Consider equation of a straight line,

```
y = c + mx
```

- where y is the dependent variable
- c is a constant,
- x is independent variable
- m is an coefficient i.e. slope of the line.

.

We are going to use LinearRegression class from sklearn.linear_model library.To implement simple linear regression we are going to create a new dataset containing at least 30 records of year of experience and total salary as follows.

```
YearsExperience,Salary
1.1,39343.00
1.3,46205.00
1.5,37731.00
2.0,43525.00
2.2,39891.00
2.9,56642.00
3.0,60150.00
3.2,54445.00
3.2,64445.00
3.7,57189.00
3.9,63218.00
4.0,55794.00
4.0,56957.00
4.1,57081.00
4.5,61111.00
4.9,67938.00
5.1,66029.00
5.3,83088.00
5.9,81363.00
6.0,93940.00
6.8,91738.00
7.1,98273.00
7.9,101302.00
8.2,113812.00
8.7,109431.00
9.0,105582.00
9.5,116969.00
9.6,112635.00
10.3,122391.00
10.5,121872.00
```

Preprocess the dataset and also divide the dataset into train and test dataset as follows.

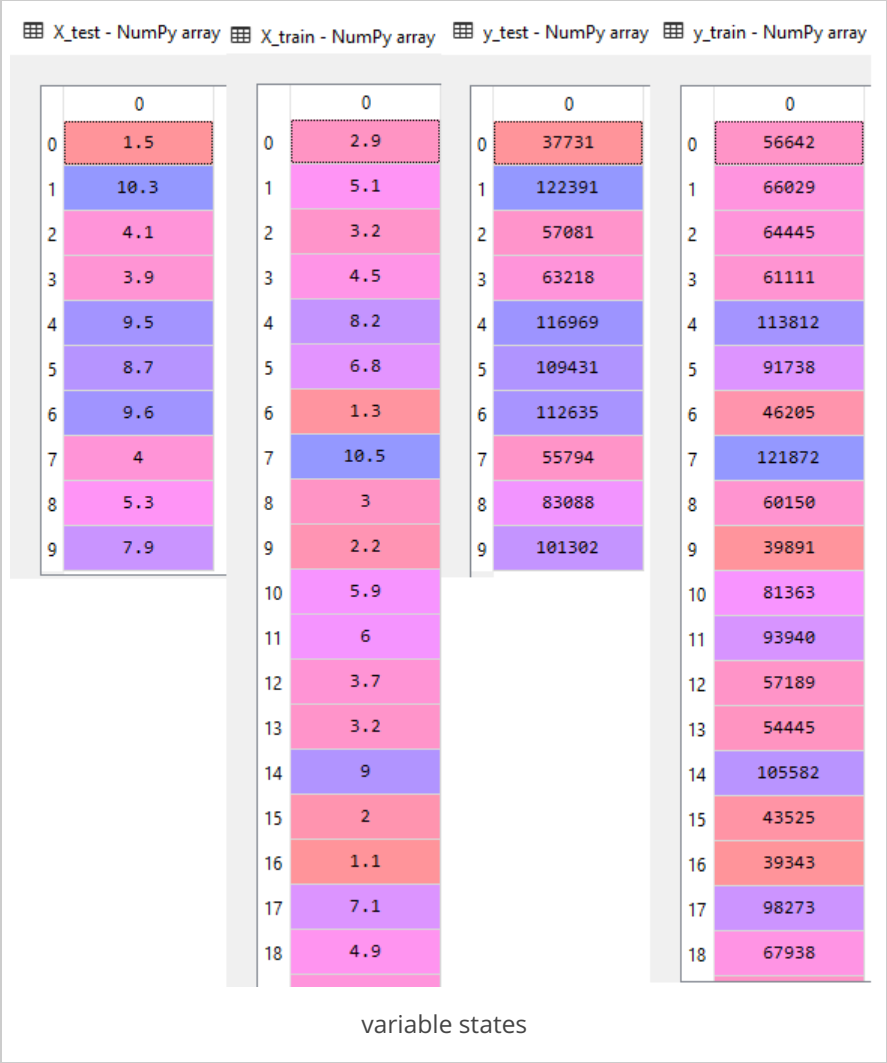
```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
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.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)
```

- x_train : training data of independent variables. i.e. years of experience
- x_test : test data for which we want to predict salaries
- y_train : training data o dependent variables i.e. salaries based on years of experience
- y_test : actual salaries for years of experience in x_test

variables are as below. Please note x_train and y_train contain 20 values.



variable states

We are to fit our training dataset into simple linear regression model.To do this create an object **regressor** of class **LinearRegression**. Fit training data i.e. x_train and y_train in regressor as below.

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

regressor.fit() method takes dependent and independent variables as parameters. We are actually teaching the regressor that y_train values are all corresponding to X_train values.

Predicting salaries

We are now going to predict the salaries related to X_test values i.e years of experience and compare them with actual i.e values of y_test as below.

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

regressor.predict() method predicts the values of salaries depending on the years of experience in X_test.

y_pred values are predicted salaries and we will compare them with actual salaries which we have in y_test.

Image besides shows years of experience, predicted salaries and actual salaries.

- Regressor has predicted 40835.1 salary for an employee with 1.5 years of experience whose actual salary is 37731.
- Regressor has predicted 123079 salary for an employee with 10.3 experience whose actual salary is 122391

Here,machine has learned to predict the salaries based on years of experiences.

Simple Linear Regression Graph:

Training Set

X_test - NumPy array | y_pred - NumPy array | y_test - NumPy array

	0
0	1.5
1	10.3
2	4.1
3	3.9
4	9.5
5	8.7
6	9.6
7	4
8	5.3
9	7.9

	0
0	40835.1
1	123079
2	65134.6
3	63265.4
4	115603
5	108126
6	116537
7	64200
8	76349.7
9	100649

	0
0	37731
1	122391
2	57081
3	63218
4	116969
5	109431
6	112635
7	55794
8	83088
9	101302

x_test y_pred and y_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()
```

- `plt.scatter(X_train, y_train , color = 'red')` plots scatter graph of salaries against years of experience for values in X_train and y_train
- `plt.plot(X_train, regressor.predict(X_train), color = 'blue')` plots the graph of predicted salaries against years of experience.
- Red dots represents co-relation between X_train and y_train i.e salaries and years of experience
- Blue line is the simple linear regression.

Training Set graph

Test Set:

```
# 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()
```

`plt.scatter(X_test, y_test , color = 'red')` plots scatter graph of salaries against years of experience for values in X_test and y_test.

Please note that blue regression line remains the same as it shows all predicted salaries for any years of experience.

From the graph and our comparison of y_pred and y_test we can say that we have successfully predicted salaries for any given number of years of experience using Simple Linear Regression using python.

I hope this article helped understand Simple Linear Regression. In next article we will learn about multiple linear regression.