


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import drive
drive.mount('/content/drive')
```

 Mounted at /content/drive

```
df=pd.read_csv('/content/retail_sales_dataset.csv')
df.head()
```



	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-	CUST003	Male	50	Electronics	1	30	30

Next steps:

[Generate code with df](#)
[View recommended plots](#)
[New interactive sheet](#)

```
df.tail()
```



	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-	CUST998	Female	33	Beauty	1	35	100

```
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

```

```
df.shape
```

```
(1000, 9)
```

```
df.isnull().sum()
```

```

0
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
```


```
df.describe()
```

```

Transaction ID      Age      Quantity  Price per Unit  Total Amount
count      1000.000000  1000.000000  1000.000000      1000.000000      1000.000000
mean         500.500000    41.392000    2.514000      179.890000      456.000000
std         288.819436    13.681430    1.132734      189.681356      559.997632
min           1.000000    18.000000    1.000000      25.000000      25.000000
25%          250.750000    29.000000    1.000000      30.000000      60.000000
50%          500.500000    42.000000    3.000000      50.000000     135.000000
75%          750.250000    53.000000    4.000000     300.000000     900.000000
max         1000.000000    64.000000    4.000000     500.000000    2000.000000

```

```
df.nunique()
```

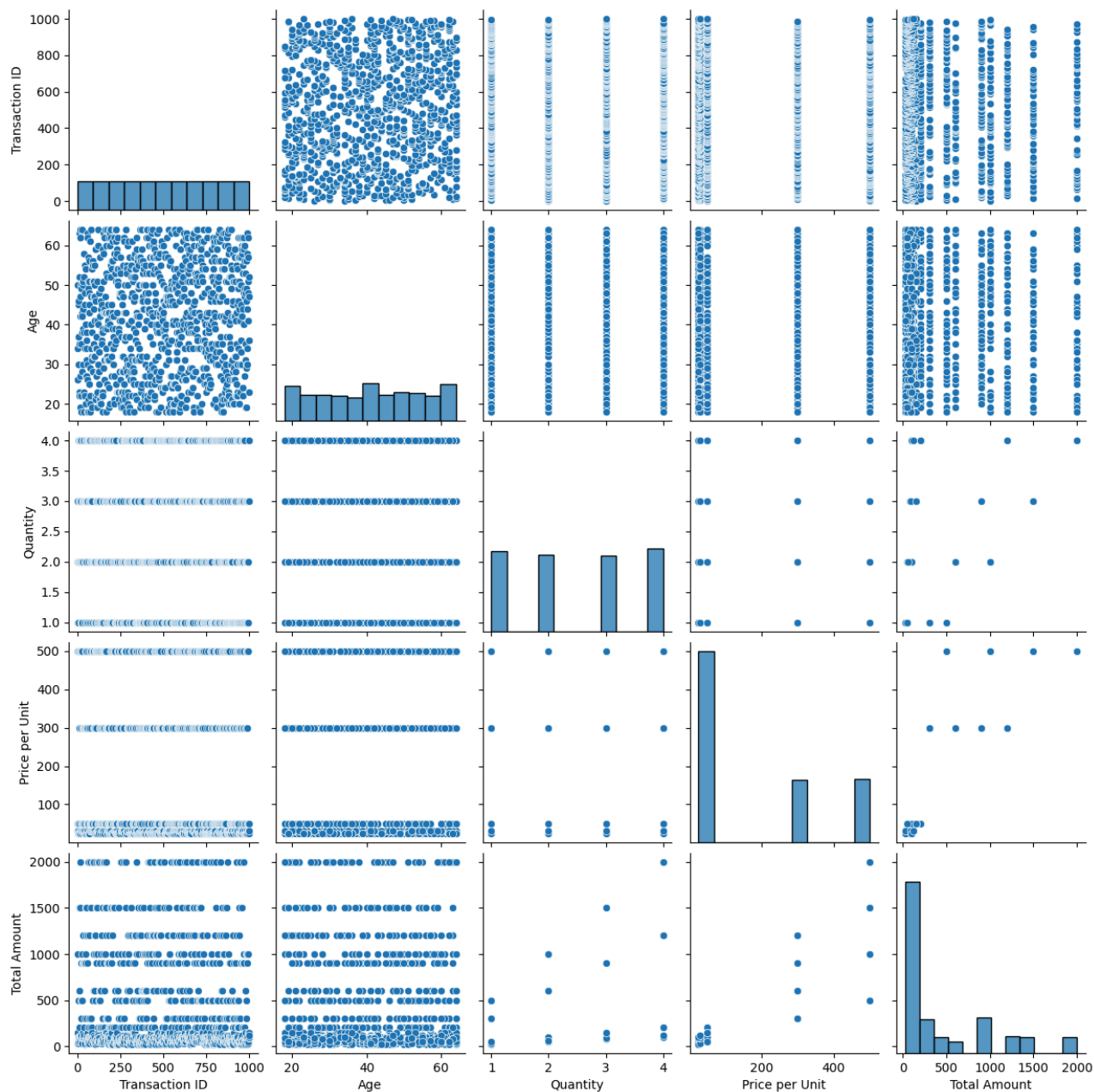


	0
Transaction ID	1000
Date	345
Customer ID	1000
Gender	2
Age	47
Product Category	3
Quantity	4
Price per Unit	5
Total Amount	18

dtype: int64

```
sns.pairplot(df)
```

↩ <seaborn.axisgrid.PairGrid at 0x7ee129db21b0>



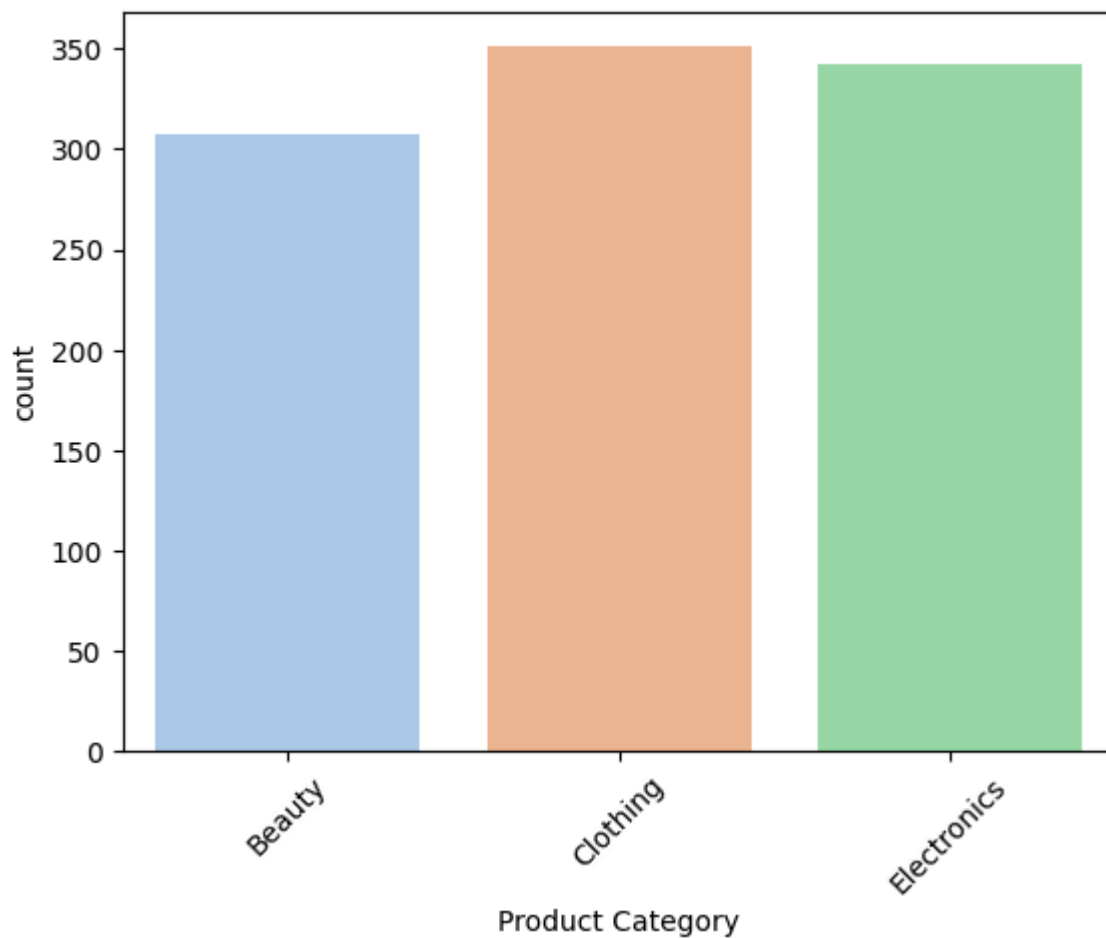
```
df[df['Product Category'].isin(['Clothing','Beauty','Electronics'])]['Product Category'].
```



	count
Product Category	
Clothing	351
Electronics	342
Beauty	307

dtype: int64

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x=df['Product Category'],palette='pastel')
plt.xticks(rotation=45)
plt.show()
```



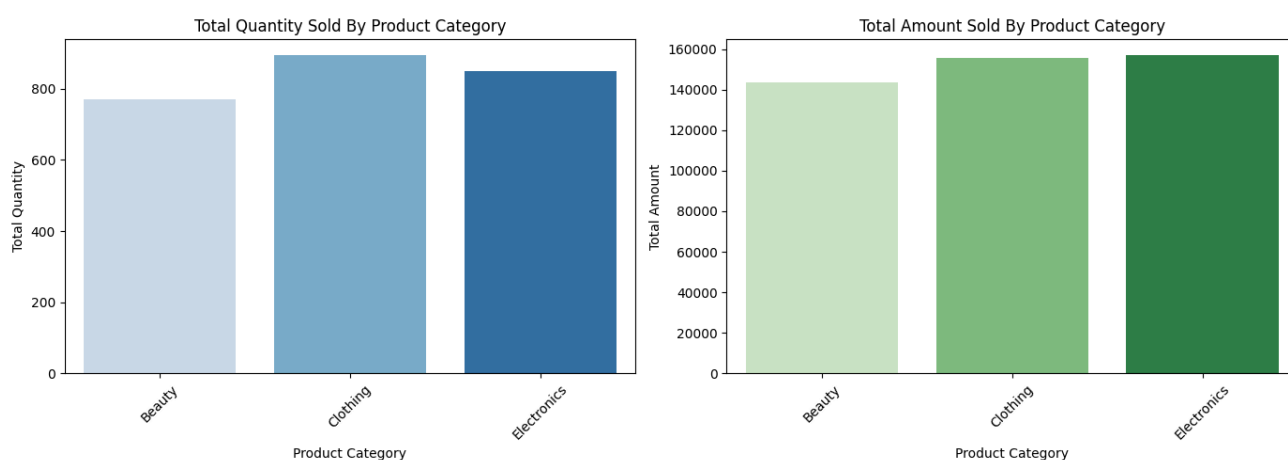
```

category_summary = df.groupby('Product Category')[['Quantity', 'Total Amount']].sum()
fig, ax = plt.subplots(1, 2, figsize=(14, 5))
sns.barplot(x=category_summary.index, y=category_summary['Quantity'], ax=ax[0], palette='Bl
ax[0].set_title('Total Quantity Sold By Product Category')
ax[0].set_xlabel('Product Category')
ax[0].set_ylabel('Total Quantity')
ax[0].tick_params(axis='x', rotation=45)

sns.barplot(x=category_summary.index, y=category_summary['Total Amount'], ax=ax[1], palette
ax[1].set_title('Total Amount Sold By Product Category')
ax[1].set_xlabel('Product Category')
ax[1].set_ylabel('Total Amount')
ax[1].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()

```



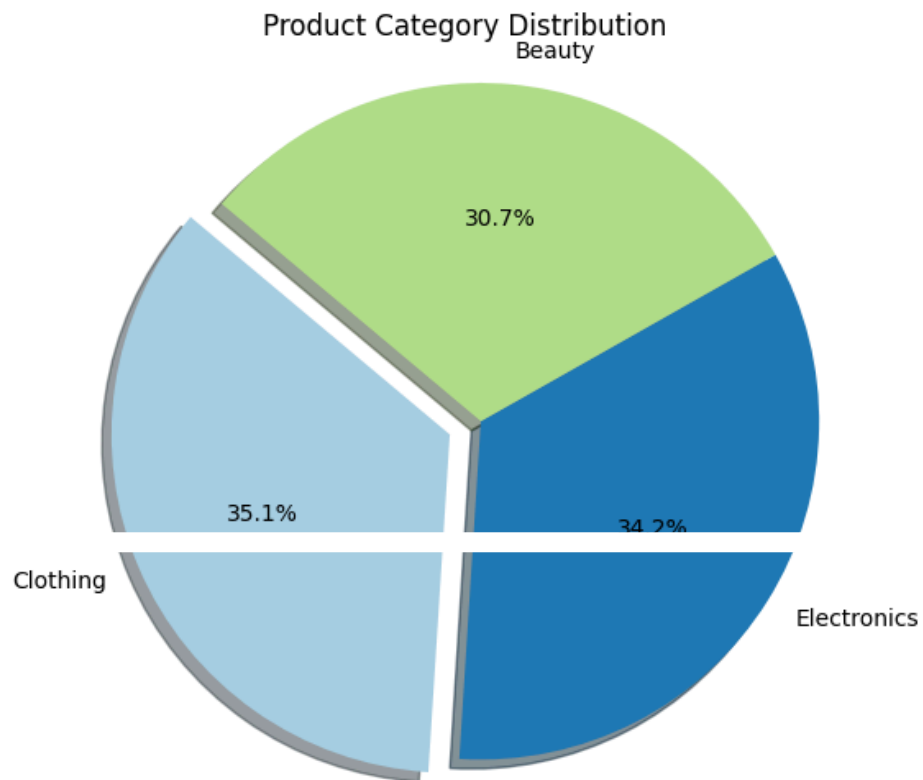
```

category_counts = df['Product Category'].value_counts()
explode_list = [0.1 if i == category_counts.idxmax() else 0 for i in category_counts.index]
plt.figure(figsize=(10, 6))
plt.pie(
    category_counts,
    labels=category_counts.index,
    autopct='%1.1f%%',
    colors=plt.cm.Paired.colors,
    explode=explode_list,
    startangle=140,
    shadow=True
)

plt.title("Product Category Distribution")
plt.axis("equal")

```

```
plt.show()
```



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

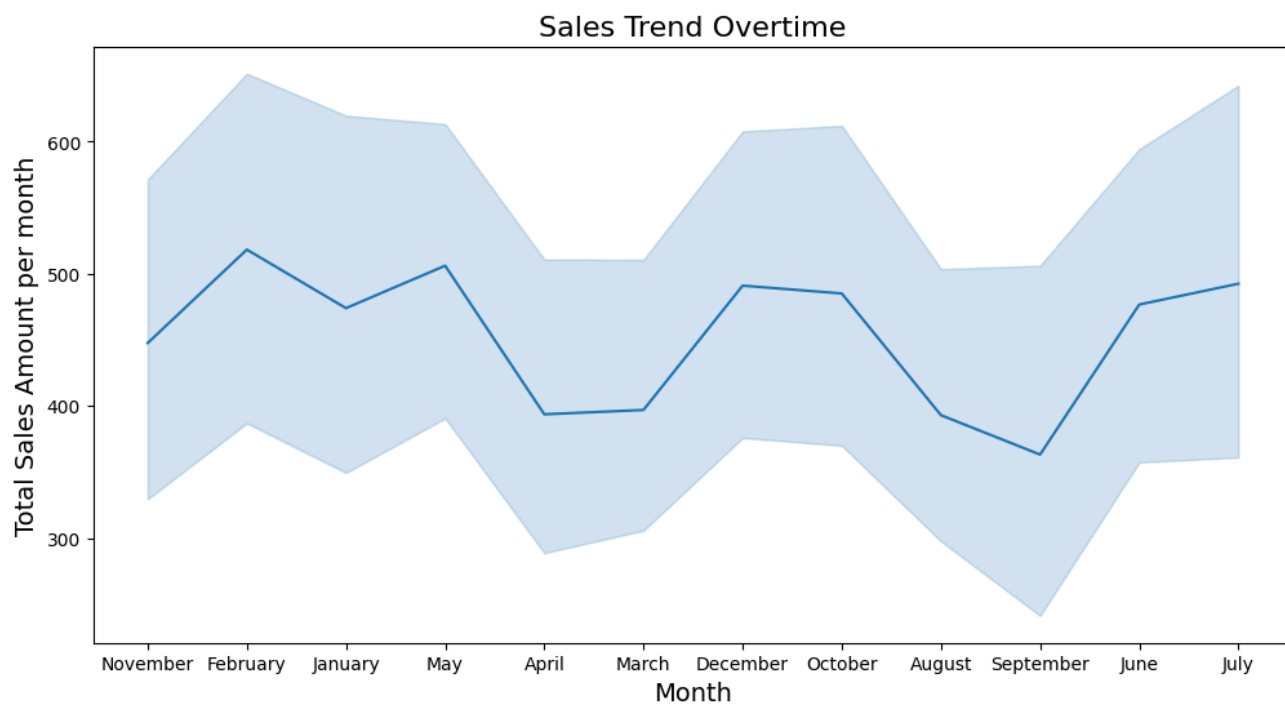


	Month
0	November
1	February
2	January
3	May
4	May
...	...
995	May
996	November
997	October
998	December
999	April

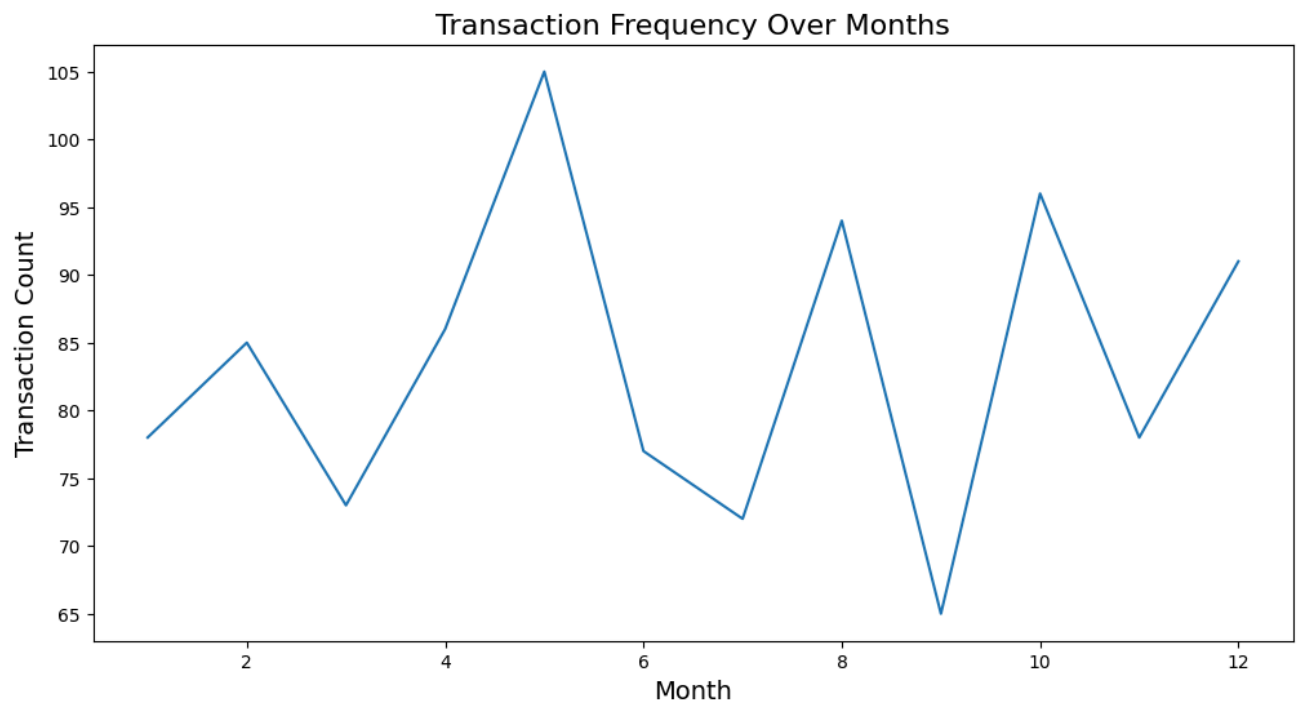
1000 rows × 1 columns

dtype: object

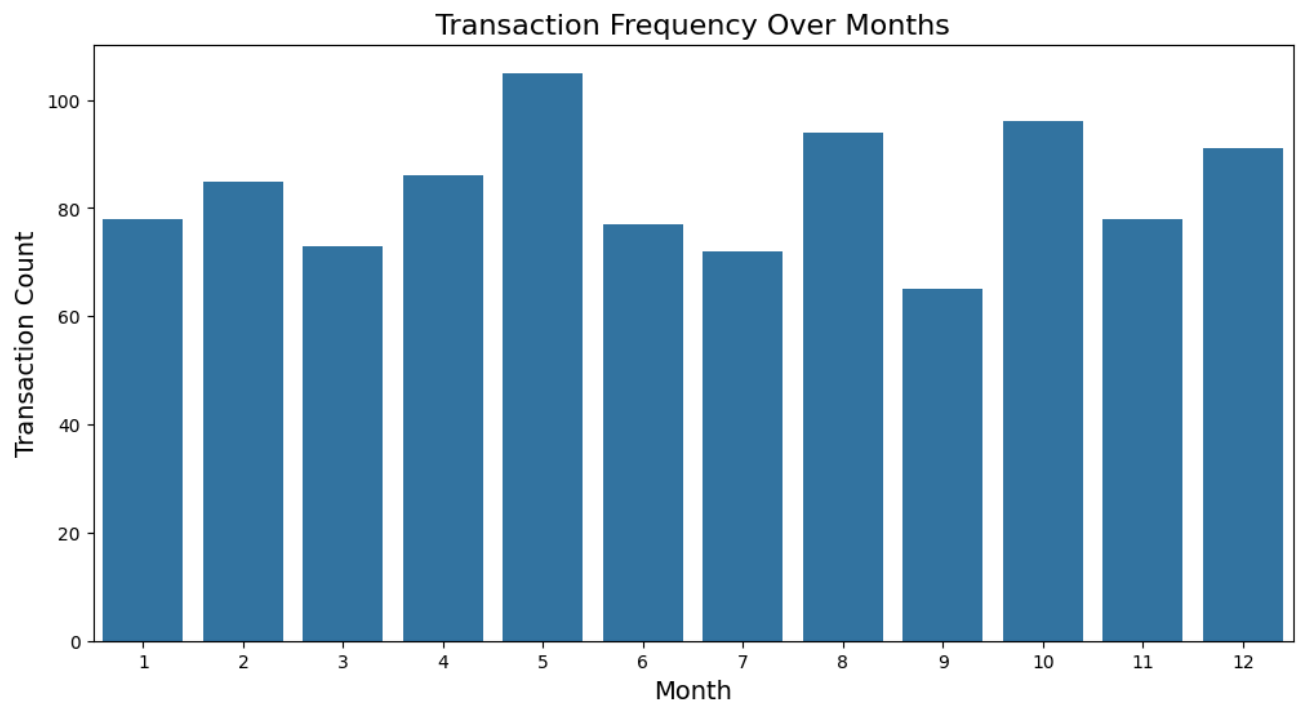
```
plt.figure(figsize=(12, 6))
sns.lineplot(x='Month', y='Total Amount', data=df)
plt.title('Sales Trend Overtime', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Total Sales Amount per month', fontsize=14)
plt.show()
```

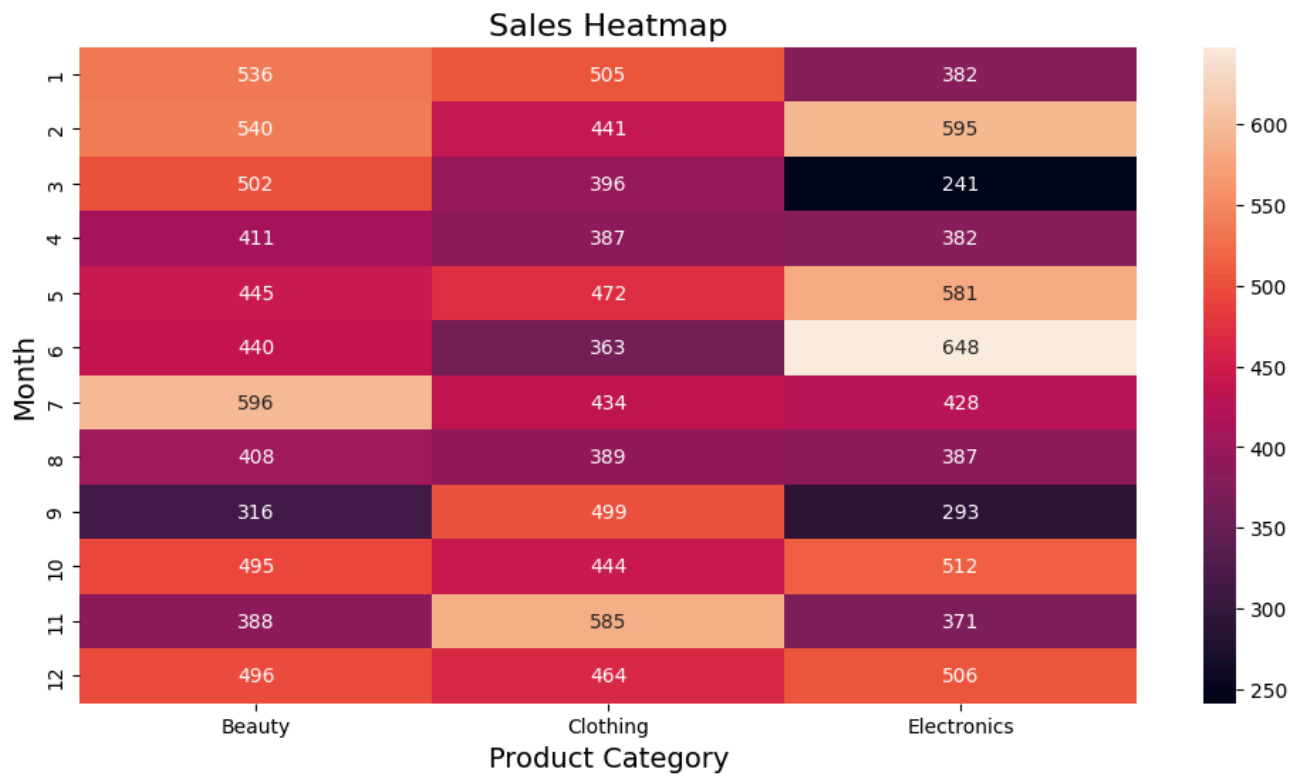
```
df['Month'] = df['Date'].dt.month
transaction_count = df.groupby('Month')['Transaction ID'].count()
plt.figure(figsize=(12, 6))
sns.lineplot(x=transaction_count.index, y=transaction_count.values)
plt.title('Transaction Frequency Over Months', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Transaction Count', fontsize=14)
plt.show()
```



```
plt.figure(figsize=(12, 6))
sns.barplot(x=transaction_count.index, y=transaction_count.values)
plt.title('Transaction Frequency Over Months', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Transaction Count', fontsize=14)
plt.show()
```



```
heatmap = df.pivot_table(index='Month', columns='Product Category', values='Total Amount')
plt.figure(figsize=(12, 6))
sns.heatmap(heatmap, annot=True, fmt='.0f')
plt.title('Sales Heatmap', fontsize=16)
plt.xlabel('Product Category', fontsize=14)
plt.ylabel('Month', fontsize=14)
plt.show()
```



```
df['Age'].count()
```

```
np.int64(1000)
```

```
df[df['Gender'].isin(['Male', 'Female'])['Gender'].value_counts()
```



```
count
```

Gender	
Female	510
Male	490

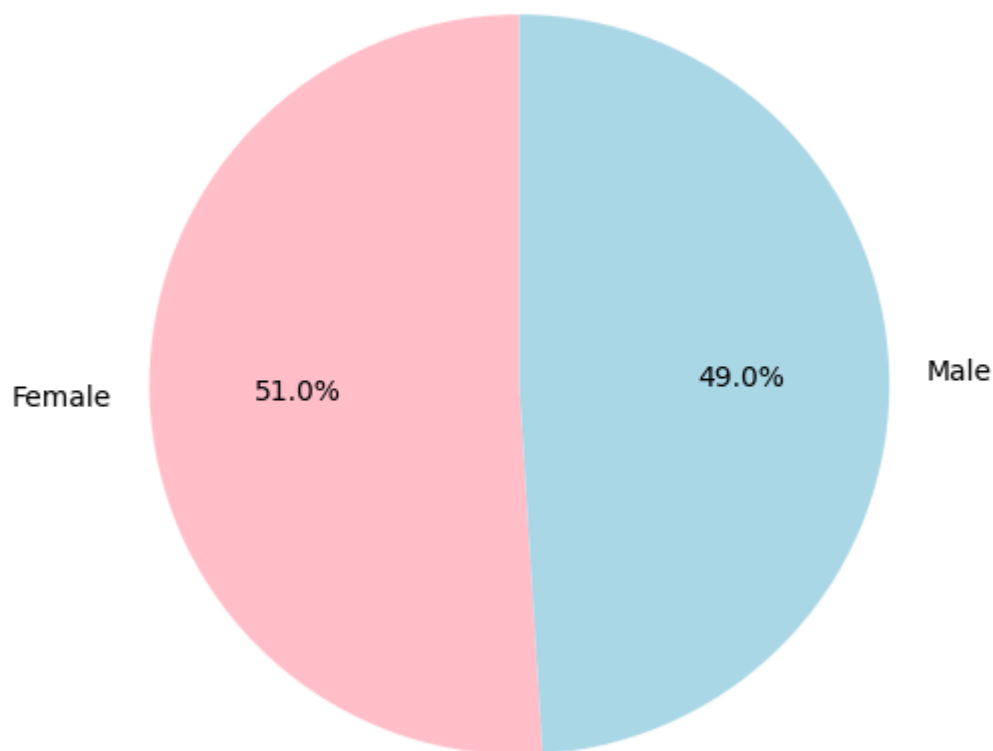
dtype: int64

```
gender_counts = {'Female': 510, 'Male': 490}
plt.figure(figsize=(6, 6))
plt.pie(
    gender_counts.values(),
    labels=gender_counts.keys(),
    autopct='%1.1f%%',
    colors=['pink', 'lightblue'],
```

```
startangle=90
)
plt.title('Gender Distribution')
plt.show()
```



Gender Distribution



```
age_bins = [0, 18, 25, 35, 50, 100]
age_group = ['0-18', '19-25', '26-35', '36-50', '50+']
df['Age Group'] = pd.cut(df['Age'], bins=age_bins, labels=age_group)
df.head()
```



	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	Mont
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150	1
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000	
2	3	2023-	CUST003	Male	50	Electronics	1	20	20	

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(12, 6))
sns.swarmplot(x='Product Category', y='Age', hue='Gender', data=df, palette="Set2")
plt.title('Distribution of Age and Gender across Product Category', fontsize=16)
plt.xlabel('Product Category', fontsize=14)
plt.ylabel('Age', fontsize=14)
plt.legend(title='Gender', loc='upper right', fontsize=10)
plt.show()
```



```
plt.figure(figsize=(20, 10))
plt.title("Age of Persons")
plt.xlabel("Age")
```