

# **WEBSITE TRAFFIC ANALYSIS**

## **PHASE 3 : DEVELOPMENT PART 1**

**PRIYADARSHINI ENGINEERING COLLEGE**

**RAM PRASATH / 511921104055**

### **INTRODUCTION:**

Website traffic analysis is the process of collecting, examining, and interpreting data related to the visitors and interactions on a website. It provides invaluable insights into user behavior, preferences, and trends, helping organizations make informed decisions, optimize their online presence, and enhance user experiences.

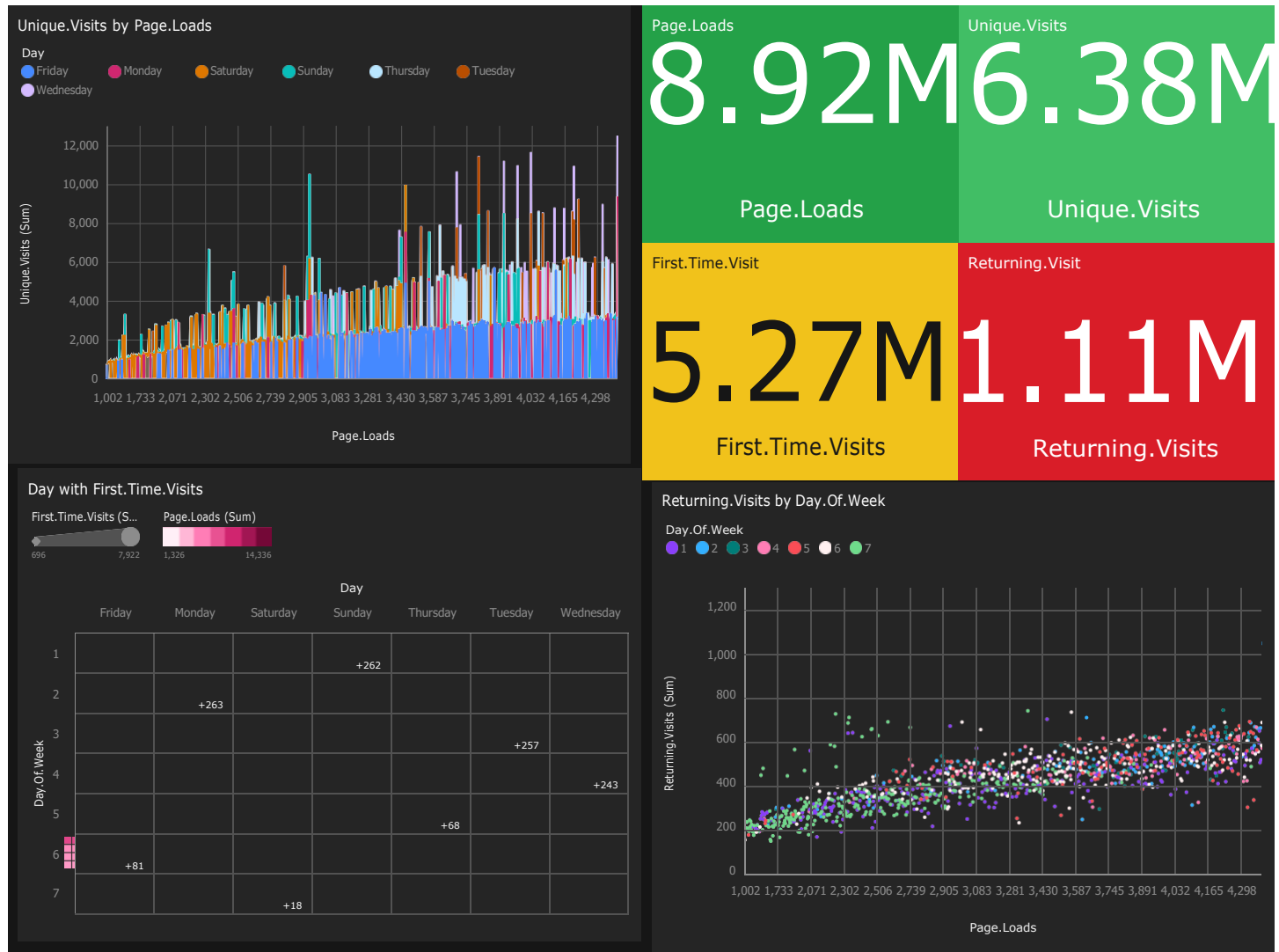
### **Abstract:**

This project aims to analyze website traffic data for insights into user behavior, popular pages, and traffic sources. It involves data collection, visualization using IBM Cognos, and Python for advanced analysis. The goal is to optimize user experiences and enhance website performance.

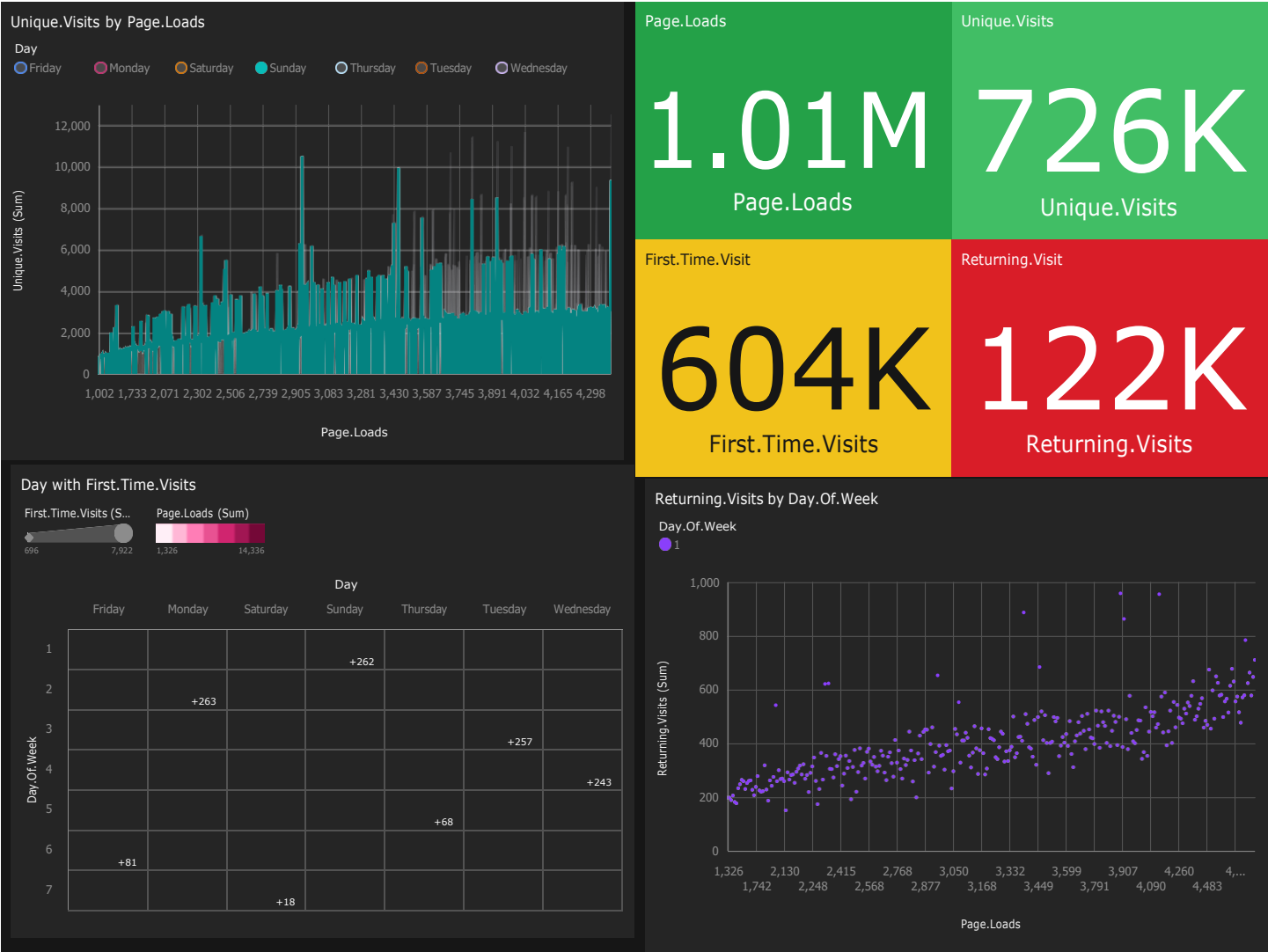
### **Problem Statement:**

The problem of "Website Traffic Analysis" lies in the need for organizations to effectively understand and leverage user behavior on their websites

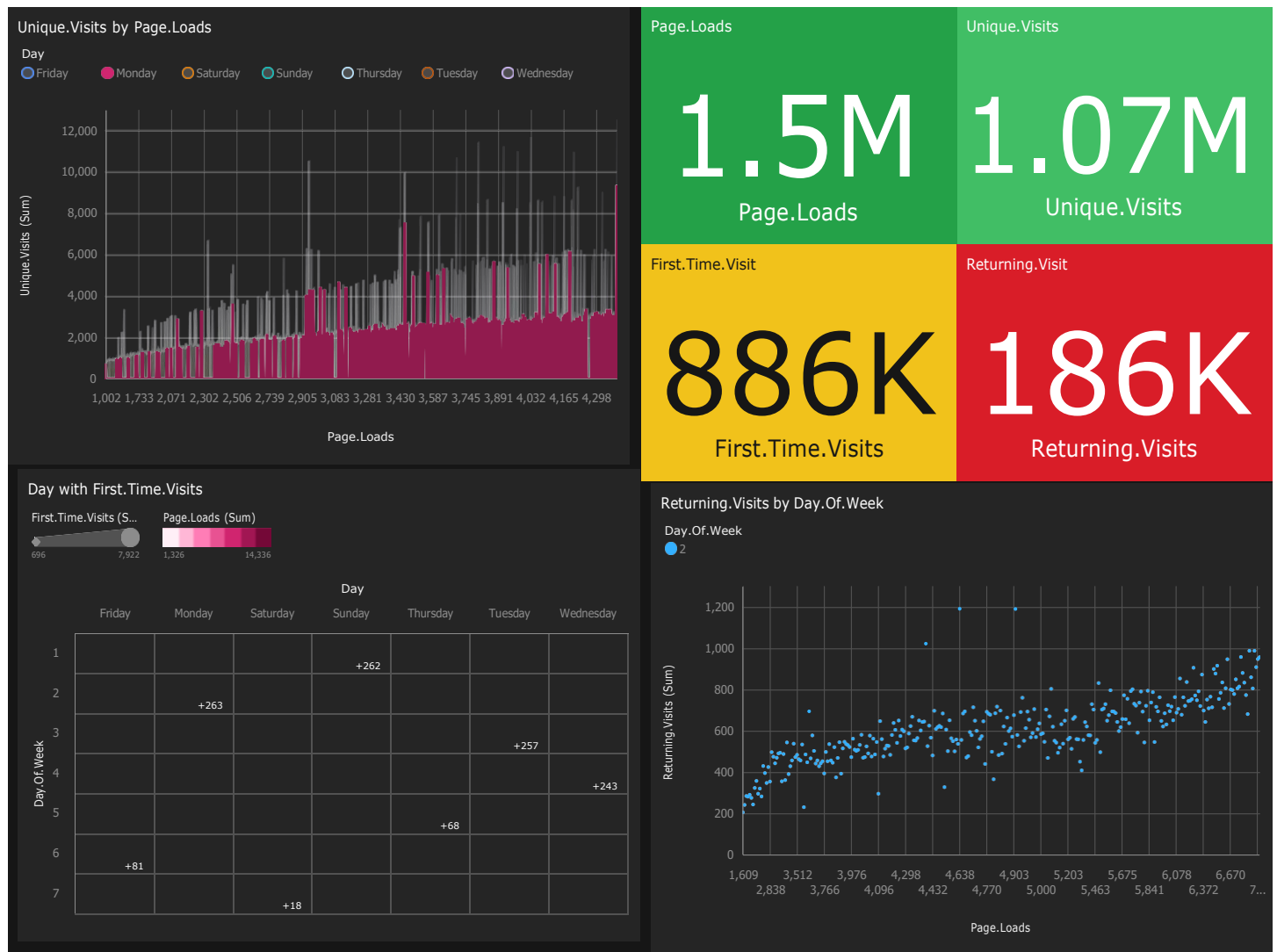
## Tab 1



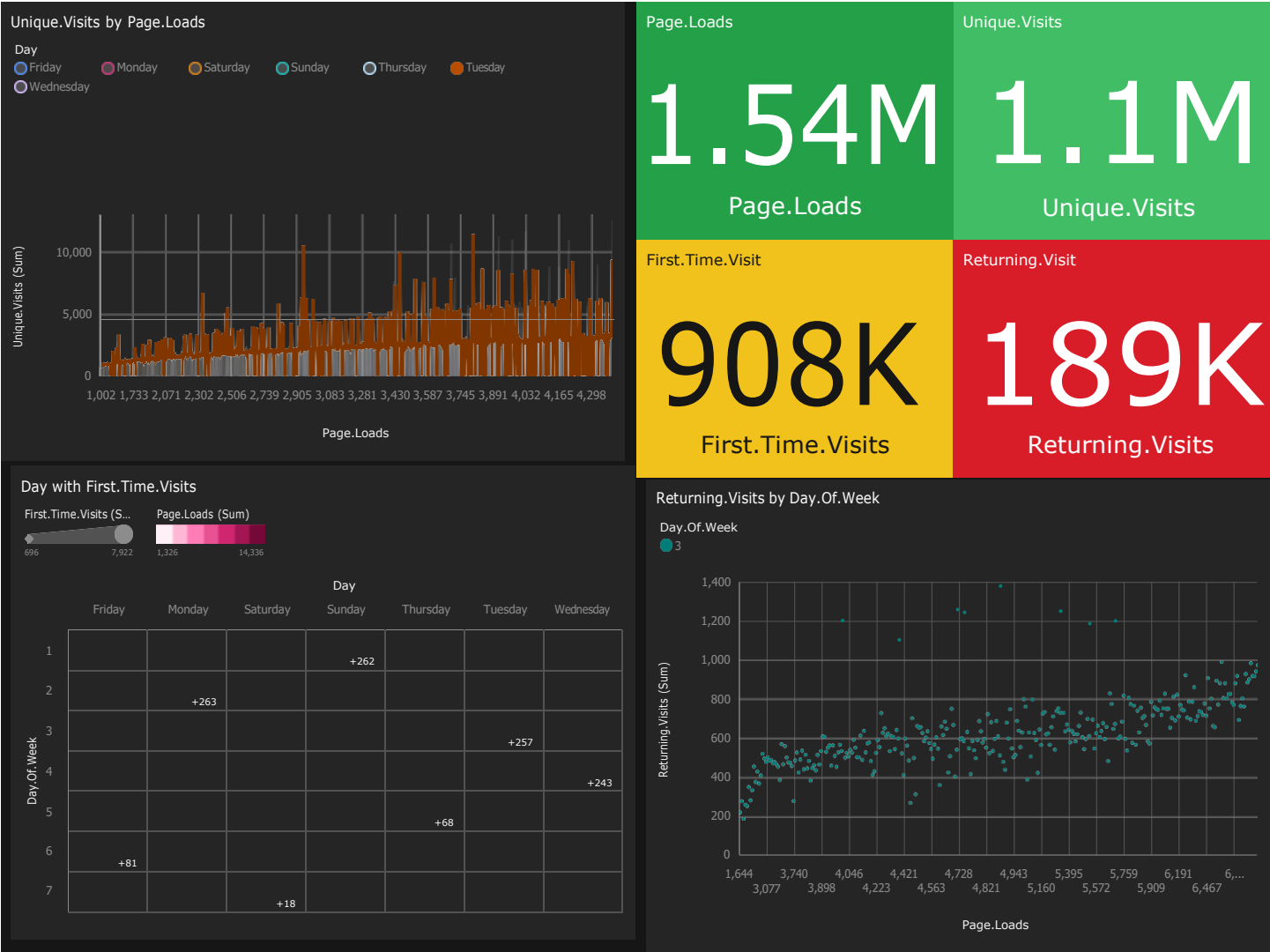
Tab 1



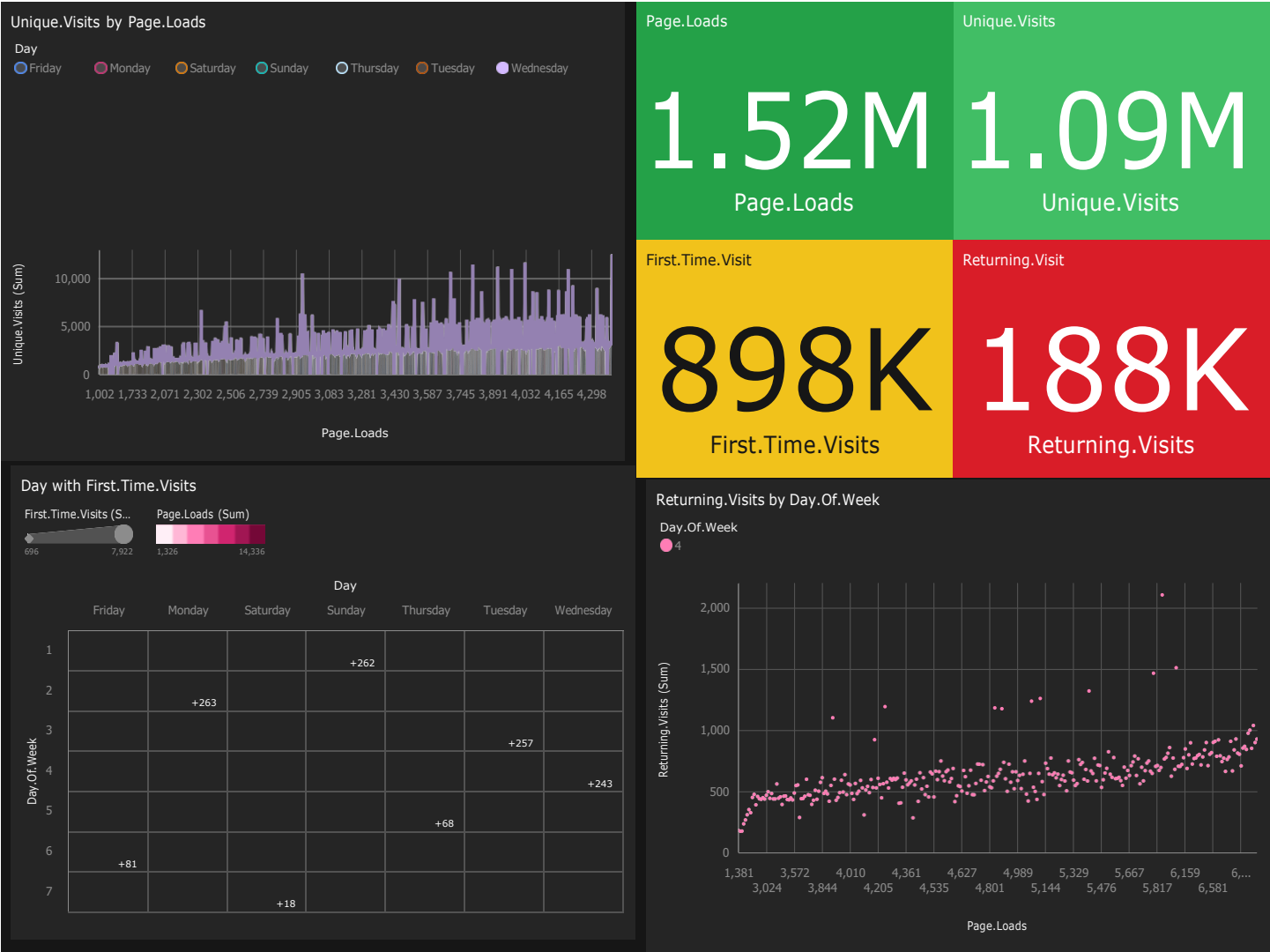
## Tab 1



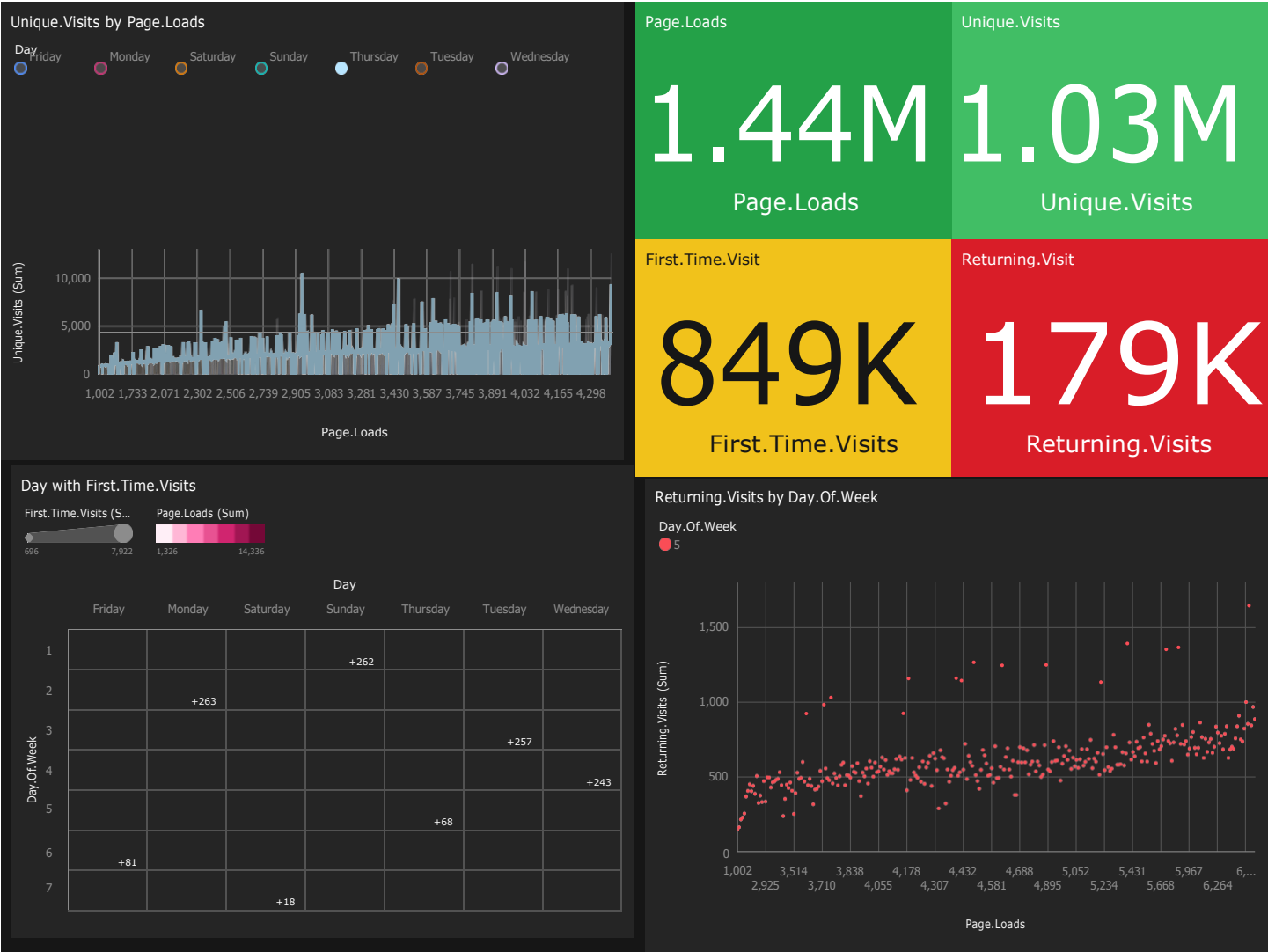
Tab 1



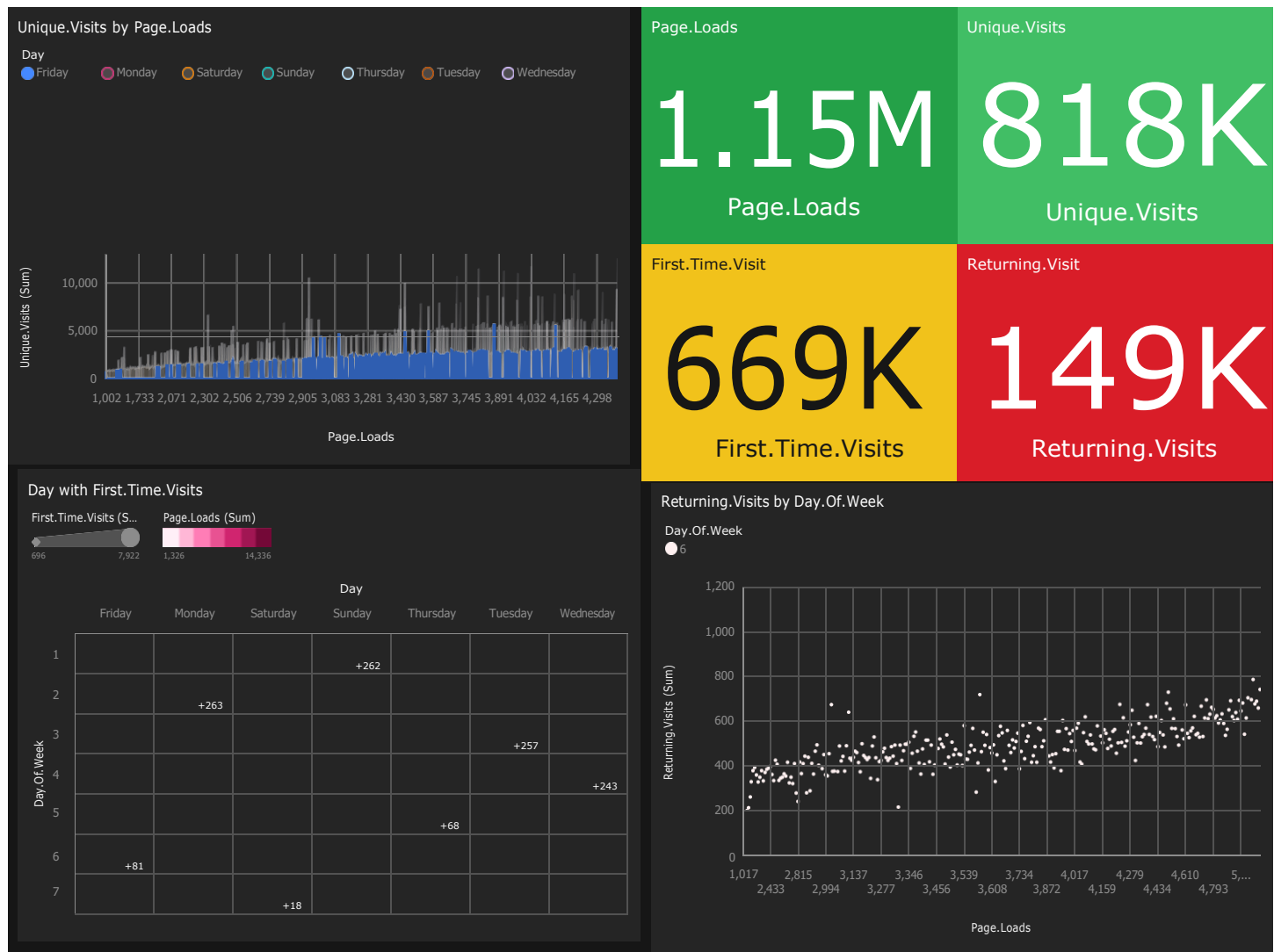
Tab 1



Tab 1

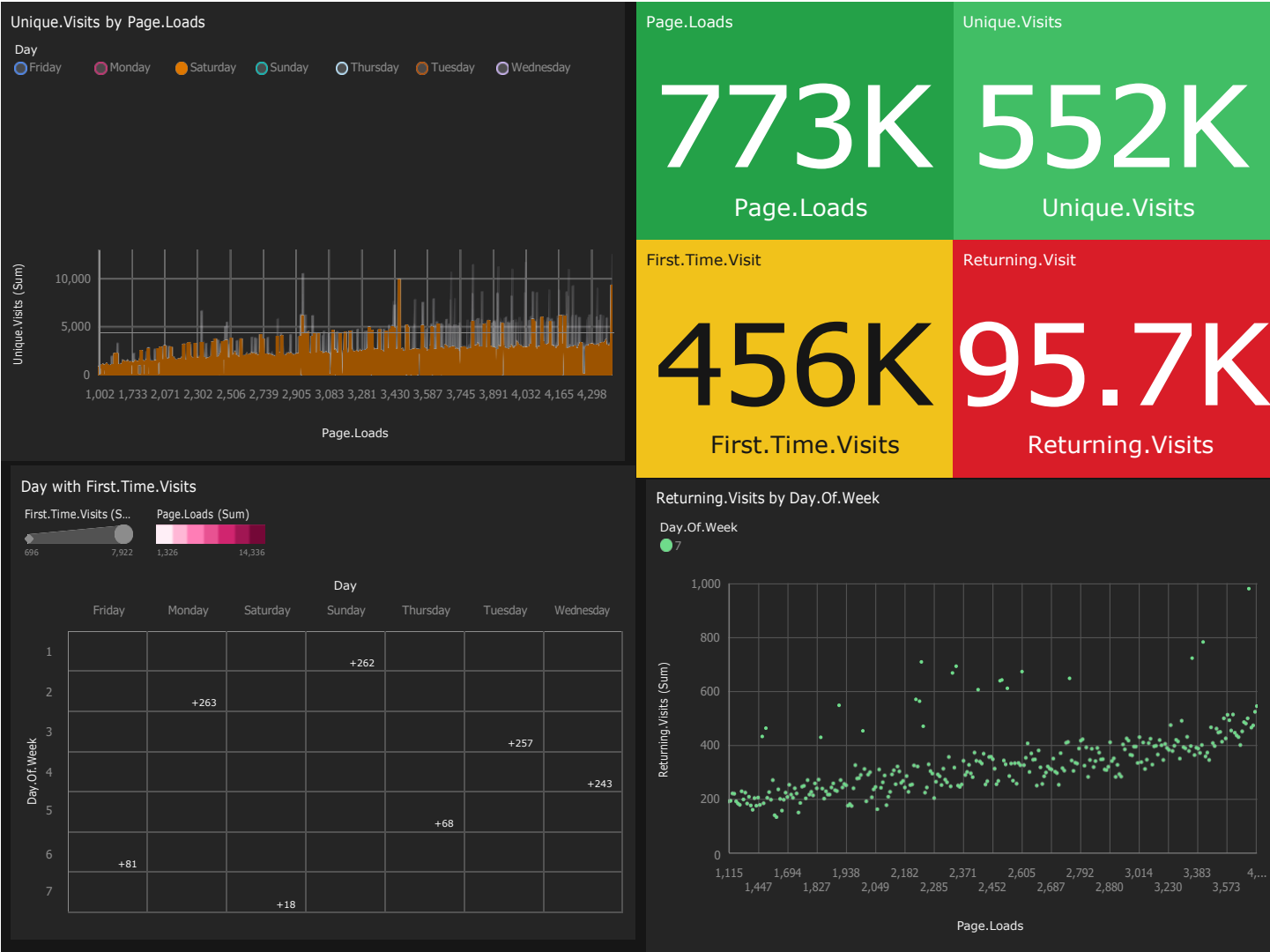


## Tab 1

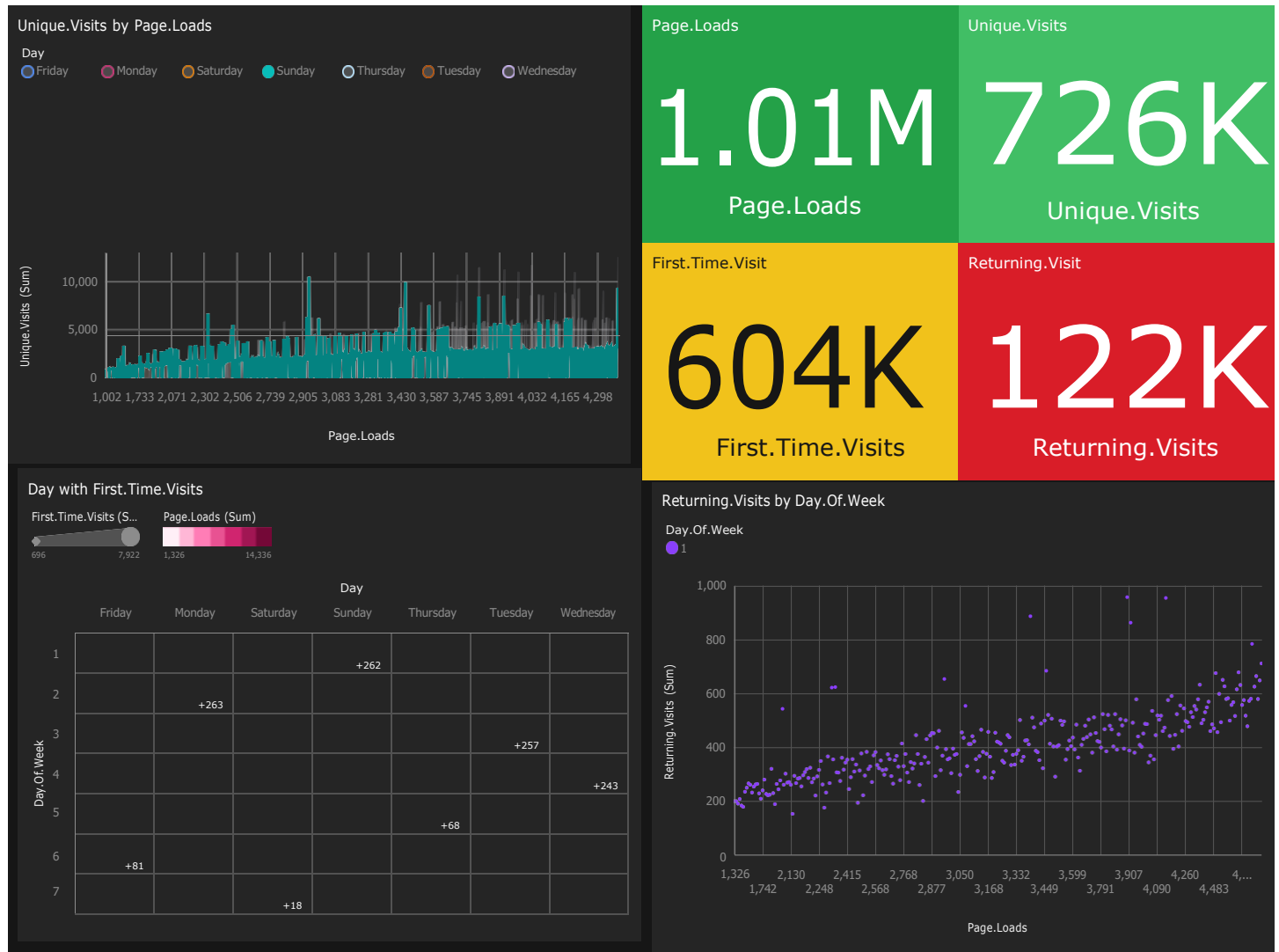




Tab 1



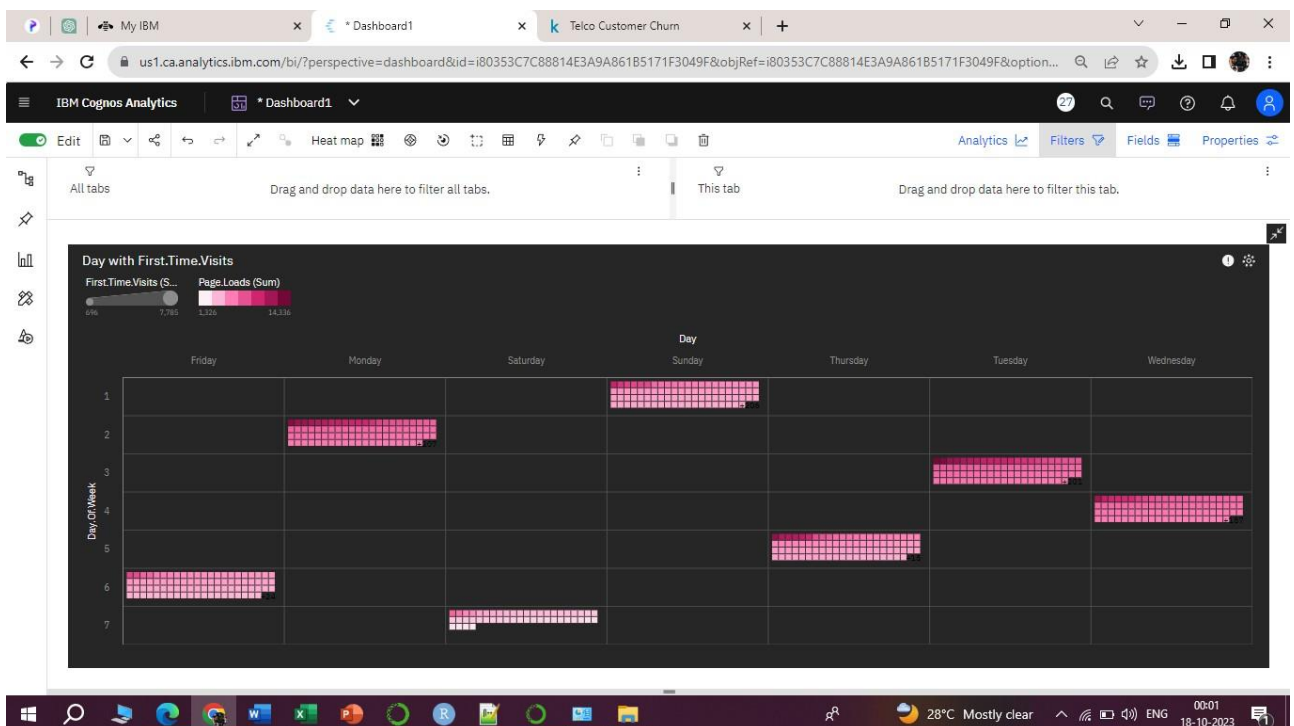
## Tab 1



# \*Insights for above “IBM Cognos Analytics”

## Unique Visits :

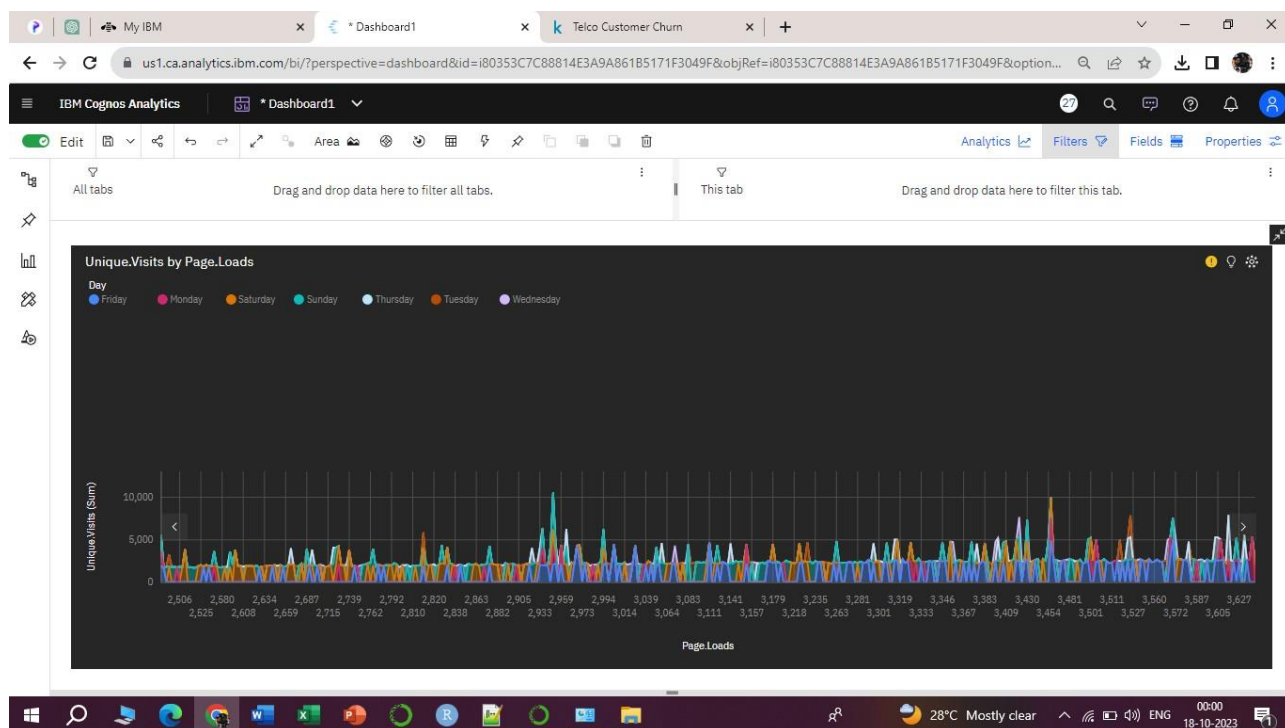
- ◆ Unique.Visits is unusually low when Day is Saturday.
- ◆ Based on the current forecasting, Unique.Visits may reach almost 481 thousand by Day Monday+1.
- ◆ It is projected that by Monday+1, 4205 will exceed 3973 in Unique.Visits by almost 1500.
- ◆ Page.Loads 4376 has the highest Total Returning.Visits but is ranked #5 in Total Unique.Visits.
- ◆ Page.Loads 4638 has the highest Total Unique.Visits but is ranked #3 in Total Returning.Visits.
- ◆ Over all values of Page.Loads and Day, the sum of Unique.Visits is almost 6.4 million.
- ◆ The summed values of Unique.Visits range from 667 to nearly 13 thousand.
- ◆ For Unique.Visits, the most significant values of Day are Tuesday, Wednesday, Monday, Thursday, and Friday, whose respective Unique.Visits values add up to over 5.1 million, or 80 % of the total.



## First Time Visitors :

- ◆ Day.Of.Week 7 has the highest Unaggregated First.Time.Visits but is ranked #7 in Total Returning.Visits.
- ◆ Day Saturday has the highest Unaggregated First.Time.Visits but is ranked #7 in Total Returning.Visits.
- ◆ Day.Of.Week 3 has the highest Total Returning.Visits but is ranked #5 in Unaggregated First.Time.Visits.
- ◆ Day Tuesday has the highest Total Returning.Visits but is ranked #5 in Unaggregated First.Time.Visits.
- ◆ First.Time.Visits 3133 has the highest Total Returning.Visits but is ranked #2 in Total Page.Loads.
- ◆ First.Time.Visits 3146 has the highest Total Page.Loads but is ranked #2 in Total Returning.Visits.
- ◆ 1 (14.3 %), 2 (14.3 %), 3 (14.3 %), and 4 (14.3 %) are the most frequently occurring categories of Day.Of.Week with a combined count of 1240 items with First.Time.Visits values (57.2 % of the total).
- ◆ 1 (14.3 %), 2 (14.3 %), 3 (14.3 %), and 4 (14.3 %) are the most frequently occurring categories of Day.Of.Week with a combined count of 1240 items with Page.Loads values (57.2 % of the total).
- ◆ Across all values of Day.Of.Week, the sum of Page.Loads is over 8.9 million.
- ◆ The summed values of First.Time.Visits range from 0 to over 1500.
- ◆ The summed values of Page.Loads range from over a thousand to over fourteen thousand.
- ◆ For First.Time.Visits, the most significant values of Day.Of.Week are 3, 4, 2, 5, and 6, whose respective First.Time.Visits values add up to over 4.2 million, or 79.9 % of the total.s
- ◆ For Page.Loads, the most significant values of Day.Of.Week are 3, 4, 2, 5, and 6, whose respective Page.Loads values add up to over 7.1 million, or 80.1 % of the total.

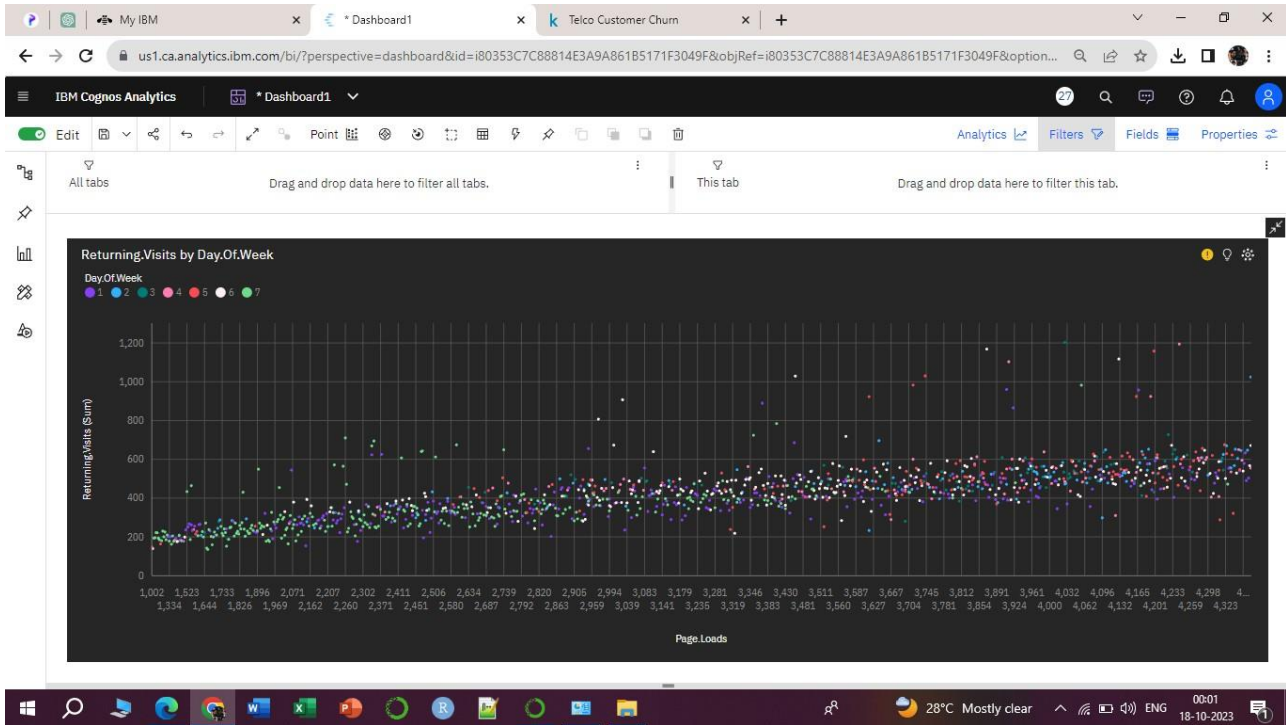
- ◆ For First.Time.Visits, the most significant values of Day are Tuesday, Wednesday, Monday, Thursday, and Friday, whose respective First.Time.Visits values add up to over 4.2 million, or 79.9 % of the total.
- ◆ For Page.Loads, the most significant values of Day are Tuesday, Wednesday, Monday, Thursday, and Friday, whose respective Page.Loads values add up to over 7.1 million, or 80.1 % of the total.



## Returning Visit :

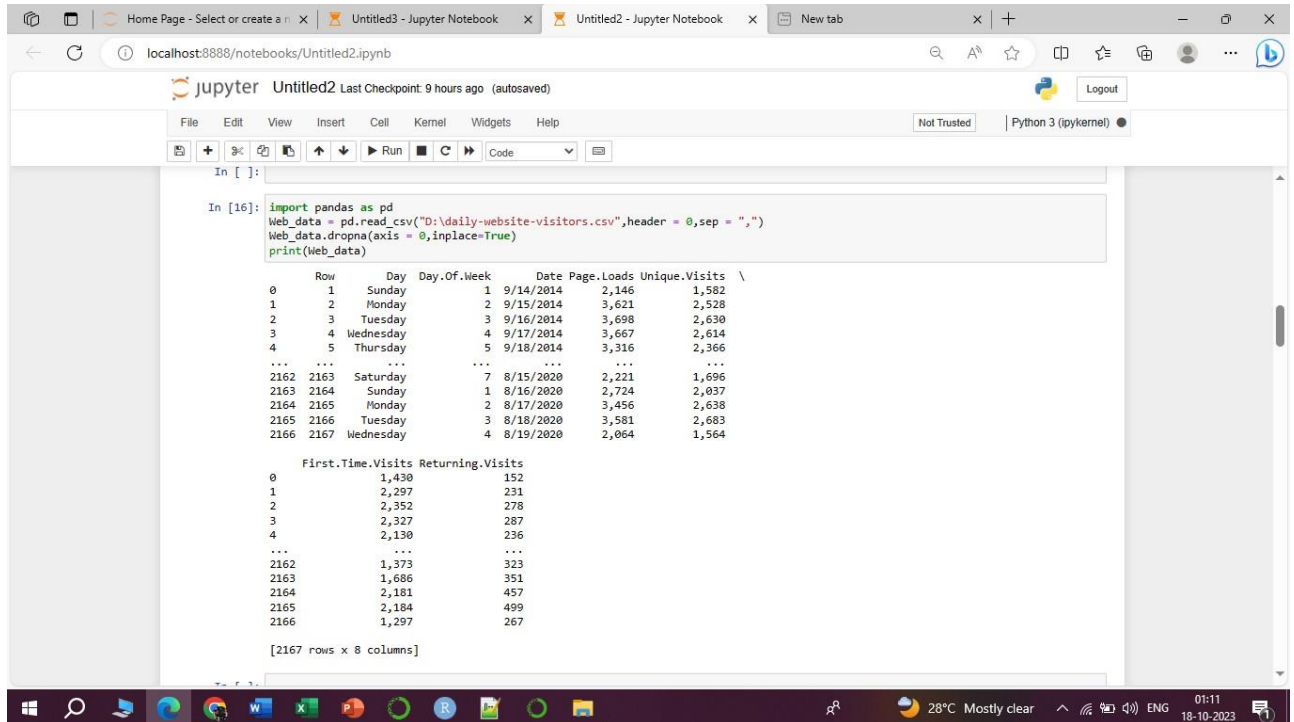
- ◆ Returning.Visits is unusually low when Day.Of.Week is 7.
- ◆ Based on the current forecasting, Returning.Visits may reach over 87 thousand by Day.Of.Week 9.
- ◆ It is projected that by 9, 4205 will exceed 3973 in Returning.Visits by 227.
- ◆ Across all values of Page.Loads and Day.Of.Week, the sum of Returning.Visits is over 1.1 million.
- ◆ The summed values of Returning.Visits range from 133 to over two thousand.

- ◆ For Returning.Visits, the most significant values of Day.Of.Week are 3, 4, 2, 5, and 6, whose respective



# \*'Python Integration' for Website Traffic Analysis

Extraction the Data file from the directory to the python text editor to excecute the data set .



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [16]: import pandas as pd
Web_data = pd.read_csv("D:\daily-website-visitors.csv", header = 0, sep = ",")
Web_data.dropna(axis = 0, inplace=True)
print(Web_data)
```

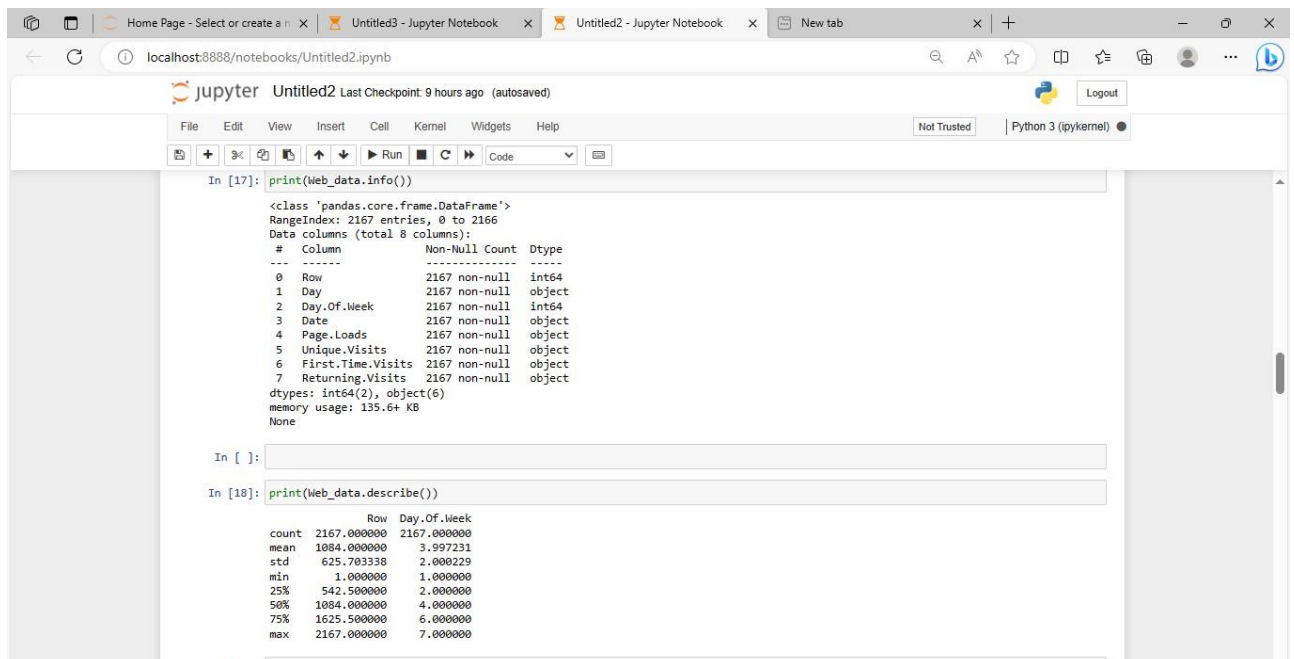
Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits
0	1	Sunday	1 9/14/2014	2,146	1,582
1	2	Monday	2 9/15/2014	3,621	2,528
2	3	Tuesday	3 9/16/2014	3,698	2,630
3	4	Wednesday	4 9/17/2014	3,667	2,614
4	5	Thursday	5 9/18/2014	3,316	2,366
...	...	...	...	...	...
2162	2163	Saturday	7 8/15/2020	2,221	1,696
2163	2164	Sunday	1 8/16/2020	2,724	2,037
2164	2165	Monday	2 8/17/2020	3,456	2,638
2165	2166	Tuesday	3 8/18/2020	3,581	2,683
2166	2167	Wednesday	4 8/19/2020	2,064	1,564

	First.Time.Visits	Returning.Visits
0	1,430	152
1	2,297	231
2	2,352	278
3	2,327	287
4	2,130	236
...	...	...
2162	1,373	323
2163	1,686	351
2164	2,181	457
2165	2,184	499
2166	1,297	267

[2167 rows x 8 columns]

Getting of data set information using info function.



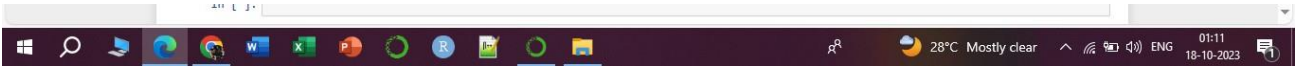
The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [17]: print(Web_data.info())
```

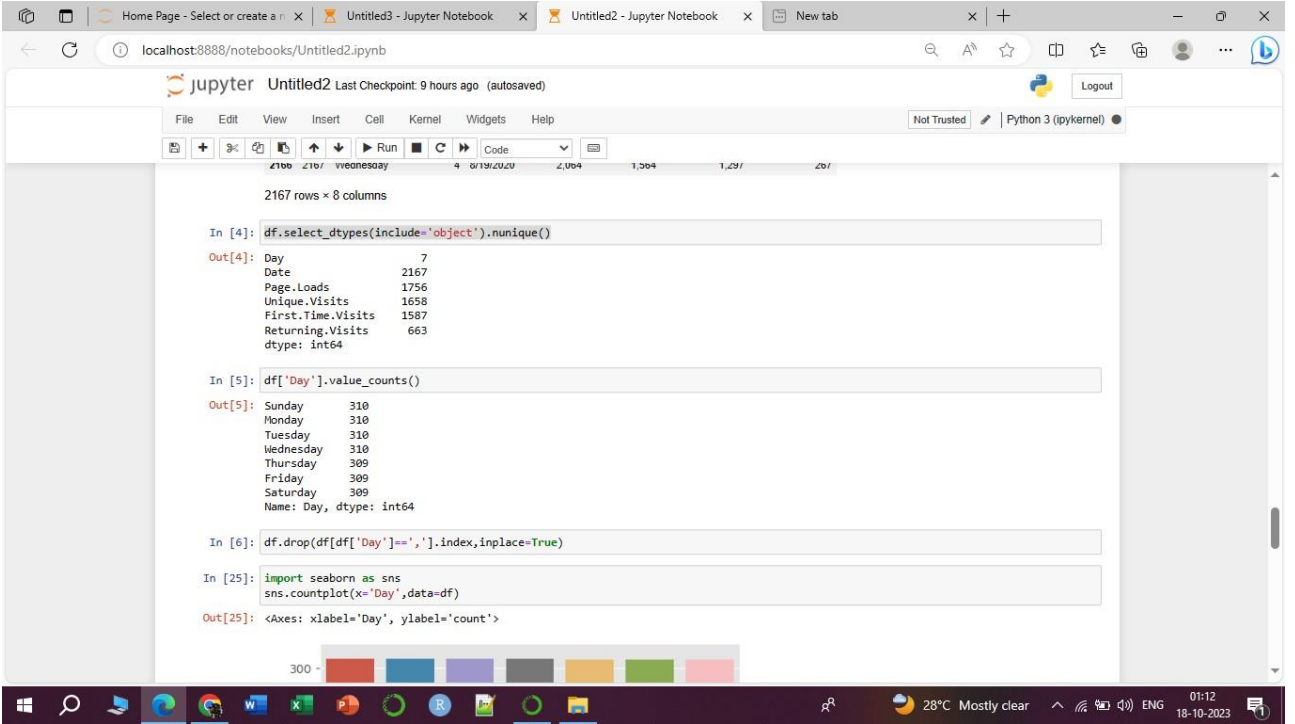
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2167 entries, 0 to 2166
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Row                   2167 non-null  int64  
1   Day                   2167 non-null  object  
2   Day.Of.Week           2167 non-null  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
None
```

```
In [18]: print(Web_data.describe())
```

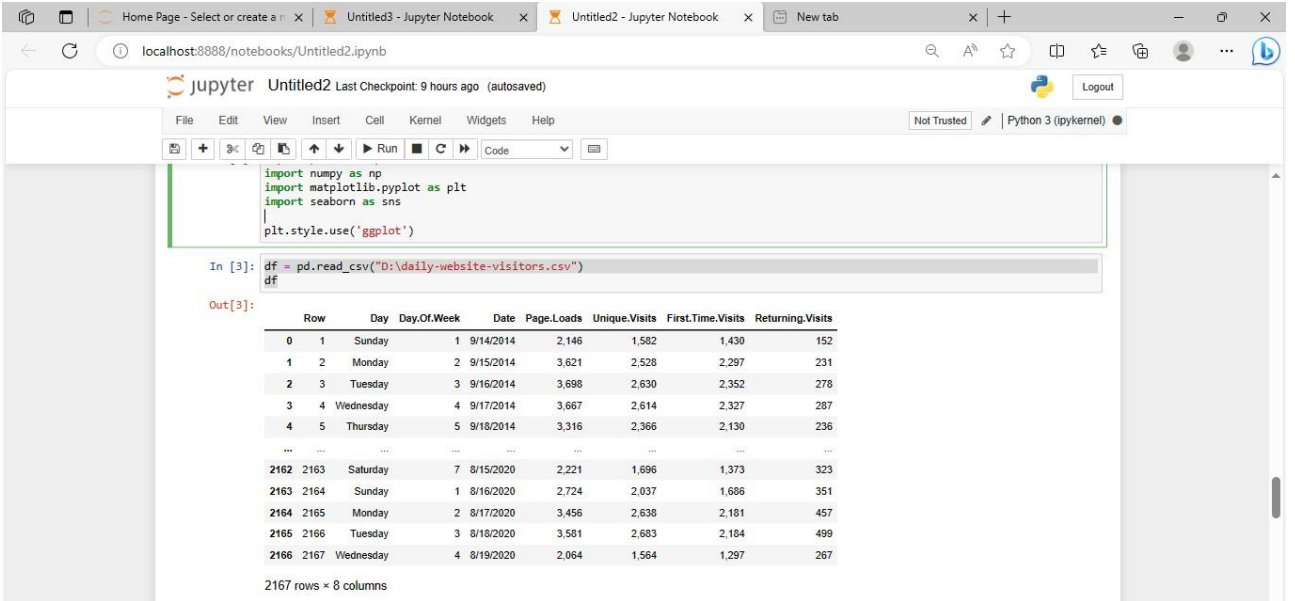
	Row	Day.Of.Week
count	2167.000000	2167.000000
mean	1084.000000	3.997231
std	625.703338	2.000229
min	1.000000	1.000000
25%	542.500000	2.000000
50%	1084.000000	4.000000
75%	1625.500000	6.000000
max	2167.000000	7.000000



# Value Counts of each Insights of the data set content and Object Integration of the data set



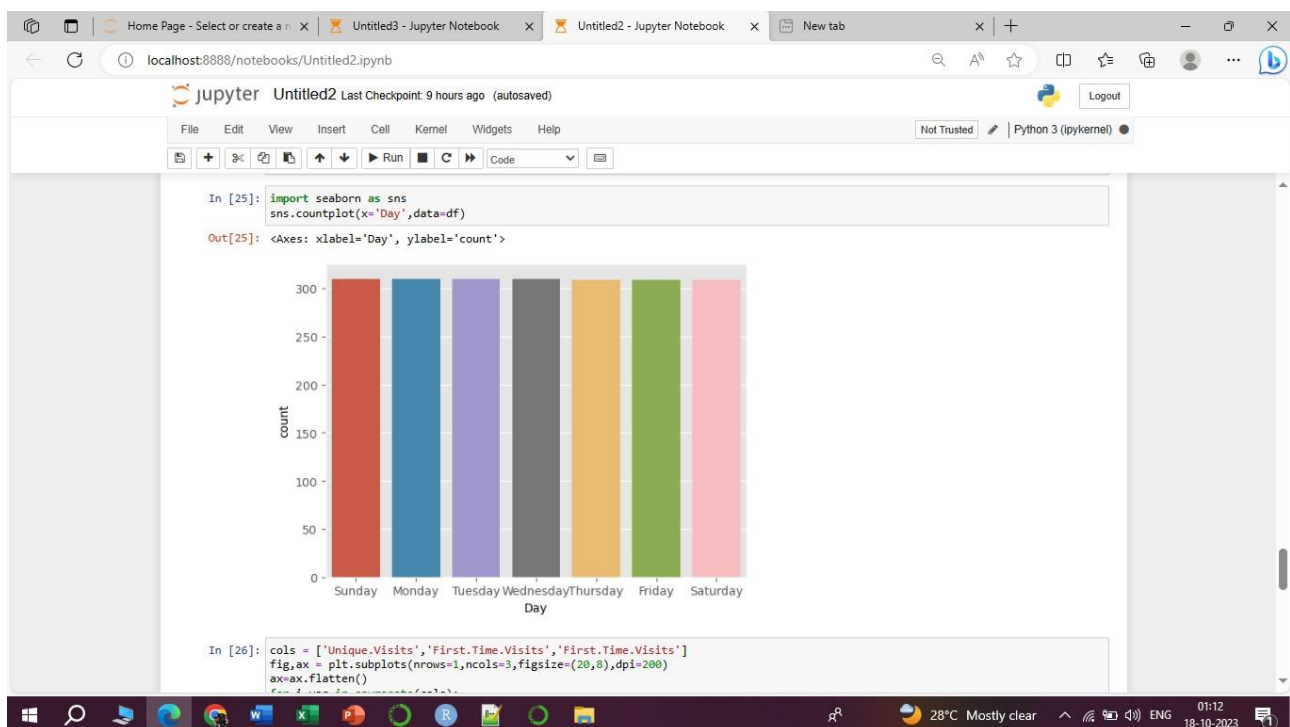
# Again pointing out the data set with the help of Pandas Library.



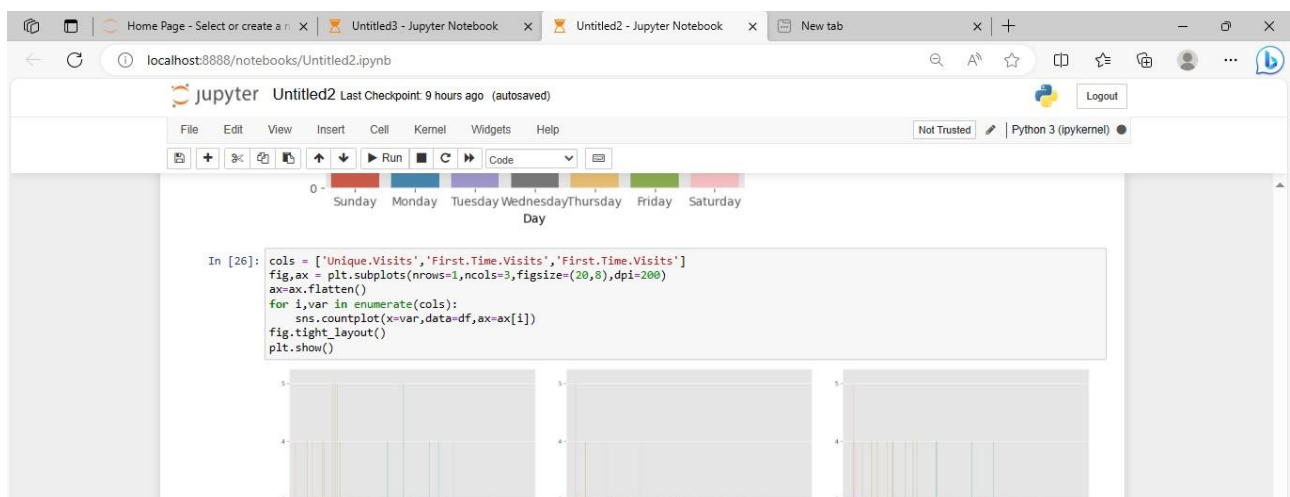


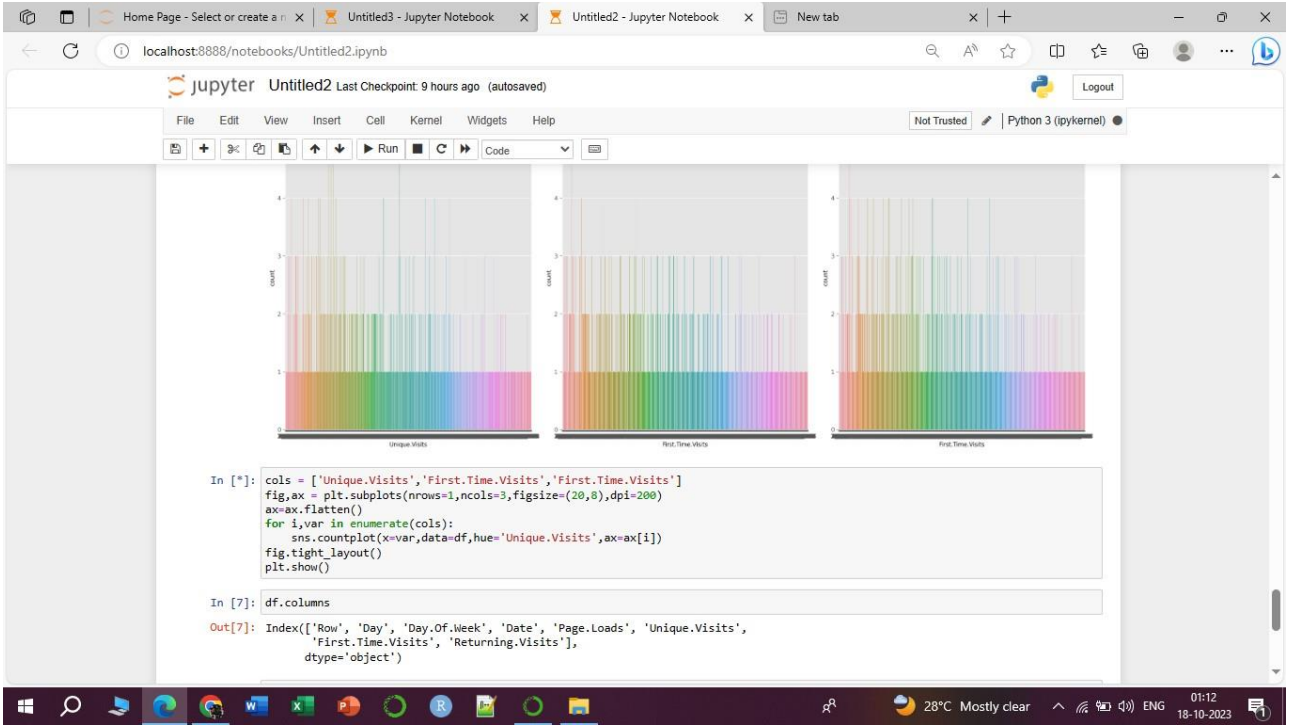
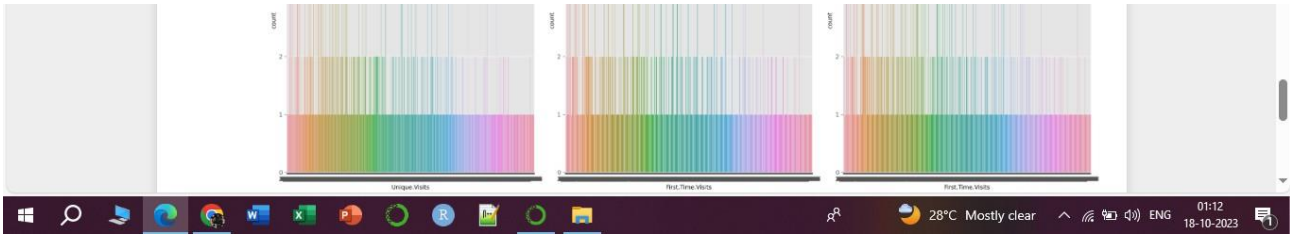


Listing the Number of Days that had been observed by the data set using Seaborn Library.

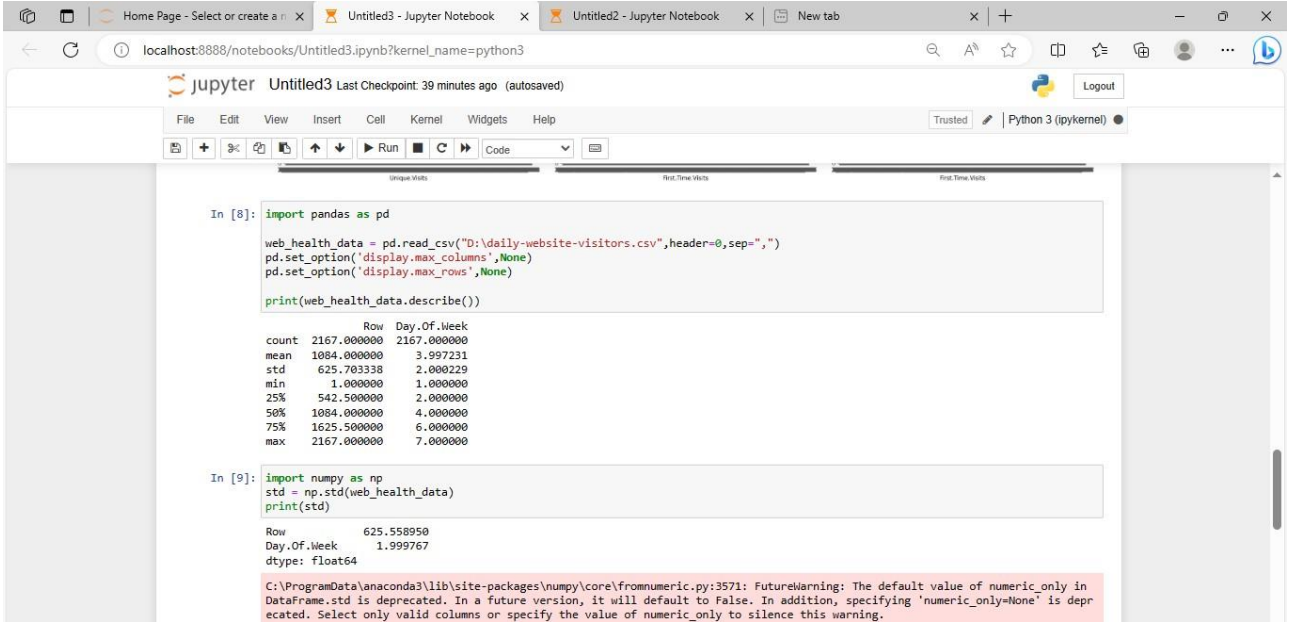


Plotting of Three important Stuffs like Unique visitor,First Time Visitor,Returning Visitor.





Mathematical Calculations of the data set with the help of Numpy Library.



```
return std(axis=axis, dtype=dtype, out=out, ddof=ddof, **kwargs)

In [10]: #correlation matrix
corr_matrix=round(web_health_data.corr(),2)
```

localhost:8888/notebooks/Untitled3.ipynb?kernel\_name=python3

jupyter Untitled3 Last Checkpoint: 39 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [11]: import numpy as np
cv = np.std(web_health_data)/np.mean(web_health_data)
print(cv)

Row      0.577084
Day.Of.Week 0.500288
dtype: float64

C:\ProgramData\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3571: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  return std(axis=axis, dtype=dtype, out=out, ddof=ddof, **kwargs)

C:\ProgramData\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3430: FutureWarning: In a future version, DataFrame.mean(axis=None) will return a scalar mean over the entire DataFrame. To retain the old behavior, use 'frame.mean(axis=0)' or just 'frame.mean()'
  return mean(axis=axis, dtype=dtype, out=out, **kwargs)

C:\ProgramData\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3430: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  return mean(axis=axis, dtype=dtype, out=out, **kwargs)

In [12]: import numpy as np
var = np.var(web_health_data)
print(var)

Row      391324.000000
Day.Of.Week 3.999069
dtype: float64

C:\ProgramData\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3713: FutureWarning: The default value of numeric_only in DataFrame.var is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  return var(axis=axis, dtype=dtype, out=out, ddof=ddof, **kwargs)
```

An Basic Observation with the Limited Data points and to plotting in the Line Graph.

localhost:8888/notebooks/Untitled4.ipynb

jupyter Untitled4 Last Checkpoint: 27 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Kernel

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt

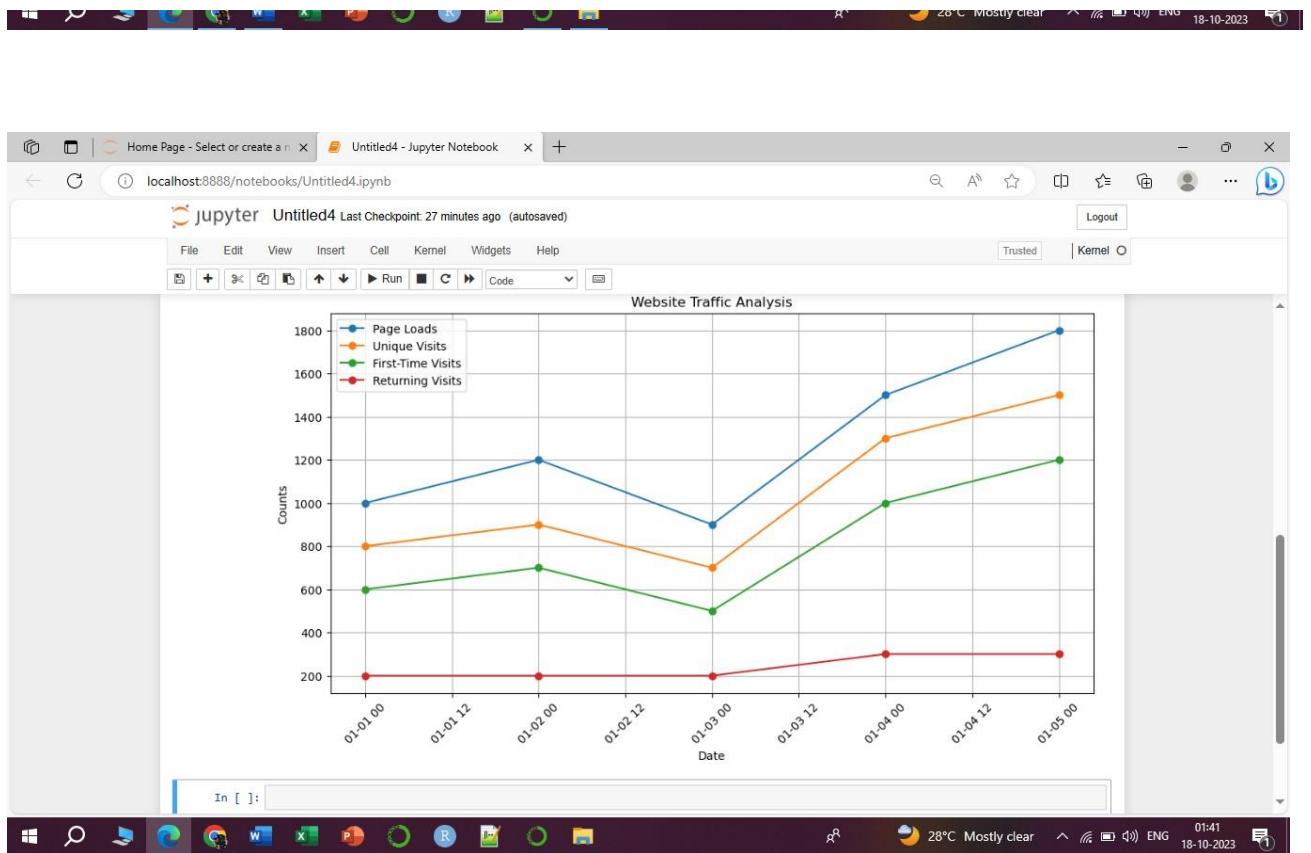
# Create a sample dataset (replace this with your actual data)
data = {
    'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04', '2023-01-05'],
    'Page_Loads': [1000, 1200, 900, 1500, 1800],
    'Unique_Visits': [800, 900, 700, 1300, 1500],
    'First_Time_Visits': [600, 700, 500, 1000, 1200],
    'Returning_Visits': [200, 200, 200, 300, 300]
}

# Create a DataFrame from the sample data
df = pd.DataFrame(data)

# Convert the 'Date' column to a datetime type
df['Date'] = pd.to_datetime(df['Date'])

# Set the 'Date' column as the DataFrame's index
df.set_index('Date', inplace=True)

# Plot page Loads and visits
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Page_Loads'], label='Page Loads', marker='o')
plt.plot(df.index, df['Unique_Visits'], label='Unique Visits', marker='o')
plt.plot(df.index, df['First_Time_Visits'], label='First-Time Visits', marker='o')
plt.plot(df.index, df['Returning_Visits'], label='Returning Visits', marker='o')
plt.title('Website Traffic Analysis')
plt.xlabel('Date')
plt.ylabel('Counts')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
```



## Conclusion :

In this Phase 3 : Development Part 1 Using IBM Cognos and Python Integration made the Basic Developing of the “Website Traffic Analysis” for Data Analytics with help of provided Kaggle Dataset.