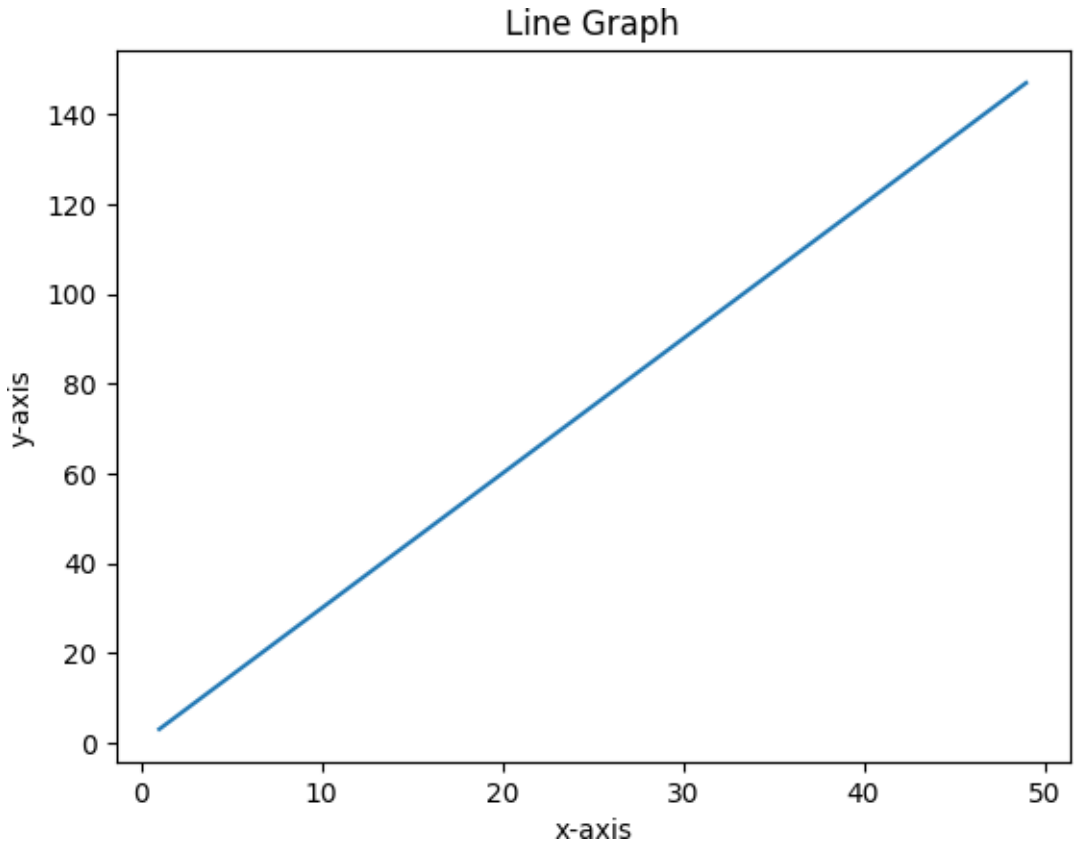
Data Visualisation
Assignment 8

#Matplotlib

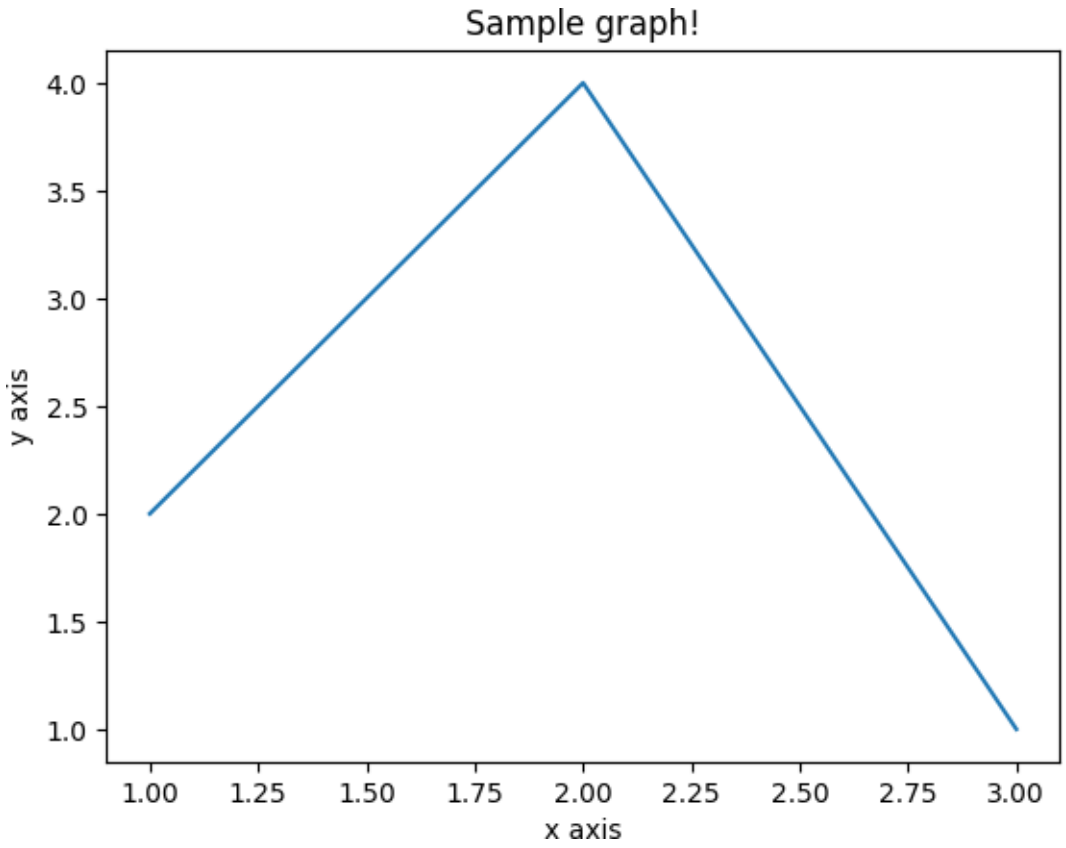
##1. 1. Write a Python program to draw a line with suitable label in the x axis, y axis and a title. The code snippet gives the output shown in the following screenshot:

[5]:
```
import matplotlib.pyplot as plt
x = range(1, 50)
y = [value * 3 for value in x]

plt.plot(x, y)
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('Line
Graph') plt.show()
```



##2 Write a Python program to draw a line using given axis values with suitable label in the x axis , y axis and a title. The code snippet gives the output shown in the following screenshot:

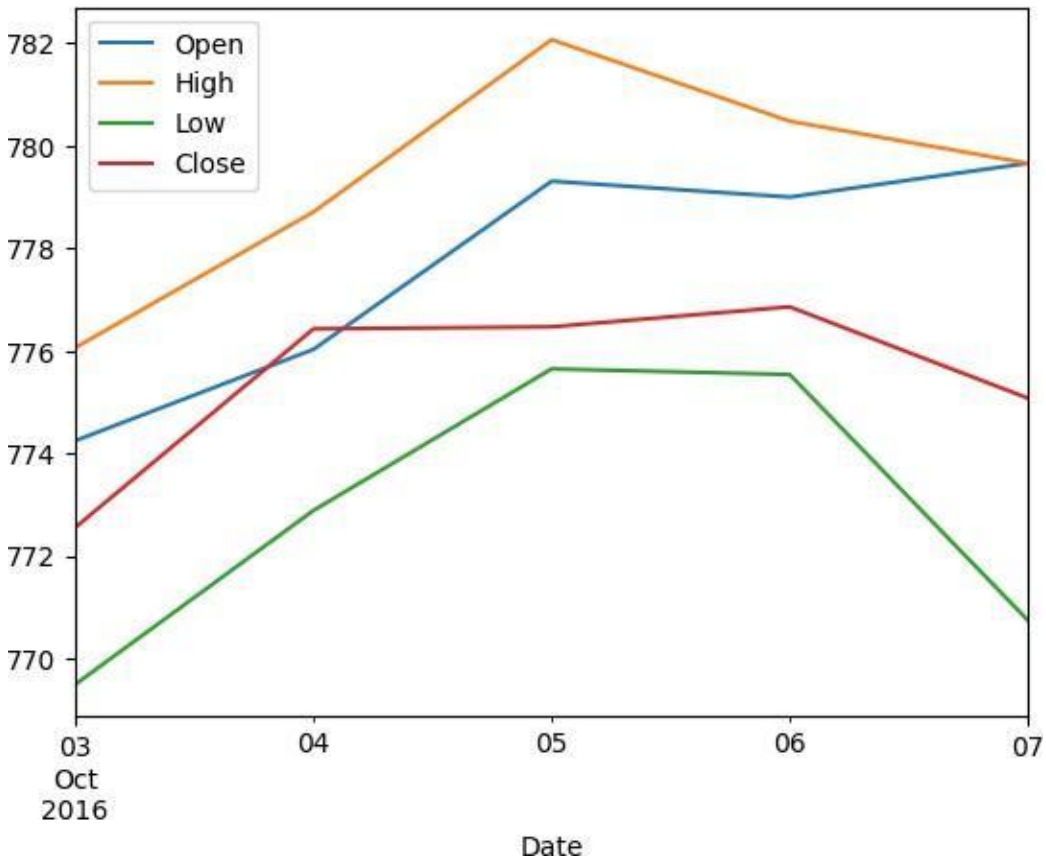
[6]:
```
import matplotlib.pyplot as plt
x = [1,2,3]
y = [2,4,1]
plt.plot(x, y)
 plt.xlabel('x axis')
 plt.ylabel('y axis')
 plt.title('Sample
 graph!') plt.show()
```



##3. Write a Python program to draw line charts of the financial data of Alphabet Inc. between October 3, 2016 to October 7, 2016.

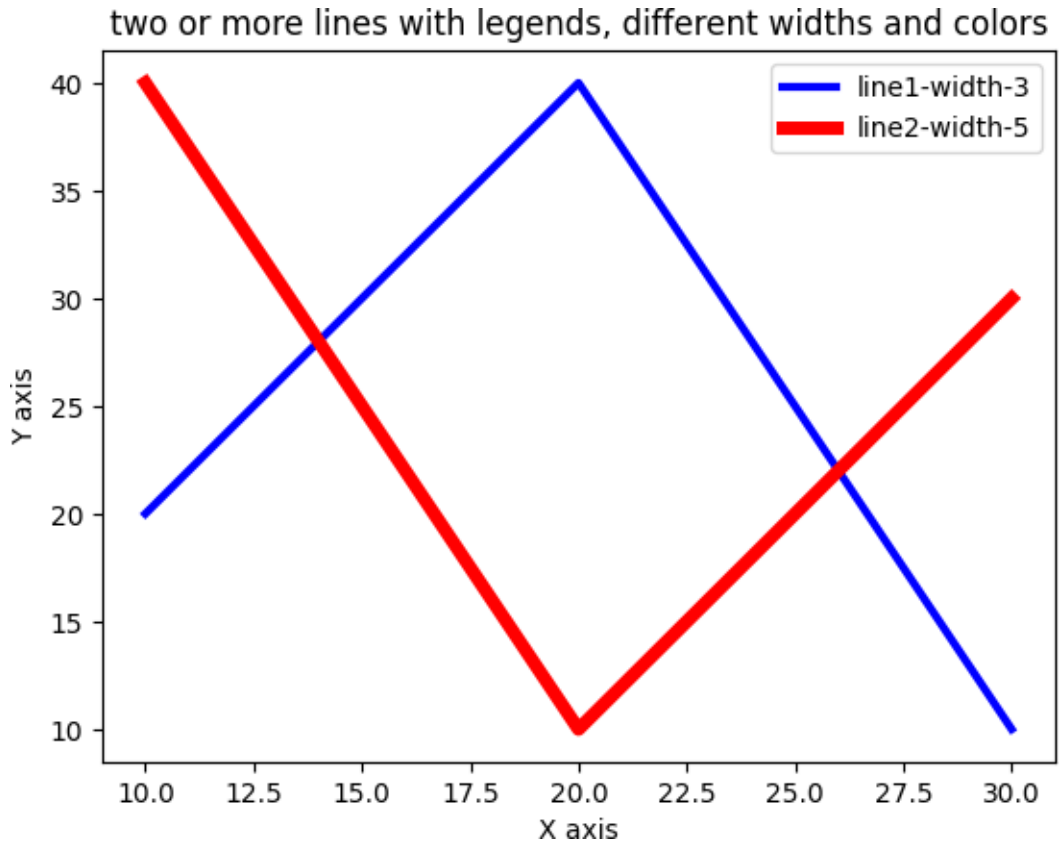
Sample Financial data (fdata.csv):
Date,Open,High,Low,Close
10-03-16,774.25,776.065002,769.5,772.559998
10-04-16,776.030029,778.710022,772.890015,776.429993
10-05-16,779.309998,782.070007,775.650024,776.469971
10-06-16,779,780.47998,775.539978,776.859985
10-07-16,779.659973,779.659973,770.75,775.080017
The code snippet gives the output shown in the following screenshot:

[12]:
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("fdata.csv",sep=',', parse_dates=True, index_col=0)
df.plot()
plt.show()
```



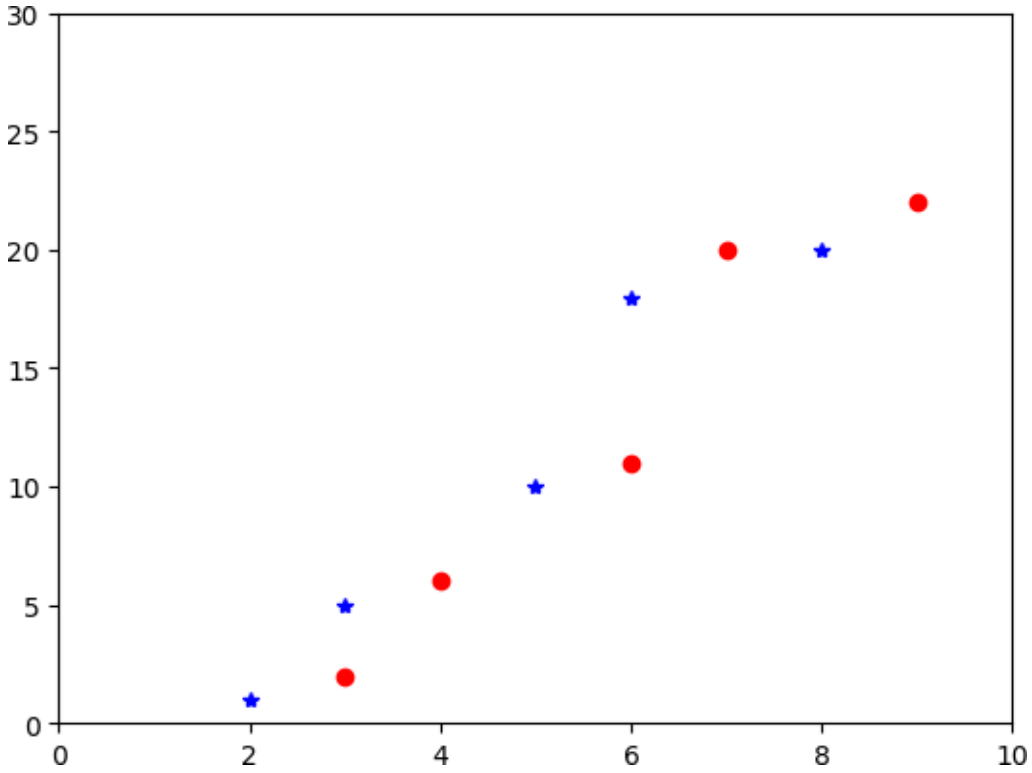
##4. Write a Python program to plot two or more lines with legends, different widths and colors. The code snippet gives the output shown in the following screenshot:

[14]:
```
import matplotlib.pyplot as plt
x1= [10,20,30]
y1 = [20,40,10]
x2 = [10,20,30]
y2 = [40,10,30]
plt.plot(x1,y1, color='blue', linewidth = 3,          label =
'line1-width-3') plt.plot(x2,y2, color='red', linewidth = 5,
label = 'line2-width-5') plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.legend()
plt.title("two or more lines with legends, different widths and
colors") plt.show()
```



##5 Write a Python program to plot quantities which have an x and y position. The code snippet gives the output shown in the following screenshot:

[17]:
```
import numpy as np
import pylab as pl x1 =
[2, 3, 5, 6, 8]
y1 = [1, 5, 10, 18, 20]
x2 = [3, 4, 6, 7, 9]
y2 = [2, 6, 11, 20, 22]
pl.axis([0, 10, 0, 30])
pl.plot(x1, y1,'b*', x2, y2,
'ro') pl.show()
```



##6. Write a Python programming to display a bar chart of the popularity of programming Languages.

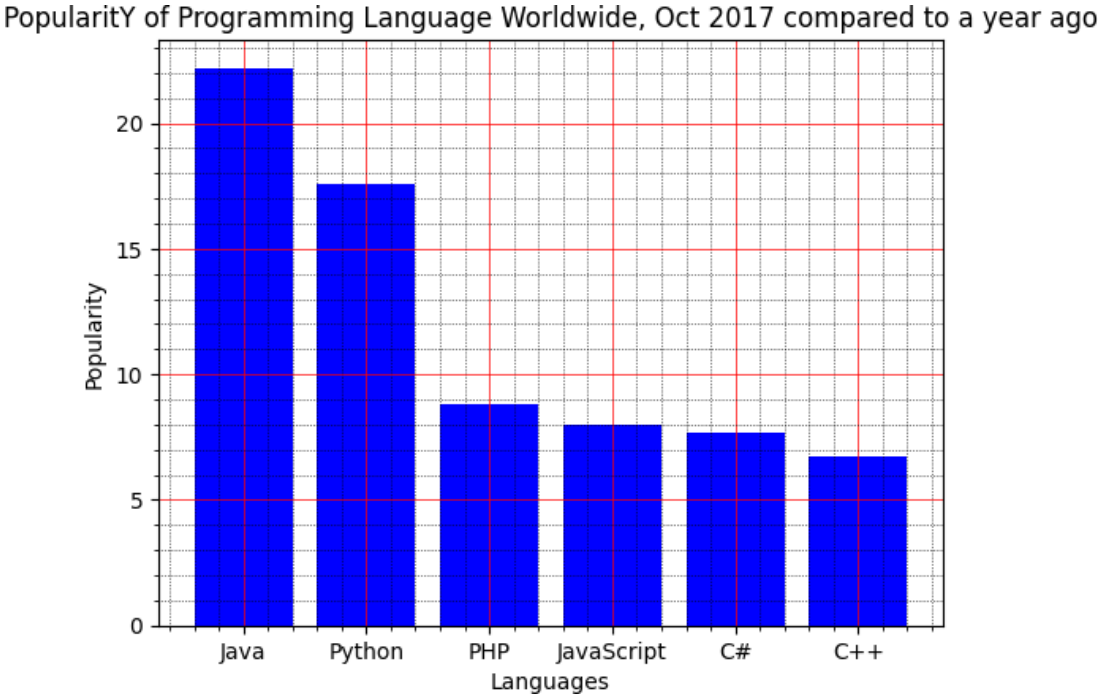
Sample data:

Programming languages: Java, Python, PHP, JavaScript, C#, C++ Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

The code snippet gives the output shown in the following screenshot:

[23]:
```
import matplotlib.pyplot as plt
x = ['Java', 'Python', 'PHP', 'JavaScript', 'C#',
'C++'] popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
x_pos = [i for i, _ in enumerate(x)]
plt.bar(x_pos, popularity,
color='blue')
plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.title("PopularitY of Programming Language Worldwide, Oct 2017
compared to a␣
 year ago")
plt.xticks(x_pos, x)
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')
```

```
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')

plt.show()
```



##7. Write a Python programming to display a horizontal bar chart of the popularity of programming Languages.

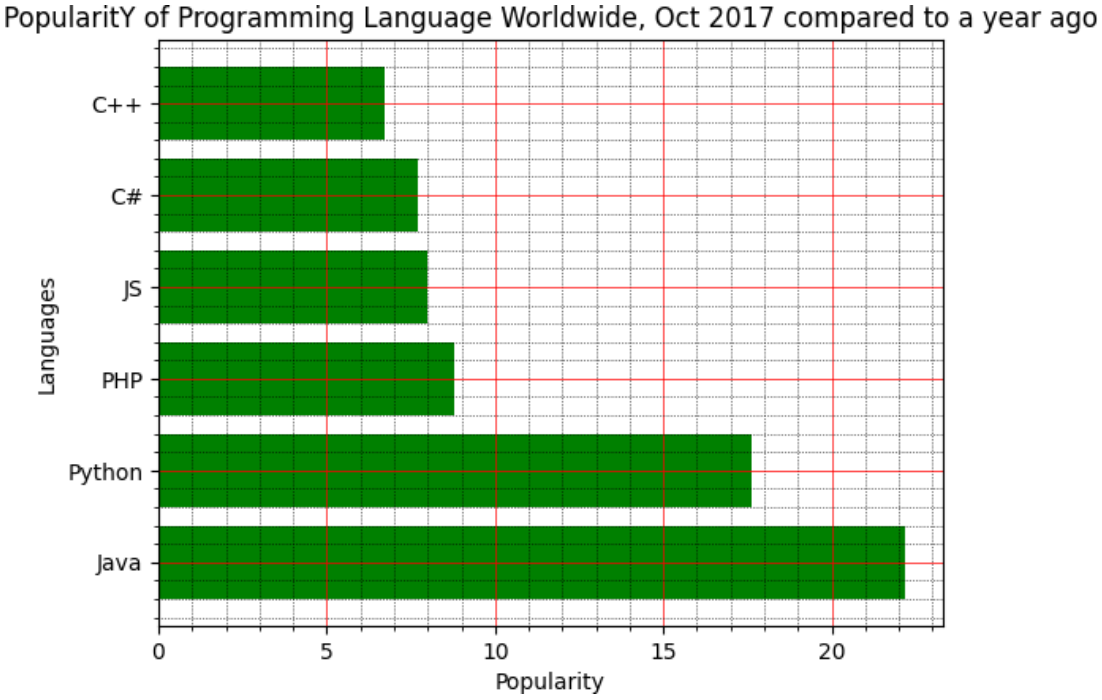
Sample data:

Programming languages: Java, Python, PHP, JavaScript, C#, C++ Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

The code snippet gives the output shown in the following screenshot:

[24]:
```
import matplotlib.pyplot as plt
x = ['Java', 'Python', 'PHP', 'JS', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7,
6.7]
x_pos = [i for i, _ in enumerate(x)]
plt.barh(x_pos, popularity,
color='green')
plt.xlabel("Popularity")
plt.ylabel("Languages")
plt.title("PopularitY of Programming Language Worldwide, Oct 2017
compared to a
year ago")
plt.yticks(x_pos, x)
```

```
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```



[25]:

1 Use the following CSV file for this exercise. Read this file using Pandas

```
!wget https://pynative.com/wp-content/uploads/201
```

```
--2023-11-
04
17:29:13--

https://py
native.com
/wp-
content/up
loads/201
9/01/com
pany_sales
_data.csv
Resolving pynative.com (pynative.com)…
172.66.43.37, 172.66.40.219,
2606:4700:3108::ac42:28db, …
Connecting to pynative.com
(pynative.com)|172.66.43.37|:443…
connected. HTTP request sent,
awaiting response… 200 OK
```

Length: 659 [text/csv]
Saving to: ‘company_sales_data.csv’

company_sales_data.
100%[==================>]

659

--.-KB/s

in 0s

2023-11-04 17:29:13 (515 MB/s) -
‘company_sales_data.csv’ saved [659/659]

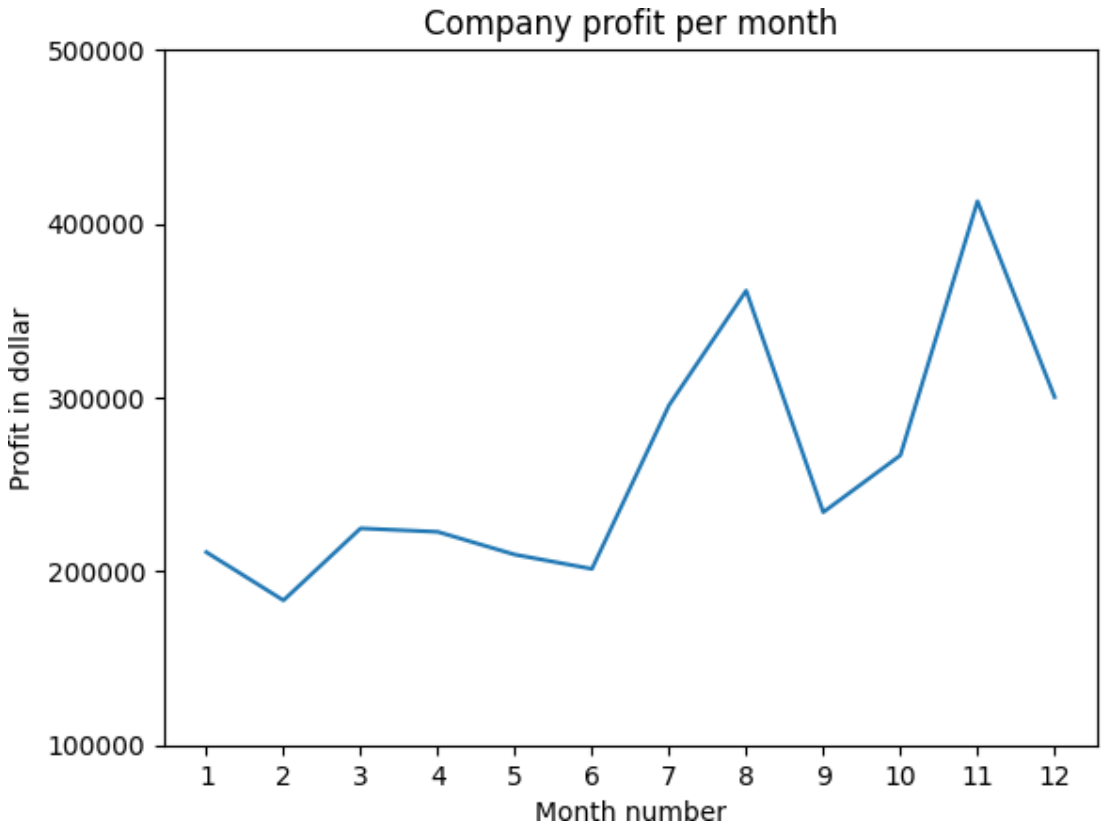
##Exercise 1: Read Total profit of all months and show it using a line plot

Total profit data provided for each month. Generated line plot must include the following properties: –

X label name = Month
Number Y label name = Total
profit

The line plot graph should look like this.

[26]:
```
import pandas as pd
import matplotlib.pyplot as plt

df =
 pd.read_csv('company_sales_data.c
 sv') profitList = df
 ['total_profit'].tolist() monthList= df
 ['month_number'].tolist()
plt.plot(monthList, profitList, label = 'Month-wise Profit data of last
 year') plt.xlabel('Month number')
plt.ylabel('Profit in dollar')
 plt.xticks(monthList)
 plt.title('Company profit per
 month')
plt.yticks([100000, 200000, 300000, 400000, 500000])
plt.show()
```



##Exercise 2: Get total profit of all months and show line plot with the following Style properties

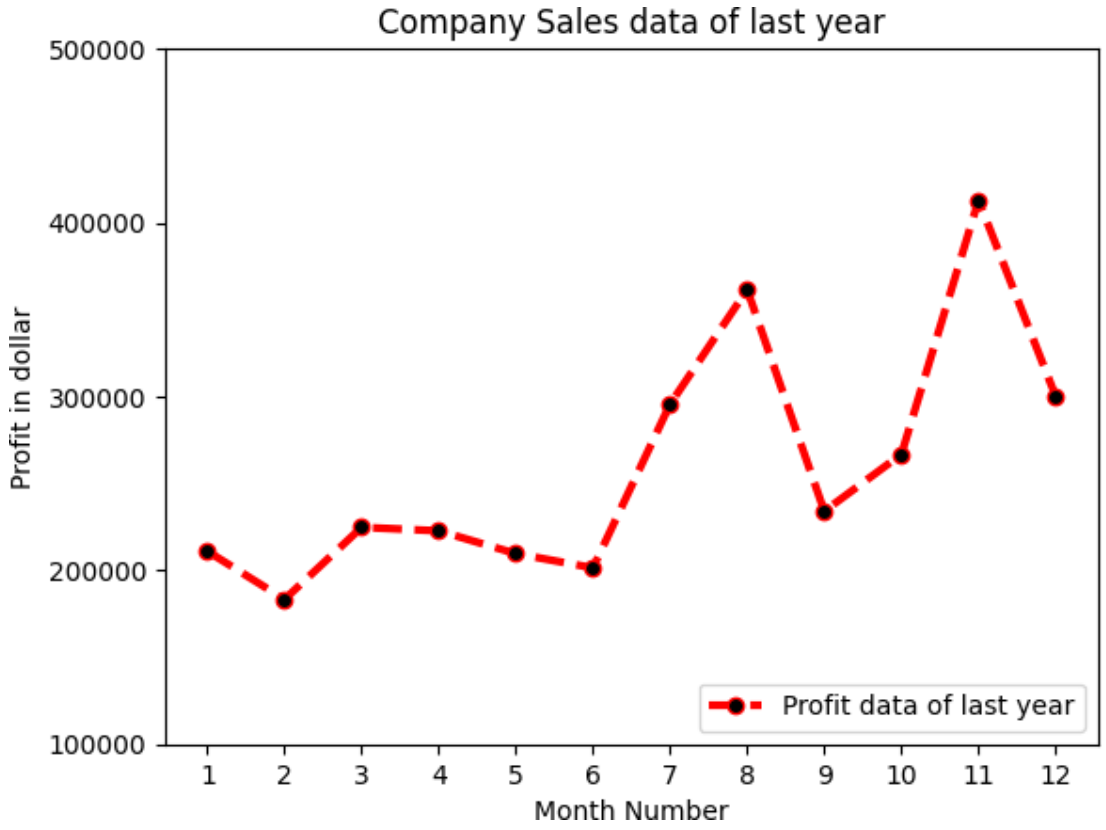
Generated line plot must include following Style properties: –

Line Style dotted and Line-color should be red Show
legend at the lower right location.
X label name = Month Number
Y label name = Sold units
number Add a circle marker.
Line marker color as read
Line width should be 3

The line plot graph should look like this.

[27]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("company_sales_data.csv")
profitList = df ['total_profit'].tolist()
```

```
monthList = df ['month_number'].tolist()

plt.plot(monthList, profitList, label = 'Profit data of last
      year', color='r', marker='o', markerfacecolor='k',
linestyle='--', linewidth=3)

plt.xlabel('Month Number')
plt.ylabel('Profit in dollar')
plt.legend(loc='lower right')
plt.title('Company Sales data of last
year') plt.xticks(monthList)
plt.yticks([100000, 200000, 300000, 400000, 500000])
plt.show()
```



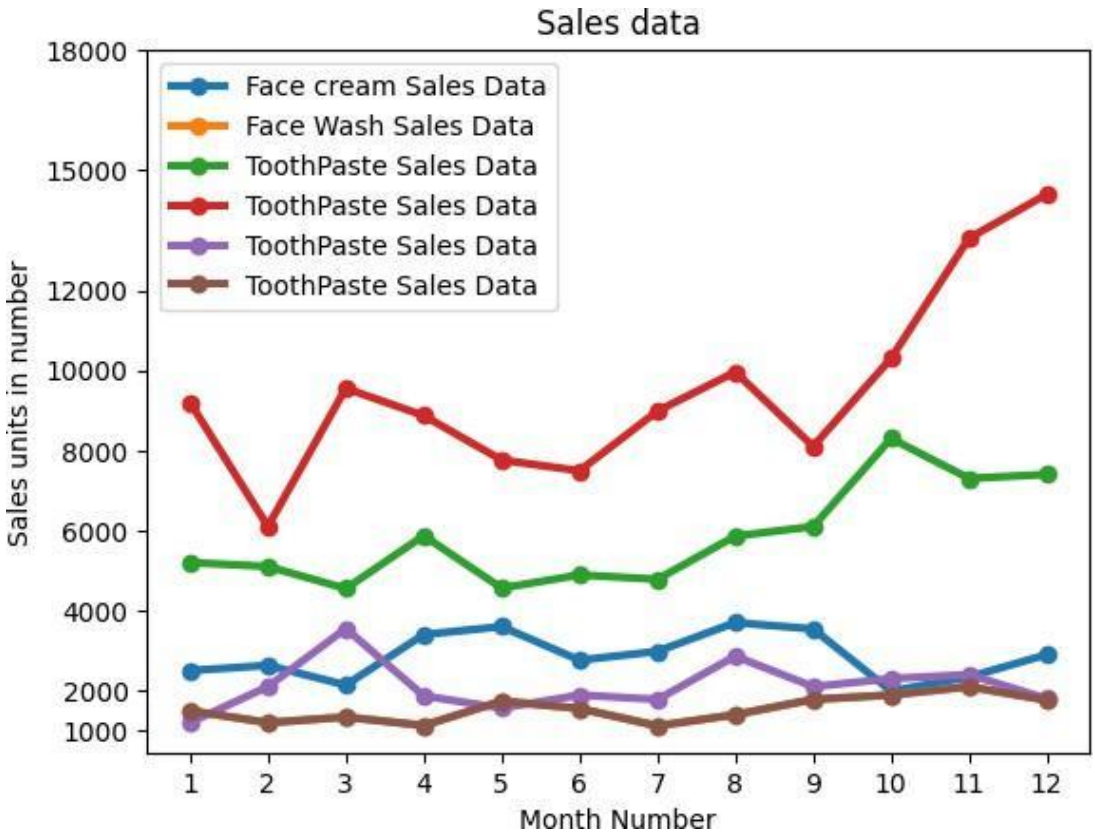
##Exercise 3: Read all product sales data and show it using a multiline plot Display the number of units sold per month for each product using multiline plots. (i.e., Separate Plotline for each product).

The graph should look like this.

[28]:

```
import pandas as pd
import matplotlib.pyplot as plt

df =
pd.read_c
sv("comp
any_sales
_data.csv"
)
monthList

= df
['month_n
umber'].t
olist()
faceCrem
SalesData

= df
['facecrea
m'].tolist()
faceWash
SalesData

= df
['facewas
h'].tolist()
toothPast
eSalesDat
a = df
['toothpas
te'].tolist()
bathingsoapS
alesData

= df
['bathingsoap'
].tolist()
shampooSales
Data

= df
['shampoo'].to
list()
moisturizerSal
esData = df
['moisturizer'].
tolist()

plt.plot(monthList, faceCremSalesData,
label = 'Face cream Sales Data',␣
 ↪marker='o', linewidth=3)
plt.plot(monthList,  faceWashSalesData,
label = 'Face Wash Sales Data', ␣
 ↪marker='o', linewidth=3)
plt.plot(monthList, toothPasteSalesData, label
= 'ToothPaste Sales Data',␣
 ↪marker='o', linewidth=3)
```

```
plt.plot(monthList, bathingsoapSalesData,
label = 'ToothPaste Sales Data',␣
  ↪marker='o', linewidth=3)
plt.plot(monthList, shampooSalesData, label =
'ToothPaste Sales Data',␣
  ↪marker='o', linewidth=3)
plt.plot(monthList, moisturizerSalesData, label
= 'ToothPaste Sales Data',␣
  ↪marker='o', linewidth=3)

plt.xlabel('Month Number')
plt.ylabel('Sales units in number')
plt.legend(loc='upper left')
plt.xticks(monthList)
plt.yticks([1000, 2000, 4000, 6000, 8000,
10000, 12000, 15000, 18000])
plt.title('Sales data')
plt.show()
```

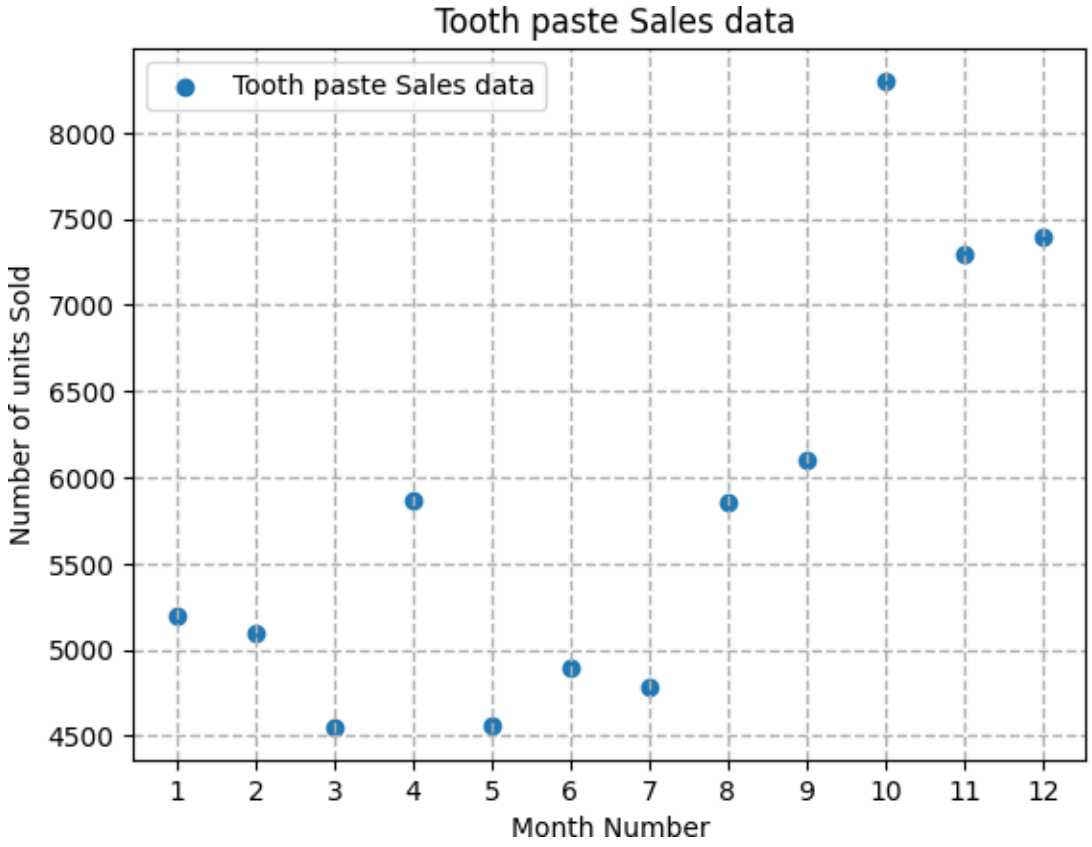


##Exercise 4: Read toothpaste sales data of each month and show it using a scatter plot Also, add a grid in the plot. gridline style should "–".

The scatter plot should look like this.

[29]:
```
import pandas as pd
import matplotlib.pyplot as plt

df =
pd.read_csv("company_sales_data.csv")
monthList= df ['month_number'].tolist()
toothPasteSalesData = df
['toothpaste'].tolist()
plt.scatter(monthList, toothPasteSalesData, label = 'Tooth paste
Sales data') plt.xlabel('Month Number')
plt.ylabel('Number of units
Sold') plt.legend(loc='upper
left') plt.title(' Tooth paste Sales
data') plt.xticks(monthList)
plt.grid(True, linewidth= 1, linestyle="--")
```

```
plt.show()
```

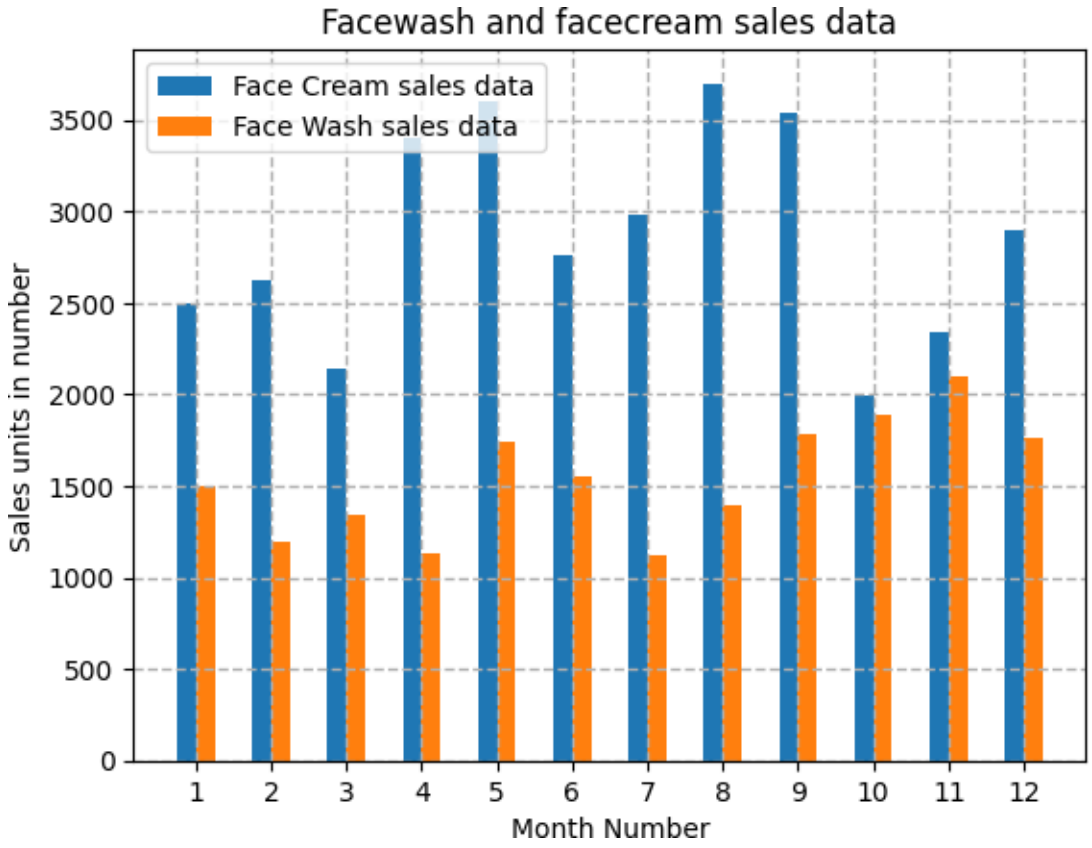


##Exercise 5: Read face cream and facewash product sales data and show it using the bar chart The bar chart should display the number of units sold per month for each product. Add a separate bar for each product in the same chart.

The bar chart should look like this.

[30]:
```
import pandas as pd
import matplotlib.pyplot as plt

df =
pd.read_csv("company_sales_data.csv")
monthList= df ['month_number'].tolist()
faceCremSalesData  = df
['facecream'].tolist() faceWashSalesData
= df ['facewash'].tolist()

plt.bar([a-0.25 for a in monthList], faceCremSalesData, width= 0.25,
label =␣
'Face Cream sales data', align='edge')
```

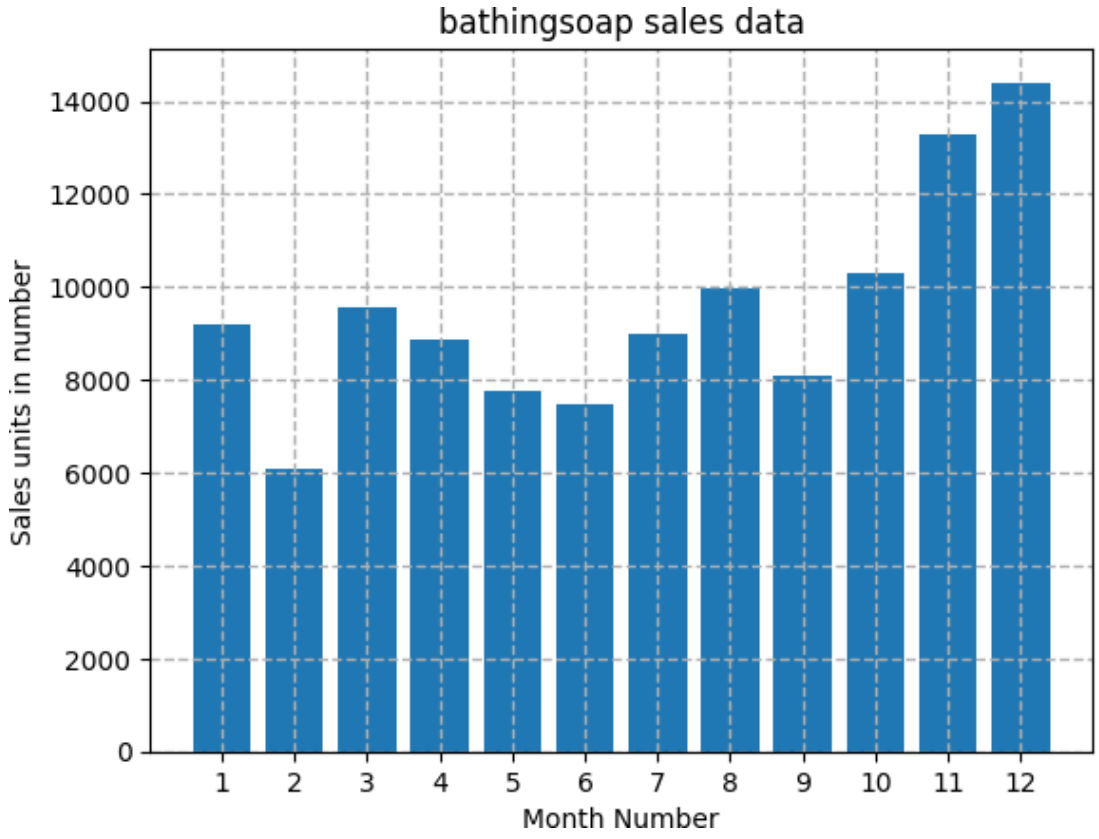
```
plt.bar([a+0.25 for a in monthList], faceWashSalesData, width= -0.25,
label =␣
  .'Face Wash sales data',
align='edge') plt.xlabel('Month
Number') plt.ylabel('Sales units
in number')
plt.legend(loc='upper  left')
plt.title(' Sales data')

plt.xticks(monthList)
plt.grid(True, linewidth= 1, linestyle="--")
 plt.title('Facewash and facecream sales
 data') plt.show()
```



##Exercise 6: Read sales data of bathing soap of all months and show it using a bar chart. Save this plot to your hard disk The bar chart should look like this.

[31]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

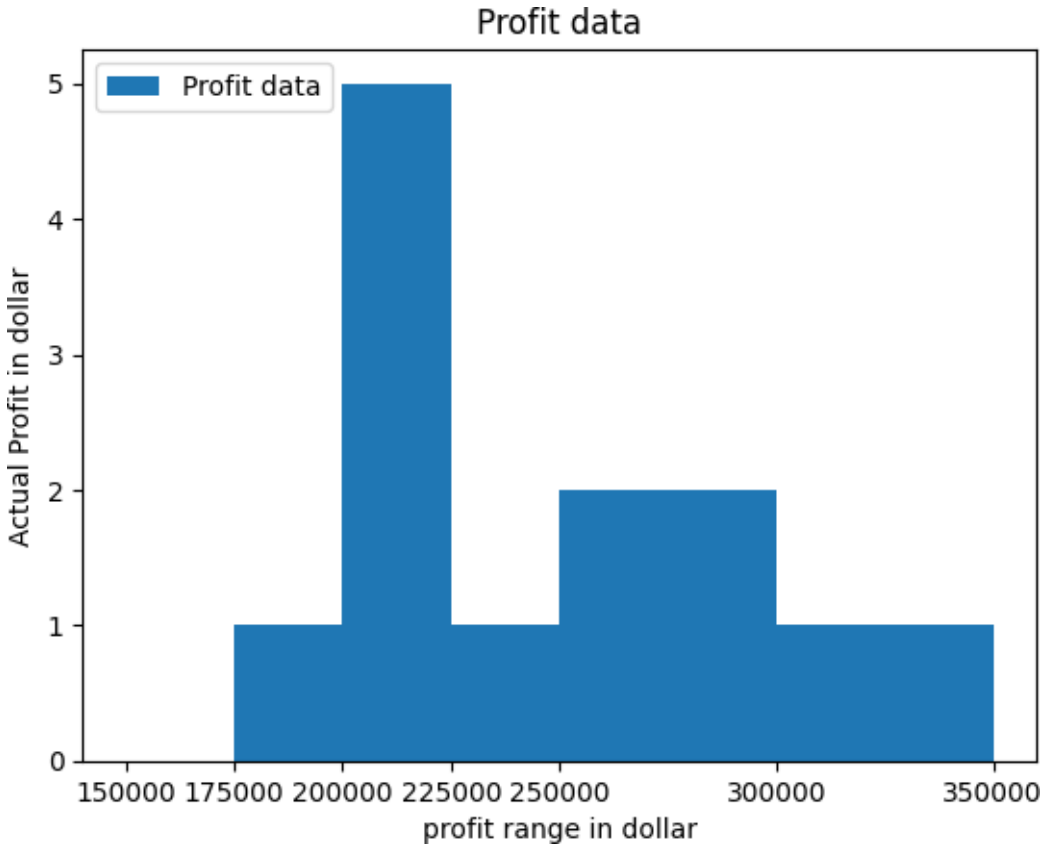
```
df =
 pd.read_csv("company_sales_data.c
 sv") monthList  = df
 ['month_number'].tolist()
bathingsoapSalesData       = df
 ['bathingsoap'].tolist() plt.bar(monthList,
 bathingsoapSalesData) plt.xlabel('Month
 Number')
plt.ylabel('Sales units in
 number') plt.title(' Sales
 data') plt.xticks(monthList)
plt.grid(True, linewidth= 1, linestyle="--")
 plt.title('bathingsoap sales data')
 plt.savefig('Sales_data_of_bathingsoap.png'
 , dpi=150) plt.show()
```



##Exercise 7: Read the total profit of each month and show it using the histogram to see the most common profit ranges The histogram should look like this.

[32]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

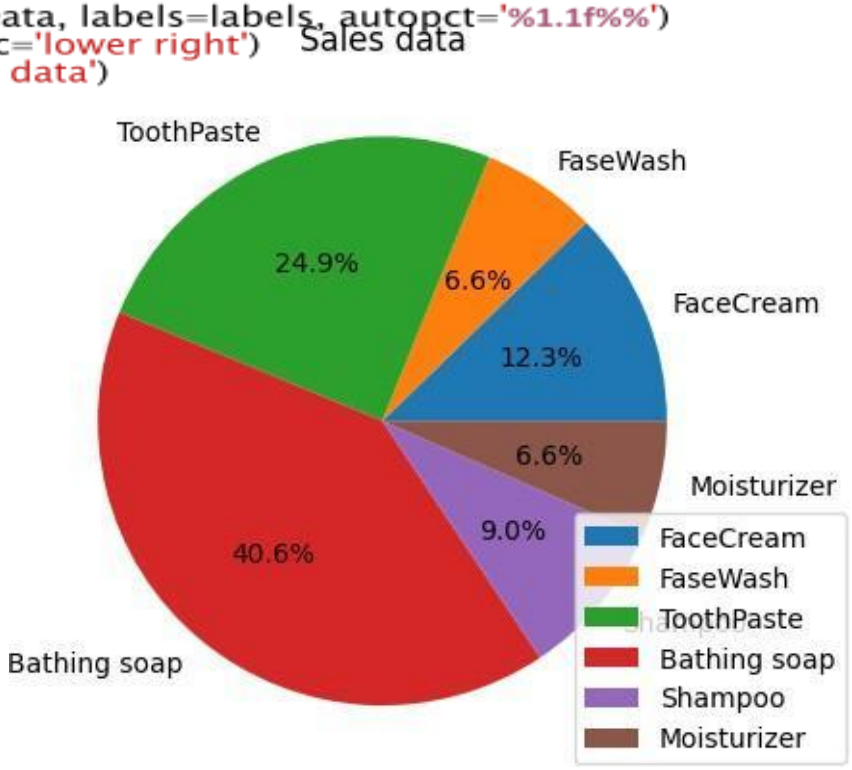
```
df =
 pd.read_csv("company_sales_data.cs
v") profitList = df
['total_profit'].tolist() labels = ['low',
'average', 'Good', 'Best']
profit_range = [150000, 175000, 200000, 225000, 250000,
 300000, 350000]
plt.hist(profitList, profit_range, label = 'Profit data')
 plt.xlabel('profit range in dollar')
plt.ylabel('Actual Profit in
 dollar') plt.legend(loc='upper
 left') plt.xticks(profit_range)
 plt.title('Profit data')
plt.show()
```



##Exercise 8: Calculate total sale data for last year for each product and show it using a Pie chart
Note: In Pie chart display Number of units sold per year for each product in percentage.

The Pie chart should look like this.

[33]:
```
import pandas as pd
import matplotlib.pyplot as plt

monthList = df ['month_number'].tolist()

labels = ['FaceCream', 'FaseWash', 'ToothPaste', 'Bathing soap',
'Shampoo',
'Moisturizer']
salesData = [df ['facecream'].sum(), df ['facewash'].sum(), df
['toothpaste'].
  sum(),
df ['bathingsoap'].sum(), df ['shampoo'].sum(), df ['moisturizer'].
  sum()]
plt.axis("equal")
plt.pie(salesData, labels=labels, autopct='%1.1f%%')
plt.legend(loc='lower right')
plt.title('Sales data')
plt.show()
```



##Exercise 9: Read Bathing soap facewash of all months and display it using the Subplot The Subplot should look like this

[34]:
```
import matplotlib.pyplot as plt
```

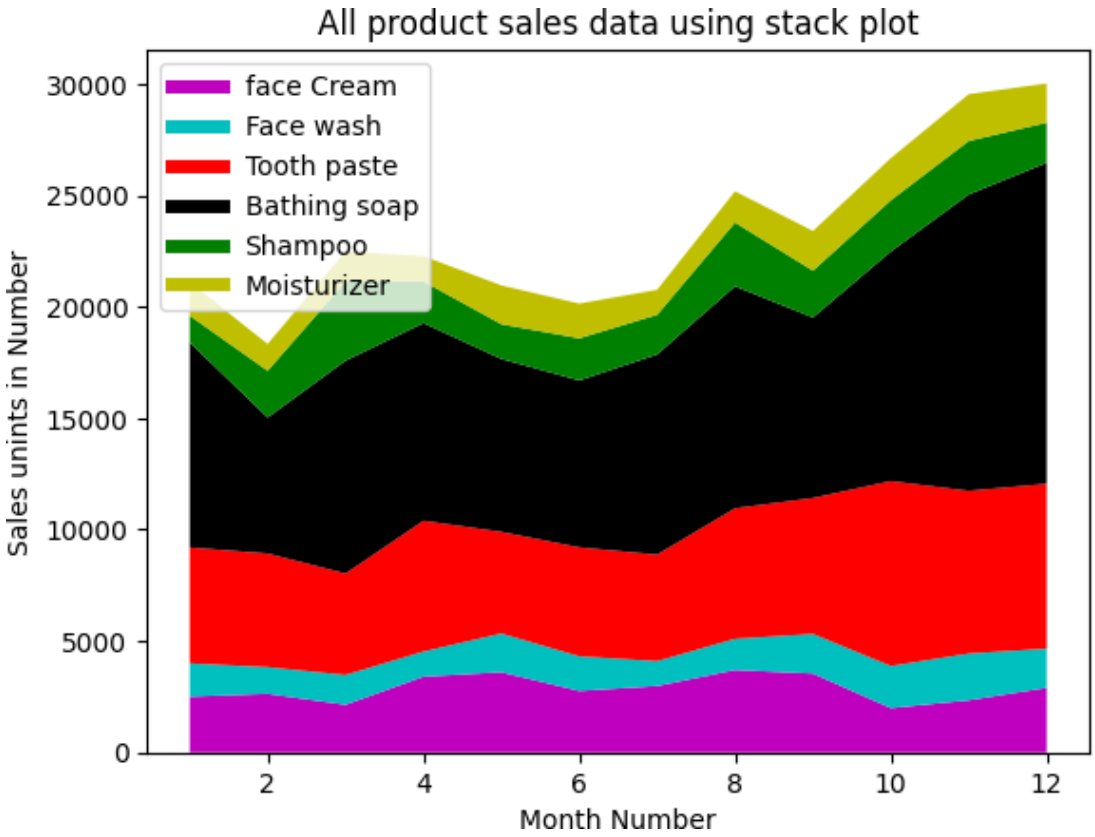
```
monthList = df
['month_number'].tolist() bathingsoap
= df ['bathingsoap'].tolist()
faceWashSalesData   = df
['facewash'].tolist()

f, axarr = plt.subplots(2, sharex=True)
axarr[0].plot(monthList, bathingsoap, label = 'Bathingsoap Sales
Data',
  color='k', marker='o', linewidth=3)
axarr[0].set_title('Sales data of  a
Bathingsoap')
axarr[1].plot(monthList, faceWashSalesData, label = 'Face Wash Sales
Data',
  color='r', marker='o', linewidth=3)
axarr[1].set_title('Sales data of  a
facewash')

plt.xticks(monthList)
plt.xlabel('Month Number')
plt.ylabel('Sales units in
number') plt.show()
```

##Exercise Question 10: Read all product sales data and show it using the stack plot The Stack plot should look like this.

[36]:

```
import matplotlib.pyplot as plt

faceCremSale
sData


= df
['facecream'].t
olist()
faceWashSales
Data


= df
['facewash'].to
list()
toothPasteSal
esData = df
['toothpaste'].t
olist()
bathingsoapS
alesData

= df
['bathingsoap'
].tolist()
shampooSales
Data

= df
['shampoo'].to
list()
moisturizerSal
esData = df
['moisturizer'].
tolist()

plt.plot([],[],color='m', label='face
Cream', linewidth=5)
plt.plot([],[],color='c', label='Face
wash', linewidth=5)
plt.plot([],[],color='r', label='Tooth
paste', linewidth=5)
plt.plot([],[],color='k', label='Bathing
soap', linewidth=5)
plt.plot([],[],color='g',
label='Shampoo', linewidth=5)
plt.plot([],[],color='y',
label='Moisturizer', linewidth=5)

plt.stackplot(monthList, faceCremSalesData,
faceWashSalesData,␣
 ↪toothPasteSalesData,
                    bathingsoapSalesData,
                    shampooSalesData,
                    moisturizerSalesData,
                    colors=['m','c','r','k','g','y
                    '])
```

```
plt.x
labe
l('M
onth
Nu
mbe
r')
plt.y
labe
l('Sal
es
unin
ts in
Nu
mbe
r')
plt.title('All product sales data using
stack plot') plt.legend(loc='upper left')
plt.show()
```


All product sales data using stack plot
face Cream
Face wash
Tooth paste
Bathing soap
Shampoo
Moisturizer
Sales unints in Number
Month Number
0
5000
10000
15000
20000
25000
30000
2
4
6
8
10
12