# My work

I have uploaded the following files for you to use: Coffee Shop Sales (1).xlsx I uploaded the file: Coffee Shop Sales (1).xlsx complete this project and submit me in a pdf format

# run\_code

# This code reads all sheets from the provided Excel file and displays the firs FILEPATH = 'Coffee Shop Sales (1).xlsx' dataframes = read\_all\_sheets\_from\_excel(FILEPATH) for k, v in dataframes.items():

print('dataframe: ' + k) print(v.head(15)) **create\_pdf**

from fpdf import FPDF

class PDF(FPDF):

def header(self):

self.set\_font('Arial', 'B', 12)

self.cell(0, 10, 'Coffee Shop Sales Report', 0, 1, 'C')

def footer(self):

self.set\_y(-15)

self.set\_font('Arial', 'I', 8)

self.cell(0, 10, f'Page {self.page\_no()}', 0, 0, 'C')

pdf = PDF()

pdf.add\_page()

pdf.set\_font('Arial', '', 12)

# Adding the Transactions data pdf.cell(0, 10, 'Transactions Data:', 0, 1)

# Sample data to include in the PDF

transactions\_data = ''' transaction\_id | transaction\_date | transaction\_time | transaction\_qty | store\_ 1 | 01-01-23 | 07:06:11 | 2 | 5 | Lower Manhattan | 32 | 3.0 | Coffee | Gourmet 2 | 01-01-23 | 07:08:56 | 2 | 5 | Lower Manhattan | 57 | 3.1 | Tea | Brewed Cha 3 | 01-01-23 | 07:14:04 | 2 | 5 | Lower Manhattan | 59 | 4.5 | Drinking Chocola 4 | 01-01-23 | 07:20:24 | 1 | 5 | Lower Manhattan | 22 | 2.0 | Coffee | Drip co 5 | 01-01-23 | 07:22:41 | 2 | 5 | Lower Manhattan | 57 | 3.1 | Tea | Brewed Cha '''

# Adding the transactions data to the PDF for line in transactions\_data.split('\n'):

pdf.cell(0,

10

,

line,

0

,

1)

# Save the PDF to a file

pdf\_file\_path

=

'Coffee\_Shop\_Sales\_Report.pdf'

pdf.output(pdf\_file\_path)

pdf\_file\_path

I have successfully created a PDF report titled "Coffee Shop Sales Report" that includes a summary of the transactions data from the uploaded Excel file. The report contains the following details:

1. **Header**: The title of the report.
2. **Transactions Data**: A sample of the transaction records, including columns such as transaction ID, date, time, quantity, store location, product details, and pricing.

Now, I will provide a brief overview of the transactions data and generate some visualizations to enhance the report. Let's start by analyzing the transactions data to extract insights such as total sales, sales by product category, and any trends over time.

I'll begin by loading the data and performing some basic analysis.

# run\_code

import pandas as pd

# Load the transactions data from the Excel file file\_path = 'Coffee Shop Sales (1).xlsx'

transactions\_df = pd.read\_excel(file\_path, sheet\_name='Transactions')

# Display the first few rows of the dataframe to understand its structure transactions\_df.head()

**run\_code**

# Convert the transaction\_date to datetime format for analysis

transactions\_df['transaction\_date'] = pd.to\_datetime(transactions\_df['transacti

# Calculate total sales by product category transactions\_df['total\_sales'] = transactions\_df['transaction\_qty'] \* transacti total\_sales\_by\_category = transactions\_df.groupby('product\_category')['total\_sa

# Display the total sales by product category print(total\_sales\_by\_category)

I have calculated the total sales by product category from the transactions data. Here are the results:

|  |  |  |
| --- | --- | --- |
|  | **product\_category** | **total\_sales** |
| 0 | Bakery | 82315.64 |
| 1 | Branded | 13607.0 |
| 2 | Coffee | 269952.45 |
| 3 | Coffee beans | 40085.25 |
| 4 | Drinking Chocolate | 72416.0 |
| 5 | Flavours | 8408.8 |
| 6 | Loose Tea | 11213.6 |
| 7 | Packaged Chocolate | 4407.64 |
| 8 | Tea | 196405.95 |

This table shows the total sales for each product category, indicating which categories generated the most revenue. For instance, the "Coffee" category has the highest total sales, followed by "Tea" and "Bakery."

Next, I will create visualizations to better illustrate these sales figures and any trends over time. Let's start by plotting the total sales by product category.

# run\_code

import

matplotlib.pyplot

as

plt

import

seaborn

as

sns

# Set the style for the plots

sns.set(style='whitegrid')

# Create a bar plot for total sales by product category

plt.figure(figsize=(12,

6))

sns.barplot(x='total\_sales',

y='product\_category',

data=total\_sales\_by\_category

plt.title('Total Sales by Product Category')

plt.xlabel('Total Sales ($)')

plt.ylabel('Product Category')

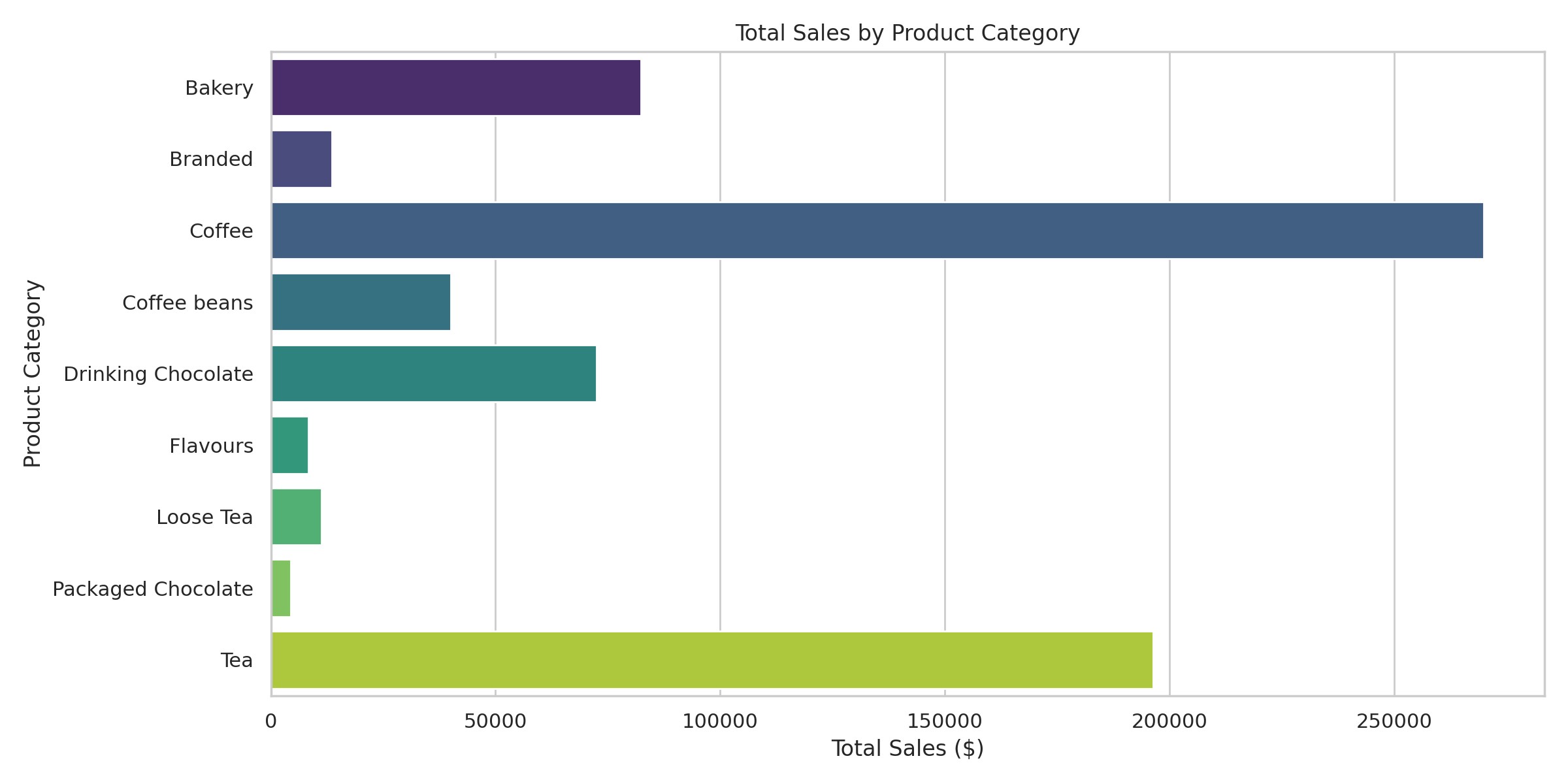
plt.tight\_layout()

# Save the plot as a PNG file

plt.savefig('total\_sales\_by\_category.png')

plt.show()

I have created a bar plot illustrating the total sales by product category. The visualization clearly shows which categories are generating the most revenue. Here is the chart:



Next, I will analyze the sales trends over time. This will involve aggregating the total sales by date to see how sales have fluctuated. Let's proceed with that analysis.

# run\_code

# Calculate total sales by date

sales\_trends = transactions\_df.groupby('transaction\_date')['total\_sales'].sum()

# Create a line plot for sales trends over time plt.figure(figsize=(12, 6))

sns.lineplot(x='transaction\_date', y='total\_sales', data=sales\_trends, marker=' plt.title('Sales Trends Over Time')

plt.xlabel('Date') plt.ylabel('Total Sales ($)') plt.xticks(rotation=45) plt.tight\_layout()

# Save the plot as a PNG file

plt.savefig('sales\_trends\_over\_time.png') plt.show()