
```

Interpretation

We see that the residual deviance for negative binomial model is 123 compared to 189 for poission regression model. Thus, this model gives a better fit.

Question 4 Use the PR4_Dataset data, Y is a dichotomous response variable. X2, X3, and X4 are categorical variables: X2 has 3 levels, X3 and X4 have 2 levels (create dummy variables for the categorical variables). Answer the questions below: (20pts)

```r
pr4_data = read.csv("PR4_Dataset.csv")
summary(pr4_data)
```

```
##        X1              X2              X3               X4
##  Min.   : 1.00   Min.   :1.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:10.75   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :21.00   Median :2.000   Median :0.0000   Median :0.0000
```

```
##  Mean   :25.18   Mean   :1.964   Mean   :0.4031   Mean   :0.2908
##  3rd Qu.:35.00   3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :85.00   Max.   :3.000   Max.   :1.0000   Max.   :1.0000
##        Y
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.5459
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```
pr4_data$X2_1 = ifelse(pr4_data$X2==1,1,0)
pr4_data$X2_2 = ifelse(pr4_data$X2==2,1,0)

pr4_data$X1.X2_1 = pr4_data$X1*pr4_data$X2_1
pr4_data$X1.X2_2 = pr4_data$X1*pr4_data$X2_2
pr4_data$X1.X3 = pr4_data$X1*pr4_data$X3
pr4_data$X1.X4 = pr4_data$X1*pr4_data$X4

pr4_data$X2_2.X2_1 = pr4_data$X2_2*pr4_data$X2_1
pr4_data$X2_2.X3 = pr4_data$X2_2*pr4_data$X3
pr4_data$X2_2.X4 = pr4_data$X2_2*pr4_data$X4

pr4_data$X2_1.X3 = pr4_data$X2_2*pr4_data$X3
pr4_data$X2_1.X4 = pr4_data$X2_2*pr4_data$X4

pr4_data$X3.X4 = pr4_data$X3*pr4_data$X4

new_pr4_data = pr4_data[,-which(colnames(pr4_data)%in%c("X2"))]
```

(a) Fit a regression model containing the predictor variables in first-order terms and interaction terms (e.g X1*X2) for all pairs of predictor variables. (5pts)

```
lmod_pr4 = glm(Y~., data=new_pr4_data, family=binomial)
summary(lmod_pr4)
```

```
##
## Call:
## glm(formula = Y ~ ., family = binomial, data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4104  -0.8787   0.4004   0.8188   1.9986
##
## Coefficients: (3 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.927079   0.514948  -3.742 0.000182 ***
## X1           0.037842   0.015132   2.501 0.012394 *
## X3           1.090199   0.687652   1.585 0.112877
## X4          -1.013539   0.962094  -1.053 0.292125
## X2_1         2.035043   0.683299   2.978 0.002899 **
## X2_2         0.778272   0.786762   0.989 0.322561
## X1.X2_1     -0.002263   0.023817  -0.095 0.924290
## X1.X2_2      0.006189   0.027358   0.226 0.821028
## X1.X3       -0.021063   0.022458  -0.938 0.348320
```

```
## X1.X4        0.021014   0.025682   0.818 0.413210
## X2_2.X2_1          NA         NA      NA       NA
## X2_2.X3     -0.311135   0.786612  -0.396 0.692446
## X2_2.X4     -0.057054   0.930643  -0.061 0.951115
## X2_1.X3            NA         NA      NA       NA
## X2_1.X4            NA         NA      NA       NA
## X3.X4        0.958622   0.831750   1.153 0.249101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 213.04  on 183  degrees of freedom
## AIC: 239.04
##
## Number of Fisher Scoring iterations: 5
```

(b) Use the likelihood ratio test to determine whether all interaction terms can be dropped from the regression model; State the alternatives, full and reduced models, decision rule, and conclusion. (5pts)

```r
lmod_red =  glm(Y~X1+X2_1+X2_2+X3+X4, data=new_pr4_data, family=binomial)
```

```r
anova(lmod_pr4, lmod_red, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X2_1 + X1.X2_2 + X1.X3 +
##     X1.X4 + X2_2.X2_1 + X2_2.X3 + X2_2.X4 + X2_1.X3 + X2_1.X4 +
##     X3.X4
## Model 2: Y ~ X1 + X2_1 + X2_2 + X3 + X4
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       183     213.04
## 2       190     215.36 -7  -2.3173   0.9402
```

Interpretation

We see that the p-value is 0.94 which means that there is not much difference between the deviance of the two models. Thus, we can drop all the interaction terms.

(c) Perform the backward variable selection method to find a model where all variables are significant and Conduct the Hosmer-Lemeshow goodness of fit test for the appropriateness of the logistic regression function by forming five groups. State the alternatives, decision rule, and conclusion. (5pts)

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X2_1 + X1.X2_2 +
    X1.X3 + X1.X4 + X2_2.X3 + X2_2.X4 + X3.X4, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X2_1 + X1.X2_2 +
##     X1.X3 + X1.X4 + X2_2.X3 + X2_2.X4 + X3.X4, family = binomial,
##     data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4104  -0.8787   0.4004   0.8188   1.9986
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.927079   0.514948  -3.742 0.000182 ***
## X1           0.037842   0.015132   2.501 0.012394 *
## X3           1.090199   0.687652   1.585 0.112877
## X4          -1.013539   0.962094  -1.053 0.292125
## X2_1         2.035043   0.683299   2.978 0.002899 **
## X2_2         0.778272   0.786762   0.989 0.322561
## X1.X2_1     -0.002263   0.023817  -0.095 0.924290
## X1.X2_2      0.006189   0.027358   0.226 0.821028
## X1.X3       -0.021063   0.022458  -0.938 0.348320
## X1.X4        0.021014   0.025682   0.818 0.413210
## X2_2.X3     -0.311135   0.786612  -0.396 0.692446
## X2_2.X4     -0.057054   0.930643  -0.061 0.951115
## X3.X4        0.958622   0.831750   1.153 0.249101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 213.04  on 183  degrees of freedom
## AIC: 239.04
##
## Number of Fisher Scoring iterations: 5
```

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X2_1 + X1.X2_2 +
    X1.X3 + X1.X4 + X2_2.X3 + X3.X4, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X2_1 + X1.X2_2 +
##     X1.X3 + X1.X4 + X2_2.X3 + X3.X4, family = binomial, data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4060  -0.8797   0.4009   0.8146   1.9981
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.925905   0.514440  -3.744 0.000181 ***
## X1           0.037888   0.015116   2.506 0.012194 *
## X3           1.096494   0.680014   1.612 0.106862
## X4          -1.026495   0.938639  -1.094 0.274131
## X2_1         2.034763   0.683270   2.978 0.002902 **
## X2_2         0.778291   0.786689   0.989 0.322505
## X1.X2_1     -0.002271   0.023817  -0.095 0.924035
## X1.X2_2      0.005821   0.026684   0.218 0.827304
## X1.X3       -0.021107   0.022448  -0.940 0.347098
## X1.X4        0.021098   0.025636   0.823 0.410521
## X2_2.X3     -0.325586   0.750181  -0.434 0.664281
## X3.X4        0.949308   0.817269   1.162 0.245414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 213.04  on 184  degrees of freedom
## AIC: 237.04
##
## Number of Fisher Scoring iterations: 5
```

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X2_2 +
    X1.X3 + X1.X4 + X2_2.X3 + X3.X4, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X2_2 + X1.X3 +
##     X1.X4 + X2_2.X3 + X3.X4, family = binomial, data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4190  -0.8808   0.3993   0.8131   1.9906
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.907322   0.474366  -4.021 5.80e-05 ***
## X1           0.037241   0.013462   2.766  0.00567 **
## X3           1.100685   0.677232   1.625  0.10411
## X4          -1.031495   0.938206  -1.099  0.27158
## X2_1         1.982712   0.408844   4.850 1.24e-06 ***
## X2_2         0.755695   0.749565   1.008  0.31337
## X1.X2_2      0.006678   0.025114   0.266  0.79033
## X1.X3       -0.021543   0.022019  -0.978  0.32788
## X1.X4        0.021203   0.025647   0.827  0.40839
## X2_2.X3     -0.322173   0.749101  -0.430  0.66714
## X3.X4        0.954840   0.815876   1.170  0.24187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 213.05  on 185  degrees of freedom
## AIC: 235.05
##
## Number of Fisher Scoring iterations: 5
```

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2 +
    X1.X3 + X1.X4 + X2_2.X3 + X3.X4, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X3 + X1.X4 +
##     X2_2.X3 + X3.X4, family = binomial, data = new_pr4_data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4450  -0.8876   0.3944   0.8233   2.0006
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.93227    0.46676  -4.140 3.48e-05 ***
## X1           0.03815    0.01307   2.919  0.00351 **
## X3           1.07292    0.66908   1.604  0.10881
## X4          -1.03739    0.93942  -1.104  0.26947
## X2_1         1.98726    0.40969   4.851 1.23e-06 ***
## X2_2         0.89893    0.52013   1.728  0.08394 .
## X1.X3       -0.02049    0.02178  -0.941  0.34664
## X1.X4        0.02143    0.02570   0.834  0.40437
## X2_2.X3     -0.31419    0.74576  -0.421  0.67353
## X3.X4        0.97091    0.81481   1.192  0.23343
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 213.12  on 186  degrees of freedom
## AIC: 233.12
##
## Number of Fisher Scoring iterations: 5
```

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2 +
    X1.X3 + X1.X4 +  X3.X4, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X3 + X1.X4 +
##     X3.X4, family = binomial, data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4152  -0.8609   0.4051   0.8137   1.9836
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.89266    0.45440  -4.165 3.11e-05 ***
## X1           0.03794    0.01306   2.904  0.00368 **
## X3           0.95796    0.60956   1.572  0.11605
## X4          -1.02895    0.93778  -1.097  0.27255
## X2_1         1.99344    0.40830   4.882 1.05e-06 ***
## X2_2         0.77027    0.42146   1.828  0.06761 .
## X1.X3       -0.01991    0.02167  -0.919  0.35812
## X1.X4        0.02069    0.02558   0.809  0.41865
## X3.X4        0.97262    0.81407   1.195  0.23218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 213.30  on 187  degrees of freedom
## AIC: 231.3
##
## Number of Fisher Scoring iterations: 5
```

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2 +
    X1.X3 + X3.X4, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X1.X3 + X3.X4,
##     family = binomial, data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3109  -0.8482   0.4070   0.8146   2.0031
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.94273    0.45183  -4.300 1.71e-05 ***
## X1           0.04048    0.01281   3.161  0.00157 **
## X3           0.85261    0.59333   1.437  0.15072
## X4          -0.41941    0.54927  -0.764  0.44512
## X2_1         1.97289    0.40681   4.850 1.24e-06 ***
## X2_2         0.77169    0.42097   1.833  0.06679 .
## X1.X3       -0.01411    0.02066  -0.683  0.49440
## X3.X4        0.86949    0.80061   1.086  0.27746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 213.98  on 188  degrees of freedom
## AIC: 229.98
##
## Number of Fisher Scoring iterations: 4
```

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2 +
    X3.X4, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2 + X3.X4, family = binomial,
##     data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3778  -0.8715   0.3752   0.8070   1.9513
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.81309    0.40397  -4.488 7.18e-06 ***
## X1           0.03533    0.01008   3.505 0.000456 ***
## X3           0.57251    0.42719   1.340 0.180186
## X4          -0.37945    0.54131  -0.701 0.483317
## X2_1         1.94640    0.40339   4.825 1.40e-06 ***
## X2_2         0.74916    0.41932   1.787 0.074005 .
## X3.X4        0.74359    0.77981   0.954 0.340312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 214.44  on 189  degrees of freedom
## AIC: 228.44
##
## Number of Fisher Scoring iterations: 4
```

```r
model = glm(Y ~ X1 + X3 + X4 + X2_1 + X2_2, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4 + X2_1 + X2_2, family = binomial,
##     data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2845  -0.8649   0.3885   0.8206   1.9874
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.89685    0.39614  -4.788 1.68e-06 ***
## X1           0.03563    0.01003   3.552 0.000382 ***
## X3           0.79651    0.36120   2.205 0.027441 *
## X4          -0.02908    0.39303  -0.074 0.941026
## X2_1         1.95235    0.40287   4.846 1.26e-06 ***
## X2_2         0.77767    0.41703   1.865 0.062210 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 215.36  on 190  degrees of freedom
## AIC: 227.36
##
## Number of Fisher Scoring iterations: 4
```

```r
model = glm(Y ~ X1 + X3 + X2_1 + X2_2, data=new_pr4_data, family=binomial)
summary(model)
```

```
##
## Call:
```

```
## glm(formula = Y ~ X1 + X3 + X2_1 + X2_2, family = binomial, data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2898  -0.8648   0.3887   0.8149   1.9887
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.899522   0.394647  -4.813 1.49e-06 ***
## X1           0.035471   0.009796   3.621 0.000294 ***
## X3           0.789524   0.348572   2.265 0.023511 *
## X2_1         1.953575   0.402550   4.853 1.22e-06 ***
## X2_2         0.779244   0.416551   1.871 0.061386 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 215.36  on 191  degrees of freedom
## AIC: 225.36
##
## Number of Fisher Scoring iterations: 4
```

NOTE I performed a manual backward elimination due to a bug in the ols_step_backward_p() function – same library gave me error in the earlier question.

Interpretation

We remove the variable with highest p-value at each step and the above model where all variables (X1, X3, X2_1 and X2_2) are significant.

```r
lmod_pr4_best = glm(Y ~ X1 + X3 + X2_1 + X2_2, data=new_pr4_data, family=binomial)
summary(lmod_pr4_best)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X2_1 + X2_2, family = binomial, data = new_pr4_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2898  -0.8648   0.3887   0.8149   1.9887
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.899522   0.394647  -4.813 1.49e-06 ***
## X1           0.035471   0.009796   3.621 0.000294 ***
## X3           0.789524   0.348572   2.265 0.023511 *
## X2_1         1.953575   0.402550   4.853 1.22e-06 ***
## X2_2         0.779244   0.416551   1.871 0.061386 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
```

```
## Residual deviance: 215.36  on 191  degrees of freedom
## AIC: 225.36
##
## Number of Fisher Scoring iterations: 4
```

(d) Use the model developed in part c and predict probability of Y for the following two cases and calculate 95% confidence interval. (5pts)

```
Xh = data.frame(cbind(X1=c(60,11),X2_1=c(1,0),X2_2=c(0,1), X3=c(0,1), X4=c(0,1)))
pre1 = predict(lmod_pr4_best, Xh, type="link", se.fit=T)
LowerCL = pre1$fit-1.96*pre1$se.fit; UpperCL = pre1$fit+1.96*pre1$se.fit
Prediction = pre1$fit
results = round(cbind(LowerCL,Prediction,UpperCL),3)
ilogit(results)
```

```
##     LowerCL Prediction   UpperCL
## 1 0.7724153  0.8986214 0.9586320
## 2 0.3389448  0.5147457 0.6871868
```

Question 5 Use the PR4_Dataset data. All variables including Y are continuous variables. Fit a regression model to predict Y. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? check to see if auto-correlation persists in the data set, write null and alternatives hypothesis and calculate p value. (5 pts)

```
pr5_data = read.csv("PR5_Dataset.csv")
summary(pr5_data)
```

```
##        Y                X1                 X2              X3
##  Min.   :  -5.0   Min.   :0.04750   Min.   :59.00   Min.   :0.0000
##  1st Qu.: 262.5   1st Qu.:0.06250   1st Qu.:63.00   1st Qu.:0.0000
##  Median : 754.0   Median :0.07500   Median :65.00   Median :0.0000
##  Mean   : 937.4   Mean   :0.07448   Mean   :65.53   Mean   :0.3721
##  3rd Qu.:1167.0   3rd Qu.:0.08750   3rd Qu.:68.00   3rd Qu.:0.0000
##  Max.   :5105.0   Max.   :0.09500   Max.   :72.00   Max.   :5.0000
##        X4             X5               X6
##  Min.   :1151   Min.   : 538.0   Min.   :0.020
##  1st Qu.:1796   1st Qu.: 724.0   1st Qu.:0.445
##  Median :2422   Median : 832.0   Median :0.860
##  Mean   :3052   Mean   : 926.1   Mean   :1.190
##  3rd Qu.:4018   3rd Qu.:1002.5   3rd Qu.:2.130
##  Max.   :7142   Max.   :2388.0   Max.   :3.420
```

```
lm_pr5 = lm(Y~., data=pr5_data)
summary(lm_pr5)
```

```
##
## Call:
## lm(formula = Y ~ ., data = pr5_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1063.26  -329.03   -77.92   239.84  1434.78
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.287e+03  2.171e+03  -0.593   0.5570
## X1           9.509e+03  7.828e+03   1.215   0.2324
```

```
## X2            1.889e+01  3.119e+01   0.606   0.5484
## X3            6.129e+02  8.021e+01   7.641 4.82e-09 ***
## X4           -1.670e-01  8.161e-02  -2.046   0.0481 *
## X5            6.445e-01  2.513e-01   2.564   0.0146 *
## X6           -3.102e+01  8.881e+01  -0.349   0.7289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 528.4 on 36 degrees of freedom
## Multiple R-squared:  0.771,  Adjusted R-squared:  0.7329
## F-statistic:  20.2 on 6 and 36 DF,  p-value: 3.491e-10
```

Multi-collinearity

```r
vif(lm_pr5)
```

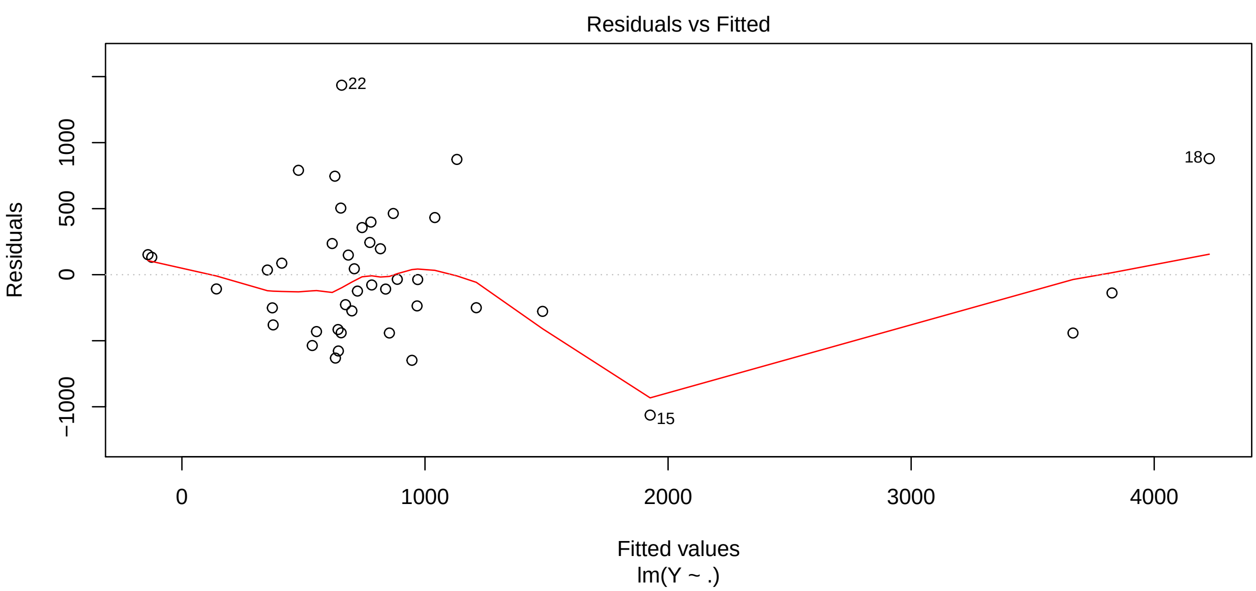
```
##       X1       X2       X3       X4       X5       X6
## 2.656652 1.653578 1.337545 2.686929 1.367983 1.098401
```

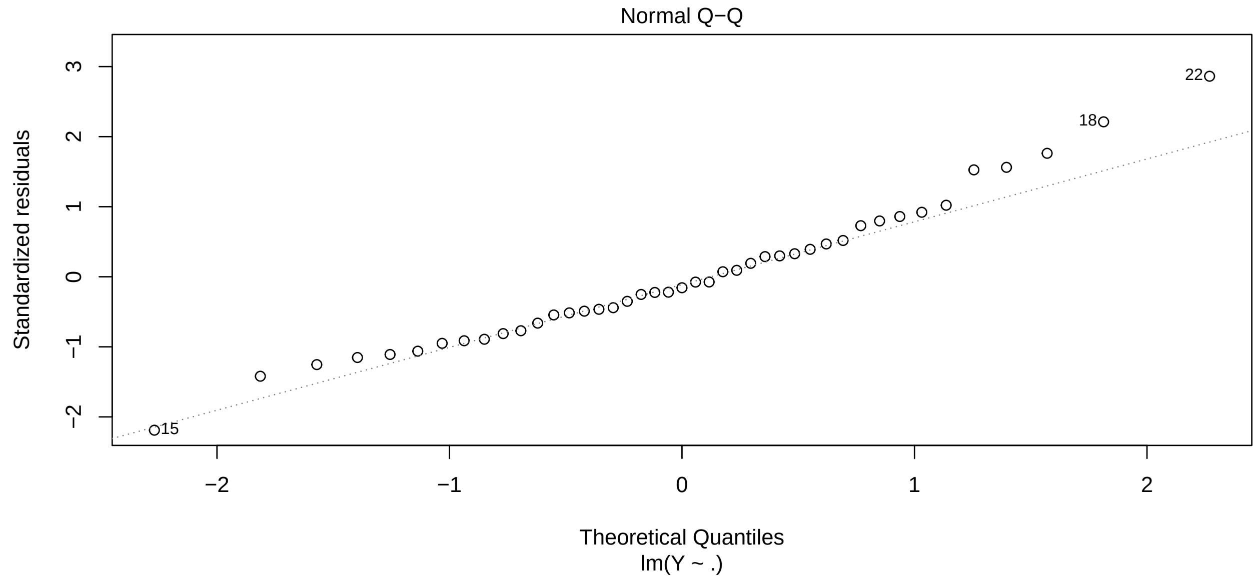
Interpretation

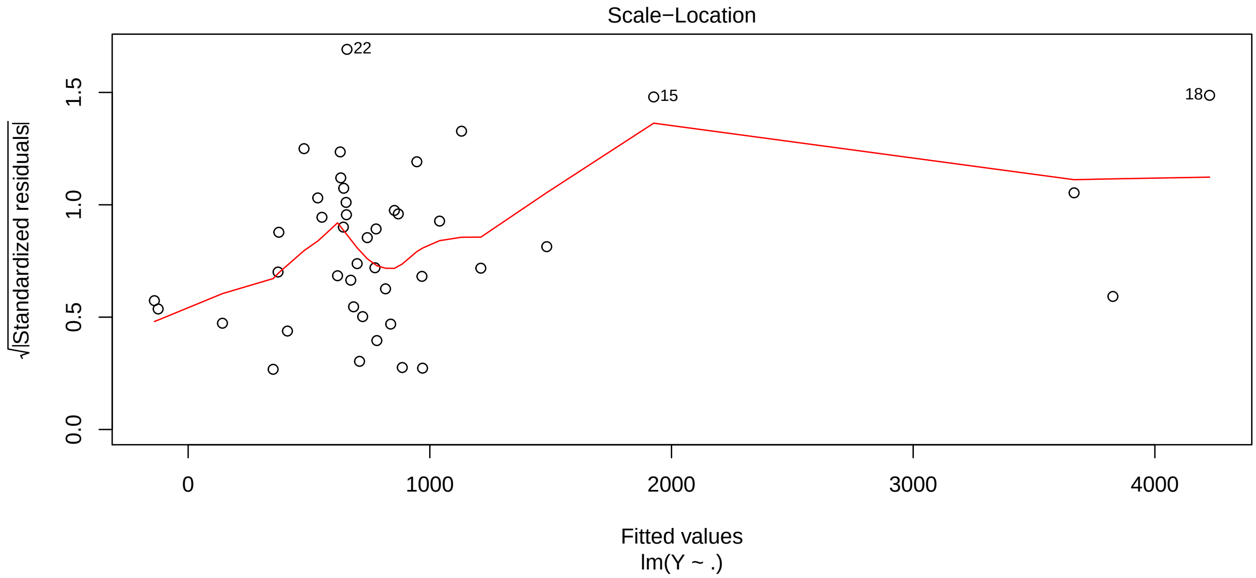
Since all the VIFs are <10, we can say that there is not any serious multi-collinearity in the given data.

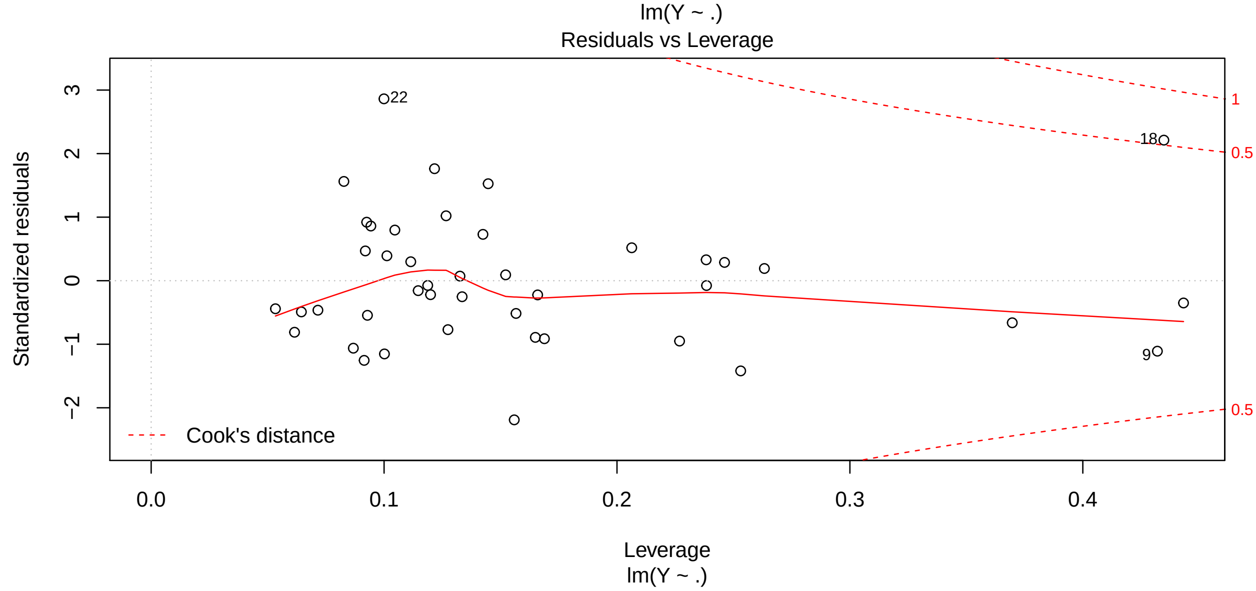
Normal Error & Constant Variance

```r
plot(lm_pr5)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Y ~ .)
22
18
15


Scale−Location
√|Standardized residuals|
Fitted values
lm(Y ~ .)
22
15
18


Residuals vs Leverage
Standardized residuals
Leverage
lm(Y ~ .)
Cook's distance
22
18
9
1
0.5
0.5

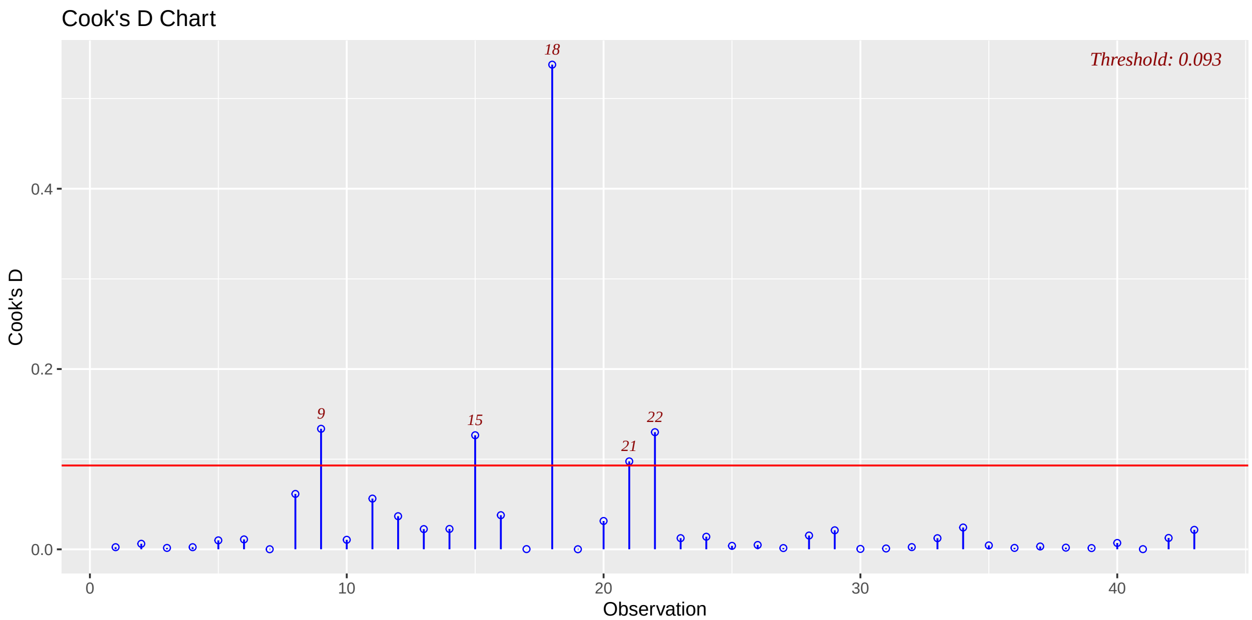
Interpretation

Normal Probability Plot: We can see that the this plot is mostly linear, so the error terms are in agreement with the normal distribution.

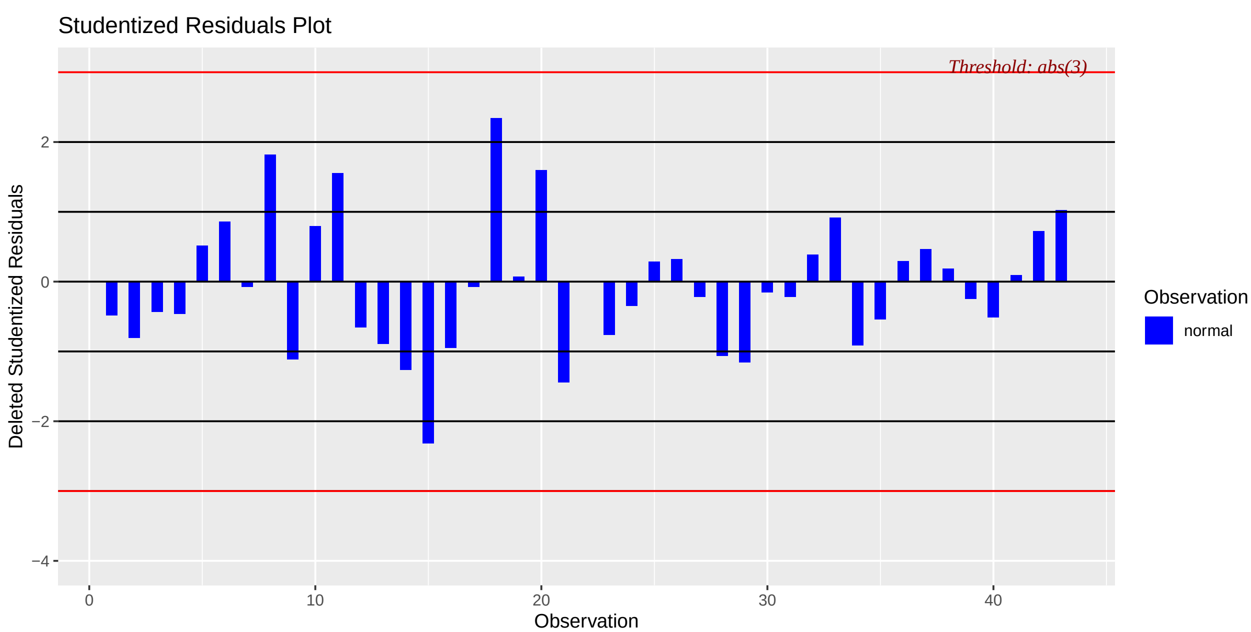
It also show that the error terms have a constant variance. However, we do see some outliers.

Outliers/Influential Points

```
ols_plot_cooksd_chart(lm_pr5)
```




```
ols_plot_resid_stud(lm_pr5)
```




```
#outliers in Xs
model = lm_pr5
df = pr5_data
n = nrow(df)
```

```
p = length(model$coefficients)
hii = hatvalues(model)
index = hii>2*p/n
print("Hat values outliers")
```

```
## [1] "Hat values outliers"
```

```
index[index]
```

```
##    9   12   18   24
## TRUE TRUE TRUE TRUE
```

Interpretation

Cook's distance and studentised residuals don't show any clear outliers in the dataset.

Hat values do show some outliers as printed above.

Overall I think the model is a good fit to the data with no outliers, multicollinearity and with error constant variance.

```
dwtest(lm_pr5)
```

```
##
##  Durbin-Watson test
##
## data:  lm_pr5
## DW = 1.9618, p-value = 0.2903
## alternative hypothesis: true autocorrelation is greater than 0
```

Interpretation

There is no auto-correlation in the data.