**DAC Phase 3:**

**Problem Statement:** Website Traffic Analysis

**Loading data:**

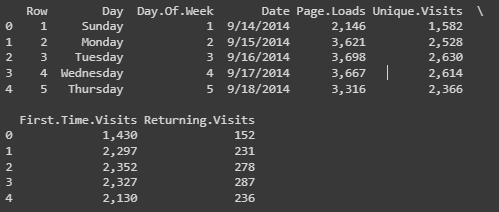
import pandas as pd

import matplotlib.pyplot as plt

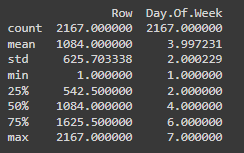
import seaborn as sns

data = pd.read\_csv('daily-website-visitors.csv')

print(data.head())

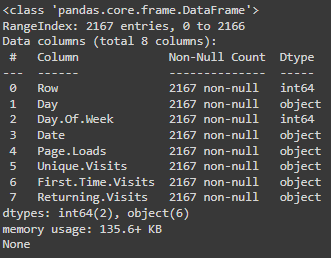


print(data.describe())



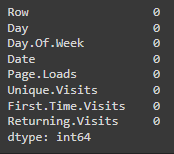
This command is used to view the brief summary of the dataset. We can see the mathematical parameters such as percentiles, standard deviation , mean, minimum and maximum values and count of each column.

print(data.info())



Info command is used check the datatype of every column and the count of each column. The difference between the describe() and info() is that describe command will give the mathematical parameters but info command will not give the mathematical parameters such as mean and standard deviation

data.isna().sum()



The above command is used to check for null values in each column. We can see that there are null values in the columns such as date,page.loads. It is very necessary to take action to clear the null values in the data set

**#Bar chart for Day of the Week and Unique Visits**

day\_of\_week\_counts = data['Day.Of.Week'].value\_counts().sort\_index()

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

day\_of\_week\_counts.plot(kind='bar')

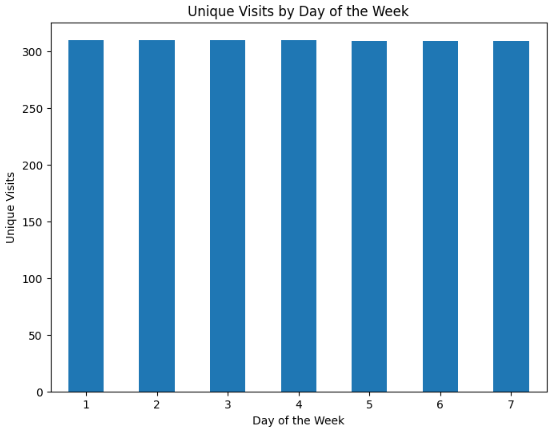
plt.title('Unique Visits by Day of the Week')

plt.xlabel('Day of the Week')

plt.ylabel('Unique Visits')

plt.xticks(rotation=0)

plt.show()



**INSIGHTS:**

Identify which day(s) of the week have the highest bar(s).Compare the bars corresponding to weekdays (Monday to Friday) with those of the weekend (Saturday and Sunday). Analyze the distribution of bars throughout the week.

**# Pie chart for Unique Visits by Day of the Week**

unique\_visits\_by\_day = data.groupby('Day.Of.Week')['Unique.Visits'].sum()

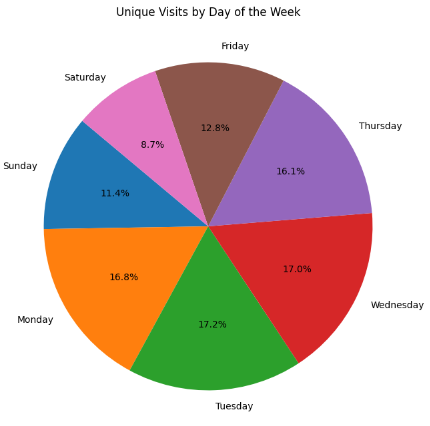
day\_labels = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']

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

plt.pie(unique\_visits\_by\_day, labels=day\_labels, autopct='%1.1f%%', startangle=140)

plt.title('Unique Visits by Day of the Week')

plt.show()



**INSIGHTS:**

The size of each slice in the pie chart represents the proportion of unique visits for each day of the week in relation to the total unique visits. This provides a clear visual indication of the relative popularity of each day. Pie charts can easily highlight outliers. If one day's slice significantly deviates from the others, it could indicate a special event, promotion, or anomaly that led to a surge or drop in traffic on that specific day.

**# Creating a pairplot for numeric columns**

columns\_to\_plot = ['Page.Loads', 'Unique.Visits', 'First.Time.Visits', 'Returning.Visits']

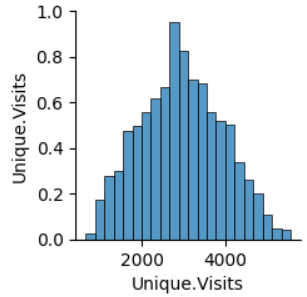
if all(col in data.columns for col in columns\_to\_plot):

    sns.pairplot(data[columns\_to\_plot])

    plt.show()

else:

    print("One or more specified columns do not exist in the dataset.")



**INSIGHTS:**

Diagonal elements of the pair plot show the distribution of individual variables. You can observe if any of the variables have a normal distribution or if they are skewed. Skewed distributions might need transformation for certain types of analyses. If there are multiple variables with strong correlations, it suggests multicollinearity.

**# Heatmap**

import seaborn as sns

data = pd.read\_csv('daily-website-visitors.csv')

data['Page.Loads'] = data['Page.Loads'].str.replace(',', '').astype(int)

data['Hour'] = pd.to\_datetime(data['Date']).dt.hour

traffic\_pivot = data.pivot\_table(index='Hour', columns='Day.Of.Week', values='Page.Loads', aggfunc='sum')

plt.figure(figsize=(10, 6))

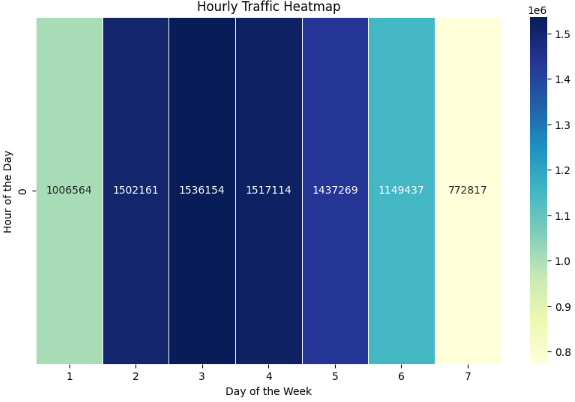
sns.heatmap(traffic\_pivot, cmap='YlGnBu', linewidths=.5, annot=True, fmt='d')

plt.title('Hourly Traffic Heatmap')

plt.xlabel('Day of the Week')

plt.ylabel('Hour of the Day')

plt.show()



**INSIGHTS:**

The hours of the day and the corresponding days of the week with the highest intensity of color. These represent peak traffic hours when your website experiences the most visits. This information is valuable for understanding the busiest time slots.

**# Histogram**

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

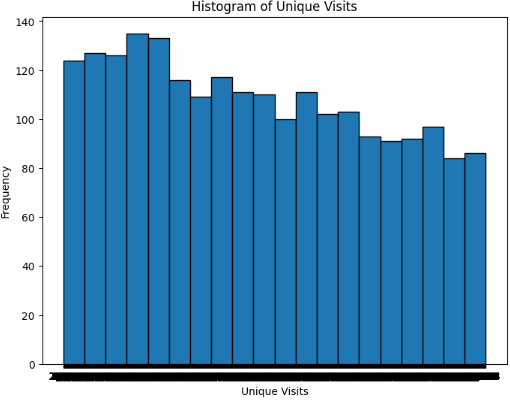
plt.hist(data['Unique.Visits'], bins=20, edgecolor='k')

plt.title('Histogram of Unique Visits')

plt.xlabel('Unique Visits')

plt.ylabel('Frequency')

plt.show()



**INSIGHTS:**

The histogram provides an overview of the distribution of unique visits on your website. You can observe the range of unique visits and how they are distributed across different values.A wider spread indicates a larger range of unique visits, suggesting variability in visitor numbers across different days.