```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("retail_sales_dataset.csv")
df.shape
df.columns = df.columns.str.strip()

df.head(30)
```

```
    Transaction ID        Date Customer ID  Gender  Age Product
Category  \
0                1  2023-11-24     CUST001    Male   34
Beauty
1                2  2023-02-27     CUST002  Female   26
Clothing
2                3  2023-01-13     CUST003    Male   50
Electronics
3                4  2023-05-21     CUST004    Male   37
Clothing
4                5  2023-05-06     CUST005    Male   30
Beauty
5                6  2023-04-25     CUST006  Female   45
Beauty
6                7  2023-03-13     CUST007    Male   46
Clothing
7                8  2023-02-22     CUST008    Male   30
Electronics
8                9  2023-12-13     CUST009    Male   63
Electronics
9               10  2023-10-07     CUST010  Female   52
Clothing
10              11  2023-02-14     CUST011    Male   23
Clothing
11              12  2023-10-30     CUST012    Male   35
Beauty
12              13  2023-08-05     CUST013    Male   22
Electronics
13              14  2023-01-17     CUST014    Male   64
Clothing
14              15  2023-01-16     CUST015  Female   42
Electronics
15              16  2023-02-17     CUST016    Male   19
Clothing
16              17  2023-04-22     CUST017  Female   27
Clothing
17              18  2023-04-30     CUST018  Female   47
Electronics
18              19  2023-09-16     CUST019  Female   62
```

```
Clothing
19              20  2023-11-05     CUST020    Male   22
Clothing
20              21  2023-01-14     CUST021  Female   50
Beauty
21              22  2023-10-15     CUST022    Male   18
Clothing
22              23  2023-04-12     CUST023  Female   35
Clothing
23              24  2023-11-29     CUST024  Female   49
Clothing
24              25  2023-12-26     CUST025  Female   64
Beauty
25              26  2023-10-07     CUST026  Female   28
Electronics
26              27  2023-08-03     CUST027  Female   38
Beauty
27              28  2023-04-23     CUST028  Female   43
Beauty
28              29  2023-08-18     CUST029  Female   42
Electronics
29              30  2023-10-29     CUST030  Female   39
Beauty

    Quantity  Price per Unit  Total Amount
0          3              50           150
1          2             500          1000
2          1              30            30
3          1             500           500
4          2              50           100
5          1              30            30
6          2              25            50
7          4              25           100
8          2             300           600
9          4              50           200
10         2              50           100
11         3              25            75
12         3             500          1500
13         4              30           120
14         4             500          2000
15         3             500          1500
16         4              25           100
17         2              25            50
18         2              25            50
19         3             300           900
20         1             500           500
21         2              50           100
22         4              30           120
23         1             300           300
```

```
24          1               50            50
25          2              500          1000
26          2               25            50
27          1              500           500
28          1               30            30
29          3              300           900
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Transaction ID    1000 non-null   int64
 1   Date              1000 non-null   object
 2   Customer ID       1000 non-null   object
 3   Gender            1000 non-null   object
 4   Age               1000 non-null   int64
 5   Product Category  1000 non-null   object
 6   Quantity          1000 non-null   int64
 7   Price per Unit    1000 non-null   int64
 8   Total Amount      1000 non-null   int64
dtypes: int64(5), object(4)
memory usage: 70.4+ KB
```

To know details about my csv file.

```
pd.isnull(df)
```

```
     Transaction ID   Date  Customer ID  Gender    Age  Product
Category  \
0             False  False        False   False  False
False
1             False  False        False   False  False
False
2             False  False        False   False  False
False
3             False  False        False   False  False
False
4             False  False        False   False  False
False
..              ...    ...          ...     ...    ...
..
995           False  False        False   False  False
False
996           False  False        False   False  False
False
```

```
997           False  False        False   False  False
False
998           False  False        False   False  False
False
999           False  False        False   False  False
False

     Quantity  Price per Unit  Total Amount
0       False           False         False
1       False           False         False
2       False           False         False
3       False           False         False
4       False           False         False
..        ...             ...           ...
995     False           False         False
996     False           False         False
997     False           False         False
998     False           False         False
999     False           False         False

[1000 rows x 9 columns]
```

To check null value.

```
#To check for null value clearly.'
pd.isnull(df).sum()

Transaction ID      0
Date                0
Customer ID         0
Gender              0
Age                 0
Product Category    0
Quantity            0
Price per Unit      0
Total Amount        0
dtype: int64
```

#I don't have any null value but if i have null value then i do df.dropna(inplace=True)

```
df.describe()

       Transaction ID         Age     Quantity  Price per Unit  Total
Amount
count     1000.000000  1000.00000  1000.000000     1000.000000
1000.000000
mean       500.500000    41.39200     2.514000      179.890000
456.000000
```

```
std         288.819436    13.68143     1.132734      189.681356
559.997632
min           1.000000    18.00000     1.000000       25.000000
25.000000
25%         250.750000    29.00000     1.000000       30.000000
60.000000
50%         500.500000    42.00000     3.000000       50.000000
135.000000
75%         750.250000    53.00000     4.000000      300.000000
900.000000
max        1000.000000    64.00000     4.000000      500.000000
2000.000000
```

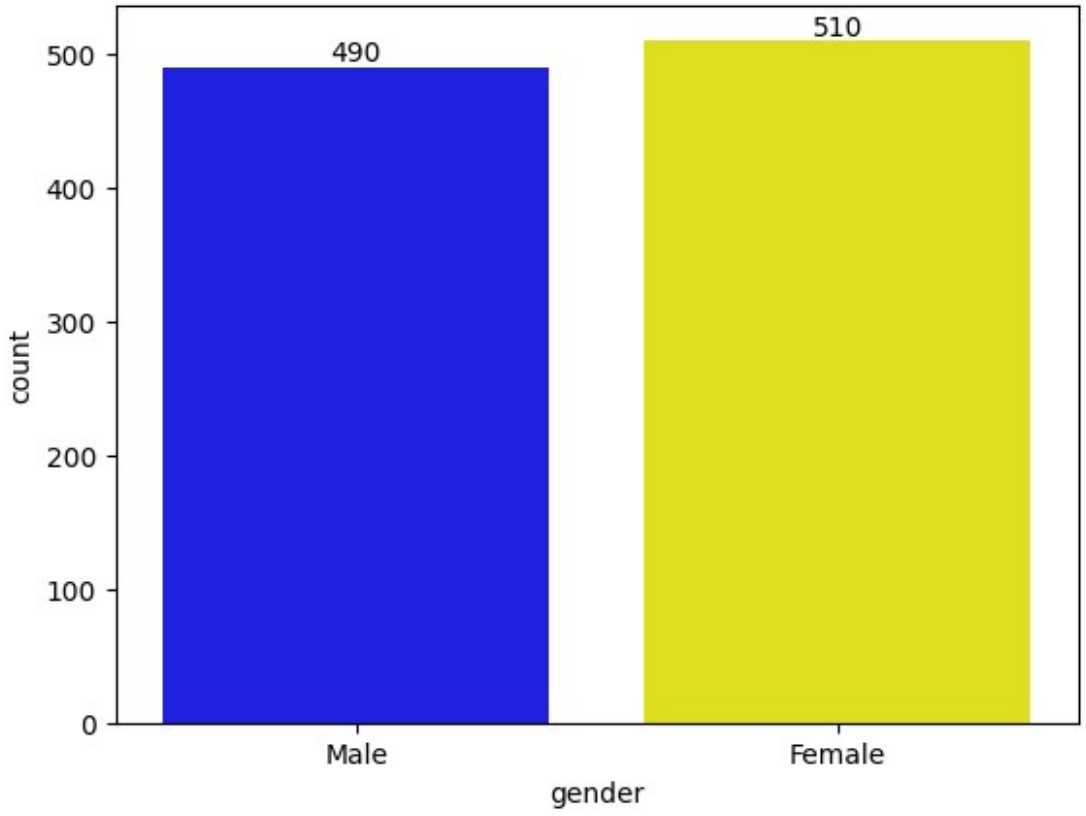
Exploratory data Analysis

Gender

```
df['date'] = pd.to_datetime(df['date'])

# Create Year and Month columns
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month

df['month_name'] = df['date'].dt.month_name()
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
df['date'] = pd.to_datetime(df['date'])

df.head()
```

```
   transaction_id       date customer_id  gender  age product_category
\
0               1 2023-11-24     CUST001    Male   34           Beauty

1               2 2023-02-27     CUST002  Female   26         Clothing

2               3 2023-01-13     CUST003    Male   50      Electronics

3               4 2023-05-21     CUST004    Male   37         Clothing

4               5 2023-05-06     CUST005    Male   30           Beauty


   quantity  price_per_unit  total_amount  year  month month_name
0         3              50           150  2023     11   November
1         2             500          1000  2023      2   February
2         1              30            30  2023      1    January
```

```
3          1              500            500  2023      5         May
4          2               50            100  2023      5         May

df.columns

Index(['transaction id', 'date', 'customer id', 'gender', 'age',
       'product category', 'quantity', 'price per unit', 'total
amount'],
      dtype='object')

ax = sns.countplot(x = 'gender',data = df,hue=
'gender',palette={'Male':'blue','Female':'yellow'},legend= False)

for bars in ax.containers:
    ax.bar_label(bars)
```



From above graph we can see that female are slightly higher than Male means purchasing power of Female are higher than male.

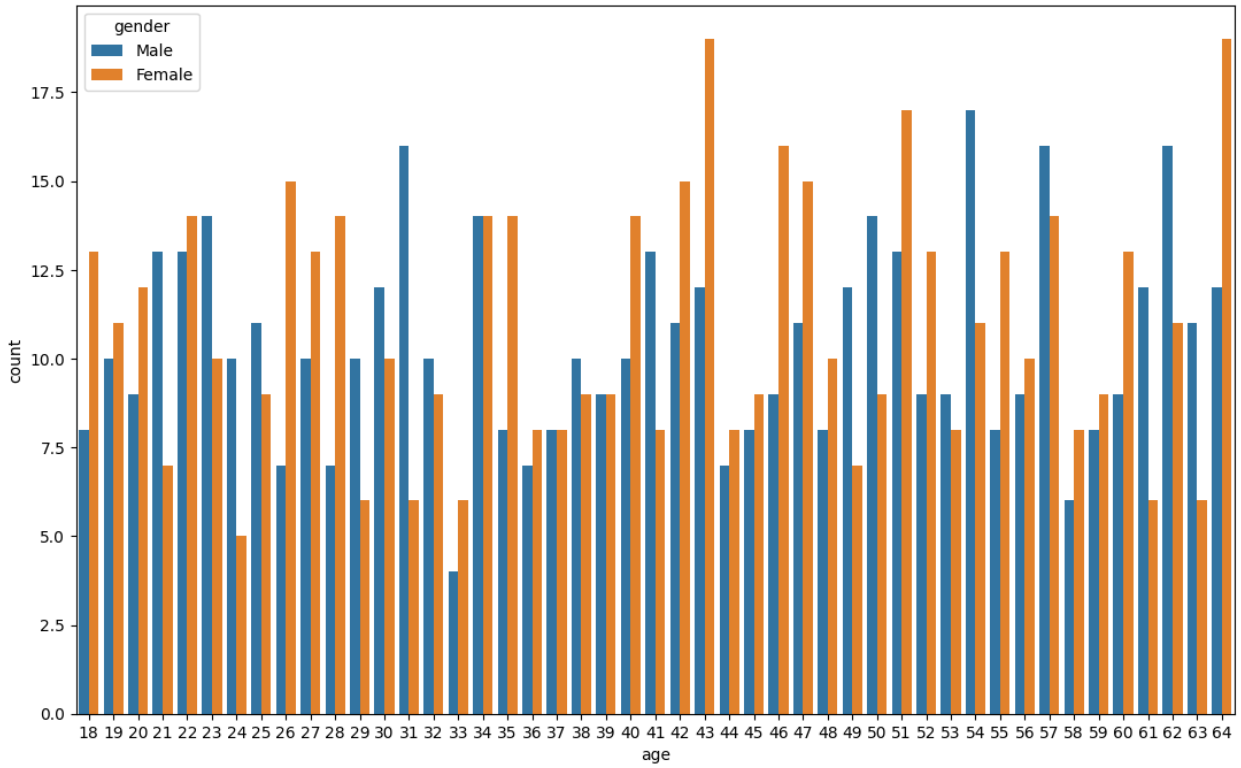
Age

```
df.columns
```

```
Index(['Transaction ID', 'Date', 'Customer ID', 'Gender', 'Age',
       'Product Category', 'Quantity', 'Price per Unit', 'Total
Amount'],
      dtype='object')
```

```
plt.figure(figsize=(13,8))
sns.countplot(x = 'age',data = df,hue= 'gender')
```

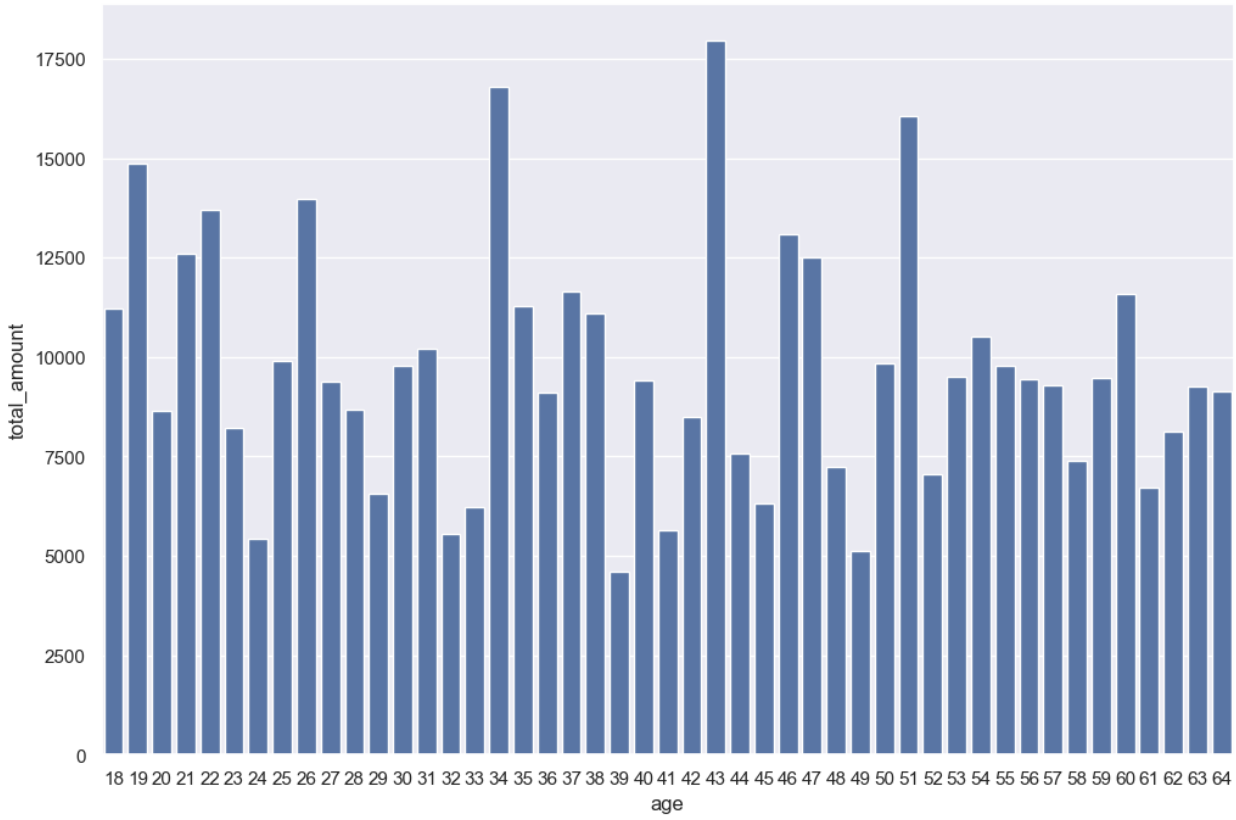
```
<Axes: xlabel='age', ylabel='count'>
```




```
#Total amount vs Age group
df.columns = (df.columns
    .str.strip()
    .str.lower()
    .str.replace(' ', '_'))


sales_age = (df.groupby('age', as_index=False)['total_amount']
      .sum()
      .sort_values(by='total_amount', ascending=False))
```

```
plt.figure(figsize=(12,8))
sns.barplot(x='age', y='total_amount', data=sales_age)
```

```
<Axes: xlabel='age', ylabel='total_amount'>
```



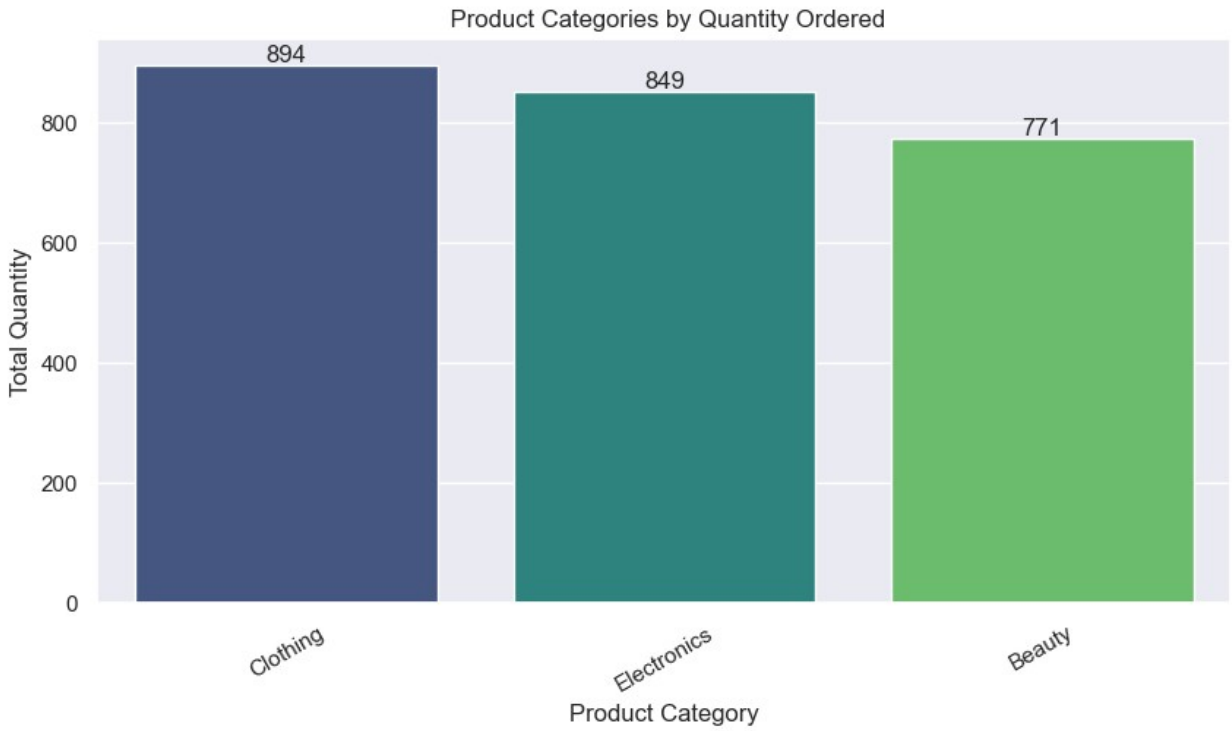
From above graphs we can see that most of the buyers age are 43,34,51

```
df.columns
```

```
Index(['transaction_id', 'date', 'customer_id', 'gender', 'age',
       'product_category', 'quantity', 'price_per_unit',
'total_amount'],
      dtype='object')
```

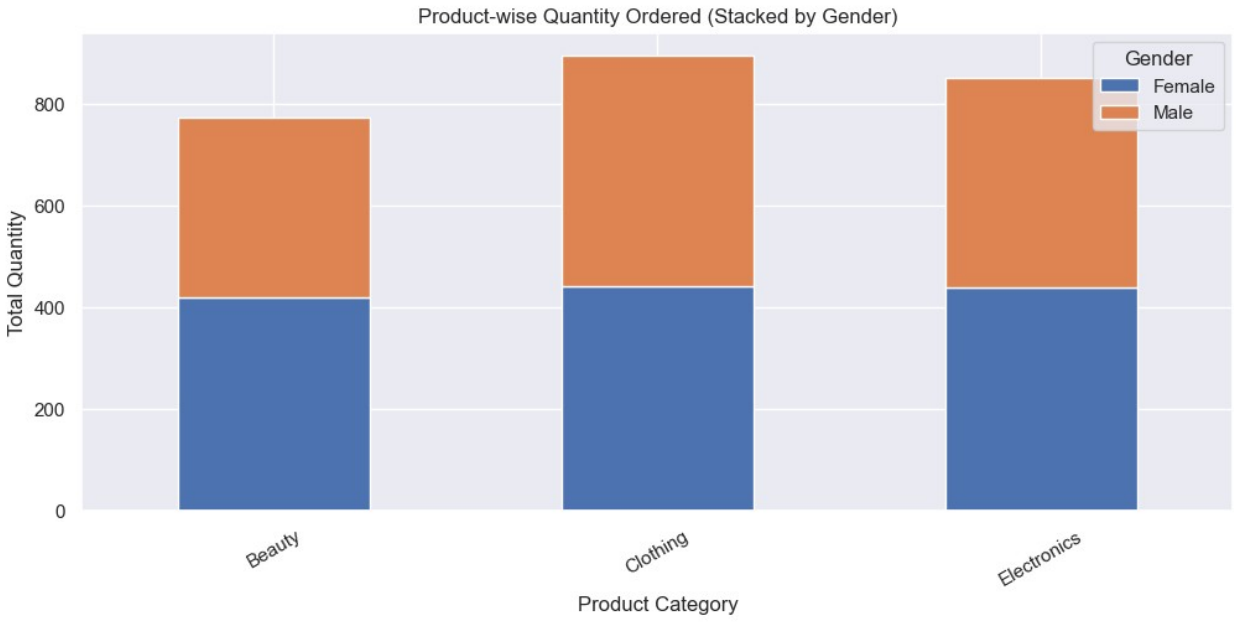
```
ax = sns.barplot( data=total_product_category,
    x='product_category',
    y='quantity',
    hue='product_category',
    palette='viridis',
    legend=False )
```

```
for container in ax.containers:
    ax.bar_label(container)

plt.title(" Product Categories by Quantity Ordered")
plt.xlabel("Product Category")
plt.ylabel("Total Quantity")
plt.xticks(rotation=30)
plt.show()
```



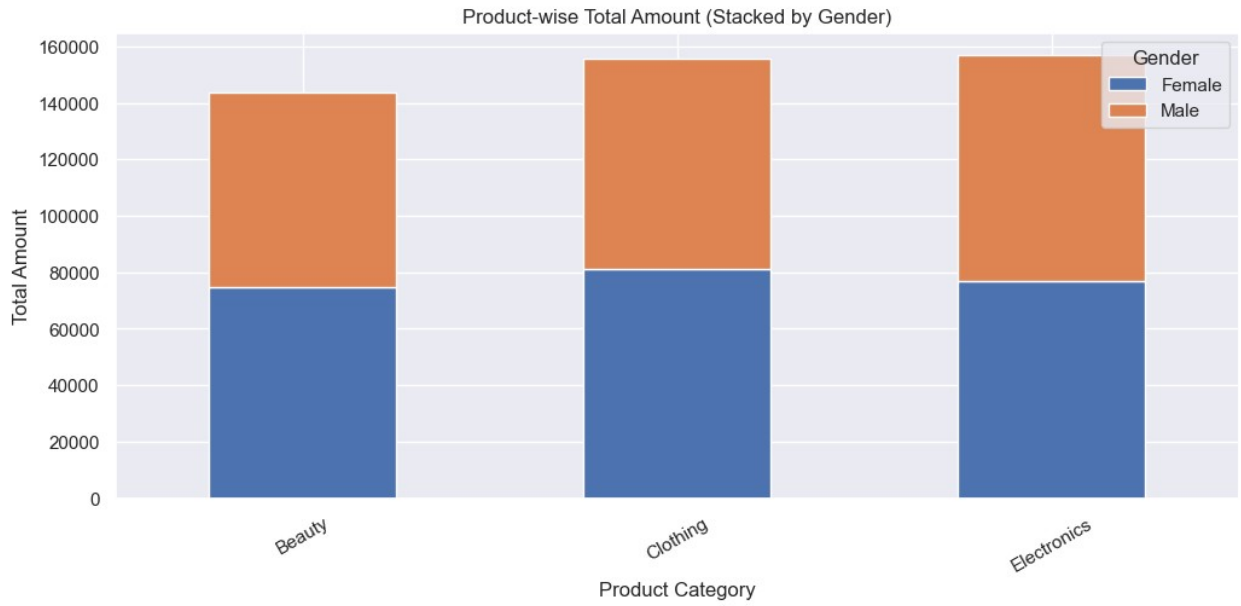

```
stacked_data = ( df.groupby(['product_category', 'gender'])
['quantity']
      .sum()
      .unstack())

stacked_data.plot( kind='bar',
    stacked=True,
    figsize=(12,5))

plt.title("Product-wise Quantity Ordered (Stacked by Gender)")
plt.xlabel("Product Category")
plt.ylabel("Total Quantity")
```

```
plt.xticks(rotation=30)
plt.legend(title="Gender")
plt.show()
```



from above graphs we can see that most of the orders are from clothing,then electronics and also show gender wise bar plot.

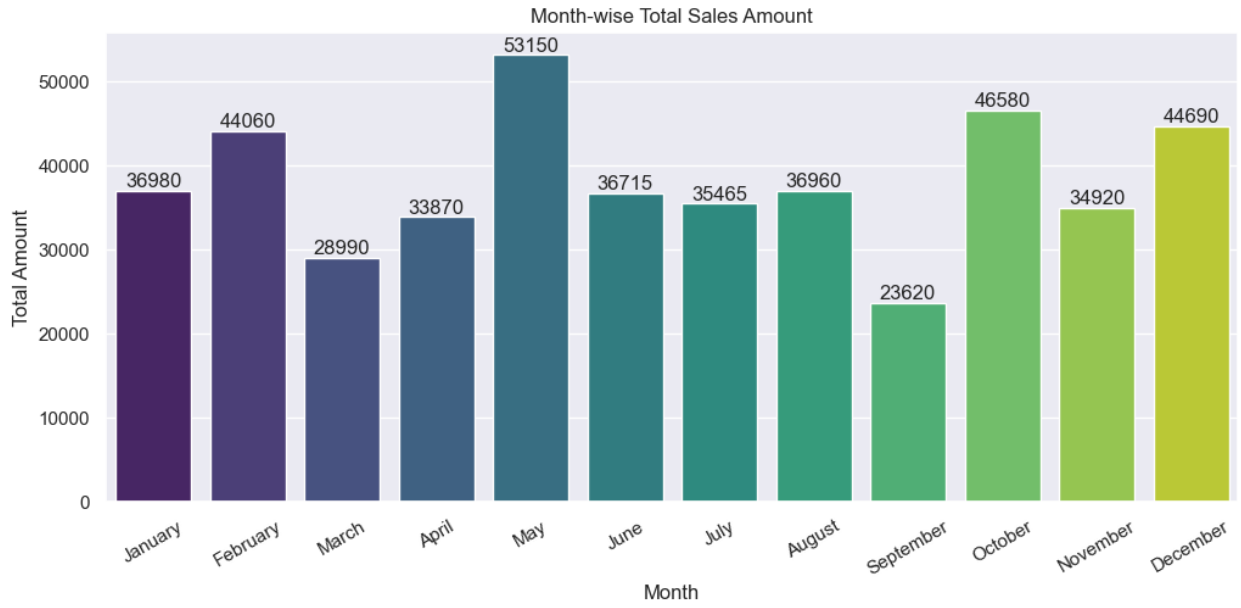
```
stacked_amount =(df.groupby(['product_category', 'gender'])
['total_amount']
      .sum()
      .unstack())

stacked_amount.plot(kind='bar',
    stacked=True,
    figsize=(12,5))

plt.title("Product-wise Total Amount (Stacked by Gender)")
plt.xlabel("Product Category")
plt.ylabel("Total Amount")
plt.xticks(rotation=30)
plt.legend(title="Gender")
plt.show()
```



#From above graphs we see gender wise total amount of product buy. we see male spent their money on clothing than beauty.

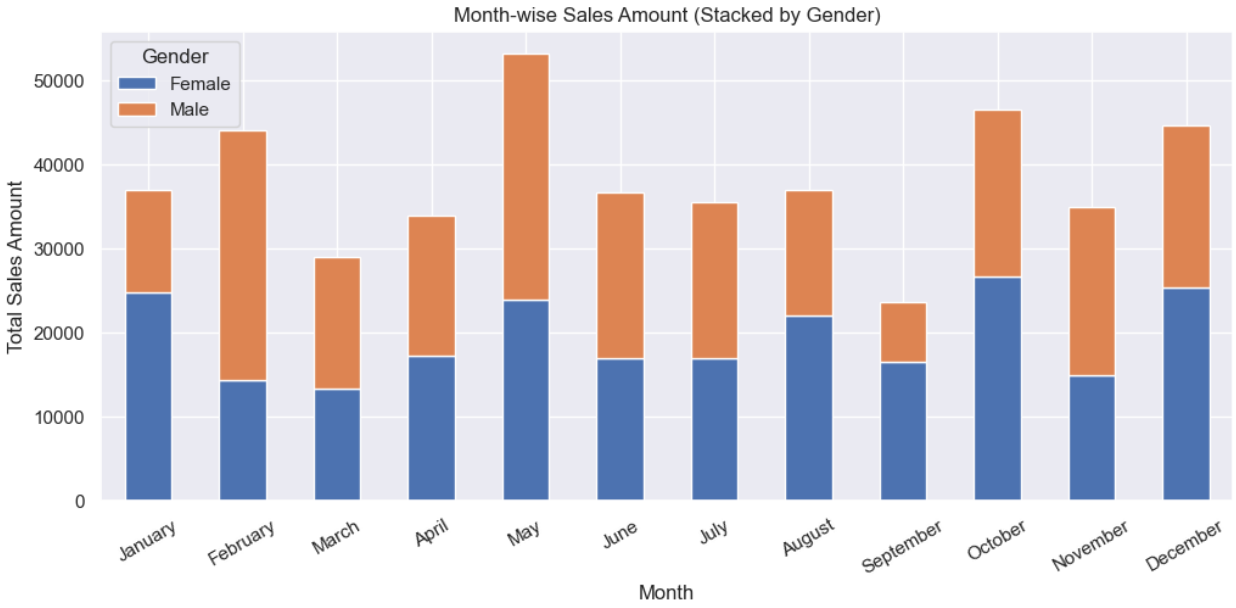
```
df['date'] = pd.to_datetime(df['date'])
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month
df['month_name'] = df['date'].dt.month_name()

monthly_sales = ( df.groupby('month_name', as_index=False)
['total_amount']
      .sum())

month_order = ['January','February','March','April','May','June',
    'July','August','September','October','November','December']

monthly_sales['month_name'] = pd.Categorical(
    monthly_sales['month_name'],
    categories=month_order,
    ordered=True)

monthly_sales = monthly_sales.sort_values('month_name')

plt.figure(figsize=(12,5))

ax = sns.barplot(
    data=monthly_sales,
    x='month_name',
    y='total_amount',
    hue='month_name',
    palette='viridis',
    legend=False)
```

```
for container in ax.containers:
    ax.bar_label(container, fmt='%.0f')

plt.title("Month-wise Total Sales Amount")
plt.xlabel("Month")
plt.ylabel("Total Amount")
plt.xticks(rotation=30)
plt.show()
```




```
df['date'] = pd.to_datetime(df['date'])
df['month_name'] = df['date'].dt.month_name()

month_order = [ 'January','February','March','April','May','June',
    'July','August','September','October','November','December']

df['month_name'] = pd.Categorical(
    df['month_name'],
    categories=month_order,
    ordered=True)

stacked_month_gender = (df.groupby(['month_name','gender'],observed=
False)['total_amount']
      .sum()
      .unstack())

stacked_month_gender.plot( kind='bar',
    stacked=True,
    figsize=(12,5))
```

```
plt.title("Month-wise Sales Amount (Stacked by Gender)")
plt.xlabel("Month")
plt.ylabel("Total Sales Amount")
plt.xticks(rotation=30)
plt.legend(title="Gender")
plt.show()
```



we see above geaph we see high amount buy of quantity , male purchasing may and february and female are purchasing january and december