

**AI ML(MCA 475)**

**Lab Assignment-05**

**BY**

**Himanshu Heda (24225013)**

**SUBMITTED TO**

**Prof. ManjulaShanbhog**

**SCHOOL OF SCIENCES**

**2024-2025**

# Multiple Regression on Tips Dataset

Libraries Used

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

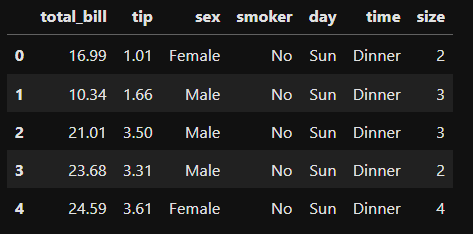
from sklearn.metrics import r2\_score,mean\_squared\_error

from scipy import stats

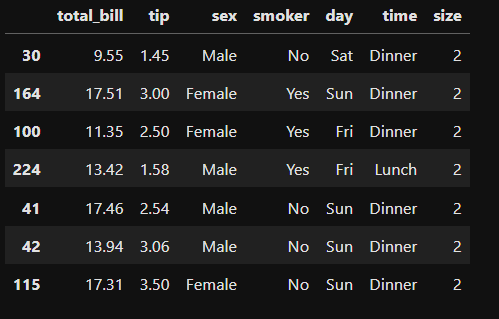
df=pd.read\_csv("C:/Users/samdc/OneDrive/Desktop/AI ML/tips - tips.csv")

# Exploratory Dataset Analysis

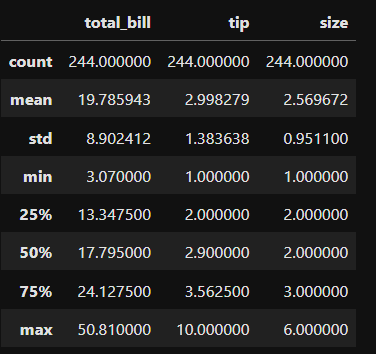
df.head()



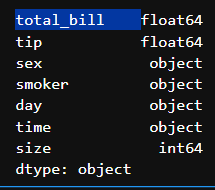
df.sample(7)



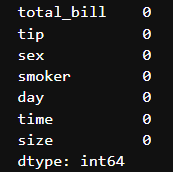
df.describe()



df.dtypes



df.isnull().sum()



**Visualization**

**plt.figure(figsize=(8, 5))**

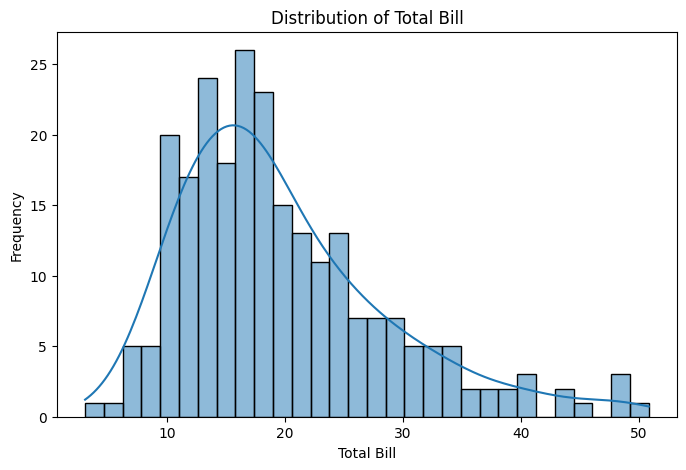
sns.histplot(df['total\_bill'], bins=30, kde=True)

plt.title('Distribution of Total Bill')

plt.xlabel('Total Bill')

plt.ylabel('Frequency')

plt.show()



**plt.figure(figsize=(8, 5))**

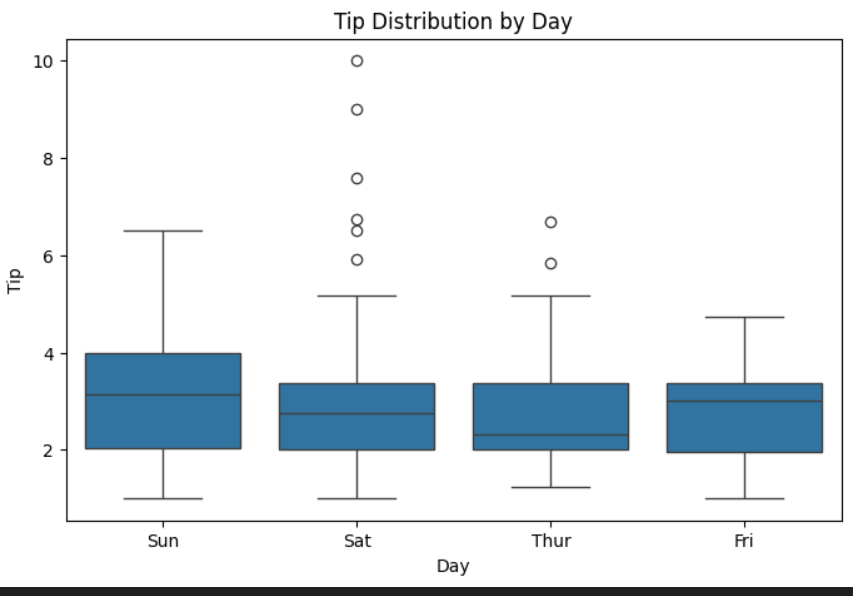
sns.boxplot(x='day', y='tip', data=df)

plt.title('Tip Distribution by Day')

plt.xlabel('Day')

plt.ylabel('Tip')

plt.show()

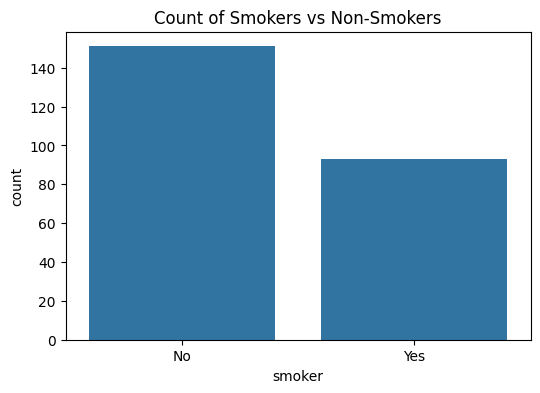


**plt.figure(figsize=(6, 4))**

sns.countplot(x='smoker', data=df)

plt.title('Count of Smokers vs Non-Smokers')

plt.show()



**plt.figure(figsize=(8, 5))**

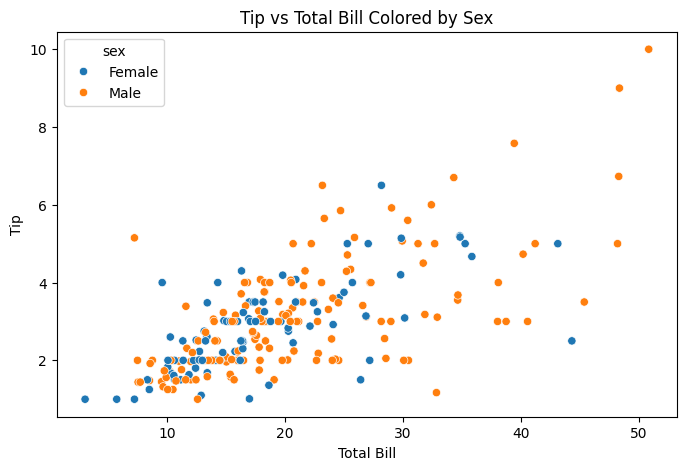
sns.scatterplot(x='total\_bill', y='tip', hue='sex', data=df)

plt.title('Tip vs Total Bill Colored by Sex')

plt.xlabel('Total Bill')

plt.ylabel('Tip')

plt.show()



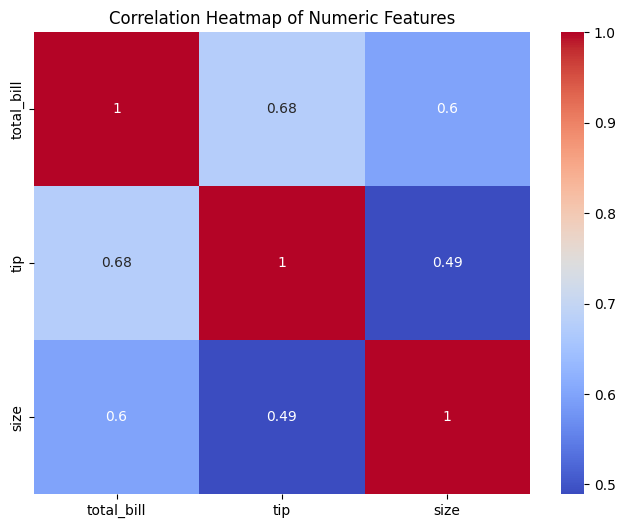
**numeric\_df = df.select\_dtypes(include=['float64', 'int64'])  # Select only numeric columns**

plt.figure(figsize=(8, 6))

sns.heatmap(numeric\_df.corr(), annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap of Numeric Features')

plt.show()



**Outlier detection (IQR and Z-score) for numerical columns**

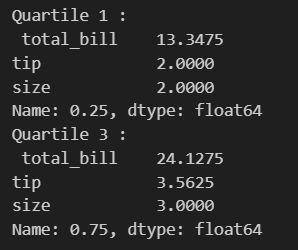
num\_cols=df.select\_dtypes(include=[np.number]).columns

Q1=df[num\_cols].quantile(0.25)

Q3=df[num\_cols].quantile(0.75)

print("Quartile 1 :\n",Q1)

print("Quartile 3 :\n",Q3)



IQR=Q3-Q1

iqr\_outlier\_mask = (df[num\_cols] < (Q1 - 1.5 \* IQR)) | (df[num\_cols] > (Q3 + 1.5 \* IQR))

iqr\_outlier=((df[num\_cols]<Q1-(1.5\*IQR)) | (df[num\_cols]>Q3 +(1.5\*IQR))).sum()

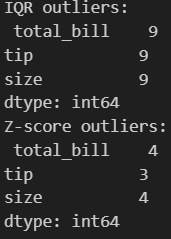
z = np.abs(stats.zscore(df[num\_cols]))

zscore\_outlier\_mask = z > 3

outlier\_zscore = (z > 3).sum(axis=0)

print("IQR outliers:\n", iqr\_outlier)

print("Z-score outliers:\n", outlier\_zscore)



for col in num\_cols:

    plt.figure(figsize=(10,4))

    plt.subplot(1,2,1)

    sns.histplot(df[col], kde=True)

    plt.title(f'Histogram: {col}')

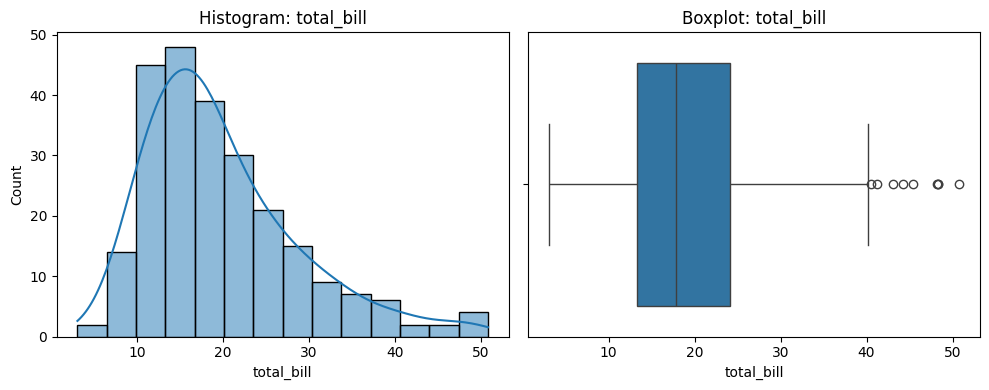
    plt.subplot(1,2,2)

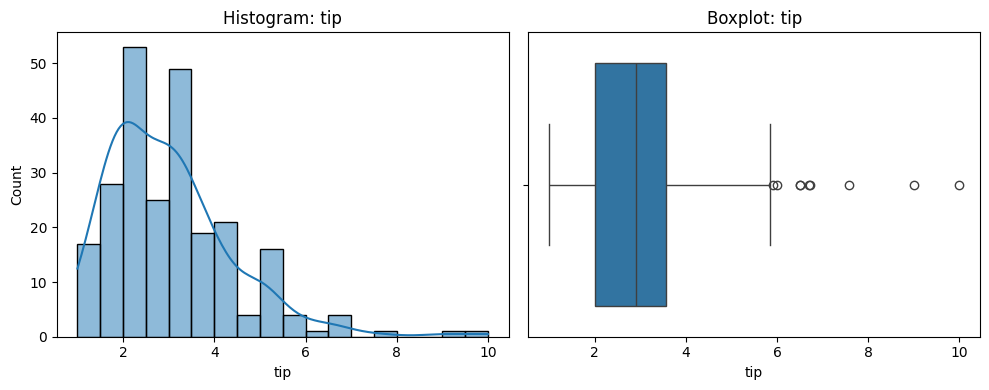
    sns.boxplot(x=df[col])

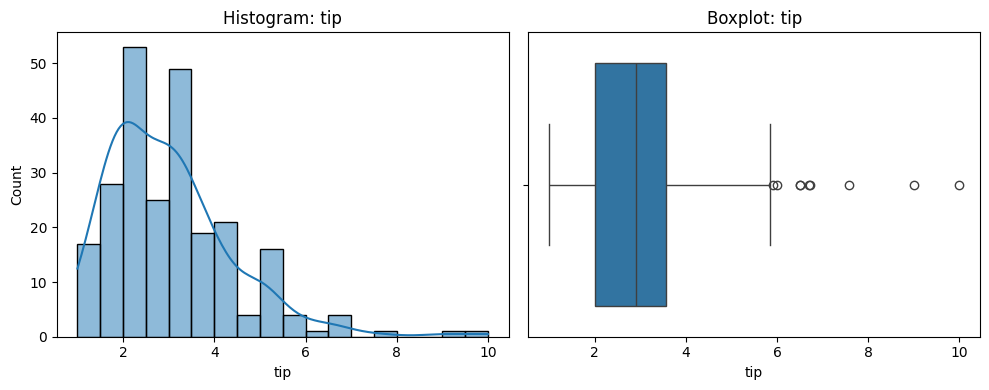
    plt.title(f'Boxplot: {col}')

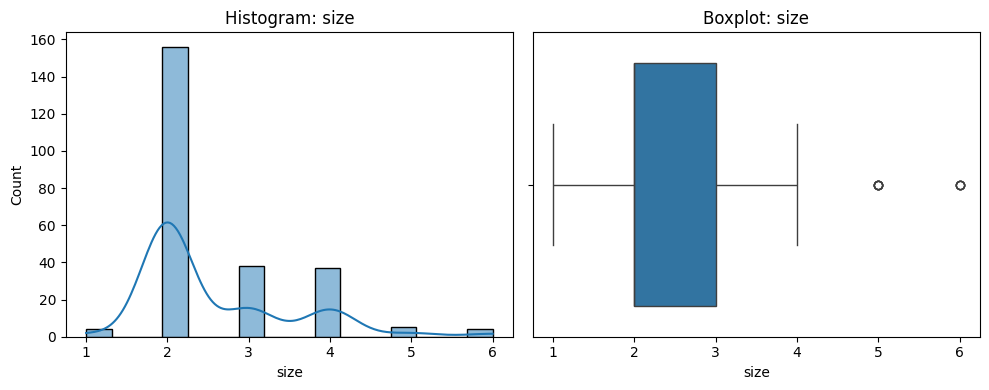
    plt.tight\_layout()

    plt.show()









# Combine masks: rows with outliers by either method

combined\_outlier\_mask = iqr\_outlier\_mask.any(axis=1) | zscore\_outlier\_mask.any(axis=1)

# Create cleaned dataset by removing outlier rows

df\_cleaned = df[~combined\_outlier\_mask].copy()

# Check shapes before and after

print("Original shape:", df.shape)

print("Cleaned shape:", df\_cleaned.shape)



## Multilinear regression

X = pd.get\_dummies(df\_cleaned.drop('tip', axis=1), drop\_first=True)

y = df\_cleaned['tip']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

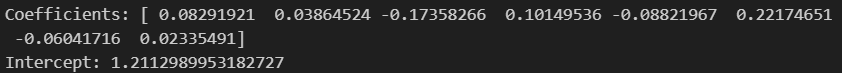
print("Mean Squared Error:", mse)

print("R-squared:", r2)



print("Coefficients:", model.coef\_)

print("Intercept:", model.intercept\_)



# Scatter plot: Actual vs Predicted

plt.figure(figsize=(8, 6))

sns.scatterplot(x=y\_test, y=y\_pred)

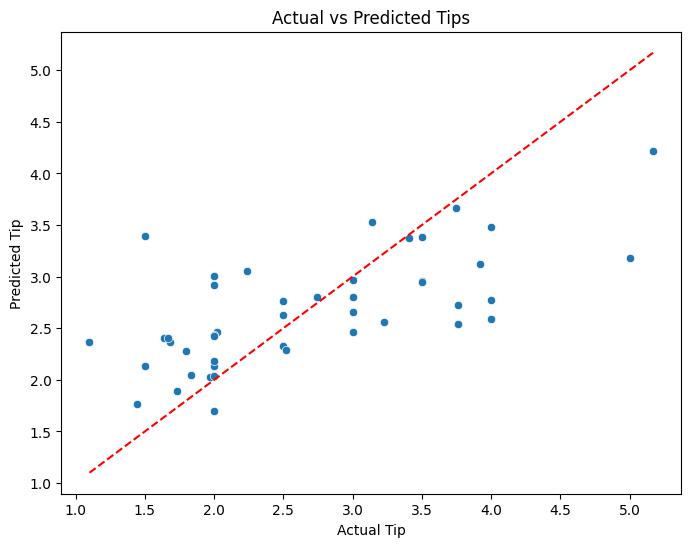
plt.xlabel('Actual Tip')

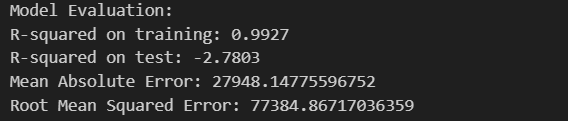
plt.ylabel('Predicted Tip')

plt.title('Actual vs Predicted Tips')

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'r--')  # Perfect fit line

plt.show()





# Histogram: Residuals

residuals = y\_test - y\_pred

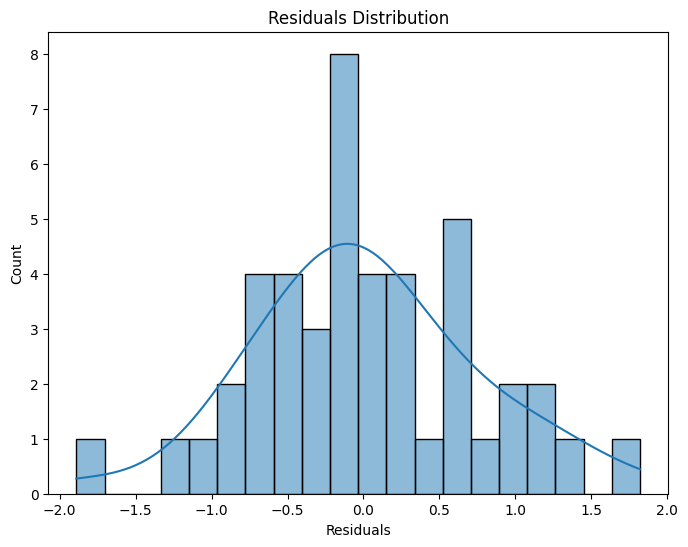
plt.figure(figsize=(8, 6))

sns.histplot(residuals, kde=True, bins=20)

plt.title('Residuals Distribution')

plt.xlabel('Residuals')

plt.show()



plt.plot(y\_test.values, label='Actual', color='red', linewidth=2)

plt.scatter(range(len(y\_pred)), y\_pred, label='Predicted', color='blue')

plt.title('Actual vs Predicted')

plt.xlabel('Sample Index')

plt.ylabel('Target Value')

plt.show()

