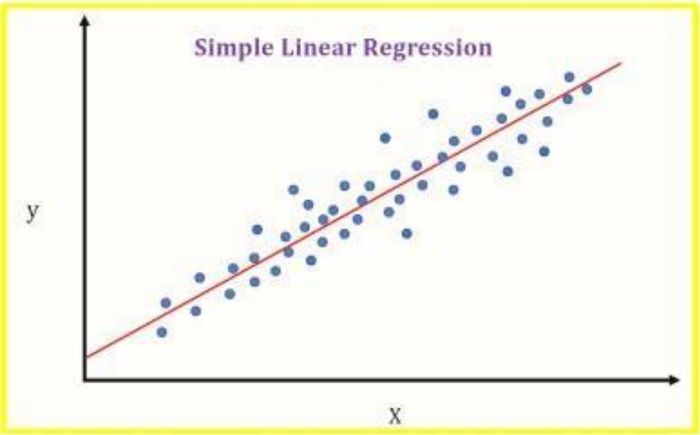
Experiment 1

Aim: Perform linear Regression on database imported from web storage

Theory:

Linear Regression is a machine learning algorithm based on supervised learning. Linear regression attempts to model the relationship between two variables by **fitting a linear equation to observed data**.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a **linear relationship** between x (input) and y(output) and variation in predicted value from actual answer is due to random noise .



The motive of the simple linear regression algorithm is to find the best values for m and c and then use to predict values. The line having that value of m and c which is used for prediction is called Best Fit

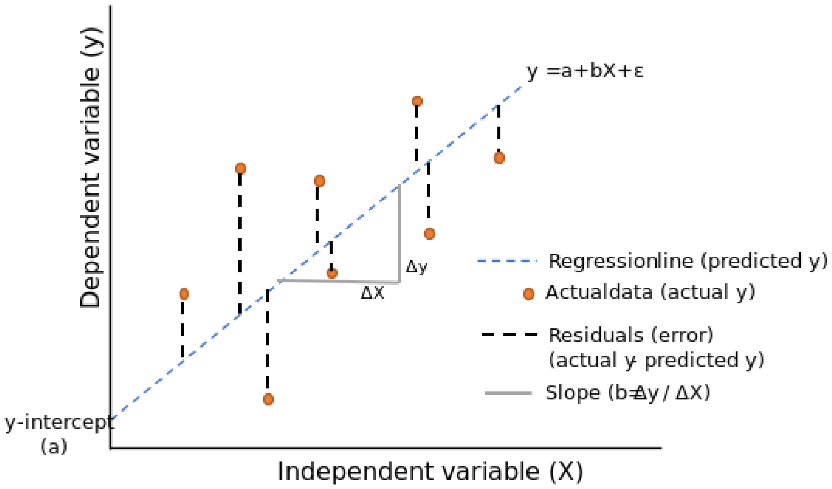
The line can be modelled based on the linear equation shown below.

$Y=m\times X+c$ is the general equation of univariate linear relation between variables. Where-

m is the slope of line,

X is the input data,

c is y intercept, also called bias.



Program: Write a program in Python or R programming language to implement the concepts discussed above.

Sample Python Program

`Structure of 'Salary_Data.csv'`

YearsExperience	Salary
1.1	39343
1.3	46205
1.5	37731
2	43525
2.2	39891
2.9	56642
3	60150
3.2	54445
3.2	64445

```
# Simple Linear Regression

# Importing the required libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset ( Here we used salary data)
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
1/3, random_state = 0)

# Training the Simple Linear Regression model on the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualising the Training set results
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()

# Visualising the Test set results
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

Sample Program in R

```
# Importing the dataset
dataset = read.csv('Salary_Data.csv')


# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Salary, SplitRatio = 2/3)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
```

```r
# Feature Scaling
# training_set = scale(training_set)
# test_set = scale(test_set)

# Fitting Simple Linear Regression to the Training set
regressor = lm(formula = Salary ~ YearsExperience, data = training_set)

# Predicting the Test set results
y_pred = predict(regressor, newdata = test_set)

# Visualising the Training set results
install.packages('ggplot2')
library(ggplot2)
ggplot() +
  geom_point(aes(x = training_set$YearsExperience, y = training_set$Salary),  colour = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, newdata = training_set)),
            colour = 'blue') +
  ggtitle('Salary vs Experience (Training set)') +
  xlab('Years of experience') +
  ylab('Salary')

# Visualising the Test set results

library(ggplot2)
ggplot() +
  geom_point(aes(x = test_set$YearsExperience, y = test_set$Salary), colour = 'red') +
  geom_line(aes(x = training_set$YearsExperience, y = predict(regressor, newdata = training_set)),
            colour = 'blue') +
  ggtitle('Salary vs Experience (Test set)') +
  xlab('Years of experience') +
  ylab('Salary')
```

Show the Results to the Supervisor.

Conclusion: Write 4 to 5 lines of conclusion in your own words.