3. Linear Regression – Assumptions

The regression has five important assumptions:

- **Linear relationship** → Linear regression assumes the relationship between the independent and dependent variables to be linear.
- **Multivariate normality**→ The data is normally distributed
- **Little or No multicollinearity** → Independent variables are not correlated with each other.
- **No auto-correlation** → Autocorrelation occurs when the residuals are not independent from each other.
- **Homoscedasticity** → The scatter plot is good way to check whether the data are homoscedastic which means the residuals are equal across the regression line.
- Note : Errors and Residuals are two closely related measures of the deviation of an observed value of an element .The error of an observed value is the deviation of the observed value from the true value of a quantity of interest (for example, a population mean), and the residual of an observed value is the difference between the observed value and the estimated value of the quantity of interest (for example, a sample mean).

3. Linear Regression – Types

Simple and Multiple Linear Regression

- **Simple Linear Regression** → Simple linear regression is used to find relationship between two continuous variables One predictor or independent variable and other target or dependent variable.
- **Multiple Linear Regression** → Multiple linear regression is used to find relationship between more than two continuous variables. More than one predictors or independent variables and one target or dependent variable.

2. Linear Regression – Split Data- numpy

We can use python library numpy to split the data.

```
import numpy as np
print (X_data.shape)
#Take the first number from X_data.shape
num_of_rows = (enter your number here) * 0.8
np.random_shuffle(X_data)
#shuffles data to make it random
train_data = X_data.iloc[:num_of_rows]
#indexes rows for training data
test_data = X_data.iloc[num_of_rows:]
#indexes rows for test data
train_data.sort()
```

3. Linear Regression – Split Data- sklearn

We can use python library scikit-Learn to split the data.

```
# Read the csv file
df   = pd.read_csv('data.csv')
# create the target variable
y = diabetes.target
# create training and testing data split
X_train, X_test, y_train, y_test = train_test_split(df, y, test_si
ze=0.2)
```