In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:

```
df=pd.read_csv("Student_Performance (2).csv")
df
```

Out[5]:

	hours	scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	Yes	9	1	91
1	4	82	No	4	2	65
2	8	51	Yes	7	2	45
3	5	52	Yes	5	2	36
4	7	75	No	8	5	66
...	...	...	...	...	...	...
9995	1	49	Yes	4	2	23
9996	7	64	Yes	8	5	58
9997	6	83	Yes	8	5	74
9998	9	97	Yes	7	0	95
9999	7	74	No	8	1	64

10000 rows × 6 columns

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column                            Non-Null Count  Dtype 
---  ------                            --------------  ----- 
 0   hours                             10000 non-null  int64 
 1   scores                            10000 non-null  int64 
 2   Extracurricular Activities        10000 non-null  object
 3   Sleep Hours                       10000 non-null  int64 
 4   Sample Question Papers Practiced  10000 non-null  int64 
 5   Performance Index                 10000 non-null  int64 
dtypes: int64(5), object(1)
memory usage: 468.9+ KB
```

In [7]:

```
df=df.drop("Extracurricular Activities",axis=1)
df
```

Out[7]:

	hours	scores	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	9	1	91
1	4	82	4	2	65
2	8	51	7	2	45
3	5	52	5	2	36
4	7	75	8	5	66
...	...	...	...	...	...
9995	1	49	4	2	23
9996	7	64	8	5	58
9997	6	83	8	5	74
9998	9	97	7	0	95
9999	7	74	8	1	64

10000 rows × 5 columns

In [8]:

```
x=df.iloc[:,0:-1]
x
```

Out[8]:

	hours	scores	Sleep Hours	Sample Question Papers Practiced
0	7	99	9	1
1	4	82	4	2
2	8	51	7	2
3	5	52	5	2
4	7	75	8	5
...	...	...	...	...
9995	1	49	4	2
9996	7	64	8	5
9997	6	83	8	5
9998	9	97	7	0
9999	7	74	8	1

10000 rows × 4 columns

In [9]:

```
y=df["Performance Index"]
y
```

Out[9]:

```
0       91
1       65
2       45
3       36
4       66
        ..
9995    23
9996    58
9997    74
9998    95
9999    64
Name: Performance Index, Length: 10000, dtype: int64
```

In [10]:

```
# train_test_split
```

In [11]:

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

In [12]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_sta
```

In [13]:

```
print("x_train shape=",x_train.shape)
print("y_train shape=",y_train.shape)
print("x_test shape=",x_test.shape)
print("y_test shape=",y_test.shape)
```

```
x_train shape= (7000, 4)
y_train shape= (7000,)
x_test shape= (3000, 4)
y_test shape= (3000,)
```

In [14]:

```
# ml training
# linear regression for multi variate
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[14]:

```
▾ LinearRegression
LinearRegression()
```

In [15]:

```
lr.score(x_test,y_test)
```

Out[15]:

```
0.9890524179572492
```

In [16]:

```
y_predicted=lr.predict(x_test)
y_predicted
```

Out[16]:

```
array([69.2605599 , 63.08380235, 68.2720853 , ..., 72.14148532,
       47.67166147, 86.79903125])
```

In [17]:

```
x_test
```

Out[17]:

	hours	scores	Sleep Hours	Sample Question Papers Practiced
3465	6	80	8	3
2574	7	70	8	9
2223	6	80	4	8
4520	3	53	6	7
9998	9	97	7	0
...	...	...	...	...
8106	7	87	8	2
4725	6	93	5	9
3000	8	78	6	4
3486	2	70	6	8
5202	6	96	9	7

3000 rows × 4 columns

In [18]:

```
y_test
```

Out[18]:

```
3465    68
2574    62
2223    70
4520    35
9998    95
        ..
8106    79
4725    82
3000    72
3486    50
5202    87
Name: Performance Index, Length: 3000, dtype: int64
```

In [19]:

```
x_test["actual"]=y_test
x_test
```

Out[19]:

	hours	scores	Sleep Hours	Sample Question Papers Practiced	actual
3465	6	80	8	3	68
2574	7	70	8	9	62
2223	6	80	4	8	70
4520	3	53	6	7	35
9998	9	97	7	0	95
...	...	...	...	...	...
8106	7	87	8	2	79
4725	6	93	5	9	82
3000	8	78	6	4	72
3486	2	70	6	8	50
5202	6	96	9	7	87

3000 rows × 5 columns

In [20]:

```
x_test["predicted"]=y_predicted
x_test
```

Out[20]:

	hours	scores	Sleep Hours	Sample Question Papers Practiced	actual	predicted
3465	6	80	8	3	68	69.260560
2574	7	70	8	9	62	63.083802
2223	6	80	4	8	70	68.272085
4520	3	53	6	7	35	33.027399
9998	9	97	7	0	95	94.047217
...	...	...	...	...	...	...
8106	7	87	8	2	79	79.042100
4725	6	93	5	9	82	82.181626
3000	8	78	6	4	72	72.141485
3486	2	70	6	8	50	47.671661
5202	6	96	9	7	87	86.799031

3000 rows × 6 columns

In [21]:

```
x_test.head()
```

Out[21]:

	hours	scores	Sleep Hours	Sample Question Papers Practiced	actual	predicted
3465	6	80	8	3	68	69.260560
2574	7	70	8	9	62	63.083802
2223	6	80	4	8	70	68.272085
4520	3	53	6	7	35	33.027399
9998	9	97	7	0	95	94.047217

In [22]:

```
lr.predict([[7,90,7,6]])
```

```
C:\Users\Admin\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X doe
s not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[22]:

```
array([82.37593655])
```

In [23]:

```
lr.coef_
```

Out[23]:

```
array([2.84924947, 1.01774177, 0.48699588, 0.19190178])
```

In [24]:

```
lr.intercept_
```

Out[24]:

```
-33.7259509956904
```

In [25]:

```
def predicted(h,s,sh,sqp):
    k=2.84924947*h+1.01774177*s+0.48699588*sh+0.19190178*sqp-33.72595099569034
    return k
```

In [26]:

```
predicted(7,90,7,6)
```

Out[26]:

```
82.37593643430966
```

In [ ]: