Data Collection,Cleaning,Manipulation,Visualization

In [1]:

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

In [2]:

```
df=pd.read_excel(r"C:\Users\user\Desktop\Datasets\student_info.xlsx")
df
```

Out[2]:

	study_hours	student_marks	Result
0	6.83	78.50	Pass
1	6.56	76.74	Pass
2	NaN	78.68	Pass
3	5.67	71.82	Fail
4	8.67	84.19	Pass
...	...	...	...
195	7.53	81.67	Pass
196	8.56	84.68	Pass
197	8.94	86.75	Pass
198	6.60	78.05	Pass
199	8.35	83.50	Pass

200 rows × 3 columns

In [3]:

```
df.isnull().sum()
```

Out[3]:

```
study_hours      6
student_marks    3
Result           2
dtype: int64
```

In [4]:

```
df['study_hours'].mean()
```

Out[4]:

```
7.006185567010309
```

In [5]:

```
df['study_hours']=df['study_hours'].fillna(df['study_hours'].mean())
df['student_marks']=df['student_marks'].fillna(df['student_marks'].mean())
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
study_hours      0
student_marks    0
Result           2
dtype: int64
```

In [7]:

```
df.dropna(inplace=True)
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
study_hours      0
student_marks    0
Result           0
dtype: int64
```

In [9]:

```
df.shape
```

Out[9]:

```
(198, 3)
```

In [10]:

```
df2=df.groupby('Result').count()
```

In [11]:

```
df2
```

Out[11]:

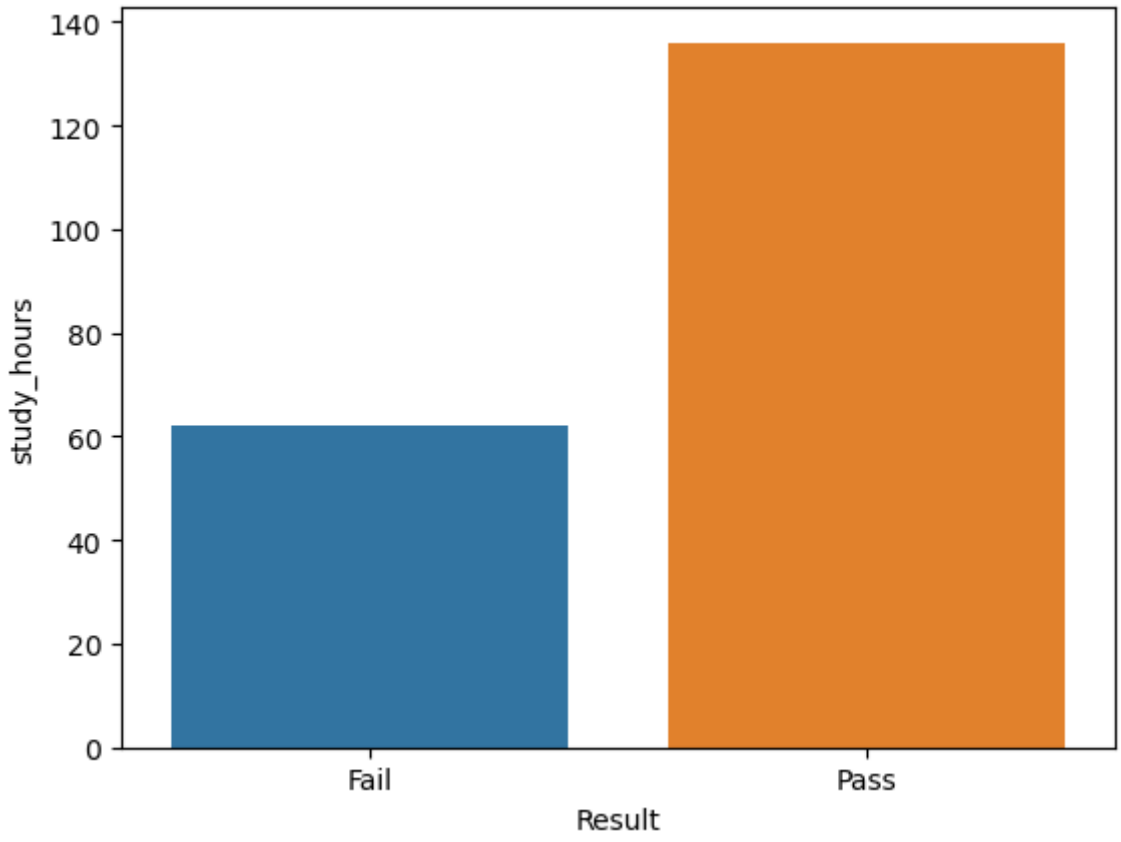
Result	study_hours	student_marks
Fail	62	62
Pass	136	136

In [12]:

```
sns.barplot(x=df2.index,y="study_hours",data=df2)
```

Out[12]:

```
<AxesSubplot:xlabel='Result', ylabel='study_hours'>
```



Regression---->Linear Regression

#Data Prediction

In [13]:

```
df
```

Out[13]:

	study_hours	student_marks	Result
0	6.830000	78.50	Pass
1	6.560000	76.74	Pass
2	7.006186	78.68	Pass
3	5.670000	71.82	Fail
4	8.670000	84.19	Pass
...	...	...	...
195	7.530000	81.67	Pass
196	8.560000	84.68	Pass
197	8.940000	86.75	Pass
198	6.600000	78.05	Pass
199	8.350000	83.50	Pass

198 rows × 3 columns

In [15]:

```
#x is indepedent variable
#y is depedent variable
x=df[['study_hours']]
y=df[['student_marks']]
```

In [16]:

```
print(x)
print(y)
```

```
     study_hours
0       6.830000
1       6.560000
2       7.006186
3       5.670000
4       8.670000
..           ...
195     7.530000
196     8.560000
197     8.940000
198     6.600000
199     8.350000

[198 rows x 1 columns]
     student_marks
0            78.50
1            76.74
2            78.68
3            71.82
4            84.19
..             ...
195          81.67
196          84.68
197          86.75
198          78.05
199          83.50

[198 rows x 1 columns]
```

In [17]:

```
#Spiliting Dataset
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [22]:

```
#Shape of All Datasets
print("Shape of All Dataset : ", df.shape)
print("Shape of x_train : ",x_train.shape)
print("Shape of x_test : ",x_test.shape)
print("Shape of y_train : ",y_train.shape)
print("Shape of y_test : ",y_test.shape)
```

```
Shape of All Dataset :  (198, 3)
Shape of x_train :  (138, 1)
Shape of x_test :  (60, 1)
Shape of y_train :  (138, 1)
Shape of y_test :  (60, 1)
```

In [24]:

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
```

Out[24]:

```
LinearRegression()
```

In [25]:

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

Out[25]:

```
0.9154499992224483
```

In [27]:

```
y_pred=model.predict(x_test)
#Predicted Result
y_pred[0:5]
```

Out[27]:

```
array([[84.89864714],
       [70.46496814],
       [79.12517554],
       [70.85506757],
       [77.95487724]])
```

In [28]:

```
#Actual Result
y_test.head()
```

Out[28]:

	student_marks
89	84.60
78	70.05
74	77.59
8	70.66
46	77.46

In [32]:

```
student=np.array([7.00]).reshape(1,-1)
student
```

Out[32]:

```
array([[7.]])
```

In [33]:

```
prediction=model.predict(student)
prediction
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarn
ing: X does not have valid feature names, but LinearRegression was fitte
d with feature names
  warnings.warn(
```

Out[33]:

```
array([[77.9158673]])
```

In [ ]:

```

```