

In [1]: ☐ `print("CS-Cult Program By Amit Basantani")`

CS-Cult Program By Amit Basantani

In [2]: ☐ `#http://bit.ly/w-data
import pandas as pd
import matplotlib.pyplot as plt`

In [3]: ☐ `data_url = "http://bit.ly/w-data"
df = pd.read_csv(data_url)`

In [4]: ☐ df

Out[4]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [5]: ☐ `df.head()`

Out[5]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [6]: ☐ `df.tail()`

Out[6]:

	Hours	Scores
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [7]: ☐ `df.shape`

Out[7]: (25, 2)

In [8]: `df.isna()`

Out[8]:

	Hours	Scores
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False
20	False	False
21	False	False
22	False	False
23	False	False
24	False	False

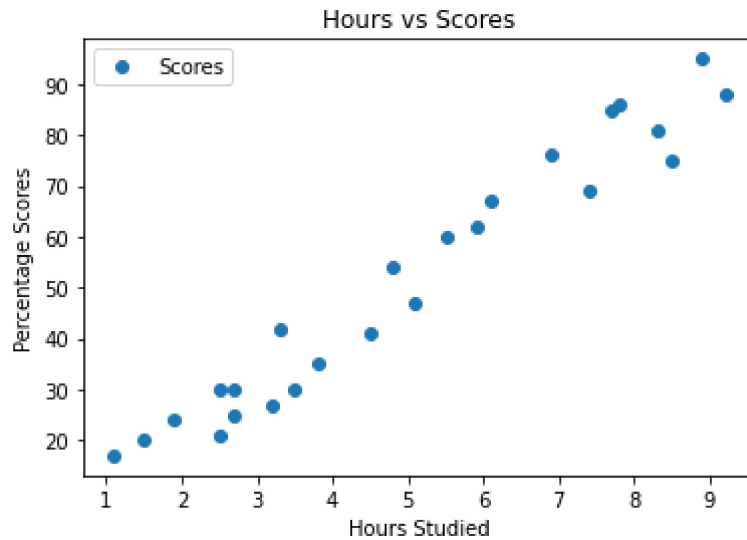
In [9]: `df.isna().sum()`

Out[9]:

Hours	0
Scores	0

dtype: int64

```
In [10]: df.plot(x= 'Hours', y = 'Scores', style = 'o')
plt.title('Hours vs Scores')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Scores')
plt.show()
```



```
In [11]: df.describe()
```

Out[11]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

```
In [12]: x= df.iloc[:, :-1].values
y= df.iloc[:, 1].values
```

In [13]: ☐ x

```
Out[13]: array([[2.5],
                [5.1],
                [3.2],
                [8.5],
                [3.5],
                [1.5],
                [9.2],
                [5.5],
                [8.3],
                [2.7],
                [7.7],
                [5.9],
                [4.5],
                [3.3],
                [1.1],
                [8.9],
                [2.5],
                [1.9],
                [6.1],
                [7.4],
                [2.7],
                [4.8],
                [3.8],
                [6.9],
                [7.8]])
```

In [14]: ☐ y

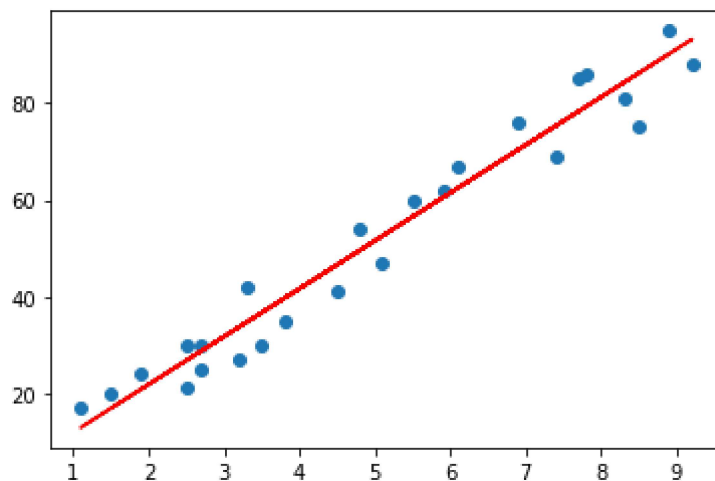
```
Out[14]: array([21, 47, 27, 75, 30, 20, 88, 60, 81, 25, 85, 62, 41, 42, 17, 95, 30,
                24, 67, 69, 30, 54, 35, 76, 86], dtype=int64)
```

In [15]: ☐ `from sklearn.model_selection import train_test_split`
`X_train , X_test , y_train , y_test = train_test_split(x, y, test_size = 0.2`

In [16]: ☐ `from sklearn.linear_model import LinearRegression`
`regressor = LinearRegression()`
`regressor.fit(X_train , y_train) #training complete!`

```
Out[16]: LinearRegression()
```

```
In [17]: # y = mx + c
line = regressor.coef_*x+regressor.intercept_
plt.scatter(x,y)
plt.plot(x, line, color = 'red')
plt.show()
```



```
In [18]: print(X_test)
y_pred = regressor.predict(X_test)
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

```
In [19]: data_frame = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
```

In [20]: ☐ data_frame

Out[20]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

In [22]: ☐

```
hours = [[1.1]]
own_pred = regressor.predict(hours)
print("No. of hours you entered for checking = {}".format(hours))
print("Predicted score the hour you enteres = {}".format(own_pred[0]))
```

No. of hours you entered for checking = [[1.1]]
Predicted score the hour you enteres = 12.919882170141145

In []: ☐