Homework 6_1

Jyoti Chaudhary

March 22, 2018

Solution 1.a

```
# Fitting the MLR model using LM function

data_B15 <- read.csv(paste(getwd(),"/data_table_B15.csv",sep = ""),header=T)

colnames(data_B15) <- c('x6','y', 'x1', 'x2', 'x3', 'x4', 'x5')

data_B15_fit <- lm(formula = data_B15$y ~ data_B15$x1 + data_B15$x2 + data_B15$x3 + data_B15$x4
 + data_B15$x5, data = data_B15)

summary(data_B15_fit)
```

```
##
## Call:
## lm(formula = data_B15$y ~ data_B15$x1 + data_B15$x2 + data_B15$x3 +
##     data_B15$x4 + data_B15$x5, data = data_B15)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -91.38 -18.97  -3.56  16.00  91.83
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 995.63646   91.64099  10.865 3.35e-15 ***
## data_B15$x1   1.40734    0.68914   2.042 0.046032 *
## data_B15$x2 -14.80139    7.02747  -2.106 0.039849 *
## data_B15$x3   3.19909    0.62231   5.141 3.89e-06 ***
## data_B15$x4  -0.10797    0.13502  -0.800 0.427426
## data_B15$x5   0.35518    0.09096   3.905 0.000264 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.09 on 54 degrees of freedom
## Multiple R-squared:  0.6746, Adjusted R-squared:  0.6444
## F-statistic: 22.39 on 5 and 54 DF,  p-value: 4.407e-12
```

The MLR equation from above R output is:

y-hat = 996 + 1.41 * x1 - 14.8 * x2 + 3.2 * x3 - 0.10 * x4 + 0.3552 * x5

where

City = x6 Mort = y PRECIP = x1 EDUC = x2 Nonwhite = x3 NOX = x4 SO2 = x5

Solution (1.b)

H0: beta1 = beta2 = beta3 = beta4 = beta5 = 0 vs. H1: at least one of 5 is non-zero.

From above, it can be seen that F-statistic: 22.39 with p-value approximately equals 0.000. So, we reject the null hypothesis

So, the data provides sufficient evidence to support the fact that the regression model is significant.

Solution (1.c)

Use t tests to assess the contribution of each regressor to the model. Discuss your findings.

From above model fit summary statistics, it can be seen that beta4 = NOX has a high p-value of approx = 0.43. Beta4 is barely significant at 5% level but not for any alpha < 0.427. All other predictors have very low p-value, hence all other predictors are significant for smaller values of alpha.

PRECIP (x1), EDUC(x2), NONWHITE(x3) and SO2(x5) contribute to the model.

solution (1.d)

Find a 95% CI for the regression coefficient for SO2.

```
confint(data_B15_fit, "data_B15$x5", level = 0.95)
```

```
##                  2.5 %    97.5 %
## data_B15$x5 0.1728118 0.5375405
```

From above output, the 95% C.I for SO2 is (0.1728, 0.5375)

solution (1.e)

Run all possible models and choose the best one with justifications.

```
fit_all =regsubsets(y ~  x1 + x2 + x3 + x4 + x5, data=data_B15, nbest=10,really.big=T, intercept
=T)


model_fit_all =summary(fit_all)[[1]]
RSQ =summary(fit_all)[[2]]
SSE=summary(fit_all)[[3]]
ADJR2 =summary(fit_all)[[4]]
Cp=summary(fit_all)[[5]]
BIC=summary(fit_all)[[6]]

fit_satistics =cbind(model_fit_all,RSQ,SSE,ADJR2,Cp,BIC)
fit_satistics
```

```
##   (Intercept) x1 x2 x3 x4 x5         RSQ       SSE       ADJR2         Cp
## 1           1  0  0  1  0  0 0.419250810 132570.75  0.40923789  40.364406
## 1           1  0  1  0  0  0 0.261104467 168671.67  0.24836489  66.605818
## 1           1  1  0  0  0  0 0.259582453 169019.11  0.24681663  66.858367
## 1           1  0  0  0  0  1 0.181435802 186858.07  0.16732263  79.825334
## 1           1  0  0  0  1  0 0.005731729 226966.98 -0.01141083 108.980121
## 2           1  0  1  1  0  0 0.566777866  98893.95  0.55157709  17.885066
## 2           1  0  0  1  0  1 0.525110918 108405.49  0.50844814  24.798912
## 2           1  1  0  0  0  1 0.493064509 115720.90  0.47527730  30.116411
## 2           1  1  0  1  0  0 0.492959663 115744.83  0.47516877  30.133808
## 2           1  0  0  1  1  0 0.427758360 130628.69  0.40767971  40.952738
## 2           1  0  1  0  0  1 0.360314920 146024.36  0.33786983  52.143709
## 2           1  1  1  0  0  0 0.349356056 148526.00  0.32652644  53.962126
## 2           1  1  0  0  1  0 0.298159792 160212.85  0.27353382  62.457183
## 2           1  0  1  0  1  0 0.262948911 168250.63  0.23708747  68.299767
## 2           1  0  0  0  1  1 0.257386457 169520.40  0.23132984  69.222752
## 3           1  1  0  1  0  1 0.640564447  82050.29  0.62130897   7.641570
## 3           1  0  1  1  0  1 0.629800512  84507.43  0.60996840   9.427643
## 3           1  0  0  1  1  1 0.587097292  94255.53  0.56497750  16.513439
## 3           1  1  1  1  0  0 0.578671932  96178.83  0.55610079  17.911469
## 3           1  0  1  1  1  0 0.566777911  98893.94  0.54356959  19.885058
## 3           1  1  1  0  0  1 0.514980267 110718.07  0.48899707  28.479903
## 3           1  1  0  1  1  0 0.497230898 114769.82  0.47029684  31.425077
## 3           1  1  0  0  1  1 0.493080859 115717.17  0.46592448  32.113698
## 3           1  1  1  0  1  0 0.386682523 140005.29  0.35382623  49.768501
## 3           1  0  1  0  1  1 0.376902004 142237.94  0.34352175  51.391394
## 4           1  1  1  1  0  1 0.670710400  75168.71  0.64676207   4.639416
## 4           1  0  1  1  1  1 0.649430493  80026.39  0.62393453   8.170417
## 4           1  1  0  1  1  1 0.647828947  80391.99  0.62221651   8.436163
## 4           1  1  1  1  1  0 0.582676441  95264.70  0.55232564  19.246996
## 4           1  1  1  0  1  1 0.515301392 110644.77  0.48005058  30.426618
## 5           1  1  1  1  1  1 0.674563903  74289.05  0.64443093   6.000000
##          BIC
## 1 -24.417489
## 1  -9.967235
## 1  -9.843771
## 1  -3.823518
## 1   7.843796
## 2 -37.907247
## 2 -32.397407
## 2 -28.479258
## 2 -28.466849
## 2 -21.208602
## 2 -14.523724
## 2 -13.504530
## 2  -8.959938
## 2  -6.022850
## 2  -5.571736
## 3 -45.015845
## 3 -43.245418
## 3 -36.695219
## 3 -35.483231
## 3 -33.812908
```

```
## 3 -27.036564
## 3 -24.880077
## 3 -24.386848
## 3 -12.954976
## 3 -12.005710
## 4 -46.177338
## 4 -42.420054
## 4 -42.146574
## 4 -31.961883
## 4 -22.981958
## 5 -42.789282
```

We can see from above result that the model that excludes NOX is the best model:

MORT (y) ~ PRECIP (x1) + EDUC(x2) + NONWHITE(x3) + SO2(x5)

This is based on this model's highest Cp value, highest Adjusted R-square and lowest BIC value.

As we had determined in solution (1.c), we predictor x4(NOX) is not a significant parameter, hence removing this predictor resulted in a better model fit.

Solution (1.f)

Run forward, backward and stepwise regression on the data.

```
fit1 <- lm(y ~  x1 + x2 + x3 + x4 + x5, data=data_B15)

# forward selection
step(lm(y ~ 1, data=data_B15), scope=list(lower=~1, upper=fit1), direction='forward')
```

```
## Start:  AIC=496.64
## y ~ 1
##
##         Df Sum of Sq    RSS    AIC
## + x3    1     95705 132571 466.03
## + x2    1     59604 168672 480.48
## + x1    1     59256 169019 480.61
## + x5    1     41417 186858 486.63
## <none>              228275 496.64
## + x4    1      1308 226967 498.29
##
## Step:  AIC=466.03
## y ~ x3
##
##         Df Sum of Sq    RSS    AIC
## + x2    1     33677  98894 450.45
## + x5    1     24165 108405 455.96
## + x1    1     16826 115745 459.89
## <none>              132571 466.03
## + x4    1      1942 130629 467.15
##
## Step:  AIC=450.45
## y ~ x3 + x2
##
##         Df Sum of Sq   RSS    AIC
## + x5    1   14386.5 84507 443.02
## <none>              98894 450.45
## + x1    1    2715.1 96179 450.78
## + x4    1       0.0 98894 452.45
##
## Step:  AIC=443.02
## y ~ x3 + x2 + x5
##
##         Df Sum of Sq   RSS    AIC
## + x1    1    9338.7 75169 437.99
## + x4    1    4481.0 80026 441.75
## <none>              84507 443.02
##
## Step:  AIC=437.99
## y ~ x3 + x2 + x5 + x1
##
##         Df Sum of Sq   RSS    AIC
## <none>              75169 437.99
## + x4    1    879.66 74289 439.28
```

```
##
## Call:
## lm(formula = y ~ x3 + x2 + x5 + x1, data = data_B15)
##
## Coefficients:
## (Intercept)           x3           x2           x5           x1
##    995.8224       3.0998     -15.5697       0.3263       1.6350
```

The above output shows the best model as per Forward selection. The best model with lowest AIC value (437.99) is:

y ~ x3 + x2 + x5 + x1

```
# Backward Selection
step(lm(y ~ x1 + x2 + x3 + x4 + x5, data=data_B15), scope=list(lower=~1, upper=fit1), direction=
'backward', trace=FALSE)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x5, data = data_B15)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x5
##    995.8224       1.6350     -15.5697       3.0998       0.3263
```

The above output shows the best model as per Backward selection. (Trace=FALSE removes the other models from the output and only display the best model)

```
# stepwise selection

step(lm(y ~ x1 + x2 + x3 + x4 + x5, data=data_B15), scope=list(lower=~1, upper=fit1), direction=
'both', trace=FALSE)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x5, data = data_B15)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x5
##    995.8224       1.6350     -15.5697       3.0998       0.3263
```

Same model is selected as the best model from Forward, Backward and Stepwise selection.

```
# Mallow's Cp criteria for model selection

leaps(x=data_B15[,3:7], y=data_B15[,2], names=names(data_B15[,3:7]), method="Cp")
```

```
## $which
##      x1    x2    x3    x4    x5
## 1 FALSE FALSE  TRUE FALSE FALSE
## 1 FALSE  TRUE FALSE FALSE FALSE
## 1  TRUE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE  TRUE
## 1 FALSE FALSE FALSE  TRUE FALSE
## 2 FALSE  TRUE  TRUE FALSE FALSE
## 2 FALSE FALSE  TRUE FALSE  TRUE
## 2  TRUE FALSE FALSE FALSE  TRUE
## 2  TRUE FALSE  TRUE FALSE FALSE
## 2 FALSE FALSE  TRUE  TRUE FALSE
## 2 FALSE  TRUE FALSE FALSE  TRUE
## 2  TRUE  TRUE FALSE FALSE FALSE
## 2  TRUE FALSE FALSE  TRUE FALSE
## 2 FALSE  TRUE FALSE  TRUE FALSE
## 2 FALSE FALSE FALSE  TRUE  TRUE
## 3  TRUE FALSE  TRUE FALSE  TRUE
## 3 FALSE  TRUE  TRUE FALSE  TRUE
## 3 FALSE FALSE  TRUE  TRUE  TRUE
## 3  TRUE  TRUE  TRUE FALSE FALSE
## 3 FALSE  TRUE  TRUE  TRUE FALSE
## 3  TRUE  TRUE FALSE FALSE  TRUE
## 3  TRUE FALSE  TRUE  TRUE FALSE
## 3  TRUE FALSE FALSE  TRUE  TRUE
## 3  TRUE  TRUE FALSE  TRUE FALSE
## 3 FALSE  TRUE FALSE  TRUE  TRUE
## 4  TRUE  TRUE  TRUE FALSE  TRUE
## 4 FALSE  TRUE  TRUE  TRUE  TRUE
## 4  TRUE FALSE  TRUE  TRUE  TRUE
## 4  TRUE  TRUE  TRUE  TRUE FALSE
## 4  TRUE  TRUE FALSE  TRUE  TRUE
## 5  TRUE  TRUE  TRUE  TRUE  TRUE
##
## $label
## [1] "(Intercept)" "x1"          "x2"          "x3"          "x4"
## [6] "x5"
##
## $size
##  [1] 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 6
##
## $Cp
##  [1]  40.364406  66.605818  66.858367  79.825334 108.980121  17.885066
##  [7]  24.798912  30.116411  30.133808  40.952738  52.143709  53.962126
## [13]  62.457183  68.299767  69.222752   7.641570   9.427643  16.513439
## [19]  17.911469  19.885058  28.479903  31.425077  32.113698  49.768501
## [25]  51.391394   4.639416   8.170417   8.436163  19.246996  30.426618
## [31]   6.000000
```

As per above ouput, the best model is “4 TRUE TRUE TRUE FALSE TRUE” which is the following model: y ~ x1 + x2 + x3 + x5 This is the same model as chosen by all the above selection methods.

```r
# Ajusted R-square criteria for model selection
leaps(x=data_B15[,3:7], y=data_B15[,2], names=names(data_B15[,3:7]), method="adjr2")
```

```
## $which
##      x1    x2    x3    x4    x5
## 1 FALSE FALSE  TRUE FALSE FALSE
## 1 FALSE  TRUE FALSE FALSE FALSE
## 1  TRUE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE  TRUE
## 1 FALSE FALSE FALSE  TRUE FALSE
## 2 FALSE  TRUE  TRUE FALSE FALSE
## 2 FALSE FALSE  TRUE FALSE  TRUE
## 2  TRUE FALSE FALSE FALSE  TRUE
## 2  TRUE FALSE  TRUE FALSE FALSE
## 2 FALSE FALSE  TRUE  TRUE FALSE
## 2 FALSE  TRUE FALSE FALSE  TRUE
## 2  TRUE  TRUE FALSE FALSE FALSE
## 2  TRUE FALSE FALSE  TRUE FALSE
## 2 FALSE  TRUE FALSE  TRUE FALSE
## 2 FALSE FALSE FALSE  TRUE  TRUE
## 3  TRUE FALSE  TRUE FALSE  TRUE
## 3 FALSE  TRUE  TRUE FALSE  TRUE
## 3 FALSE FALSE  TRUE  TRUE  TRUE
## 3  TRUE  TRUE  TRUE FALSE FALSE
## 3 FALSE  TRUE  TRUE  TRUE FALSE
## 3  TRUE  TRUE FALSE FALSE  TRUE
## 3  TRUE FALSE  TRUE  TRUE FALSE
## 3  TRUE FALSE FALSE  TRUE  TRUE
## 3  TRUE  TRUE FALSE  TRUE FALSE
## 3 FALSE  TRUE FALSE  TRUE  TRUE
## 4  TRUE  TRUE  TRUE FALSE  TRUE
## 4 FALSE  TRUE  TRUE  TRUE  TRUE
## 4  TRUE FALSE  TRUE  TRUE  TRUE
## 4  TRUE  TRUE  TRUE  TRUE FALSE
## 4  TRUE  TRUE FALSE  TRUE  TRUE
## 5  TRUE  TRUE  TRUE  TRUE  TRUE
##
## $label
## [1] "(Intercept)" "x1"          "x2"          "x3"          "x4"
## [6] "x5"
##
## $size
##  [1] 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 6
##
## $adjr2
##  [1]  0.40923789  0.24836489  0.24681663  0.16732263 -0.01141083
##  [6]  0.55157709  0.50844814  0.47527730  0.47516877  0.40767971
## [11]  0.33786983  0.32652644  0.27353382  0.23708747  0.23132984
## [16]  0.62130897  0.60996840  0.56497750  0.55610079  0.54356959
## [21]  0.48899707  0.47029684  0.46592448  0.35382623  0.34352175
## [26]  0.64676207  0.62393453  0.62221651  0.55232564  0.48005058
## [31]  0.64443093
```

The model with highest Adjusted R-square value is the best model. Again the highest R-square value (0.64676207) is for model “4 TRUE TRUE TRUE FALSE TRUE” which is actually (y ~ x1 + x2 + x3 + x5) model.

Solution (1.g)

Do all 3 procedures picked the same model? If yes: Should it happen all the time, If NO: Why don’t they pick the same?

Yes, all 3 procedures picked the same model - MORT (y) ~ PRECIP (x1) + EDUC(x2) + NONWHITE(x3) + SO2(x5)

But it does not happen all the time. These 3 methods can pick different models based on various criteria. In this particular case, NOX is a weak parameter and its P-value is lesser at all steps in all three methods. In case of another dataset, the P-value of a variable may vary at different steps and can result in different model selection.

solution (1.h)

Perform the residual analysis of your final model and provide the final estimated model

```
fit2 <- lm(y ~ x1 + x2 + x3 + x5, data=data_B15)

#e. Residual analysis
e=residuals(fit2)           ## RESIDUAL
std_e=stdres(fit2)  ## STANDARDIZED RESIDUAL
std_e
```

```
##             1             2             3             4             5 
## -1.3598827028 -1.4759548626 -0.2631211982 -2.6804192907 -0.2362788337 
##             6             7             8             9            10 
## -1.5354005019 -2.6872066609 -0.8883417599 -0.6790199876  0.5066891711 
##            11            12            13            14            15 
## -0.6168068923 -0.6995926630 -0.3668076236 -0.0283660424 -0.1857097341 
##            16            17            18            19            20 
##  0.3345870951 -0.6273781409 -0.2651710415  0.1679042739 -1.2715134876 
##            21            22            23            24            25 
##  0.0001167551 -0.0834399193  0.3649333526 -0.0507272184 -0.0488352698 
##            26            27            28            29            30 
## -0.0464615290  0.7504235322  0.4216126707  0.5217288726  0.5653351100 
##            31            32            33            34            35 
## -0.1316212571  1.3881670282 -0.1798955073 -0.6364588804  0.3850702058 
##            36            37            38            39            40 
##  1.1704760104 -0.2379237115 -1.1547469007  0.9661580479 -0.5039459305 
##            41            42            43            44            45 
##  0.7528619474 -0.5548814528 -0.0968002716  1.5201696880 -0.2650636596 
##            46            47            48            49            50 
##  1.0737264503 -0.1464990085 -0.2868016907  0.7307738731  2.5441693843 
##            51            52            53            54            55 
##  2.0214776153  1.6258751359 -1.0042745403  0.2028608029  0.2439231350 
##            56            57            58            59            60 
##  0.1109301454  0.6967063749 -1.0739814690  0.3446762675  2.4206296501
```

From above, none of the standardized residuals are greater than 3 therefore none of them are outliers.

```
r=studres(fit2)             ## STUDENTIZED RESIDUAL
max(abs(r))
```

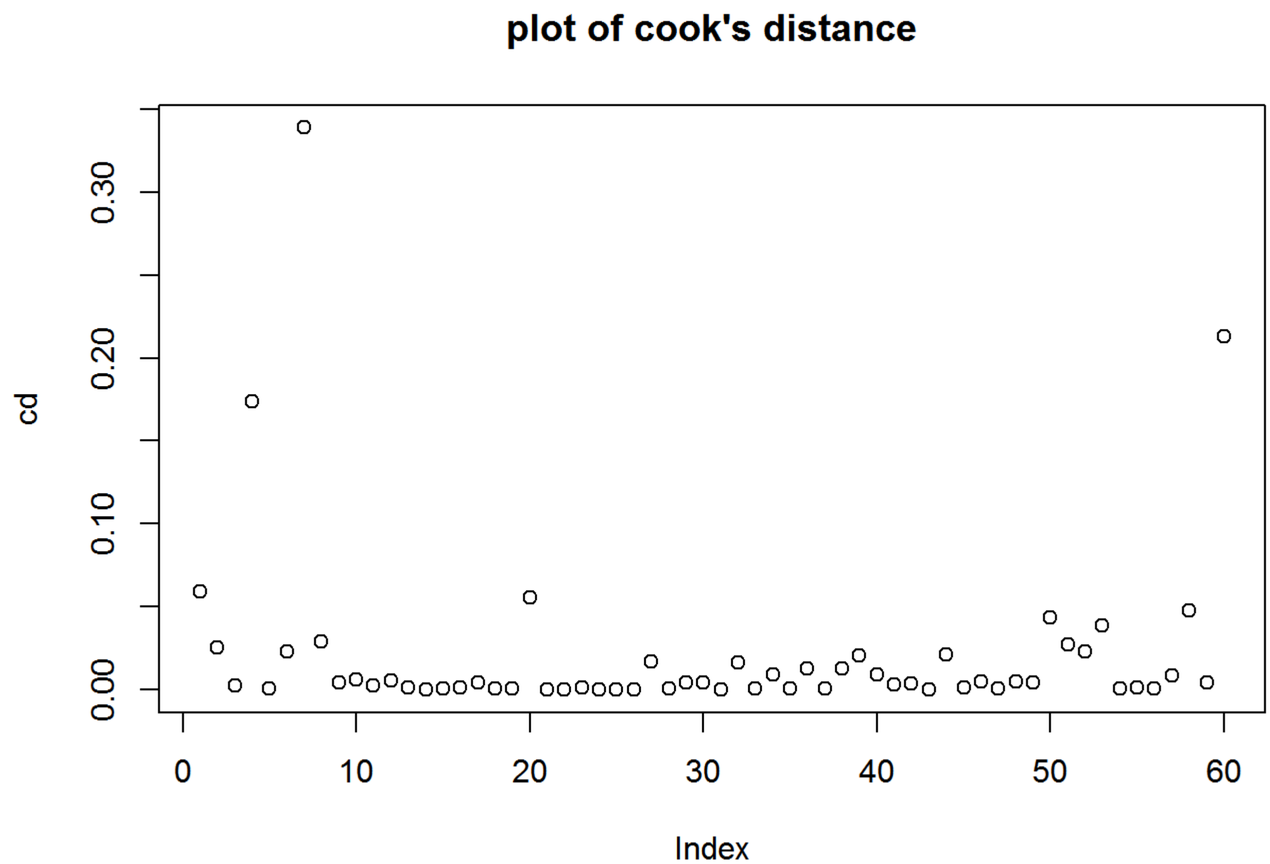
```
## [1] 2.8568
```

The studentized residuals also don't show any outlier. However the 50th and 60th point have values greater than 2.5 and hence these points need closer analysis.

```
# Influence analysis

## COOKS DISTANCE
cd=cooks.distance(fit2)
cd
```

```
##            1            2            3            4            5 
## 5.920384e-02 2.566065e-02 2.748780e-03 1.735884e-01 7.104871e-04 
##            6            7            8            9           10 
## 2.293407e-02 3.388075e-01 2.896764e-02 4.342522e-03 6.385758e-03 
##           11           12           13           14           15 
## 2.665218e-03 5.301815e-03 1.155392e-03 1.213187e-05 5.964713e-04 
##           16           17           18           19           20 
## 1.537115e-03 4.445000e-03 9.679526e-04 6.784888e-04 5.567658e-02 
##           21           22           23           24           25 
## 1.837655e-10 4.696907e-05 1.381437e-03 1.303433e-05 2.715753e-05 
##           26           27           28           29           30 
## 2.400820e-05 1.678024e-02 9.548235e-04 4.317069e-03 4.089604e-03 
##           31           32           33           34           35 
## 3.716435e-04 1.621505e-02 4.469483e-04 8.889603e-03 6.311853e-04 
##           36           37           38           39           40 
## 1.275908e-02 5.785972e-04 1.299847e-02 2.032373e-02 9.331727e-03 
##           41           42           43           44           45 
## 3.273842e-03 3.942597e-03 1.462202e-04 2.142370e-02 1.242379e-03 
##           46           47           48           49           50 
## 4.992725e-03 6.346837e-04 4.970891e-03 4.378052e-03 4.329587e-02 
##           51           52           53           54           55 
## 2.729008e-02 2.331134e-02 3.855466e-02 6.497139e-04 1.160931e-03 
##           56           57           58           59           60 
## 7.844472e-04 8.622769e-03 4.793206e-02 4.095782e-03 2.132721e-01
```

```
plot(cd,main="plot of cook's distance")
```

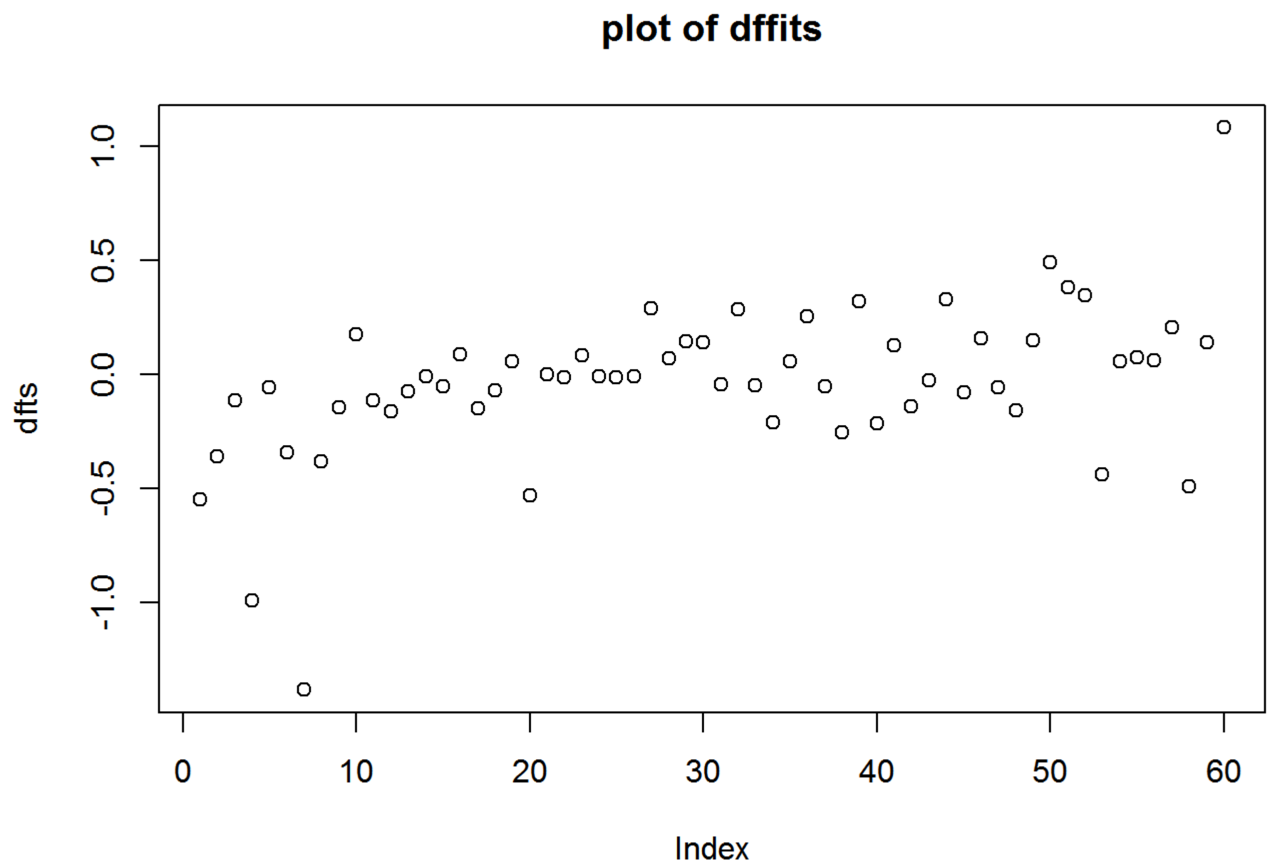


There is no point which has Cook's distance greater than 2. Therefore, as per cook's distance, no point unduly impacts the regression coefficients.

```
## DFFITS
dfts=dffits(fit2)
dfts
```

```
##             1             2             3             4             5 
## -5.484061e-01 -3.621680e-01 -1.162369e-01 -9.900532e-01 -5.908798e-02 
##             6             7             8             9            10 
## -3.429684e-01 -1.383695e+00 -3.798353e-01 -1.466222e-01  1.774691e-01 
##            11            12            13            14            15 
## -1.147821e-01 -1.620518e-01 -7.540447e-02 -7.717339e-03 -5.412919e-02 
##            16            17            18            19            20 
##  8.695531e-02 -1.482504e-01 -6.897717e-02  5.772756e-02 -5.306591e-01 
##            21            22            23            24            25 
##  3.003534e-05 -1.518567e-02  8.245036e-02 -7.999355e-03 -1.154663e-02 
##            26            27            28            29            30 
## -1.085647e-02  2.884924e-01  6.857488e-02  1.459393e-01  1.421041e-01 
##            31            32            33            34            35 
## -4.272009e-02  2.872131e-01 -4.685511e-02 -2.096751e-01  5.573977e-02 
##            36            37            38            39            40 
##  2.534473e-01 -5.332272e-02 -2.557268e-01  3.185806e-01 -2.145293e-01 
##            41            42            43            44            45 
##  1.274321e-01 -1.395117e-01 -2.679423e-02  3.313361e-01 -7.814571e-02 
##            46            47            48            49            50 
##  1.582230e-01 -5.582953e-02 -1.563301e-01  1.473193e-01  4.908090e-01 
##            51            52            53            54            55 
##  3.804235e-01  3.467209e-01 -4.390944e-01  5.649684e-02  7.553336e-02 
##            56            57            58            59            60 
##  6.206274e-02  2.066564e-01 -4.902481e-01  1.419510e-01  1.082503e+00
```

```
plot(dfts,main="plot of dffits")
```




```
#threshold value
# 2*square-root(p/n) = 0.52  (p = 4, n = 60)
```

The 50th and 60th point have a dffits value greater than the cutoff value of 0.52. Therefore according to this analysis, these 2 data points unduly impact the parameter estimation.

```
# removing data points - 50th and 60th and see if there is improvement in model fit

data1 <- data_B15[-c(1,6)]

data2 <- data1[-50,]
data3 <- data2[-59,]

fit_subset <- lm(y ~ ., data = data3)

summary(fit2)
```

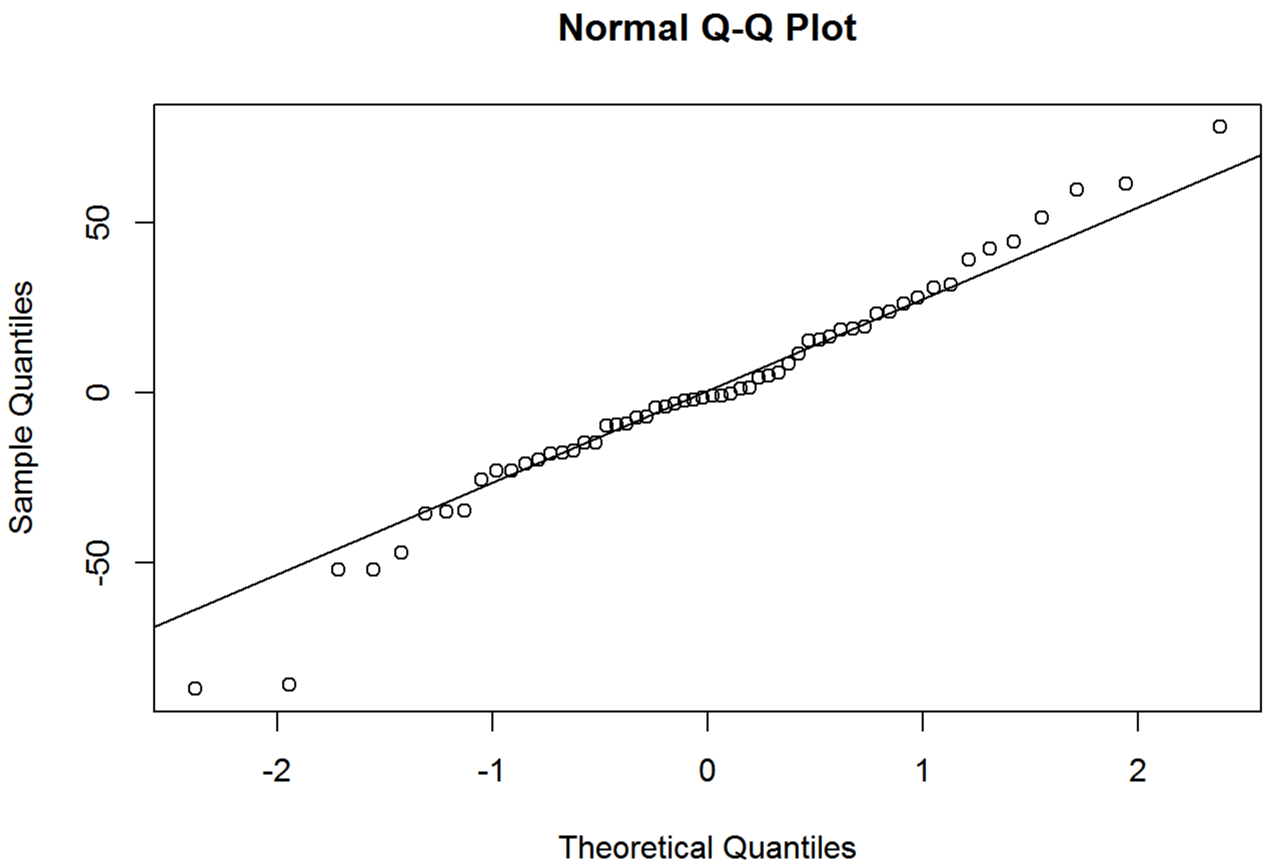
```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x5, data = data_B15)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -93.600 -20.499  -2.443  17.891  92.521
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 995.82238   91.33980  10.902 2.31e-15 ***
## x1            1.63505    0.62550   2.614 0.011522 *
## x2          -15.56968    6.93862  -2.244 0.028883 *
## x3            3.09979    0.60779   5.100 4.33e-06 ***
## x5            0.32634    0.08323   3.921 0.000247 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.97 on 55 degrees of freedom
## Multiple R-squared:  0.6707, Adjusted R-squared:  0.6468
## F-statistic: 28.01 on 4 and 55 DF,  p-value: 1.052e-12
```

```
summary(fit_subset)
```

```
##
## Call:
## lm(formula = y ~ ., data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -86.926 -17.467  -1.177  18.878  78.041
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 958.70795   82.93197  11.560 4.16e-16 ***
## x1            1.63894    0.56235   2.914  0.00521 **
## x2          -12.45716    6.32690  -1.969  0.05420 .
## x3            2.87892    0.57029   5.048 5.60e-06 ***
## x5            0.36838    0.07628   4.829 1.21e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.23 on 53 degrees of freedom
## Multiple R-squared:  0.6986, Adjusted R-squared:  0.6759
## F-statistic: 30.71 on 4 and 53 DF,  p-value: 3.074e-13
```

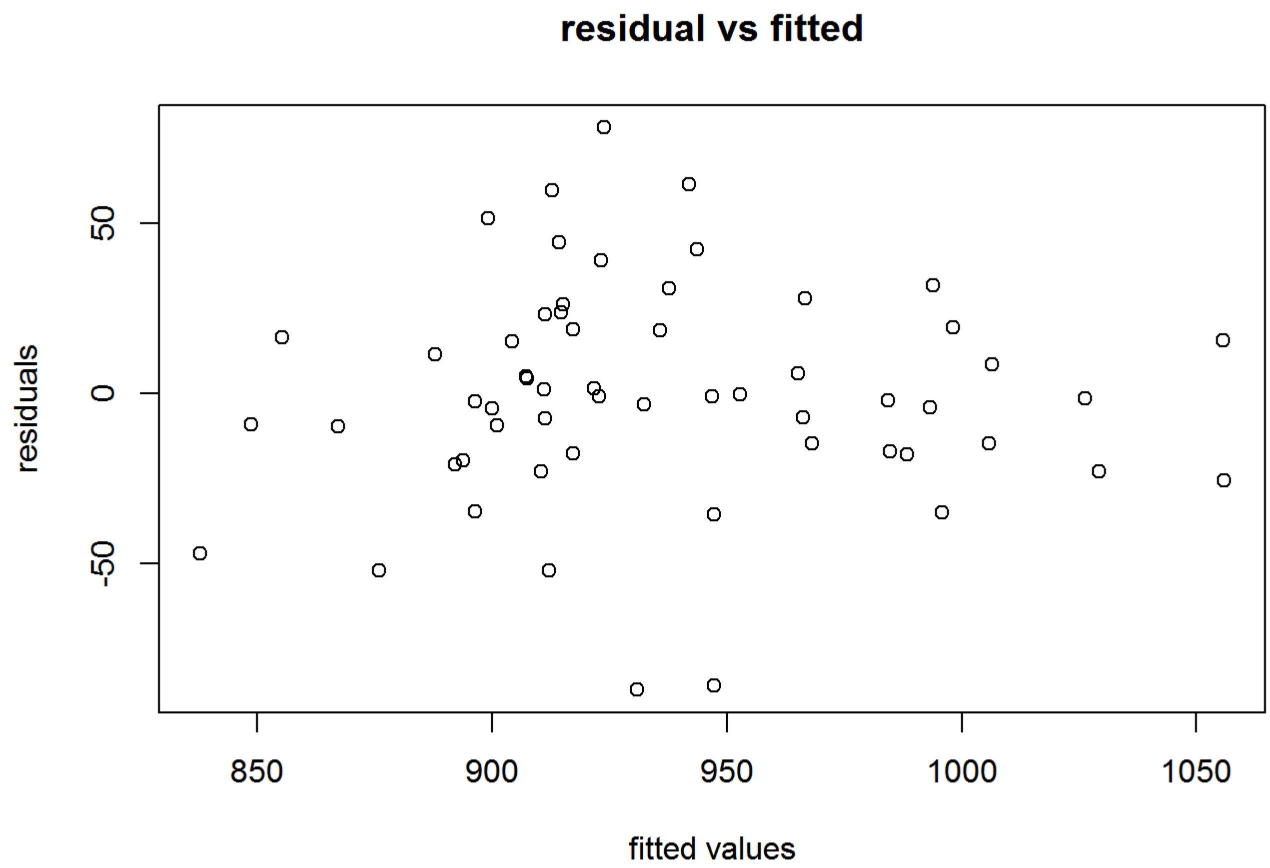
Removal of 50th and 60th data point has resulted in decrease in Residual standard error and increase in Adjusted R-square. Hence these data points should be excluded to attain better model fitting.

```
# Normality plot of residuals

qqnorm(fit_subset$residuals)
qqline(fit_subset$residuals)
```



The normal probability plot shows that the residuals are much more closer to a normal distribution and there is no departure from the straight line towards the tails. This indicates that errors are IID normal.

```
# residual vs fitted plot

plot(y=fit_subset$residuals,x=fit_subset$fit,xlab="fitted values",ylab="residuals",main="residua
l vs fitted")
```



There does not appear to be any strong pattern to the residuals, and they all appear to lie randomly in a horizontal band, so this plot supports the assumption that the errors are independently, identically distributed. The plot does not indicate any outliers.