



Matplotlib

- Matplotlib is a plotting library for the Python programming language and easy to get started for simple plots .It support for custom labels and texts

In [1]:

```
import matplotlib.pyplot as plt
```

In [2]:

```
import numpy as np
import pandas as pd
```

In [3]:

```
df = pd.DataFrame({
    'StudyHours': [10, 11.5, 9, 16, 9.25, 1, 11.5, 9, 8.5, 14.5, 15.5, 13.75, 9, 8, 15.5],
    'Grade': [50, 50, 47, 97, 49, 3, 53, 42, 26, 74, 82, 62, 37, 119, 165],
    'Gender': ["M", "F", "M", "M", "F", "M", "M", "F", "M", "F", "M", "F", "M", "F", "M"],
    'no_of_assignments': [1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4]})

df
```

Out[3]:

	StudyHours	Grade	Gender	no_of_assignments
0	10.00	50	M	1
1	11.50	50	F	1
2	9.00	47	M	1
3	16.00	97	M	1
4	9.25	49	F	2
5	1.00	3	M	2
6	11.50	53	M	2
7	9.00	42	F	3
8	8.50	26	M	3
9	14.50	74	F	3
10	15.50	82	M	3
11	13.75	62	F	3
12	9.00	37	M	4
13	8.00	119	F	4
14	15.50	165	M	4

Histogram

- In histograms, x axis contains a variable and y axis will be a frequency of that variable

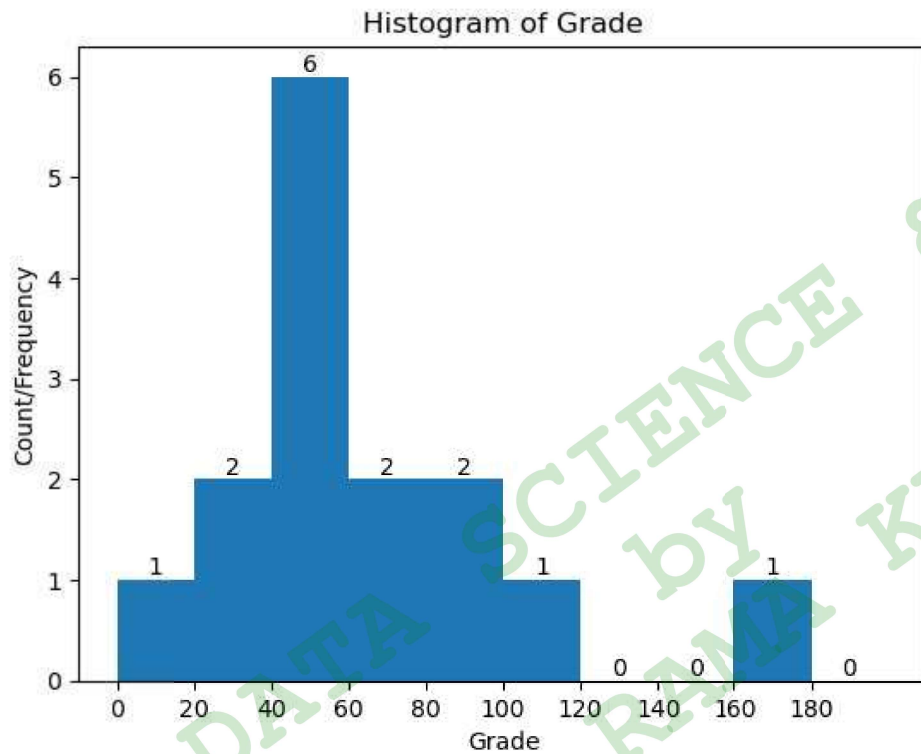
Important Paramemeters in Histogram

- data



In [4]:

```
n,bins,patches = plt.hist(df['Grade'],bins=10,range=(0,200))
plt.bar_label(patches)           # count of each bar
plt.xlabel("Grade")              # Set the label for the x-axis
plt.ylabel("Count/Frequency")    # Set the label for the y-axis
plt.title("Histogram of Grade")  # Set title of plot
plt.xticks(list(range(0,200,20))) # x-axis values presentation
plt.show()
```



In [5]:

```
df["Grade"].skew()
```

Out[5]:

1.218038125849298

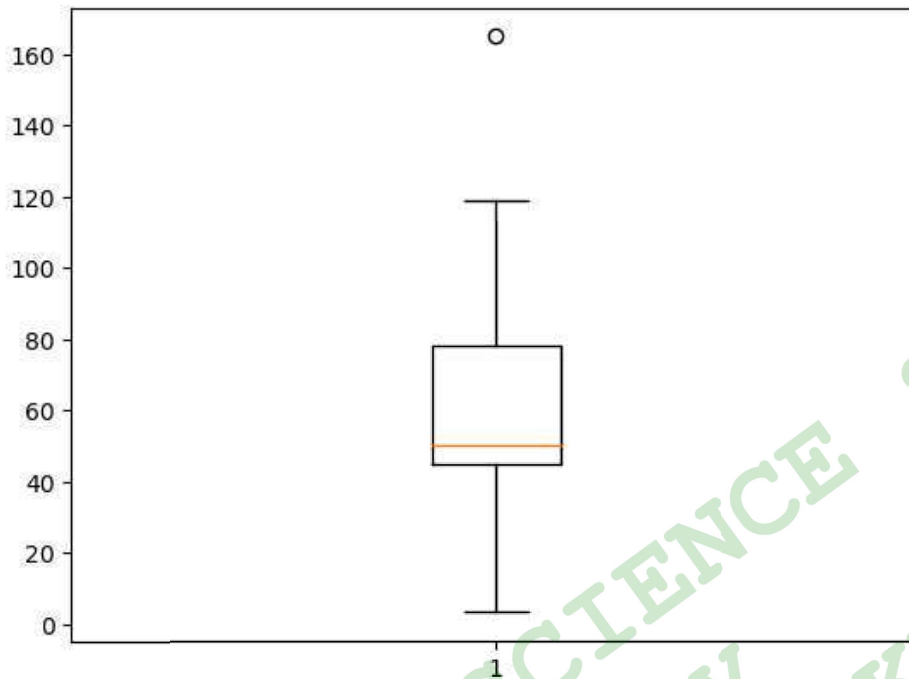
Box Plot

- **Use:** to check, the given data has outliers or not



In [6]:

```
plt.boxplot(df["Grade"])
plt.show()
```



In [7]:

```
Q3 = df["Grade"].quantile(0.75)
Q1 = df["Grade"].quantile(0.25)
IQR = Q3 - Q1
upper_limit = Q3 + (1.5 * IQR)
lower_limit = Q1 - (1.5 * IQR)

df[(df["Grade"] > upper_limit) | (df["Grade"] < lower_limit)]
```

Out[7]:

StudyHours	Grade	Gender	no_of_assignments
14	15.5	165	M
			4

Scatter Plot

- Marking the data points on the graph
- **Use:** to check 1. linearity, 2. Direction, 3. Strength

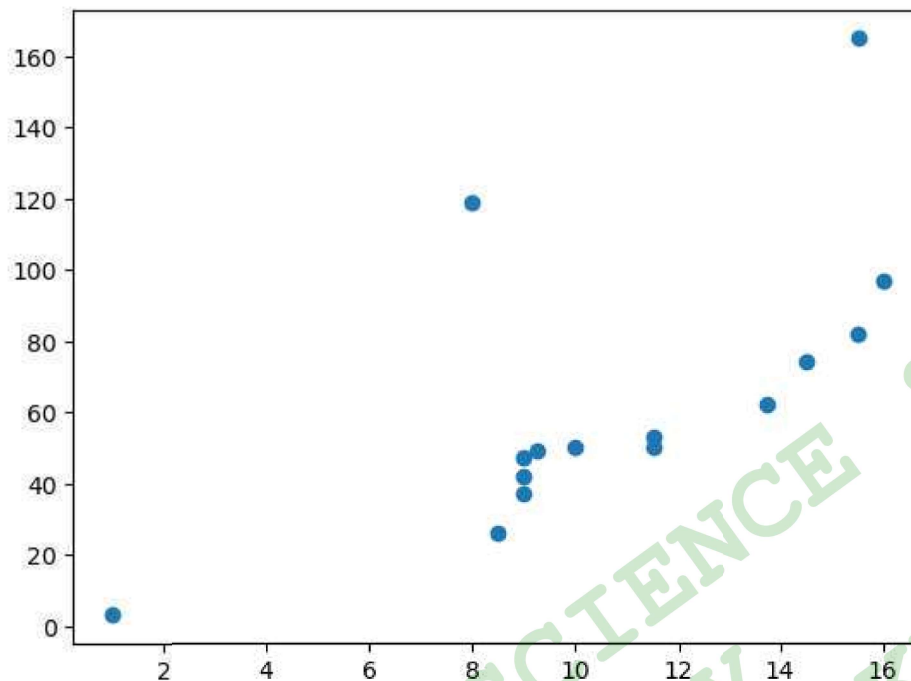
Important Paramemeters in Scatterplot

- X-axis values
- y-axis values
- color (default=blue)
- marker (default = "o")



In [8]:

```
plt.scatter(x=df['StudyHours'],y=df['Grade'])  
plt.show()
```

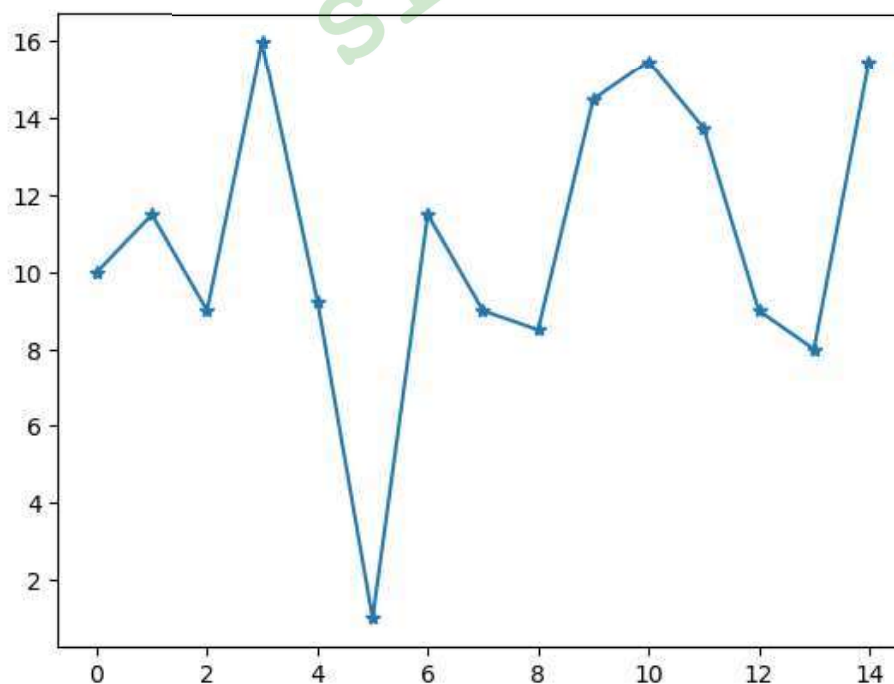


Line Plot

- **Use:** represents the time series data

In [9]:

```
plt.plot(df['StudyHours'],marker="*")  
plt.show()
```



- the x-axis values, should be in the sequential order

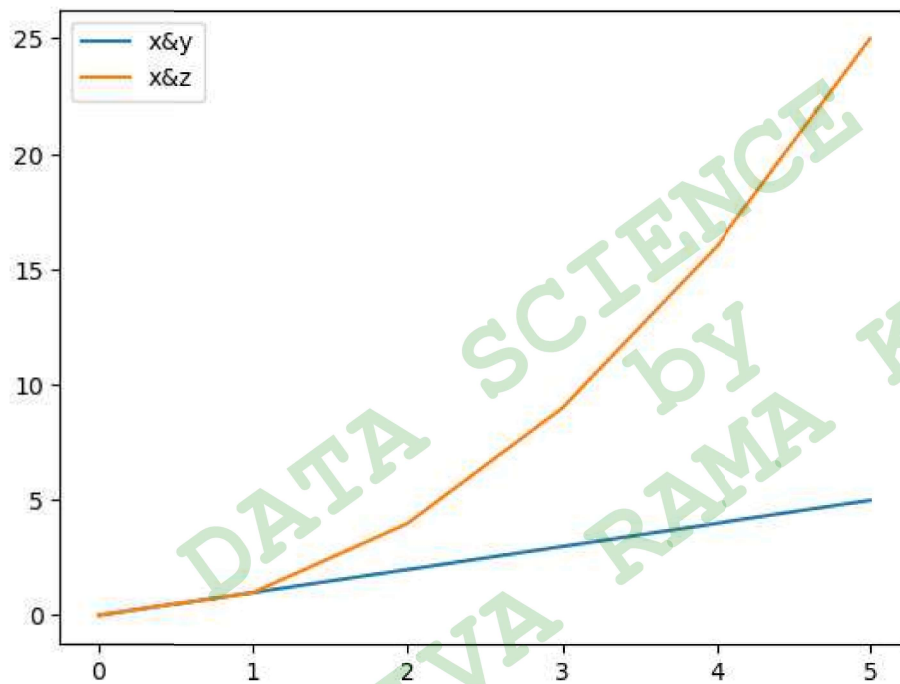


In [10]:

```
x = np.arange(6)
y = x
z = x**2
```

In [11]:

```
plt.plot(x,y,label="x&y")
plt.plot(x,z,label="x&z")
plt.legend()
plt.show()
```

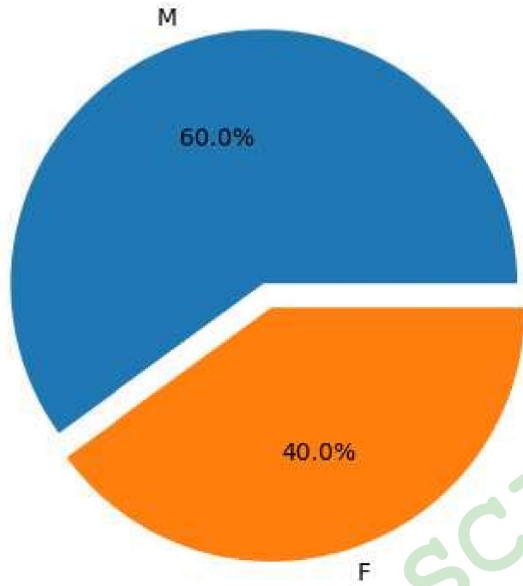




Pie Chart

In [12]:

```
plt.pie(x=df["Gender"].value_counts(), labels=df["Gender"].unique(), autopct='%0.1f%%',explode=[0.1,0.1],  
plt.show())
```



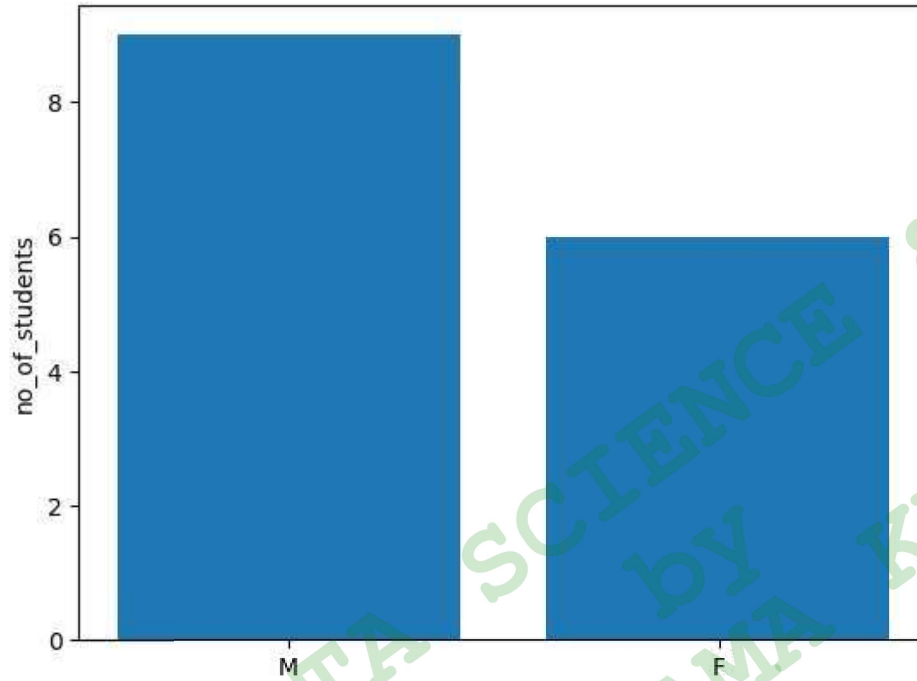
DATA SCIENCE & AI
by
SIVA RAMA KRISHNA



Bar plot

In [13]:

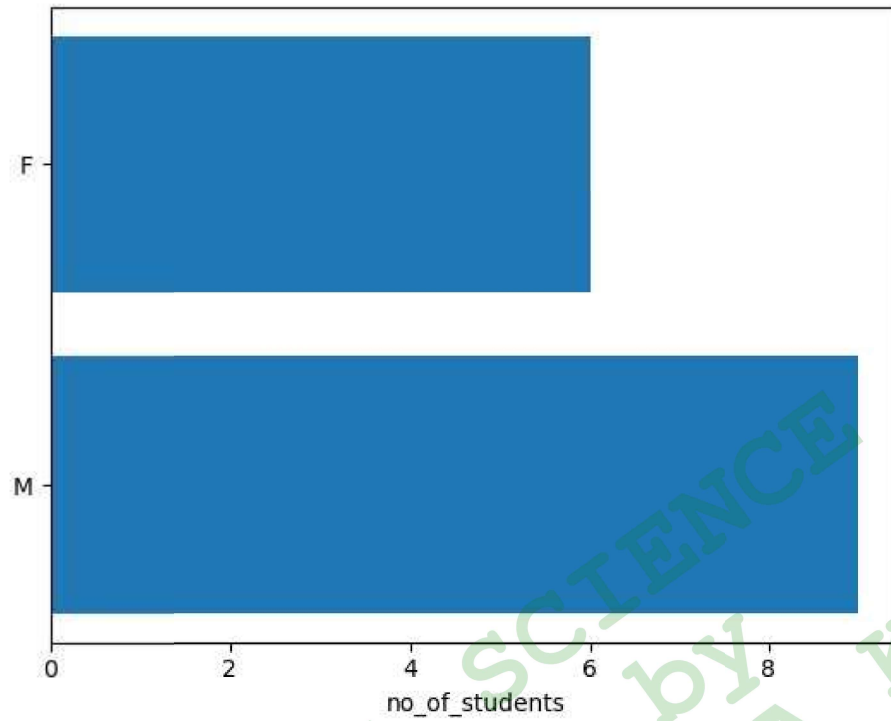
```
plt.bar(df["Gender"].value_counts().index,df["Gender"].value_counts())  
plt.ylabel("no_of_students")  
plt.show()
```





In [14]:

```
plt.barh(df["Gender"].unique(),df["Gender"].value_counts())  
plt.xlabel("no_of_students")  
plt.show()
```

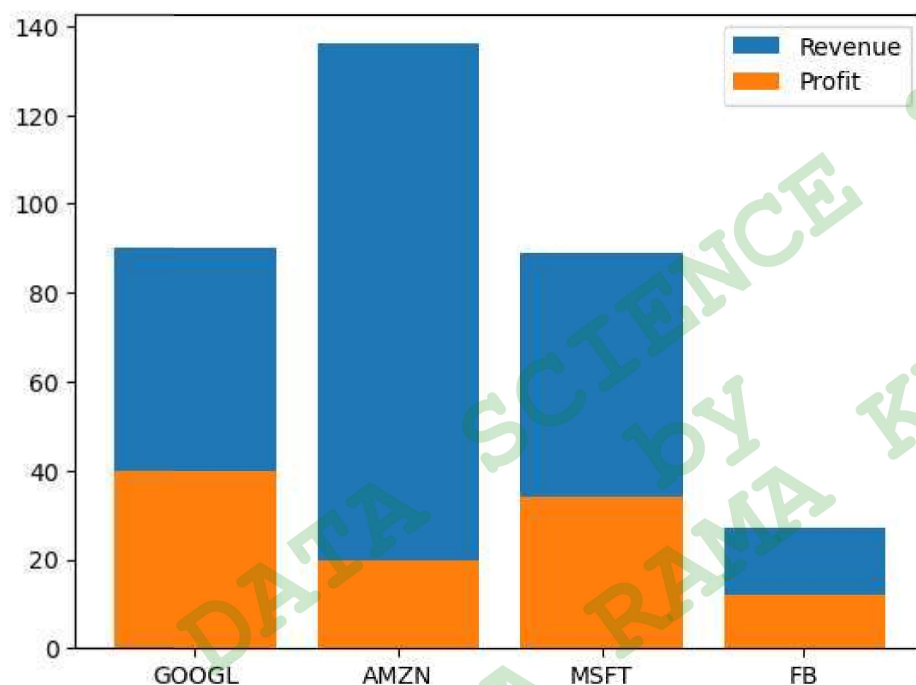




Stacked Bar plot

In [15]:

```
company = ['GOOGL', 'AMZN', 'MSFT', 'FB']  
turnover = [90, 136, 89, 27]  
profit = [40, 20, 34, 12]  
  
plt.bar(company, turnover, label="Revenue")  
plt.bar(company, profit, label="Profit")  
plt.legend()  
plt.show()
```

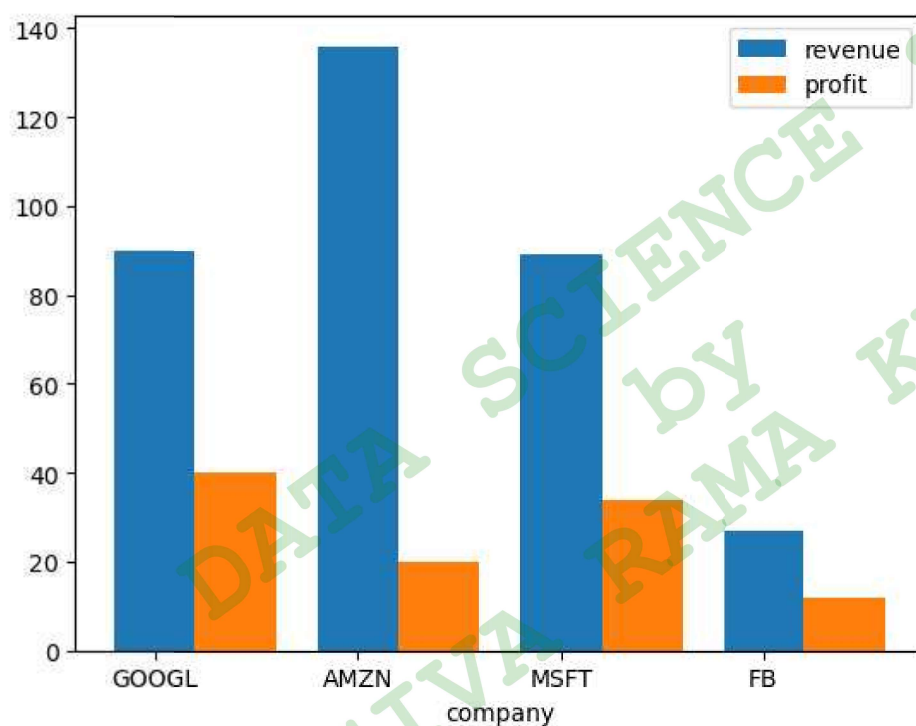




Unstacked Bar plot

In [16]:

```
company=[ 'GOOGL', 'AMZN', 'MSFT', 'FB' ]  
revenue=[90,136,89,27]  
profit=[40,20,34,12]  
  
plt.bar([0,1,2,3],revenue,width=0.4,label="revenue")  
plt.bar([0.4,1.4,2.4,3.4],profit,width=0.4,label="profit")  
plt.xlabel("company")  
plt.xticks([0,1,2,3],company)  
plt.legend()  
plt.show()
```





Subplots

In [17]:

```
#Program for drawing multiple bar charts on one image.
ecommerce=['Snapdeal', 'Alibaba', 'Amazon', 'Flipkart']
Q1_Profit=[45, 100, 70, 40]
Q2_Profit=[40, 105, 65, 45]
Q3_Profit=[42, 120, 72, 50]
Q4_Profit=[34, 115, 60, 69]

#Creating different bar charts on one image using subplot () function.
plt.figure(figsize=(10,10))

#Creating bar chart in first cell of figure having 3 rows, 2 columns.
plt.subplot(2,2,1)
plt.bar(ecommerce,Q1_Profit)

#Creating a bar chart in second cell.
plt.subplot(2,2,2)
plt.bar(ecommerce,Q2_Profit)

#Creating a bar chart in third cell.
plt.subplot(2,2,3)
plt.bar(ecommerce, Q3_Profit)

#Creating a bar chart in sixth cell.
plt.subplot(2,2,4)
plt.bar(ecommerce,Q4_Profit)

#Displaying the chart
plt.show ()
```

