Lab 9: Detection of Multicollinearity in R

Jeevan Koshy ~ 1740256

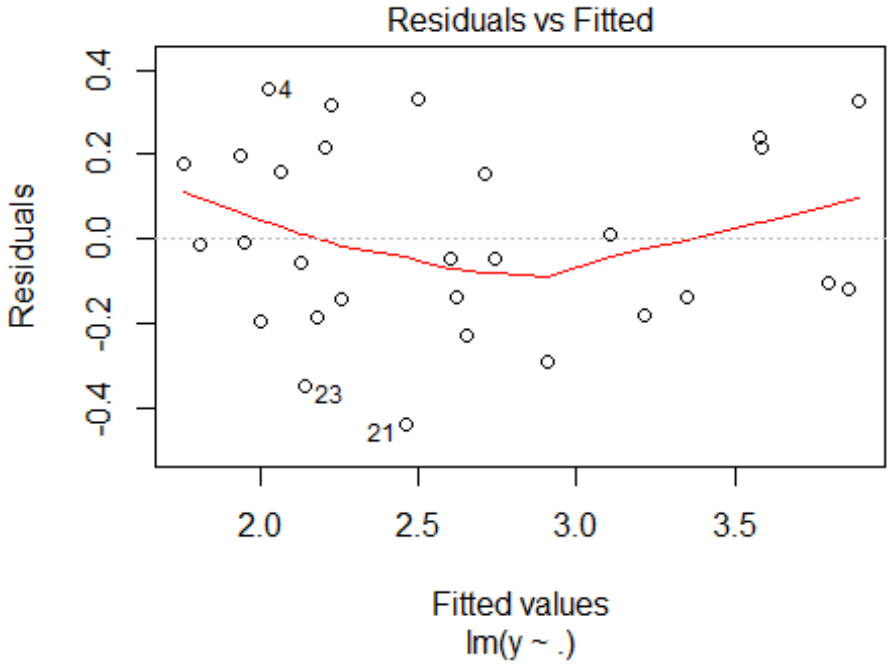
September 18, 2019

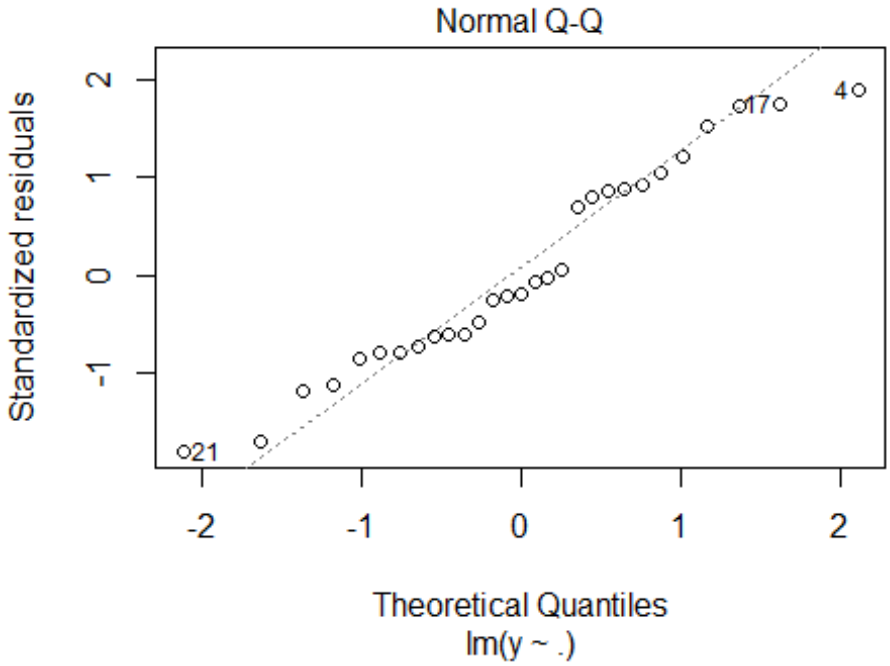
Aim

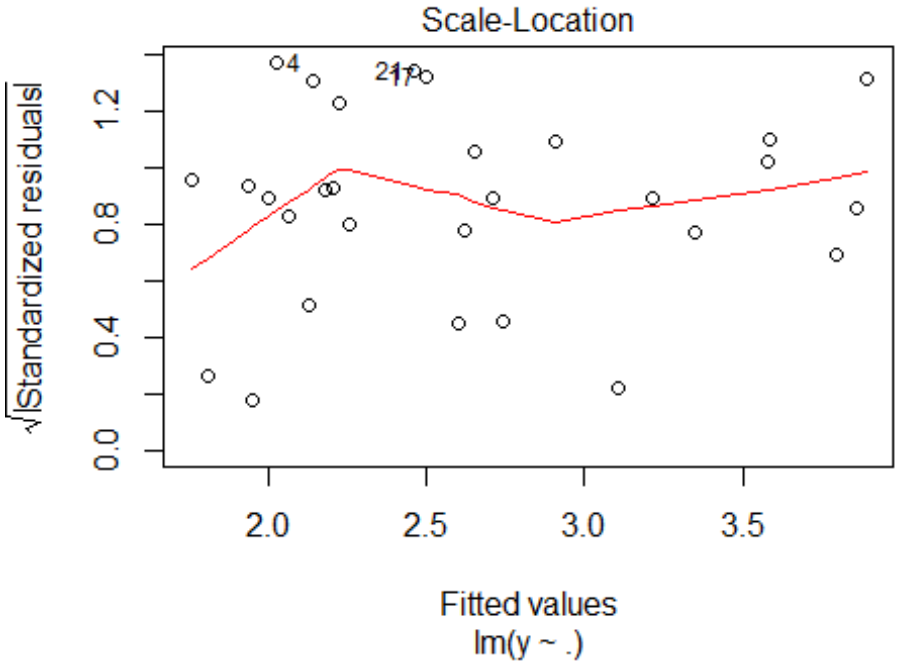
To check the multicollinearity of the given data and analyze the data using R package.

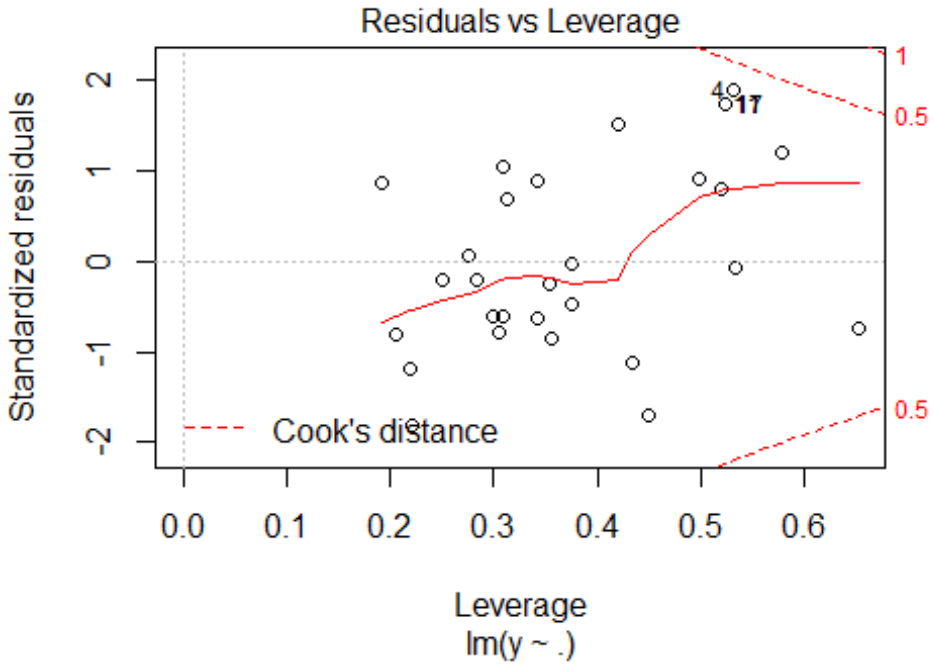
Procedure

```
#install.packages("GGally")
# install.packages("mctest")
library(GGally)

## Warning: package 'GGally' was built under R version 3.5.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.5.2

library(readxl)

## Warning: package 'readxl' was built under R version 3.5.2

library(mctest)

## Warning: package 'mctest' was built under R version 3.5.2

multicollinearitydata <- read_excel("C:/Users/Jeevan/Desktop/Christ Universit
y/Statistics/Linear Regression/multicollinearitydata.xlsx")
# View(multicollinearitydata)
attach(multicollinearitydata)
fit = lm(y~.,data = multicollinearitydata)
formula(fit)

## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

plot(fit)
```

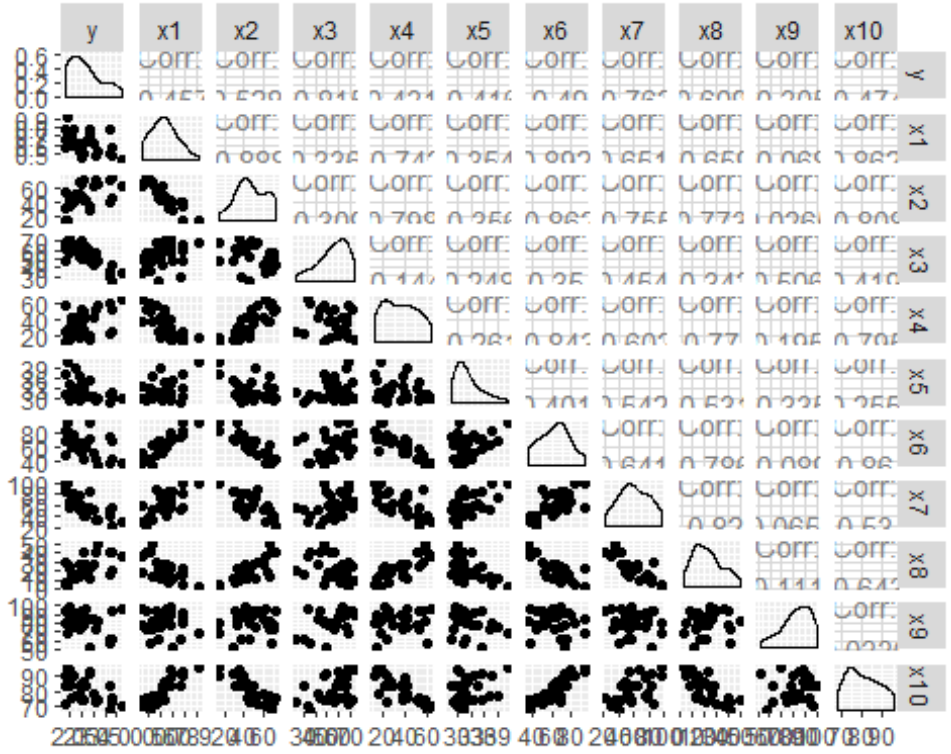

Residuals vs Fitted
Residuals
Fitted values
lm(y ~ .)
4
23
21
-0.4
-0.2
0.0
0.2
0.4
2.0
2.5
3.0
3.5


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(y ~ .)
21
17
4
-2
-1
0
1
2


Scale-Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
2.0
2.5
3.0
3.5
Fitted values
lm(y ~ .)


Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
0.5
0.5
Cook's distance
Leverage
lm(y ~ .)

```
ggpairs(multicollinearitydata)
```




```
x = multicollinearitydata[,2:10]
x
```

```
## # A tibble: 29 x 9
##       x1    x2    x3    x4    x5    x6    x7    x8    x9
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 0.789  39.8  66.9  23.4  33.4  77.3  79.2  15.3  92.1
##  2 0.644  41.7  63.4  41.4  30.4  60.4  42.1  27.7  95.6
##  3 0.681  36.1  72.6  14.4  29.9  79.5  66    10.1  88.4
##  4 0.601  44.7  52.6  16.1  32    53.9  77.2  14.4  80.8
##  5 0.679  41.7  63.3  21.6  29.7  68.7  56.1  25.3  99.5
##  6 0.537  65.3  47.2  58.4  30.2  36.2  33.9  49.2  81.8
##  7 0.628  41.9  59.3  26    32.4  64.6  47.9  18.8  87.4
##  8 0.516  70.8  53.2  50.5  33    44.9  41.8  39.9  77.9
##  9 0.488  69.5  55.9  52.6  29.2  44.4  32.6  40.7  74.2
## 10 0.49   72.7  43.6  52.2  29.8  44.8  22.5  40.5  93.7
## # ... with 19 more rows
```

```
omcdiag(x,multicollinearitydata$y,data = multicollinearitydata)
```

```
## Warning in omcdiag(x, multicollinearitydata$y, data =
## multicollinearitydata): Extra argument 'data' is ignored
```

```
##
## Call:
## omcdiag(x = x, y = multicollinearitydata$y, data = multicollinearitydata)
```

```
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.0001         1
## Farrar Chi-Square:       230.3103         1
## Red Indicator:             0.5562         1
## Sum of Lambda Inverse:    52.4495         1
## Theil's Method:           -0.3339         0
## Condition Number:        101.8065         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

```
imcdiag(x,multicollinearitydata$y)
```

```
##
## Call:
## imcdiag(x = x, y = multicollinearitydata$y)
##
##
## All Individual Multicollinearity Diagnostics Result
##
##        VIF    TOL      Wi      Fi Leamer    CVIF Klein
## x1  8.5152 0.1174 18.7879 22.5455 0.3427 -0.4908     0
## x2  8.7085 0.1148 19.2714 23.1256 0.3389 -0.5019     0
## x3  2.5346 0.3945  3.8364  4.6037 0.6281 -0.1461     0
## x4  5.2636 0.1900 10.6589 12.7907 0.4359 -0.3034     0
## x5  2.1149 0.4728  2.7872  3.3446 0.6876 -0.1219     0
## x6 11.0201 0.0907 25.0503 30.0603 0.3012 -0.6352     1
## x7  5.2569 0.1902 10.6424 12.7708 0.4361 -0.3030     0
## x8  6.7809 0.1475 14.4524 17.3428 0.3840 -0.3908     0
## x9  2.2548 0.4435  3.1369  3.7643 0.6660 -0.1300     0
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## x1 , x2 , x5 , x6 , x8 , x9 , coefficient(s) are non-significant may be du
e to multicollinearity
##
## R-square of y on all x: 0.8965
##
## * use method argument to check which regressors may be the reason of colli
nearity
## ===================================
```

```
summary(fit)
```

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

```
## lm(formula = y ~ ., data = multicollinearitydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43837 -0.14250 -0.04833  0.19676  0.35463
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.686737   1.970421   2.379 0.028660 *
## x1           0.613751   1.488450   0.412 0.684958
## x2          -0.001258   0.009745  -0.129 0.898732
## x3          -0.033710   0.007148  -4.716 0.000172 ***
## x4           0.019574   0.008449   2.317 0.032514 *
## x5          -0.024824   0.024090  -1.030 0.316427
## x6           0.005273   0.011015   0.479 0.637931
## x7          -0.017476   0.006283  -2.782 0.012310 *
## x8          -0.007767   0.011157  -0.696 0.495198
## x9          -0.006798   0.006318  -1.076 0.296125
## x10          0.012862   0.018141   0.709 0.487409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2751 on 18 degrees of freedom
## Multiple R-squared:  0.8994, Adjusted R-squared:  0.8434
## F-statistic: 16.08 on 10 and 18 DF,  p-value: 5.185e-07
```

```
new = multicollinearitydata[,-7]
new
```

```
## # A tibble: 29 x 10
##        y    x1    x2    x3    x4    x5    x7    x8    x9   x10
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  2.13 0.789  39.8  66.9  23.4  33.4  79.2  15.3  92.1  90.2
##  2  2.69 0.644  41.7  63.4  41.4  30.4  42.1  27.7  95.6  83.4
##  3  1.94 0.681  36.1  72.6  14.4  29.9  66    10.1  88.4  94
##  4  2.38 0.601  44.7  52.6  16.1  32    77.2  14.4  80.8  78.1
##  5  1.99 0.679  41.7  63.3  21.6  29.7  56.1  25.3  99.5  82.9
##  6  3.21 0.537  65.3  47.2  58.4  30.2  33.9  49.2  81.8  73.9
##  7  2.55 0.628  41.9  59.3  26    32.4  47.9  18.8  87.4  86.3
##  8  2.62 0.516  70.8  53.2  50.5  33    41.8  39.9  77.9  74.1
##  9  3.12 0.488  69.5  55.9  52.6  29.2  32.6  40.7  74.2  73.5
## 10  3.82 0.49   72.7  43.6  52.2  29.8  22.5  40.5  93.7  76.2
## # ... with 19 more rows
```

```
fit1 = lm(y~.,data = new)
formula(fit1)
```

```
## y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8 + x9 + x10
```

```
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ ., data = new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42439 -0.16256 -0.06006  0.22185  0.32720
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.876626   1.890518   2.580 0.018370 *
## x1           0.989738   1.238412   0.799 0.434053
## x2          -0.001715   0.009499  -0.181 0.858656
## x3          -0.032935   0.006820  -4.829 0.000116 ***
## x4           0.018210   0.007792   2.337 0.030529 *
## x5          -0.023376   0.023409  -0.999 0.330540
## x7          -0.018585   0.005720  -3.249 0.004225 **
## x8          -0.009947   0.009976  -0.997 0.331253
## x9          -0.006975   0.006178  -1.129 0.272942
## x10          0.013037   0.017765   0.734 0.471996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2695 on 19 degrees of freedom
## Multiple R-squared:  0.8981, Adjusted R-squared:  0.8498
## F-statistic:  18.6 on 9 and 19 DF,  p-value: 1.277e-07

xx = new[,2:9]
xx

## # A tibble: 29 x 8
##       x1    x2    x3    x4    x5    x7    x8    x9
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 0.789  39.8  66.9  23.4  33.4  79.2  15.3  92.1
##  2 0.644  41.7  63.4  41.4  30.4  42.1  27.7  95.6
##  3 0.681  36.1  72.6  14.4  29.9  66    10.1  88.4
##  4 0.601  44.7  52.6  16.1  32    77.2  14.4  80.8
##  5 0.679  41.7  63.3  21.6  29.7  56.1  25.3  99.5
##  6 0.537  65.3  47.2  58.4  30.2  33.9  49.2  81.8
##  7 0.628  41.9  59.3  26    32.4  47.9  18.8  87.4
##  8 0.516  70.8  53.2  50.5  33    41.8  39.9  77.9
##  9 0.488  69.5  55.9  52.6  29.2  32.6  40.7  74.2
## 10 0.49   72.7  43.6  52.2  29.8  22.5  40.5  93.7
## # ... with 19 more rows

imcdiag(xx,new$y)

##
## Call:
## imcdiag(x = xx, y = new$y)
##
```

```
##
## All Individual Multicollinearity Diagnostics Result
##
##       VIF    TOL      Wi      Fi Leamer    CVIF Klein
## x1 5.5402 0.1805 13.6207 16.6476 0.4248 -0.3734     0
## x2 8.6189 0.1160 22.8566 27.9359 0.3406 -0.5810     0
## x3 2.3634 0.4231  4.0903  4.9992 0.6505 -0.1593     0
## x4 4.4741 0.2235 10.4224 12.7385 0.4728 -0.3016     0
## x5 2.0816 0.4804  3.2448  3.9658 0.6931 -0.1403     0
## x7 4.4087 0.2268 10.2260 12.4984 0.4763 -0.2972     0
## x8 5.6462 0.1771 13.9386 17.0361 0.4208 -0.3806     0
## x9 2.2470 0.4450  3.7411  4.5725 0.6671 -0.1515     0
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## x1 , x2 , x5 , x8 , x9 , coefficient(s) are non-significant may be due to
multicollinearity
##
## R-square of y on all x: 0.8952
##
## * use method argument to check which regressors may be the reason of colli
nearity
## ===================================
```

```
step(fit1,direction = "both")
```

```
## Start:  AIC=-68.31
## y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS     AIC
## - x2    1   0.00237 1.3824 -70.261
## - x10   1   0.03912 1.4191 -69.500
## - x1    1   0.04639 1.4264 -69.352
## - x8    1   0.07221 1.4522 -68.831
## - x5    1   0.07243 1.4525 -68.827
## - x9    1   0.09259 1.4726 -68.427
## <none>              1.3800 -68.310
## - x4    1   0.39675 1.7768 -62.982
## - x7    1   0.76662 2.1467 -57.498
## - x3    1   1.69408 3.0741 -47.084
##
## Step:  AIC=-70.26
## y ~ x1 + x3 + x4 + x5 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS     AIC
## - x10   1   0.04245 1.4248 -71.384
## - x5    1   0.07323 1.4556 -70.764
## - x8    1   0.07709 1.4595 -70.687
## - x1    1   0.08164 1.4640 -70.597
```

```
## - x9    1   0.09143 1.4738 -70.403
## <none>              1.3824 -70.261
## + x2    1   0.00237 1.3800 -68.310
## - x4    1   0.39712 1.7795 -64.938
## - x7    1   0.86042 2.2428 -58.227
## - x3    1   1.87671 3.2591 -47.389
##
## Step:  AIC=-71.38
## y ~ x1 + x3 + x4 + x5 + x7 + x8 + x9
##
##        Df Sum of Sq    RSS     AIC
## - x5    1   0.07429 1.4991 -71.910
## - x8    1   0.08457 1.5094 -71.712
## - x9    1   0.08776 1.5126 -71.650
## <none>              1.4248 -71.384
## + x10   1   0.04245 1.3824 -70.261
## + x2    1   0.00570 1.4192 -69.500
## - x1    1   0.35012 1.7750 -67.012
## - x4    1   0.37385 1.7987 -66.627
## - x7    1   1.16522 2.5901 -56.053
## - x3    1   2.00859 3.4334 -47.878
##
## Step:  AIC=-71.91
## y ~ x1 + x3 + x4 + x7 + x8 + x9
##
##        Df Sum of Sq    RSS     AIC
## - x9    1   0.03540 1.5345 -73.233
## - x8    1   0.04403 1.5432 -73.070
## <none>              1.4991 -71.910
## + x5    1   0.07429 1.4248 -71.384
## + x10   1   0.04351 1.4556 -70.764
## + x2    1   0.00696 1.4922 -70.045
## - x4    1   0.30425 1.8034 -68.551
## - x1    1   0.31815 1.8173 -68.329
## - x7    1   1.23626 2.7354 -56.469
## - x3    1   2.36467 3.8638 -46.454
##
## Step:  AIC=-73.23
## y ~ x1 + x3 + x4 + x7 + x8
##
##        Df Sum of Sq    RSS     AIC
## - x8    1    0.0457 1.5802 -74.382
## <none>              1.5345 -73.233
## + x10   1    0.0404 1.4942 -72.006
## + x9    1    0.0354 1.4991 -71.910
## + x5    1    0.0219 1.5126 -71.650
## + x2    1    0.0004 1.5342 -71.240
## - x4    1    0.2951 1.8297 -70.132
## - x1    1    0.3247 1.8593 -69.666
## - x7    1    1.2009 2.7354 -58.469
```

```
## - x3    1    4.1080 5.6425 -37.472
##
## Step:  AIC=-74.38
## y ~ x1 + x3 + x4 + x7
##
##        Df Sum of Sq    RSS     AIC
## <none>              1.5802 -74.382
## + x8    1    0.0457 1.5345 -73.233
## + x10   1    0.0456 1.5346 -73.231
## + x9    1    0.0371 1.5432 -73.070
## + x5    1    0.0054 1.5749 -72.481
## + x2    1    0.0021 1.5782 -72.420
## - x4    1    0.2615 1.8417 -71.941
## - x1    1    0.3024 1.8826 -71.305
## - x7    1    1.5685 3.1487 -56.389
## - x3    1    4.0640 5.6442 -39.463

##
## Call:
## lm(formula = y ~ x1 + x3 + x4 + x7, data = new)
##
## Coefficients:
## (Intercept)           x1           x3           x4           x7
##    4.270727     1.524122    -0.034690     0.009727    -0.017116
```

```
be = lm(y~x1+x3+x4+x7,data = new)
summary(be)
```

```
##
## Call:
## lm(formula = y ~ x1 + x3 + x4 + x7, data = new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48391 -0.17071  0.01805  0.18276  0.45733
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.270727   0.542943   7.866 4.25e-08 ***
## x1           1.524122   0.711229   2.143   0.0425 *
## x3          -0.034690   0.004416  -7.856 4.34e-08 ***
## x4           0.009727   0.004881   1.993   0.0578 .
## x7          -0.017116   0.003507  -4.881 5.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2566 on 24 degrees of freedom
## Multiple R-squared:  0.8833, Adjusted R-squared:  0.8638
## F-statistic: 45.41 on 4 and 24 DF,  p-value: 7.411e-11
```

Conclusion

From the analysis performed above, it can be seen that there exists multicollinearity in the given dataset as indicated by the following methods:

	MC Results	detection
1. Determinant |X'X|:	0.0001	1
2. Farrar Chi-Square:	230.3103	1
3. Red Indicator:	0.5562	1
4. Sum of Lambda Inverse:	52.4495	1
5. Condition Number:	101.8065	1

On diagnosing for Individual Multicollinearity, it is observed that *x6*, i.e., the percentage of female literacy is the cause for multicollinearity in the dataset. On removing *x6* from the dataset it can be seen that there exists no multicollinearity in the data.

Using Stepwise Elimination Method, it can be seen that *x4*, i.e., the female age at marriage is the best variable($p>0.05$) that can be selected from the data since it has almost no effect on the dataset.

---\---