Name: - L Prathyusha

TASK 1 - Data science and Business Analytics Internship

In [42]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [49]:

```
data=pd.read_csv("C:/Users/Prathyu Lachireddy/Desktop/student_scores - student_scores.csv.c
data.head(20)
```

Out[49]:

	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

In [6]:

```
data.head()
```

Out[6]:

	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 [7]:

```
data.columns
```

Out[7]:

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

In [8]:

```
data.dtypes
```

Out[8]:

```
Hours     float64
Scores      int64
dtype: object
```

In [9]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [10]:

```
data.describe()
```

Out[10]:

	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 [11]:

```
data.corr()
```

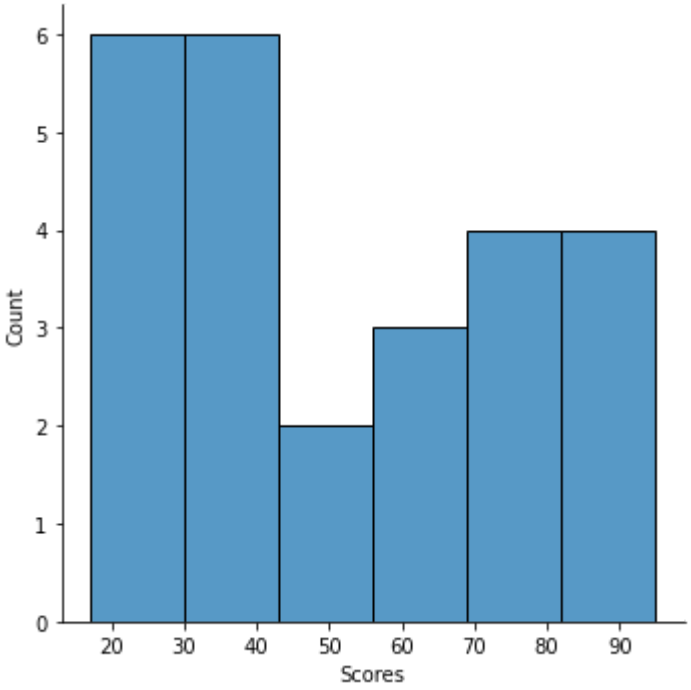
Out[11]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

DATA VISUALISATION

In [12]:

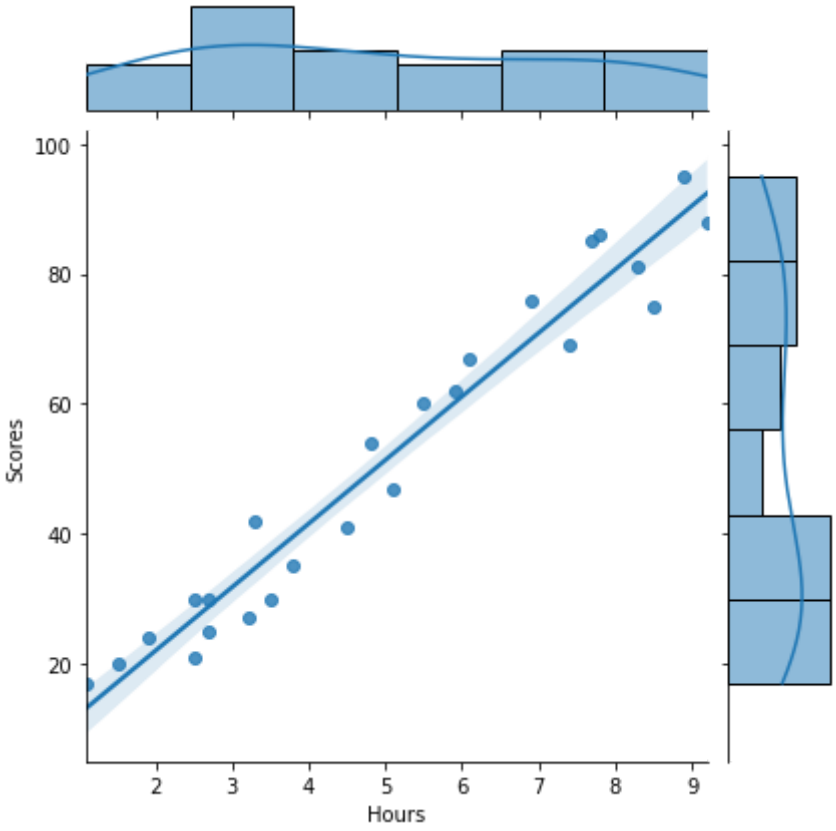
```
sns.displot(data["Scores"])
plt.show()
```



In [13]:

```
sns.jointplot(data["Hours"],data["Scores"], kind="reg")
plt.show()
```

```
C:\Users\Prathyu Lachireddy\anaconda3\lib\site-packages\seaborn\_decorators.
py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and pass
ing other arguments without an explicit keyword will result in an error or m
isinterpretation.
  warnings.warn(
```



In [14]:

```
def null_detection(data):
    num_cols = []

    count = 0
    t = []
    for i in num_cols:
        z=np.sbs(stats.zscore(data[i]))
        for j in range(len(z)):
            if z[j]>3 or z[j]<3:
                t.append(j)
                count+=1
    data=data.drop(list(set(t)))
    data=data.reset_index()
    data=data.drop('index',axis=1)
    print(count)
    return data
```

In [15]:

```
df = null_detection(data)
```

```
0
```

Simple linear regression

In [19]:

```
mean_x=np.mean(data['Hours'])
mean_y=np.mean(data['Scores'])
num=0
den=0
x = list(data['Hours'])
y = list(data['Scores'])
for i in range(len(data)):
    num += (x[i]-mean_x)*(y[i]-mean_y)
    den += (x[i]-mean_x)**2
B1=num/den
B1
```

Out[19]:

```
9.775803390787475
```

In [20]:

```
B0 = mean_y - B1*mean_x
B0
```

Out[20]:

```
2.4836734053731746
```

In [21]:

```
data['Prediction'] = B0 + B1*data['Hours']
data.head()
```

Out[21]:

	Hours	Scores	Prediction
0	2.5	21	26.923182
1	5.1	47	52.340271
2	3.2	27	33.766244
3	8.5	75	85.578002
4	3.5	30	36.698985

solution of 9.25 hours

In [22]:

```
B0 + B1*9.25
```

Out[22]:

```
92.90985477015732
```

In [23]:

```
y = list (data['Scores'].values)
y_pred = list(data['Prediction'].values)
```

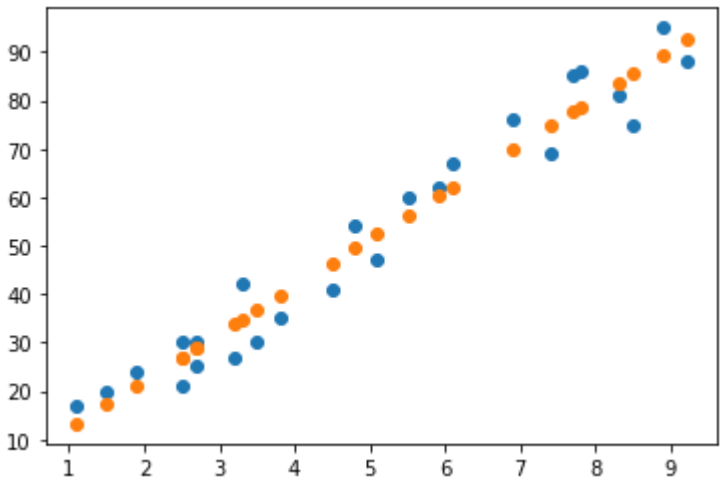
Predictions

In [24]:

```
plt.scatter(data['Hours'],data['Scores'])
plt.scatter(data['Hours'],data['Prediction'])
plt.plot()
```

Out[24]:

```
[]
```



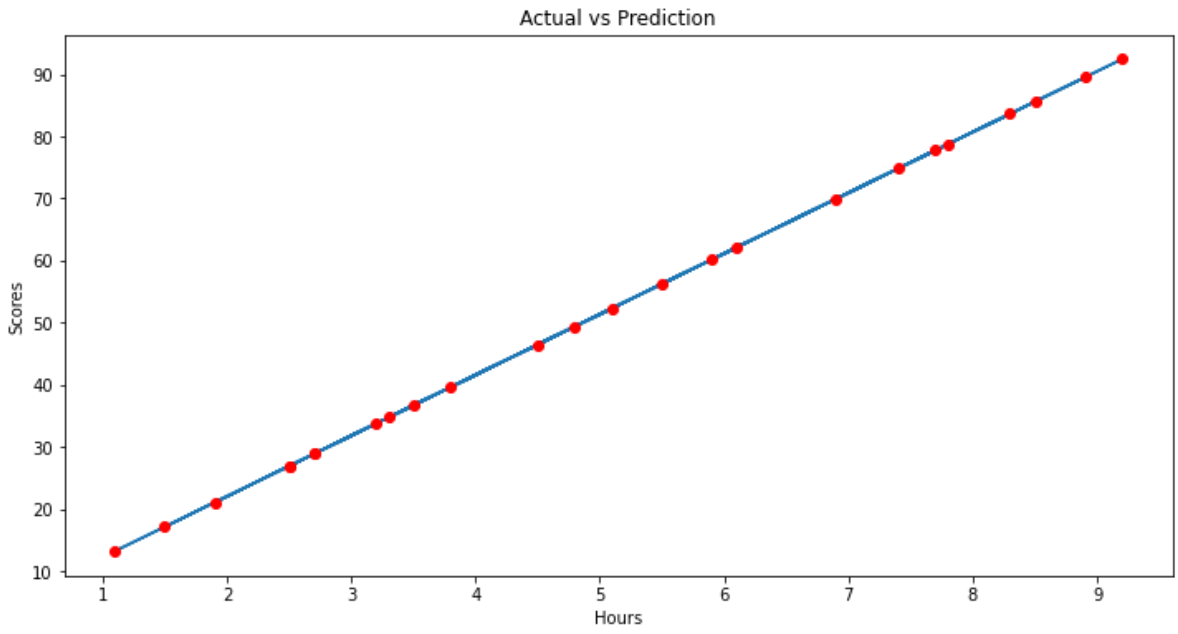
OLS Model

In [35]:

```
plt.figure(figsize=(12,6))
plt.plot(data['Hours'],data['Prediction'])
plt.plot(data['Hours'],data['Prediction'],'ro')
plt.title('Actual vs Prediction')
plt.xlabel('Hours')
plt.ylabel('Scores')
```

Out[35]:

```
Text(0, 0.5, 'Scores')
```



It could be observed that the prediction shpws that for 9.25 hours is marked at 92

Categorical prediction

In [37]:

```
cut_off = 40
```

In [39]:

```
data['Passed?'] = data['Scores']>=40
```

In [43]:

```
data.head()
```

Out[43]:

	Hours	Scores	Prediction	Passed?
0	2.5	21	26.923182	False
1	5.1	47	52.340271	True
2	3.2	27	33.766244	False
3	8.5	75	85.578002	True
4	3.5	30	36.698985	False

Plotting the given data's results

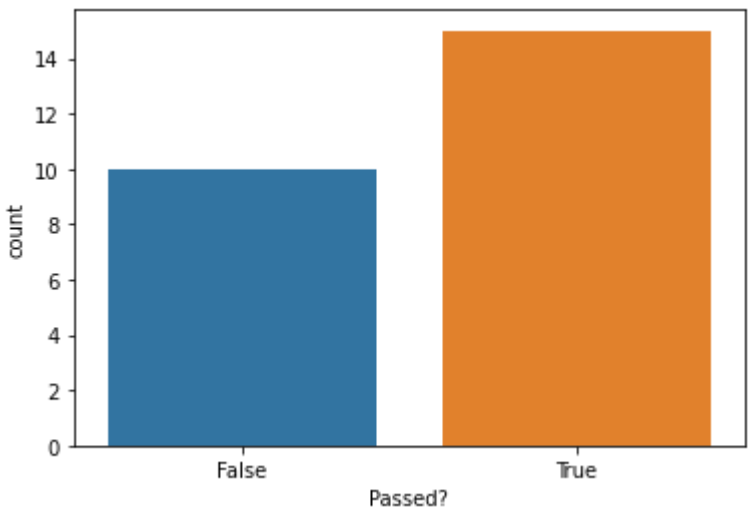
In [46]:

```
import seaborn as sns
sns.countplot(data['Passed?'])
```

```
C:\Users\Prathyu Lachireddy\anaconda3\lib\site-packages\seaborn\_decorators.
py:36: FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or misin
terpretation.
  warnings.warn(
```

Out[46]:

```
<AxesSubplot:xlabel='Passed?', ylabel='count'>
```



In [48]:

```
get_results = [[9.25]]
get_results
```

Out[48]:

```
[[9.25]]
```

In [ ]:

```

```