K.R. Mangalam University

School of Engineering & Technology

Assignment

Probabilistic Modeling and Reasoning

Submitted by:

Name: Bhavya Rattan

Roll No: 2401201004

Course: BCA (AI & DS) Sec: B

Submitted to:

Dr. Kunal Rai

ASSIGNMENT 1

Q1. Bike Prices:

(a) **Calculate the Mean Price**

- Sum all prices: 12000 + 15000 + 18000 + 20000 + 50000 = 119000
- Total bikes counted: 5
- Mean price = Total sum ÷ Number of bikes = 119000 ÷ 5 = ₹23,800

(b) **Find the Median Price**

- Order the prices: 12000, 15000, 18000, 20000, 50000
- With an odd number of entries, the median is the middle (3rd) value = ₹18,000

(c) **Which Average to Highlight?**

- The mean is strongly influenced by the high priced ₹50,000 bike.
- The median remains stable and better reflects typical prices.
- Therefore, promoting the median price (₹18,000) will make the bikes appear more budget-friendly.

Q2. Relationship Between Histogram Shape and Central Tendency in Left-Skewed Distribution:

- Left-skewed data has a longer tail on the left side.
- The mode is greater than the median, which is greater than the mean (Mode > Median > Mean).
- The mean is drawn in the direction of the skew (left) making it less than the median.

Q3. Exam Scores Analysis:

(a) **Mean Calculation:**

- Sum of scores: 67 + 72 + 78 + 85 + 90 + 91 + 95 = 578
- Total scores: 7

- Mean = 578 ÷ 7 ≈ 82.57

(b) **Median Determination:**

- With 7 values, the median is the 4th number when sorted = 85

(c) **Mode Identification:**

- Since each score appears once, there is no mode.

(d) **Distribution Shape:**

- The distribution appears fairly symmetrical because the mean and median are close.

(e) **Five-Number Summary:**

- Minimum = 67
- First Quartile (Q1) = Median of lower half = 72
- Median = 85
- Third Quartile (Q3) = Median of upper half = 91
- Maximum = 95

Q4. Difference Between Population Parameter and Sample Statistic:

- **Population parameter:** A measure summarizing an entire population (e.g., average income of all residents).
- **Sample statistic:** A measure derived from a subset (sample) of the population (e.g., average income from surveyed individuals).

Q5. Employee Salary Data:

(a) **Calculate the Mean Salary:**

- Salaries: 45000 + 50000 + 55000 + 60000 + 250000 = 460000
- Number of employees: 5
- Mean salary = 460000 ÷ 5 = $92,000

(b) **Median Salary:**

- Middle salary in ordered list is $55,000

(c) **Choosing Which Average to Advertise:**

- Company may prefer advertising mean salary ($92,000) due to its higher value.
- Employees might look more favorably on the median ($55,000) since it reflects the typical wage.
- The disparity is caused by an exceptionally high salary skewing the mean upwards.

Q6. Students Scores Analysis:

Same as Q3

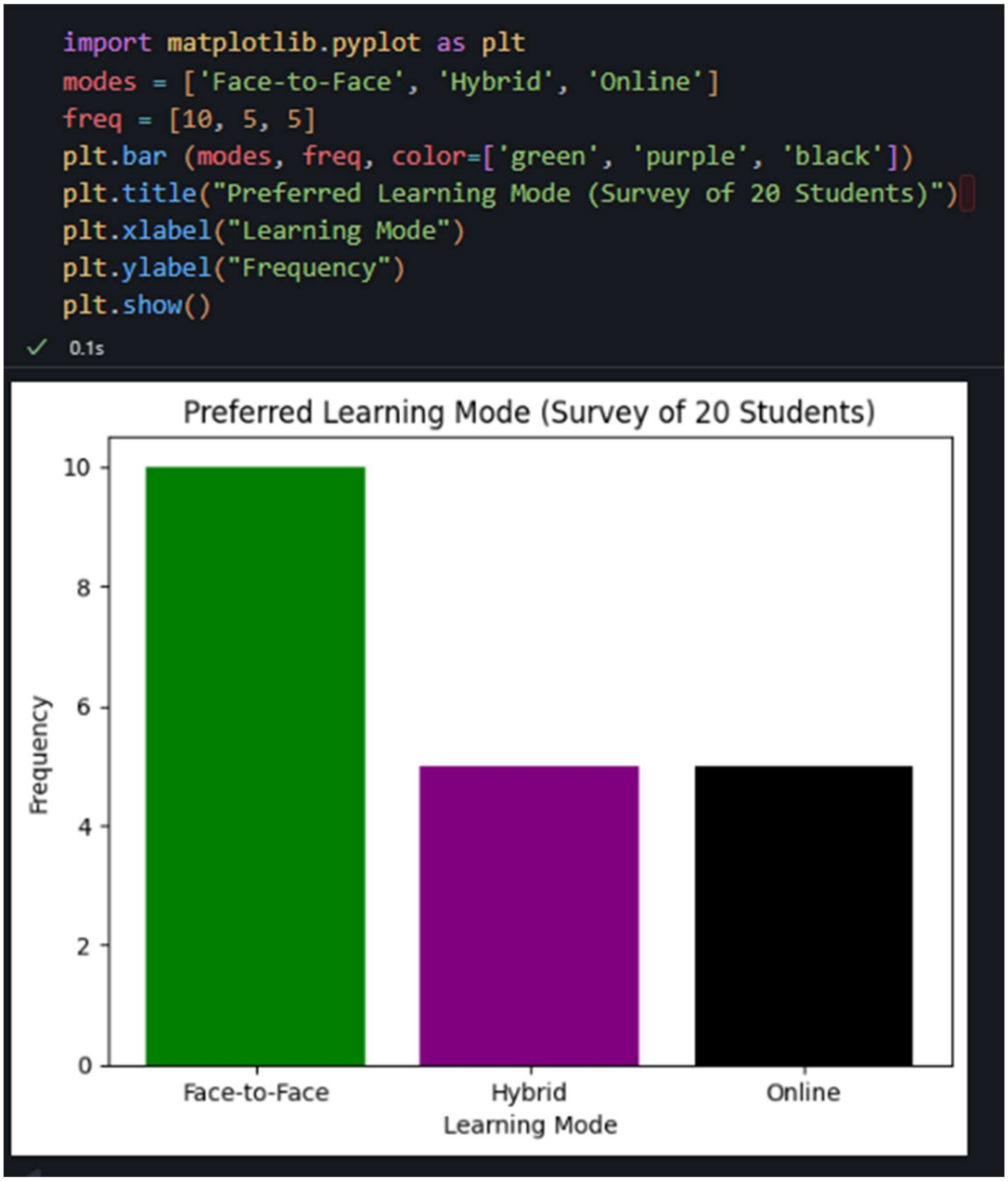
Q7. Modes of transportation:

Sol:

(a) Frequency table:

Mode	**Frequency**
F (In-person)	11
H (Hybrid)	5
O (Online)	4

(b) Bar chart would plot mode categories vs frequencies.



```
import matplotlib.pyplot as plt
modes = ['Face-to-Face', 'Hybrid', 'Online']
freq = [10, 5, 5]
plt.bar (modes, freq, color=['green', 'purple', 'black'])
plt.title("Preferred Learning Mode (Survey of 20 Students)")
plt.xlabel("Learning Mode")
plt.ylabel("Frequency")
plt.show()
```

✓ 0.1s

(c) Most popular mode: F (In-Person), 55% (11/20 students)

(d) Histogram not suitable for categorical data; bar chart is appropriate.

ASSIGNMENT 2

Q1. Probability of Second Marble Being Red Given First is Blue:

- Total marbles: 5 red + 7 blue = 12
- First marble drawn is blue, so remaining marbles = 11
- Remaining red marbles = 5
- Probability(second marble red | first blue) = Number of remaining red marbles ÷ Total remaining = 5/11

Q2. Probability of Having Disease Given Positive Test Result (Bayes' Theorem):

- Given:
  - P(Disease) = 0.01
  - P(No disease) = 0.99
  - P(Positive | Disease) = 0.95
  - P(Positive | No disease) = 0.02
- Using Bayes' Theorem:
  - Numerator = P(Positive | Disease) × P(Disease) = 0.95 × 0.01 = 0.0095
  - Denominator = (0.95 × 0.01) + (0.02 × 0.99) = 0.0095 + 0.0198 = 0.0293
  - P(Disease | Positive) = 0.0095 / 0.0293 ≈ 0.324

There is approximately a 32.4% chance the person has the disease after a positive test.

Q3. Revising Doctor's Opinion After Positive Test:

- Prior probability (doctor's estimate): P(Disease) = 0.30
- Compute posterior probability:
  - Numerator = 0.95 × 0.30 = 0.285

    - Denominator = 0.285 + (0.02 × 0.70) = 0.285 + 0.014 = 0.299
    - P(Disease | Positive) = 0.285 / 0.299 ≈ 0.952

The positive test increases the disease probability from 30% to about 95.2%.

Q4 & Q5. Probability Involving Students Studying Physics and Chemistry:

- Total students = 50
- Physics students = 30
- Chemistry students = 25
- Students studying both = 15

Calculate students studying only Physics:

- Only Physics = Total Physics - Both = 30 - 15 = 15

Calculate the probability a randomly chosen student studies only Physics:

- P(Only Physics) = 15 / 50 = 0.3

There is a 30% chance a chosen student studies Physics but not Chemistry.

ASSIGNMENT 3:

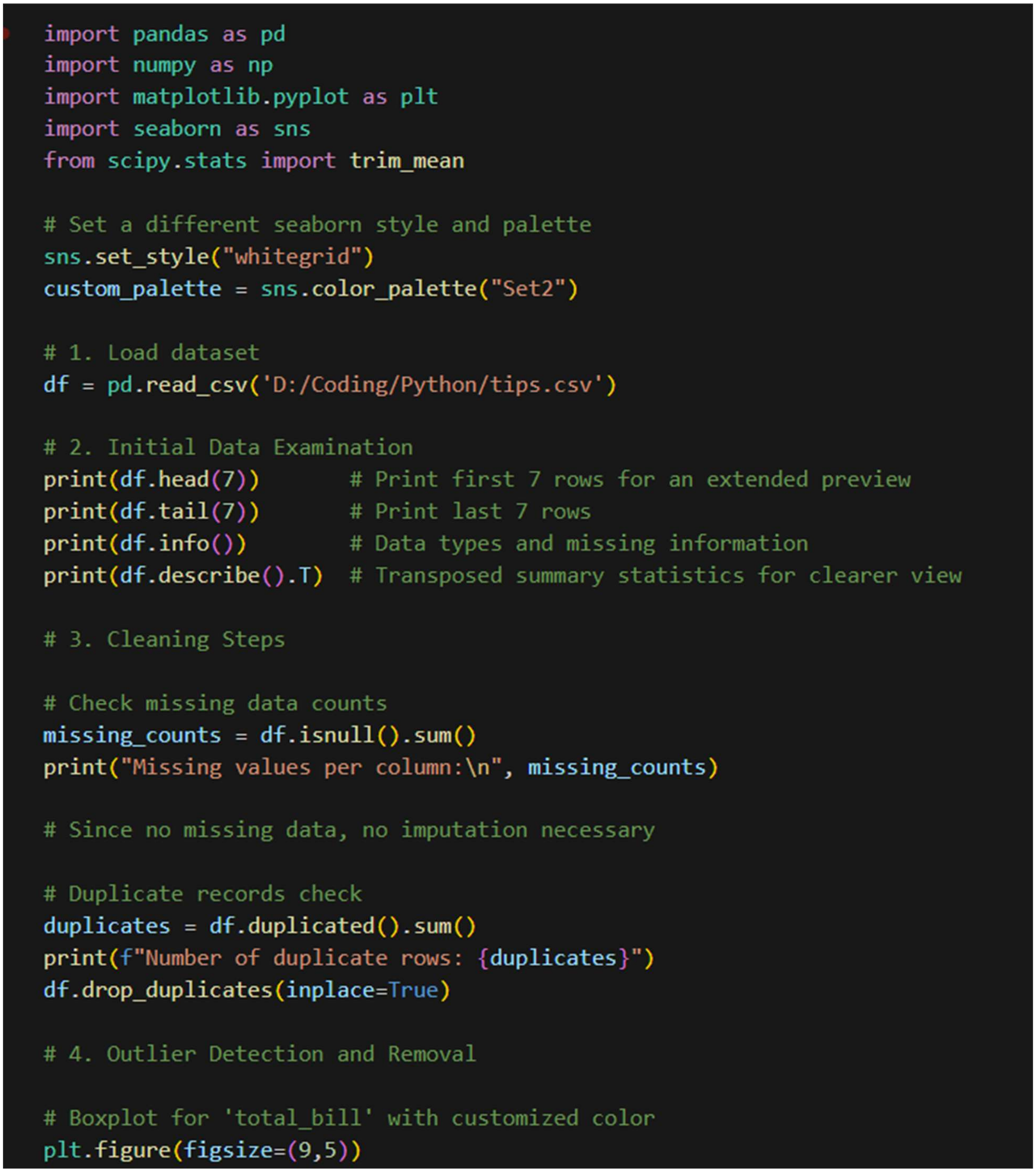
1. Project Objective:
Perform a complete Exploratory Data Analysis (EDA) on a dataset of your choice. Your goal is to understand the underlying structure of the data, discover patterns and relationships, identify anomalies and outliers, and test your initial hypotheses.

2. Dataset Selection:
You must select a dataset from (link unavailable) Choose a dataset that is rich enough to allow for meaningful analysis. It should have:
- At least 5 variables (columns).
- A mix of numerical and categorical variables is highly recommended.
- A sufficient number of rows (e.g., >100) to make analysis interesting.

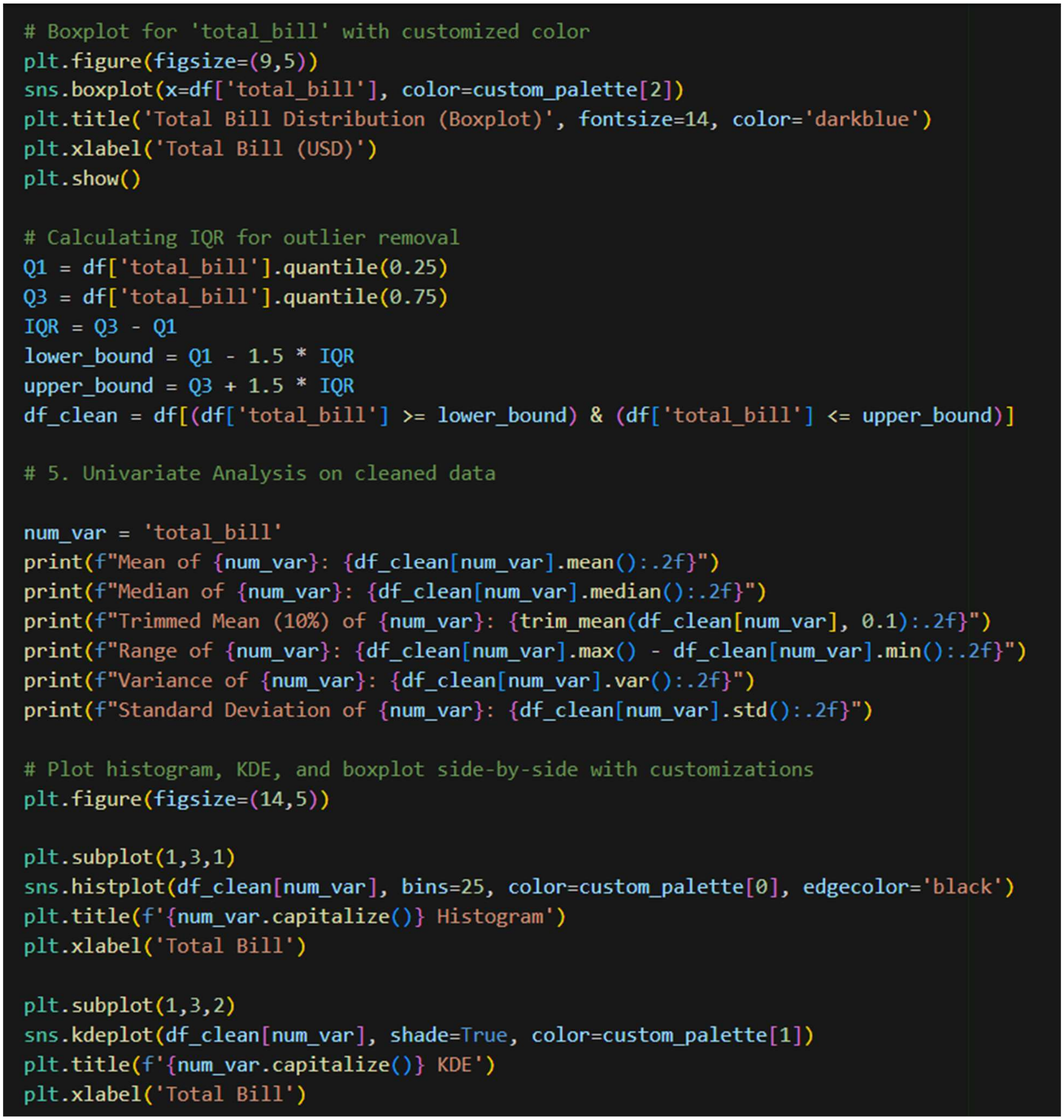
Popular beginner-friendly datasets on Kaggle include: Titanic, Iris, House Prices, Netflix Movies and TV Shows, Wine Reviews, or Pokemon Datasets. You are free to choose any that interests you.

3. Technical Requirements:
Your analysis must include the following, implemented in Python:
- Data Loading & Inspection:
    - Load the dataset using pandas.
    - Display the first and last few rows (.head(), .tail()).
    - Check the data types and summary info (.info(), .describe()).
- Data Cleaning (Mandatory):
    - Identify and handle missing values. Explain your method (e.g., removal, imputation).
    - Check for and handle any duplicate entries.
    - Identify outliers using visualizations (e.g., boxplots) and describe how you treated them.
- Univariate Analysis (From Project 1):
    - For numerical variables: Calculate and interpret Mean, Median, Trimmed Mean, Range, Variance, and Standard Deviation.
    - For categorical variables: Calculate frequency counts and modes.
    - Visualize distributions using histograms, KDE plots, and boxplots.
- Bivariate/Multivariate Analysis (From Project 2):
    - Create scatter plots to explore relationships between two numerical variables.
    - Create a correlation matrix and visualize it using a heatmap.
    - Use grouped boxplots or bar charts to explore relationships between categorical and numerical variables.
- Conclusion:
    - Summarize the 3-5 most important insights you discovered from your analysis.

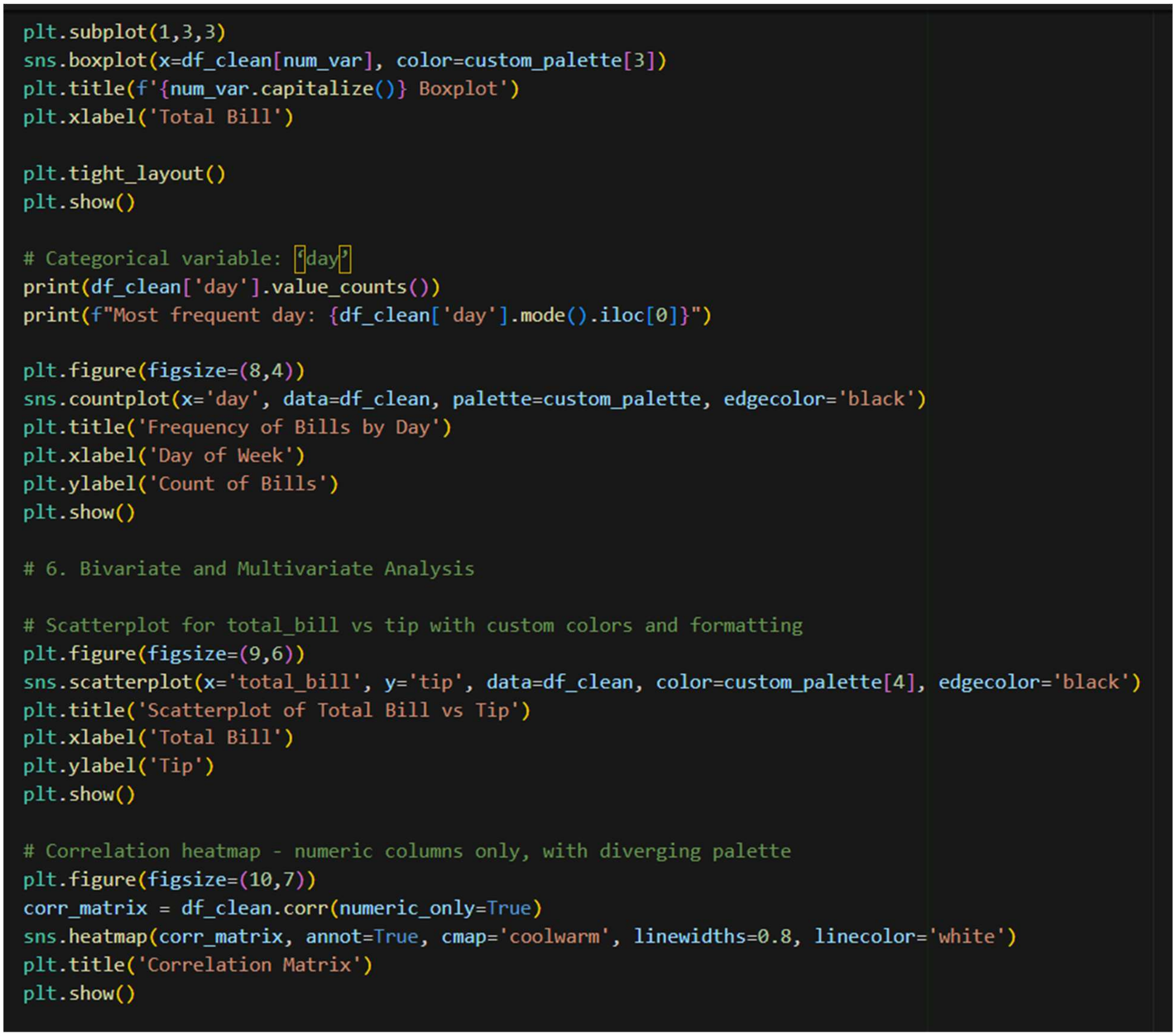
Code:



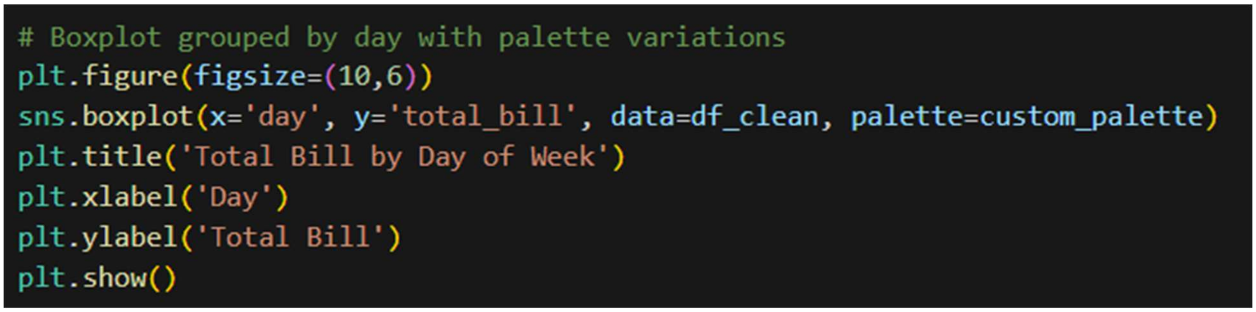
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import trim_mean

# Set a different seaborn style and palette
sns.set_style("whitegrid")
custom_palette = sns.color_palette("Set2")

# 1. Load dataset
df = pd.read_csv('D:/Coding/Python/tips.csv')

# 2. Initial Data Examination
print(df.head(7))       # Print first 7 rows for an extended preview
print(df.tail(7))       # Print last 7 rows
print(df.info())        # Data types and missing information
print(df.describe().T)  # Transposed summary statistics for clearer view

# 3. Cleaning Steps

# Check missing data counts
missing_counts = df.isnull().sum()
print("Missing values per column:\n", missing_counts)

# Since no missing data, no imputation necessary

# Duplicate records check
duplicates = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicates}")
df.drop_duplicates(inplace=True)

# 4. Outlier Detection and Removal

# Boxplot for 'total_bill' with customized color
plt.figure(figsize=(9,5))
```



```
# Boxplot for 'total_bill' with customized color
plt.figure(figsize=(9,5))
sns.boxplot(x=df['total_bill'], color=custom_palette[2])
plt.title('Total Bill Distribution (Boxplot)', fontsize=14, color='darkblue')
plt.xlabel('Total Bill (USD)')
plt.show()

# Calculating IQR for outlier removal
Q1 = df['total_bill'].quantile(0.25)
Q3 = df['total_bill'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df_clean = df[(df['total_bill'] >= lower_bound) & (df['total_bill'] <= upper_bound)]

# 5. Univariate Analysis on cleaned data

num_var = 'total_bill'
print(f"Mean of {num_var}: {df_clean[num_var].mean():.2f}")
print(f"Median of {num_var}: {df_clean[num_var].median():.2f}")
print(f"Trimmed Mean (10%) of {num_var}: {trim_mean(df_clean[num_var], 0.1):.2f}")
print(f"Range of {num_var}: {df_clean[num_var].max() - df_clean[num_var].min():.2f}")
print(f"Variance of {num_var}: {df_clean[num_var].var():.2f}")
print(f"Standard Deviation of {num_var}: {df_clean[num_var].std():.2f}")

# Plot histogram, KDE, and boxplot side-by-side with customizations
plt.figure(figsize=(14,5))

plt.subplot(1,3,1)
sns.histplot(df_clean[num_var], bins=25, color=custom_palette[0], edgecolor='black')
plt.title(f'{num_var.capitalize()} Histogram')
plt.xlabel('Total Bill')

plt.subplot(1,3,2)
sns.kdeplot(df_clean[num_var], shade=True, color=custom_palette[1])
plt.title(f'{num_var.capitalize()} KDE')
plt.xlabel('Total Bill')
```

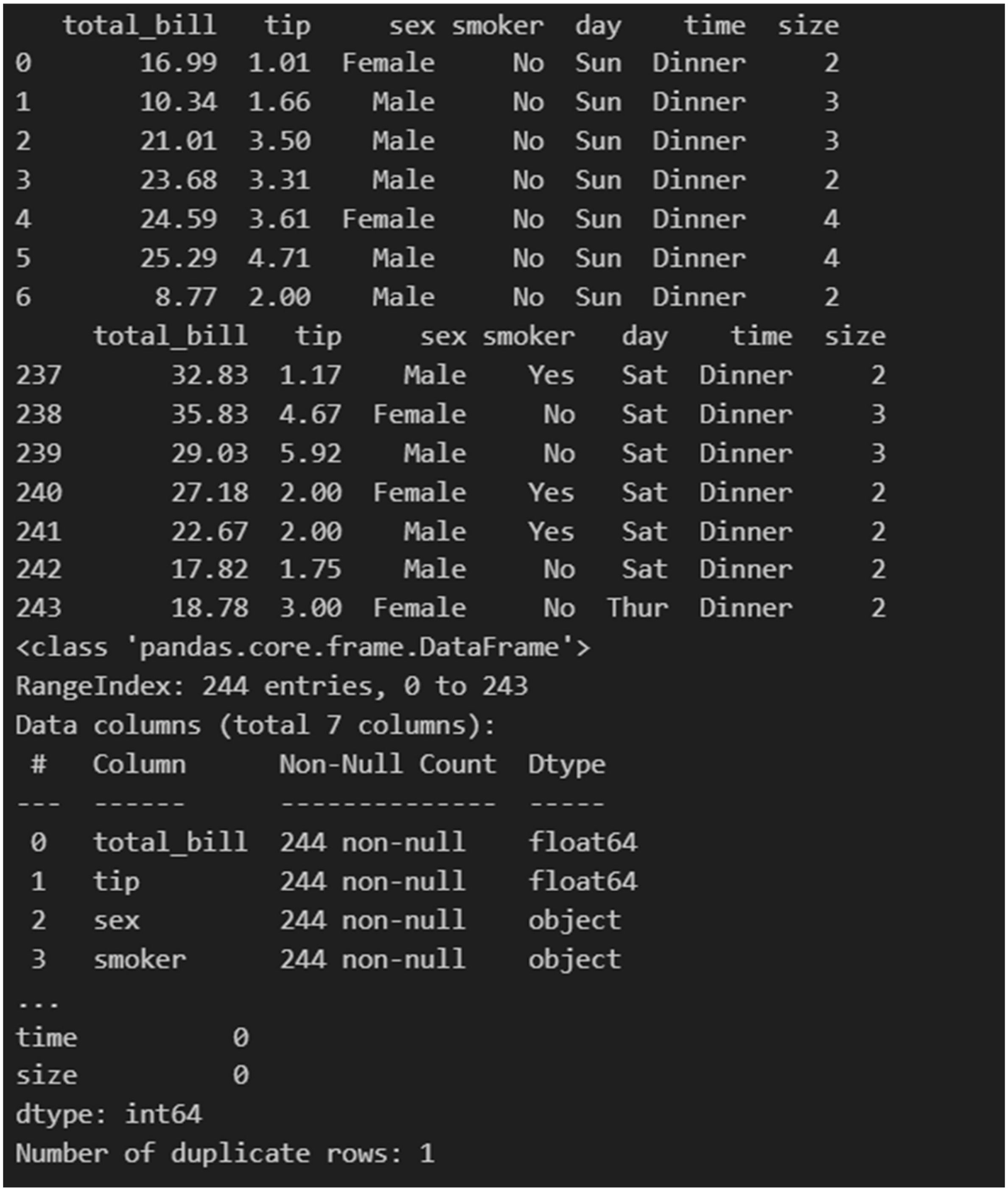


```python
plt.subplot(1,3,3)
sns.boxplot(x=df_clean[num_var], color=custom_palette[3])
plt.title(f'{num_var.capitalize()} Boxplot')
plt.xlabel('Total Bill')

plt.tight_layout()
plt.show()

# Categorical variable: ‘day’
print(df_clean['day'].value_counts())
print(f"Most frequent day: {df_clean['day'].mode().iloc[0]}")

plt.figure(figsize=(8,4))
sns.countplot(x='day', data=df_clean, palette=custom_palette, edgecolor='black')
plt.title('Frequency of Bills by Day')
plt.xlabel('Day of Week')
plt.ylabel('Count of Bills')
plt.show()

# 6. Bivariate and Multivariate Analysis

# Scatterplot for total_bill vs tip with custom colors and formatting
plt.figure(figsize=(9,6))
sns.scatterplot(x='total_bill', y='tip', data=df_clean, color=custom_palette[4], edgecolor='black')
plt.title('Scatterplot of Total Bill vs Tip')
plt.xlabel('Total Bill')
plt.ylabel('Tip')
plt.show()

# Correlation heatmap - numeric columns only, with diverging palette
plt.figure(figsize=(10,7))
corr_matrix = df_clean.corr(numeric_only=True)
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.8, linecolor='white')
plt.title('Correlation Matrix')
plt.show()
```

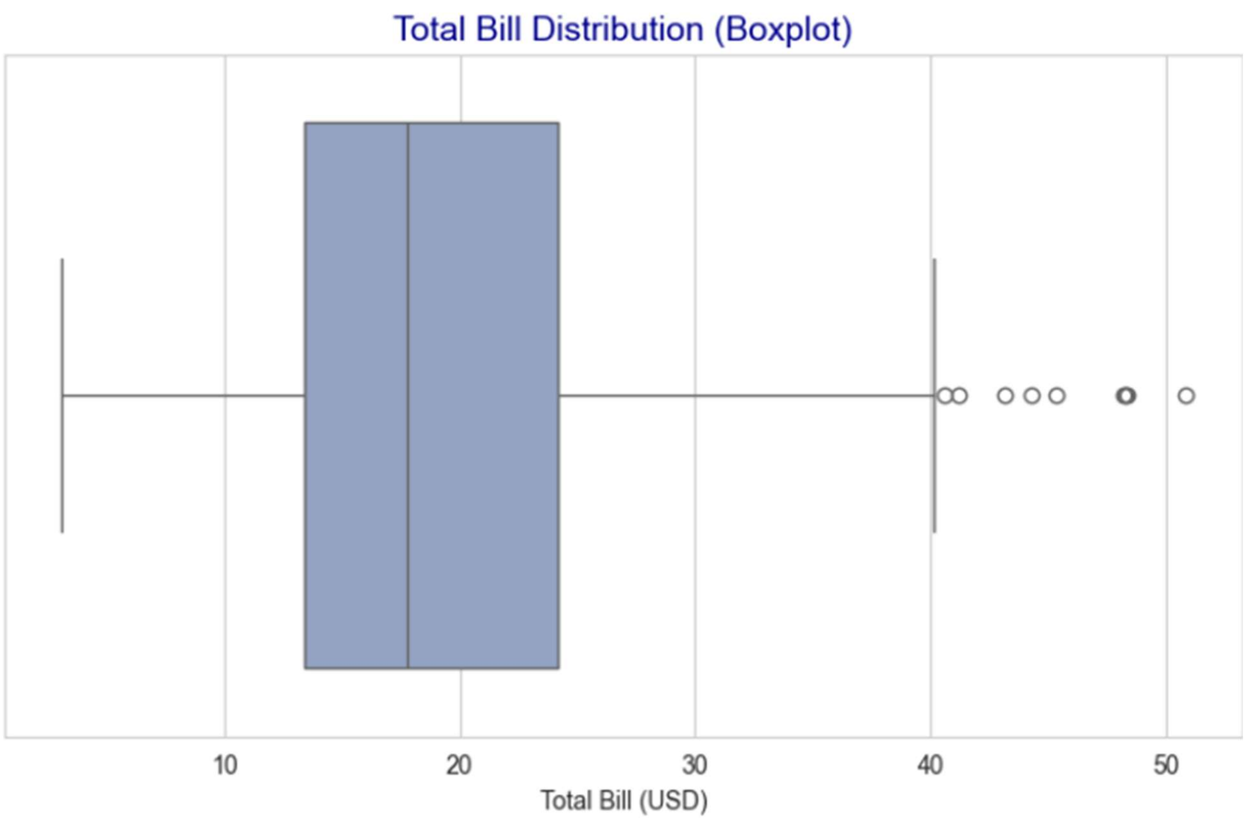


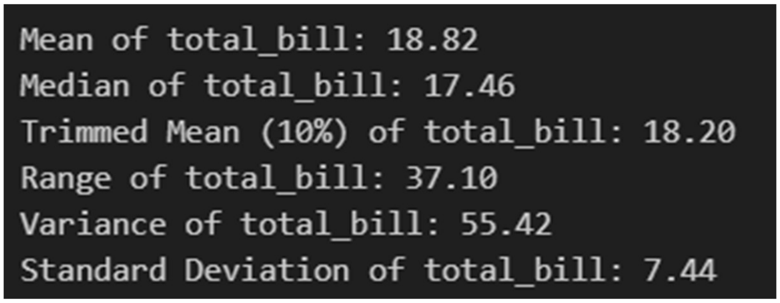
```python
# Boxplot grouped by day with palette variations
plt.figure(figsize=(10,6))
sns.boxplot(x='day', y='total_bill', data=df_clean, palette=custom_palette)
plt.title('Total Bill by Day of Week')
plt.xlabel('Day')
plt.ylabel('Total Bill')
plt.show()
```

Output:

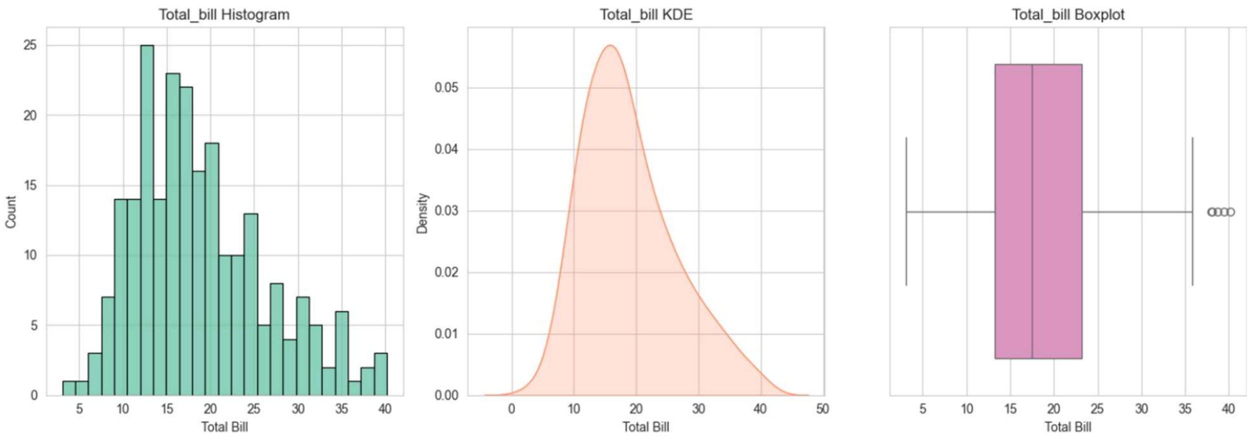


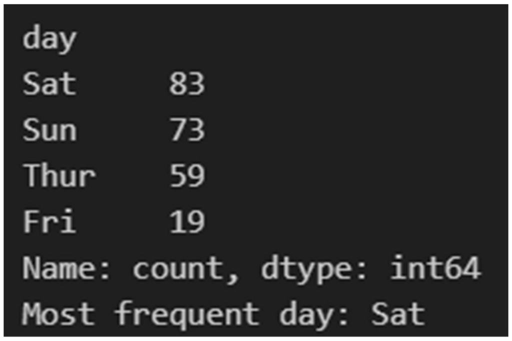
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
5       25.29  4.71    Male     No  Sun  Dinner     4
6        8.77  2.00    Male     No  Sun  Dinner     2
     total_bill   tip     sex smoker   day    time  size
237       32.83  1.17    Male    Yes   Sat  Dinner     2
238       35.83  4.67  Female     No   Sat  Dinner     3
239       29.03  5.92    Male     No   Sat  Dinner     3
240       27.18  2.00  Female    Yes   Sat  Dinner     2
241       22.67  2.00    Male    Yes   Sat  Dinner     2
242       17.82  1.75    Male     No   Sat  Dinner     2
243       18.78  3.00  Female     No  Thur  Dinner     2
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   total_bill  244 non-null    float64
 1   tip         244 non-null    float64
 2   sex         244 non-null    object
 3   smoker      244 non-null    object
...
time          0
size          0
dtype: int64
Number of duplicate rows: 1
```



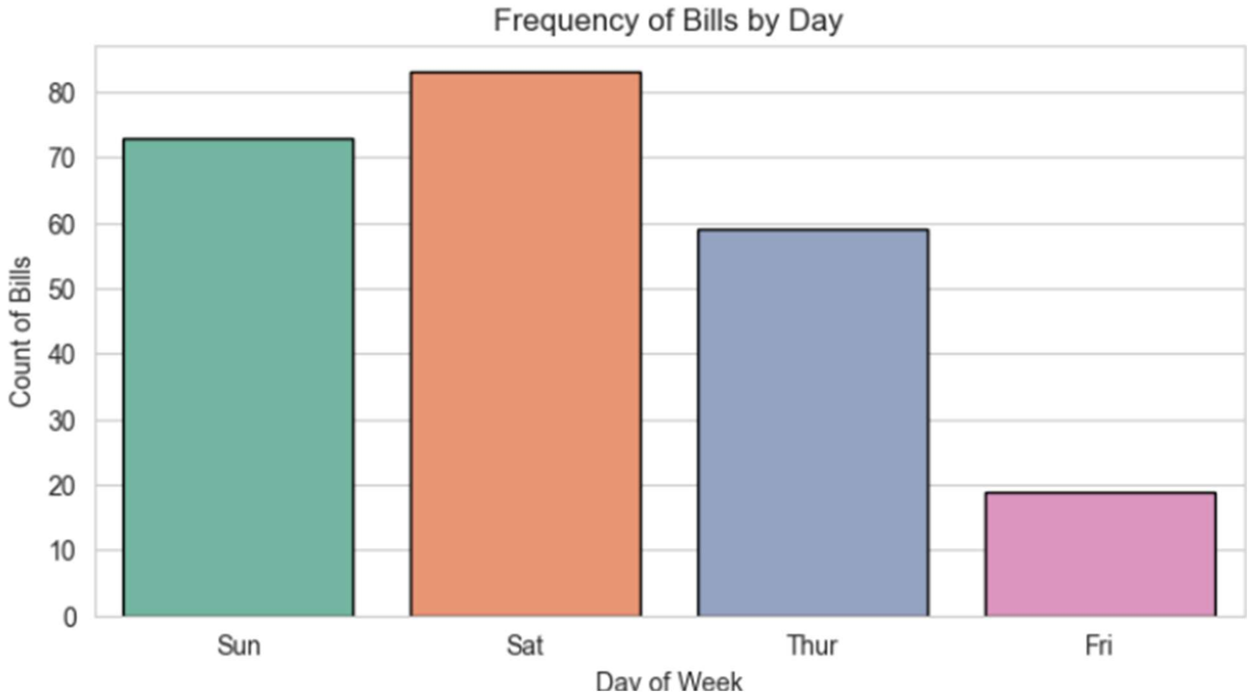


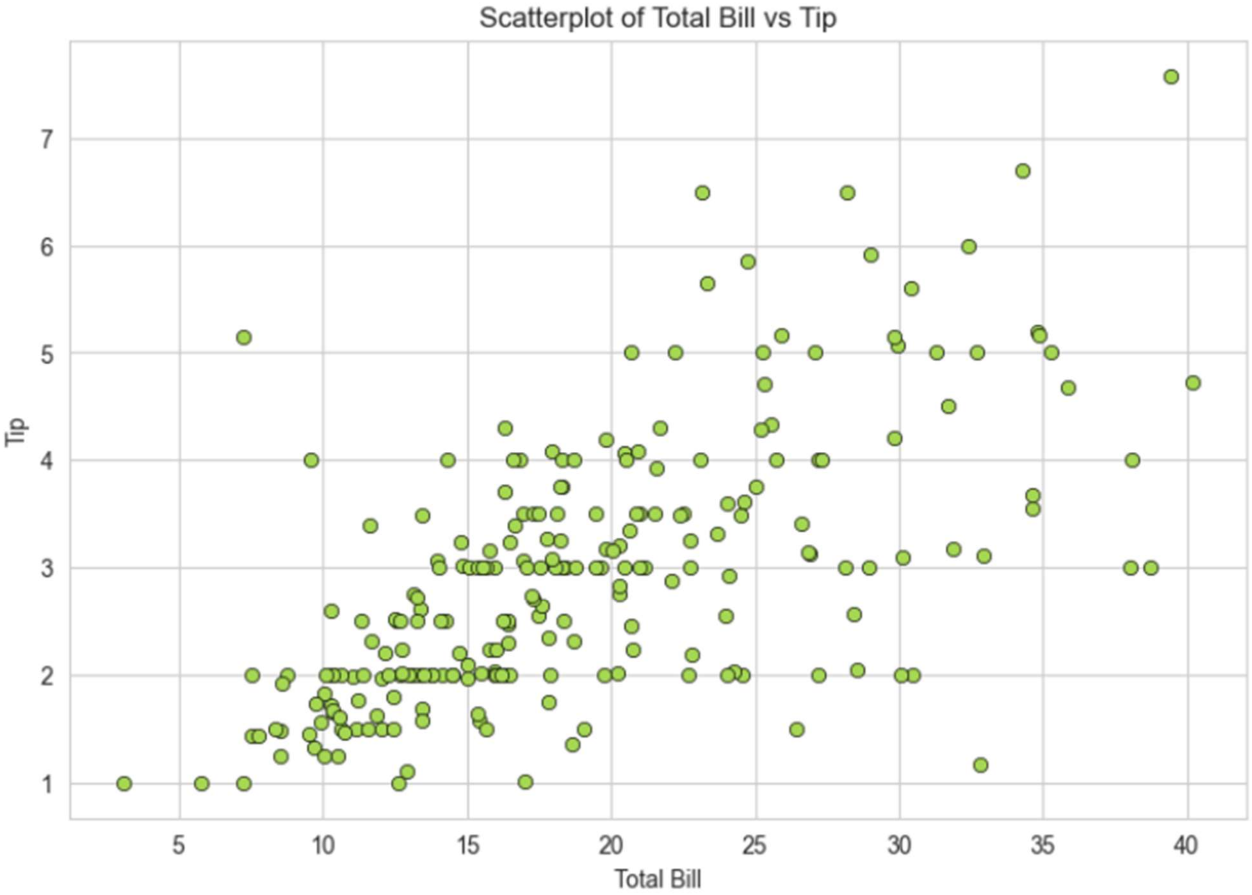
```
Mean of total_bill: 18.82
Median of total_bill: 17.46
Trimmed Mean (10%) of total_bill: 18.20
Range of total_bill: 37.10
Variance of total_bill: 55.42
Standard Deviation of total_bill: 7.44
```

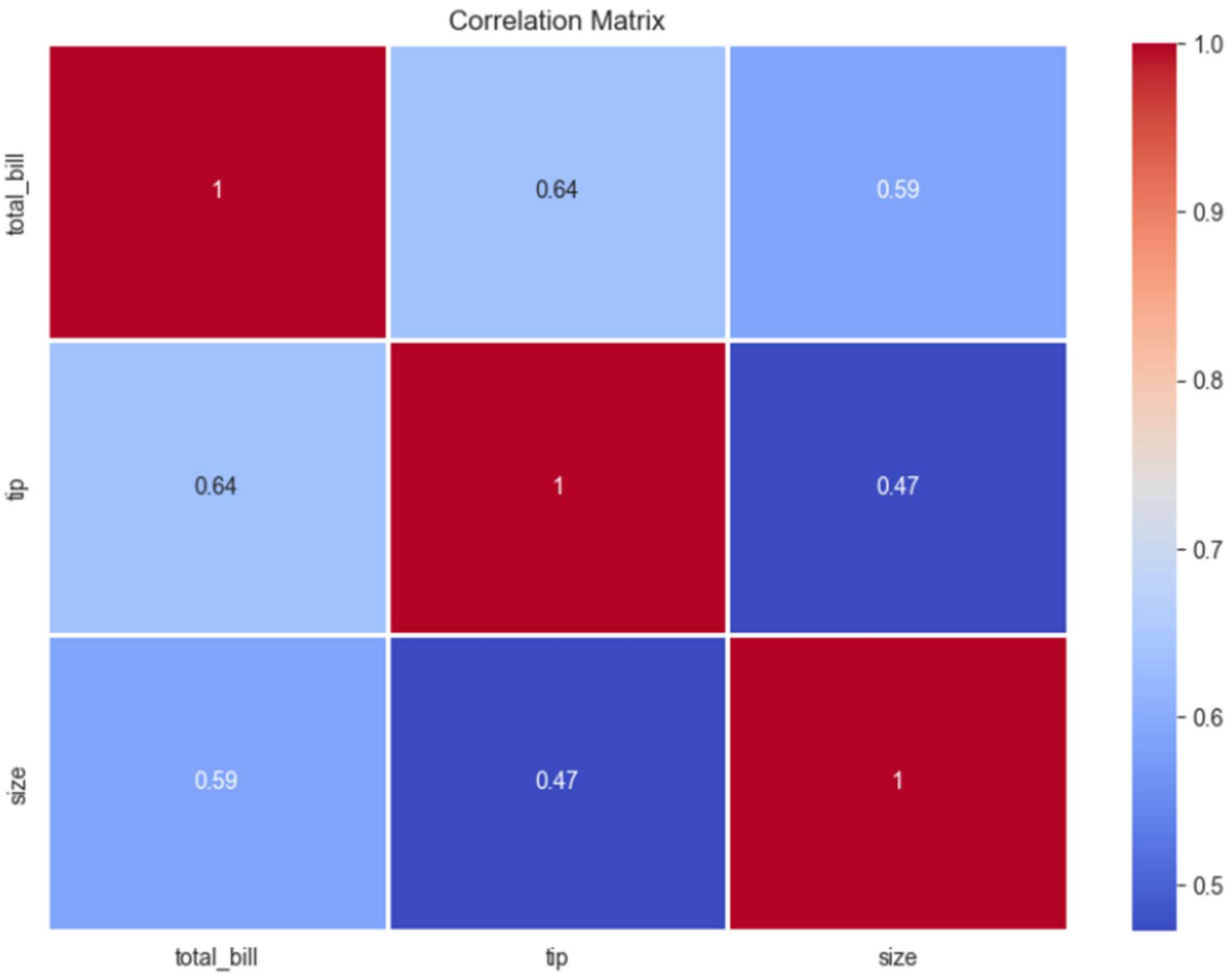


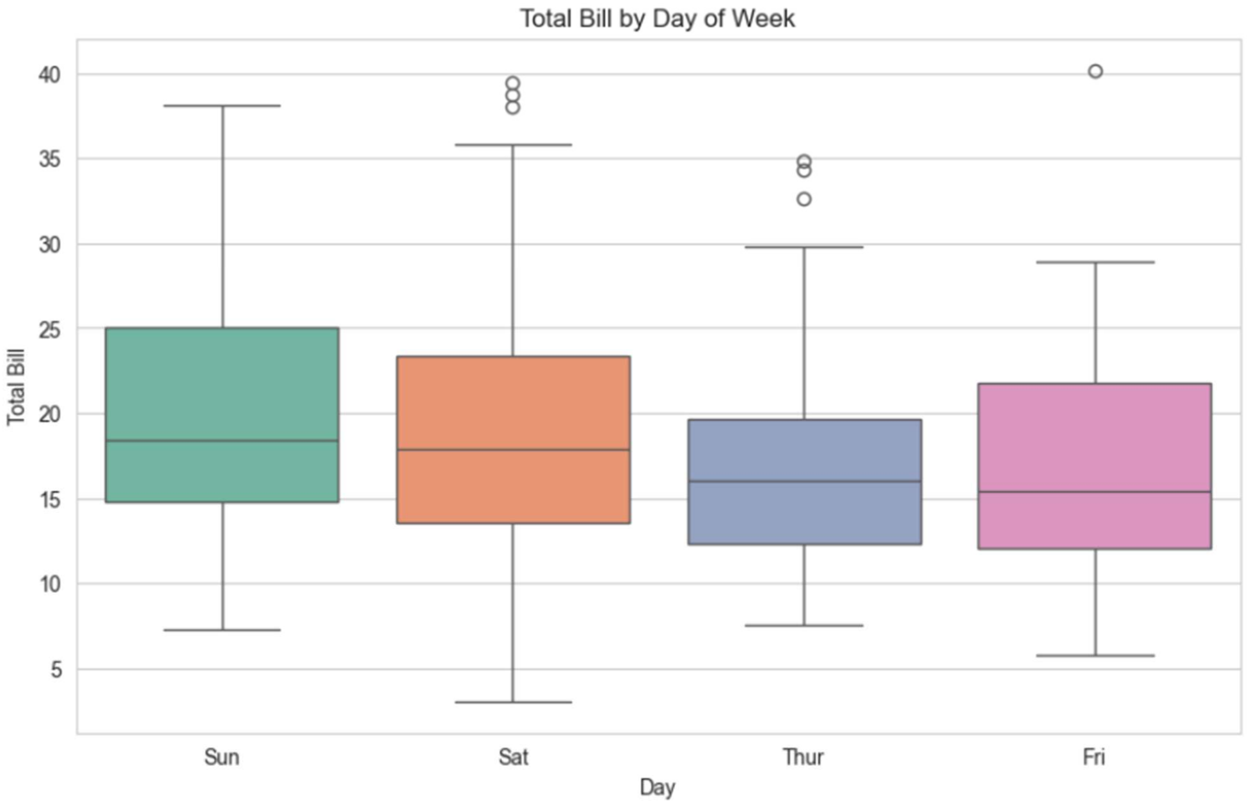


```
day
Sat     83
Sun     73
Thur    59
Fri     19
Name: count, dtype: int64
Most frequent day: Sat
```




Scatterplot of Total Bill vs Tip
Tip
1
2
3
4
5
6
7
Total Bill
5
10
15
20
25
30
35
40


Correlation Matrix
total_bill
tip
size
1
0.64
0.59
0.64
1
0.47
0.59
0.47
1
total_bill
tip
size
1.0
0.9
0.8
0.7
0.6
0.5


Total Bill by Day of Week
Total Bill
40
35
30
25
20
15
10
5
Sun
Sat
Thur
Fri
Day