In [1]:

```
#Importing library files
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
#Loading data in DataFrame
df = pd.read_csv("http://bit.ly/w-data")
df.head()
```

Out[2]:

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

In [3]:

```
df.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 [4]:

```
df.describe()
```

Out[4]:

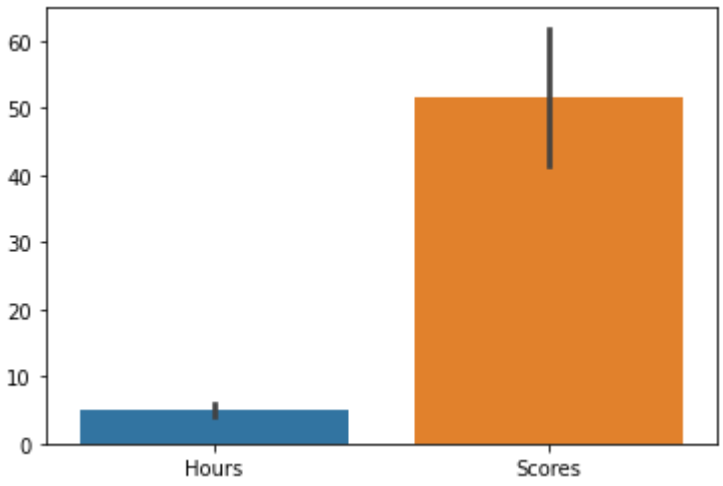
	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [5]:

```
#Barplot
sns.barplot(data=df[["Hours","Scores"]])
```

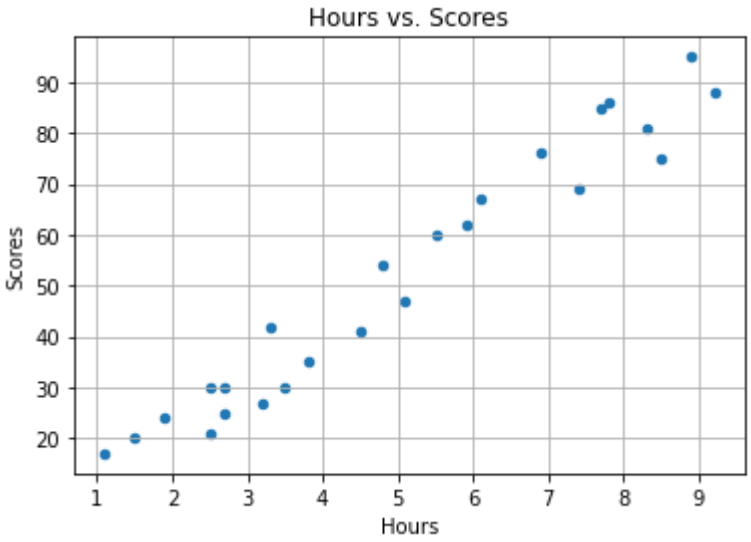
Out[5]:

```
<AxesSubplot:>
```



In [6]:

```
#Scatterplot
df.plot.scatter(x="Hours",y="Scores")
plt.title("Hours vs. Scores")
plt.grid()
plt.show()
```



In [7]:

```
#Preparing the data
X = df.iloc[:, :-1].values
y = df.iloc[:, 1].values
#Splitting our Dataset into Train and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state =
```

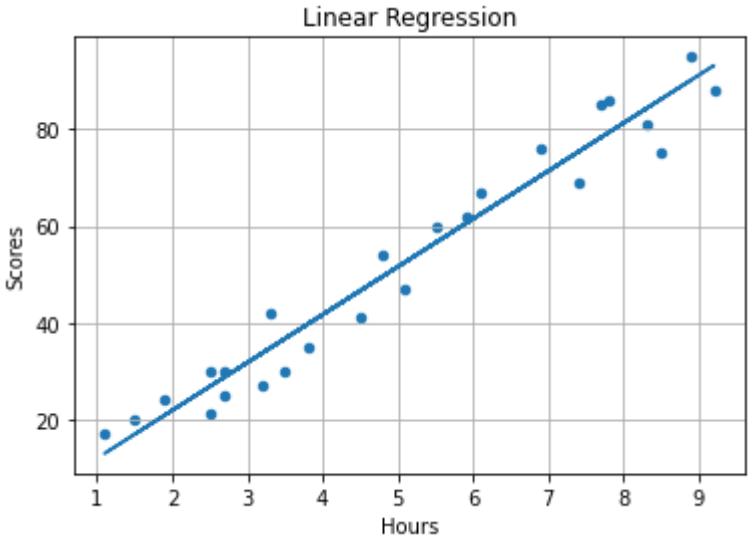
In [8]:

```
#Training the Algorithm
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
print("Training Successful.")
```

```
Training Successful.
```

In [9]:

```
line = regressor.coef_*X+regressor.intercept_
df.plot.scatter(x="Hours",y="Scores")
plt.plot(X, line);
plt.title('Linear Regression')
plt.grid()
plt.show()
```



In [10]:

```
#Predicting the scores
y_pred = regressor.predict(X_test)
print(y_pred)
```

```
[16.88414476 33.73226078 75.357018   26.79480124 60.49103328]
```

In [11]:

```
#Comparing Actual Score vs. Predicted Score
df_compare = pd.DataFrame({"Actual Score":y_test,"Predicted Score":y_pred})
df_compare
```

Out[11]:

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

In [12]:

```
#Predicting the score for a student who studies 9.25 hrs/ day
my_hours = np.array([[9.25]])
my_pred = regressor.predict(my_hours)
print("No of Hours = {}".format(my_hours[0][0]))
print("Predicted Score = {}".format(my_pred[0]))
```

```
No of Hours = 9.25
Predicted Score = 93.69173248737538
```

In [13]:

```
#Evaluating the Model
import sklearn.metrics as metrics
variance=metrics.explained_variance_score(y_test, y_pred)
mae=metrics.mean_absolute_error(y_test, y_pred)
mse=metrics.mean_squared_error(y_test, y_pred)
msle=metrics.mean_squared_log_error(y_test, y_pred)
mae=metrics.median_absolute_error(y_test, y_pred)
r2=metrics.r2_score(y_test, y_pred)
print('Explained Variance: ', round(variance,5))
print('Mean squared log error: ', round(msle,5))
print('Mean absolute error: ', round(mae,5))
print('Mean squared error: ', round(mse,5))
print('Coefficient of determination: ', round(r2,5))
print('Root mean squared error: ', round(np.sqrt(mse),5))
```

```
Explained Variance:  0.94828
Mean squared log error:  0.01845
Mean absolute error:  3.2052
Mean squared error:  21.59877
Coefficient of determination:  0.94549
Root mean squared error:  4.64745
```

In [ ]:

```

```