In [12]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline
```

In [13]:
```
url="http://bit.ly/w-data"
```

In [14]:
```
data=pd.read_csv(url)
```

In [15]:
```
data.head(10)
```

Out[15]:

	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 [16]:
```
data.isnull().sum()
```

Out[16]:
```
Hours     0
Scores    0
dtype: int64
```

In [17]:
```
data.describe()
```

Out[17]:

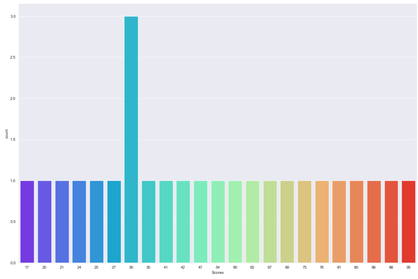
	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [18]:
```
data.info()
```

Out[18]:
```
<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
```

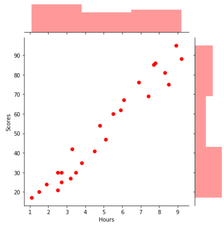
In [36]:
```
plt.figure(figsize=(18,12))
sns.set_style("darkgrid")
sns.countplot("Scores",data=data,palette="rainbow")
```

Out[36]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24a4ed2ffc8>
```



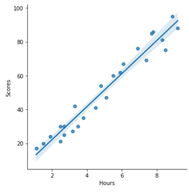
In [29]:
```
sns.jointplot(x="Hours",y="Scores",data=data,kind="scatter",color="red")
```

Out[29]:
```
<seaborn.axisgrid.JointGrid at 0x24a4d17e048>
```



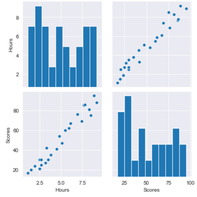
In [31]:
```
sns.lmplot(x='Hours',y='Scores',data=data)
```

Out[31]:
```
<seaborn.axisgrid.FacetGrid at 0x24a4d9e3c08>
```



In [39]:
```
sns.pairplot(data)
```

Out[39]:
```
<seaborn.axisgrid.PairGrid at 0x24a4f708f08>
```



In [57]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

In [58]:
```
from sklearn.model_selection import train_test_split
```

In [71]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [72]:
```
from sklearn.linear_model import LinearRegression
```

In [73]:
```
lm=LinearRegression()
```

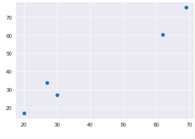
In [74]:
```
lm.fit(X_train,y_train)
```

Out[74]:
```
LinearRegression()
```

In [75]:
```
predictions=lm.predict(X_test)
```

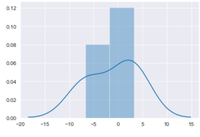
In [103]:
```
plt.scatter(y_test,predictions)
```

Out[103]:
```
<matplotlib.collections.PathCollection at 0x24a50395208>
```



In [101]:
```
sns.distplot((y_test-predictions),bins=2)
```

Out[101]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24a5190fe08>
```



In [86]:
```
from sklearn import metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test,predictions))
```

```
Mean Absolute Error: 4.183859899002982
```

In [92]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': predictions})
df
```

Out[92]:

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

In [95]:
```
df=lm.predict([[9.25]])
```

In [98]:
```
print("The predicted Score of a Student who reads 9.25 hpours is {}".format(df))
```

```
The predicted Score of a Student who reads 9.25 hpours is [93.69173249]
```