Author: Saurabh Kumar Singh

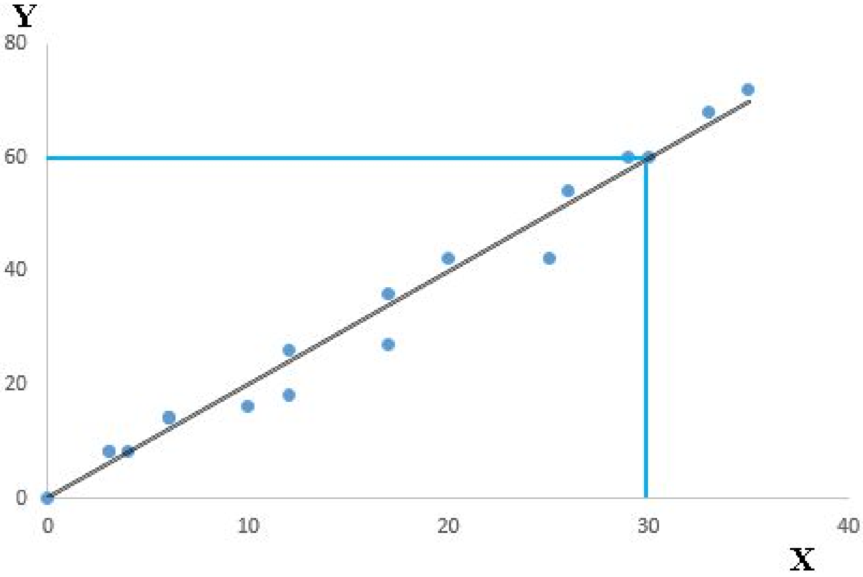
What is Linear Regression?

Linear Regression is used for predictive analysis. It is a technique which explains the degree of relationship between two or more variables (multiple regression, in that case) using a best fit line / plane. Simple Linear Regression is used when we have, one independent variable and one dependent variable.

Regression technique tries to fit a single line through a scatter plot (see below). The simplest form of regression with one dependent and one independent variable is defined by the formula:

$$Y = aX + b$$

Let’s understand this equation using the scatter plot below:



Above, you can see that a black line passes through the data points. Now, you carefully notice that this line intersects the data points at coordinates (0,0), (4,8) and (30,60). Here’s a question. Find the equation that describe this line? Your answer should be:

$$Y= a * X + b$$

Now, find the value of a and b?

With out going in its working, the outcome after solving these equations is:

a = 2, b = 0

Hence, our regression equation becomes: Y= 2*X + 0 i.e. Y= 2*X

Here, Slope = 8/4 =2 or 60/30 =2 and Intercept = 0 (as Y =0 when x is 0). So, equation would be

$$Y = 2*X + 0$$

This equation is known as linear regression equation, where Y is target variable, X is input variable. 'a' is known as slope and 'b' as intercept. It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on input variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a *X + b.

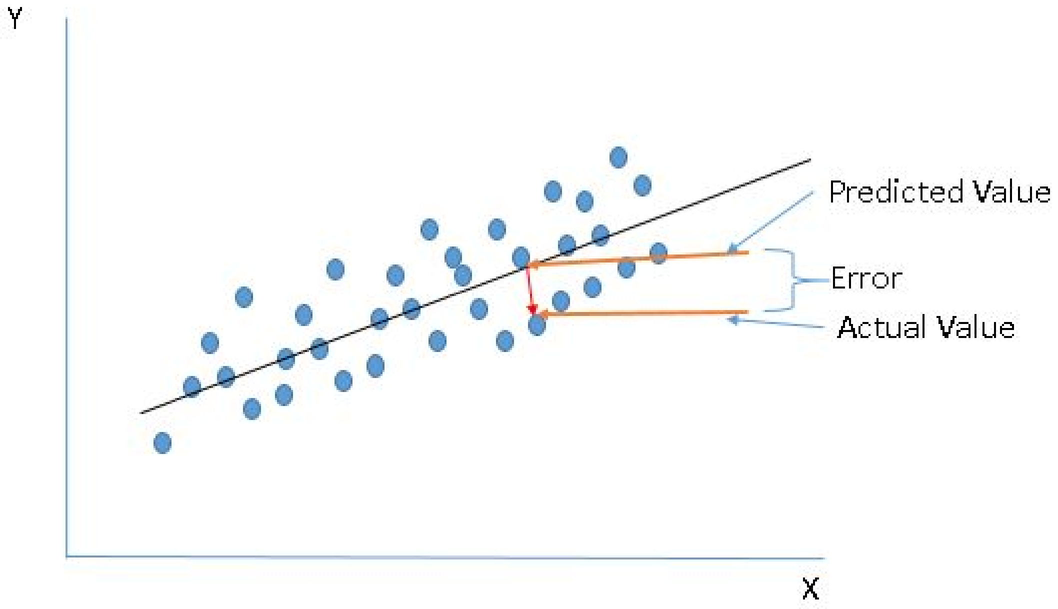
Now, you might think that in above example, there can be multiple regression lines those can pass through the data points. So, how to choose the best fit line or value of co-efficients a and b.

Let's look at the methods to find the best fit line.

How to find the best regression line?

We discussed above that regression line establishes a relationship between independent and dependent variable(s). A line which can explain the relationship better is said to be best fit line.

In other words, the best fit line tends to return most accurate value of Y based on X i.e. causing a minimum difference between actual and predicted value of Y (lower prediction error). Make sure you understand the image below.



Here are some methods which check for error:

- Sum of all errors (∑error)
- Sum of absolute value of all errors (∑|error|)
- Sum of square of all errors (∑error^2)

Let’s evaluate performance of above discussed methods using an example. Below I have plotted three lines (y=2.3x+4, y=1.8x+3.5 and y=2x+8) to find the relationship between y and x.

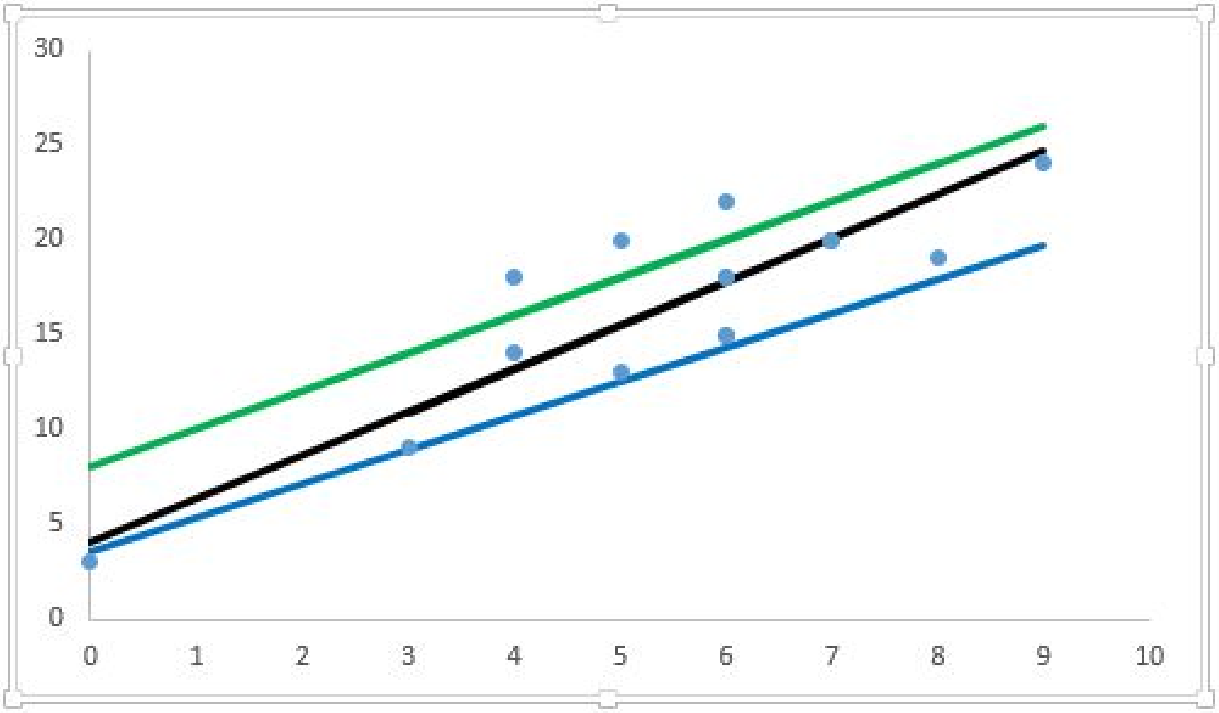
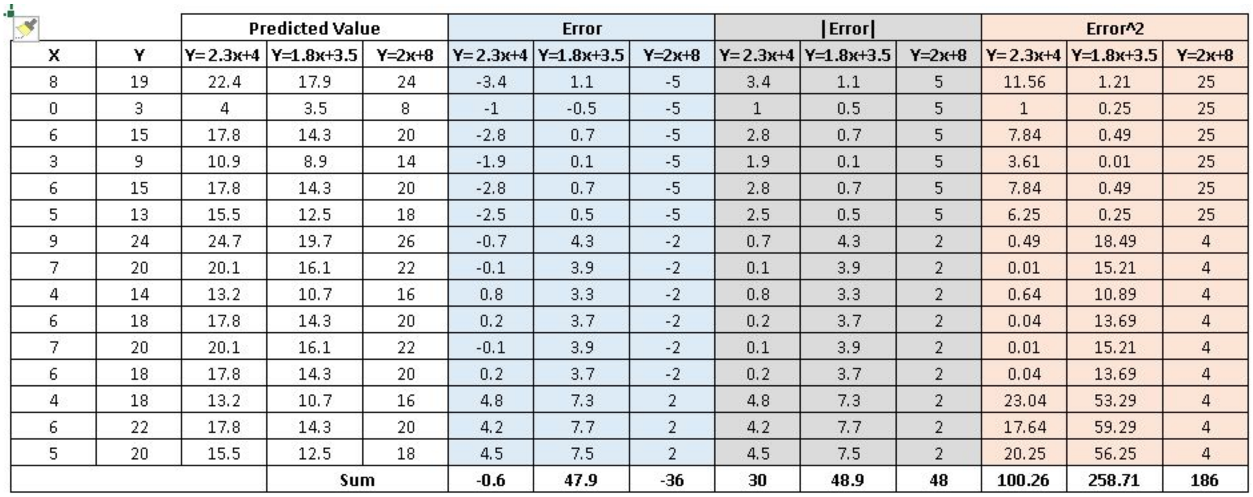


Table shown below calculates the error value of each data point and the total error value (E) using the three methods discussed above:



		Predicted Value			Error			|Error|			Error^2		
X	Y	Y=2.3x+4	Y=1.8x+3.5	Y=2x+8	Y=2.3x+4	Y=1.8x+3.5	Y=2x+8	Y=2.3x+4	Y=1.8x+3.5	Y=2x+8	Y=2.3x+4	Y=1.8x+3.5	Y=2x+8
8	19	22.4	17.9	24	-3.4	1.1	-5	3.4	1.1	5	11.56	1.21	25
0	3	4	3.5	8	-1	-0.5	-5	1	0.5	5	1	0.25	25
6	15	17.8	14.3	20	-2.8	0.7	-5	2.8	0.7	5	7.84	0.49	25
3	9	10.9	8.9	14	-1.9	0.1	-5	1.9	0.1	5	3.61	0.01	25
6	15	17.8	14.3	20	-2.8	0.7	-5	2.8	0.7	5	7.84	0.49	25
5	13	15.5	12.5	18	-2.5	0.5	-5	2.5	0.5	5	6.25	0.25	25
9	24	24.7	19.7	26	-0.7	4.3	-2	0.7	4.3	2	0.49	18.49	4
7	20	20.1	16.1	22	-0.1	3.9	-2	0.1	3.9	2	0.01	15.21	4
4	14	13.2	10.7	16	0.8	3.3	-2	0.8	3.3	2	0.64	10.89	4
6	18	17.8	14.3	20	0.2	3.7	-2	0.2	3.7	2	0.04	13.69	4
7	20	20.1	16.1	22	-0.1	3.9	-2	0.1	3.9	2	0.01	15.21	4
6	18	17.8	14.3	20	0.2	3.7	-2	0.2	3.7	2	0.04	13.69	4
4	18	13.2	10.7	16	4.8	7.3	2	4.8	7.3	2	23.04	53.29	4
6	22	17.8	14.3	20	4.2	7.7	2	4.2	7.7	2	17.64	59.29	4
5	20	15.5	12.5	18	4.5	7.5	2	4.5	7.5	2	20.25	56.25	4
			Sum		**-0.6**	**47.9**	**-36**	**30**	**48.9**	**48**	**100.26**	**258.71**	**186**

After looking at the table, the following inferences can be generated:

- Sum of all errors (∑error): Using this method leads to cancellation of positive and negative errors, which certainly isn't our motive. Hence, it is not the right method.

- The other two methods perform well but, if you notice, ∑error^2, we penalize the error value much more compared to ∑|error|. You can see that two equations has almost similar value for ∑|error| whereas in case of ∑error^2 there is significant difference.

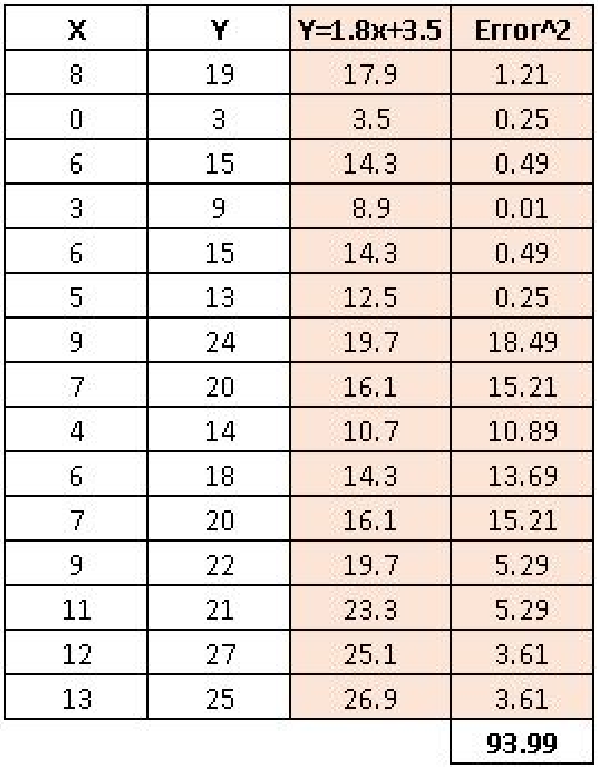
Therefore, we can say that these coefficients a and b are derived based on minimizing the sum of squared difference of distance between data points and regression line.

There are two common algorithms to find the right coefficients for minimum sum of squared errors, first one is Ordinary Least Sqaure (OLS, used in python library sklearn) and other one is gradient descent.

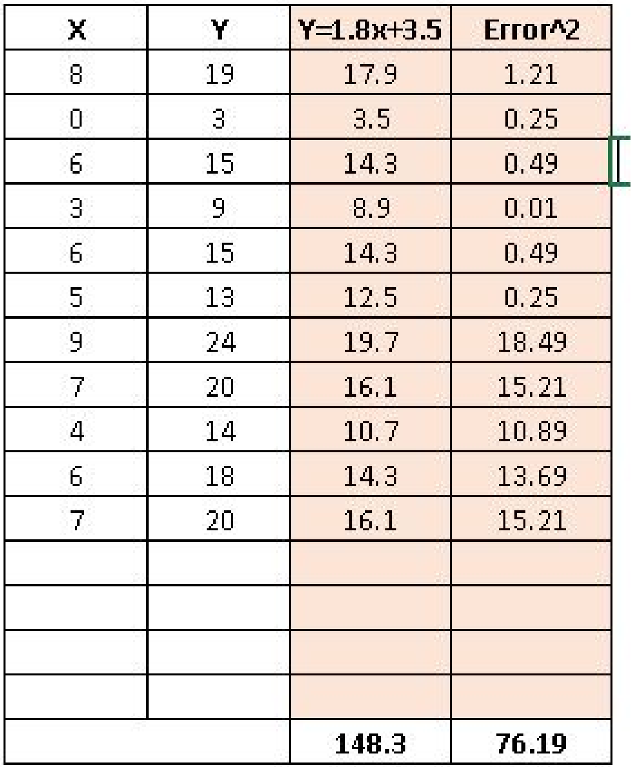
What are the performance evaluation metrics in Regression?

As discussed above, to evaluate the performance of regression line, we should look at the minimum sum of squared errors (SSE). It works well but when it has one concern!

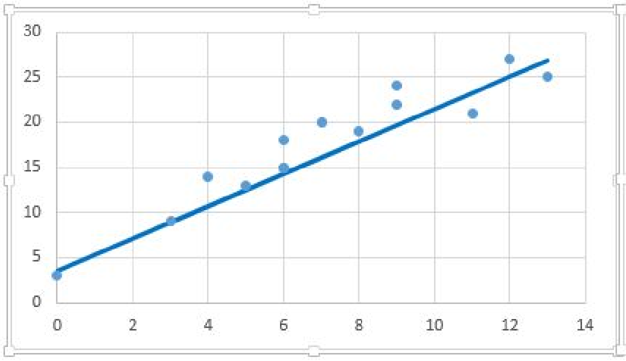
Let’s understand it using the table shown below:

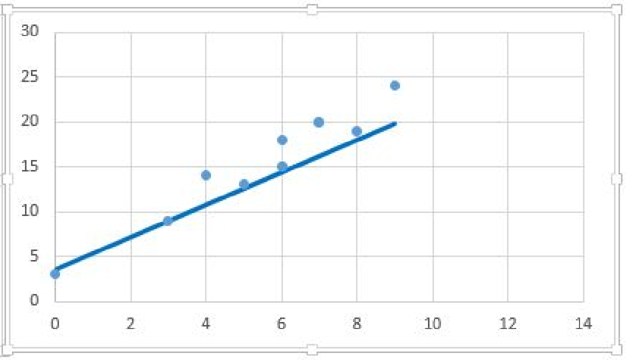


X	Y	Y=1.8x+3.5	Error^2
8	19	17.9	1.21
0	3	3.5	0.25
6	15	14.3	0.49
3	9	8.9	0.01
6	15	14.3	0.49
5	13	12.5	0.25
9	24	19.7	18.49
7	20	16.1	15.21
4	14	10.7	10.89
6	18	14.3	13.69
7	20	16.1	15.21
9	22	19.7	5.29
11	21	23.3	5.29
12	27	25.1	3.61
13	25	26.9	3.61
			93.99



X	Y	Y=1.8x+3.5	Error^2
8	19	17.9	1.21
0	3	3.5	0.25
6	15	14.3	0.49
3	9	8.9	0.01
6	15	14.3	0.49
5	13	12.5	0.25
9	24	19.7	18.49
7	20	16.1	15.21
4	14	10.7	10.89
6	18	14.3	13.69
7	20	16.1	15.21
		148.3	**76.19**





Above you can see, we've removed 4 data points in right table and therefore the SSE has reduced (with same regression line). Further, if you look at the scatter plot, removed data points have almost similar relationship between x and y. It means that SSE is highly sensitive to number of data points.

Other metric to evaluate the performance of linear regression is **R-square** and most common metric to judge the performance of regression models. R² measures, "How much the change in output variable (y) is explained by the change in input variable(x).

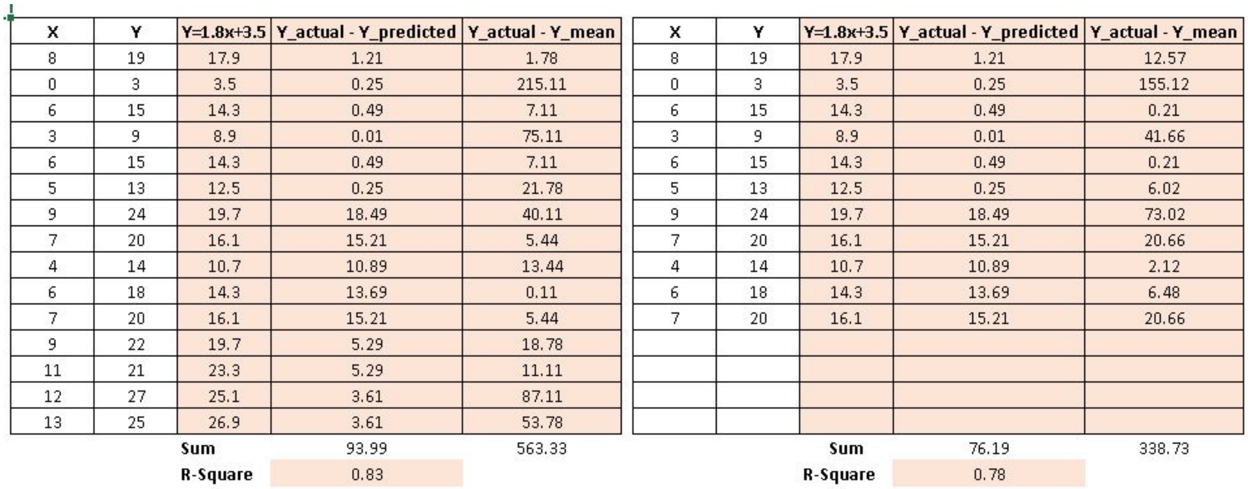
$$\text{R-Square} = 1 - \frac{\sum(Y\_actual - Y\_predicted)^2}{\sum(Y\_actual - Y\_mean)^2}$$

R-squared is always between 0 and 1:

- 0 indicates that the model explains NIL variability in the response data around its mean.
- 1 indicates that the model explains full variability in the response data around its mean.

In general, higher the R², more robust will be the model. However, there are important conditions for this guideline that I'll talk about in my future posts..

Let's take the above example again and calculate the value of R-square.



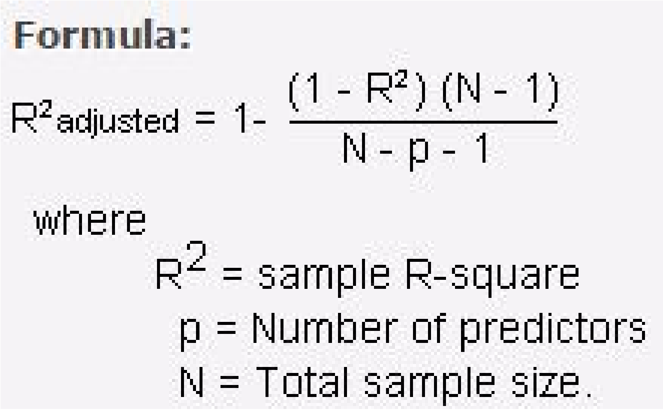
X	Y	Y=1.8x+3.5	Y_actual - Y_predicted	Y_actual - Y_mean
8	19	17.9	1.21	1.78
0	3	3.5	0.25	215.11
6	15	14.3	0.49	7.11
3	9	8.9	0.01	75.11
6	15	14.3	0.49	7.11
5	13	12.5	0.25	21.78
9	24	19.7	18.49	40.11
7	20	16.1	15.21	5.44
4	14	10.7	10.89	13.44
6	18	14.3	13.69	0.11
7	20	16.1	15.21	5.44
9	22	19.7	5.29	18.78
11	21	23.3	5.29	11.11
12	27	25.1	3.61	87.11
13	25	26.9	3.61	53.78
		Sum	93.99	563.33
		R-Square	0.83	

X	Y	Y=1.8x+3.5	Y_actual - Y_predicted	Y_actual - Y_mean
8	19	17.9	1.21	12.57
0	3	3.5	0.25	155.12
6	15	14.3	0.49	0.21
3	9	8.9	0.01	41.66
6	15	14.3	0.49	0.21
5	13	12.5	0.25	6.02
9	24	19.7	18.49	73.02
7	20	16.1	15.21	20.66
4	14	10.7	10.89	2.12
6	18	14.3	13.69	6.48
7	20	16.1	15.21	20.66
		Sum	76.19	338.73
		R-Square	0.78	

As you can see, R² has less variation in score compare to SSE.

One disadvantage of R-squared is that it can only increase as predictors are added to the regression model. This increase is artificial when predictors are not actually improving the model's fit. To cure this, we use "Adjusted R-squared".

Adjusted R-squared is nothing but the change of R-square that adjusts the number of terms in a model. Adjusted R square calculates the proportion of the variation in the dependent variable accounted by the explanatory variables. It incorporates the model's degrees of freedom. Adjusted R-squared will decrease as predictors are added if the increase in model fit does not make up for the loss of degrees of freedom. Likewise, it will increase as predictors are added if the increase in model fit is worthwhile. Adjusted R-squared should always be used with models with more than one predictor variable. It is interpreted as the proportion of total variance that is explained by the model.



Formula:

$$R^2_{\text{adjusted}} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

where

R^2 = sample R-square

p = Number of predictors

N = Total sample size.

What is Multi-Variate Regression?

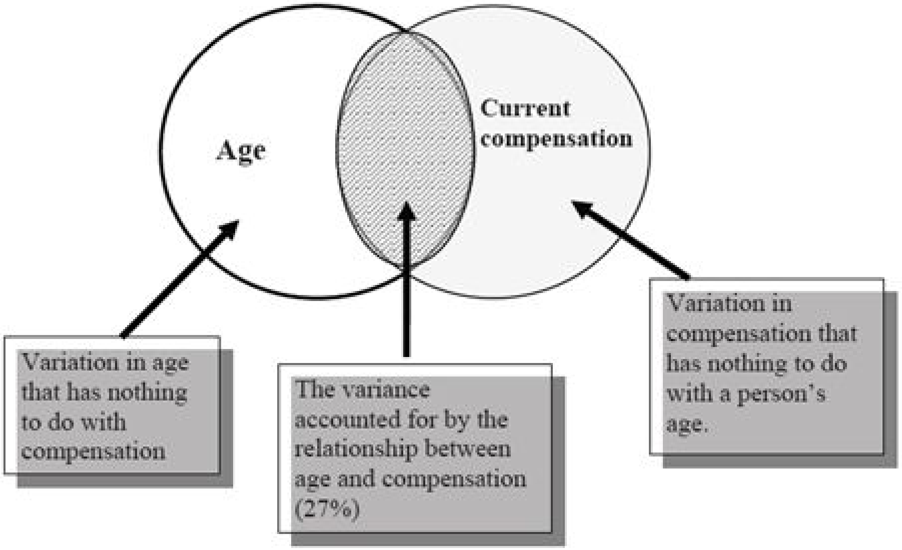
Let’s now examine the process to deal with **multiple independent variables** related to a dependent variable.

Once you have identified the level of significance between independent variables(IV) and dependent variables(DV), use these significant IVs to make more powerful and accurate predictions. This technique is known as “Multi-variate Regression”.

Let’s take an example here to understand this concept further.

We know that, compensation of a person depends on his age i.e. the older one gets, the higher he/she earns as compared to previous year. You build a simple regression model to explain this effect of age on a person’s compensation . You obtain R_2 of 27%. What does this mean?

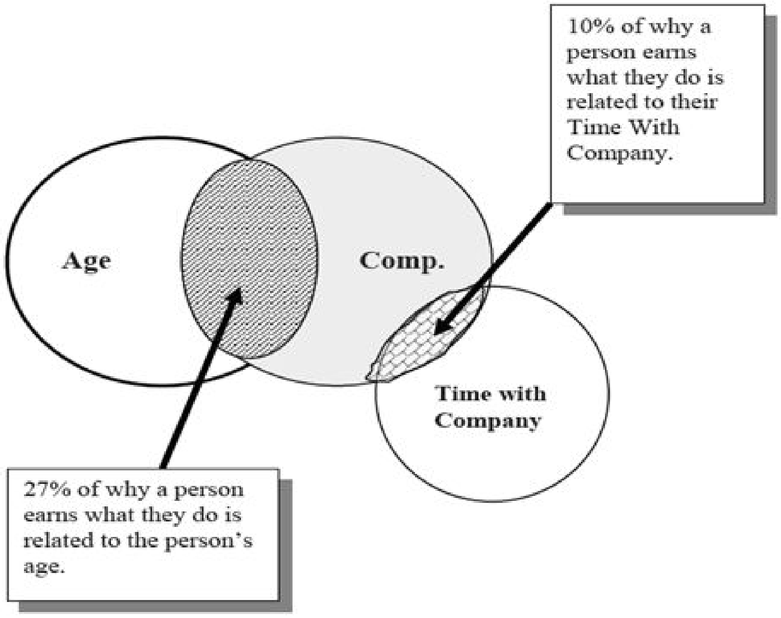
Let’s try to think over it graphically.


Age
Current compensation
Variation in age that has nothing to do with compensation
The variance accounted for by the relationship between age and compensation (27%)
Variation in compensation that has nothing to do with a person's age.

In this example, R² as 27%, says, only 27% of variance in compensation is explained by Age. In other words, if you know a person's age, you'll have 27% information to make an accurate prediction about their compensation.

Now, let's take an additional variable as 'time spent with the company' to determine the current compensation. By this, R2 value increases to 37%. How do we interpret this value now?

Let's understand this graphically once again:



10% of why a person earns what they do is related to their Time With Company.

27% of why a person earns what they do is related to the person's age.

Notice that a person's time with company holds only 10% responsible for his/her earning by profession. In other words, by adding this variable to our study, we improved our understanding of their compensation from 27% to 37%.

Therefore, we learnt, by using two variables rather than one, improved the ability to make accurate predictions about a person's salary.

Things get much more complicated when your multiple independent variables are related to with each other. This phenomenon is known as Multicollinearity. This is undesirable. To avoid such situation, it is advisable to look for Variance Inflation Factor (VIF). For no multicollinearity, VIF should be (VIF < 2). In case of high VIF, look for correlation table to find highly correlated variables and drop one of correlated ones.

Along with multi-collinearity, regression suffers from Autocorrelation, Heteroskedasticity.

In an multiple regression model, we try to predict

$$Y = a + b_1X_1 + b_2X_2 + \ldots.. + b_kX_k$$

Here, b1, b2, b3 …bk are slopes for each independent variables X1, X2, X3….Xk and a is intercept.

Example: Net worth = a+ b1 (Age) +b2 (Time with company)

How to implement regression in Python and R?

Linear regression has commonly known implementations in R packages and Python scikit-learn. Let’s look at the code of loading linear regression model in R and Python below:

Python Code

```
#Import Library

#Import other necessary libraries like pandas, numpy

from sklearn import linear_model

#Load Train and Test datasets

#Identify feature and response variable(s) and values must be numeric and numpy arrays

x_train=input_variables_values_training_datasets

y_train=target_variables_values_training_datasets

x_test=input_variables_values_test_datasets

# Create linear regression object

linear = linear_model.LinearRegression()

# Train the model using the training sets and check score

linear.fit(x_train, y_train)

linear.score(x_train, y_train)

#Equation coefficient and Intercept

print('Coefficient: \n', linear.coef_)

print('Intercept: \n', linear.intercept_)

#Predict Output
```

```
predicted= linear.predict(x_test)
```

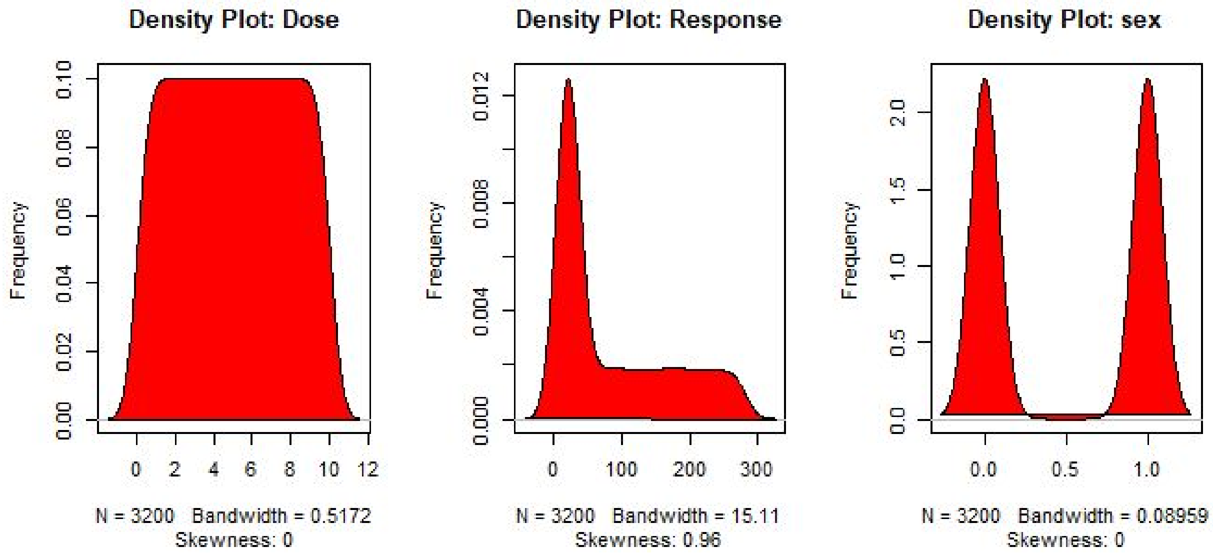
R Code

```
#Load Train and Test datasets

#Identify feature and response variable(s) and values must be numeric

x_train <- input_variables_values_training_datasets

y_train <- target_variables_values_training_datasets

x_test <- input_variables_values_test_datasets

x <- cbind(x_train,y_train)

# Train the model using the training sets and check score

linear <- lm(y_train ~ ., data = x)

summary(linear)

#Predict Output

predicted= predict(linear,x_test)
```

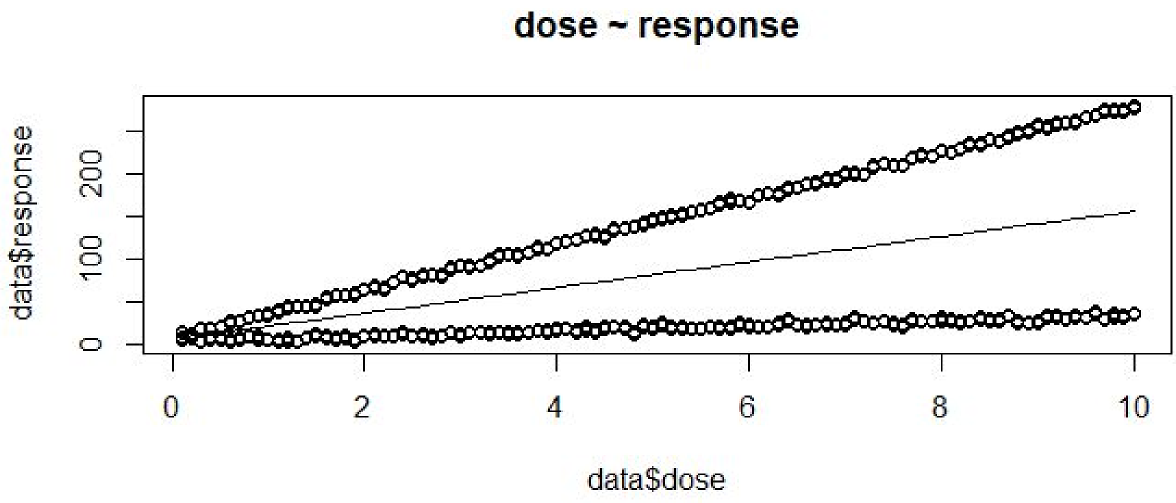
Implementation in R

```
>data=read.csv("C:/Users/imsau/Desktop/6th Sem/ML/ML_lab/Lab3(6-Feb)/drug2.csv")

> summary(data)
      sex          dose           response
 Min.   :0.0   Min.   : 0.100   Min.   :  1.92
 1st Qu.:0.0   1st Qu.: 2.575   1st Qu.: 19.53
 Median :0.5   Median : 5.050   Median : 32.37
 Mean   :0.5   Mean   : 5.050   Mean   : 83.01
 3rd Qu.:1.0   3rd Qu.: 7.525   3rd Qu.:147.09
 Max.   :1.0   Max.   :10.000   Max.   :280.73

> head(data)
  sex dose response
1   1  0.1    13.75
2   1  0.2    12.90
3   1  0.3    19.26
4   1  0.4    20.34
5   1  0.5    19.97
6   1  0.6    26.80

> cor(data$dose, data$response)
[1] 0.5136214
> cor(data$sex, data$response)
[1] 0.7516308
```

```
> par(mfrow=c(1, 3))  # divide graph area in 2 columns
> plot(density(data$dose), main="Density Plot: Dose", ylab="Frequency",
sub=paste("Skewness:", round(e1071::skewness(data$dose), 2)))  # density plot for 'speed'
 polygon(density(data$dose), col="red")
> plot(density(data$response), main="Density Plot: Response", ylab="Frequency",
sub=paste("Skewness:", round(e1071::skewness(data$response), 2)))  # density plot for 'dist'
 polygon(density(data$response), col="red")
> plot(density(data$sex), main="Density Plot: sex", ylab="Frequency", sub=paste("Skewness:",
round(e1071::skewness(data$sex), 2)))  # density plot for 'dist'
 polygon(density(data$sex), col="red")
```



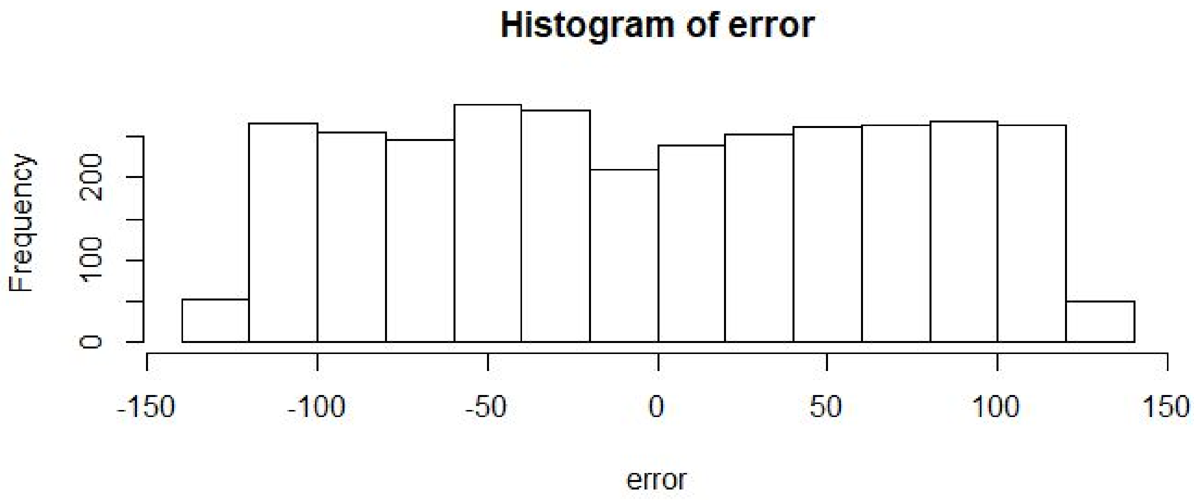
```
>scatter.smooth(x=data$dose, y=data$response, main="dose ~ response")
```



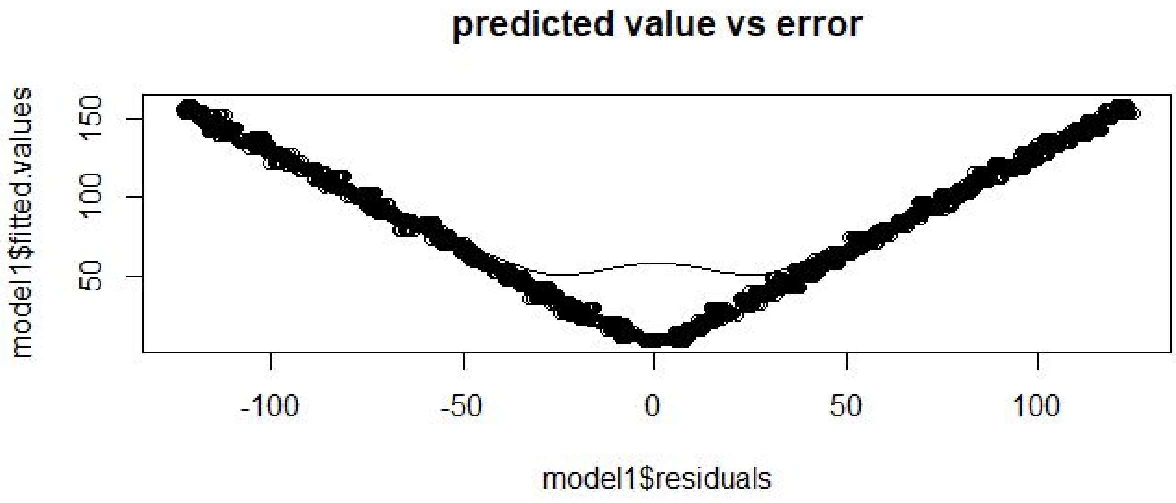
```
model1 = lm(data$response~data$dose)
> summary(model1)

Call:
lm(formula = data$response ~ data$dose)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-123.514  -62.764    0.401   63.669  124.707

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.2534     2.5778   2.814  0.00493 **
data$dose    15.0020     0.4432  33.852  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 72.36 on 3198 degrees of freedom
Multiple R-squared:  0.2638,  Adjusted R-squared:  0.2636
F-statistic:  1146 on 1 and 3198 DF,  p-value: < 2.2e-16
```

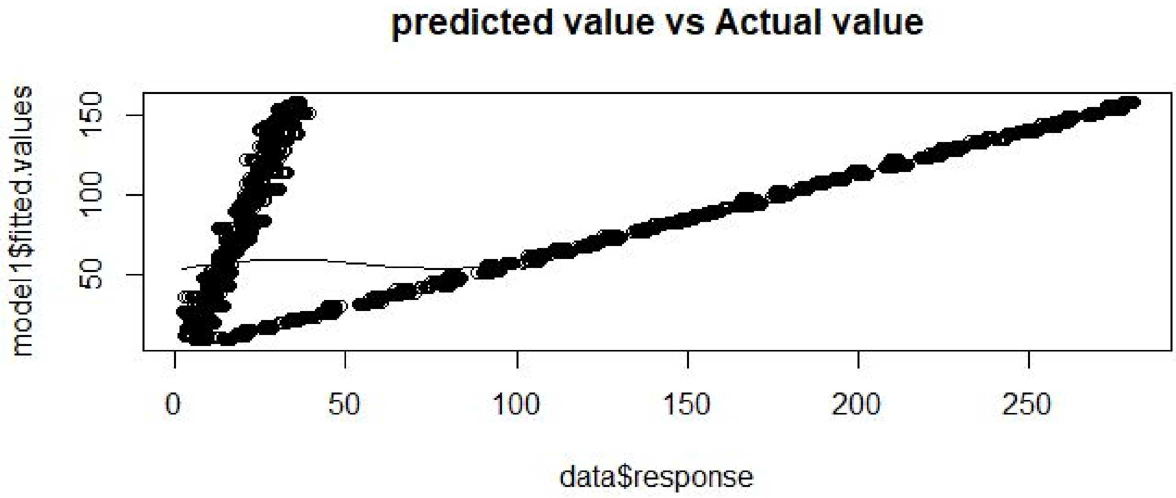
```
>error = residuals(model1)
> hist(error)
```



```
>scatter.smooth(x=model1$residuals, y=model1$fitted.values, main="predicted value vs error")
```



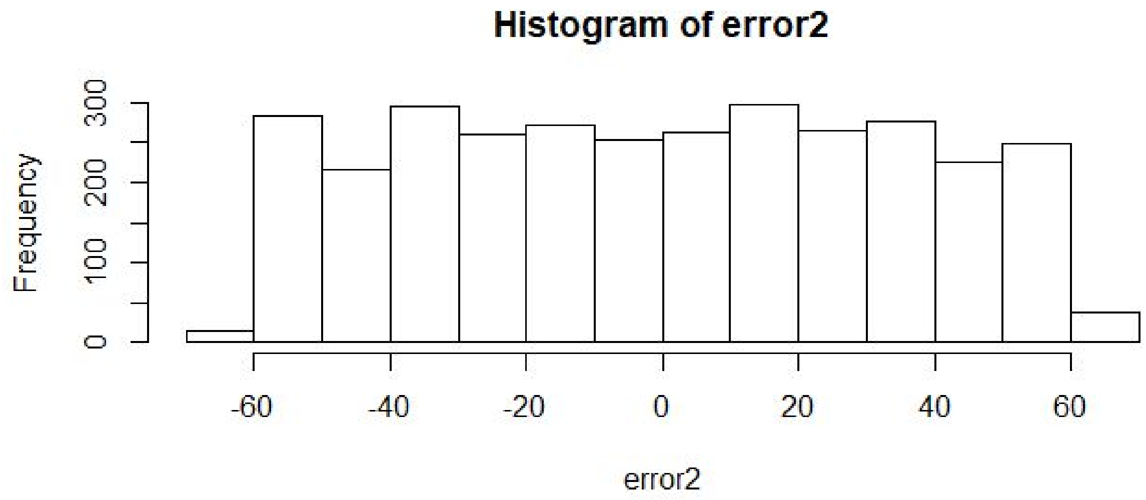
```
> scatter.smooth(x=data$response, y=model1$fitted.values, main="predicted value vs Actual
value")
```



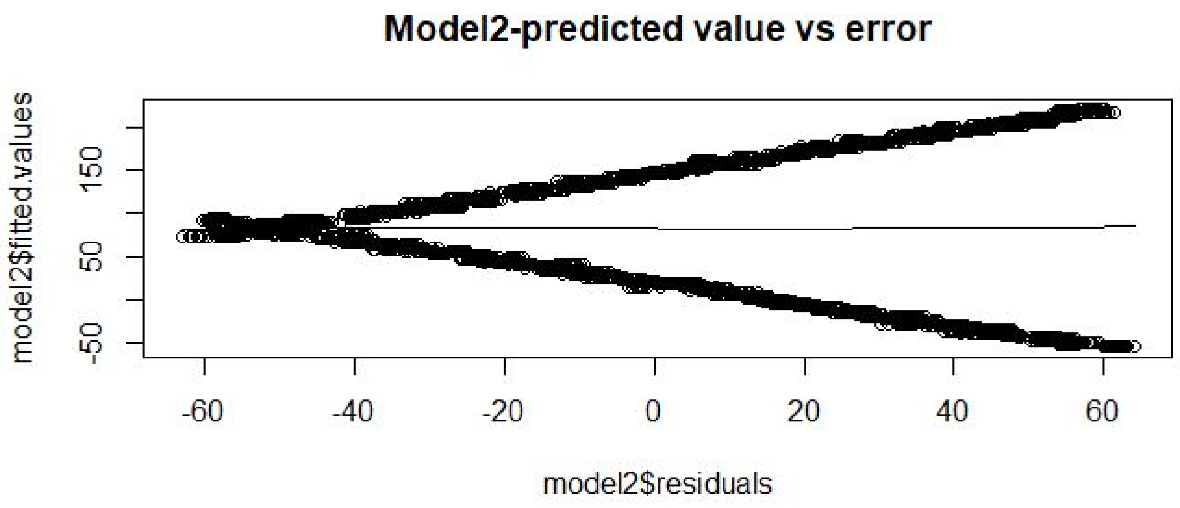
```
> model2 = lm(data$response~data$dose+data$sex)
> summary(model2)

Call:
lm(formula = data$response ~ data$dose + data$sex)

Residuals:
    Min      1Q  Median      3Q     Max 
-62.986 -30.350   0.306  29.360  64.009 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -56.1189     1.3881  -40.43   <2e-16 ***
data$dose    15.0020     0.2138   70.18   <2e-16 ***
data$sex    126.7445     1.2341  102.70   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.91 on 3197 degrees of freedom
Multiple R-squared:  0.8288,  Adjusted R-squared:  0.8286
F-statistic:  7736 on 2 and 3197 DF,  p-value: < 2.2e-16
```

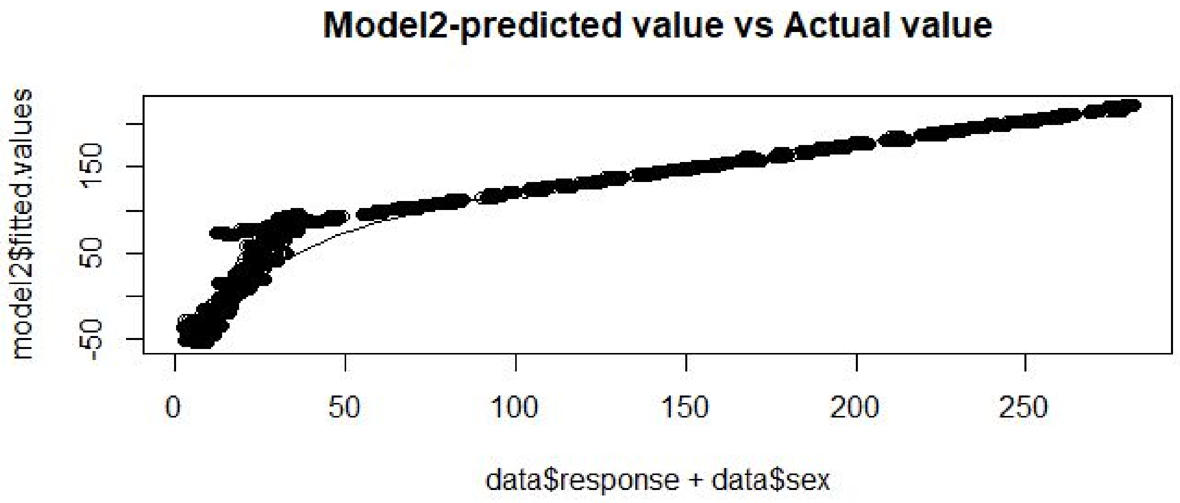
```
> error2 = residuals(model2)
> hist(error2)
```



```
> scatter.smooth(x=model2$residuals, y=model2$fitted.values, main="Model2-predicted value
vs error")
```

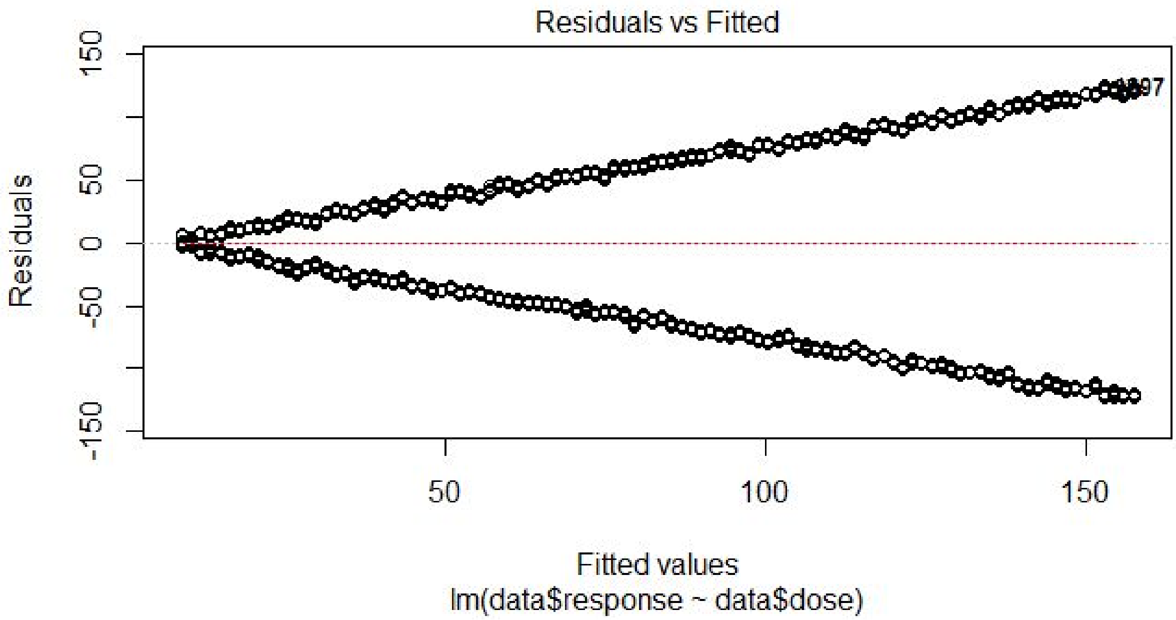


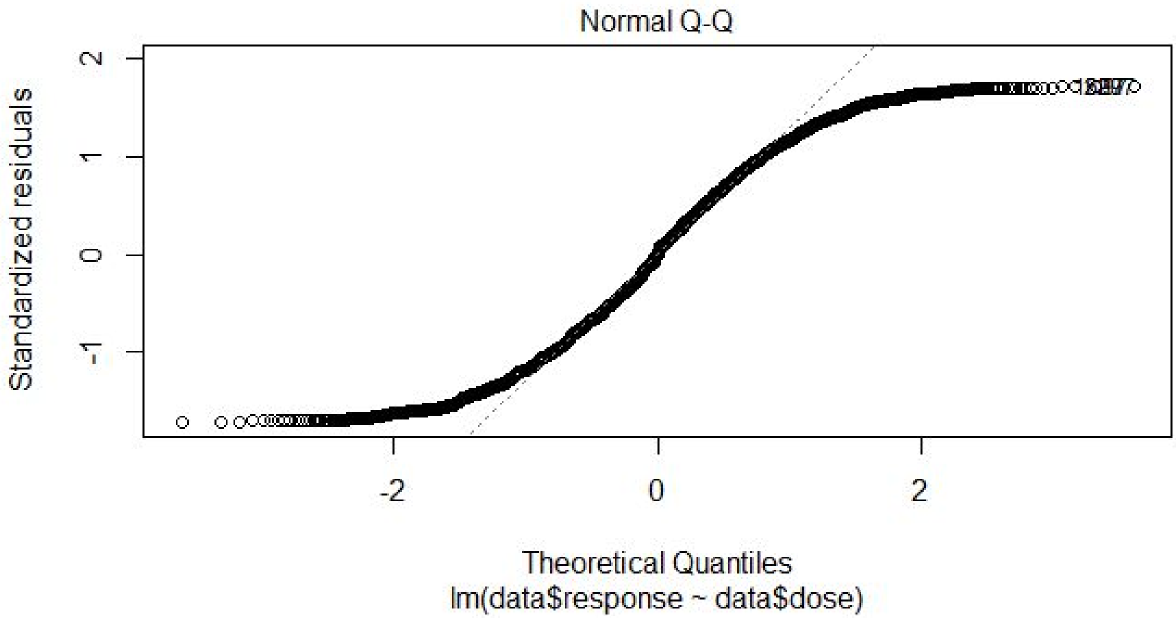
```
> scatter.smooth(x=data$response+data$sex, y=model2$fitted.values, main="Model2-predicted value vs Actual value")
```

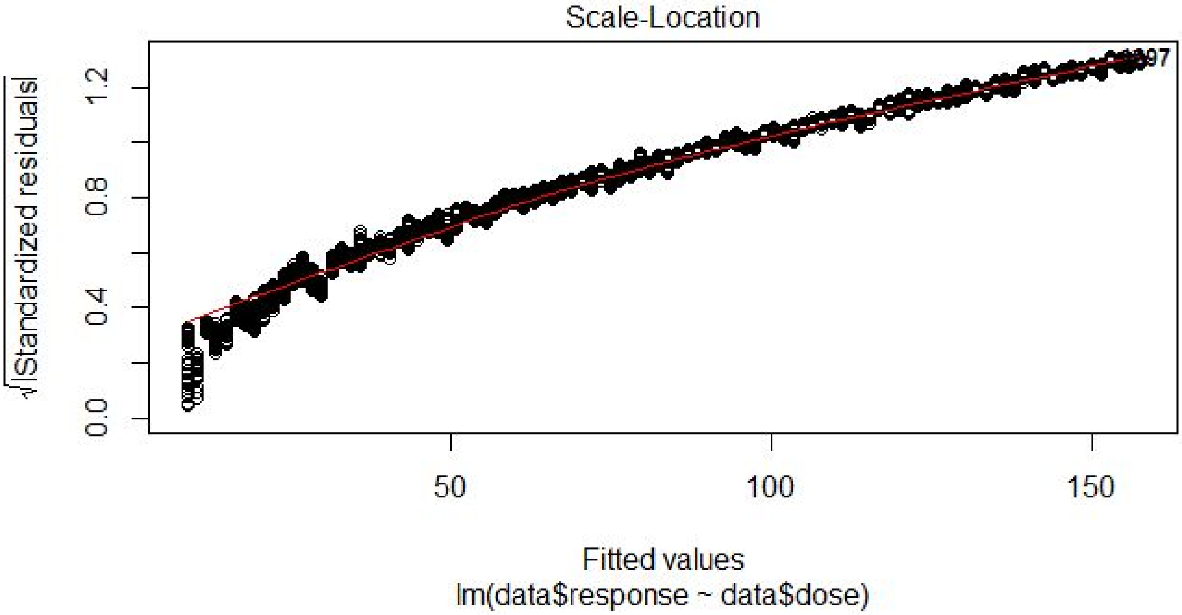


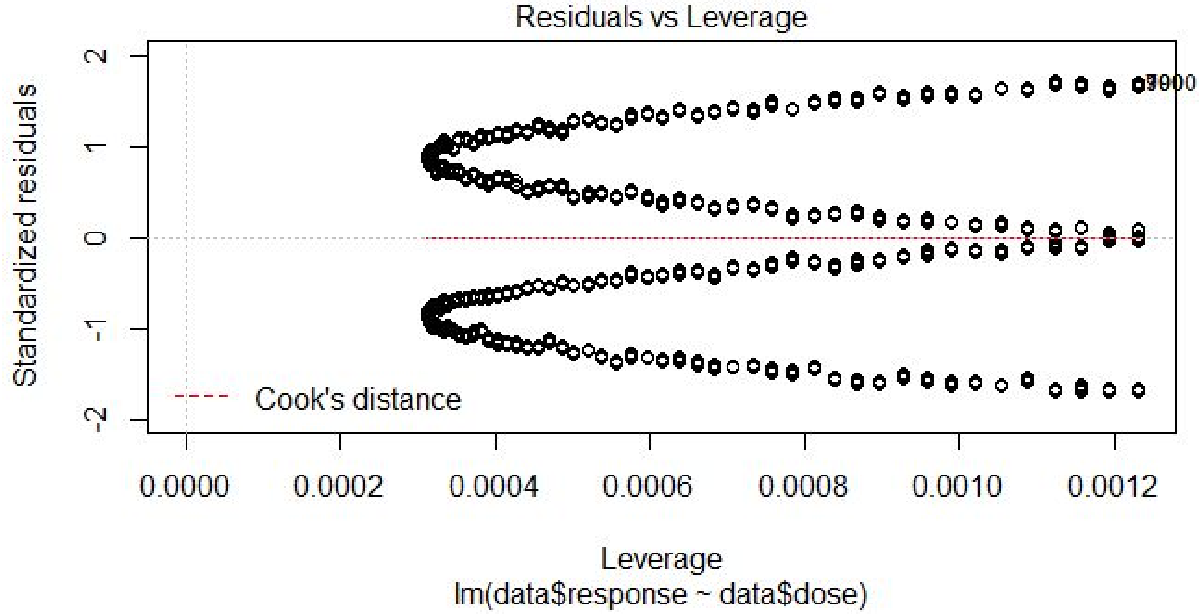
Ploting Model1

```
>plot(model1)
```


Residuals vs Fitted
Residuals
150
50
0
-50
-150
50
100
150
Fitted values
lm(data$response ~ data$dose)


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
0
2
Theoretical Quantiles
lm(data$response ~ data$dose)


Scale-Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
50
100
150
Fitted values
lm(data$response ~ data$dose)
97


Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
Leverage
lm(data$response ~ data$dose)
Cook's distance
9000

How to know if the model is best fit for your data?

The most common metrics to look at while selecting the model are:

STATISTIC	CRITERION
R-Squared	Higher the better *(> 0.70)*
Adj R-Squared	Higher the better
F-Statistic	Higher the better
Std. Error	Closer to zero the better
t-statistic	Should be greater 1.96 for p-value to be less than 0.05
AIC	Lower the better
BIC	Lower the better
Mallows cp	Should be close to the number of predictors in model
MAPE (Mean absolute percentage error)	Lower the better

MSE (Mean squared error)	Lower the better
Min_Max Accuracy => mean(min(actual, predicted)/max(actual, predicted))	Higher the better

----
-

Assumptions in Regression

Regression is a parametric approach. ‘Parametric’ means it makes assumptions about data for the purpose of analysis. Due to its parametric side, regression is restrictive in nature. It fails to deliver good results with data sets which doesn’t fulfill its assumptions. Therefore, for a successful regression analysis, it’s essential to validate these assumptions.

So, how would you check (validate) if a data set follows all regression assumptions? You check it using the regression plots (explained below) along with some statistical test.

Let’s look at the important assumptions in regression analysis:

1. There should be a linear and additive relationship between dependent (response) variable and independent (predictor) variable(s). A linear relationship suggests that a change in response Y due to one unit change in X^1 is constant, regardless of the value of X^1. An additive relationship suggests that the effect of X^1 on Y is independent of other variables.
2. There should be no correlation between the residual (error) terms. Absence of this phenomenon is known as Autocorrelation.
3. The independent variables should not be correlated. Absence of this phenomenon is known as multicollinearity.
4. The error terms must have constant variance. This phenomenon is known as homoskedasticity. The presence of non-constant variance is referred to heteroskedasticity.
5. The error terms must be normally distributed.

What if these assumptions get violated ?

Let’s dive into specific assumptions and learn about their outcomes (if violated):

1. Linear and Additive: If you fit a linear model to a non-linear, non-additive data set, the regression algorithm would fail to capture the trend mathematically, thus resulting in an inefficient model. Also, this will result in erroneous predictions on an unseen data set.

How to check: Look for residual vs fitted value plots (explained below). Also, you can include polynomial terms (X, X^2, X^3) in your model to capture the non-linear effect.

2. Autocorrelation: The presence of correlation in error terms drastically reduces model’s accuracy. This usually occurs in time series models where the next instant is dependent on previous instant. If the error terms are correlated, the estimated standard errors tend to underestimate the true standard error.

If this happens, it causes confidence intervals and prediction intervals to be narrower. Narrower confidence interval means that a 95% confidence interval would have lesser probability than 0.95 that it would contain the actual value of coefficients. Let’s understand narrow prediction intervals with an example:

For example, the least square coefficient of X^1 is 15.02 and its standard error is 2.08 (without autocorrelation). But in presence of autocorrelation, the standard error reduces to 1.20. As a result, the prediction interval narrows down to (13.82, 16.22) from (12.94, 17.10).

Also, lower standard errors would cause the associated p-values to be lower than actual. This will make us incorrectly conclude a parameter to be statistically significant.

How to check: Look for Durbin – Watson (DW) statistic. It must lie between 0 and 4. If DW = 2, implies no autocorrelation, $0 < DW < 2$ implies positive autocorrelation while $2 < DW < 4$ indicates negative autocorrelation. Also, you can see residual vs time plot and look for the seasonal or correlated pattern in residual values.

3. Multicollinearity: This phenomenon exists when the independent variables are found to be moderately or highly correlated. In a model with correlated variables, it becomes a tough task to figure out the true relationship of a predictors with response variable. In other words, it becomes difficult to find out which variable is actually contributing to predict the response variable.

Another point, with presence of correlated predictors, the standard errors tend to increase. And, with large standard errors, the confidence interval becomes wider leading to less precise estimates of slope parameters.

Also, when predictors are correlated, the estimated regression coefficient of a correlated variable depends on which other predictors are available in the model. If this happens, you'll end up with an incorrect conclusion that a variable strongly / weakly affects target variable. Since, even if you drop one correlated variable from the model, its estimated regression coefficients would change. That's not good!

How to check: You can use scatter plot to visualize correlation effect among variables. Also, you can also use VIF factor. VIF value <= 4 suggests no multicollinearity whereas a value of >= 10 implies serious multicollinearity. Above all, a correlation table should also solve the purpose.

4. Heteroskedasticity: The presence of non-constant variance in the error terms results in heteroskedasticity. Generally, non-constant variance arises in presence of outliers or extreme leverage values. Look like, these values get too much weight, thereby disproportionately influences the model's performance. When this phenomenon occurs, the confidence interval for out of sample prediction tends to be unrealistically wide or narrow.

How to check: You can look at residual vs fitted values plot. If heteroskedasticity exists, the plot would exhibit a funnel shape pattern (shown in next section). Also, you can use Breusch-Pagan / Cook – Weisberg test or White general test to detect this phenomenon.

5. Normal Distribution of error terms: If the error terms are non- normally distributed, confidence intervals may become too wide or narrow. Once confidence interval becomes unstable, it leads to difficulty in estimating coefficients based on minimization of least squares. Presence of non – normal distribution suggests that there are a few unusual data points which must be studied closely to make a better model.

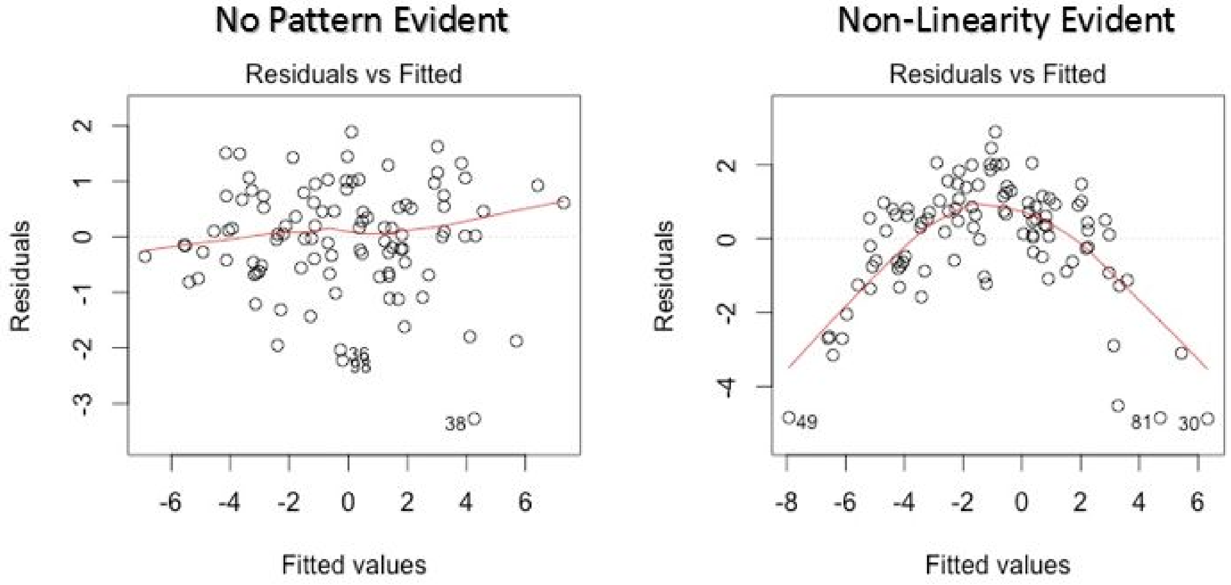
How to check: You can look at QQ plot (shown below). You can also perform statistical tests of normality such as Kolmogorov-Smirnov test, Shapiro-Wilk test.

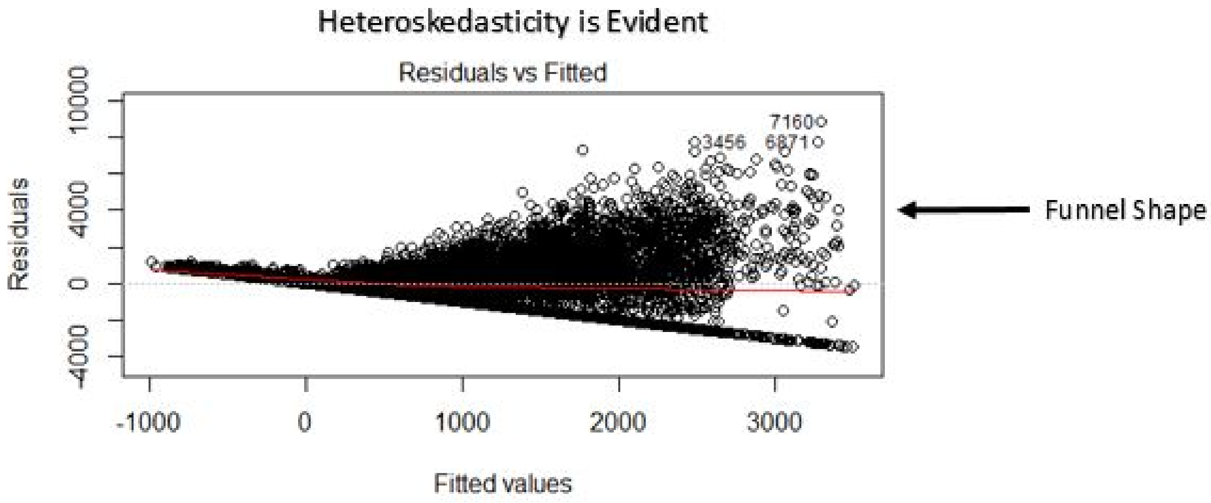
Interpretation of Regression Plots

Until here, we’ve learnt about the important regression assumptions and the methods to undertake, if those assumptions get violated.

But that’s not the end. Now, you should know the solutions also to tackle the violation of these assumptions. In this section, I’ve explained the 4 regression plots along with the methods to overcome limitations on assumptions.

1. Residual vs Fitted Values





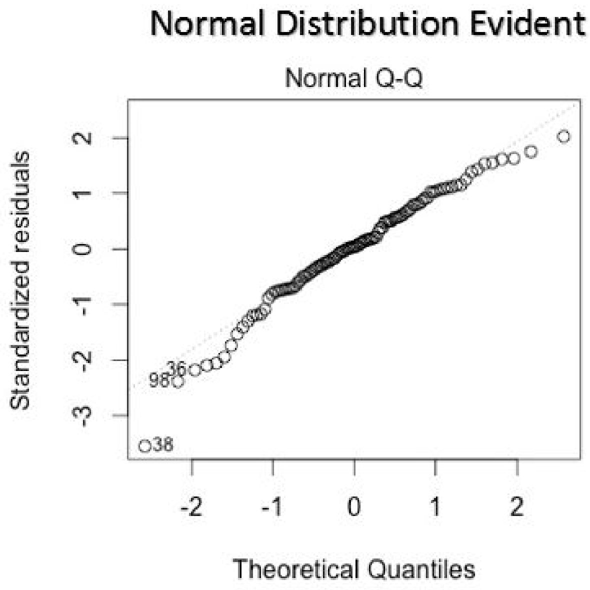
This scatter plot shows the distribution of residuals (errors) vs fitted values (predicted values). It is one of the most important plot which everyone must learn. It reveals various useful insights including outliers. The outliers in this plot are labeled by their observation number which make them easy to detect.

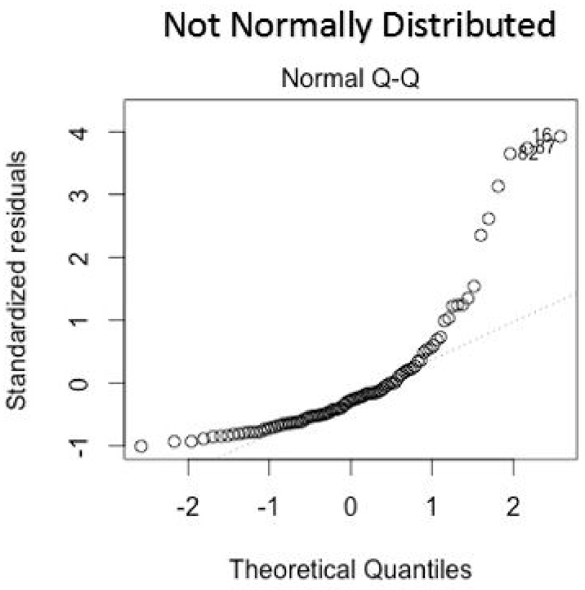
There are two major things which you should learn:

1. If there exist any pattern (may be, a parabolic shape) in this plot, consider it as signs of non-linearity in the data. It means that the model doesn't capture non-linear effects.
2. If a funnel shape is evident in the plot, consider it as the signs of non constant variance i.e. heteroskedasticity.

Solution: To overcome the issue of non-linearity, you can do a non linear transformation of predictors such as log (X), $\sqrt{X}$ or X^2 transform the dependent variable. To overcome heteroskedasticity, a possible way is to transform the response variable such as log(Y) or $\sqrt{Y}$. Also, you can use weighted least square method to tackle heteroskedasticity.

2. Normal Q-Q Plot



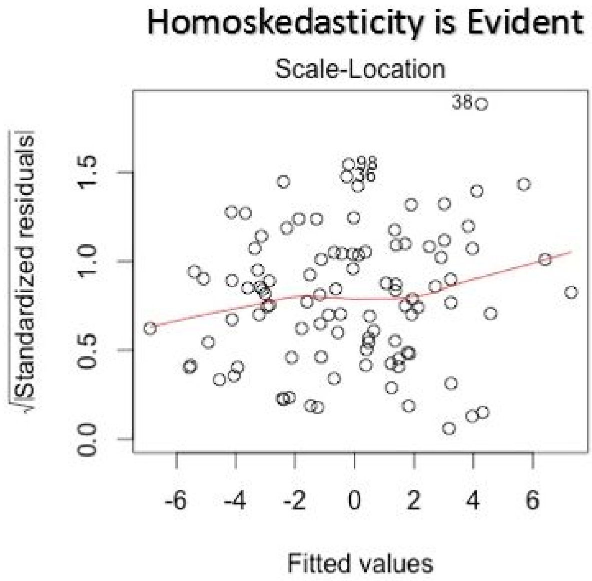


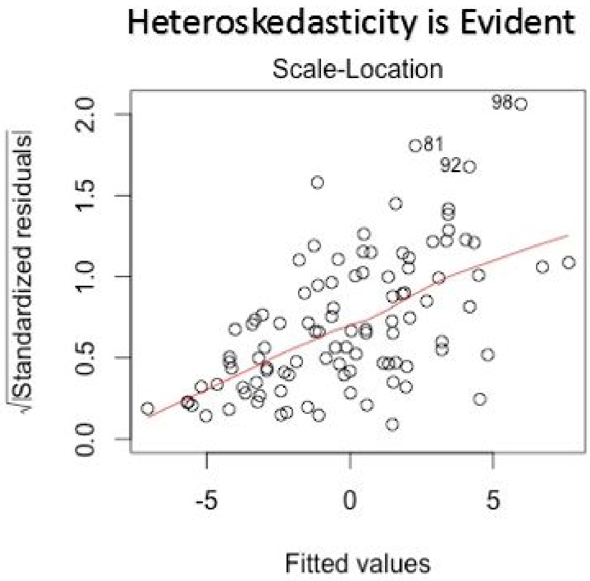
This q-q or quantile-quantile is a scatter plot which helps us validate the assumption of normal distribution in a data set. Using this plot we can infer if the data comes from a normal distribution. If yes, the plot would show fairly straight line. Absence of normality in the errors can be seen with deviation in the straight line.

If you are wondering what is a ‘quantile’, here’s a simple definition: Think of quantiles as points in your data below which a certain proportion of data falls. Quantile is often referred to as percentiles. For example: when we say the value of 50th percentile is 120, it means half of the data lies below 120.

Solution: If the errors are not normally distributed, non – linear transformation of the variables (response or predictors) can bring improvement in the model.

3. Scale Location Plot

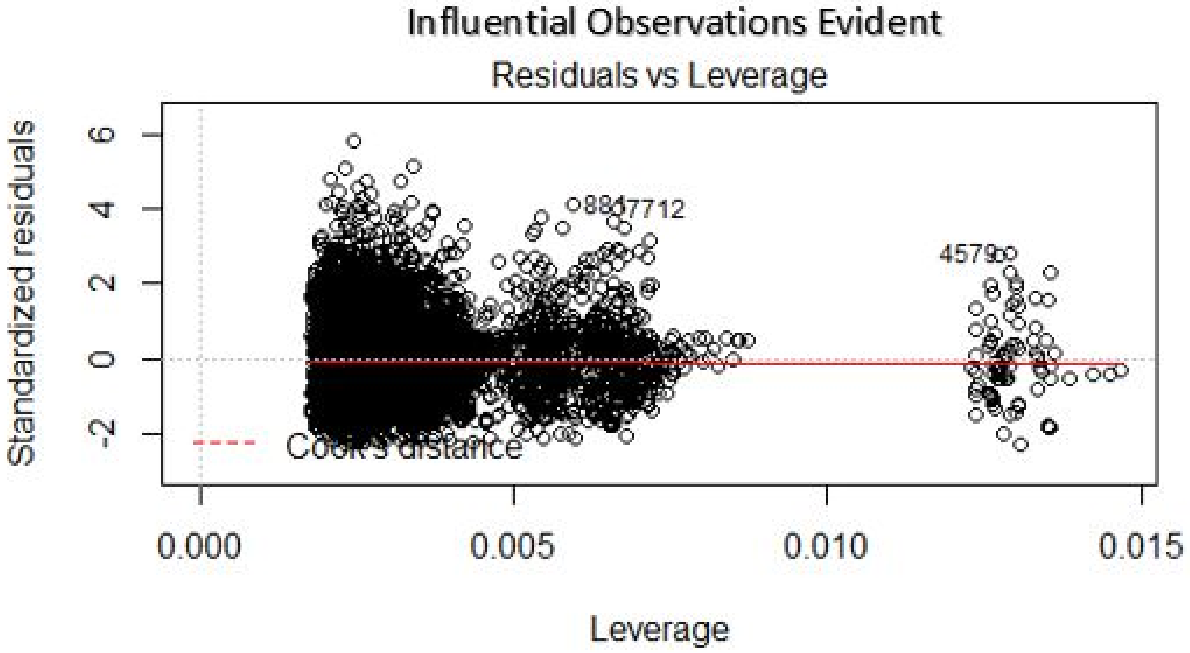




This plot is also used to detect homoskedasticity (assumption of equal variance). It shows how the residual are spread along the range of predictors. It's similar to residual vs fitted value plot except it uses standardized residual values. Ideally, there should be no discernible pattern in the plot. This would imply that errors are normally distributed. But, in case, if the plot shows any discernible pattern (probably a funnel shape), it would imply non-normal distribution of errors.

Solution: Follow the solution for heteroskedasticity given in plot 1.

4. Residuals vs Leverage Plot



It is also known as Cook’s Distance plot. Cook’s distance attempts to identify the points which have more influence than other points. Such influential points tends to have a sizable impact of the regression line. In other words, adding or removing such points from the model can completely change the model statistics.

But, can these influential observations be treated as outliers? This question can only be answered after looking at the data. Therefore, in this plot, the large values marked by cook’s distance might require further investigation.

Solution: For influential observations which are nothing but outliers, if not many, you can remove those rows. Alternatively, you can scale down the outlier observation with maximum value in data or else treat those values as missing values.

Ref.
https://www.analyticsvidhya.com
http://r-statistics.co/Linear-Regression.html