multiple_linear_regression

Vinoth Aryan Nagabosshanam

March 22, 2017

simple multiple linear regression problem

```
cars_data<-read.csv("B:\\data science courses\\Datasets_BA 2\\Cars.csv")
head(cars_data)
```

```
##   HP      MPG VOL       SP       WT
## 1 49 53.70068  89 104.1854 28.76206
## 2 55 50.01340  92 105.4613 30.46683
## 3 55 50.01340  92 105.4613 30.19360
## 4 70 45.69632  92 113.4613 30.63211
## 5 53 50.50423  92 104.4613 29.88915
## 6 70 45.69632  89 113.1854 29.59177
```

```
str(cars_data)
```

```
## 'data.frame':    81 obs. of  5 variables:
##  $ HP : int  49 55 55 70 53 70 55 62 62 80 ...
##  $ MPG: num  53.7 50 50 45.7 50.5 ...
##  $ VOL: int  89 92 92 92 92 89 92 50 50 94 ...
##  $ SP : num  104 105 105 113 104 ...
##  $ WT : num  28.8 30.5 30.2 30.6 29.9 ...
```

Exploratory Data Analysis

Measures of Central Tendency to know the maen medium and mode if we summary dataset

we will get all central tendency

```
summary(cars_data)
```

```
##        HP             MPG             VOL               SP        
##  Min.   : 49.0   Min.   :12.10   Min.   : 50.00   Min.   : 99.56  
##  1st Qu.: 84.0   1st Qu.:27.86   1st Qu.: 89.00   1st Qu.:113.83  
##  Median :100.0   Median :35.15   Median :101.00   Median :118.21  
##  Mean   :117.5   Mean   :34.42   Mean   : 98.77   Mean   :121.54  
##  3rd Qu.:140.0   3rd Qu.:39.53   3rd Qu.:113.00   3rd Qu.:126.40  
##  Max.   :322.0   Max.   :53.70   Max.   :160.00   Max.   :169.60  
##        WT       
##  Min.   :15.71  
##  1st Qu.:29.59  
##  Median :32.73  
##  Mean   :32.41  
```

```
##   3rd Qu.:37.39
##   Max.   :53.00
```

Measures of Dispersion means find out the variance and sd of data in data set

```r
sapply(cars_data,var)
```

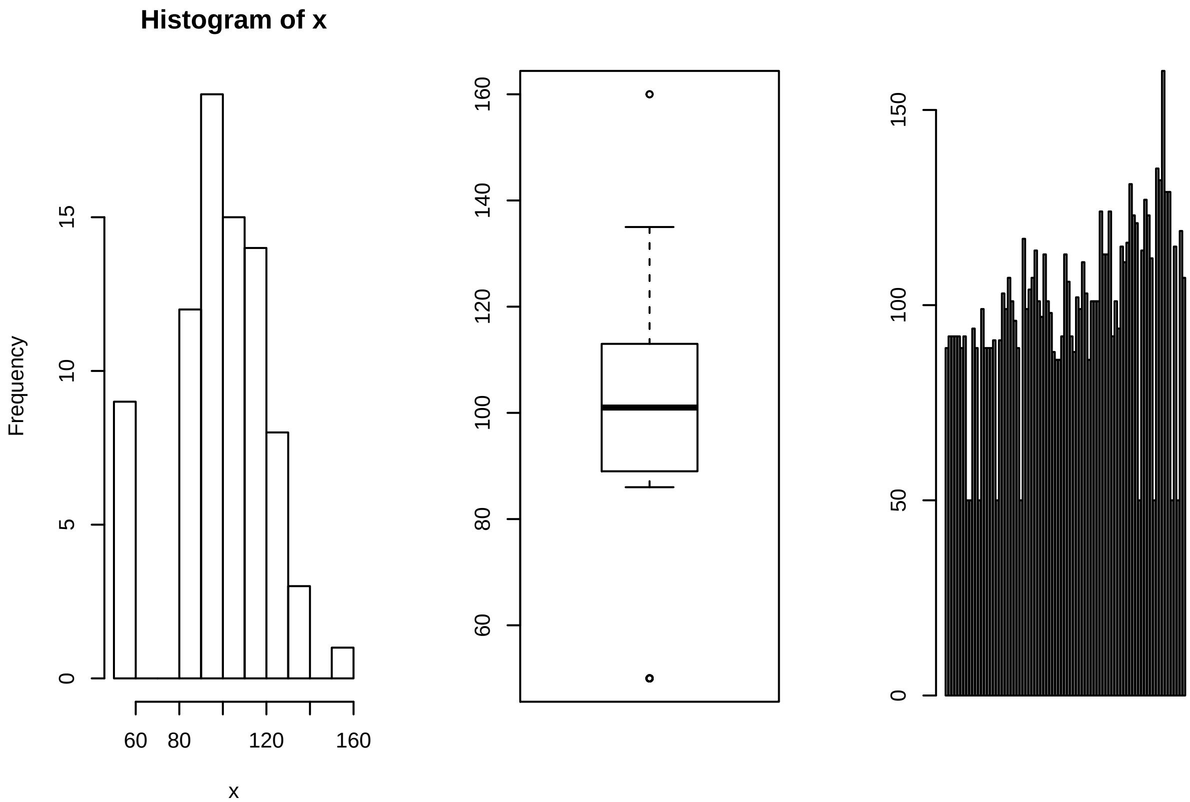
```
##         HP        MPG        VOL         SP         WT
## 3261.95216   83.38328  497.35679  201.11300   56.14225
```

```r
sapply(cars_data,sd)
```

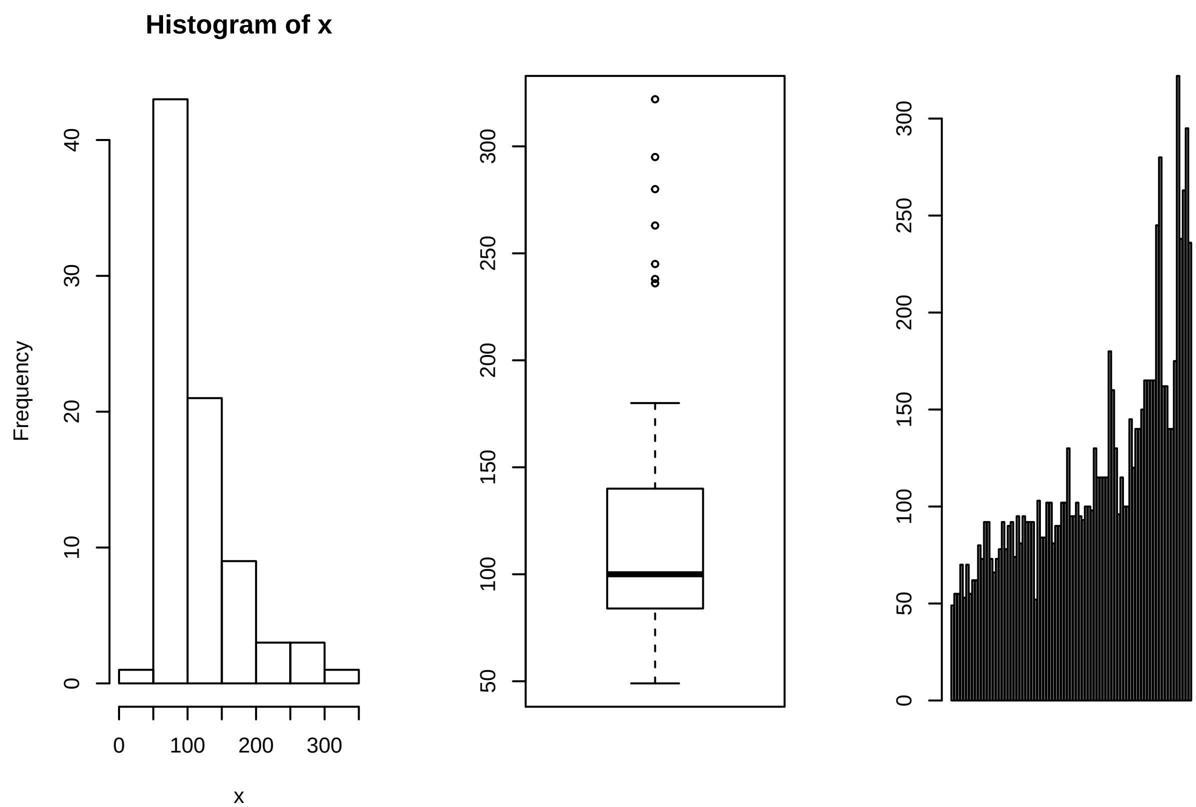
```
##        HP       MPG       VOL        SP        WT
## 57.113502  9.131445 22.301497 14.181432  7.492813
```

Graphical representations

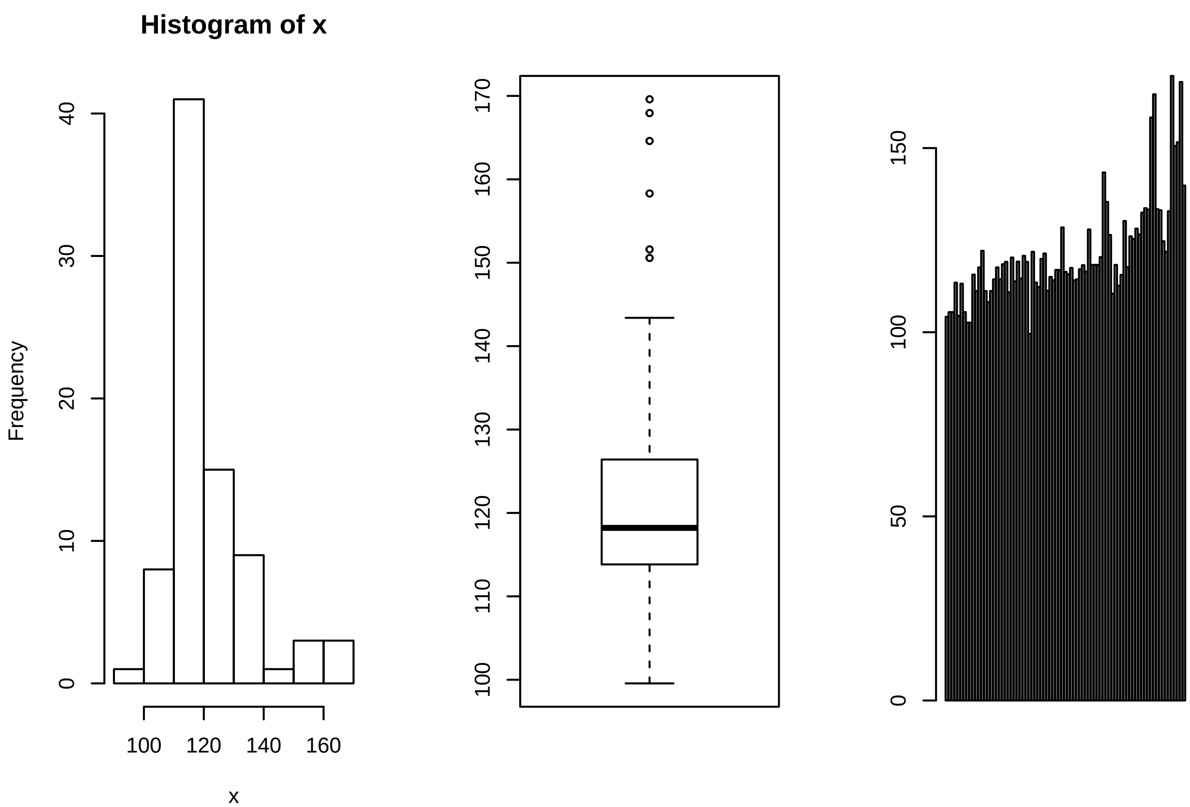
```r
plott<-function(x){
  par(mfrow=c(1,3))
  hist(x)
  boxplot(x)
  barplot(x)

 }

plott(cars_data$VOL)
```



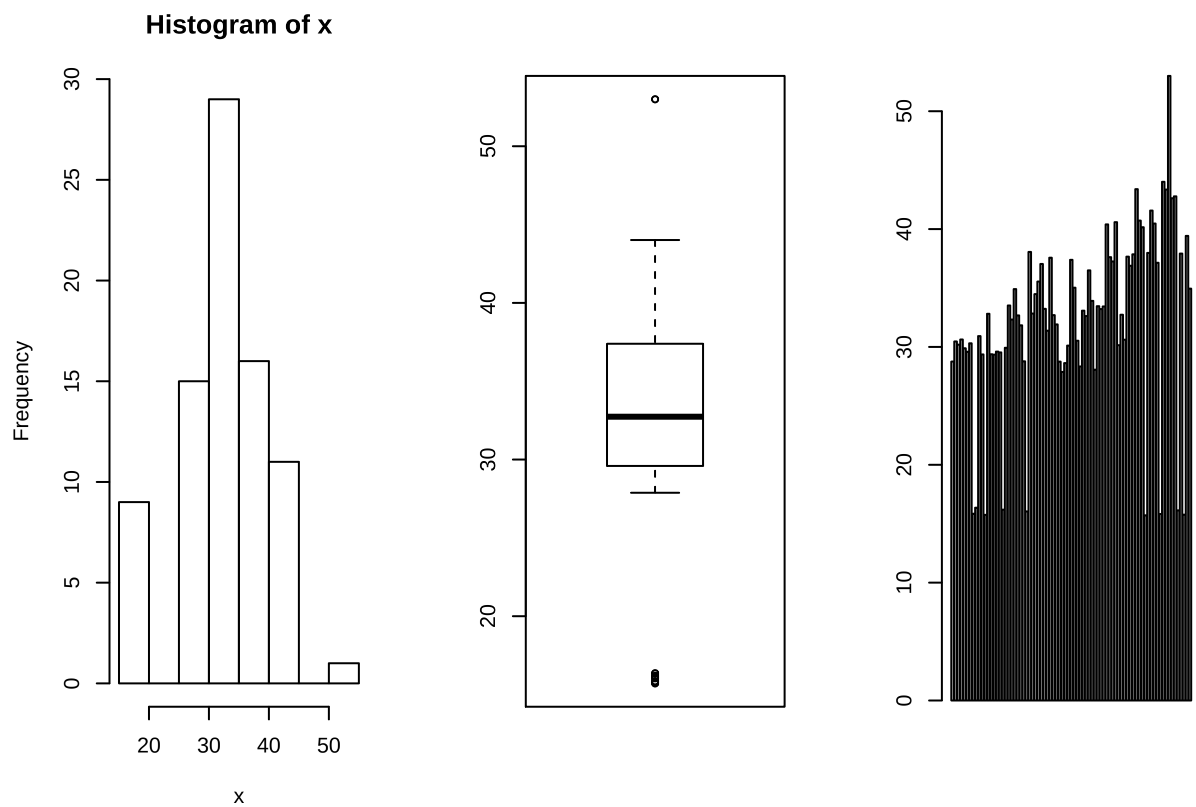

```
plott(cars_data$HP)
```




```r
plott(cars_data$SP)
```




```r
plott(cars_data$WT)
```

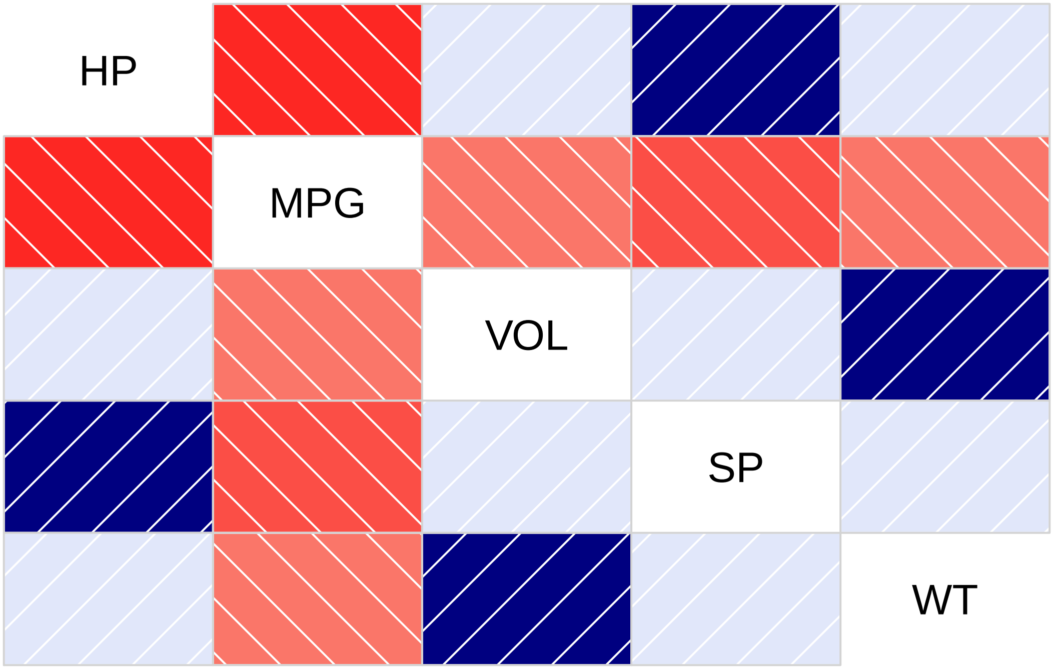


Find the correlation b/n Output and input

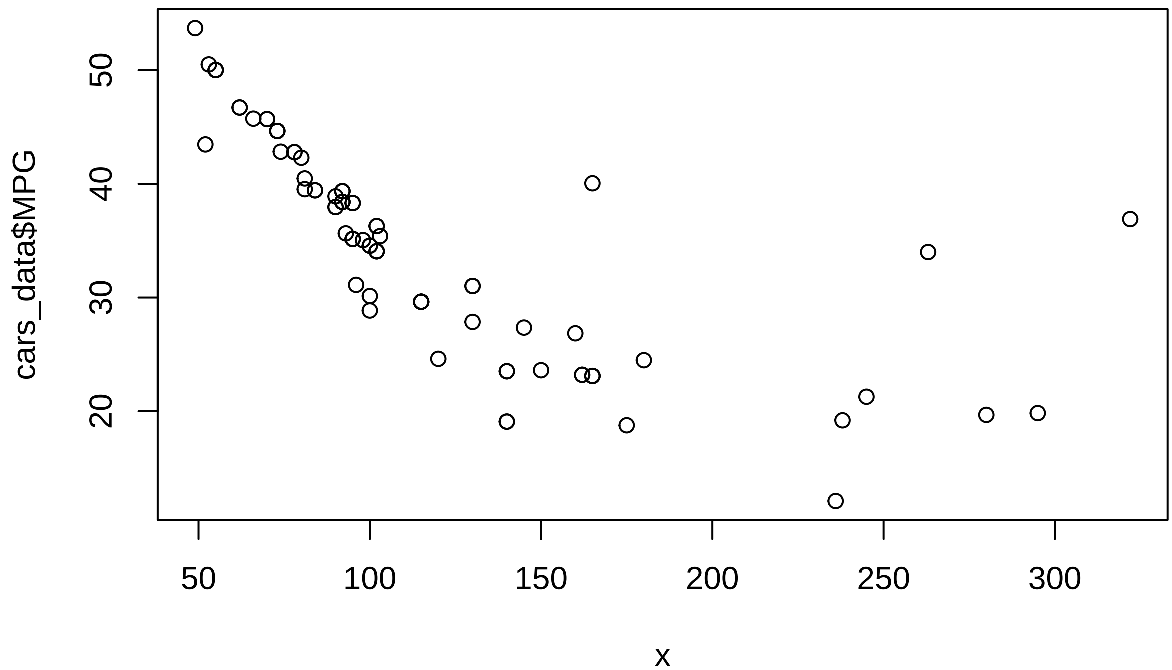
```r
library(corrgram)
```

```
## Warning: package 'corrgram' was built under R version 3.3.3
```

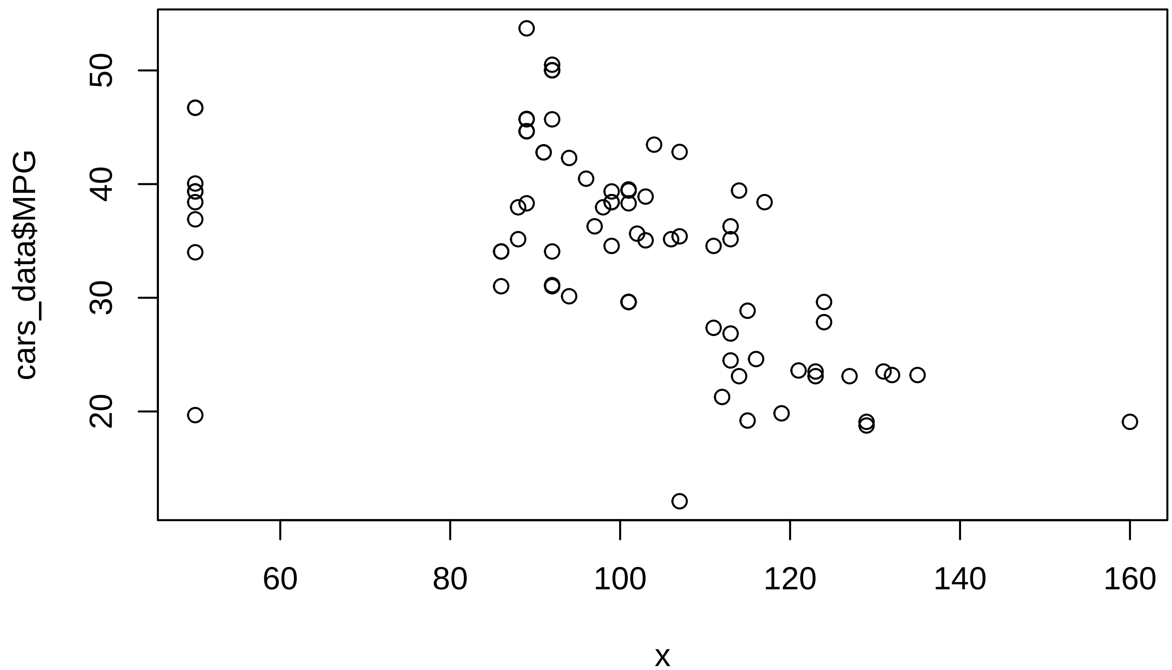
```r
corrgram(cars_data)
```



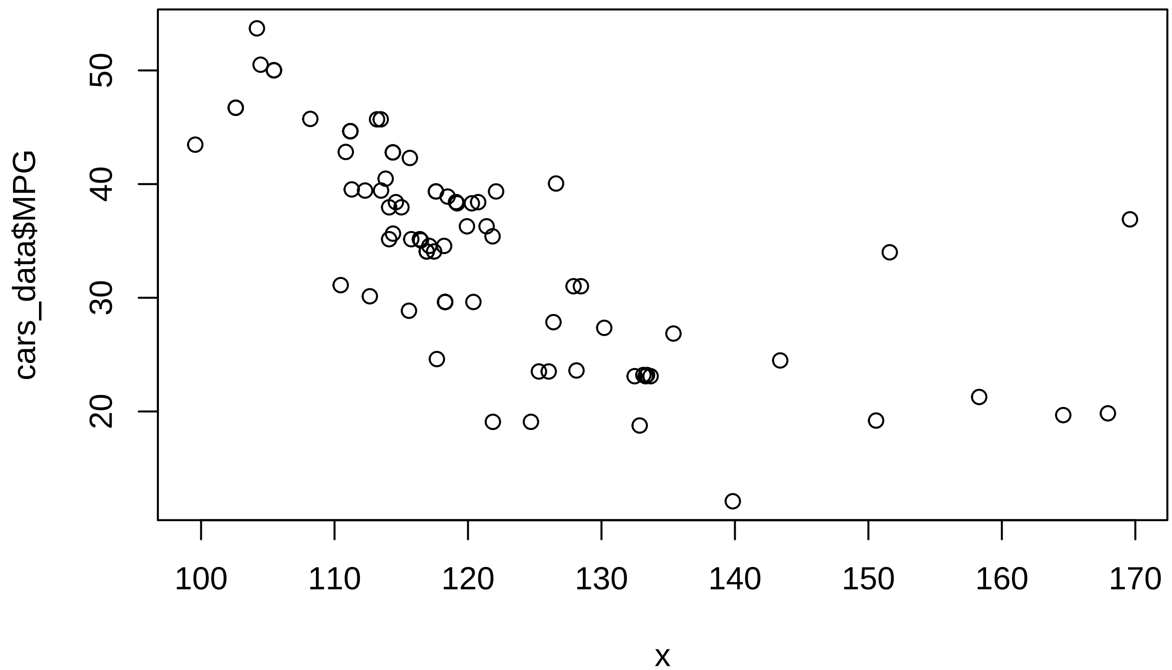

```
cor_plot<-function(x){
  plot(x,cars_data$MPG)
}

cor_plot(cars_data$HP)
```



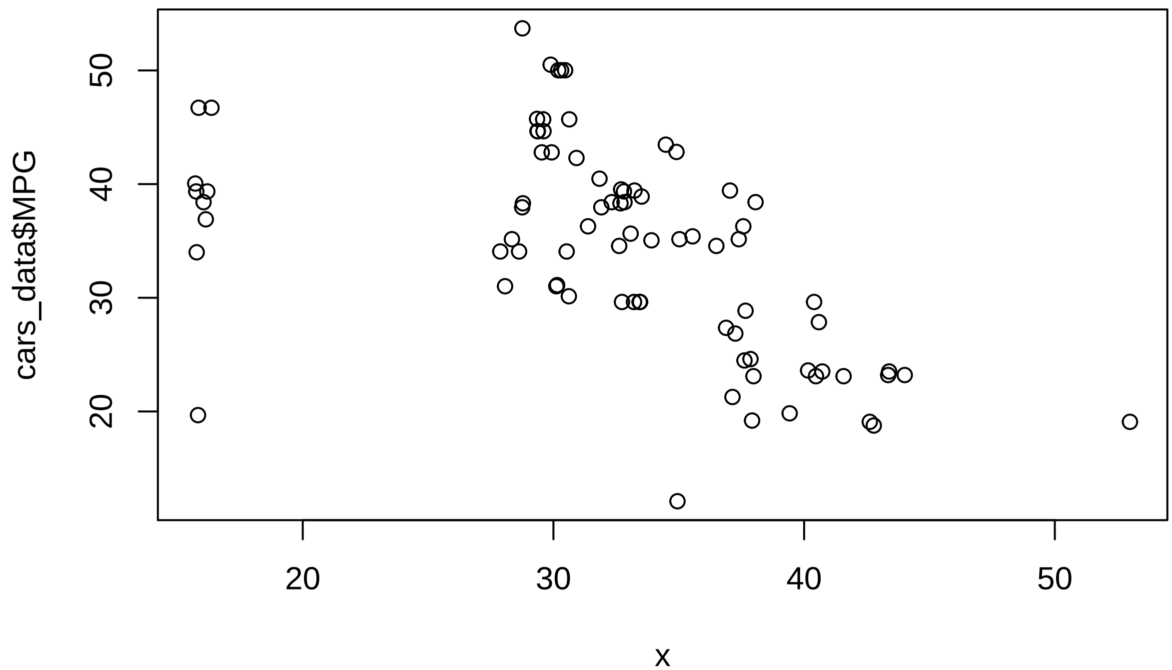

```
# by look at the below given diagram its negative correled


cor_plot(cars_data$VOL)
```



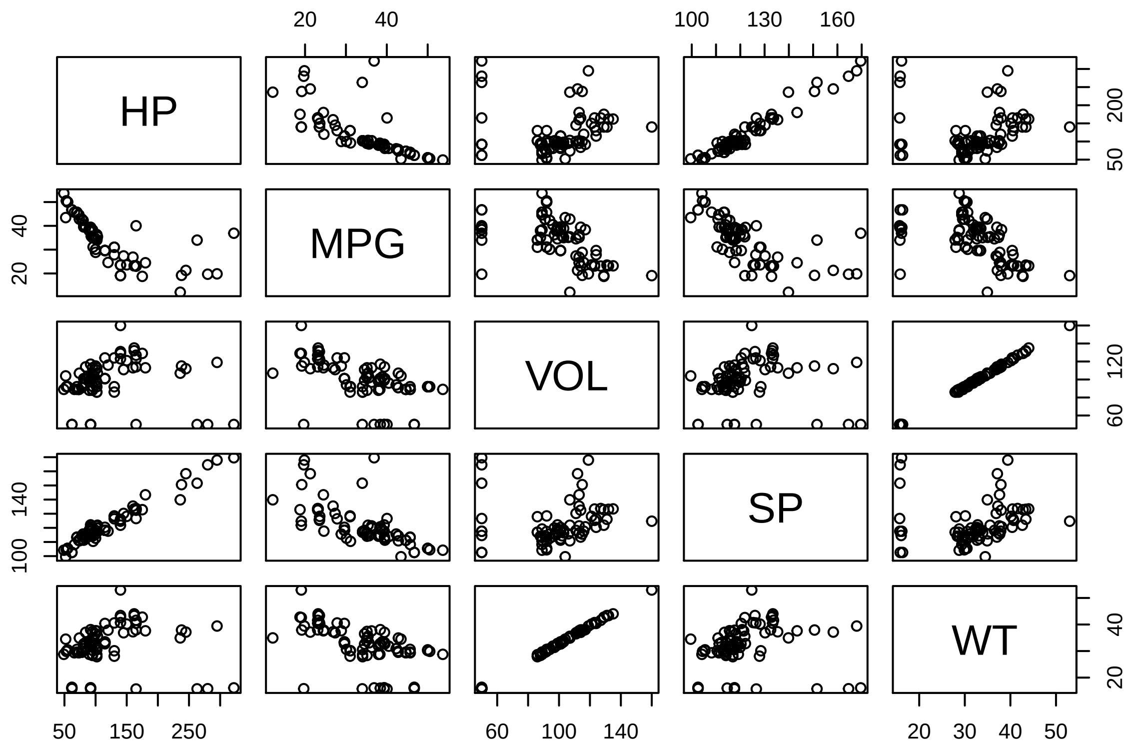

```
# no correlation

cor_plot(cars_data$SP)
```




```
# neagtive correlation

cor_plot(cars_data$WT)
```




```r
# no correlation

# to all the correlation
pairs(cars_data)
```




```r
cor(cars_data)
```

```
##             HP        MPG         VOL         SP          WT
## HP   1.00000000 -0.7250383  0.07745947  0.9738481  0.07651307
## MPG -0.72503835  1.0000000 -0.52905658 -0.6871246 -0.52675909
## VOL  0.07745947 -0.5290566  1.00000000  0.1021700  0.99920308
## SP   0.97384807 -0.6871246  0.10217001  1.0000000  0.10243919
## WT   0.07651307 -0.5267591  0.99920308  0.1024392  1.00000000
```

if we see that above given plot its clear that there is no correlation mpg with any other variabler and but the coffiency problem which has the hp is storng correlated to sp and vol to wt which leads collinearity problem. so we can take one variable insteat of two variable.

Partial Correlation matrix - Pure Correlation b/n the varibles

```r
#install.packages("corpcor")
library(corpcor)
```

```
## Warning: package 'corpcor' was built under R version 3.3.3
```

```r
cor2pcor(cor(cars_data))
```

```
##             [,1]        [,2]        [,3]       [,4]        [,5]
## [1,]  1.00000000 -0.51507804  0.07802551  0.9448373 -0.10251007
## [2,] -0.51507804  1.00000000 -0.06763373  0.2756467  0.02712318
## [3,]  0.07802551 -0.06763373  1.00000000 -0.1056994  0.99838084
```

```
## [4,]  0.94483727  0.27564673 -0.10569943  1.0000000  0.12170021
## [5,] -0.10251007  0.02712318  0.99838084  0.1217002  1.00000000
```

even we cross check correlation between the vol and wt and hp and sp it has high correlation value

now building a muiltple linear regression model

```r
model_car<-lm(MPG~HP+SP+VOL+WT,data = cars_data)
summary(model_car)
```

```
##
## Call:
## lm(formula = MPG ~ HP + SP + VOL + WT, data = cars_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6320 -2.9944 -0.3705  2.2149 15.6179
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.67734   14.90030   2.059   0.0429 *
## HP          -0.20544    0.03922  -5.239  1.4e-06 ***
## SP           0.39563    0.15826   2.500   0.0146 *
## VOL         -0.33605    0.56864  -0.591   0.5563
## WT           0.40057    1.69346   0.237   0.8136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.488 on 76 degrees of freedom
## Multiple R-squared:  0.7705, Adjusted R-squared:  0.7585
## F-statistic:  63.8 on 4 and 76 DF,  p-value: < 2.2e-16
```

if we see the pvaluse of vol and wt are in sufficient beaceuse it greater than o.05

now we are creatting module using weight and volume

```r
mode_c_vol<-lm(cars_data$MPG~cars_data$VOL)
summary(mode_c_vol)
```

```
##
## Call:
## lm(formula = cars_data$MPG ~ cars_data$VOL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.3074  -5.2026   0.1902   5.4536  17.1632
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   55.81709    3.95696  14.106  < 2e-16 ***
## cars_data$VOL -0.21662    0.03909  -5.541 3.82e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.798 on 79 degrees of freedom
## Multiple R-squared:  0.2799, Adjusted R-squared:  0.2708
## F-statistic: 30.71 on 1 and 79 DF,  p-value: 3.823e-07
```

```r
# if we see the pavalue its not insufficient for volum

#now we will check for weight
mode_c_w<-lm(cars_data$MPG~cars_data$WT)
summary(mode_c_w)
```

```
##
## Call:
## lm(formula = cars_data$MPG ~ cars_data$WT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.3933  -5.4377   0.2738   5.2951  16.9351
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   55.2296     3.8761  14.249  < 2e-16 ***
## cars_data$WT  -0.6420     0.1165  -5.508 4.38e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.811 on 79 degrees of freedom
## Multiple R-squared:  0.2775, Adjusted R-squared:  0.2683
## F-statistic: 30.34 on 1 and 79 DF,  p-value: 4.383e-07
```

```r
#if we see the pavalue its not insufficient for weight

#let comebind both together
mode_c_vw<-lm(cars_data$MPG~cars_data$WT+cars_data$VOL)
summary(mode_c_vw)
```

```
##
## Call:
## lm(formula = cars_data$MPG ~ cars_data$WT + cars_data$VOL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.9939  -4.9460   0.0028   5.3905  17.6972
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    56.8847     4.5342   12.55   <2e-16 ***
## cars_data$WT    1.4349     2.9291    0.49    0.626
## cars_data$VOL  -0.6983     0.9841   -0.71    0.480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

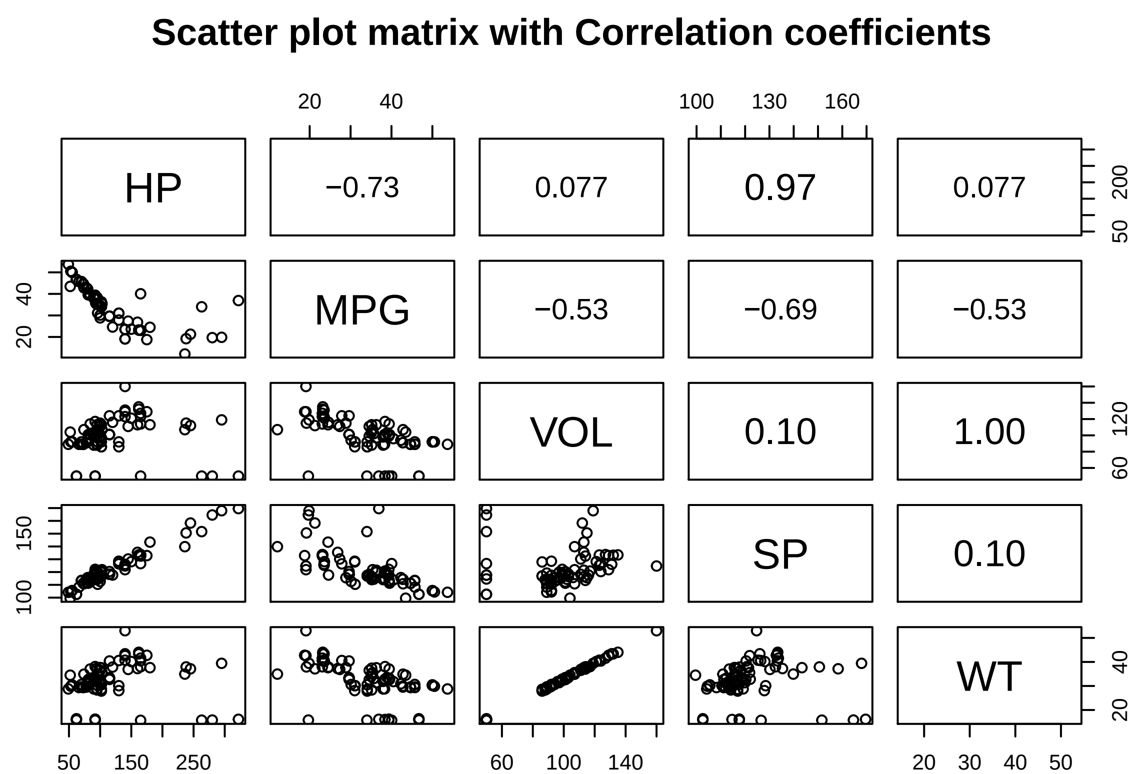
```
##
## Residual standard error: 7.835 on 78 degrees of freedom
## Multiple R-squared:  0.2821, Adjusted R-squared:  0.2637
## F-statistic: 15.33 on 2 and 78 DF,  p-value: 2.434e-06
```

```
# if we see the p value both insiffcent
#so there colinearnity problem
```

So there exists a collinearity problem b/n volume and weight

Scatter plot matrix along with Correlation Coefficients

```
panel.cor<-function(x,y,digits=2,prefix="",cex.cor)
{
  usr<- par("usr"); on.exit(par(usr))
  par(usr=c(0,1,0,1))
  r=(cor(x,y))
  txt<- format(c(r,0.123456789),digits=digits)[1]
  txt<- paste(prefix,txt,sep="")
  if(missing(cex.cor)) cex<-0.4/strwidth(txt)
  text(0.5,0.5,txt,cex=cex)
}
pairs(cars_data,upper.panel = panel.cor,main="Scatter plot matrix with Correlation coefficients")
```



It is Better to delete influential observations rather than deleting entire column which is

costliest process

Deletion Diagnostics for identifying influential observations

```r
influence.measures(model_car)
```

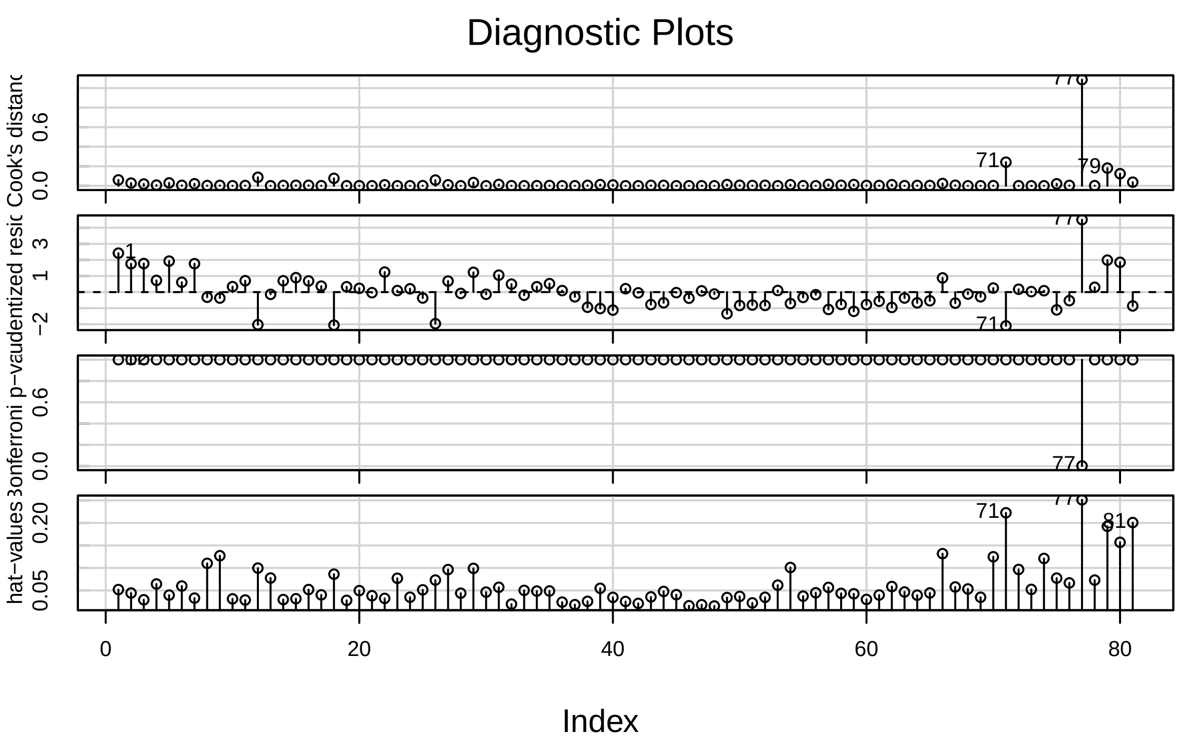
```
## Influence measures of
##   lm(formula = MPG ~ HP + SP + VOL + WT, data = cars_data) :
##
##       dfb.1_    dfb.HP    dfb.SP  dfb.VOL    dfb.WT    dffit cov.r
## 1   0.027438 -0.064490 -1.06e-02  0.34491 -0.348732  0.56724 0.774
## 2   0.130255  0.035945 -8.57e-02 -0.22541  0.224097  0.37945 0.913
## 3   0.090410  0.010673 -6.06e-02 -0.03970  0.038239  0.30826 0.896
## 4  -0.050842 -0.086513  7.38e-02 -0.12604  0.124908  0.19016 1.103
## 5   0.086150  0.014157 -7.17e-02  0.17799 -0.179572  0.39154 0.874
## 6  -0.038227 -0.068912  5.82e-02 -0.09931  0.097995  0.15449 1.108
## 7   0.106859  0.021234 -7.10e-02 -0.11741  0.116017  0.32649 0.901
## 8  -0.065348 -0.044642  5.30e-02 -0.01492  0.017852 -0.11238 1.193
## 9  -0.093081 -0.062806  7.25e-02  0.05912 -0.055744 -0.14161 1.213
## 10 -0.031771 -0.041975  3.74e-02 -0.00857  0.008142  0.06104 1.095
## 11  0.010849 -0.018542  5.55e-03 -0.05995  0.058726  0.12152 1.064
## 12  0.185184  0.269479 -2.60e-01 -0.24251  0.264240 -0.67399 0.909
## 13  0.029117  0.032757 -3.20e-02  0.01302 -0.012894 -0.04051 1.157
## 14  0.011711 -0.017970  4.99e-03 -0.06396  0.062745  0.12343 1.066
## 15  0.054751  0.013861 -3.43e-02 -0.07598  0.074656  0.16218 1.046
## 16  0.024610 -0.009612 -3.19e-03 -0.12416  0.122996  0.16212 1.092
## 17 -0.035743 -0.043967  3.84e-02  0.04293 -0.043618  0.07590 1.103
## 18  0.109967  0.222667 -2.14e-01  0.11288 -0.091298 -0.63072 0.889
## 19 -0.021167 -0.032416  2.75e-02 -0.01180  0.011223  0.05655 1.091
## 20 -0.034346 -0.035729  3.34e-02  0.03309 -0.033011  0.05350 1.120
## 21  0.005188  0.005631 -5.32e-03 -0.00310  0.003118 -0.00748 1.111
## 22 -0.009172 -0.027690  1.91e-03  0.13117 -0.128786  0.22782 0.997
## 23 -0.017641 -0.017551  1.69e-02  0.02021 -0.020237  0.02750 1.157
## 24 -0.000285 -0.007742  4.68e-03 -0.02650  0.026417  0.03917 1.104
## 25  0.048301  0.052463 -5.01e-02 -0.05565  0.056583 -0.08596 1.116
## 26 -0.082883  0.016880 -2.74e-03  0.01903  0.000512 -0.55124 0.897
## 27 -0.158459 -0.151772  1.46e-01  0.14820 -0.146183  0.22349 1.146
## 28  0.008111  0.010392 -9.73e-03  0.00955 -0.009524 -0.01799 1.117
## 29  0.323026  0.261335 -3.06e-01 -0.18454  0.187890  0.40784 1.073
## 30  0.013519  0.015781 -1.53e-02  0.01685 -0.017023 -0.02991 1.119
## 31 -0.037571 -0.035690  1.77e-02  0.20218 -0.198784  0.25885 1.053
## 32  0.026770  0.015135 -2.32e-02 -0.01523  0.015672  0.07028 1.072
## 33  0.020540  0.020495 -1.95e-02 -0.03504  0.035187 -0.04458 1.122
## 34 -0.030604 -0.035152  3.35e-02 -0.04250  0.043347  0.07496 1.115
## 35  0.010936  0.005937 -1.56e-02  0.09112 -0.090704  0.11893 1.103
## 36 -0.001573 -0.002726  1.58e-03  0.00893 -0.008936  0.01476 1.094
## 37 -0.009721 -0.002428  5.92e-03 -0.00385  0.004414 -0.04023 1.082
## 38 -0.017382 -0.004671  1.08e-02 -0.08410  0.086403 -0.15334 1.034
```

```
## 39 -0.083121 -0.045772  5.21e-02  0.20544 -0.203131 -0.24677 1.054
## 40  0.152467  0.151483 -1.63e-01 -0.02970  0.032106 -0.21212 1.019
## 41  0.002327 -0.000434 -1.92e-03 -0.01530  0.015947  0.03409 1.093
## 42 -0.001566 -0.000752  1.27e-03  0.00332 -0.003399 -0.00694 1.092
## 43 -0.039107 -0.016950  2.19e-02  0.11655 -0.115675 -0.14952 1.064
## 44 -0.027999 -0.021059  2.85e-02 -0.11046  0.111714 -0.14919 1.089
## 45 -0.000546 -0.000433  8.74e-04 -0.00544  0.005411 -0.00711 1.114
## 46 -0.009063 -0.002513  5.52e-03  0.01889 -0.018974 -0.04849 1.076
## 47 -0.002202 -0.002566  1.93e-03  0.00112 -0.000935  0.01048 1.088
## 48 -0.002937 -0.001218  2.35e-03  0.00350 -0.003634 -0.01517 1.084
## 49  0.159432  0.162450 -1.76e-01 -0.04698  0.051421 -0.25338 0.982
## 50 -0.101653 -0.090453  9.35e-02  0.10473 -0.105584 -0.16303 1.059
## 51 -0.081377 -0.076501  7.95e-02  0.02529 -0.026046 -0.12191 1.047
## 52 -0.099716 -0.089107  9.21e-02  0.09732 -0.098168 -0.15790 1.057
## 53 -0.002025  0.000521 -9.74e-04  0.01874 -0.018224  0.02530 1.138
## 54  0.153429  0.131520 -1.56e-01  0.09921 -0.100335 -0.23838 1.149
## 55  0.041217  0.031761 -3.87e-02  0.00259 -0.003279 -0.06524 1.102
## 56  0.014599  0.009917 -1.06e-02 -0.02015  0.019391 -0.03490 1.117
## 57 -0.235589 -0.211317  2.27e-01  0.03237 -0.032098 -0.26485 1.049
## 58 -0.048198 -0.054597  5.77e-02 -0.11798  0.117361 -0.16443 1.074
## 59 -0.186548 -0.171879  1.86e-01 -0.06523  0.065393 -0.25265 1.015
## 60 -0.037707 -0.039495  4.82e-02 -0.05542  0.052562 -0.13667 1.058
## 61  0.024059  0.018819 -2.64e-02  0.07541 -0.076628 -0.11635 1.089
## 62 -0.126961 -0.143815  1.49e-01 -0.10712  0.103065 -0.23796 1.069
## 63 -0.015838 -0.024985  2.28e-02  0.02280 -0.025296 -0.08159 1.111
## 64 -0.047739 -0.060306  5.62e-02  0.04873 -0.052219 -0.13562 1.080
## 65 -0.035161 -0.045946  3.98e-02  0.06111 -0.063646 -0.11608 1.097
## 66  0.190254  0.189618 -1.71e-01  0.08129 -0.089976  0.34353 1.169
## 67 -0.036320 -0.049478  3.59e-02  0.12250 -0.124629 -0.16959 1.099
## 68  0.006786  0.000503 -2.83e-03 -0.01561  0.014952 -0.02907 1.128
## 69  0.006291 -0.005653  9.04e-05 -0.00920  0.007847 -0.05551 1.101
## 70 -0.052619 -0.034244  5.05e-02 -0.02289  0.023153  0.09494 1.216
## 71  0.375081  0.225899 -4.17e-01 -0.20353  0.231060 -1.12358 1.033
## 72 -0.016888 -0.005558  8.66e-03  0.04110 -0.039847  0.05801 1.180
## 73 -0.001556 -0.000118  6.18e-04  0.00260 -0.002414  0.00678 1.127
## 74  0.006863  0.011133 -1.10e-02 -0.00131  0.002486  0.03234 1.215
## 75 -0.206329 -0.234704  2.32e-01  0.04805 -0.055962 -0.32060 1.068
## 76 -0.054797 -0.076780  6.57e-02  0.05050 -0.053946 -0.14193 1.124
## 77  0.214209  0.605131 -1.15e-01 -0.29653  0.240034  2.60978 0.431
## 78 -0.001403  0.020701 -4.72e-03 -0.00835  0.009157  0.08386 1.146
## 79  0.328332  0.443574 -3.01e-01  0.14545 -0.167508  0.97032 1.024
## 80 -0.249491 -0.044284  2.05e-01 -0.14528  0.150314  0.79955 1.013
## 81 -0.316601 -0.384383  3.44e-01 -0.03288  0.030357 -0.43138 1.273
##      cook.d    hat inf
## 1  6.05e-02 0.0520   *
## 2  2.80e-02 0.0443
## 3  1.85e-02 0.0293
## 4  7.28e-03 0.0643
## 5  2.96e-02 0.0396
## 6  4.81e-03 0.0598
## 7  2.07e-02 0.0330
## 8  2.56e-03 0.1102
## 9  4.06e-03 0.1271   *
## 10 7.54e-04 0.0313
```

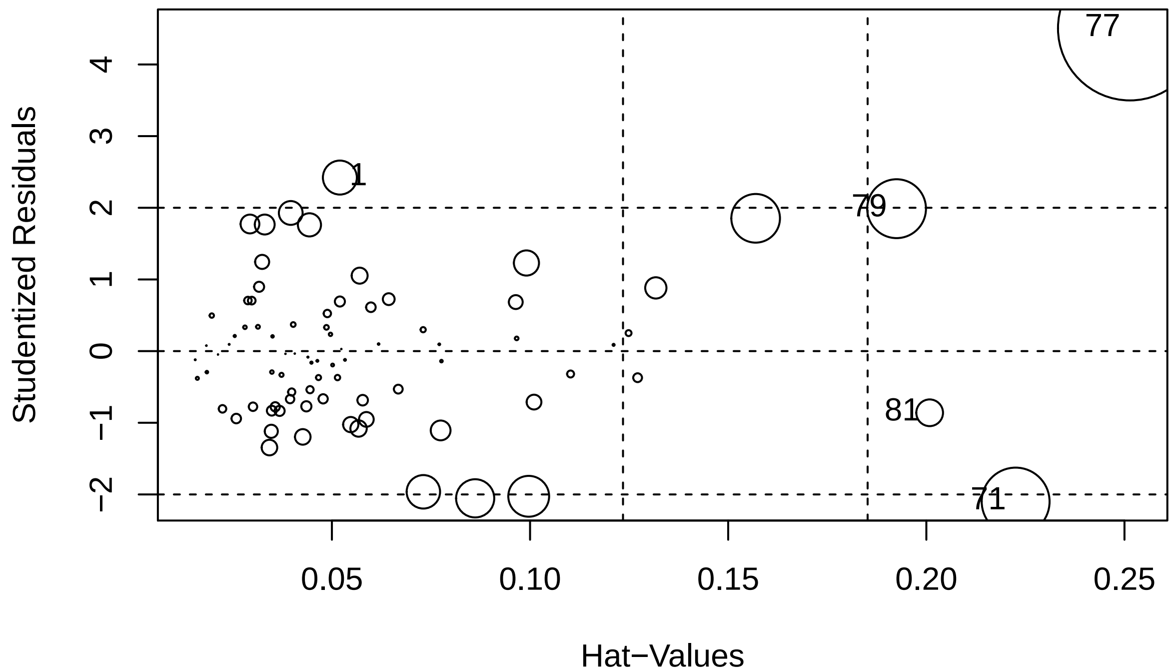
```
## 11 2.97e-03 0.0288
## 12 8.73e-02 0.0997
## 13 3.32e-04 0.0776
## 14 3.07e-03 0.0298
## 15 5.27e-03 0.0316
## 16 5.29e-03 0.0520
## 17 1.17e-03 0.0402
## 18 7.63e-02 0.0861
## 19 6.47e-04 0.0281
## 20 5.80e-04 0.0496
## 21 1.13e-05 0.0383
## 22 1.03e-02 0.0324
## 23 1.53e-04 0.0771
## 24 3.11e-04 0.0350
## 25 1.49e-03 0.0514
## 26 5.86e-02 0.0731
## 27 1.01e-02 0.0964
## 28 6.56e-05 0.0440
## 29 3.30e-02 0.0991
## 30 1.81e-04 0.0463
## 31 1.34e-02 0.0570
## 32 9.98e-04 0.0197
## 33 4.03e-04 0.0502
## 34 1.14e-03 0.0486
## 35 2.86e-03 0.0488
## 36 4.41e-05 0.0241
## 37 3.28e-04 0.0184
## 38 4.71e-03 0.0259
## 39 1.22e-02 0.0548
## 40 8.97e-03 0.0347
## 41 2.35e-04 0.0255
## 42 9.76e-06 0.0213
## 43 4.49e-03 0.0357
## 44 4.48e-03 0.0478
## 45 1.03e-05 0.0406
## 46 4.76e-04 0.0160
## 47 2.23e-05 0.0183
## 48 4.66e-05 0.0155
## 49 1.27e-02 0.0342
## 50 5.34e-03 0.0368
## 51 2.99e-03 0.0224
## 52 5.01e-03 0.0348
## 53 1.30e-04 0.0618
## 54 1.14e-02 0.1010
## 55 8.61e-04 0.0373
## 56 2.47e-04 0.0448
## 57 1.40e-02 0.0567
## 58 5.44e-03 0.0435
## 59 1.27e-02 0.0426
## 60 3.76e-03 0.0301
## 61 2.73e-03 0.0399
## 62 1.13e-02 0.0587
## 63 1.35e-03 0.0466
## 64 3.71e-03 0.0394
```

```
## 65 2.72e-03 0.0445
## 66 2.37e-02 0.1317
## 67 5.79e-03 0.0578
## 68 1.71e-04 0.0533
## 69 6.24e-04 0.0349
## 70 1.83e-03 0.1249   *
## 71 2.42e-01 0.2225   *
## 72 6.82e-04 0.0966
## 73 9.33e-06 0.0524
## 74 2.12e-04 0.1211   *
## 75 2.05e-02 0.0774
## 76 4.07e-03 0.0667
## 77 1.09e+00 0.2514   *
## 78 1.42e-03 0.0730
## 79 1.81e-01 0.1925   *
## 80 1.24e-01 0.1569   *
## 81 3.73e-02 0.2008   *
```

```r
library(car)
## plotting Influential measures
influenceIndexPlot(model_car,id.n=3) # index plots for infuence measures
```




```r
influencePlot(model_car,id.n=3) # A user friendly representation of the above
```




```
##        StudRes        Hat      CookD
## 1    2.4217621 0.05200781 0.06047977
## 71  -2.1001313 0.22253511 0.24164401
## 77   4.5036028 0.25138750 1.08651940
## 79   1.9873749 0.19249263 0.18126775
## 81  -0.8605141 0.20083657 0.03734554
```

after see the plot we can see that 77 row and 71 row are influence data

```r
model_car2<-lm(MPG~VOL+SP+HP+WT,data=cars_data[-c(71,77),])
summary(model_car2)
```

```
##
## Call:
## lm(formula = MPG ~ VOL + SP + HP + WT, data = cars_data[-c(71,
##     77), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9343 -2.3434 -0.5155  1.9756 10.8897
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 25.26269   13.49494   1.872   0.0652 .
## VOL         -0.13878    0.50979  -0.272   0.7862
## SP           0.44336    0.14391   3.081   0.0029 **
## HP          -0.22953    0.03537  -6.489 8.68e-09 ***
## WT          -0.13051    1.51940  -0.086   0.9318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.001 on 74 degrees of freedom
## Multiple R-squared:  0.8162, Adjusted R-squared:  0.8063
## F-statistic: 82.15 on 4 and 74 DF,  p-value: < 2.2e-16
```

after see p value of weight its that its great influence so we remove wt variable

Variance Inflation factor to check collinearity b/n variables

vif>10 then there exists collinearity among all the variables

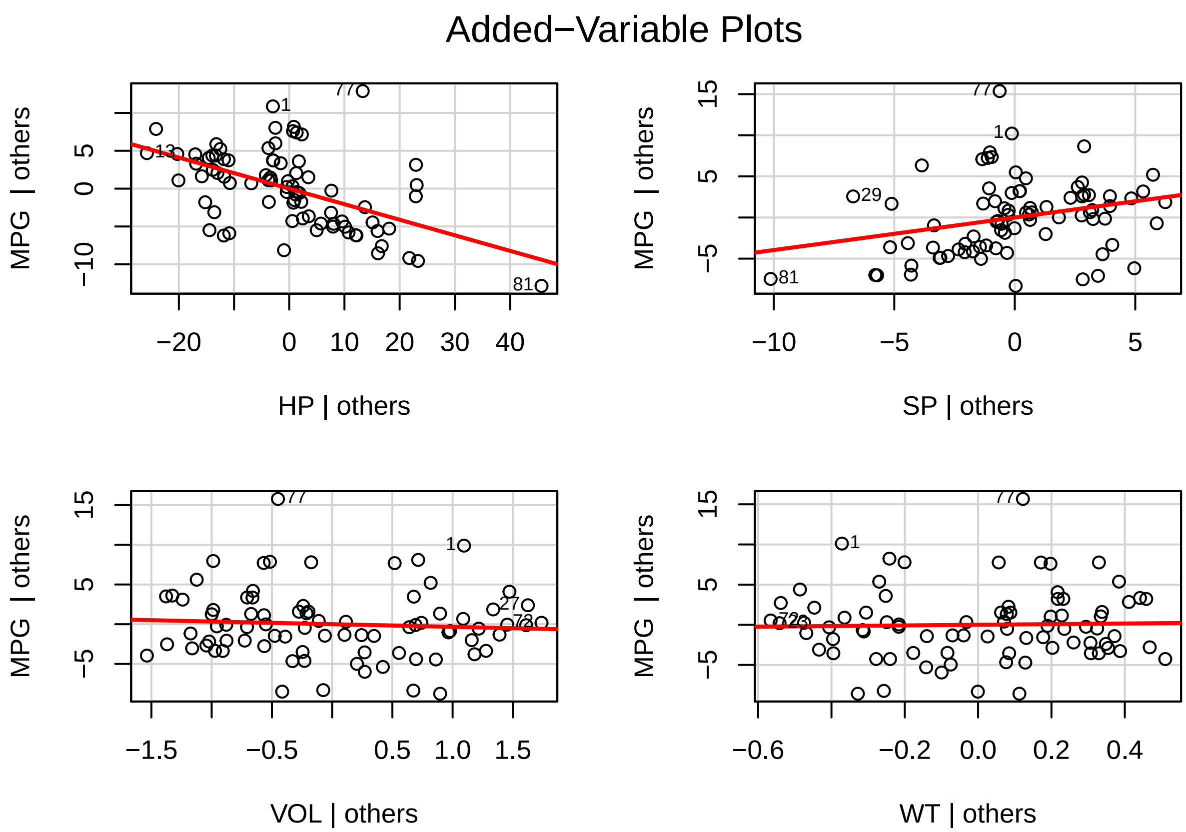
```
vif(model_car)
```

```
##        HP        SP       VOL        WT
##  19.92659  20.00764 638.80608 639.53382
```

wt has high variance influence so remove the wt in model

Added Variable plot to check correlation b/n variables and o/p variable

```
avPlots(model_car,id.n=2,id.cex=0.7)
```



VIF and AV plot has given us an indication to delete “wt” variable

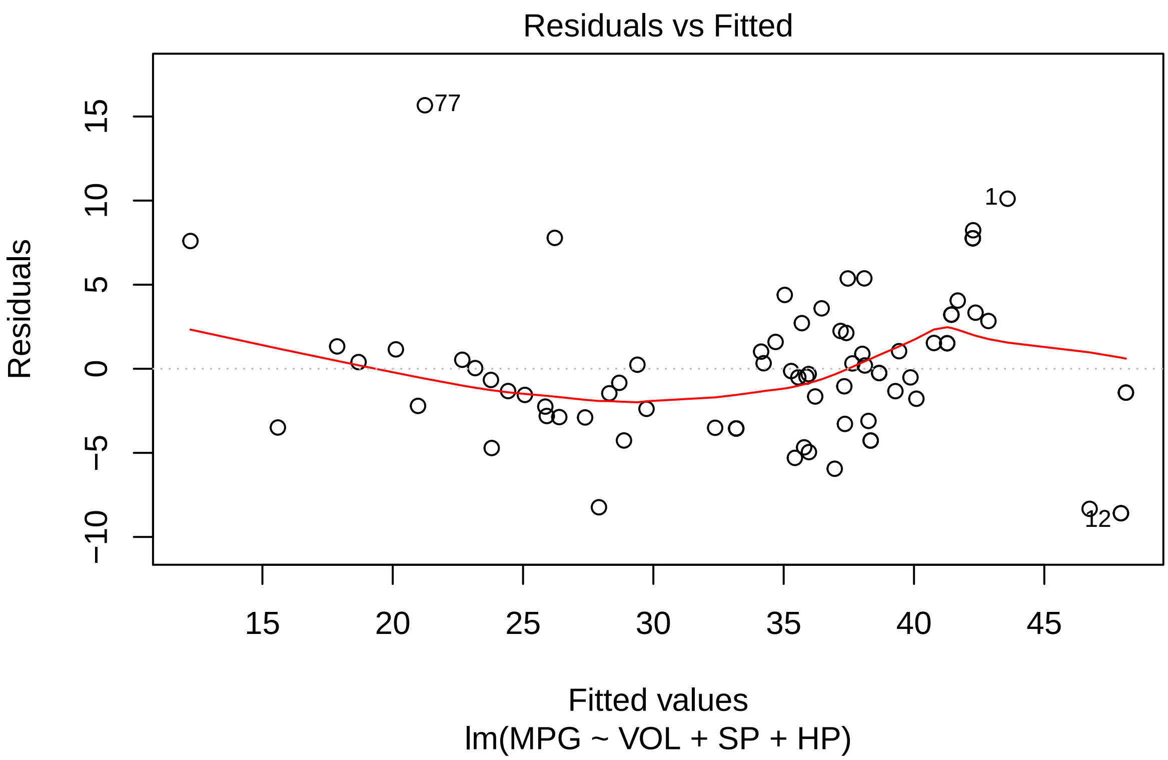
atlast the coreect fianal mode

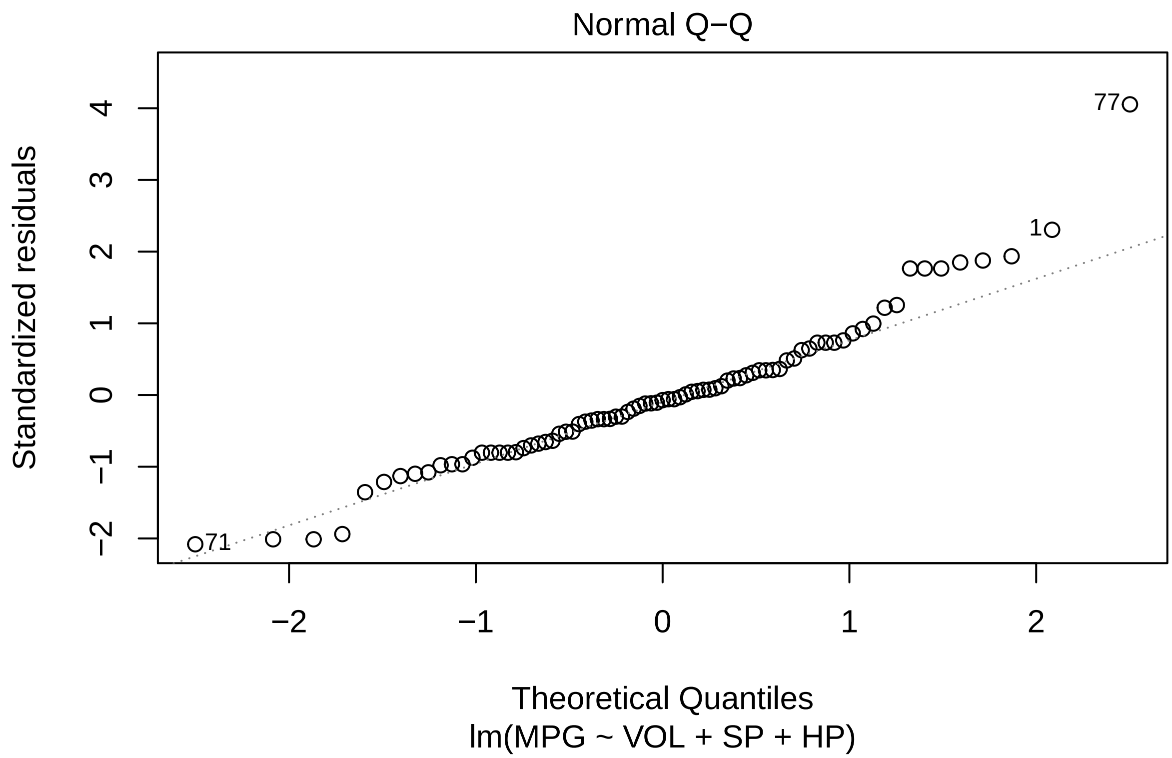
```r
## Final model
final_model<-lm(MPG~VOL+SP+HP,data=cars_data)
summary(final_model)
```

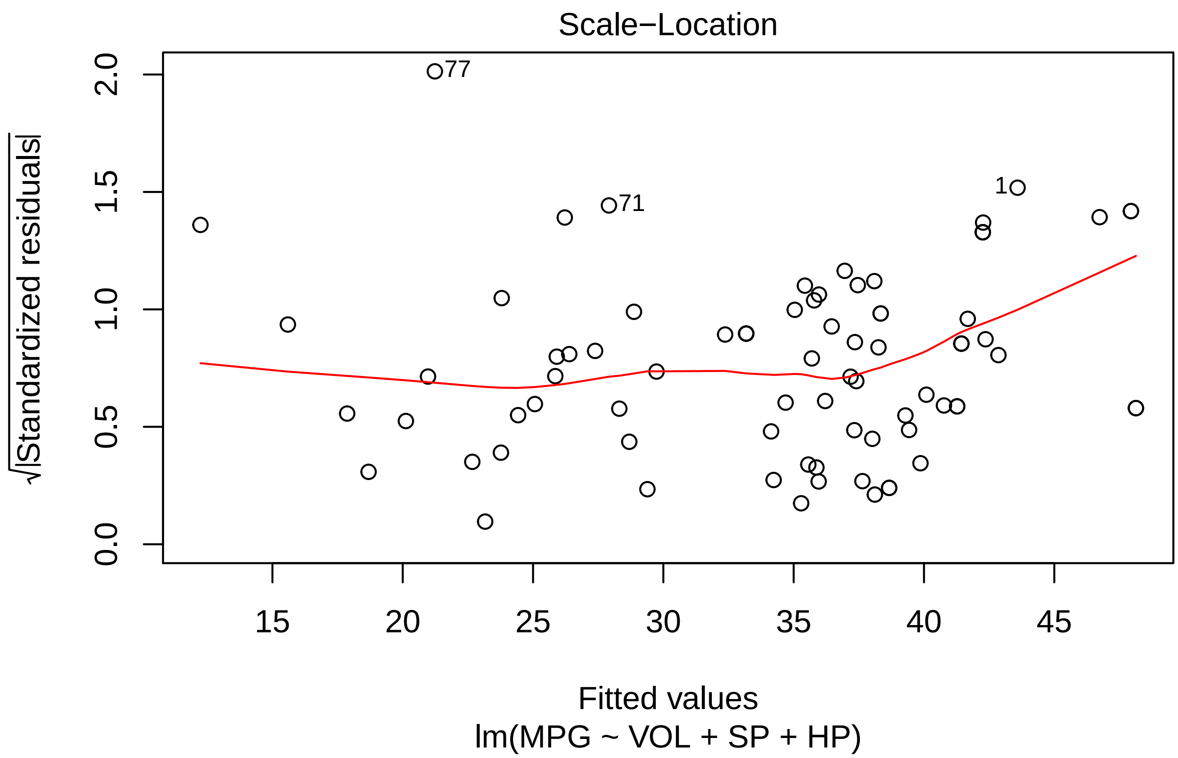
```
##
## Call:
## lm(formula = MPG ~ VOL + SP + HP, data = cars_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5869 -2.8942 -0.3157  2.1291 15.6669
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.92339   14.46589   2.069   0.0419 *
## VOL         -0.20165    0.02259  -8.928 1.65e-13 ***
## SP           0.40066    0.15586   2.571   0.0121 *
## HP          -0.20670    0.03861  -5.353 8.64e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

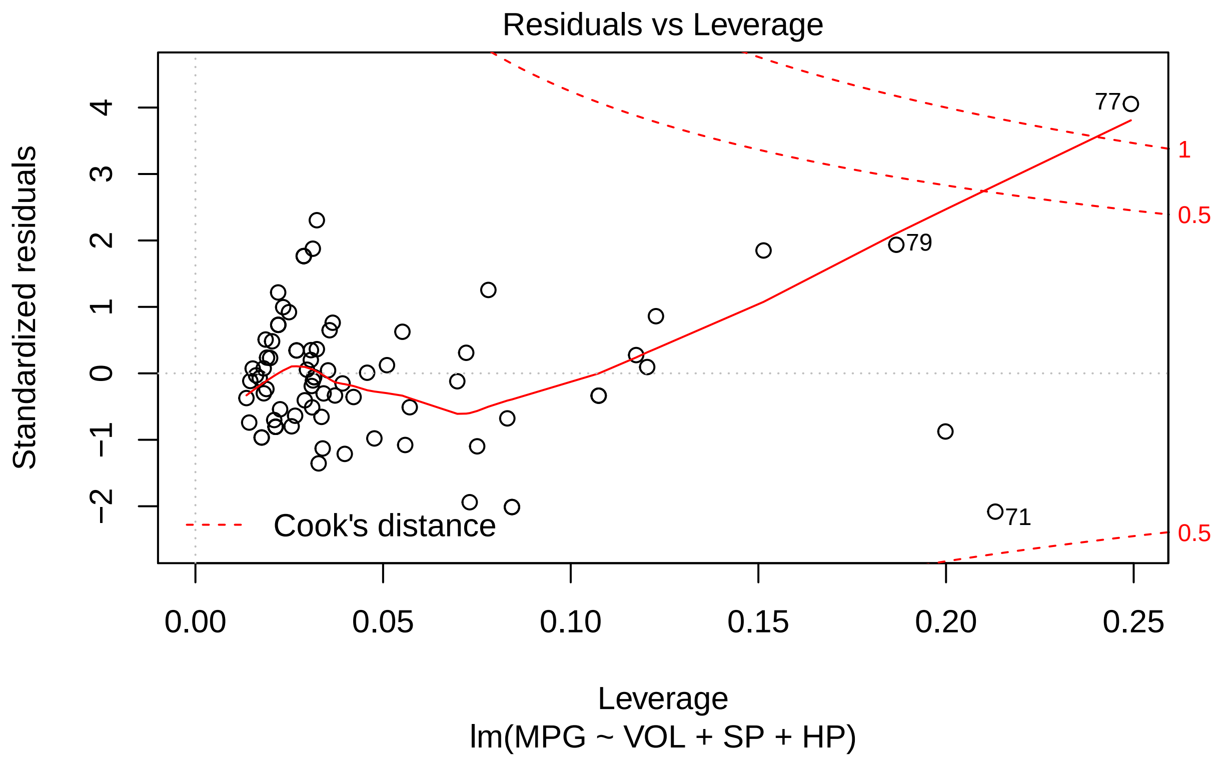
```
##
## Residual standard error: 4.46 on 77 degrees of freedom
## Multiple R-squared:  0.7704, Adjusted R-squared:  0.7614
## F-statistic: 86.11 on 3 and 77 DF,  p-value: < 2.2e-16
```

```
# Evaluate model LINE assumptions
plot(final_model)
```

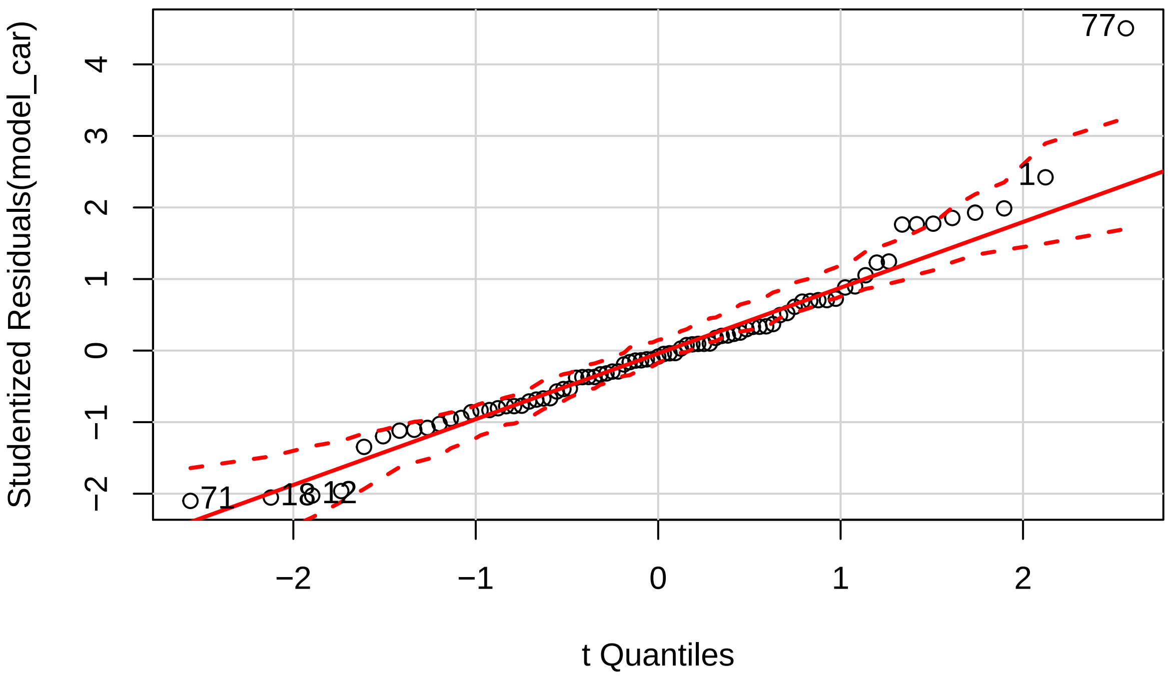



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(MPG ~ VOL + SP + HP)
77
1
71
−2
−1
0
1
2
3
4


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
77
71
1
15
20
25
30
35
40
45
Fitted values
lm(MPG ~ VOL + SP + HP)




```
#Residual plots,QQplot,std-Residuals Vs Fitted,Cook's Distance
qqPlot(model_car,id.n = 5)
```




```
## 71 18 12  1 77
##  1  2  3 80 81
# QQ plot of studentized residuals helps in identifying outlier
```