REGRESSION ANALYSIS AND PREDICTIVE MODELING

DSM-1003

FEBRUARY 24, 2022

MOHAMMAD WASIQ

MS (Data Science)

Regression Analysis and Predictive Modeling
DSM-1003

Table of Contents

1 Regression Analysis and Predictive Modeling

Book :-: **An Introduction to Statistical Learning with Applications in R**
Teacher :-: **Prof. Ahmed ur Rehman Sir**
Composer :-: **Mohammad Wasiq** *(Data Science)*

2 Linear Models

- **Linear Models:** Summarising the data in the forms of equation is known as Linear Models.
- **Regression Analysis**: Regression Analysis is a simple method for ivestigation relationship among variables.

2.1 Simple Linear Regression / SLR

Model **:-** $\boldsymbol{y = \beta_0 + \beta_1 x_1 + \epsilon}$
where , $\boldsymbol{y}$ is `Response variable / Outcome of Study / Dependent Variable`
$\boldsymbol{\beta_0}$ is `Intercept`
$\boldsymbol{\beta_1}$ is `Slope` i.e. $\boldsymbol{\frac{\Delta y}{\Delta x}}$
$\boldsymbol{x_1}$ is `Explanatory variable / Predictor / Regressor / Independent Variable`
$\boldsymbol{\epsilon}$ is `Error`

1. **Response Variable**(y)**:** A response variable measures an outcome of a study.

2. **Explanatory Variable**(x)**:** Explanatory variable explains or cause change in the response variable.
   Ex- Beer Drinking and Blood Alcohol Level. How does drinking beer affect the level of alcohol in our blood.
   Model: Blood Alcohol Level (y)=$\beta_0(intercept) + \beta_1(slope) * Beer - Drink(x) + \epsilon$

3. **Slope**(β_1)**:** $\beta_1 = \frac{\Delta y}{\Delta x}$ is slope, the amount by which y changes, when x changes by one unit. The slope is an important numerical description of the relationship b/w two variables. Ex- $Weight = \widehat{\beta_0} + \widehat{\beta_1} Age \Rightarrow$ Weight(kg) =3+0.2 Age(yrs) *Interpretation*- If age changes by one unit(i.e. 1 year) then weight changes by 0.2 kg.

4. **Intercept**(β_0)**:** β_0 is the intercept , the value of y when $x = 0$. Prediction: we can use a regression line to predict the response y for a specific value of the explanatory variable x.

5. **Residual:* Observed(y) - Predict(y) $\Rightarrow (y - \hat{y})$

6. **Assumption of Linear Model**
Linear in Parameter: The model (A) is linear in the parameters β_0 & β_1.
Random Sampling: We have a random sample of n observation i.e. we draw samples from the population by simple random sampling method .
Normality: The error will follow normal distribution with $mean = 0$ & $variance = \sigma^2$ i.e. X~ $N(0, \sigma^2)$
Homoscedasticity: The error has the same variance given any values of the explanatory variables.
i.e. Variance is constant at every value x. $\Rightarrow V(e|x_1, x_2 \dots . x_n) = \sigma^2$
No Perfect Multicollinearity/No Auto Correlation: In the Model(A), there is no perfect linear relationship b/w regression.
(That's why we call x is independent variable) i.e. $Cov(e_i, e_j) = 0$

7. **Some Other Definition**:-
Error: Error of the dataset is the difference b/w the observed value and the unobserved value.
Residuals: Residual is calculated after running the regression model and is the difference b/w observed value and the estimated value.
$e_i = (y_i - \widehat{y_i}) = y_i - (\widehat{\beta_0} + \widehat{\beta_1} x)$
Sum of Squares: Sum of squares is one of the most important output in regression analysis. The general rule is that a smaller sum of squares indicate a better model,as there is less variation in the data.
Coefficient of Determination / $R^2 - Value$ It can be noted that a fitted model can be said to be good model when residuals are small for the measure of Goodness of Model, we use the following formula: $\boldsymbol{R^2 = \frac{SSR}{SST} = 1 - \frac{SS_{res}}{SST}}$**,** this is called, the coefficient of determination. The ratio $\frac{SSR}{SST}$ describe the proportion of variability i.e. explained by the regression in relation to the total varialbility of y . The ratio $\frac{SS_{res}}{SST}$ describe the proportion of variability that is not explained by the regression. The value of R^2 lies $0 \le R^2 \le 1$. $R^2 = 0$, indicates that poorest fit of the model. $R^2 = 1$, indicates that best fit of the model. $R^2 = 0.95$, indicates that 95% of the variation in y is explained by R^2. In simple words, the model is 95% good .
Drawbacks of R^2- As R^2 always increase with an increase in the no. of explanatory vaiables in the model.The main drawback of this property is that even when the

irrelevant explanatory variables. are odded in the model, R^2 still increases. This indicates that the model is getting better, which is not really correct. With a purpose of correction in the overly optimstic picture ,Adjusted R^2, denoted by R^2_{adj} is used , which is defined as: $\boldsymbol{R^2_{adj} = 1 - \frac{SS_{res}/(n-k-1)}{SST/(n-1)}}$ OR $\boldsymbol{R^2_{adj} = 1 - \frac{SS_{res}}{SST} \times \frac{(n-1)}{(n-k-1)}}$ OR $\boldsymbol{R^2_{adj} = 1 - \frac{n-1}{n-k-1}(1 - R^2)}$

8. **Types of Sum of Square**:-
   (i).*Total Sum of Square(SST)*: $\sum_{i=1}^{n}(y_i - \bar{y})^2$ where y_i=value in a sample and $\bar{y}$=mean value of the sample
   (ii). *Regression Sum of Square(SSR)*: $\sum_{i=1}^{n}(\hat{y_i} - \bar{y})^2$, where $\hat{y_i}$=value estimated by regression line. and $\bar{y}$=Mean value of the sample. $\boldsymbol{SSR \propto \frac{1}{fitting-of-model}}$
   (iii). *Residual Sum of Square(SSres)*: $SS_{res} = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$
   ,where y_i=Observed Value and $\hat{y_i}$=Estimated by regression line
   $\boldsymbol{SS_{res} \propto \frac{1}{Explanation-of-Data}}$
   $SST = SSR + SS_{res}$

9. Hypothesis of SLR:
   Null Hypothesis $H_0: \beta_0 = \beta_{00}$
   Alternative Hypothesis $H_1: \beta_0 \neq \beta_{00}$

Accuracy of the Model :

1. **Residual Standard Error (RSE) :** The RSE is considered a measure of the lack of fit of the model to the data. If the predictions obtained using the model are very close to the true outcome values—that is, if $\hat{y_i} \approx y_i$ for $i = 1, \ldots, n$ then RSE will be small, and we can conclude that the model fits the data very well. On the other hand, if $\hat{y_i}$ is very far from y_i for one or more observations, then the RSE may be quite large, indicating that the model doesn't fit the data well.

2. R^2 **Statistics :** The RSE provides an absolute measure of lack of fit of the model to the data. But since it is measured in the units of Y , it is not always clear what constitutes a good RSE. The R^2 statistic provides an alternative measure of fit. It takes the form of a *proportion* the proportion of variance explained and so it always takes on a value between 0 and 1, and is independent of the scale of Y .

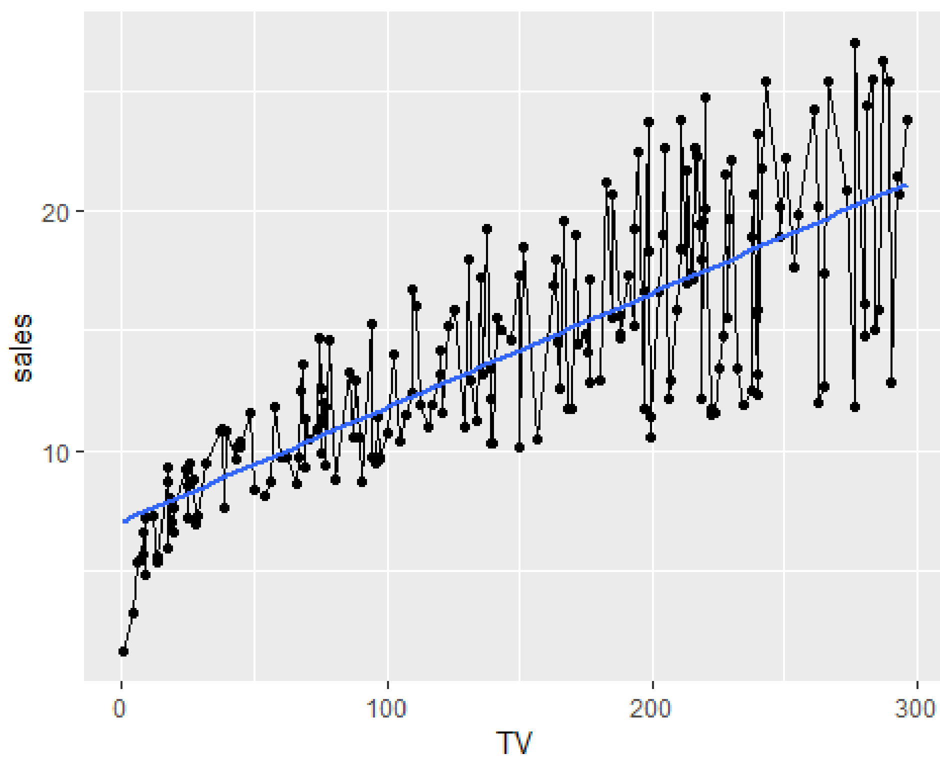
Linear Model with R Syntax -: $\boldsymbol{lm(formula, \quad data)}$

2.1.1 SLR with Advertising Data

- Here we fit the Simple Linear Models of Advertising Data . Our 1st model is between sales and TV $Sales = \beta_0 + \beta_1(TV) + \epsilon$

```
# library(ISLR)
# Watch the dataset in a particular package
# data(package = "ISLR")

# Load advertising dataset
```

```
library(readr)
library(ggplot2)
advertising <- read_csv("Advertising.csv")

## New names:
## * `` -> ...1

## Rows: 200 Columns: 5

## -- Column specification --------------------------------------------
## Delimiter: ","
## dbl (5): ...1, TV, radio, newspaper, sales

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

names(advertising)

## [1] "...1"      "TV"        "radio"     "newspaper" "sales"

ggplot(advertising , aes(TV , sales)) + geom_point() + geom_line() +
geom_smooth(method = "lm" , se = F)

## `geom_smooth()` using formula 'y ~ x'
```



```r
# linear model b/w sale ~ Tv
sale_tv <- lm(sales ~ TV , data = advertising)
sale_tv
```

```
##
## Call:
## lm(formula = sales ~ TV, data = advertising)
##
## Coefficients:
## (Intercept)           TV
##     7.03259      0.04754
```

```r
# Summary  of our Model
summary(sale_tv)
```

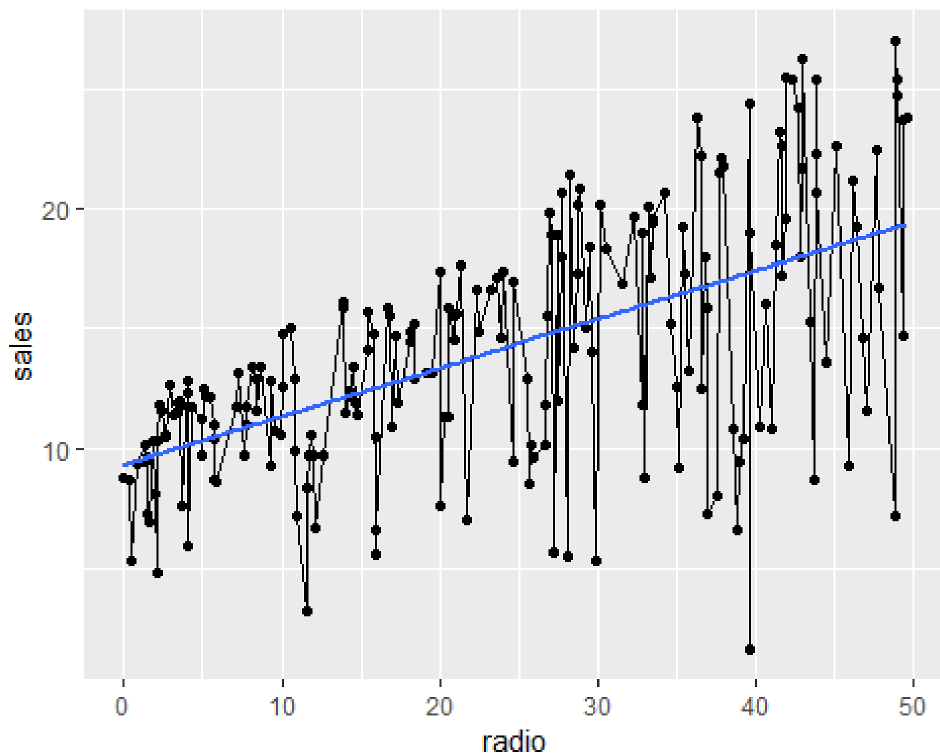
```
##
## Call:
## lm(formula = sales ~ TV, data = advertising)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

- **Fitted Model-:** $Sales = 7.03 + 0.047(TV)$ * The value of R^2 is 0.61 ,which tells that our fitted **61%** good . **OR 61%** of variability is explained in our model , so our model is quite good .

Our 2nd Model b/w Sales and Radio $y = \beta_0 + \beta_1(Radio) + \epsilon$

```r
ggplot(advertising , aes(radio , sales)) + geom_point() + geom_line() +
geom_smooth(method = "lm" , se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



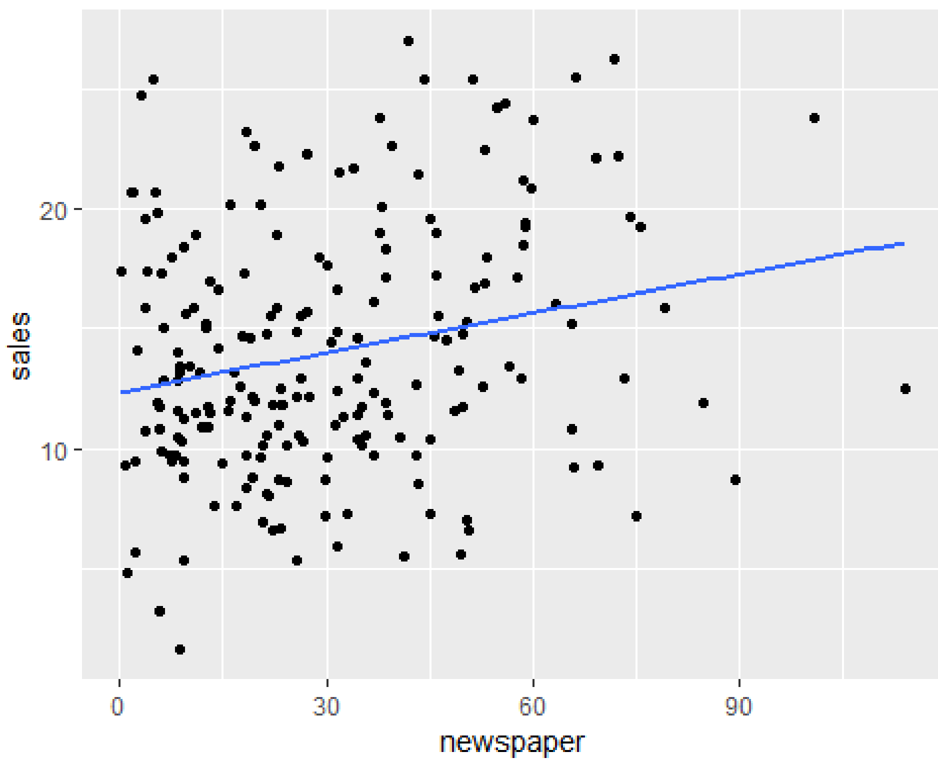

```
# Linear Model b/w Sales and Rado
sale_radio <- lm(sales ~ radio , advertising)
sale_radio

##
## Call:
## lm(formula = sales ~ radio, data = advertising)
##
## Coefficients:
## (Intercept)        radio
##      9.3116       0.2025

# Summary
summary(sale_radio)

##
## Call:
## lm(formula = sales ~ radio, data = advertising)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.7305  -2.1324   0.7707   2.7775   8.1810
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.31164    0.56290  16.542   <2e-16 ***
## radio        0.20250    0.02041   9.921   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.275 on 198 degrees of freedom
## Multiple R-squared:  0.332,  Adjusted R-squared:  0.3287
## F-statistic: 98.42 on 1 and 198 DF,  p-value: < 2.2e-16
```

- **Fitted Model-:** $Sales = 9.31 + 0.20(Radio)$ * The value of R^2 is 0.33 ,which tells that our fitted **33%** good that means our fill model is Bad . **OR 33%** of variability is explained in our model , so ou model is very bad .

Our 3rd Model b/w Sales and Newspaper $y = \beta_0 + \beta_1(Newspaper) + \epsilon$

```
ggplot(advertising , aes(newspaper , sales)) + geom_point() +
geom_smooth(method = "lm" , se = F)

## `geom_smooth()` using formula 'y ~ x'
```




```
# Linear Model b/w Sales and Radio
sale_news <- lm(sales ~ newspaper , advertising)
sale_news

##
## Call:
## lm(formula = sales ~ newspaper, data = advertising)
##
## Coefficients:
```

```
## (Intercept)    newspaper
##    12.35141      0.05469

# Summary
summary(sale_news)

##
## Call:
## lm(formula = sales ~ newspaper, data = advertising)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2272  -3.3873  -0.8392   3.5059  12.7751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.35141    0.62142   19.88  < 2e-16 ***
## newspaper    0.05469    0.01658    3.30  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.092 on 198 degrees of freedom
## Multiple R-squared:  0.05212,    Adjusted R-squared:  0.04733
## F-statistic: 10.89 on 1 and 198 DF,  p-value: 0.001148
```

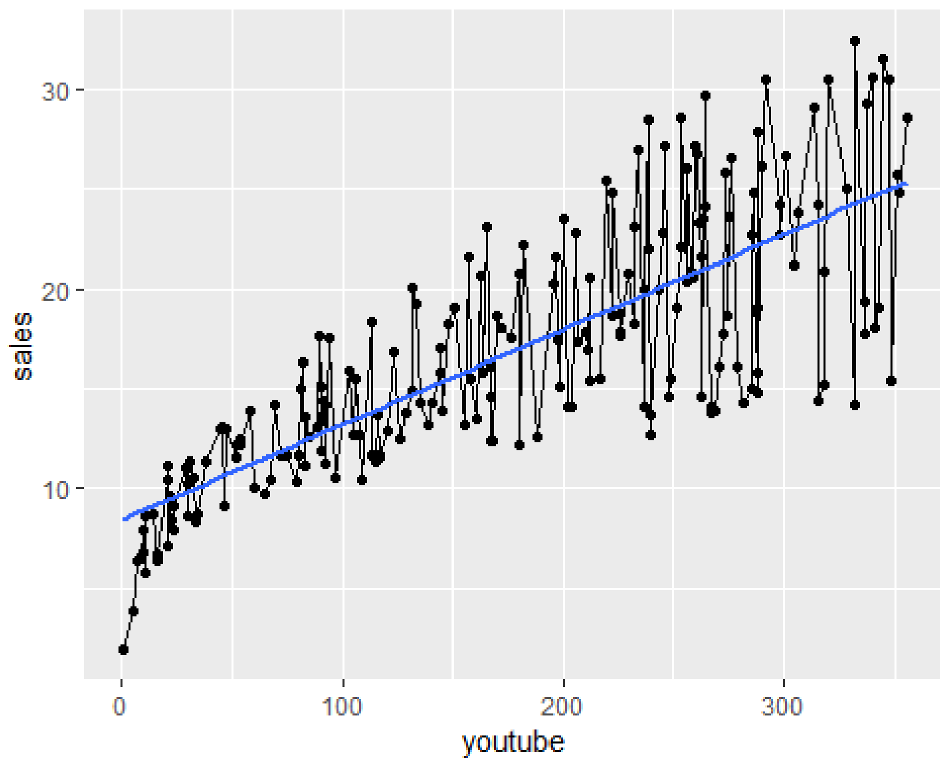
- **Fitted Model-:** $Sales = 12.35 + 0.054(Newspaper)$ * The value of R^2 is 0.052 ,which tells that our fitted **0.052%** good that means our fill model is very Bad . **OR 5%** of variability is explained in our model , so ou model is very bad .

2.1.2 SLR with Marketing Data

This **marketing** data from **datarium** package. In this data there are three advertising medias (youtube, facebook and newspaper) on sales. Data are the advertising budget in thousands of dollars along with the sales. The advertising experiment has been repeated 200 times.

1. **Model :** $sales = \beta_o + \beta_1(youtube) + \epsilon$

```
library(datarium)
data(marketing)
names(marketing)

## [1] "youtube"   "facebook"  "newspaper" "sales"

ggplot(marketing , aes(youtube , sales)) + geom_point() + geom_line() +
geom_smooth(method = "lm" , se = F)

## `geom_smooth()` using formula 'y ~ x'
```

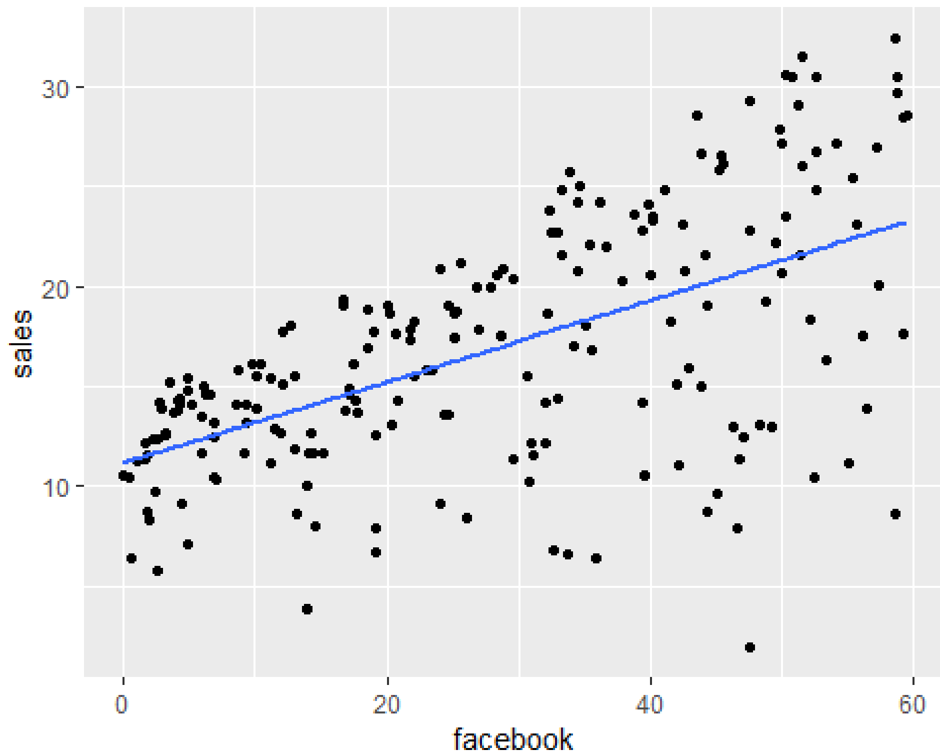



```
sale_yt <- lm(sales ~ youtube , data = marketing)

summary(sale_yt)

##
## Call:
## lm(formula = sales ~ youtube, data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0632  -2.3454  -0.2295   2.4805   8.6548
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.439112   0.549412   15.36   <2e-16 ***
## youtube     0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.91 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

- **Fitted Model:*** $sales = 8.84 + 0.048(YouTube)$ * **Interpretation :** One advertising on *YouTube* increase the *Sale* by 0.048 or *4.8%* . * **Model Accuracy :** The value of $R^2 = 0.61$, it's mean that our model is *61%* good . Our model is just good.

2. **MOdel :** $Sales = \beta_0 + \beta_1(Facebook) + \epsilon$

```
ggplot(marketing , aes(facebook , sales)) + geom_point() + geom_smooth(method
= "lm" , se = F)

## `geom_smooth()` using formula 'y ~ x'
```



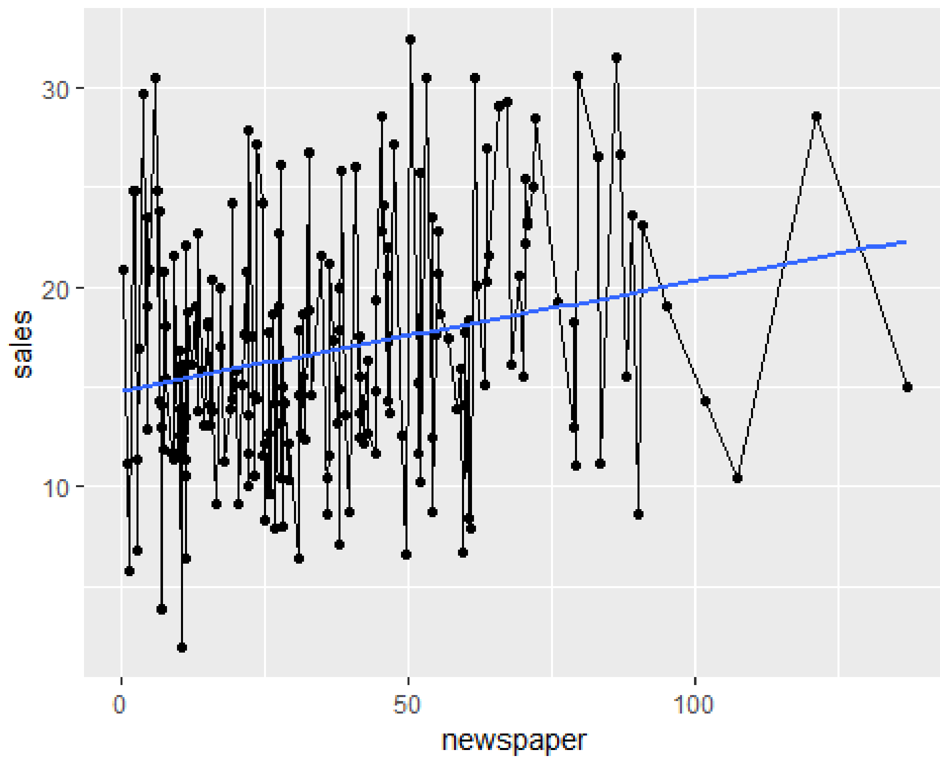

```
sale_face <- lm(sales ~ facebook , data = marketing)

summary(sale_face)

##
## Call:
## lm(formula = sales ~ facebook, data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.8766  -2.5589   0.9248   3.3330   9.8173
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.17397    0.67548  16.542   <2e-16 ***
## facebook     0.20250    0.02041   9.921   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.13 on 198 degrees of freedom
```

```
## Multiple R-squared:  0.332,  Adjusted R-squared:  0.3287
## F-statistic: 98.42 on 1 and 198 DF,  p-value: < 2.2e-16
```

- **Fitted Model:*** $sales = 11.17 + 0.202(Facebook)$
  Interpretation : One advertising on $Facebook$ increase the $Sale$ by 0.2 or *20%* . *
  Model Accuracy : The value of $R^2 = 0.33$, it's mean that our model is *33%* good .
  Our model is not good .

3. **Model:** $Sales = \beta_0 + \beta_1(Newspaper) + \epsilon$

```
ggplot(marketing , aes(newspaper , sales)) + geom_point() + geom_line() +
geom_smooth(method = "lm" , se = F)

## `geom_smooth()` using formula 'y ~ x'
```




```
sale_news <- lm(sales ~ newspaper , marketing)

summary(sale_news)

##
## Call:
## lm(formula = sales ~ newspaper, data = marketing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.473  -4.065  -1.007   4.207  15.330
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

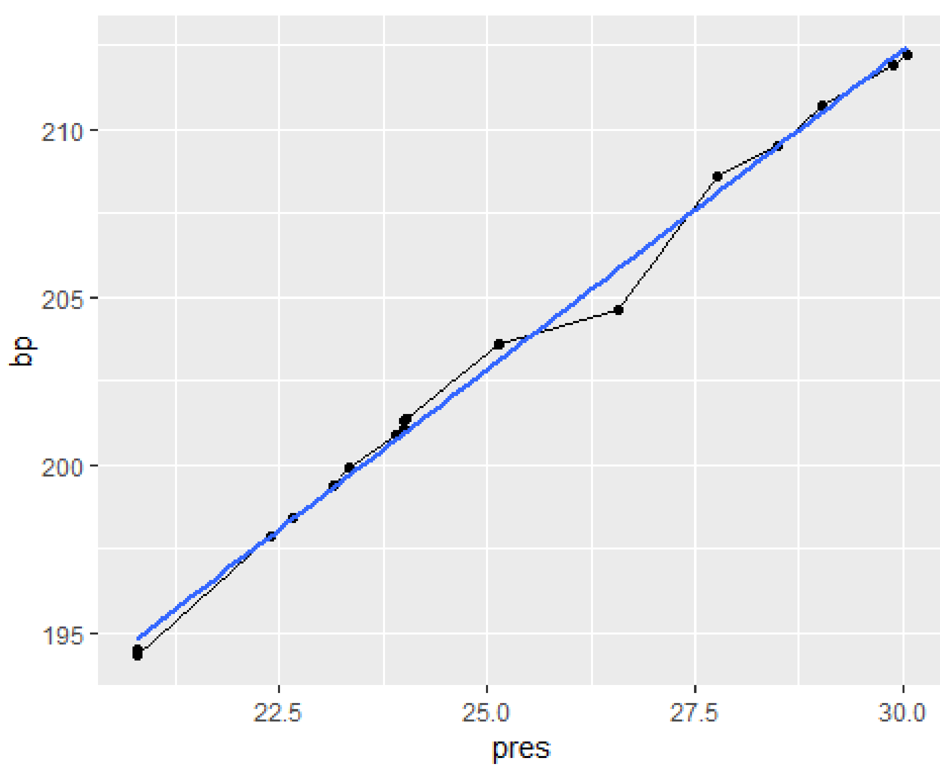
```
## (Intercept) 14.82169    0.74570   19.88  < 2e-16 ***
## newspaper    0.05469    0.01658    3.30  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.111 on 198 degrees of freedom
## Multiple R-squared:  0.05212,    Adjusted R-squared:  0.04733
## F-statistic: 10.89 on 1 and 198 DF,  p-value: 0.001148
```

- **Fitted Model:*** $sales = 14.82 + 0{,}055(Newspaper)$ * **Interpretation :** One advertising on *Newspaper* increase the *Sale* by 0.05 or *5%* . * **Model Accuracy :** The value of $R^2 = 0.052$, it's mean that our model is *5%* good . Our model is very bad .

2.1.3 SLR with Forbes Data

Model : $bp = \beta_0 + \beta_1(pres) + \epsilon$

```
library(MASS)
data(forbes)
# dim(forbes)
# names(forbes)

library(ggplot2)
ggplot(forbes , aes(pres , bp)) + geom_point() + geom_line() +
geom_smooth(method = "lm" , se = F)

## `geom_smooth()` using formula 'y ~ x'
```



```r
lm_forbes <- lm(bp ~ pres , data = forbes)
summary(lm_forbes)

##
## Call:
## lm(formula = bp ~ pres, data = forbes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22687 -0.22178  0.07723  0.19687  0.51001
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 155.29648    0.92734  167.47   <2e-16 ***
## pres          1.90178    0.03676   51.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.444 on 15 degrees of freedom
## Multiple R-squared:  0.9944, Adjusted R-squared:  0.9941
## F-statistic:  2677 on 1 and 15 DF,  p-value: < 2.2e-16
```

- **Fitted Model :** $bp = 155 + 1.9(pres)$
  Interpretation : bp increase by 1.9 , if the press increase by one unit .
  Model Accuracy : The value of $R^2 = 0.99$, it's mean that our model is *99%* good .

2.1.4 SLR with Trees Data

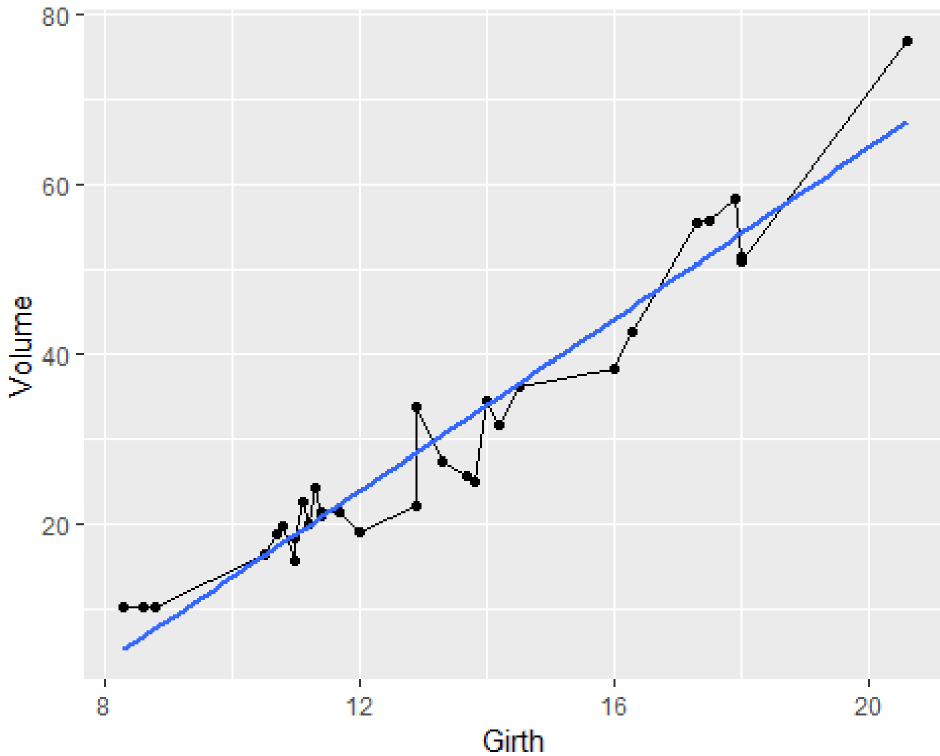
Model: $Volume = \beta_0 + \beta_1(Girth) + \epsilon$

1. **Correlation b/w Girth and Volume of Trees Data**

```r
cor(trees$Girth , trees$Volume)

## [1] 0.9671194
```

The value of correlation is 0.967 , which is very close to 1. So , we can say that there is a positive correlation b/w Girth and Volume . We can also see this **Graphically**

```r
ggplot(trees , aes(Girth , Volume)) + geom_point() + geom_line() +
geom_smooth(method = "lm" , se = F)

## `geom_smooth()` using formula 'y ~ x'
```



After seeing the *Graph* , we can easily say that there is linear relationship b/w *Girth* and *Volume* because most points are linear and also we a approximately straight . To prove that we fit the linear model .

```
lm_girth <- lm(Volume ~ Girth , data = trees)
summary(lm_girth)

##
## Call:
## lm(formula = Volume ~ Girth, data = trees)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## Girth         5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

- **Complete Interpretation :**

- **Fitted Model :** $Volume = -36.94 + 5.06(Girth)$ It means a change of one unit in *Girth* will bring **5.06** units to change in *Volume* . **OR** If the *Girth* increase by one unit then the *Volume* will increase by **5.06** units .

- Ths **Std.Error** is variability to expect in coefficient which captures sampling variability so the variation in *intercept* can be 3.36 and variation in *Girth* will be 0.24 not more than that .

- **T value :** t-value is coefficient divided by standard error it is basically how big is estimated relative to error bigger the coefficient relative to *Std.Error* the bigger that t score and t score comes with a p-value because its a distribution p-value is how significantly significant the variable is to the model for a *Confidence Interval* of 95 we will compare this value with α which will be 0.05 , so in our case *p-value* of both intercept and Girth is less than α $(\alpha = 0.05)$ this implies that both are statistically significant to iur model .

- **Residual Standard Error** or the std. error of the model is basically the average error for the model which is 4.252 in our case and it means that our model can be off by on an average of **4.252** , while predicting the Volume lesser the error the better the model while predicting .

- **Model Accuracy :** The value of $R^2 = 0.94$, it's mean that our model is *94%* good .

- **F - statistics** is the ratio of the mesn sum square of the model and mean sum square of the error. In othe word it's the ratio of how
well the model is doing and what the errror is doing and the higher the F-value is the better the model is doing on compared to error . *1* is the df of numerator and *29* is the df of the F-statistics .

Predict the value Volume at Girth 10 .

```
p =as.data.frame(10)
colnames(p) = "Girth"

predict(lm_girth  , newdata = p)

##        1
## 13.71511
```

So, the predicted value of the *Volume* is 13.71 at *Girth* 10 .

2.1.5 SLR by Own Function

Here we fit Simple Linear Regression model **Model:** $Volume = \beta_0 + \beta_1(Height) + \epsilon$ by our own Function .

```
slr <- function(x , y){
# A function which returns simple Regression Analysis
X <- cbind(1 , x)  # Model Matrix
```

```r
p1 <- ncol(X)
n <- nrow(X)

# (X^T X)
xtx <- crossprod(X)

# (X^T y)
xty <- crossprod(X , y)

# beta= (X^T y)^-1 (X^T y)
beta <- solve(xtx , xty)

# Resid = y-X*beta
resid <- y - X %*% beta

# Residual Sum Square
rss <- sum(resid^2)

# Mean Sum Square
msresid <- rss / (n-p1)

# Std. Error
sebeta <- sqrt(diag(msresid*solve(xtx)))

# t - value
tratio <- beta / sebeta

# p - value
pvalue <- 2*(1 - pt(abs(tratio) , df = (n - p1)))

# Output
out <- data.frame(Reg_Coeff = beta, SE_Beta = sebeta , T_value = tratio ,
P_value = pvalue)

# Return the output that we find
return(out)
}

# dump and source
dump("slr" , file = "slr.txt")
source("slr.txt")

y <- trees$Volume
x <- trees$Height

# Fit the Model
model_slr <- slr(x = x , y = y)
```

```r
round(model_slr , 3)

##   Reg_Coeff SE_Beta T_value P_value
##     -87.124  29.273  -2.976   0.006
## x     1.543   0.384   4.021   0.000
```

Model: $Volume = -87.12 + 1.5(Height)$

To verify the above result , we fit the model by using `lm()` function.

```r
m = lm(y~x , data = trees) ; summary(m)

## 
## Call:
## lm(formula = y ~ x, data = trees)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -21.274  -9.894  -2.894  12.068  29.852 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -87.1236    29.2731  -2.976 0.005835 ** 
## x             1.5433     0.3839   4.021 0.000378 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 13.4 on 29 degrees of freedom
## Multiple R-squared:  0.3579, Adjusted R-squared:  0.3358 
## F-statistic: 16.16 on 1 and 29 DF,  p-value: 0.0003784
```

It's $\beta's$ ae same as above . **Interpretation:** The value of $R^2 = 0.36$, It's means only *36%* of variability are explain in this model .

2.1.5.1 Extraction from the fitted SLR

Model: $Volume = \beta_0 + \beta_1(Girth) + \epsilon$
Here we extract **names , names(summary())** , **coefficients** , R^2 , **coef , residuals , sum of residuals , deviance , model mtx , MS_residuals , sigma**

```r
m1 <- lm(Volume ~ Girth , trees)
# print m1
print(m1)

## 
## Call:
## lm(formula = Volume ~ Girth, data = trees)
## 
## Coefficients:
```

```
## (Intercept)       Girth
##     -36.943       5.066

# Summary m1
summary(m1)

##
## Call:
## lm(formula = Volume ~ Girth, data = trees)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## Girth         5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16

# Names of m1
names(m1)

##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"

# Names of summary of m1
names(summary(m1))

##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"

# Coef. of m1
m1$coefficients

## (Intercept)      Girth
##  -36.943459    5.065856

# R_square
summary(m1)$r.squared

## [1] 0.9353199

# beta's
coef(m1)
```

```
## (Intercept)       Girth
##  -36.943459    5.065856
```

```r
summary(m1)$coef
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -36.943459   3.365145 -10.97827 7.621449e-12
## Girth         5.065856   0.247377  20.47829 8.644334e-19
```

```r
# Residuals
residuals(m1)
```

```
##          1          2          3          4          5          6
##  5.1968508  3.6770939  2.5639226  0.1519667  1.5387954  1.9322098
##          7          8          9         10         11         12
## -3.1809615 -0.5809615  3.3124528  0.1058672  3.8992815  0.1926959
##         13         14         15         16         17         18
##  0.5926959 -1.0270610 -4.7468179 -6.2060887  5.3939113 -3.0324313
##         19         20         21         22         23         24
## -6.7587739 -8.0653595  0.5214692 -3.2917021 -0.2114590 -5.8102436
##         25         26         27         28         29         30
## -3.0300006  4.7041430  3.9909717  4.5646292 -2.7419565 -3.2419565
##         31
##  9.5868168
```

```r
# Residuals sum of Square
sum(residuals(m1)^2)
```

```
## [1] 524.3025
```

```r
# Deviance od m1
deviance(m1)
```

```
## [1] 524.3025
```

```r
# Model Matrix X
X <- model.matrix(m1)
X[c(1:3,25)]
```

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

```r
# MS_residuals
d <- deviance(m1) / df.residual(m1) ; d
```

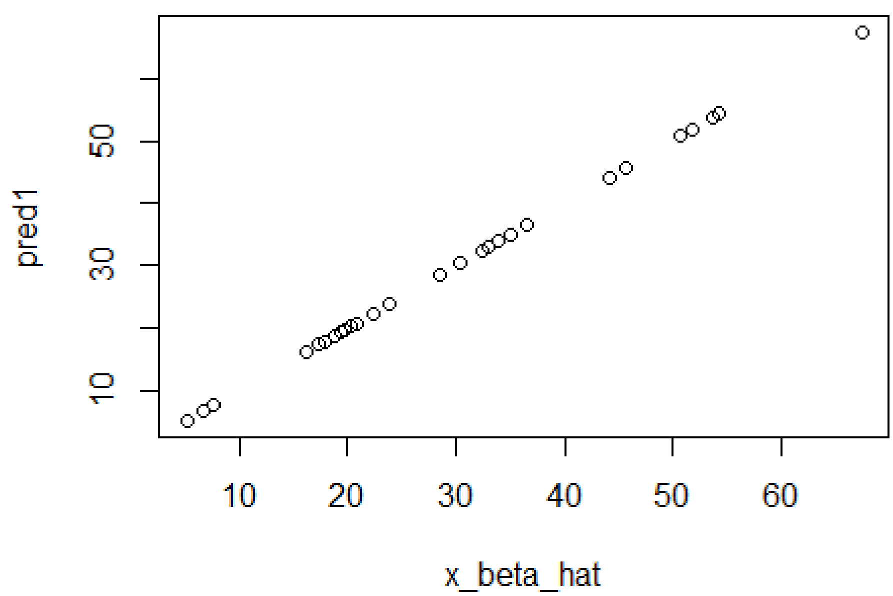
```
## [1] 18.0794
```

```r
# Sqrt of MS_residuals
sqrt(d)
```

```
## [1] 4.251988
```

```r
# sigma
summary(m1)$sigma  # Same as Above
```

```
## [1] 4.251988

# Fitted Value =s of x*beta^T
x_beta_hat <- fitted(m1)  ; head(x_beta_hat)

##         1         2         3         4         5         6
##  5.103149  6.622906  7.636077 16.248033 17.261205 17.767790

# Predicted Values
pred1 <- predict(m1)  ; head(pred1)

##         1         2         3         4         5         6
##  5.103149  6.622906  7.636077 16.248033 17.261205 17.767790

# Plot the graph b/w fitted(m1) & predict(m1)
plot(x_beta_hat , pred1)
```




```
# Variance - Covaiance of beta
vcov(m1)

##             (Intercept)      Girth
## (Intercept)  11.3242005 -0.81073976
## Girth        -0.8107398  0.06119536
```

2.1.6 Centered Form of SLR

The form of the model

$$y_i = \beta_0 + \beta_1(x_i - \bar{x}) + \epsilon_i \quad ; i = 1,2,\cdots,n$$

is called centered form of simple linear regression model. Note that in this form $\widehat{\beta_1}$ remains same, but $\widehat{\beta_0 = \bar{y}}$ in this form. Moreover, x_i is replaced by $(x_i - \bar{x})$ in the centered form. Thus to implement `slr()` function, define `x = x - mean(x)`, and call it into slr as argument x. Similar changes are also required in `lm()` to implement it. We shall make use of the `transform()` function for this data manipulation. Following set of commands will make the things more clear:

```
# Use the transform() to transform the variable
d1 <- trees
d1 <- transform(d1 , Girth.c = Girth - mean(Girth))
head(d1)

##   Girth Height Volume   Girth.c
## 1   8.3     70   10.3 -4.948387
## 2   8.6     65   10.3 -4.648387
## 3   8.8     63   10.2 -4.448387
## 4  10.5     72   16.4 -2.748387
## 5  10.7     81   18.8 -2.548387
## 6  10.8     83   19.7 -2.448387

# Fit the centered form
m1c <- lm(Volume ~ Girth.c , data = d1)

summary(m1c)

##
## Call:
## lm(formula = Volume ~ Girth.c, data = d1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.1710     0.7637   39.51   <2e-16 ***
## Girth.c       5.0659     0.2474   20.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16

m11 <- lm(Volume ~ Girth , d1)

# beta's of both models
print(m1c)
```

```
##
## Call:
## lm(formula = Volume ~ Girth.c, data = d1)
##
## Coefficients:
## (Intercept)      Girth.c
##      30.171        5.066
```

```
print(m11)
```

```
##
## Call:
## lm(formula = Volume ~ Girth, data = d1)
##
## Coefficients:
## (Intercept)        Girth
##     -36.943        5.066
```

```
# variance - covariance of beta's
vcov(m1c)
```

```
##              (Intercept)      Girth.c
## (Intercept) 5.832064e-01 1.952396e-17
## Girth.c     1.952396e-17 6.119536e-02
```

```
vcov(m11)
```

```
##             (Intercept)       Girth
## (Intercept)  11.3242005 -0.81073976
## Girth        -0.8107398  0.06119536
```

```
# Correlation Mtx.
cov2cor(vcov(m1c))
```

```
##              (Intercept)      Girth.c
## (Intercept) 1.000000e+00 1.033469e-16
## Girth.c     1.033469e-16 1.000000e+00
```

```
cov2cor(vcov(m11))
```

```
##             (Intercept)      Girth
## (Intercept)   1.0000000 -0.9739092
## Girth        -0.9739092  1.0000000
```

Note -: Note that estimates are highly correlated in non-centered form, whereas they are not in centered form. Moreover, $\widehat{\beta_0}$ is simply $\bar{x}$, which is mean of the response vector y. For these reasons, centered form is preferred over non-centered form of the model. These ideas can be extended to multiple regression model also .

2.1.7 Centered Form of SLR by Own Function

```
slrc=function(y,x1) {
X=cbind(1,x1) # model matrix
```

```
n=nrow(X)
p1=ncol(X)
XtX=crossprod(X,X)
Xty=crossprod(X,y)
beta= solve(XtX,Xty)
resid=y-X %*% beta
rss= sum(resid^2)
msresid=rss/(n-p1)
sebeta=sqrt(diag(msresid*solve(XtX)))
tratio=beta/sebeta
pvalue=2*(1-pt(abs(tratio),df=n-p1))
out=data.frame(Reg_Coeff=beta, SE_beta=sebeta, T_value=tratio,
P_value=pvalue)
out=round(out,3) # Round upto 3 digits
return(out)
}
dump("slrc",file="slrc.txt")

## Analyse the data 'trees' using 'volume' as response and
# 'Girth' and 'Height' as regressors.
d1=trees

y=d1$Volume
x1=d1$Girth-mean(d1$Girth)
srmc=slrc(y,x1)
print(srmc)

##    Reg_Coeff SE_beta T_value P_value
##       30.171   0.764  39.507       0
## x1     5.066   0.247  20.478       0
```

2.2 Multiple Linear Regression / MLR

(Multiple Linear Regression Analysis) The basic difference between simple and multiple regression is that in simple there is only one predictor x, whereas in multiple regression it must be 2 or *more*. We shall write a function to implement multiple regression analysis with 2 regressors or covariates.

1. **Model:** The Multiple Linear Regression Model is denoted as: $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots \beta_i x_{ip} + \epsilon$ where, y is the response vaiable , $\beta_1 + \beta_2 + \cdots + \beta_i$ is regression coefficient and $x_1 + x_2 + \cdots + x_{ip}$ are predictors.

2. **Regressiom Coefficient:** Change in response y per unit change in regressor x.

3. **Formulas for Calculation** $(y, X, \beta, \sigma^2, I)$ It is to be noted that y is the vector of responses, X is termed as model matrix and β iis known as vector of regression coefficients.However, σ^2 is known as residual variance, I stands for indentity matrix of order $n \times n$. The method of least square is used to estimate β . This method states that we will close that value of β which will minimize error sum of squares

defined as : $errorSS = e^T e = (y - X\beta)^T(y - X\beta)$ and the result is solution normal equations defined as: $(X^T X)\hat{\beta} = X^T y$ alternatively least square estimate of β is defined as: $\hat{\beta} = (X^T X)^{-1}(X^T y)$ This implies that variance covariance matrix of $\hat{\beta}$ is : $Var(\hat{\beta}) = \sigma^2(X^T X)^{-1}$ and its estimate is $\widehat{Var(\hat{\beta})} = \widehat{\sigma^2}(X^T X)^{-1}$ The diagonal elements of this matrix are variances and non-diagonal elements are co-variances, Thus standard error of β is $SE(\hat{\beta}) = \sqrt{diag\left(\widehat{Var(\hat{\beta})}\right)}$ where $\widehat{\sigma^2} = \frac{ResidSS}{n-(p+1)} = MSresidual$ where, $ResidSS = \left(y - X\hat{\beta}\right)^T\left(y - X\hat{\beta}\right)$

4. **Sum of Squares -**
   Total Sum of Square: $SST = Y^T Y - n\bar{Y}^2$ with degree of freedom $n - 1$
   Regression Sum of Square: $SS_{res} = \hat{\beta}^T X^T Y - n\bar{X}^2$ with degree of freedom k
   Residual Sum of Square: $SSR = Y^T Y \hat{\beta}^T X^T Y$ with degree of freedom n-k-1

5. **Hypothesis of SLR:** Null Hypothesis $H_0: \beta_1 = \beta_2 = \cdots = \beta_i = \cdots = \beta_k = 0$
   Alternative Hypothesis H_1: At least one $\beta_i's \neq 0 \quad ; i = 1,2,\ldots,k$

Steps for Best Fitting of MLR :

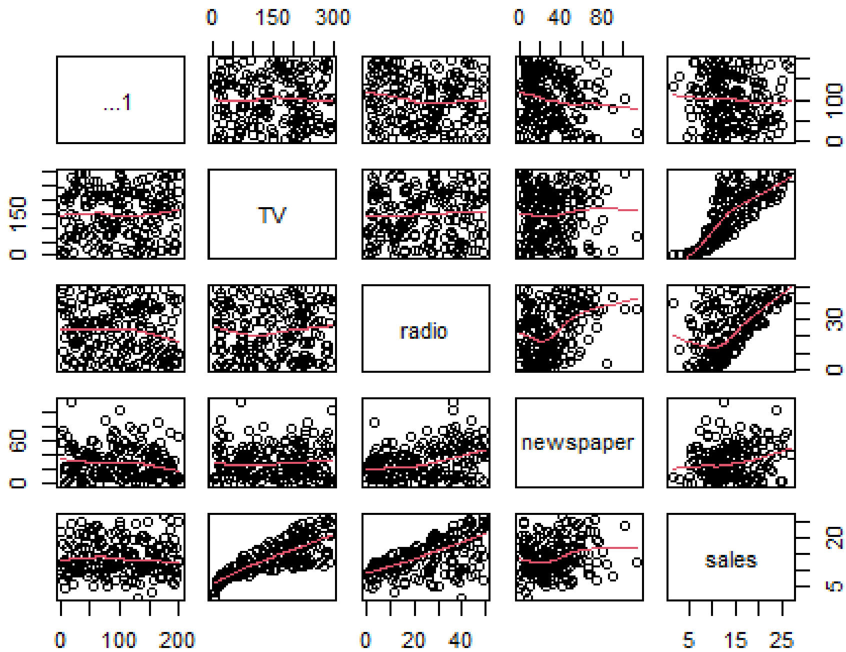
1. **Relationship between the Response and Predictors**
2. **Decide the Important Variables**
3. **Fitting Model**
4. **Predictions**

2.2.1 MLR with Advertising Data

- Here we fit the Multiple Linear Models of *Advertising* Data.
  Model : $Sales = \beta_0 + \beta_1(TV) + \beta_2(Radio) + \beta_3(Newspaper) + \epsilon$

```
# Load advertising dataset
library(readr)
library(ggplot2)
advertising <- read_csv("Advertising.csv")

## New names:
## * `` -> ...1

## Rows: 200 Columns: 5

## -- Column specification --------------------------------------------------
## Delimiter: ","
## dbl (5): ...1, TV, radio, newspaper, sales

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```r
# we see the relation Graphically
pairs(advertising , panel = panel.smooth)
```




```r
# We see correlation mtx
cor(advertising)

##                   ...1         TV      radio   newspaper       sales
## ...1        1.00000000 0.01771469 -0.11068044 -0.15494414 -0.05161625
## TV          0.01771469 1.00000000  0.05480866  0.05664787  0.78222442
## radio      -0.11068044 0.05480866  1.00000000  0.35410375  0.57622257
## newspaper  -0.15494414 0.05664787  0.35410375  1.00000000  0.22829903
## sales      -0.05161625 0.78222442  0.57622257  0.22829903  1.00000000

# Now we want to prove the above  results
ad_mlr <- lm(sales ~ TV + radio + newspaper , data = advertising)

ad_mlr

##
## Call:
## lm(formula = sales ~ TV + radio + newspaper, data = advertising)
##
## Coefficients:
## (Intercept)           TV        radio    newspaper
##    2.938889     0.045765     0.188530    -0.001037

summary(ad_mlr)
```

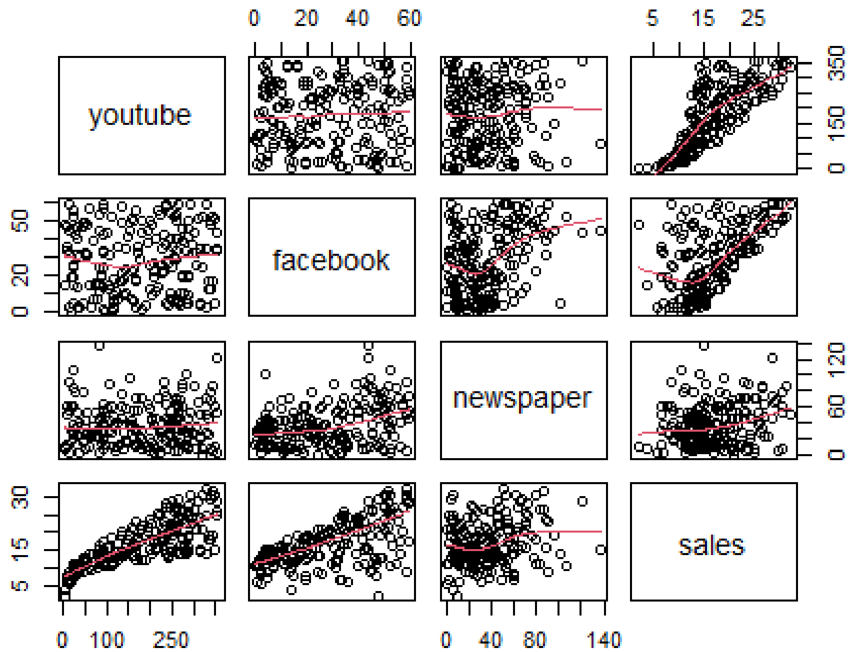
```
##
## Call:
## lm(formula = sales ~ TV + radio + newspaper, data = advertising)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

- **Fitted Model:** $Sales = 2.94 + 0.046(TV) + 0.188(Radio) - 0.001(Newspaper)$
- The value of R^2 is 0.9 , which tells that **90%** of variability explain in our model . **OR** Our Model **90%** Good .

2.2.2 MLR with Marketing Data

- Here we fit the Multiple Linear Models of *Marketing* Data .
  Model : $Sales = \beta_0 + \beta_1(TV) + \beta_2(Radio) + \beta_3(Newspaper) + \epsilon$

```
# Load advertising dataset
library(datarium)
data("marketing")
# we see the relation Graphically
pairs(marketing , panel = panel.smooth)
```




```
# We see correlation mtx
cor(marketing)

##                youtube   facebook  newspaper     sales
## youtube   1.00000000 0.05480866 0.05664787 0.7822244
## facebook  0.05480866 1.00000000 0.35410375 0.5762226
## newspaper 0.05664787 0.35410375 1.00000000 0.2282990
## sales     0.78222442 0.57622257 0.22829903 1.0000000

# Now we want to prove the above  results
mar_mlr <- lm(sales ~ youtube + facebook + newspaper , data = marketing)

mar_mlr

##
## Call:
## lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)
##
## Coefficients:
## (Intercept)      youtube     facebook    newspaper
##    3.526667     0.045765     0.188530    -0.001037

summary(mar_mlr)

##
## Call:
## lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -10.5932  -1.0690   0.2902   1.4272   3.3951 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  3.526667   0.374290   9.422   <2e-16 ***
## youtube      0.045765   0.001395  32.809   <2e-16 ***
## facebook     0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.023 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956 
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

- **Fitted Model:** $Sales = 3.52 + 0.045(YouTube) + 0.188(Facebook) - 0.001(Newspaper)$
- The value of R^2 is 0.89 , which tells that **89%** of variability explain in our model . **OR** Our Model **89%** Good .

2.2.3 MLR with Trees Data Data

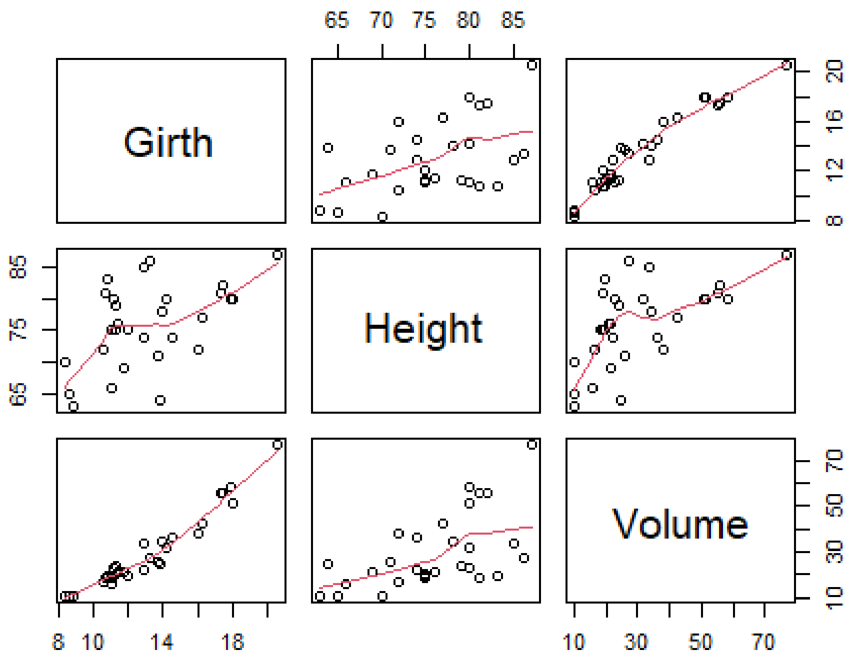
*Model:** $Volume = \beta_0 + \beta_1(Girth) + \beta_2(Height)\epsilon$

1. **Correlation Matrix of Trees Data**

```
cor(trees)

##            Girth    Height    Volume
## Girth  1.0000000 0.5192801 0.9671194
## Height 0.5192801 1.0000000 0.5982497
## Volume 0.9671194 0.5982497 1.0000000
```

We can also see this **Graphically**

```
pairs(trees , panel = panel.smooth)
```



To prove that we fit the linear model .

```
lmr_girth <- lm(Volume ~ Girth + Height , data = trees)
summary(lmr_girth)

##
## Call:
## lm(formula = Volume ~ Girth + Height, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4065 -2.6493 -0.2876  2.2003  8.4847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57.9877     8.6382  -6.713 2.75e-07 ***
## Girth         4.7082     0.2643  17.816  < 2e-16 ***
## Height        0.3393     0.1302   2.607   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.882 on 28 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9442
## F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
```

- **Complete Interpretation :**

- **Fitted Model :** $Volume = -57.99 + 4.70(Girth) + 0.33(Heigt)$ It means for a change of one unit in *Girth* will bring **4.70** units to change in *Volume* and one unit change in *Height* will bring **0.33** units to change in *Volume*. **OR** If the *Girth* increase by one unit then the *Volume* will increase by **4.70** units *&* If the *Height* increase by one unit then the *Volume* will increase by **0.33** units.

- The **Std.Error** is variability to expect in coefficient which captures sampling variability so the variation in *intercept* can be up 8.64 and variation in *Girth* will be 0.26 and and variation in *Height* will be 0.13 not more than that .

- **T value :** t-value is coefficient divided by standard error it is basically how big is estimated relative to error bigger the coefficient relative to *Std.Error* the bigger that t score and t score comes with a p-value because its a distribution p-value is how significantly significant the variable is to the model for a *Confidence Interval* of 95 we will compare this value with α which will be 0.05 , so in our case *p-value* of both *intercept* , *Girth* is less than α $(\alpha = 0.05)$ this implies that both are statistically significant to iur model **&** *Height* is greater than α $(\alpha = 0.05)$ this implies that *height* is not statistically significant to our model.

- **Residual Standard Error** or the std. error of the model is basically the average error for the model which is 3.88 in our case and it means that our model can be off by on an average of **3.88** , while predicting the Volume lesser the error the better the model while predicting .

- **Model Accuracy :** The value of $R^2 = 0.94$, it's mean that our model is *94%* good . **OR** There is **94%** variability is explain in our Model.

- **F - statistics** is the ratio of the mean sum square of the model and mean sum square of the error. In othe word it's the ratio of how well the model is doing and what the errror is doing and the higher the F-value is the better the model is doing on compared to error . *2* is the df of numerator and *23* is the df of the F-statistics . The value of *F-statistic* is **255* and the corresponding *p-value* is $2.2e^{-16}$.

2.2.4 MLR by Own Function

Here we will fit the *MLR Model* by defing our own fumction. **Model:** $Volume = \beta_0 + \beta_1(Girth) + \beta_2(Height) + \epsilon$

```
mlr <- function(y , x1 , x2){
# define a function to implement multiple linear regression
# y is the response variable
# x1 is one regressor
# x2 is another regressor
# this function returns regresion analysis
X<-cbind(1,x1,x2)
p1<-ncol(X)
n<-nrow(X)

# (x^t x)
```

```
xtx<-crossprod(X)

# (X^T y)
xty<-crossprod(X,y)

# beta= (X^T y)^-1 (X^T y)
beta<-solve(xtx,xty)

# Resid = y-X*beta
resid<-y-X %*% beta

# Residual Sum Square
rss<-sum(resid^2)

# Mean Sum Square
msresid<-rss/(n-p1)

# Std. Error
sebeta<-sqrt(diag(msresid*solve(xtx)))

# T- Value
tratio<-beta/sebeta

# P - value
pvalue<-2*(1-pt(abs(tratio),df=n-p1))

# Output
out<-data.frame(Reg_Coef = beta , SE_beta = sebeta , tvalue = tratio ,
P_value = pvalue)

#round output up to 3 digits
out<-round(out,3)
return(out)
}

dump("mlr",file="mlr.txt")

## Analyse the data `trees` using `Volume` as response and #`Girth` and
`Height` as regressors.
data(trees)
y<-trees$Volume
x1<-trees$Girth
x2<-trees$Height

M2<-mlr(y,x1,x2)
print(M2)

##    Reg_Coef SE_beta tvalue P_value
##     -57.988   8.638 -6.713   0.000
```

```
## x1    4.708   0.264 17.816   0.000
## x2    0.339   0.130  2.607   0.014
```

- **Fitted Model :** $Volume = -57.99 + 4.70(Girth) + 0.33(Heigt)$

Compare the above results by `lm()` function.

```
mlr_tree <- lm(Volume ~ Girth + Height , trees)
mlr_tree

##
## Call:
## lm(formula = Volume ~ Girth + Height, data = trees)
##
## Coefficients:
## (Intercept)        Girth       Height
##    -57.9877       4.7082       0.3393
```

The result we get by **slr()** and **lm()** functions are approximately same

2.2.4.1 Extraction from the fitted MSLR

```
m2 <- lm(Volume ~ Girth + Height , trees)
print(m2)

##
## Call:
## lm(formula = Volume ~ Girth + Height, data = trees)
##
## Coefficients:
## (Intercept)        Girth       Height
##    -57.9877       4.7082       0.3393

summary(m2)

##
## Call:
## lm(formula = Volume ~ Girth + Height, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4065 -2.6493 -0.2876  2.2003  8.4847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57.9877     8.6382  -6.713 2.75e-07 ***
## Girth         4.7082     0.2643  17.816  < 2e-16 ***
## Height        0.3393     0.1302   2.607   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.882 on 28 degrees of freedom
```

```
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9442
## F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
```

```r
# Names of m2
names(m2)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```r
# Names of summary of m2
names(summary(m2))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
# Coef. of m2
m2$coefficients
```

```
## (Intercept)       Girth      Height
## -57.9876589   4.7081605   0.3392512
```

```r
# R_square
summary(m2)$r.squared
```

```
## [1] 0.94795
```

```r
# beta's
coef(m2)
```

```
## (Intercept)       Girth      Height
## -57.9876589   4.7081605   0.3392512
```

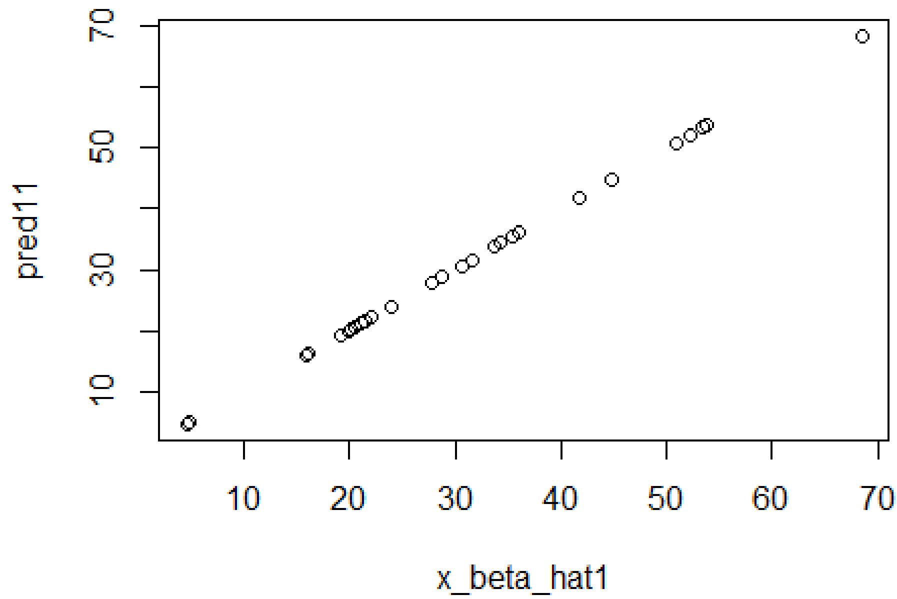
```r
summary(m2)$coef
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -57.9876589  8.6382259 -6.712913 2.749507e-07
## Girth         4.7081605  0.2642646 17.816084 8.223304e-17
## Height        0.3392512  0.1301512  2.606594 1.449097e-02
```

```r
# Residuals
residuals(m2)
```

```
##           1           2           3           4           5
##  5.46234035  5.74614837  5.38301873  0.52588477 -1.06900844
##           6           7           8           9          10
## -1.31832696 -0.59268807 -1.04594918  1.18697860 -0.28758128
##          11          12          13          14          15
##  2.18459773 -0.46846462 -0.06846462  0.79384587 -4.85410969
##          16          17          18          19          20
## -5.65220290  2.21603352 -6.40648192 -4.90097760 -3.79703501
##          21          22          23          24          25
```

```
##  0.11181561 -4.30831896  0.91474029 -3.46899800 -2.27770232
##          26          27          28          29          30
##  4.45713224  3.47624891  4.87148717 -2.39932888 -2.89932888
##          31
##  8.48469518

# Residuals sum of Square
sum(residuals(m2)^2)

## [1] 421.9214

# Deviance of m2
deviance(m2)

## [1] 421.9214

# Model Matrix X
X <- model.matrix(m2)
X[c(1:3,25)]

## [1] 1 1 1 1

# MS_residuals
d1 <- deviance(m2) / df.residual(m2) ; d1

## [1] 15.06862

# Sqrt of MS_residuals
sqrt(d1)

## [1] 3.881832

# sigma
summary(m2)$sigma  # Same as Above

## [1] 3.881832

# Fitted Value =s of x*beta^T
x_beta_hat1 <- fitted(m2)  ; head(x_beta_hat1)

##         1         2         3         4         5         6
##  4.837660  4.553852  4.816981 15.874115 19.869008 21.018327

# Predicted Values
pred11 <- predict(m2)  ; head(pred11)

##         1         2         3         4         5         6
##  4.837660  4.553852  4.816981 15.874115 19.869008 21.018327

# Plot the graph b/w fitted(m1) & predict(m1)
plot(x_beta_hat1 , pred11)
```




```
# Variance - Covaiance of beta
vcov(m2)

##               (Intercept)       Girth      Height
## (Intercept)  74.6189461  0.43217138 -1.05076889
## Girth         0.4321714  0.06983578 -0.01786030
## Height       -1.0507689 -0.01786030  0.01693933
```

2.2.5 Centered Form of MLR

General Form of Centered Model of MLR

$$y_i = \beta_0 + \beta_1(x_{i1} - \bar{x_1}) + \beta_2(x_{i2} - \bar{x_2}) + \cdots + \beta_j\left(x_{ij} - \bar{x_j}\right) + \epsilon_{ij} \quad ; i \neq j = 1,2,\cdots,n$$

Our Fitted Model for *trees* data . $Volume = \beta_0 + \beta_1\big(Girth - mean(Girth)\big) + \beta_2\big(Height - nean(Height)\big) + error$

```
d1 <- trees # Assign trees to d1
d1 <- transform(d1, Girth.c = Girth - mean(Girth) , Height.c = Height -
mean(Height))
head(d1)

##   Girth Height Volume   Girth.c Height.c
## 1   8.3     70   10.3 -4.948387       -6
## 2   8.6     65   10.3 -4.648387      -11
## 3   8.8     63   10.2 -4.448387      -13
## 4  10.5     72   16.4 -2.748387       -4
```

```
## 5  10.7     81   18.8 -2.548387        5
## 6  10.8     83   19.7 -2.448387        7

# Fit the centered form
mlr_c=lm(Volume ~ Girth.c + Height.c , data=d1)

# Summary of mlr_c
summary(mlr_c)

##
## Call:
## lm(formula = Volume ~ Girth.c + Height.c, data = d1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4065 -2.6493 -0.2876  2.2003  8.4847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.1710     0.6972  43.275   <2e-16 ***
## Girth.c       4.7082     0.2643  17.816   <2e-16 ***
## Height.c      0.3393     0.1302   2.607   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.882 on 28 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9442
## F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16

# Comparision of Centered and Non - centered Model
m22 <- lm(Volume ~ Girth + Height , trees)
# Coefficients
print(mlr_c)

##
## Call:
## lm(formula = Volume ~ Girth.c + Height.c, data = d1)
##
## Coefficients:
## (Intercept)      Girth.c     Height.c
##     30.1710       4.7082       0.3393

print(m22)

##
## Call:
## lm(formula = Volume ~ Girth + Height, data = trees)
##
## Coefficients:
## (Intercept)        Girth       Height
##    -57.9877       4.7082       0.3393
```

```r
# Variance - Covariance of beta's
vcov(mlr_c)
```

```
##                 (Intercept)       Girth.c      Height.c
## (Intercept)  4.860845e-01  1.658242e-17 -2.938295e-19
## Girth.c      1.658242e-17  6.983578e-02 -1.786030e-02
## Height.c    -2.938295e-19 -1.786030e-02  1.693933e-02
```

```r
vcov(m22)
```

```
##               (Intercept)       Girth      Height
## (Intercept)  74.6189461  0.43217138 -1.05076889
## Girth         0.4321714  0.06983578 -0.01786030
## Height       -1.0507689 -0.01786030  0.01693933
```

```r
# Correlation Matrix
cov2cor(vcov(mlr_c))
```

```
##                 (Intercept)       Girth.c      Height.c
## (Intercept)  1.000000e+00  9.000217e-17 -3.238109e-18
## Girth.c      9.000217e-17  1.000000e+00 -5.192801e-01
## Height.c    -3.238109e-18 -5.192801e-01  1.000000e+00
```

```r
cov2cor(vcov(m22))
```

```
##               (Intercept)      Girth     Height
## (Intercept)   1.0000000  0.1893182 -0.9346189
## Girth         0.1893182  1.0000000 -0.5192801
## Height       -0.9346189 -0.5192801  1.0000000
```

2.2.6 Centered Form of MLR by Own Function

```r
mlrc=function(y,x1,x2) {
X=cbind(1,x1,x2) # model matrix

n=nrow(X)

p1=ncol(X)

XtX=crossprod(X,X)

Xty=crossprod(X,y)

beta= solve(XtX,Xty)

resid=y-X %*% beta

rss= sum(resid^2)

msresid=rss/(n-p1)

sebeta=sqrt(diag(msresid*solve(XtX)))
```

```
tratio=beta/sebeta

pvalue=2*(1-pt(abs(tratio),df=n-p1))

out=data.frame(Reg_Coeff = beta , SE_beta = sebeta , T_value = tratio ,
P_value = pvalue)

out=round(out,3) # Round upto 3 digits

return(out)
}

dump("mlrc",file="mlrc.txt")

## Analyse the data 'trees' using 'volume' as response and
# 'Girth' and 'Height' as regressors.
d2=trees

y=d2$Volume
x1=trees$Girth-mean(d2$Girth)
x2=trees$Height-mean(d2$Height)
M4c=mlrc(y,x1,x2)

print(M4c)

##    Reg_Coeff SE_beta T_value P_value
##       30.171   0.697  43.275   0.000
## x1     4.708   0.264  17.816   0.000
## x2     0.339   0.130   2.607   0.014
```

2.2.7 Interactive MLR

Model : $Sales = \beta_0 + \beta_1(TV) + \beta_2(Radio) + \beta_3(TV \times Radio) + \epsilon$

```
library(readr)

advertising <- read_csv("Advertising.csv")

## New names:
## * `` -> ...1

## Rows: 200 Columns: 5

## -- Column specification ----------------------------------------------
## Delimiter: ","
## dbl (5): ...1, TV, radio, newspaper, sales

##
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
int_model <- lm(sales ~ TV + radio + TV:radio , data = advertising)

# int_model <- lm(sales ~ TV + radio + TV * radio , data = advertising)

summary(int_model)
```

```
##
## Call:
## lm(formula = sales ~ TV + radio + TV:radio, data = advertising)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3366 -0.4028  0.1831  0.5948  1.5246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.750e+00  2.479e-01  27.233   <2e-16 ***
## TV          1.910e-02  1.504e-03  12.699   <2e-16 ***
## radio       2.886e-02  8.905e-03   3.241   0.0014 **
## TV:radio    1.086e-03  5.242e-05  20.727   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9435 on 196 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## F-statistic:  1963 on 3 and 196 DF,  p-value: < 2.2e-16
```

Fitted Model :** $\widehat{Sales} = 6.19 + 0.0423(TV) + 0.0422(Radio) + 0.0004(TV \times Radio)$

2.3 Non - Linear Relationship

we assume that there is a *Linear Regression* between Response and Predictors , but in many case there is a *Non-Linear Relationship*. We present a very simple way to directly extend the linear model to accommodate non-linear relationships, using **Polynomial Regression**.

We can see in *Auto* dataset from *ISLR* package .

```
library(ISLR)
data("Auto")
head(Auto)
```

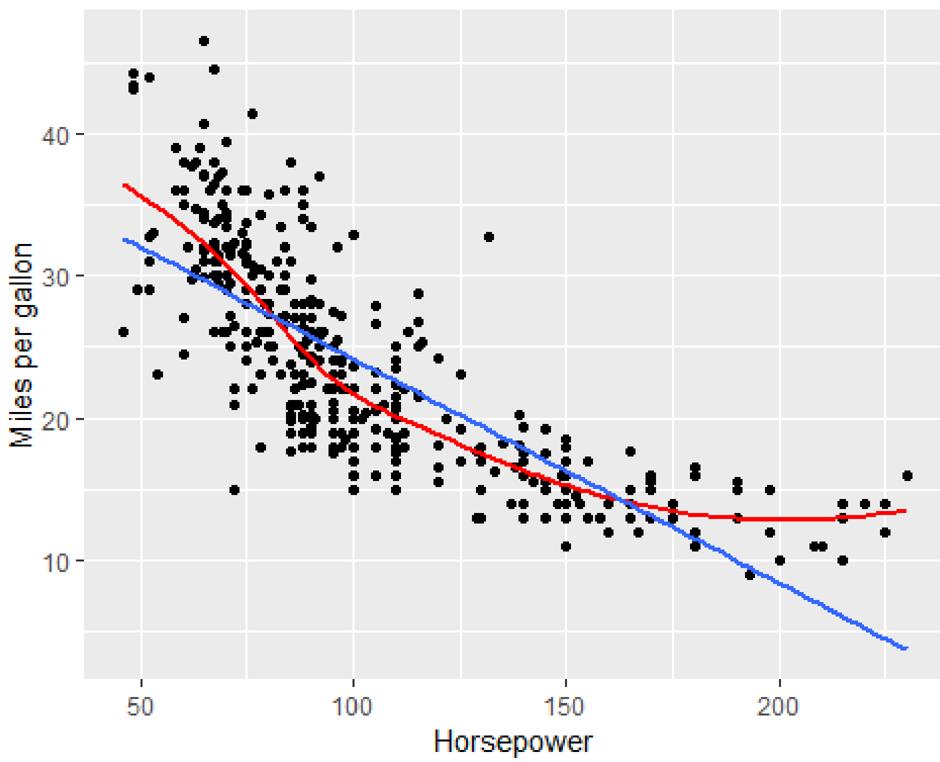
```
##   mpg cylinders displacement horsepower weight acceleration year
## 1  18         8          307        130   3504         12.0   70
## 2  15         8          350        165   3693         11.5   70
## 3  18         8          318        150   3436         11.0   70
## 4  16         8      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

```
##   origin                      name
## 1      1 chevrolet chevelle malibu
## 2      1         buick skylark 320
## 3      1        plymouth satellite
## 4      1             amc rebel sst
## 5      1               ford torino
## 6      1          ford galaxie 500
```

```
# WE can see by using Scatterplot
ggplot(Auto , aes(horsepower , mpg)) +
  geom_point() + geom_smooth(se = F , col= "red") +
  geom_smooth(method = "lm", se = F) +
  labs(x = "Horsepower" , y = "Miles per gallon")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Here `Red` line is the *best fitted line* while `Steelblue` line is *linear* which is not a 'best fitted line' . so , here we can not directly apply Linear Regression . Firstly we convert it into LR .

Model : $mpg = \beta_0 + \beta_1(horsepower) + \beta_2(horsepoer)^2 + \epsilon$

```
h_lm <- lm(mpg ~ horsepower + I(horsepower^2)  , data = Auto)
summary(h_lm)
```

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

```
## lm(formula = mpg ~ horsepower + I(horsepower^2), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7135  -2.5943  -0.0859   2.2868  15.8961
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     56.9000997  1.8004268   31.60   <2e-16 ***
## horsepower      -0.4661896  0.0311246  -14.98   <2e-16 ***
## I(horsepower^2)  0.0012305  0.0001221   10.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.374 on 389 degrees of freedom
## Multiple R-squared:  0.6876, Adjusted R-squared:  0.686
## F-statistic:   428 on 2 and 389 DF,  p-value: < 2.2e-16
```

Fitted Model : $\widehat{mpg} = 56.9 - 0.466(horsepower) + 0.001(horsepower)^2$ The value of $R^2 = 0.68$, that means our Model is **68%** Good.

2.3.0.1 Potential problems

When we fit a *Linear regression Model* to a particular dataset. There are many problems which we are face during fitting a Model . Some of them are as follows : 1. *Non-Linearity of the Response - Predictor Relationship* 2. *Correlation of Error Terms* 3. *Non-Constant variance of error terms* 4. *Outliers* 5. *Higher-Leverage Points* 6. *Collinearity*

2.4 Comparison of Linear Regression with K-Nearest Neighbors

2.4.1 K-Nearest Neighbors (KNN) Regression

One of the simplest and best-known non-parametric methods, K-nearest neighbors regression (KNN regression).

The *MSE* for *KNN* as a function of *1/K* (on the log scale). Linear regression achieves a lower test MSE than does KNN regression, since f(X) is in fact linear. For KNN regression, the best results occur with a very large value of K, corresponding to a small value of 1/K $KNN \propto \frac{1}{k}$

2.5 Labs

Load Some packages for differeent datasets .

```
library(MASS)
library(ISLR)
library(ISLR2)
```

2.5.1 Simple Linear Regression (SLR)

Here we fit a Model of Simple Linear Regression Model of *Boston* dataset from *MASS* package , which records *medv (median house value)* for 506 census tracts in Boston. We will seek to predict medv using 12 predictors such as *rm* (average number of rooms per house), *age* (average age of houses), and *lstat* (percent of households with low socioeconomic status) .

Model : $medv = \beta_0 + \beta_1(lstat) + \epsilon$

```
data("Boston")

# Names of columns od data
names(Boston)

##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"
##  [7] "age"     "dis"     "rad"     "tax"     "ptratio" "lstat"
## [13] "medv"

# Head of Dataset
head(Boston)

##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7

# Fit the Model
boston_model <- lm(medv ~ lstat , data = Boston)
summary(boston_model)

##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16

# OR we can also fitt as
# attach (Boston)
# model <- lm(medv ~ lstat) ; summary(model)
```

Fitted Model : $\widehat{medv} = 34.55 - 0.95(lstat)$ The value of $R^2 = 0.54$, that means **54%** of variability is explained in Our Model .

Task : Find out the other information stored in above fitted model .

```
#  Formula of Model
boston_model$call

## lm(formula = medv ~ lstat, data = Boston)

# Names
names(boston_model)

##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"

# Coefficents
boston_model$coefficients

## (Intercept)       lstat
##  34.5538409  -0.9500494

# Residuals
res <- boston_model$residuals
head(res)

##          1          2          3          4          5          6
## -5.8225951 -4.2703898  3.9748580  1.6393042  6.7099222 -0.9040837

# Effect
effect <- boston_model$effects
head(effect)

## (Intercept)       lstat
## -506.862945 -152.459549    4.426601    2.117838    7.129713
##
##   -0.481343
```

```r
# Rank
boston_model$rank

## [1] 2

# Fitted Values y_cap
fitted_value <- boston_model$fitted.values
head(fitted_value)

##        1        2        3        4        5        6
## 29.82260 25.87039 30.72514 31.76070 29.49008 29.60408

# Assign
boston_model$assign

## [1] 0 1

# Qr
qr <- boston_model$qr
head(qr)

## $qr
##      (Intercept)         lstat
## 1   -22.49444376 -2.846236e+02
## 2     0.04445542  1.604754e+02
## 3     0.04445542  5.169934e-02
## 4     0.04445542  5.849166e-02
## 500   0.04445542 -1.728319e-02


##  [ reached getOption("max.print") -- omitted 6 rows ]
## attr(,"assign")
## [1] 0 1
##
## $qraux
## [1] 1.044455 1.019856
##
## $pivot
## [1] 1 2
##
## $tol
## [1] 1e-07
##
## $rank
## [1] 2

# Residuals
boston_model$df.residual

## [1] 504
```

```r
# Terms
boston_model$terms

## medv ~ lstat
## attr(,"variables")
## list(medv, lstat)
## attr(,"factors")
##       lstat
## medv      0
## lstat     1
## attr(,"term.labels")
## [1] "lstat"
## attr(,"order")
## [1] 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
## attr(,"predvars")
## list(medv, lstat)
## attr(,"dataClasses")
##      medv     lstat
## "numeric" "numeric"

# Values of Dependent and Independent Columns
val <- boston_model$model
head(val)

##   medv lstat
## 1 24.0  4.98
## 2 21.6  9.14
## 3 34.7  4.03
## 4 33.4  2.94
## 5 36.2  5.33
## 6 28.7  5.21

# Confidence Interval
confint(boston_model)

##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505

# confidence intervals
predict(boston_model , data.frame(lstat = (c(5, 10, 15))),
interval = "confidence")

##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
```

```
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461

# R-Square Value
summary(boston_model)$r.sq

## [1] 0.5441463

# RSE
summary(boston_model)$sigma

## [1] 6.21576
```

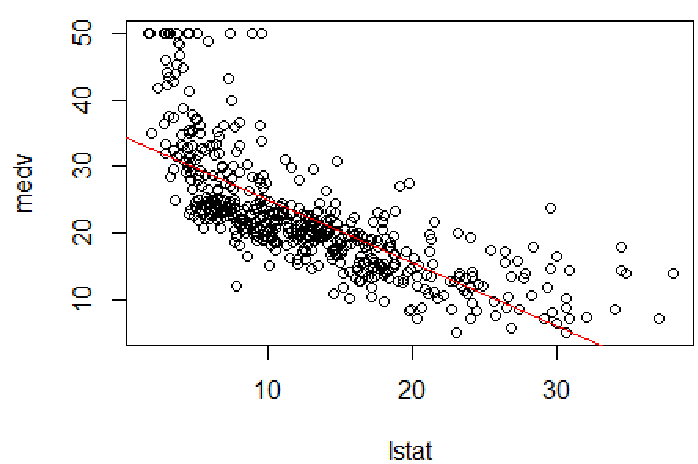
The *95 % Confidence Interval* associated with a *lstat* value of *10* is **(24.47 , 25.63)**

```
# predict() prediction intervals
predict(boston_model , data.frame(lstat = (c(5, 10, 15))),
interval = "prediction")

##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

The *95 % Prediction Interval* associated with a *lstat* value of *10* is **(12.828 , 37.28)**

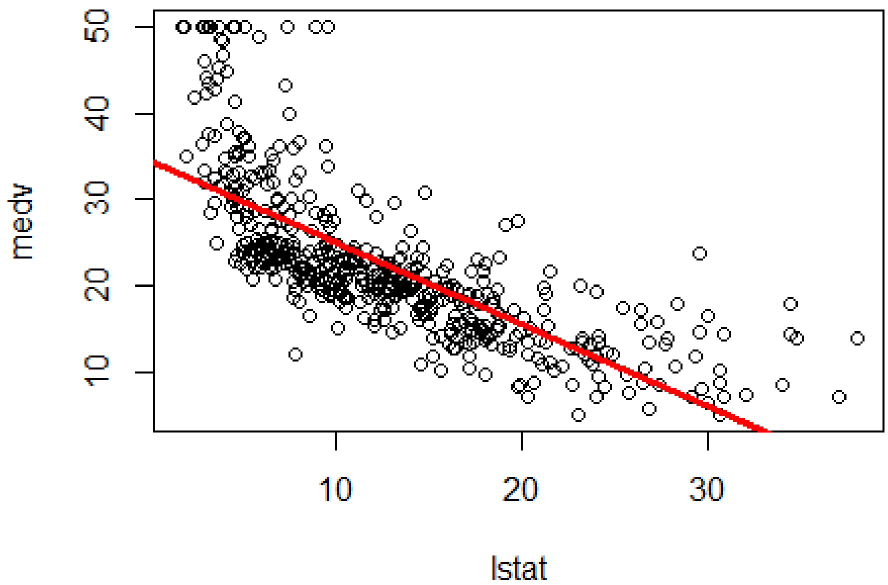
Task : Plot *medv* and *lstat* along with the least squares regression line .

```
plot(lstat , medv)  # Scatter Plot
abline (boston_model , col = "red")
```

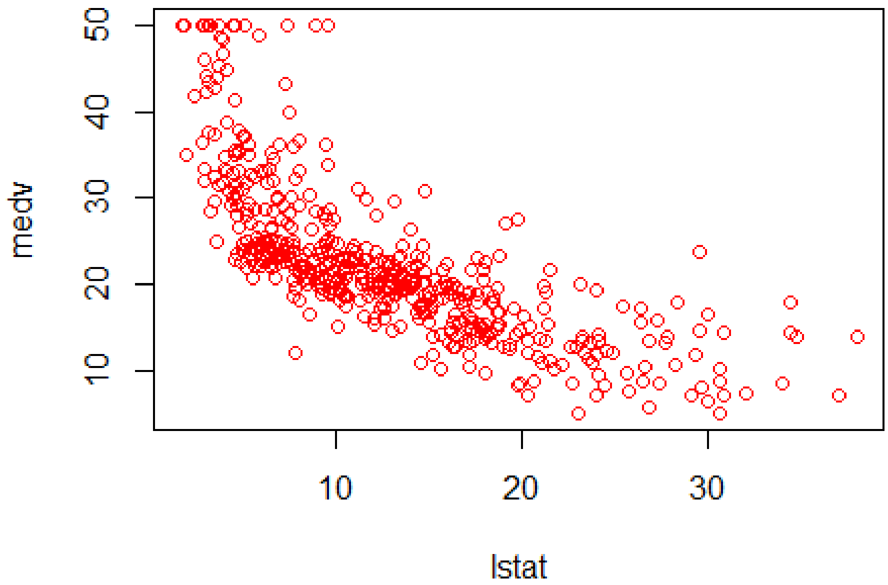


There is some evidence for non-linearity in the relationship between *lstat* and *medv* .

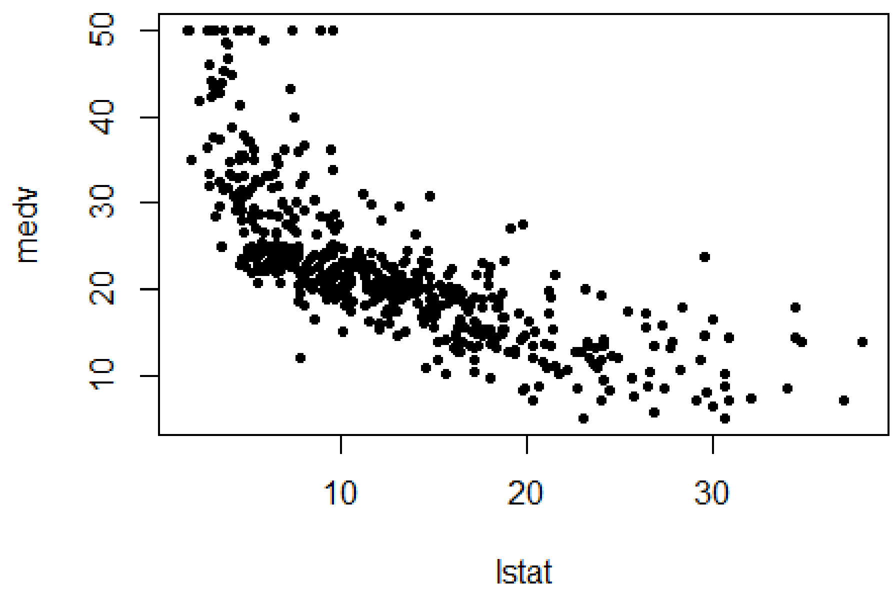
```
plot(lstat , medv)
abline (lm.fit , lwd = 3)
abline (lm.fit , lwd = 3, col = " red ")
```



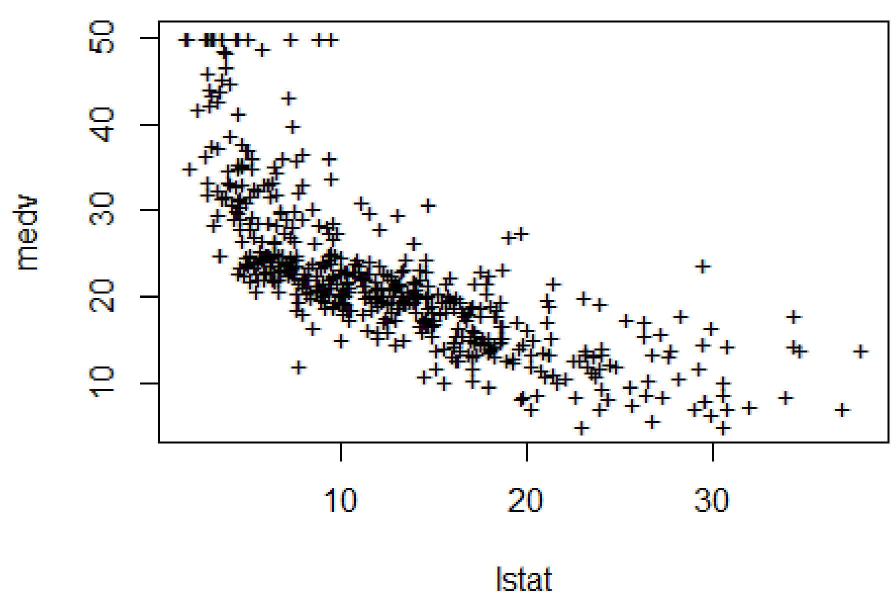
```r
plot (lstat , medv , col = " red ")
```



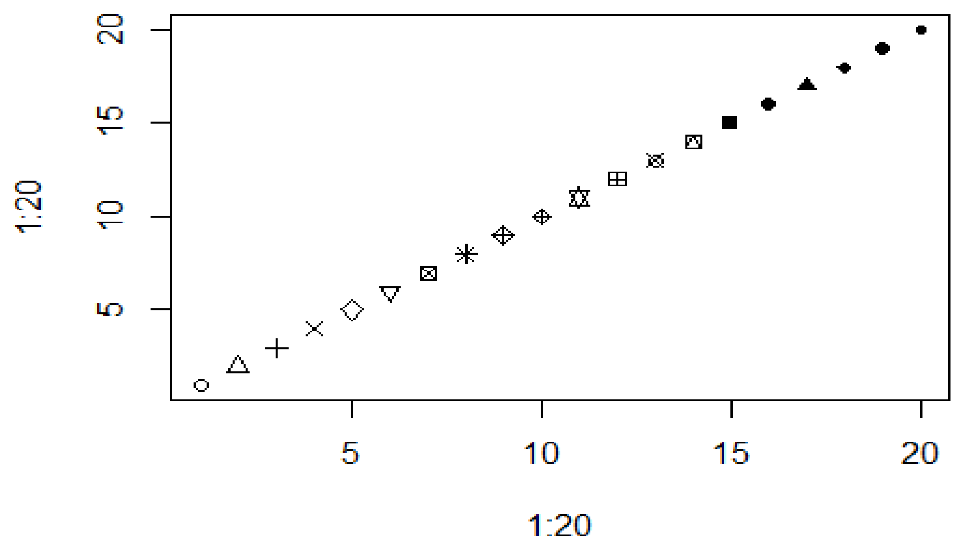
```
plot (lstat , medv , pch = 20)
```




```
plot (lstat , medv , pch = "+")
```

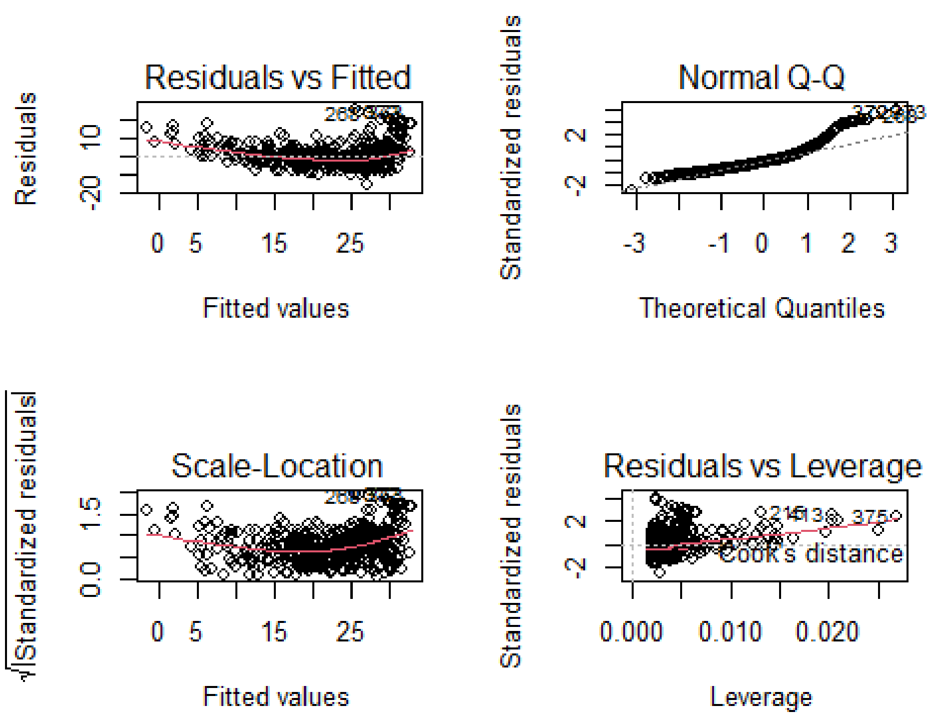


```r
plot (1:20, 1:20, pch = 1:20)
```



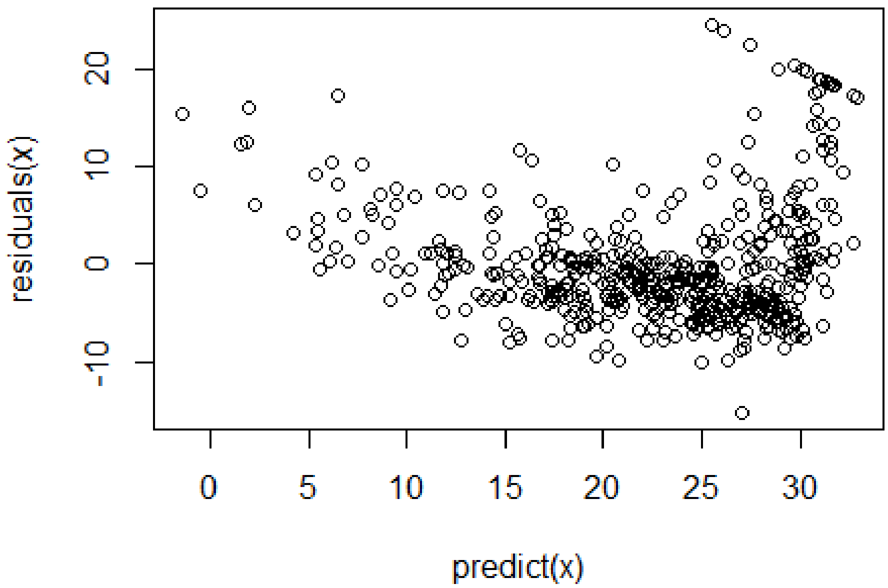
Task : Plot the Graphs b/w *Predicted* and *Residuals* values.

```r
par(mfrow = c(2 , 2))
plot(boston_model)
```

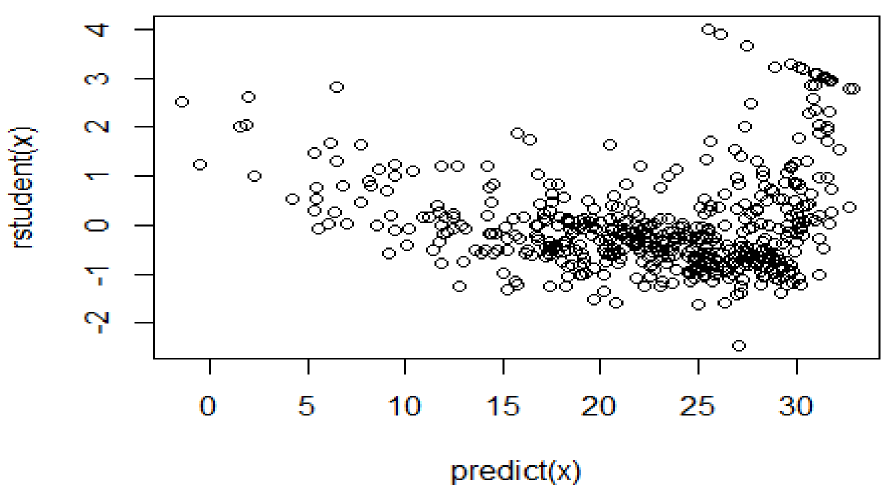


Alternatively, we can compute the residuals from a linear regression fit

```
x <- boston_model
plot(predict(x) , residuals(x)) # plot(x$predict , x$residuals)
```

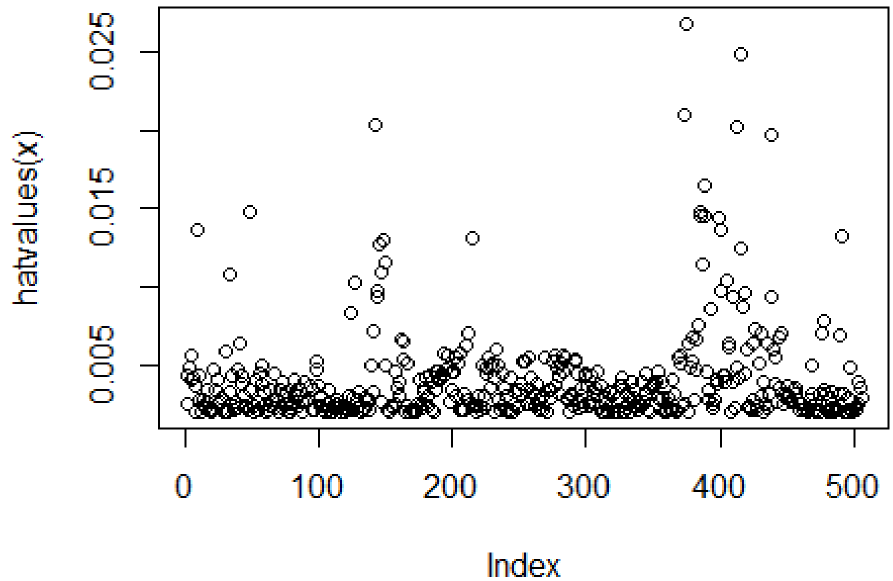



```
plot(predict(x) , rstudent(x))
```



On the basis of the residual plots, there is some evidence of non-linearity. Leverage statistics can be computed for any number of predictors using the *hatvalues()* function

```
plot(hatvalues(x))
```




```
# Maximum hat value
which.max(hatvalues(x))

## 375
## 375
```

375 is the the largest *leverage statistic*.

2.5.2 Multiple Linear Regression (MLR)

We will again fitt the Model of same data *(Boston)*

Model : $medv = \beta_0 + \beta_1(lstat) + \beta_2(age) + \epsilon$

```
m <- lm(medv ~ lstat + age , data = Boston)
summary(m)

##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

Fitted Model : $\widehat{medv} = 33.22 - 1.03(lstat) + 0.034(age)$ The value of R^2 is 0.55 , which means that our Model is **55%** is Good .

TASK : Fit the Model for all the variables of *Boston* data.

Model : $medv = \beta_0 + \beta_1(crime) + \beta_2(zn) + \beta_3(indus) + \beta_4(chas) + \beta_5(nox) + \beta_6(rm) + \beta_7(age) + \beta_8(dis) + \beta_9(rad) + \beta_{10}(tax) + \beta_{11}(ptratio) + \beta_{12}(lstat)$

```
mm <- lm(medv ~ . , Boston)
summary(mm)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1304  -2.7673  -0.5814   1.9414  26.2526
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
## crim         -0.121389   0.033000  -3.678 0.000261 ***
## zn            0.046963   0.013879   3.384 0.000772 ***
## indus         0.013468   0.062145   0.217 0.828520
## chas          2.839993   0.870007   3.264 0.001173 **
## nox         -18.758022   3.851355  -4.870 1.50e-06 ***
## rm            3.658119   0.420246   8.705  < 2e-16 ***
## age           0.003611   0.013329   0.271 0.786595
## dis          -1.490754   0.201623  -7.394 6.17e-13 ***
## rad           0.289405   0.066908   4.325 1.84e-05 ***
## tax          -0.012682   0.003801  -3.337 0.000912 ***
## ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
## lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.798 on 493 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7278
## F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```

Fitted Model : $medv = 41.61 - 0.12(crime) + 0.046(zn) + 0.013(indus) + 2.84(chas) - 18.76(nox) + 3.66(rm) + 0.003(age) - 1.49(dis) + 0.29(rad) - 0.012(tax) - 0.94(ptratio) - 0.55(lstat)$

The value of R^2 is 0.73,that means **73%** variability is explain in Our Model .

Task : Find the R^2 , RSE , VIF Values.

```
# R-Square
summary(mm)$r.sq

## [1] 0.734307

# RSE
summary(mm)$sigma

## [1] 4.798034
```

To find the **VIF** , firstly we load **car** *library* .

```
library(car)
vif(mm)

##     crim       zn    indus     chas      nox       rm      age
## 1.767486 2.298459 3.987181 1.071168 4.369093 1.912532 3.088232
##      dis      rad      tax  ptratio    lstat
## 3.954037 7.445301 9.002158 1.797060 2.870777
```

In the above model *mm* **age** variable is not significant . So we want to remove only this variable then

```
mm1 <- lm(medv ~ . -age , data = Boston)  ; mm1

##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Coefficients:
## (Intercept)         crim           zn        indus         chas
##    41.52513     -0.12143      0.04651      0.01345      2.85277
##         nox           rm          dis          rad          tax
##   -18.48507      3.68107     -1.50678      0.28794     -0.01265
##     ptratio        lstat
##    -0.93465     -0.54741

# We can also update the the above model (mm)
mm2 <- update(mm , ~ . -age)   ; mm2
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + chas + nox + rm + dis +
##     rad + tax + ptratio + lstat, data = Boston)
##
## Coefficients:
## (Intercept)         crim           zn        indus         chas
##    41.52513     -0.12143      0.04651      0.01345      2.85277
##         nox           rm          dis          rad          tax
##   -18.48507      3.68107     -1.50678      0.28794     -0.01265
##     ptratio        lstat
##    -0.93465     -0.54741
```

The results of *mm1* and *mm2* are same .

2.5.3 Intractive MOdel

We will again use same Data **Boston**

Model : $medv = \beta_0 + \beta_1(lstat \times age) + \epsilon$

```
im <- lm(medv ~ lstat*age , data = Boston)
summary(im)
```

```
##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Fitted Model : $\widehat{medv} = 36.08 - 1.39(lstat) - 0.0007(age) + 0.004(lstat\!:age)$ The value of R^2 is 0.55 , that means only **55%** of variability is explain in our model.

2.5.4 Non - Linear Tansformations of the Predictos

We can also use **lm()** function for *non-linear models* as well by using their *transformation* .

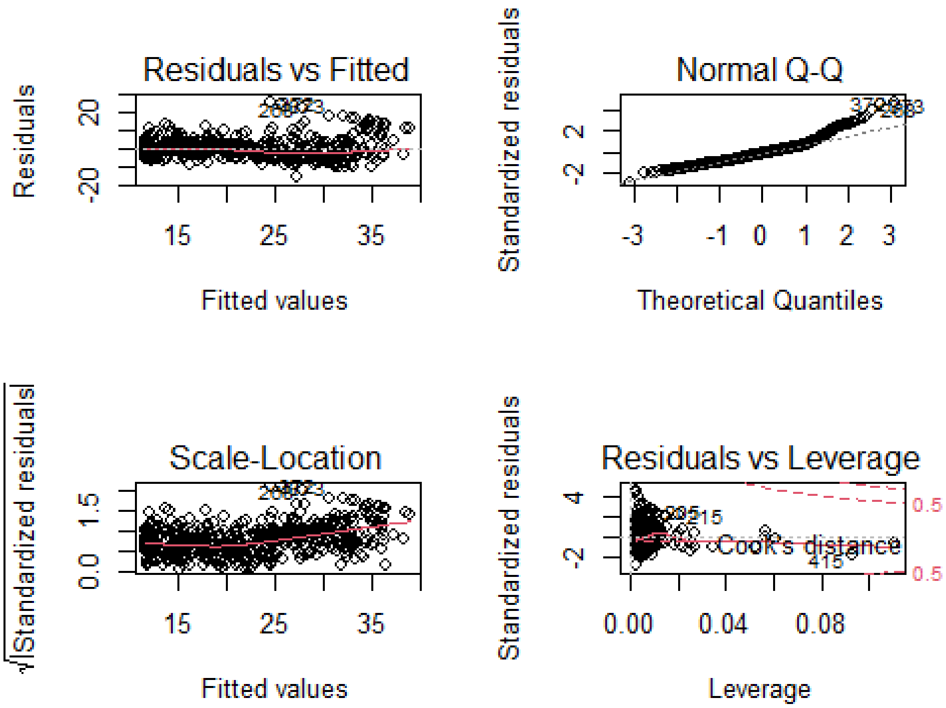
Model : $medv = \beta_0 + \beta_1(lstat) + \beta_2(lstat)^2 + \epsilon$

```
tm <- lm(medv ~ lstat + I(lstat^2) , data = Boston)
summary(tm)

##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

Fitted Model : $\widehat{medv} = 42.86 - 2.33(lstat) + 0.043(lstat)^2$ The value of R^2 is 0.64 , that means only **64%** of variability is explain in our model.

Task : Plotting

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



Ploy Fitted Model :

In order to create a cubic fit, we can include a predictor of the form $I(X^3)$. However, this approach can start to get cumbersome for higher-order polynomials. A better approach involves using the **poly()** function poly() to create the **polynomial** within **lm()** .
For example, the following command produces a fifth-order polynomial fit:
Model : $medv = \beta_0 + \beta_1(lstat) + \beta_2(lstat)^2 + \beta_3(lstat)^3 + \beta_4(lstat)^4 + \beta_5(lstat)^5 + \epsilon$

```
pm <- lm(medv ~ poly(lstat , 5))
summary(pm)

##
## Call:
## lm(formula = medv ~ poly(lstat, 5))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5433  -3.1039  -0.7052   2.0844  27.1153
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         22.5328     0.2318  97.197  < 2e-16 ***
## poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
## poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
## poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
## poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
## poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.215 on 500 degrees of freedom
## Multiple R-squared:  0.6817, Adjusted R-squared:  0.6785
## F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

This suggests that including additional polynomial terms, up to fifth order, leads to an improvement in the model fit! .

Log Fitted Model : A linear model applied to the output of the *poly()* function will have the same fitted values as a linear model applied to the raw polynomials (although the coefficient estimates, standard errors, and p-values will differ). In order to obtain the raw polynomials from the poly() function, the argument *raw = TRUE* must be used .

Model : $medv = \beta_0 + \beta_1 \times log(rm) + \epsilon$

```
summary(lm(medv ~ log(rm) , data = Boston))

##
## Call:
## lm(formula = medv ~ log(rm), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.487  -2.875  -0.104   2.837  39.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -76.488      5.028  -15.21   <2e-16 ***
## log(rm)       54.055      2.739   19.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.915 on 504 degrees of freedom
## Multiple R-squared:  0.4358, Adjusted R-squared:  0.4347
## F-statistic: 389.3 on 1 and 504 DF,  p-value: < 2.2e-16
```

Fitted Model : $medv = -76.48 + 54.05 \times log(rm)$

2.5.5 Qualitative Predictors

Here we will work on **Carseats** data from *ISLR2* package .

```
data("Carseats")
names(Carseats)

##  [1] "Sales"       "CompPrice"   "Income"      "Advertising"
##  [5] "Population"  "Price"       "ShelveLoc"   "Age"
##  [9] "Education"   "Urban"       "US"

head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age
## 1  9.50       138     73          11        276   120       Bad  42
## 2 11.22       111     48          16        260    83      Good  65
## 3 10.06       113     35          10        269    80    Medium  59
## 4  7.40       117    100           4        466    97    Medium  55
## 5  4.15       141     64           3        340   128       Bad  38
## 6 10.81       124    113          13        501    72       Bad  78
##   Education Urban  US
## 1        17   Yes Yes
## 2        10   Yes Yes
## 3        12   Yes Yes
## 4        14   Yes Yes
## 5        13   Yes  No
## 6        16    No Yes
```

Model : $Sales = \beta_0 + \beta_1(Income{:}Advertising) + \beta_2(Price{:}Age) + \epsilon$

```
cm <- lm(Sales ~ . + Income : Advertising + Price : Age , data = Carseats)
summary(cm)
```

```
##
## Call:
## lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9208 -0.7503  0.0177  0.6754  3.3413
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.5755654  1.0087470   6.519 2.22e-10 ***
## CompPrice           0.0929371  0.0041183  22.567  < 2e-16 ***
## Income              0.0108940  0.0026044   4.183 3.57e-05 ***
## Advertising         0.0702462  0.0226091   3.107 0.002030 **
## Population          0.0001592  0.0003679   0.433 0.665330
## Price              -0.1008064  0.0074399 -13.549  < 2e-16 ***
## ShelveLocGood       4.8486762  0.1528378  31.724  < 2e-16 ***
## ShelveLocMedium     1.9532620  0.1257682  15.531  < 2e-16 ***
## Age                -0.0579466  0.0159506  -3.633 0.000318 ***
## Education          -0.0208525  0.0196131  -1.063 0.288361
## UrbanYes            0.1401597  0.1124019   1.247 0.213171
## USYes              -0.1575571  0.1489234  -1.058 0.290729
## Income:Advertising  0.0007510  0.0002784   2.698 0.007290 **
## Price:Age           0.0001068  0.0001333   0.801 0.423812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.011 on 386 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8719
## F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16
```

The **contrasts()** function returns the coding that R uses for the dummy contrasts() variables . For further detail use *??contrasts()*.

```
attach (Carseats)  # To Loading Data

## The following objects are masked from Carseats (pos = 3):
##
##     Advertising, Age, CompPrice, Education, Income,
##     Population, Price, Sales, ShelveLoc, Urban, US

## The following objects are masked from Carseats (pos = 8):
##
##     Advertising, Age, CompPrice, Education, Income,
##     Population, Price, Sales, ShelveLoc, Urban, US

contrasts (ShelveLoc)

##        Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```

R has created a *ShelveLocGood* dummy variable that takes on a value of *1* if the shelving location is good, and 0 otherwise.

3 Econometrics

1. **Econometrics :** Econometrics has developed methods for dealing with the random component of economic relation .
   Econometrics is an amalgam of *economic theory , mathematical economics , economics statistics* and *mathematical statistics* .

2. **Aims of econometrics :** i. Formulation and specification of econometric models . ii. Estimation and testing of models iii. Use of Models

3. **Types of Data :**

i. **Time Series Data :** Time series data give information about the numerical values of variables from period to period and are collected over time. For example, the data during the years 1990-2010 for monthly income constitutes a time series of data.

ii. **Cross - Sectional Data :** The cross-section data give information on the variables concerning individual agents (e.g., consumers or produces) at a given point of time. For example, a cross-section of a sample of consumers is a sample of family budgets showing expenditures on various commodities by each family, as well as information on family income, family composition and other demographic, social or financial characteristics.

iii. **Panel Data :** The panel data are the data from a repeated survey of a single (cross-section) sample in different periods of time.

3.1 Multicollinearity

Multicollinearity : The situation where the explanatory variables are intercorrelated is reffered to as Multicollinearity . When some or all of the explanatory variables are highly but not perfect collinear .

3.1.1 Sources of Multicollinearity

1. **Method of Data Collection**
2. **Model and Population Constraints**
3. **Existence of Identities** OR **Definitional Relationships**
4. **Imprecise Formulation of Model**
5. **An Over - Determined Model**

3.1.2 Consequences of Multicollinearity

1. The Precision of estimation falls . The loss of precision has three aspects :-
   i. Specific estimates may have very large errors .
   ii. These errors may be highly correlated one with another .
   iii. The sampling variance of the coefficients will be very large.
2. Investigators are sometimes led to drop variables incorrectly from an analysis because their coefficients are not significantly different from zero .

3.1.3 Multicollinearity Diagnostics

1. **Determinant of** $X'X$ $(|X'X|)$
2. **Inspection of Correlation Matrix**
3. **Determinant of Correlation Matrix :** Thus a value close to 0 is an indication of a high degree of multicollinearity. Any value of D between 0 and 1 gives an idea of the degree of multicollinearity .
4. **Measure Based on Partial Regression** 5. **Variance Inflation Factor (VIF)** :

$$\boldsymbol{VIF_i = \frac{1}{1-R_i^2}}$$

In practice, usually, a **VIF > 5** or **10** indicates that the associated regression coefficients are poorly estimated because of multicollinearity. If regression coefficients are estimated by OLSE and its variance is $\sigma^2(X'X)$ So VIF indicates that a part of this variance is given by VIF_i .

Limitations :

(i) It sheds no light on the number of dependencies among the explanatory variables.

(ii) The rule of **VIF > 5** or **10** is a rule of thumb which may differ from one situation to another situation.

3.1.4 Remedies for Multicollinearity

1. **Obtain More Data**
2. **Drop some Variables that are Collinear**
3. **Use some Relevant Prior Information**
4. **Employ Generalized Inverse**
5. **Use of Principal component Regression** 6. **Ridge Regression**

3.2 Auto-Correlation

Auto - Correlation : In Regression Model $Y = X\beta + \epsilon$, the assumption $E(\epsilon\epsilon') = \sigma^2 I$, [i.e. The distribution term ϵ has constant variance σ^2 and $E(\epsilon_i, \epsilon_j) = 0$] is violated . Here we also consider that $E(\epsilon) = 0$ and $E(\epsilon\epsilon') = \sigma^2 \Omega$

3.2.1 Tests for Autocorrelation :**

1. **Durbin Watson test**

3.2.2 Consequences of Auto-Correaltion

1. β is unbiased but the Sampling variance is large
   2. We will obtain *Inefficient Prediction* i.e. Prediction with needlessly large sampling variance.

3.2.3 Source of Auto-Correlation

1. Carry over effect atleast in part is an important source of autocorrelation.
   2. Deleting of some variance .
   3. The misspecification of the form of relationship can introduced in the data .

3.3 Hetroscedasticity

Hetroscedasticity : In Regression Model $Y = X\beta + \epsilon$, the assumption $V(\epsilon) = \sigma^2 I$, violated .

3.3.1 Tests for Heteroscedasticity

1. **Bartlett Test**
   2. **Breusch Pagan Test**
   3. **Goldfeld Quandt Test**
   4. **Glesjer Test**
   5. **Test based on Spearman's rank correlation coefficient**
   6. **White Test**
   7. **Ramsey Test**
   8. **Harvey Phillips Test**
   9. **Szroeter Test**
   10. **Peak test (Non-Parametric) Test**