```
import faker
import pandas as pd
import random
import numpy as np
from datetime import datetime, timedelta

# Initialize faker
fake = faker.Faker()

# Define categories
categories = ['Electronics', 'Clothing', 'Books', 'Beauty', 'Home',
'Sports']

# Define function to generate synthetic e-commerce data
def generate_shopping_data(num_rows):
    data = {
        'order_id': [fake.random_int(min=1000, max=9999) for _ in
range(num_rows)],
        'product_name': [fake.word() for _ in range(num_rows)],
        'price': [fake.random_number(digits=2) for _ in
range(num_rows)],
        'quantity': [fake.random_int(min=1, max=10) for _ in
range(num_rows)],
        'customer_name': [fake.name() for _ in range(num_rows)],
        'address': [fake.address() for _ in range(num_rows)],
        'email': [fake.email() for _ in range(num_rows)],
        'gender': [fake.random_element(elements=('Male', 'Female'))
for _ in range(num_rows)],
        'date': [fake.date_between(start_date='-1y', end_date='today')
for _ in range(num_rows)],
        'location': [fake.country() for _ in range(num_rows)],
        'category': [random.choice(categories) for _ in
range(num_rows)]
    }

    # Introduce outliers
    outlier_indices = random.sample(range(num_rows), min(int(num_rows
* 0.05), 10))
    for idx in outlier_indices:
        data['price'][idx] *= random.uniform(5, 10)

    # Introduce duplicates
    duplicate_indices = random.sample(range(num_rows),
min(int(num_rows * 0.1), 20))
    for idx in duplicate_indices:
        data['order_id'][idx] = data['order_id']
[random.choice(range(num_rows))]

    # Introduce null values
    null_indices = random.sample(range(num_rows), min(int(num_rows *
```

```
0.1), 20))
    for idx in null_indices:
        for key in data.keys():
            data[key][idx] = np.nan

    # Calculate sale amount and profit
    data['sale_amount'] = [data['price'][i] * data['quantity'][i] for
i in range(num_rows)]
    data['profit'] = [data['sale_amount'][i] * random.uniform(0.05,
0.2) for i in range(num_rows)]

    return pd.DataFrame(data)

# Generate synthetic data
num_rows = 1000
Shopping_data = generate_shopping_data(num_rows)

# Display synthetic data

Shopping_data
```

```
     order_id product_name       price  quantity      customer_name  \
0      7335.0         that   68.000000      10.0       Amanda Adams
1      1893.0       budget   33.000000       4.0  Christine Daniels
2      5388.0        chair   97.000000       7.0        Daniel Dean
3      2624.0         drop   27.000000       3.0         Kevin King
4      1571.0        court   56.000000      10.0     Jamie Campbell
..        ...          ...         ...       ...                ...
995    8201.0        local   75.000000       6.0    Stephanie Heath
996    2197.0       myself   66.000000       3.0      James Vasquez
997    9063.0      machine   54.000000       7.0        Erica Moore
998    4366.0      several  770.203536       7.0       Kathryn Reid
999    7956.0      suggest   61.000000       9.0        Mary Obrien

                                              address  \
0    89848 Christine Station Suite 357\nValeriebury...
1      0553 Wendy Ways Suite 420\nWest Scott, IL 76790
2           060 Horton Row\nNew Melissaport, NE 61802
3    380 Potter Forges Suite 210\nWilsontown, NE 48889
4    520 William Highway Suite 684\nWest Robinburgh...
..                                                 ...
995  049 Bauer Avenue Apt. 312\nNew Carlaborough, G...
996      85451 Michael Locks\nSouth Mitchell, NJ 93434
997   5266 Jose Ville Suite 673\nLake Dennis, MN 92154
998  64107 John Trace Apt. 047\nEast Timothy, AZ 73201
999                  PSC 1807, Box 3944\nAPO AP 67318

                           email  gender        date
location  \
0        hawkinslinda@example.com    Male  2024-02-26            Saudi
```

```
Arabia
1          dawnchavez@example.com    Male  2024-04-30
Angola
2       erinschroeder@example.org  Female  2024-01-09
Belize
3               joy83@example.com  Female  2023-08-18
Anguilla
4         mariafoster@example.org    Male  2024-03-16
Kuwait
..                            ...     ...         ...
...
995  christinebarrera@example.net  Female  2023-11-26
Ukraine
996     mcdonaldjames@example.org  Female  2024-04-13
Burkina Faso
997       reyesjoshua@example.com    Male  2024-01-20  British Virgin
Islands
998       morenodavid@example.net  Female  2024-02-12
Namibia
999       donnabrewer@example.com    Male  2023-05-14
Ghana

        category  sale_amount      profit
0    Electronics   680.000000  117.487249
1         Beauty   132.000000   19.945602
2          Books   679.000000   34.981492
3         Beauty    81.000000   10.032915
4    Electronics   560.000000   38.366661
..           ...          ...         ...
995  Electronics   450.000000   56.436091
996         Home   198.000000   23.383709
997     Clothing   378.000000   53.919608
998  Electronics  5391.424752  302.026660
999       Sports   549.000000   97.478372

[1000 rows x 13 columns]
```

```
Location=r'C:\\Users\\DELL\\Documents\\.ipynb_checkpoints\\
Note_new.csv'
Shopping_data.to_csv(Location,index=False)


Shopping_data = Shopping_data.dropna(how='all')
Shopping_data
```

```
     order_id product_name  price  quantity        customer_name  \
0      2330.0         face   43.0       3.0       Patricia Moore
1      2954.0      prepare   51.0       9.0          Tony Hunter
2      9308.0         send   57.0       9.0      Jeffery Herrera
3      6152.0         show   15.0       6.0  Mrs. Andrea Owen MD
```

```
4      8749.0        stage    7.0       3.0          Erik Rivera
..        ...          ...    ...       ...                 ...
995    6984.0     campaign    3.0       3.0        Brianna Mejia
996    1574.0        thing   10.0      10.0       Nicholas Chase
997    1804.0        human   83.0       8.0     Michael Peterson
998    5953.0          win   53.0      10.0       Terrence Brown
999    4631.0     behavior    4.0       6.0          Matthew Lee

                                                  address  \
0    5584 Davis Inlet Suite 161\nJennifershire, WA ...
1    81478 Blankenship Roads Suite 388\nSouth Markv...
2                     Unit 7948 Box 1497\nDPO AA 45527
3         212 Jennifer Station\nTimothyburgh, NY 13319
4    7922 Tracy Tunnel Suite 312\nSanchezbury, AL 9...
..                                                 ...
995          06468 Carol Forges\nAndersonton, MI 30655
996                      USS Christensen\nFPO AE 34459
997    02053 Ashley Way\nSouth Stephaniebury, WA 67533
998  029 Pearson Grove Apt. 461\nWest Matthew, MA 6...
999        604 Martin Lakes\nSouth Lisamouth, NE 92122

                              email  gender        date  \
0             angela26@example.org    Male  2024-01-18
1               zoneal@example.net  Female  2023-08-20
2             wjohnson@example.com  Female  2023-05-08
3    dominiquephillips@example.com  Female  2023-10-30
4             bgalloway@example.com    Male  2023-07-16
..                              ...     ...         ...
995           thomas41@example.net  Female  2023-05-21
996           jilldavid@example.com    Male  2024-02-22
997       bakerfrancis@example.com  Female  2023-09-06
998          garciaamy@example.org  Female  2023-12-07
999            zknight@example.org  Female  2023-05-31

                              location     category  sale_amount
profit
0                               Brazil         Home        129.0
19.520525
1                               Uganda         Home        459.0
64.628182
2                                Egypt         Home        513.0
40.707420
3                             Honduras  Electronics         90.0
8.500197
4                 Syrian Arab Republic       Beauty         21.0
1.078037
..                                 ...          ...          ...
...
995     Holy See (Vatican City State)         Home          9.0
0.699745
```

```
996                         Cook Islands        Books         100.0
19.050987
997                              Guernsey        Books         664.0
53.450919
998  Lao People's Democratic Republic       Beauty         530.0
49.936220
999                               Greece          Home          24.0
4.460721

[980 rows x 13 columns]
```

```
Shopping_data.drop_duplicates(subset=['order_id'],inplace=True)
```

```
Shopping_data.isnull().sum()
```

```
order_id          1
product_name      1
price             1
quantity          1
customer_name     1
address           1
email             1
gender            1
date              1
location          1
category          1
sale_amount       1
profit            1
dtype: int64
```

```
outliers=[]
import numpy as np

data = np.array(Shopping_data['price']).values  # Replace [...] with
your dataset

# Calculate mean and standard deviation
mean = np.mean(data)
std_dev = np.std(data)

# Calculate Z-scores
z_scores = (data - mean) / std_dev

# Identify outliers
outliers = np.where(np.abs(z_scores) > 3)

print("Indices of outliers:", outliers[0])
print("Values of outliers:", data[outliers])
```

```
-------------------------------------------------------------------------
-----
```

```
AttributeError                            Traceback (most recent call
last)
Cell In[11], line 4
      1 outliers=[]
      2 import numpy as np
----> 4 data = np.array(Shopping_data['price']).values  # Replace
[...] with your dataset
      6 # Calculate mean and standard deviation
      7 mean = np.mean(data)

AttributeError: 'numpy.ndarray' object has no attribute 'values'
```

```
Location=r'C:\\Users\\DELL\\Documents\\.ipynb_checkpoints\\
Updated_data.csv'
Shopping_data.to_csv(Location,index=False)
Shopping_data
```

```
     order_id product_name       price  quantity      customer_name  \
0      7335.0         that   68.000000      10.0       Amanda Adams
1      1893.0       budget   33.000000       4.0  Christine Daniels
2      5388.0        chair   97.000000       7.0        Daniel Dean
3      2624.0         drop   27.000000       3.0         Kevin King
4      1571.0        court   56.000000      10.0     Jamie Campbell
..        ...          ...         ...       ...                ...
994    1455.0        cover   68.000000       4.0   Steven Dominguez
995    8201.0        local   75.000000       6.0    Stephanie Heath
996    2197.0       myself   66.000000       3.0      James Vasquez
997    9063.0      machine   54.000000       7.0        Erica Moore
998    4366.0      several  770.203536       7.0       Kathryn Reid

                                                address  \
0    89848 Christine Station Suite 357\nValeriebury...
1      0553 Wendy Ways Suite 420\nWest Scott, IL 76790
2            060 Horton Row\nNew Melissaport, NE 61802
3    380 Potter Forges Suite 210\nWilsontown, NE 48889
4    520 William Highway Suite 684\nWest Robinburgh...
..                                                 ...
994        905 Christopher Mills\nJasonmouth, MS 52818
995  049 Bauer Avenue Apt. 312\nNew Carlaborough, G...
996      85451 Michael Locks\nSouth Mitchell, NJ 93434
997   5266 Jose Ville Suite 673\nLake Dennis, MN 92154
998  64107 John Trace Apt. 047\nEast Timothy, AZ 73201

                            email  gender        date
location  \
0        hawkinslinda@example.com    Male  2024-02-26            Saudi
Arabia
1          dawnchavez@example.com    Male  2024-04-30
Angola
2       erinschroeder@example.org  Female  2024-01-09
```

```
Belize
3               joy83@example.com  Female  2023-08-18
Anguilla
4         mariafoster@example.org    Male  2024-03-16
Kuwait
..                            ...     ...         ...
...
994           carol86@example.com  Female  2024-02-23             Saudi
Arabia
995  christinebarrera@example.net  Female  2023-11-26
Ukraine
996     mcdonaldjames@example.org  Female  2024-04-13
Burkina Faso
997       reyesjoshua@example.com    Male  2024-01-20  British Virgin
Islands
998       morenodavid@example.net  Female  2024-02-12
Namibia

        category  sale_amount      profit
0    Electronics   680.000000  117.487249
1         Beauty   132.000000   19.945602
2          Books   679.000000   34.981492
3         Beauty    81.000000   10.032915
4    Electronics   560.000000   38.366661
..           ...          ...         ...
994       Sports   272.000000   46.340875
995  Electronics   450.000000   56.436091
996         Home   198.000000   23.383709
997     Clothing   378.000000   53.919608
998  Electronics  5391.424752  302.026660

[914 rows x 13 columns]

#Capitalized the All Heading of first coloum
Shopping_data.columns = Shopping_data.columns.str.capitalize()
Shopping_data.columns

Index(['Order_id', 'Product_name', 'Price', 'Quantity',
'Customer_name',
       'Address', 'Email', 'Gender', 'Date', 'Location', 'Category',
       'Sale_amount', 'Profit'],
      dtype='object')

#Shopping_data.columns
Shopping_data['Date']

0      2024-02-26
1      2024-04-30
2      2024-01-09
3      2023-08-18
```

```
4      2024-03-16
          ...
994    2024-02-23
995    2023-11-26
996    2024-04-13
997    2024-01-20
998    2024-02-12
Name: Date, Length: 914, dtype: object
```

```
pd.to_datetime(Shopping_data['Date'])
```

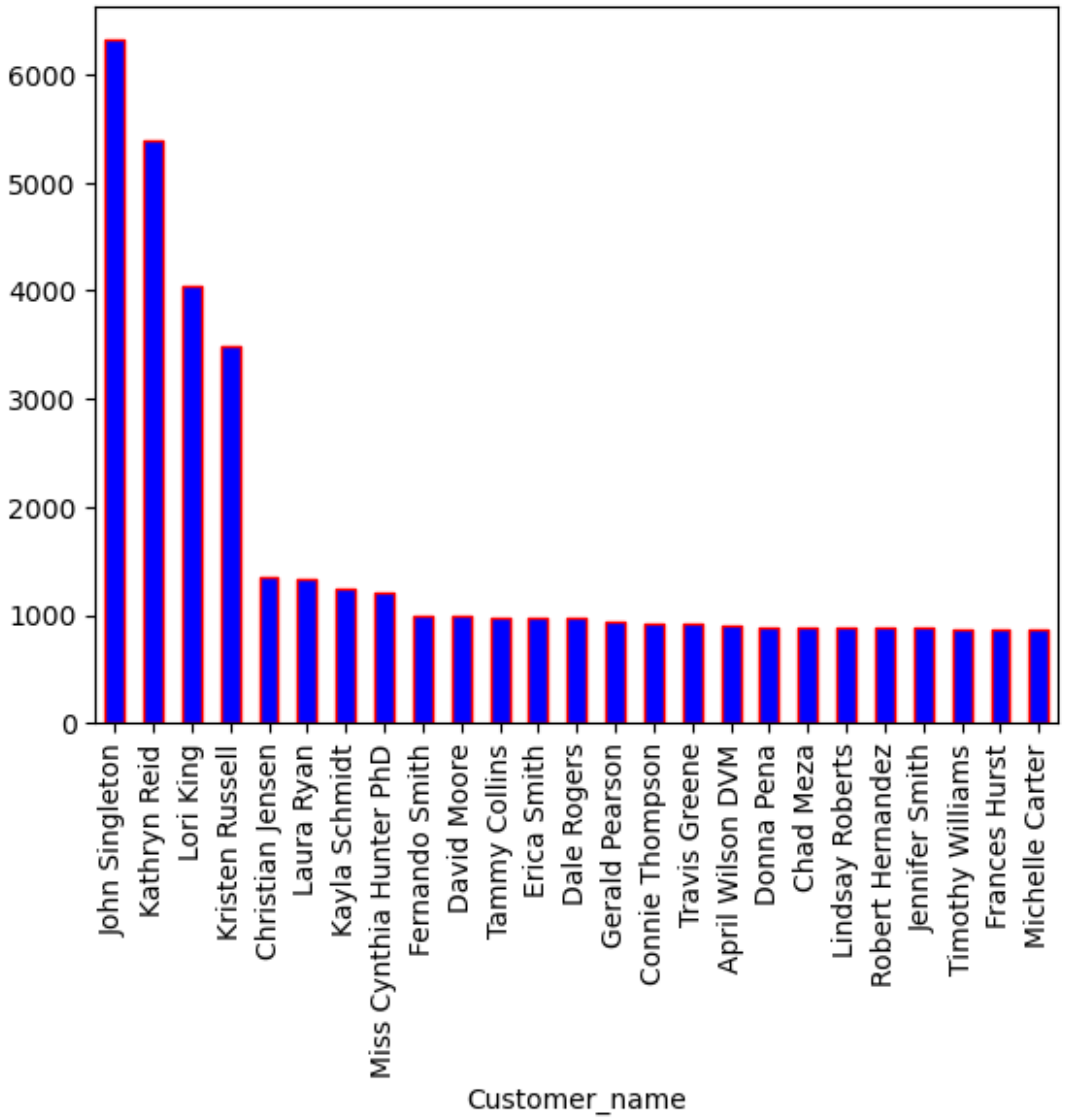
```
0     2024-02-26
1     2024-04-30
2     2024-01-09
3     2023-08-18
4     2024-03-16
          ...
994   2024-02-23
995   2023-11-26
996   2024-04-13
997   2024-01-20
998   2024-02-12
Name: Date, Length: 914, dtype: datetime64[ns]
```

```
Shopping_data['Price']
```

```
0       68.000000
1       33.000000
2       97.000000
3       27.000000
4       56.000000
           ...
994     68.000000
995     75.000000
996     66.000000
997     54.000000
998    770.203536
Name: Price, Length: 914, dtype: float64
```

```
#top 15 customer who purchase most
import matplotlib.pyplot as plt
Shopping_data.groupby('Customer_name')
['Sale_amount'].sum().sort_values(ascending =
False).head(25).plot(kind='bar',color='blue',edgecolor='red')
```

```
<Axes: xlabel='Customer_name'>
```



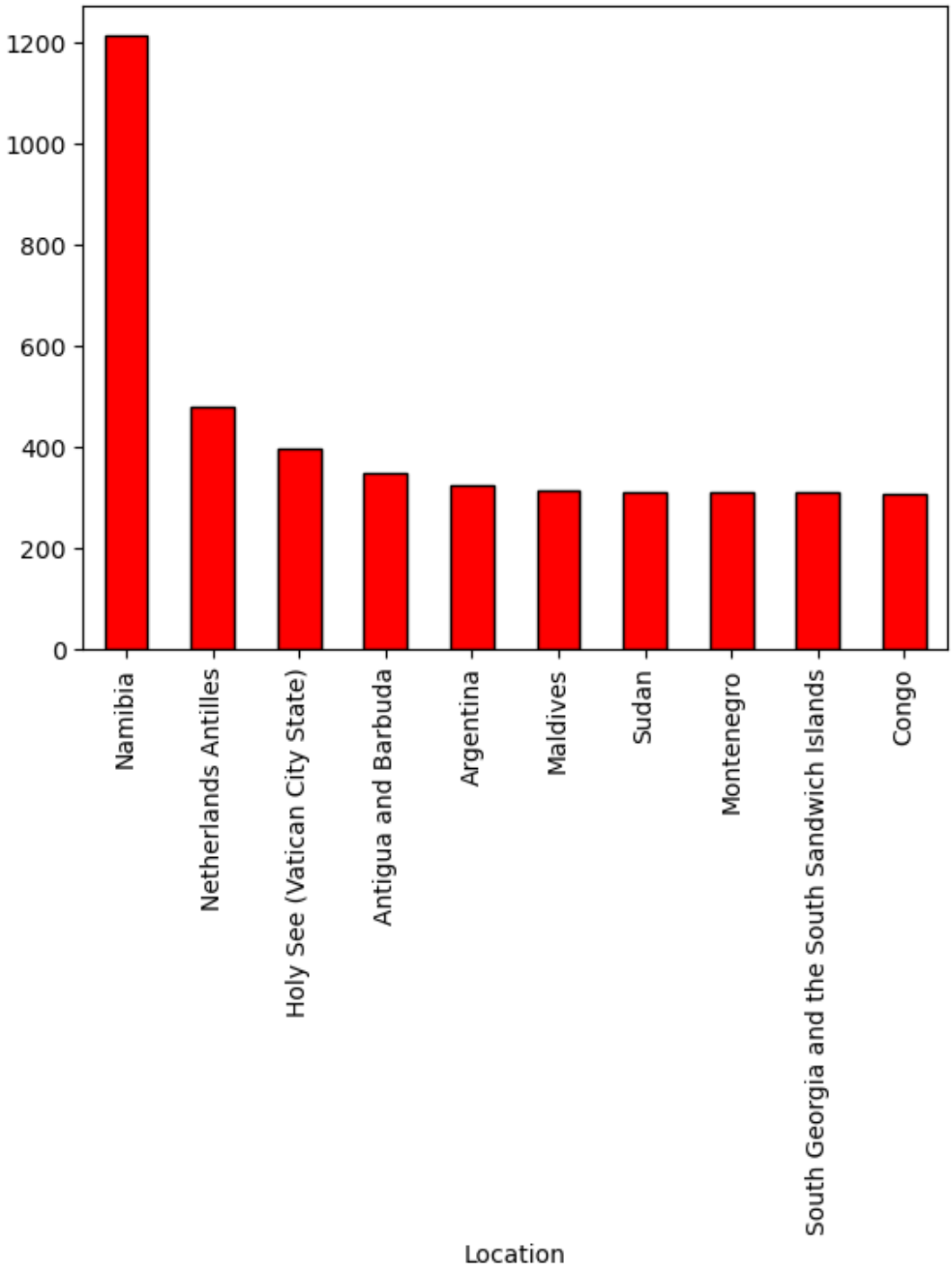

```
Shopping_data.isnull().sum()
```

```
Order_id          1
Product_name      1
Price             1
Quantity          1
Customer_name     1
Address           1
Email             1
Gender            1
Date              1
Location          1
Category          1
Sale_amount       1
```

```
Profit            1
dtype: int64
```

```
Shopping_data.groupby('Location')
['Profit'].sum().sort_values( ascending
=False).head(10).plot(kind='bar',color='red',edgecolor='black')
```

```
<Axes: xlabel='Location'>
```



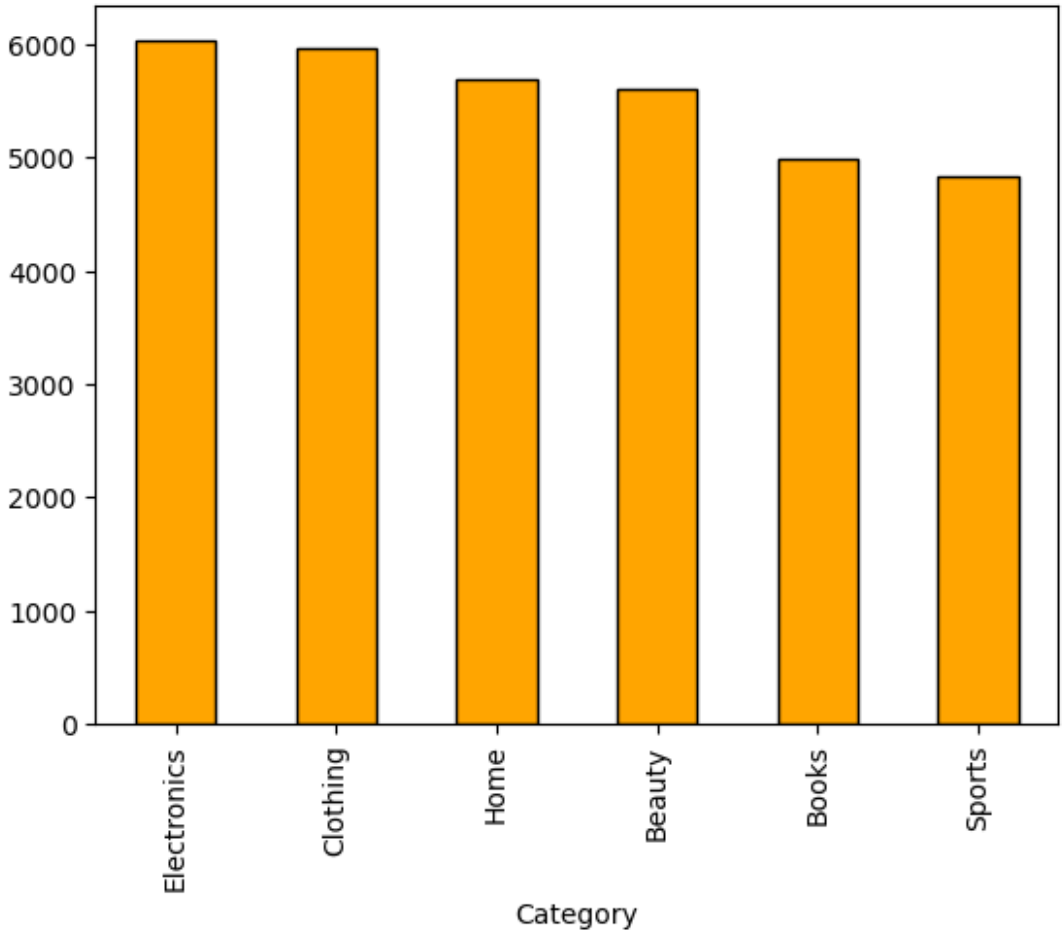

```
#What is the total sales amount and profit for each product category?
Shopping_data.columns
```

```
Index(['Order_id', 'Product_name', 'Price', 'Quantity',
'Customer_name',
       'Address', 'Email', 'Gender', 'Date', 'Location', 'Category',
```

```
       'Sale_amount', 'Profit'],
      dtype='object')
```

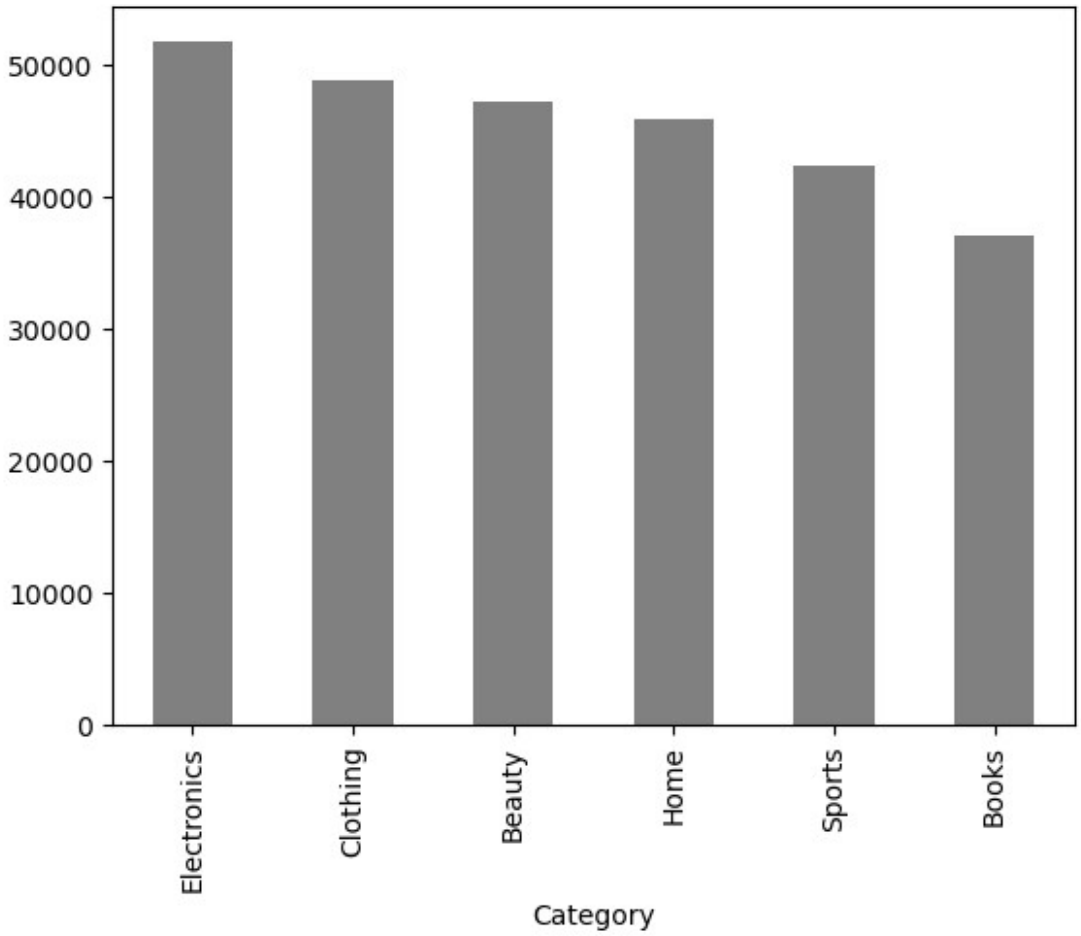
```
Shopping_data.groupby('Category')[
'Profit'].sum().sort_values( ascending=False).plot(kind='bar',color='o
range',edgecolor='black')
```

```
<Axes: xlabel='Category'>
```

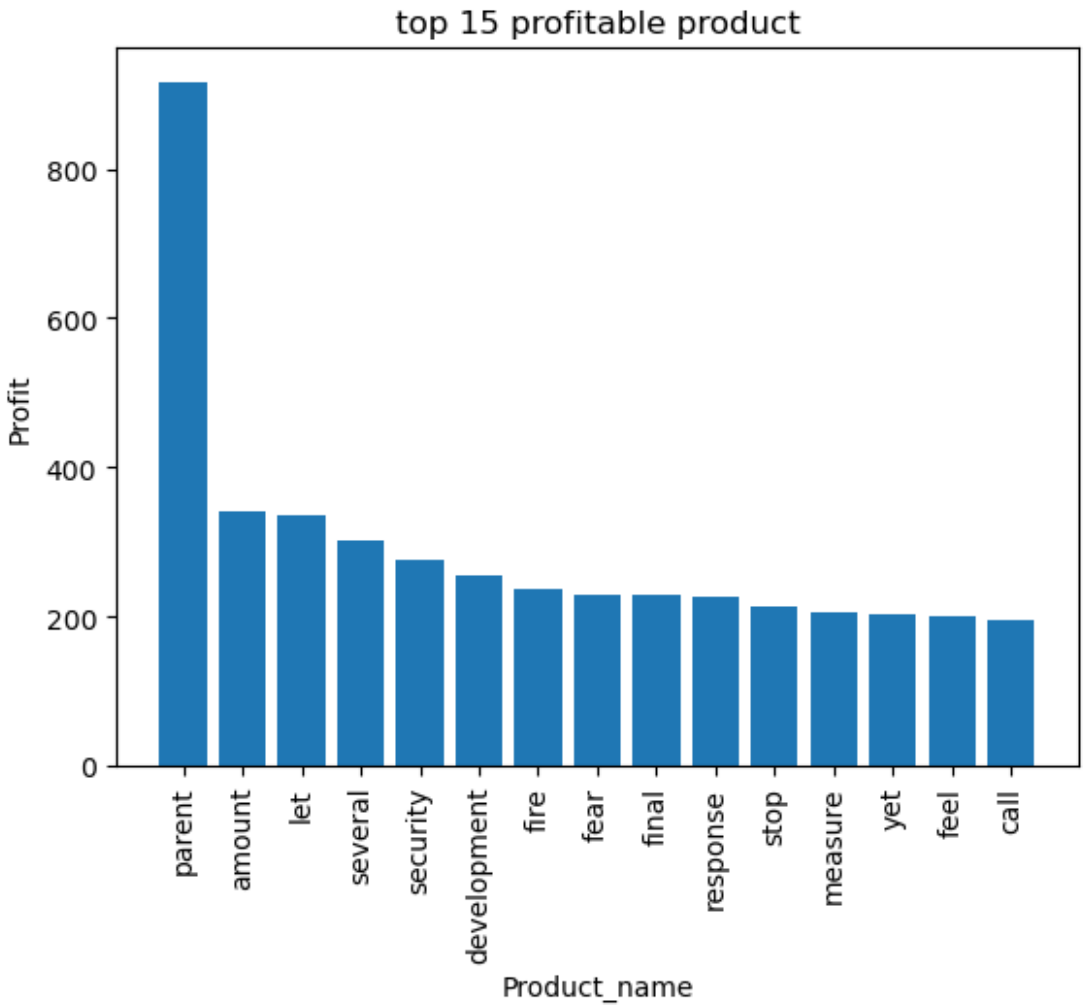



```
Shopping_data.groupby('Category')
['Sale_amount'].sum().sort_values(ascending=False).plot(kind='bar',
color='Gray')
```

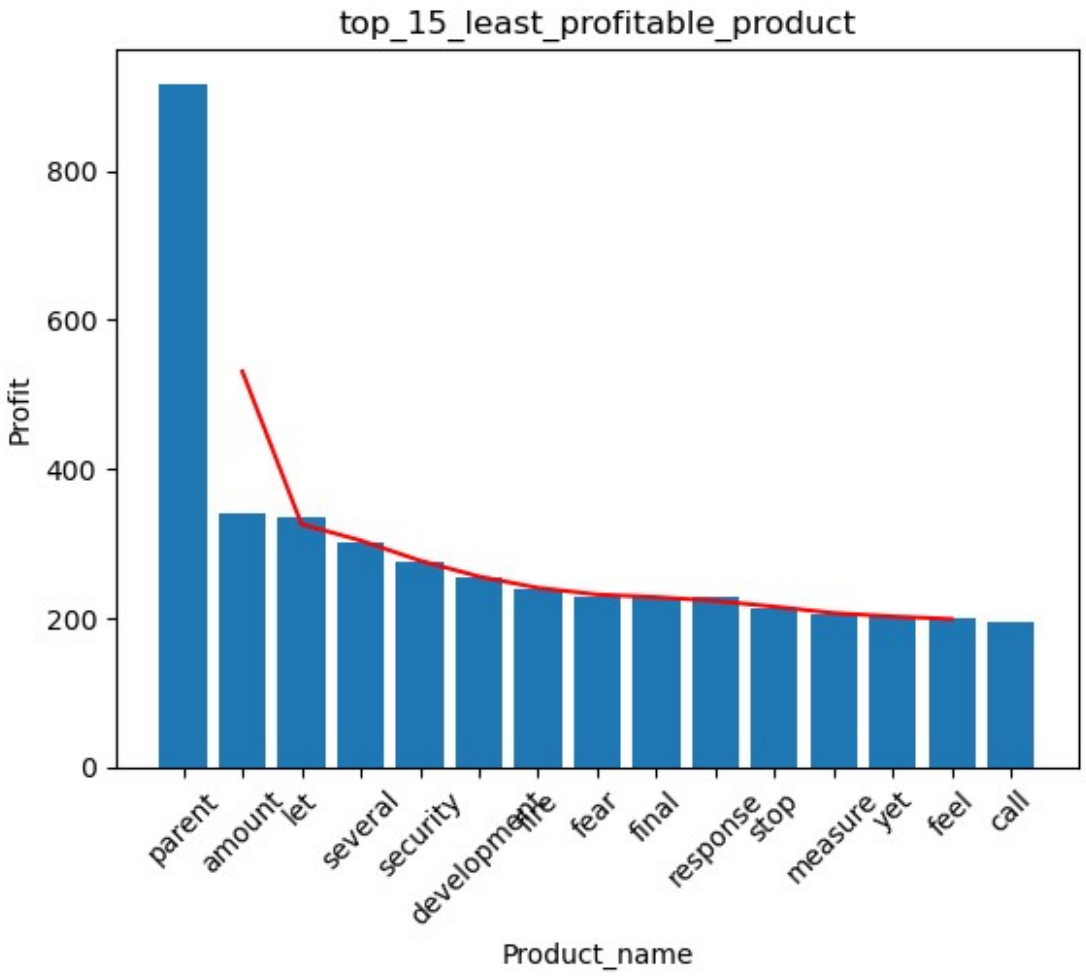
```
<Axes: xlabel='Category'>
```



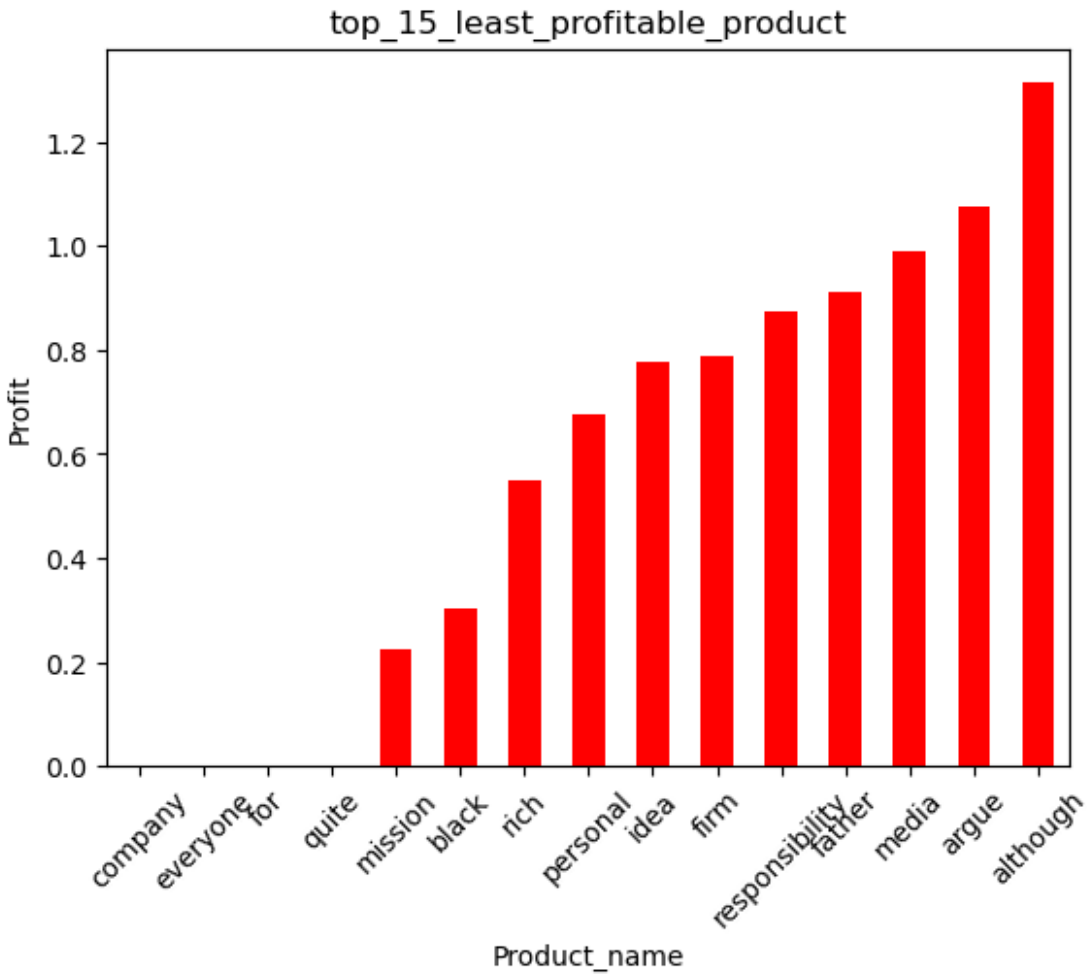

```
#Analyze the sales performance of products to identify best-selling
items and optimize inventory management strategies.

New_Pr_1=Shopping_data.groupby('Product_name')
['Profit'].sum().sort_values(ascending = False).head(15)
plt.bar(New_Pr_1.index,New_Pr_1.values)
plt.xticks(rotation=90)
plt.xlabel('Product_name')
plt.ylabel('Profit')
plt.title('top 15 profitable product')
plt.show()
```




```
New_Pr_2=Shopping_data.groupby('Product_name')
['Profit'].sum().sort_values(ascending = True).tail(15)
A=New_Pr_1.index
B=New_Pr_1.values
plt.bar(A,B)
plt.xticks(rotation=45)
plt.xlabel('Product_name')
plt.ylabel('Profit')
plt.title('top_15_least_profitable_product')
window = 3
trend = np.convolve(B, np.ones(window)/window, mode='valid')
trend_x = range(window//2, len(B) - window//2)
plt.plot(trend_x, trend, color='red', linestyle='-', label='Trend
(Moving Average)')
plt.show()
```

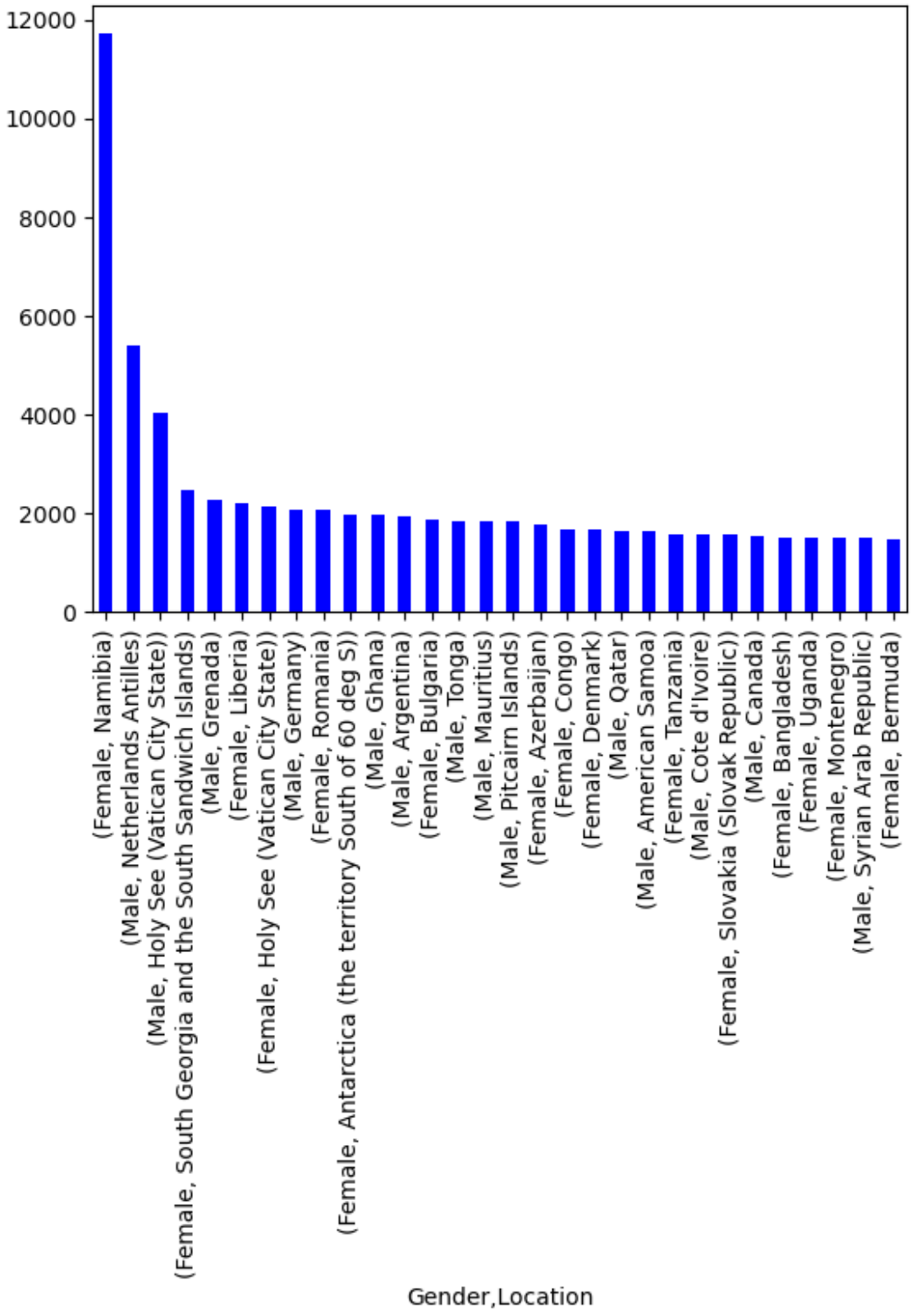



```
Shopping_data.groupby('Product_name')
['Profit'].sum().sort_values(ascending =
True).head(15).plot(kind='bar',color='red')
plt.xticks(rotation=45)
plt.xlabel('Product_name')
plt.ylabel('Profit')
plt.title('top_15_least_profitable_product')

Text(0.5, 1.0, 'top_15_least_profitable_product')
```




```
#How do sales vary across different customer segments (e.g., gender,
location)?

Shopping_data.groupby(['Gender','Location'])
['Sale_amount'].sum().sort_values(ascending
=False).head(30).plot(kind='bar',color='blue')
```

```
<Axes: xlabel='Gender,Location'>
```




#Gender_wise_sale_amount

```
Shopping_data.groupby('Gender')['Sale_amount'].sum()
```

```
Gender
Female    140298.012395
Male      132568.949688
Name: Sale_amount, dtype: float64
```

```
Pie_chart=Shopping_data.groupby('Gender')['Sale_amount'].sum()
Pie_chart.plot(kind='pie',autopct='%1.1f%%', startangle=45)
```

```
<Axes: ylabel='Sale_amount'>
```