By:- subhayan mukherjee

In [5]:

```
# import pandas as pd
import numpy as np
import pandas as pd
df = pd.read_csv('http://bit.ly/w-data')
df
```

Out[5]:

	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
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

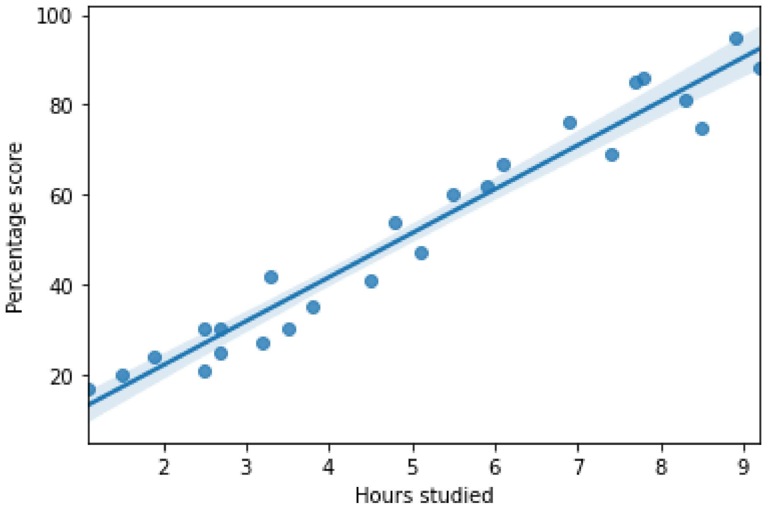
In [4]:
```
df.describe()
```

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 [37]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.regplot(x='Hours',y='Scores',data=df)
print('This is the regression line with 95% confidence interval for that regressi
plt.xlabel('Hours studied')
plt.ylabel('Percentage score')
plt.show()
```

```
This is the regression line with 95% confidence interval for that regression:
```



In [4]:

```
#for checking null values
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 [9]:

```
print('min score:', df['Hours'].min())
print('max score:', df['Hours'].max())
```

```
min score: 1.1
max score: 9.2
```

In [11]:

```
print('min score:-', df['Scores'].min())
print('max score:-', df['Scores'].max())
```

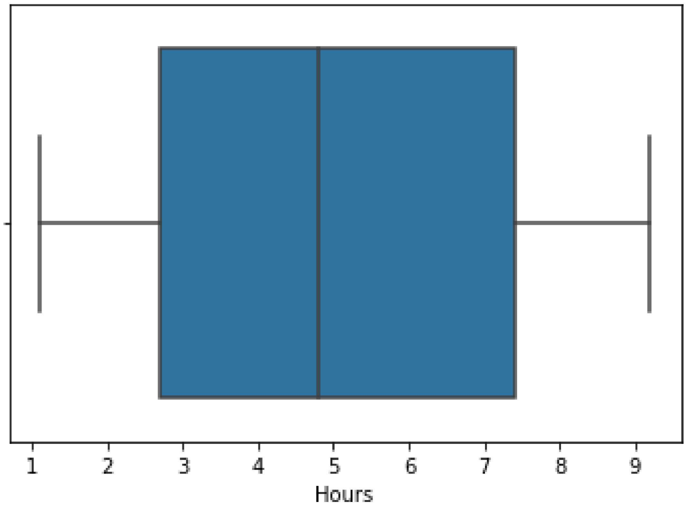
```
min score:- 17
max score:- 95
```

In [14]:

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('http://bit.ly/w-data')
sns.boxplot(df["Hours"])
print('There is no outlier present')
```

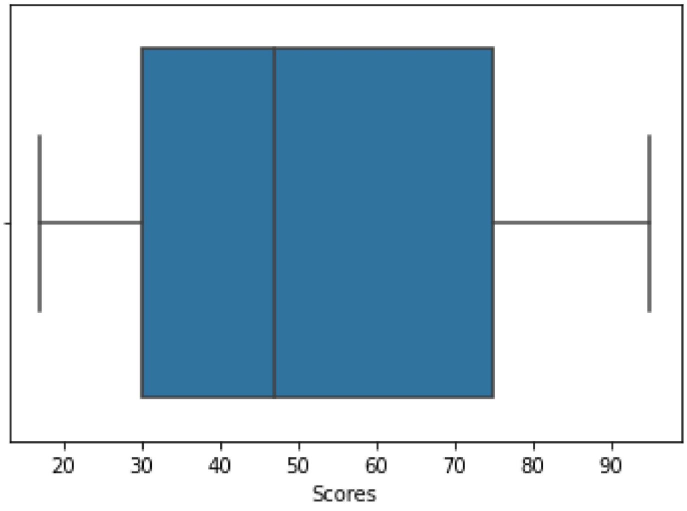
```
There is no outlier present
```

```
C:\Users\Subhayan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12, th
e only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



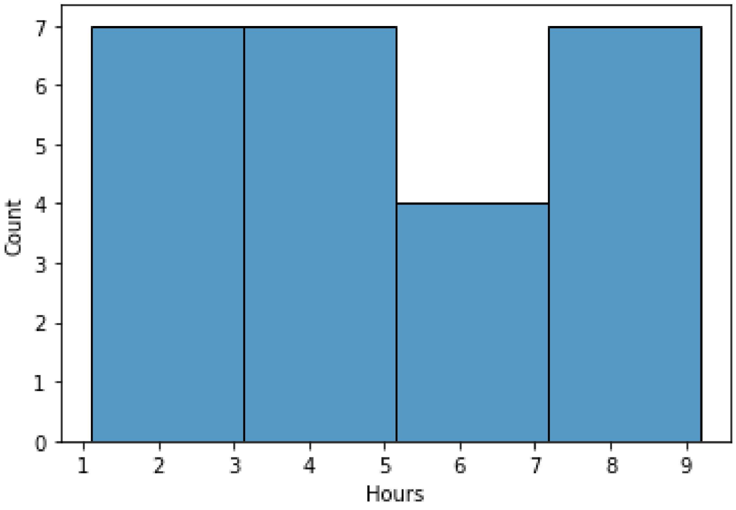
In [16]:
```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('http://bit.ly/w-data')
sns.boxplot(df["Scores"])
print('There is no outlier present')
```

```
There is no outlier present
```



In [18]:
```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('http://bit.ly/w-data')
sns.histplot(df["Hours"], bins=4)
print('There is no outlier present')
```

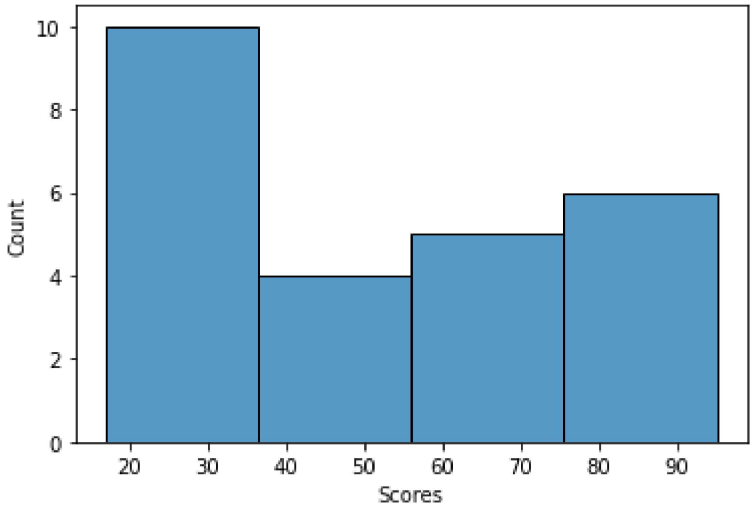
```
There is no outlier present
```



In [19]:

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('http://bit.ly/w-data')
sns.histplot(df["Scores"], bins=4)
print('There is no outlier present')
```

```
There is no outlier present
```



In [20]:

```
#The hours and Scores are distributed normally and we can perform linear regressi
```

In [21]:

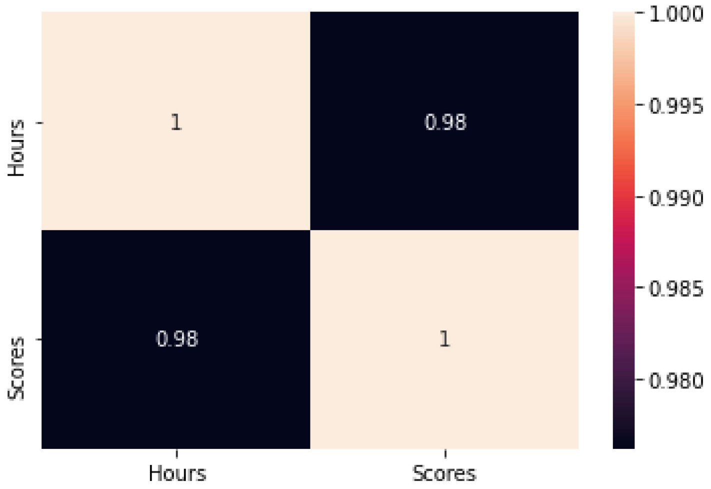
```
df = pd.read_csv('http://bit.ly/w-data')
column_1 = df["Hours"]
column_2 = df["Scores"]
correlation = column_1.corr(column_2)
correlation
```

Out[21]:

```
0.9761906560220887
```

In [34]:

```
%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
sns.heatmap(df.corr(),annot=True)
plt.show()
print('The correlation value is greater zero')
```



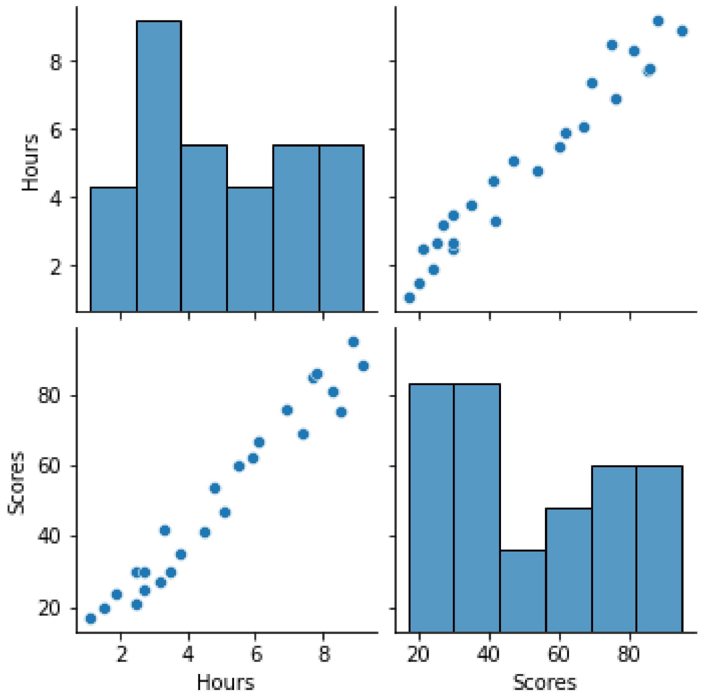
```
The correlation value is greater zero
```

In [38]:

```
sns.pairplot(df)
```

Out[38]:

```
<seaborn.axisgrid.PairGrid at 0x1f4be2d8520>
```



In [39]:

```
from sklearn.model_selection import train_test_split
```

In [41]:
```
x=df.iloc[:,:-1].values
y=df.iloc[:,1].values
x_train, x_test, y_train, y_test= train_test_split(x, y,train_size=0.60,test_size
```

In [42]:
```
from sklearn.linear_model import LinearRegression
model= LinearRegression()
model.fit(x_train, y_train)
```

Out[42]:
```
LinearRegression()
```

In [43]:
```
y_pred = model.predict(x_test)
y_pred
```

Out[43]:
```
array([15.9477618 , 32.77394723, 74.344523  , 25.84551793, 59.49788879,
       38.71260091, 19.90686425, 78.30362545, 69.39564493, 11.98865934])
```

In [44]:
```
print('Test Score')
print(model.score(x_test, y_test))
print('Training Score')
print(model.score(x_train, y_train))
```

```
Test Score
0.956640847232559
Training Score
0.9440108159733135
```

In [48]:
```
print('Score of student who studied for 9.25 hours a day is:-', model.predict([[
```

```
Score of student who studied for 9.25 hours a day is: [92.65537185]
```

summary:-
The dataset with 2 attributes Hours and Scores contains no null values. With the help of numpy, pandas, matplotlib, seaborn we have done the data analysis and visualization. e performed Linear Regression operation on the given dataset and the model had an accuracy of 95%. Thus, the model could predict the score for a student who studies for 9.25hrs in a day which is 92.65%.

In [ ]: