PHASE 5

WEBSITE TRAFFIC ANALYSIS

JCT College of Engineering and Technology

BY,
AFSAL P A (AU720921243004)
ASWIN GEORGE (AU720921243015)
HARI KRISHNA K R (AU720921243026)
ROSHAN ROY (AU720921243047)

Website Traffic Analysis

➢ **Objective:**

The primary objective of this project is to gain valuable insights into the traffic of a specific website. These insights are aimed at improving the overall user experience by understanding user behavior, identifying popular pages, and recognizing traffic sources. In essence, the project seeks to answer critical questions about how visitors interact with the website.

➢ **Design Thinking:**

Design Thinking is a user-centric problem-solving approach that focuses on creating solutions that meet the needs of users. Here's how we can apply Design Thinking principles to the steps of your Website Traffic Analysis project:

Analysis Objectives:

- Empathize: Begin by empathizing with the website owners and stakeholders. Understand their goals, concerns, and what they hope to achieve through website traffic analysis. Conduct interviews and surveys to gather their insights.
- Define: Based on your empathy phase, clearly define the objectives of the analysis. What specific questions are the website owners trying to answer? What are the key performance indicators (KPIs) they want to track? Define success criteria for the project.
- Ideate: Brainstorm different ways to achieve the defined objectives. Consider various metrics, tools, and approaches that can help answer the questions and meet the project goals.
- Prototype: Create a preliminary plan outlining the objectives, metrics, and data sources that will be used to achieve the project's goals. Share this plan with stakeholders for feedback and refinement.

Data Collection (from provided Kaggle link):

- Empathize: Understand the data source (Kaggle) and the nature of the data available. Consider any limitations or biases in the data.
- Define: Clearly specify what data needs to be collected from the Kaggle dataset. Define the data collection process, including the frequency of data updates, data cleaning steps, and data storage.
- Ideate: Explore different ways to extract and preprocess the data from Kaggle to ensure it aligns with the project objectives. Consider data quality and completeness.
- Prototype: Create a data collection plan, which outlines the steps for acquiring, cleaning, and storing the data from Kaggle. Test the data collection process to ensure it works effectively.

(https://www.kaggle.com/datasets/bobnau/daily-website-visitors)

Visualization (IBM Cognos):

- Empathize: Put yourself in the shoes of the end-users who will be interacting with the visualizations. Understand their preferences and needs when it comes to data presentation.
- Define: Determine the specific visualizations that will be created using IBM Cognos. Define the key insights that need to be conveyed through these visualizations. Consider the best visualization types for different types of data (e.g., bar charts, line graphs, heatmaps).
- Ideate: Explore different design options for the visualizations. Consider the use of color, layout, and interactivity to make the data more engaging and understandable.
- Prototype: Create prototype visualizations using IBM Cognos. Test them with potential users or stakeholders to gather feedback. Iterate on the designs based on feedback.

Python Integration:

- Empathize: Understand the technical capabilities and constraints of the project team regarding Python integration. Consider their familiarity with Python and data analysis libraries.
- Define: Clearly define the specific tasks that require Python integration. For example, identify the advanced analyses or custom calculations that cannot be achieved through IBM Cognos alone.
- Ideate: Brainstorm potential Python code solutions for the defined tasks. Consider the libraries and tools that will be most suitable for the analysis. Explore ways to ensure the Python code integrates seamlessly with the rest of the project.
- Prototype: Develop prototype Python code for the identified tasks. Test the code to ensure it produces the desired results and aligns with the project's objectives. Collaborate with the project team to integrate Python code into the overall workflow.

➢ Innovation Description:

To elevate the Website Traffic Analysis project, we propose incorporating an innovative feature called "**User Persona Profiling**" that leverages the power of artificial intelligence and machine learning to enhance user-centric decision-making. This feature will take the project's insights to the next level by providing a deeper understanding of website visitors, allowing website owners to tailor their content and strategies even more effectively.

User Persona Profiling will involve the following steps:

1. ***Data Enrichment:*** In addition to standard traffic data, we will gather additional information about users, such as their click behavior, time spent on specific pages, and the sequence of pages visited. This data will be collected and anonymized to ensure privacy compliance.

2. ***Machine Learning Models:*** Implement machine learning models, such as clustering and classification algorithms, to group users into distinct personas based on their behavior. This step will help identify common traits and preferences among visitors.

3. ***Persona Insights:*** Once user personas are established, create dynamic dashboards within IBM Cognos that visualize the characteristics, preferences, and behaviors of each persona segment. This will help website owners see a clear picture of their diverse audience.

4. ***Content Personalization Recommendations*:** Use Python integration to develop recommendation engines that suggest personalized content or product recommendations for each user persona. This could include suggesting relevant blog articles, products, or services based on the user's persona.

5. ***A/B Testing:*** Implement A/B testing experiments to validate the effectiveness of content personalization for different user personas. Continuously optimize content based on the results of these tests.

6. ***Real-time Personalization*:** For returning visitors, employ real-time personalization techniques to adapt the website's content and layout based on the detected persona. This ensures that users see the most relevant content immediately.

7. ***Feedback Loop:*** Incorporate feedback mechanisms to collect user feedback on the personalized experience. Analyze this feedback to fine-tune the persona profiling models and content recommendations further.

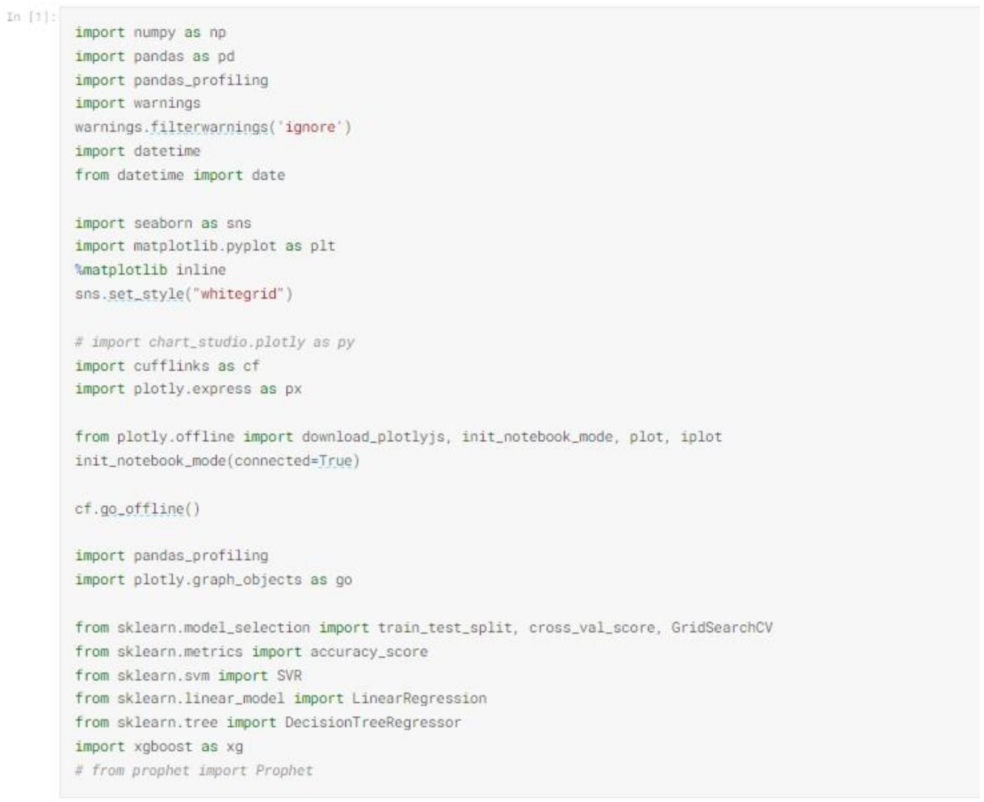
➢ <u>Data Preprocessing and Python Integration:</u>

Data Cleansing:

- Data Cleansing is the process of removing inconsistencies and incorrect values in the dataset.
- It also involves handling missing values either by removing them or assigning the average values.
- It helps to improve the efficiency of the model.

Importing Necessary Libraries and Initializing Plotly & Cufflinks:

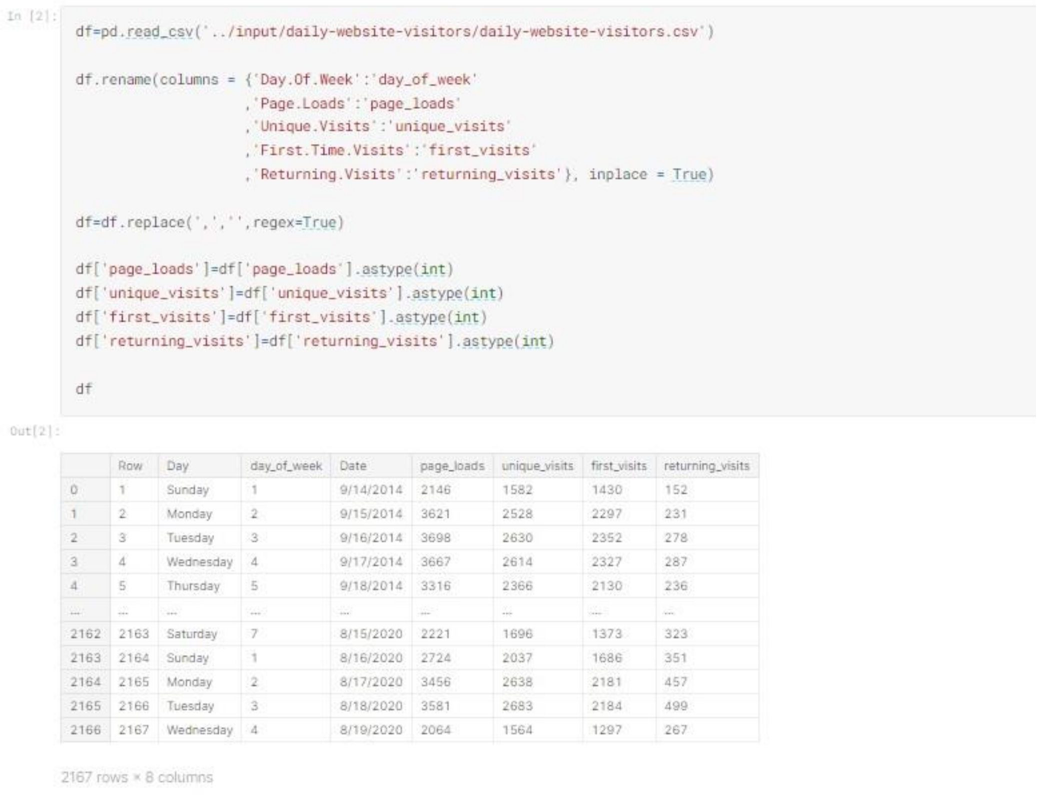
- Import various Python Libraries such as Numpy, Pandas ,Seaborn ,Matplotlib , Plotly Express ,Cufflinks and others. These libraries are used for data manipulation, visualization, and machine leaning.
- Set up the notebook environment for Plotly and Cufflinks to enable interactive plotting in Jupyter notebooks.



```
In [1]:
import numpy as np
import pandas as pd
import pandas_profiling
import warnings
warnings.filterwarnings('ignore')
import datetime
from datetime import date

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set_style("whitegrid")

# import chart_studio.plotly as py
import cufflinks as cf
import plotly.express as px

from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)

cf.go_offline()

import pandas_profiling
import plotly.graph_objects as go

from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
import xgboost as xg
# from prophet import Prophet
```

Reading Data & Data Preprocessing:

- Read a CSV file containing website traffic data into a Pandas DataFrame. Ensure that the file path is correctly specified.
- Rename columns in the DataFrame for easier reference and remove commas from numeric columns. The columns 'page_loads', 'unique_visits', 'first_visits', and 'returning_visits' are converted to integer data types.



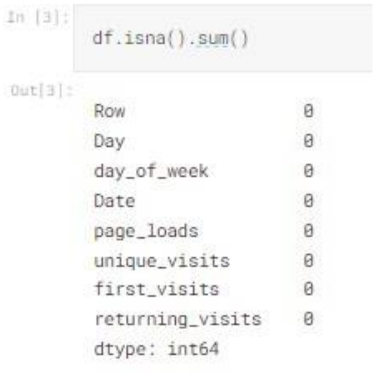
In [2]:
```
df=pd.read_csv('../input/daily-website-visitors/daily-website-visitors.csv')

df.rename(columns = {'Day.Of.Week':'day_of_week'
                    ,'Page.Loads':'page_loads'
                    ,'Unique.Visits':'unique_visits'
                    ,'First.Time.Visits':'first_visits'
                    ,'Returning.Visits':'returning_visits'}, inplace = True)

df=df.replace(',','',regex=True)

df['page_loads']=df['page_loads'].astype(int)
df['unique_visits']=df['unique_visits'].astype(int)
df['first_visits']=df['first_visits'].astype(int)
df['returning_visits']=df['returning_visits'].astype(int)

df
```
Out[2]:

	Row	Day	day_of_week	Date	page_loads	unique_visits	first_visits	returning_visits
0	1	Sunday	1	9/14/2014	2146	1582	1430	152
1	2	Monday	2	9/15/2014	3621	2528	2297	231
2	3	Tuesday	3	9/16/2014	3698	2630	2352	278
3	4	Wednesday	4	9/17/2014	3667	2614	2327	287
4	5	Thursday	5	9/18/2014	3316	2366	2130	236
...	...	...	...	...	...	...	...	...
2162	2163	Saturday	7	8/15/2020	2221	1696	1373	323
2163	2164	Sunday	1	8/16/2020	2724	2037	1686	351
2164	2165	Monday	2	8/17/2020	3456	2638	2181	457
2165	2166	Tuesday	3	8/18/2020	3581	2683	2184	499
2166	2167	Wednesday	4	8/19/2020	2064	1564	1297	267

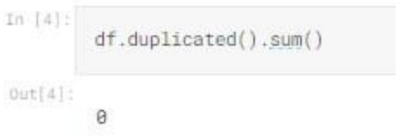
2167 rows × 8 columns

Data Quality Check & Data Information:

- Check for missing values using `df.isna().sum()` and duplicate rows using `df.duplicated().sum()` in the DataFrame.
- Display information about the DataFrame using `df.info()`. This provides details on column data types, non-null counts, and memory usage.



In [3]:
```
df.isna().sum()
```
Out[3]:
```
Row                 0
Day                 0
day_of_week         0
Date                0
page_loads          0
unique_visits       0
first_visits        0
returning_visits    0
dtype: int64
```



In [4]:
```
df.duplicated().sum()
```
Out[4]:
```
0
```

```
In [5]: df.info()
```

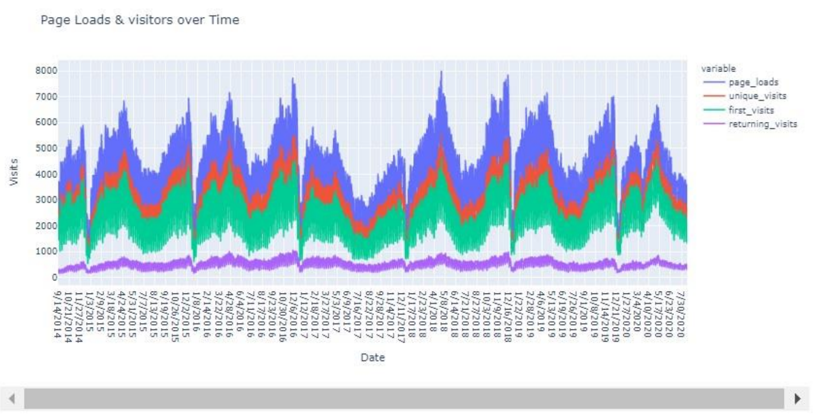
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2167 entries, 0 to 2166
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Row               2167 non-null   int64
 1   Day               2167 non-null   object
 2   day_of_week       2167 non-null   int64
 3   Date              2167 non-null   object
 4   page_loads        2167 non-null   int64
 5   unique_visits     2167 non-null   int64
 6   first_visits      2167 non-null   int64
 7   returning_visits  2167 non-null   int64
dtypes: int64(6), object(2)
memory usage: 135.6+ KB
```

Data Visualization(In Python Platform):

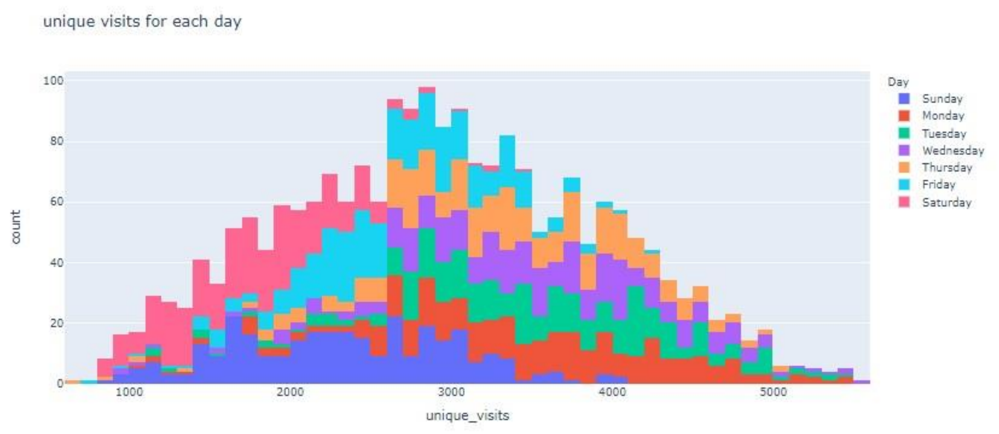
Create various data visualizations using Plotly Express and Matplotlib:

- A line plot (`fig`) showing page loads and visitors over time.
- A histogram of unique visits for each day.
- A bar chart of the sum of unique visits for each day.
- A histogram showing the sum of unique visits for each day over time.
- A bar chart showing the sum of various types of visits (page loads, unique visits, firsttime visits, returning visits) for each day.
- A correlation heatmap using Seaborn, illustrating the correlation between the numeric variables.
- A scatter matrix using Plotly Express, showing scatter plots and histograms for the numeric variables.

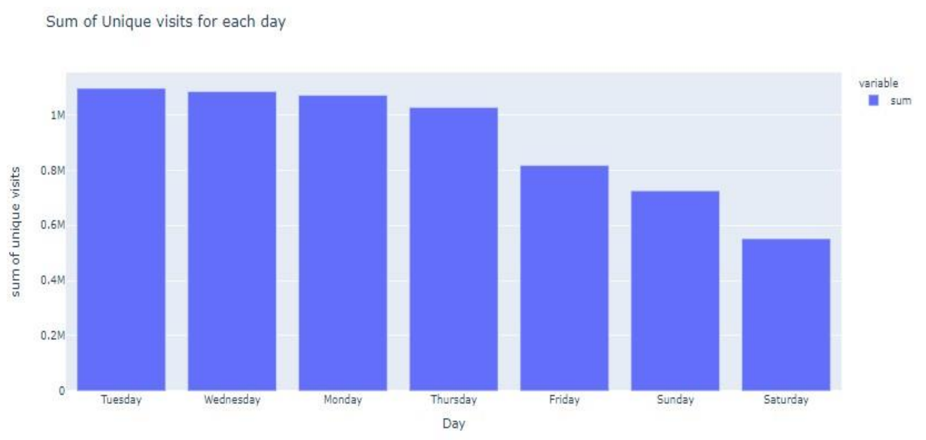
```
In [6]: px.line(df,x='Date',y=['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits'],
               labels={'value':'Visits'}
               ,title='Page Loads & visitors over Time')
```



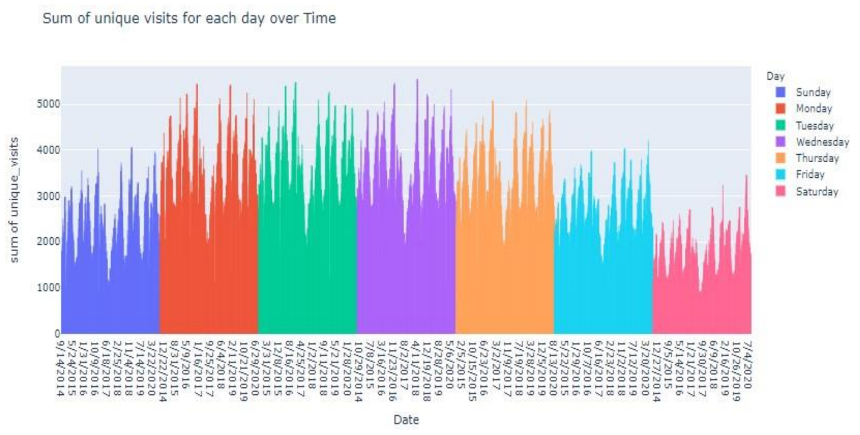
In [7]:
```python
px.histogram(df,x='unique_visits',color='Day',title='unique visits for each day')
```



In [8]:
```python
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')
```



In [9]:
```python
px.histogram(df,x='Date',y='unique_visits',color='Day',title='Sum of unique visits for each day over Time')
```



[illegible]

In [10]:

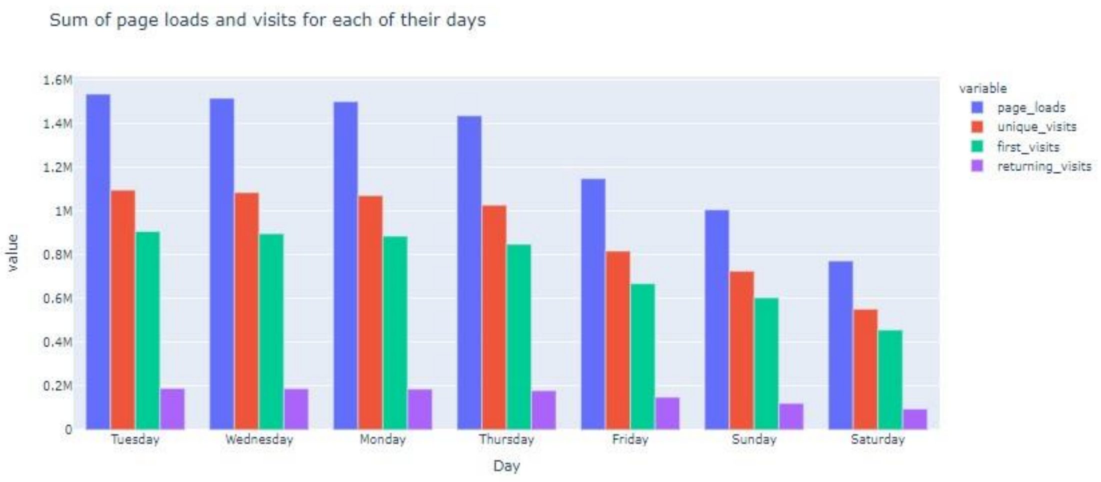
```
sums=df.groupby(['Day'])[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']].sum().sort_values(
    by='unique_visits',ascending=False)
sums
```

Out[10]:

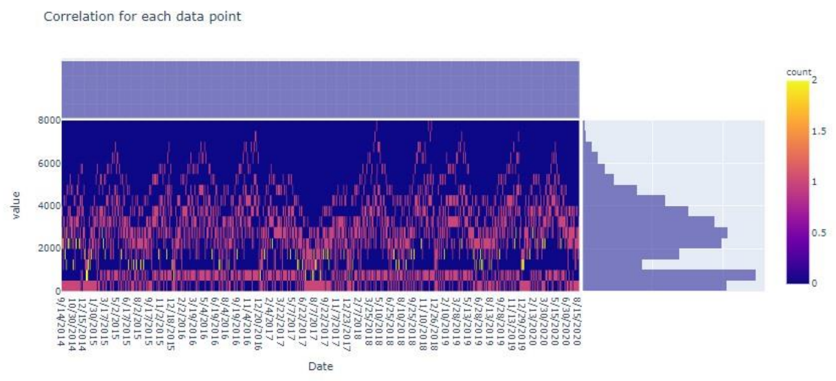
	page_loads	unique_visits	first_visits	returning_visits
Day				
Tuesday	1536154	1097181	907752	189429
Wednesday	1517114	1085624	897602	188022
Monday	1502161	1072112	886036	186076
Thursday	1437269	1028214	848921	179293
Friday	1149437	817852	668805	149047
Sunday	1006564	725794	604198	121596
Saturday	772817	552105	456449	95656

In [11]:

```
px.bar(sums,barmode='group',title='Sum of page loads and visits for each of their days')
```

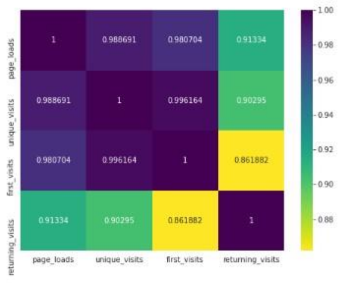


In [12]:
```
px.density_heatmap(df, x='Date',y=['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']
#                    color_continuous_scale="Viridis"
                   ,marginal_x="histogram", marginal_y="histogram",title='Correlation for each data point')
```



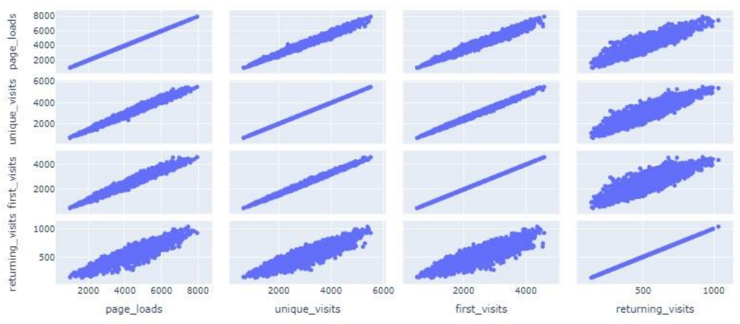
In [13]:
```
fig, ax = plt.subplots()
fig.set_size_inches(8, 6)
sns.heatmap(df[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']].corr(),
           annot=True,
           cmap='viridis_r',
           fmt='g')
```

Out[13]:
```
<AxesSubplot:>
```

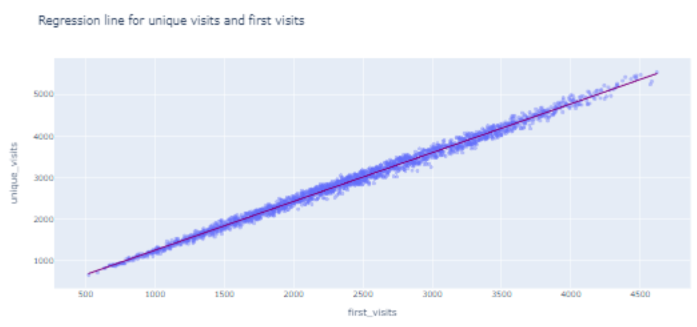


[illegible]

In [14]:
```
px.scatter_matrix(df[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']])
```

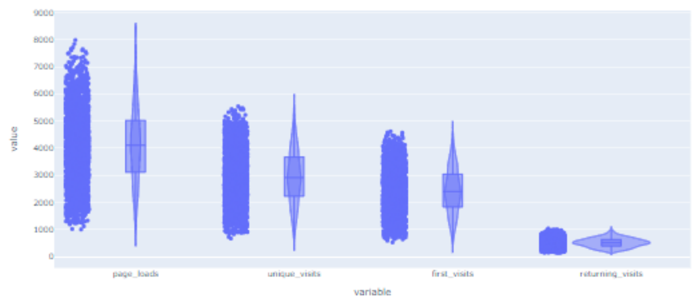


In [15]:
```
px.scatter(
    df, x='first_visits', y='unique_visits',opacity=0.4,
    trendline='ols', trendline_color_override='purple',title="Regression line for unique visits and first visits"
)
```



there are no outliears that need to be dealt with, data is tightly packed with no dispersion except for returning visits, this column was also less correlated with our target variable.

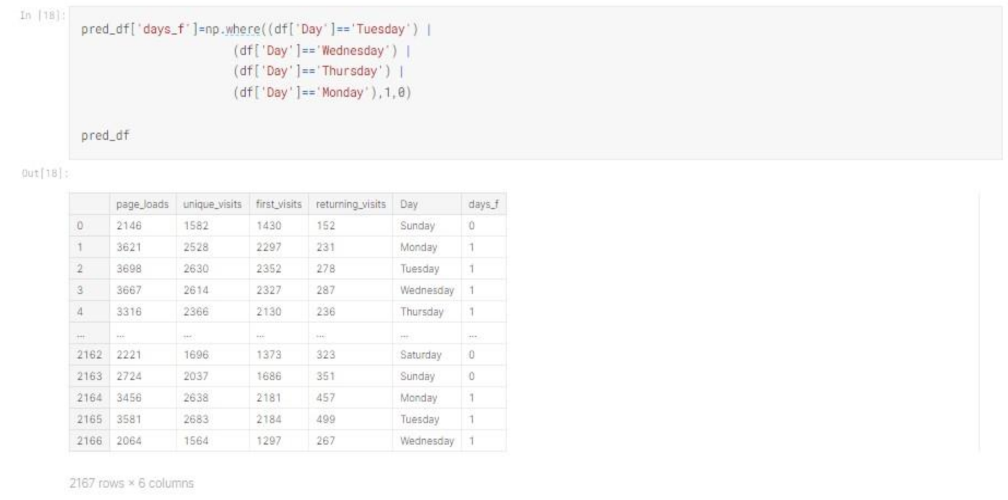
In [16]:
```
px.violin(df,y=['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits'],box=True,points='all')
```



Feature Engineering:

In [17]:
```
pred_df=df[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits','Day']]
```

- Create a new DataFrame `pred_df` as a copy of the original data, and add a new binary feature 'days_f' based on the day of the week. This feature is set to 1 for Monday, Tuesday, Wednesday, and Thursday, and 0 otherwise.



In [18]:
```
pred_df['days_f']=np.where((df['Day']=='Tuesday') |
                           (df['Day']=='Wednesday') |
                           (df['Day']=='Thursday') |
                           (df['Day']=='Monday'),1,0)

pred_df
```

Out[18]:

	page_loads	unique_visits	first_visits	returning_visits	Day	days_f
0	2146	1582	1430	152	Sunday	0
1	3621	2528	2297	231	Monday	1
2	3698	2630	2352	278	Tuesday	1
3	3667	2614	2327	287	Wednesday	1
4	3316	2366	2130	236	Thursday	1
...	...	...	...	...	...	...
2162	2221	1696	1373	323	Saturday	0
2163	2724	2037	1686	351	Sunday	0
2164	3456	2638	2181	457	Monday	1
2165	3581	2683	2184	499	Tuesday	1
2166	2064	1564	1297	267	Wednesday	1

2167 rows × 6 columns

Data Modelling Preparation:

- Define the feature matrix `X` and the target variable `y` for machine learning modeling. The features include 'page_loads', 'first_visits', 'returning_visits', and 'days_f'. The target variable is 'unique_visits'.

In [21]:
```
X2=pred_df[['page_loads','first_visits' ,'returning_visits','days_f']]
y2=pred_df['unique_visits']
```

Splitting data and Training the Model:

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split(X2, y2, test_size=0.3,random_state=42)
```

train the model with train sample

In [23]:
```
regressor2 = LinearRegression(fit_intercept=False,normalize=True)
regressor2.fit(X_train, y_train)
```

Out[23]:
```
LinearRegression(fit_intercept=False, normalize=True)
```

In [24]:
```
y_pred2 = regressor2.predict(X_test)
```

Model Evaluation:

In [25]:
```
lr2 = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred2})
lr2
```

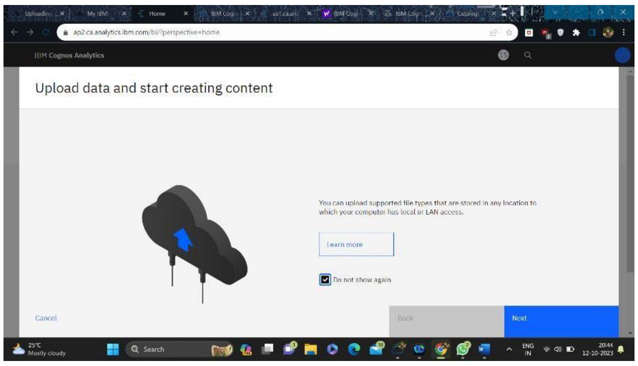
Out[25]:

	Actual	Predicted
1486	4173	4173.0
1602	1902	1902.0
1460	2870	2870.0
1134	2142	2142.0
1513	4329	4329.0
...	...	...
439	2579	2579.0
271	2494	2494.0
244	1818	1818.0
1159	3332	3332.0
1701	2565	2565.0

651 rows × 2 columns

➢ <u>Data Analysis and Visualization in IBM Cognos:</u>

Dataset Loading and Preprocessing:

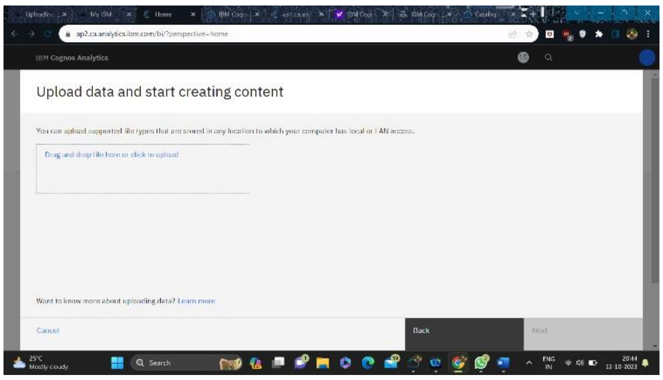


Upload data and start creating content

You can upload supported file types that are stored in any location to which your computer has local or LAN access.

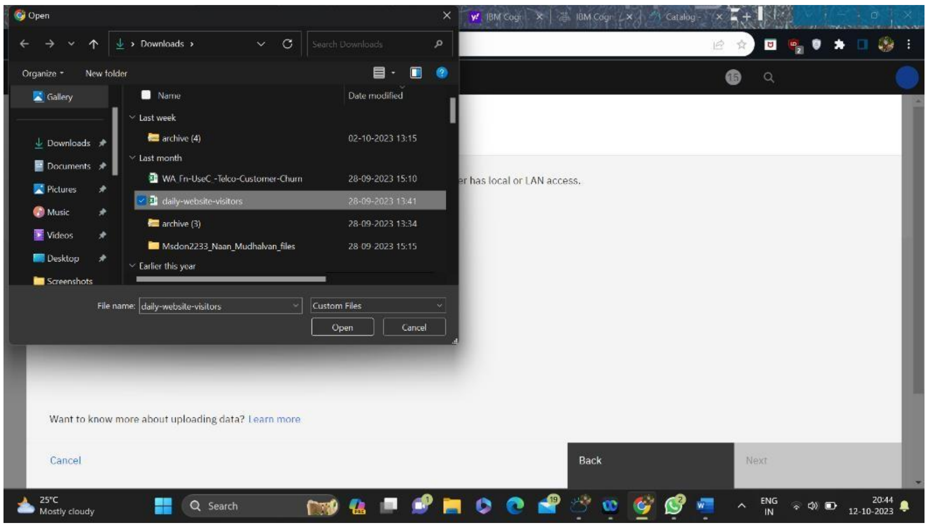
Loading or Uploading Dataset:

- In this phase of the project we are going upload our dataset for analysis which is already provide through Kaggle link in the SkillUp platform.
- For this purpose, first of all we have to access to the IBM Cloud services and IBM Cognos Analytics suit.
- All must create a IBM Account and register as lite account for gaining access to the some premium membership accessibilities.



IBM Cognos Analytics
Upload data and start creating content
You can upload supported file types that are stored in any location to which your computer has local or LAN access.
Drag and drop file here or click to upload
Want to know more about uploading data? Learn more
Cancel
Back
Next

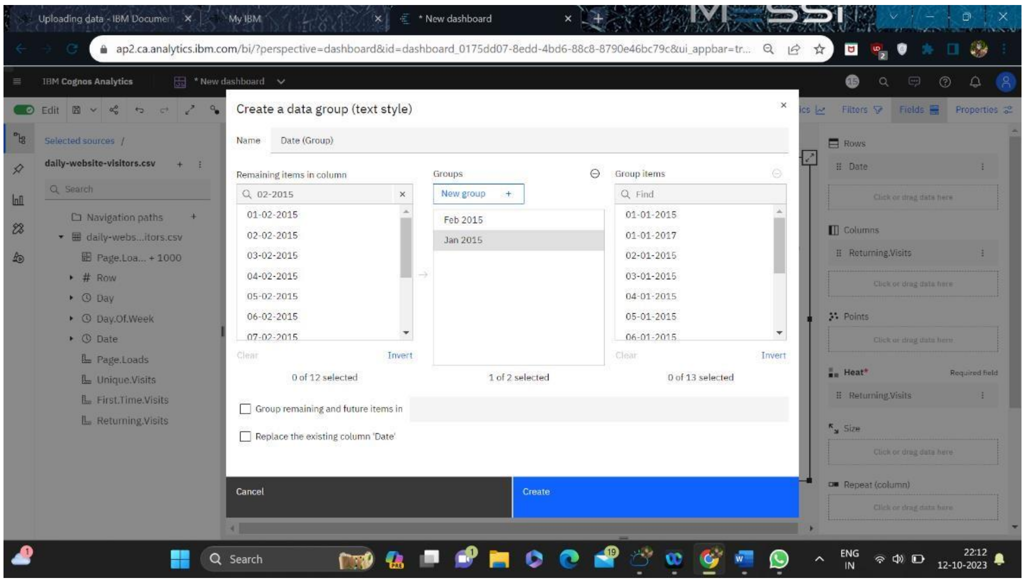
- A Lite account can use the features of IBM software of web applications upto a certain limits in time period , no.of sessions per month, storage limits etc.
- After the account is created and accessibility gained we have create a project and Upload our Dataset that need to be analysed into the My Contents section. Using selecting from storage option or drag and drop option.



Open
Downloads
Organize
New folder
Name
Date modified
Last week
archive (4) 02-10-2023 13:15
Last month
WA_Fn-UseC_-Telco-Customer-Churn 28-09-2023 15:10
daily-website-visitors 28-09-2023 13:41
archive (3) 28-09-2023 13:34
Msdon2233_Naan_Mudhalvan_files 28-09-2023 15:15
Earlier this year
File name: daily-website-visitors
Custom Files
Open
Cancel
Want to know more about uploading data? Learn more

Preprocessing Dataset:

- Clean the data by handling missing values , outliers , and data quality issues.
- Transform and reshape the data as needed . This might include feature engineering , data aggregation ,or other data preparation.


Create a data group (text style)

Name: Date (Group)

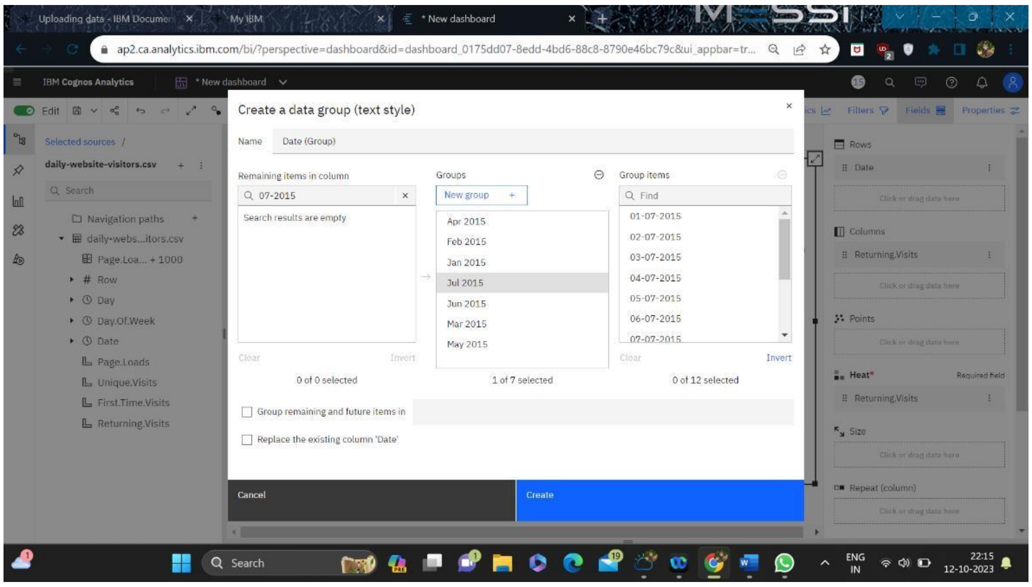
Remaining items in column: 02-2015 — 01-02-2015, 02-02-2015, 03-02-2015, 04-02-2015, 05-02-2015, 06-02-2015, 07-02-2015 — 0 of 12 selected

Groups: Feb 2015, Jan 2015 — 1 of 2 selected

Group items: 01-01-2015, 01-01-2017, 02-01-2015, 03-01-2015, 04-01-2015, 05-01-2015, 06-01-2015 — 0 of 13 selected

Group remaining and future items in

Replace the existing column 'Date'


Create a data group (text style)

Name: Date (Group)

Remaining items in column: 07-2015 — Search results are empty — 0 of 0 selected

Groups: Apr 2015, Feb 2015, Jan 2015, Jul 2015, Jun 2015, Mar 2015, May 2015 — 1 of 7 selected

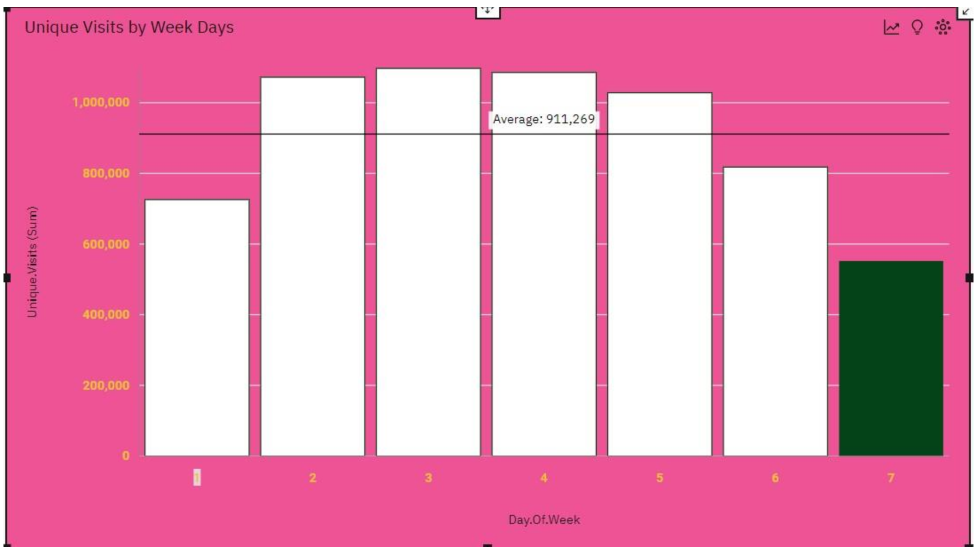
Group items: 01-07-2015, 02-07-2015, 03-07-2015, 04-07-2015, 05-07-2015, 06-07-2015, 07-07-2015 — 0 of 12 selected

Group remaining and future items in

Replace the existing column 'Date'

- Data Preprocess is one of the crucial step in the Data Analytics . Otherwise our analysis can be irrelevant due to unwanted noises in the data.
- This includes null value removal , formatting ,etc.

Basic Analysis and Visualization (Dashboard Creation):



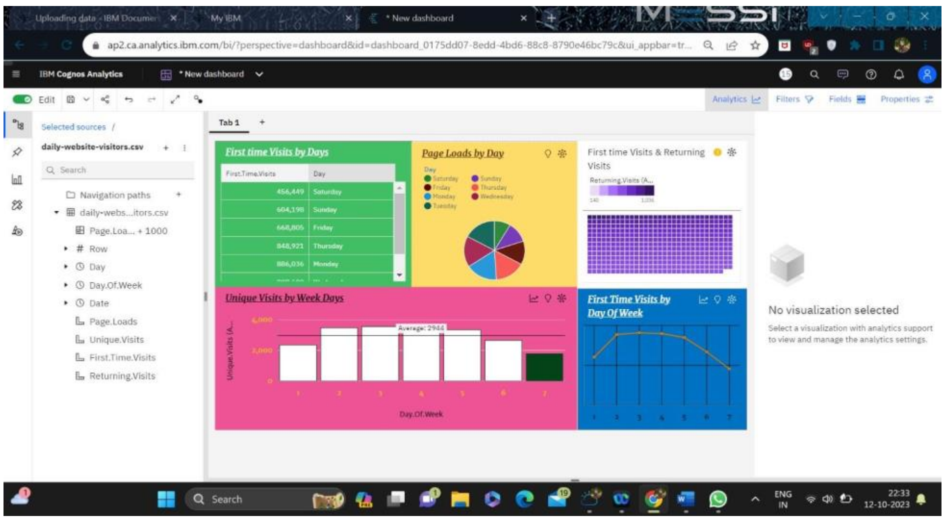
Analysis:

- Use IBM Cognos for descriptive and exploratory data analysis. Conduct statistical analysis to understand the dataset's characteristics.

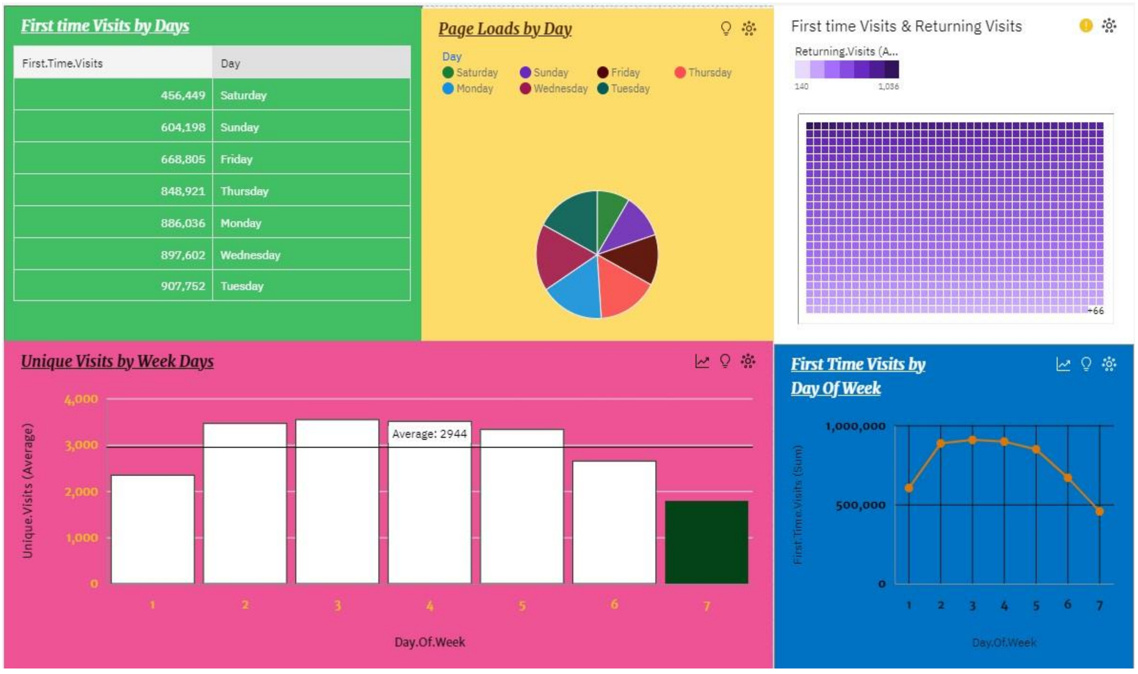
- Identify patterns, trends, and correlations within the data. Look for potential factors that may contribute to website traffic analysis.

Visualization:

- Create visualizations using IBM Cognos to present your findings. This might include:
- Histograms, scatter plots, and box plots to visualize data distributions and relationships.
- Line charts or time series visualizations to explore trends over time.
- Dashboards that summarize key insights in a user-friendly manner.



➢ Basic Dashboard:



First time Visits by Days

First.Time.Visits	Day
456,449	Saturday
604,198	Sunday
668,805	Friday
848,921	Thursday
886,036	Monday
897,602	Wednesday
907,752	Tuesday

➢ Conclusion:

Website traffic analysis using IBM Cognos integrated with Python code can provide valuable insights to website owners, helping them improve the user experience in several ways:

Understanding User Behavior: By analyzing website traffic data, website owners can gain insights into how users interact with their site. They can identify popular pages, the flow of user journeys, and the content that attracts the most attention. This knowledge can help optimize the layout and content of the site for a better user experience.

Identifying Traffic Sources: Website owners can determine the sources of website traffic, such as organic search, social media, paid advertising, or direct traffic. This information helps in allocating resources to the most effective marketing channels and improving the targeting of campaigns.

User Segmentation: With website traffic analysis, you can segment users based on various criteria, such as demographics, location, device type, or referral source. Segmenting users allows website owners to tailor content and features to specific audience groups, enhancing user satisfaction.

Detecting Bounce Rates: High bounce rates (users leaving the site quickly) can indicate issues with the landing page or content. By identifying pages with high bounce rates, website owners can optimize those pages to keep users engaged.

Conversion Rate Optimization (CRO): Analyzing conversion funnels and user behavior can help website owners identify bottlenecks in the conversion process. They can then make adjustments to improve conversion rates, leading to increased user satisfaction and achieving business goals.

Page Load Times: Website speed is critical for user experience. Website traffic analysis can provide insights into the average page load times and highlight slow-loading pages. Website owners can take steps to optimize these pages for a faster user experience.

Content Effectiveness: By analyzing which content generates the most traffic, engagement, and conversions, website owners can refine their content strategy. This includes creating more of what works and eliminating or updating less effective content.

Device Compatibility: Analyzing user traffic by device type (desktop, mobile, tablet) can help ensure that the website is responsive and provides an optimal experience on all devices. Mobile optimization is particularly important as mobile users continue to grow.

A/B Testing: Website traffic analysis can help identify areas for A/B testing. By running experiments and comparing user engagement and conversion rates, website owners can refine design, content, and functionality based on real user preferences.

Data-Driven Decision Making: Combining IBM Cognos with Python code allows for advanced data analysis and predictive modeling. By using machine learning and predictive analytics, website owners can forecast trends, make data-driven decisions, and continuously improve the user experience.

Security and Anomaly Detection: Monitoring website traffic can also help detect suspicious activities, such as unusual traffic patterns or security threats. Identifying and addressing these issues promptly enhances the security and trustworthiness of the website.

In summary, website traffic analysis using IBM Cognos and Python empowers website owners with data-driven insights to make informed decisions, optimize their websites, and enhance the overall user experience. By understanding user behavior, traffic sources, and content effectiveness, they can tailor their websites to better meet user expectations and achieve their business objectives.