Experiment No.2
Implement R Program for Simple Linear Regression
Date of Performance:
Date of Submission:

EXPERIMENT NO 2

Aim: Towrite the implementation of linear regression.

Objective:- Tounderstand the use of simple linear regression techniques by implementing user define dataset and importing dataset

Description:

Regression analysis is a very widely used statistical tool to establish a relationship model between two variables. One of these variables is called a predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

In Linear Regression these two variables are related through an equation, where the exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

The general mathematical equation for a linear regression is

$y = ax + b$

Following is the description of the parameters used-

y is the response variable. x is the predictor variable.

a and b are constants which are called the coefficients.

Procedure:

The steps to create the relationship is

1. Carry out the experiment of gathering a sample of observed values of height and corresponding weights.
2. Create a relationship model using the **lm()** functions in R.
3. Find the coefficients from the model created and create the mathematical equation using these Get a summary of the relationship model to know the average error in prediction. Also called **residuals.**

To predict the weight of new persons, use the **predict()** function in R.

Program:

```
import numpy as np
class LinearRegression:
    def __init__ (self):
        self.b_0 = 0
        self.b_1 = 0

    def fit (self, X, y):
        X_mean = np.mean (X)
        y_mean = np.mean (y)
        ssxy, ssx = 0, 0
        for _ in range (len (X)):
            ssxy += (X[_]-X_mean)*(y[_]-y_mean)
            ssx += (X[_]-X_mean)**2

        self.b_1 = ssxy / ssx
        self.b_0 = y_mean - (self.b_1*X_mean)
        return self.b_0, self.b_1

    def predict (self, X):
        y_hat = self.b_0 + (X * self.b_1)
        return y_hat

if __name__ == '__main__':
    X = np.array ([173, 182, 165, 154, 170], ndmin=2)
    X = X.reshape(5, 1)
    y = np.array ([68, 79, 65, 57, 64])
    model = LinearRegression ()
    model.fit (X, y)
    y_pred = model.predict ([161])
    print (y_pred)
```

OUTPUT:

```
[60.85051546]
```

Conclusion:

1. **Function used for linear regression in R is** lm() Function (Function_name (Parameters))
2. **Explain use of summary().**
   The summary() function is a useful tool in data analysis for providing a concise overview of key statistics and characteristics of a dataset. It typically generates descriptive statistics such as mean, median, standard deviation, minimum, maximum, and quartiles for numerical variables, offering insights into the central tendency, dispersion, and distribution of the data. Additionally, summary() can also display the count of non-missing values, helping to identify missing data and assess data completeness. Overall, summary() serves as a quick and efficient way to understand the essential properties of a dataset, aiding in the initial exploration and assessment of data quality.