

Module 5 Lab Worksheet: Visualizing Data with AI Templates

Using Matplotlib & Seaborn in Jupyter with Live Online Datasets

Participant Name: _____

Date: _____

Section 1: The Core Python Visualization Engines

Python's charting landscape relies heavily on two packages: `Matplotlib` (offering explicit canvas manipulation and granular layout control) and `Seaborn` (built on top of Matplotlib to create clean, statistical plots effortlessly with pre-configured themes).

Section 2: No-Syntax Customization Workflow

Instead of manually looking up complex plotting configuration syntax, analysts use generative AI templates to refine plots based on clean requirements:

 **Professional Charting Best Practices Checklist:**

- **Action Titles:** Inform the user of the key takeaway (e.g., `plt.title("Revenue Increased by 15% in Q2")`).
- **Readable Axis Labels:** Always include descriptive variables and standard metrics/units.
- **Brand Palette Coordination:** Use customized palette arrays instead of default high-contrast primary colors.

Section 3: Jupyter Active Code Sandbox

Run the code block below inside your Jupyter Notebook workspace to load a live public dataset stream (the famous "Tips" restaurant tracking dataset) and plot customized metrics:

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

# Step 1: Ingest public live online dataset stream
url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/tips.csv"
tips = pd.read_csv(url)

# Step 2: Establish Corporate Styling Rules
sns.set_theme(style="whitegrid")
plt.figure(figsize=(8, 5))

# Step 3: Draw the Visualization Map via Seaborn
# Prompt: "Create a barplot of average total bill by day using a professional blue palette"
sns.barplot(x="day", y="total_bill", data=tips, palette="Blues_d", errorbar=None)

# Step 4: Granular Customization Rules
plt.title("Average Total Bill Distribution by Day of Week", fontsize=14, pad=15,
weight='bold', color='#005088')
plt.xlabel("Day of the Week", fontsize=11, labelpad=10)
plt.ylabel("Average Bill Value ($)", fontsize=11, labelpad=10)

# Render clean visual map canvas
plt.tight_layout()
plt.show()
```

Section 4: Active Verification & Critique

Execute the code framework inside your local environment, inspect the generated chart, and complete the questions below:

Question 1: Feature Extraction Identification

Review the loaded dataset schema inside your notebook (using `tips.head()`). List three separate variables available for tracking client behavior within this public online table.

Question 2: Decoupling Visualization Libraries

In the workspace script above, identify which specific command utilizes **Seaborn** capabilities, and contrast that with an example line that requires **Matplotlib** directly:

Question 3: Prompting Customizations

Draft an alternative descriptive natural language prompt to give your AI assistant to modify this code. Ask it to replace the daily breakdown bar chart with a color-coded categorical boxplot mapping bill distributions by gender or smoking preference.
