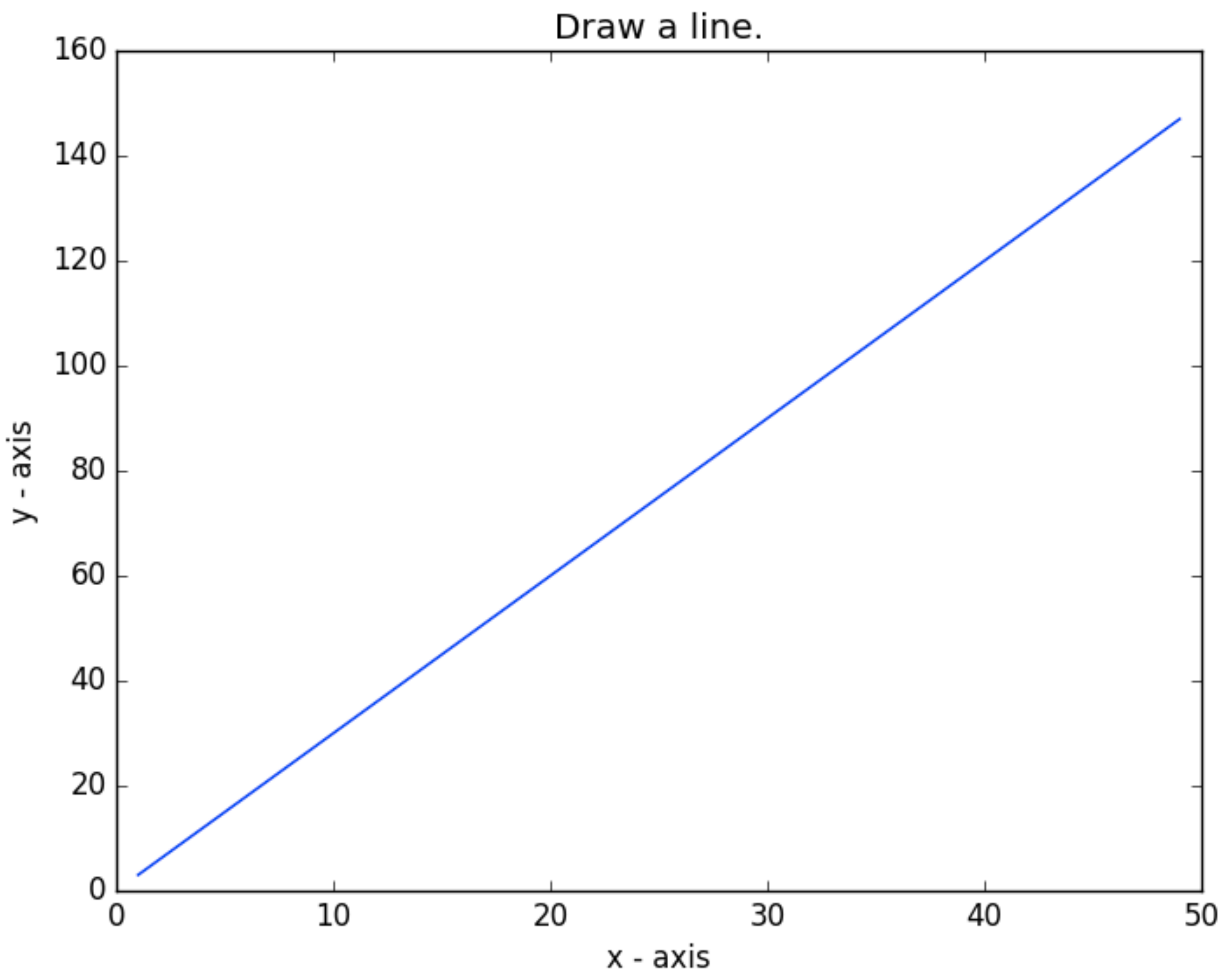
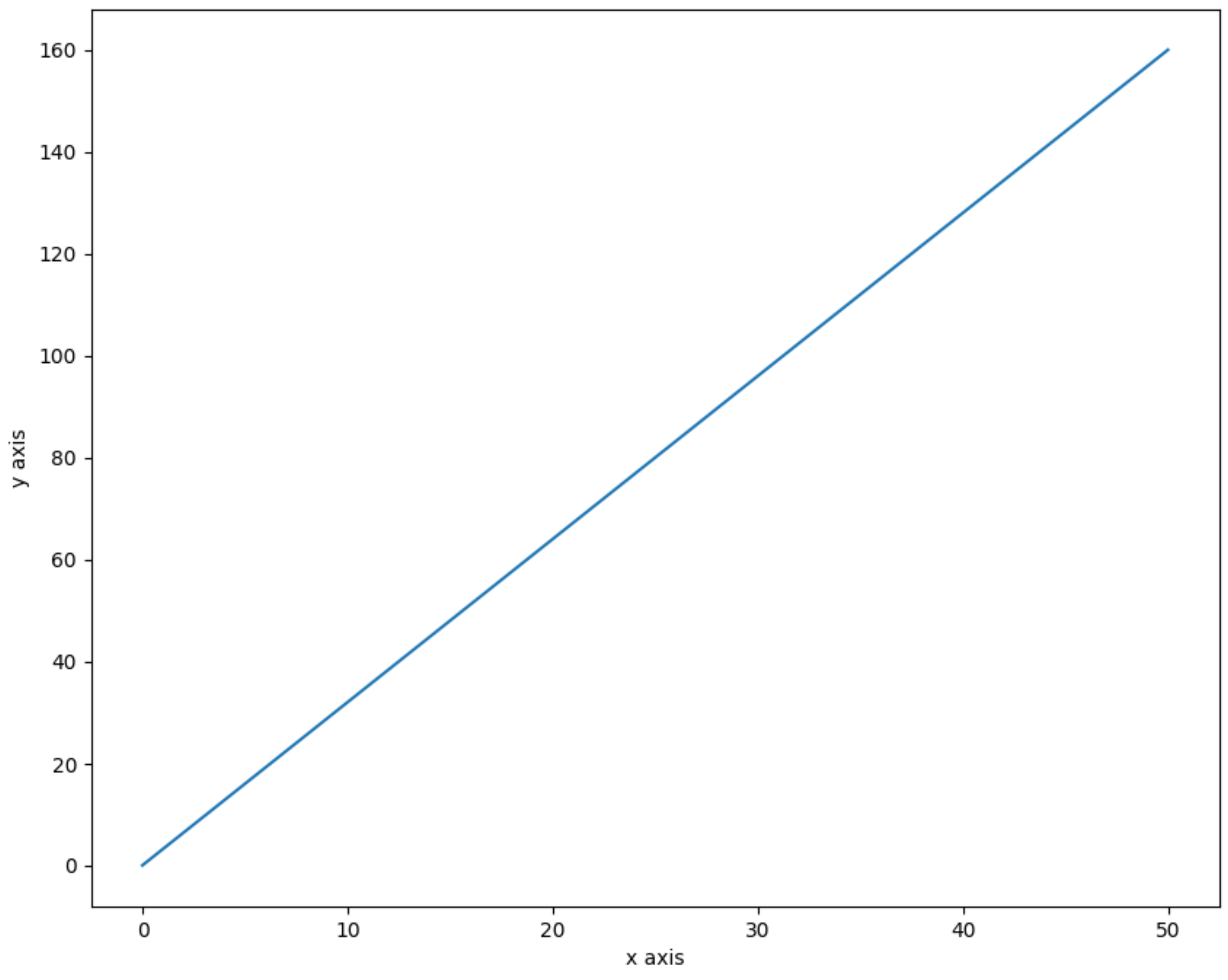
Matplotlib

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:

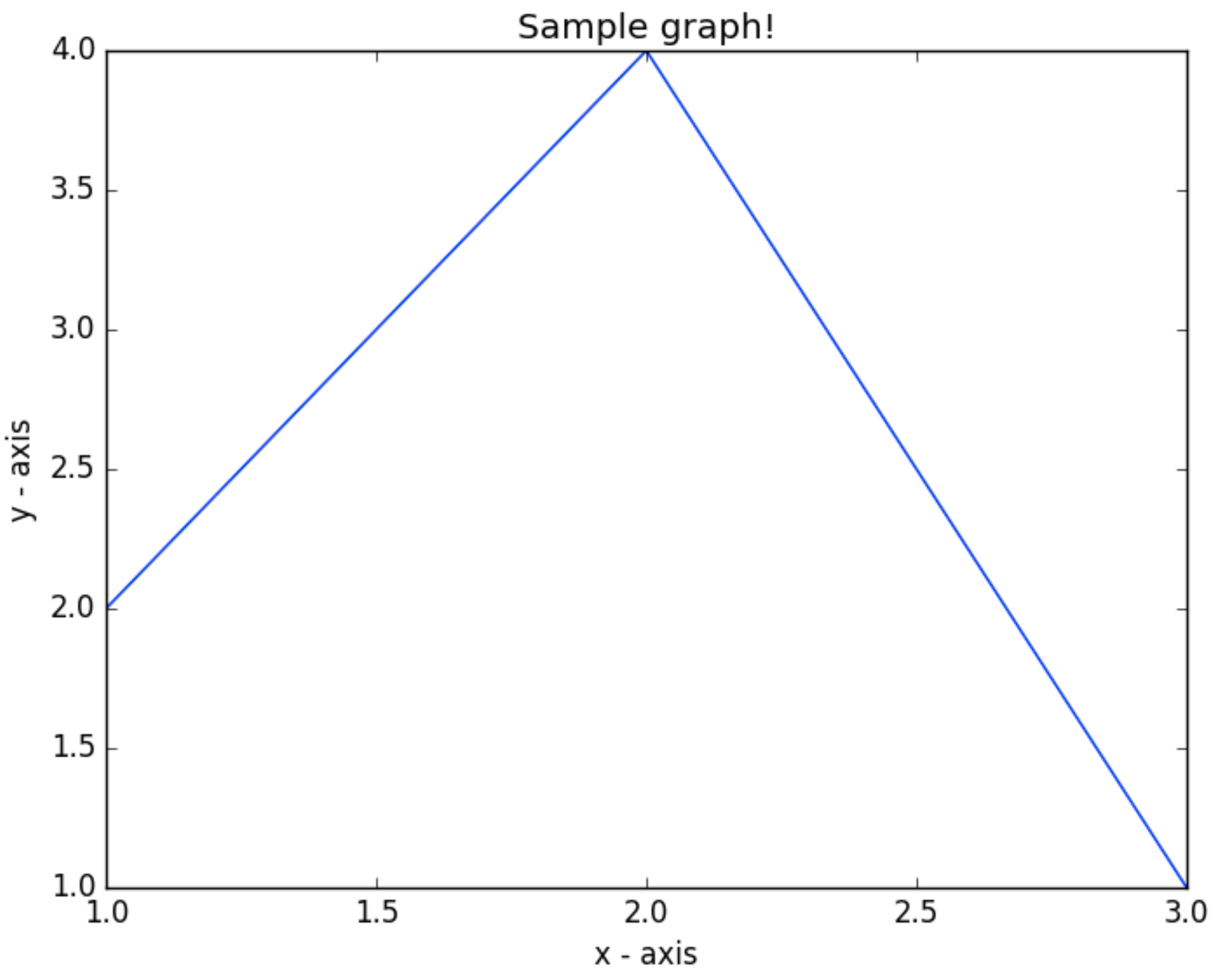


```
from matplotlib import pyplot as plt
import numpy as np

xpoints = np.array([0, 50])
ypoints = np.array([0, 160])

myfigure = plt.figure(figsize=(10, 8))
```

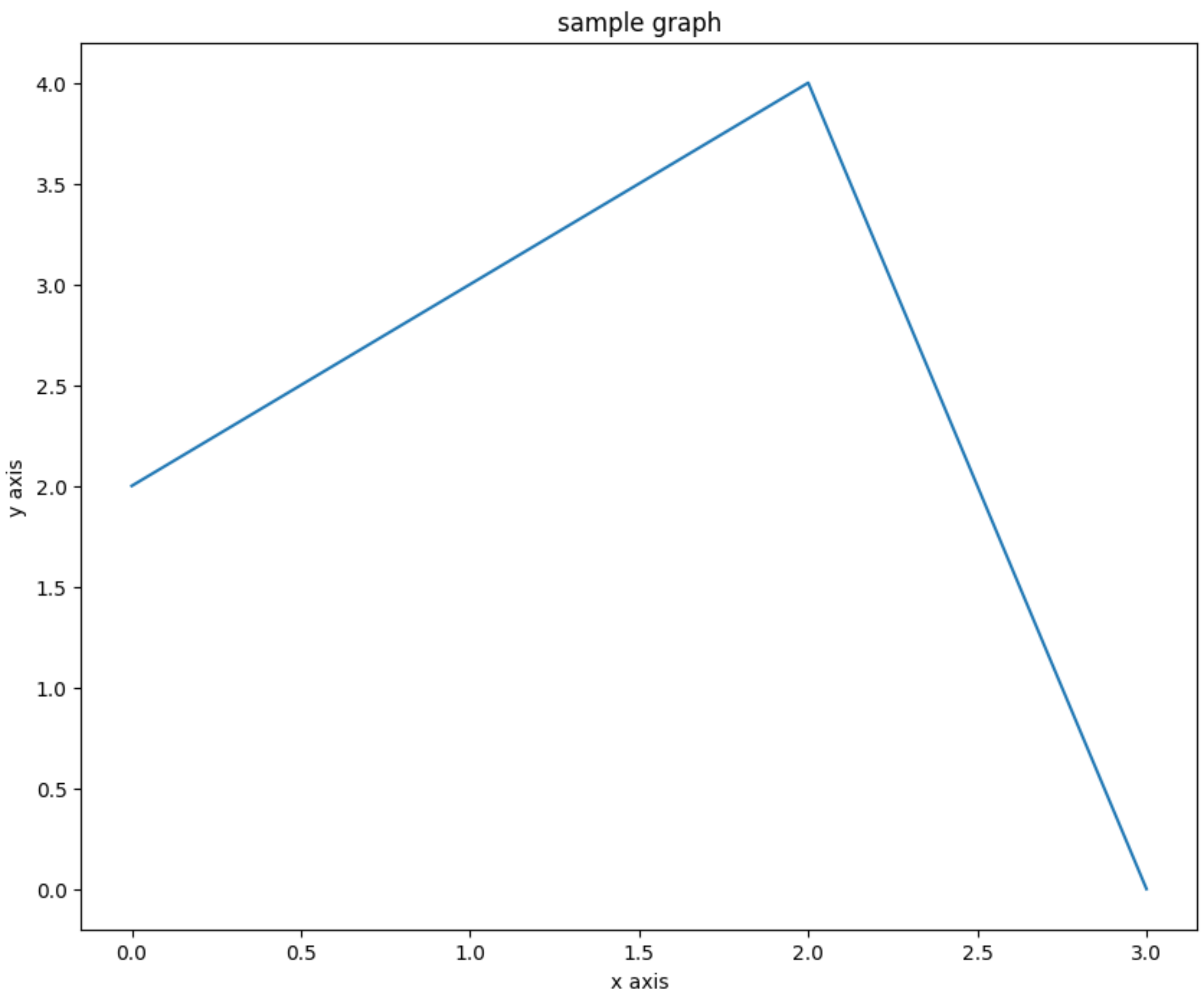
```
plt.xlabel("x axis")
plt.ylabel("y axis")


plt.plot(xpoints, ypoints)
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:

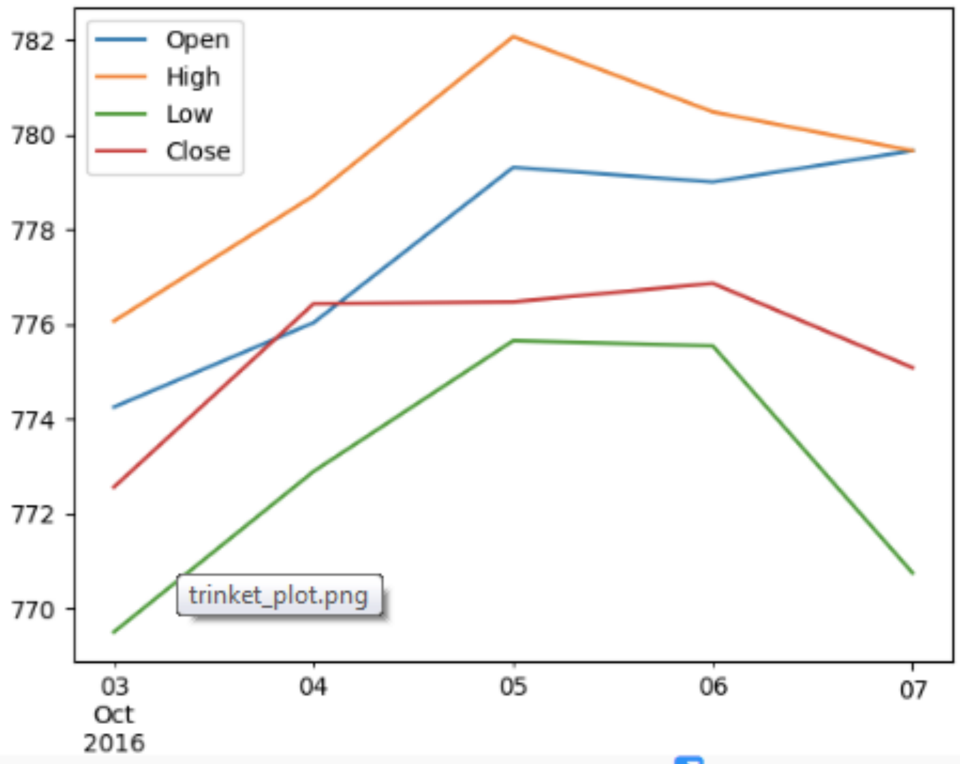


```
from matplotlib import pyplot as plt
import numpy as np

ypoints = np.array([2.0, 4.0,0])
xpoints = np.array([0, 2.0,3.0])

myfigure = plt.figure(figsize=(10, 8))

plt.xlabel("x axis")
plt.ylabel("y axis")
plt.title("sample graph")


plt.plot(xpoints, ypoints)
plt.show()
```



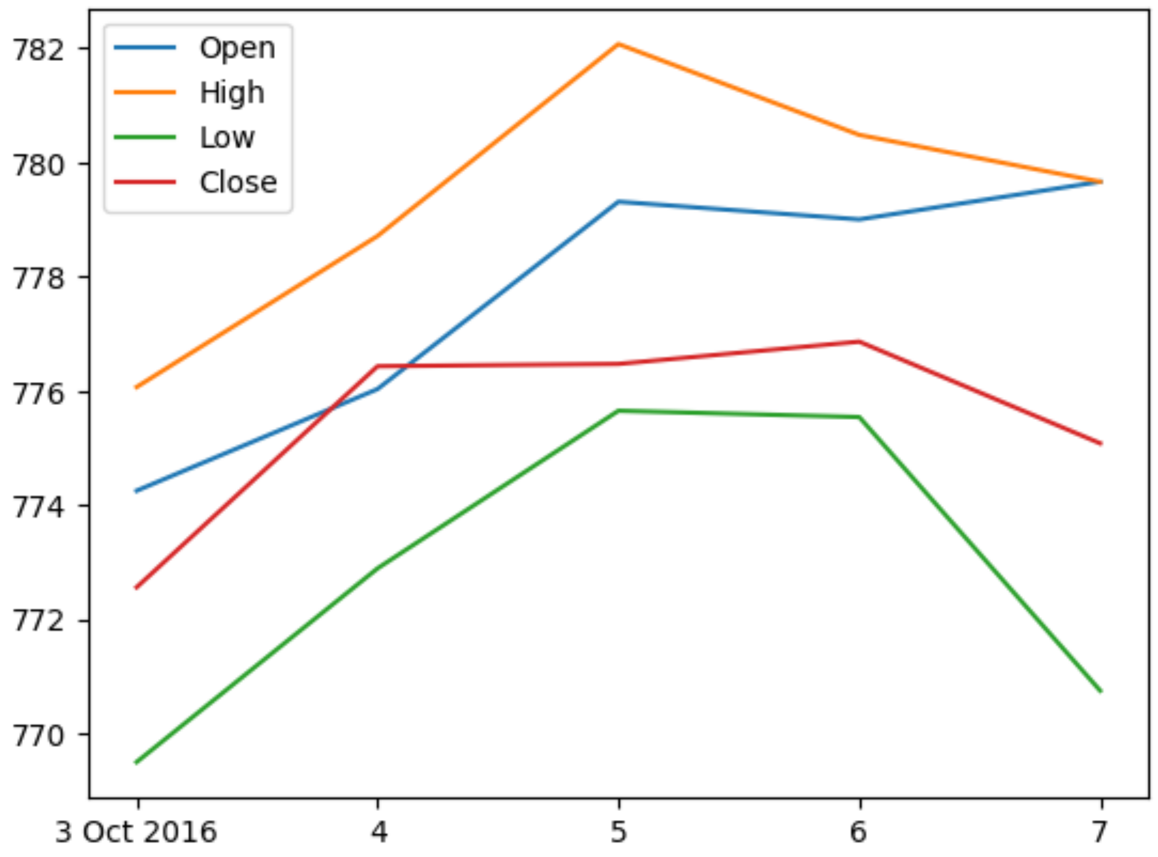
3. Write a Python program to draw line charts of the financial data of Alphabet Inc. between October 3, 2016 to October 7, 2016.

```
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:
```

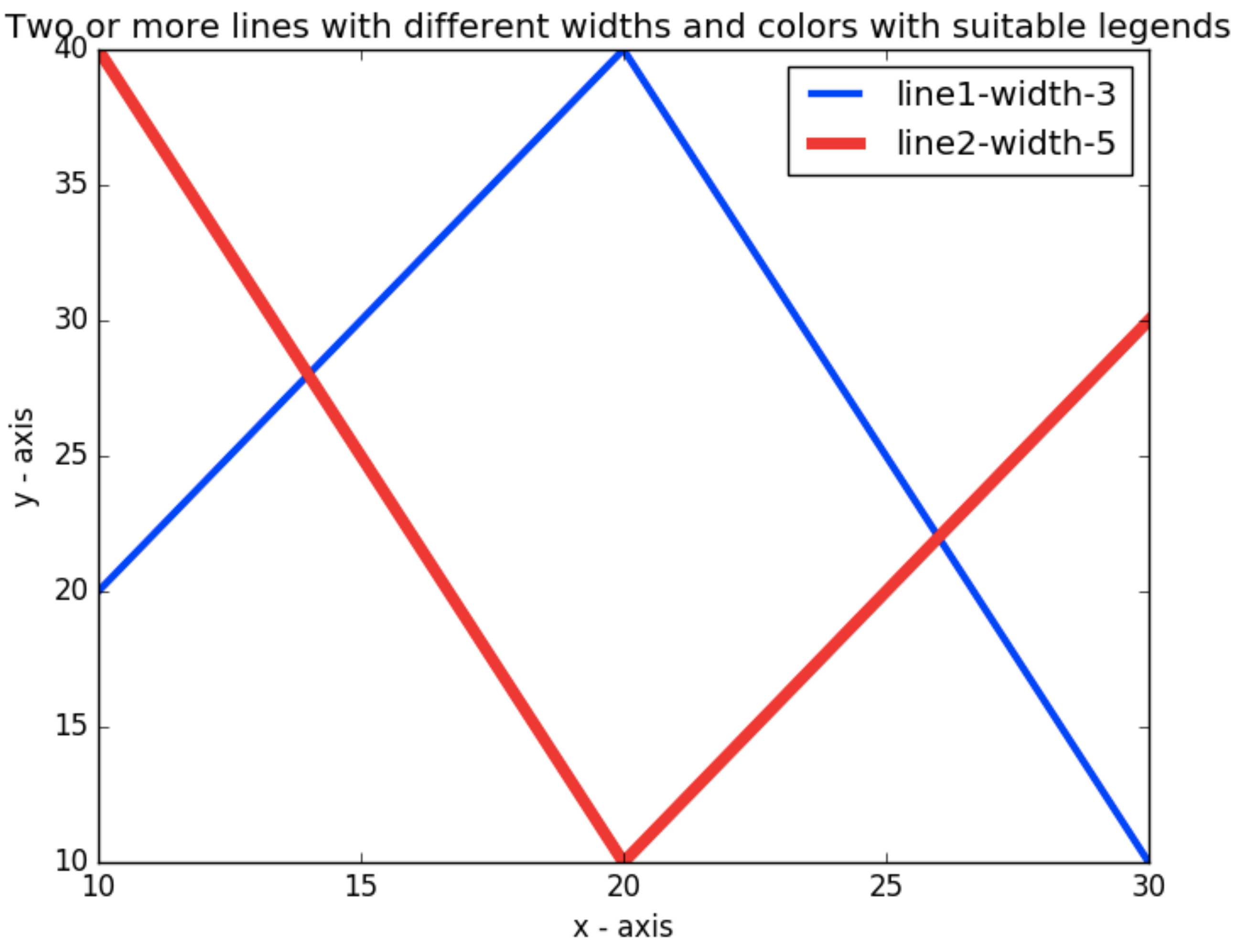


trinket_plot.png

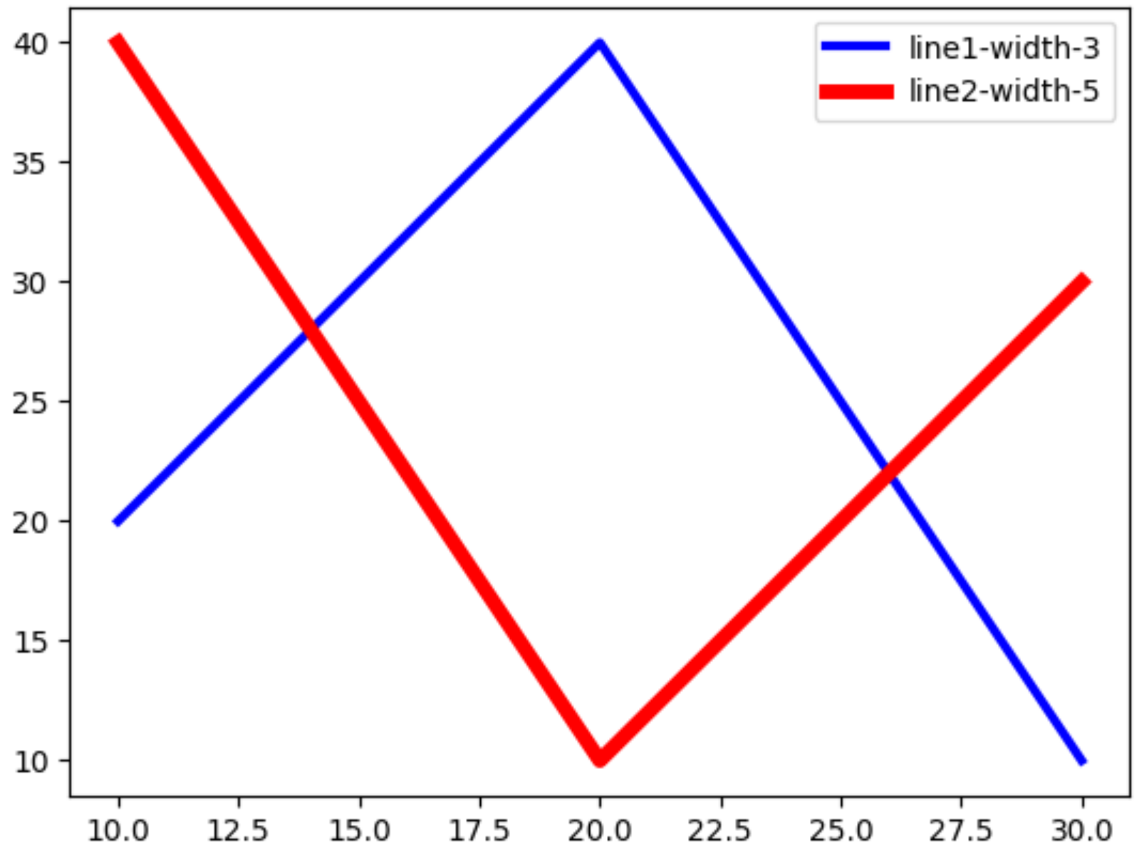
```
import matplotlib.pyplot as plt
import numpy as np
Date = np.array(['3 Oct 2016','4','5','6','7'])
Open = np.array([774.25,776.030029,779.309998,779,779.659973])
High = np.array([776.065002,778.710022,782.070007,780.47998,779.659973])
Low = np.array([769.5,772.890015,775.650024,775.539978,770.75])
Close = np.array([772.559998,776.429993,776.469971,776.859985,775.080017])
plt.plot(Date, Open, label='Open')
plt.plot(Date, High, label='High')
plt.plot(Date, Low, label='Low')
plt.plot(Date, Close, label='Close')
plt.legend()
plt.show()
```



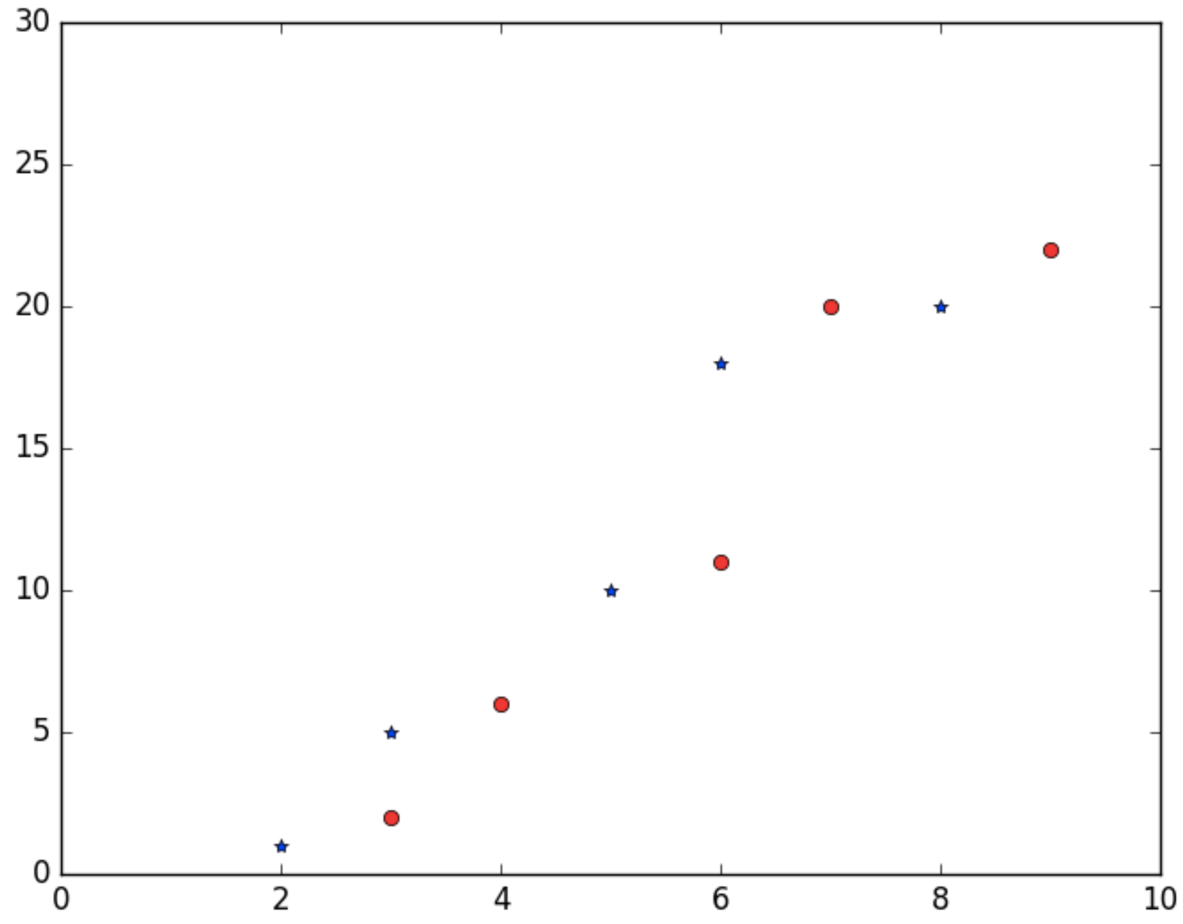
4. Write a Python program to plot two or more lines with legends, different widths and colors. The code snippet gives the output shown in the following screenshot:


Two or more lines with different widths and colors with suitable legends
line1-width-3
line2-width-5
y - axis
x - axis
10
15
20
25
30
35
40
10
15
20
25
30

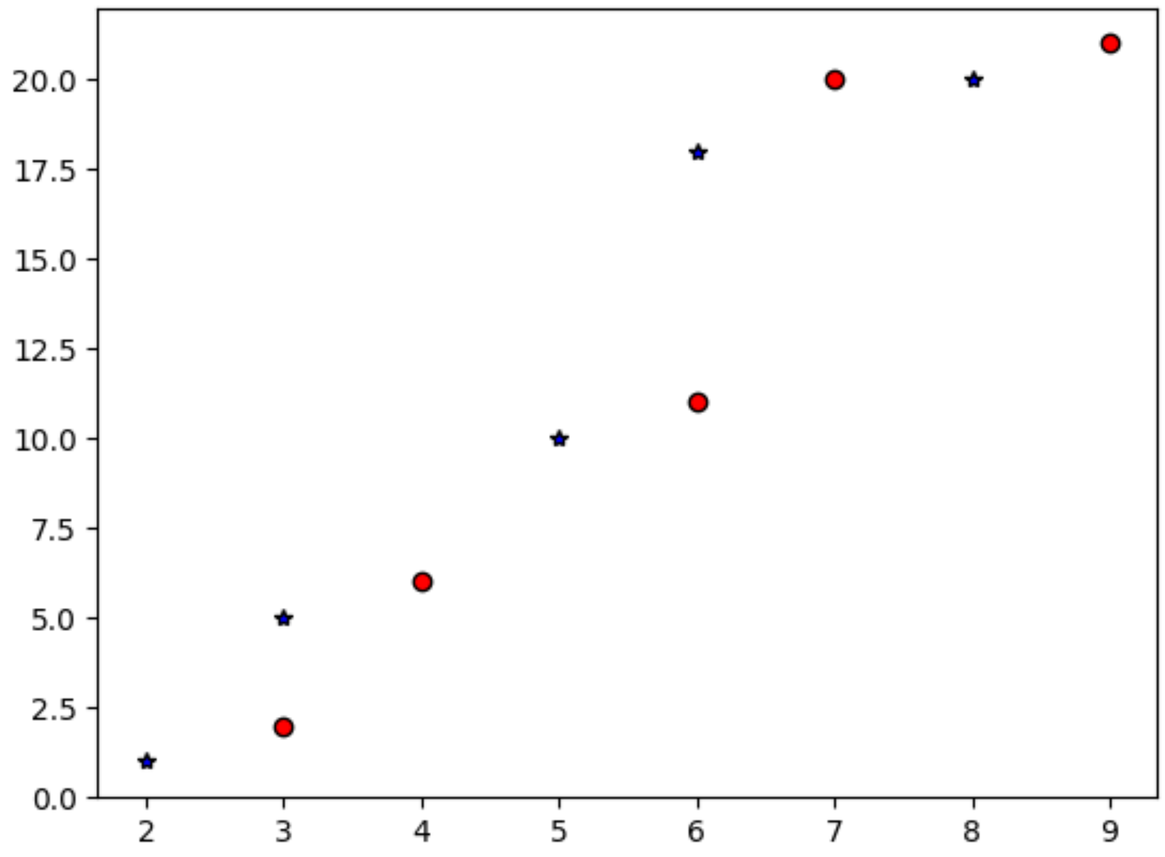
```
import matplotlib.pyplot as plt

x1 = 10,20,30
y1 = 20,40,10
x2 = 10,20,30
y2 = 40,10,30
plt.plot(x1,y1, color='b', lw=3 ,label= 'line1-width-3')
plt.plot(x2,y2, color='r', lw=5, label= 'line2-width-5')
plt.legend()
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:

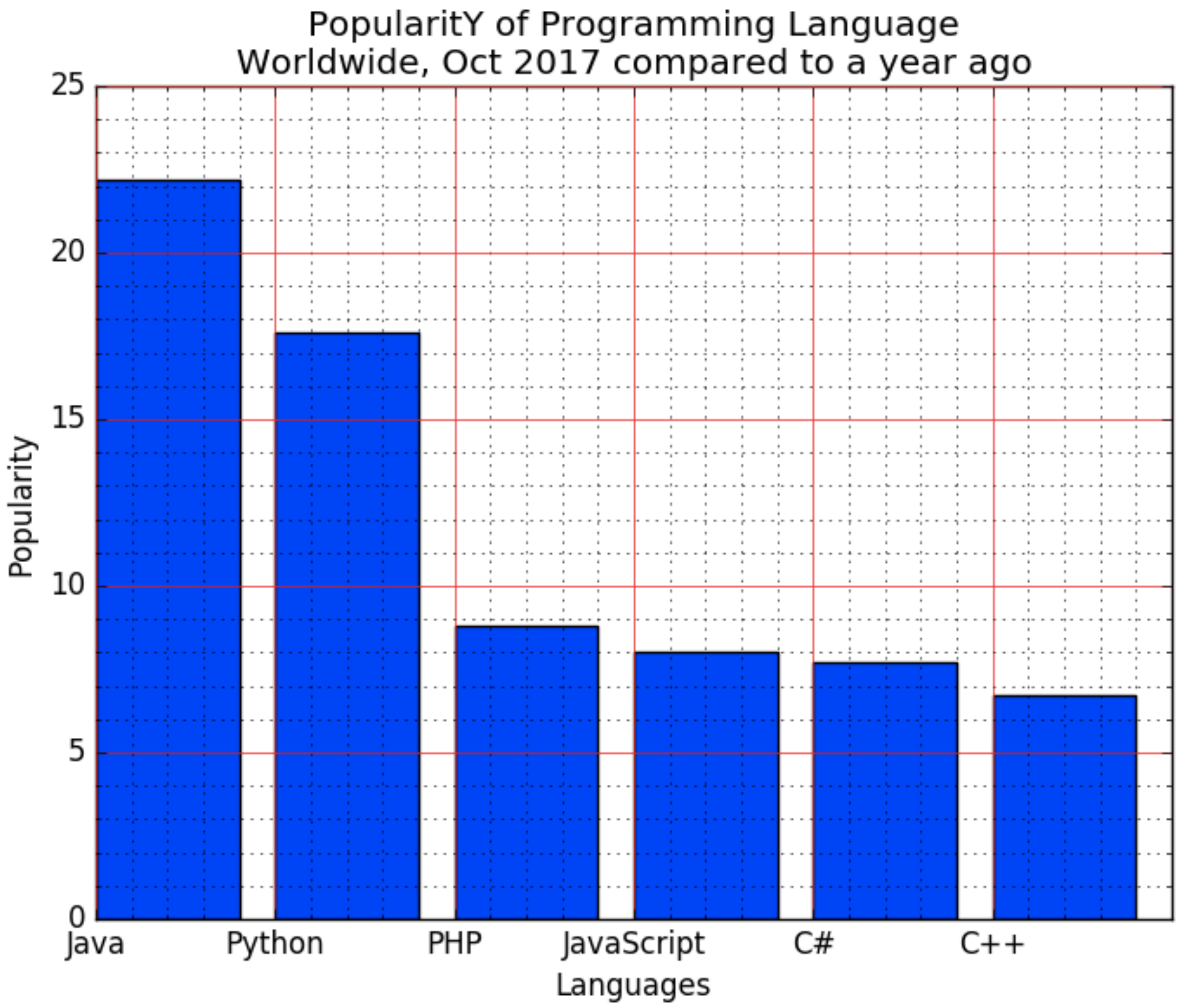

30
25
20
15
10
5
0
0
2
4
6
8
10

```
import matplotlib.pyplot as plt
x1 = 2,3,5,6,8
y1 = 1,5,10,18,20
x2 = 3,4,6,7,9
y2 = 2,6,11,20,21
plt.plot(x1,y1,'*', mec='k', mfc='b')
plt.plot(x2,y2,'o', mec='k', mfc='r')
plt.show()
```

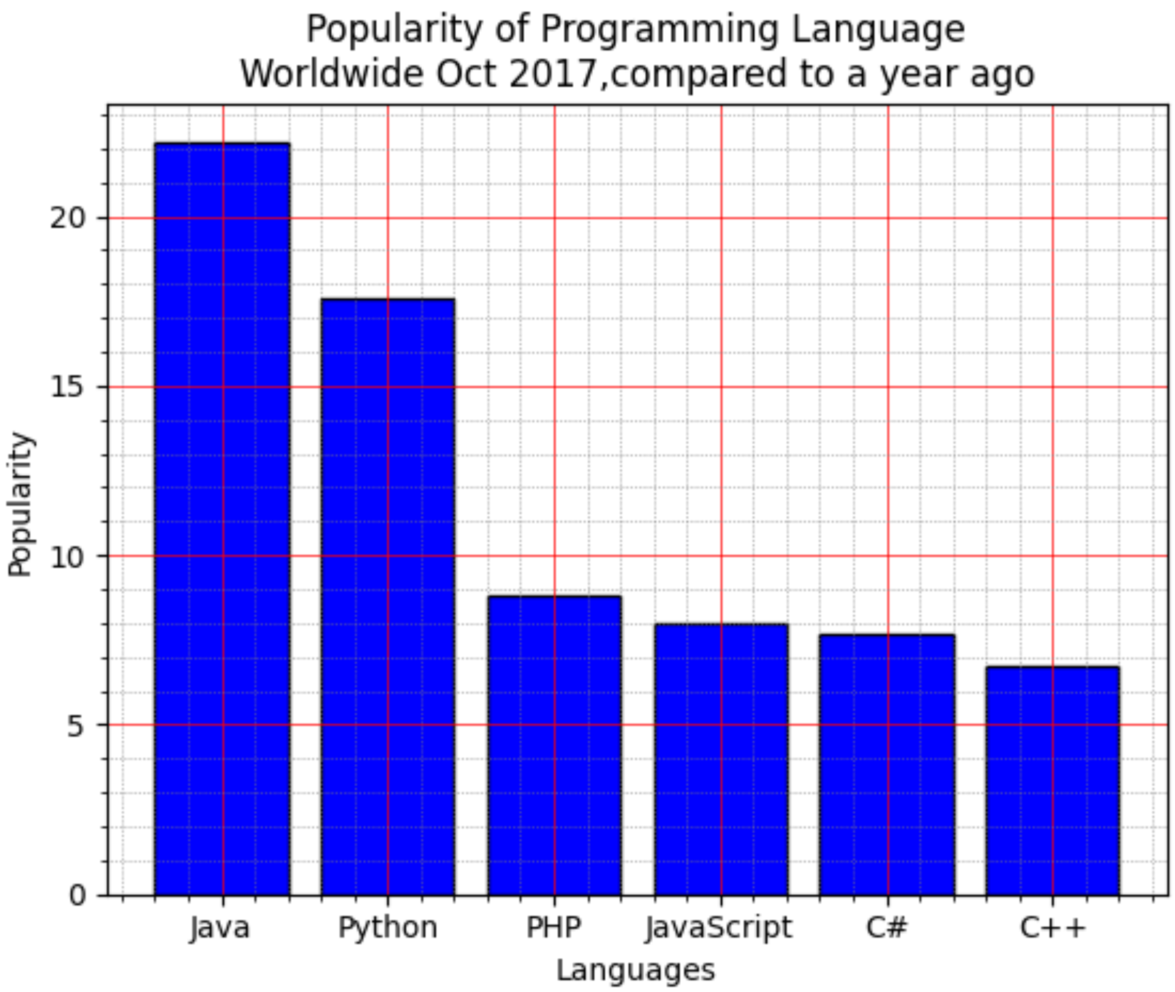


6. Write a Python programming to display a bar chart of the popularity of programming Languages.

```
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:
```

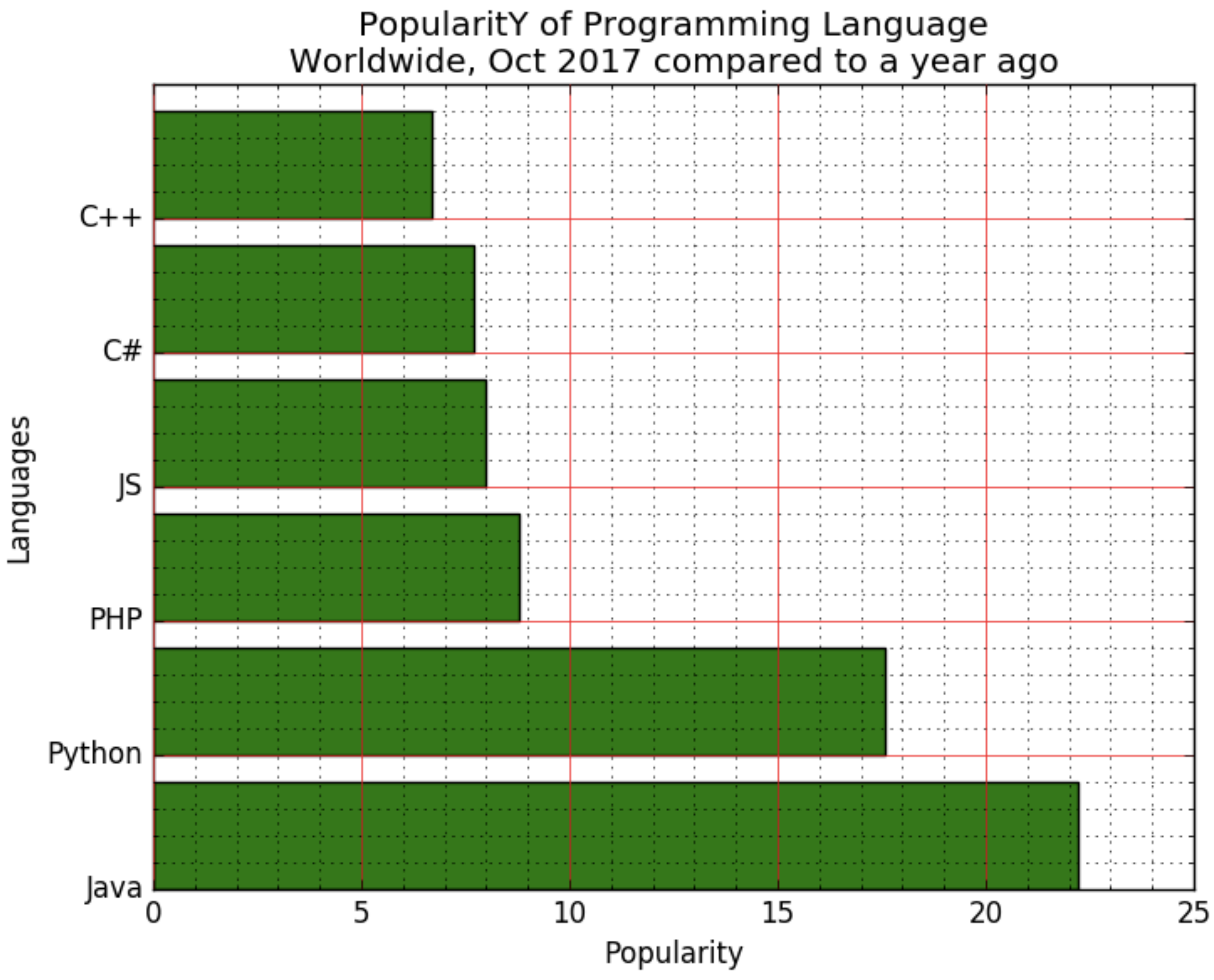

PopularitY of Programming Language
Worldwide, Oct 2017 compared to a year ago
Popularity
Languages
Java
Python
PHP
JavaScript
C#
C++
0
5
10
15
20
25

```
import matplotlib.pyplot as plt
x = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
y = 22.2, 17.6, 8.8, 8, 7.7, 6.7
plt.bar(x,y,color= 'b', ec='k')
plt.xlabel('Languages')
plt.ylabel('Popularity')
plt.title('Popularity of Programming Language\nWorldwide Oct 2017,compared to a
plt.grid(color='r', which= 'major', lw='0.5')
plt.grid(which= 'minor', linestyle= ':', color= 'gray', lw='0.5')
plt.minorticks_on()

plt.show()
```

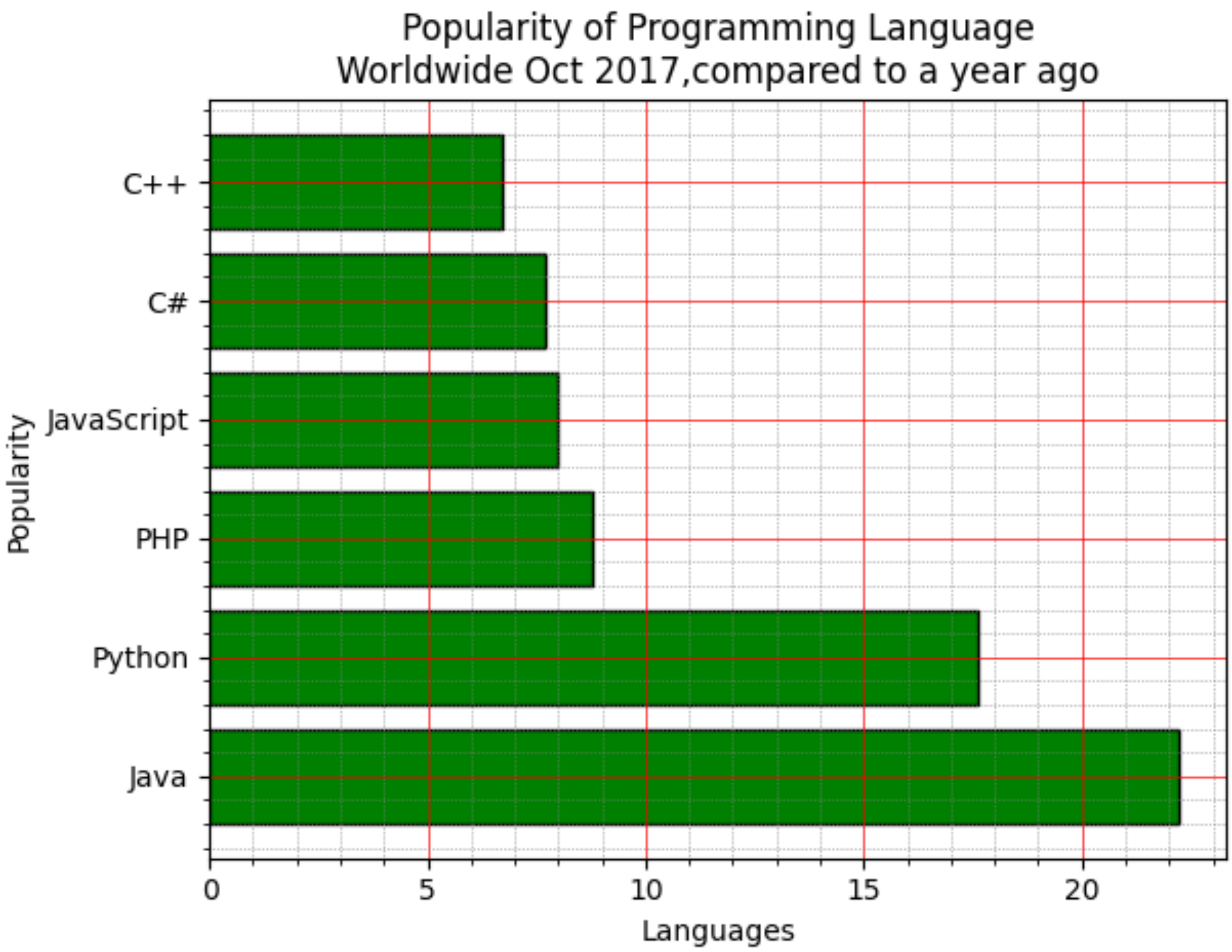


7. Write a Python programming to display a horizontal bar chart of the popularity of programming Languages.

```
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:
```



```
import matplotlib.pyplot as plt
x = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
y = 22.2, 17.6, 8.8, 8, 7.7, 6.7
plt.barh(x,y,color= 'g', ec='k')
plt.xlabel('Languages')
plt.ylabel('Popularity')
plt.title('Popularity of Programming Language\nWorldwide Oct 2017,compared to a
plt.grid(color='r', which= 'major', lw='0.5')
plt.grid(which= 'minor', linestyle= ':', color= 'gray', lw='0.5')
plt.minorticks_on()

plt.show()
```



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

```
!wget https://pynative.com/wp-content/uploads/2019/01/company_sales_data.csv
```

```
--2023-12-02 01:54:51--  https://pynative.com/wp-content/uploads/2019/01/co
Resolving pynative.com (pynative.com)... 172.66.43.37, 172.66.40.219, 2606:
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

2023-12-02 01:54:52 (201 MB/s) - ‘company_sales_data.csv’ saved [659/659]
```

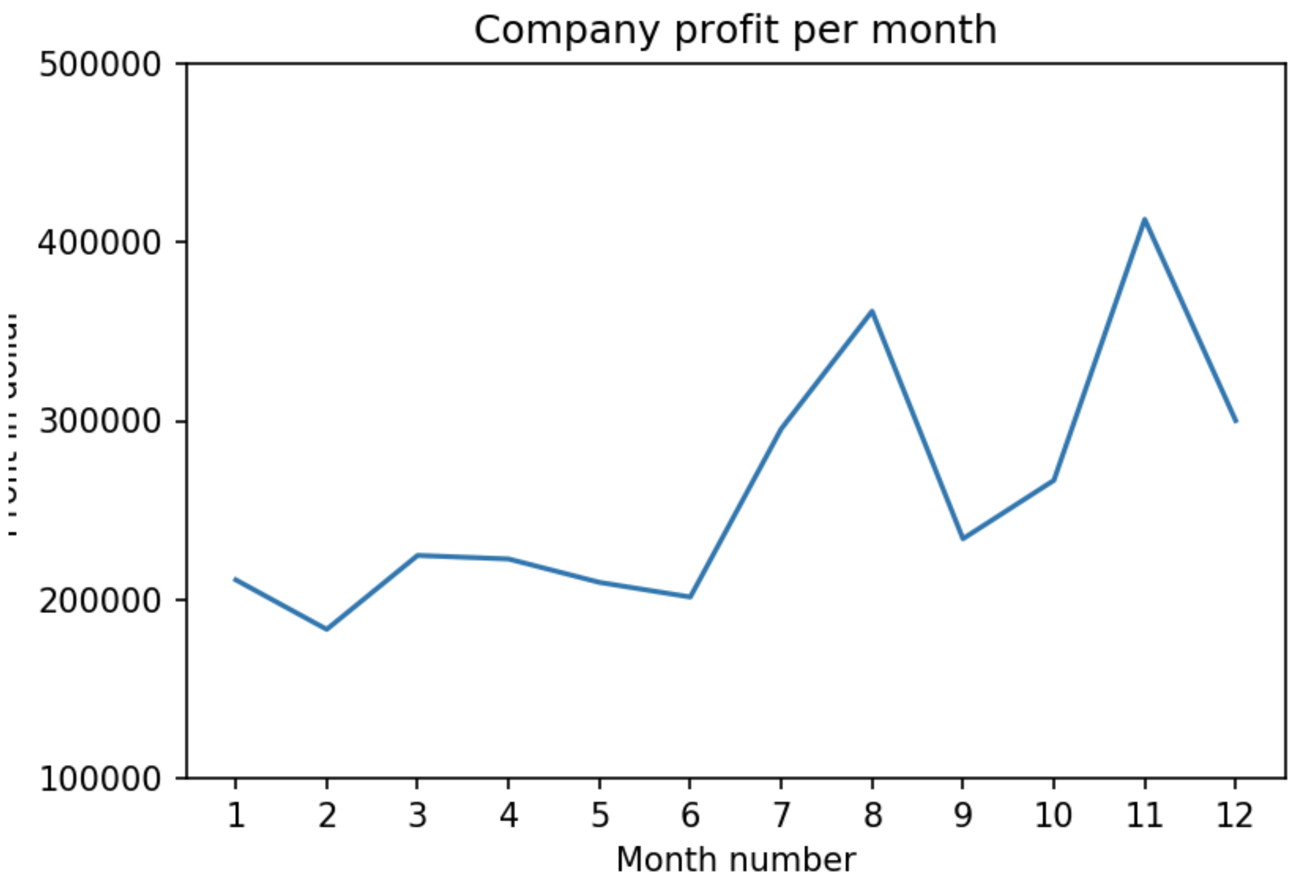
Double-click (or enter) to edit

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

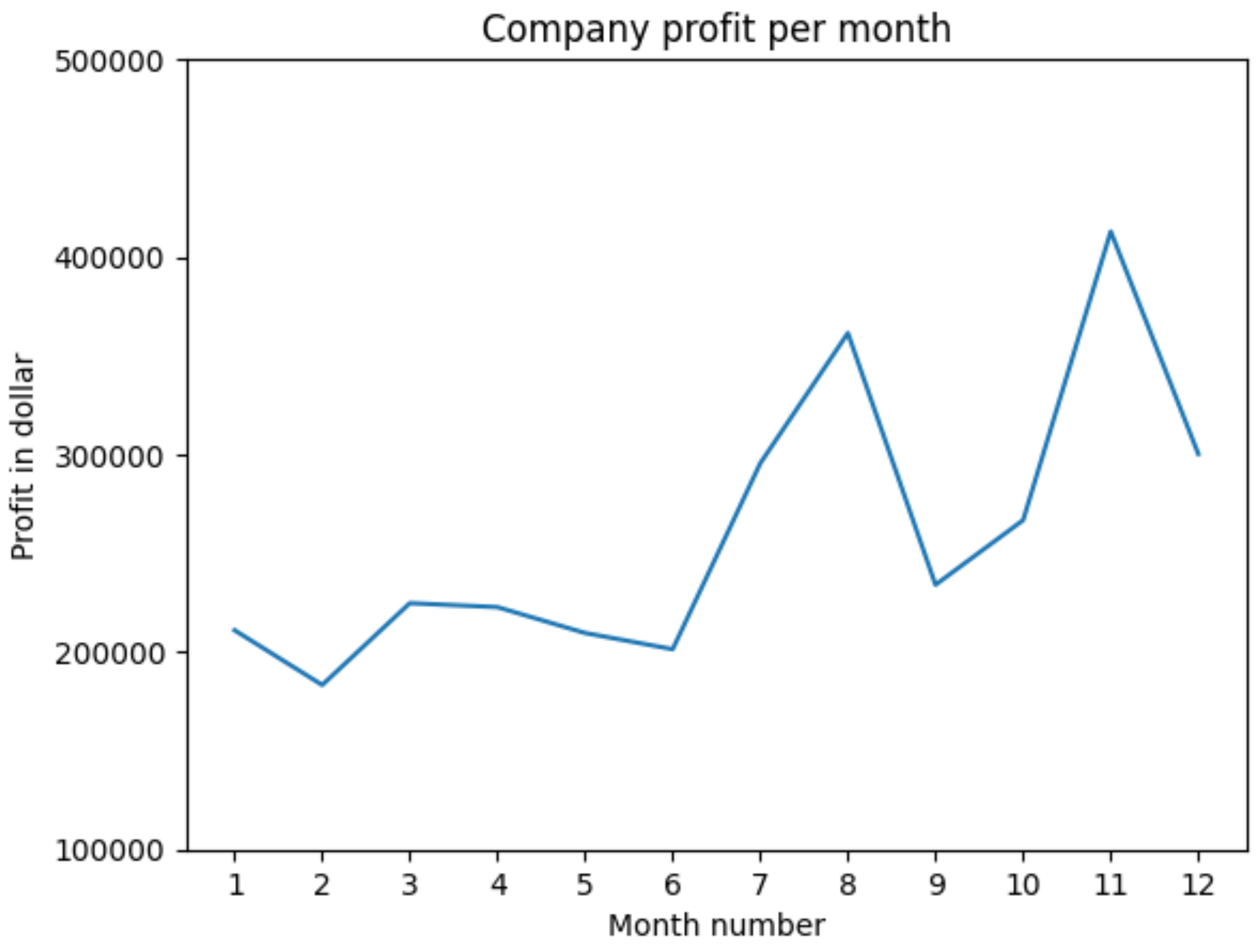
Total profit data provided for each month. Generated line plot must include the following properties: –

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

The line plot graph should look like this.


Company profit per month
500000
400000
300000
200000
100000
1
2
3
4
5
6
7
8
9
10
11
12
Month number

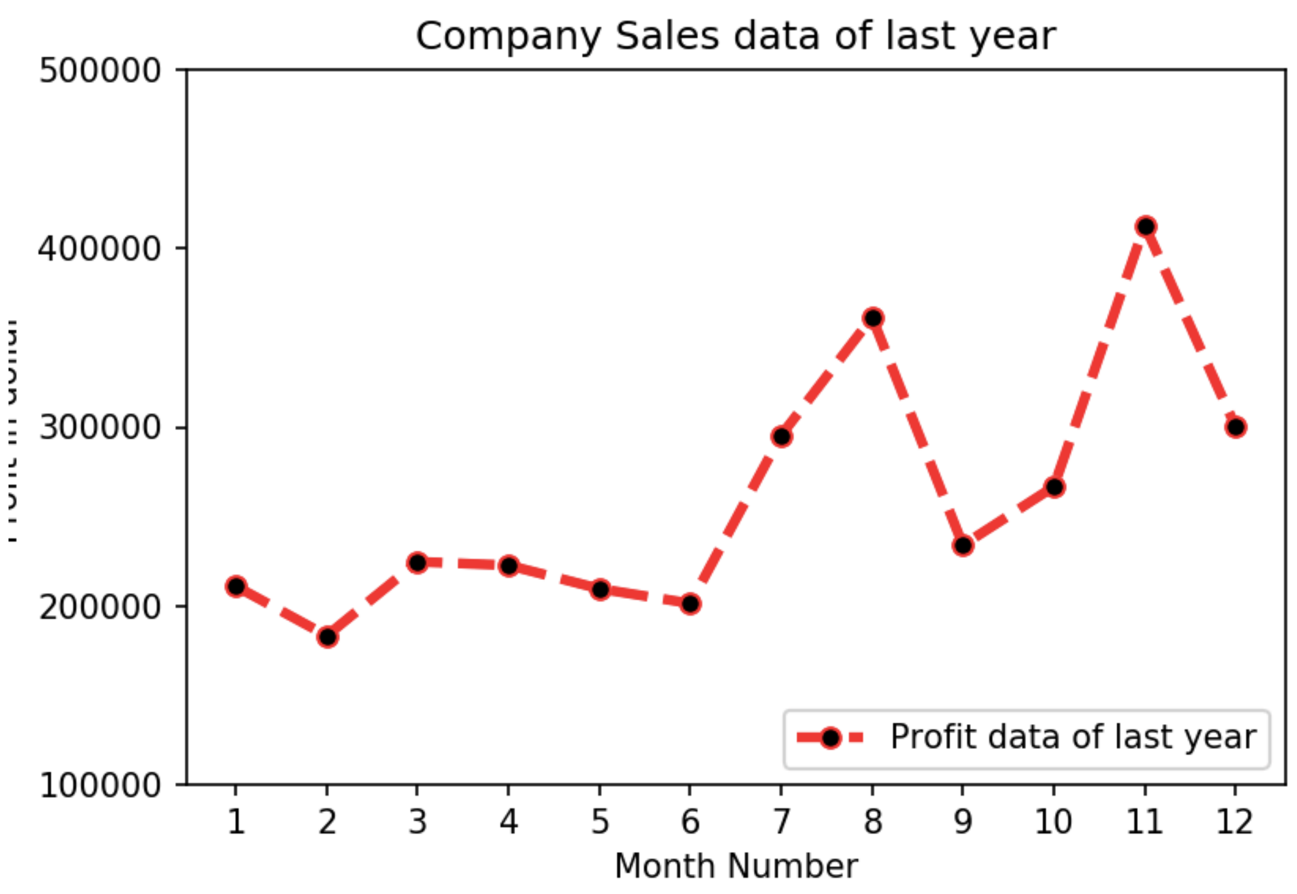
```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
profitList = dataframe ['total_profit'].tolist()
monthList  = dataframe ['month_number'].tolist()
plt.plot(monthList, profitList, label = 'Month-wise Profit data of last year')
plt.xlabel('Month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



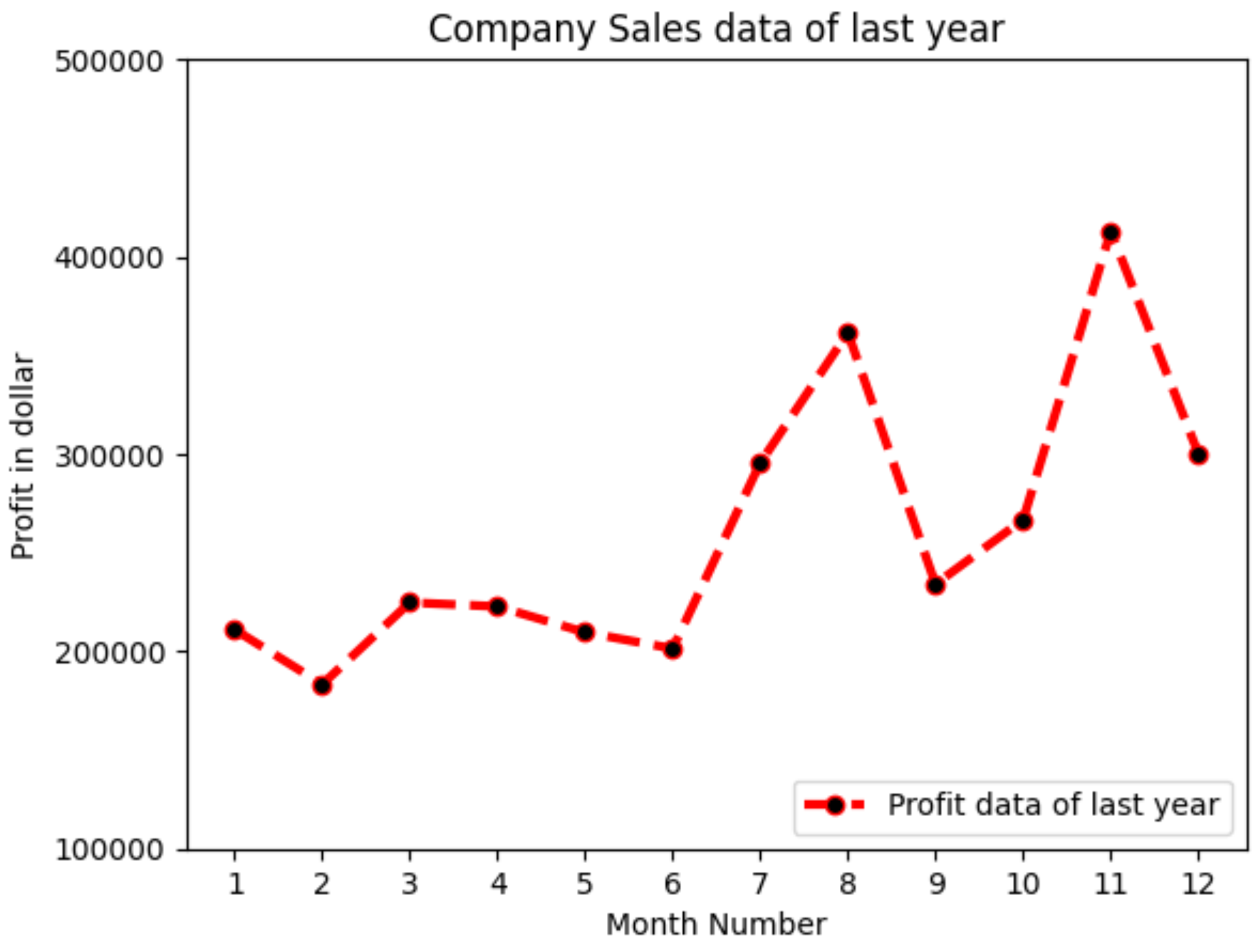
Exercise 2: Get total profit of all months and show line plot with the following Style properties

```
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.



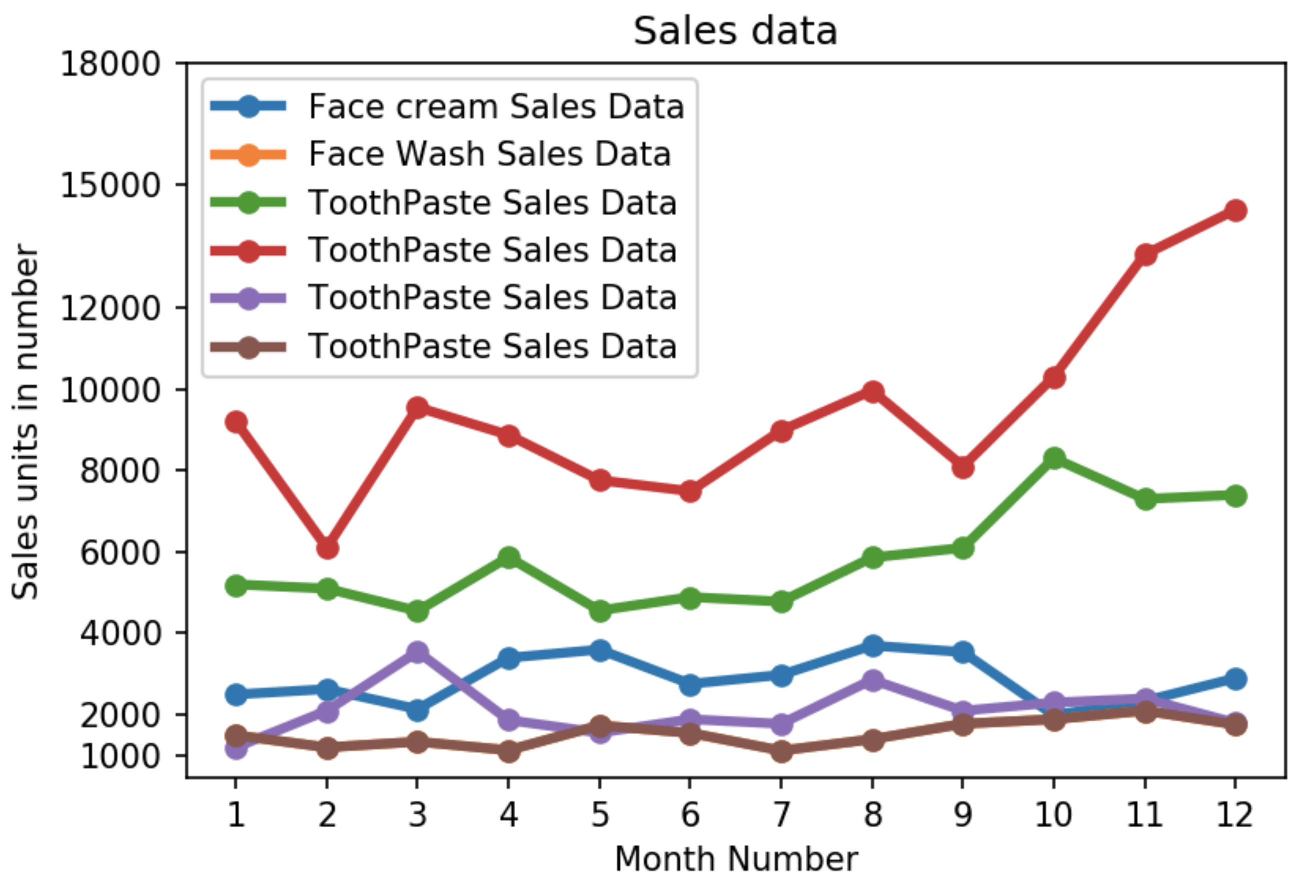
```
import pandas as pd
import matplotlib.pyplot as plt

dataframef = pd.read_csv("company_sales_data.csv")
profitList = dataframe ['total_profit'].tolist()
monthList  = dataframe ['month_number'].tolist()
plt.plot(monthList, profitList, label = 'Profit data of last year', color='r',
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()
```



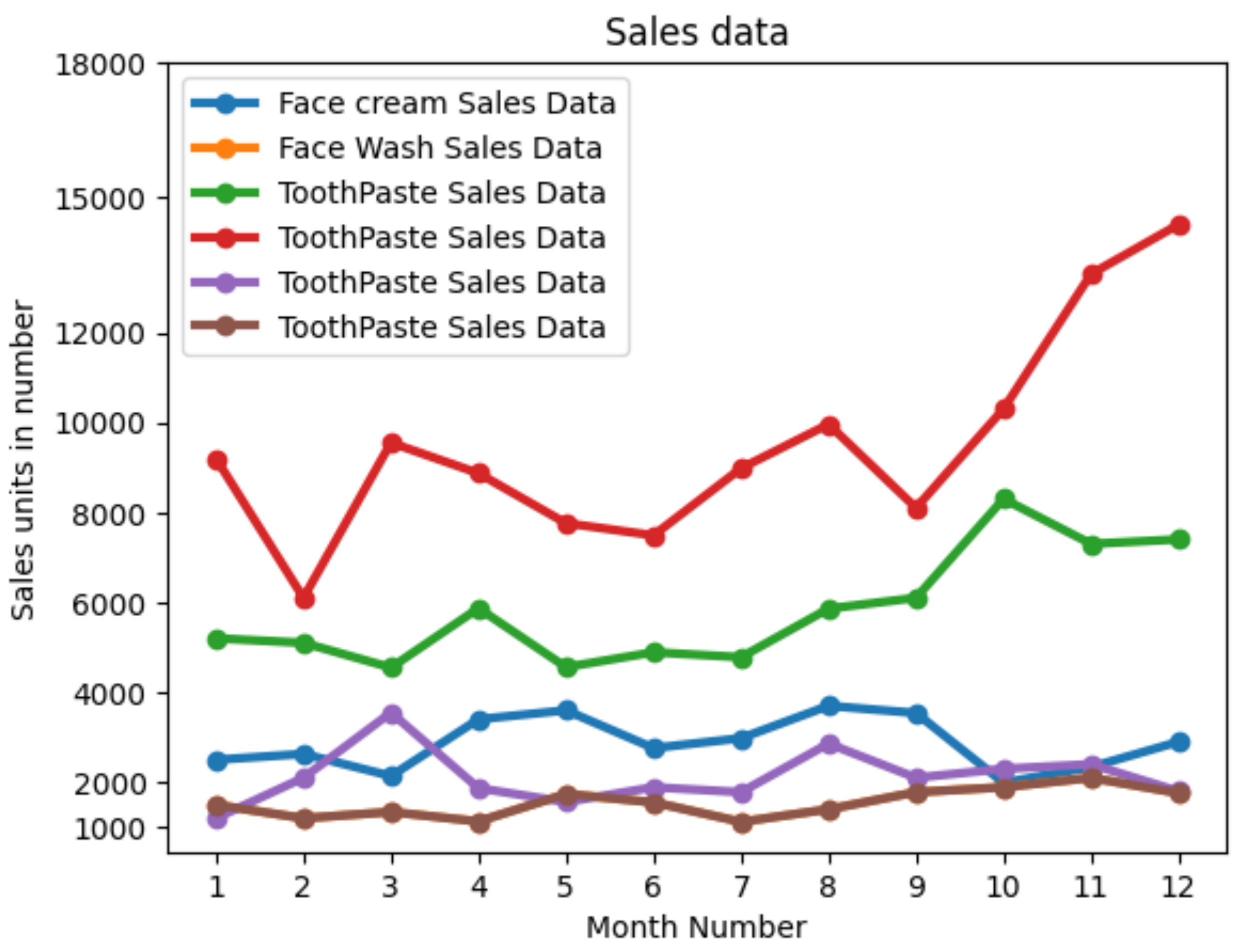
Exercise 3: Read all product sales data and show it using a multiline plot

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.



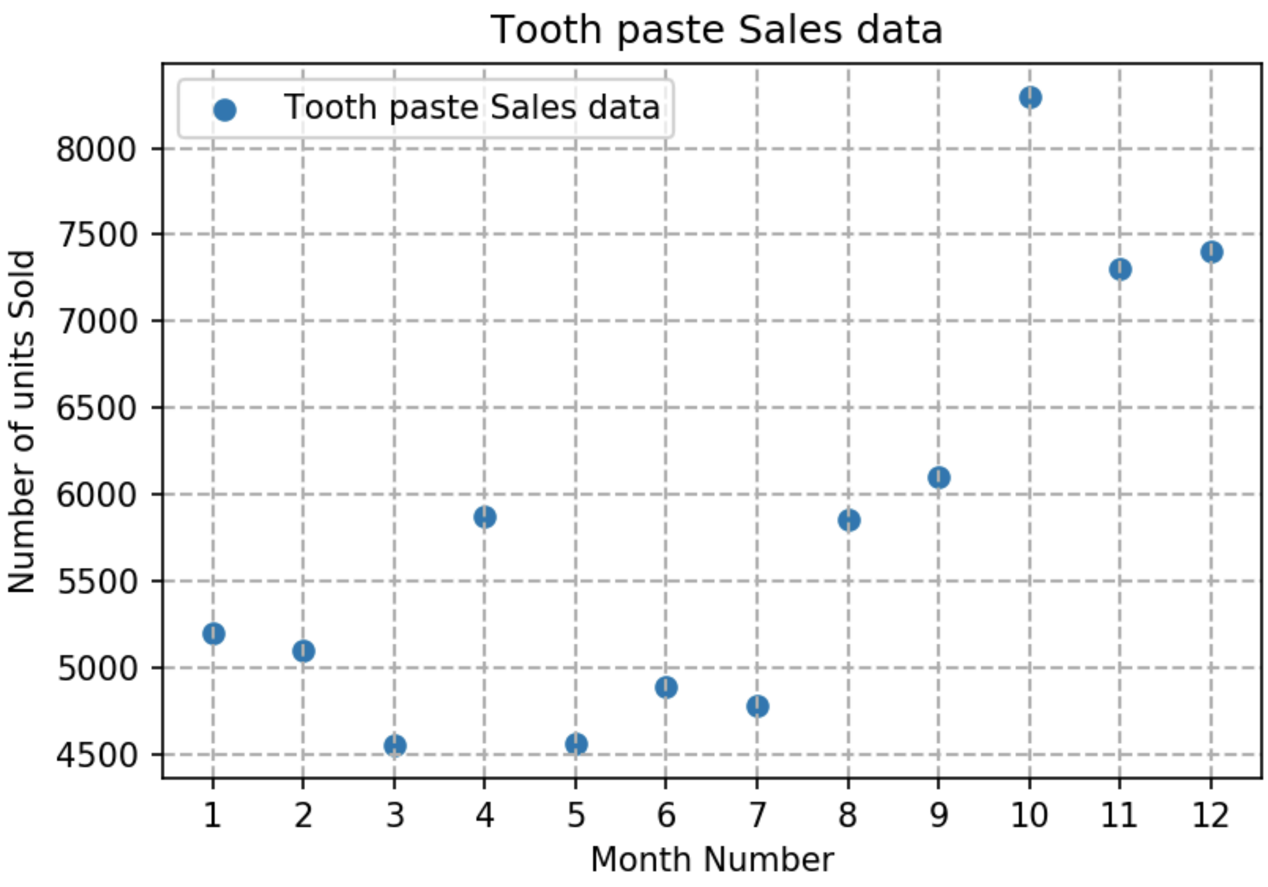
```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
monthList  = dataframe ['month_number'].tolist()
faceCream= dataframe ['facecream'].tolist()
facewash= dataframe ['facewash'].tolist()
toothPaste= dataframe ['toothpaste'].tolist()
bathingsoap= dataframe ['bathingsoap'].tolist()
shampoo= dataframe ['shampoo'].tolist()
moisturizer= dataframe ['moisturizer'].tolist()
plt.plot(monthList, faceCream,   label = 'Face cream Sales Data', marker='o', l
plt.plot(monthList, facewash,   label = 'Face Wash Sales Data',  marker='o', li
plt.plot(monthList, toothPaste, label = 'ToothPaste Sales Data', marker='o', li
plt.plot(monthList, bathingsoap, label = 'ToothPaste Sales Data', marker='o', l
plt.plot(monthList, shampoo, label = 'ToothPaste Sales Data', marker='o', linew
plt.plot(monthList, moisturizer, label = 'ToothPaste Sales Data', marker='o', l
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()
```



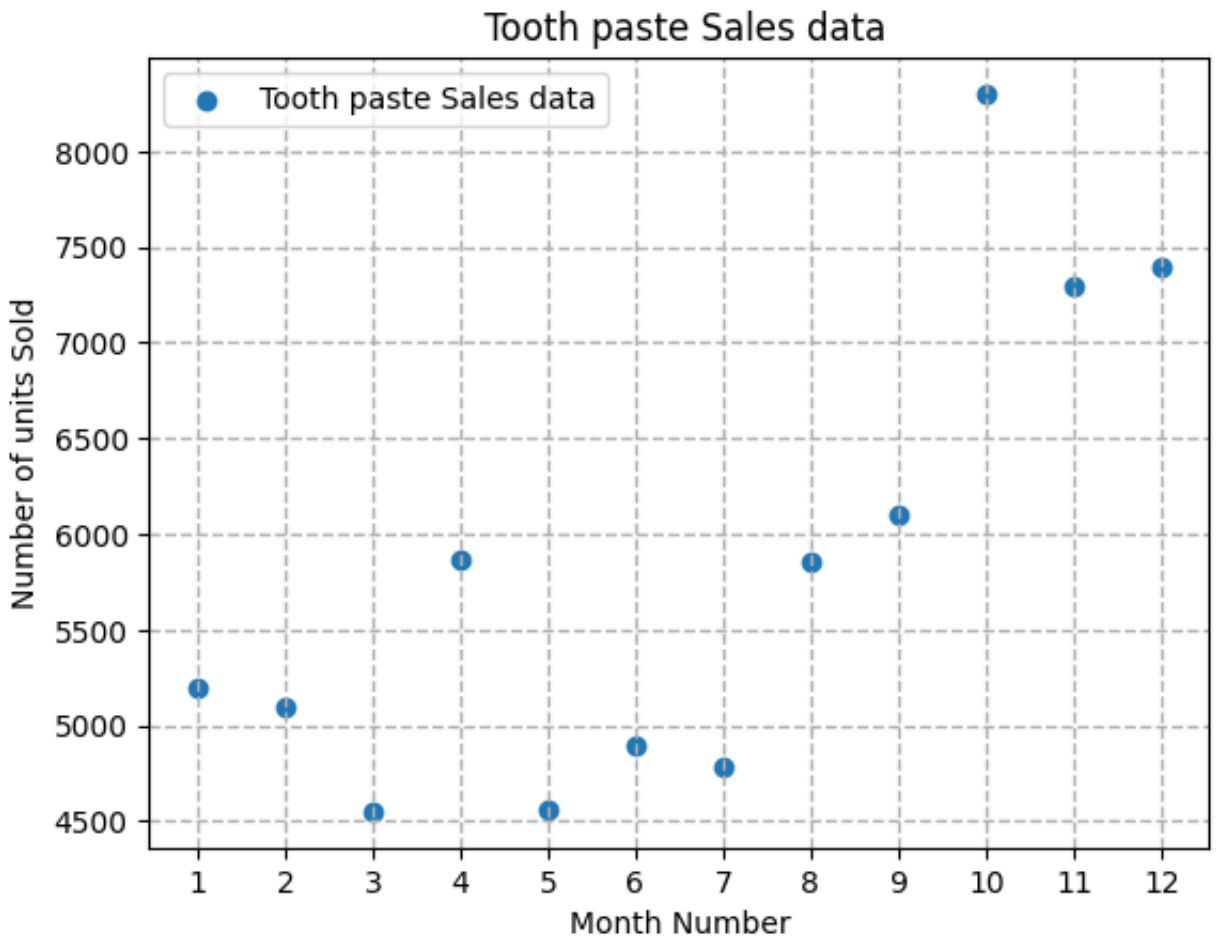
Exercise 4: Read toothpaste sales data of each month and show it using a scatter plot

Also, add a grid in the plot. gridline style should “–“.

The scatter plot should look like this.


Tooth paste Sales data
● Tooth paste Sales data
Number of units Sold
8000
7500
7000
6500
6000
5500
5000
4500
1
2
3
4
5
6
7
8
9
10
11
12
Month Number

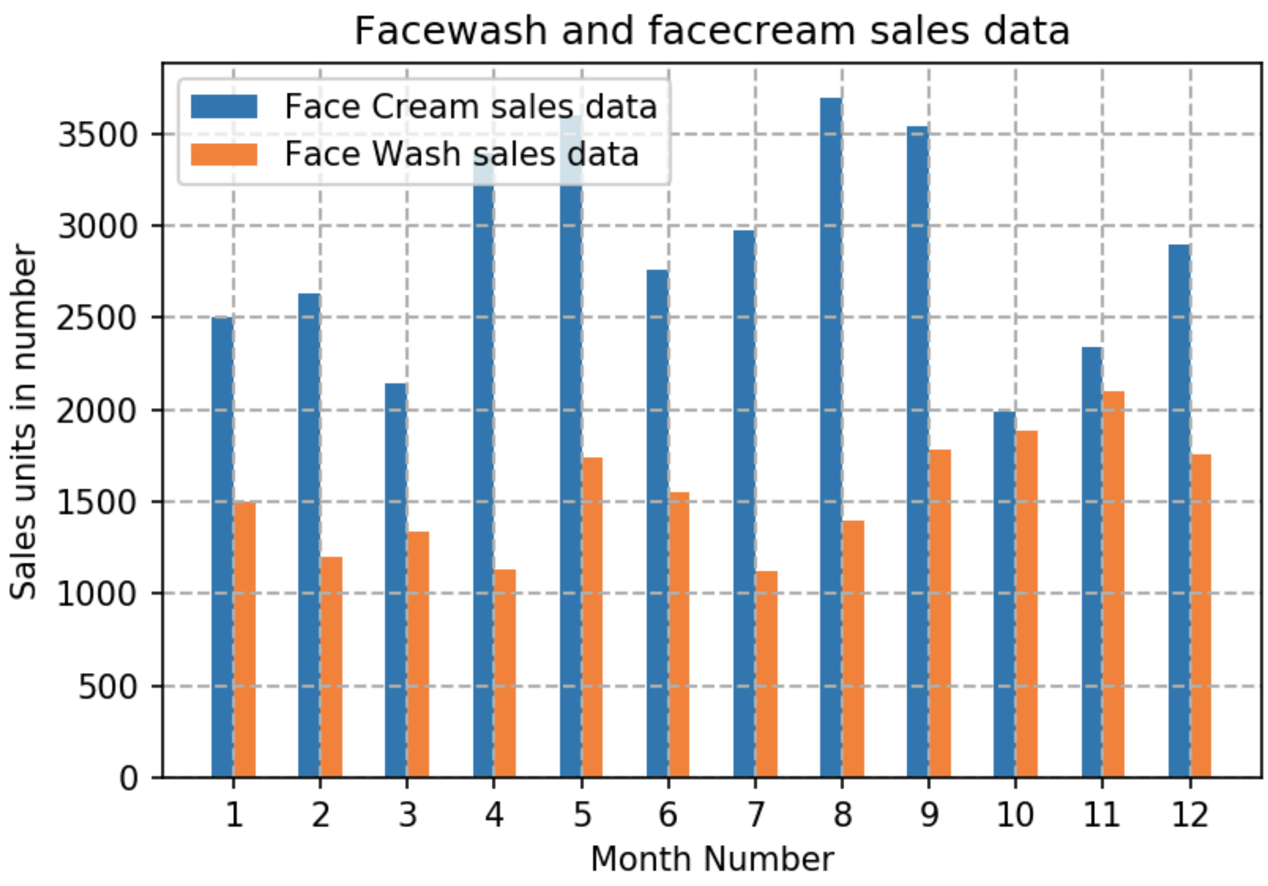
```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
monthList  = dataframe ['month_number'].tolist()
toothPaste= dataframe ['toothpaste'].tolist()
plt.scatter(monthList, toothPaste, label = 'Tooth paste Sales data')
plt.xlabel('Month Number')
plt.ylabel('Number 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()
```



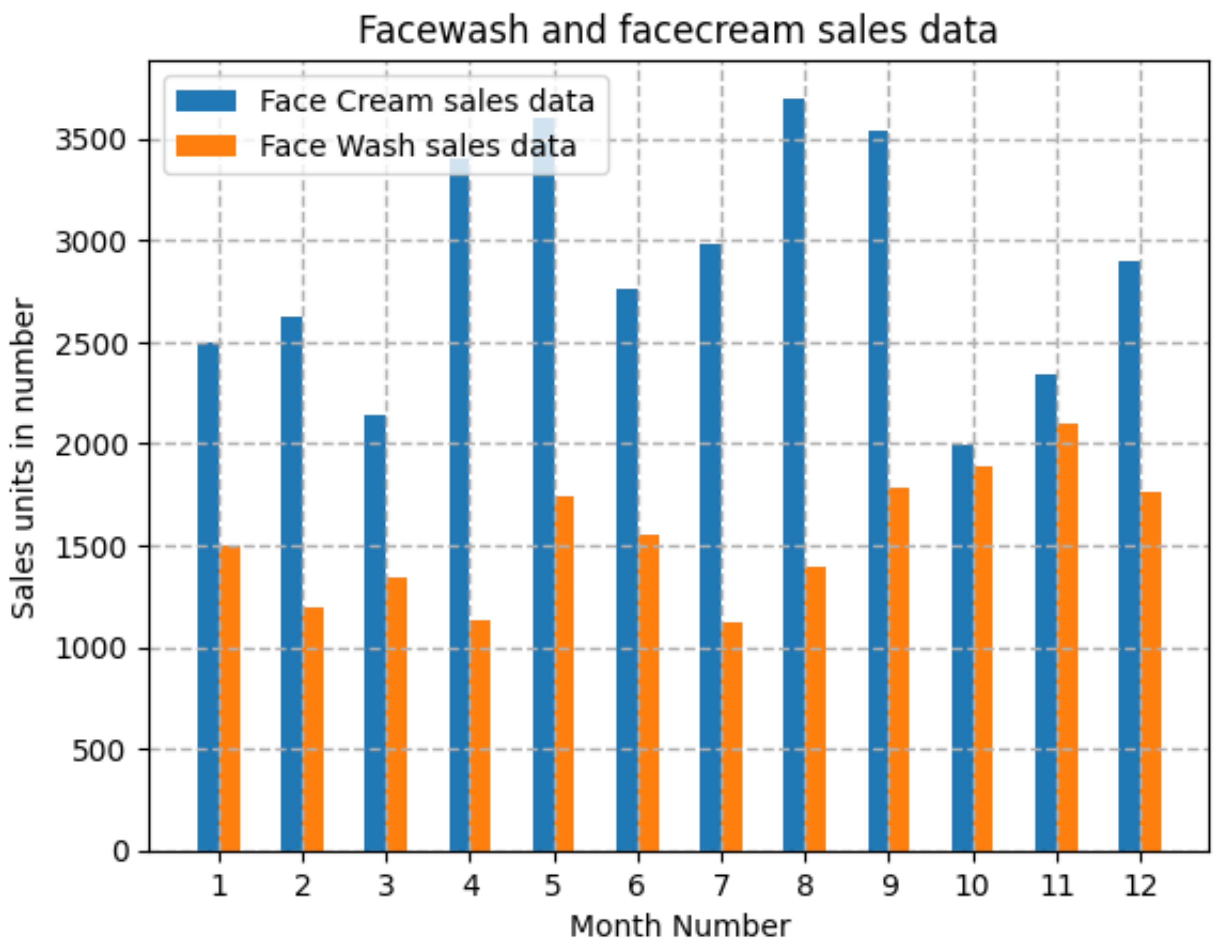
Exercise 5: Read face cream and facewash product sales data and show it using the bar chart

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.

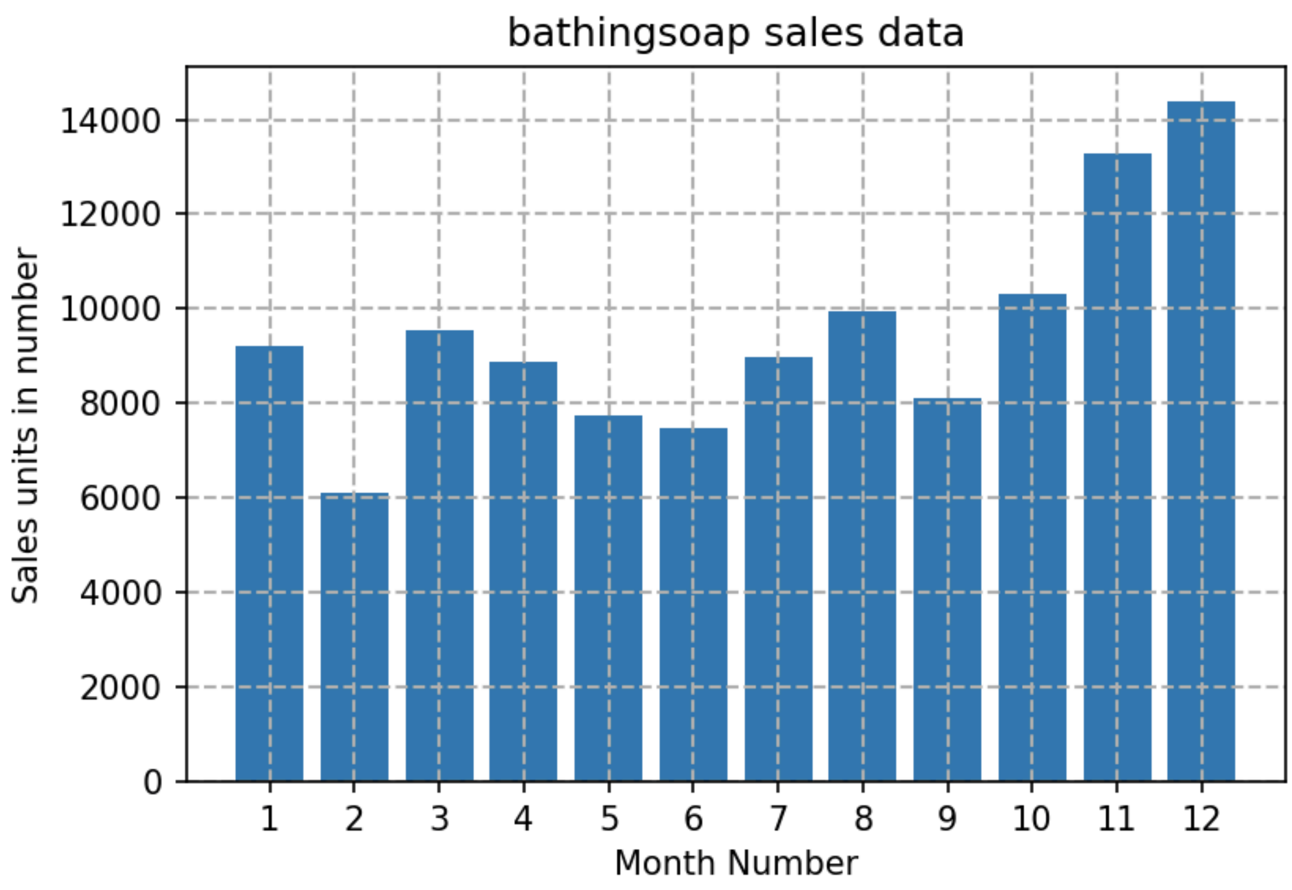


```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
monthList  = dataframe ['month_number'].tolist()
faceCream= dataframe ['facecream'].tolist()
faceWash= dataframe ['facewash'].tolist()
plt.bar([a-0.25 for a in monthList], faceCream, width= 0.25, label = 'Face Crea
plt.bar([a+0.25 for a in monthList], faceWash, width= -0.25, label = 'Face Wash
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()
```

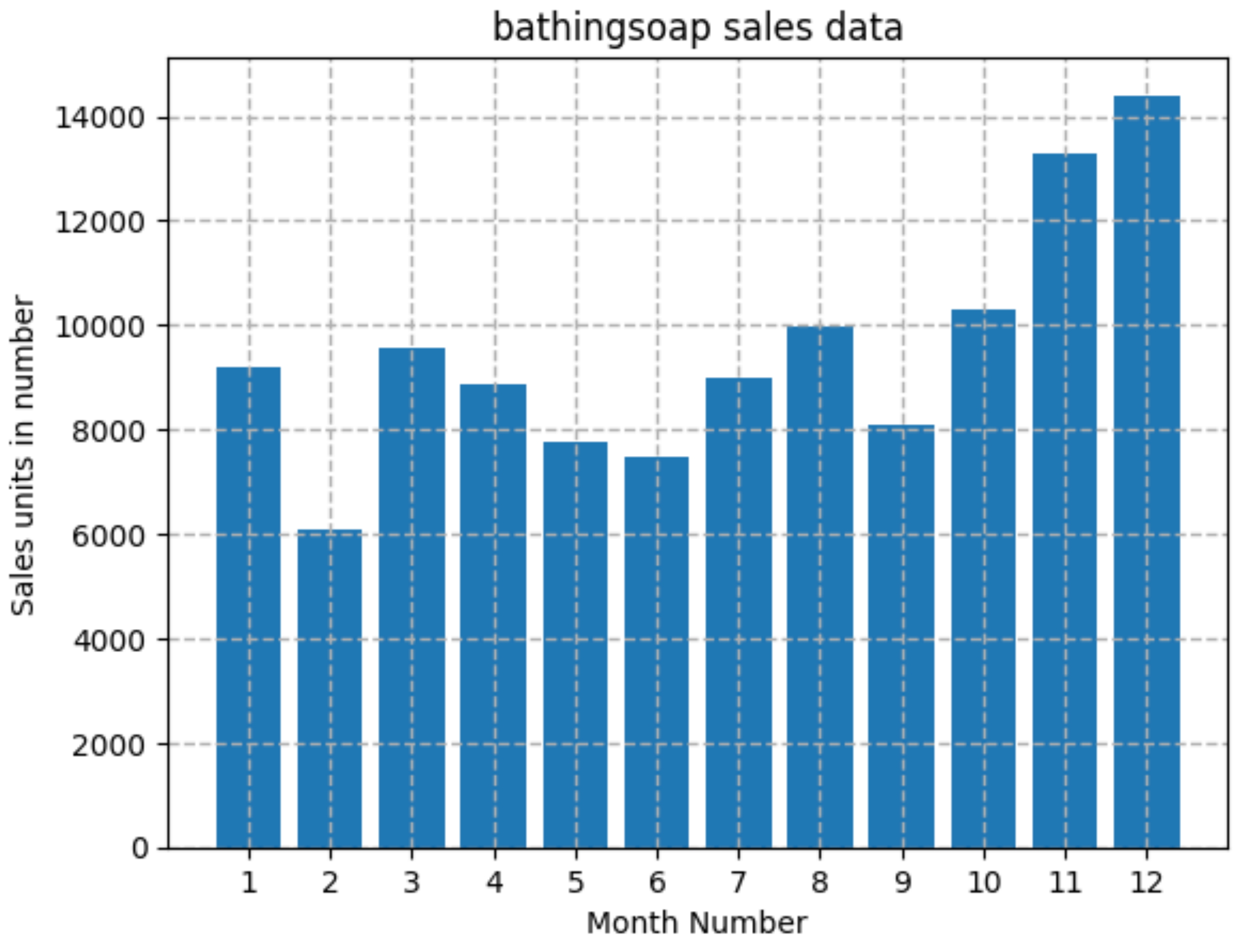


Exercise 6: Read sales data of bathing soap of all months and show it using a bar chart. Save this plot to your hard disk

The bar chart should look like this.

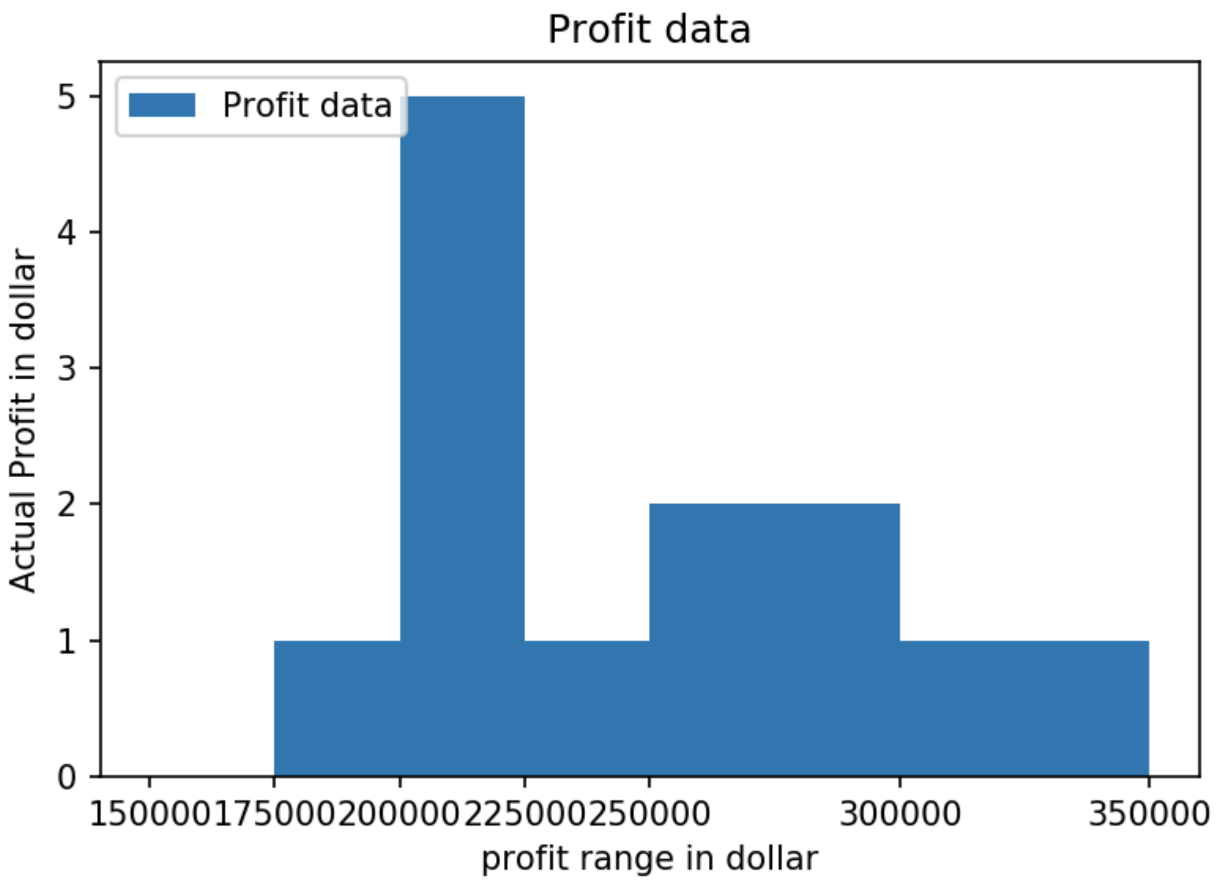


```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
monthList  = dataframe ['month_number'].tolist()
bathingsoap = dataframe ['bathingsoap'].tolist()
plt.bar(monthList, bathingsoap)
plt.xlabel('Month Number')
plt.ylabel('Sales units in number')
plt.title(' 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()
```

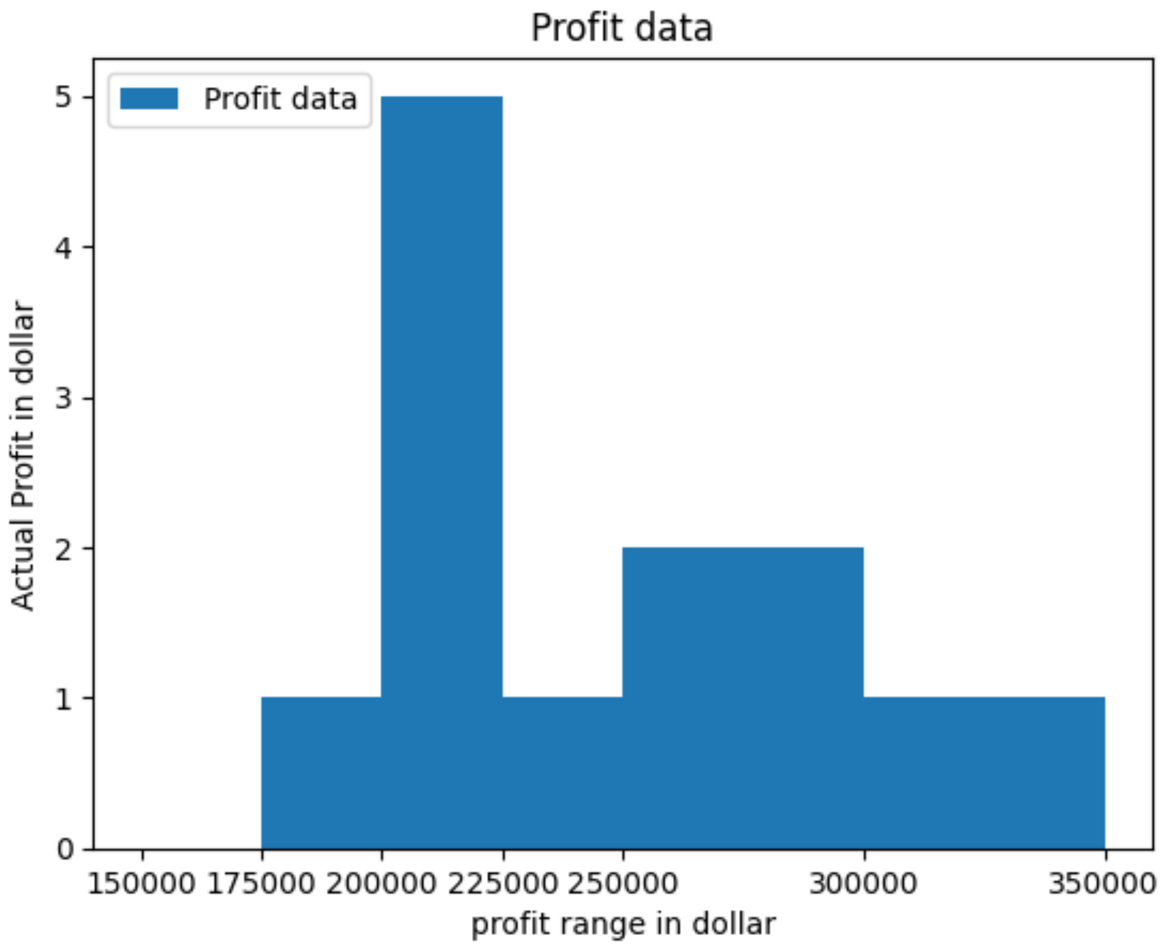


Exercise 7: Read the total profit of each month and show it using the histogram to see the most common profit ranges

The histogram should look like this.



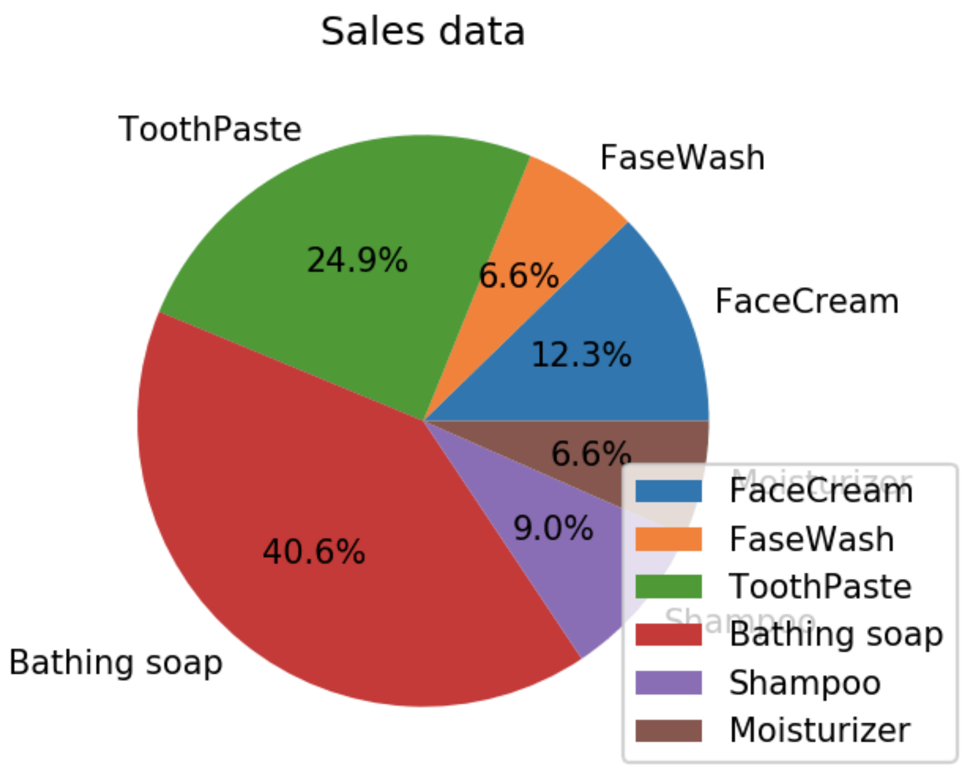
```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
profitList = dataframe ['total_profit'].tolist()
labels = ['low', 'average', 'Good', 'Best']
profit_range = [150000, 175000, 200000, 225000, 250000, 300000, 350000]
plt.hist(profitList, profit_range, 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()
```



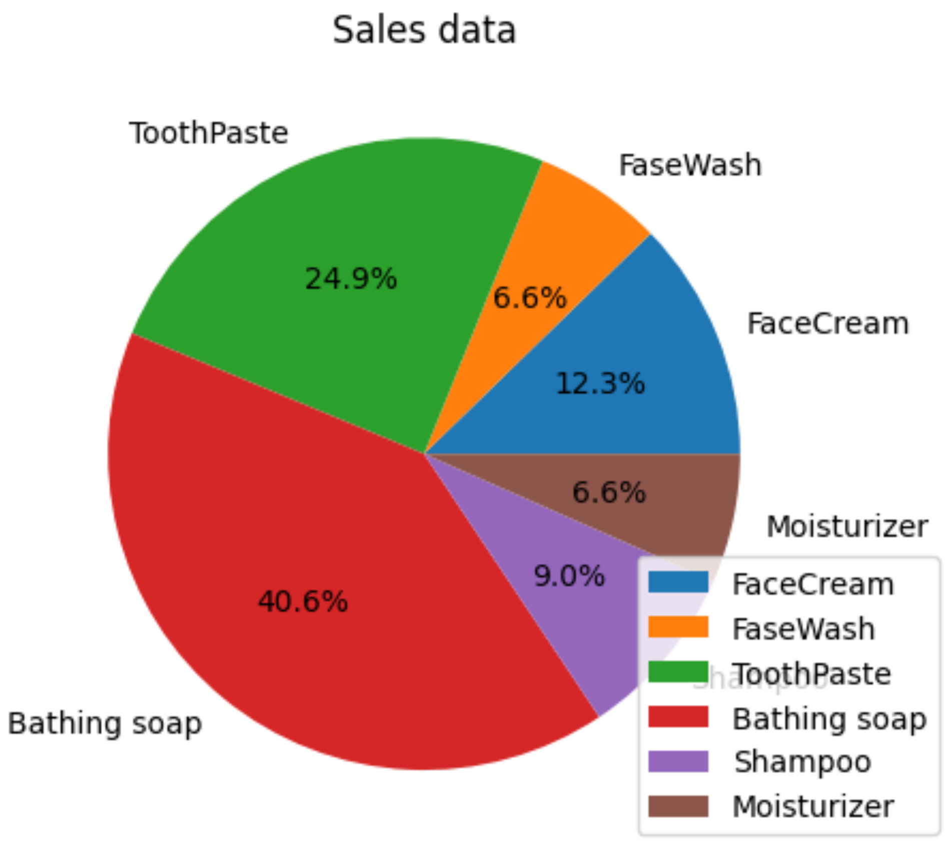
Exercise 8: Calculate total sale data for last year for each product and show it using a Pie chart

Note: In Pie chart display Number of units sold per year for each product in percentage.

The Pie chart should look like this.

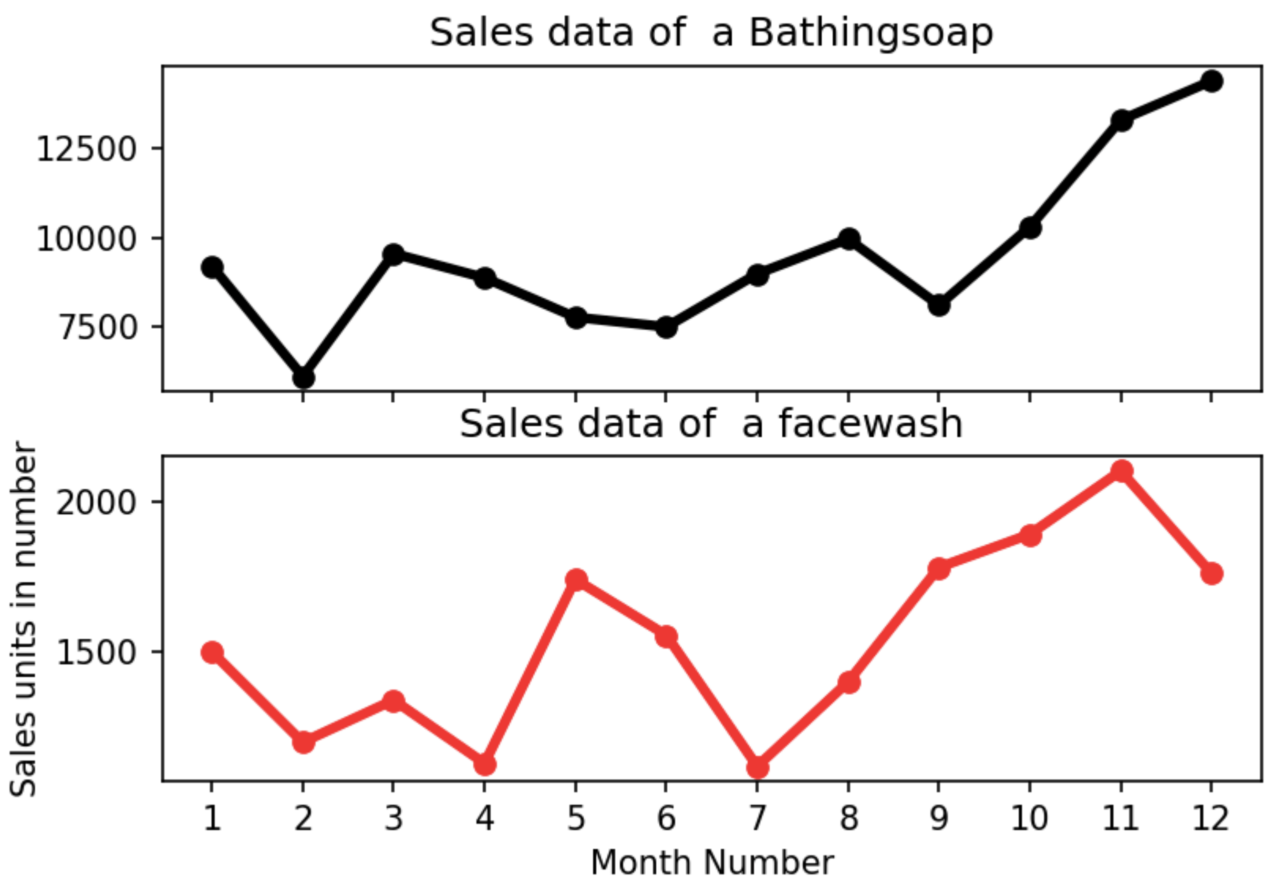


```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
monthList  = dataframe ['month_number'].tolist()
labels = ['FaceCream', 'FaseWash', 'ToothPaste', 'Bathing soap', 'Shampoo', 'Mc
salesData   = [dataframe ['facecream'].sum(), dataframe ['facewash'].sum(), dat
plt.axis("equal")
plt.pie(salesData, labels=labels, autopct='%1.1f%%')
plt.legend(loc='lower right')
plt.title('Sales data')
plt.show()
```

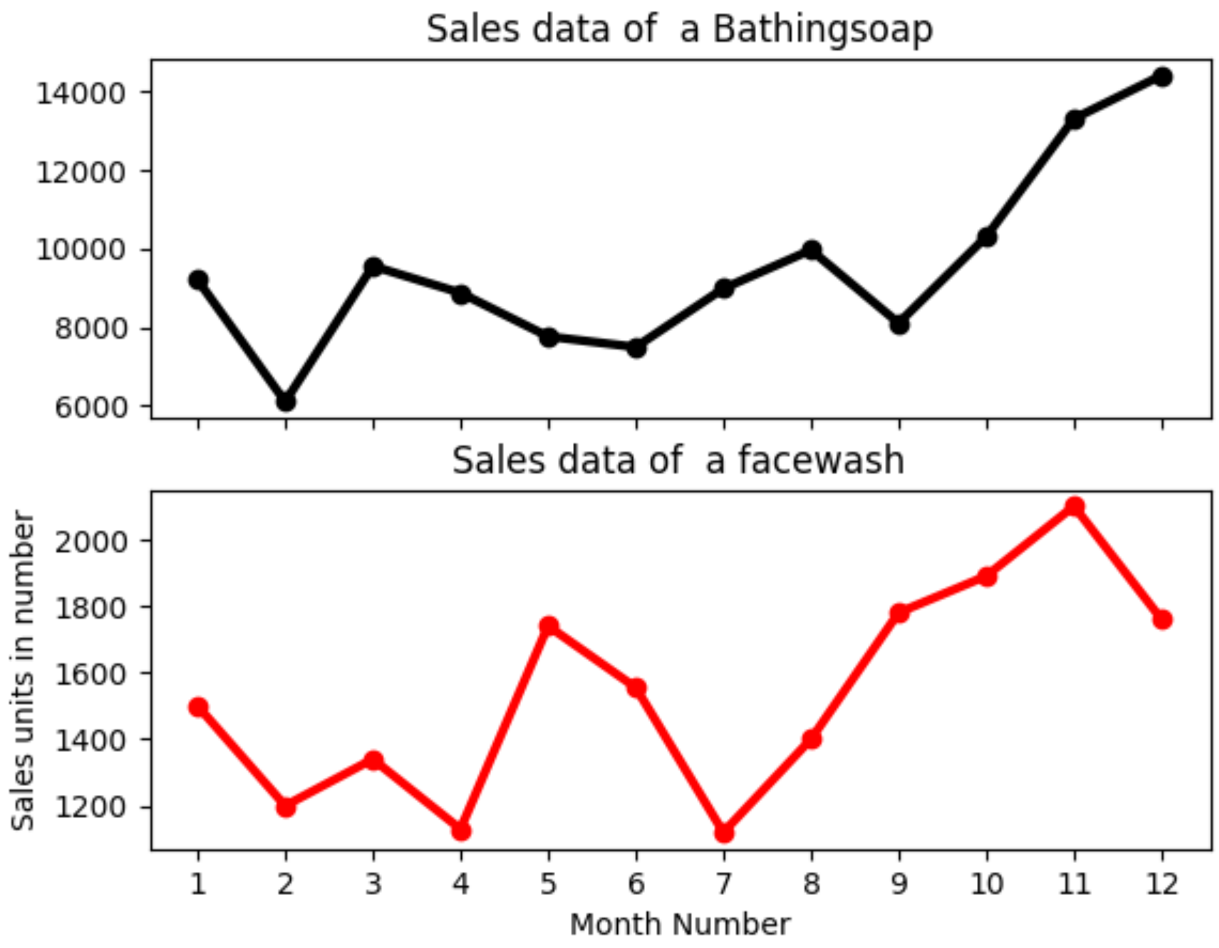


Exercise 9: Read Bathing soap facewash of all months and display it using the Subplot

The Subplot should look like this.

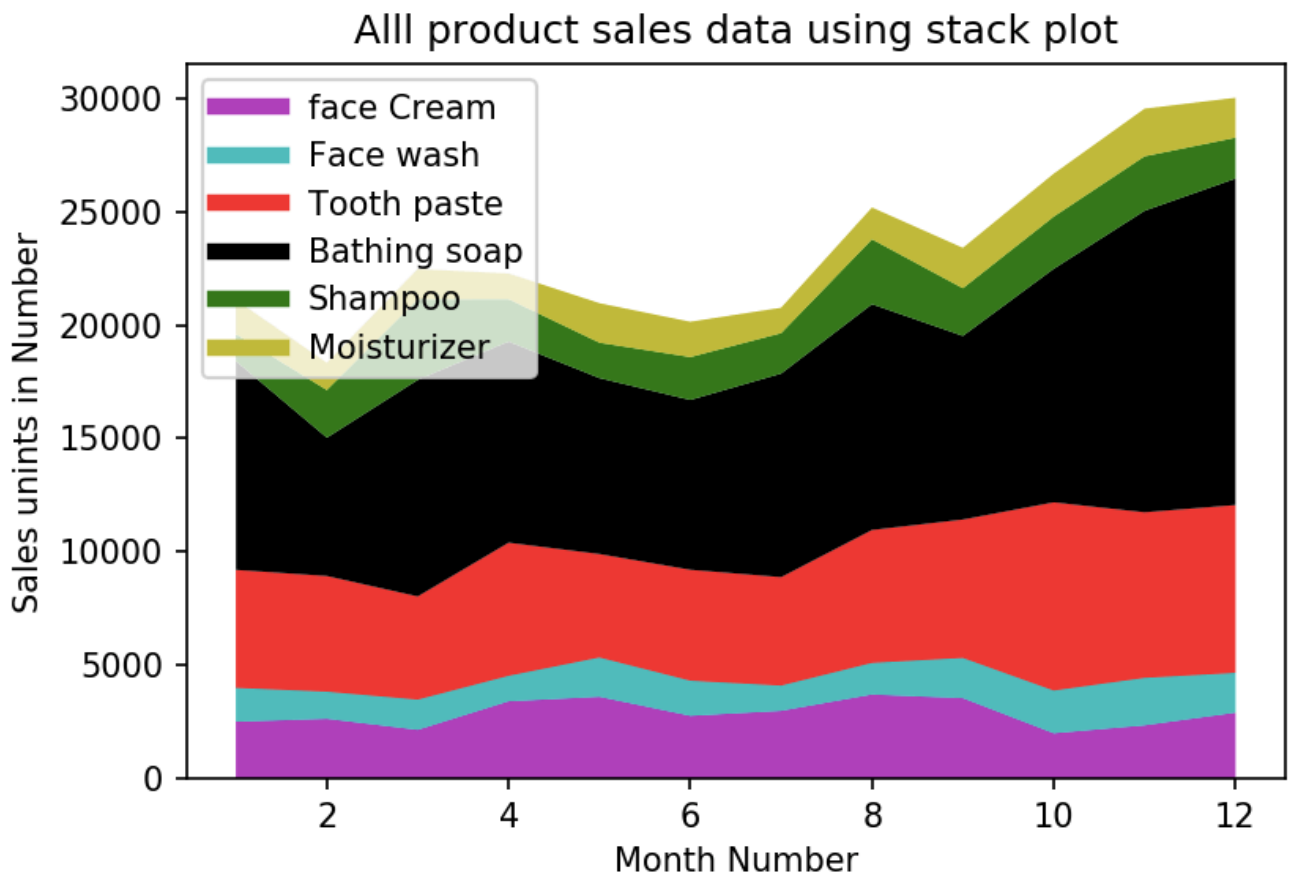


```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
monthList  = dataframe ['month_number'].tolist()
bathingsoap   = dataframe ['bathingsoap'].tolist()
faceWash= dataframe ['facewash'].tolist()
f, axarr = plt.subplots(2, sharex=True)
axarr[0].plot(monthList, bathingsoap, label = 'Bathingsoap Sales Data', color='
axarr[0].set_title('Sales data of  a Bathingsoap')
axarr[1].plot(monthList, faceWash, label = 'Face Wash Sales Data', color='r', m
axarr[1].set_title('Sales data of  a facewash')
plt.xticks(monthList)
plt.xlabel('Month Number')
plt.ylabel('Sales units in number')
plt.show()
```



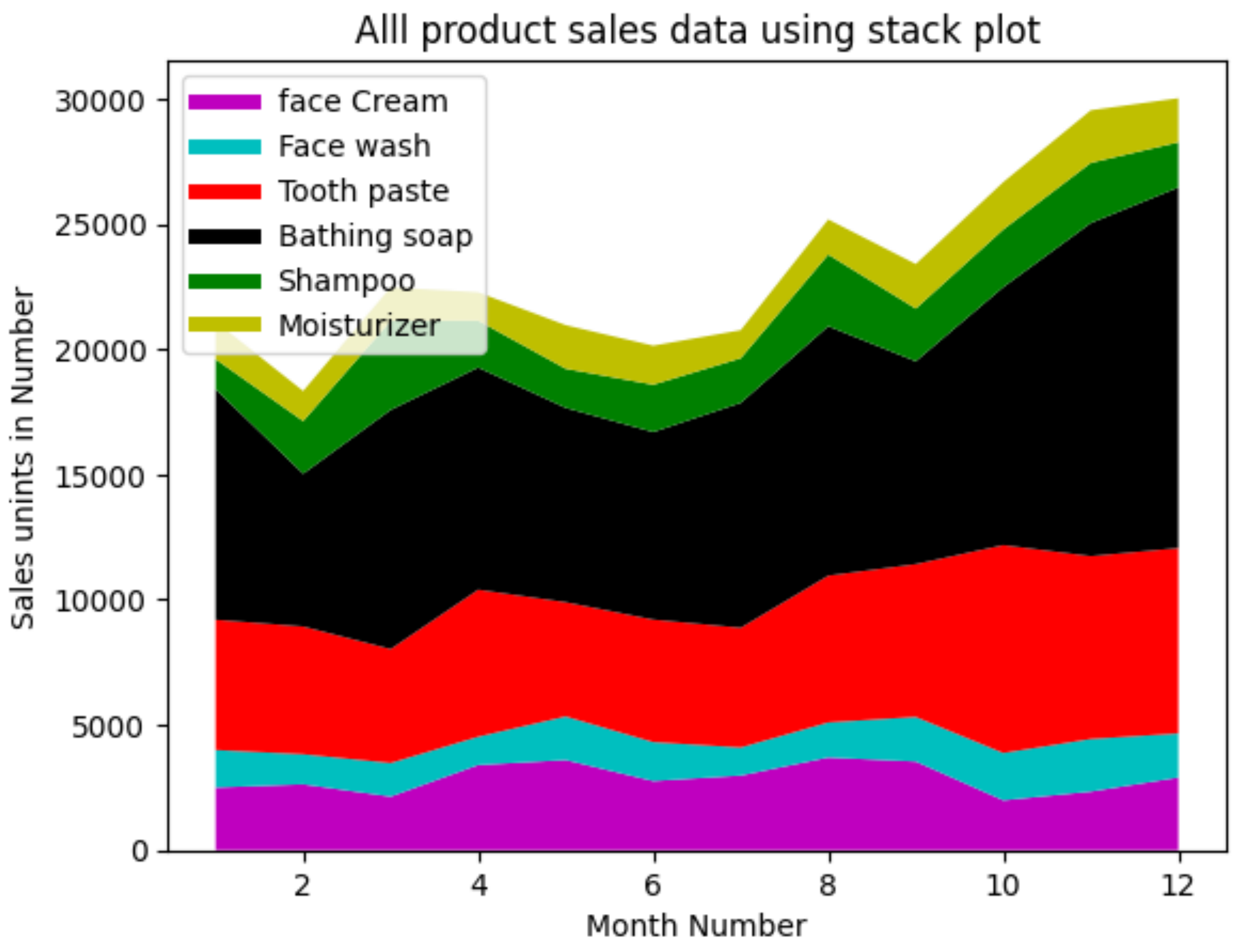
Exercise Question 10: Read all product sales data and show it using the stack plot

The Stack plot should look like this.



Double-click (or enter) to edit

```
import pandas as pd
import matplotlib.pyplot as plt
dataframe = pd.read_csv("company_sales_data.csv")
monthList  = dataframe ['month_number'].tolist()
faceCream= dataframe ['facecream'].tolist()
faceWash= dataframe ['facewash'].tolist()
toothPaste= dataframe ['toothpaste'].tolist()
bathingsoap= dataframe ['bathingsoap'].tolist()
shampoo= dataframe ['shampoo'].tolist()
moisturizer== dataframe ['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, faceCream, faceWash, toothPaste, bathingsoap, shampoo,
plt.xlabel('Month Number')
plt.ylabel('Sales unints in Number')
plt.title('Alll product sales data using stack plot')
plt.legend(loc='upper left')
plt.show()
```


Alll product sales data using stack plot
face Cream
Face wash
Tooth paste
Bathing soap
Shampoo
Moisturizer
Month Number
Sales unints in Number