DATASCIENCE & BUSINESS ANALYTICS

SUPERVISED ML-LINEAR REGRESSION

Predict the percentage of an student based on the no. of study hours.

AUTHOR:Kaviya.G

Imports

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Get the data

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

In [4]:
```
data.head()
```

Out[4]:

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

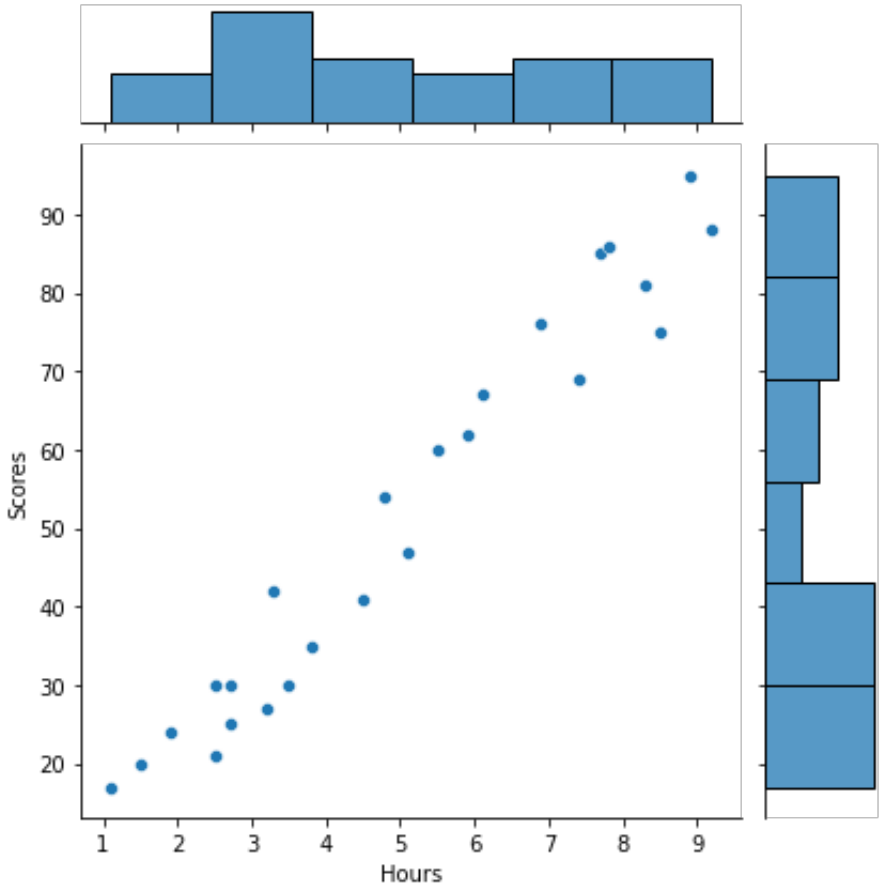
EXPLORATORY DATA ANALYSIS

In [4]:
```
data.columns
```

Out[4]:
```
Index(['Hours', 'Scores'], dtype='object')
```

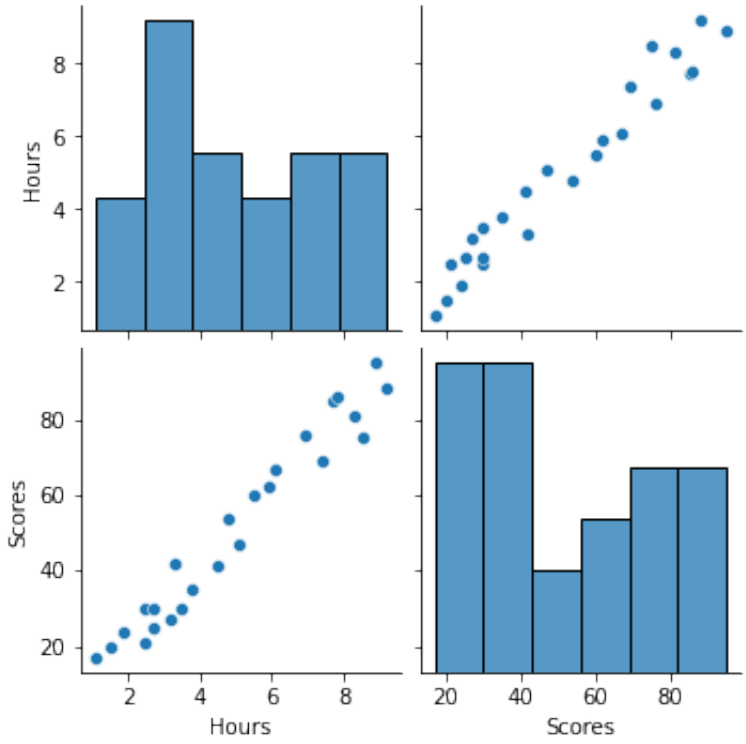
In [5]:
```
sns.jointplot(x='Hours',y='Scores',data=data)
```

Out[5]:
```
<seaborn.axisgrid.JointGrid at 0x7faca073af40>
```



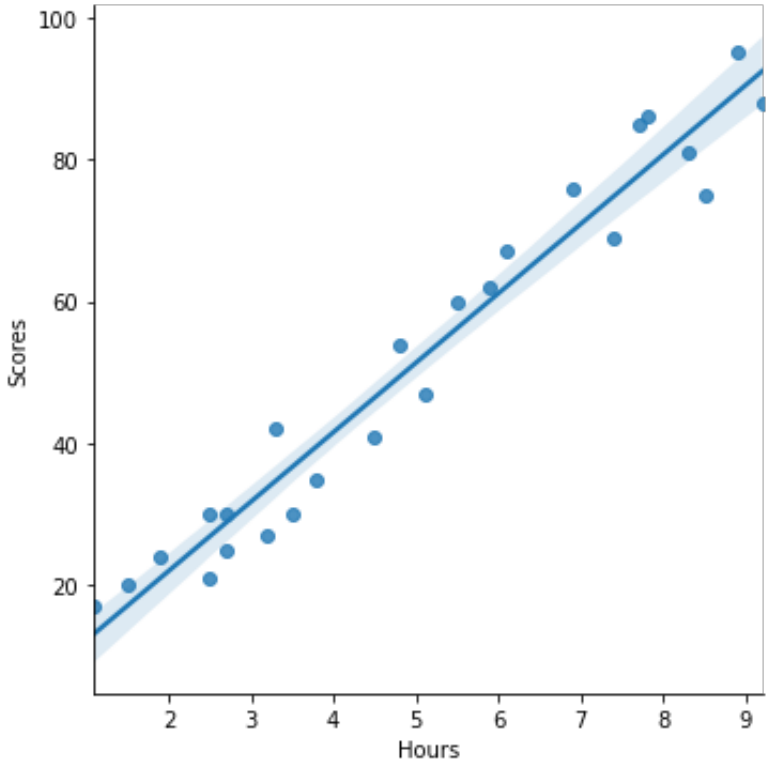
In [6]:
```
sns.pairplot(data=data)
```

Out[6]: <seaborn.axisgrid.PairGrid at 0x7faca073ae20>



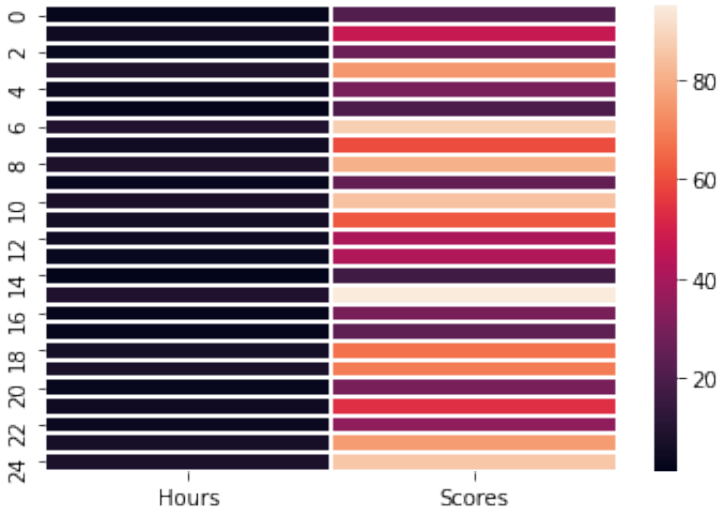
In [7]:
```
sns.lmplot(x='Hours',y='Scores',data = data)
```

Out[7]: <seaborn.axisgrid.FacetGrid at 0x7faca1726580>



In [6]:
```
sns.heatmap(data,linecolor='white',linewidth='1')
```

Out[6]: <AxesSubplot:>



Splitting the data

In [7]:
```
x=data[["Hours"]]
```

In [8]:
```
y=data["Scores"]
```

MODEL SELECTION

Training and testing data

In [9]:
```
from sklearn.model_selection import train_test_split
```

In [10]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size
```

Training the model

In [11]:
```
from sklearn.linear_model import LinearRegression
```

Create an instance of a LinearRegression() model named lm.

In [12]:
```
lm=LinearRegression()
```

In [13]:
```
lm.fit(x_train,y_train)
```

Out[13]:
```
LinearRegression()
```

Print out the coefficients of the model

In [14]:
```
print(lm.coef_)
```

```
[9.94167834]
```

In [15]:
```
print(lm.intercept_)
```

```
1.932204253151646
```

In [17]:
```
df1=pd.DataFrame(lm.coef_,x.columns,columns=["coeff"])
df1
```

Out[17]:

	coeff
Hours	9.941678

PREDICTING TEST DATA

In [18]:
```
p=lm.predict(x_test)
print(p)
```

```
[16.84472176 33.74557494 75.50062397 26.7864001  60.58810646 39.71
058194
 20.8213931 ]
```

In [19]:
```
df2=pd.DataFrame(p,y_test)
df2
```

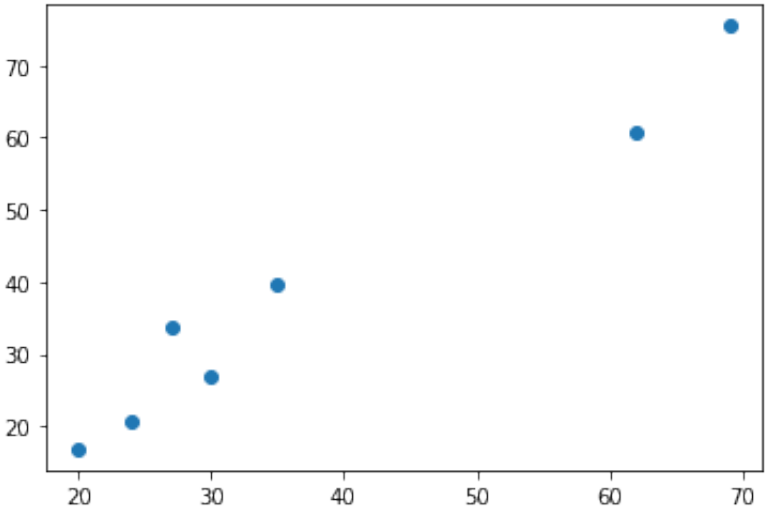
Out[19]:

	0
Scores	
20	16.844722
27	33.745575
69	75.500624
30	26.786400
62	60.588106
35	39.710582
24	20.821393

Create a scatterplot of the real test values versus the predicted values.

In [20]:
```
plt.scatter(y_test,p)
```

Out[20]:
```
<matplotlib.collections.PathCollection at 0x7f9d0f1fcb20>
```



Evaluating the Model

In [22]:
```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test,p))
print('MSE:', metrics.mean_squared_error(y_test, p))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, p)))
```

```
MAE: 4.130879918502486
MSE: 20.33292367497997
RMSE: 4.5092043283688055
```

In [53]:
```
print(lm.predict([[9.25]]))
```

```
[93.89272889]
```

In [54]:
```
df2=pd.DataFrame({'Actual':y_test,'Predicted':p})
```

df2

In [55]:
```
df2
```

Out[55]:

	Actual	Predicted
5	20	16.844722
2	27	33.745575
19	69	75.500624
16	30	26.786400
11	62	60.588106
22	35	39.710582
17	24	20.821393

Conculsion

Predicted score as per algorithm for student studied for 9.25 hrs/day is 93.89 %

In [ ]:
```

```