

```
from google.colab import files
uploaded = files.upload()
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving student_scores.csv to student_scores.csv

```
import pandas as pd
```

```
import io
df2 = pd.read_csv(io.BytesIO(uploaded['student_scores.csv']))
print(df2)
```

	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

```
import pandas as pd
```

```
df = pd.read_csv('student_scores.csv')
```

```
row_count, column_count = df.shape
```

```
print("Number of rows ", row_count)
print("Number of columns ", column_count)
```

```
Number of rows 25
Number of columns 2
```

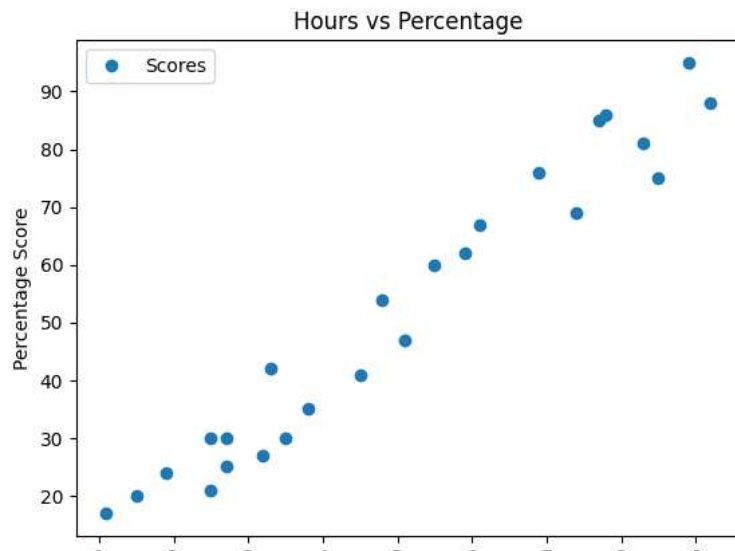
```
import pandas as pd
```

```
df = pd.read_csv('student_scores.csv')
#Display the first 10 rows
result = df.head(5)
print("First 5 rows of the DataFrame:")
print(result)
```

```
First 5 rows of the DataFrame:
   Hours  Scores
0    2.5     21
1    5.1     47
2    3.2     27
3    8.5     75
4    3.5     30
```

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error
```

```
df2.plot(x= 'Hours', y='Scores',style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours studied ')
plt.ylabel('Percentage Score')
plt.show()
```



```
x = df2.iloc[:, :-1].values
print(x)
y=df2.iloc[:,1].values
#print(y )
```

```
[[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]]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,random_state=0)
print("training -x",X_train)
print("training -y",y_train)
print("testing -x",X_test)
print("testing -y",y_test)
```

```
training -x [[6.9]
 [1.1]
 [5.1]
 [7.7]
 [3.3]
 [8.3]
 [9.2]
 [6.1]
 [3.5]
 [2.7]
 [5.5]
 [2.7]
 [8.5]
 [2.5]
 [4.8]
 [8.9]
 [4.5]]
training -y [76 17 47 85 42 81 88 67 30 25 60 30 75 21 54 95 41]
testing -x [[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]
 [3.8]
 [1.9]]
```

```
[7.8]]
testing -y [20 27 69 30 62 35 24 86]

model = linear_model.LinearRegression()
model.fit(X_train, y_train)
y_predicted = model.predict(X_test)
print(X_test)

[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]
 [3.8]
 [1.9]
 [7.8]]

print("Mean Squared Error : ",mean_squared_error(y_test,y_predicted))

Mean Squared Error : 22.965097212700428

#Print "Weights and Intercept : "
print("Hours : ",model.coef_)
print("Score : ",model.intercept_)

Hours : [9.78856669]
Score : 2.3708153823418883

plt.scatter(X_test,y_test, color="blue")
plt.plot(X_test,y_predicted, color="orange", lw=3)
plt.show()
```

