Name: RUTUJA SOHANI

Domain: DATA SCIENCE AND BUSINESS ANALYTICS

Task 1: PREDICTION USING SUPERVISED ML

Language:Python

Dataset Link:http://bit.ly/w-data

In [14]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
from sklearn.linear_model import LinearRegression as lr
from sklearn.model_selection import train_test_split as tts
import seaborn as sns
```

In [19]:
```
# reading the data
url="http://bit.ly/w-data'
data = pd.read_csv(url)
df=pd.read_csv(url)
data.head(10)
```

Out[19]:

	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

In [5]:
```
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: 464.0 bytes
```

In [6]:
```
#to check whether any duplicate value or missing value is present or not
data.isnull().sum()
```

Out[6]:
```
Hours     0
Scores    0
dtype: int64
```

In [7]:
```
#analysis on data
data.describe()
```

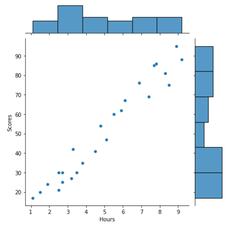
Out[7]:

	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

Plotting the distribution of scores

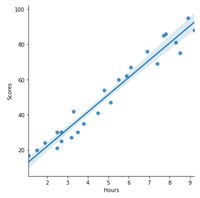
In [21]:
```
sns.jointplot(x='Hours',y='Scores', data=df)
plt.xlabel('Hours' )
plt.ylabel('Score')
```

Out[21]:
```
Text(336.9714285714286, 0.5, 'Score')
```



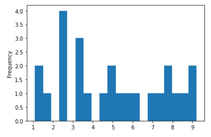
In [25]:
```
sns.lmplot(x='Hours',y='Scores',data=df,palette='Rainbow')
plt.xlabel('Hours')
plt.ylabel('Scores')
```

Out[25]:
```
Text(3.6749999999999997, 0.5, 'Scores')
```



In [28]:
```
df['Hours'].plot.hist(bins=20)
```

Out[28]:
```
<AxesSubplot:ylabel='Frequency'>
```



The data

In [30]:
```
from sklearn.model_selection import train_test_split
X = df.iloc[:, :-1].values
Y = df.iloc[:, 1].values
```

In [34]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,test_size=0.2, random_state=0)
```

Training the Data

In [37]:
```
from sklearn.linear_model import LinearRegression
Regression= LinearRegression()
Regression.fit(X_train,Y_train)
```

Out[37]:
```
LinearRegression()
```

In [38]:
```
Prediction =Regression.predict(X_test)
```

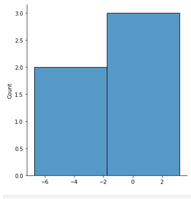
Evaluating the model

In [44]:
```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(Y_test, Prediction))
print('MSE:', metrics.mean_squared_error(Y_test, Prediction))
print('RMSE:', np.sqrt(metrics.mean_squared_error(Y_test, Prediction)))
```

```
MAE: 4.183859899002982
MSE: 21.598769307217456
RMSE: 4.647447612100373
```

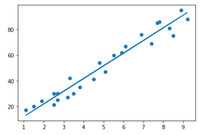
In [51]:
```
sns.displot(Y_test-Prediction,bins=2)
```

Out[51]:
```
<seaborn.axisgrid.FacetGrid at 0x92944a8>
```



In [55]:
```
Line = Regression.coef_*X+Regression.intercept_

# Plotting for the test data
plt.scatter(X, Y)
plt.plot(X, Line);
plt.show()
```



Making Predictions

In [56]:
```
Hours=9.25
Newprediction=Regression.predict([[9.25]])
```

In [57]:
```
Newprediction
```

Out[57]:
```
array([93.69173249])
```

Result –

In [66]:
```
print('\nHour Studied: 9.25\nPredicted Score: ',Newprediction)
```

```
Hour Studied: 9.25
Predicted Score:  [93.69173249]
```