In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df=pd.read_csv(r"C:\Users\Seth-Ese Joey\Downloads\archive (4)\score.csv")
df.head()
```

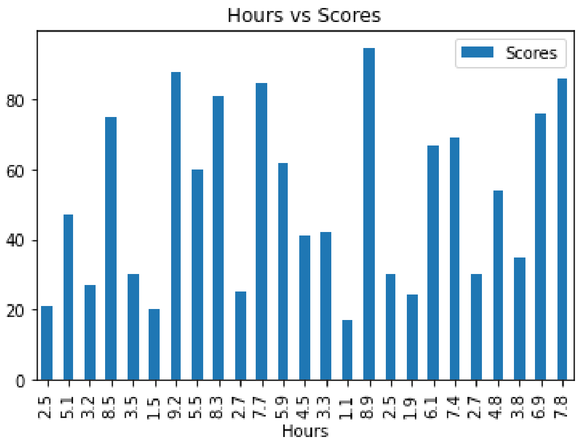
Out[2]:

	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.isnull().sum()
```

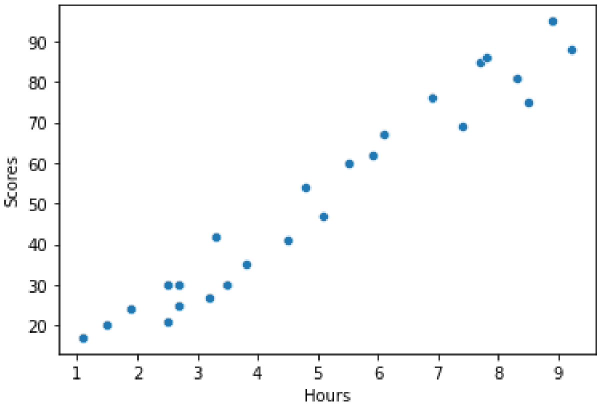
Out[3]:
```
Hours     0
Scores    0
dtype: int64
```

In [39]:
```
#Plotting the hours vs score graph
df.plot(x='Hours',y='Scores',kind='bar')
plt.title("Hours vs Scores")
plt.show()
```

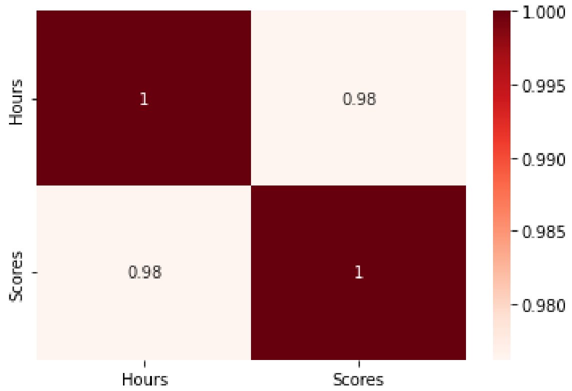


In [5]:
```
sns.scatterplot(x='Hours',y='Scores',data=df)
```

Out[5]:
```
<AxesSubplot:xlabel='Hours', ylabel='Scores'>
```



In [6]:
```
cor=df.corr()
sns.heatmap(cor,annot=True,cmap='Reds')
plt.show()
```



In [7]:
```
# the score column
df_marks=df['Scores']
```

In [10]:
```
#droping score column(dependent variable) for prediction
z=df.drop(['Scores'],axis=1)
```

In [11]:
```
#Training and carring out a test
X_train,X_test,y_train,y_test=train_test_split(z,df_marks,test_size=0.30,random_state=4)
X_train.head()
```

Out[11]:

	Hours
2	3.2
4	3.5
19	7.4
17	1.9
15	8.9

In [15]:
```
#Applying linear regression
lin_reg=LinearRegression()
lin_reg.fit(X_train,y_train)
```

```
Out[15]:LinearRegression()
```

In [18]:
```
#Checking for accuracy
print(f'The accuracy score is => {round(lin_reg.score(X_train,y_train),2)*100}%')
```

```
The accuracy score is => 96.0%
```

In [36]:
```
X_test
```

Out[36]:

	Hours
22	3.8
21	4.8
0	2.5
3	8.5
12	4.5
10	7.7
6	9.2
11	5.9

In [35]:
```
lin_reg.predict(X_test)
```

```
Out[35]:array([40.5054997 , 50.46249282, 27.56140864, 87.30336737, 47.47539489,
       79.33777287, 94.27326256, 61.41518526])
```

In [40]:
```
pred=[]
for i in range(0,len(X_train.columns)):
    v=float(input(f'enter study {X_train.columns[i]}'))
    pred.append(v)
prediction=lin_reg.predict([pred])
print(f'students who study {pred} hours will score {prediction}')
```

```
enter study Hours7
students who study [7.0] hours will score [72.36787769]
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [ ]: