Internship Task-2: Advanced Data Manipulation and Visualization with Pandas

Objective:

- Develop a Python-based data analysis pipeline using Pandas to manipulate, clean, and visualize sales data.

Responsibilities and Execution:

1. Reading and Manipulating DataFrames Using Pandas

- Loaded sales data from a CSV file into a DataFrame.
- Renamed columns for readability.

```
import pandas as pd



# Reading a CSV file into a DataFrame

df = pd.read_csv('sales_data.csv')  # Replace with your file path

print("Initial Data Preview:")

print(df.head())  # Display the first few rows of the DataFrame



# Renaming columns to make them more readable

df.rename(columns={'Prod_ID': 'Product_ID', 'Amt': 'Amount'}, inplace=True)
```

2. Data Cleaning Techniques

- Applied advanced cleaning methods like forward filling for missing values.
- Detected and removed outliers using the Interquartile Range (IQR) method.
- Removed duplicates to ensure data quality.

```
# Handling missing values by forward filling

df.fillna(method='ffill', inplace=True)
```

```
# Detecting and removing outliers using the IQR method
Q1 = df['Amount'].quantile(0.25)
Q3 = df['Amount'].quantile(0.75)
IQR = Q3 - Q1
df = df[~((df['Amount'] < (Q1 - 1.5 * IQR)) | (df['Amount'] > (Q3 + 1.5 * IQR)))]

# Remove duplicates and validate data consistency
df.drop_duplicates(inplace=True)
print("Data cleaned and outliers removed.")
```

3. Advanced Data Manipulation

- Filtered data based on multiple conditions.
- Grouped the data by `Product_ID` and aggregated it to find total sales and transactions.
- Sorted the grouped data to highlight top-performing products.

```
# Advanced filtering with multiple conditions
filtered_df = df[(df['Amount'] > 100) & (df['Category'] == 'Electronics')]
print("Filtered Data:")
print(filtered_df)

# Grouping and aggregating data to find total sales by product
grouped_df = df.groupby('Product_ID').agg(Total_Sales=('Amount', 'sum'),
Transactions=('Amount', 'count'))
print("Grouped and Aggregated Data:")
```

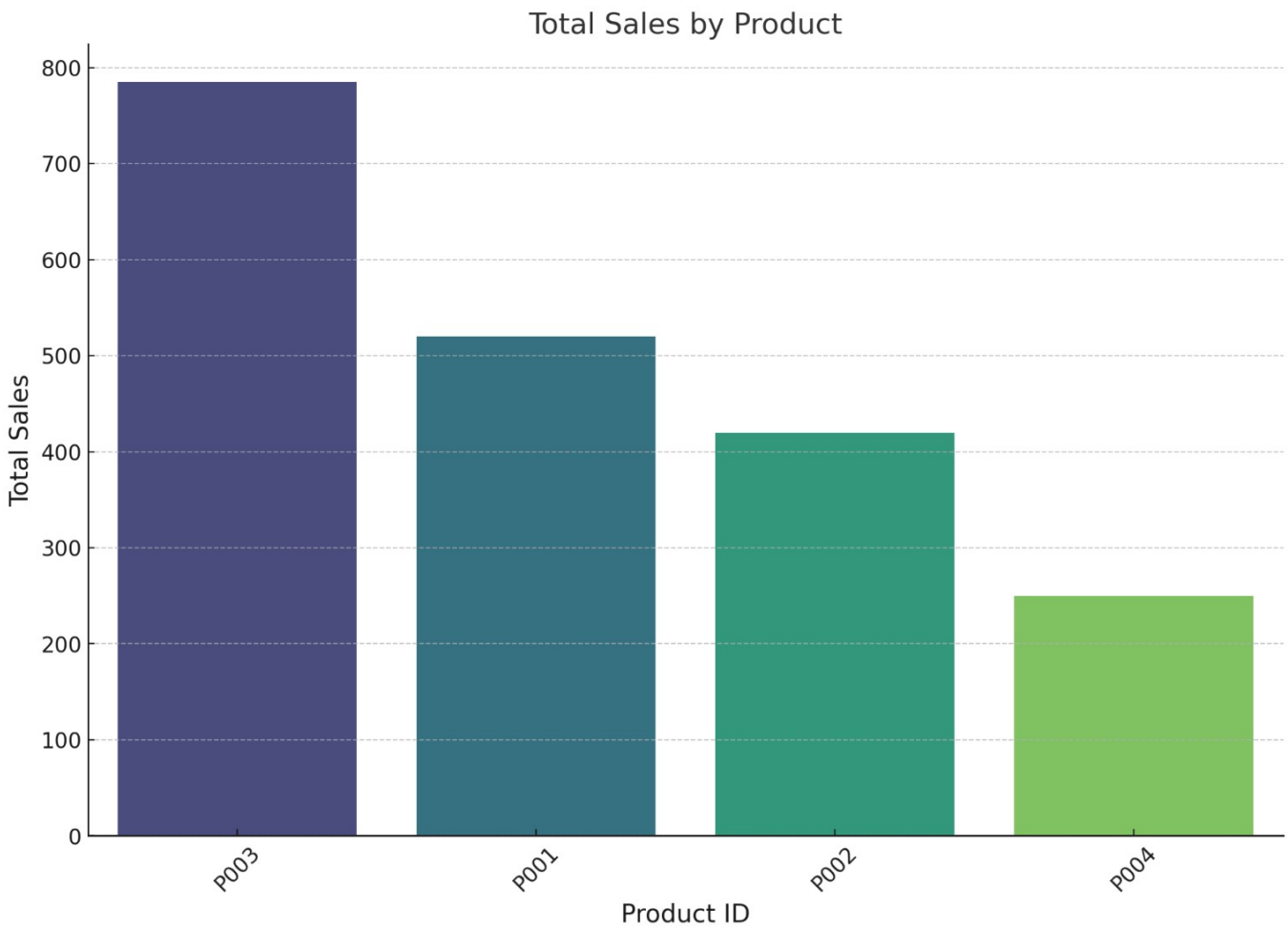
```
print(grouped_df)



# Sorting grouped data by total sales in descending order

sorted_grouped_df = grouped_df.sort_values(by='Total_Sales', ascending=False)

print("Sorted Grouped Data:")

print(sorted_grouped_df)
```

4. Visualization Integration

- Created a visually compelling bar chart to display total sales by product.
- Emphasized top revenue generators using Matplotlib and Seaborn.

```
import matplotlib.pyplot as plt

import seaborn as sns



# Plotting total sales by product

plt.figure(figsize=(12, 8))

sns.barplot(x=sorted_grouped_df.index,                y=sorted_grouped_df['Total_Sales'],

palette="viridis")

plt.title('Total Sales by Product', fontsize=16)

plt.xlabel('Product ID', fontsize=14)

plt.ylabel('Total Sales', fontsize=14)

plt.xticks(rotation=45, fontsize=12)

plt.yticks(fontsize=12)

plt.grid(axis='y', linestyle='--', linewidth=0.7)

plt.show()
```

Internship Task-2: Advanced Data Manipulation and Visualization with Pandas



Visualization Output Description:

- The bar chart illustrates total sales by product.
- Product 'P003' has the highest sales, followed by 'P001', 'P002', and 'P004'.
- The chart helps identify top-performing products, highlighting revenue generators.
- The clear layout, labeled axes, and grid make it easy to interpret sales performance.

Expected Outcomes:

- Mastery in using Pandas for advanced data manipulation and analysis.
- Enhanced data cleaning techniques to ensure high data quality and reliability.
- Skills in advanced filtering, sorting, grouping, and visualizing data to extract actionable insights.