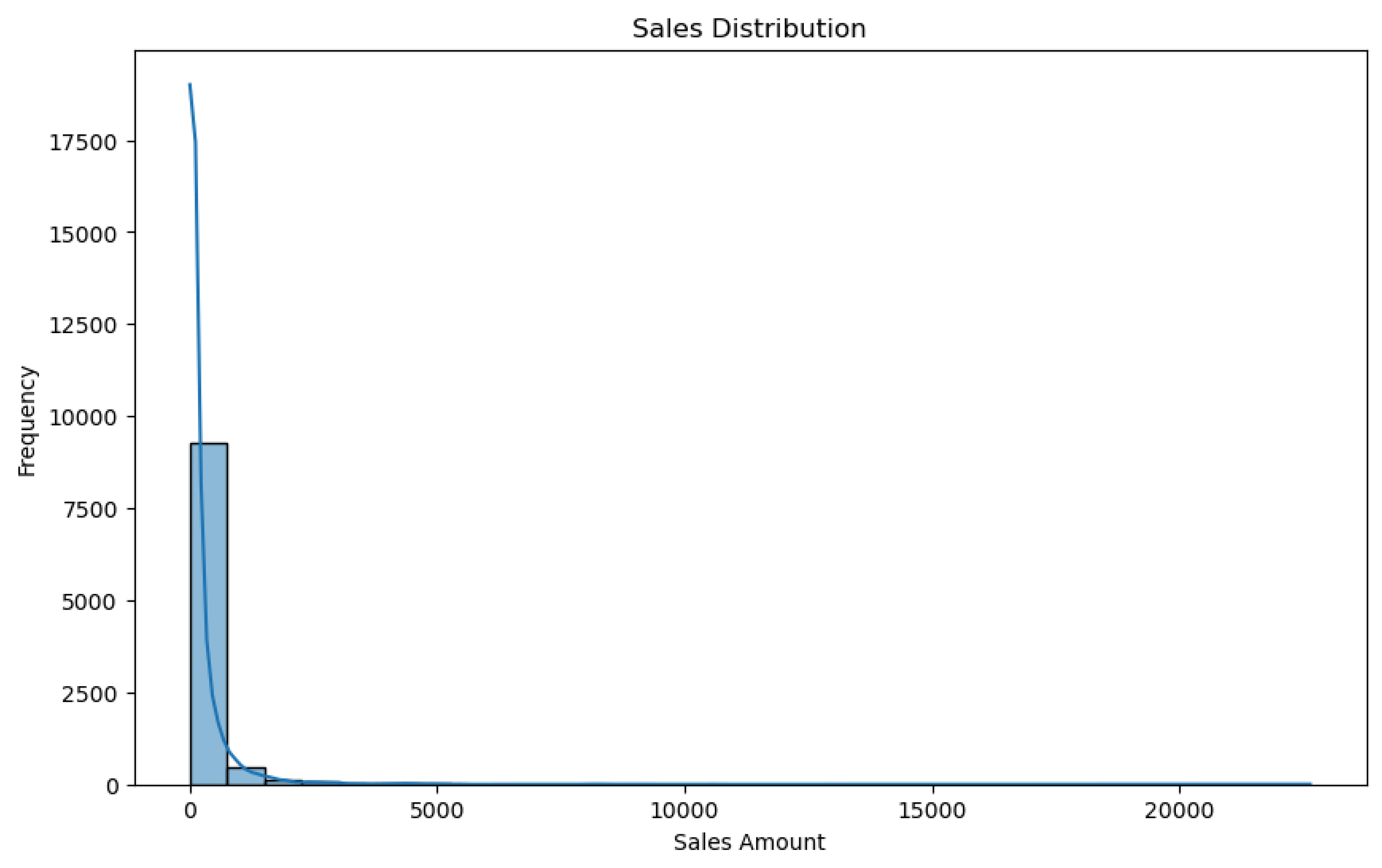
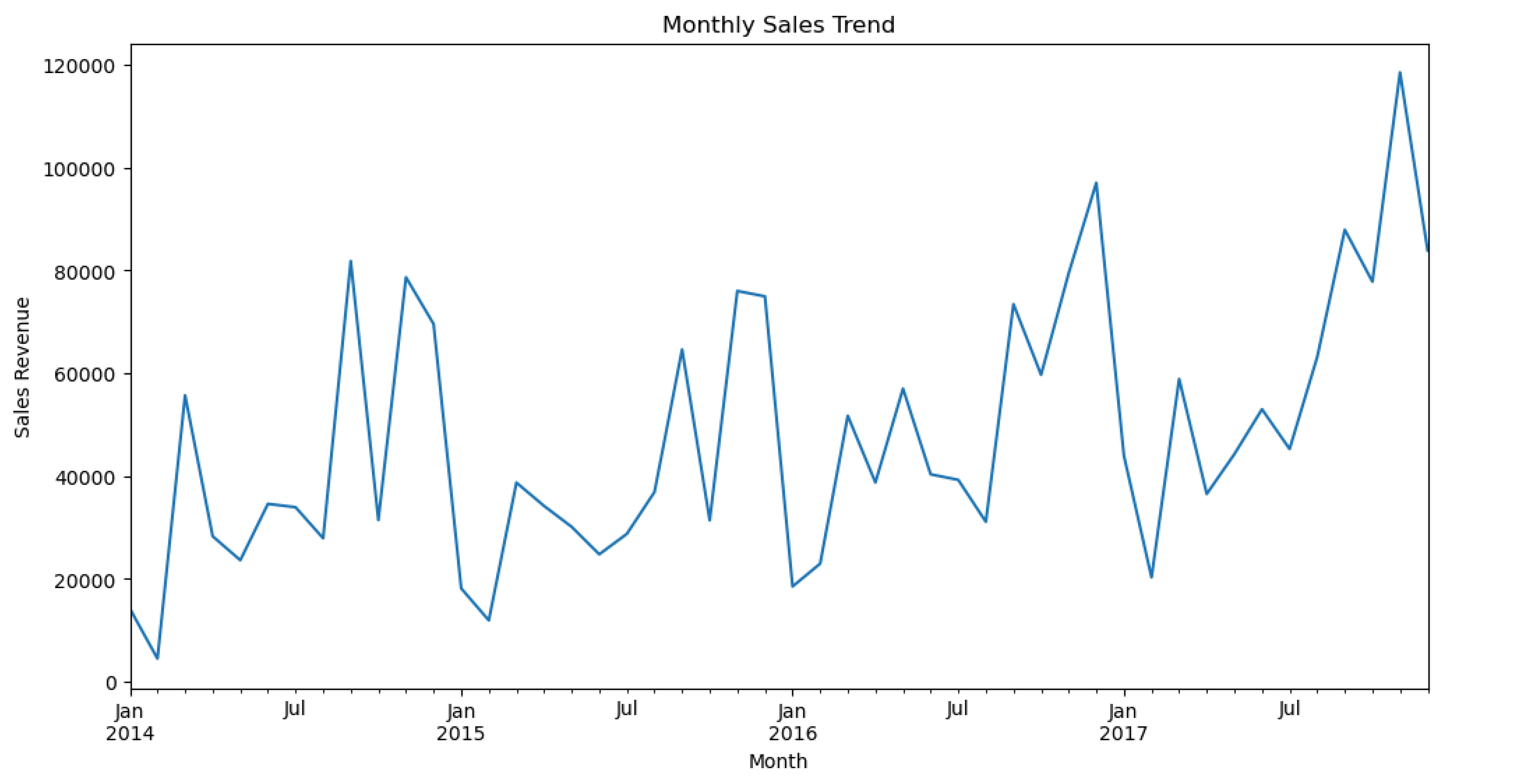
In [6]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans

# Step 1: Load Data with Specified Encoding
data_frame = pd.read_csv('Sample - Superstore.csv', encoding='ISO-8859-1')

# Step 2: Data Exploration
print("Data Preview:")
print(data_frame.head())

# Check data types and missing values
print("\nData Summary:")
print(data_frame.info())

# Step 3: Data Cleaning
# Convert date columns to datetime
data_frame['Order Date'] = pd.to_datetime(data_frame['Order Date'])
data_frame['Ship Date'] = pd.to_datetime(data_frame['Ship Date'])

# Fill missing values with zeros
data_frame.fillna(0, inplace=True)

# Remove duplicate rows
data_frame.drop_duplicates(inplace=True)

# Step 4: Compute Key Metrics
total_revenue = data_frame['Sales'].sum()
unique_orders = len(data_frame['Order ID'].unique())
avg_order_value = total_revenue / unique_orders
print(f"\nTotal Revenue: ${total_revenue}")
print(f"Average Order Value: ${avg_order_value:.2f}")

# Step 5: Visualize Sales Distribution
plt.figure(figsize=(10, 6))
sns.histplot(data_frame['Sales'], bins=30, kde=True)
plt.title('Sales Distribution')
plt.xlabel('Sales Amount')
plt.ylabel('Frequency')
plt.savefig('sales_distribution.png')

# Step 6: RFM Analysis (Customer Segmentation)
current_date = pd.to_datetime('today')
rfm_data = data_frame.groupby('Customer ID').agg({
    'Order Date': lambda x: (current_date - x.max()).days,
    'Order ID': 'count',
    'Sales': 'sum'
}).reset_index()

rfm_data.rename(columns={
    'Order Date': 'Recency',
    'Order ID': 'Frequency',
    'Sales': 'Monetary'
}, inplace=True)

# Step 7: Apply K-means Clustering
kmeans_model = KMeans(n_clusters=3)
rfm_data['Cluster'] = kmeans_model.fit_predict(rfm_data[['Recency', 'Frequency', 'Monetary']])

# Step 8: Identify Top Products
top_products = data_frame.groupby('Product Name')['Sales'].sum().nlargest(10)
print("\nTop Products:")
print(top_products)

# Step 9: Analyze Monthly Sales Trends
monthly_sales = data_frame.resample('M', on='Order Date')['Sales'].sum()
plt.figure(figsize=(12, 6))
monthly_sales.plot()
plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Sales Revenue')
plt.savefig('monthly_sales_trend.png')

# Step 10: Conclusion and Recommendations
print("\nFinal Insights:")
print("Based on the analysis, ... (insert key findings and recommendations)")

# Step 11: Documentation
with open('project_summary.txt', 'w') as file:
    file.write("Project Workflow:\n")
    file.write("- Data Loading\n")
    file.write("- Data Exploration\n")
    file.write("- Data Cleaning\n")
    file.write("- Key Metric Computation\n")
    file.write("- Sales Visualization\n")
    file.write("- Customer Segmentation\n")
    file.write("- Product Analysis\n")
    file.write("- Monthly Sales Analysis\n")
    file.write("- Final Insights\n")
```

```
Data Preview:
   Row ID        Order ID  Order Date   Ship Date       Ship Mode Customer ID  \
0       1  CA-2016-152156   11/8/2016  11/11/2016    Second Class    CG-12520
1       2  CA-2016-152156   11/8/2016  11/11/2016    Second Class    CG-12520
2       3  CA-2016-138688   6/12/2016   6/16/2016    Second Class    DV-13045
3       4  US-2015-108966  10/11/2015  10/18/2015  Standard Class    SO-20335
4       5  US-2015-108966  10/11/2015  10/18/2015  Standard Class    SO-20335

     Customer Name    Segment        Country             City  ...  \
0      Claire Gute   Consumer  United States        Henderson  ...
1      Claire Gute   Consumer  United States        Henderson  ...
2  Darrin Van Huff  Corporate  United States      Los Angeles  ...
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...

  Postal Code  Region       Product ID         Category Sub-Category  \
0       42420   South  FUR-BO-10001798        Furniture    Bookcases
1       42420   South  FUR-CH-10000454        Furniture       Chairs
2       90036    West  OFF-LA-10000240  Office Supplies       Labels
3       33311   South  FUR-TA-10000577        Furniture       Tables
4       33311   South  OFF-ST-10000760  Office Supplies      Storage

                                        Product Name     Sales  Quantity  \
0                  Bush Somerset Collection Bookcase  261.9600         2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.9400         3
2  Self-Adhesive Address Labels for Typewriters b...   14.6200         2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                     Eldon Fold 'N Roll Cart System   22.3680         2

   Discount    Profit
0      0.00   41.9136
1      0.00  219.5820
2      0.00    6.8714
3      0.45 -383.0310
4      0.20    2.5164

[5 rows x 21 columns]

Data Summary:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   object
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
None

Total Revenue: $2297200.8603000003
Average Order Value: $458.61

Top Products:
Product Name
Canon imageCLASS 2200 Advanced Copier                                          61599.824
Fellowes PB500 Electric Punch Plastic Comb Binding Machine with Manual Bind    27453.384
Cisco TelePresence System EX90 Videoconferencing Unit                          22638.480
HON 5400 Series Task Chairs for Big and Tall                                   21870.576
GBC DocuBind TL300 Electric Binding System                                     19823.479
GBC Ibimaster 500 Manual ProClick Binding System                               19024.500
Hewlett Packard LaserJet 3310 Copier                                           18839.686
HP Designjet T520 Inkjet Large Format Printer - 24" Color                      18374.895
GBC DocuBind P400 Electric Binding System                                      17965.068
High Speed Automatic Electric Letter Opener                                    17030.312
Name: Sales, dtype: float64

Final Insights:
Based on the analysis, ... (insert key findings and recommendations)
```





DASHBOARD ON SUPERSTORE:

In [16]:

```python
import pandas as pd
import ipywidgets as widgets
from IPython.display import display
import plotly.graph_objs as go

# Load the Superstore dataset
df = pd.read_csv('Sample - Superstore.csv', encoding='ISO-8859-1')

# Create dropdown for selecting Category
category_dropdown = widgets.Dropdown(
    options=df['Category'].unique(),
    value='Furniture',
    description='Category:',
)

# Initialize bar chart widget
bar_chart = go.FigureWidget()

# Initialize text widget to display selected category
selected_category_text = widgets.Text(
    value='Furniture',
    description='Selected Category:',
    disabled=True
)

# Function to update dashboard based on selected category
def update_dashboard(change):
    selected_category = change['new']
    filtered_df = df[df['Category'] == selected_category]
    sales_by_subcategory = filtered_df.groupby('Sub-Category')['Sales'].sum().reset_index()

    with bar_chart.batch_update():
        bar_chart.data = []
        bar_chart.add_trace(go.Bar(
            x=sales_by_subcategory['Sub-Category'],
            y=sales_by_subcategory['Sales'],
            marker=dict(color='blue')
        ))
        bar_chart.update_layout(
            title=f'Sales by Sub-Category for {selected_category}',
            xaxis=dict(title='Sub-Category'),
            yaxis=dict(title='Sales'),
        )

    selected_category_text.value = f'Selected Category: {selected_category}'

# Attach update function to dropdown's 'value' trait
category_dropdown.observe(update_dashboard, names='value')

# Initialize dashboard with default category
update_dashboard({'new': category_dropdown.value})

# Display dashboard components
display(widgets.VBox([category_dropdown, selected_category_text, bar_chart]))
```

```
VBox(children=(Dropdown(description='Category:', options=('Furniture', 'Office Supplies', 'Technology'), value…
```

DOCUMENTION:

Project Documentation: Superstore Sales Analysis Introduction This project aims to analyze and visualize sales data from a Superstore dataset using Python. The analysis covers various aspects, including data cleaning, key metric computation, customer segmentation, product analysis, and monthly sales trends. The primary objective is to gain insights into sales performance, customer behavior, and product popularity to make informed business decisions.

Project Workflow

Data Loading Library Used: Pandas Method: pd.read_csv('Sample - Superstore.csv', encoding='ISO-8859-1') Description: The dataset is loaded into a pandas DataFrame using the specified encoding. Data Exploration Methods: head(), info() Description: Displays the first few rows of the dataset. Provides a summary of the dataset structure, including data types and missing values. Data Cleaning Methods: pd.to_datetime(), fillna(), drop_duplicates() Description: Converts date columns to datetime format. Fills missing values with zeros. Removes duplicate rows to ensure data integrity. Compute Key Metrics Metrics: Total Revenue, Unique Orders, Average Order Value Description: Calculates the total revenue by summing up the 'Sales' column. Computes the number of unique orders and average order value. Visualize Sales Distribution Libraries Used: Matplotlib, Seaborn Methods: histplot(), savefig() Description: Plots a histogram to visualize the distribution of sales amounts. Saves the plot as 'sales_distribution.png'. RFM Analysis (Customer Segmentation) Methods: groupby(), agg() Metrics: Recency, Frequency, Monetary Description: Groups the data by 'Customer ID' and calculates RFM values. Renames the columns for better readability. Apply K-means Clustering Library Used: Scikit-learn Method: KMeans() Description: Applies K-means clustering to segment customers into three clusters based on RFM values. Identify Top Products Methods: groupby(), nlargest() Description: Groups the data by 'Product Name' and identifies the top 10 products based on sales. Analyze Monthly Sales Trends Methods: resample(), plot(), savefig() Description: Resamples the data to analyze monthly sales trends. Plots the monthly sales revenue and saves the plot as 'monthly_sales_trend.png'. Conclusion and Recommendations Description: Placeholder for inserting key findings, insights, and recommendations based on the analysis. Documentation Method: open(), write() Description: Writes a summary of the project workflow to a text file named 'project_summary.txt'. Conclusion This project provides a comprehensive analysis of the Superstore sales data, covering various aspects from data cleaning and exploration to customer segmentation and product analysis. The insights gained from this analysis can be used to make data-driven decisions to optimize sales strategies, improve customer relationships, and enhance overall business performance. Future Work Perform advanced analytics and machine learning models for sales forecasting. Incorporate additional datasets for more comprehensive analysis, such as customer reviews and product categories. Implement interactive dashboards for real-time monitoring and visualization of key metrics.

In [ ]: