1. import the dataset.

```
import pandas as pd
df = pd.read_csv('/supermarket_sales - Sheet1.csv')
print(df)
```

```
      Invoice ID Branch       City Customer type  Gender  \
0    750-67-8428      A     Yangon        Member  Female
1    226-31-3081      C  Naypyitaw        Normal  Female
2    631-41-3108      A     Yangon        Normal    Male
3    123-19-1176      A     Yangon        Member    Male
4    373-73-7910      A     Yangon        Normal    Male
..           ...    ...        ...           ...     ...
995  233-67-5758      C  Naypyitaw        Normal    Male
996  303-96-2227      B   Mandalay        Normal  Female
997  727-02-1313      A     Yangon        Member    Male
998  347-56-2442      A     Yangon        Normal    Male
999  849-09-3807      A     Yangon        Member  Female

               Product line  Unit price  Quantity   Tax 5%      Total  \
0         Health and beauty       74.69         7  26.1415   548.9715
1    Electronic accessories       15.28         5   3.8200    80.2200
2        Home and lifestyle       46.33         7  16.2155   340.5255
3         Health and beauty       58.22         8  23.2880   489.0480
4         Sports and travel       86.31         7  30.2085   634.3785
..                      ...         ...       ...      ...        ...
995       Health and beauty       40.35         1   2.0175    42.3675
996      Home and lifestyle       97.38        10  48.6900  1022.4900
997      Food and beverages       31.84         1   1.5920    33.4320
998      Home and lifestyle       65.82         1   3.2910    69.1110
999     Fashion accessories       88.34         7  30.9190   649.2990

          Date   Time      Payment    cogs  gross margin percentage  \
0     1/5/2019  13:08      Ewallet  522.83                 4.761905
1     3/8/2019  10:29         Cash   76.40                 4.761905
2     3/3/2019  13:23  Credit card  324.31                 4.761905
3    1/27/2019  20:33      Ewallet  465.76                 4.761905
4     2/8/2019  10:37      Ewallet  604.17                 4.761905
..         ...    ...          ...     ...                      ...
995  1/29/2019  13:46      Ewallet   40.35                 4.761905
996   3/2/2019  17:16      Ewallet  973.80                 4.761905
997   2/9/2019  13:22         Cash   31.84                 4.761905
998  2/22/2019  15:33         Cash   65.82                 4.761905
999  2/18/2019  13:28         Cash  618.38                 4.761905

     gross income  Rating
0         26.1415     9.1
1          3.8200     9.6
2         16.2155     7.4
3         23.2880     8.4
4         30.2085     5.3
..            ...     ...
995        2.0175     6.2
996       48.6900     4.4
997        1.5920     7.7
998        3.2910     4.1
999       30.9190     6.6
```

```
[1000 rows x 17 columns]
```

Dispay the first 5 rows.

```
import pandas as pd
print(df.head(5))
```

```
     Invoice ID Branch       City Customer type  Gender  \
0  750-67-8428      A     Yangon        Member  Female
1  226-31-3081      C  Naypyitaw        Normal  Female
2  631-41-3108      A     Yangon        Normal    Male
3  123-19-1176      A     Yangon        Member    Male
4  373-73-7910      A     Yangon        Normal    Male

              Product line  Unit price  Quantity   Tax 5%     Total       Date  \
0       Health and beauty       74.69         7  26.1415  548.9715   1/5/2019
1  Electronic accessories       15.28         5   3.8200   80.2200   3/8/2019
2      Home and lifestyle       46.33         7  16.2155  340.5255   3/3/2019
3       Health and beauty       58.22         8  23.2880  489.0480  1/27/2019
4       Sports and travel       86.31         7  30.2085  634.3785   2/8/2019

    Time      Payment    cogs  gross margin percentage  gross income  Rating
0  13:08      Ewallet  522.83                 4.761905       26.1415     9.1
1  10:29         Cash   76.40                 4.761905        3.8200     9.6
2  13:23  Credit card  324.31                 4.761905       16.2155     7.4
3  20:33      Ewallet  465.76                 4.761905       23.2880     8.4
4  10:37      Ewallet  604.17                 4.761905       30.2085     5.3
```

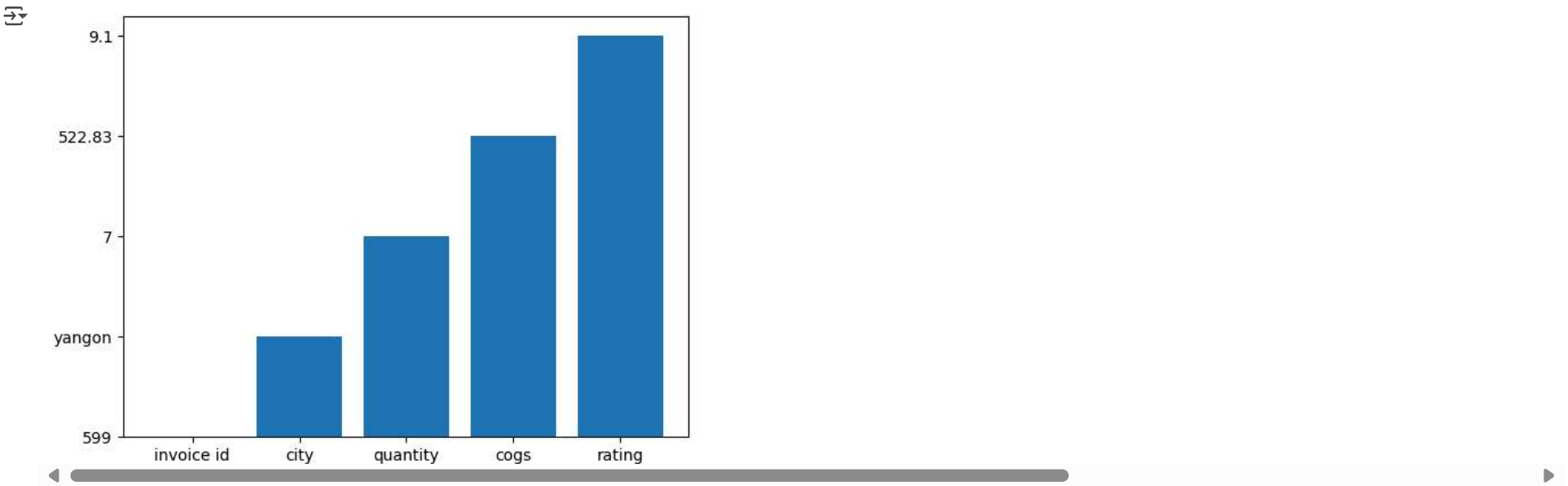
2. USE NUMPY TO CALCULATE:

```
import numpy as np
total_tax = np.sum([26.1415] + [16.2155])
print("tax:", total_tax)
import numpy as np
total_gross_income = np.sum([3.8200 ]+[23.2880])
print("total", total_gross_income)
```

```
tax: 42.357
total 27.108
```

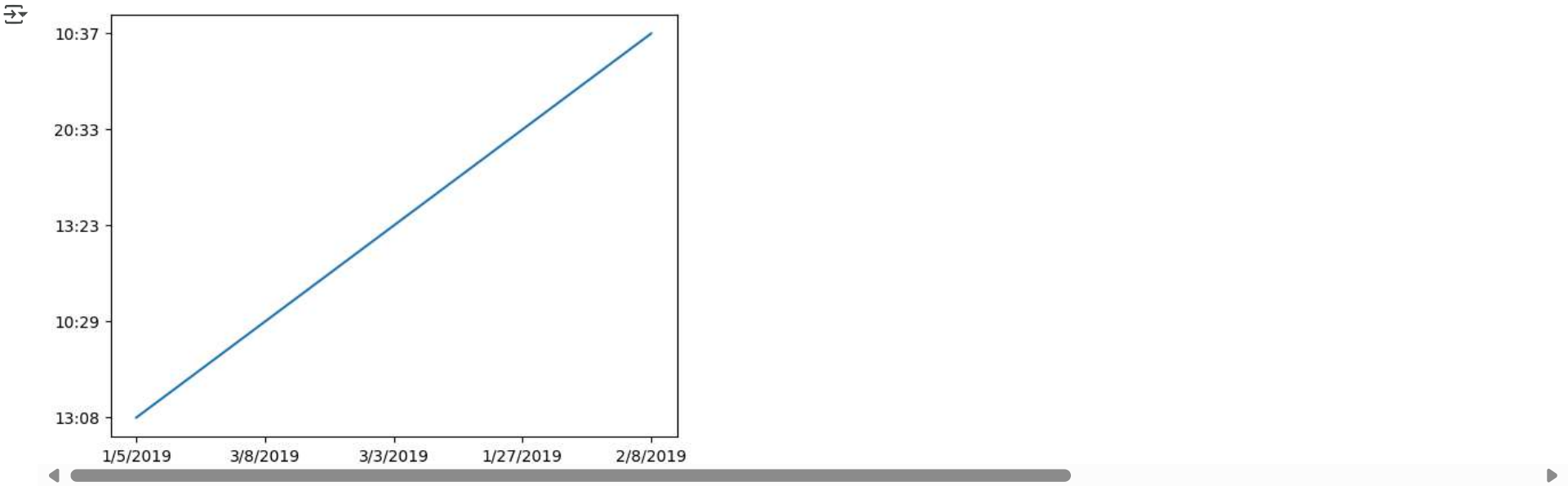
3. create a bar chart:

```
import matplotlib.pyplot as plo
import numpy as np
x = np.array(["invoice id","city", "quantity", "cogs", "rating"])
y = np.array([750-67-84, "yangon", 7, 522.83, 9.1])
plo.bar(x,y)
plo.show()
```



4. create a line chart.

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

xpoints = np.array(['1/5/2019', '3/8/2019', '3/3/2019', '1/27/2019', '2/8/2019'])
ypoints = np.array(['13:08', '10:29', '13:23', '20:33', '10:37'])

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

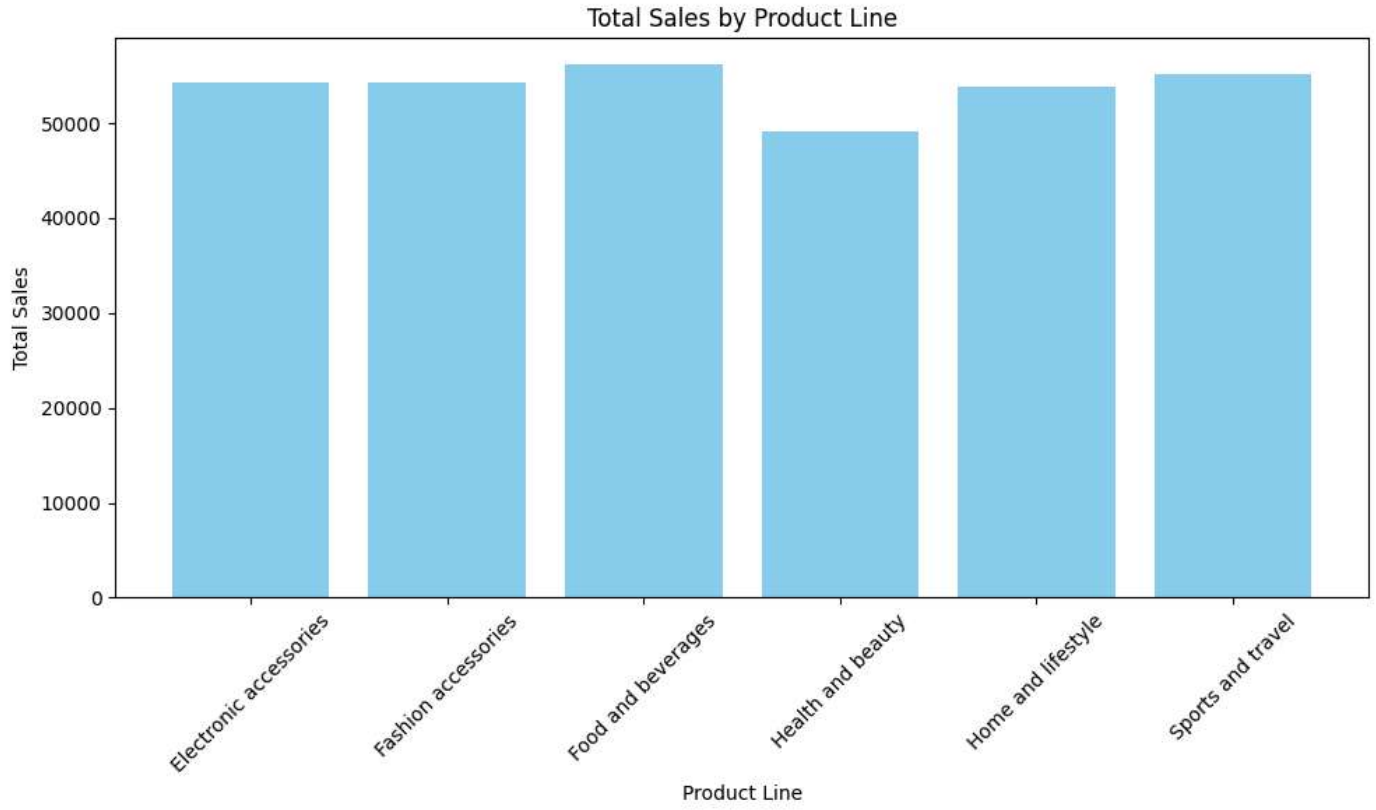


5. identify the best-selling-product and visualize its performance

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

# Load the CSV file
df = pd.read_csv('/content/supermarket_sales - Sheet1 (1).csv')

# Optional: View the first few rows to understand the structure
print(df.head())

# Check for missing product names
df = df.dropna(subset=['Product line'])

# Group sales by product line (summing total sales per category)
sales_by_product = df.groupby('Product line')['Total'].sum().reset_index()

# Identify best-selling product
best_selling = sales_by_product.loc[sales_by_product['Total'].idxmax()]

print(f"Best-selling product: {best_selling['Product line']} with {best_selling['Total']} total sales.")

# Visualize sales performance
plt.figure(figsize=(10, 6))
plt.bar(sales_by_product['Product line'], sales_by_product['Total'], color='skyblue')
plt.xlabel('Product Line')
plt.ylabel('Total Sales')
plt.title('Total Sales by Product Line')
plt.xticks(rotation=45)
plt.tight_layout()
```

```
plt.show()
```

```
    Invoice ID Branch       City Customer type  Gender  \
0  750-67-8428      A     Yangon        Member  Female
1  226-31-3081      C  Naypyitaw        Normal  Female
2  631-41-3108      A     Yangon        Normal    Male
3  123-19-1176      A     Yangon        Member    Male
4  373-73-7910      A     Yangon        Normal    Male

             Product line  Unit price  Quantity   Tax 5%     Total       Date  \
0       Health and beauty       74.69         7  26.1415  548.9715   1/5/2019
1  Electronic accessories       15.28         5   3.8200   80.2200   3/8/2019
2      Home and lifestyle       46.33         7  16.2155  340.5255   3/3/2019
3       Health and beauty       58.22         8  23.2880  489.0480  1/27/2019
4       Sports and travel       86.31         7  30.2085  634.3785   2/8/2019

    Time      Payment    cogs  gross margin percentage  gross income  Rating
0  13:08      Ewallet  522.83                 4.761905       26.1415     9.1
1  10:29         Cash   76.40                 4.761905        3.8200     9.6
2  13:23  Credit card  324.31                 4.761905       16.2155     7.4
3  20:33      Ewallet  465.76                 4.761905       23.2880     8.4
4  10:37      Ewallet  604.17                 4.761905       30.2085     5.3
Best-selling product: Food and beverages with 56144.844 total sales.
```



6. provide a clear written analysis of insights and trends.

```
import pandas as pd
import matplotlib.pyplot as plt
data =pd.read_csv('//content/supermarket_sales - Sheet1 (1).csv')
data['Total_sales'] = data['Unit price'] * data['Quantity']
print(data.columns)
best_product = data.groupby('Product line') ['Total_sales'].sum().idxmax()
product_data = data[data['Product line'] == best_product]
product_data['Data'] = pd.to_datetime(product_data['Date'])
daily_sales = product_data.groupby(product_data['Date'])['Total_sales'].sum()
daily_sales.plot(title=f"Daily sales of {best_product}", figsize=(10,2))
plt.xlabel("Date")
plt.ylabel("Total sales")
plt.grid(True)
plt.show()
```

```
Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
       'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date',
       'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
       'Rating', 'Total_sales'],
      dtype='object')
<ipython-input-23-122eef3b9fa1>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  product_data['Data'] = pd.to_datetime(product_data['Date'])
```