

```
In [1]: import pandas as pd
a="E:/DA_Projects/jar/List_of_Orders_55FFC79CF8.csv"
b="E:/DA_Projects/jar/Sales_target_DD2E9B96A0.csv"
c="E:/DA_Projects/jar/Order_Details_19795F61CF.csv"
transactions = pd.read_csv(a)
products = pd.read_csv(b)
customers = pd.read_csv(c)

print(customers.head())
print(products.head())
print(transactions.head())
```

	Order ID	Amount	Profit	Quantity	Category	Sub-Category
0	B-25601	1275	-1148	7	Furniture	Bookcases
1	B-25601	66	-12	5	Clothing	Stole
2	B-25601	8	-2	3	Clothing	Hankerchief
3	B-25601	80	-56	4	Electronics	Electronic Games
4	B-25602	168	-111	2	Electronics	Phones

	Month of	Order Date	Category	Target
0		01-04-2018	Furniture	10400
1		01-05-2018	Furniture	10500
2		01-06-2018	Furniture	10600
3		01-07-2018	Furniture	10800
4		01-08-2018	Furniture	10900

	Order ID	Order Date	CustomerName	State	City
0	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad
1	B-25602	01-04-2018	Pearl	Maharashtra	Pune
2	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal
3	B-25604	03-04-2018	Divsha	Rajasthan	Jaipur
4	B-25605	05-04-2018	Kasheen	West Bengal	Kolkata

```
In [2]: file_path = a
df = pd.read_csv(file_path)

print("Basic Info:")
print(df.info())
```

```
Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Order ID        500 non-null   object
1   Order Date      500 non-null   object
2   CustomerName    500 non-null   object
3   State           500 non-null   object
4   City            500 non-null   object
dtypes: object(5)
memory usage: 19.7+ KB
None
```

```
In [4]: file_path = c
df = pd.read_csv(file_path)
```

```
print("Basic Info:")
print(df.info())
```

```
Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Order ID        1500 non-null   object
1   Amount          1500 non-null   int64
2   Profit          1500 non-null   int64
3   Quantity        1500 non-null   int64
4   Category        1500 non-null   object
5   Sub-Category    1500 non-null   object
dtypes: int64(3), object(3)
memory usage: 70.4+ KB
None
```

```
In [5]: file_path = a
df = pd.read_csv(file_path)
null_values = df.isnull().sum()
print("Null values in each column:")
print(null_values)

if null_values.any():
    print("\nThere are null values in the dataset.")
else:
    print("\nNo null values found in the dataset.")
```

```
Null values in each column:
Order ID      0
Order Date    0
CustomerName  0
State         0
City          0
dtype: int64
```

No null values found in the dataset.

```
In [6]: file_path = b
df = pd.read_csv(file_path)
null_values = df.isnull().sum()
print("Null values in each column:")
print(null_values)

if null_values.any():
    print("\nThere are null values in the dataset.")
else:
    print("\nNo null values found in the dataset.")
```

```
Null values in each column:
Month of Order Date  0
Category            0
Target              0
dtype: int64
```

No null values found in the dataset.

```
In [7]: file_path = c
df = pd.read_csv(file_path)
null_values = df.isnull().sum()
print("Null values in each column:")
print(null_values)

if null_values.any():
    print("\nThere are null values in the dataset.")
else:
    print("\nNo null values found in the dataset.")
```

Null values in each column:

Order ID	0
Amount	0
Profit	0
Quantity	0
Category	0
Sub-Category	0

dtype: int64

No null values found in the dataset.

```
In [8]: file_path = a
df = pd.read_csv(file_path)
duplicates = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicates}")
if duplicates > 0:
    print("\nDuplicate rows:")
    print(df[df.duplicated()])
else:
    print("\nNo duplicate rows found.")
```

Number of duplicate rows: 0

No duplicate rows found.

```
In [9]: file_path = b
df = pd.read_csv(file_path)
duplicates = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicates}")
if duplicates > 0:
    print("\nDuplicate rows:")
    print(df[df.duplicated()])
else:
    print("\nNo duplicate rows found.")
```

Number of duplicate rows: 0

No duplicate rows found.

```
In [10]: file_path = c
df = pd.read_csv(file_path)
duplicates = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicates}")
if duplicates > 0:
    print("\nDuplicate rows:")
    print(df[df.duplicated()])
else:
    print("\nNo duplicate rows found.")
```

Number of duplicate rows: 0

No duplicate rows found.

```
In [12]: #Part 1 : Sales and Profitability Analysis
transactions = pd.read_csv(c)

customers = pd.read_csv(a)
merged_df = pd.merge(transactions, customers, on="Order ID", how="left")
print(merged_df)
merged_df.to_csv("E:\DA_Projects\jar\mergerd_orders.csv", index=False)
```

	Order ID	Amount	Profit	Quantity	Category	Sub-Category \
0	B-25601	1275	-1148	7	Furniture	Bookcases
1	B-25601	66	-12	5	Clothing	Stole
2	B-25601	8	-2	3	Clothing	Hankerchief
3	B-25601	80	-56	4	Electronics	Electronic Games
4	B-25602	168	-111	2	Electronics	Phones
...	...	...	...	...	...	...
1495	B-26099	835	267	5	Electronics	Phones
1496	B-26099	2366	552	5	Clothing	Trousers
1497	B-26100	828	230	2	Furniture	Chairs
1498	B-26100	34	10	2	Clothing	T-shirt
1499	B-26100	72	16	2	Clothing	Shirt

	Order Date	CustomerName	State	City
0	01-04-2018	Bharat	Gujarat	Ahmedabad
1	01-04-2018	Bharat	Gujarat	Ahmedabad
2	01-04-2018	Bharat	Gujarat	Ahmedabad
3	01-04-2018	Bharat	Gujarat	Ahmedabad
4	01-04-2018	Pearl	Maharashtra	Pune
...	...	...	...	...
1495	30-03-2019	Bhishm	Maharashtra	Mumbai
1496	30-03-2019	Bhishm	Maharashtra	Mumbai
1497	31-03-2019	Hitika	Madhya Pradesh	Indore
1498	31-03-2019	Hitika	Madhya Pradesh	Indore
1499	31-03-2019	Hitika	Madhya Pradesh	Indore

[1500 rows x 10 columns]

```
In [43]: d="E:\DA_Projects\jar\mergerd_orders.csv"
```

```
In [44]: df = pd.read_csv(d)

print("Basic Info:")
print(df.info())
```

```
Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Order ID        1500 non-null   object
1   Amount          1500 non-null   int64
2   Profit          1500 non-null   int64
3   Quantity        1500 non-null   int64
4   Category        1500 non-null   object
5   Sub-Category    1500 non-null   object
6   Order Date      1500 non-null   object
7   CustomerName    1500 non-null   object
8   State           1500 non-null   object
9   City            1500 non-null   object
dtypes: int64(3), object(7)
memory usage: 117.3+ KB
None
```

```
In [4]: revenue_by_category=df.groupby("Category")["Amount"].sum()
print("Total Revenue by Category")
print(revenue_by_category)
```

```
Total Revenue by Category
Category
Clothing      139054
Electronics   165267
Furniture     127181
Name: Amount, dtype: int64
```

```
In [5]: revenue_by_sub_category=df.groupby("Sub-Category")["Amount"].sum()
print("Total Revenue by Sub-Category")
print(revenue_by_sub_category)
```

```
Total Revenue by Sub-Category
Sub-Category
Accessories      21728
Bookcases        56861
Chairs           34222
Electronic Games 39168
Furnishings      13484
Hankerchief     14608
Kurti            3361
Leggings         2106
Phones          46119
Printers         58252
Saree           53511
Shirt            7555
Skirt            1946
Stole            18546
T-shirt          7382
Tables           22614
Trousers         30039
Name: Amount, dtype: int64
```

```
In [6]: profit_category=df.groupby("Category")["Profit"].mean()
print("Average Profit by Category")
print(profit_category)
```

```
Average Profit by Category
Category
Clothing      11.762908
Electronics   34.071429
Furniture      9.456790
Name: Profit, dtype: float64
```

```
In [45]: profit_category=(df.groupby("Sub-Category")["Profit"].mean()).sort_values(ascending=False)
print("Average Profit by Sub-Category")
print(profit_category)
```

```
Average Profit by Sub-Category
Sub-Category
Printers      80.594595
Trousers      73.000000
Bookcases     61.873418
Accessories   49.430556
Phones        26.590361
T-shirt       19.480519
Shirt         16.391304
Stole         13.328125
Furnishings   11.561644
Hankerchief   10.595960
Chairs        7.797297
Leggings      4.905660
Kurti         3.851064
Skirt         3.671875
Saree         1.676190
Electronic Games -15.645570
Tables        -235.941176
Name: Profit, dtype: float64
```

```
In [8]: total_profit_category=df.groupby("Category")["Profit"].sum()
print("Total Profit by Category")
print(total_profit_category)
```

```
Total Profit by Category
Category
Clothing      11163
Electronics   10494
Furniture      2298
Name: Profit, dtype: int64
```

```
In [46]: total_profit_sub_category=(df.groupby("Sub-Category")["Profit"].sum()
        .sort_values(ascending=False))
print("Total Profit by Category")
print(total_profit_sub_category)
```

```
Total Profit by Category
Sub-Category
Printers          5964
Bookcases         4888
Accessories       3559
Trousers          2847
Stole             2559
Phones            2207
Hankerchief       2098
T-shirt           1500
Shirt             1131
Furnishings        844
Chairs            577
Saree             352
Leggings          260
Skirt             235
Kurti             181
Electronic Games -1236
Tables            -4011
Name: Profit, dtype: int64
```

```
In [56]: Profit_margin_by_category=((total_profit_category/revenue_by_category)*100).sort_valu
print("Profit_margin_by_category")
print(Profit_margin_by_category)
```

```
Profit_margin_by_category
Category
Clothing      8.027817
Electronics   6.349725
Furniture     1.806874
dtype: float64
```

```
In [58]: profitmargin_subcategory=((total_profit_sub_category / revenue_by_sub_category) * 100
print("Total Profit Margin (%) by Sub-Category")
print(profitmargin_subcategory)
```

```
Total Profit Margin (%) by Sub-Category
Sub-Category
T-shirt          20.319697
Accessories      16.379786
Shirt            14.970218
Hankerchief      14.361993
Stole            13.798124
Leggings         12.345679
Skirt            12.076053
Printers         10.238275
Trousers         9.477679
Bookcases        8.596402
Furnishings      6.259270
Kurti            5.385302
Phones           4.785446
Chairs           1.686050
Saree            0.657809
Electronic Games -3.155637
Tables          -17.736800
dtype: float64
```

In [48]: *#Part 2 : Target Achievement Analysis*

In [13]: `file_path = b  
df = pd.read_csv(file_path)`

```
print("Basic Info:")  
print(df.info())
```

```
Basic Info:  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 36 entries, 0 to 35  
Data columns (total 3 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   Month of Order Date    36 non-null     object  
1   Category               36 non-null     object  
2   Target                 36 non-null     int64  
dtypes: int64(1), object(2)  
memory usage: 992.0+ bytes  
None
```

In [14]: `df.head()`

Out[14]:

	Month of Order Date	Category	Target
0	01-04-2018	Furniture	10400
1	01-05-2018	Furniture	10500
2	01-06-2018	Furniture	10600
3	01-07-2018	Furniture	10800
4	01-08-2018	Furniture	10900

In [15]: `target_by_category=df.groupby("Category")["Target"].sum()  
print("Total Sales Target by Category")  
print(target_by_category)`

```
Total Sales Target by Category  
Category  
Clothing      174000  
Electronics   129000  
Furniture     132900  
Name: Target, dtype: int64
```



```
In [40]: x = df[df['Category'] == 'Furniture']
x['Month of Order Date'] = pd.to_datetime(x['Month of Order Date'], format='%d-%m-%Y')
m_sales = x.groupby(x['Month of Order Date'].dt.to_period('M'))['Target'].sum()
m_sales_pct = m_sales.pct_change() * 100
print(m_sales_pct)
```

```
Month of Order Date
2018-04      NaN
2018-05      0.961538
2018-06      0.952381
2018-07      1.886792
2018-08      0.925926
2018-09      0.917431
2018-10      0.909091
2018-11      1.801802
2018-12      0.884956
2019-01      0.877193
2019-02      0.869565
2019-03      1.724138
Freq: M, Name: Target, dtype: float64
```

C:\Users\mayan\AppData\Local\Temp\ipykernel\_17768\594903558.py:2: 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](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
x['Month of Order Date'] = pd.to_datetime(x['Month of Order Date'], format='%d-%m-%Y') # Adjust format if needed
```

```
In [87]: #Part 3 : Regional Performance Insights
```

```
In [92]: df=pd.read_csv(d)
top_sales=df.groupby('State')['Quantity'].sum().sort_values(ascending=False).head(5)
print(top_sales)
```

```
State
Madhya Pradesh      1360
Maharashtra         1056
Gujarat              328
Uttar Pradesh        288
Rajasthan            282
Name: Quantity, dtype: int64
```

```
In [106]: statewise_slaes=(df.groupby("State")["Amount"].sum().sort_values(ascending=False))  
print(statewise_slaes)
```

```
State  
Madhya Pradesh      105140  
Maharashtra         95348  
Delhi               22531  
Uttar Pradesh       22359  
Rajasthan           21149  
Gujarat             21058  
Punjab              16786  
Karnataka           15058  
West Bengal         14086  
Kerala              13459  
Andhra Pradesh      13256  
Bihar               12943  
Nagaland            11903  
Jammu and Kashmir   10829  
Haryana             8863  
Himachal Pradesh    8666  
Goa                  6705  
Tamil Nadu          6087  
Sikkim              5276  
Name: Amount, dtype: int64
```

```
In [107]: statewise_slaes=(df.groupby("State")["Profit"].mean().sort_values(ascending=False))  
print(statewise_slaes)
```

```
State  
Haryana              50.961538  
Uttar Pradesh        47.602941  
Kerala               41.577778  
Delhi                40.364865  
West Bengal          39.682540  
Himachal Pradesh     22.620690  
Maharashtra          21.296552  
Rajasthan            16.986486  
Sikkim               16.708333  
Madhya Pradesh       16.326471  
Karnataka            11.944444  
Goa                   8.604651  
Gujarat              5.344828  
Nagaland             3.288889  
Jammu and Kashmir    0.163265  
Bihar               -5.177419  
Punjab              -10.150000  
Andhra Pradesh      -11.809524  
Tamil Nadu          -88.640000  
Name: Profit, dtype: float64
```

```
In [108]: statewise_slaes=(df.groupby("City")["Profit"].mean().sort_values(ascending=False))  
print(statewise_slaes)
```

```
City  
Allahabad      102.700000  
Udaipur        67.000000  
Pune           54.686747  
Surat          53.800000  
Delhi          43.308642  
Thiruvananthapuram 41.577778  
Kolkata        39.682540  
Amritsar       36.266667  
Simla          22.620690  
Gangtok        16.708333  
Indore         15.576779  
Bhopal         13.196970  
Bangalore      11.944444  
Goa            8.604651  
Mumbai         7.908213  
Lucknow        4.105263  
Kohima         3.288889  
Chandigarh     2.422535  
Kashmir        0.163265  
Patna         -5.177419  
Hyderabad     -11.809524  
Ahmedabad     -14.193548  
Jaipur        -17.113636  
Chennai       -88.640000  
Name: Profit, dtype: float64
```