gogogo

Xiancheng Lin

12/1/2019

```
library(car)
```

```
## Loading required package: carData
```

```
library(stringr)
library(leaps)
library(MASS)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
data<-read.table("auto-mpg.txt",col.names = c("mpg","cylinders","displacement",
                                              "horsepower","weight","acceleration",
                                              "year","origin","name"),
                 colClasses = c("numeric","integer","numeric",
                                "numeric","numeric","numeric","integer","factor","character"),na.strings

data[,9]=word(data[,9],1)

### description of our data
head(data)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##        name
## 1 chevrolet
## 2     buick
## 3  plymouth
## 4       amc
## 5      ford
## 6      ford
```

```
dim(data)
```

```
## [1] 398   9
```

```
summary(data)
```

```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0
##  Median :23.00   Median :4.000   Median :148.5   Median : 93.5
##  Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
```

```
##                                                NA's   :6
##      weight      acceleration        year       origin      name
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   1:249   Length:398
##  1st Qu.:2224   1st Qu.:13.82   1st Qu.:73.00   2: 70   Class :character
##  Median :2804   Median :15.50   Median :76.00   3: 79   Mode  :character
##  Mean   :2970   Mean   :15.57   Mean   :76.01
##  3rd Qu.:3608   3rd Qu.:17.18   3rd Qu.:79.00
##  Max.   :5140   Max.   :24.80   Max.   :82.00
##
```

```
###count the numbers of data group by predictor "name"
aggregate(mpg~name,data=data,length)
```

```
##             name mpg
## 1            amc  28
## 2           audi   7
## 3            bmw   2
## 4          buick  17
## 5       cadillac   2
## 6          capri   1
## 7      chevroelt   1
## 8      chevrolet  43
## 9          chevy   3
## 10      chrysler   6
## 11        datsun  23
## 12         dodge  28
## 13          fiat   8
## 14          ford  51
## 15            hi   1
## 16         honda  13
## 17         maxda   2
## 18         mazda  10
## 19      mercedes   1
## 20 mercedes-benz   2
## 21       mercury  11
## 22        nissan   1
## 23    oldsmobile  10
## 24          opel   4
## 25       peugeot   8
## 26      plymouth  31
## 27       pontiac  16
## 28       renault   5
## 29          saab   4
## 30        subaru   4
## 31        toyota  25
## 32       toyouta   1
## 33       triumph   1
## 34     vokswagen   1
## 35    volkswagen  15
## 36         volvo   6
## 37            vw   6
```

```
### according to the result,we do not have enough dataset for each group, so we delete this predictor

data=data[,-9]
```

```r
aggregate(mpg~year,data=data,length)
```

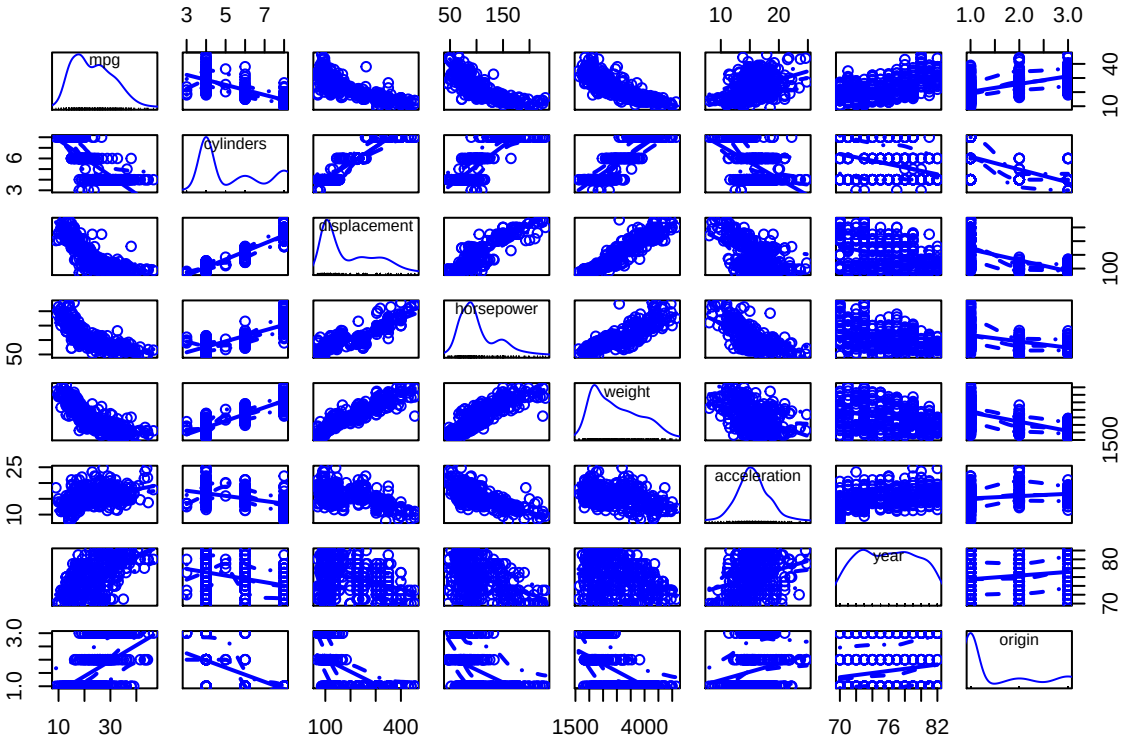
```
##    year mpg
## 1    70  29
## 2    71  28
## 3    72  28
## 4    73  40
## 5    74  27
## 6    75  30
## 7    76  34
## 8    77  28
## 9    78  36
## 10   79  29
## 11   80  29
## 12   81  29
## 13   82  31
```

```r
aggregate(mpg~year+origin,data=data,length)  ###not enough dataset for each group
```

```
##    year origin mpg
## 1    70      1  22
## 2    71      1  20
## 3    72      1  18
## 4    73      1  29
## 5    74      1  15
## 6    75      1  20
## 7    76      1  22
## 8    77      1  18
## 9    78      1  22
## 10   79      1  23
## 11   80      1   7
## 12   81      1  13
## 13   82      1  20
## 14   70      2   5
## 15   71      2   4
## 16   72      2   5
## 17   73      2   7
## 18   74      2   6
## 19   75      2   6
## 20   76      2   8
## 21   77      2   4
## 22   78      2   6
## 23   79      2   4
## 24   80      2   9
## 25   81      2   4
## 26   82      2   2
## 27   70      3   2
## 28   71      3   4
## 29   72      3   5
## 30   73      3   4
## 31   74      3   6
## 32   75      3   4
## 33   76      3   4
## 34   77      3   6
```

```
## 35   78       3    8
## 36   79       3    2
## 37   80       3   13
## 38   81       3   12
## 39   82       3    9
```

```
#data=data[-which(data[,4]=="?"),]
#data[,4]=as.numeric(data[,4])
####plot
scatterplotMatrix(data)
```

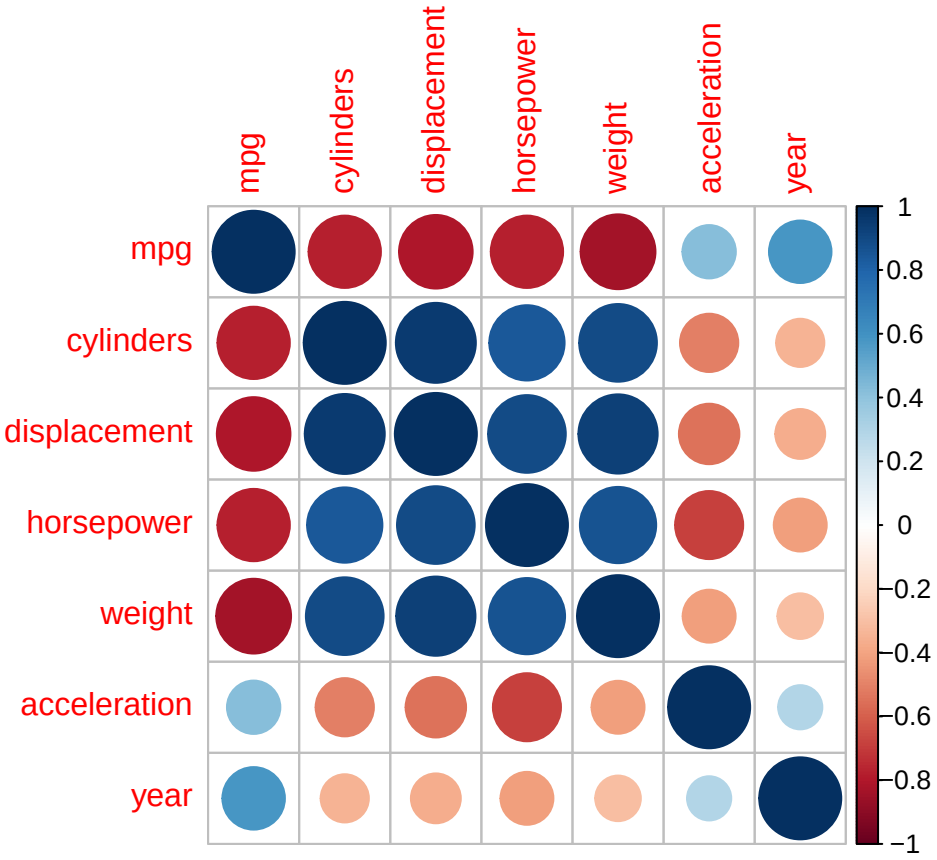



```
##we can see that the predictors and response are not symmetric, so we can consider variables
##transformation later on

###cor matrix
cor(data[,1:7])
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7753963   -0.8042028         NA -0.8317409
## cylinders    -0.7753963  1.0000000    0.9507214         NA  0.8960168
## displacement -0.8042028  0.9507214    1.0000000         NA  0.9328241
## horsepower           NA         NA           NA          1         NA
## weight       -0.8317409  0.8960168    0.9328241         NA  1.0000000
## acceleration  0.4202889 -0.5054195   -0.5436841         NA -0.4174573
## year          0.5792671 -0.3487458   -0.3701642         NA -0.3065643
##              acceleration       year
## mpg             0.4202889  0.5792671
```

```
## cylinders      -0.5054195 -0.3487458
## displacement   -0.5436841 -0.3701642
## horsepower            NA         NA
## weight         -0.4174573 -0.3065643
## acceleration    1.0000000  0.2881370
## year            0.2881370  1.0000000
```

```
corrplot(cor(na.omit(data[,-8])))
```




```
####model selection
data=data[,-c(7,8)]
regsubsets<-regsubsets(mpg~.,data=data)
sumreg<-summary(regsubsets)
sumreg
```

```
## Subset selection object
## Call: regsubsets.formula(mpg ~ ., data = data)
## 5 Variables  (and intercept)
##              Forced in Forced out
## cylinders        FALSE      FALSE
## displacement     FALSE      FALSE
## horsepower       FALSE      FALSE
## weight           FALSE      FALSE
## acceleration     FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          cylinders displacement horsepower weight acceleration
```

```
## 1  ( 1 ) " "       " "          " "        "*"    " "
## 2  ( 1 ) " "       " "          "*"        "*"    " "
## 3  ( 1 ) "*"       " "          "*"        "*"    " "
## 4  ( 1 ) "*"       " "          "*"        "*"    "*"
## 5  ( 1 ) "*"       "*"          "*"        "*"    "*"
```

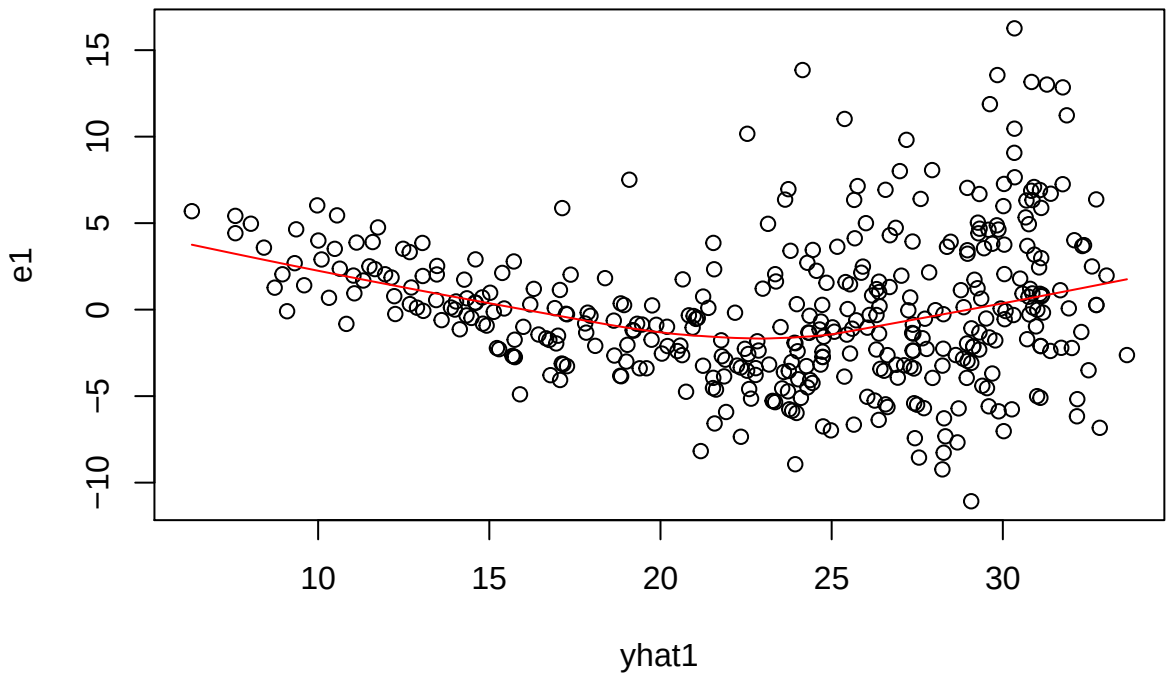
```
data.frame(adjr2=sumreg$adjr2,cp=sumreg$cp,bic=sumreg$bic)
```

```
##       adjr2        cp       bic
## 1 0.6918423 17.890053 -450.5016
## 2 0.7048656  1.739609 -462.4637
## 3 0.7053915  2.053796 -458.2005
## 4 0.7046713  4.000084 -452.2838
## 5 0.7039063  6.000000 -446.3126
```

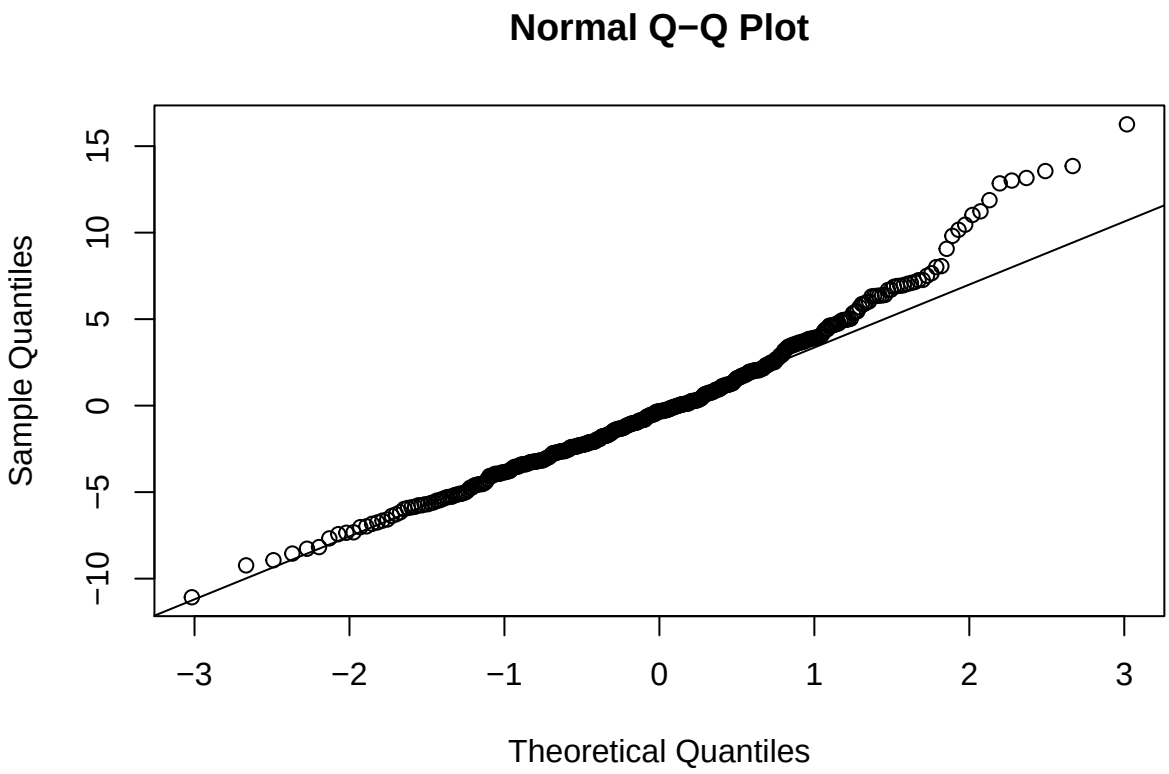
```
###Consider two variables to do regression
lm.fit1<-lm(mpg~horsepower+weight,data=na.omit(data))
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + weight, data = na.omit(data))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0762  -2.7340  -0.3312   2.1752  16.2601
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 45.6402108  0.7931958  57.540  < 2e-16 ***
## horsepower  -0.0473029  0.0110851  -4.267 2.49e-05 ***
## weight      -0.0057942  0.0005023 -11.535  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.24 on 389 degrees of freedom
## Multiple R-squared:  0.7064, Adjusted R-squared:  0.7049
## F-statistic: 467.9 on 2 and 389 DF,  p-value: < 2.2e-16
```

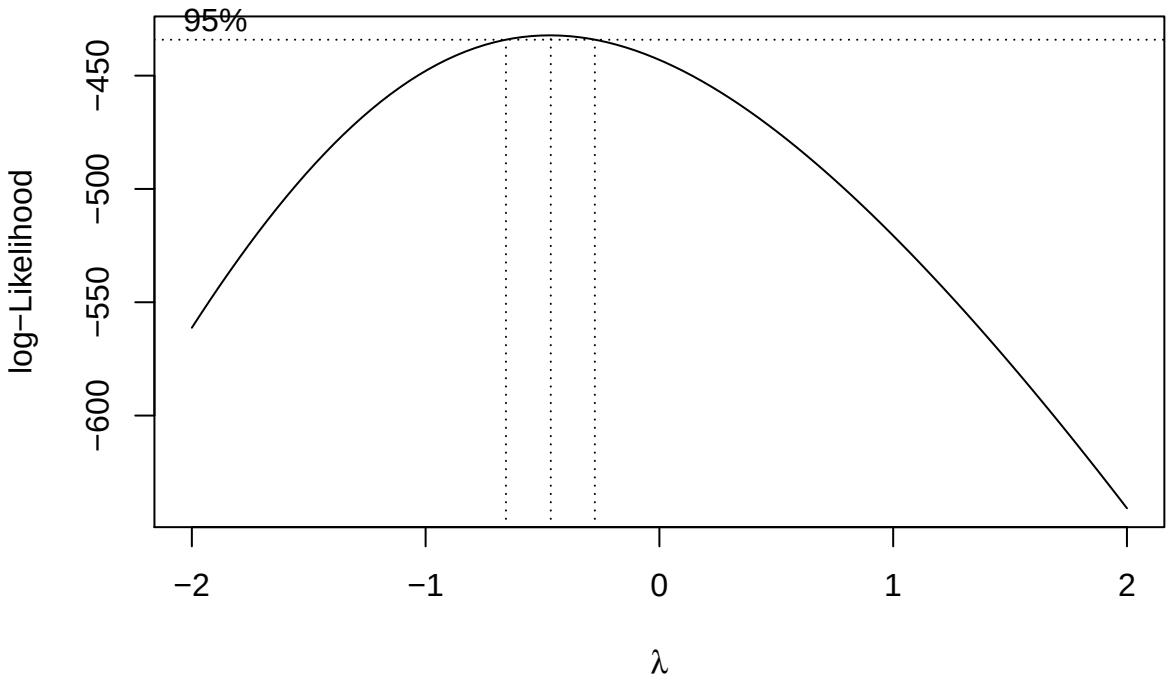
```
###residual analysis
#step1  normality
e1=residuals(lm.fit1)
yhat1=fitted(lm.fit1)
plot(yhat1,e1)
resid.lowess=lowess(yhat1,e1,f=0.8)
lines(resid.lowess,col=2)
```



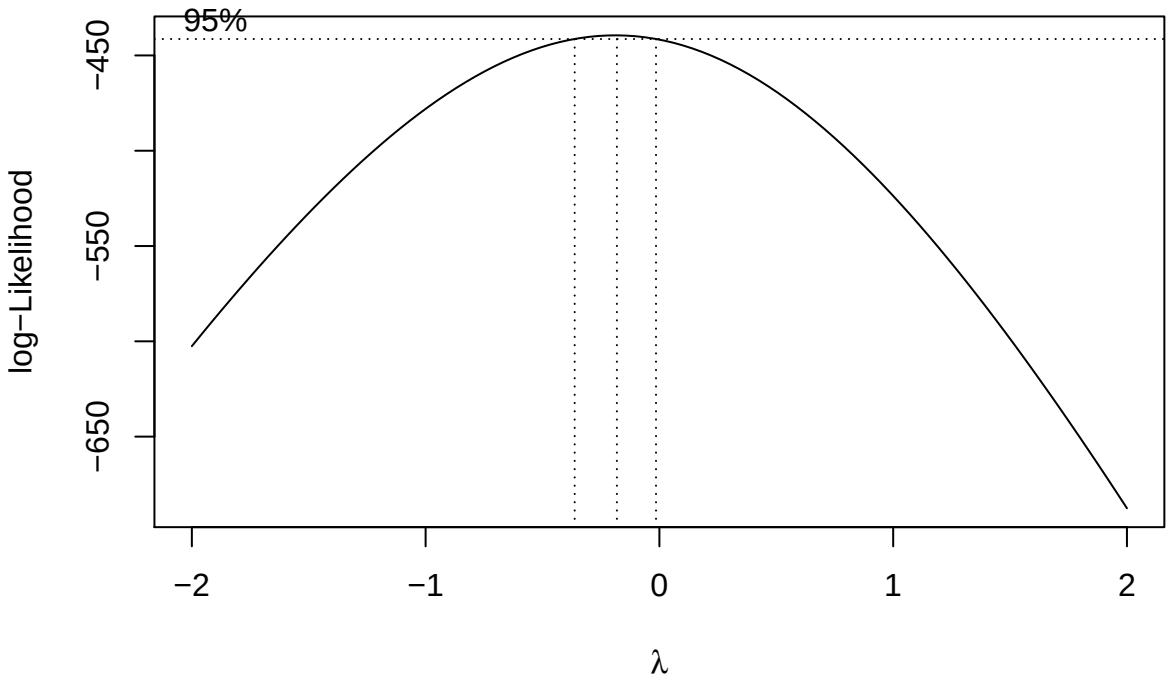

```
###from the plot we can see that the residuals are not linear, and has a tendency of non-linear
qqnorm(e1)
qqline(e1)
```



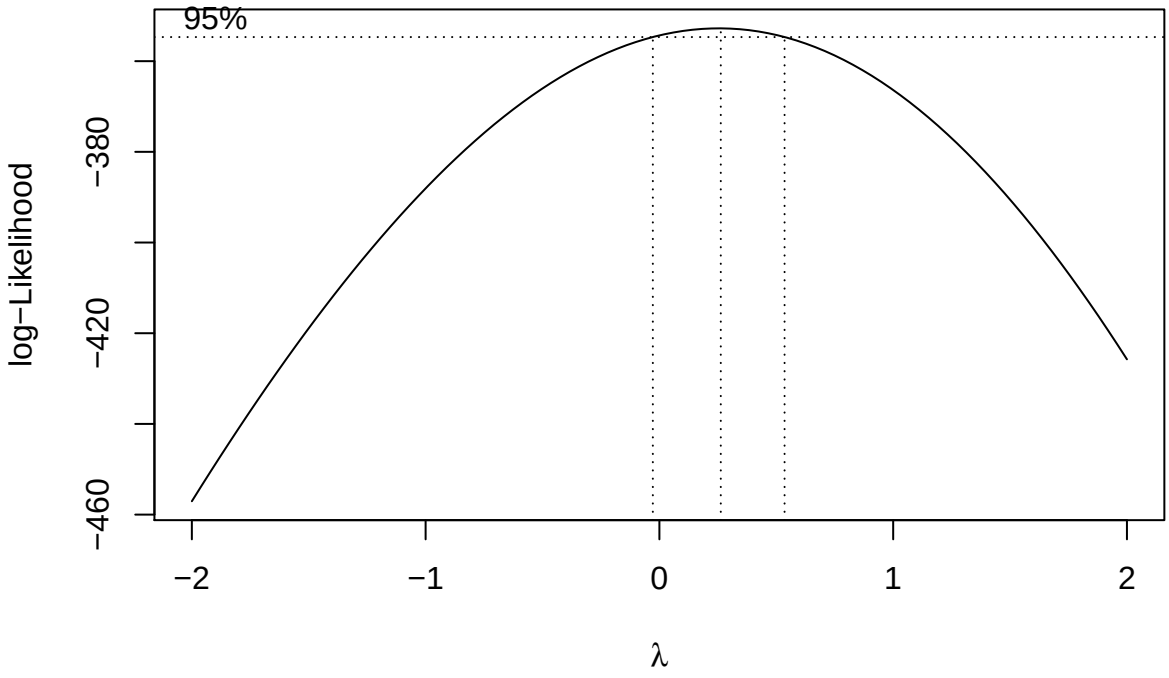

```
boxcox(mpg ~ horsepower+weight,data=data)
```




```
boxcox(horsepower~mpg+weight,data=data)
```




```
boxcox(weight~mpg+horsepower,data=data)
```

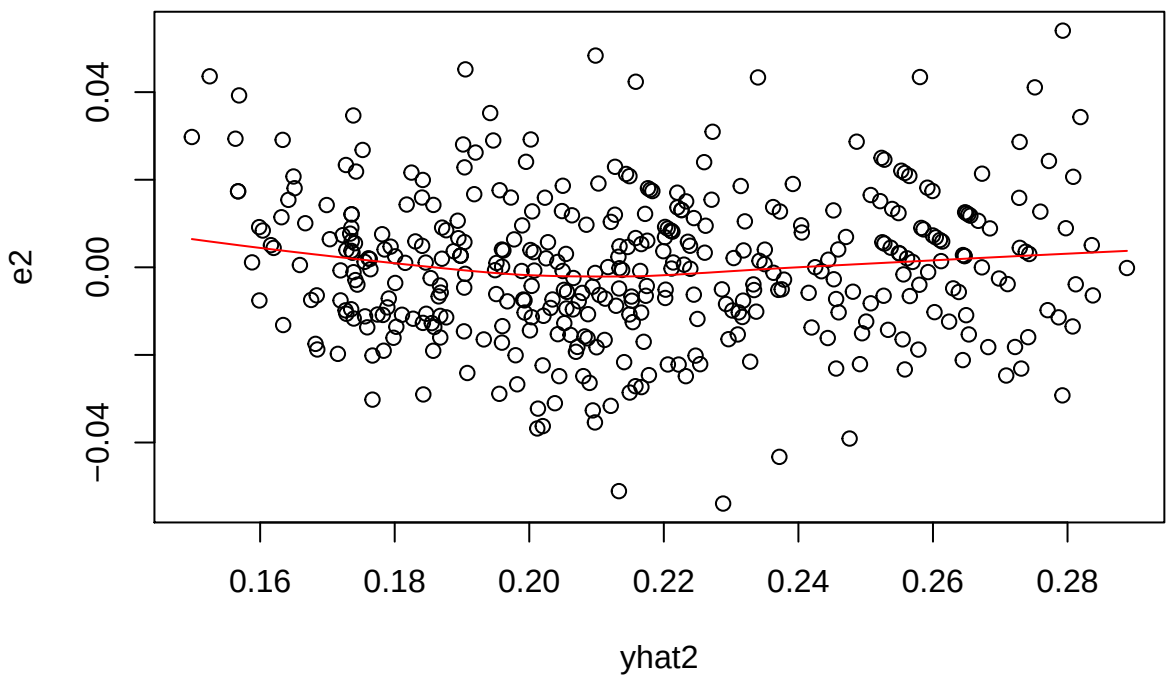



```
lm.fit2=lm(mpg^(-1/2)~log(horsepower)+log(weight),data=na.omit(data))
summary(lm.fit2)
```

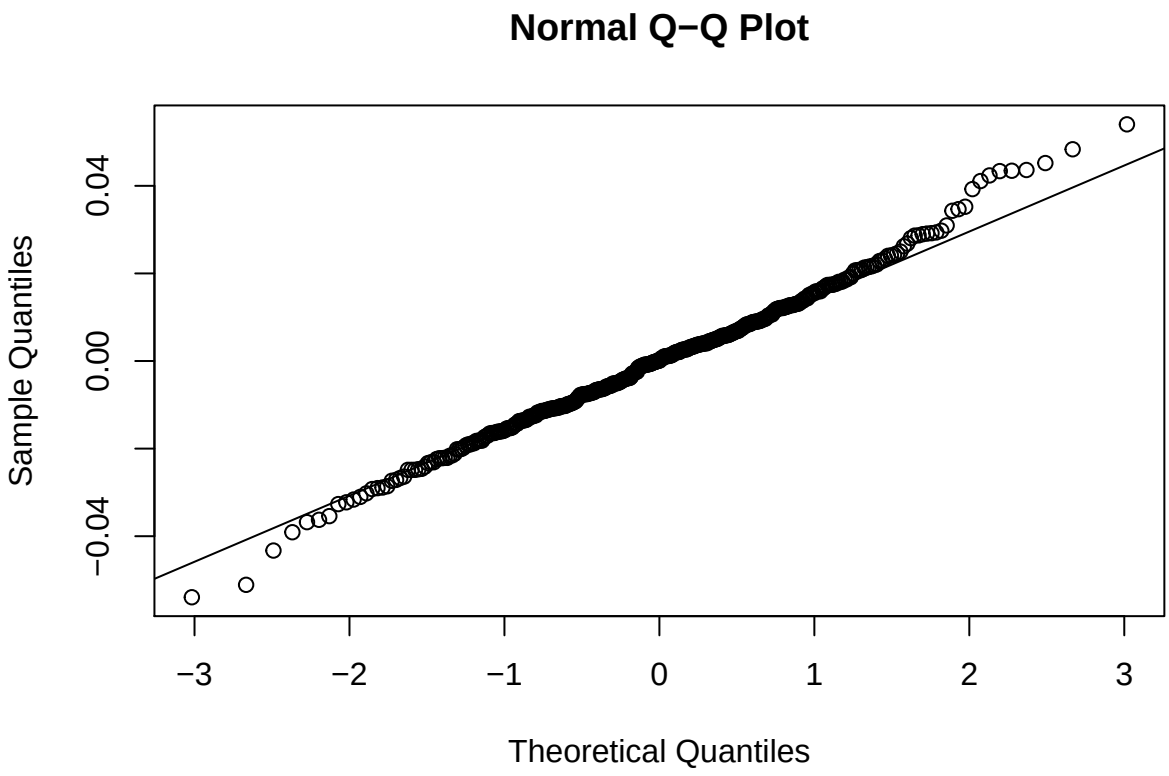
```
##
## Call:
## lm(formula = mpg^(-1/2) ~ log(horsepower) + log(weight), data = na.omit(data))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.053937 -0.010768  0.000105  0.009574  0.054037
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.544097   0.030664 -17.744  < 2e-16 ***
## log(horsepower)  0.040972   0.005001   8.193 3.71e-15 ***
## log(weight)      0.071823   0.006107  11.761  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0165 on 389 degrees of freedom
## Multiple R-squared:  0.8027, Adjusted R-squared:  0.8017
## F-statistic: 791.5 on 2 and 389 DF,  p-value: < 2.2e-16
```

```
e2=residuals(lm.fit2)
yhat2=fitted(lm.fit2)
plot(yhat2,e2)
resid.lowess=lowess(yhat2,e2,f=0.8)
```

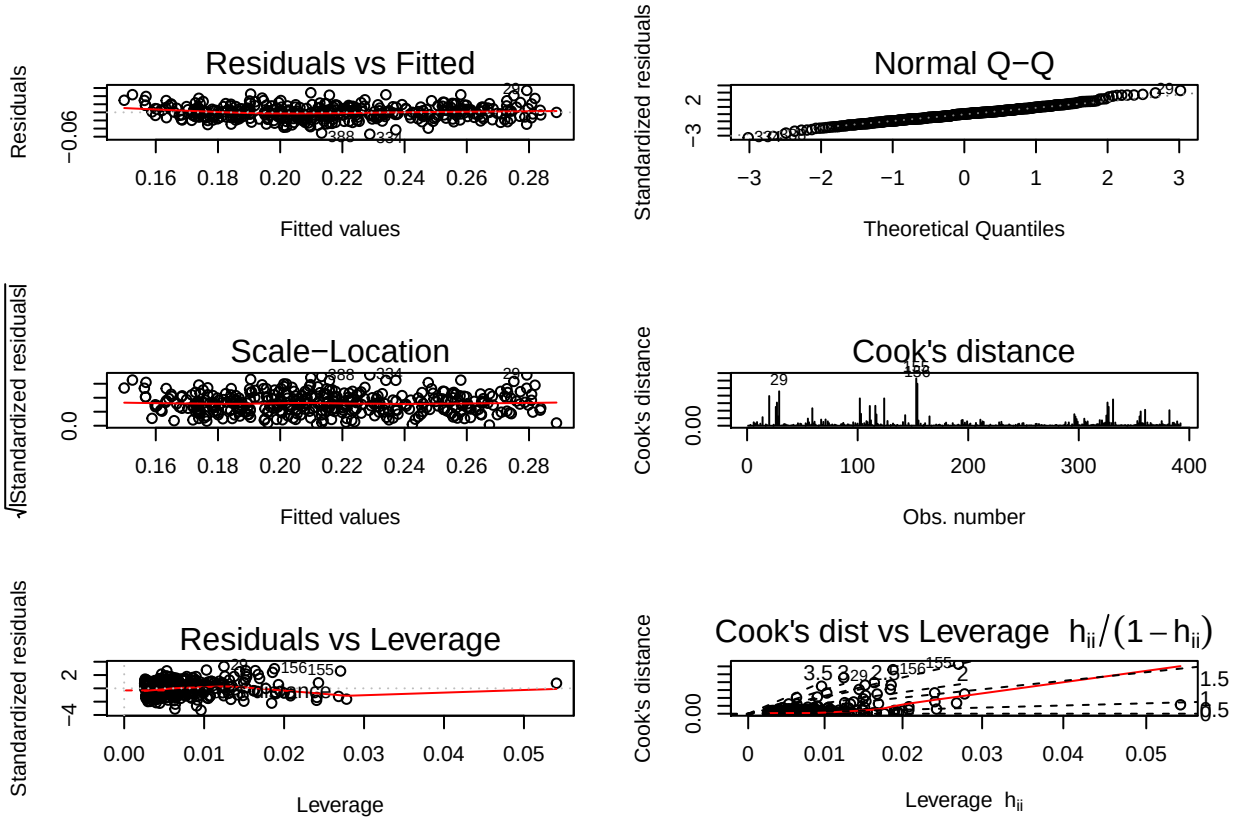
```
lines(resid.lowess,col=2)
```




```
###from the plot we can see that the residuals are not linear, and has a tendency of non-linear
qqnorm(e2)
qqline(e2)
```




```
###influence measure
par(mfrow=c(3,2))
plot(lm.fit2,which=1:6)
```




```
mean(abs(lm.fit2$residuals))
```

```
## [1] 0.01277699
```

```
#consider the possible high leverage and outlier points 29,155,156

data=data[-c(29,155,156),]
lm.fit3=lm(mpg^(-1/2)~log(horsepower)+log(weight),data=na.omit(data))
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = mpg^(-1/2) ~ log(horsepower) + log(weight), data = na.omit(data))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.054448 -0.010581  0.000463  0.010231  0.044428
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.521151   0.030094 -17.318   <2e-16 ***
## log(horsepower)  0.044272   0.004948   8.947   <2e-16 ***
## log(weight)      0.066989   0.006029  11.112   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01599 on 386 degrees of freedom
```

```
## Multiple R-squared:   0.81,  Adjusted R-squared:  0.8091
## F-statistic:   823 on 2 and 386 DF,  p-value: < 2.2e-16
```

```
mean(abs(lm.fit3$residuals))
```

```
## [1] 0.01250845
```

```
##we can see that after deleting the high influence points, we enhance the adjusted R2
## and reduce the average absolute residuals.
###multilinear
vif(lm.fit3)
```

```
## log(horsepower)     log(weight)
##        4.353742        4.353742
```

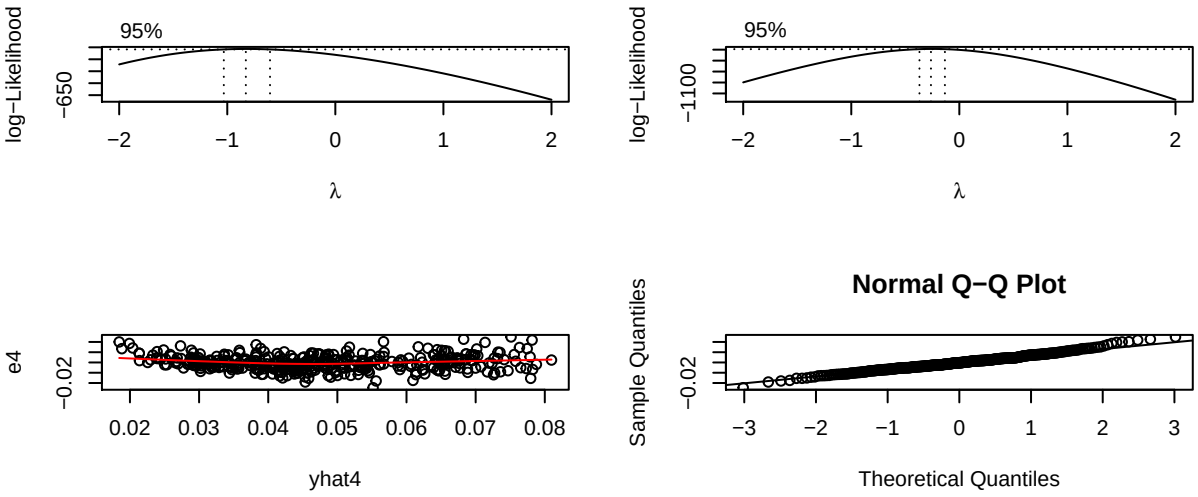
```
#they are both lower than 10, so we believe there is no multilinearlity.



###observe that horsepower and weight have the same transformation and their coefficients
###are almost the same, so consider a new predictor horsepower*weight,named new
data[,7]=data[,4]*data[,5]
colnames(data)[7]="new"
boxcox(mpg~new,data=data) ###inverse transformation
boxcox(new~mpg,data=data) ###log transformation
lm.fit4=lm(1/mpg~log(new),data=na.omit(data))
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = 1/mpg ~ log(new), data = na.omit(data))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0245847 -0.0047885 -0.0003301  0.0042065  0.0248786
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.256436   0.007662  -33.47   <2e-16 ***
## log(new)     0.024233   0.000610   39.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007258 on 387 degrees of freedom
## Multiple R-squared:  0.803,  Adjusted R-squared:  0.8025
## F-statistic:  1578 on 1 and 387 DF,  p-value: < 2.2e-16
```

```
mean(abs(lm.fit4$residuals))
```

```
## [1] 0.00563477
```

```
e4=residuals(lm.fit4)
yhat4=fitted(lm.fit4)
plot(yhat4,e4)
resid.lowess=lowess(yhat4,e4,f=0.8)
lines(resid.lowess,col=2)
qqnorm(e4)
qqline(e4)
```



Compare lm.fit4 to lm.fit3, lm.fit4 reduce the absolute mean of residuals significantly without a much reduce on R2, so maybe lm.f4 is now the best model we get.