Regression Analysis of Fitness Dataset
Instructor : Dr. Swagata Nandi

Adrija Bhar, Srijani Das, Yenisi Das

October 29, 2023

Abstract

We have a data on various measures of heart and pulse rate that were taken on men in a physical fitness course. In the course few tasks were given to the subjects. It has been observed that people who are more fit use less oxygen in those tasks. Our goal is to develop a model that can predict or analyse whether a man is fit or not. We shall the response variable using the explanatory variables. We also have tried to assess how well our model fits or performs and if any problems arise, we tried to overcome that and modify our model accordingly. We mainly tried to implement our knowledge gained from the Regression Techniques course to this real life data.

Contents

1 Introduction **3**

2 Exploratory Data Analysis **3**

3 Regression Analysis **8**
3.1 Fitting Mulitiple Linear Regression Model . . . 8
3.2 Fitting Regression Model by considering quadratic relation of response (OXY) with weight . . . 9
3.3 Comparing the above two fitted model . . . 11

4 Regression Diagnostics **13**
4.1 Checking for different aspects and assumptions of the model . . . 13
4.1.1 Checking for normality assumptions: . . . 13
4.1.2 Checking for homoscedasticity Assumptions . . . 14
4.1.3 Checking for Autocorrelation . . . 17
4.1.4 Detecting Influential Points . . . 20
4.1.5 Checking for Multicollinearity . . . 27
4.2 Remedies for the problems found . . . 29
4.2.1 Removing Influential Points . . . 29
4.2.2 Remedies For Collinearity . . . 31
4.2.3 Handling Non-normality . . . 32

5 Model Selection **37**
5.1 Stepwise Selection . . . 37
5.2 Best Subset Selection . . . 43

6 Alternative Estimation Methods **48**
6.1 Lasso Estimation . . . 48
6.2 Ridge Estimation . . . 51

7 Conclusion **53**

8 Acknowledgement **54**

1 Introduction

✍ Data Description

In the dataset we have observations on 31 men corresponding to the following variables :-

OXY : the outcome measure; people who are more fit use less oxygen in this task
AGE :age of the subjects
WEIGHT : weights of the subjects
RUNTIME : the time to run a given distance on a treadmill
RSTPULSE : measure of pulse rate at rest
RUNPULSE : measure of average during the run
MAXPULSE : measure of maximum during the run

Here our response variable is OXY and others are all covariates.
AGE and WEIGHT are obvious control variables.
To understand the data and to draw conclusions from it, we first perform the exploratory data analysis.

2 Exploratory Data Analysis

✍ Loading the dataset

We first load the dataset in R and save it as a data-frame. Below we have printed first few datapoints from the data using the head() function.

```
library(MASS)
library(lattice)
library(olsrr)
library(ggplot2)
library(lmtest)
library(L1pack)
library(car)
library(ggplot2)
library(faraway)
index <- 1:31
fitness <- read.csv(file = "D:/Regression Project/fitness.csv")
attach(fitness)
data <- cbind.data.frame(index,oxy, age, weight, runtime, restpulse, runpulse, maxpulse)
mydata = data[,-1]
n=31 #total number of observations
p=7  #number of parameters with the intercept term
```

We have total 31 observations and there are no null values.

✍ Type of the variables

We have used the str() functin, to know the type of the each the variables :-

```
str(mydata)

'data.frame': 31 obs. of  7 variables:
 $ oxy      : num  44.6 45.3 54.3 59.6 49.9 ...
 $ age      : int  44 40 44 42 38 47 40 43 44 38 ...
 $ weight   : num  89.5 75.1 85.8 68.2 89 ...
 $ runtime  : num  11.37 10.07 8.65 8.17 9.22 ...
 $ restpulse: int  62 62 45 40 55 58 70 64 63 48 ...
 $ runpulse : int  178 185 156 166 178 176 176 162 174 170 ...
 $ maxpulse : int  182 185 168 172 180 176 180 170 176 186 ...
```

So, our dataset does not contain any categorical variables.

Overall Summary of the dataset

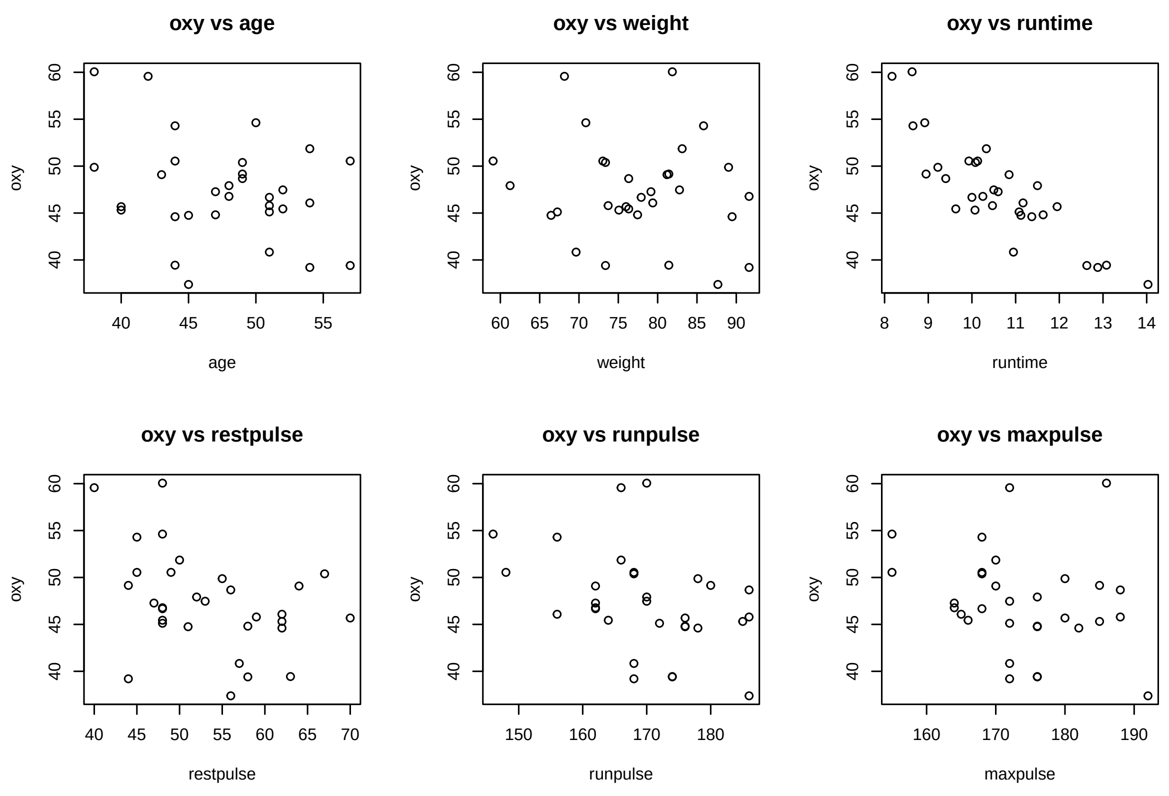
To get an oveview of the data, we have used the summary() function :-

```
summary(mydata)

      oxy             age            weight          runtime
 Min.   :37.39   Min.   :38.00   Min.   :59.08   Min.   : 8.17
 1st Qu.:44.96   1st Qu.:44.00   1st Qu.:73.20   1st Qu.: 9.78
 Median :46.77   Median :48.00   Median :77.45   Median :10.47
 Mean   :47.38   Mean   :47.68   Mean   :77.44   Mean   :10.59
 3rd Qu.:50.13   3rd Qu.:51.00   3rd Qu.:82.33   3rd Qu.:11.27
 Max.   :60.05   Max.   :57.00   Max.   :91.63   Max.   :14.03
   restpulse        runpulse        maxpulse
 Min.   :40.00   Min.   :146.0   Min.   :155.0
 1st Qu.:48.00   1st Qu.:163.0   1st Qu.:168.0
 Median :52.00   Median :170.0   Median :172.0
 Mean   :53.45   Mean   :169.6   Mean   :173.8
 3rd Qu.:58.50   3rd Qu.:176.0   3rd Qu.:180.0
 Max.   :70.00   Max.   :186.0   Max.   :192.0
```

Scatterplots of response (OXY) with the covariates

To get an idea about the relationship between the covariates and response, we make scatterplots using plot() function :-

```r
par(mfrow=c(2,3))
for(i in 2:7){
  plot(mydata[,i], mydata[,1], xlab=colnames(mydata)[i], ylab=colnames(mydata)[1],
  main=paste(colnames(mydata)[1], "vs", colnames(mydata)[i]))
}
```

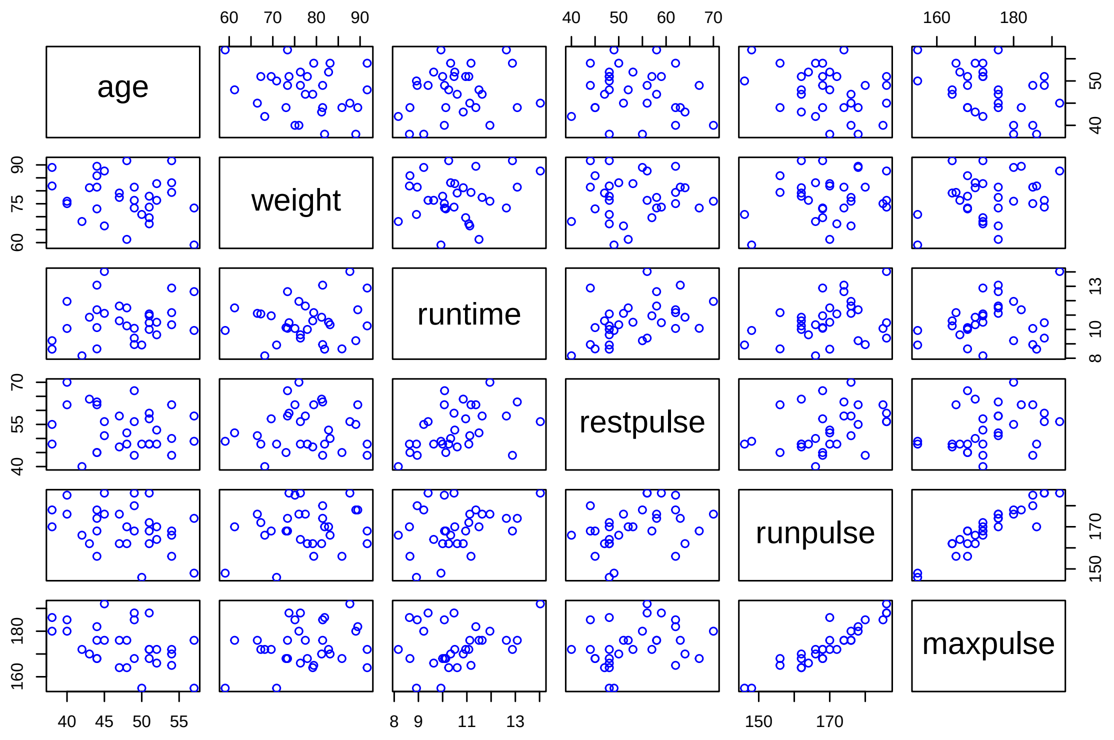


- ***From the plot, we observe that our response, oxy seems to be linearly related to mainly runtime.***
- ***From the plot, we also observe that oxy possibly has some quadratic relation with weight. So, we shall explore it further.***

✍ Pairwise Scatterplot of the covariates

To get an idea about the relationship between the covariates and response and also between the covariates, we make pairwise scatterplots using pairs() function :-

```
pairs(mydata[,-1],col="blue")
```



- *From the plot, there seems to be a correlation between the covariates runpulse and maxpulse. This could possibly lead to the problem of multicollinearity which we will diagnose in due course.*

✍ Renaming the variables

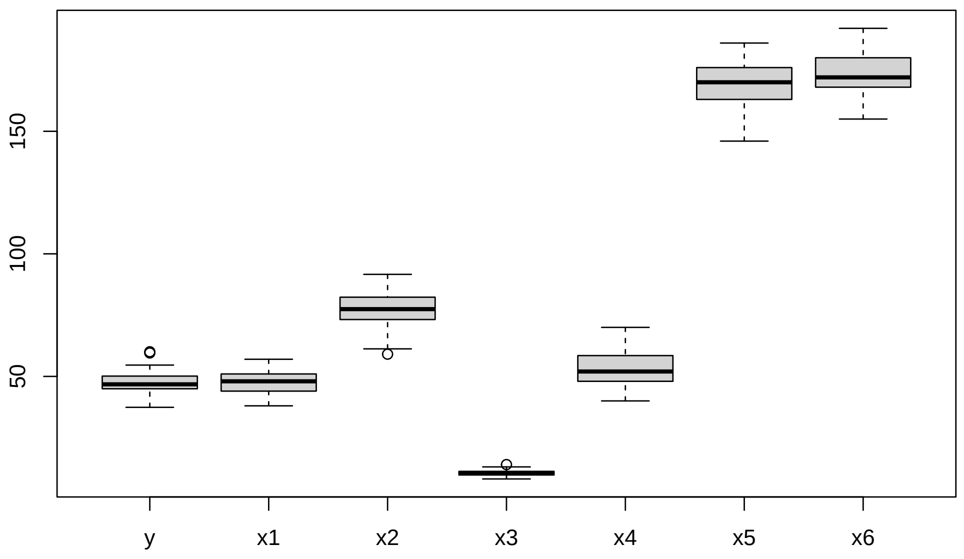
For our conveinience we have renamed the variables as following :
y = oxy, and x_1 : age, x_2 = weight, x_3 = runtime, x_4 = restpulse, x_5 = runpulse, x_6 = maxpulse
Here our response variable is y and $x_1, x_2, x_3, x_4, x_4, x_5, x_6$ are covariates.
After changing the column names our data looks like the following :

```
colnames(mydata) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6")
#y = oxy, x1 = age, x2 = weight, x3 = runtime, x4 = restpulse, x5 = runpulse, x6 = maxpulse
head(mydata)

       y x1    x2    x3 x4  x5  x6
1 44.609 44 89.47 11.37 62 178 182
2 45.313 40 75.07 10.07 62 185 185
3 54.297 44 85.84  8.65 45 156 168
4 59.571 42 68.15  8.17 40 166 172
5 49.874 38 89.02  9.22 55 178 180
6 44.811 47 77.45 11.63 58 176 176

attach(mydata)
```

✍ Boxplots of Covariates and Response

We draw the boxplots for differnet covariates to see whether there are any outlier / high leavarage points or not :-

```
boxplot(mydata)
```



- ***From the plots we observe presence of such points which are possibly outliers/ high leverage points.***
- ***From inintial exploratory data analysis these points do not seem very far from the bulk of the data and since we have very few data points, we shall proceed with detection and handling of influential points after model selection.***

3 Regression Analysis

3.1 Fitting Mulitiple Linear Regression Model

There are no categorical variables in the data. Hence, we start with the very basic model and try to fit a multiple linear regression model with all the available covariates, with an intercept term :

Our model under consideration is $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \epsilon_i, i = 1, 2, .., n,$ (Model-1) with the assumptions:

1. The errors are normally distributed with mean 0.
2. The errors are homoscedastic with common variance σ^2 (unknown).
3. ϵ_i's are uncorrelated.

$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$ are the unknown parameters to be estimated. In vector-matrix notation the model is :

$$\boldsymbol{Y}^{n\times 1} = \boldsymbol{X}^{n\times p}\boldsymbol{\beta}^{p\times 1} + \boldsymbol{\epsilon}^{n\times 1}$$

where $n = 31$ (total number of observed responses) and $p = 7$ where the regression matrix,

$$\mathbf{X} = [\mathbf{1_n} \quad \mathbf{x_1} \quad \mathbf{x_2} \quad \mathbf{x_3} \quad \mathbf{x_4} \quad \mathbf{x_5} \quad \mathbf{x_6}]$$

and parameter vector, $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)'$.

✍ Summary of the fitted model

We fit the linear model stated above in the dataset using lm() function and to get overall idea about the estimates we use the summary() function :-

```
model <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(model)


Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4026 -0.8991  0.0706  1.0496  5.3847 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
x1           -0.22697    0.09984  -2.273  0.03224 *  
x2           -0.07418    0.05459  -1.359  0.18687    
x3           -2.62865    0.38456  -6.835 4.54e-07 ***
x4           -0.02153    0.06605  -0.326  0.74725    
x5           -0.36963    0.11985  -3.084  0.00508 ** 
x6            0.30322    0.13650   2.221  0.03601 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.317 on 24 degrees of freedom
Multiple R-squared:  0.8487,Adjusted R-squared:  0.8108 
F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09
```

Here we observe that the null hypothesis for test of significance for the x_2, x_4 components respectively $H_0 : \beta_2 = 0, H_0 : \beta_4 = 0$ are getting accepted at level 0.05. But before we draw conclusion about this,we inspect some aspects of our model

✍ Observations from the fitted model

We observe that we do not have enough evidence to reject the null hypothesis $H_0 : \beta_2 = 0, H_0 : \beta_4 = 0$ for test of significance for the x_2, x_4 components respectively therefore we connot reject the null at level 0.05. So, we see the coefficients $\beta_1, \beta_3, \beta_5, \beta_6$ for covariates x_1, x_3, x_5, x_6 are not statistically insignificant.

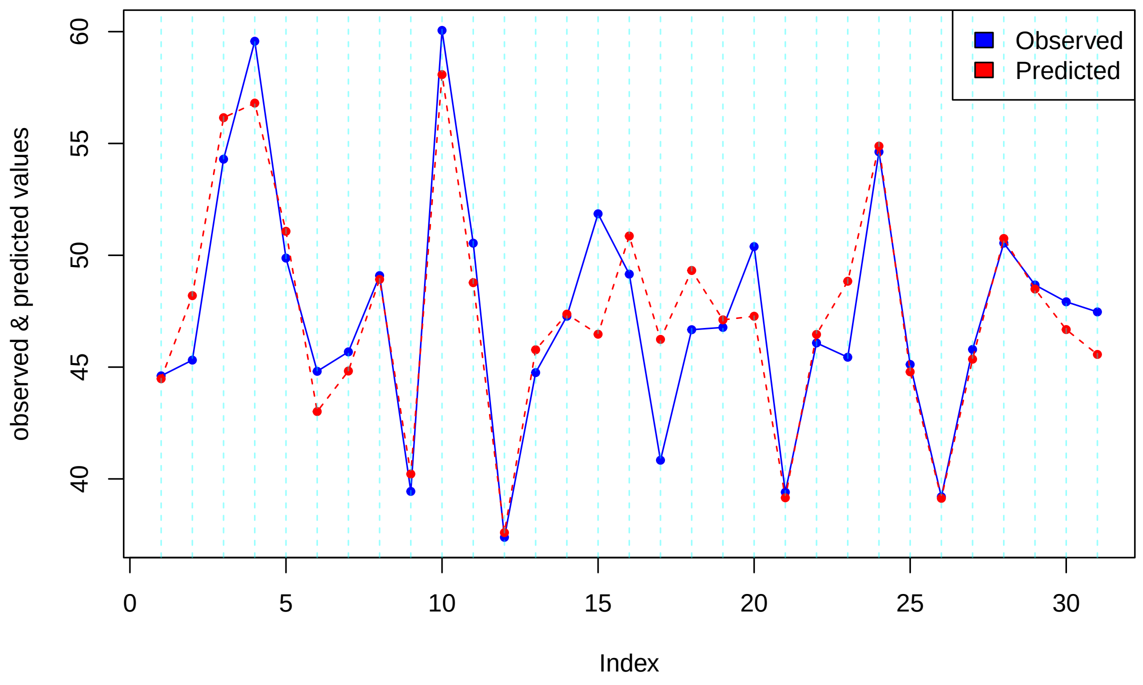
This does not imply that other covariates are insignificant since there maybe some problems, hidden in the model, that are affecting the results.

So, before drawing conclusion about this,we inspect some more aspects of our model.

✍ Observed values vs Predicted Values Plot

We plot the observed vs fitted or predicted values to get some idea about the prediction :-

```r
plot(1:nrow(mydata),mydata$y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index", col = "blue")
lines(1:nrow(mydata),model$fitted.values,type = "o",pch = 20,col = "red",lty = 2)
abline(v = 1:nrow(mydata),lty = 2,col = rgb(0,1,1,alpha = 0.4))
legend("topright",legend = c("Observed","Predicted"),fill = c("blue","red"))
```



The fit is good except a few observations.
There may be many reasons for this which we will eventually look into.

3.2 Fitting Regression Model by considering quadratic relation of response (OXY) with weight

We have observed from the scatterplots of response (OXY) with the covariates that oxy possibly has some quadratic relation with weight, so we shall now try to fit a regression model by considering quadratic relation of response with weight and linear relation with the other remaining covariates :

New model under consideration is $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5 + \beta_7 x_6 + \epsilon_i, i = 1, 2, .., n,$ (Model-2) with the assumptions:

1. The errors are normally distributed with mean 0.
2. The errors are homoscedastic with common variance σ^2 (unknown).
3. ϵ_i's are uncorrelated.

$\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7$ are the unknown parameters to be estimated. In vector-matrix notation the model is :

$$\boldsymbol{Y}^{n\times 1} = \boldsymbol{X}^{n\times p}\boldsymbol{\beta}^{(p+1)\times 1} + \boldsymbol{\epsilon}^{n\times 1}$$

where $n = 31$ (total number of observed responses) and $p = 7$ where the regression matrix,

$$\mathbf{X} = [\mathbf{1_n} \quad \mathbf{x_1} \quad \mathbf{x_2} \quad (\mathbf{x_2})^2 \quad \mathbf{x_3} \quad \mathbf{x_4} \quad \mathbf{x_5} \quad \mathbf{x_6}]$$

and parameter vector, $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7)'$.

✍ Summary of the fitted model

We fit the model stated above in the dataset using lm() function and to get overall idea about the estimates we use the summary() function :-

```
model1 <- lm(y ~ x1 + x2 + I(x2^2) + x3 + x4 + x5 + x6)
summary(model1)


Call:
lm(formula = y ~ x1 + x2 + I(x2^2) + x3 + x4 + x5 + x6)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4009 -0.8964  0.0618  1.0429  5.3886

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.036e+02  3.376e+01   3.070  0.00542 **
x1          -2.268e-01  1.023e-01  -2.218  0.03675 *
x2          -9.403e-02  8.757e-01  -0.107  0.91542
I(x2^2)      1.298e-04  5.715e-03   0.023  0.98207
x3          -2.632e+00  4.187e-01  -6.286 2.05e-06 ***
x4          -2.095e-02  7.214e-02  -0.290  0.77406
x5          -3.693e-01  1.234e-01  -2.992  0.00651 **
x6           3.031e-01  1.396e-01   2.171  0.04047 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.367 on 23 degrees of freedom
Multiple R-squared:  0.8487,Adjusted R-squared:  0.8026
F-statistic: 18.43 on 7 and 23 DF,  p-value: 4.886e-08
```

Here we observe that we do not have enough evidence to reject the null hypothesis $H_0 : \beta_2 = 0, H_0 : \beta_3 = 0, H_0 : \beta_5 = 0$ for test of significance for the components x_2, x_2^2, x_4 that is we fail to reject the null hypotheses at level 0.05.

✍ Observations from the fitted model

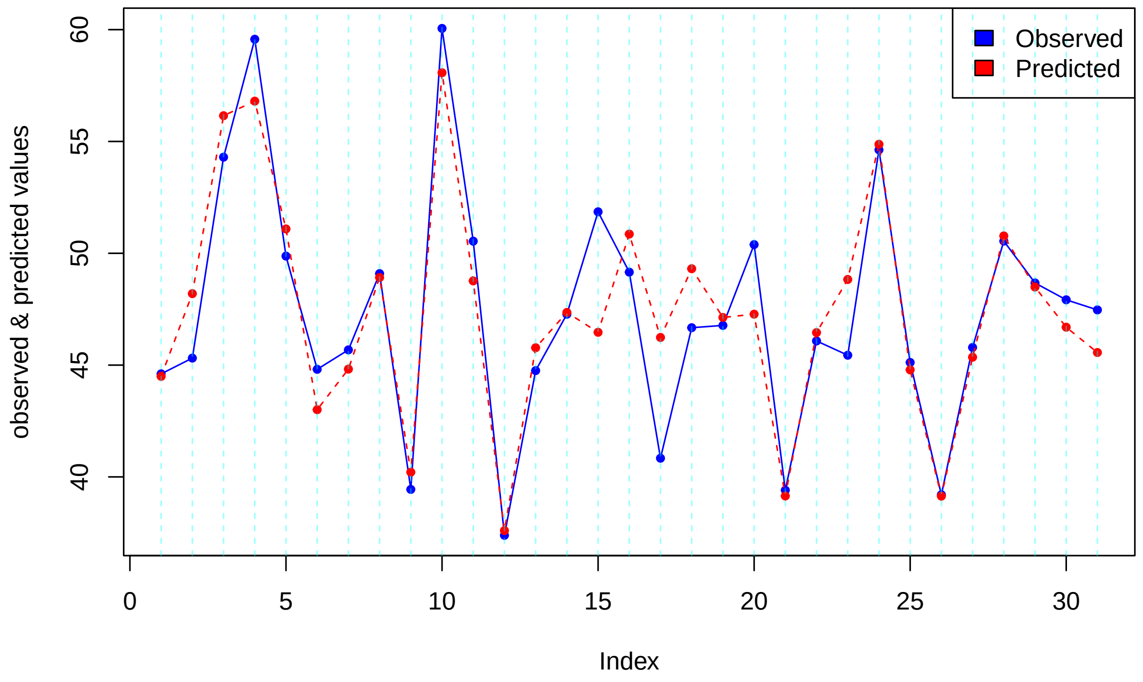
We observe that we do not have enough evidence to reject the null hypothesis $H_0 : \beta_2 = 0, H_0 : \beta_3 = 0, H_0 : \beta_5 = 0$ for test of significance for the components x_2, x_2^2, x_4 that is we fail to reject the null hypotheses at level 0.05. So, we see the coefficients $\beta_1, \beta_4, \beta_6, \beta_7$ for covariates x_1, x_3, x_5, x_6 are not statistically insignificant.

This does not imply that other covariates are insignificant since there maybe some problems of model misspecification.

✍ Observed values vs Predicted Values Plot

We plot the observed vs fitted or predicted values to get some idea about the prediction :-

```
plot(1:nrow(mydata),mydata$y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index", col = "blue")
lines(1:nrow(mydata),model1$fitted.values,type = "o",pch = 20,col = "red",lty = 2)
abline(v = 1:nrow(mydata),lty = 2,col = rgb(0,1,1,alpha = 0.4))
legend("topright",legend = c("Observed","Predicted"),fill = c("blue","red"))
```

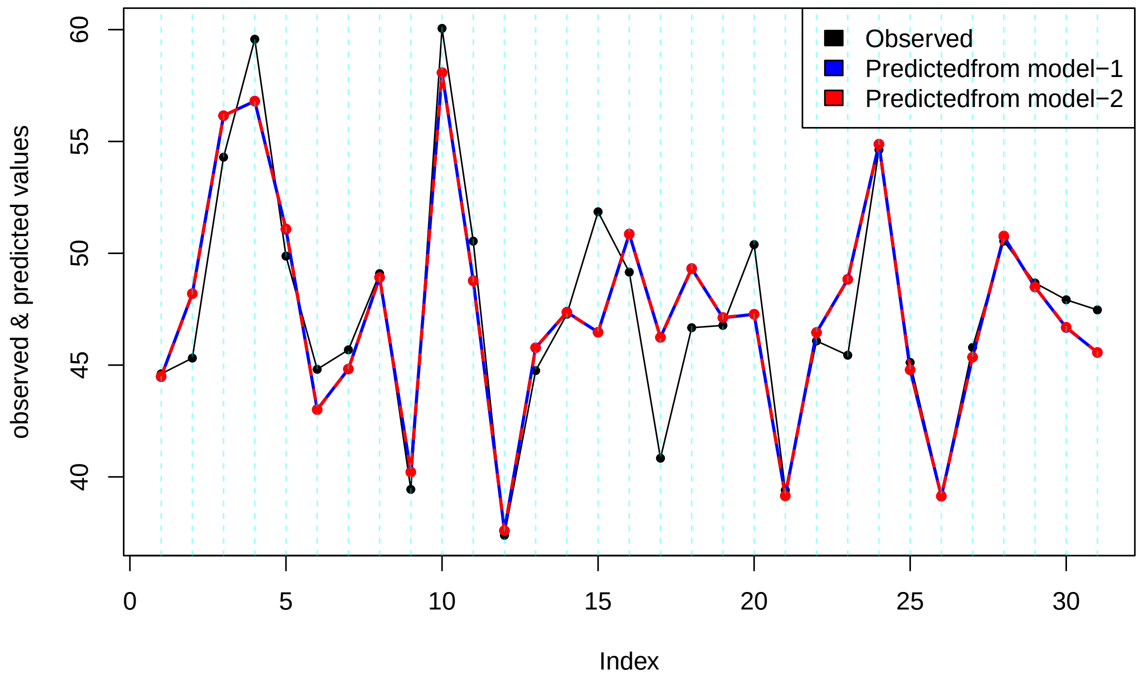


The plot looks like more or less same as the plot corresponding to the multiple linear regression model.

3.3 Comparing the above two fitted model

Model	R^2	R^2_{adj}
(1) $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \epsilon$	0.8487	0.8108
(2) $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5 + \beta_7 x_6 + \epsilon$	0.8487	0.8026

```
plot(1:nrow(mydata),mydata$y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index", col = "black")
lines(1:nrow(mydata),model$fitted.values,type = "o",pch = 20,col = "blue", lwd=2)
lines(1:nrow(mydata),model1$fitted.values,type = "o",pch = 20,col = "red",lty = 2, lwd=2)
abline(v = 1:nrow(mydata),lty = 2,col = rgb(0,1,1,alpha = 0.4))
legend("topright",legend = c("Observed","Predictedfrom model-1","Predictedfrom model-2"),fill = c("black","blue","red"))
```



After observing the adjusted R^2 values and from the plot we see that the fitted values from Model 1 are identical to the one in Model 2, hence we conclude that we are not gaining any extra information by considering a quadratic relation of x_2(weight) with the response, rather we are losing degrees of freedom and compromising with the precision of the parameter estimates. Therefore we shall proceed with the simple linear regression model for now.

4 Regression Diagnostics

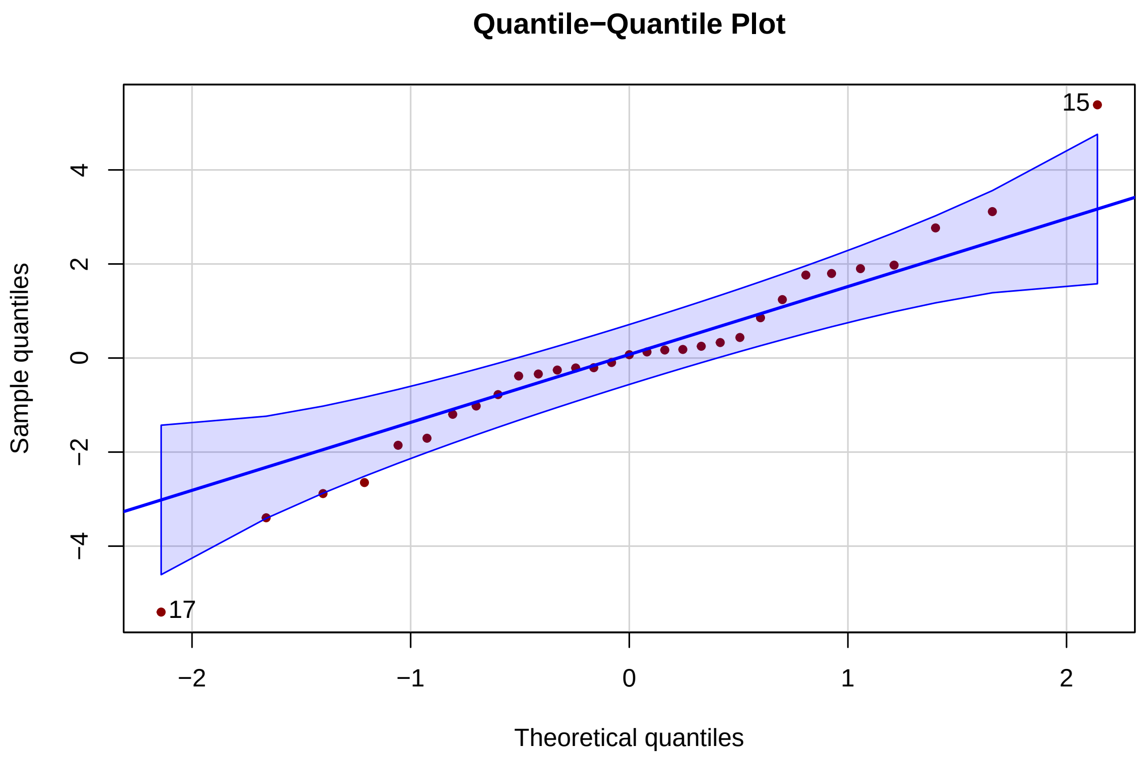
4.1 Checking for different aspects and assumptions of the model

4.1.1 Checking for normality assumptions:

✍ Quantile-Quantile plot of residuals

Here, we plot the sorted residuals (sample quantiles) against the theoretical quantiles of a normal distribution :

```
resi<-residuals(model)
with(ToothGrowth, qqPlot(resi,pch=20, col="darkred", xlab = "Theoretical quantiles"
     ,ylab ="Sample quantiles", main="Quantile-Quantile Plot"))
```



```
[1] 17 15
```

The diagram indicates long-tailed distribution, i.e., quantiles of the residuals obtained from our data do not match the theoretical quantiles of the normal distribution towards the tail, and thus, there is a possible deviation from normality.

We also perform the Shapiro-Wilk's test to check for normality.

✍ Shapiro-Wilk Test

We test the following hypothesis H_0 : residuals are normally distributed against $H_1 : H_0$ is false using Shapiro-Wilk test in R as :-

```
shapiro.test(resi)
```

```
Shapiro-Wilk normality test

data:  resi
W = 0.96779, p-value = 0.4603
```

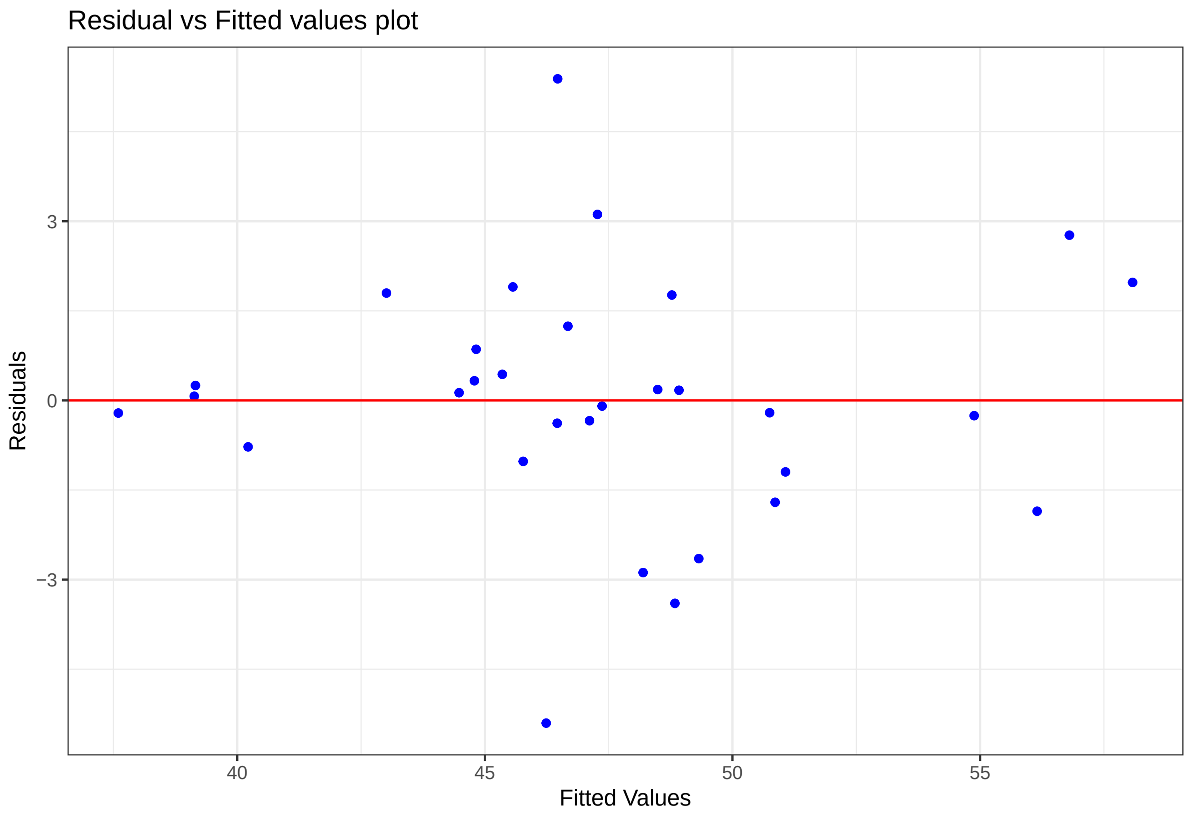
Looking at a non-significant p-value(0.4603) doesn't lend support to the null hypothesis. In this case, a non-significant SW doesn't show normality-- it just means the sample doesn't have enough information to suggest stronger incompatibility with normality which may be due to sample size(31) or just due to the actual distribution (or some kind of bias) therefore relying too much on formal tests of normality can lead us astray.

It is important to note that the Shapiro-Wilk test is a one-tailed test, meaning that it only detects deviations from normality in one tail of the distribution. Therefore, it is possible for a data set to be non-normal but not detected by the Shapiro-Wilk test.

4.1.2 Checking for homoscedasticity Assumptions

First to check homoscedasticity assumption, we make the residuals ($\hat{\varepsilon}$) vs fitted values($\hat{y}$) plots :-

```
ggplot(mydata, aes(fitted(model), residuals(model))) +
  geom_point(col="blue")+geom_abline(intercept=0,slope=0,col="red")+
  labs(x="Fitted Values", y="Residuals", title="Residual vs Fitted values plot")+
  theme_bw()
```

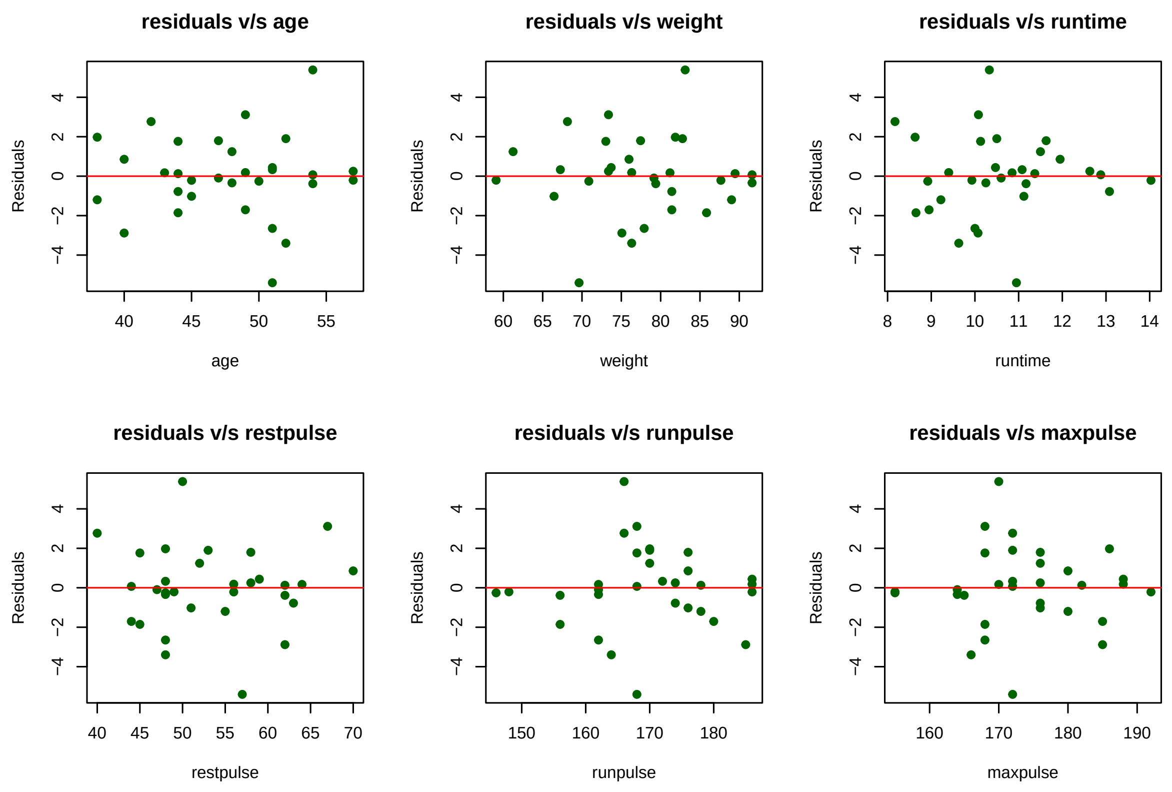


The datapoints seem to be equally distributed below and above the residual = 0 line from the above plot. Since our sample size is quite small it is very difficult to infer anything concrete from the plot, so we perform confirmatory tests to find out if the data satisfies the homoscedasticity assumption.

✍ Residuals vs Covariates plot

We plot the Residuals vs Covariates to observe how the residuals vary with the observations of each covariate and to identify a relationship between them if at all they exist.

```
par(mfrow=c(2,3))
for(i in 3:8)
{
plot(data[,i], residuals(model), xlab=colnames(data)[i], ylab="Residuals",
main=paste("residuals v/s", colnames(data)[i]), pch=19, col= "darkgreen", grid=TRUE)
abline(h=0, col="red")
}
```



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

We observe:

- **Residuals vs Age:** We observe a well-behaved plot that bounces randomly and forms a roughly horizontal band around the residual = 0 line.
- **Residuals vs Weight:** We observe that almost no data points stand out from the basic random pattern of the other residuals.
- **Residuals vs Runtime:** We observe that there is a pattern in this plot which gradually converges to the residual = 0 line.
- **Residuals vs Restpulse:** We observe that most data points do not stand out from the basic random pattern of the other residuals.
- **Residuals vs Runpulse:** There seems to be a hyperbolic pattern. This indicates that the variance of the residuals is proportional to the value of this covariate.
- **Residuals vs Maxpulse:** We do not observe a random pattern and there is no formation of a roughly horizontal band around the residual = 0 line.

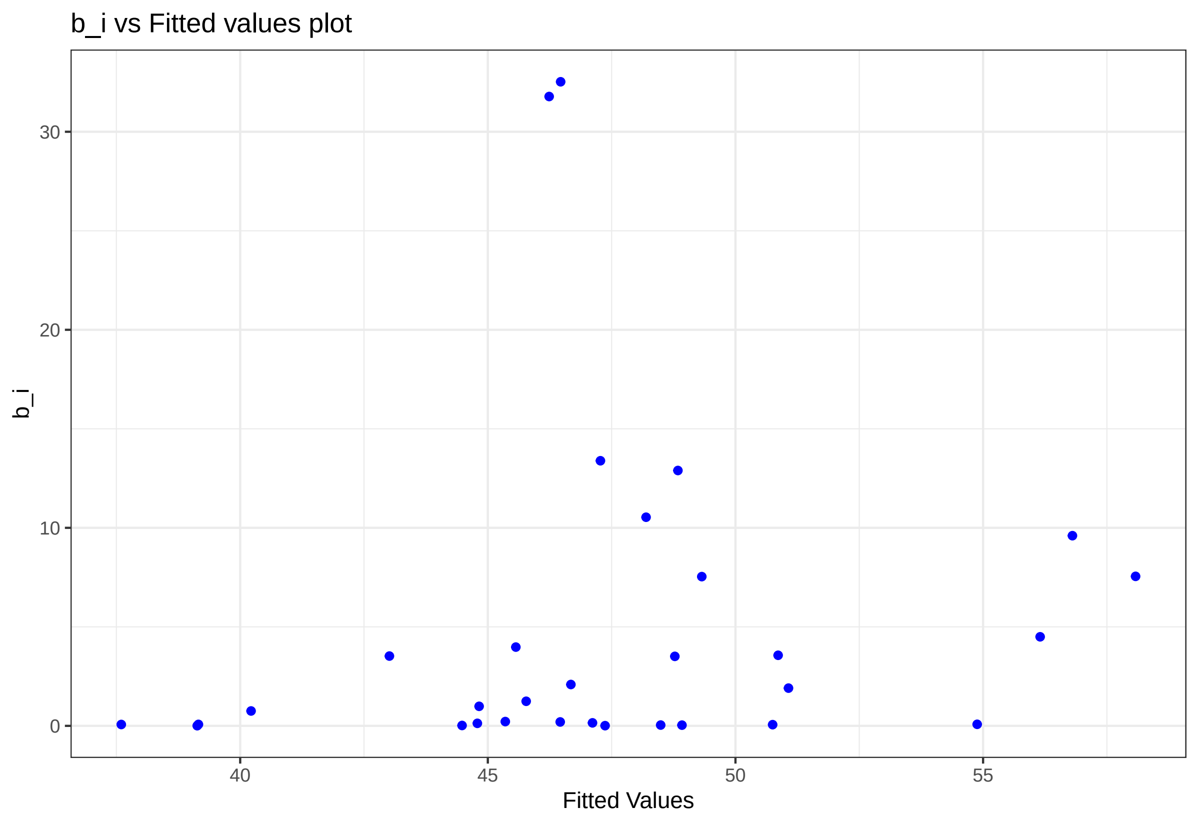
The homoscedasticity assumption of the error might not be true.

✍ b_i vs $\widehat{y}_i$ plot

A standard technique to detect presence of Heteroscedasticity is to plot the quantities $b_i = \frac{e_i^2}{1-h_i}$ against the fitted values $\widehat{y}_i$.

We make the plot using R :-

```
A = as.matrix(mydata[,-1])
H = A%*%solve(t(A)%*%A)%*%t(A)
H_i = diag(H)
e_i = residuals(model)
b_i = (e_i^2)/(1-H_i)
ggplot(mydata, aes(fitted(model), b_i)) +
  geom_point(col="blue")+
  labs(x="Fitted Values", y="b_i", title="b_i vs Fitted values plot")+
  theme_bw()
```



This plot show that the errors seems not to be homoscedastic as we see a wedge like shape in the plot. Since we have very limited data it is not wise to conclude anything from the above plots alone.
We will perform the Breusch-Pagan test to check whether it gives any further information.

✍ Breusch-Pagan Test

We perform the Breusch–Pagan test for testing the homoscedasticity assumptions using R :-

```
bptest(model)


studentized Breusch-Pagan test
```

```
data:  model
BP = 2.8268, df = 6, p-value = 0.8302
```

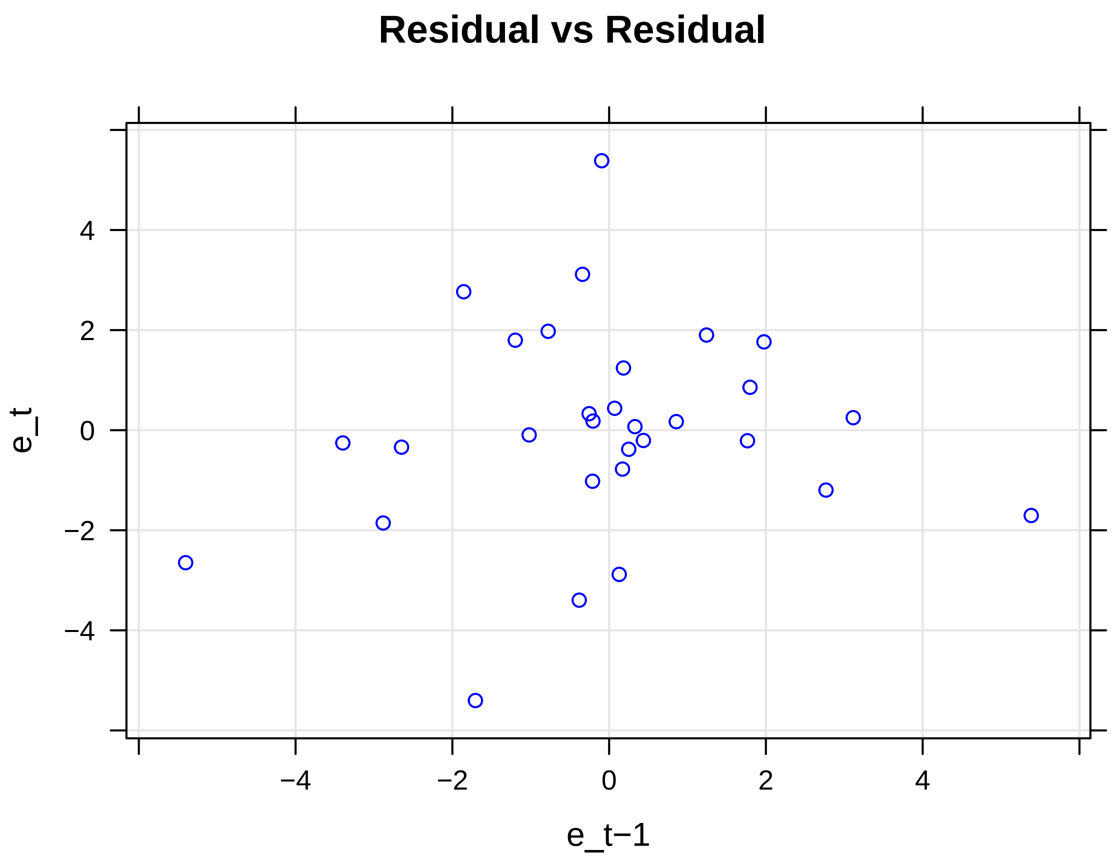
Here we see that the p-value is 0.8302(quite high) and we fail to reject the null hypothesis(errors are homoscedastic) using the Breusch-Pagan Test.In Breusch-Pagan test, the test assumes that the variance is a function of some of the covariates. The Breusch-Pagan test can fail when the heteroscedasticity in the data is not predictable using the covariates,but among some of the residual vs covariate plots, we have seen a pattern so we cannot claim that the heteroscedasticity in the data(if present) cannot be predicted using the covariates. In other words, heteroscedasticity might or might not be present in the data.

We shall check for the other model assumptions and if something is violated we shall try to fix it and then will come back to this problem.

4.1.3 Checking for Autocorrelation

✍ **$\hat{\epsilon}_t$ vs $\hat{\epsilon_{t-1}}$plot**

```
xyplot(residuals(model)[2:31]~residuals(model)[1:30], grid= TRUE, col = "blue", xlab="e_t-1", ylab="e_t", main="Residual vs Residual")
```

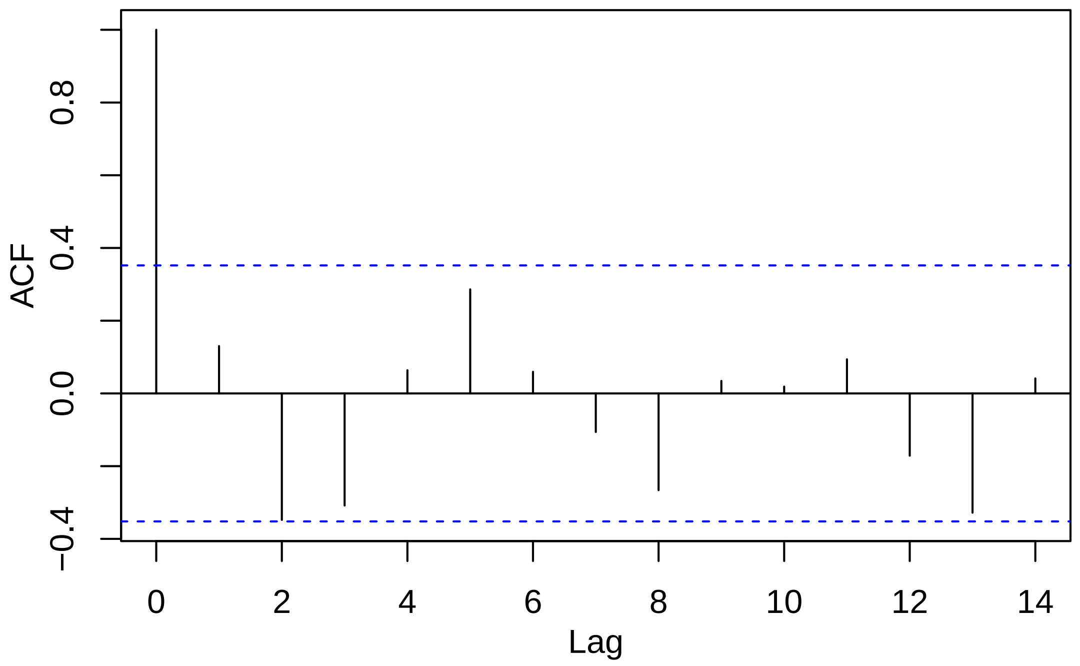


There is no specific pattern to the plot hence we can believe that there is no correlation among the errors although we shall perform some more tests to strengthen our belief.

✍ **ACF plot**

If the errors in the model are truely independent, then we will expect the sample autocorrelation coefficients for different lags k to be insignificant.

```r
acf(resi,ylab = "",xlab = "",main = "")
title(xlab="Lag", ylab="ACF", line=2)
```

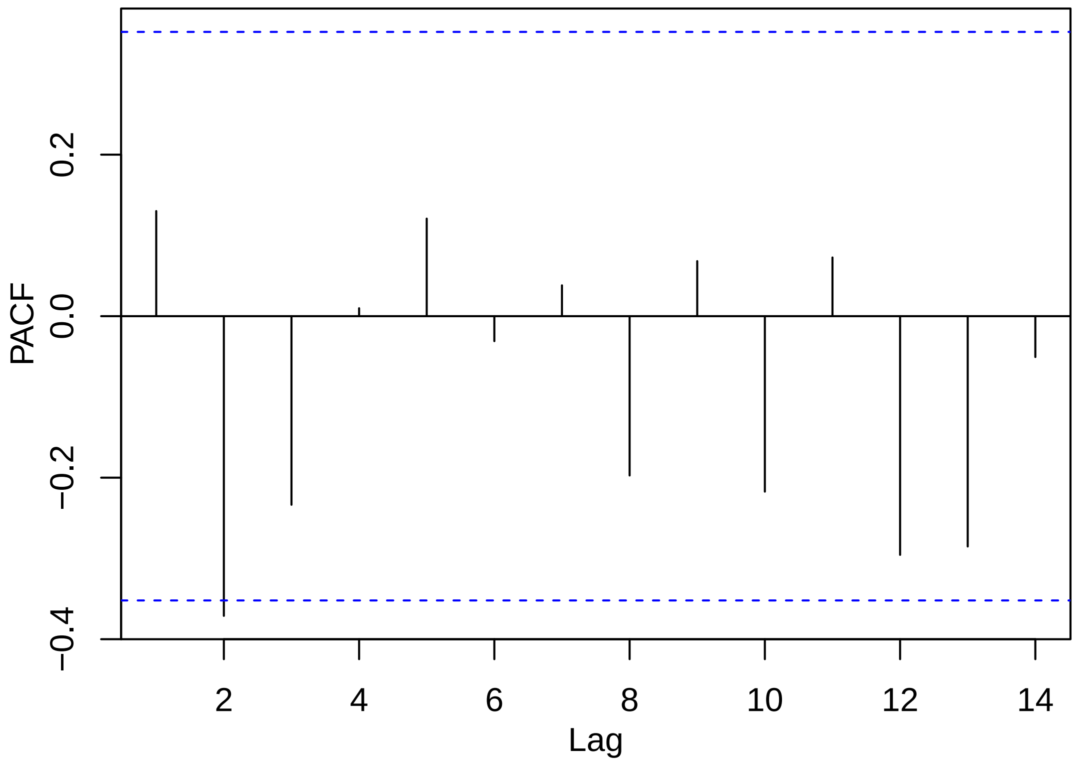


This plot gives indication of no presence of any type of correlation between the residuals.

✍ PACF plot

Similarly, we plot the sample partial autocorrelation coefficients for different lags. We get the same kind of observations indicating no presence of correlations.

```r
pacf(resi,ylab = "",xlab = "",main = "")
title(xlab="Lag", ylab="PACF", line=2)
```



✍ **Durbin-Watson Test**

To test the null hypothesis H_0 : errors are uncorrelated against H_1 : errors are correlated, we perform Durbin-Watson test which gives the following results :-

```
dwtest(y~x1+x2+x3+x4+x5+x6,data=mydata, alternative="two.sided")


Durbin-Watson test

data:  y ~ x1 + x2 + x3 + x4 + x5 + x6
DW = 1.7115, p-value = 0.3733
alternative hypothesis: true autocorrelation is not 0
```

Since the test gives high p-value (0.377), we do not reject H_0. Hence, the assumption of uncorrelated residuals can be assumed to be satisfied.

✍ **Breusch–Godfrey test**

To check whether residuals are uncorrelated for higher orders, we perform the Breusch–Godfrey test upto order 20.

```
bgtest(model,order = 20)


Breusch-Godfrey test for serial correlation of order up to 20

data:  model
LM test = 22.633, df = 20, p-value = 0.3072
```

Here also the p-value is fairly high favouring the null assumption.

4.1.4 Detecting Influential Points

✍ Hat Matrix Diagonals

To detect high leverage points, we compute the hat matrix diagonals h_i of the matrix $\boldsymbol{H} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$:-

```
hat_d <- hatvalues(model)
head(sort(hat_d,dec=TRUE))

       10        12        26        28        20        22
0.4915736 0.4199574 0.3751986 0.3260652 0.2788765 0.2766513
```

We find out if there is any diagonal element with value $> \frac{2p}{n}$ as they should be looked at more closely.

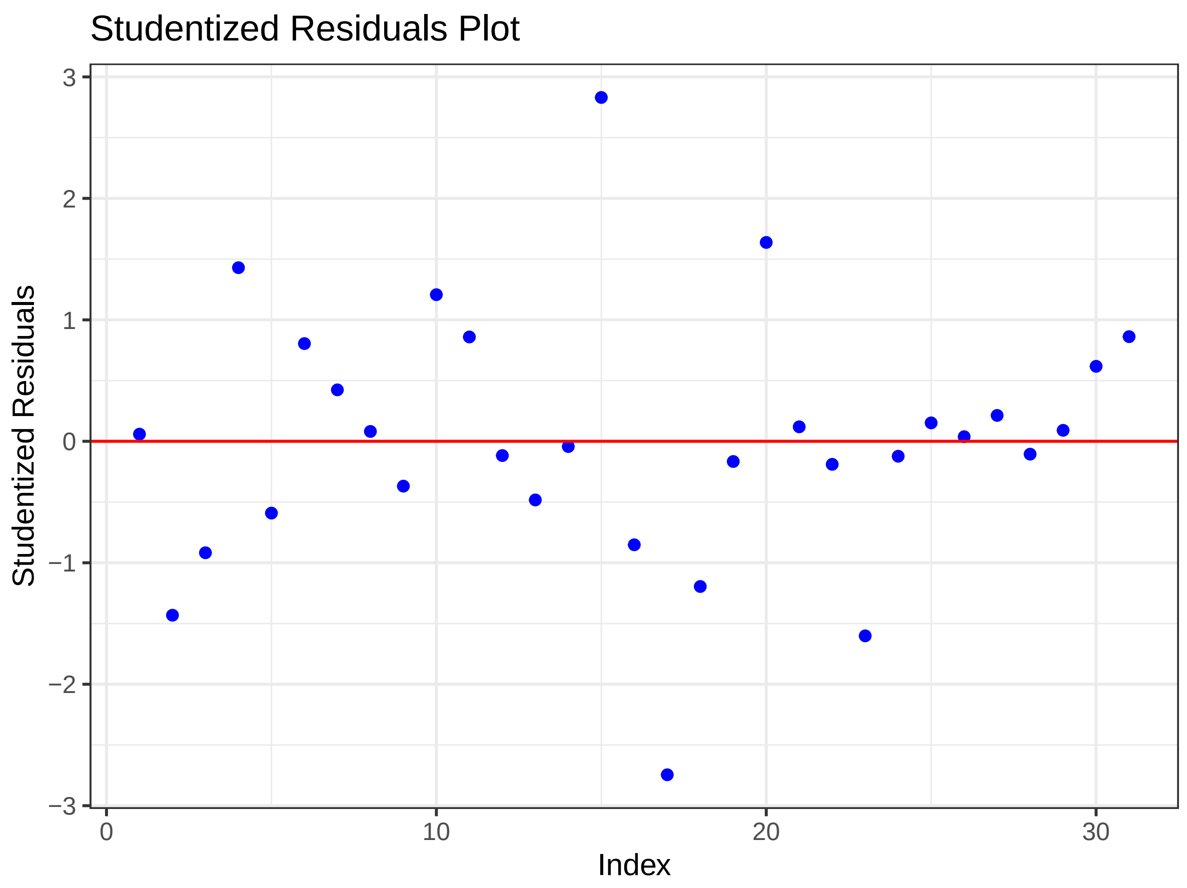
```
hat_d[hat_d > 2*(p/n)]

       10
0.4915736
```

Hence we will apply other procedures also to confirm whether these points are influential or not.

✍ Externally Studentized Residuals

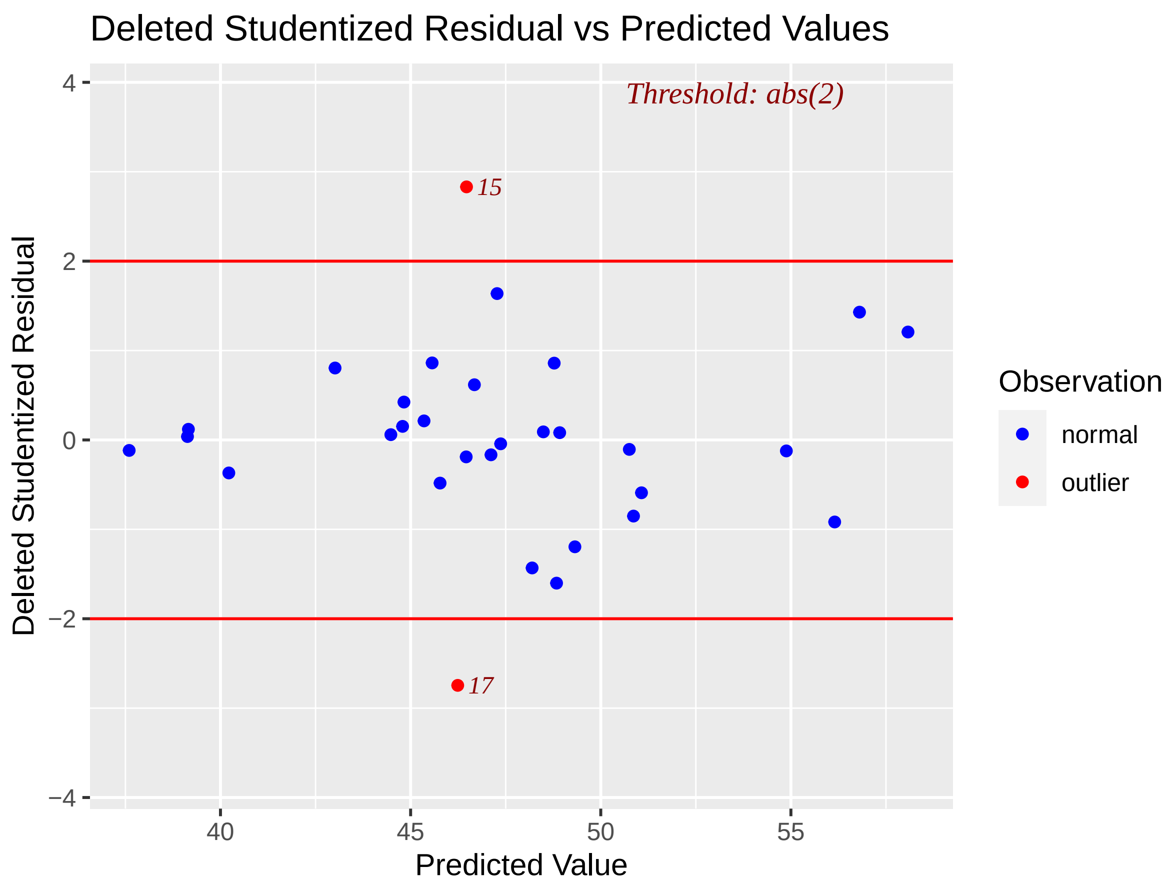
We plot the externally studentized residuals using the formula $t_i^2 = r_i^2\left(\frac{n-p-1}{n-p-r_i^2}\right)$:-

```
stud <- rstudent(model)
ggplot(data, aes(x=index,y=stud),ylim=c(-3,3)) +
  geom_point(col="blue")+geom_abline(intercept=0,slope=0,col="red")+
  labs(x="Index", y="Studentized Residuals", title="Studentized Residuals Plot")+
  theme_bw()
```



If the assumptions are correct i.e. $\boldsymbol{\epsilon} \sim N_n\left(\mathbf{0}, \sigma^2 \boldsymbol{I}\right)$ then we should get that $t_i \sim t_{n-p-1}$.
Hence the significant externally studentized residuals will have values$|t_i| > t_{n-p-1;\frac{\alpha}{2}} \iff t_i^2 > F_{n-p-1;\alpha}$:-

```
ols_plot_resid_stud_fit(model)
```



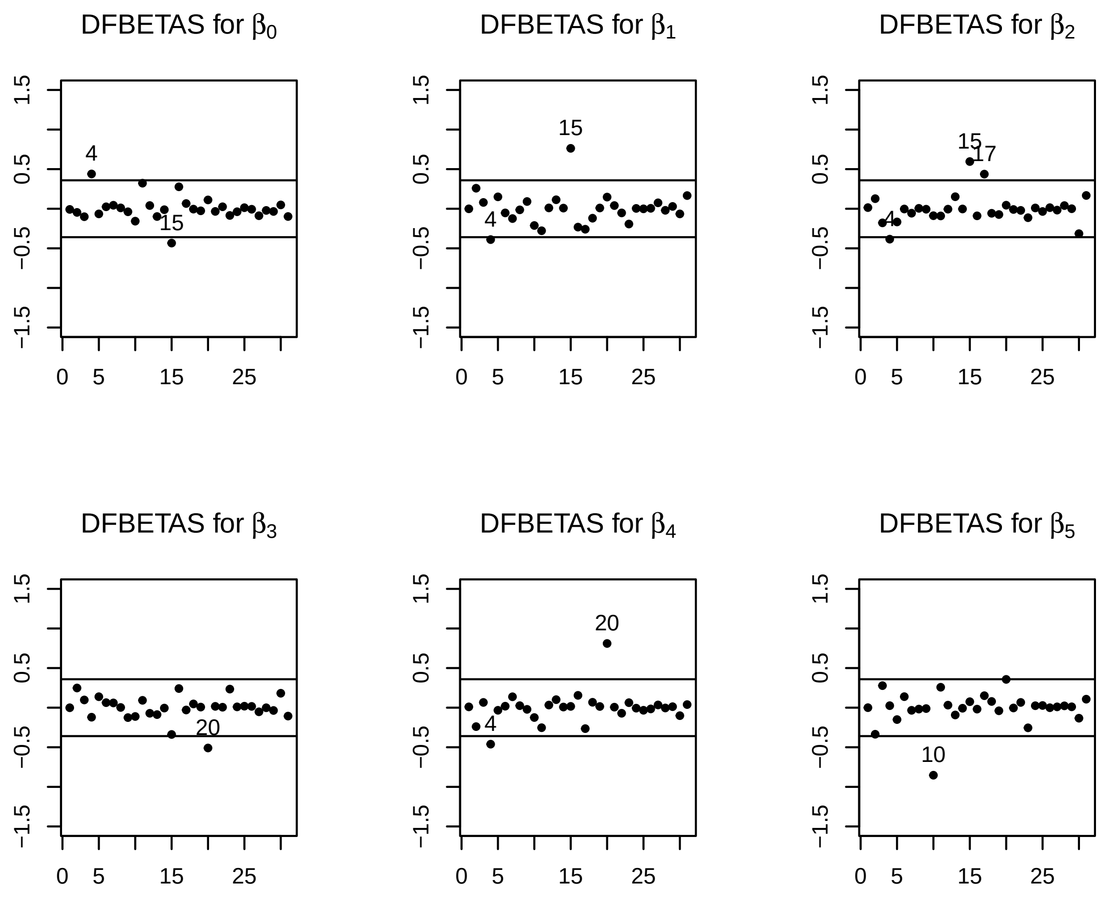
We can see from the plot that two residuals are significant hence we treat them as outliers.

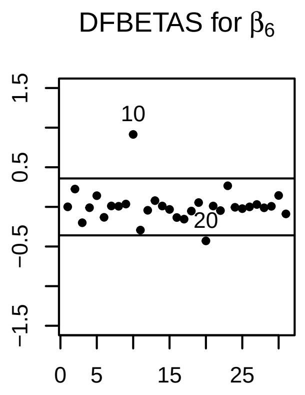
✍ **DFBETAS**

Now, we will detect the high leverage points by analyzing the DFBETAS measure for different parameters, which is calculated as $DFBETAS_{ij} = \frac{\widehat{\beta}_j - \widehat{\beta}(i)_j}{S(i)\sqrt{\sum_i c^2_{j+1,i}}}$ where $\boldsymbol{C} = ((c_{ij})) = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}$.

We will check for points that have $|DFBETAS_{ij}| > \frac{2}{\sqrt{n}}$. Here we plot the values for all the 7 coefficients $\beta_i, i = 0, ..., 6$.

```r
par(mfrow = c(2,3))
DFBETAS = dfbetas(model)
for(i in 1:7)
{
  plot(DFBETAS[,i],main=bquote("DFBETAS for" ~ beta[.(i-1)]),ylab="",ylim=c(-1.5,1.5),xlab="",pch=20)
  abline(h=c(-2/sqrt(n),2/sqrt(n)))
  ind = which(abs(DFBETAS[,i]) > 2/sqrt(n)) # beta_0
  text = text(ind,DFBETAS[ind,i],pos = 3,labels = ind)
}
```

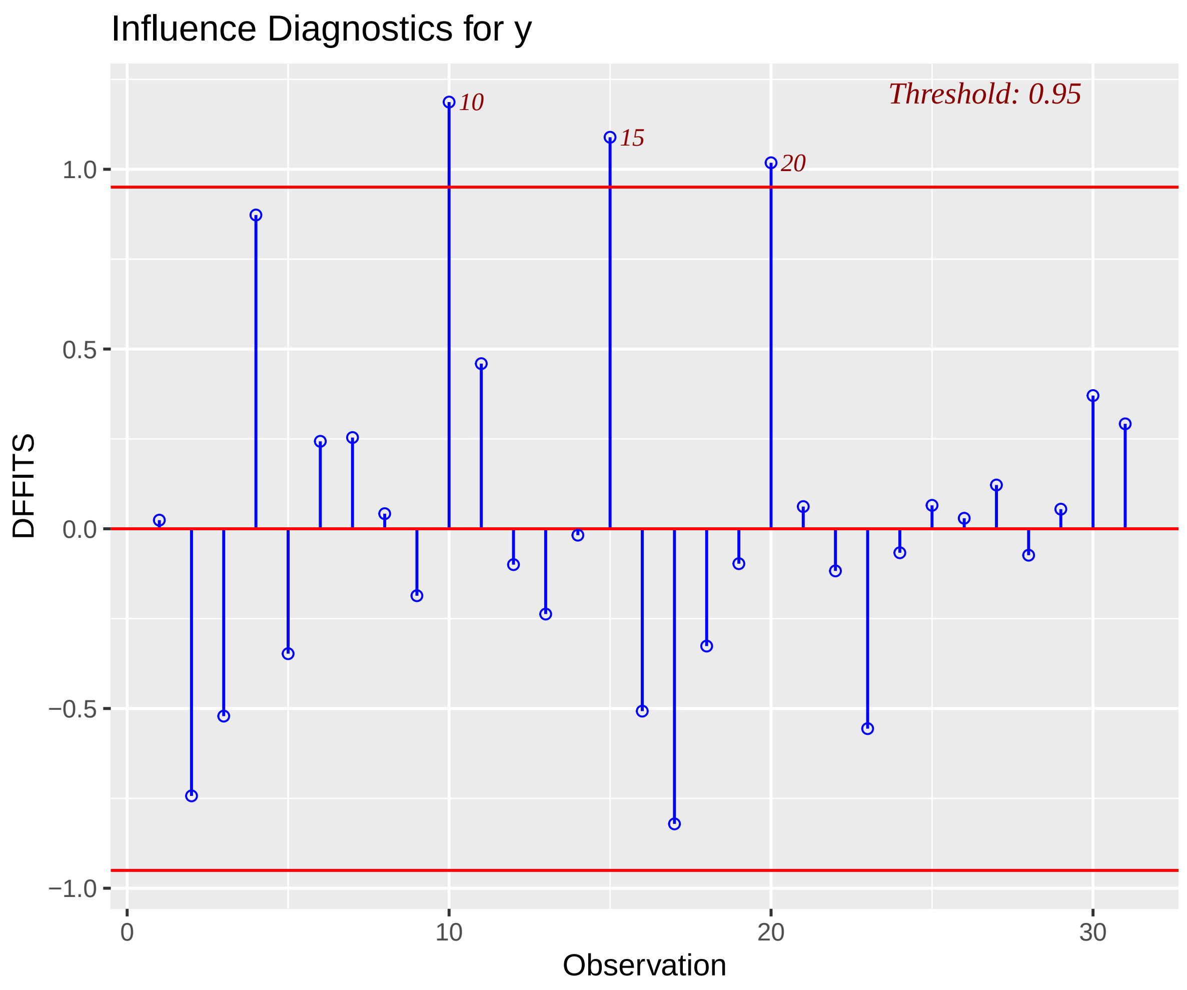

DFBETAS for β_0
4
15
DFBETAS for β_1
15
4
DFBETAS for β_2
15
17
4
DFBETAS for β_3
20
DFBETAS for β_4
20
4
DFBETAS for β_5
10
1.5
0.5
−0.5
−1.5
0
5
15
25



✍ **DFFITS**

We plot the DFFITS values to observe the change in fitted values, where $DFFITS_i = t_i \left(\frac{h_i}{1-h_i}\right)^{12}$, and examine the corresponding points for which $|DFFITS_i| > 2\sqrt{\frac{p}{n}}$.

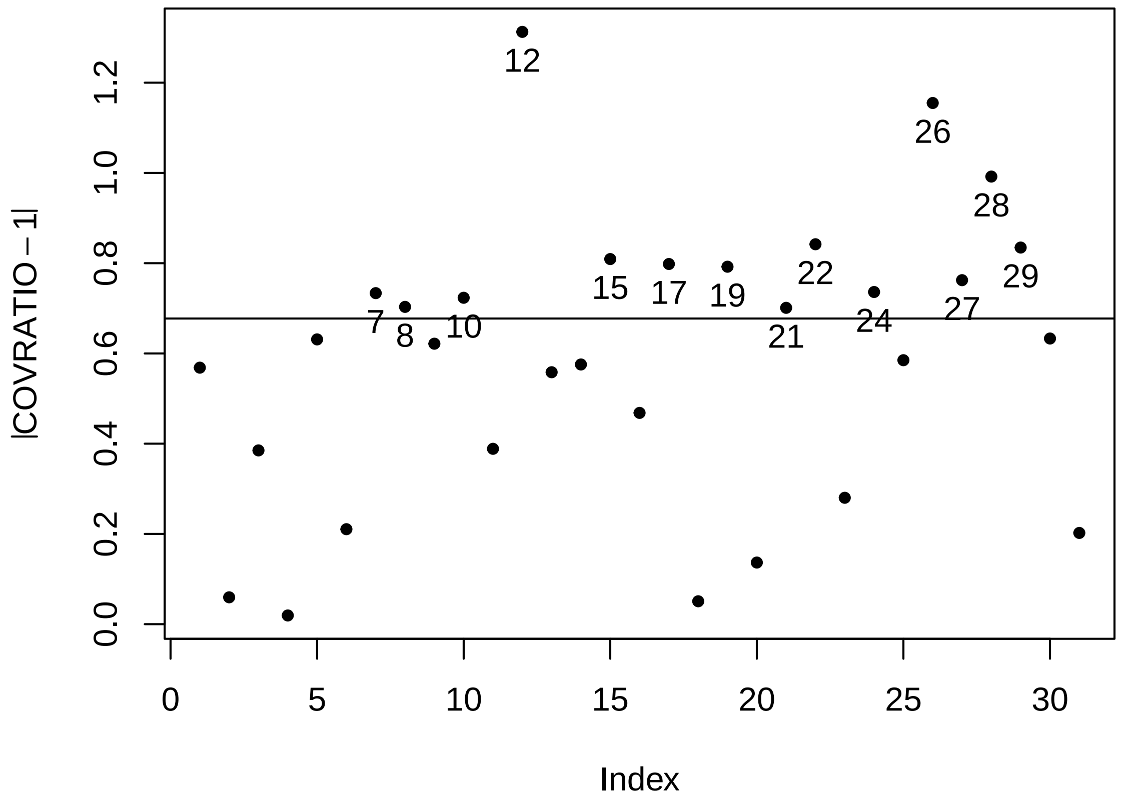
```
ols_plot_dffits(model)
```



✍ **COVRATIO**

We plot the COVRATIO values, given by $COVRATIO_i = \left(\frac{n-p-1}{n-p} + \frac{t_i^2}{n-p}\right)^{-p}(1-h_i)^{-1}$. We deem the points to possess significant influence when $|COVRATIO - 1| > \frac{3p}{n}$.

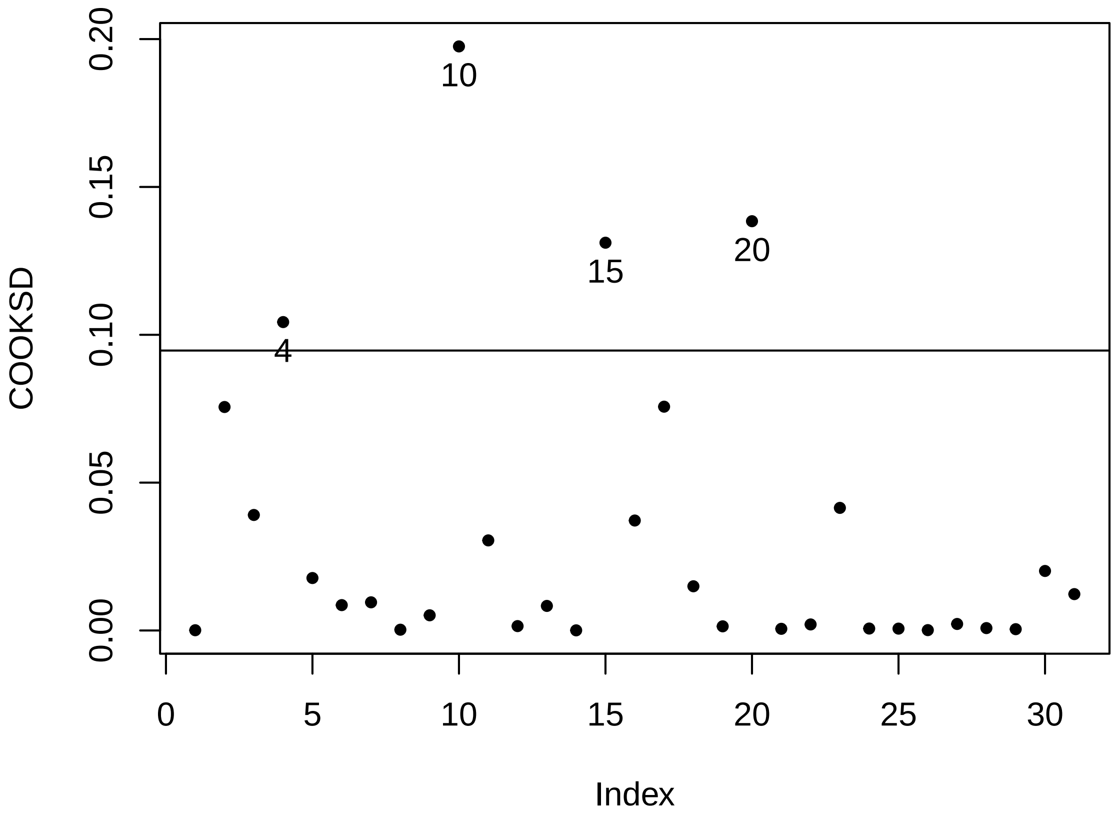
```
COVRATIO = covratio(model)
plot(abs(COVRATIO-1),ylab=expression(abs(COVRATIO-1)),pch = 20)
abline(h = 3*p/n)
ind = which(abs(COVRATIO-1) > 3*p/n)
text(ind,(abs(COVRATIO-1))[ind],pos = 1,labels = ind)
```



✍ **Cook's D**

The Cook's Distance, D_i, is calculated as $r_i^2 \frac{h_i}{p(1-h_i)}$ for each of the n points. We designate points as suspicious if $D_i > \frac{4}{n}$, with n=31 and p=7 resulting in a value of 0.129. By plotting the values, we can determine if any influential points exist.

```
COOKSD = cooks.distance(model)
plot(COOKSD,pch=20)
abline(h = 3*mean(COOKSD))
ind = which(COOKSD > 3*mean(COOKSD)) # beta_0
text = text(ind,COOKSD[ind],pos = 1,labels = ind)
```



The values of D_i for points 4, 10, 15, 20 are clearly significant, leading us to investigate them further.

✍ **Conclusion**

From all the diagnostics performed for finding influential observations, we can make the following table of our findings :-

Diagnostic Measures	Points Detected
h_i	10 (high-leverage)
t_i	15, 17 (outlier)
$DFBETAS$	4, 10, 15, 17, 20 (influential)
$DFFITS$	10, 15, 20 (influential)
$COVRATIO$	7, 8, 10, 12, 14, 15, 17, 21, 22, 24, 26, 27, 28, 29 (influential)
$Cook's\ D$	4, 10, 15, 20 (influential)

Consequently, the table indicates that observations 10 and 15 are potentially influential based on most diagnostic measures, so we will assess if removing them enhances the fitted model.

4.1.5 Checking for Multicollinearity

✍ **Pairwise Correlation**

```
round(cor(mydata[,-1]),2)
```

```
      x1    x2   x3    x4    x5    x6
x1  1.00 -0.23 0.19 -0.16 -0.34 -0.43
x2 -0.23  1.00 0.14  0.04  0.18  0.25
x3  0.19  0.14 1.00  0.45  0.31  0.23
x4 -0.16  0.04 0.45  1.00  0.35  0.31
x5 -0.34  0.18 0.31  0.35  1.00  0.93
x6 -0.43  0.25 0.23  0.31  0.93  1.00
```

The correlation between runpulse (x_5) and maxpulse (x_6) is very large and there are also moderately large pairwise correlation between some of the other covariates.

✍ Eigen-decomposition of $\mathbf{X^*}^{\mathbf{T}}\mathbf{X}^*$ and Condition Number

In the next step, we analyze the issue of multicollinearity in our dataset. To measure this, we compute the condition number for the scaled and centered model matrix $\mathbf{X}^*$ and check the eigen decomposition of $\mathbf{X^*}^{\mathbf{T}}\mathbf{X}^*$, which is $\sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$, with λ_i's representing the eigenvalues of $\mathbf{X^*}^{\mathbf{T}}\mathbf{X}^* = \mathbf{R_{xx}}$. We calculate $\kappa\left(\boldsymbol{X}^*\right)$ using R :-

```
X_mdl = scale(model.matrix(model)[,-1])
e <- eigen(t(X_mdl) %*% X_mdl)
e$val

[1] 77.247573 39.832887 27.752696 22.296657 11.060271  1.809916

kappa(X_mdl)

[1] 6.834833
```

We note that there is a wide range in the eigen values and the square of condition number $\kappa^2\left(\mathbf{X}^*\right)$ is approximately 46.715. This upper bound for the VIFs is quite large.

✍ Calculating VIF Values

To determine whether or not a covariate with corresponding column $\mathbf{x_j}$ can be predicted accurately using other covariates, we compute the variance inflation factors $VIF_j = \frac{1}{1-R_j^2}$, where R_j^2 is the coefficient of determination of the regression of $\mathbf{x^*}_{(j)}$ on the columns of $\mathbf{X^*}_{(j)}$. Here $\mathbf{X^*}_{(j)}$ is the resulting matrix after deleting the jth column of $\mathbf{X}^*$ and $\mathbf{x^*}_{(j)}$ is the jth column of $\mathbf{X}^*$.

In R, we calculate VIF$_j$ values for $j = 1, 2, 3, 4,$ 5, 6 :-

```
vif(lm(y~x1+x2+x3+x4+x5+x6,data = mydata))#high values imply collinearity

      x1       x2       x3       x4       x5       x6
1.512836 1.155329 1.590868 1.415589 8.437274 8.743848
```

We can observe that the VIF values for variables x_5 and x_6 are respectively 8.437274 and 8.743848 which are really high and even close to 10. Also, we may deduce that "the standard error of $\widehat{\beta_5}$ and $\widehat{\beta_6}$ would be respectively $\sqrt{8.437} \approx 2.905$ and $\sqrt{8.744} \approx 2.957$ times more in presence of collinearity".

✍ Collinearity and Influential points

We first calculate the VIF values using the initial model:

```
vif(lm(y~x1+x2+x3+x4+x5+x6,data = mydata[]))

      x1       x2       x3       x4       x5       x6
1.512836 1.155329 1.590868 1.415589 8.437274 8.743848
```

After removing the influential points and then calculating the VIF values, we get :

```
vif(lm(y~x1+x2+x3+x4+x5+x6,data = mydata[-c(10,15),]))#high implies collinearity

       x1        x2        x3        x4        x5        x6
 1.399130  1.181753  1.511645  1.394318 12.617895 12.865483
```

We observed an increase in the VIF values corresponding to x_5 and x_6 after eliminating the influential points.

So, it is intuitive to infer that the actual linear dependence between the covariates x_5 and x_6 was slightly nullified by the presence of influential points.

From the above analysis we conclude that there is high collinearity between covariates runpulse(x_5)and maxpulse (x_6) which are having a significant effect in their estimates by inflating their variances.

4.2 Remedies for the problems found

4.2.1 Removing Influential Points

We exclude the influential data points and subsequently reapply a linear model using all the covariates.

```
model3=lm(y~x1+x2+x3+x4+x5+x6,data=mydata[-c(10,15),])
summary(model3)


Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6, data = mydata[-c(10,
    15), ])

Residuals:
    Min      1Q  Median      3Q     Max
-5.2552 -0.6293  0.2912  0.8625  3.1680

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 109.37640   10.97296   9.968 1.28e-09 ***
x1           -0.27350    0.09117  -3.000  0.00659 **
x2           -0.09795    0.04857  -2.017  0.05611 .
x3           -2.47657    0.33744  -7.339 2.39e-07 ***
x4           -0.01490    0.05764  -0.258  0.79845
x5           -0.28357    0.12749  -2.224  0.03670 *
x6            0.19239    0.14869   1.294  0.20911
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.011 on 22 degrees of freedom
Multiple R-squared:  0.8653,Adjusted R-squared:  0.8285
F-statistic: 23.55 on 6 and 22 DF,  p-value: 1.589e-08

summary(model)


Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4026 -0.8991  0.0706  1.0496  5.3847

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
x1           -0.22697    0.09984  -2.273  0.03224 *
x2           -0.07418    0.05459  -1.359  0.18687
x3           -2.62865    0.38456  -6.835 4.54e-07 ***
x4           -0.02153    0.06605  -0.326  0.74725
x5           -0.36963    0.11985  -3.084  0.00508 **
x6            0.30322    0.13650   2.221  0.03601 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.317 on 24 degrees of freedom
Multiple R-squared:  0.8487,Adjusted R-squared:  0.8108
F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09
```

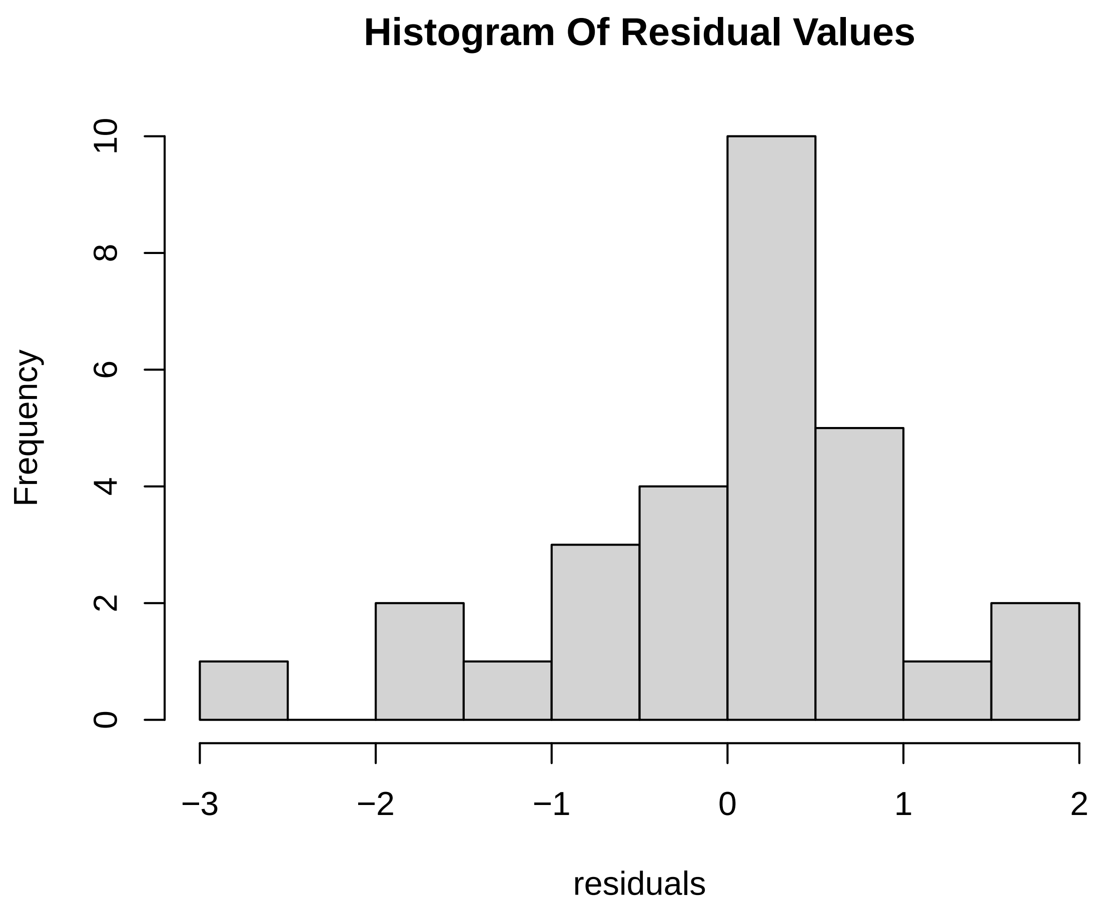
We observe that the standard errors of the coefficients of those covariates that we had detected multicollinearity in, have increased. This reflects that the influential points were masking the effect of multicollinearity and it becomes more pronounced after removing the influential points. After comparing the newly fitted model with the initial multiple linear regression model, we observe an increase in the value of $R^2_{adj} = 0.8285$.According to the model, the estimate of maxpulse(x_6) turns out to be signifcant.We shall verify later if maxpulse(x_6) is signifcant or not.

✍ CHECKING FOR IMPROVEMENTS

We shall perform the diagnostics to check the normality and heteroscedasticity assumption again with the data devoid of the influential points.

➽ Histogram of Residuals to check normality

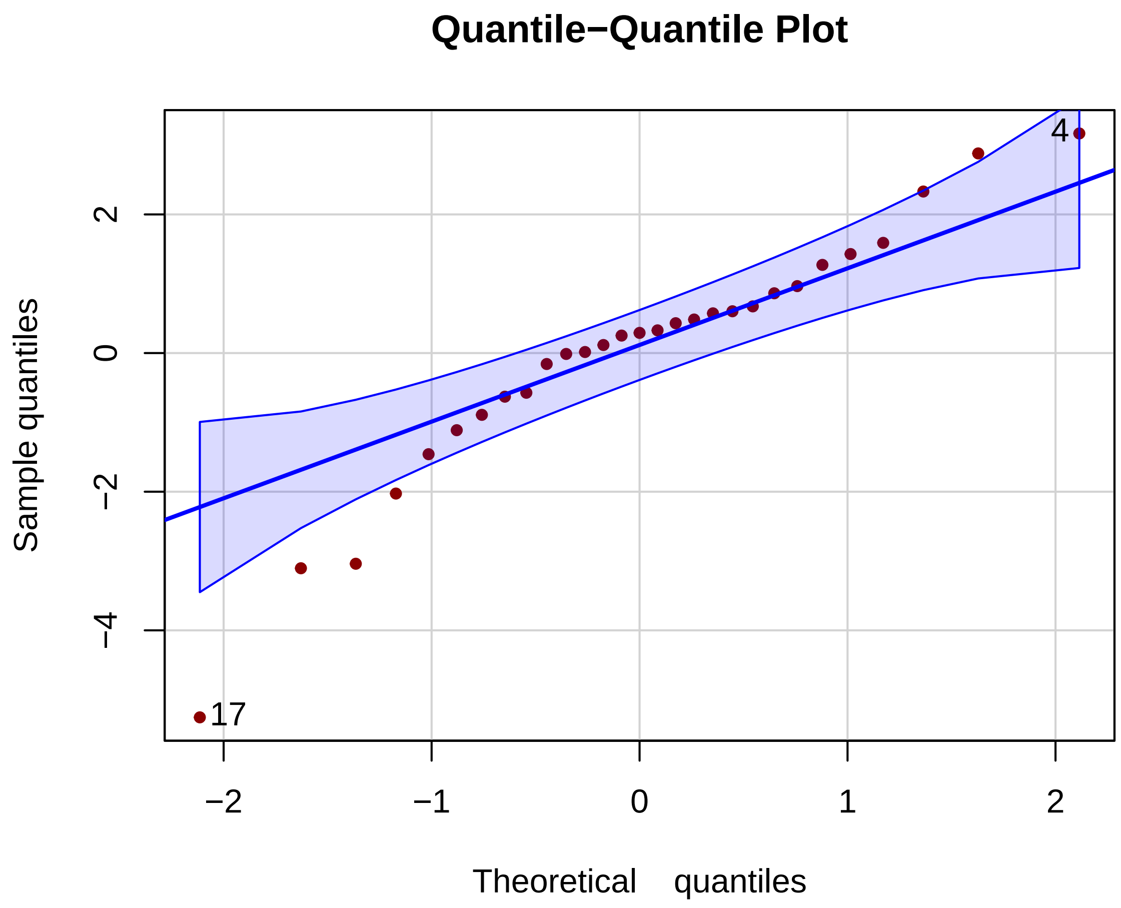
```
hist(rstandard(model3),breaks = 10,main = "Histogram Of Residual Values",xlab="residuals")
```



We again plot the histogram of the residual values but still observe that it indicates non-normality of the residuals.

➽ Quantile-Quantile plot to check normality

```
with(ToothGrowth,qqPlot(residuals(model3),pch=20,col="darkred",
xlab = "Theoretical    quantiles",ylab ="Sample quantiles", main="Quantile-Quantile Plot"))
```



```
17  4
15  4
```

For checking the assumptions for the residuals we again make the quantile-quantile plot of the residual.We observe that even after removing high influential points the q-q plot indicates that normality has been disrupted at the left tail and indicates some long tailed distribution of the residuals.

4.2.2 Remedies For Collinearity

✍ Dealing with Collinearity

We first try to remove one of the correlated covariates runpulse (x_5) or maxpulse (x_6) and see what changes occur in the model

Model	Condition Number (κ)	R^2_{adj}
$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \epsilon$	8.805	0.8285
$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \epsilon$	2.930	0.8235
$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_6 x_6 + \epsilon$	2.967	0.799

The R^2_{adj} decreases on removing either of the covariates x_5 or x_6. This means that the initially fitted model can explain the variability in the response better than the models without *runpulse* or without *maxpulse*. Thus even though the condition number decreases, we do not remove either of the covariates.

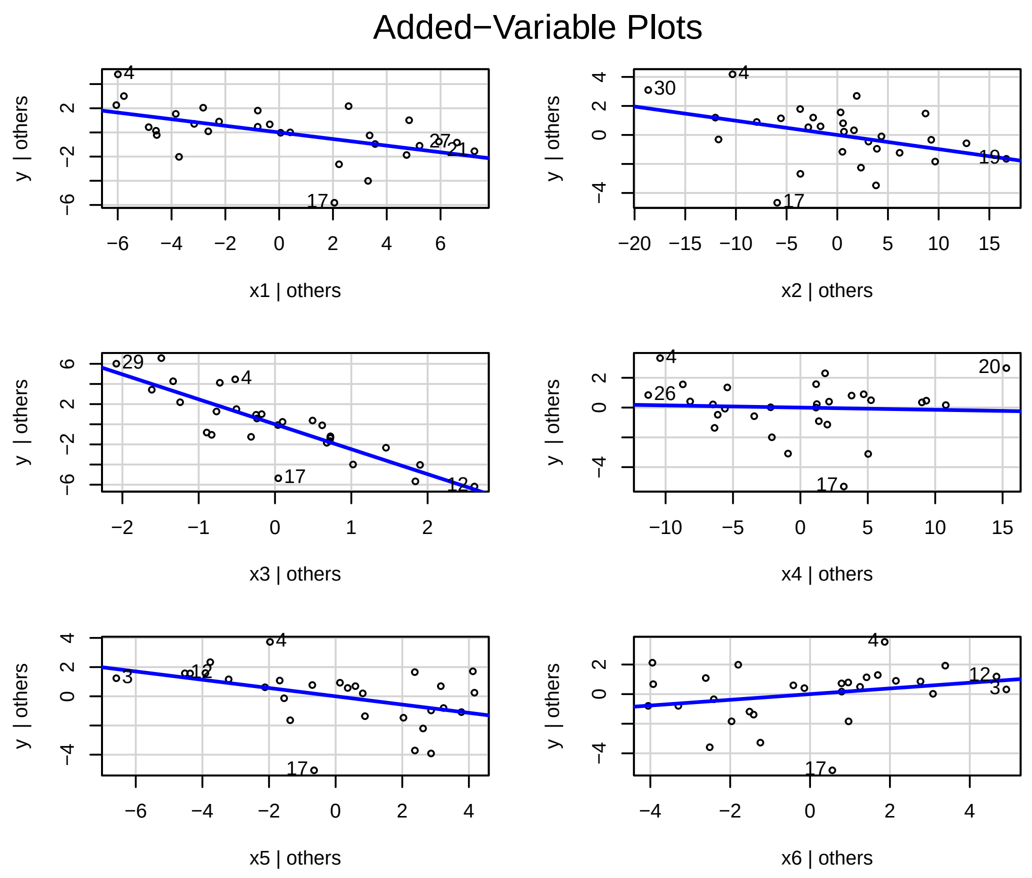
Multicollinearity is a sample phenomenon. It might not have been present if we had a different sample. It makes our estimates unreliable, other than that, it causes no problem in inference. Hence, we shall opt for biased estimation techniques such as ridge estimation.

While checking the homoscedasticity assumption we noticed that *runpulse vs residuals* and *maxpulse vs residuals* plots were not randomly scattered about the residual = 0 line, rather had a pattern. This shows that these variables carry additional information about the response. Thus, we should not remove them immediately.

✍ Added Variable Plot

For better understanding the contribution of a covariate in the regression model, we make a scatter plot of $\boldsymbol{e}^{(j)} = (\boldsymbol{I} - \boldsymbol{P}_j)\boldsymbol{Y}$ against $(\boldsymbol{I} - \boldsymbol{P}_j)\boldsymbol{x}^{(j)}$ where $\boldsymbol{e}^{(j)}$ are the residuals of the model with variable $\boldsymbol{x}^{(j)}$ excluded where $\boldsymbol{x}^{(j)}$ is the column of observations of x_j. This is also called the added variable plot.

```
avPlots(lm(y~x1+x2+x3+x4+x5+x6,data = mydata[-c(10,15),]))
```



✍ Conclusion

From the added variable plots, we can see that slopes of the fitted lines for the added variable plots are significant for all and even the collinear covariates x_5 & x_6.

We can make some important conclusions from here.

There is a strong evidence that we should not remove *runpulse* and *maxpulse* from our model.

4.2.3 Handling Non-normality

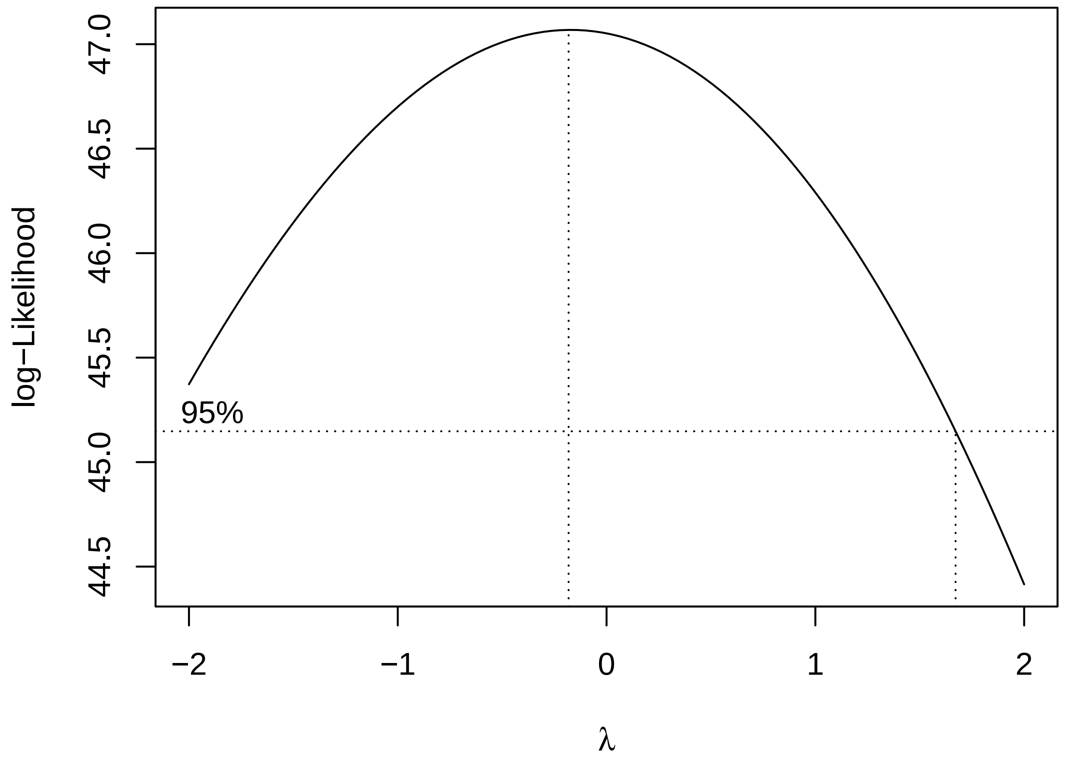
Now we will handle non-normality of the residuals by suitably transforming the data. First we do the well known box-cox transformation:

- **Box-Cox Transformation:**

$$g(y,\lambda) = \begin{cases} y^{\lambda} - 1, & \lambda \neq 0 \\ log(y), & \lambda = 0 \end{cases}$$

We now fit our model using the transformed data and check the summary output :

```
#remedies for non-normality
boxcox(model3)
par(mfrow=c(1,1))
bc <- boxcox(y~., data=mydata[-c(10,15),], plotit=TRUE)
```

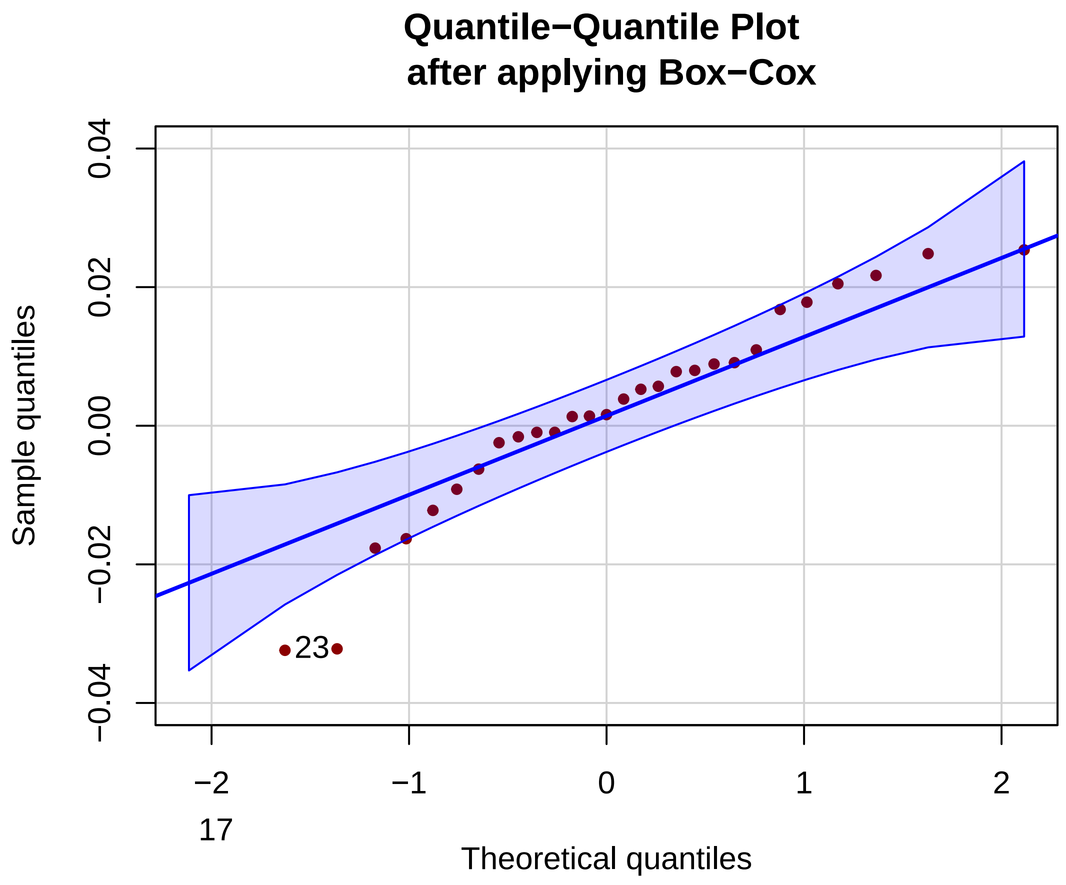


```
lambda_boxcox <- bc$x[which.max(bc$y)]
ynew_boxcox <- (y^lambda_boxcox-1)/lambda_boxcox
mydata1 = cbind.data.frame(ynew_boxcox, x1, x2, x3, x4, x5, x6)
model6 = lm(ynew_boxcox~., data = mydata1[-c(10,15),])
summary(model6)



Call:
lm(formula = ynew_boxcox ~ ., data = mydata1[-c(10, 15), ])

Residuals:
      Min        1Q    Median        3Q       Max 
-0.058725 -0.006261  0.001602  0.009108  0.025376 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  3.4275671  0.1141792  30.019  < 2e-16 ***
x1          -0.0026843  0.0009486  -2.830  0.00975 **
x2          -0.0009794  0.0005054  -1.938  0.06557 .
x3          -0.0277612  0.0035112  -7.906 7.17e-08 ***
x4           0.0001285  0.0005997   0.214  0.83230
x5          -0.0024501  0.0013266  -1.847  0.07825 .
x6           0.0014109  0.0015472   0.912  0.37169
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02092 on 22 degrees of freedom
Multiple R-squared:  0.8712,Adjusted R-squared:  0.8361
F-statistic:  24.8 on 6 and 22 DF,  p-value: 9.818e-09
```

To check whether the transformation resolves non-normality or not, we draw the qqplot for the transformed data :

```
resinew_boxcox <-residuals(model6)
with(ToothGrowth, qqPlot(resinew_boxcox ,pch=20, col="darkred",xlab = "Theoretical quantiles",
ylab ="Sample quantiles", main="Quantile-Quantile Plot \n after applying Box-Cox"
,ylim=c(-0.04,0.04)))
```



```
17 23
15 21
```

From the plot we see that the heavy-tailed feature of the data persists, hence we also perform the shapiro-wilk test to check if there is normality in the transformed residuals:

```
shapiro.test(resinew_boxcox)



Shapiro-Wilk normality test

data:  resinew_boxcox
W = 0.90461, p-value = 0.01265
```

The Shapiro-Wilk level $\alpha = 0.05$ test has a p-value of 0.01265 which is less than 0.05. Thus the test rejects the normality assumption of the residuals.

Since the data has long-tail, the box-cox transformation worsens the situation and deviates it from normality. So we perform some other transformations.

– **Draper and John Transformation:** $g(y,\lambda) = \begin{cases} \frac{sign(y)(((y)+1)^{\lambda}-1)}{\lambda}, & \lambda \neq 0 \\ sign((y+1)), & \lambda = 0 \end{cases}$

First we find the optimum λ by minimizing the likelihood.

```
l= seq(from=-10,to=10,by=0.3)
like=NULL
for(i in 1:length(l))
{
ynew = sign(y)*((abs(y)+1)^l[i]-1)/l[i]
mydata2 = cbind.data.frame(ynew, x1, x2, x3, x4, x5, x6)
mod=lm(ynew~., data=mydata2[-c(10,15),])
s=sum(residuals(mod)^2)
t=s/23
like[i]=(2*pi*t)^(-29/2)*exp(-1/(2*t)*s)*prod(y^(l[i]-1))
}
lambda_john=l[which.max(like)]
lambda_john

[1] 6.8
```

We found the value of $\lambda = 6.8$.

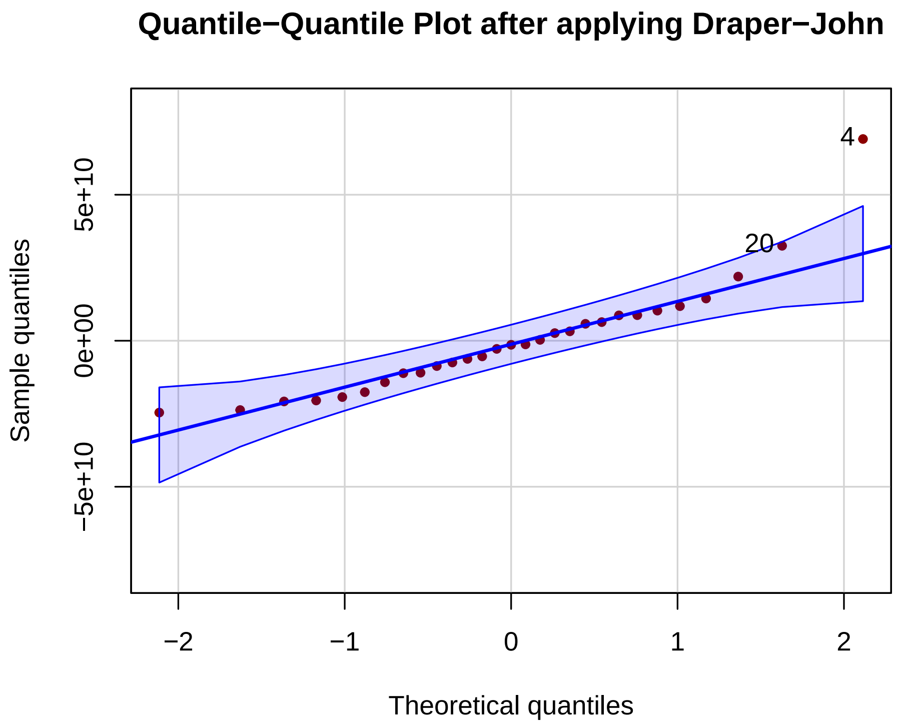
Again we fit our model in this newly transformed data :

```
ynew_john =sign(y)*((abs(y)+1)^lambda_john-1)/lambda_john
mydata3 = cbind.data.frame(ynew_john, x1, x2, x3, x4, x5, x6)
model7 = lm(ynew_john ~., data = mydata3[-c(10,15),])
summary(model7)



Call:
lm(formula = ynew_john ~ ., data = mydata3[-c(10, 15), ])

Residuals:
       Min         1Q     Median         3Q        Max
-2.460e+10 -1.112e+10 -1.400e+09  8.690e+09  6.907e+10

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.882e+11  1.184e+11   4.122 0.000448 ***
x1          -2.627e+09  9.840e+08  -2.670 0.014000 *
x2          -9.566e+08  5.243e+08  -1.825 0.081666 .
x3          -1.267e+10  3.642e+09  -3.477 0.002137 **
x4          -1.017e+09  6.221e+08  -1.634 0.116389
x5          -3.367e+09  1.376e+09  -2.446 0.022881 *
x6           2.993e+09  1.605e+09   1.865 0.075577 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.171e+10 on 22 degrees of freedom
Multiple R-squared:  0.729,Adjusted R-squared:  0.6551
F-statistic: 9.865 on 6 and 22 DF,  p-value: 2.551e-05
```

Also, we draw qqplot to check whether this new transformation handles the non-normality case better or not :

```
resinew_john <- residuals(model7)
with(ToothGrowth, qqPlot(resinew_john,pch=20, col="darkred",
                         xlab = "Theoretical quantiles"
     ,ylab ="Sample quantiles", main="Quantile-Quantile Plot after applying Draper-John",
ylim= c(-8*10^10,8*10^10)))
```



```
 4 20
 4 18
```

From the plot, we can see that after transformation of the response, the residuals still deviate significantly from normality.

We perform the Shapiro Wilk test below :

```
shapiro.test(resinew_john)


Shapiro-Wilk normality test

data:  resinew_john
W = 0.87181, p-value = 0.00221
```

Shapiro-Wilk's test also rejects the normality assumption of the variables and strengthen our belief from the QQplot.

This transformation also doesn't perform well. We do not transform the data because of the following reasons:

1. Normality is disrupted only at the lower tail of the data after removing the influential points.
2. Transforming the response makes it difficult to interpret the relation of the covariates with the response oxy.
3. Box-Cox and John-Draper transformations worsens the situation.
4. Since we have only 29 datapoints which is quite low, we do not transform the covariates.

5 Model Selection

5.1 Stepwise Selection

Stepwise regression is a statistical method used to build a regression model from a set of predictor variables. The procedure involves entering and removing predictors in a stepwise manner into the model until there is no statistically valid reason to enter or remove any more.

First we perform stepwise regression based on Akaike's Information Criterion (AIC) and note down the AIC values for the optimal models given in the different subsequent steps of this selection process. Then, we perform stepwise regression based on the adjusted R^2criteria and note down the adjusted R^2 values and absolute values of (Mallow's $C_p{-}p$) for the optimal models given in the different subsequent steps of this selection process. Based on AIC criteria we choose that model with lowest AIC value and based on adjusted R^2criteria we choose the model with maximum adjusted R^2value whereas based on |Mallow's $C_p{-}p$| criteria we choose the model with least absolute value of (Mallow's $C_p{-}p$).

```
Stepwise Selection Method
-------------------------

Candidate Terms:

1 . x1
2 . x2
3 . x3
4 . x4
5 . x5
6 . x6

 Step 0: AIC = 176.9396
 y ~ 1


Variables Entered/Removed:

                          Enter New Variables
-----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
-----------------------------------------------------------------------
x3           1    137.813    500.480    159.924    0.758        0.749
x5           1    172.595    129.775    530.629    0.197        0.167
x6           1    174.076    101.960    558.444    0.154        0.123
x4           1    174.558     92.602    567.802    0.140        0.108
x2           1    176.862     45.666    614.739    0.069        0.035
x1           1    177.297     36.356    624.048    0.055        0.020
-----------------------------------------------------------------------

- x3 added


 Step 1 : AIC = 137.8132
 y ~ x3

                          Enter New Variables
-----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
-----------------------------------------------------------------------
x5           1    136.140    519.506    140.898    0.787        0.770
x6           1    137.705    511.691    148.713    0.775        0.757
x1           1    137.743    511.497    148.907    0.775        0.757
x2           1    138.470    507.719    152.685    0.769        0.751
x4           1    139.812    500.484    159.920    0.758        0.739
```

```
-----------------------------------------------------------------------

- x5 added


 Step 2 : AIC = 136.1399
 y ~ x3 + x5

                      Remove Existing Variables
-----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
-----------------------------------------------------------------------
x5           1    137.813    500.480    159.924    0.758        0.749
x3           1    172.595    129.775    530.629    0.197        0.167
-----------------------------------------------------------------------

                          Enter New Variables
-----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
-----------------------------------------------------------------------
x1           1    130.659    551.544    108.860    0.835        0.815
x6           1    136.622    526.689    133.715    0.798        0.773
x2           1    137.227    523.871    136.533    0.793        0.768
x4           1    137.849    520.912    139.493    0.789        0.763
-----------------------------------------------------------------------

- x1 added


 Step 3 : AIC = 130.6587
 y ~ x3 + x5 + x1

                      Remove Existing Variables
-----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
-----------------------------------------------------------------------
x1           1    136.140    519.506    140.898    0.787        0.770
x5           1    137.743    511.497    148.907    0.775        0.757
x3           1    167.493    245.032    415.372    0.371        0.323
-----------------------------------------------------------------------

                          Enter New Variables
-----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
-----------------------------------------------------------------------
x2           1    128.991    564.477     95.927    0.855        0.831
x6           1    131.726    554.989    105.415    0.840        0.814
x4           1    132.658    551.547    108.858    0.835        0.808
-----------------------------------------------------------------------

- x2 added


 Step 4 : AIC = 128.9909
 y ~ x3 + x5 + x1 + x2

                      Remove Existing Variables
-----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
-----------------------------------------------------------------------
x2           1    130.659    551.544    108.860    0.835        0.815
x5           1    136.822    525.765    134.640    0.796        0.772
x1           1    137.227    523.871    136.533    0.793        0.768
x3           1    165.911    293.298    367.106    0.444        0.377
-----------------------------------------------------------------------

                          Enter New Variables
----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq      RSS      R-Sq     Adj. R-Sq
----------------------------------------------------------------------
x6           1    128.894    571.168    89.236    0.865        0.836
x4           1    130.933    564.668    95.737    0.855        0.824
----------------------------------------------------------------------

- x6 added


 Step 5 : AIC = 128.8942
 y ~ x3 + x5 + x1 + x2 + x6

                      Remove Existing Variables
```

```
----------------------------------------------------------------------
Variable     DF      AIC      Sum Sq       RSS      R-Sq     Adj. R-Sq
----------------------------------------------------------------------
x6           1    128.991    564.477     95.927    0.855        0.831
x2           1    131.726    554.989    105.415    0.840        0.814
x5           1    132.786    551.066    109.338    0.834        0.807
x1           1    136.943    534.211    126.193    0.809        0.777
x3           1    167.343    300.421    359.984    0.455        0.364
----------------------------------------------------------------------

                          Enter New Variables
---------------------------------------------------------------------
Variable     DF      AIC      Sum Sq      RSS      R-Sq     Adj. R-Sq
---------------------------------------------------------------------
x4           1    130.806    571.438    88.966    0.865        0.829
---------------------------------------------------------------------


No more variables to be added or removed.

Final Model Output
------------------

                        Model Summary
-------------------------------------------------------------
R                       0.930       RMSE               1.970
R-Squared               0.865       Coef. Var          4.210
Adj. R-Squared          0.836       MSE                3.880
Pred R-Squared          0.813       MAE                1.245
-------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                               ANOVA
-------------------------------------------------------------------
               Sum of
              Squares        DF    Mean Square      F         Sig.
-------------------------------------------------------------------
Regression    571.168         5        114.234    29.443    0.0000
Residual       89.236        23          3.880
Total         660.404        28
-------------------------------------------------------------------

                                  Parameter Estimates
------------------------------------------------------------------------------------------
      model       Beta    Std. Error    Std. Beta      t        Sig      lower      upper
------------------------------------------------------------------------------------------
(Intercept)    108.910        10.602                 10.273    0.000    86.979    130.841
         x3     -2.513         0.301       -0.716    -8.354    0.000    -3.135     -1.890
         x5     -0.284         0.125       -0.620    -2.276    0.032    -0.542     -0.026
         x1     -0.268         0.087       -0.272    -3.086    0.005    -0.448     -0.088
         x2     -0.097         0.047       -0.169    -2.042    0.053    -0.194      0.001
         x6      0.191         0.146        0.361     1.313    0.202    -0.110      0.492
------------------------------------------------------------------------------------------


                               Stepwise Summary
-----------------------------------------------------------------------------
Variable     Method       AIC        RSS      Sum Sq      R-Sq      Adj. R-Sq
-----------------------------------------------------------------------------
x3         addition    137.813    159.924    500.480    0.75784      0.74887
x5         addition    136.140    140.898    519.506    0.78665      0.77024
x1         addition    130.659    108.860    551.544    0.83516      0.81538
x2         addition    128.991     95.927    564.477    0.85474      0.83054
x6         addition    128.894     89.236    571.168    0.86488      0.83550
-----------------------------------------------------------------------------
[1] 130.8063
```

```
Subset selection object
Call: regsubsets.formula(y ~ ., data = mydata[-c(10, 15), ], method = "seqrep",
    nbest = 1)
6 Variables  (and intercept)
   Forced in Forced out
x1     FALSE      FALSE
x2     FALSE      FALSE
x3     FALSE      FALSE
x4     FALSE      FALSE
x5     FALSE      FALSE
x6     FALSE      FALSE
```

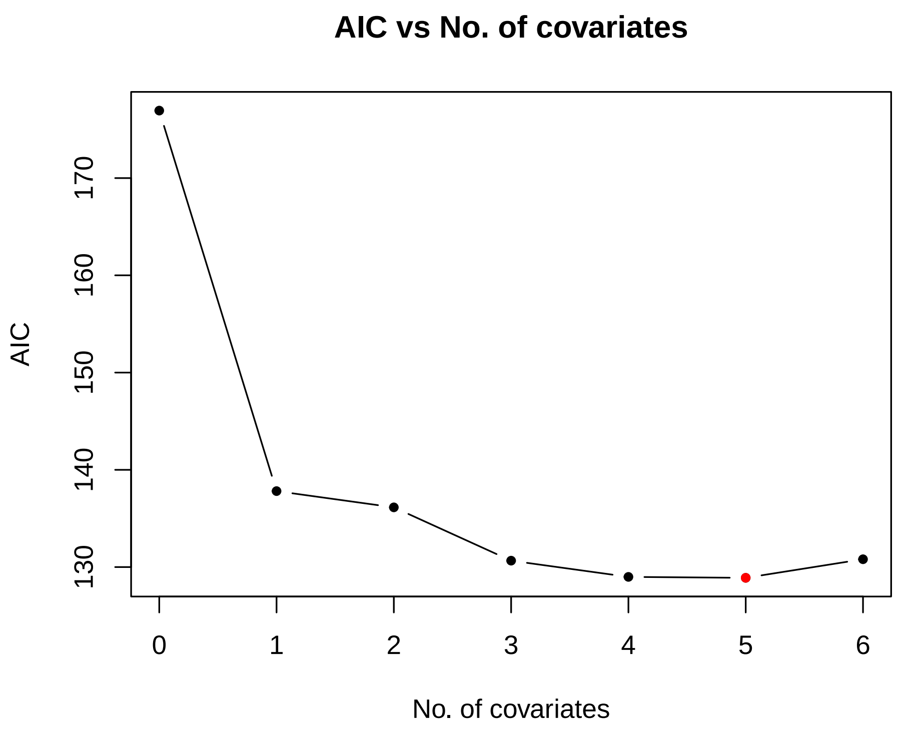
```
1 subsets of each size up to 6
Selection Algorithm: 'sequential replacement'
         x1  x2  x3  x4  x5  x6
1  ( 1 ) " " " " "*" " " " " " "
2  ( 1 ) " " " " "*" " " "*" " "
3  ( 1 ) "*" "*" "*" " " " " " "
4  ( 1 ) "*" "*" "*" "*" " " " "
5  ( 1 ) "*" "*" "*" " " "*" "*"
6  ( 1 ) "*" "*" "*" "*" "*" "*"
[1] 0.7488701 0.7702370 0.7716602 0.7653846 0.8355016 0.8285450
[1] 13.54686875  9.84201802  9.29441861  9.84109579  0.06680304  1.00000000
```

Table of various optimal models and values of their Goodness of fit criteria (Adjusted R^2), criteria based on Prediction error ($|$Mallow's $C_p - p|$), criteria based on Distributional discrepencies (AIC) :

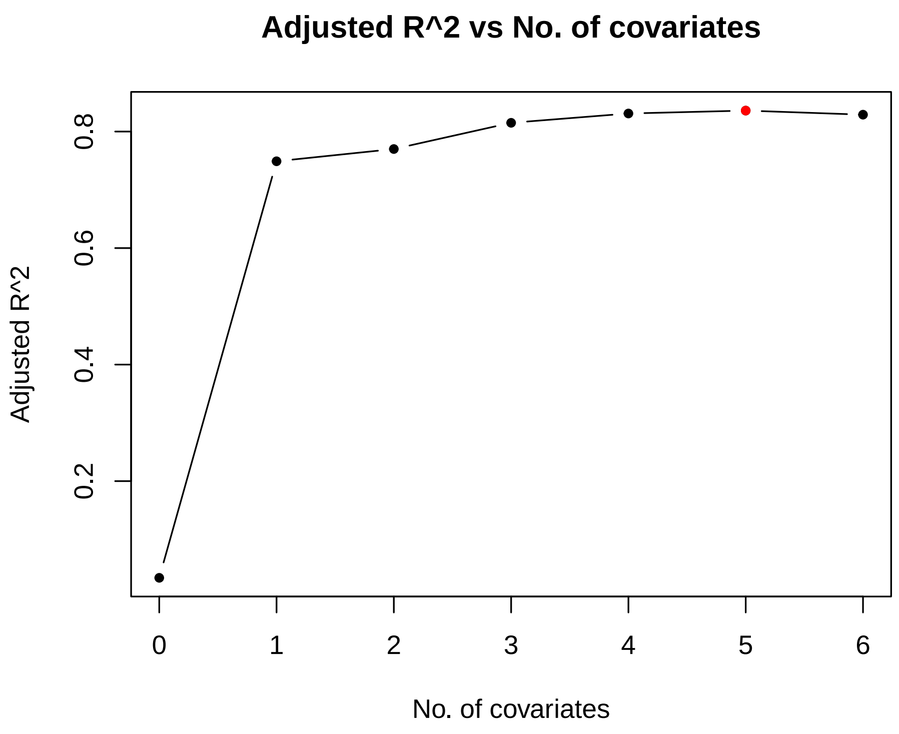
Model	AIC Value	Adjusted R^2	$\lvert$Mallow's $C_p - p\rvert$
$Y = \beta_0 + \epsilon$	176.940	0.034	141.7312
$Y = \beta_0 + \beta_3 x_3 + \epsilon$	137.813	0.749	13.547
$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \epsilon$	136.140	0.770	9.842
$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \epsilon$	130.659	0.815	2.919
$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$	128.991	0.831	0.721
$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \epsilon$	128.894	0.836	0.067
$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \beta_4 x_4 + \epsilon$	130.806	0.829	1.000

Now we plot the values of the above criterias corresponding to the optimal models :

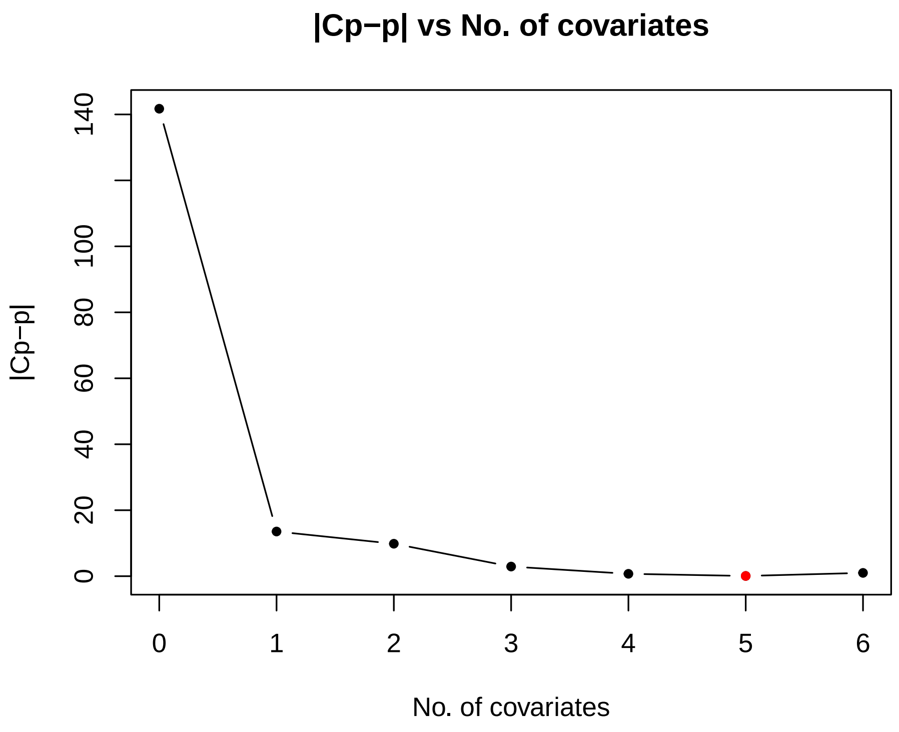
```r
P=0:6
aic <- c(176.940
, 137.813,
136.140,
130.659,
128.991,
128.894
, 130.806
)
adj_r <- c(0.034, 0.749, 0.770, 0.815, 0.831, 0.836, 0.829)
mal_p <- c(141.7312, 13.547, 9.842, 2.919, 0.721, 0.067, 1.000)
par(mfrow=c(1,1))

ind = which.min(aic)
plot(P, aic, pch = 20,type="b",ylab="AIC",xlab="No. of covariates",
main="AIC vs No. of covariates")
points(ind-1,aic[ind],col="red",pch=20)
```



```r
ind = which.max(adj_r)
plot(P, adj_r, pch = 20,type="b",ylab="Adjusted R^2",xlab="No. of covariates",
main="Adjusted R^2 vs No. of covariates")
points(ind-1,adj_r[ind],col="red",pch=20)
```



```
ind = which.min(mal_p)
plot(P, mal_p, pch = 20,type="b",ylab="|Cp-p|",xlab="No. of covariates",main="|Cp-p| vs No. of covariates")
points(ind-1,mal_p[ind],col="red",pch=20)
```



From the above plots and also by comparing the values of different criterias for model selection (AIC, Adjusted R^2, |Mallow's $C_p - p$|) while performing the stepwise selection method, we find that our final optimal model is : $Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \epsilon$.

5.2 Best Subset Selection

Again we perforn the Best subset selection method to check whether we get the same optimal model if we go by best subset selection rather than stepwise selection .

```
X1 <- mydata[-c(10,15),]
names(X1) <- c("R","V1","V2","V3","V4","V5","V6")
models<-list()
models[["V1"]]<-lm(R~V1,X1)
models[["V2"]]<-lm(R~V2,X1)
models[["V3"]]<-lm(R~V3,X1)
models[["V4"]]<-lm(R~V4,X1)
models[["V5"]]<-lm(R~V5,X1)
models[["V6"]]<-lm(R~V6,X1)
models[["V12"]]<-lm(R~V1+V2,X1)
models[["V13"]]<-lm(R~V1+V3,X1)
models[["V14"]]<-lm(R~V1+V4,X1)
models[["V15"]]<-lm(R~V1+V5,X1)
models[["V16"]]<-lm(R~V1+V6,X1)
models[["V23"]]<-lm(R~V2+V3,X1)
models[["V24"]]<-lm(R~V2+V4,X1)
models[["V25"]]<-lm(R~V2+V5,X1)
models[["V26"]]<-lm(R~V2+V6,X1)
models[["V34"]]<-lm(R~V3+V4,X1)
models[["V35"]]<-lm(R~V3+V5,X1)
models[["V36"]]<-lm(R~V3+V6,X1)
models[["V45"]]<-lm(R~V4+V5,X1)
models[["V46"]]<-lm(R~V4+V6,X1)
models[["V56"]]<-lm(R~V5+V6,X1)
models[["V123"]]<-lm(R~V1+V2+V3,X1)
models[["V124"]]<-lm(R~V1+V2+V4,X1)
models[["V125"]]<-lm(R~V1+V2+V5,X1)
models[["V126"]]<-lm(R~V1+V2+V6,X1)
models[["V134"]]<-lm(R~V1+V3+V4,X1)
models[["V135"]]<-lm(R~V1+V3+V5,X1)
models[["V136"]]<-lm(R~V1+V3+V6,X1)
models[["V145"]]<-lm(R~V1+V4+V5,X1)
models[["V146"]]<-lm(R~V1+V4+V6,X1)
models[["V156"]]<-lm(R~V1+V5+V6,X1)
models[["V234"]]<-lm(R~V2+V3+V4,X1)
models[["V235"]]<-lm(R~V2+V3+V5,X1)
models[["V236"]]<-lm(R~V2+V3+V6,X1)
models[["V245"]]<-lm(R~V2+V4+V5,X1)
models[["V246"]]<-lm(R~V2+V4+V6,X1)
models[["V256"]]<-lm(R~V2+V5+V6,X1)
models[["V345"]]<-lm(R~V3+V4+V5,X1)
models[["V346"]]<-lm(R~V3+V4+V6,X1)
models[["V356"]]<-lm(R~V3+V5+V6,X1)
models[["V456"]]<-lm(R~V4+V5+V6,X1)
models[["V1234"]]<-lm(R~V1+V2+V3+V4,X1)
models[["V1235"]]<-lm(R~V1+V2+V3+V5,X1)
models[["V1236"]]<-lm(R~V1+V2+V3+V6,X1)
models[["V1245"]]<-lm(R~V1+V2+V4+V5,X1)
models[["V1246"]]<-lm(R~V1+V2+V4+V6,X1)
models[["V1256"]]<-lm(R~V1+V2+V5+V6,X1)
models[["V1345"]]<-lm(R~V1+V3+V4+V5,X1)
models[["V1346"]]<-lm(R~V1+V3+V4+V6,X1)
models[["V1356"]]<-lm(R~V1+V3+V5+V6,X1)
models[["V1456"]]<-lm(R~V1+V4+V5+V6,X1)
models[["V2345"]]<-lm(R~V2+V3+V4+V5,X1)
models[["V2346"]]<-lm(R~V2+V3+V4+V6,X1)
models[["V2356"]]<-lm(R~V2+V3+V5+V6,X1)
```

```
models[["V2456"]]<-lm(R~V2+V4+V5+V6,X1)
models[["V3456"]]<-lm(R~V3+V4+V5+V6,X1)
models[["V12345"]]<-lm(R~V1+V2+V3+V4+V5,X1)
models[["V12346"]]<-lm(R~V1+V2+V3+V4+V6,X1)
models[["V12456"]]<-lm(R~V1+V2+V4+V5+V6,X1)
models[["V13456"]]<-lm(R~V1+V3+V4+V5+V6,X1)
models[["V12356"]]<-lm(R~V1+V2+V3+V5+V6,X1)
models[["V23456"]]<-lm(R~V2+V3+V4+V5+V6,X1)
models[["V123456"]]<-lm(R~V1+V2+V3+V4+V5+V6,X1)

mnames<- factor(names(models),levels = names(models))
```

We calculate the values for different criterias for model selection for all possible combinations (65) of model below and print that model which is optimal based on those criterias :

✍ R^2_{adj}

```
adj.R2 <- sapply(models, function(fit) summary(fit)$adj.r.squared)
adj.R2[which.max(adj.R2)]


   V12356
0.8355016
```

✍ Absolute value of (Mallow's $C_p - p$)

```
    V12356
0.06680304
```

✍ Akaike information criterion (AIC)

```
AIC <- sapply(models, function(fit) AIC(fit))
AIC[which.min(AIC)]


  V12356
128.8942
```

✍ Leave-One-Out CV

```
CV = NULL
for(i in 1:15)
{
  X_mdl_mat = model.matrix(models[[i]])
  head(X_mdl_mat)
  Y_vec = X1$V5
  H = X_mdl_mat%*%solve(t(X_mdl_mat)%*%X_mdl_mat)%*%t(X_mdl_mat)
  h = diag(H)
  n_h = nrow(X1)
  CV[i] = (1/n_h)*sum((Y_vec-H%*%Y_vec)^2/(1-h)^2)
}
mnames[which.max(CV)]


[1] V2
63 Levels: V1 V2 V3 V4 V5 V6 V12 V13 V14 V15 V16 V23 V24 V25 V26 V34 ... V123456


CV[which.max(CV)]


[1] 121.9613
```

Hence, we see that even using Best subset selection method, we get our final model to be : $Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \epsilon$.

Conclusion

Now, we write down the optimals model and different model selection criterions with corresponding values :-

Criterions	Optimum Model	Value
R^2_{adj}	$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \epsilon$.	0.836
$\lvert$Mallow's $C_{p-p}\rvert$	$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \epsilon$.	0.067
AIC	$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \epsilon$.	128.894
$CV(1)$	$Y = \beta_0 + \beta_3 x_3 + \beta_5 x_5 + \beta_1 x_1 + \beta_2 x_2 + \beta_6 x_6 + \epsilon$.	121.961

Hence, clearly this indicates among all the linear models, $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_5 x_5 + \beta_6 x_6 + \epsilon$ is optimum based on several criterions.

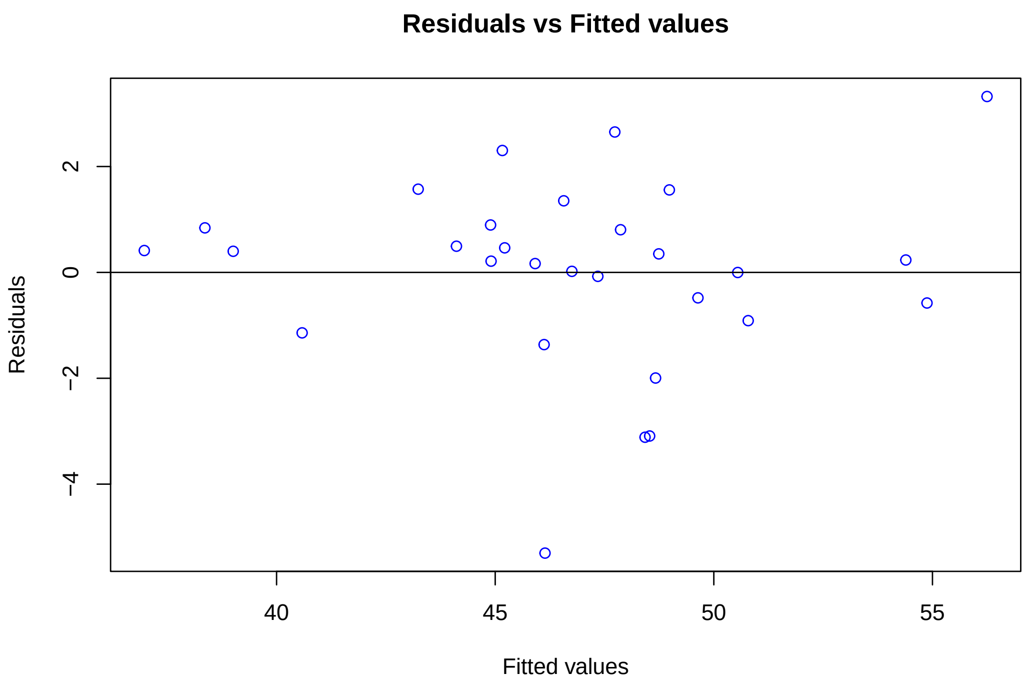
In terms of terminology of the given dataset, the optimum predictors of Y =Oxygen consumed are Age, Weight, Runtime, Runpulse, and Maxpulse.

So the optimum fitted model can be written as :-

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_5 x_5 + \beta_6 x_6 + \epsilon$$

We again check the error assumptions fitting the multiple linear regression model as described above:

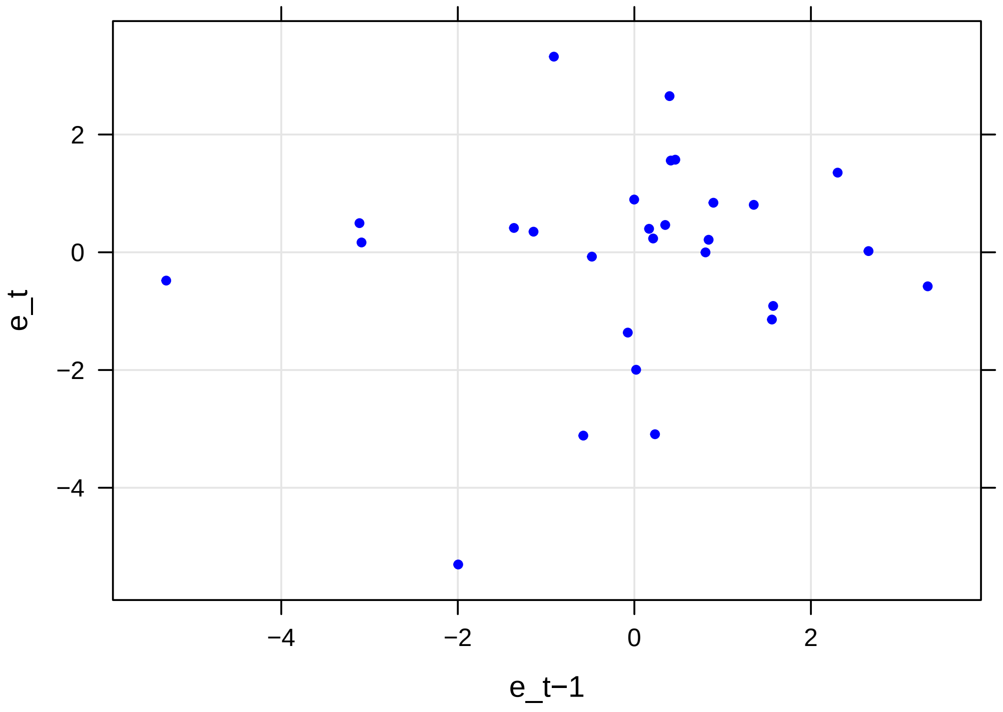
```
mnew= lm(y~x1+x2+x3+x5+x6, data=mydata[-c(10,15),])
plot(fitted(mnew), residuals(mnew), col = "blue", xlab="Fitted values",
ylab="Residuals", main="Residuals vs Fitted values")
abline(h=0)
```



```
bptest(mnew)


studentized Breusch-Pagan test

data:  mnew
BP = 2.9379, df = 5, p-value = 0.7096
```

We observe that the residuals are more or less randomly scattered around the residuals=0 line and even the confirmatory test accepts the null hypothesis that there is no heteroscedasticity with a very high p-value.

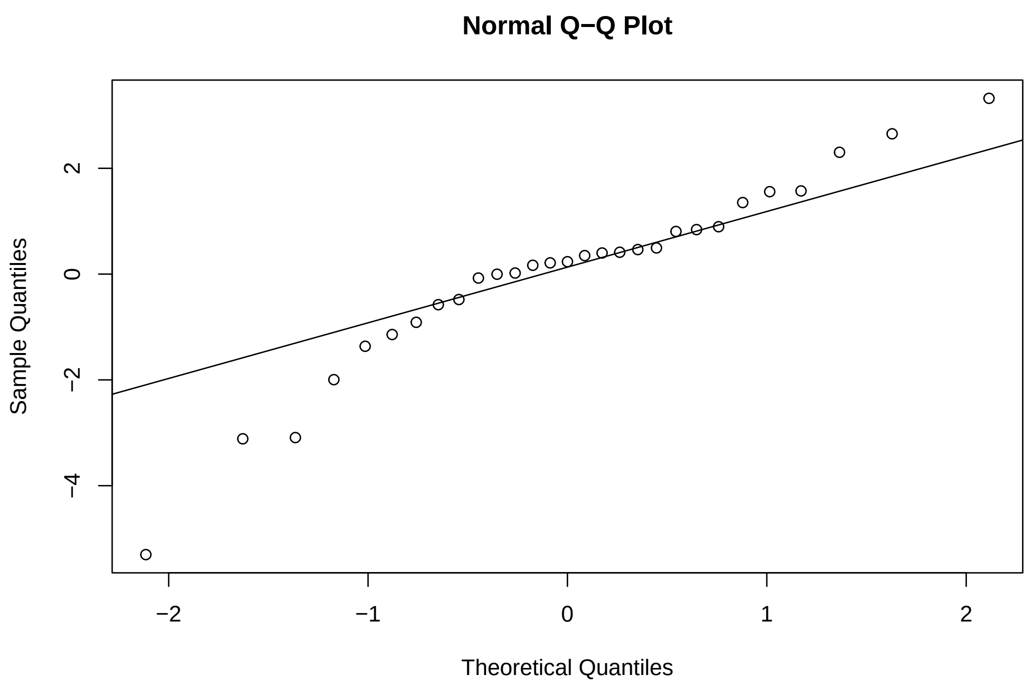
```
xyplot(as.vector(residuals(mnew))[-29]~as.vector(residuals(mnew))[-1],pch = 20,
ylab = "e_t",xlab = "e_t-1", col = "blue", grid=TRUE)
```



```
bgtest(mnew)


Breusch-Godfrey test for serial correlation of order up to 1

data:  mnew
LM test = 0.69812, df = 1, p-value = 0.4034

dwtest(mnew)


Durbin-Watson test

data:  mnew
DW = 1.6908, p-value = 0.1914
alternative hypothesis: true autocorrelation is greater than 0
```

There is no observable pattern in the plot, so we do not find any autocorrelation in the residuals. The Durbin Watson test and the Breusch Godfrey test also fail to reject the null hypothesis.

```
qqnorm(residuals(mnew))
qqline(residuals(mnew))
```

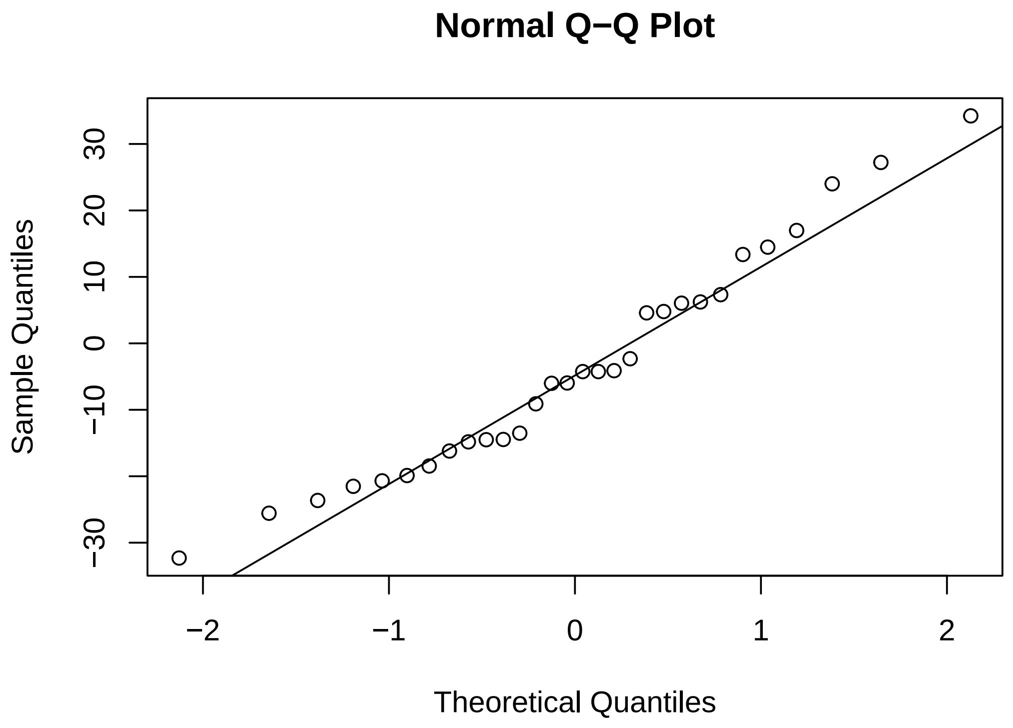


```
shapiro.test(residuals(mnew))


Shapiro-Wilk normality test

data:  residuals(mnew)
W = 0.93258, p-value = 0.06416
```

We observe that the Shapiro-Wilk's test fails to reject the null hypothesis indicating that the errors are normally distributed but from the QQplot it seems that the data is non-normal. This may happen because of less number of data-points (29). We shall demonstrate by simulation to make our point clear:

We generate 30 data-points from a normal distribution using rnorm function in R. Then we make the QQplot for these 30 data-points.

```
d=rnorm(30,0,15)
qqnorm(d)
qqline(d)
```



Thus, we see that having a very small number of datapoints from a normal distribution with a fairly large variance can lead us to wrong conclusions from just the plots.

6 Alternative Estimation Methods

After model selection we see that neither of the collinear variables (runpulse (x_5) and maxpulse (x_6)) were excluded from the model. Hence, we cannot use the Ordinary Least Squares method for estimating the model parameters. So we opt for Ridge and Lasso estimation methods.

6.1 Lasso Estimation

Ridge regression is a method we can use to fit a regression model when multicollinearity is present in the data. Ridge regression seeks to minimize the following:

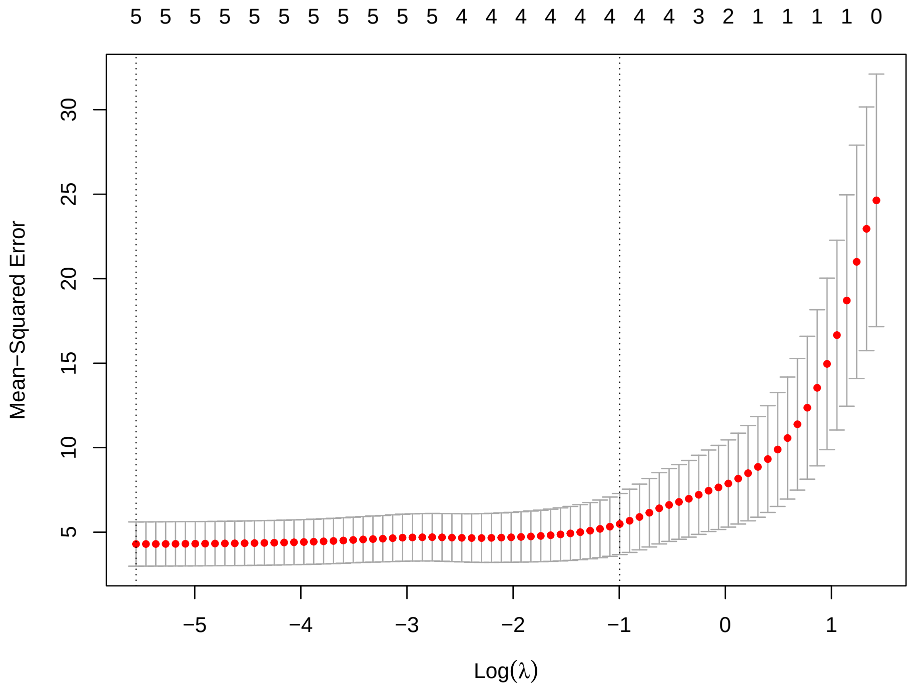
$RSS + |\beta_j|$

We perform lasso estimation in R using the glmnet library:

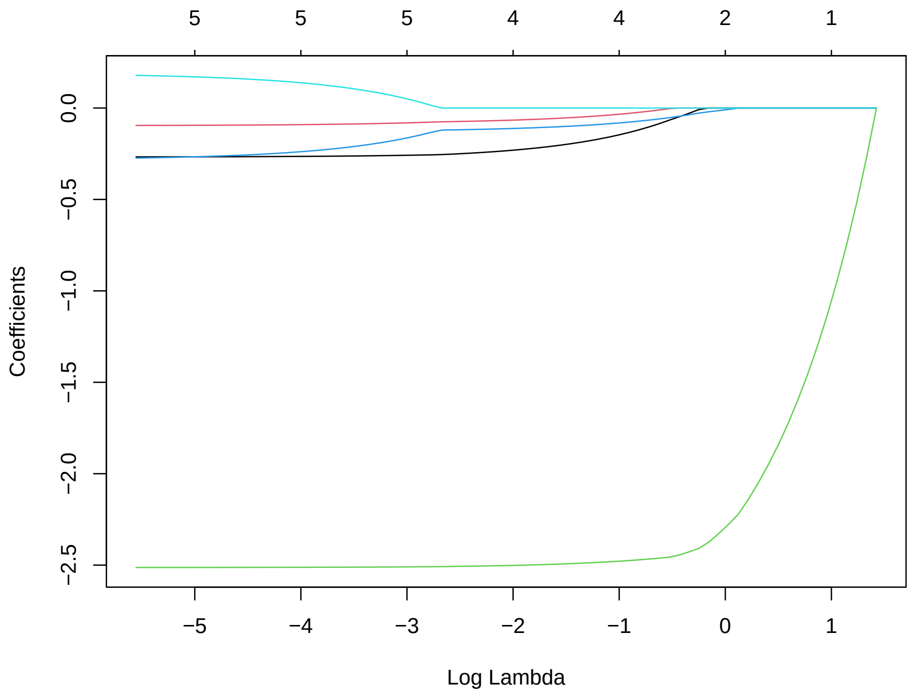
```
library(glmnet)
yn=mydata[-c(10,15),1]
xn=data.matrix(mydata[-c(10,15),-c(1,5)])
model7=glmnet(xn, yn,alpha=1)
summary(model7)
```

```
          Length Class     Mode
a0         76    -none-    numeric
beta      380    dgCMatrix S4
df         76    -none-    numeric
dim         2    -none-    numeric
lambda     76    -none-    numeric
```

```
dev.ratio  76    -none-    numeric
nulldev     1    -none-    numeric
npasses     1    -none-    numeric
jerr        1    -none-    numeric
offset      1    -none-    logical
call        4    -none-    call
nobs        1    -none-    numeric

cvmodel=cv.glmnet(xn,yn,alpha=1)
bl=cvmodel$lambda.min
bl

[1] 0.003874284

plot(cvmodel)
```



```
bmodel=glmnet(xn,yn,alpha=1,lambda=bl)
coef(bmodel)

6 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) 109.10869077
x1           -0.26721016
x2           -0.09523417
x3           -2.51297678
x5           -0.27340400
x6            0.17866633

plot(model7,xvar="lambda")
```



```
y_pr=predict(model7,s=bl,newx=xn)
sst=sum((yn-mean(yn))^2)
sse=sum((yn-y_pr)^2)
rsq=1-sse/sst; rsq

[1] 0.8648295

adjr2=1-(28)*(1-rsq)/24; adjr2

[1] 0.8423011

summary(lm(y~x1+x2+x3+x5+x6, data=mydata[-c(10,15),-5]))


Call:
lm(formula = y ~ x1 + x2 + x3 + x5 + x6, data = mydata[-c(10,
    15), -5])

Residuals:
    Min      1Q  Median      3Q     Max
-5.3032 -0.5782  0.2341  0.8414  3.3231

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 108.90986   10.60162  10.273 4.59e-10 ***
x1           -0.26801    0.08684  -3.086  0.00521 **
x2           -0.09655    0.04728  -2.042  0.05278 .
x3           -2.51272    0.30079  -8.354 2.03e-08 ***
x5           -0.28420    0.12486  -2.276  0.03246 *
x6            0.19116    0.14557   1.313  0.20206
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.97 on 23 degrees of freedom
Multiple R-squared:  0.8649,Adjusted R-squared:  0.8355
F-statistic: 29.44 on 5 and 23 DF,  p-value: 2.827e-09
```

We compare the R^2_{adj} values for this model and the previous model (done after model selection) and see that the $R^2_{lasso} = 0.842$ $R^2_{previous} = 0.835$. Thus we conclude that the model with lasso estimates can explain the variability in our data better than the model using ordinary least squares estimator.

We check if Ridge estimation gives better estimates than Lasso estimation.

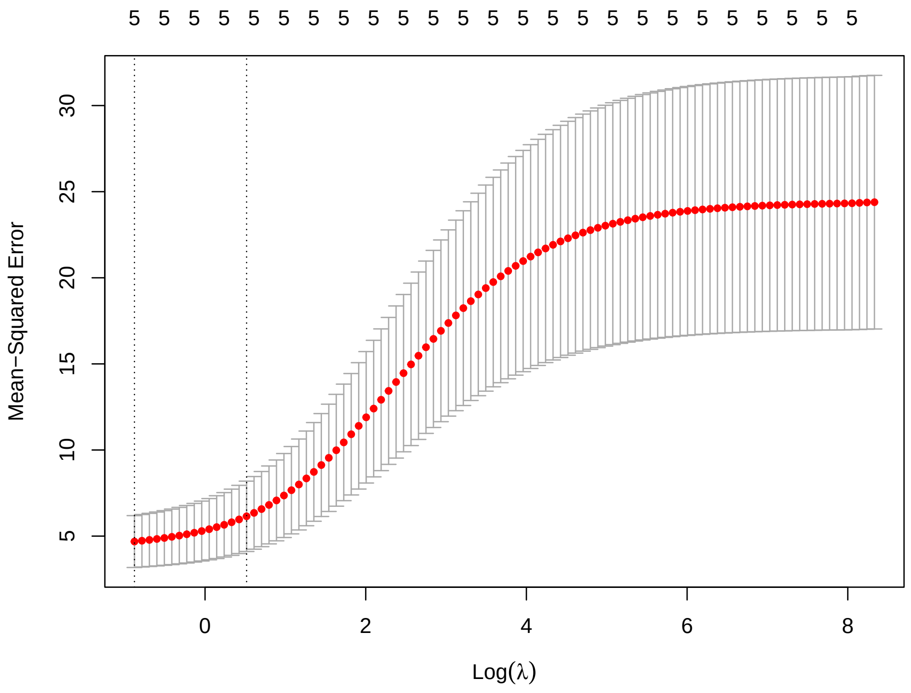
6.2 Ridge Estimation

Ridge regression is a method we can use to fit a regression model when multicollinearity is present in the data. Ridge regression seeks to minimize the following:

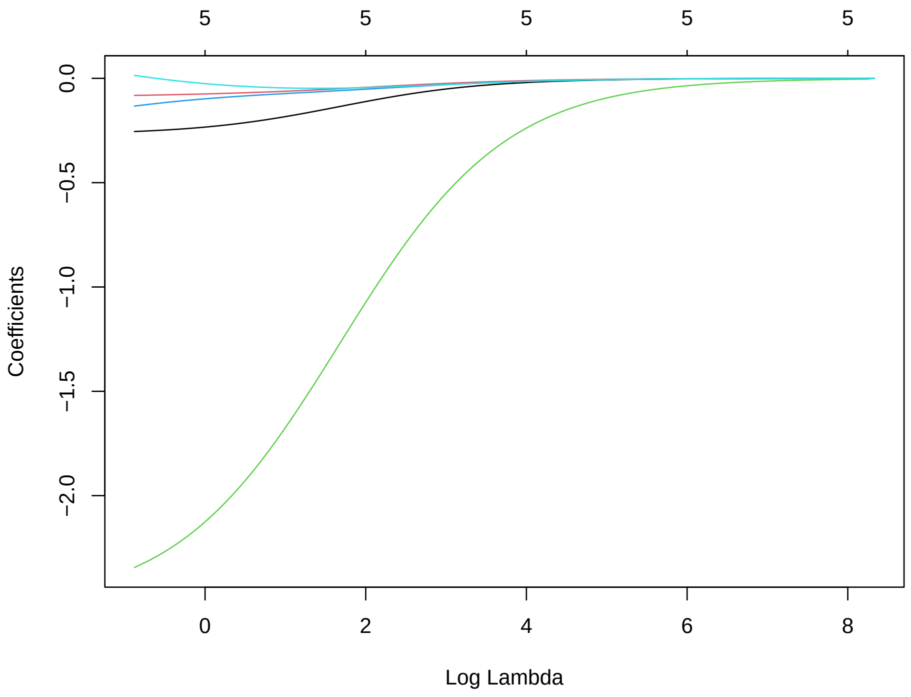
$RSS + \lambda\beta_j^2$

We perform ridge regression in R using the glmnet library:

```
yp=mydata[-c(10,15),1]
xp=data.matrix(mydata[-c(10,15),-c(1,5)])
model8=glmnet(xp, yp,alpha=0)
summary(model8)

          Length Class     Mode
a0        100    -none-    numeric
beta      500    dgCMatrix S4
df        100    -none-    numeric
dim         2    -none-    numeric
lambda    100    -none-    numeric
dev.ratio 100    -none-    numeric
nulldev     1    -none-    numeric
npasses     1    -none-    numeric
jerr        1    -none-    numeric
offset      1    -none-    logical
call        4    -none-    call
nobs        1    -none-    numeric

cvmodel1=cv.glmnet(xp,yp,alpha=0)
bl1=cvmodel1$lambda.min
bl1

[1] 0.4154267

plot(cvmodel1)
```



```
bmodel1=glmnet(xp,yp,alpha=0,lambda=bl1)
coef(bmodel1)

6 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) 110.39982942
x1           -0.25482082
x2           -0.08195607
x3           -2.34432327
x5           -0.13298071
x6            0.01413544

plot(model8,xvar="lambda")
```



```
y_pr1=predict(model8,s=bl1,newx=xp)
sst1=sum((yp-mean(yp))^2)
sse1=sum((yp-y_pr1)^2)
rsq1=1-sse1/sst1; rsq1

[1] 0.8523082

adjr21=1-(28)*(1-rsq1)/24; adjr21

[1] 0.8276929
```

We observe that the adjusted R^2 values for the Ridge estimate is 0.827 and that of the Lasso estimate was 0.842. So Lasso estimators explains the data better than the Ridge estimators.

7 Conclusion

Thus, our final fitted regression model is :

$$\hat{y} = 109.108 - 0.267x_1 - 0.095x_2 - 2.512x_3 - 0.273x_5 + 0.178x_6$$

The coefficients in the model represent the change in the dependent variable (y) associated with a one-unit change in the corresponding independent variable, while holding all other variables constant.

1. Intercept (109.108): The intercept represents the expected value of y when all predictor variables are set to zero.
2. Age (x_1, coefficient: -0.267): For each one-year increase in age, the average oxygen consumption decreases (average fitness of a person increases) by 0.267 units, holding all other variables constant.

3. Weight (x_2, coefficient: -0.095): For each one-unit increase in weight, the average oxygen consumption decreases (average fitness of a person increases) by 0.095 units, holding all other variables constant.
4. Runtime (x_3, coefficient: -2.512): For each one-unit increase in runtime, the average oxygen consumption decreases (average fitness of a person increases) by 2.512 units, holding all other variables constant.
5. Runpulse (x_5, coefficient: -0.273): For each one-unit increase in runpulse, the average oxygen consumption decreases (average fitness of a person increases) by 0.273 units, holding all other variables constant.
6. Maxpulse (x_6, coefficient: 0.178): For each one-unit increase in maxpulse, the average oxygen consumption increases (average fitness of a person decreases) by 0.178 units, holding all other variables constant.

8 Acknowledgement

We want to extend our heartfelt gratitude to our esteemed professor, Dr. Swagata Nandi, for her invaluable assistance with our presentation and for providing us with this fantastic learning opportunity.