

Analyzing Website Traffic Data

***Project Title: Analyzing
Website Traffic Data***

Author: Sudeeksha Singh

Date: 11 March 2025

Institution: KIET

Introduction

Understanding website traffic is crucial for improving user engagement, marketing strategies, and overall performance. This project aims to analyze website traffic data to identify trends, peak visiting hours, and user behavior. The insights gained will help improve content strategy and enhance user experience.

Objectives

- Identify peak traffic periods.
- Understand user demographics and behavior.
- Visualize key metrics for clear insights.

Methodology

Data Collection

The data was collected from Google Analytics reports, covering metrics like:

- Sessions
- Page Views
- Bounce Rate
- User Demographics (age, location, device)

Tools Used

- **Python** for data analysis.
- **Pandas** for data manipulation.
- **Matplotlib** and **Seaborn** for data visualization.
- **Jupyter Notebook** for code execution.

Code

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
traffic_data = pd.read_csv('/content/drive/MyDrive/traffic_data.csv')
```

```
# Display basic information about the dataset
```

```
print(traffic_data.info())
```

```
print(traffic_data.describe())
```

```
import pandas as pd
```

```
# Sample data
```

```
data = {'Date': ['01-01-2020', '01-02-2020', '01-03-2020', '01-04-2020', '01-05-2020'],
```

```
        'Visits': [1554.0, 2820.0, 2970.0, 2111.0, 2393.0]}
```

```
# Create DataFrame
```

```
traffic_data = pd.DataFrame(data)
```

```
# Convert 'Date' column to datetime format
traffic_data['Date'] = pd.to_datetime(traffic_data['Date'], format='%m-%d-%Y')

# Now 'Date' column can be used for time-based analysis
traffic_data.set_index('Date', inplace=True)

# Display the DataFrame to ensure everything is correct
print(traffic_data)

import pandas as pd
import matplotlib.pyplot as plt

# Sample data (replace this with your actual data)
data = {'Date': ['01-01-2020', '01-02-2020', '01-03-2020', '01-04-2020', '01-05-2020'],
        'Visits': [1554.0, 2820.0, 2970.0, 2111.0, 2393.0]}

# Create DataFrame
traffic_data = pd.DataFrame(data)

# Convert 'Date' column to datetime format
traffic_data['Date'] = pd.to_datetime(traffic_data['Date'], format='%m-%d-%Y')

# Set 'Date' as the index
traffic_data.set_index('Date', inplace=True)

# Plot the 'Visits' column (instead of 'page_views')
```

```
plt.figure(figsize=(12, 6))
plt.plot(traffic_data.index, traffic_data['Visits'], label='Visits')
plt.title('Website Traffic Over Time')
plt.xlabel('Date')
plt.ylabel('Visits')
plt.legend()
plt.show()
```

```
# Create DataFrame
```

```
traffic_data = pd.DataFrame(data)
```

```
# Convert 'Date' column to datetime format
```

```
traffic_data['Date'] = pd.to_datetime(traffic_data['Date'], format='%m-%d-%Y')
```

```
# Set 'Date' as the index
```

```
traffic_data.set_index('Date', inplace=True)
```

```
# Calculate the bounce rate (if it's not already calculated)
```

```
traffic_data['bounce_rate'] = (traffic_data['Bounces'] / traffic_data['Sessions'])
* 100
```

```
# Plot the bounce rate over time
```

```
plt.figure(figsize=(12, 6))
plt.plot(traffic_data.index, traffic_data['bounce_rate'], label='Bounce Rate',
color='red')
plt.title('Bounce Rate Over Time')
plt.xlabel('Date')
```

```
plt.ylabel('Bounce Rate (%)')
```

```
plt.legend()
```

```
plt.show()
```

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

```
plt.scatter(y_test, y_pred, label='Predicted', color='orange' )
```

```
plt.plot(y_test, y_test, label='Actual', color='blue')
```

```
plt.title('Predicted vs Actual Page Views')
```

```
plt.xlabel('Actual Page Views')
```

```
plt.ylabel('Predicted Page Views')
```

```
plt.legend()
```

```
plt.show()
```

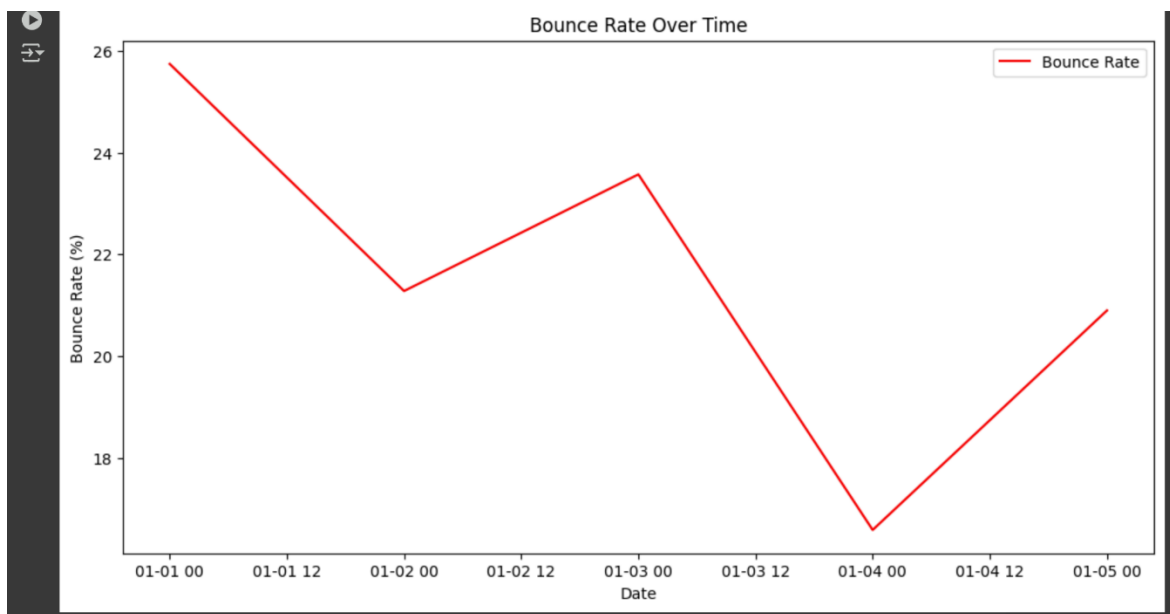
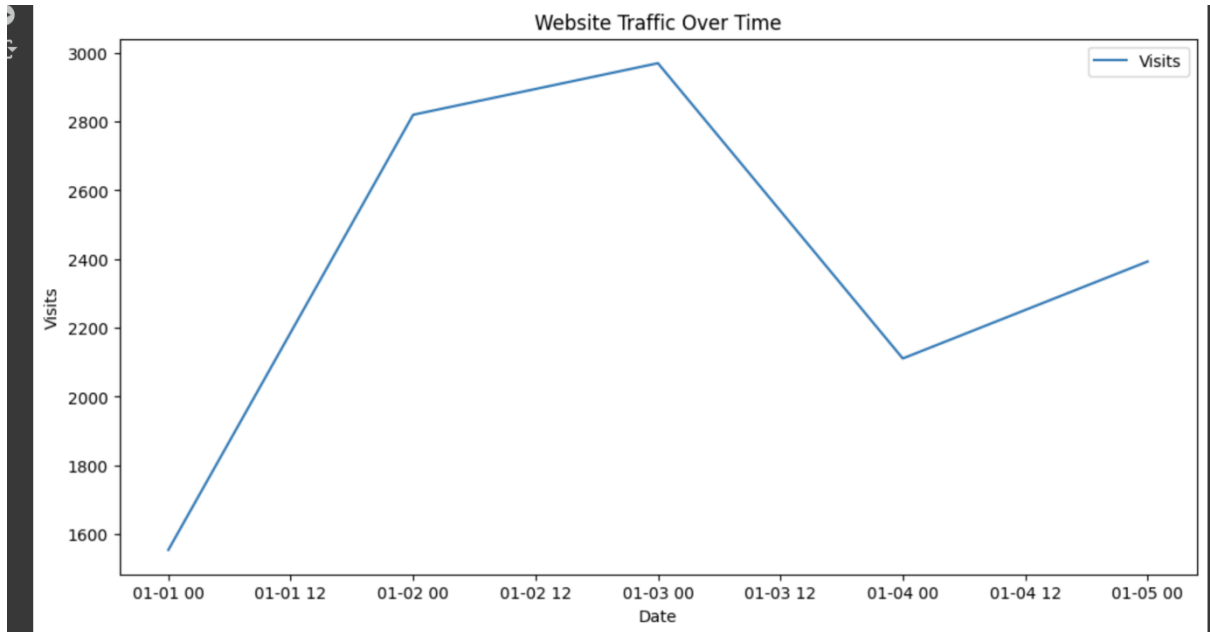

Screenshot of output

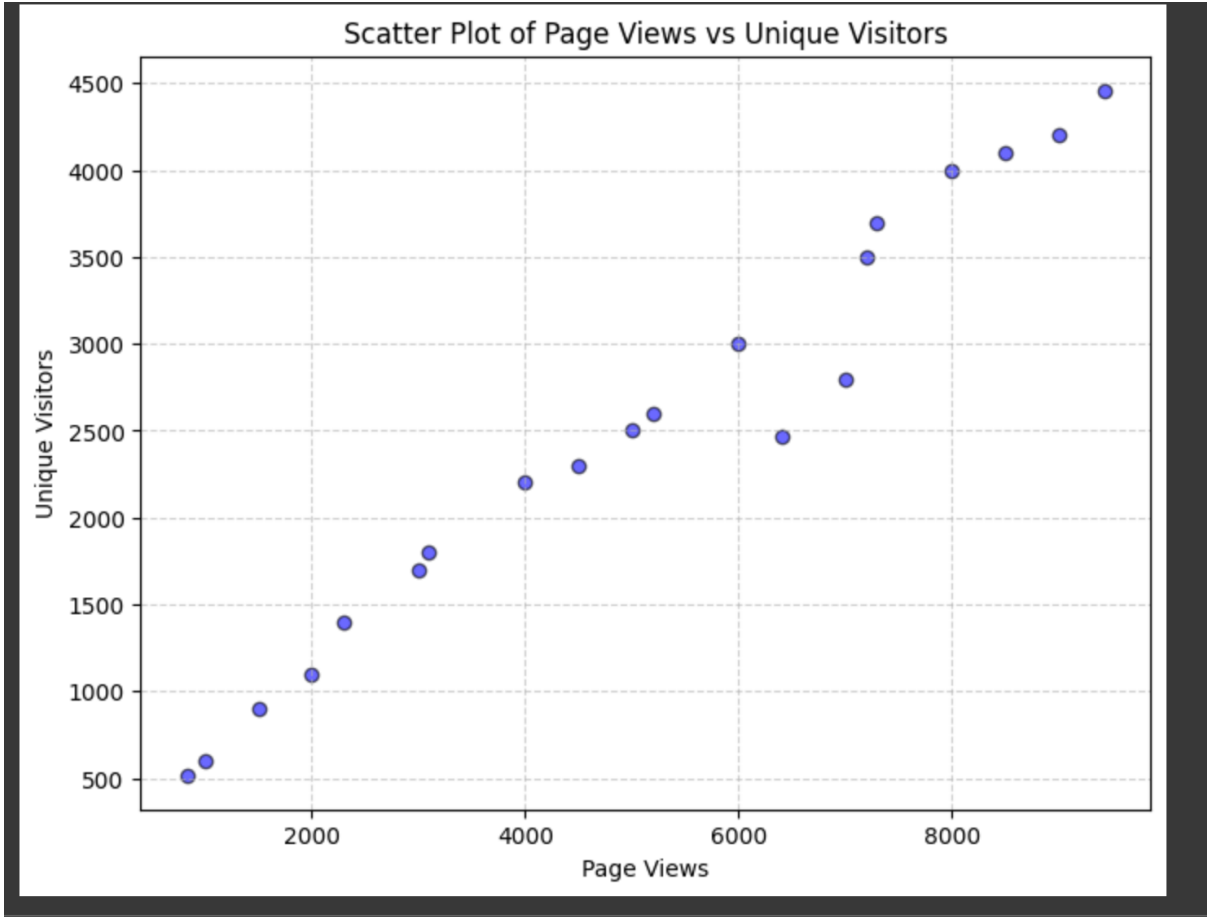
```
>>> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                   20 non-null    object
1   PageViews              20 non-null    int64
2   UniqueVisitors        20 non-null    int64
3   BounceRate             20 non-null    float64
dtypes: float64(1), int64(2), object(1)
memory usage: 772.0+ bytes
None
```

	PageViews	UniqueVisitors	BounceRate
count	20.00000	20.00000	20.000000
mean	5533.20000	2435.05000	49.150658
std	2595.96585	1383.40109	15.286241
min	828.00000	518.00000	28.581849
25%	3218.50000	1115.25000	37.609458
50%	6405.00000	2466.50000	49.061288
75%	7288.75000	3696.25000	60.163514
max	9432.00000	4459.00000	79.981676

```
>>>
```

Date	Visits
2020-01-01	1554.0
2020-01-02	2820.0
2020-01-03	2970.0
2020-01-04	2111.0
2020-01-05	2393.0





Conclusion

From the analysis, key insights were drawn regarding traffic trends, bounce rate patterns, and device usage. These insights can guide improvements in website design, content strategy, and marketing approaches.