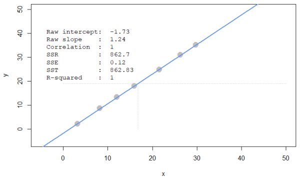
BACS HW (Week 10)

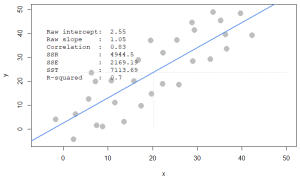
108020024

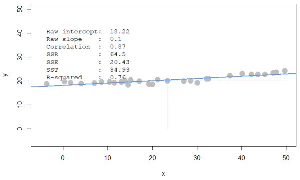
due on 04/23 (Sun) Helped by 108020033

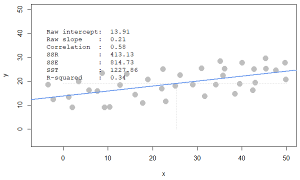
Question 1)

```
library(compstatslib)
```









a)Comparing scenarios 1 and 2, which do we expect to have a stronger R^2 ?

scenarios 1

b)Comparing scenarios 3 and 4, which do we expect to have a stronger R^2 ?

scenarios 3

c)Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

SSE: scenarios 1 smaller, scenarios 2 bigger.

SSR: scenarios 1 smaller, scenarios 2 bigger.

SST: scenarios 1 smaller, scenarios 2 bigger.

d)Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

SSE: scenarios 3 smaller, scenarios 4 bigger.

SSR: scenarios 3 smaller, scenarios 4 bigger.

SST: scenarios 3 smaller, scenarios 4 bigger.

Q2)Let's analzye the programmer_salaries.txt dataset we saw in class. Read the file using read.csv("programmer_salaries.txt", sep="̂") because the columns are separated by tabs ()̂.

```
q2 <- read.csv("programmer_salaries.txt", sep="\t")
```

a) Use the lm() function to estimate the regression model Salary ~ Experience + Score + Degree. Show the beta coefficients, R2, and the first 5 values of y ($fitted.values)and($**residuals)**

```
md1 <- lm(Salary ~ Experience + Score + Degree, data = q2)
summary(md1)
```

```
##
## Call:
## lm(formula = Salary ~ Experience + Score + Degree, data = q2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8963 -1.7290 -0.3375  1.9699  5.0480
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9448     7.3808   1.076   0.2977
## Experience    1.1476     0.2976   3.856   0.0014 **
## Score         0.1969     0.0899   2.191   0.0436 *
## Degree        2.2804     1.9866   1.148   0.2679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.396 on 16 degrees of freedom
## Multiple R-squared:  0.8468, Adjusted R-squared:  0.8181
## F-statistic: 29.48 on 3 and 16 DF,  p-value: 9.417e-07
```

R^2 : 0.8468

First 5 values of $\hat{y}$:

```
head(md1$fitted.values,5)
```

```
##        1        2        3        4        5
## 27.89626 37.95204 26.02901 32.11201 36.34251
```

First 5 values of ϵ :

```
head(md1$residuals,5)
```

```
##          1          2          3          4          5
## -3.8962605  5.0479568 -2.3290112  2.1879860 -0.5425072
```

b) Use only linear algebra and the geometric view of regression to estimate the regression yourself

```
    i) Create an X matrix that has a first column of 1s followed by columns of the independent variables
    (only show the code)
```

```
x <-data.matrix( cbind(rep(1, length(q2)),q2[,-4]))
colnames(x)[1] <- "Intercept"
```

```
    ii) Create a y vector with the Salary values (only show the code)
```

```
y <- q2[,4]
```

```
    iii) Compute the beta_hat vector of estimated regression coefficients (show the code and values)
```

```
beta_hat <- solve(t(x)%*%x)%*%t(x)%*%y
beta_hat
```

```
##                 [,1]
## Intercept  7.944849
## Experience 1.147582
## Score      0.196937
## Degree     2.280424
```

```
    iv) Compute a y_hat vector of estimated y values, and a res vector of residuals
    (show the code and the first 5 values of y_hat and res)
```

```
y_hat <- x%*%beta_hat
head(y_hat,5)
```

```
##          [,1]
## [1,] 27.89626
## [2,] 37.95204
## [3,] 26.02901
## [4,] 32.11201
## [5,] 36.34251
```

```
    v) Using only the results from (i) - (iv), compute SSR, SSE and SST (show the code and values)
```

```
SSE <- sum((y-y_hat)^2)
SST <- SSE / (1 - cor(y,y_hat)^2)
SSR <- cor(y,y_hat)^2 * SST
```

c) Compute R2 for in two ways, and confirm you get the same results (show code and values):

```
    i) Use any combination of SSR, SSE, and SST
```

```
SSR/SST
```

```
##           [,1]
## [1,] 0.8467961
```

```
    ii) Use the squared correlation of vectors y and y hat
```

```
cor(y,y_hat)^2
```

```
##           [,1]
## [1,] 0.8467961
```

Q3

```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
```

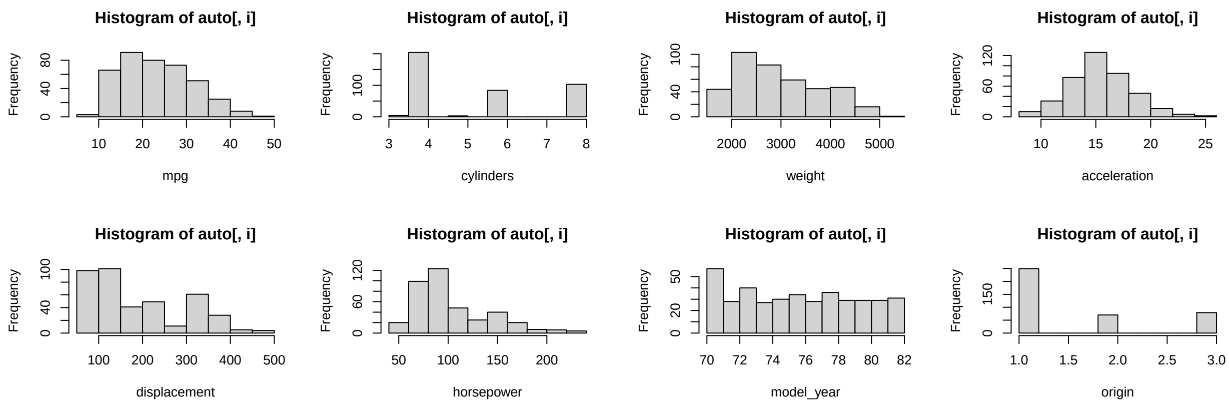
a) Let's first try exploring this data and problem:

```
    i) Visualize the data as you wish (report only relevant/interesting plots)
```

```
par(mfrow=c(2,2))

library(GGally)
```

```
##     ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
for (i in seq(1:8)) {
  hist(auto[,i],xlab=(colnames(auto)[i]))
}
```



```
par(mfrow=c(1,1))
```

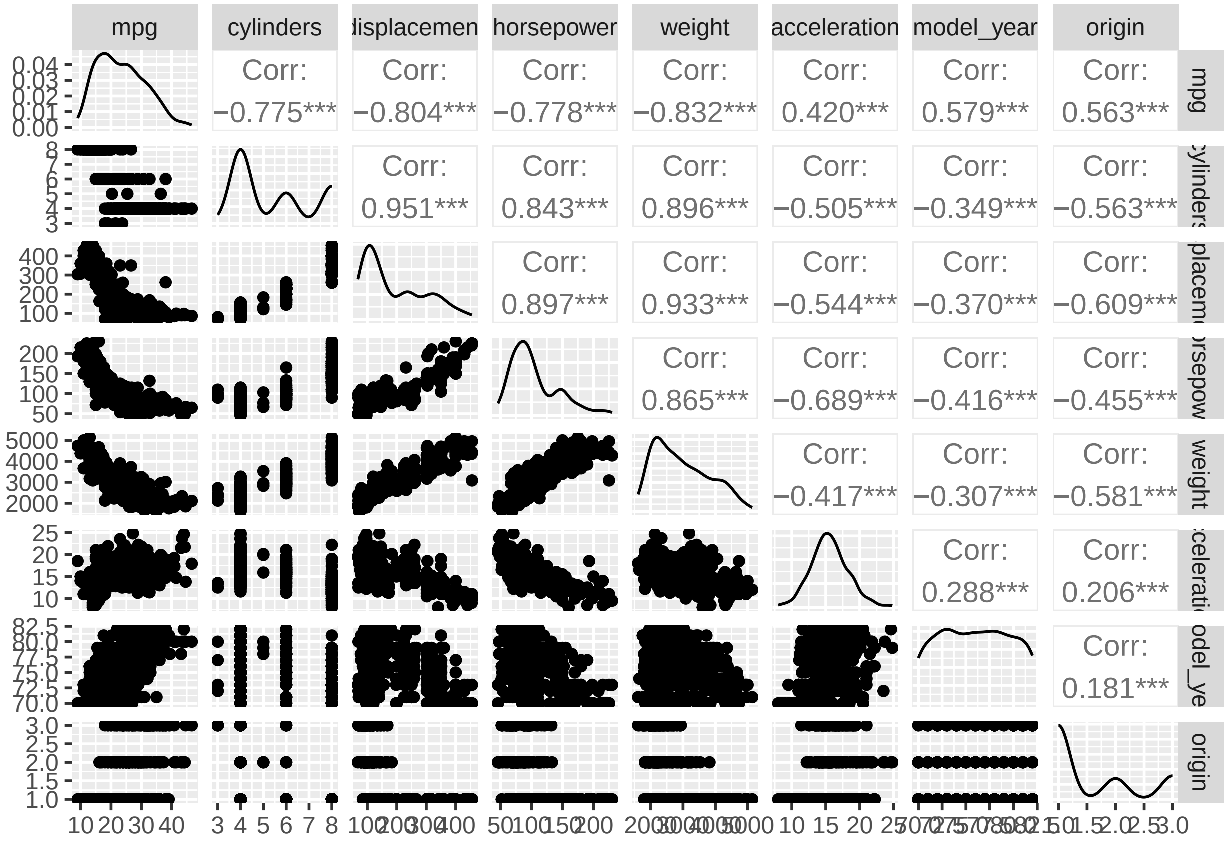
```
    ii) Report a correlation table of all variables, rounding to two decimal places
    (in the cor() function, set use="pairwise.complete.obs" to handle missing values)
```

```
round(cor(auto[,-9],use="pairwise.complete.obs"),2)
```

```
##                mpg cylinders displacement horsepower weight acceleration
## mpg           1.00     -0.78        -0.80      -0.78  -0.83         0.42
## cylinders    -0.78      1.00         0.95       0.84   0.90        -0.51
## displacement -0.80      0.95         1.00       0.90   0.93        -0.54
## horsepower   -0.78      0.84         0.90       1.00   0.86        -0.69
## weight       -0.83      0.90         0.93       0.86   1.00        -0.42
## acceleration  0.42     -0.51        -0.54      -0.69  -0.42         1.00
## model_year    0.58     -0.35        -0.37      -0.42  -0.31         0.29
## origin        0.56     -0.56        -0.61      -0.46  -0.58         0.21
##              model_year origin
## mpg                0.58   0.56
## cylinders         -0.35  -0.56
## displacement      -0.37  -0.61
## horsepower        -0.42  -0.46
## weight            -0.31  -0.58
## acceleration       0.29   0.21
## model_year         1.00   0.18
## origin             0.18   1.00
```

```
    iii) From the visualizations and correlations, which variables appear to relate to mpg?
```

```
ggpairs (auto[,-9])
```

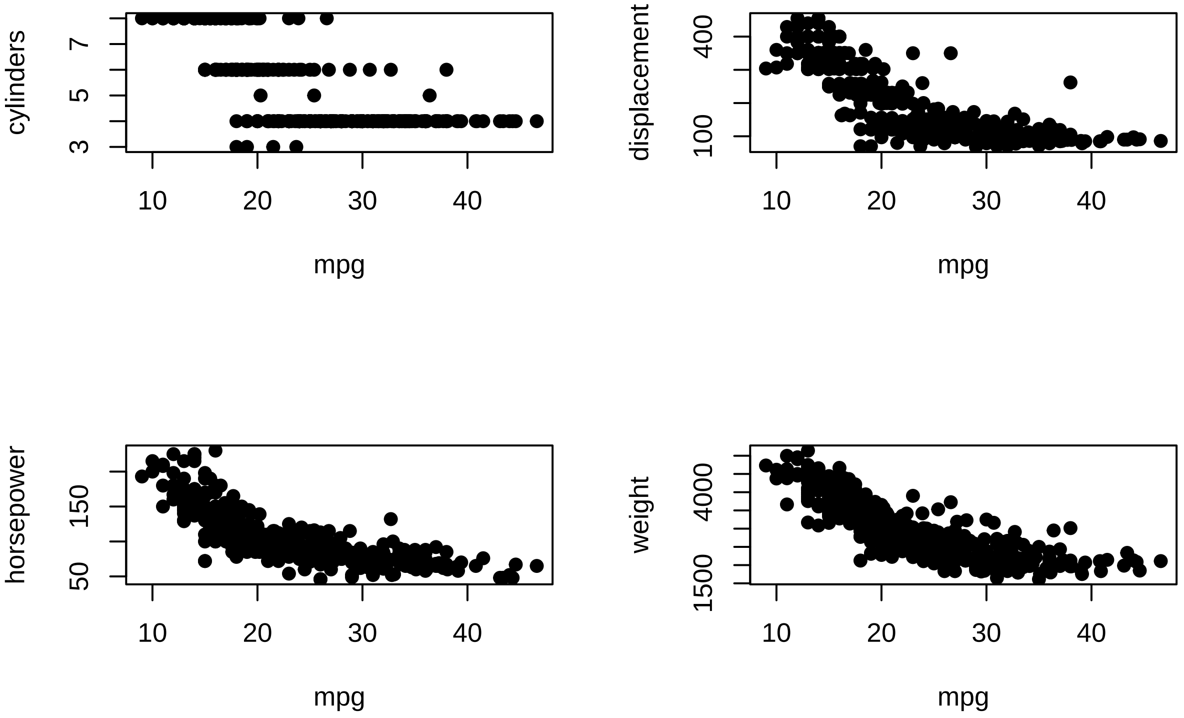


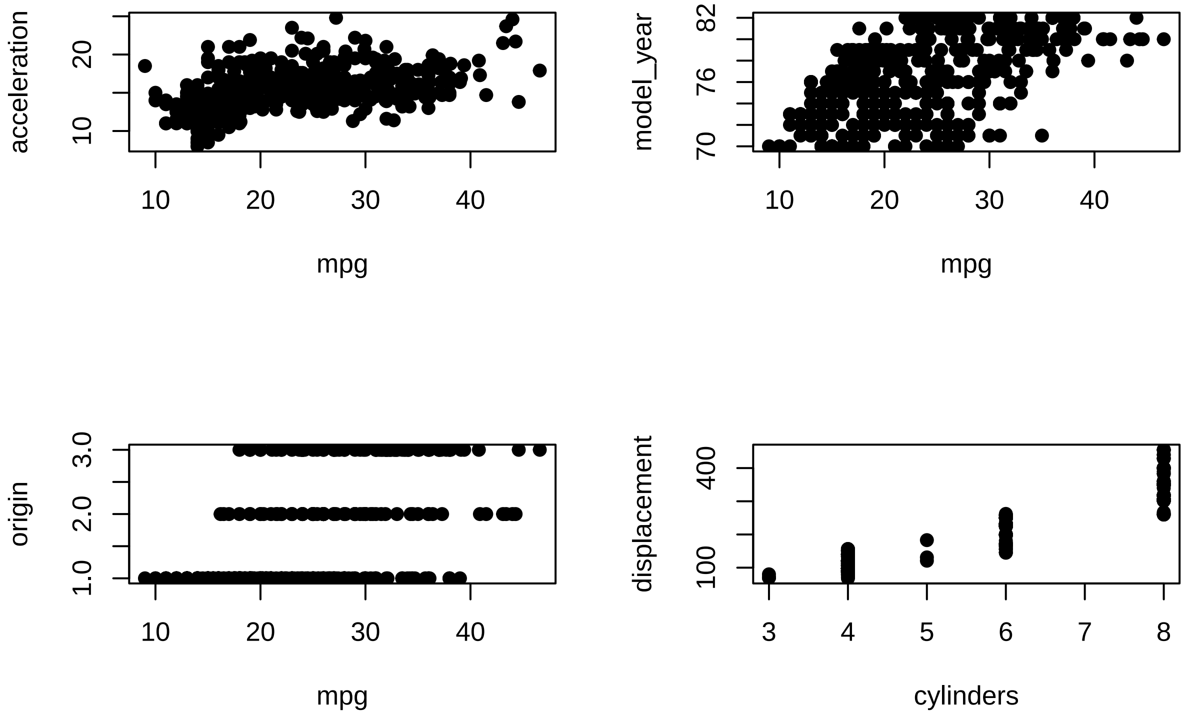
From the correlation and the plotting result of ggpairs, ggapirs did cor.test during the plotting, and the testing result of mpg with other variables are all significant, so I will say cylinders, displacement, horsepower, weight, acceleration, model_year, origin are all appear to relate to mpg.

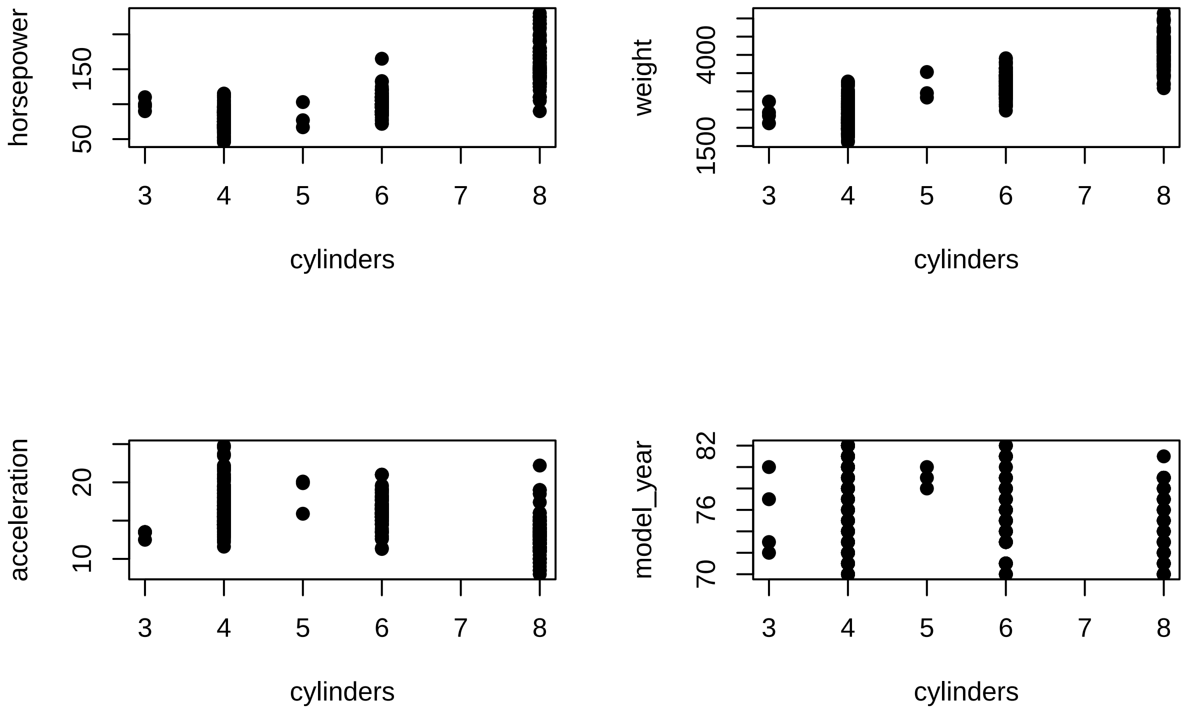
```
iv) Which relationships might not be linear? (don't worry about linearity for rest of this HW)
```

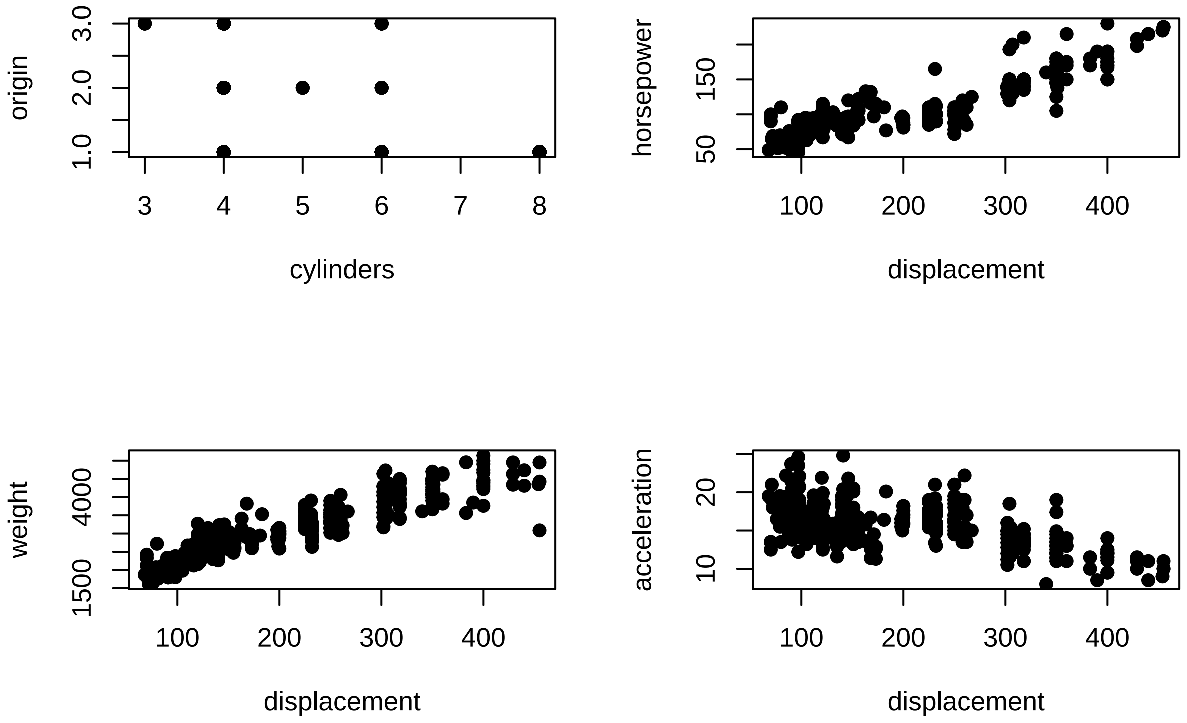
For closer look to find non linear relationships, draw scatter plot against two variables.

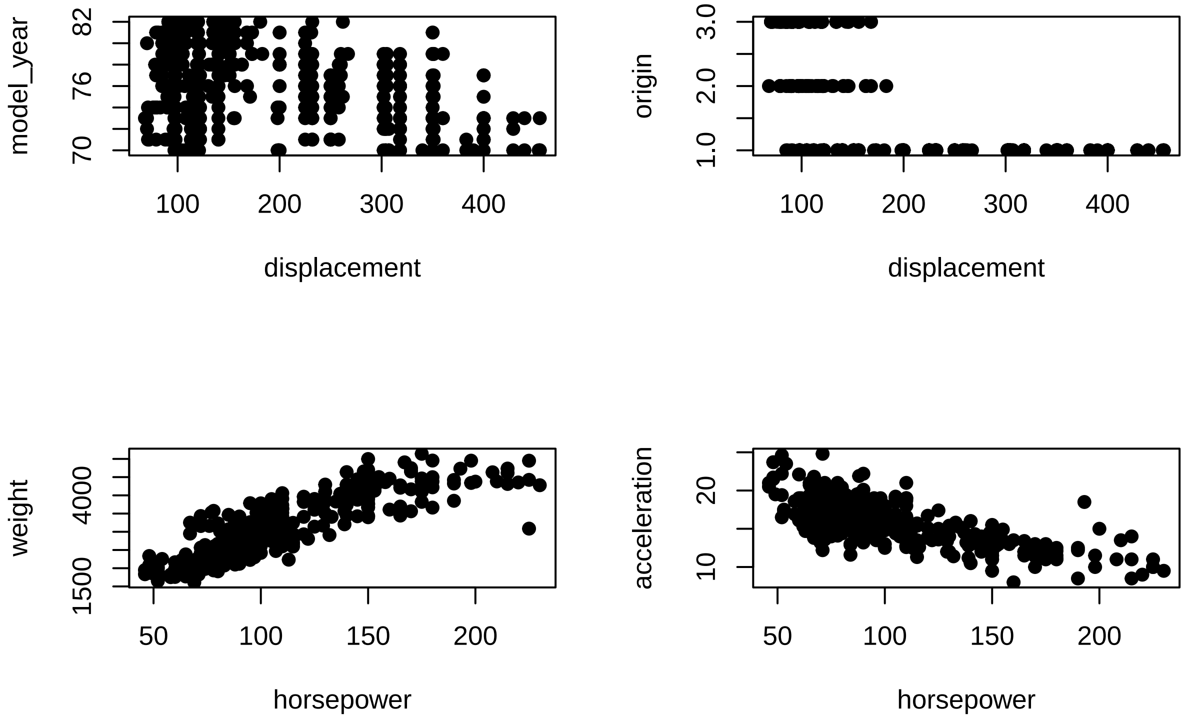
```
par(mfrow=c(2,2))

for (i in seq(1:8)) {
  for (j in seq(1:8)) {
    if(i == j) next
    if(i>j) next
    plot(auto[,i],auto[,j],xlab = (colnames(auto)[i]),ylab = (colnames(auto)[j]), pch=19)
  }
}
```

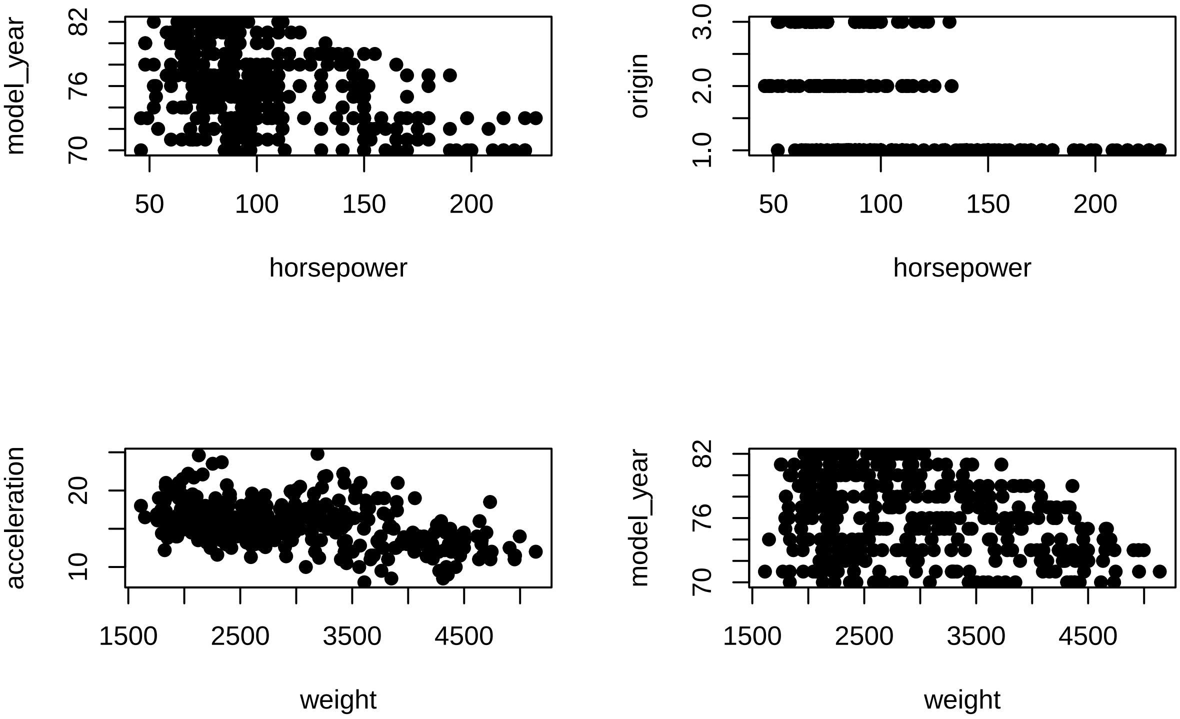

cylinders
3
5
7
10
20
30
40
mpg
displacement
100
400
10
20
30
40
mpg
horsepower
50
150
10
20
30
40
mpg
weight
1500
4000
10
20
30
40
mpg

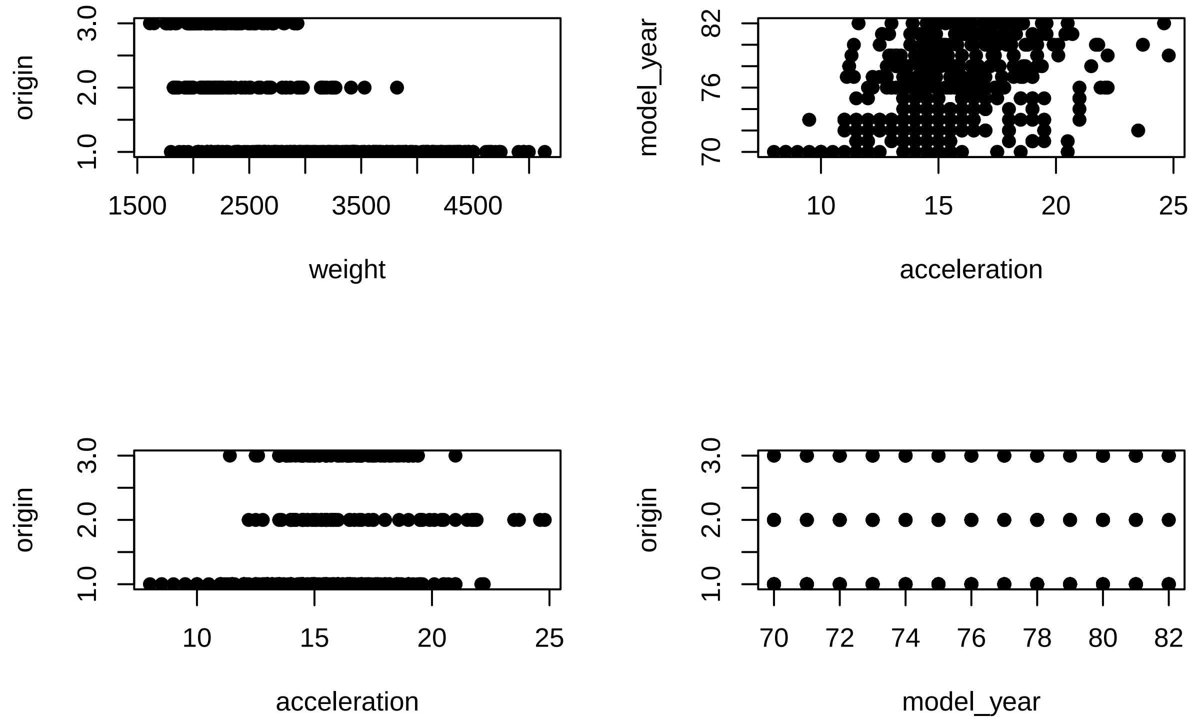

acceleration
10
20
10
20
30
40
mpg
model_year
70
76
82
10
20
30
40
mpg
origin
1.0
2.0
3.0
10
20
30
40
mpg
displacement
100
400
3
4
5
6
7
8
cylinders


horsepower
50
150
3
4
5
6
7
8
cylinders
weight
1500
4000
3
4
5
6
7
8
cylinders
acceleration
10
20
3
4
5
6
7
8
cylinders
model_year
70
76
82
3
4
5
6
7
8
cylinders


origin
1.0
2.0
3.0
3
4
5
6
7
8
cylinders
horsepower
50
150
100
200
300
400
displacement
weight
1500
4000
100
200
300
400
displacement
acceleration
10
20
100
200
300
400
displacement


model_year
70
76
82
displacement
100
200
300
400
origin
1.0
2.0
3.0
displacement
100
200
300
400
weight
1500
4000
horsepower
50
100
150
200
acceleration
10
20
horsepower
50
100
150
200


model_year
70
76
82
50
100
150
200
horsepower
origin
1.0
2.0
3.0
50
100
150
200
horsepower
acceleration
10
20
1500
2500
3500
4500
weight
model_year
70
76
82
1500
2500
3500
4500
weight



```r
par(mfrow=c(1,1))
```

From plot "mpg and displacement", "mpg and horsepower", "mpg and weight", "mpg and acceleration", there may be quadratic relation.

From plot "displacement and origin", "horsepower and origin", "weight and origin" there may be quadratic relation.

From plot "horsepower and weight", "horsepower and acceleration", there may be quadratic relation.

```
    v) Are there any pairs of independent variables that are highly correlated (r > 0.7)?
```

```r
round(cor(auto[,-9],use="pairwise.complete.obs"),2)>0.7
```

```
##                mpg cylinders displacement horsepower weight acceleration
## mpg           TRUE     FALSE        FALSE      FALSE  FALSE        FALSE
## cylinders    FALSE      TRUE         TRUE       TRUE   TRUE        FALSE
## displacement FALSE      TRUE         TRUE       TRUE   TRUE        FALSE
## horsepower   FALSE      TRUE         TRUE       TRUE   TRUE        FALSE
## weight       FALSE      TRUE         TRUE       TRUE   TRUE        FALSE
## acceleration FALSE     FALSE        FALSE      FALSE  FALSE         TRUE
## model_year   FALSE     FALSE        FALSE      FALSE  FALSE        FALSE
## origin       FALSE     FALSE        FALSE      FALSE  FALSE        FALSE
##              model_year origin
## mpg               FALSE  FALSE
## cylinders         FALSE  FALSE
## displacement      FALSE  FALSE
## horsepower        FALSE  FALSE
## weight            FALSE  FALSE
## acceleration      FALSE  FALSE
## model_year         TRUE  FALSE
## origin            FALSE   TRUE
```

Yes, there are many pairs of independent variables that can beconsider as highly correlated. "cylinders and displacement", "cylinders and horsepower", "cylinders and weight" "displacement and horsepower", "displacement and weight" "horsepower and weight"

b) Let's create a linear regression model where mpg is dependent upon all other suitable variables (Note: origin is categorical with three levels, so use factor(origin) in lm(...) to split it into two dummy variables)

```
    i) Which independent variables have a 'significant' relationship with mpg at 1% significance?
```

```
md2 <- lm(mpg~cylinders+displacement+horsepower+weight++acceleration+model_year+factor(origin), data = a
summary(md2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     +acceleration + model_year + factor(origin), data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders       -4.897e-01  3.212e-01  -1.524 0.128215
## displacement     2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower      -1.818e-02  1.371e-02  -1.326 0.185488
## weight          -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration     7.910e-02  9.822e-02   0.805 0.421101
## model_year       7.770e-01  5.178e-02  15.005  < 2e-16 ***
## factor(origin)2  2.630e+00  5.664e-01   4.643 4.72e-06 ***
## factor(origin)3  2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
##   (   6      )
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

Find displacement, weight, model_year, factor(origin)2, factor(origin)3 have a 'significant' relationship with mpg at 1% significance.

```
ii) Looking at the coefficients, is it possible to determine which independent variables are the mo
```

No, since different variables have different scale of units, it's hard for us to compare which variables are most effective at increasing mpg based on the value of the estimation of beta in the linear .

c) Let's try to resolve some of the issues with our regression model above.

```
i) Create fully standardized regression results: are these slopes easier to compare?
(note: consider if you should standardize origin)
```

Don't standardize origin since it is a categorical data.

```
temp <- as.data.frame(cbind(scale(auto[,c(-8,-9)]),auto[,8]))
md2 <- lm(mpg~cylinders+displacement+horsepower+weight++acceleration+model_year+factor(V8), data = temp
summary(md2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     +acceleration + model_year + factor(V8), data = temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.15270 -0.26593 -0.01257  0.25404  1.70942
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.13323    0.03174  -4.198 3.35e-05 ***
## cylinders    -0.10658    0.06991  -1.524  0.12821
## displacement  0.31989    0.10210   3.133  0.00186 **
## horsepower   -0.08955    0.06751  -1.326  0.18549
## weight       -0.72705    0.07098 -10.243  < 2e-16 ***
## acceleration  0.02791    0.03465   0.805  0.42110
## model_year    0.36760    0.02450  15.005  < 2e-16 ***
## factor(V8)2   0.33649    0.07247   4.643 4.72e-06 ***
## factor(V8)3   0.36505    0.07072   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.423 on 383 degrees of freedom
##   (   6      )
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

Yes, it becomes more easily to compare which variables are most effective at increasing mpg. Even though the meaning for the variable may not be easy to interpret after standardization, but we can still say **weight** has the largest value, -0.72705, in all beta, meaning **weight** is the variable that are most effective at increasing mpg.

```
ii) Regress mpg over each nonsignificant independent variable, individually.
Which ones become significant when we regress mpg over them individually?
```

```
md3 <- lm(mpg~cylinders, data = temp )
summary(md3)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders, data = temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82455 -0.43297 -0.08288  0.32674  2.29046
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.834e-15  3.169e-02    0.00        1
## cylinders   -7.754e-01  3.173e-02  -24.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6323 on 396 degrees of freedom
## Multiple R-squared:  0.6012, Adjusted R-squared:  0.6002
## F-statistic: 597.1 on 1 and 396 DF,  p-value: < 2.2e-16
```

```
md4 <- lm(mpg~horsepower, data = temp )
summary(md4)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73632 -0.41699 -0.04395  0.35351  2.16531
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.008784   0.031701  -0.277    0.782
## horsepower  -0.777334   0.031742 -24.489   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6277 on 390 degrees of freedom
##   (    6      )
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
md5 <- lm(mpg~acceleration, data = temp )
summary(md5)
```

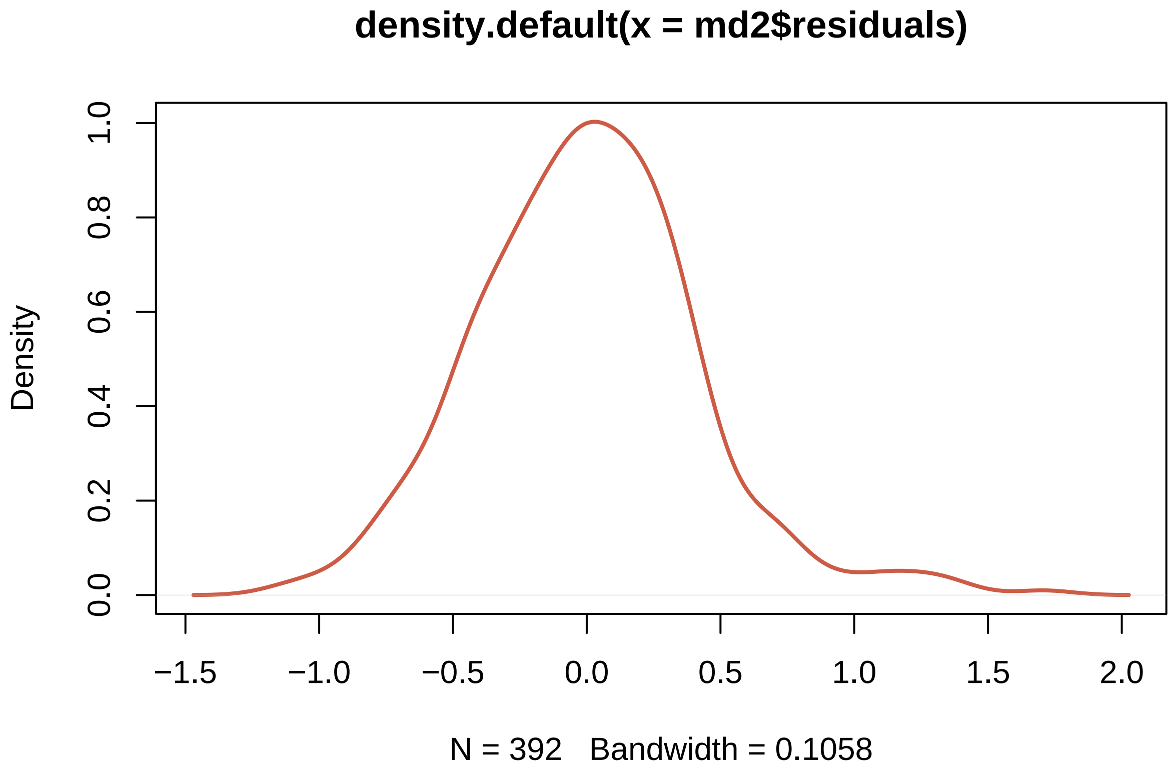
```
##
## Call:
## lm(formula = mpg ~ acceleration, data = temp)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3039 -0.7210 -0.1589  0.6087  2.9672
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.004e-16  4.554e-02   0.000        1
## acceleration 4.203e-01  4.560e-02   9.217   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9085 on 396 degrees of freedom
## Multiple R-squared:  0.1766, Adjusted R-squared:  0.1746
## F-statistic: 84.96 on 1 and 396 DF,  p-value: < 2.2e-16
```

The three variables all become significant

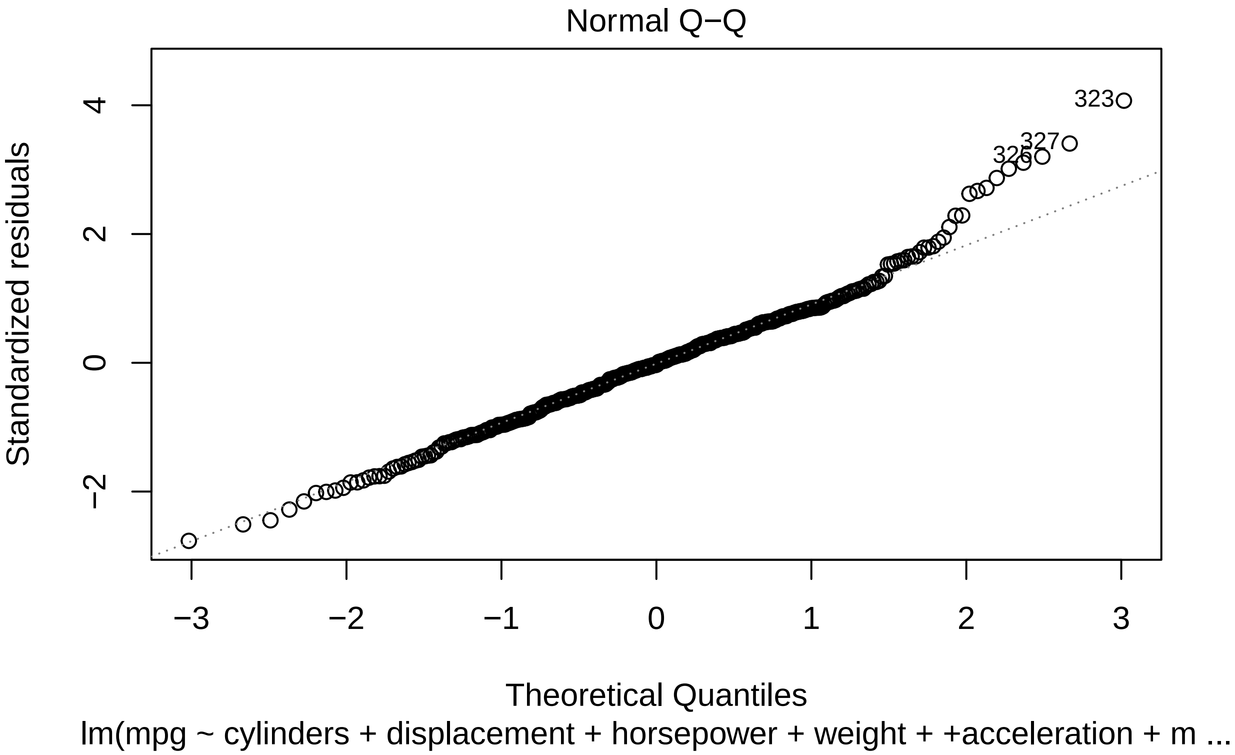
```
    iii) Plot the distribution of the residuals: are they normally distributed and centered around zero
    (get the residuals of a fitted linear model, e.g. regr <- lm(...), using regr$residuals
```

```
plot(density(md2$residuals), col="coral3", lwd=2)
```



See a bell shape centered at zero with some skewness. However it may not be considered as normally distributed.

```r
plot(md2, which = c(2,2))
```



From Q-Q plot, find the distribution is right skew.

```r
shapiro.test(md2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  md2$residuals
## W = 0.98243, p-value = 0.0001061
```

The residuals did not pass the normality test, so the distribution of the residuals are not normally distributed and centered around zero.