In [19]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

#to ignore warnings
import warnings as wg
wg.filterwarnings("ignore")
```

In [21]:
```
#Reading dataset from remote link
url="http://bit.ly/w-data"
data=pd.read_csv(url)
```

In [22]:
```
print(data.shape)
data.head()
```

```
(25, 2)
```

Out[22]:

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

In [23]:
```
data.describe()
```

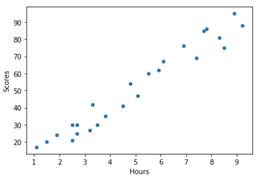
Out[23]:

	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 [24]:
```
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 [25]:
```
data.plot(kind='scatter',x='Hours',y='Scores');
plt.show()
```



In [26]:
```
data.corr(method='pearson')
```

Out[26]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

In [27]:
```
data.corr(method='spearman')
```

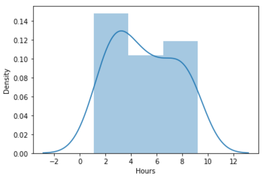
Out[27]:

	Hours	Scores
Hours	1.000000	0.971891
Scores	0.971891	1.000000

In [28]:
```
hours=data['Hours']
scores=data['Scores']
```

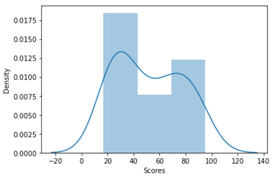
In [29]:
```
sns.distplot(hours)
```

Out[29]:
```
<AxesSubplot:xlabel='Hours', ylabel='Density'>
```



In [30]:
```
sns.distplot(scores)
```

Out[30]:
```
<AxesSubplot:xlabel='Scores', ylabel='Density'>
```



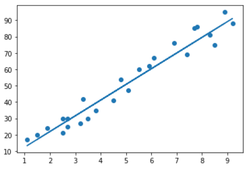
In [31]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

In [35]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=50)
```

In [36]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(X_train, y_train)
```

Out[36]:
```
LinearRegression()
```

In [37]:
```
m=reg.coef_
c=reg.intercept_
line=m*X+c
plt.scatter(X,y)
plt.plot(X, line);
plt.show()
```



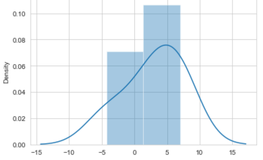
In [38]:
```
y_pred=reg.predict(X_test)
```

In [39]:
```
actual_predicted=pd.DataFrame({'Target':y_test,'Predicted':y_pred})
actual_predicted
```

Out[39]:

	Target	Predicted
0	95	88.211394
1	30	28.718453
2	76	69.020122
3	35	39.273652
4	17	13.365436

In [40]:
```
sns.set_style('whitegrid')
sns.distplot(np.array(y_test-y_pred))
plt.show()
```



In [41]:
```
h=9.25
s=reg.predict([[h]])
print("If a student studies for {} hours per day he/she will score {} % in exam.".format(h,s))
```

```
If a student studies for 9.25 hours per day he/she will score [91.56986604] % in exam.
```

In [45]:
```
from sklearn import metrics
from sklearn.metrics import r2_score
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, y_pred))
print('R2 Score:',r2_score(y_test,y_pred))
```

```
Mean Absolute Error: 4.591649530630285
R2 Score: 0.971014141329942
```

In [ ]:

In [ ]: