In [ ]:

```
```

In [337…

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

In [183…

```
df= pd.read_csv(r'D:\My projects\Othoba Data 2023-2024.csv')
```

```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\2162910310.py:1: DtypeWarning: Co
lumns (15,25) have mixed types. Specify dtype option on import or set low_memory=Fal
se.
  df= pd.read_csv(r'D:\My projects\Othoba Data 2023-2024.csv')
```

In [184…

```
df.head()
```

Out[184…

	SubOrderId	OrderShippingExclTax	ShipmentBy	ShippingStatus	OrderStatus	FinalOrde
0	1151770-1	0.0	Daily shopping	Delivered	Complete	
1	1151770-1	0.0	Daily shopping	Delivered	Complete	
2	1151769-1	0.0	Daily shopping	Delivered	Complete	
3	1151769-1	0.0	Daily shopping	Delivered	Complete	
4	1151768-1	0.0	Vision Emporium	Delivered	Complete	

5 rows × 41 columns

In [185…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 930318 entries, 0 to 930317
Data columns (total 41 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   SubOrderId            930318 non-null  object 
 1   OrderShippingExclTax  778266 non-null  float64
 2   ShipmentBy            930318 non-null  object 
 3   ShippingStatus        930318 non-null  object 
 4   OrderStatus           930318 non-null  object 
 5   FinalOrderStatus      930318 non-null  object 
 6   PaymentStatus         930318 non-null  object 
 7   Pre-ShipmentStatus    930318 non-null  object 
 8   Order Update Status   930318 non-null  object 
 9   DeliveredDate         930318 non-null  object 
 10  ShippingFullName      930318 non-null  object 
 11  ShippingDivision      930318 non-null  object 
 12  ShippingStateProvince 930318 non-null  object 
 13  Order Via             930318 non-null  object 
 14  ShippingPhoneNumber   930318 non-null  float64
 15  Mobile Number         930318 non-null  object 
 16  Order CreatedOn       930318 non-null  object 
 17  Month                 930318 non-null  object 
 18  Year                  930318 non-null  int64  
 19  Payment Method        930318 non-null  object 
 20  Order PaidOn          930318 non-null  object 
 21  Order Cancel Reason   817005 non-null  object 
 22  SL No                 930318 non-null  int64  
 23  Product Name          930318 non-null  object 
 24  Product Attribute     110925 non-null  object 
 25  AttributeItemCode     70 non-null      object 
 26  SKU                   930079 non-null  object 
 27  Vendor Group          930318 non-null  object 
 28  Delivery Channel Type 930315 non-null  object 
 29  Quantity              930318 non-null  int64  
 30  RegularPrice          930318 non-null  float64
 31  UnitSellingPrice      930318 non-null  float64
 32  TotalSellingPrice     930318 non-null  float64
 33  UnitCostPrice         930318 non-null  float64
 34  TotalCostPrice        930318 non-null  float64
 35  Commission            930318 non-null  float64
 36  Profit                930318 non-null  float64
 37  Profit Margin         864349 non-null  float64
 38  Sales Person          930318 non-null  object 
 39  Category              930318 non-null  object 
 40  B2C/B2B               930318 non-null  object 
dtypes: float64(10), int64(3), object(28)
memory usage: 291.0+ MB
```

In [9]:

```
df.count()
```

Out[9]:

```
SubOrderId                930318
OrderShippingExclTax      778266
ShipmentBy                930318
ShippingStatus            930318
OrderStatus               930318
FinalOrderStatus          930318
PaymentStatus             930318
Pre-ShipmentStatus        930318
Order Update Status       930318
DeliveredDate             930318
ShippingFullName          930318
ShippingDivision          930318
ShippingStateProvince     930318
Order Via                 930318
ShippingPhoneNumber       930318
Mobile Number             930318
Order CreatedOn           930318
Month                     930318
Year                      930318
Payment Method            930318
Order PaidOn              930318
Order Cancel Reason       817005
SL No                     930318
Product Name              930318
Product Attribute         110925
AttributeItemCode             70
SKU                       930079
Vendor Group              930318
Delivery Channel Type     930315
Quantity                  930318
RegularPrice              930318
UnitSellingPrice          930318
TotalSellingPrice         930318
UnitCostPrice             930318
TotalCostPrice            930318
Commission                930318
Profit                    930318
Profit Margin             864349
Sales Person              930318
Category                  930318
B2C/B2B                   930318
dtype: int64
```

In [189…

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

```
SubOrderId                    0
OrderShippingExclTax     152052
ShipmentBy                    0
ShippingStatus                0
OrderStatus                   0
FinalOrderStatus              0
PaymentStatus                 0
Pre-ShipmentStatus            0
Order Update Status           0
DeliveredDate                 0
ShippingFullName              0
ShippingDivision              0
ShippingStateProvince         0
Order Via                     0
ShippingPhoneNumber           0
Mobile Number                 0
Order CreatedOn               0
Month                         0
Year                          0
Payment Method                0
Order PaidOn                  0
Order Cancel Reason      113313
SL No                         0
Product Name                  0
Product Attribute        819393
AttributeItemCode        930248
SKU                         239
Vendor Group                  0
Delivery Channel Type         3
Quantity                      0
RegularPrice                  0
UnitSellingPrice              0
TotalSellingPrice             0
UnitCostPrice                 0
TotalCostPrice                0
Commission                    0
Profit                        0
Profit Margin             65969
Sales Person                  0
Category                      0
B2C/B2B                       0
dtype: int64
```

In [191…

```
####Droping Unnecessary Column Those have mostly Null values
df.drop(columns=['AttributeItemCode', 'Product Attribute','OrderShippingExclTax'],
```

In [193…

```
######Rename null columns on OCR which are Sales by Successful
df['Order Cancel Reason'].fillna('Successful', inplace = True)
```

```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\824469097.py:2: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assignm
ent using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because
the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  df['Order Cancel Reason'].fillna('Successful', inplace = True)
```

In [195…

```
####Changing Column object to  date time
df['Order CreatedOn'] = pd.to_datetime(df['Order CreatedOn'], errors= 'coerce')
df['DeliveredDate']= pd.to_datetime(df['DeliveredDate'], errors= 'coerce')
```

In [196…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 930318 entries, 0 to 930317
Data columns (total 38 columns):
 #   Column                 Non-Null Count   Dtype         
---  ------                 --------------   -----         
 0   SubOrderId             930318 non-null  object        
 1   ShipmentBy             930318 non-null  object        
 2   ShippingStatus         930318 non-null  object        
 3   OrderStatus            930318 non-null  object        
 4   FinalOrderStatus       930318 non-null  object        
 5   PaymentStatus          930318 non-null  object        
 6   Pre-ShipmentStatus     930318 non-null  object        
 7   Order Update Status    930318 non-null  object        
 8   DeliveredDate          930318 non-null  datetime64[ns]
 9   ShippingFullName       930318 non-null  object        
 10  ShippingDivision       930318 non-null  object        
 11  ShippingStateProvince  930318 non-null  object        
 12  Order Via              930318 non-null  object        
 13  ShippingPhoneNumber    930318 non-null  float64       
 14  Mobile Number          930318 non-null  object        
 15  Order CreatedOn        930318 non-null  datetime64[ns]
 16  Month                  930318 non-null  object        
 17  Year                   930318 non-null  int64         
 18  Payment Method         930318 non-null  object        
 19  Order PaidOn           930318 non-null  object        
 20  Order Cancel Reason    930318 non-null  object        
 21  SL No                  930318 non-null  int64         
 22  Product Name           930318 non-null  object        
 23  SKU                    930079 non-null  object        
 24  Vendor Group           930318 non-null  object        
 25  Delivery Channel Type  930315 non-null  object        
 26  Quantity               930318 non-null  int64         
 27  RegularPrice           930318 non-null  float64       
 28  UnitSellingPrice       930318 non-null  float64       
 29  TotalSellingPrice      930318 non-null  float64       
 30  UnitCostPrice          930318 non-null  float64       
 31  TotalCostPrice         930318 non-null  float64       
 32  Commission             930318 non-null  float64       
 33  Profit                 930318 non-null  float64       
 34  Profit Margin          864349 non-null  float64       
 35  Sales Person           930318 non-null  object        
 36  Category               930318 non-null  object        
 37  B2C/B2B                930318 non-null  object        
dtypes: datetime64[ns](2), float64(9), int64(3), object(24)
memory usage: 269.7+ MB
```

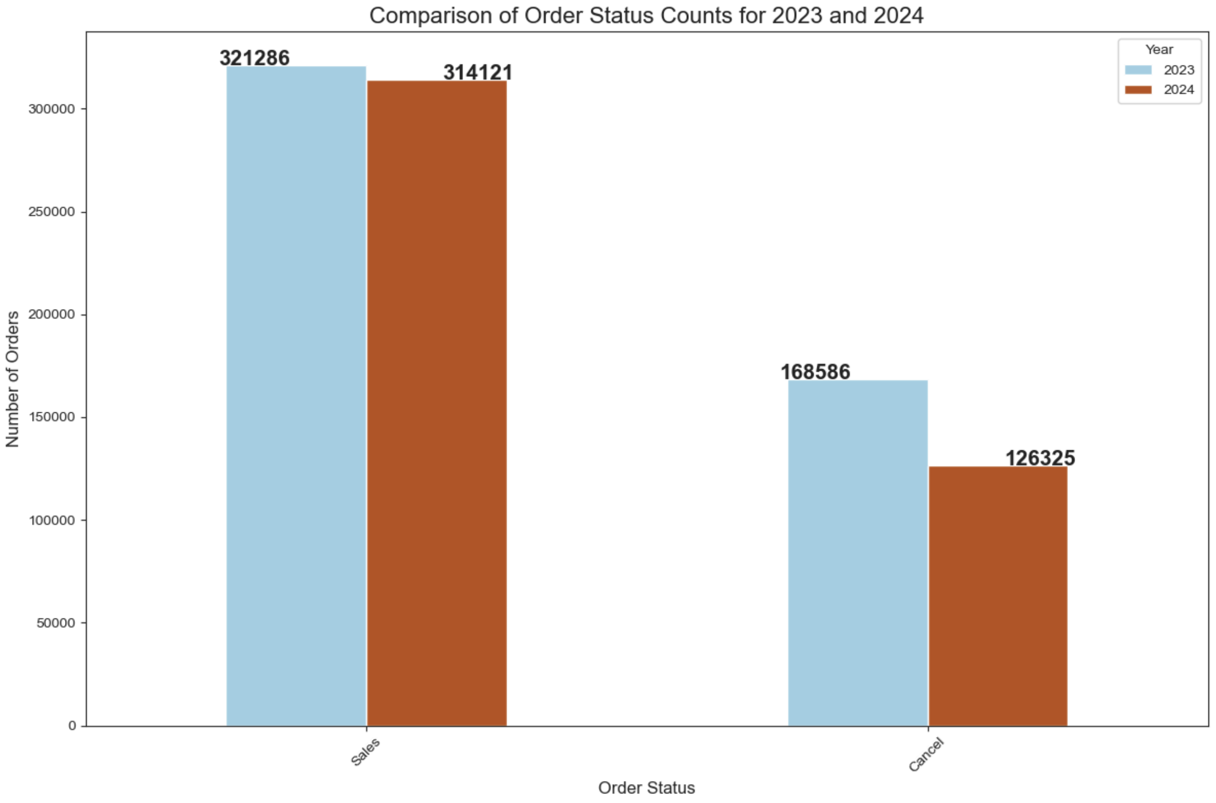
Explanatory 360° Analysis of Othoba's Monthly Sales for the Year 2023-2024

In [199…

```
###Total order Sale/Cancel

# Filter the DataFrame for the years 2023 and 2024
df_2023 = df[df['Year'] == 2023]
```

```
df_2024 = df[df['Year'] == 2024]

# Count the number of orders for each order status for 2023 and 2024
order_status_counts_2023 = df_2023['FinalOrderStatus'].value_counts()
order_status_counts_2024 = df_2024['FinalOrderStatus'].value_counts()

# Create a DataFrame for plotting
comparison_df = pd.DataFrame({
    '2023': order_status_counts_2023,
    '2024': order_status_counts_2024
}).fillna(0)

# Plotting the comparison bar graph
plt.figure(figsize=(12, 8))
comparison_df.plot(kind='bar', rot=45, figsize=(12, 8), colormap='Paired')
plt.title('Comparison of Order Status Counts for 2023 and 2024', fontsize=16)
plt.xlabel('Order Status', fontsize=12)
plt.ylabel('Number of Orders', fontsize=12)

plt.legend(title='Year')

# Annotate each bar with the exact count
for i in range(len(comparison_df)):
    for j in range(len(comparison_df.columns)):
        plt.text(i - 0.2 + j * 0.4, comparison_df.iloc[i, j] + 10, f'{int(compariso

plt.tight_layout()
plt.show()
```

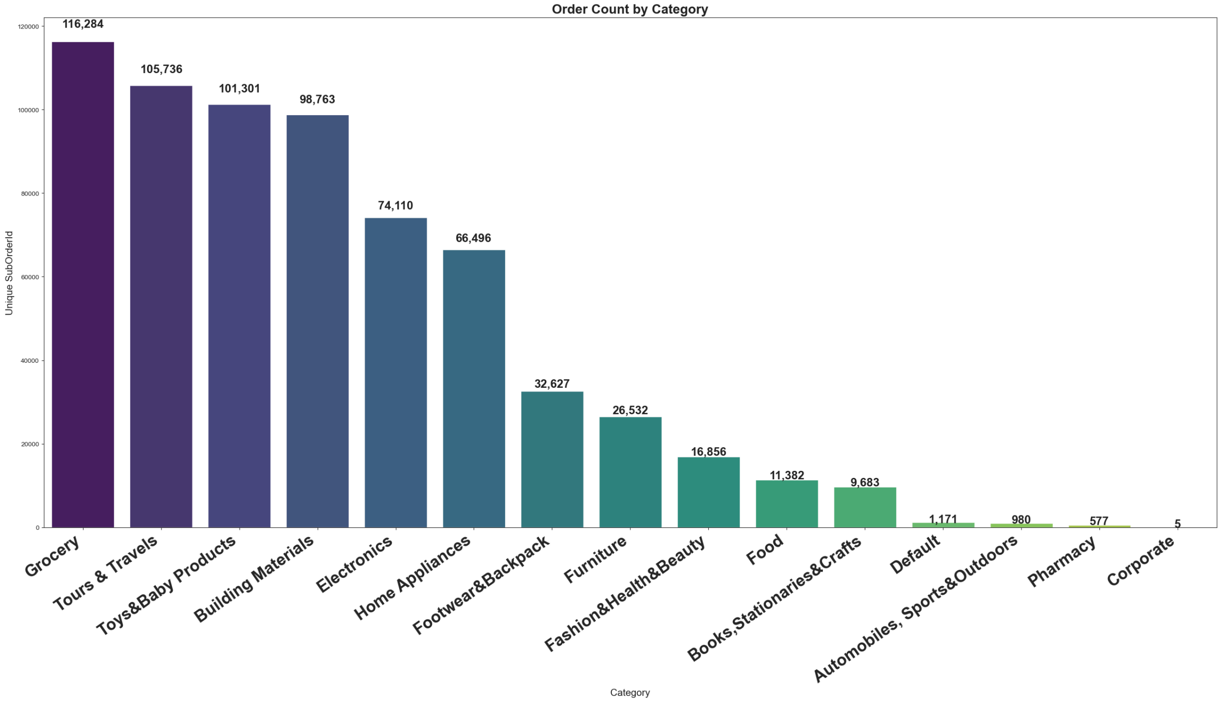
```
<Figure size 1200x800 with 0 Axes>
```



In [201…

```
# Count unique SubOrderId based on category
category_sales = df.groupby('Category')['SubOrderId'].nunique().sort_values(ascendi

# Set the figure size and style
plt.figure(figsize=(26, 15))
sns.set_style("ticks")

# Create a barplot with a vibrant color palette
colors = sns.color_palette("viridis", len(category_sales))
sns.barplot(x=category_sales.index, y=category_sales.values, palette=colors)

# Set the title and labels with larger font sizes for better readability
plt.title('Order Count by Category', fontsize=20, weight='bold')
plt.xlabel('Category', fontsize=15)
plt.ylabel('Unique SubOrderId', fontsize=15)

# Rotate x-axis labels for better clarity
plt.xticks(rotation=35, ha='right', fontsize=25, weight='bold')

# Adding value labels on top of each bar for more detailed information
for index, value in enumerate(category_sales.values):
    plt.text(index, value + (value * 0.03), f'{value:,.0f}', ha='center', fontsize=

# Remove the legend as it is not necessary in this context
plt.legend([], [], frameon=False)

# Show the plot
plt.tight_layout()
plt.show()
```

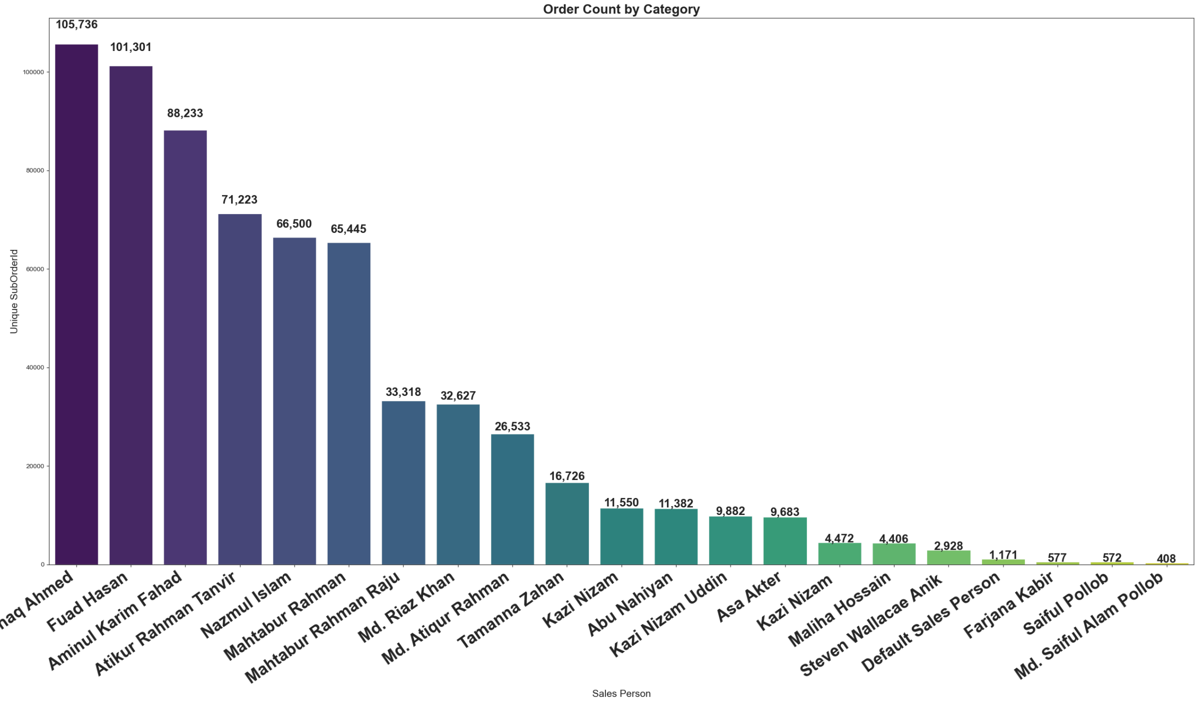
```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\2642553284.py:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=category_sales.index, y=category_sales.values, palette=colors)
```



In [203…

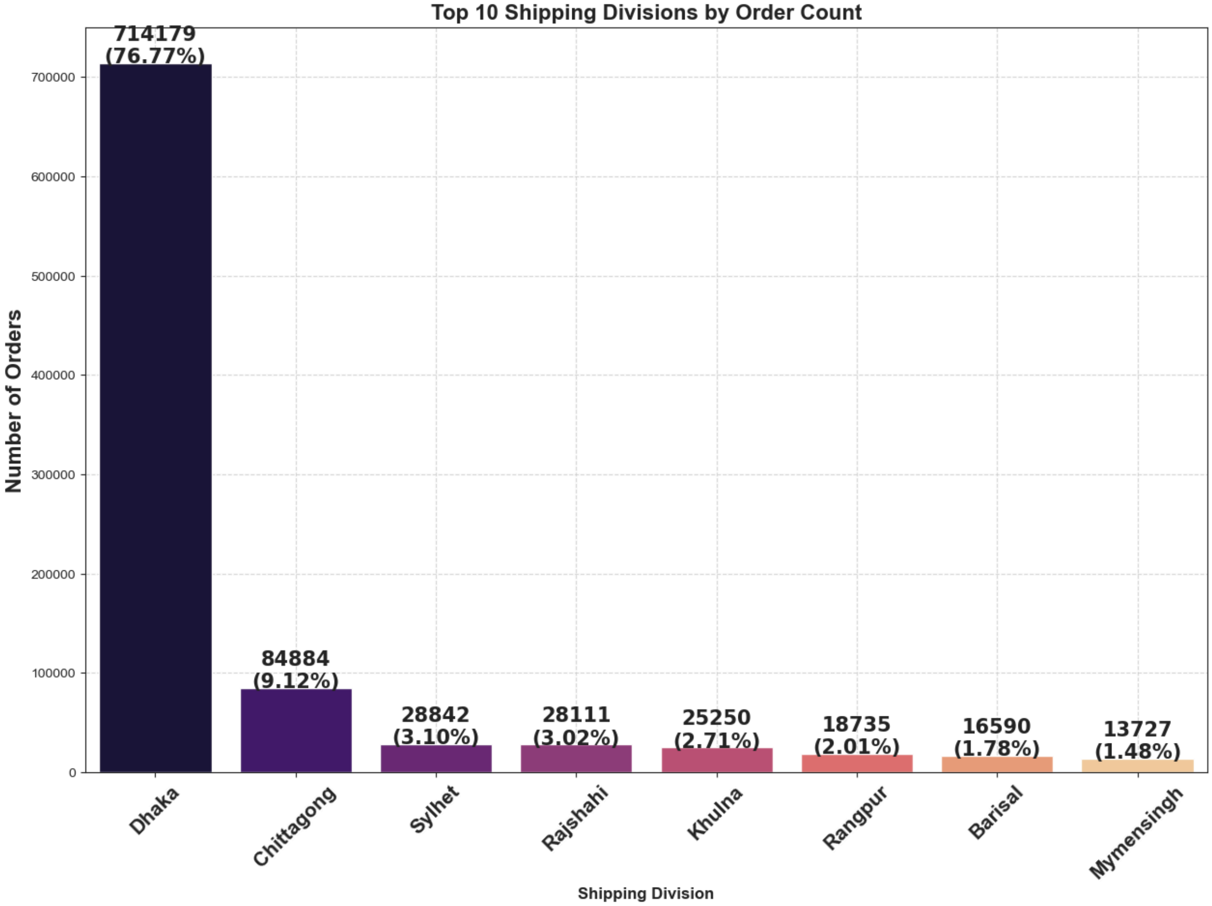
```
# Count unique SubOrderId based on category
category_sales = df.groupby('Sales Person')['SubOrderId'].nunique().sort_values(asc

# Set the figure size and style
plt.figure(figsize=(26, 15))
sns.set_style("ticks")

# Create a barplot with a vibrant color palette
colors = sns.color_palette("viridis", len(category_sales))
sns.barplot(x=category_sales.index, y=category_sales.values, palette=colors)

# Set the title and labels with larger font sizes for better readability
plt.title('Order Count by Category', fontsize=20, weight='bold')
plt.xlabel('Sales Person', fontsize=15)
plt.ylabel('Unique SubOrderId', fontsize=15)

# Rotate x-axis labels for better clarity
plt.xticks(rotation=35, ha='right', fontsize=25, weight='bold')

# Adding value labels on top of each bar for more detailed information
for index, value in enumerate(category_sales.values):
    plt.text(index, value + (value * 0.03), f'{value:,.0f}', ha='center', fontsize=

# Remove the legend as it is not necessary in this context
plt.legend([], [], frameon=False)

# Show the plot
plt.tight_layout()
plt.show()
```

```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\1217810342.py:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=category_sales.index, y=category_sales.values, palette=colors)
```



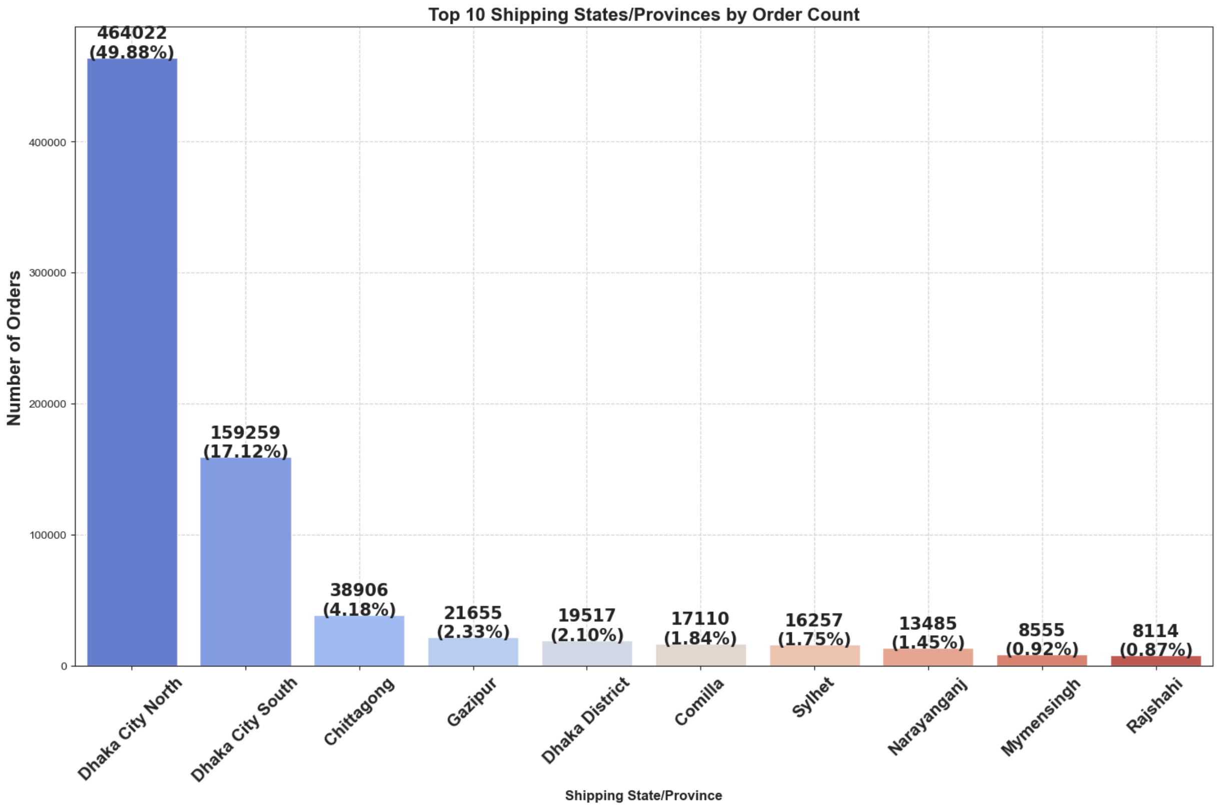
In [204…

```python
# Group by 'ShippingDivision' and count the number of orders in each division
division_order_counts = df['ShippingDivision'].value_counts()

# Calculate the total number of orders for percentage calculation
total_orders_division = division_order_counts.sum()



# Visualization of the top 10 shipping divisions by order count
plt.figure(figsize=(15, 10))
sns.barplot(x=division_order_counts.head(10).index, y=division_order_counts.head(10
plt.title('Top 10 Shipping Divisions by Order Count', fontsize=16, fontweight='bold
plt.xlabel('Shipping Division', fontsize=12, fontweight='bold')
plt.ylabel('Number of Orders', fontsize=16, fontweight='bold')
plt.xticks(rotation=45, fontsize=15, fontweight='bold')
plt.grid(True, linestyle='--', alpha=0.7)

# Annotate each bar with the exact value and percentage in bold
for i, v in enumerate(division_order_counts.head(10).values):
    percentage = (v / total_orders_division) * 100
    plt.text(i, v + 10, f'{v}\n({percentage:.2f}%)', ha='center', fontsize=15, font
```



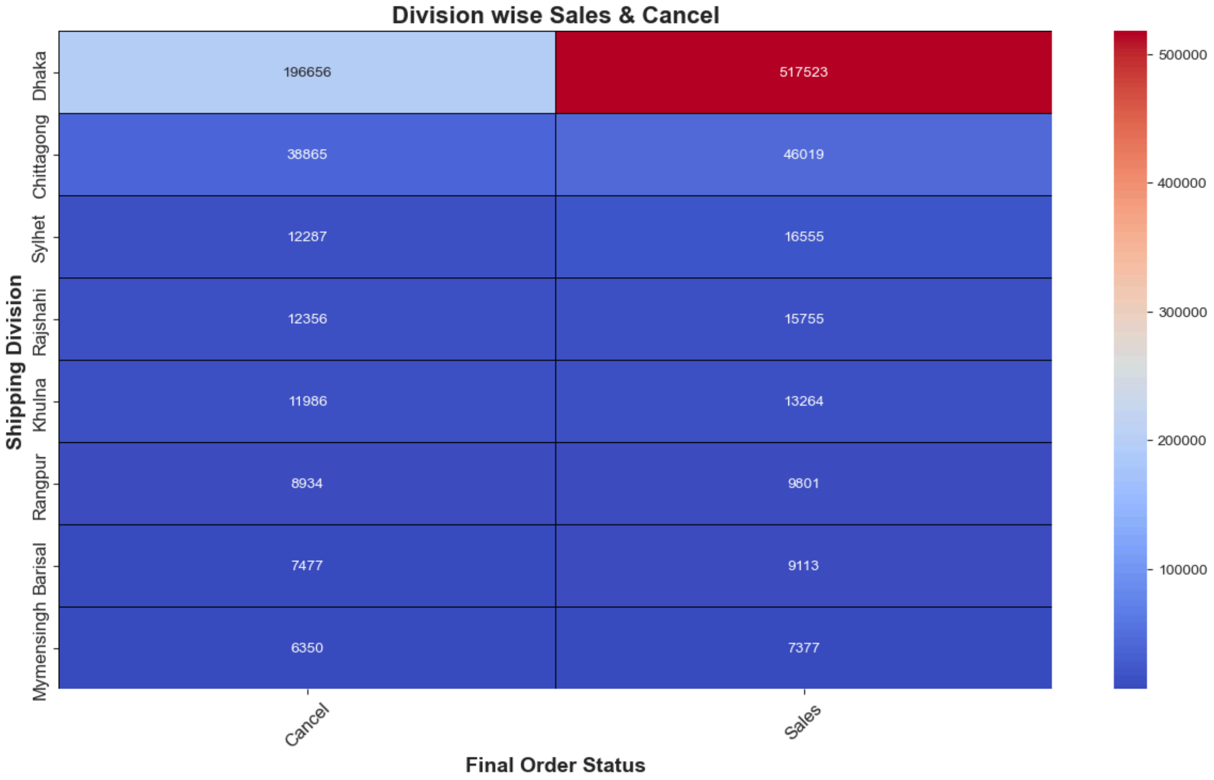
In [206…

```python
state_order_counts = df['ShippingStateProvince'].value_counts()
total_orders_state = state_order_counts.sum()
# Visualization of the top 10 shipping states/provinces by order count
plt.figure(figsize=(15, 10))
sns.barplot(x=state_order_counts.head(10).index, y=state_order_counts.head(10).valu
plt.title('Top 10 Shipping States/Provinces by Order Count', fontsize=16, fontweigh
plt.xlabel('Shipping State/Province', fontsize=12, fontweight='bold')
plt.ylabel('Number of Orders', fontsize=16, fontweight='bold')
plt.xticks(rotation=45, fontsize=15, fontweight='bold')
plt.grid(True, linestyle='--', alpha=0.7)

# Annotate each bar with the exact value and percentage in bold
for i, v in enumerate(state_order_counts.head(10).values):
    percentage = (v / total_orders_state) * 100
    plt.text(i, v + 10, f'{v}\n({percentage:.2f}%)', ha='center', fontsize=15, font

plt.tight_layout()
plt.show()
```



In [209…

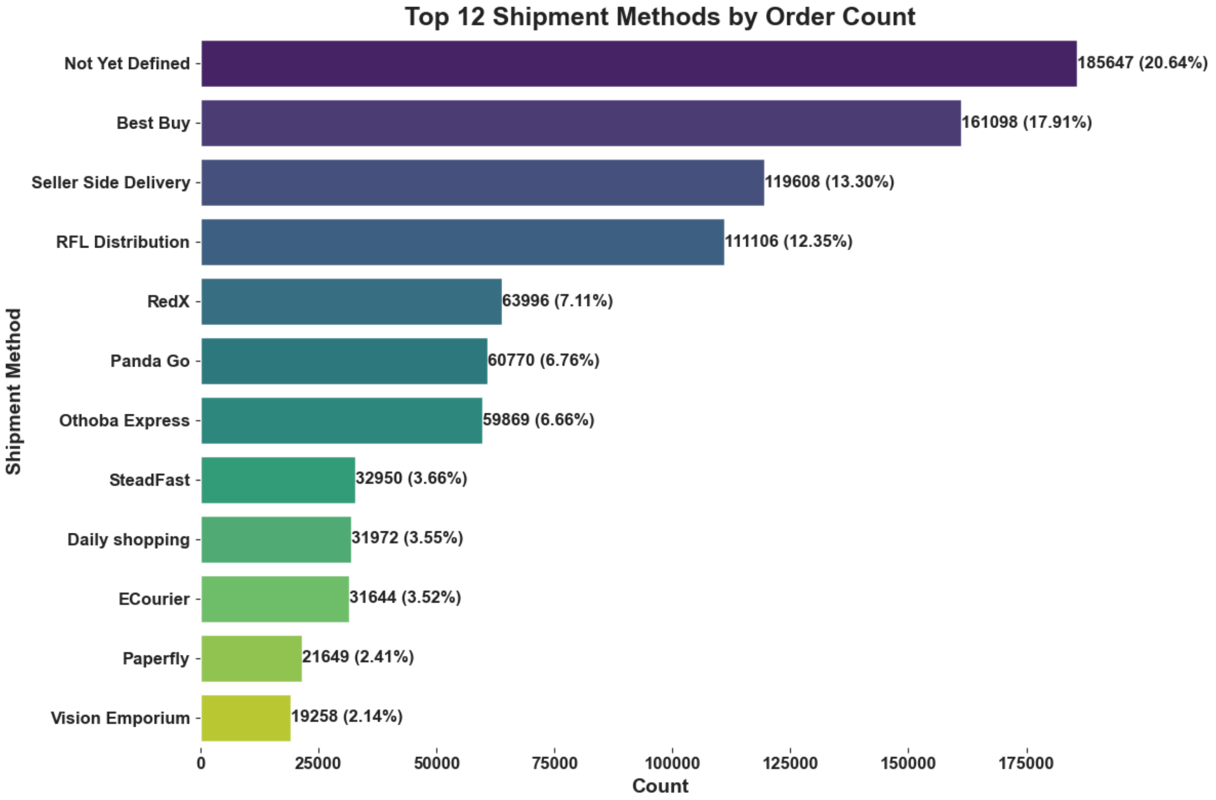
```
# Group by 'ShippingDivision' and 'FinalOrderStatus' and count orders
division_order_counts = df.groupby(['ShippingDivision', 'FinalOrderStatus']).size()

# Select the top 10 divisions by total order count
top_10_divisions = division_order_counts.sum(axis=1).nlargest(10).index
division_order_counts_top10 = division_order_counts.loc[top_10_divisions]

# Plot heatmap
plt.figure(figsize=(15, 8))
sns.heatmap(division_order_counts_top10, annot=True, fmt='d', cmap='coolwarm', line

plt.title('Division wise Sales & Cancel', fontsize=16, fontweight='bold')
plt.xlabel('Final Order Status', fontsize=14, fontweight='bold')
plt.ylabel('Shipping Division', fontsize=14, fontweight='bold')

plt.xticks(rotation=45, fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```



In [211…

```
shipment_method_counts = df['ShipmentBy'].value_counts().head(12)

# Calculate the percentages
shipment_method_percentages = (shipment_method_counts / shipment_method_counts.sum(

# Plot the count of orders by shipment method with percentages
plt.figure(figsize=(12, 8))
ax = sns.barplot(y=shipment_method_counts.index, x=shipment_method_counts.values, p

# Add annotations
for i, count in enumerate(shipment_method_counts):
    percentage = shipment_method_percentages[i]
    ax.text(count, i, f'{count} ({percentage:.2f}%)', ha='left', va='center', fonts

# Style adjustments
plt.title('Top 12 Shipment Methods by Order Count', fontsize=18, fontweight='bold')
plt.xlabel('Count', fontsize=14, fontweight='bold')
plt.ylabel('Shipment Method', fontsize=14, fontweight='bold')
plt.xticks(fontsize=12, fontweight='bold')
plt.yticks(fontsize=12, fontweight='bold')
sns.despine(left=True, bottom=True)

plt.tight_layout()
plt.show()
```

```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\218957076.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.barplot(y=shipment_method_counts.index, x=shipment_method_counts.values,
palette='viridis')
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\218957076.py:12: FutureWarning: S
eries.__getitem__ treating keys as positions is deprecated. In a future version, int
eger keys will always be treated as labels (consistent with DataFrame behavior). To
access a value by position, use `ser.iloc[pos]`
  percentage = shipment_method_percentages[i]
```



In [ ]:

In [ ]:

In [ ]:

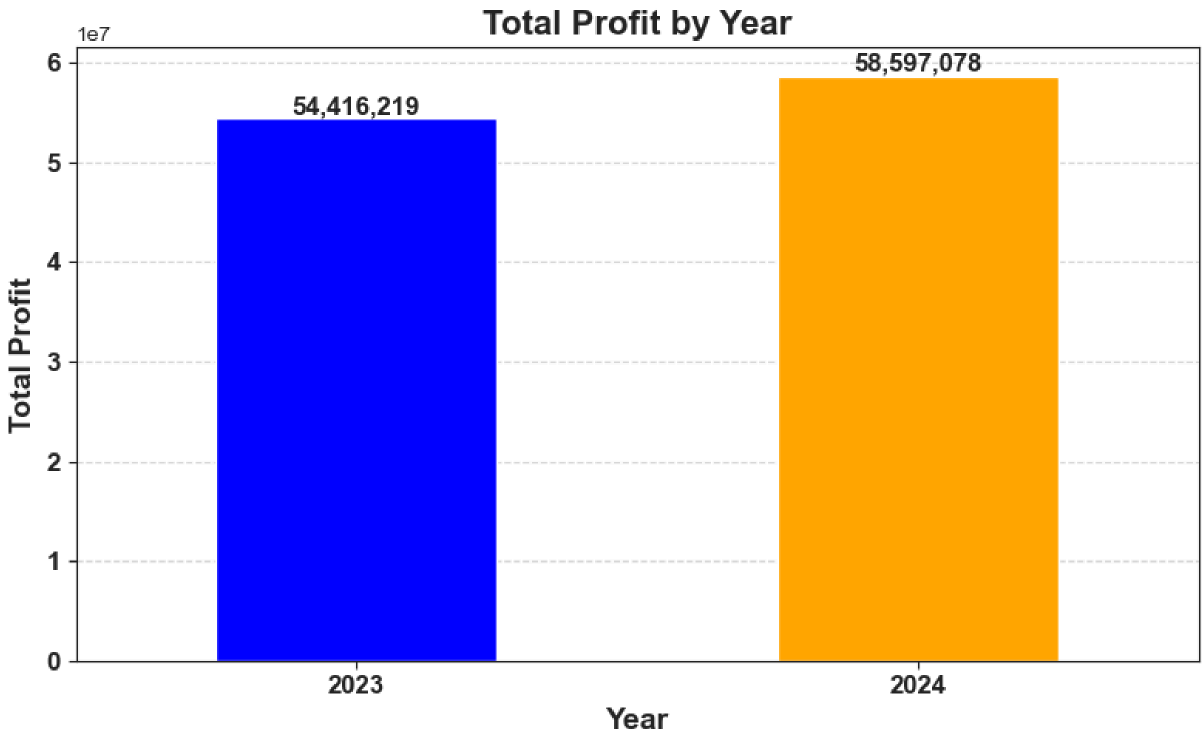
In [ ]:

Revenue Analysis

In [224…

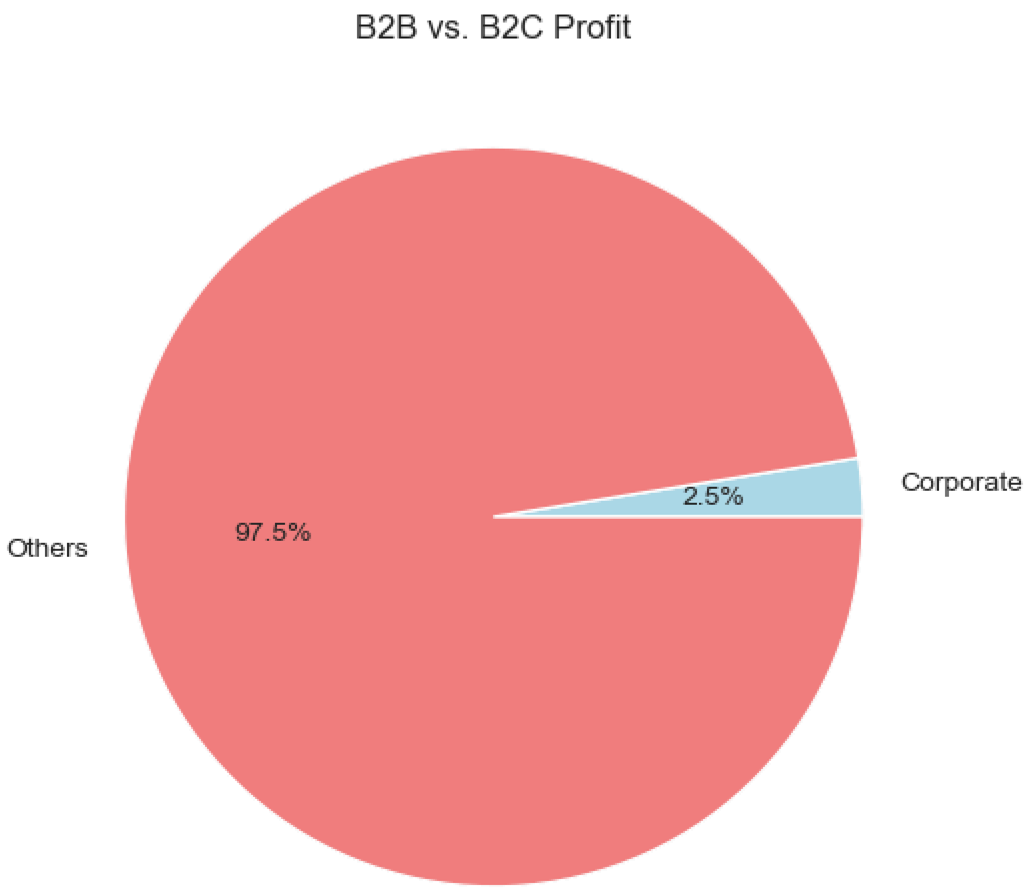
```
# Profitability Analysis
profit_by_year = df.groupby('Year')['Profit'].sum()

# Plotting the bar chart
plt.figure(figsize=(8, 5))
ax = profit_by_year.plot(kind='bar', color=['blue', 'orange'])
```

```
# Adding value labels on top of each bar
for i, value in enumerate(profit_by_year):
    ax.text(i, value + (value * 0.01), f'{value:,.0f}', ha='center', fontsize=12, f

plt.title("Total Profit by Year", fontsize=16, fontweight='bold')
plt.xlabel("Year", fontsize=14, fontweight='bold')
plt.ylabel("Total Profit", fontsize=14, fontweight='bold')
plt.xticks(rotation=0, fontsize=12, fontweight='bold')
plt.yticks(fontsize=12, fontweight='bold')
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()
plt.show()
```



In [232…

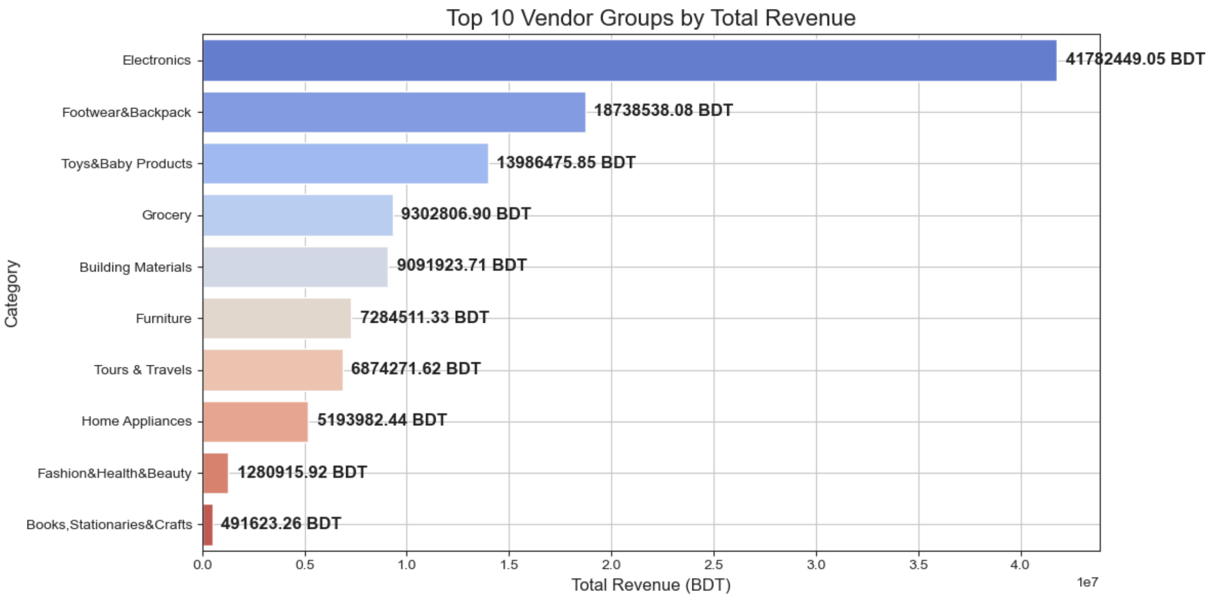
```
# B2B vs. B2C Profitability
b2b_b2c_profit = df.groupby('B2C/B2B')['Profit'].sum()
b2b_b2c_profit.plot(kind='pie', autopct='%1.1f%%', figsize=(6, 6), colors=['lightbl
plt.title("B2B vs. B2C Profit")
plt.ylabel("")
plt.show()
```



In [250…

```
top_vendors = df.groupby('Category').agg({'Profit': 'sum'}).reset_index()
top_vendors = top_vendors.sort_values(by='Profit', ascending=False).head(10)
# Visualization: Bar chart of top 10 vendors by total revenue with bold, clear numb
plt.figure(figsize=(12, 6))
sns.barplot(x='Profit', y='Category', data=top_vendors, palette='coolwarm')
plt.title('Top 10 Category by Total Profit', fontsize=16)
plt.xlabel('Total Revenue (BDT)', fontsize=12)
plt.ylabel('Category', fontsize=12)
plt.grid(True)

# Annotate each bar with bold, clear revenue numbers
for i, v in enumerate(top_vendors['Profit']):
    plt.text(v + 0.01 * max(top_vendors['Profit']), i, f'{v:.2f} BDT', ha='left', v

plt.tight_layout()
plt.show()
```

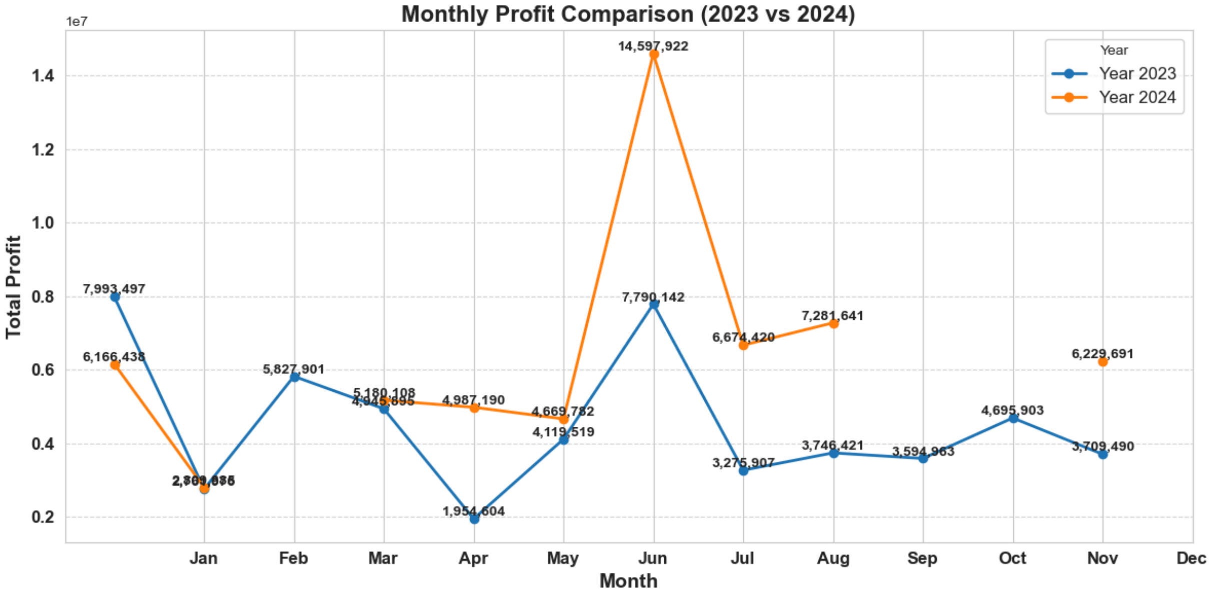
```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\874579967.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Profit', y='Category', data=top_vendors, palette='coolwarm')
```



In [252…

```
# Filter data for 2023 and 2024
df_filtered = df[df['Year'].isin([2023, 2024])]

# Group by Year and Month, then sum Profit
profit_by_month = df_filtered.groupby(['Year', 'Month'])['Profit'].sum().unstack()

# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Plot line chart for both years
for year in profit_by_month.index:
    plt.plot(profit_by_month.columns, profit_by_month.loc[year], marker='o', label=

# Adding value labels on each point
for year in profit_by_month.index:
    for month, value in enumerate(profit_by_month.loc[year]):
        plt.text(month, value, f'{value:,.0f}', ha='center', fontsize=10, fontweigh

# Titles and labels
plt.title("Monthly Profit Comparison (2023 vs 2024)", fontsize=16, fontweight='bold
plt.xlabel("Month", fontsize=14, fontweight='bold')
plt.ylabel("Total Profit", fontsize=14, fontweight='bold')
plt.xticks(ticks=range(1, 13), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'J
plt.yticks(fontsize=12, fontweight='bold')

# Show legend
plt.legend(title="Year", fontsize=12)

# Grid for readability
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()
plt.show()
```

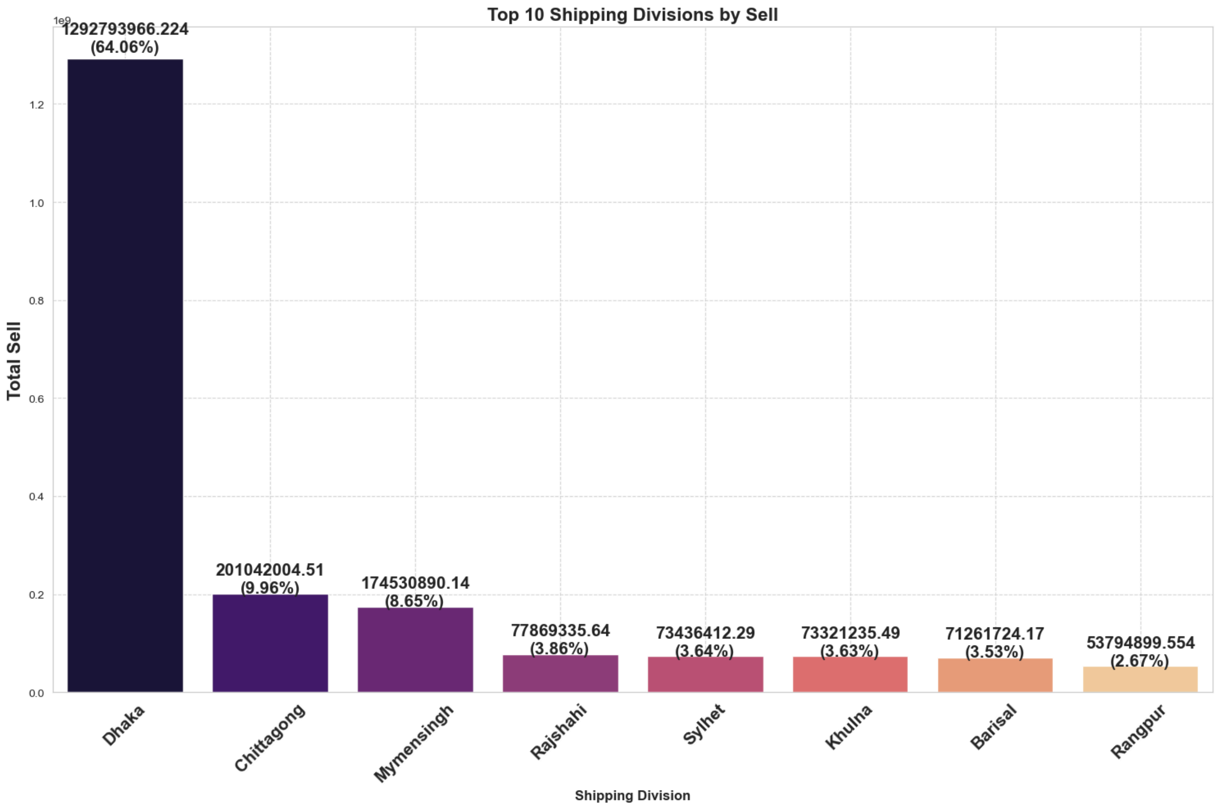
```
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
```



In [256…

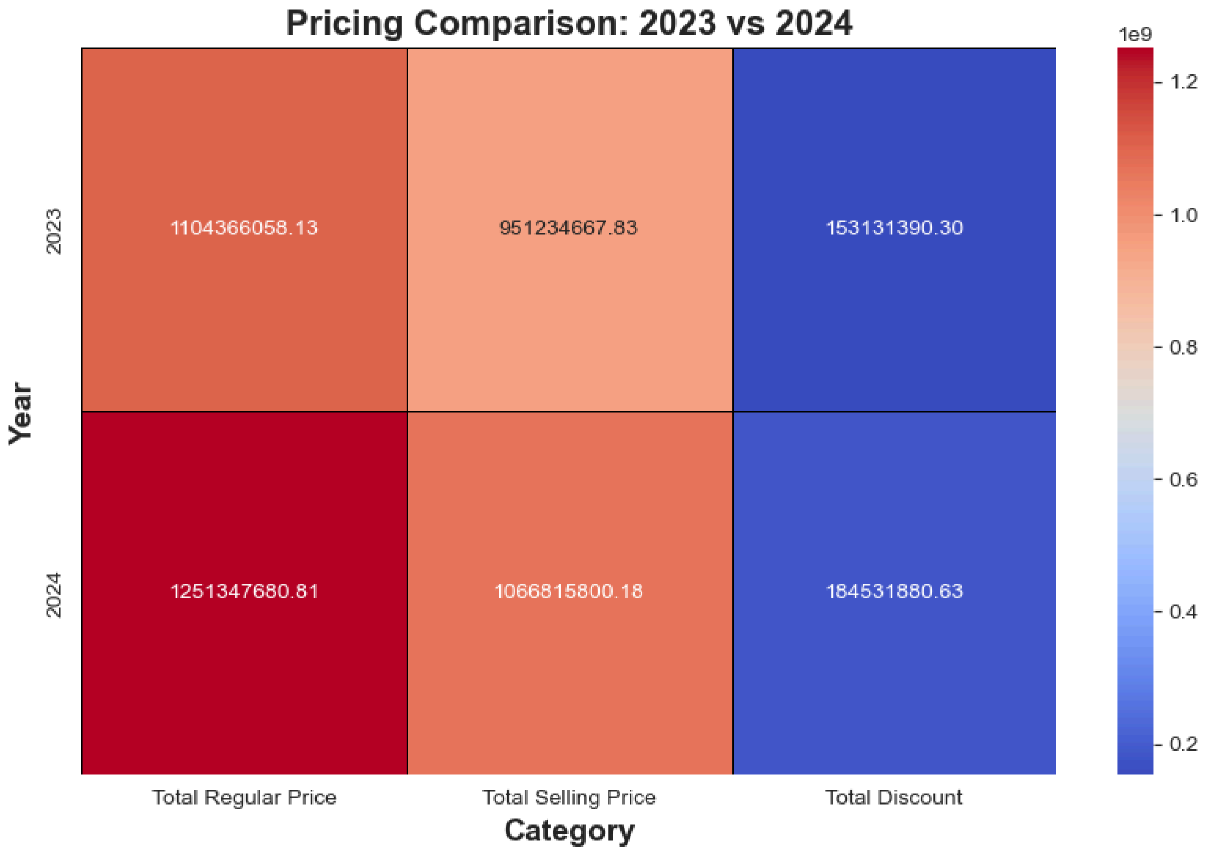
```python
# Group by 'ShippingDivision' and calculate the total profit for each division
division_profit = df.groupby('ShippingDivision')['TotalSellingPrice'].sum().sort_va

# Calculate the total profit for percentage calculation
total_profit_division = division_profit.sum()

# Visualization of the top 10 shipping divisions by total profit
plt.figure(figsize=(15, 10))
sns.barplot(x=division_profit.head(10).index, y=division_profit.head(10).values, pa
plt.title('Top 10 Shipping Divisions by Sell', fontsize=16, fontweight='bold')
plt.xlabel('Shipping Division', fontsize=12, fontweight='bold')
plt.ylabel('Total Sell', fontsize=16, fontweight='bold')
plt.xticks(rotation=45, fontsize=15, fontweight='bold')
plt.grid(True, linestyle='--', alpha=0.7)

# Annotate each bar with the exact value and percentage in bold
for i, v in enumerate(division_profit.head(10).values):
    percentage = (v / total_profit_division) * 100
    plt.text(i, v + (v * 0.01), f'{v}\n({percentage:.2f}%)', ha='center', fontsize=

plt.tight_layout()
plt.show()
```

```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\1819751983.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=division_profit.head(10).index, y=division_profit.head(10).values, p
alette='magma')
```



In [270…

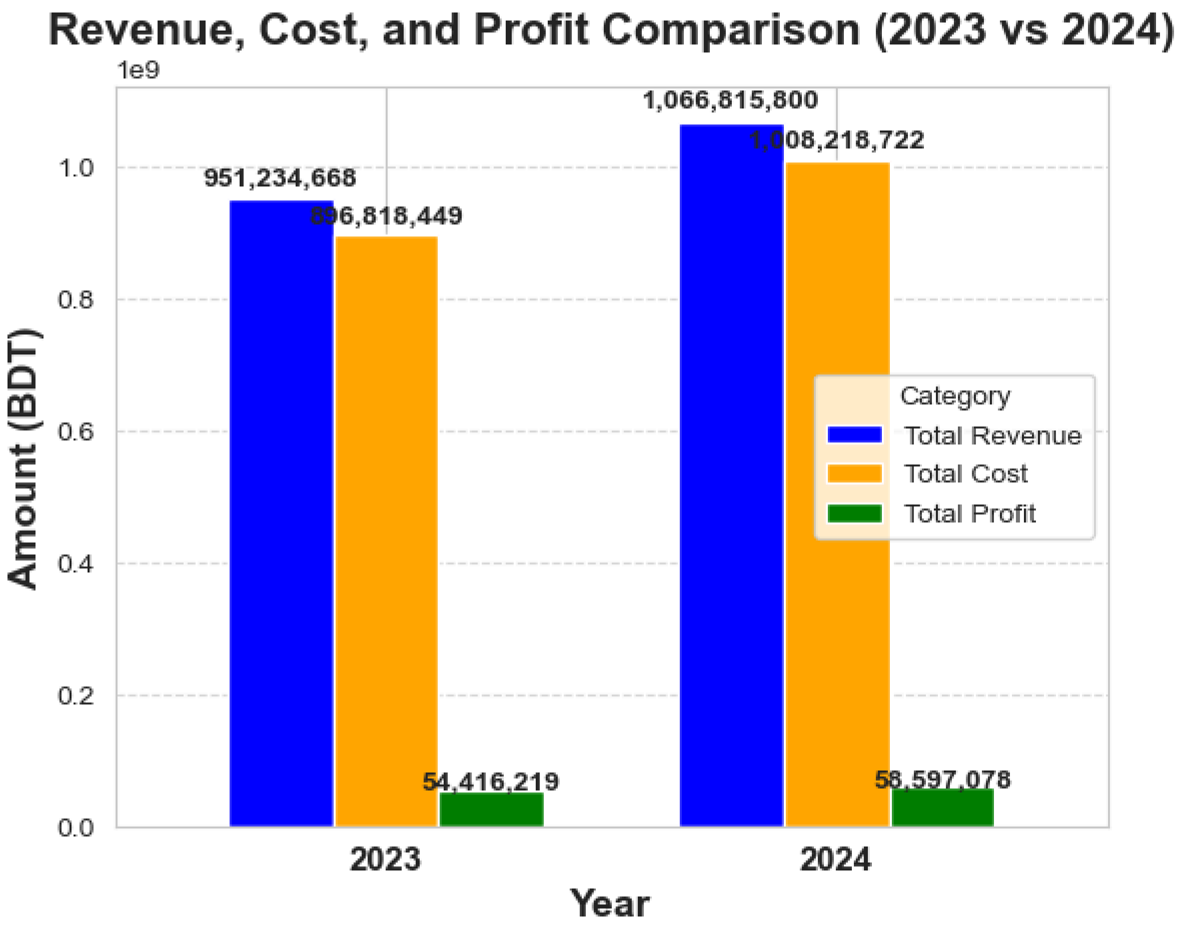
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Create DataFrame for visualization
pricing_summary_data = pd.DataFrame({
    'Year': ['2023', '2024'],
    'Total Regular Price': [total_regular_price_2023, total_regular_price_2024],
    'Total Selling Price': [total_selling_price_2023, total_selling_price_2024],
    'Total Discount': [total_discount_2023, total_discount_2024]
}).set_index('Year')

# Set figure size and style
plt.figure(figsize=(10, 6))
sns.set_style("whitegrid")

# Create heatmap
sns.heatmap(pricing_summary_data, annot=True, fmt=".2f", cmap="coolwarm", linewidth

# Title and labels
plt.title("Pricing Comparison: 2023 vs 2024", fontsize=16, fontweight='bold')
plt.xlabel("Category", fontsize=14, fontweight='bold')
plt.ylabel("Year", fontsize=14, fontweight='bold')

# Show the plot
plt.show()
```



In [287…

```
# Filter data for 2023 and 2024
df_2023 = df[df['Year'] == 2023]
df_2024 = df[df['Year'] == 2024]

# Calculate yearly totals
yearly_summary = pd.DataFrame({
    'Year': ['2023', '2024'],
    'Total Revenue': [df_2023['TotalSellingPrice'].sum(), df_2024['TotalSellingPric
    'Total Cost': [df_2023['TotalCostPrice'].sum(), df_2024['TotalCostPrice'].sum()
    'Total Profit': [df_2023['Profit'].sum(), df_2024['Profit'].sum()]
}).set_index('Year')

# Set figure size
plt.figure(figsize=(18, 9))
yearly_summary.plot(kind='bar', color=['blue', 'orange', 'green'], width=0.7)

# Set labels and title
plt.title('Revenue, Cost, and Profit Comparison (2023 vs 2024)', fontsize=16, fontw
plt.xlabel('Year', fontsize=14, fontweight='bold')
plt.ylabel('Amount (BDT)', fontsize=14, fontweight='bold')
plt.xticks(rotation=0, fontsize=12, fontweight='bold')
plt.legend(title="Category")

# Annotate bars with values
for i, bar in enumerate(plt.gca().patches):
    height = bar.get_height()
    plt.gca().text(bar.get_x() + bar.get_width()/2, height + 0.02*height, f'{height
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

```
<Figure size 1800x900 with 0 Axes>
```



In [ ]:

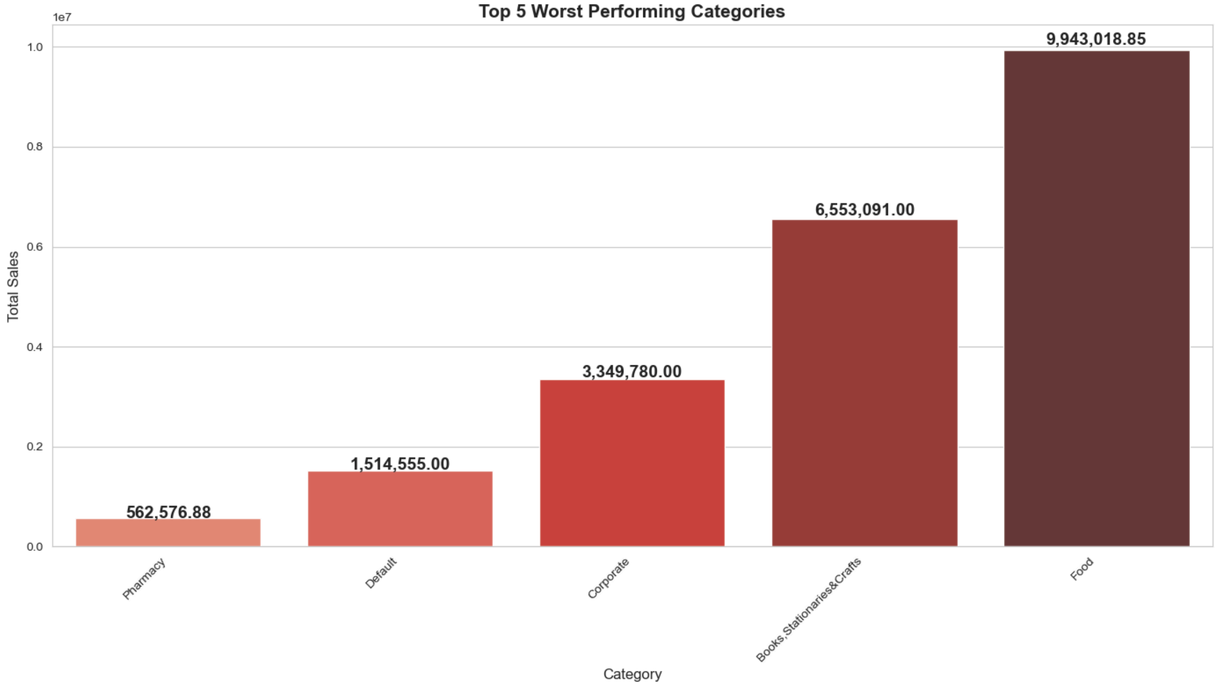
In [ ]:

In [321…

```
# Group by 'CategoryName' and sum 'TotalAmount'
sales_by_category = df.groupby('Category')['TotalSellingPrice'].sum().reset_index()

# Sort the categories by total sales in ascending order to get the worst performers
sales_by_category = sales_by_category.sort_values(by='TotalSellingPrice', ascending

# Display the top 5 worst-performing categories with formatted values
print("Top 5 Worst Performing Categories:")
print(sales_by_category.head(5).apply(lambda row: f"{row['Category']} : {row['Total

# Visualize the worst performers
plt.figure(figsize=(14, 8))
bottom_categories = sales_by_category.head(5)

# Plotting worst performers
sns.barplot(x=bottom_categories['Category'], y=bottom_categories['TotalSellingPrice
plt.title('Top 5 Worst Performing Categories', fontsize=15, weight='bold')
plt.xlabel('Category', fontsize=12)
plt.ylabel('Total Sales', fontsize=12)
plt.xticks(rotation=45, ha='right')

# Adding value labels on top of each bar
for index, value in enumerate(bottom_categories['TotalSellingPrice']):
    plt.text(index, value + (value * 0.01), f'{value:,.2f}', ha='center', fontsize=

plt.tight_layout()
plt.show()
```

```
Top 5 Worst Performing Categories:
12                        Pharmacy : 562,576.88
4                          Default : 1,514,555.00
3                        Corporate : 3,349,780.00
1      Books,Stationaries&Crafts : 6,553,091.00
7                             Food : 9,943,018.85
dtype: object
```

```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\134580547.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=bottom_categories['Category'], y=bottom_categories['TotalSellingPric
e'], palette='Reds_d')
```

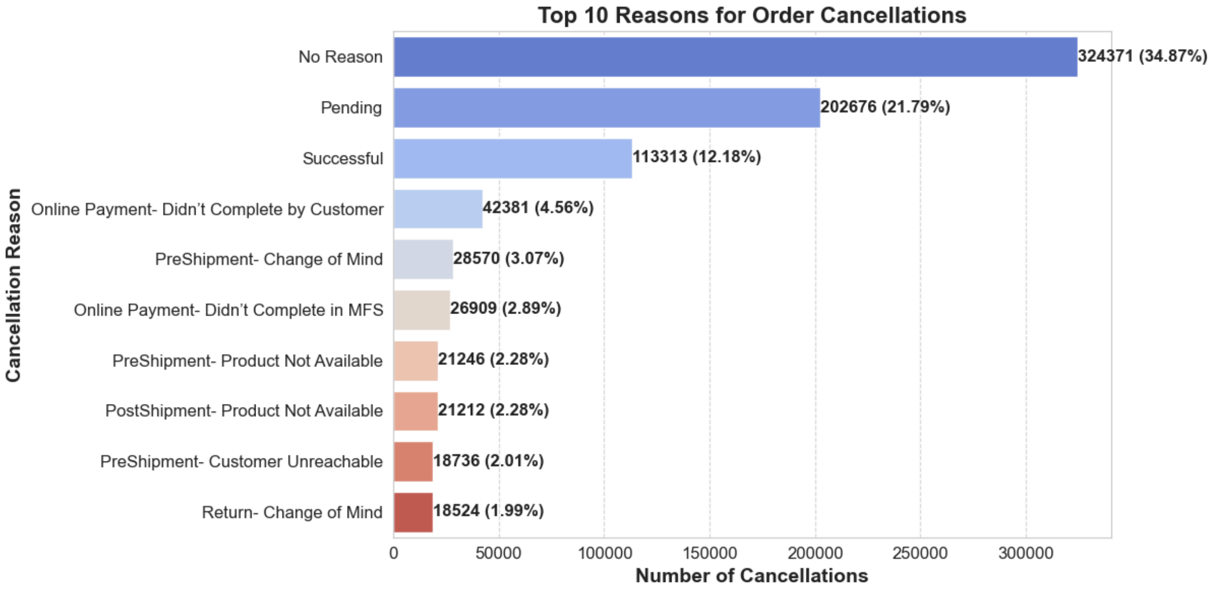


In [323…

```
# Count the number of cancellations for each reason
cancel_reason_counts = df['Order Cancel Reason'].value_counts()

# Calculate total number of cancellations
total_cancellations = cancel_reason_counts.sum()

# Select the top 10 reasons
top_10_cancellations = cancel_reason_counts.head(10)

# Calculate percentage for each reason
top_10_percentages = (top_10_cancellations / total_cancellations) * 100
```

```
# Visualization: Horizontal Bar Chart
plt.figure(figsize=(12, 6))
sns.barplot(y=top_10_cancellations.index, x=top_10_cancellations.values, palette='c

# Title and labels
plt.title('Top 10 Reasons for Order Cancellations', fontsize=16, fontweight='bold')
plt.xlabel('Number of Cancellations', fontsize=14, fontweight='bold')
plt.ylabel('Cancellation Reason', fontsize=14, fontweight='bold')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.7)

# Annotate each bar with the count and percentage
for i, (v, p) in enumerate(zip(top_10_cancellations.values, top_10_percentages.valu
    plt.text(v + 0.5, i, f'{v} ({p:.2f}%)', va='center', fontsize=12, fontweight='b

plt.tight_layout()
plt.show()
```

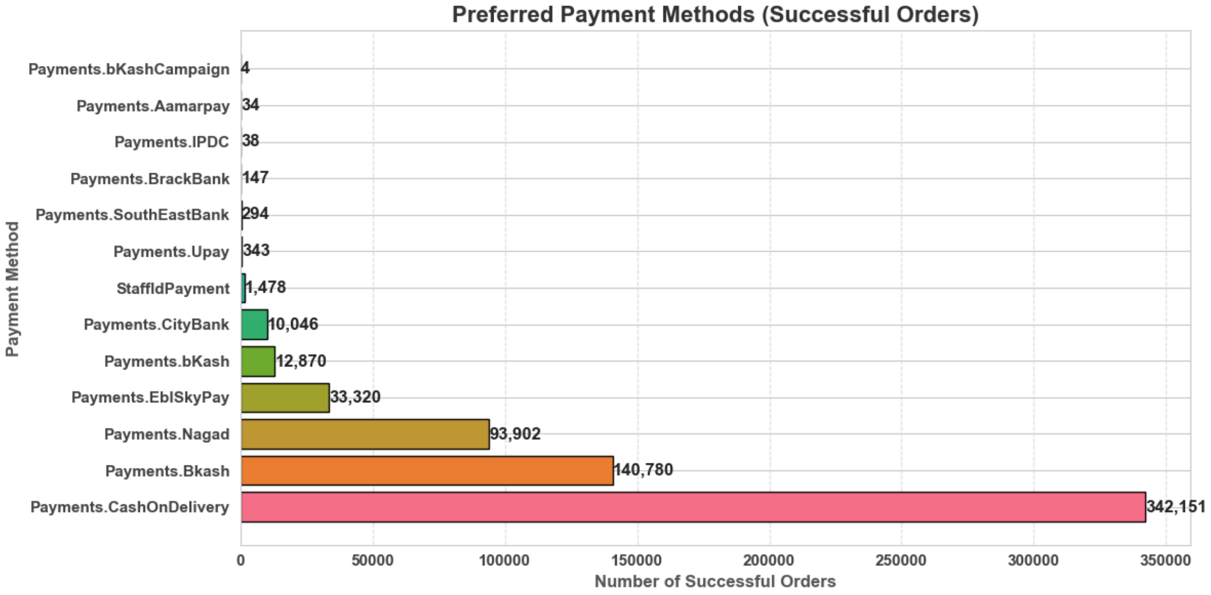
```
C:\Users\Srijon\AppData\Local\Temp\ipykernel_21500\412334030.py:15: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(y=top_10_cancellations.index, x=top_10_cancellations.values, palette
='coolwarm')
```



In [345…

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Filter only successfully completed orders
successful_orders = df[df['FinalOrderStatus'] == 'Sales']

# Count payment methods used in successful orders
payment_method_counts = successful_orders['Payment Method'].value_counts()
```

```
# Define a vibrant color palette
colors = sns.color_palette("husl", len(payment_method_counts))

# Create a horizontal bar chart
plt.figure(figsize=(12, 6))
bars = plt.barh(payment_method_counts.index, payment_method_counts.values, color=co

# Add labels to bars
for bar, count in zip(bars, payment_method_counts.values):
    plt.text(count + 50, bar.get_y() + bar.get_height()/2, f'{count:,}', va='center

# Title and labels
plt.title('Preferred Payment Methods (Successful Orders)', fontsize=16, fontweight=
plt.xlabel('Number of Successful Orders', fontsize=12, fontweight='bold', color='#5
plt.ylabel('Payment Method', fontsize=12, fontweight='bold', color='#555')

# Grid and layout adjustments
plt.grid(axis='x', linestyle="--", alpha=0.5)
plt.xticks(fontsize=11, fontweight='bold', color='#444')
plt.yticks(fontsize=11, fontweight='bold', color='#444')

# Show the plot
plt.tight_layout()
plt.show()
```



In [1]:

```
pip install --upgrade notebook
```

```
Requirement already satisfied: notebook in d:\newconda\lib\site-packages (7.2.2)
Collecting notebook
  Downloading notebook-7.3.2-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: jupyter-server<3,>=2.4.0 in d:\newconda\lib\site-pack
ages (from notebook) (2.14.1)
Requirement already satisfied: jupyterlab-server<3,>=2.27.1 in d:\newconda\lib\site-
packages (from notebook) (2.27.3)
Collecting jupyterlab<4.4,>=4.3.4 (from notebook)
  Downloading jupyterlab-4.3.5-py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: notebook-shim<0.3,>=0.2 in d:\newconda\lib\site-packa
ges (from notebook) (0.2.3)
Requirement already satisfied: tornado>=6.2.0 in d:\newconda\lib\site-packages (from
notebook) (6.4.1)
Requirement already satisfied: anyio>=3.1.0 in d:\newconda\lib\site-packages (from j
upyter-server<3,>=2.4.0->notebook) (4.2.0)
Requirement already satisfied: argon2-cffi>=21.1 in d:\newconda\lib\site-packages (f
rom jupyter-server<3,>=2.4.0->notebook) (21.3.0)
Requirement already satisfied: jinja2>=3.0.3 in d:\newconda\lib\site-packages (from
jupyter-server<3,>=2.4.0->notebook) (3.1.4)
Requirement already satisfied: jupyter-client>=7.4.4 in d:\newconda\lib\site-package
s (from jupyter-server<3,>=2.4.0->notebook) (8.6.0)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in d:\newconda\lib\site-pa
ckages (from jupyter-server<3,>=2.4.0->notebook) (5.7.2)
Requirement already satisfied: jupyter-events>=0.9.0 in d:\newconda\lib\site-package
s (from jupyter-server<3,>=2.4.0->notebook) (0.10.0)
Requirement already satisfied: jupyter-server-terminals>=0.4.4 in d:\newconda\lib\si
te-packages (from jupyter-server<3,>=2.4.0->notebook) (0.4.4)
Requirement already satisfied: nbconvert>=6.4.4 in d:\newconda\lib\site-packages (fr
om jupyter-server<3,>=2.4.0->notebook) (7.16.4)
Requirement already satisfied: nbformat>=5.3.0 in d:\newconda\lib\site-packages (fro
m jupyter-server<3,>=2.4.0->notebook) (5.10.4)
Requirement already satisfied: overrides>=5.0 in d:\newconda\lib\site-packages (from
jupyter-server<3,>=2.4.0->notebook) (7.4.0)
Requirement already satisfied: packaging>=22.0 in d:\newconda\lib\site-packages (fro
m jupyter-server<3,>=2.4.0->notebook) (24.1)
Requirement already satisfied: prometheus-client>=0.9 in d:\newconda\lib\site-packag
es (from jupyter-server<3,>=2.4.0->notebook) (0.14.1)
Requirement already satisfied: pywinpty>=2.0.1 in d:\newconda\lib\site-packages (fro
m jupyter-server<3,>=2.4.0->notebook) (2.0.10)
Requirement already satisfied: pyzmq>=24 in d:\newconda\lib\site-packages (from jupy
ter-server<3,>=2.4.0->notebook) (25.1.2)
Requirement already satisfied: send2trash>=1.8.2 in d:\newconda\lib\site-packages (f
rom jupyter-server<3,>=2.4.0->notebook) (1.8.2)
Requirement already satisfied: terminado>=0.8.3 in d:\newconda\lib\site-packages (fr
om jupyter-server<3,>=2.4.0->notebook) (0.17.1)
Requirement already satisfied: traitlets>=5.6.0 in d:\newconda\lib\site-packages (fr
om jupyter-server<3,>=2.4.0->notebook) (5.14.3)
Requirement already satisfied: websocket-client>=1.7 in d:\newconda\lib\site-package
s (from jupyter-server<3,>=2.4.0->notebook) (1.8.0)
Requirement already satisfied: async-lru>=1.0.0 in d:\newconda\lib\site-packages (fr
om jupyterlab<4.4,>=4.3.4->notebook) (2.0.4)
Requirement already satisfied: httpx>=0.25.0 in d:\newconda\lib\site-packages (from
jupyterlab<4.4,>=4.3.4->notebook) (0.27.0)
Requirement already satisfied: ipykernel>=6.5.0 in d:\newconda\lib\site-packages (fr
om jupyterlab<4.4,>=4.3.4->notebook) (6.28.0)
Requirement already satisfied: jupyter-lsp>=2.0.0 in d:\newconda\lib\site-packages
```

```
(from jupyterlab<4.4,>=4.3.4->notebook) (2.2.0)
Requirement already satisfied: setuptools>=40.8.0 in d:\newconda\lib\site-packages
(from jupyterlab<4.4,>=4.3.4->notebook) (75.1.0)
Requirement already satisfied: babel>=2.10 in d:\newconda\lib\site-packages (from ju
pyterlab-server<3,>=2.27.1->notebook) (2.11.0)
Requirement already satisfied: json5>=0.9.0 in d:\newconda\lib\site-packages (from j
upyterlab-server<3,>=2.27.1->notebook) (0.9.6)
Requirement already satisfied: jsonschema>=4.18.0 in d:\newconda\lib\site-packages
(from jupyterlab-server<3,>=2.27.1->notebook) (4.23.0)
Requirement already satisfied: requests>=2.31 in d:\newconda\lib\site-packages (from
jupyterlab-server<3,>=2.27.1->notebook) (2.32.3)
Requirement already satisfied: idna>=2.8 in d:\newconda\lib\site-packages (from anyi
o>=3.1.0->jupyter-server<3,>=2.4.0->notebook) (3.7)
Requirement already satisfied: sniffio>=1.1 in d:\newconda\lib\site-packages (from a
nyio>=3.1.0->jupyter-server<3,>=2.4.0->notebook) (1.3.0)
Requirement already satisfied: argon2-cffi-bindings in d:\newconda\lib\site-packages
(from argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->notebook) (21.2.0)
Requirement already satisfied: pytz>=2015.7 in d:\newconda\lib\site-packages (from b
abel>=2.10->jupyterlab-server<3,>=2.27.1->notebook) (2024.1)
Requirement already satisfied: certifi in d:\newconda\lib\site-packages (from httpx>
=0.25.0->jupyterlab<4.4,>=4.3.4->notebook) (2025.1.31)
Requirement already satisfied: httpcore==1.* in d:\newconda\lib\site-packages (from
httpx>=0.25.0->jupyterlab<4.4,>=4.3.4->notebook) (1.0.2)
Requirement already satisfied: h11<0.15,>=0.13 in d:\newconda\lib\site-packages (fro
m httpcore==1.*->httpx>=0.25.0->jupyterlab<4.4,>=4.3.4->notebook) (0.14.0)
Requirement already satisfied: comm>=0.1.1 in d:\newconda\lib\site-packages (from ip
ykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (0.2.1)
Requirement already satisfied: debugpy>=1.6.5 in d:\newconda\lib\site-packages (from
ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (1.6.7)
Requirement already satisfied: ipython>=7.23.1 in d:\newconda\lib\site-packages (fro
m ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (8.27.0)
Requirement already satisfied: matplotlib-inline>=0.1 in d:\newconda\lib\site-packag
es (from ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (0.1.6)
Requirement already satisfied: nest-asyncio in d:\newconda\lib\site-packages (from i
pykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (1.6.0)
Requirement already satisfied: psutil in d:\newconda\lib\site-packages (from ipykern
el>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (5.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in d:\newconda\lib\site-packages (fro
m jinja2>=3.0.3->jupyter-server<3,>=2.4.0->notebook) (2.1.3)
Requirement already satisfied: attrs>=22.2.0 in d:\newconda\lib\site-packages (from
jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook) (23.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in d:\newconda\l
ib\site-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook)
(2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in d:\newconda\lib\site-packages
(from jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in d:\newconda\lib\site-packages (from
jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook) (0.10.6)
Requirement already satisfied: python-dateutil>=2.8.2 in d:\newconda\lib\site-packag
es (from jupyter-client>=7.4.4->jupyter-server<3,>=2.4.0->notebook) (2.9.0.post0)
Requirement already satisfied: platformdirs>=2.5 in d:\newconda\lib\site-packages (f
rom jupyter-core!=5.0.*,>=4.12->jupyter-server<3,>=2.4.0->notebook) (3.10.0)
Requirement already satisfied: pywin32>=300 in d:\newconda\lib\site-packages (from j
upyter-core!=5.0.*,>=4.12->jupyter-server<3,>=2.4.0->notebook) (305.1)
Requirement already satisfied: python-json-logger>=2.0.4 in d:\newconda\lib\site-pac
kages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->notebook) (2.0.7)
```

```
Requirement already satisfied: pyyaml>=5.3 in d:\newconda\lib\site-packages (from ju
pyter-events>=0.9.0->jupyter-server<3,>=2.4.0->notebook) (6.0.1)
Requirement already satisfied: rfc3339-validator in d:\newconda\lib\site-packages (f
rom jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->notebook) (0.1.4)
Requirement already satisfied: rfc3986-validator>=0.1.1 in d:\newconda\lib\site-pack
ages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->notebook) (0.1.1)
Requirement already satisfied: beautifulsoup4 in d:\newconda\lib\site-packages (from
nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (4.12.3)
Requirement already satisfied: bleach!=5.0.0 in d:\newconda\lib\site-packages (from
nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (4.1.0)
Requirement already satisfied: defusedxml in d:\newconda\lib\site-packages (from nbc
onvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (0.7.1)
Requirement already satisfied: jupyterlab-pygments in d:\newconda\lib\site-packages
(from nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (0.1.2)
Requirement already satisfied: mistune<4,>=2.0.3 in d:\newconda\lib\site-packages (f
rom nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (2.0.4)
Requirement already satisfied: nbclient>=0.5.0 in d:\newconda\lib\site-packages (fro
m nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (0.8.0)
Requirement already satisfied: pandocfilters>=1.4.1 in d:\newconda\lib\site-packages
(from nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (1.5.0)
Requirement already satisfied: pygments>=2.4.1 in d:\newconda\lib\site-packages (fro
m nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (2.15.1)
Requirement already satisfied: tinycss2 in d:\newconda\lib\site-packages (from nbcon
vert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (1.2.1)
Requirement already satisfied: fastjsonschema>=2.15 in d:\newconda\lib\site-packages
(from nbformat>=5.3.0->jupyter-server<3,>=2.4.0->notebook) (2.16.2)
Requirement already satisfied: charset-normalizer<4,>=2 in d:\newconda\lib\site-pack
ages (from requests>=2.31->jupyterlab-server<3,>=2.27.1->notebook) (3.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in d:\newconda\lib\site-packages
(from requests>=2.31->jupyterlab-server<3,>=2.27.1->notebook) (2.2.3)
Requirement already satisfied: six>=1.9.0 in d:\newconda\lib\site-packages (from ble
ach!=5.0.0->nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (1.16.0)
Requirement already satisfied: webencodings in d:\newconda\lib\site-packages (from b
leach!=5.0.0->nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (0.5.1)
Requirement already satisfied: decorator in d:\newconda\lib\site-packages (from ipyt
hon>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (5.1.1)
Requirement already satisfied: jedi>=0.16 in d:\newconda\lib\site-packages (from ipy
thon>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (0.19.1)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in d:\newconda\lib\site
-packages (from ipython>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook)
(3.0.43)
Requirement already satisfied: stack-data in d:\newconda\lib\site-packages (from ipy
thon>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (0.2.0)
Requirement already satisfied: colorama in d:\newconda\lib\site-packages (from ipyth
on>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (0.4.6)
Requirement already satisfied: fqdn in d:\newconda\lib\site-packages (from jsonschem
a[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->notebook)
(1.5.1)
Requirement already satisfied: isoduration in d:\newconda\lib\site-packages (from js
onschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->no
tebook) (20.11.0)
Requirement already satisfied: jsonpointer>1.13 in d:\newconda\lib\site-packages (fr
om jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.
0->notebook) (2.1)
Requirement already satisfied: uri-template in d:\newconda\lib\site-packages (from j
sonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->n
otebook) (1.3.0)
Requirement already satisfied: webcolors>=24.6.0 in d:\newconda\lib\site-packages (f
rom jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.
4.0->notebook) (24.11.1)
Requirement already satisfied: cffi>=1.0.1 in d:\newconda\lib\site-packages (from ar
gon2-cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->notebook) (1.17.1)
Requirement already satisfied: soupsieve>1.2 in d:\newconda\lib\site-packages (from
beautifulsoup4->nbconvert>=6.4.4->jupyter-server<3,>=2.4.0->notebook) (2.5)
Requirement already satisfied: pycparser in d:\newconda\lib\site-packages (from cffi
>=1.0.1->argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->noteboo
k) (2.21)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in d:\newconda\lib\site-packages
(from jedi>=0.16->ipython>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->noteboo
k) (0.8.3)
Requirement already satisfied: wcwidth in d:\newconda\lib\site-packages (from prompt
-toolkit<3.1.0,>=3.0.41->ipython>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->
notebook) (0.2.5)
Requirement already satisfied: arrow>=0.15.0 in d:\newconda\lib\site-packages (from
isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-serve
r<3,>=2.4.0->notebook) (1.2.3)
Requirement already satisfied: executing in d:\newconda\lib\site-packages (from stac
k-data->ipython>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (0.8.3)
Requirement already satisfied: asttokens in d:\newconda\lib\site-packages (from stac
k-data->ipython>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (2.0.5)
Requirement already satisfied: pure-eval in d:\newconda\lib\site-packages (from stac
k-data->ipython>=7.23.1->ipykernel>=6.5.0->jupyterlab<4.4,>=4.3.4->notebook) (0.2.2)
Downloading notebook-7.3.2-py3-none-any.whl (13.2 MB)
   ---------------------------------------- 0.0/13.2 MB ? eta -:--:--
    --------------------------------------- 0.3/13.2 MB ? eta -:--:--
   --- ------------------------------------ 1.3/13.2 MB 4.5 MB/s eta 0:00:03
   ---------- ----------------------------- 3.4/13.2 MB 6.9 MB/s eta 0:00:02
   --------------- ------------------------ 5.2/13.2 MB 7.4 MB/s eta 0:00:02
   --------------------- ------------------ 7.1/13.2 MB 7.6 MB/s eta 0:00:01
   --------------------------- ------------ 8.9/13.2 MB 8.0 MB/s eta 0:00:01
   --------------------------------- ------ 11.0/13.2 MB 8.4 MB/s eta 0:00:01
   ---------------------------------------  13.1/13.2 MB 8.6 MB/s eta 0:00:01
   ---------------------------------------- 13.2/13.2 MB 8.3 MB/s eta 0:00:00
Downloading jupyterlab-4.3.5-py3-none-any.whl (11.7 MB)
   ---------------------------------------- 0.0/11.7 MB ? eta -:--:--
   ------- -------------------------------- 2.1/11.7 MB 11.8 MB/s eta 0:00:01
   -------------- ------------------------- 4.2/11.7 MB 10.5 MB/s eta 0:00:01
   -------------------- ------------------- 6.0/11.7 MB 10.0 MB/s eta 0:00:01
   --------------------------- ------------ 8.1/11.7 MB 10.1 MB/s eta 0:00:01
   ----------------------------------- ---- 10.2/11.7 MB 10.0 MB/s eta 0:00:01
   ---------------------------------------- 11.7/11.7 MB 9.6 MB/s eta 0:00:00
Installing collected packages: jupyterlab, notebook
  Attempting uninstall: jupyterlab
    Found existing installation: jupyterlab 4.2.5
    Uninstalling jupyterlab-4.2.5:
      Successfully uninstalled jupyterlab-4.2.5
  Attempting uninstall: notebook
    Found existing installation: notebook 7.2.2
    Uninstalling notebook-7.2.2:
      Successfully uninstalled notebook-7.2.2
Successfully installed jupyterlab-4.3.5 notebook-7.3.2
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:

```
pip uninstall notebook
```

In [ ]:

```
```