Linear Regression using R

LANGAT ERICK

2023-08-20

Simple Linear Regression

Simple Linear Regression (SLR) is a statistical technique for finding the existence of an association relationship between a dependent variable (response variable or outcome variable) and an independent variable (explanatory variable or predictor variable). Regression models do not establish a causal relationship between the dependent variable (Y) and the independent variable (X). In other words, using regression we cannot say that the value of Y depends on the value of X (or a change in the value of Y is caused due to a change in the value of X). We can only establish that the change in the value of Y is associated with the change in the value of X.

SLR implies that that there is only one independent variable in the model and the functional relationship between the dependent variable and the regression coefficient is linear. One of the functional forms of SLR is –

Y= b0+ b1**X** + e

where:
b0 and b1 are known as the regression beta coefficients or parameters:

- b0 is the intercept of the regression line; that is the predicted value when X = 0
- b1 is the slope of the regression line
- e is the error term (also known as the residual errors), the part of Y that cannot be explained by the regression model

The regression parameters are estimated using the method of **Ordinary Least Squares (OLS)**. In ordinary least squares, the objective is to find the optimal value of b0 and b1 that will minimize the Sum of Squares Error (SSE). The equation is used to predict a quantitative outcome Y on the basis of one single predictor variable X. Once, we build a statistically significant model, it is possible to use it for predicting future outcome on the basis of new X values.

```r
#load libraries
library(bookdown)
library(tinytex)
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(dplyr))
#import data set(campus-placement/Placement_Data_Full_Class.csv)
df <- read.csv('C:/Users/langa/OneDrive/Desktop/Dataset/Placement_Data_Full_Class.csv')
head(df)
```

```
##   sl_no gender ssc_p   ssc_b hsc_p   hsc_b    hsc_s degree_p  degree_t workex
## 1     1      M 67.00  Others 91.00  Others Commerce    58.00  Sci&Tech     No
## 2     2      M 79.33 Central 78.33  Others  Science    77.48  Sci&Tech    Yes
## 3     3      M 65.00 Central 68.00 Central     Arts    64.00 Comm&Mgmt     No
## 4     4      M 56.00 Central 52.00 Central  Science    52.00  Sci&Tech     No
## 5     5      M 85.80 Central 73.60 Central Commerce    73.30 Comm&Mgmt     No
## 6     6      M 55.00  Others 49.80  Others  Science    67.25  Sci&Tech    Yes
##   etest_p specialisation mba_p     status salary
## 1    55.0         Mkt&HR 58.80     Placed 270000
## 2    86.5        Mkt&Fin 66.28     Placed 200000
## 3    75.0        Mkt&Fin 57.80     Placed 250000
## 4    66.0         Mkt&HR 59.43 Not Placed     NA
## 5    96.8        Mkt&Fin 55.50     Placed 425000
## 6    55.0        Mkt&Fin 51.58 Not Placed     NA
```

```
names(df)
```

```
##  [1] "sl_no"          "gender"         "ssc_p"          "ssc_b"
##  [5] "hsc_p"          "hsc_b"          "hsc_s"          "degree_p"
##  [9] "degree_t"       "workex"         "etest_p"        "specialisation"
## [13] "mba_p"          "status"         "salary"
```

```
(placement_df <-df %>% dplyr::select(degree_p, mba_p)) %>%  head()
```

```
##   degree_p mba_p
## 1    58.00 58.80
## 2    77.48 66.28
## 3    64.00 57.80
## 4    52.00 59.43
## 5    73.30 55.50
## 6    67.25 51.58
```

```
 sum(is.na(placement_df))
```

```
## [1] 0
```

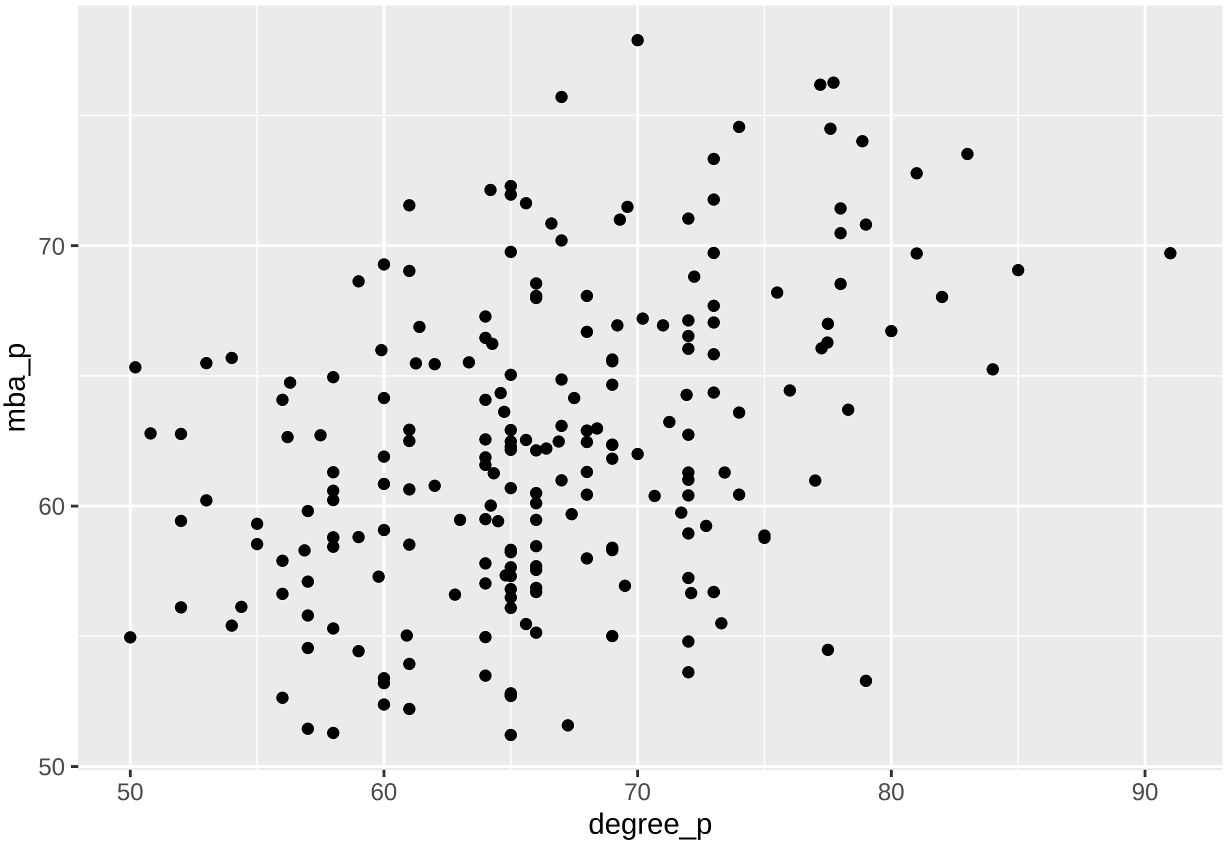
```
str(placement_df)
```

```
## 'data.frame':    215 obs. of  2 variables:
##  $ degree_p: num  58 77.5 64 52 73.3 ...
##  $ mba_p   : num  58.8 66.3 57.8 59.4 55.5 ...
```

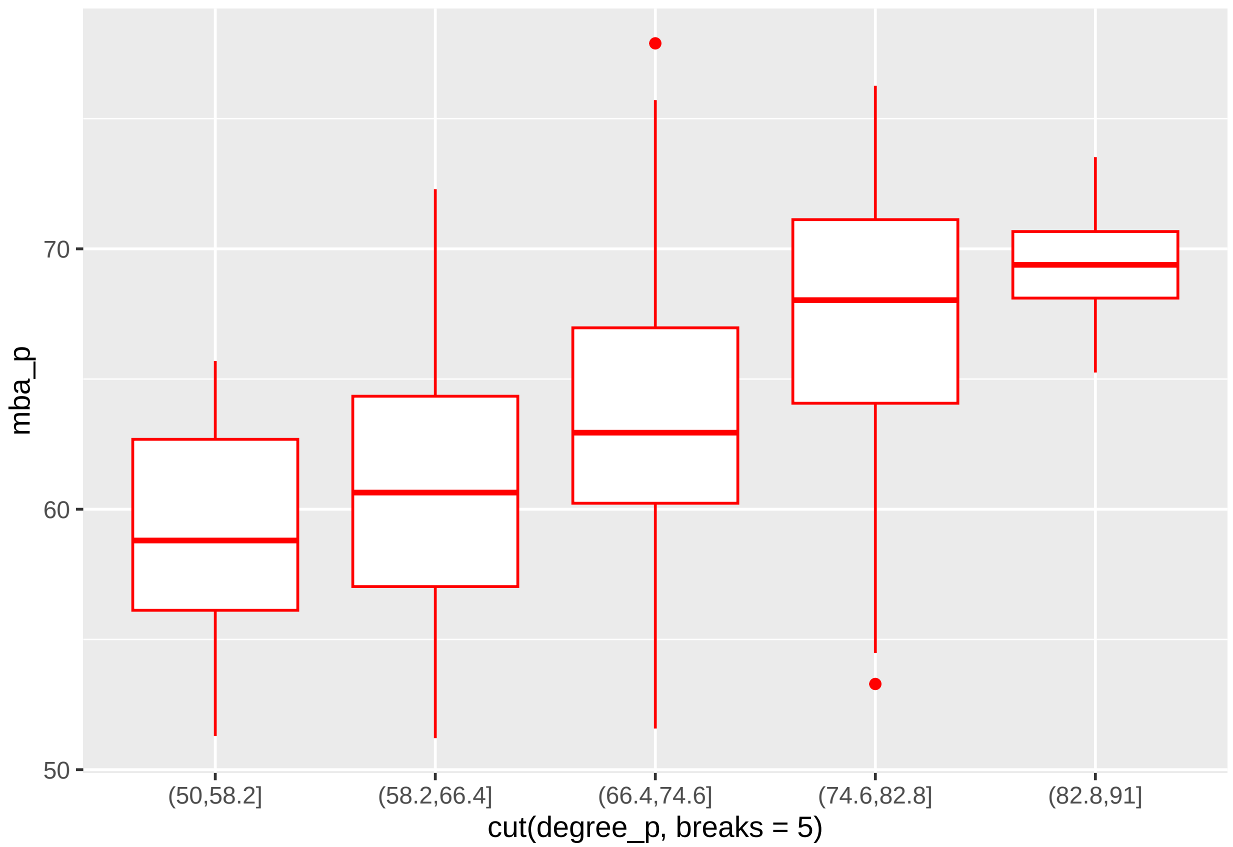
```
cor(placement_df)
```

```
##           degree_p     mba_p
## degree_p 1.0000000 0.4023638
## mba_p    0.4023638 1.0000000
```

```
# Basic Visualisation
ggplot(placement_df, aes(degree_p, mba_p)) +
  geom_point()
```

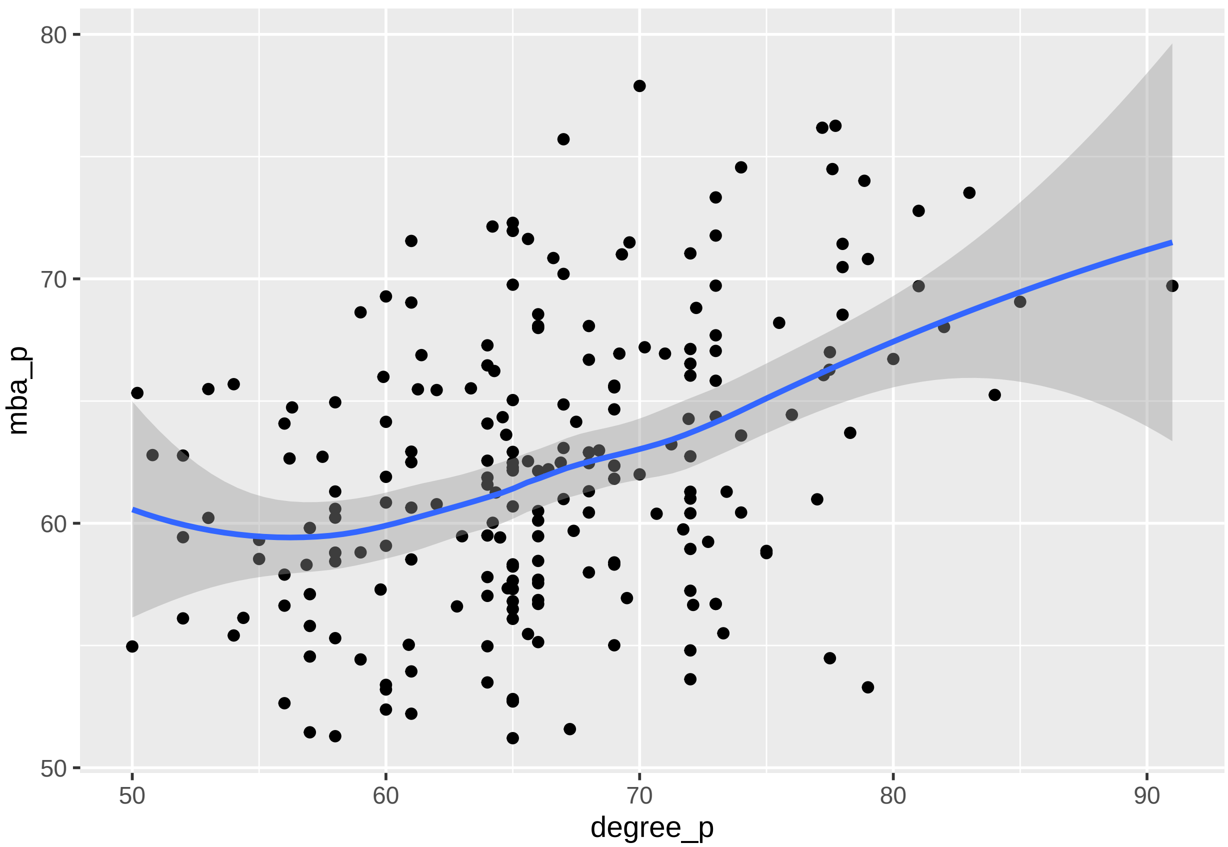


```
ggplot(placement_df, aes( cut(degree_p, breaks=5), mba_p)) +
  geom_boxplot(col='red')
```



```
ggplot(placement_df, aes(degree_p, mba_p)) +
  geom_point() + geom_smooth()
```

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



SLR Model

The function lm() can be used to determine the beta coefficients of the linear model.

```
# model
# mba_p = b0 + b1*degree_p
slr <- lm(mba_p~ degree_p, data = placement_df)
slr
```

```
##
## Call:
## lm(formula = mba_p ~ degree_p, data = placement_df)
##
## Coefficients:
## (Intercept)     degree_p
##      41.109        0.319
```

From the output, the estimated values of b0 and b1 are: b0 = 41.109 and b1 = 0.319

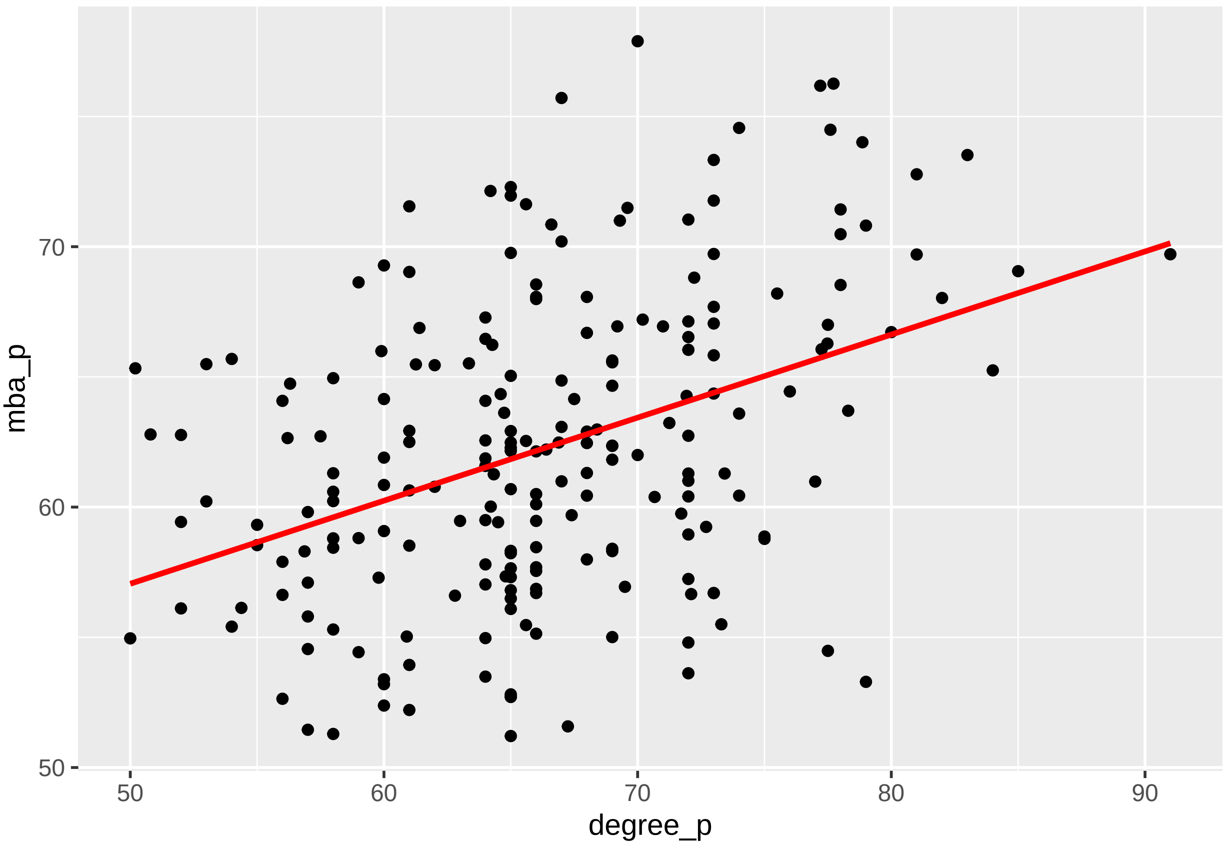
Regression equation:

The corresponding regression equation is given by- **mba__p = 41.109 + 0.319*degree__p**

Plotting the regression line:

```
ggplot(placement_df, aes(degree_p, mba_p))+ geom_point()+
       geom_smooth(method = 'lm', se=FALSE, col='red')
```

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



Validation of the SLR Model

It is important to validate the regression model to ensure its validity and goodness of fit before it can be used for practical applications. The following measures are used to validate the SLR model:

1. Coefficient of determination (R-square)
2. Hypothesis test for the regression coefficient (b1)
3. Analysis of Variance for overall model validity (relevant for MLR)
4. Residual analysis to validate the regression model assumptions
5. Outlier analysis

We start by displaying the statistical summary of the model using the function summary().

```r
#model assesment
summary(slr)
```

```
##
## Call:
## lm(formula = mba_p ~ degree_p, data = placement_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0166  -3.9567  -0.0328   3.6580  14.4540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 41.10877    3.32040  12.381  < 2e-16 ***
## degree_p     0.31896    0.04973   6.414 8.99e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 5.353 on 213 degrees of freedom
## Multiple R-squared:  0.1619, Adjusted R-squared:  0.158
## F-statistic: 41.15 on 1 and 213 DF,  p-value: 8.993e-10
```

```r
mba_P_hat <- fitted.values(slr)
head(mba_P_hat)
```

```
##        1        2        3        4        5        6
## 59.60843 65.82177 61.52219 57.69468 64.48852 62.55881
```

```r
head(placement_df$mba_p)
```

```
## [1] 58.80 66.28 57.80 59.43 55.50 51.58
```

```r
data.frame(head(mba_P_hat),head(placement_df$mba_p) )
```

```
##   head.mba_P_hat. head.placement_df.mba_p.
## 1        59.60843                    58.80
## 2        65.82177                    66.28
## 3        61.52219                    57.80
## 4        57.69468                    59.43
## 5        64.48852                    55.50
## 6        62.55881                    51.58
```

```r
mean(mba_P_hat)==mean(placement_df$mba_p)#should be the same
```

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

```r
model_residuals <- residuals(slr)
head(model_residuals)
```

```
##           1           2           3           4           5           6 
##  -0.8084339   0.4582309  -3.7221922   1.7353243  -8.9885175 -10.9788112
```

```
mean(model_residuals)#should be zero
```

```
## [1] -4.032627e-17
```

```
sum(model_residuals)#should be zero
```

```
## [1] -8.756884e-15
```

Coefficient of Determination (R-square or R2)

The coefficient of determination measures the percentage of variation in Y explained by the model. That is, R2 is the proportion of variation in the response variable explained by the regression model. The value of R2 lies between 0 and 1. Higher values of R2 implies better fit. There is no minimum threshold for R2, however a minimum value of R2 for a given significance value () can be derived using the relationship between the F-statistic and R2. Mathematically, for SLR the R-square is square of the Pearson correlation coefficient.
In our example, the R-square value is 0.1619, meaning that the model explains 16.19% of the variation in MBA percentage. The remaining variation is due to other factors that were not included in the model.
Note: A high R-square value is not necessarily a good indicator of the correctness of the model; it could be a spurious relationship. The adjusted R-squared is a statistic that modifies R2 by incorporating the sample size and number of explanatory variables.

Hypothesis test for regression coefficient (t-Test)

The regression coefficient b1 captures the existence of a linear relationship between the response variable and the explanatory variable. If b1 = 0, we can conclude that there is no statistically significant linear relationship between the two variables.
The null and alternative hypothesis for the SLR model can be stated as follows:

- Null: There is no relationship between X and Y (b1 = 0)
- Alternative: There is a relationship between X and Y (b1 **is not equal** 0)

In our example, since the p-value for degree_p is less than 0.05, we reject the Null and conclude that there is significant evidence suggesting a linear relationship between degree_p (X) and mba_p (Y).

Test for Overall Model: ANOVA (F-test)

Using ANOVA, we can test whether the overall model is statistically significant. However, for a SLR, the null and alternative hypothesis in ANOVA and t-test are exactly same and thus there will be no difference in the p-value. The Null and Alternative hypothesis for F-test are given by-

- **Null:** There is no statistically significant relationship between Y and any of the explanatory variables (i.e all regression coefficients are zero)

Alternative: Not all regression coefficients are zero

Residual analysis

The difference between the observed value of the dependent variable (Y) and the predicted value (Y , Y-hat) is called the residual (error). Each data point has one residual. Residual analysis is important to check whether the assumptions of regression models have been satisfied. It is performed to check the following-

- The functional form of regression is correctly specified
- The residuals (Y- Y_hat) are normally distributed
- The variance of the residuals is constant (homoscedasticity)
- If there are any outliers
  Regression diagnostic plots are used to check on the above.

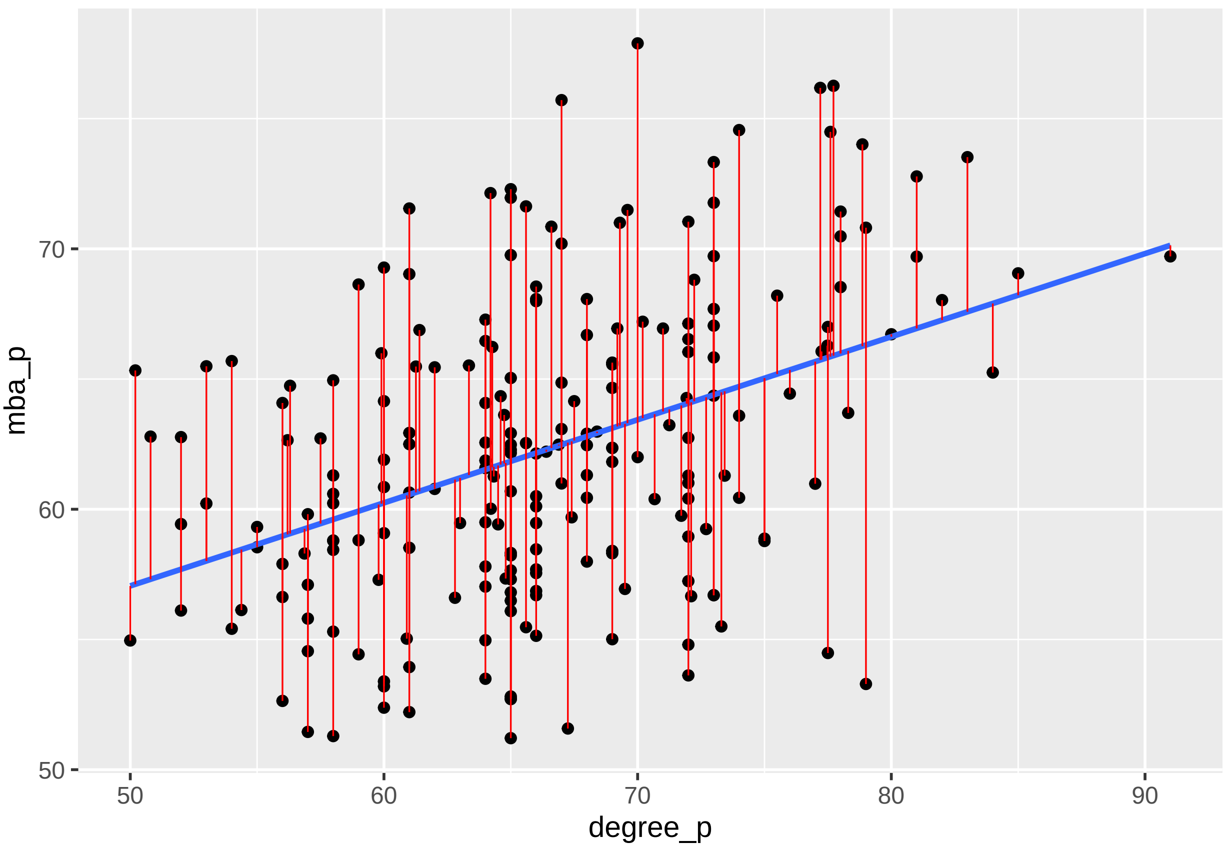
Plotting the residuals:

Each vertical red segments represents the residual error between an actual value and the corresponding predicted (i.e. fitted) value.

```
# Highlighting the Residuals
ggplot(placement_df, aes(degree_p, mba_p)) + geom_point()+
      geom_smooth(method = 'lm', se=FALSE)+
       geom_segment(aes(xend=degree_p, yend=mba_P_hat), color='red', size=0.3)
```

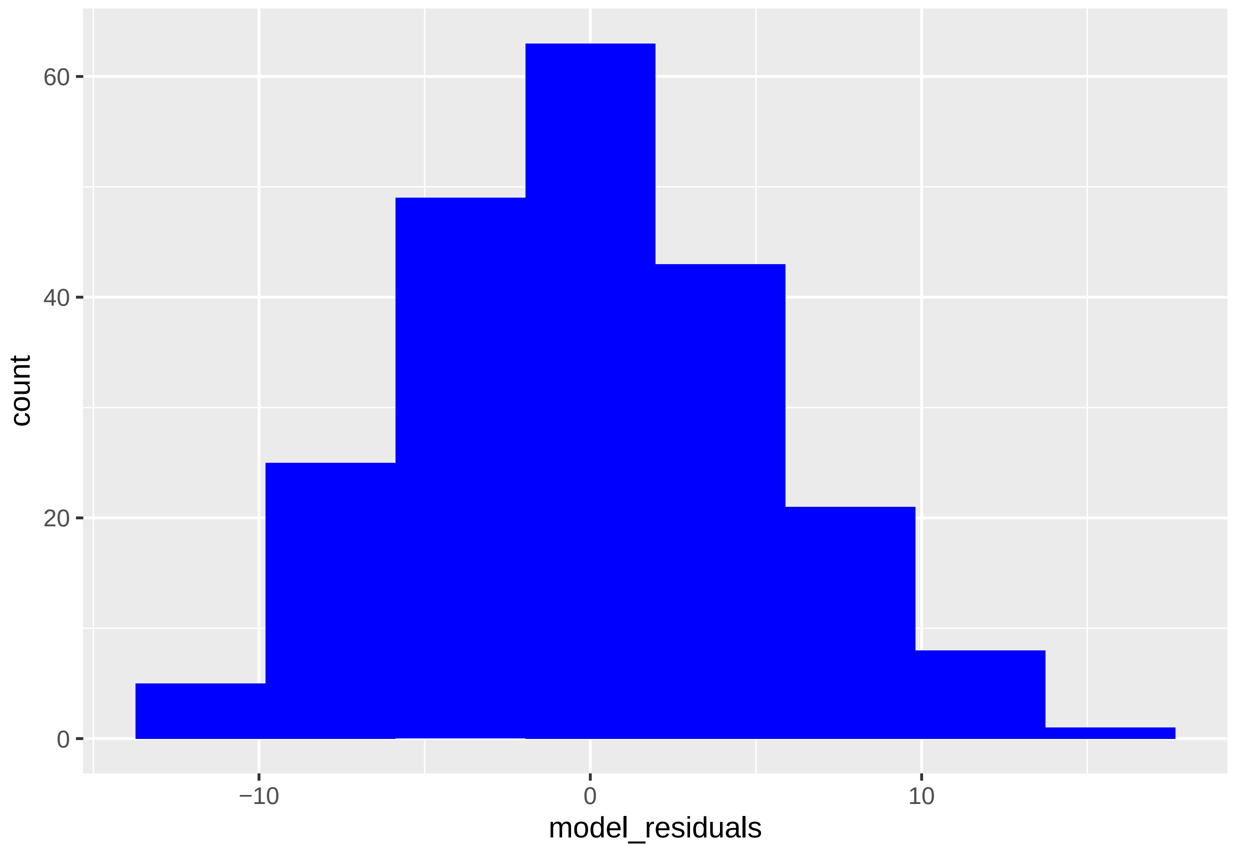
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```r
# Histogram of residuals for understanding
residual_df <- as.data.frame(model_residuals)
```

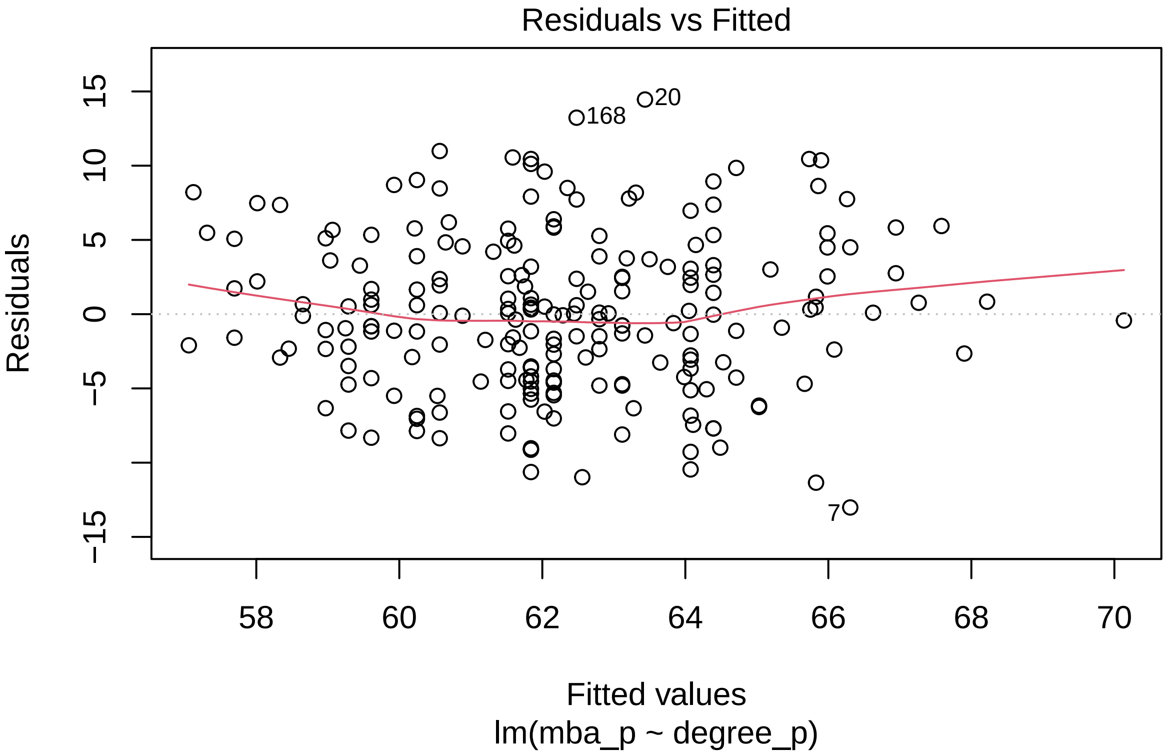
```r
#histogram of the residuals
ggplot(residual_df, aes(model_residuals))+ geom_histogram(fill='blue', bins = 8)
```

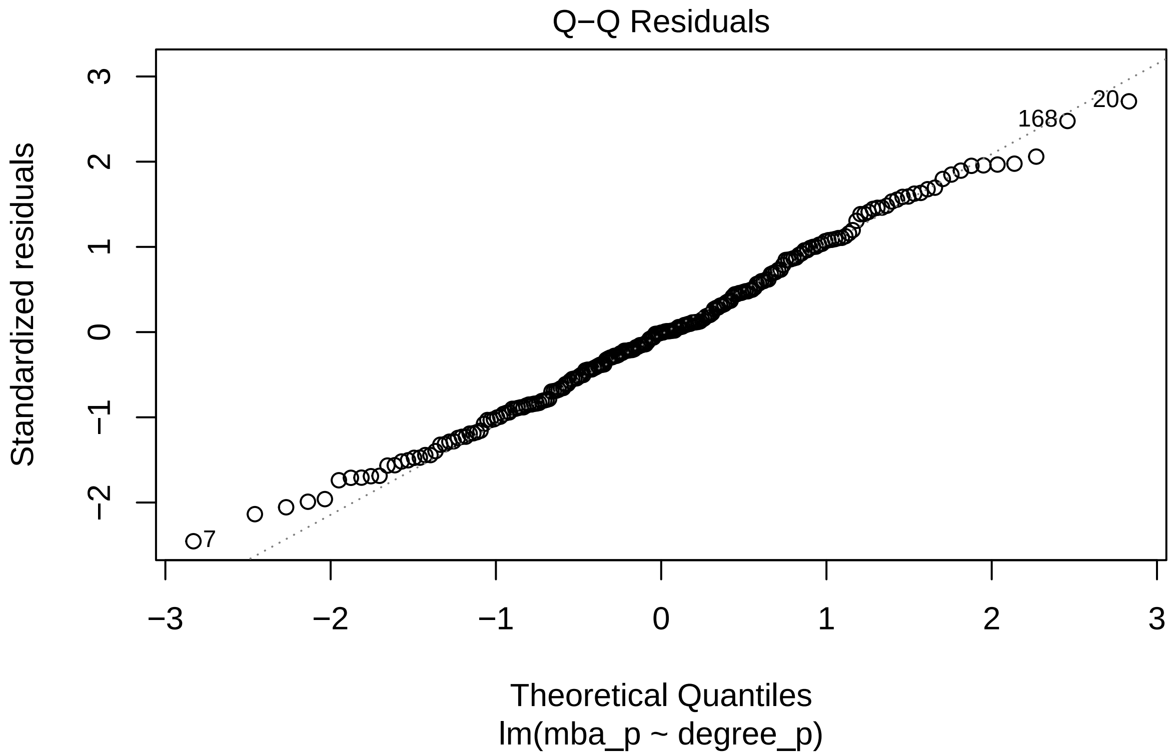


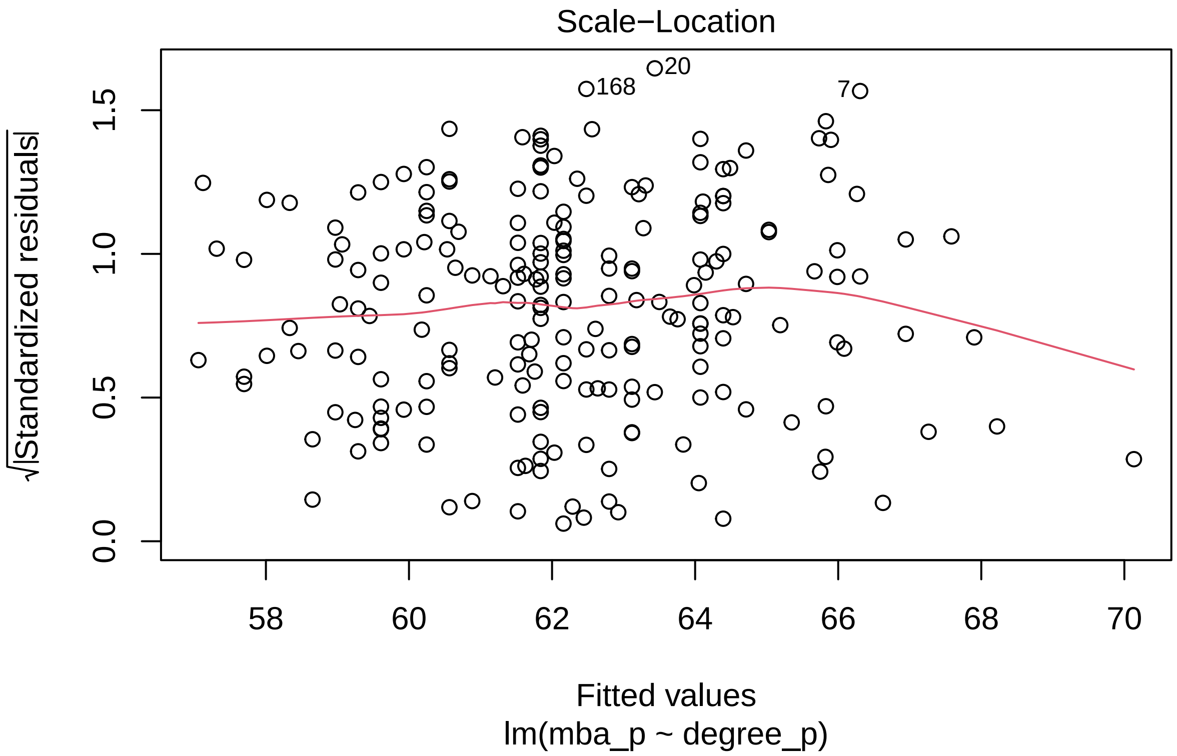
Diagnostic plots;

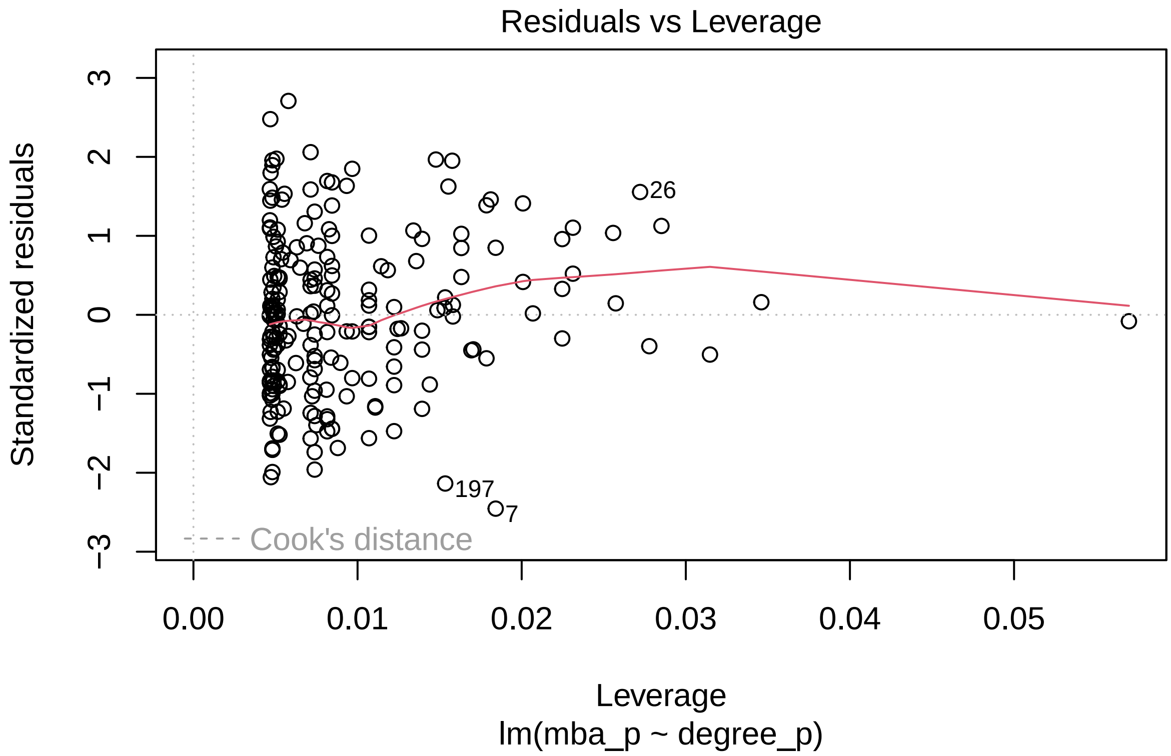
Regression diagnostics plots can be created using the R base function plot(). The diagnostic plots show residuals in four different ways.

```
plot(slr)
```


Residuals vs Fitted
Residuals
Fitted values
lm(mba_p ~ degree_p)
15
10
5
0
−5
−15
58
60
62
64
66
68
70
168
20
7


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(mba_p ~ degree_p)
−3
−2
−1
0
1
2
3
7
168
20


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
58
60
62
64
66
68
70
Fitted values
lm(mba_p ~ degree_p)
168
20
7



Note: The four plots show the top 3 most extreme data points labelled with the row numbers of the data in the data set. You might want to take a close look at them individually to check if there is anything special for the observation. The standardized residual is the residual divided by its standard deviation.

Residuals vs Fitted:

Used to check the linear relationship assumption. An approximately horizontal line (red line), without distinct patterns is an indication for a linear relationship. Any pattern in the residual plot would indicate incorrect specification of the model.
In our example, there is no pattern in the residual plot. This suggests that we can assume linear relationship between the predictor (degree_p) and the outcome variable (mba_p).
Note: If the residual plot indicates a non-linear relationship in the data, then a simple approach is to use non-linear transformations of the predictors, such as log(X), sqrt(X) and X2, in the regression model.

Multiple Linear Regression

Multiple Linear Regression (MLR) is a statistical technique for finding existence of an association relationship between a dependent variable (response variable or outcome variable) and several independent variables (explanatory variable or predictor variable). The functional form of MLR is given by-

Y = b0 + b1*X1 + b2*X2 + ... + bp*Xp + e

where, Y is the dependent variable, X1, X2 … Xp are independent variables, b0 is a constant, b1, b2 … bp are the partial regression coefficients corresponding to the explanatory variables and e is the error term (residual).

In MLR the regression coefficients are called partial regression coefficients since the relationship between an explanatory variable and the response variable is calculated after removing (partial out) the effect of all the other explanatory variables in the model.

```
#select the relevant columns
library(dplyr)
placement_mlr <- df %>% dplyr::select(ends_with('_p'), -etest_p)
```

```
(str(placement_mlr))
```

```
## 'data.frame':    215 obs. of  4 variables:
##  $ ssc_p   : num  67 79.3 65 56 85.8 ...
##  $ hsc_p   : num  91 78.3 68 52 73.6 ...
##  $ degree_p: num  58 77.5 64 52 73.3 ...
##  $ mba_p   : num  58.8 66.3 57.8 59.4 55.5 ...
```

```
## NULL
```

```
(colnames(placement_mlr))
```

```
## [1] "ssc_p"    "hsc_p"    "degree_p" "mba_p"
```

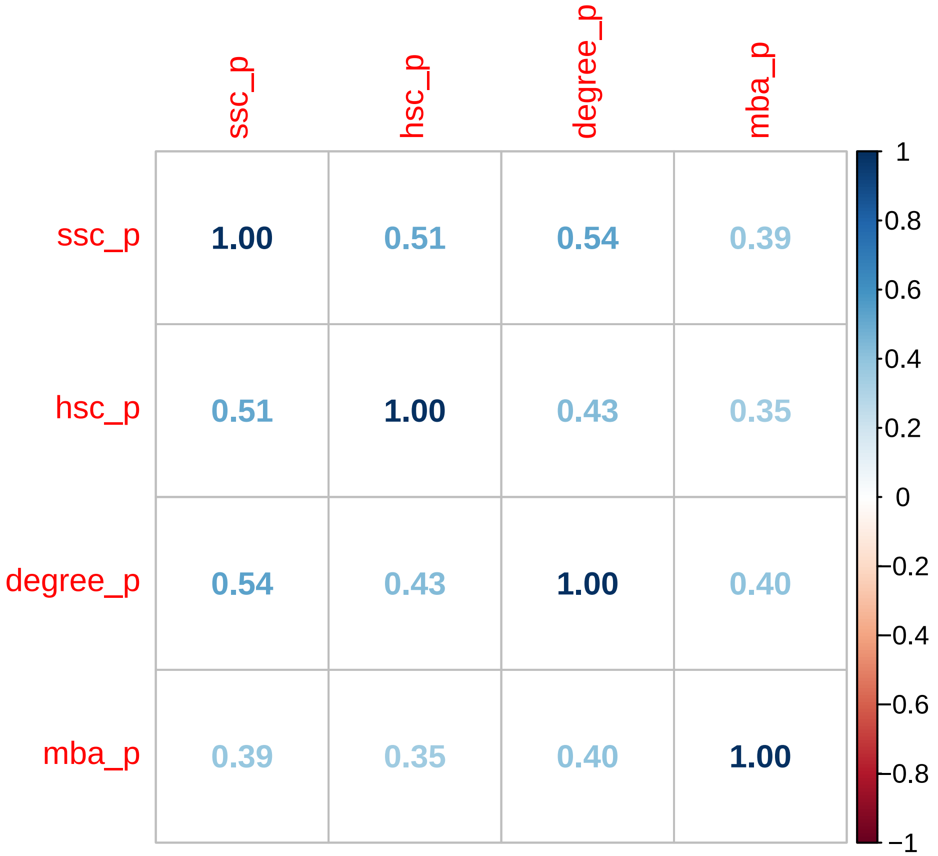
```
#correlation among numeric variables
placement_mlr %>% cor()
```

```
##              ssc_p     hsc_p  degree_p     mba_p
## ssc_p    1.0000000 0.5114721 0.5384040 0.3884776
## hsc_p    0.5114721 1.0000000 0.4342058 0.3548226
## degree_p 0.5384040 0.4342058 1.0000000 0.4023638
## mba_p    0.3884776 0.3548226 0.4023638 1.0000000
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor(placement_mlr), method = 'number')
```



```
{# {r} # #correlation vizualization # library(GGally) # ggcorr(placement_mlr) # ggpairs(placement_mlr)
```

Dividing the data into training and validation data sets

The data is randomly divided into mostly two subsets- training and validation/test. The proportion of training dataset is usually between 70% to 80%. The training data is used for developing the model and the validation data is used for model validation and selection.

```
#Train and Test Data
library(rsample)
set.seed(1001)
split <- initial_split(placement_mlr, prop = 0.7)
train <- training(split)
test <- testing(split)
```

MLR Model

The function lm() can be used to determine the partial regression coefficients of the linear model. The statistical summary of the model can be displayed using the function summary().

```
#MLR Model
mlrmodel <- lm(mba_p~.,data = placement_mlr)
mlrmodel
```

```
##
## Call:
## lm(formula = mba_p ~ ., data = placement_mlr)
##
## Coefficients:
## (Intercept)        ssc_p        hsc_p     degree_p
##    37.65289      0.09653      0.08599      0.18719
```

```
#model assessement
summary(mlrmodel)
```

```
##
## Call:
## lm(formula = mba_p ~ ., data = placement_mlr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1965  -4.0046  -0.5437   3.4835  15.5800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.65289    3.33814  11.280   <2e-16 ***
## ssc_p        0.09653    0.04172   2.314   0.0217 *
## hsc_p        0.08599    0.03878   2.217   0.0277 *
## degree_p     0.18719    0.05856   3.197   0.0016 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 5.183 on 211 degrees of freedom
## Multiple R-squared:  0.2216, Adjusted R-squared:  0.2106
## F-statistic: 20.03 on 3 and 211 DF,  p-value: 1.845e-11
```

From the output, the estimated values of the parameters are b0 = 37.65;

b1 = 0.0965; b2 = 0.086; and b3 = 0.187

Regression equation:

The regression model is given by-

- mba_p = b0 + b1*ssc_p + b2*hsc_p + b3*degree_p
- mba_p =37.65 +0.0965*ssc_p +*0.086hsc_p + 0.187*degree_p

Interpretation of the coefficients:

The partial regression coefficient provides the change in the response variable for a unit change in the explanatory variable, when all other explanatory variables are kept constant or controlled. For every one percentage increase in SSC, the MBA percent will increase by 0.0965% provided all other variables are held constant.
Since the unit of measurement of all the explanatory variables is same, we can say that the Degree percentage has a higher impact on the MBA percentage as compared to others.
Note: If the unit of measurement of the explanatory variables is different, we have to derive the standardised regression coefficients to compare the impact.

1. Validation of the overall regression model: F-Test

The research question that we are answering here is, does the regression model contain at least one predictor variable useful in predicting the response variable. Analysis of Variance (ANOVA) is used to validate the overall regression model. The null and alternative hypotheses are stated as-

```
#*H0; Null: b0 = b1 = b2 = b3 = 0
#*H1; Alternative: Not all coefficients are zero
```

In our example, it can be seen that p-value of the F-statistic is < 0.05, which is highly significant and hence we reject the null hypothesis. This means that, at least one of the predictor variables is significantly related to the response variable.

The statement in alternative hypothesis is that not all the beta's are zero, that is, some of the coefficients may be zero. This is the reason why we have to do the t-Test to check the existence of statistically significant relationship between the individual explanatory variables and the response variable. Note: We usually don't worry about the p-value for Constant. It has to do with the "intercept" of the model and seldom has any practical meaning unless it makes sense for all the independent variables to be zero simultaneously.

2. Statistical significance of individual variables in MLR: t-Test

Within a MLR model, we may want to know whether a particular independent variable is making a useful contribution to the model. That is, given the presence of the other independent variables in the model, does a particular variable help us predict or explain the dependent variable?
For a given independent variable, the t-statistic evaluates whether or not there is significant association between the independent and the dependent variable, that is, whether the beta coefficient of the independent variable is significantly different from zero.
The null and alternative hypotheses in the case of individual independent variables (Xi) and the dependent variable (Y) is stated as-

H0;There is no relationship between independent variable (Xi)and dependent variable Y (bi = 0)

H1;Alternative: There is a relationship between independent variable (Xi) and dependent variable Y (bi != 0)

When we cannot reject the null hypothesis for a particular independent variable in the above, we should say that we do not need that variable (say X1) in the model given that variables X2 and X3 will remain in the model.
In our example, we find that SSC percentage is not significant at 0.05 alpha. This means that for a fixed HSC percentage and Degree percentage, changes in the SSC percentage will not significantly affect MBA percentage.

If we remove ssc_p from the model, we will have the regression equation as below.

```r
#remove insignificant independent variable
mlrmodel1 <- lm(mba_p ~hsc_p+ degree_p,data =train)
summary(mlrmodel1)
```

```
##
## Call:
## lm(formula = mba_p ~ hsc_p + degree_p, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8316 -3.8904 -0.5384  3.5872 14.2593
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.62961    4.03469   8.831 2.86e-15 ***
## hsc_p        0.10983    0.04372   2.512   0.0131 *
## degree_p     0.29489    0.06891   4.279 3.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.314 on 147 degrees of freedom
## Multiple R-squared:  0.241,  Adjusted R-squared:  0.2307
## F-statistic: 23.34 on 2 and 147 DF,  p-value: 1.572e-09
```

The regression model is given by-
mba_p = 36.409 + 0.10017*hsc_p + 0.29018*degree_p

A good regression model should include only significant independent variables. It is not always clear what will happen when we add or remove variables from a model. Therefore, we should not consider dropping all insignificant variables at one time, but rather take a more structured approach. Automated methods like forward selection, backward elimination, stepwise regression, and best subsets methods facilitate this process. Such procedures ensure that only statistically significant variables for a given alpha are include in the model.

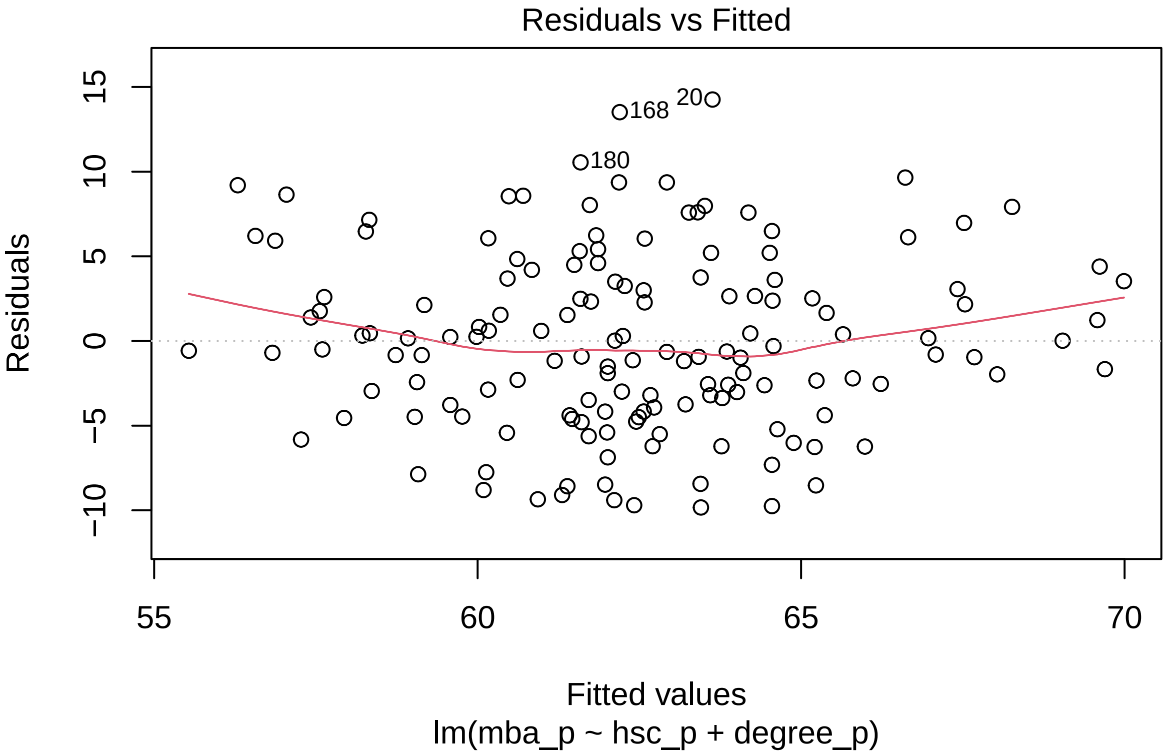
3. Coefficient of Multiple Determination (R-Square) and adjusted R-square

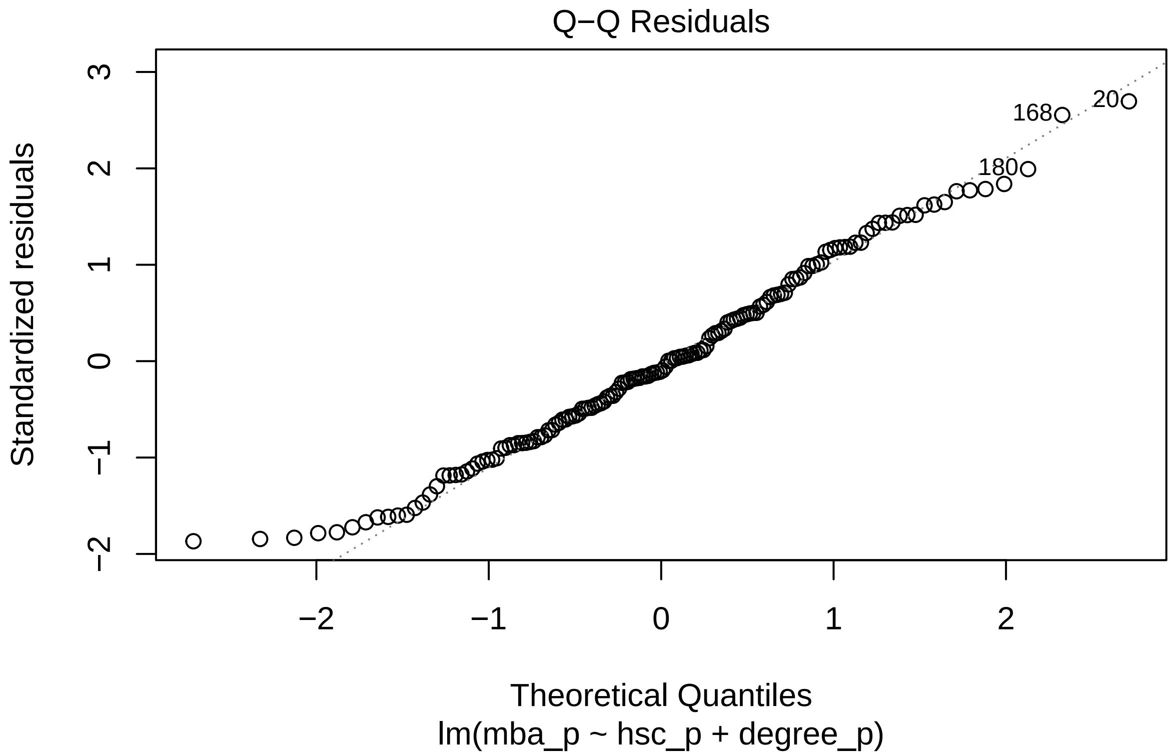
R-square measures the proportion of variation in the dependent variable explained by the model. A problem with R-square is that it will always increase when more variables are added to the model even when there is no statistically significant relationship between the independent variable and the dependent variable. A solution to this is to adjust the value of R-square by taking into account the number of predictors. The Adjusted R-square reflects both the number of independent variables and the sample size and may increase or decrease when an independent variable is added or dropped, thus giving an indication of the value of adding or removing the independent variables in the model. An increase in Adjusted R-square indicates that the model has improved. The Adjusted R-square value will always be less than or equal to R-square value. In our example, the Adjusted R-square of 0.2328 means that 23.28% of the variability in MBA percentage is explained by the model. (This is better than the R-square of approximately 16% in the SLR model)

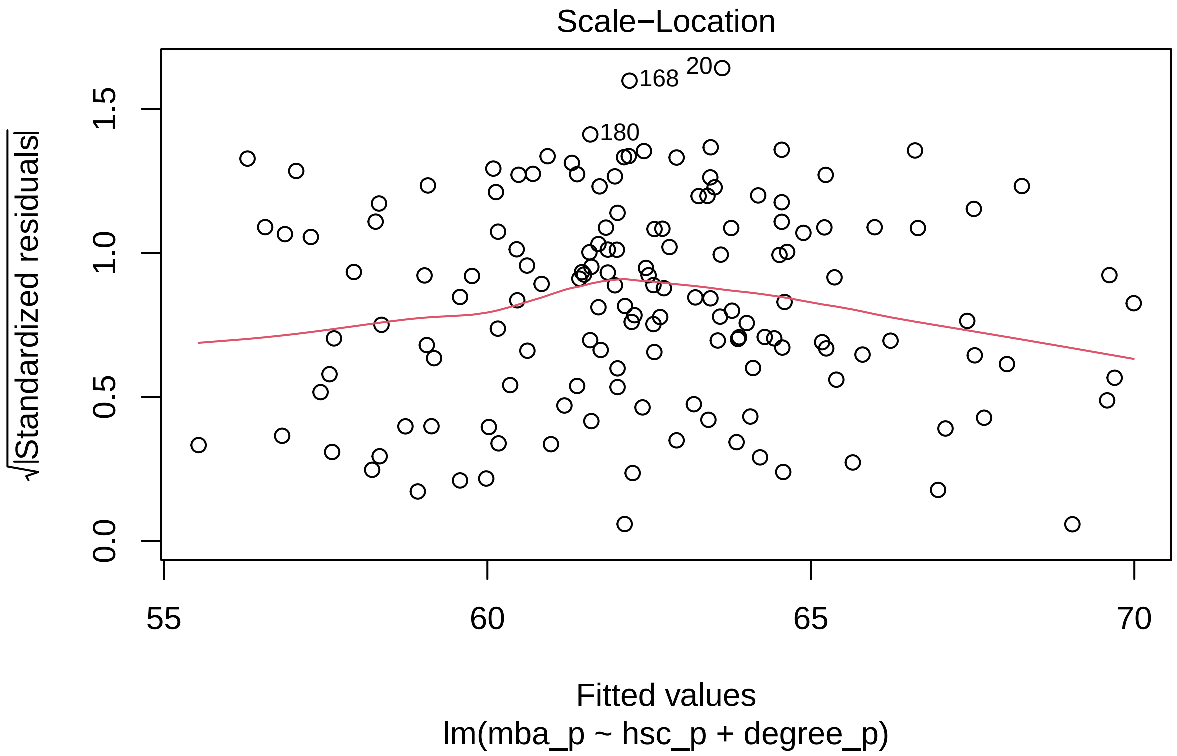
4. Residual analysis

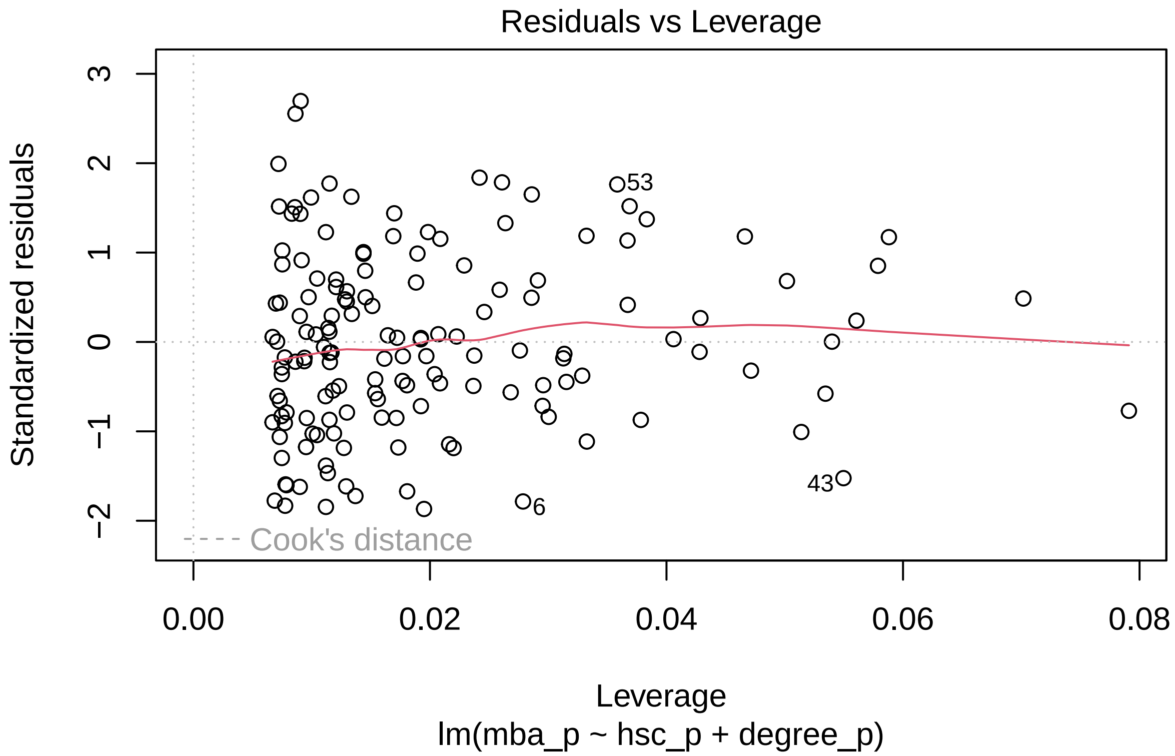
Residual analysis is important for checking assumptions about the functional form of the regression model, normal distribution of the residuals and homoscedasticity.

```
#residual analysis
plot(mlrmodel1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(mba_p ~ hsc_p + degree_p)
15
10
5
0
−5
−10
55
60
65
70
168
20
180


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(mba_p ~ hsc_p + degree_p)
−2
−1
0
1
2
3
168
20
180


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
55
60
65
70
Fitted values
lm(mba_p ~ hsc_p + degree_p)
168
20
180



In our example, based on the plots we can infer that all the assumptions are satisfied and the data don't present any outliers and influential observations in the MLR model.

5. Multi-collinearity and Variance Inflation Factor

When the data set has a large number of independent variables, it is possible that few of these independent variables may be highly correlated. Existence of high correlation between independent variables is called multi-collinearity. Presence of multi-collinearity can destabilise the MLR model. Due to presence of multi-collinearity it is possible that a statistically significant variable may be labelled as insignificant on account of inflated p-value or the sign of a regression coefficient may change. Thus, it is necessary to identify the presence of multi-collinearity and take corrective action.
Multi-collinearity can be assessed by computing a score called the variance inflation factor (VIF) for each independent variable. The smallest possible value of VIF is one (absence of multi-collinearity). As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problem. VIF greater than this requires further investigation to assess the impact of multi-collinearity. Remedies to handle multi-collinearity include using Principal Component Analysis (PCA), Ridge regression and LASSO regression.

The R function vif() [**car package**] can be used to detect multi-collinearity in a regression model. In our example, the VIF score is within limit.

```
#VIF
suppressPackageStartupMessages(library(car))
vif(mlrmodel1)
```

```
##    hsc_p degree_p
## 1.324004 1.324004
```

Making predictions on the validation dataset

After the model is built and has passed all diagnostic tests, we can apply the model on the validation dataset to predict.

```
#prediction on the test set
mbapred <-predict(mlrmodel1, test)
data.frame(test$mba_p, mbapred)
```

```
##     test.mba_p  mbapred
## 4         59.43 56.67520
## 5         55.50 65.32874
## 7         53.29 64.32966
## 9         61.29 65.53849
## 21        56.70 62.23148
## 24        63.62 61.31075
## 26        65.33 56.42997
## 29        64.15 63.93691
## 36        62.74 65.42866
## 38        55.47 63.32170
## 40        62.56 61.97121
## 47        71.63 62.88017
## 49        62.46 62.49176
## 50        56.11 55.02768
## 51        62.98 63.83986
## 52        62.65 58.91542
## 60        56.66 64.09696
## 63        59.69 62.55646
## 64        59.50 62.19088
## 65        58.78 65.76416
## 67        58.46 63.21999
## 70        68.07 63.11016
## 73        68.53 65.98982
## 76        67.00 65.29320
## 78        57.65 63.58411
## 80        67.99 61.90198
## 81        62.35 62.78665
## 82        70.20 62.30670
## 83        60.44 64.81026
## 84        66.69 64.35894
## 89        64.36 63.96620
## 90        62.36 64.21449
## 95        54.97 61.31220
## 96        62.16 63.36444
## 97        64.44 65.72954
## 101       64.95 58.99370
## 102       60.44 63.59010
## 103       61.31 62.38192
## 104       65.83 65.72355
## 109       58.31 64.98333
## 112       60.64 59.54886
## 113       53.94 60.31770
## 120       64.34 62.19208
## 124       56.70 63.63670
## 126      73.33 65.17438
## 130      68.55 64.94439
## 140      54.43 60.71643
## 141      56.94 63.24163
## 145      58.52 59.10953
## 153      65.25 67.04523
## 154      62.48 61.27759
## 155      53.20 60.24248
## 159      61.87 61.42204
## 160      60.59 58.11503
## 162      62.72 58.18725
## 166      74.56 66.05468
## 173      52.64 58.51376
## 186      71.43 66.53899
## 187      62.93 60.64721
## 191      62.50 61.32818
## 193      57.34 61.48221
## 197      54.48 65.40303
## 198      69.71 68.28569
## 199      71.96 62.48577
## 202      58.44 59.65271
```

Cross-validation

How do we know that an estimated regression model is generalizable beyond the sample data used to fit it? One way is to partition the sample data into a training (or model-building) set, which we can use to develop the model, and a validation (or prediction) set, which is used to evaluate the predictive ability of the model. This is called cross-validation. Cross-validation refers to a set of methods for measuring the performance of a given predictive model on new test data sets.
The different cross-validation methods for assessing model performance are

- The validation set approach
- Leave-one-out cross-validation
- k-fold cross-validation
- Repeated k-fold cross-validation

Validation set approach involves randomly dividing the available set of observations into two parts, a training set and a validation set (hold-out set). The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set. The resulting validation set error rate – typically assessed using Root Mean Square Error (RMSE) provides an estimate of the test error rate.

- Sum of Square Errors (SSE) = sum((Yi- Yi(hat)) 2
- Mean Square Error (MSE) = SSE/n
- Root Mean Square Error (RMSE) = sqrt(MSE)

(Note: R calls this as Residual Standard Error (RSE). RSE, also known as the model sigma, is a variant of the RMSE adjusted for the number of predictors in the model. Instead of n in the denominator in the MSE formula, it will be the degrees of freedom which is n-p-1 where p is the number of predictors. The lower the

RSE, the better the model. In practice, the difference between RMSE and RSE is very small, particularly for large multivariate data.)

If the predictions obtained using the model are very close to the actual values in the validation set, then RMSE will be small, and we can conclude that the model fits the validation set very well. The lower the RMSE, the better the model. Dividing the RMSE by the average value of the outcome variable will give us the prediction error rate, which should be as small as possible.

```
#cross validation
mse <-mean((test$mba_p-mbapred)^2)
rmse <- sqrt(mse)
rmse
```

```
## [1] 5.108589
```

```
prederror <- rmse/mean(test$mba_p)
prederror
```

```
## [1] 0.0824674
```