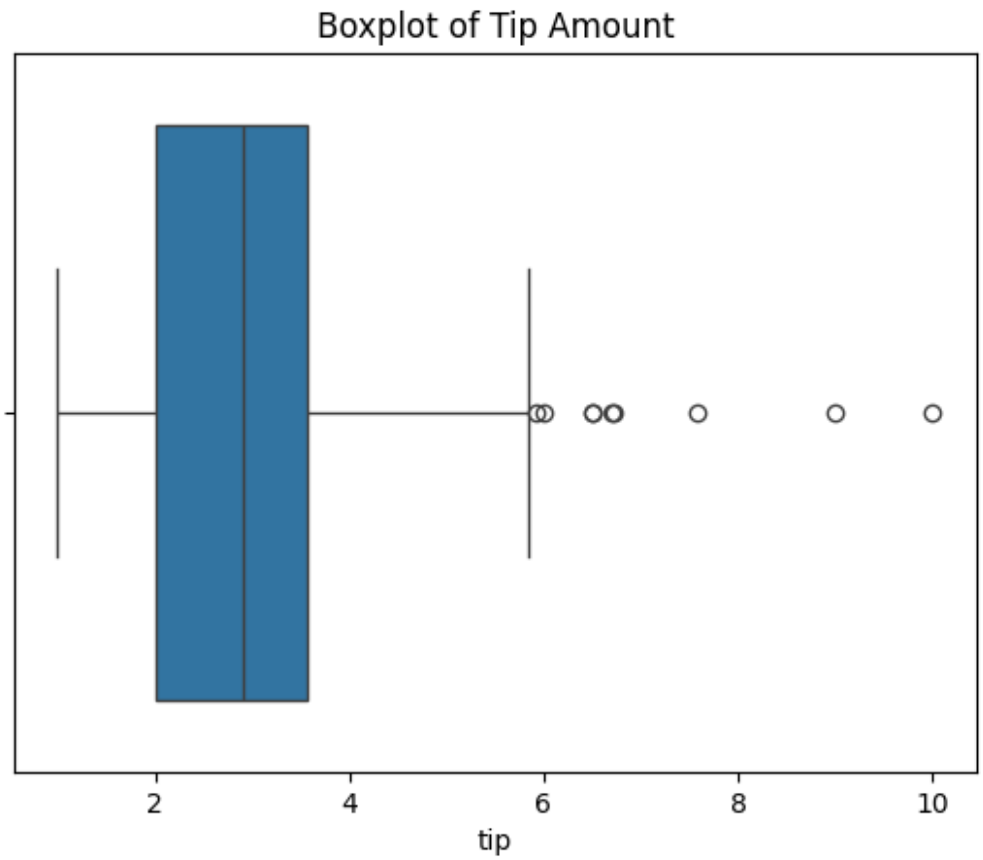
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Step 1: Understanding the Dataset
# Load the dataset
data = pd.read_csv('/Data set.csv')

# Display the first few rows of the dataset
print(data.head())
```

```
   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
```

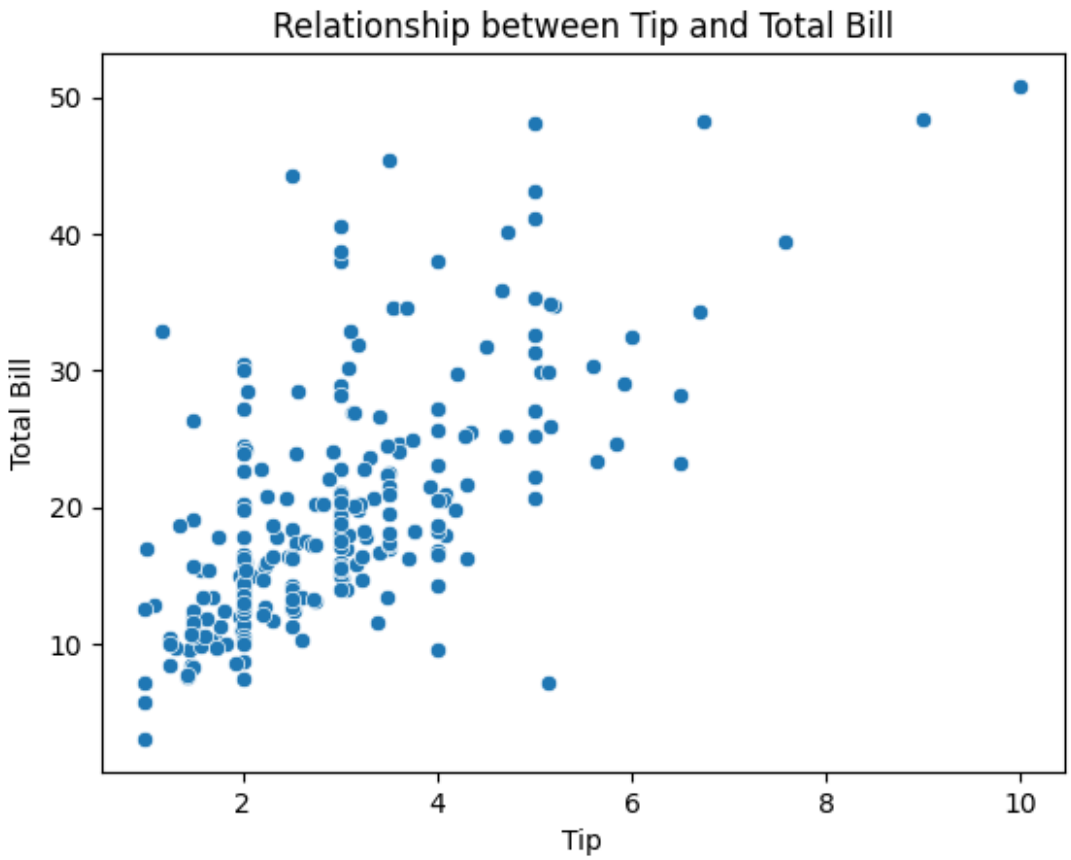
```python
# Check for missing values
print("Missing values:")
print(data.isnull().sum())
```

```
Missing values:
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

```python
# Check for outliers
sns.boxplot(x=data['tip'])
plt.title("Boxplot of Tip Amount")
plt.show()
```




```
# Step 2: Data Visualization
# Scatter plot of 'tip' against 'total_bill'
sns.scatterplot(data=data, x='tip', y='total_bill')
plt.title("Relationship between Tip and Total Bill")
plt.xlabel("Tip")
plt.ylabel("Total Bill")
plt.show()
```




```
# Step 3: Model Building
# Selecting independent and dependent variables
X = data['tip'].values.reshape(-1, 1)  # Independent variable (tip)
y = data['total_bill'].values  # Dependent variable (total bill)

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Creating and training the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

```
# Step 4: Model Evaluation
# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 41.253481147483996
R-squared: 0.5134545396054382
```

```
# Compare predicted vs actual values
results = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(results.head())
```

```
   Actual  Predicted
0   19.82  20.589182
1    8.77  15.835024
2   24.55  15.835024
3   25.89  28.566497
4   13.00  15.835024
```

```
# Plotting the regression line
plt.scatter(X_test, y_test, color='black')
plt.plot(X_test, y_pred, color='blue', linewidth=3)
plt.xlabel('Tip')
plt.ylabel('Total Bill')
plt.title('Simple Linear Regression')
plt.show()
```