Matplotlib Assignment

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

```
df = pd.read_csv('company_sales_data.csv')
df
```

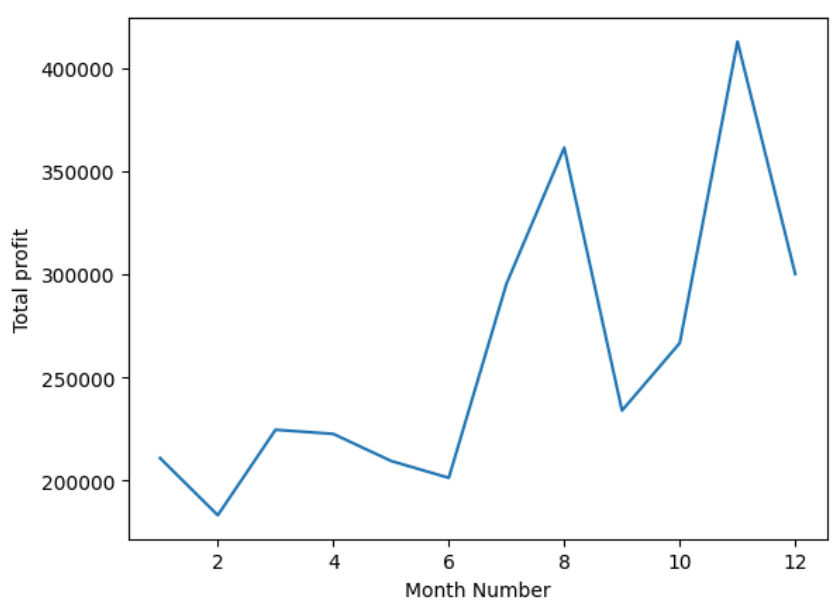
	month_number	facecream	facewash	toothpaste	bathingsoap	shampoo	moisturizer	total_units	total_profit
0	1	2500	1500	5200	9200	1200	1500	21100	211000
1	2	2630	1200	5100	6100	2100	1200	18330	183300
2	3	2140	1340	4550	9550	3550	1340	22470	224700
3	4	3400	1130	5870	8870	1870	1130	22270	222700
4	5	3600	1740	4560	7760	1560	1740	20960	209600
5	6	2760	1555	4890	7490	1890	1555	20140	201400
6	7	2980	1120	4780	8980	1780	1120	29550	295500
7	8	3700	1400	5860	9960	2860	1400	36140	361400
8	9	3540	1780	6100	8100	2100	1780	23400	234000
9	10	1990	1890	8300	10300	2300	1890	26670	266700
10	11	2340	2100	7300	13300	2400	2100	41280	412800
11	12	2900	1760	7400	14400	1800	1760	30020	300200

Next steps: Generate code with df View recommended plots

Start coding or generate with AI.

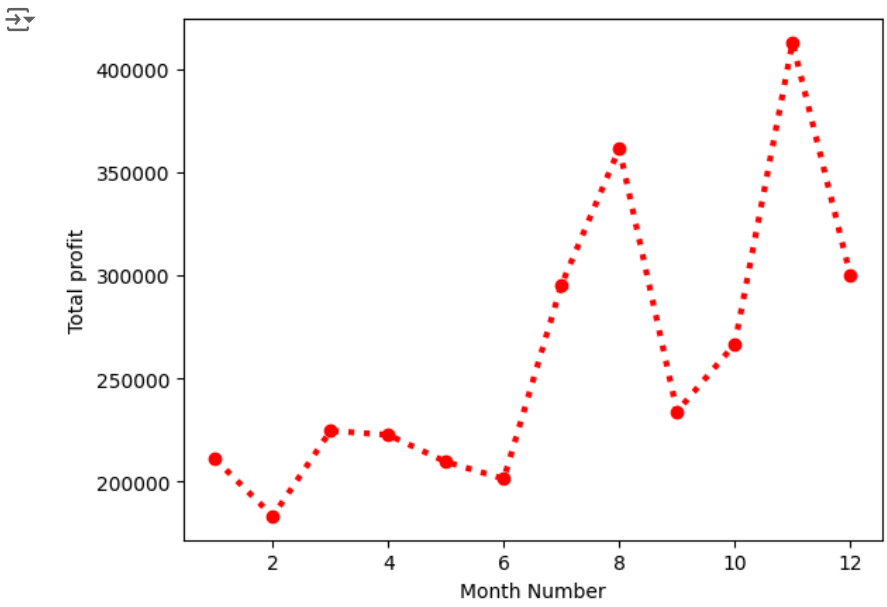
Q.1 Read Total profit of all months and show it using a line plot. Total profit data provided for each month. Generated line plot must include the following properties: X label name = Month Number Y label name = Total profit

```
x = df['month_number']
y = df['total_profit']
plt.plot(x,y)
plt.xlabel('Month Number')
plt.ylabel('Total profit')
plt.show()
```



Q.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

```
x = df['month_number']
y = df['total_profit']

plt.plot(x, y, linestyle='dotted', color='red', marker='o', markerfacecolor='red', linewidth=3)
plt.xlabel('Month Number')
plt.ylabel('Total profit')
plt.show()
```

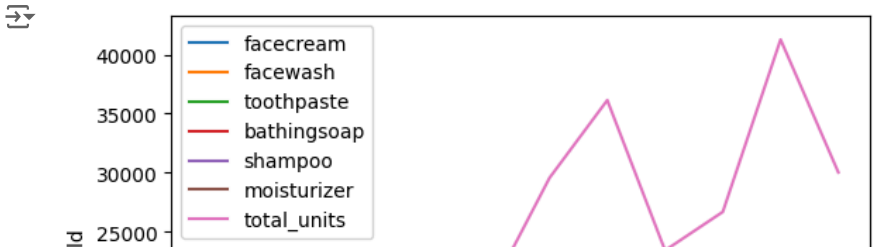


Read all product sales data and show it using a multiline plot. Display the number of units sold per month for each product using multiline plots. (i.e., Separate Plotline for each product)

Generate | 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) | Close

1 of 4 Use code with caution

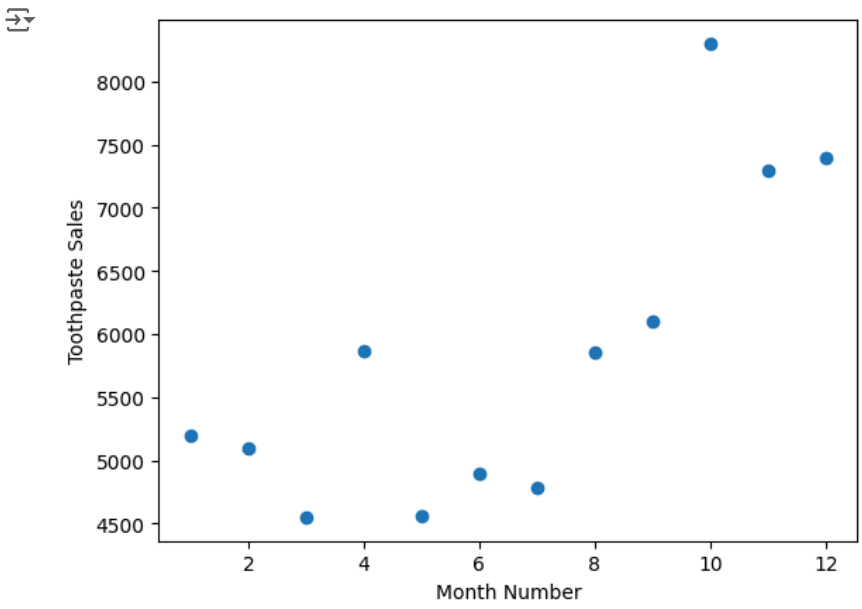
```
# prompt: Read all product sales data and show it using a multiline plot. Display the number of units sold per month

import matplotlib.pyplot as plt
products = df.drop(['month_number', 'total_profit'], axis=1)

for column in products:
  plt.plot(df['month_number'], products[column], label=column)

plt.xlabel('Month Number')
plt.ylabel('Units Sold')
plt.legend()
plt.show()
```




```
# products = df.drop(['month_number', 'total_profit'], axis=1)

# for column in products:
#     plt.plot(df['month_number'], products[column], label=column)

# plt.xlabel('Month Number')
# plt.ylabel('Units Sold')
# plt.legend()
# plt.show()
```

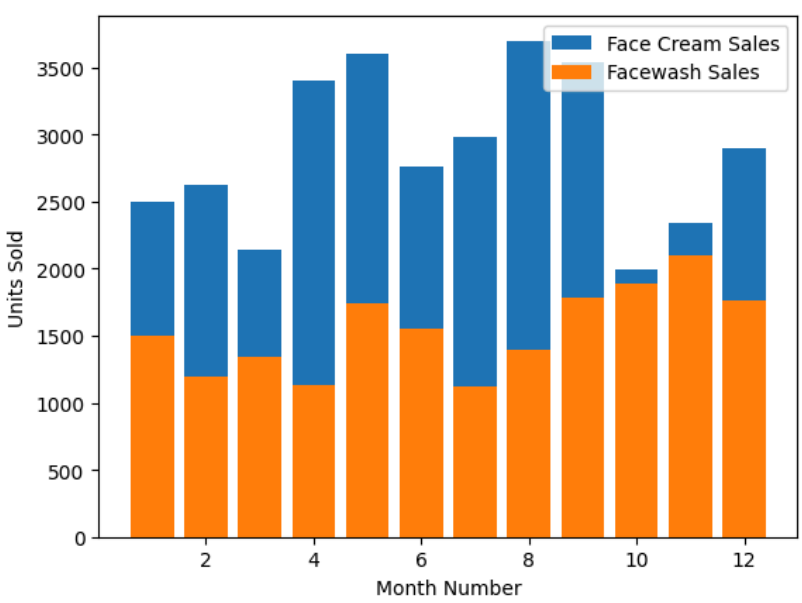
4 Read toothpaste sales data of each month and show it using a scatter plot

```
toothpaste_sales = df['toothpaste']
month_number = df['month_number']

plt.scatter(month_number, toothpaste_sales)
plt.xlabel('Month Number')
plt.ylabel('Toothpaste Sales')
plt.show()
```



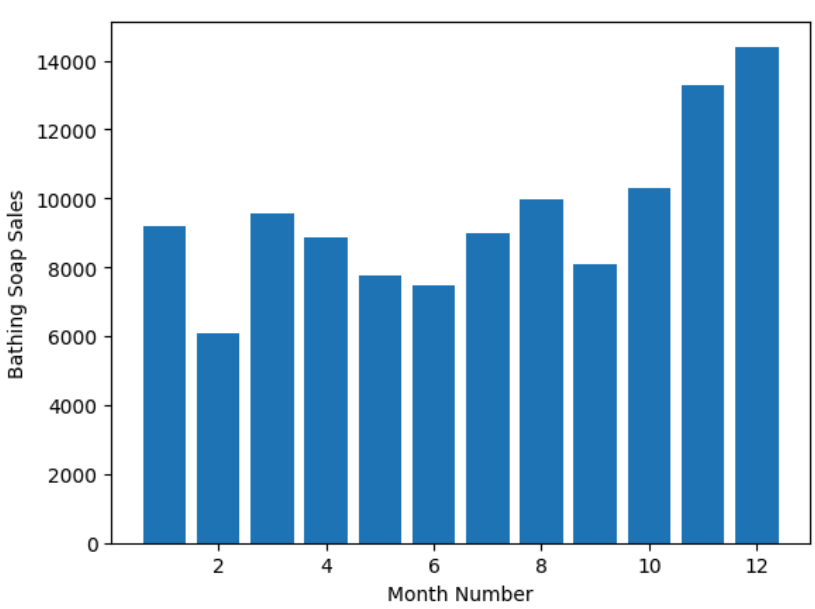
Q.5 Read face cream and facewash product sales data and show it using the bar chart. The bar chart should display the number of units sold per month for each product. Add a separate bar for each product in the same chart.

```
face_cream_sales = df['facecream']
facewash_sales = df['facewash']
month_number = df['month_number']

x = month_number
y1 = face_cream_sales
y2 = facewash_sales

plt.bar(x, y1, label='Face Cream Sales')
plt.bar(x, y2, label='Facewash Sales')
plt.xlabel('Month Number')
plt.ylabel('Units Sold')
plt.legend()
plt.show()
```



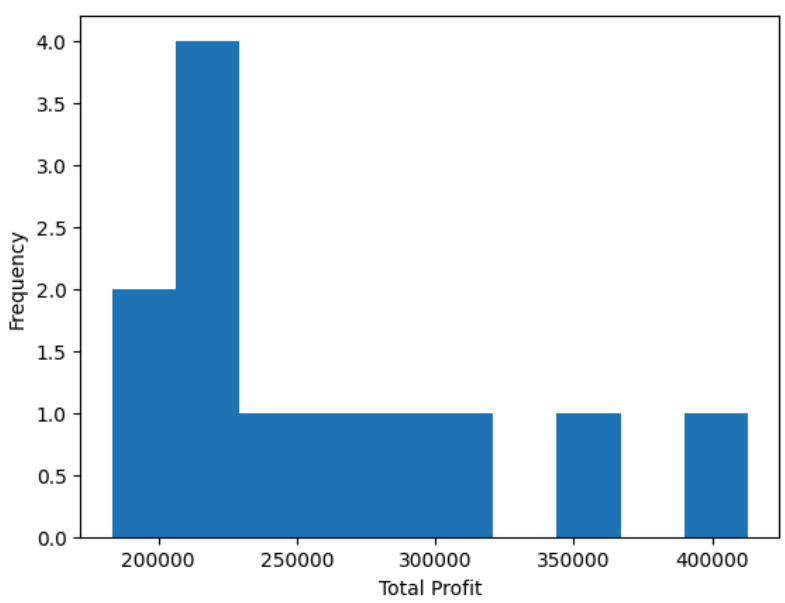
Q.6 Read sales data of bathing soap of all months and show it using a bar chart.

```
x = df['month_number']
y = df['bathingsoap']

plt.bar(x, y)
plt.xlabel('Month Number')
plt.ylabel('Bathing Soap Sales')
plt.show()
```



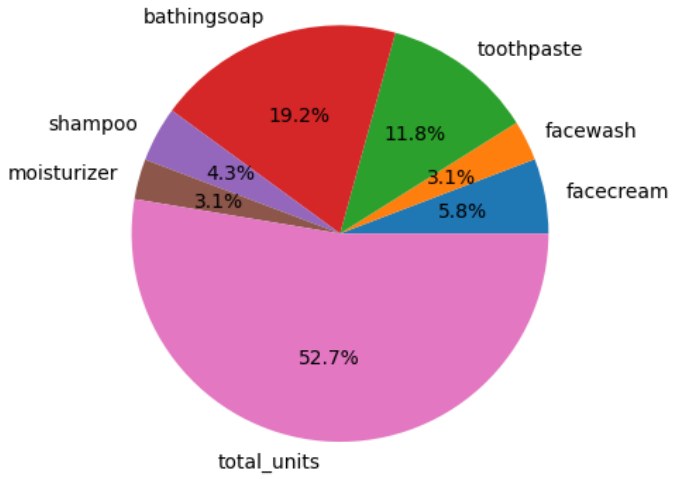
7 Read the total profit of each month and show it using the histogram to see the most common profit ranges.

```
total_profit = df['total_profit']

plt.hist(total_profit, bins=10)
plt.xlabel('Total Profit')
plt.ylabel('Frequency')
plt.show()
```



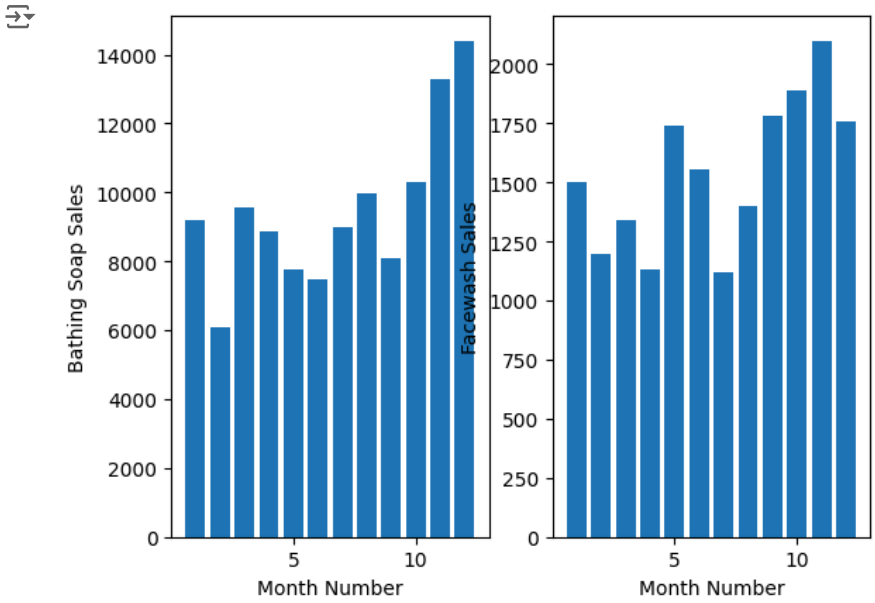
Q.8 Calculate the total sale data for last year for each product and show it using a Pie chart.

```
total_sales = df.drop(['month_number', 'total_profit'], axis=1)
plt.pie(total_sales.sum(), labels=total_sales.columns, autopct='%1.1f%%')
plt.show()
```



Q.9 Read bathing soap facewash of all months and display it using the subplot:

```
bathing_soap_sales = df['bathingsoap']
facewash_sales = df['facewash']
month_number = df['month_number']

plt.subplot(1, 2, 1)

plt.bar(month_number, bathing_soap_sales)
plt.xlabel('Month Number')
plt.ylabel('Bathing Soap Sales')

plt.subplot(1, 2, 2)

plt.bar(month_number, facewash_sales)
plt.xlabel('Month Number')
plt.ylabel('Facewash Sales')

plt.show()
```

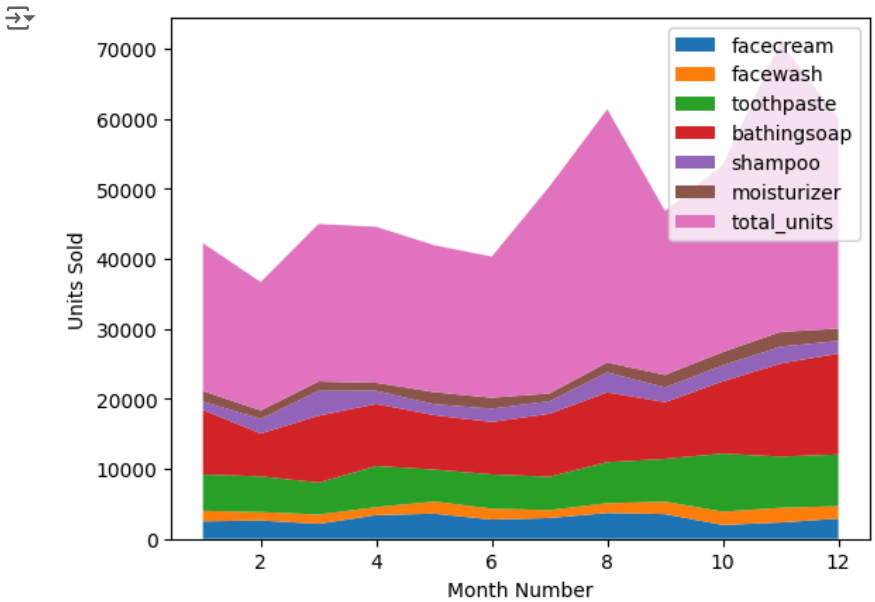


Q.10 Read all product sales data and show it using the stack plot.\

```
products.values.T
```

```
array([[ 2500,  2630,  2140,  3400,  3600,  2760,  2980,  3700,  3540,
         1990,  2340,  2900],
       [ 1500,  1200,  1340,  1130,  1740,  1555,  1120,  1400,  1780,
         1890,  2100,  1760],
       [ 5200,  5100,  4550,  5870,  4560,  4890,  4780,  5860,  6100,
         8300,  7300,  7400],
       [ 9200,  6100,  9550,  8870,  7760,  7490,  8980,  9960,  8100,
        10300, 13300, 14400],
       [ 1200,  2100,  3550,  1870,  1560,  1890,  1780,  2860,  2100,
         2300,  2400,  1800],
       [ 1500,  1200,  1340,  1130,  1740,  1555,  1120,  1400,  1780,
         1890,  2100,  1760],
       [21100, 18330, 22470, 22270, 20960, 20140, 29550, 36140, 23400,
        26670, 41280, 30020]])
```

```
products = df.drop(['month_number', 'total_profit'], axis=1)

plt.stackplot(df['month_number'], products.values.T, labels=products.columns)
plt.xlabel('Month Number')
plt.ylabel('Units Sold')
plt.legend()
plt.show()
```



Start coding or generate with AI.