WEB TRAFFIC ANALYTICS

WEB TRAFFIC ANALYTICS

Monitoring web traffic requires information such as the total number of visitors, average page views per visitor, most popular pages, average visits by visitors.

Duplicate data most often occurs during the data collection process. This typically happens when you combine data from multiple places, or receive data from clients or multiple departments.

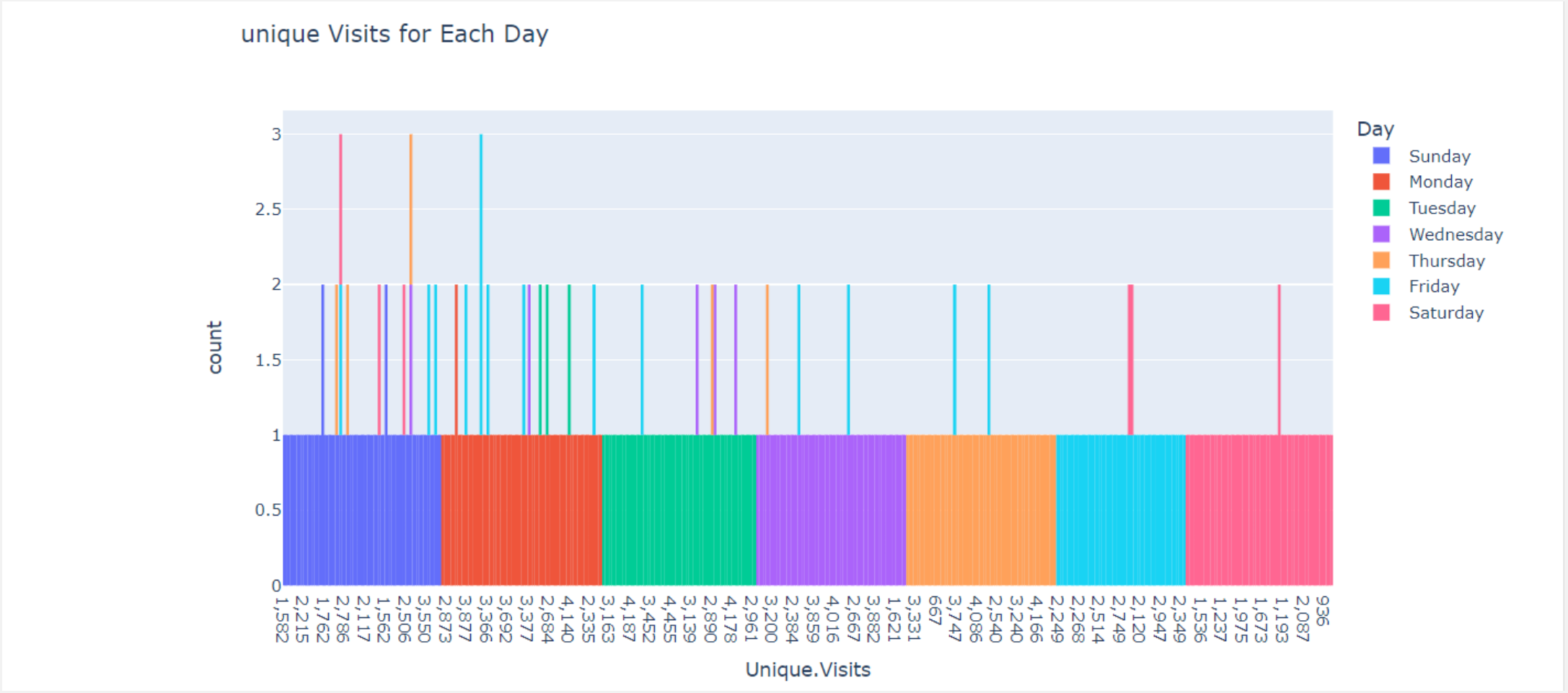
In [8]:
```
project.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2167 entries, 0 to 2166
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Row               2167 non-null   int64 
 1   Day               2167 non-null   object
 2   Day.Of.Week       2167 non-null   int64 
 3   Date              2167 non-null   object
 4   Page.Loads        2167 non-null   object
 5   Unique.Visits     2167 non-null   object
 6   First.Time.Visits  2167 non-null   object
 7   Returning.Visits  2167 non-null   object
dtypes: int64(2), object(6)
memory usage: 135.6+ KB
```

In [17]:
```
sns.barplot(project['Row'], project['Date'])
```

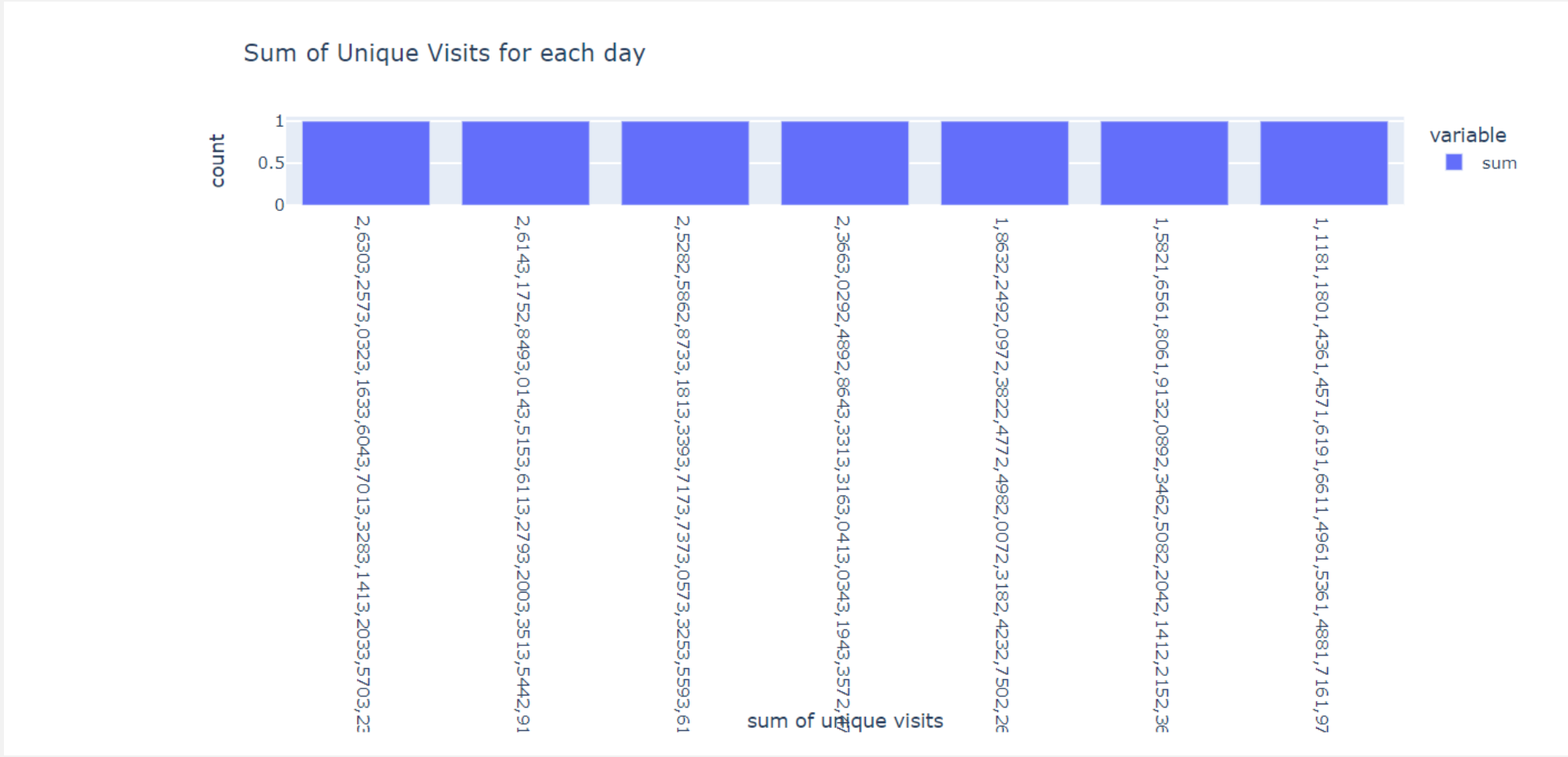
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('webtraffic-analysis.csv')

df = df.dropna()

df = df.drop_duplicates()

df.to_csv('webtraffic-analysis.csv',index=False)
```

```
import plotly.express as px

px.histogram(df,x='Unique.Visits',color='Day',title='unique Visits for Each Day')
```

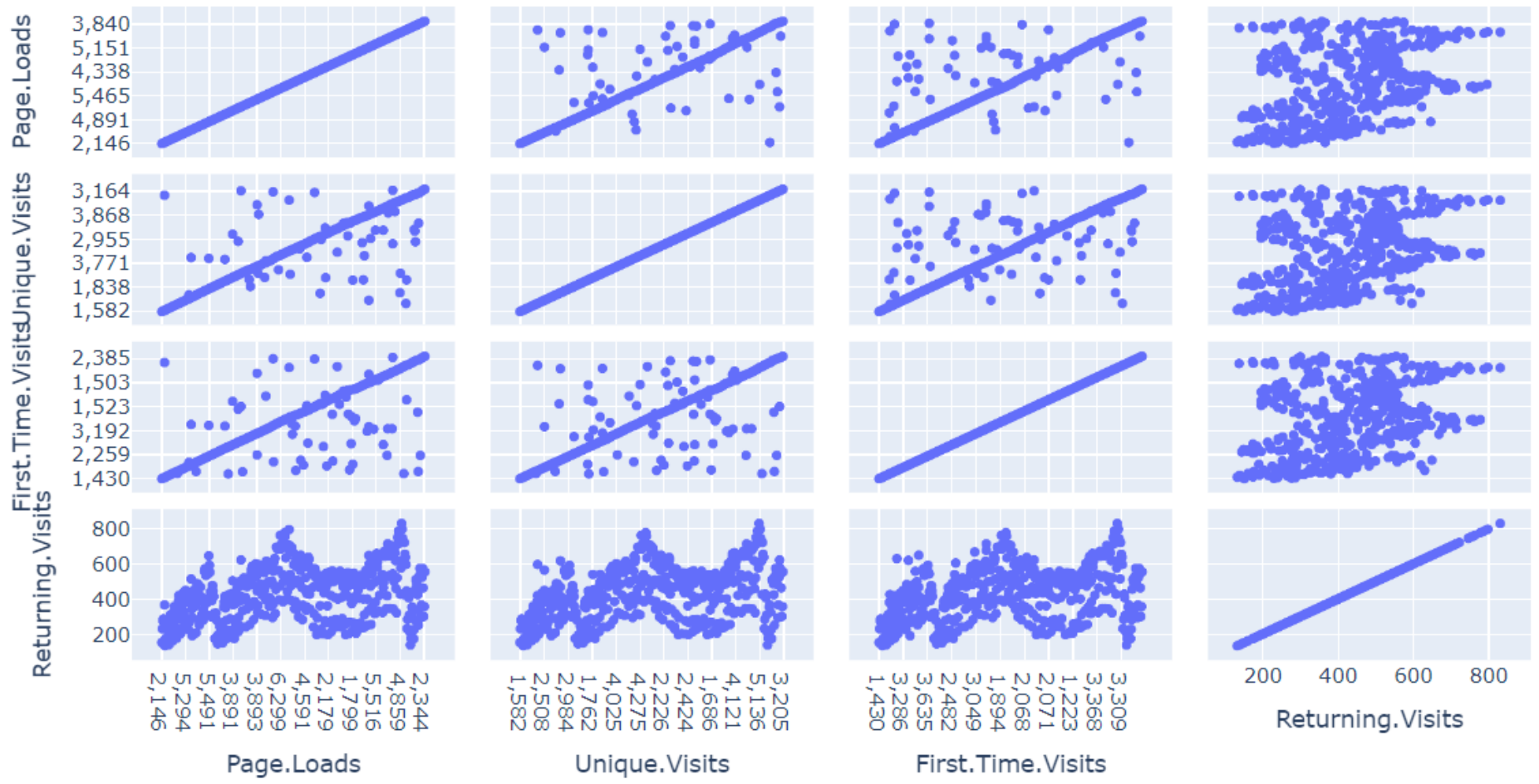


```
day_imp=df.groupby(['Day'])['Unique.Visits'].agg(['sum']).sort_values(by='sum',ascending=False)
px.bar(day_imp,labels={'value':'sum of unique visits'},title='Sum of Unique Visits for each day')

px.histogram(df,x='Date',y='Unique.Visits',color='Day',title='Sum of Unique visits for each day over Time')

px.density_heatmap(df,x='Date',y=['Page.Loads','Unique.visits','First.Time.Visit','Returning.Visits']
                   ,marginal_x="histogram",marginal_y="histogram")

px.scatter_matrix(df[['Page.Loads','Unique.Visits','First.Time.Visits','Returning.Visits']])

px.line(df,x='Data',y=['page.Loads','Unique.Visits','First.Time.Visits','Returning.Visits'],
      labels={'value':'Visits'}
      ,title='page Loads & visitors over Time')
```

```
day_imp=df.groupby(['Day'])['Unique.Visits'].agg(['sum']).sort_values(by='sum',ascending=False)
px.bar(day_imp,labels={'value':'sum of unique visits'},title='Sum of Unique Visits for each day')
```

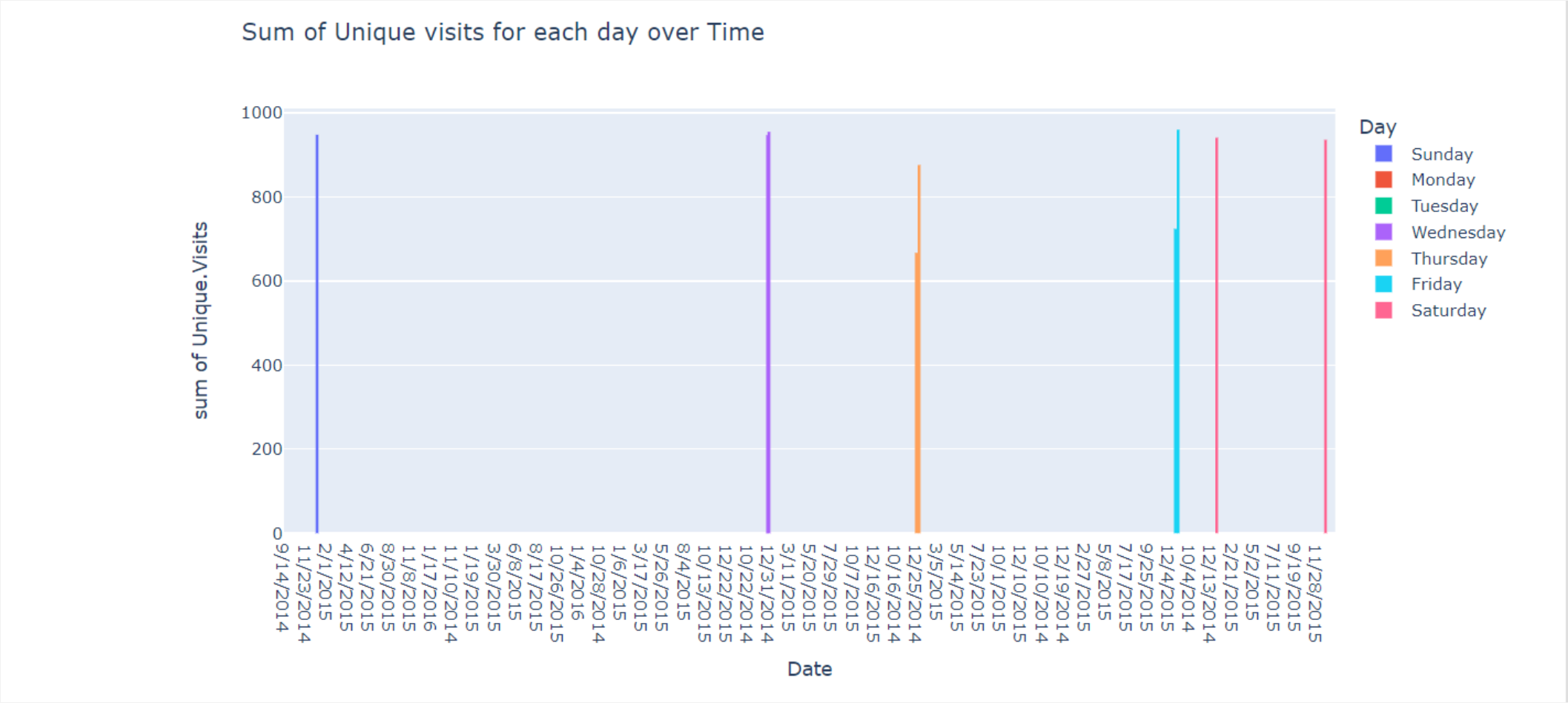


```
px.scatter_matrix(df[['Page.Loads','Unique.Visits','First.Time.Visits','Returning.Visits']])
```

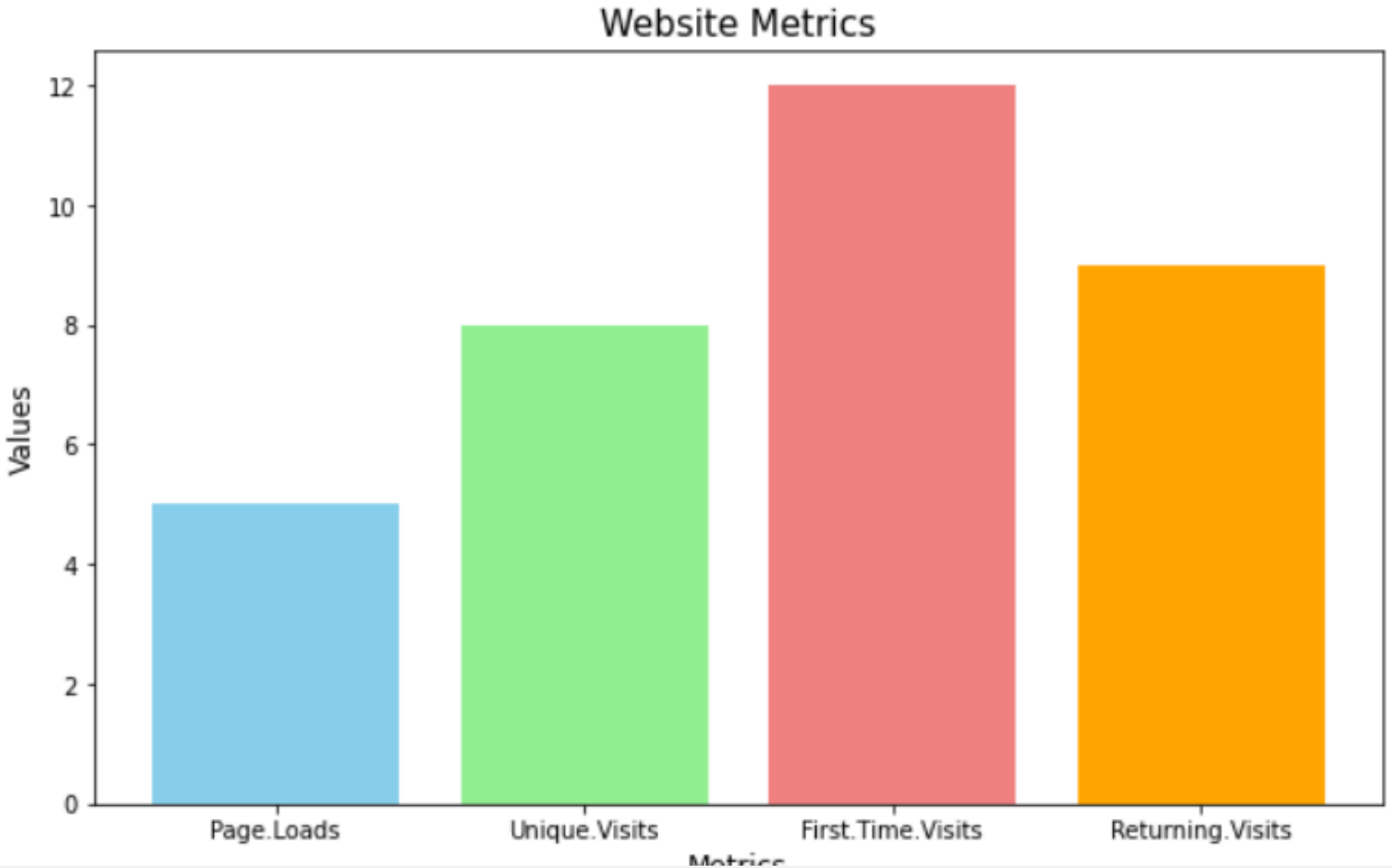


```
import matplotlib.pyplot as plt

# Data
categories = ['Page.Loads', 'Unique.Visits', 'First.Time.Visits', 'Returning.Visits']
values = [5, 10, 15, 20]

# Creating a scatter plot using Matplotlib
plt.figure(figsize=(8, 6))
plt.scatter(categories, values, color='b', marker='o')
plt.title('Website Metrics Scatter Plot', fontsize=15)
plt.xlabel('Metrics', fontsize=12)
plt.ylabel('Values', fontsize=12)
plt.show()
```

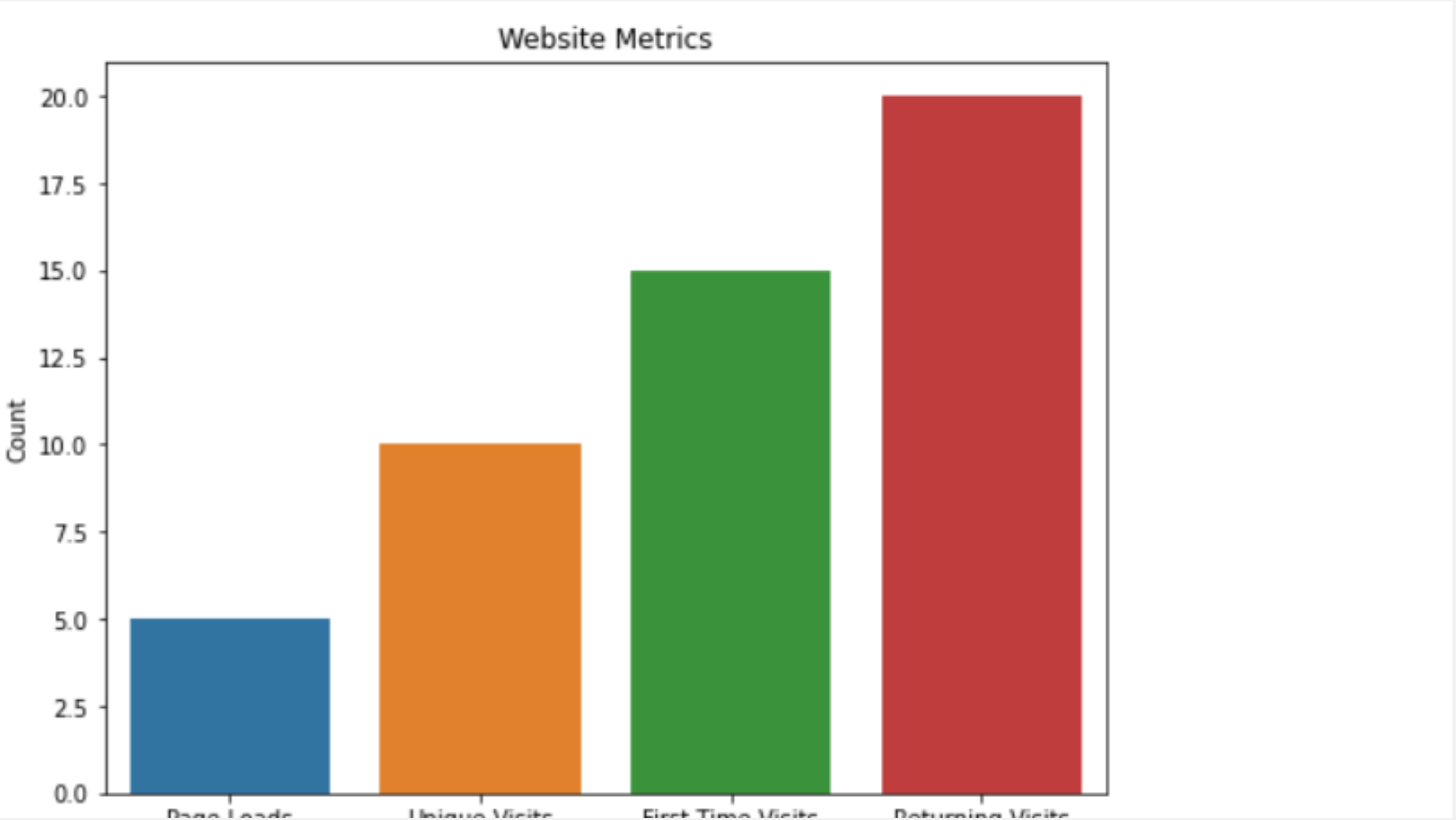
px.histogram(df,x='Date',y='Unique.Visits',color='Day',title='Sum of Unique visits for each day over Time')



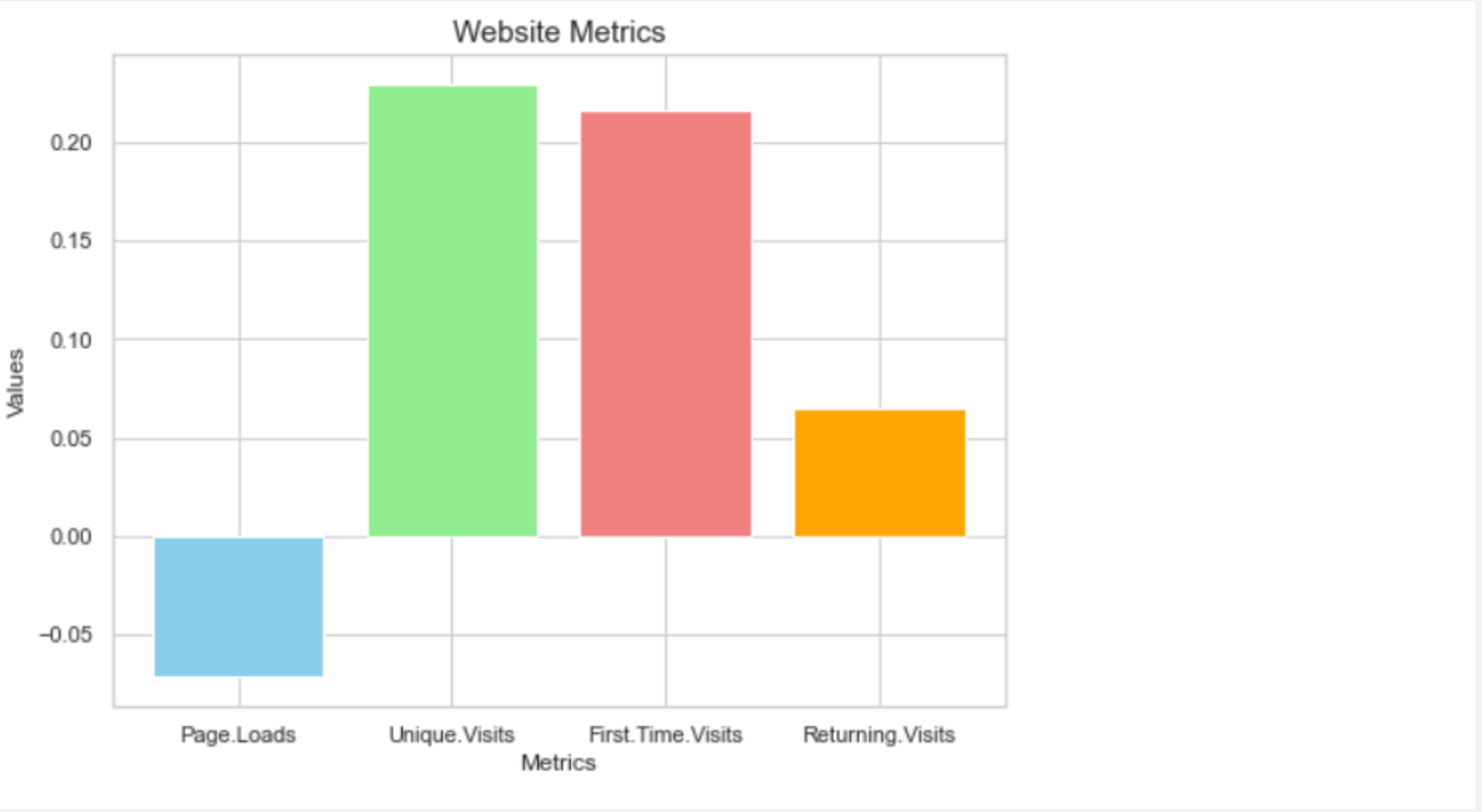
```
import plotly.express as px
px.histogram(df,x='Unique.Visits',color='Day',title='unique Visits for Each Day')


px.histogram(df,x='Date',y='Unique.Visits',color='Day',title='Sum of Unique visits for each day over Time')


import matplotlib.pyplot as plt

# Data
categories = ['Page.Loads', 'Unique.Visits', 'First.Time.Visits', 'Returning.Visits']
values = [5, 8, 12, 9]

# Creating a bar plot
plt.figure(figsize=(10,6))
plt.bar(categories, values, color=['skyblue', 'lightgreen', 'lightcoral', 'orange'])
plt.xlabel('Metrics', fontsize=12)
plt.ylabel('Values', fontsize=12)
plt.title('Website Metrics', fontsize=15)
plt.show()
```


Website Metrics
Values
0
2
4
6
8
10
12
Page.Loads
Unique.Visits
First.Time.Visits
Returning.Visits

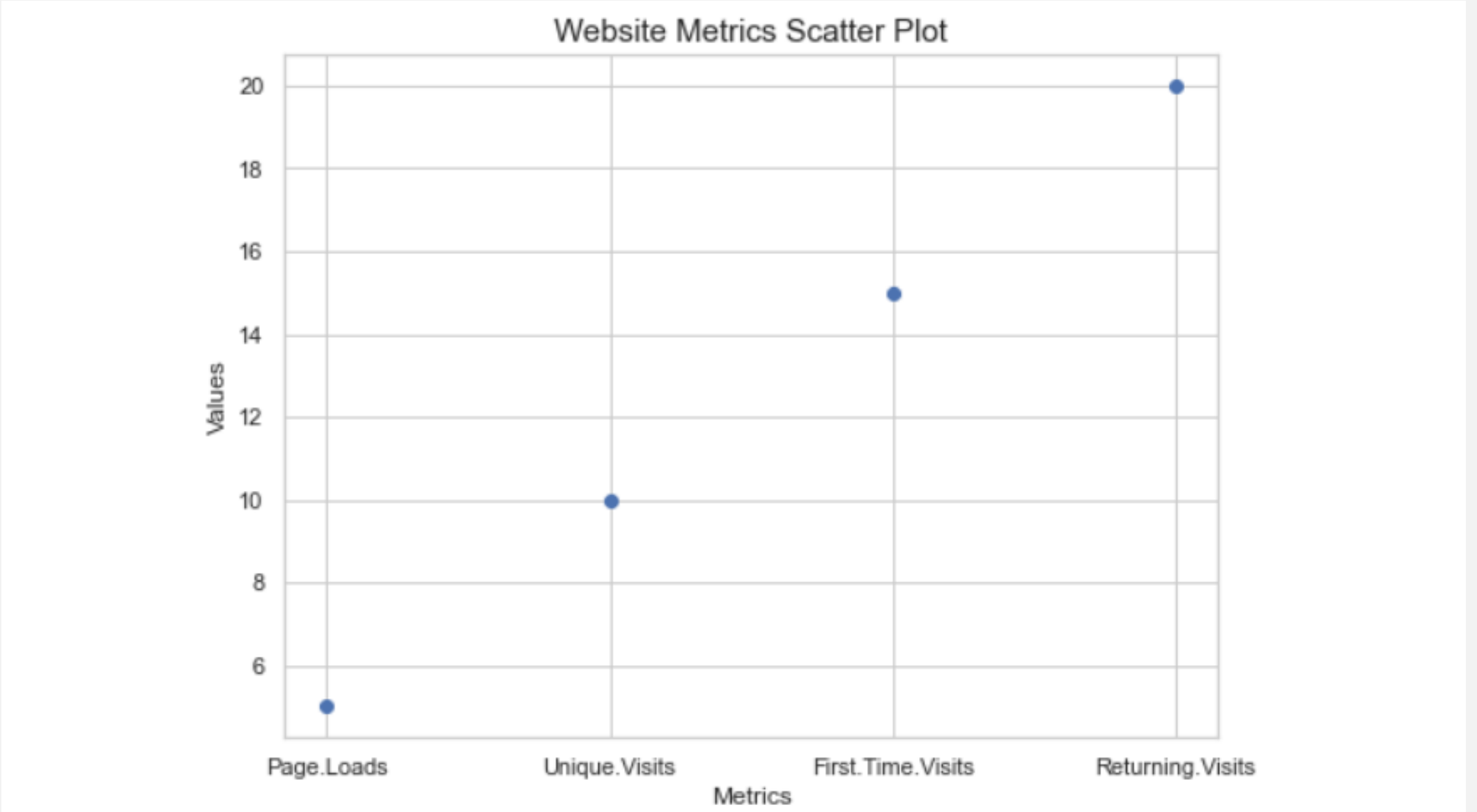
```
import seaborn as sns
import matplotlib.pyplot as plt

# Sample data
categories = ['Page.Loads', 'Unique.Visits', 'First.Time.Visits', 'Returning.Visits']
values = [5, 10, 15, 20]

# Create a bar plot using Seaborn
plt.figure(figsize=(8, 6))
sns.barplot(x=categories, y=values)
plt.title('Website Metrics')
plt.xlabel('Metrics')
plt.ylabel('Count')
plt.show()
```


Website Metrics
Count
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification

# Generating sample data using sklearn
data, _ = make_classification(n_samples=100, n_features=4, random_state=0)

# Assuming the data is represented by the variables 'Page.Loads', 'Unique.Visits', 'First.Time.Visits', and 'Returning.Visits'
categories = ['Page.Loads', 'Unique.Visits', 'First.Time.Visits', 'Returning.Visits']
values = data[0]  # Using the generated data for illustration purposes

# Creating a simple bar plot using Matplotlib
plt.figure(figsize=(8, 6))
plt.bar(categories, values, color=['skyblue', 'lightgreen', 'lightcoral', 'orange'])
plt.title('Website Metrics', fontsize=15)
plt.xlabel('Metrics', fontsize=12)
plt.ylabel('Values', fontsize=12)
plt.show()
```


Website Metrics
0.20
0.15
0.10
0.05
0.00
−0.05
Values
Page.Loads
Unique.Visits
First.Time.Visits
Returning.Visits
Metrics

```
import matplotlib.pyplot as plt

# Data
categories = ['Page.Loads', 'Unique.Visits', 'First.Time.Visits', 'Returning.Visits']
values = [5, 10, 15, 20]

# Creating a scatter plot using Matplotlib
plt.figure(figsize=(8, 6))
plt.scatter(categories, values, color='b', marker='o')
plt.title('Website Metrics Scatter Plot', fontsize=15)
plt.xlabel('Metrics', fontsize=12)
plt.ylabel('Values', fontsize=12)
plt.show()
```


Website Metrics Scatter Plot
20
18
16
14
12
10
8
6
Values
Page.Loads
Unique.Visits
First.Time.Visits
Returning.Visits
Metrics

THIS PROJECT IS PRESENTED BY:

Soalwin Thomas

Sri jaya shankaran.R

Thamizh Selvan.S

Soleswaran.N

Vinothan.V


THANK YOU!