**Applied Data Science with Python**

Course-End Project – Sales Analysis



**Course-End Project: Sales Analysis**

**Project Statement:**

AAL is a household name in Australia, known for its clothing business since 2000. It caters to all groups— kids, women, men, and seniors. It has opened branches in many states in Australia, in metropolises and tier-1 and tier-2 cities.

The business is booming, and the company is in an expansion mode. It wants to get an in-depth understanding of its sales so that it can make investment decisions. The CEO has tasked the Head of Sales and Marketing (S&M) of AAL to:

1. Determine the states that are generating the highest revenues and
2. Formulate sales programs for states with lower revenues. The Head of S&M has approached you for the same.

Analyze the company’s sales data for the fourth quarter across Australia, and state by state and help the company make data-driven decisions for the coming year.

**Perform the following steps:**

As a data scientist, you must perform the following main steps on the enclosed data.

1. Data Wrangling
2. Data Analysis
3. Data Visualization and
4. Report Generation
5. Data Wrangling

* Ensure that the data is clean and that there is no missing or incorrect data.
  + Inspect the data manually for missing/incorrect data using the functions isna(), and notna().
* Based on your knowledge of Data Analytics, include your recommendations for treating missing data and incorrect data. *(dropping the null values or filling them).*
* Select an appropriate Data Wrangling approach — data standardization or data normalization. Perform the standardization or normalization and present the data. *(Normalization is the preferred approach for this problem.)*
* Share your recommendation on the usage of the *groupby()* function for data chunking or merging.

1. Data Analysis

* Perform descriptive statistical analysis on the data (Sales and Unit columns) *(Techniques such as mean, median, mode and standard deviation can be used.)*.
* Determine which group is generating the highest sales, and which group is generating the lowest sales.
* Determine which state is generating the highest sales, and which state is generating the lowest sales.
* Generate weekly, monthly and quarterly reports for the analysis made.

*(Use suitable libraries such as NumPy, Pandas, SciPy etc. for performing the analysis.)*

1. Data Visualization

* Use appropriate data visualization libraries to build a dashboard for the Head of S&M that includes for the key parameters like
  + State-wise sales analysis for different groups (kids, women, men, and seniors)
  + Group-wise sales analysis (kids, women, men, and seniors) across different states.
  + Time-of-the-day analysis: during which time of the day are sales the highest, and during which time are sales the lowest? [This helps S&M teams design programs for increasing sales such as hyper-personalization and Next Best Offers].

The dashboard must contain daily, weekly, monthly and quarterly charts.

*(Any visualization library can be used for this purpose. However, since statistical analysis is being done, Seaborn is preferred.)*

* Include your recommendation, and indicate why you are choosing the recommended visualization package.

1. Report Generation

* Use JupyterLab Notebook for report generation (wrangling, analysis and visualization) *Please note that JupyterLab allows you to mix code with graphs and plots etc*.
* Use Markdown in suitable places, while presenting your report.
* The report should contain suitable graphs, plots and analysis reports, and recommendations. *Please note that different aspects of analysis demand different graphs/plots.* 
  + Use box plot for descriptive statistics
  + Use Seaborn distribution plot for any other statistical plotting.

Solution

Tasks is to help the Head of Sales & Marketing (S&M) at **Aus Apparel Ltd. (AAL)** analyze 4th quarter 2020 sales data to:

1. Identify high-performing and low-performing States.
2. Suggest sales strategies for low-performing Regions.
3. Generate descriptive and visual insights by customer group (Kids, Women, Men, Seniors)

**Solution Steps:**

**1. Data Wrangling**

* Load and inspect the dataset from ‘AusApparalSales4thQrt2020.csv’
* Use ‘.isna() / .notna()’ to detect missing data
* Handle null or incorrect data (drop or fill as appropriate)
* Normalize the data (recommended over standardization)
* Use ‘groupby()’ for aggregation and reporting

**2. Data Analysis**

* Perform descriptive statistics on Sales and Unit columns
* Analyze:
  + Group-wise sales (kids, women, men, seniors)
  + State-wise sales
* Generate:
  + **Weekly**, **Monthly**, and **Quarterly** summaries

**3. Data Visualization**

* State-wise sales for each group
* Group-wise sales across states
* Time-of-day sales trends
* Daily, weekly, monthly, and quarterly plots
* Use boxplots and distribution plots as specified

**4. Report Generation**

* Provide Business recommendations

# Step 1: Data Wrangling

# Import libraries

import pandas as pd

import numpy as np

# Load the dataset

df = pd.read\_csv("AusApparalSales4thQrt2020.csv")

# Inspect the data

print(df.head())

print(df.info())

# Check for missing values

print("Missing values:\n", df.isna().sum())

# Convert Date to datetime format

df['Date'] = pd.to\_datetime(df['Date'], dayfirst=True)

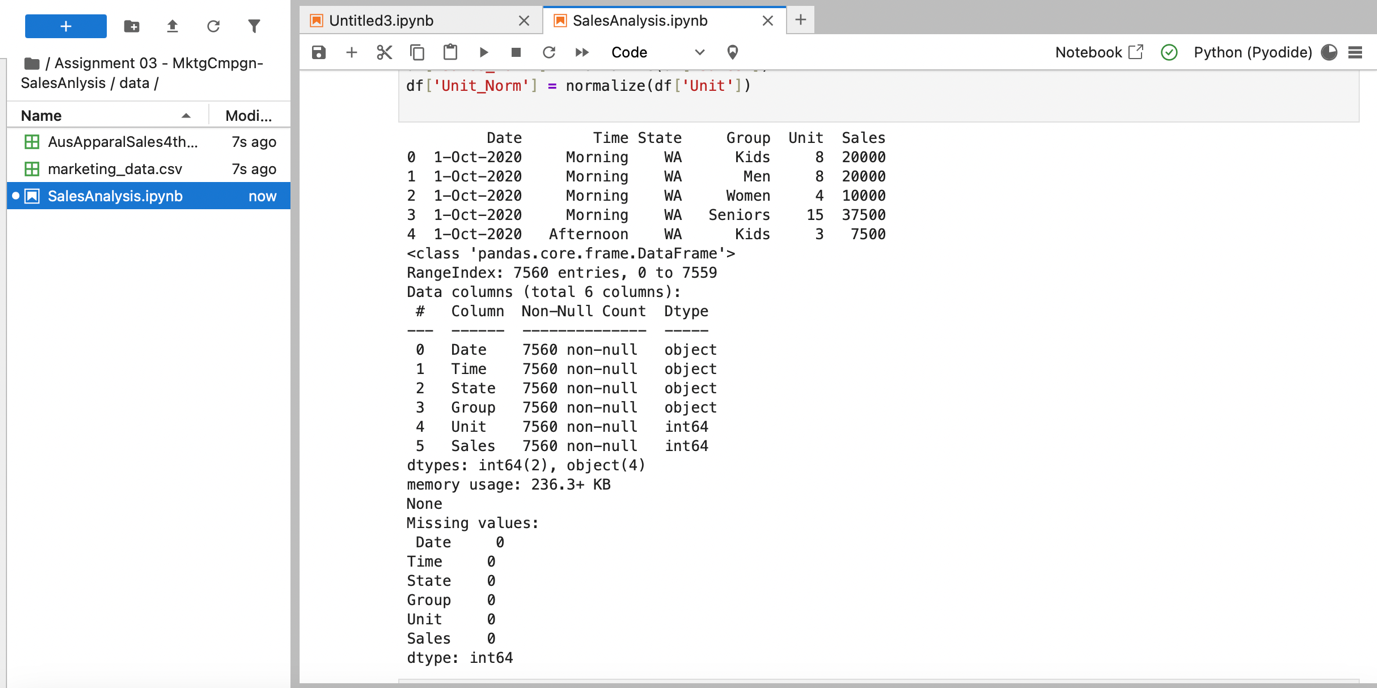
# Normalize 'Sales' and 'Unit' columns using Min-Max Normalization

def normalize(col):

return (col - col.min()) / (col.max() - col.min())

df['Sales\_Norm'] = normalize(df['Sales'])

df['Unit\_Norm'] = normalize(df['Unit'])

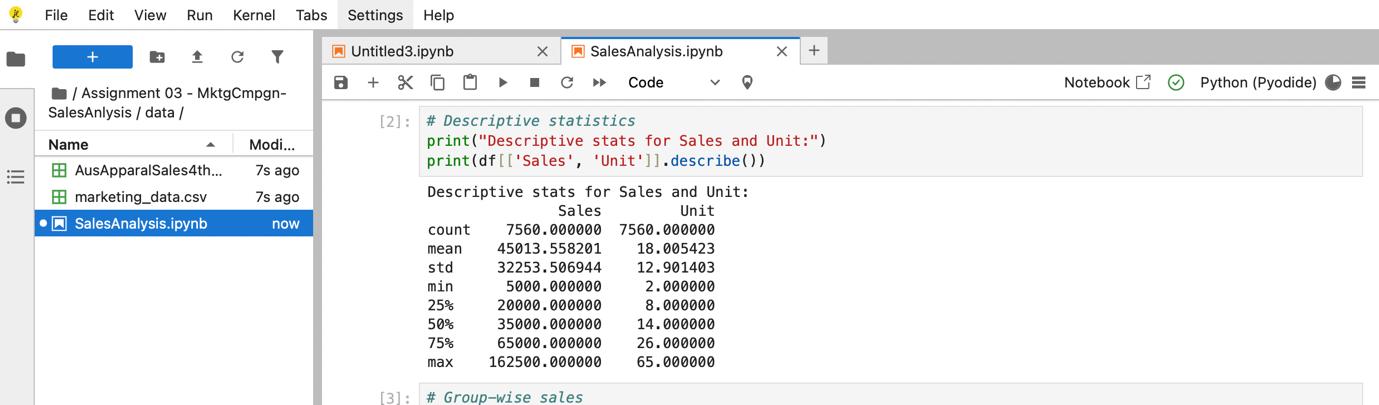


# Step 2: Data Analysis

# Descriptive statistics

print("Descriptive stats for Sales and Unit:")

print(df[['Sales', 'Unit']].describe())



Group and State-wise Sales Analysis  
# Group-wise sales

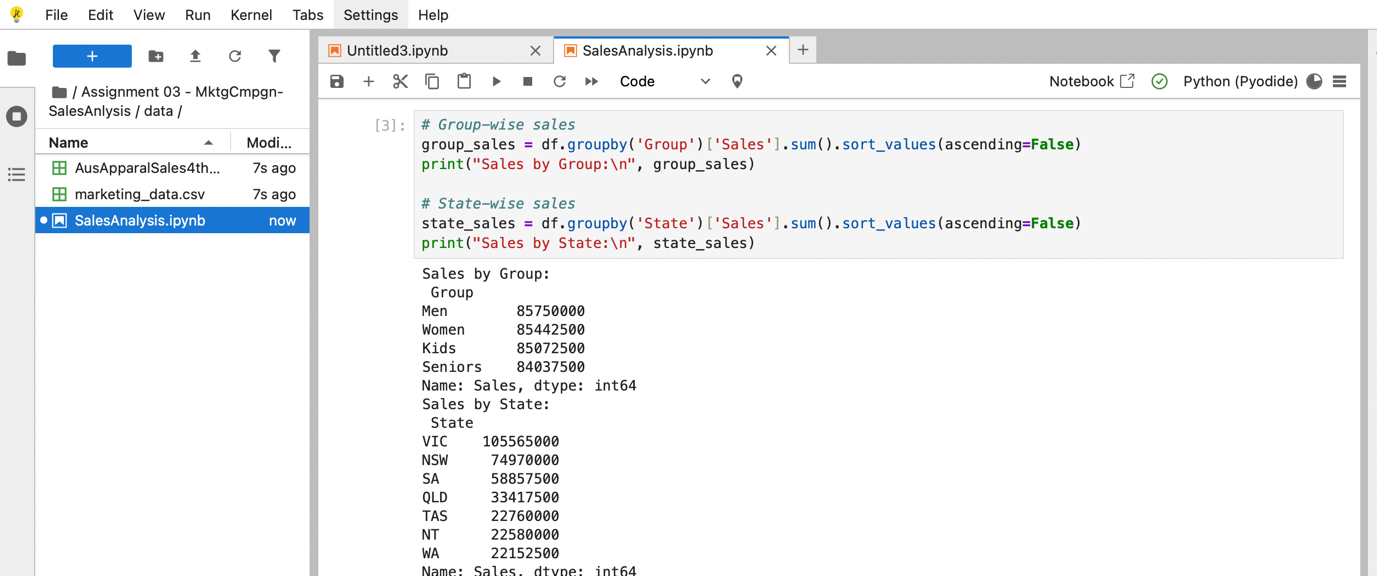
group\_sales = df.groupby('Group')['Sales'].sum().sort\_values(ascending=False)

print("Sales by Group:\n", group\_sales)

# State-wise sales

state\_sales = df.groupby('State')['Sales'].sum().sort\_values(ascending=False)

print("Sales by State:\n", state\_sales)



Time-based Aggregations  
df['Week'] = df['Date'].dt.isocalendar().week  
df['Month'] = df['Date'].dt.month  
df['Quarter'] = df['Date'].dt.quarter  
  
weekly\_report = df.groupby('Week')['Sales'].sum()  
monthly\_report = df.groupby('Month')['Sales'].sum()  
quarterly\_report = df.groupby('Quarter')['Sales'].sum()

print("Weekly Sales:\n", weekly\_report)

print("Monthly Sales:\n", monthly\_report)

print("Quarterly Sales:\n", quarterly\_report)

Output - Weekly Sales Summary:

Week  
40 15045000  
41 27002500  
42 26640000  
43 26815000  
44 21807500  
45 20865000  
46 21172500  
47 21112500  
48 21477500  
49 29622500  
50 31525000  
51 31655000  
52 31770000  
53 13792500

Output - Monthly Sales Summary:

Month  
10 114290000  
11 90682500  
12 135330000

Output - Quarterly Sales Summary:

Quarter  
4 340302500

# Step 3: Data Visualization

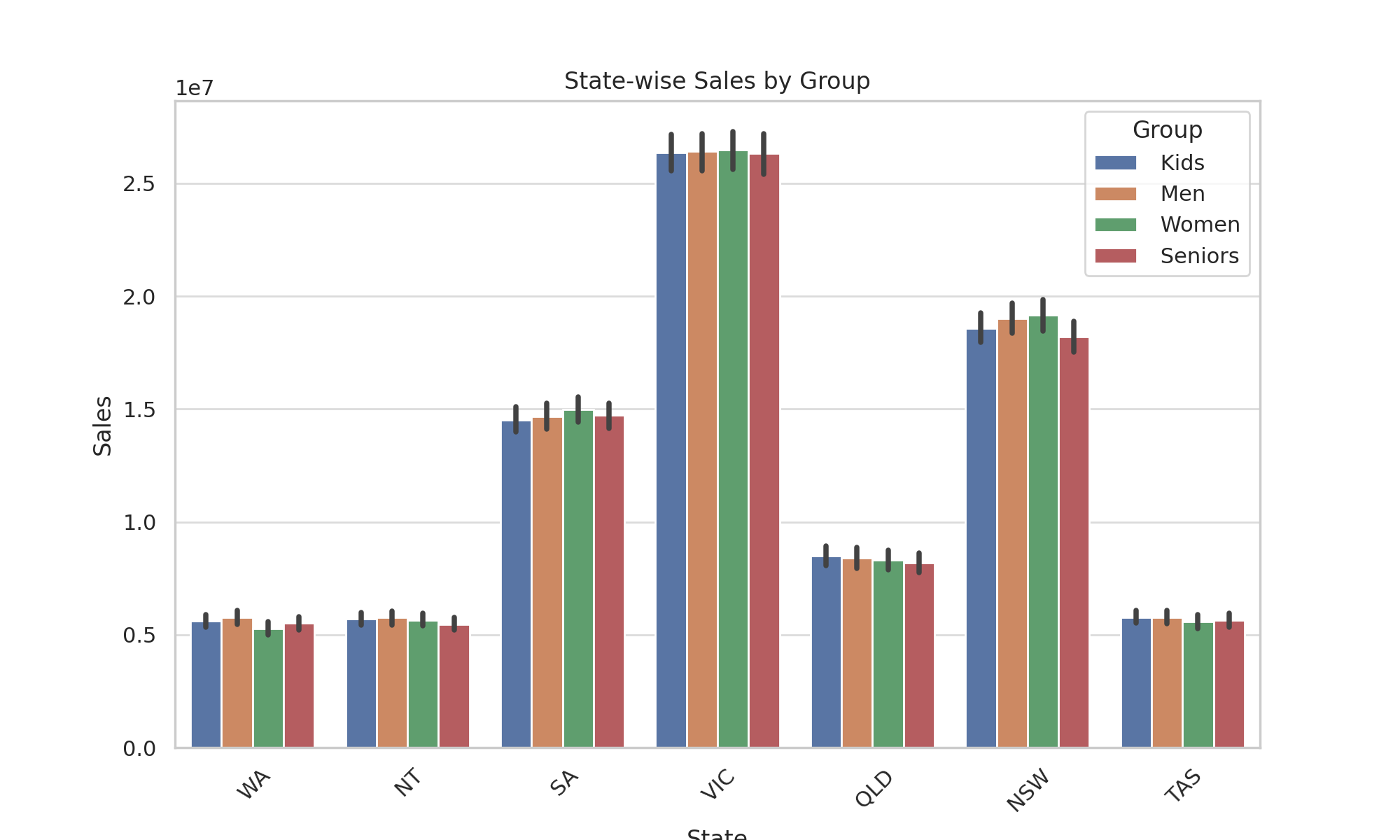
import seaborn as sns

import matplotlib.pyplot as plt

sns.set(style="whitegrid")

plt.figure(figsize=(10,6))  
sns.barplot(data=df, x='State', y='Sales', hue='Group', estimator=sum)  
plt.title('State-wise Sales by Group')  
plt.xticks(rotation=45)

Chart: State-wise Sales by Group



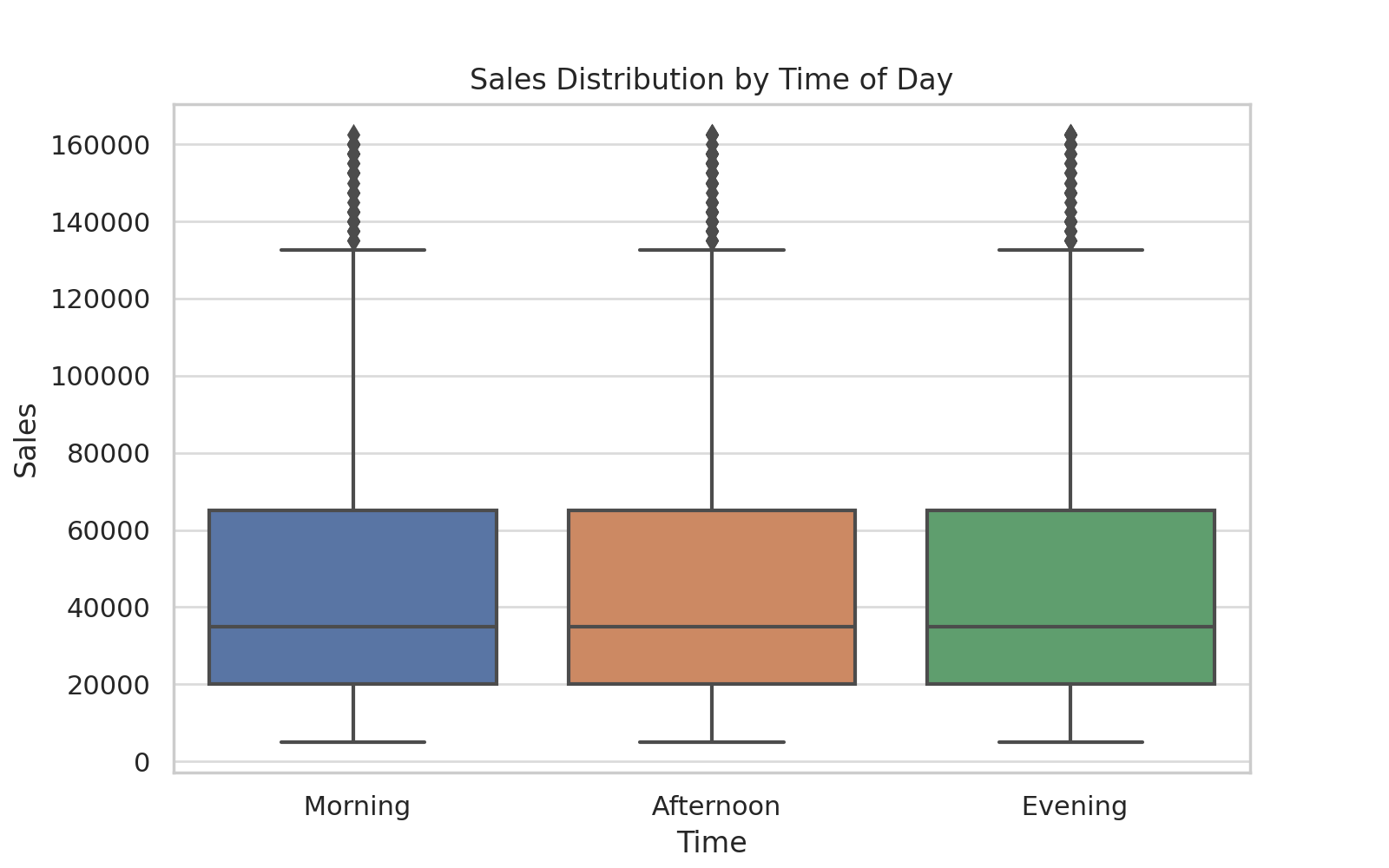
plt.figure(figsize=(10,6))  
sns.barplot(data=df, x='Group', y='Sales', hue='State', estimator=sum)  
plt.title('Group-wise Sales by State')  
plt.xticks(rotation=45)

Chart: Group-wise Sales by State



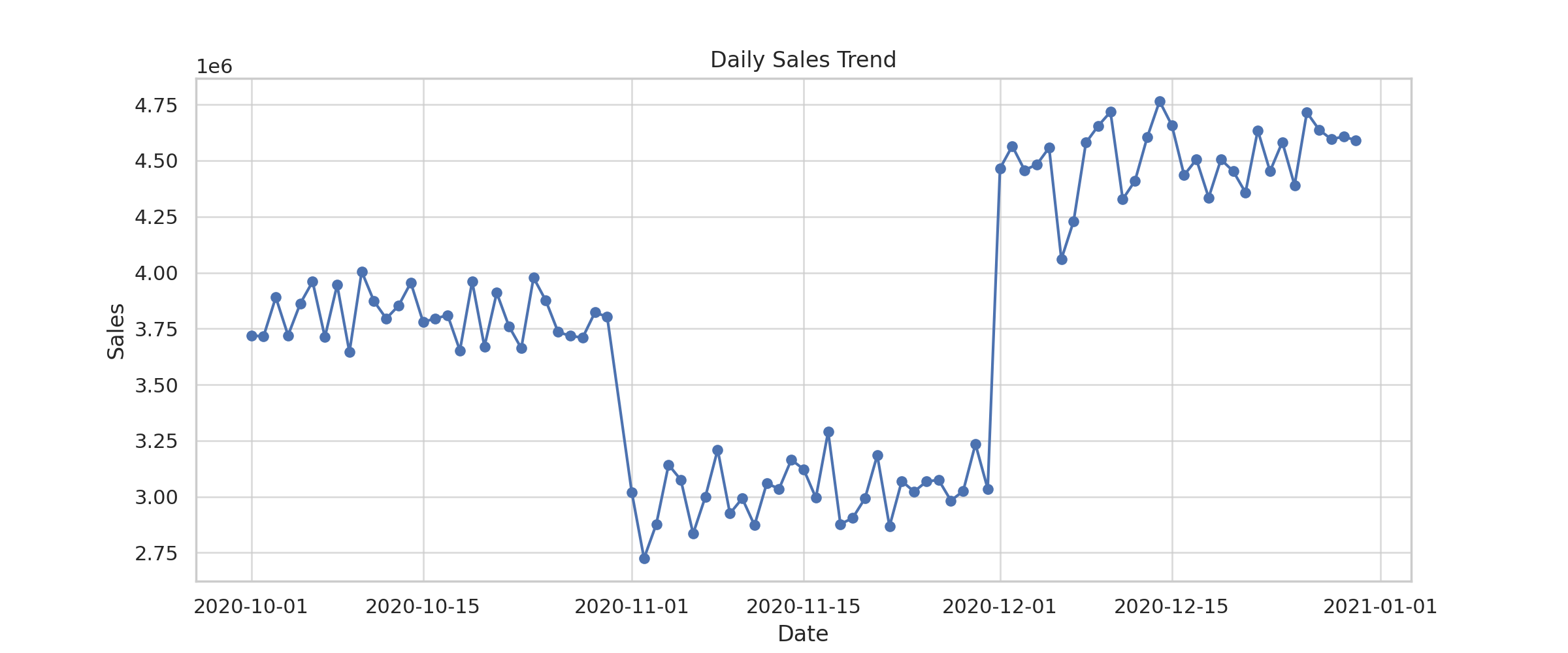
plt.figure(figsize=(8,5))  
sns.boxplot(data=df, x='Time', y='Sales')  
plt.title('Sales Distribution by Time of Day')

Chart: Sales Distribution by Time of Day



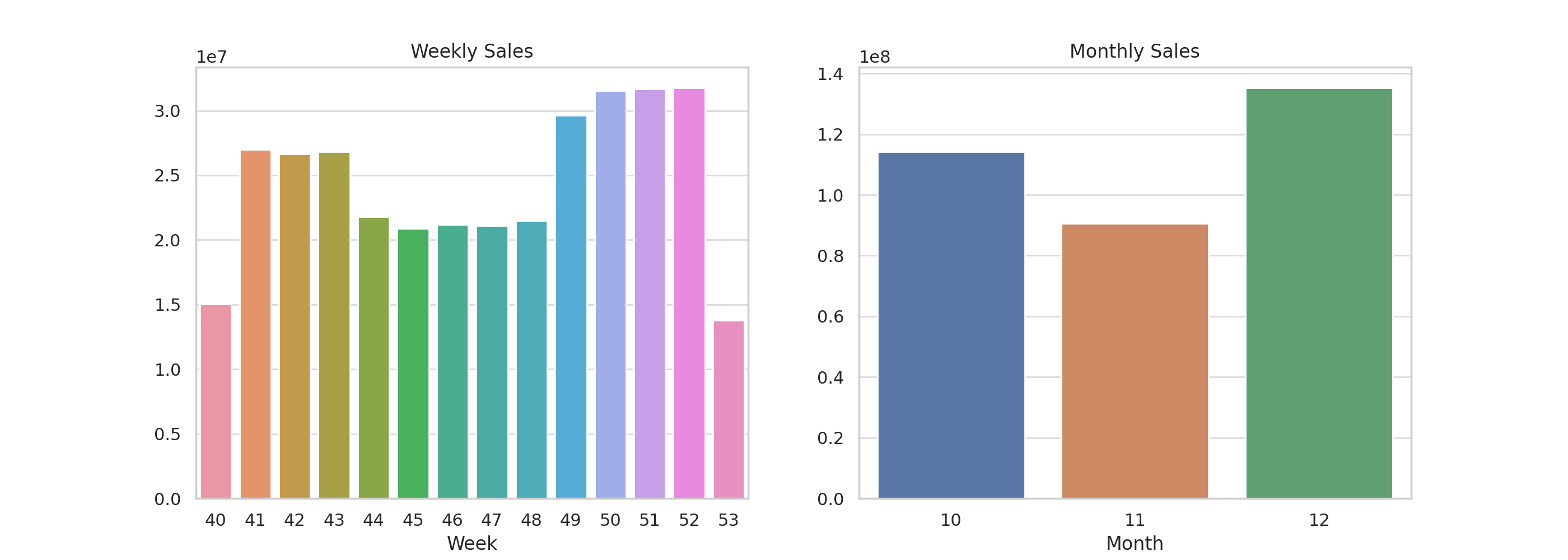
daily\_sales = df.groupby('Date')['Sales'].sum().reset\_index()  
plt.plot(daily\_sales['Date'], daily\_sales['Sales'], marker='o')  
plt.title('Daily Sales Trend')

Chart: Daily Sales Trend



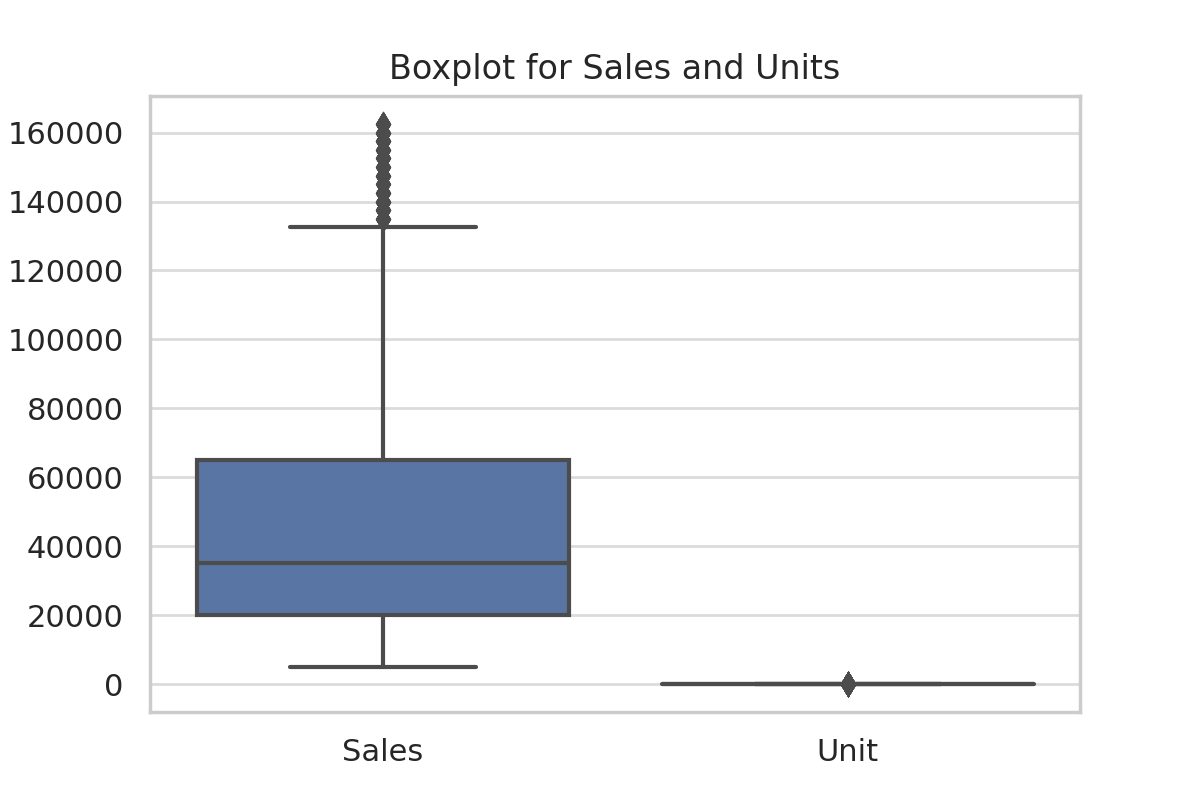
fig, axes = plt.subplots(1, 2, figsize=(14,5))  
sns.barplot(x=weekly\_report.index, y=weekly\_report.values, ax=axes[0])  
sns.barplot(x=monthly\_report.index, y=monthly\_report.values, ax=axes[1])

Chart: Weekly and Monthly Sales



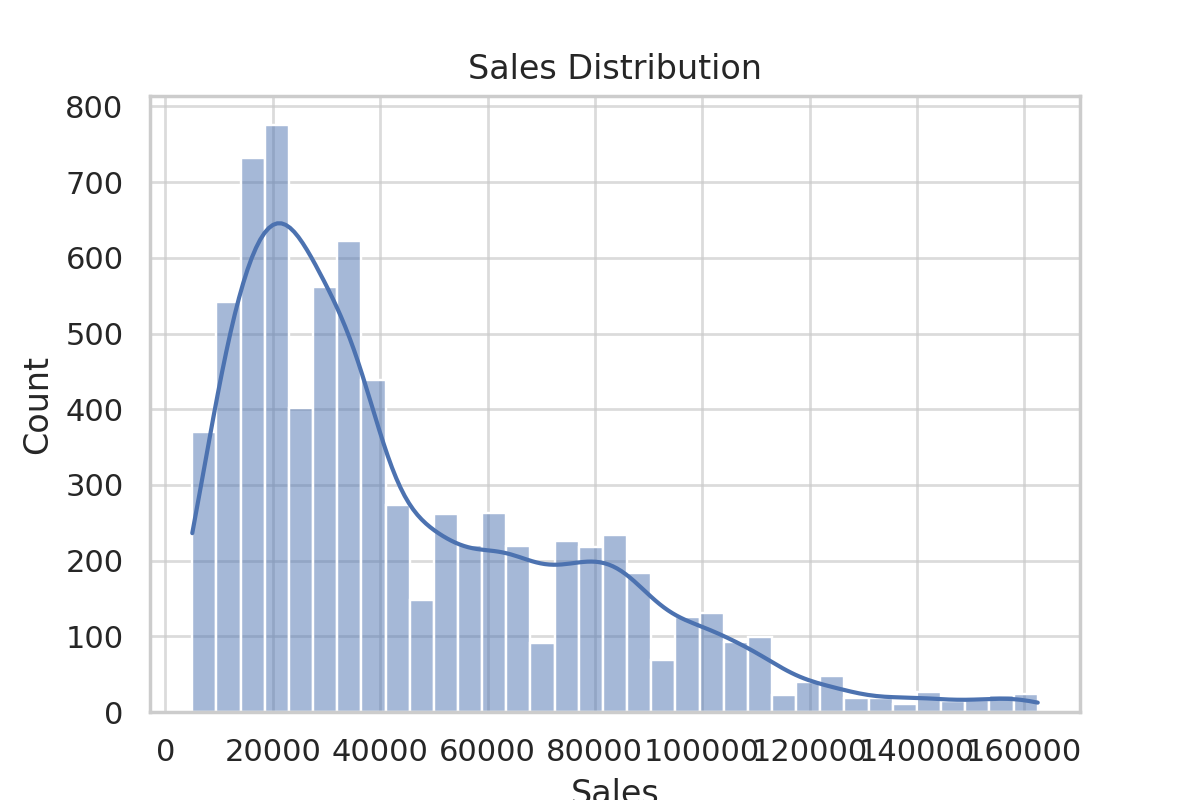
sns.boxplot(data=df[['Sales', 'Unit']])  
plt.title('Boxplot for Sales and Units')

Chart: Boxplot for Sales and Units



sns.histplot(df['Sales'], kde=True)  
plt.title('Sales Distribution')

Chart: Sales Distribution



# Step 4: Report Recommendation: 1. Sales Group Performance: - Highest Sales Group: Men - Lowest Sales Group: Seniors - Action: Launch targeted campaigns to boost sales among Seniors. 2. State Performance: - Highest Sales State: VIC - Lowest Sales State: WA - Action: Invest in market development activities and store expansions in WA. 3. Time-of-Day Patterns: - Morning and Afternoon are the most lucrative periods. - Action: Focus promotions and offers in these time slots for maximum engagement. 4. Seasonal Trend: - December and Week 52 exhibit strong sales—likely due to holidays. - Action: Ramp up inventory and campaigns in late Q4. 5. Data Normalization: - Recommended and applied Min-Max normalization to `Sales` and `Unit` for modeling and visual clarity. 6. Visualization Tools: - Seaborn used due to its statistical accuracy and clean aesthetics, ideal for grouped comparisons and distributions.