# WEBSITE TRAFFIC ANALYSIS

## ABSTRACT

The process of collecting, analyzing, and interpreting data about the interactions and behaviors of users on a website. This data includes metrics like page views, bounce rates, click-through rates, and more, which help in evaluating the website's effectiveness and making data-driven decisions for improvement.

# OBJECTIVES

Website traffic analysis serves several critical objectives for businesses and website owners.
It provides valuable insights into user behavior, allowing them to understand how visitors interact with their site. This data can be used to optimize the website's user experience, making it user-friendly.
website traffic analysis helps in measuring the effectiveness of marketing campaigns. By tracking the source of traffic, businesses can determine which channels are driving the most visitors and conversions. This information guides strategy adjustments and budget allocation for maximum ROI.

# *DESIGN THINKING*

- Empathize
- Define
- Ideate
- Prototype
- Test
- Iterate

# EMPATHIZE

"Empathize" involves understanding users' perspectives, needs, and pain points. It requires engaging with visitors, conducting user interviews, and gathering data to gain deep insights into their behaviors and preferences.

This empathetic approach helps designers create user-centric strategies, content, and experiences that ultimately boost website traffic and engagement.

# DEFINE

"Define" stage is about precisely identifying the problem or challenge related to web traffic. It involves gathering user feedback, analytics data, and stakeholder input to create a clear and actionable problem statement.

This step ensures a focused and shared understanding of the issues at hand, serving as a foundation for subsequent ideation and solution development in the design process.

# IDEATE

"Ideate" in Design Thinking for Website Traffic Analysis involves brainstorming creative solutions and ideas to enhance web traffic. Teams generate a variety of concepts, such as content optimizations, SEO strategies, or user experience improvements.

This step encourages diverse thinking and innovation, helping uncover unique approaches to attract and retain visitors, ultimately leading to data-driven solutions that can be implemented to boost website traffic.

# PROTOTYPE

A prototype in Design Thinking for Website Traffic Analysis would be a preliminary version of the analytical tool, showcasing its key features and user interface.

This visual representation allows designers and stakeholders to interact with the tool, gather feedback, and refine it before development, ensuring that the final product effectively meets user needs and goals while enhancing website traffic analysis capabilities.

# TEST

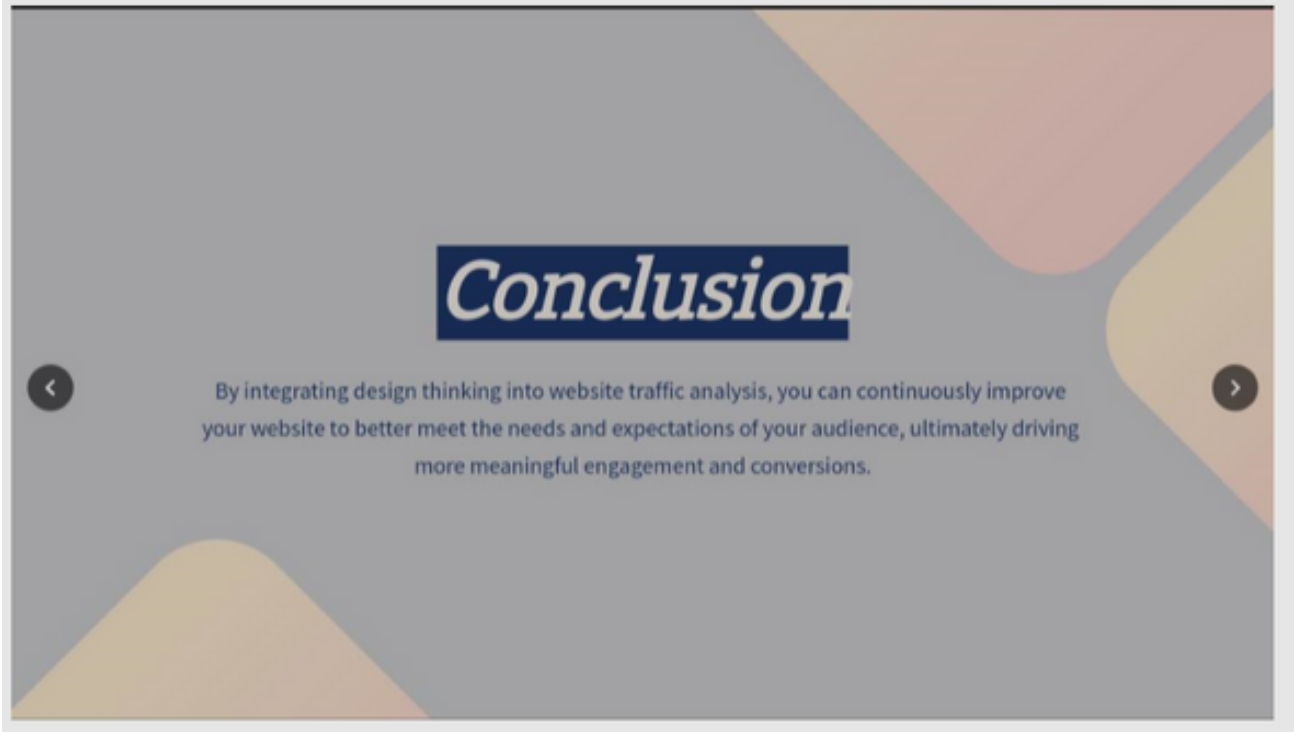
A Design Thinking test for Website Traffic Analysis involves applying empathy, ideation, and prototyping to enhance website performance. Teams empathize with users, generate creative ideas for engagement, and prototype solutions.

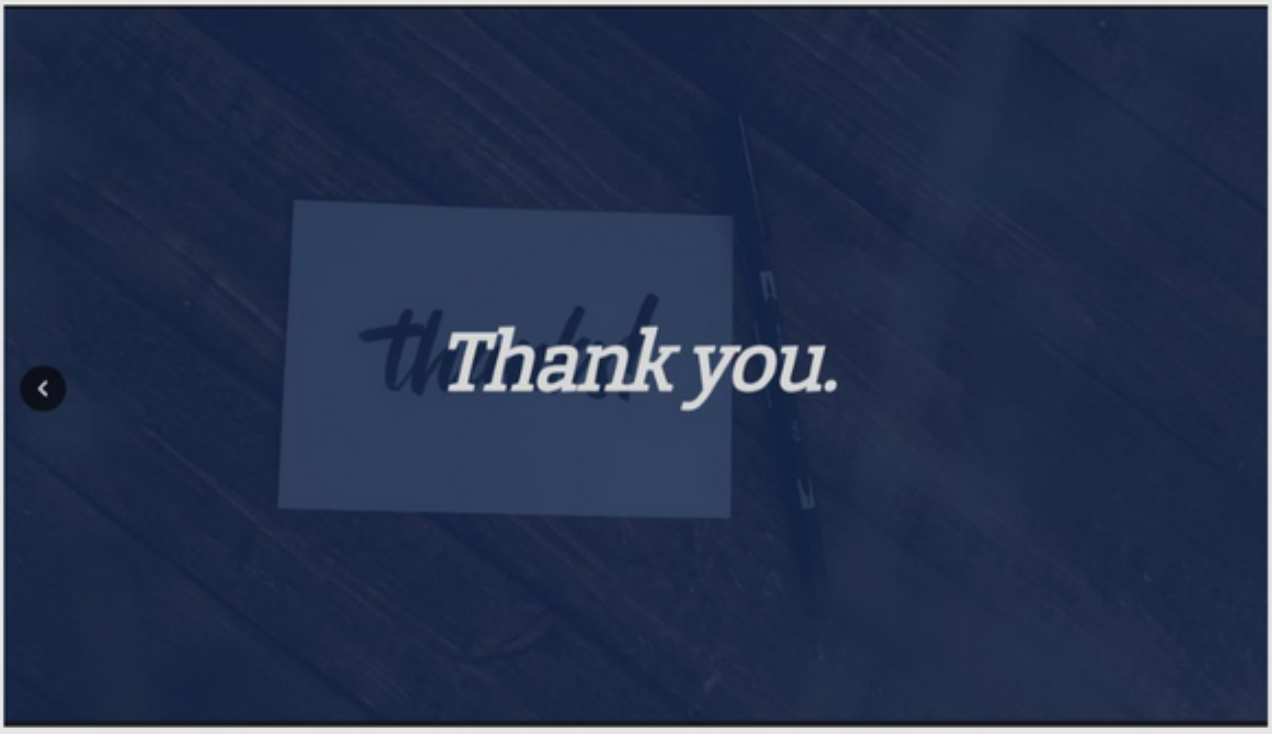
User feedback and data guide iterations. This iterative, user-centric approach aims to boost website traffic and user satisfaction.

# ITERATE

In Design Thinking for Website Traffic Analysis, iteration is a crucial step. It involves continually reviewing and refining your analysis methods and insights.

This iterative process ensures In Design Thinking for Website Traffic Analysis, iteration is a crucial step. It involves continually reviewing and refining your analysis methods and insights. This iterative process ensures that your strategies for improving website traffic are data-driven and adaptable, allowing you to make informed adjustments and maximize the effectiveness of your website's user experience and content.


Conclusion
By integrating design thinking into website traffic analysis, you can continuously improve your website to better meet the needs and expectations of your audience, ultimately driving more meaningful engagement and conversions.


Thank you.

# FRAUD DETECTION USING MACHINE LEARNING

**Team**member

810021205075: RITHIK

Phase**2: Innovation**

## PROJECT DEFINITION :

Detecting fraud in future website traffic trends or user behavior patterns typically involves utilizing advanced machine learning and data analytics techniques

## INTRODUCTION:

In the ever-evolving digital landscape, the battle against online fraud is an ongoing challenge for businesses and organizations. As technology advances, so do the tactics employed by malicious actors seeking to exploit vulnerabilities in websites and online services. Detecting and preventing fraud in future website traffic trends and user behavior patterns is crucial to maintaining the trust of customers and the integrity of digital ecosystems.

## ABSTRACTION:

Fraud detection in the realm of website traffic and user behavior patterns is a multifaceted task that combines cutting-edge technology, data analysis, and predictive modeling. It involves the abstraction of complex data points and the identification of anomalous activities that may indicate fraudulent behavior.

**Here’s a high-level overview of the process**:

- Data Collection
- DataPreprocessing
- Feature Engineering
- Anomaly Detection
- Supervised Learning
- Behavioral Analysis
- Real-time Monitoring
- Thresholds and Alerts
- Feedback Loop
- User Authentication

1. **Data Collection**:

Gather data on website traffic and user behavior. This can include user interactions, login attempts, transactions, IP addresses, geolocation, and more. The more data you have, the better your fraud detection system can perform.

**2.DataPreprocessing:**

Clean and preprocess the data. This involves handling missing values, normalizing data, and converting categorical variables into numerical formats.

**3.Feature Engineering:**

Create relevant features from the data that can help in fraud detection. These features could include user activity patterns, session duration, IP address history, and more.

**4. Detection:**

Employ anomaly detection techniques like Isolation Forests, One-Class SVM, or autoencoders to identify unusual or

fraudulent behavior. These methods can flag activities that deviate significantly from the norm.

**5.Supervised Learning:**

Utilize supervised machine learning algorithms to build predictive models. Train the model on historical data where fraud labels are known. Algorithms like Random Forest, Gradient Boosting, or Neural Networks can be effective.

6.**Behavioral Analysis**:

Analyze user behavior patterns over time. Look for sudden spikes or drops in certain activities, which might indicate fraud.

**7.Real-time Monitoring**:

Implement real-time monitoring systems to detect fraud as it occurs. This involves continuously feeding incoming data to your model and flagging suspicious activities.

**8.Thresholds and Alerts**:

Set thresholds for when activity is considered suspicious. If an event surpasses this threshold, generate alerts for further investigation.

**9.Feedback Loop**:

Continuously update and retrain your fraud detection model as new data becomes available. This helps in adapting to evolving fraud techniques.

**10.User Authentication**:

Implement strong user authentication methods, such as multi-factor authentication, to reduce the risk of unauthorized access.

**Conclusion:**

Keep in mind that fraudsters are constantly evolving their tactics, so your fraud detection system should be agile and adaptable. Regularly assess its performance and update it as needed to stay ahead of emerging threats.

***

# WEBSITE TRAFFIC ANALYSIS

## DATA ANALYTICS WITH COGNOS : GROUP 2

## PHASE : 3

This phase involves in designing of the steps that defining in each phase of the previous documentation this involves importing necessary functions, data processing and so on in this phase we have to begin our project by loading and preprocessing the dataset.

The IBM suggests using the jupyter notebook for loading and preprocess the dataset:

Here for this project title we need to define the loading the libraries, understand the data and visualize the missing values.

For this certain inputs are defined for this project.in this phase each of the input

Codes of project is given below:

untitled7

October 18, 2023

[ ]: PHASE 3

[1]: import pandas as pd import numpy as np import missingno as msno

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

[3]: df.head()

[3]: 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

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

[4]: df.tail()

[4]: Row Day Day.Of.Week Date Page.Loads Unique.Visits \
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 2162 1,373 323 2163 1,686 351 2164 2,181 457

2165 2,184 499 2166 1,297 267

[5]: df.shape

[5]: (2167, 8)

[6]: df.info()

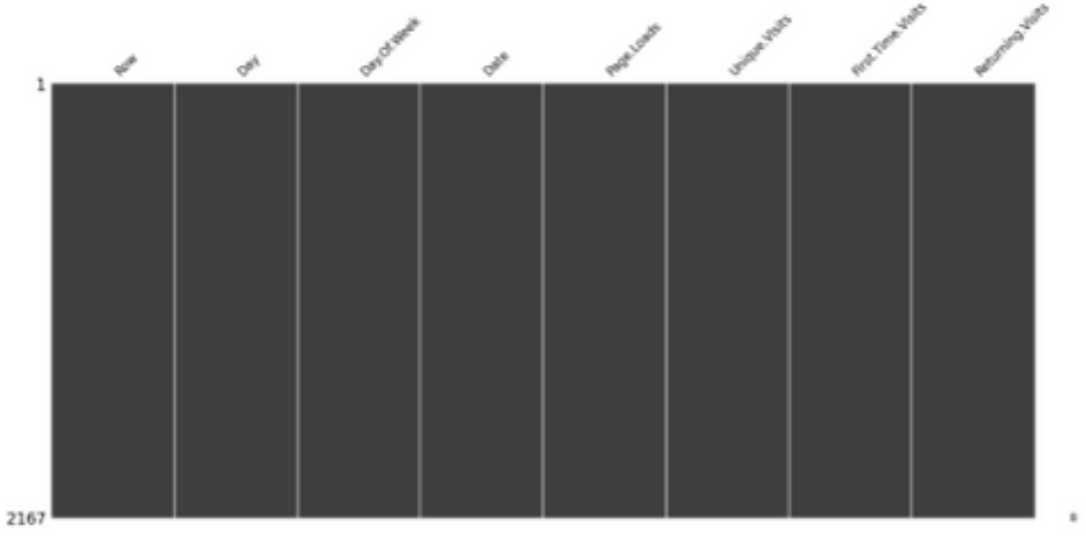
<class 'pandas.core.frame.DataFrame'> RangeIndex: 2167 entries, 0 to 2166 Data columns (total 8 columns): # Column Non-Null Count Dtype --- ------ -------------- ----0 Row 2167 non-null int64 1 Day 2167 non-null object 2 Day.Of.Week 2167 non-null int64 3 Date 2167 non-null object 4 Page.Loads 2167 non-null 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 [7]: df.columns.values

[7]: array(['Row', 'Day', 'Day.Of.Week', 'Date', 'Page.Loads', 'Unique.Visits', 'First.Time.Visits', 'Returning.Visits'], dtype=object)

[8]: df.dtypes

[8]: Row int64 Day object Day.Of.Week int64 Date object Page.Loads object Unique.Visits object First.Time.Visits object Returning.Visits object dtype: object

[9]: msno.matrix(df);




[10]: df = df.drop(['Unique.Visits'],axis = 1) df.head()

[10]: Row Day Day.Of.Week Date Page.Loads First.Time.Visits \ 0 1 Sunday 1 9/14/2014 2,146 1,430 1 2 Monday 2 9/15/2014 3,621 2,297 2 3 Tuesday 3 9/16/2014 3,698 2,352 3 4 Wednesday 4 9/17/2014 3,667 2,327 4 5 Thursday 5 9/18/2014 3,316 2,130

Returning.Visits 0 152 1 231 2 278 3 287 4 236

[11]: df.isnull()

[11]: Row Day Day.Of.Week Date Page.Loads First.Time.Visits \ 0 False False False False False False 1 False False False False False False 2 False False False False False False 3 False False False False False False 4 False False False False False False … … … … … … … … 2162 False False False False False False 2163 False False False False False False 2164 False False False False False False

2165 False False False False False False 2166 False False False False False False

Returning.Visits 0 False 1 False 2 False 3 False 4 False … … 2162 False 2163 False 2164 False 2165 False 2166 False

[2167 rows x 7 columns]

[12]: df.isnull().sum()

[12]: Row 0 Day 0 Day.Of.Week 0 Date 0 Page.Loads 0 First.Time.Visits 0 Returning.Visits 0 dtype: int64

[13]: df['Row'] = pd.to_numeric(df.Row,errors='coerce') df.isnull().sum()

[13]: Row 0 Day 0 Day.Of.Week 0 Date 0 Page.Loads 0 First.Time.Visits 0 Returning.Visits 0 dtype: int64

[14]: df[np.isnan(df['Row'])]

[14]: Empty DataFrame Columns: [Row, Day, Day.Of.Week, Date, Page.Loads, First.Time.Visits, Returning.Visits] Index: []

[15]: df.fillna(df['Row'].mean())

[15]: Row Day Day.Of.Week Date Page.Loads First.Time.Visits \ 0 1 Sunday 1 9/14/2014 2,146 1,430 1 2 Monday 2 9/15/2014 3,621 2,297 2 3 Tuesday 3 9/16/2014 3,698 2,352 3 4 Wednesday 4 9/17/2014 3,667 2,327 4 5 Thursday 5 9/18/2014 3,316 2,130 … … … … … … … 2162 2163 Saturday 7 8/15/2020 2,221 1,373 2163 2164 Sunday 1 8/16/2020 2,724 1,686 2164 2165 Monday 2 8/17/2020 3,456 2,181 2165 2166 Tuesday 3 8/18/2020 3,581 2,184 2166 2167 Wednesday 4 8/19/2020 2,064 1,297

Returning.Visits 0 152 1 231 2 278 3 287 4 236 … … 2162 323 2163 351 2164 457 2165 499 2166 267

[2167 rows x 7 columns]

[16]: df["Date"] = pd.to_datetime(df["Date"],format="%m/%d/%Y") print(df.info())

<class 'pandas.core.frame.DataFrame'> RangeIndex: 2167 entries, 0 to 2166 Data columns (total 7 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 datetime64[ns] 4 Page.Loads 2167 non-null object 5 First.Time.Visits 2167 non-null object 6 Returning.Visits 2167 non-null object dtypes: datetime64[ns](1), int64(2), object(4) memory usage: 118.6+ KB

None [17]: df.isnull().sum()

[17]: Row 0 Day 0 Day.Of.Week 0 Date 0 Page.Loads 0 First.Time.Visits 0 Returning.Visits 0 dtype: int64

[18]: df["Returning.Visits"]=df['Returning.Visits'].map({0:"no", 1: "yes"}) df.head()

[18]: Row Day Day.Of.Week Date Page.Loads First.Time.Visits \ 0 1 Sunday 1 2014-09-14 2,146 1,430 1 2 Monday 2 2014-09-15 3,621 2,297 2 3 Tuesday 3 2014-09-16 3,698 2,352 3 4 Wednesday 4 2014-09-17 3,667 2,327 4 5 Thursday 5 2014-09-18 3,316 2,130

Returning.Visits 0 NaN 1 NaN 2 NaN 3 NaN 4 NaN

[19]: df["Returning.Visits"].describe(include=['object','bool'])

[19]: count 0 unique 0 top NaN freq NaN Name: Returning.Visits, dtype: object

[20]: df[df['Row'] == 0].index

[20]: Int64Index([], dtype='int64')

[21]: numerical_cols = ['Row','First.Time.Visits','Returning.Visits'] df[numerical_cols].describe()

[21]: Row count 2167.000000

mean 1084.000000 std 625.703338 min 1.000000 25% 542.500000 50% 1084.000000 75% 1625.500000 max 2167.000000

[ ]:

**College code:8100**

**Register number:8100212053075**

# Website Traffic Analysis

## DATA ANALYTICS WITH COGNOS : GROUP 5

## PHASE : 4

### Project Overview

The Website Traffic Analysis project aims to assess and understand the patterns, trends, and user behaviour on a specific website. By collecting and analysing data, the project will provide insights into the site's performance, user engagement, and areas for improvement. The results will inform decision-making, helping to optimize the website's content, design, and marketing strategies for enhanced user experience and increased traffic.

### OBJECTIVE

Equipped with the right website traffic analysis tools, identify your top site pages, track visitor trends, calculate conversion rates, and ensure your marketing spend translates into an increase in conversions and sales.

### 1.DATA EXTRACTION:

#### INSTALLATION OF JUPYTER NOTEBOOK

Command to install jupyter notebook:

```
pip install jupyter notebook
```

Output:

```
Requirement already satisfied: jupyter in c:\users\dhaya\.conda\lib\site-packages (1.0.0)
Requirement already satisfied: notebook in c:\users\dhaya\.conda\lib\site-packages (6.5.4)
Requirement already satisfied: qtconsole in c:\users\dhaya\.conda\lib\site-packages (from jupyter) (5.4.2)
Requirement already satisfied: jupyter-console in c:\users\dhaya\.conda\lib\site-packages (from jupyter) (6.6.3)
```

#### WORKING OF JUPYTER NOTEBOOK:

**Command to open jupyternotbook:**

```
jupyter notebook
```

**Output:**


**EXTRACTION:**

Packages needed:

To extract command for those modules:

```
pip install numpy
pip install pandas
```

**Input:**

```
import numpy as np
import pandas as pd
 FILE_LOCATION = "P:\ibm\daily-website-visitors.csv"
 df = pd.read_csv(FILE_LOCATION,
                            index_col='Date',
                            thousands=',')
 df.index = pd.to_datetime(df.index)
 df
```

**CODE DESCRIPTION:**

- The code starts by importing the pandas library.
- The code then creates a variable called FILE_LOCATION and assigns it to the path of a file on your computer.
- Next, the code reads in that CSV file into a DataFrame object using read_csv().
- The index column is set to Date, which means that this DataFrame will have one row for each day of data.
- The index_col='Date' parameter specifies that the column with the date should be used as the index.
- The thousands=',' parameter tells pandas to use commas for thousands separators in this column.

- The code opens a file called "P:\ibm\daily-website-visitors.csv" and reads in the data using csv.reader().
- The data is then stored in a list, which is assigned to variable "data".
- Next, numpy is imported as np so that we can use it to analyze the data.
- Finally, the first row of our dataset is analyzed with np.array() and printed out on screen for us to see what's going on with this dataset.
- The code will open the file "P:\ibm\daily-website-visitors.csv" and read the data in as a list of tuples, one for each row of data.

**Output:**

| Date | Row | Day | Day.Of.Week | Page.Loads | Unique.Visits | First.Time.Visits | Returning.Visits |
|---|---|---|---|---|---|---|---|
| **2014-09-14** | 1 | Sunday | 1 | 2146 | 1582 | 1430 | 152 |
| **2014-09-15** | 2 | Monday | 2 | 3621 | 2528 | 2297 | 231 |
| **2014-09-16** | 3 | Tuesday | 3 | 3698 | 2630 | 2352 | 278 |
| **2014-09-17** | 4 | Wednesday | 4 | 3667 | 2614 | 2327 | 287 |
| **2014-09-18** | 5 | Thursday | 5 | 3316 | 2366 | 2130 | 236 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **2020-08-15** | 2163 | Saturday | 7 | 2221 | 1696 | 1373 | 323 |
| **2020-08-16** | 2164 | Sunday | 1 | 2724 | 2037 | 1686 | 351 |
| **2020-08-17** | 2165 | Monday | 2 | 3456 | 2638 | 2181 | 457 |
| **2020-08-18** | 2166 | Tuesday | 3 | 3581 | 2683 | 2184 | 499 |
| **2020-08-19** | 2167 | Wednesday | 4 | 2064 | 1564 | 1297 | 267 |

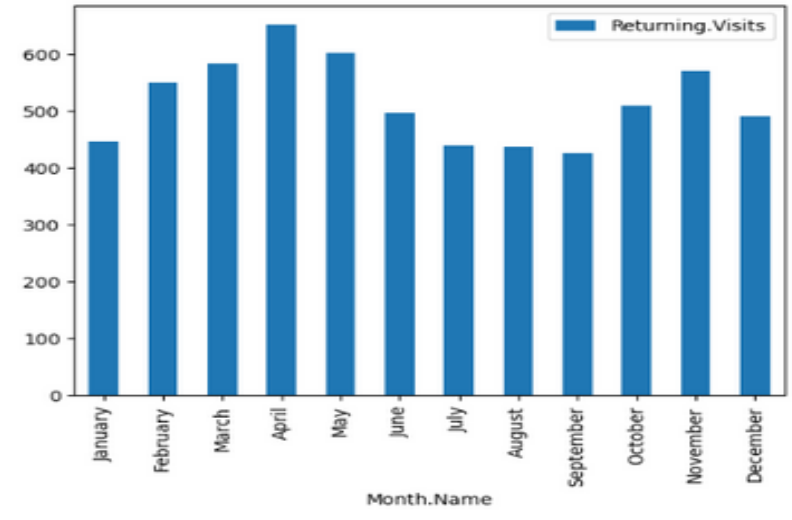
2167 rows × 7 columns

**EXTRACTION WITH VISUALIZATION:**

Extraction is the process of retrieving or pulling data from one or more sources. These sources can be diverse and include databases, spreadsheets, web services, logs, and more.

Transformation involves manipulating, cleaning, and structuring the data to make it suitable for the desired use case. This can include operations like filtering, aggregating, joining, and more.

```
pd.DataFrame(dataset_group_by_month['Returning.Visits'].mean()).loc[MONTH_NAMES].plot(kind='bar')
plt.show()
```

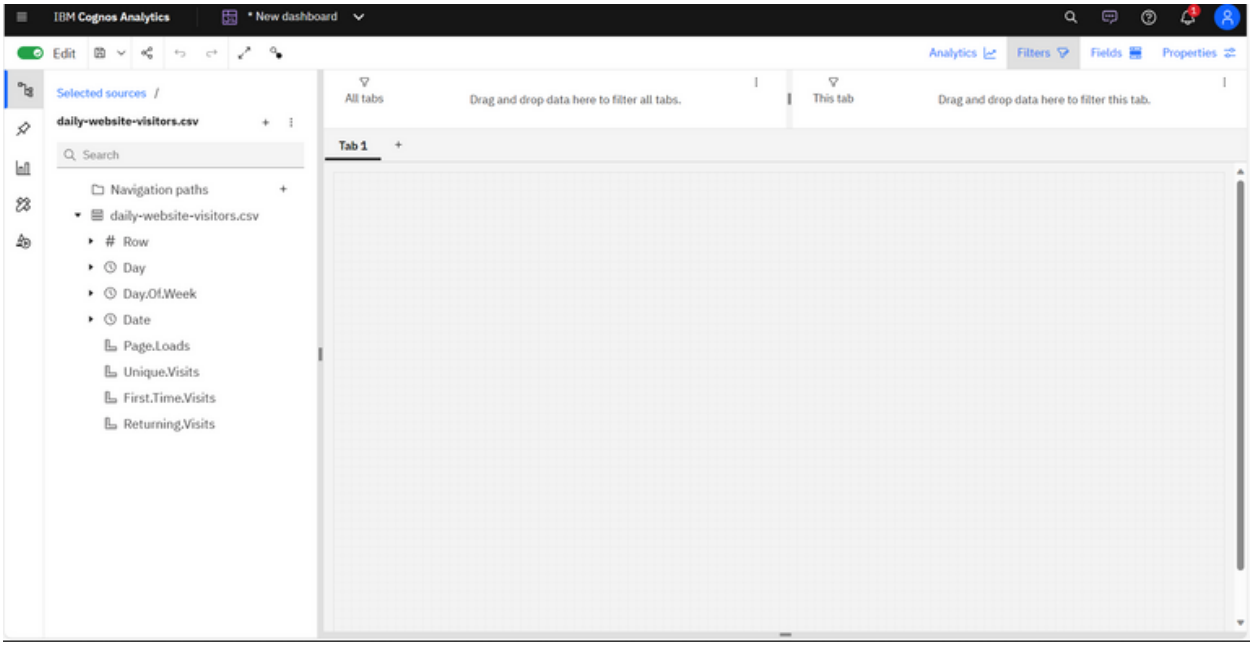



## 2.COGNOS ANALYTICS IN ACTION

Cognos Analytics serves as a powerful tool to transform our analytical findings into actionable insights. This section explores the utilization of Cognos Analytics in enhancing the accessibility and applicability of our results.

### Dashboard Design

We delve into the process of designing intuitive dashboards within Cognos Analytics. These dashboards serve as a centralized hub for visualizing key metrics, trends, and predictions derived from our analysis.




## Report Generation

Cognos Analytics enables the creation of comprehensive reports summarizing the outcomes of our efficiency analysis. This section outlines the steps involved in generating reports that cater to various stakeholders, providing customized views based on their informational needs.

### Templates and themes

IBM Cognos Analytics includes several basic report templates and color themes that you can choose from when you create a new report.

### Adding data

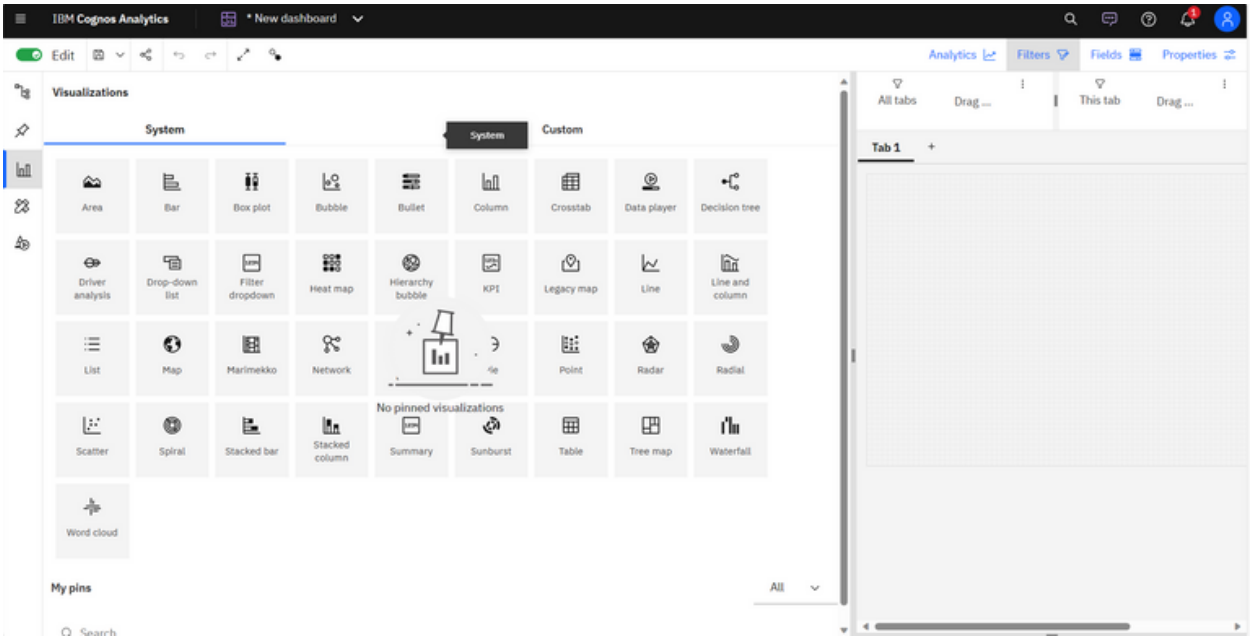
Add data to a report by loading packages or data modules in IBM Cognos Analytics - Reporting.

### Inserting a single data item

You can insert a single data item anywhere in your report using the singleton object. The singleton object retrieves only the first row value for that query. Inserting a single data item is useful when you want to show a value that is independent from the rest of the values in the report or when you want to insert some boilerplate text, such as a company name and address. For example, you can add the total revenue value in the header of each page in a report.

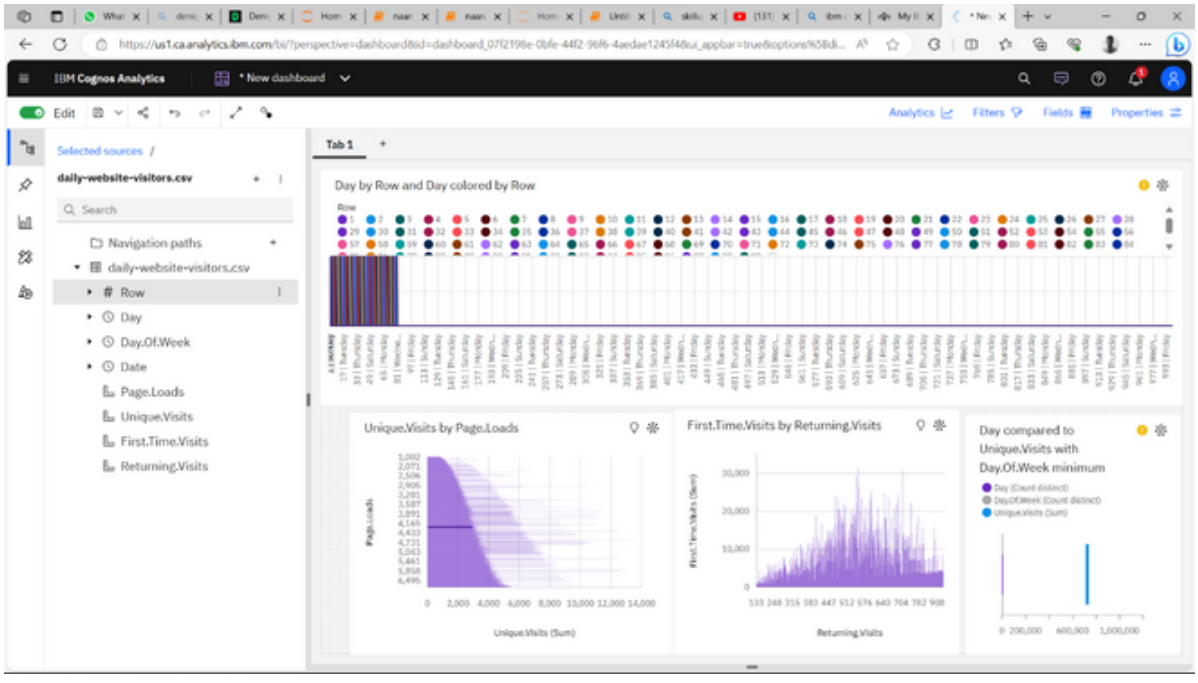
## Continuous Improvement

In a world where consumer preferences and market dynamics are in constant flux, the ability to adapt and refine sales strategies is paramount. This process begins with the collection of high-quality data, which serves as the foundation for informed decision-making. The art of continuous improvement lies in the willingness to evolve – iterating sales and marketing approaches, fostering cross-functional collaboration, and investing in the growth of sales teams.




## Actionability

The practicality of our recommendations was assessed through engagement with stakeholders responsible for implementing changes based on our analysis.




## Conclusion

Website traffic analysis is the compass that guides the visitors of the websites. It provides critical insights that are fundamental to success. By examining  data, inspection can make informed decisions about their usual, regular, unique visitors of the website. This analysis makes the average view of visitors of the website, then by upgrading to new things in the website makes the visitors more conventional and widely used for the future visits in the website.