

# Ajio\_Customer & Sales Analysis AJIO

```
In [39]: import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns
```

```
In [40]: customer = pd.read_csv("customer.csv")  
orders = pd.read_csv("orders.csv")  
products = pd.read_csv("products.csv")  
transaction = pd.read_csv("transaction.csv")  
ratings = pd.read_csv("ratings.csv")  
returns = pd.read_csv("returns.csv")  
delivery = pd.read_csv("delivery.csv")
```

```
In [43]: customer.head()
```

Out[43]:

	C_ID	C_Name	Gender	Age	City	State	Street_Address	Mobile
0	CS_11000001	Manbir Lala	Male	67	Delhi	Delhi	Park Ave, 163 , Delhi , Delhi - 529675	9607971039
1	CS_11000002	Radhika More	Female	51	Pune	Maharashtra	Elm St, 960 , Pune , Maharashtra - 328062	9109249091
2	CS_11000003	Faqid Halder	Female	57	Bengaluru	Karnataka	Maple St, 71 , Bengaluru , Karnataka - 574209	9129509047
3	CS_11000004	Chandresh Dugar	Female	26	Thane	Maharashtra	2nd St, 557 , Thane , Maharashtra - 329555	9351639395
4	CS_11000005	Logan Soni	Male	24	Ghaziabad	Uttar Pradesh	Pine St, 758 , Ghaziabad , Uttar Pradesh - 119526	9445754174

```
In [45]: customer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   C_ID             10000 non-null   object  
 1   C_Name            10000 non-null   object  
 2   Gender            10000 non-null   object  
 3   Age               10000 non-null   int64  
 4   City              10000 non-null   object  
 5   State             10000 non-null   object  
 6   Street_Address    10000 non-null   object  
 7   Mobile            10000 non-null   int64  
dtypes: int64(2), object(6)
memory usage: 625.1+ KB
```

In [47]: `customer.describe()`

Out[47]:

	Age	Mobile
<b>count</b>	10000.000000	1.000000e+04
<b>mean</b>	44.006400	9.503873e+09
<b>std</b>	15.325647	2.911932e+08
<b>min</b>	18.000000	9.000042e+09
<b>25%</b>	31.000000	9.250165e+09
<b>50%</b>	44.000000	9.507420e+09
<b>75%</b>	57.000000	9.756789e+09
<b>max</b>	70.000000	9.999973e+09

In [51]: `customer.duplicated().sum()`

Out[51]: 0

In [53]: `orders.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Or_ID       10000 non-null   object  
 1   C_ID        10000 non-null   object  
 2   P_ID        10000 non-null   object  
 3   Order_Date  10000 non-null   object  
 4   Order_Time  10000 non-null   object  
 5   Qty          10000 non-null   int64  
 6   Coupon       10000 non-null   object  
 7   DP_ID       10000 non-null   object  
 8   Discount     10000 non-null   int64  
dtypes: int64(2), object(7)
memory usage: 703.3+ KB
```

```
In [55]: orders.describe()
```

```
Out[55]:
```

	Qty	Discount
<b>count</b>	10000.000000	10000.000000
<b>mean</b>	5.513600	9.066700
<b>std</b>	2.882195	11.828941
<b>min</b>	1.000000	0.000000
<b>25%</b>	3.000000	0.000000
<b>50%</b>	6.000000	0.000000
<b>75%</b>	8.000000	15.000000
<b>max</b>	10.000000	50.000000

```
In [57]: products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   P_ID            10000 non-null   object  
 1   P_Name           10000 non-null   object  
 2   Category         10000 non-null   object  
 3   Company_Name    10000 non-null   object  
 4   Gender           10000 non-null   object  
 5   Price            10000 non-null   int64  
dtypes: int64(1), object(5)
memory usage: 468.9+ KB
```

```
In [59]: transaction.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Tr_ID            10000 non-null   object  
 1   Or_ID            10000 non-null   object  
 2   Transaction_Mode 10000 non-null   object  
 3   Reward           10000 non-null   object  
dtypes: object(4)
memory usage: 312.6+ KB
```

```
In [61]: ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   R_ID             10000 non-null   object  
 1   Or_ID            10000 non-null   object  
 2   Prod_Rating      10000 non-null   int64  
 3   Delivery_Service_Rating 10000 non-null   int64  
dtypes: int64(2), object(2)
memory usage: 312.6+ KB
```

```
In [63]: returns.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   RT_ID       2000 non-null   object  
 1   Or_ID       2000 non-null   object  
 2   Reason      2000 non-null   object  
 3   Return_Refund  2000 non-null  object  
 4   Dates       2000 non-null   object  
dtypes: object(5)
memory usage: 78.3+ KB
```

```
In [67]: delivery.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   DP_ID       5 non-null    object  
 1   DP_name     5 non-null    object  
 2   DP_Ratings  5 non-null    int64  
 3   Percent_Cut 5 non-null    int64  
dtypes: int64(2), object(2)
memory usage: 292.0+ bytes
```

## Unique customers

```
In [110...]: customer['C_ID'].nunique()    # Count of unique customer IDs
```

```
Out[110...]: 10000
```

## Total orders placed

```
In [115...]: orders['Or_ID'].nunique() #Total number of orders
```

```
Out[115...]: 10000
```

## Gender distribution

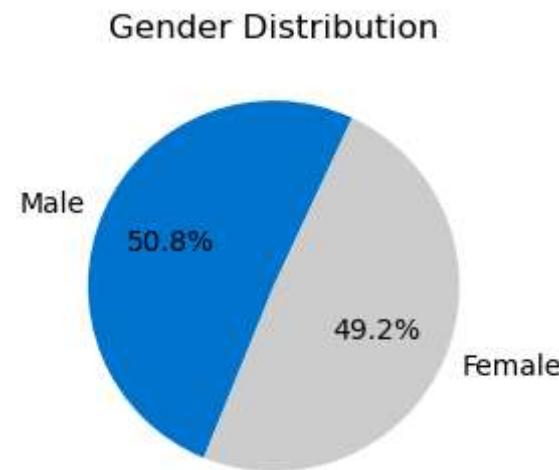
```
In [449...]: print(customer['Gender'].value_counts())

customer['Gender'].value_counts().plot.pie(autopct='%1.1f%%',
                                             title='Gender Distribution',
                                             figsize=(3, 3),
                                             colors=['#0074CC', '#CCCCCC'],
                                             ylabel='', startangle=65) # Pie chart shows the male vs female customer percentage.
```

Gender

Male 5080  
Female 4920  
Name: count, dtype: int64

```
Out[449...]: <Axes: title={'center': 'Gender Distribution'}>
```



## Top 5 Cities by Number of Customers

```
In [414...]: customer['City'].value_counts().head(5)
```

```
Out[414...]: City
Vadodara    531
Surat       529
Jaipur      525
Kanpur      520
Patna       518
Name: count, dtype: int64
```

## Top 5 states by number of customers

```
In [445...]: print(customer['State'].value_counts().head(5))

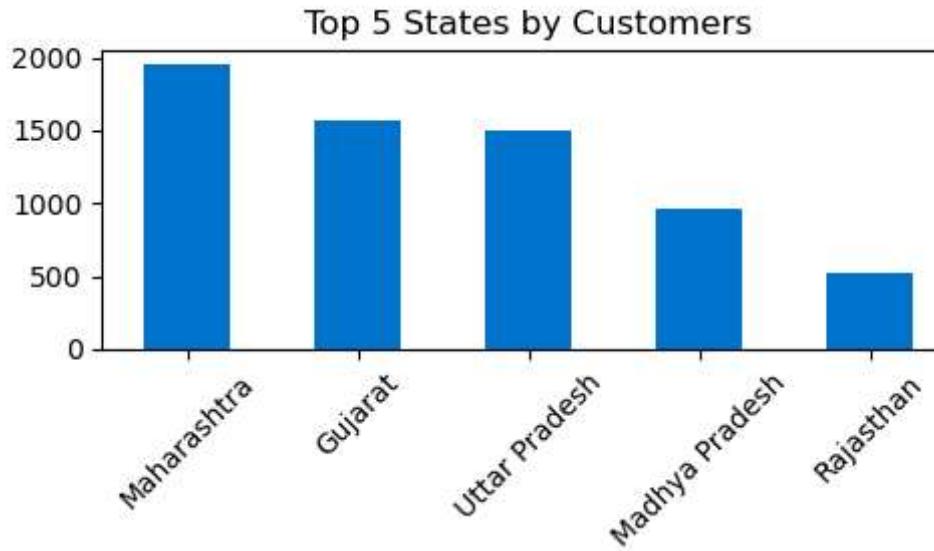
plt.figure(figsize=(5, 3))

customer['State'].value_counts().head(5).plot(
    kind='bar',
    color='#0074CC',
    title='Top 5 States by Customers'
)

plt.xlabel("")

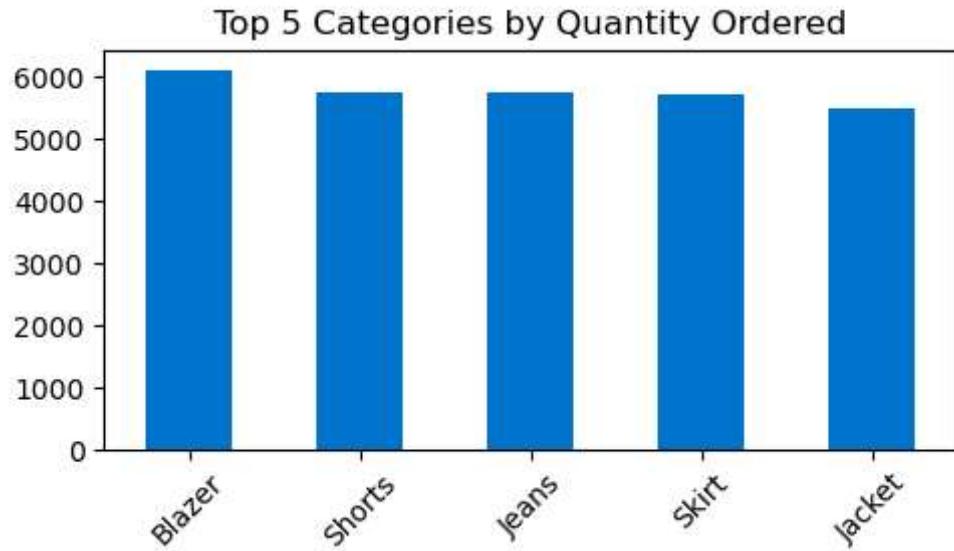
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
State
Maharashtra    1949
Gujarat        1571
Uttar Pradesh   1501
Madhya Pradesh  965
Rajasthan       525
Name: count, dtype: int64
```



## Top 5 most ordered product categories

```
In [306...]:  
orders_products = orders.merge(products, on='P_ID')  
top_categories = orders_products.groupby('Category')[['Qty']].sum().sort_values(ascending=False).head(5)  
  
plt.figure(figsize=(5, 3))  
top_categories.plot(  
    kind='bar',  
    color="#0074CC",  
    title='Top 5 Categories by Quantity Ordered'  
)  
  
plt.xlabel('')  
plt.ylabel('')  
plt.xticks(rotation=45)  
plt.tight_layout()  
plt.show()
```



## Average quantity per order

```
In [281]: orders['Qty'].mean()
```

```
Out[281]: 5.5136
```

## Top 5 most purchased products by quantity

```
In [279]: orders_merged = pd.merge(orders, products, on='P_ID')
top_products = orders_merged.groupby('P_Name')['Qty'].sum().sort_values(ascending=False).head(5)
print(top_products)
```

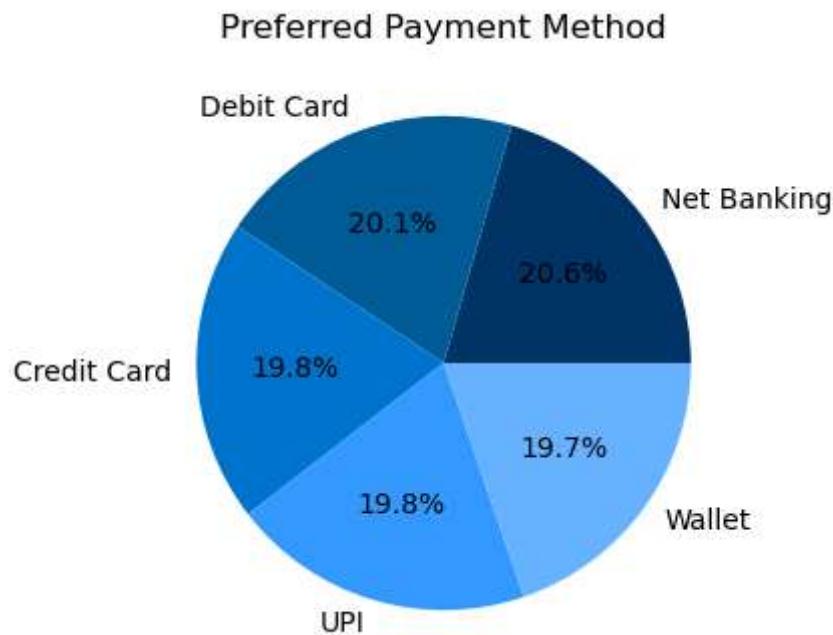
P_Name	Qty
Shift Cotton Charcoal Polka Dot Dress	43
Single-Breasted Linen Navy Blue Checkered Blazer	42
Pleated Mesh Gray Textured Shorts	40
Pencil Silk Brown Floral Skirt	37
Cotton Maroon Ribbed Shorts	37

Name: Qty, dtype: int64

## Most preferred payment method

In [443...]

```
transaction['Transaction_Mode'].value_counts().plot(  
    kind='pie',  
    autopct='%1.1f%%',  
    figsize=(4, 4),  
    colors=['#003366', '#005B96', '#0074CC', '#3399FF', '#66B2FF'],  
    title='Preferred Payment Method'  
)  
  
plt.ylabel('')  
plt.show()
```



## The average order value (AOV)

```
In [476...]  
orders_txn = pd.merge(orders, products, on='P_ID')  
orders_txn['Total_Amount'] = orders_txn['Price'] * orders_txn['Qty'] # Calculate total amount for each product in an order  
aov = orders_txn.groupby('Or_ID')['Total_Amount'].sum().mean()  
print("Average Order Value:", aov)
```

Average Order Value: 9732.7078

## Distribution of product ratings

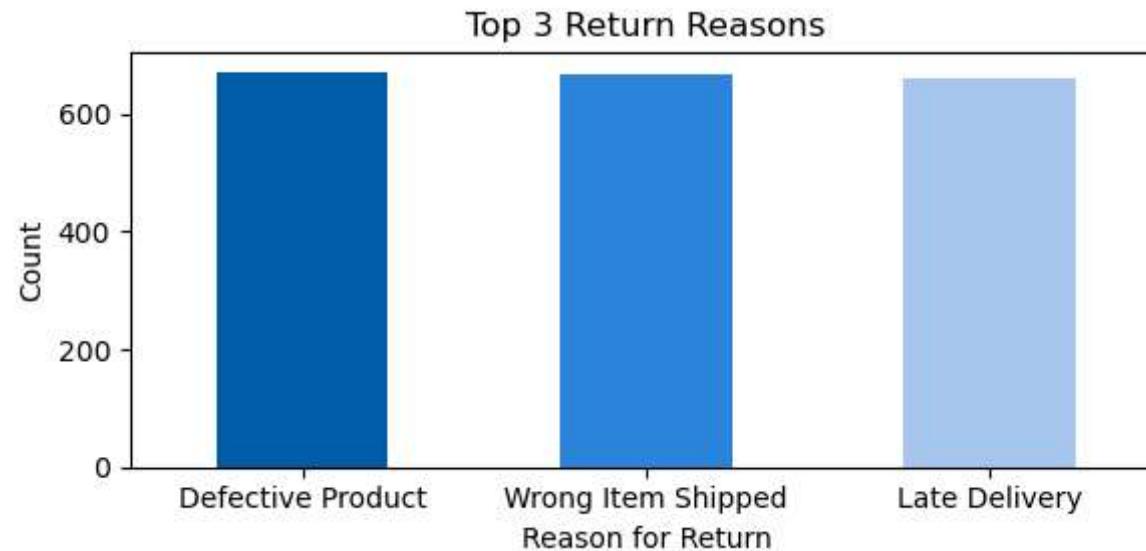
```
In [346...]  
ratings['Prod_Rating'].value_counts()
```

```
Out[346...]  
Prod_Rating  
3    2052  
4    2012  
2    2003  
1    1988  
5    1945  
Name: count, dtype: int64
```

## Top 3 return reasons

```
In [437...]  
print(returns['Reason'].value_counts())  
  
plt.figure(figsize=(6,3))  
returns['Reason'].value_counts().head(3).plot(  
    kind='bar',  
    title='Top 3 Return Reasons',  
    color=['#005DAA', '#2B84D9', '#A7C6ED'])  
plt.xticks(rotation=0)  
plt.xlabel("Reason for Return")  
plt.ylabel("Count")  
plt.tight_layout()  
plt.show()
```

```
Reason
Defective Product    670
Wrong Item Shipped   668
Late Delivery        662
Name: count, dtype: int64
```

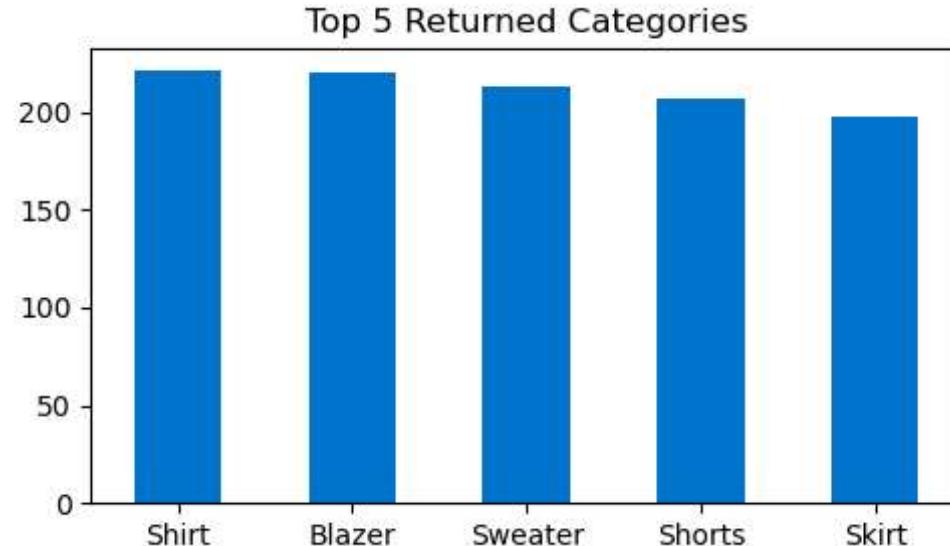


## Categories have the most returns

```
In [431]: print(returns_merged['Category'].value_counts().head(5))

plt.figure(figsize=(5, 3))
returns_merged['Category'].value_counts().head(5).plot(
    kind='bar',
    color="#0074CC"
)
plt.title('Top 5 Returned Categories')
plt.xlabel('')
plt.ylabel('')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

```
Category
Shirt      221
Blazer     220
Sweater    213
Shorts     207
Skirt      197
Name: count, dtype: int64
```



## How many people used coupons

```
In [387...]: orders_merged['Used_Coupon'] = orders_merged['Coupon'] != 'No Coupon'
orders_merged['Used_Coupon'].value_counts()
```

```
Out[387...]: Used_Coupon
False      5011
True       4989
Name: count, dtype: int64
```

## Average product rating

```
In [402...]: ratings['Prod_Rating'].mean()
```

```
Out[402...]: 2.9923
```

```
In [453...]: orders_products = orders.merge(products, on='P_ID')

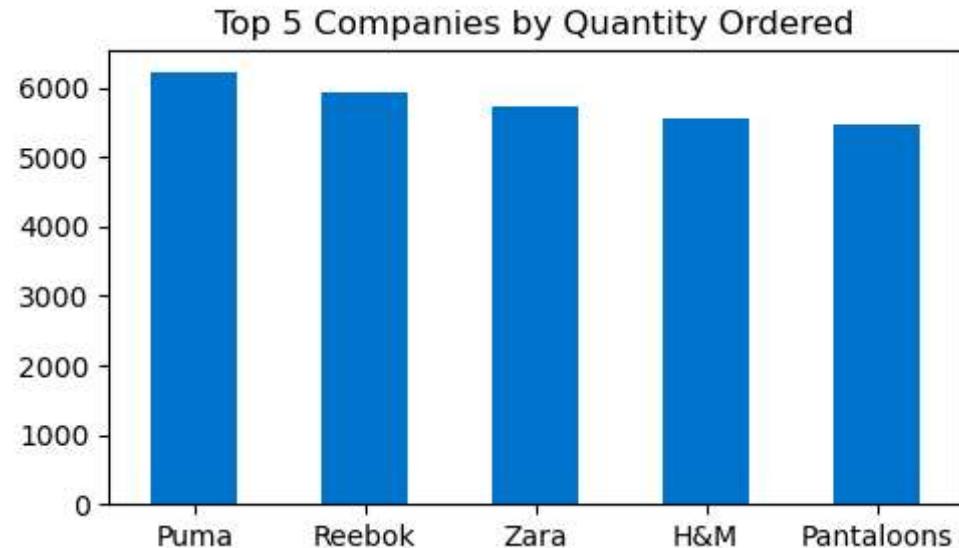
top_companies = orders_products.groupby('Company_Name')['Qty'].sum().sort_values(ascending=False).head(5)

print("Top Companies by Orders (Quantity Sold):")
print(top_companies)

plt.figure(figsize=(5, 3))
top_companies.plot(
    kind='bar',
    color='#0074CC',
    title='Top 5 Companies by Quantity Ordered'
)
plt.xlabel('')
plt.ylabel('')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

Top Companies by Orders (Quantity Sold):

```
Company_Name
Puma          6216
Reebok        5917
Zara          5723
H&M          5546
Pantaloons   5474
Name: Qty, dtype: int64
```



Insights 

Gender Split: Male 51%, Female 49%

Top Cities: Vadodara, Surat, Jaipur

Popular Categories: Blazers, Shorts, Jeans

Top Brands: Puma, Reebok, Zara

Payment Mode: Net Banking (20.6%)

**Common Return Reasons: Defective products, Wrong items**

**Avg. Order Value: ₹9,732 | Avg. Quantity/Order: 5.5**

**Ratings: Product : 2.99**

**Nearly 50% customers used coupons**

In [ ]: