**Phase 4 : Development Part 2**

**Website Traffic Analysis**

**Data Preparation:**

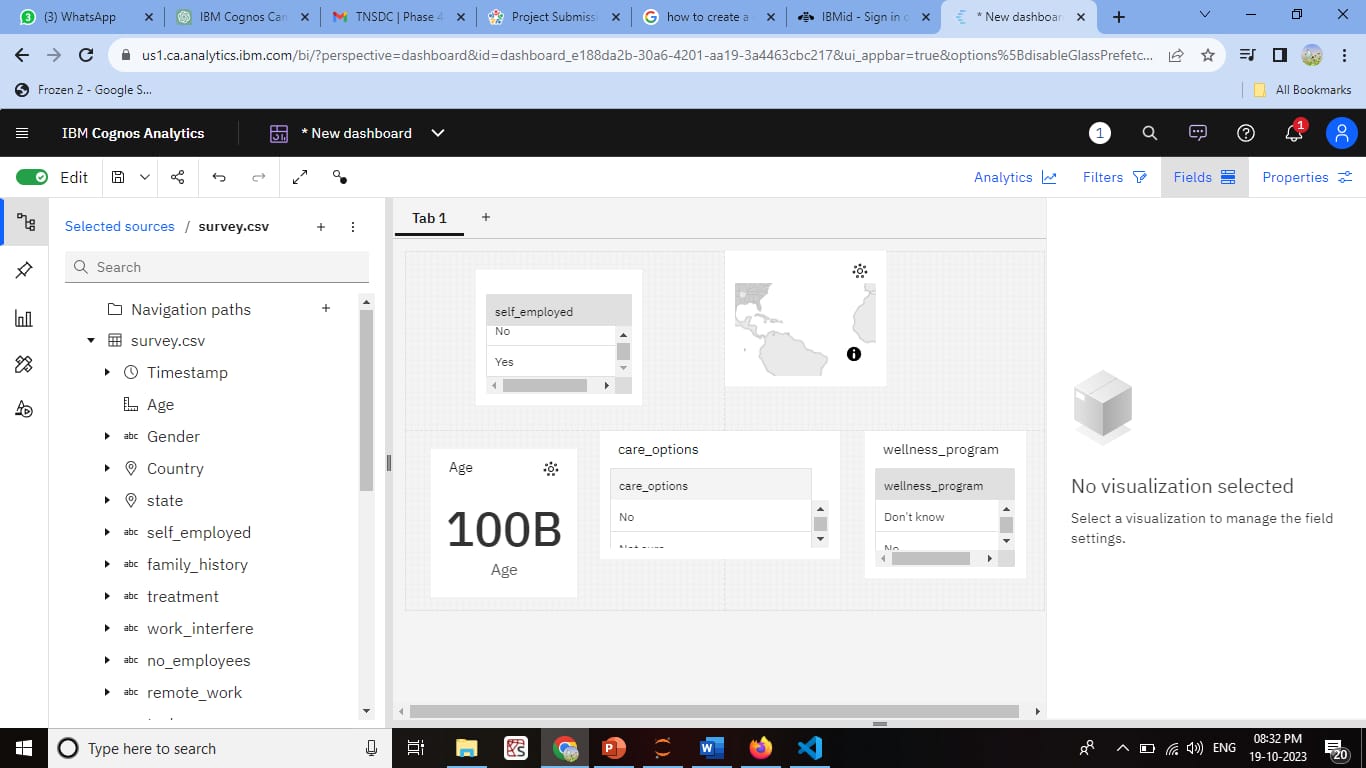
Start by collecting and cleaning your website analytics data. Ensure that the data is in a suitable format, and you've dealt with missing or erroneous values.

You can use Pandas in Python to manipulate and clean your data. For instance, you can load your data using pandas.read\_csv(), drop or fill missing values, and filter relevant columns.

**IBM Cognos Dashboard:**

Create a new dashboard in IBM Cognos that displays key insights.

For example:



Popular Pages:

Create a report that lists the most visited pages on your website. You can use Cognos' report authoring tools to generate this data.

Traffic Sources:

Show a pie chart or bar chart displaying the sources of traffic (organic search, direct, referral, etc.).

User Engagement Metrics:

Use Cognos to generate visualizations for metrics like bounce rate, average session duration, and conversion rates.

**Integration with Python:**

For more complex analysis, you can integrate Python into IBM Cognos using the Python integration feature. Here's a simplified example of how to do this:

**Program**

import pandas as pd

import matplotlib.pyplot as plt

from scipy.stats import linregress

data = pd.read\_csv("your\_analytics\_data.csv")

data['Date'] = pd.to\_datetime(data['Date'])

data.set\_index('Date', inplace=True)

monthly\_data = data.resample('M').sum()

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

plt.plot(monthly\_data.index, monthly\_data['PageViews'], label='Page Views')

plt.plot(monthly\_data.index, monthly\_data['Users'], label='Users')

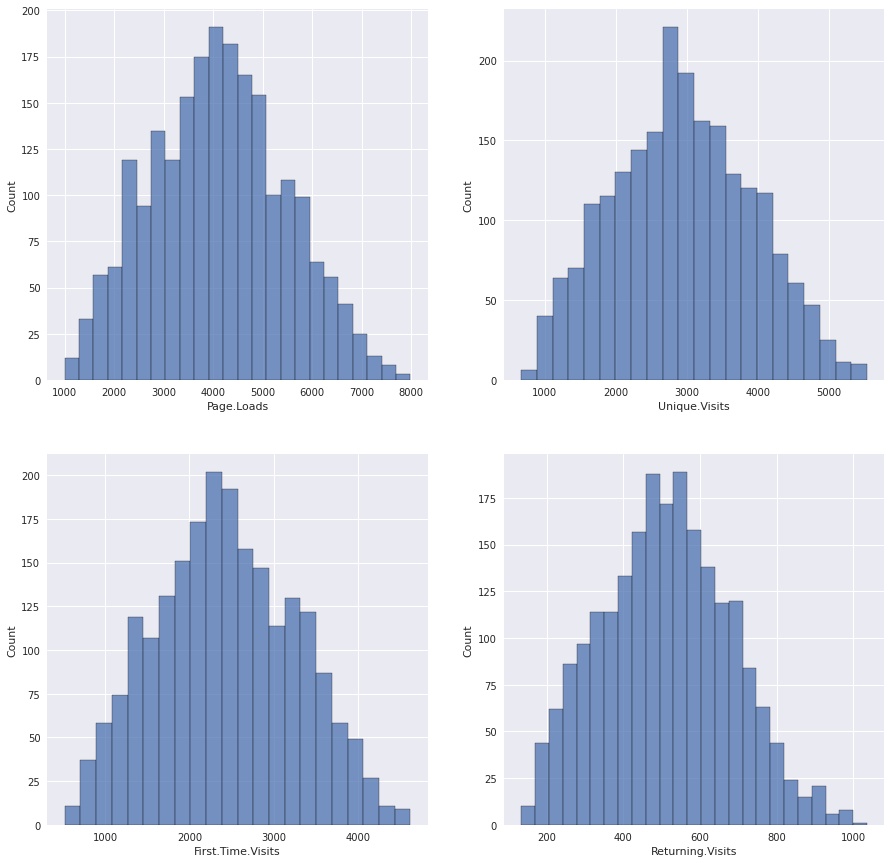
plt.xlabel('Date')

plt.ylabel('Count')

plt.legend()

plt.title('Time Series Analysis')

plt.show()



**Embedding Python Outputs:**

You can save the Python analysis output (e.g., charts, plots, or statistical results) as image files or data files.

In IBM Cognos, you can embed these files directly into your dashboards or reports by using the Image or Data components.

**Advanced Analysis:**

**Depending on your goals, you can use Python for advanced analytics:**

User Segmentation:

Utilize machine learning libraries like scikit-learn to segment users based on behavior or demographics.

Predictive Modeling:

Create predictive models using libraries like scikit-learn or TensorFlow for forecasting future traffic or user behavior.

A/B Testing: Implement statistical tests to analyze the effectiveness of changes made to your website.

**Real-time Data Updates:**

For real-time analytics, consider setting up a pipeline that regularly updates your Cognos dashboard with the latest data. Tools like Apache Kafka or AWS Kinesis can be used for real-time data ingestion.

**Program**

figure, ax = plt.subplots(2, 2, figsize=(17, 15))

plt.style.use('seaborn')

ax1 = ax[0]

ax2 = ax[1]

# Plot the Number of Page Loads with time

ax1[0].plot(data['Date'], data['Page.Loads'])

ax1[0].set\_xlabel("Date")

ax1[0].set\_ylabel("Number of Page Loads")

# Plot the Number of Unique Visits with time

ax1[1].plot(data['Date'], data['Unique.Visits'])

ax1[1].set\_xlabel("Date")

ax1[1].set\_ylabel("Number of Unique Visits")

# Plot the Number of First Time visits with time

ax2[0].plot(data['Date'], data['First.Time.Visits'])

ax2[0].set\_xlabel("Date")

ax2[0].set\_ylabel("Number of First Time visits")

# Plot the Number of Returning visits with time

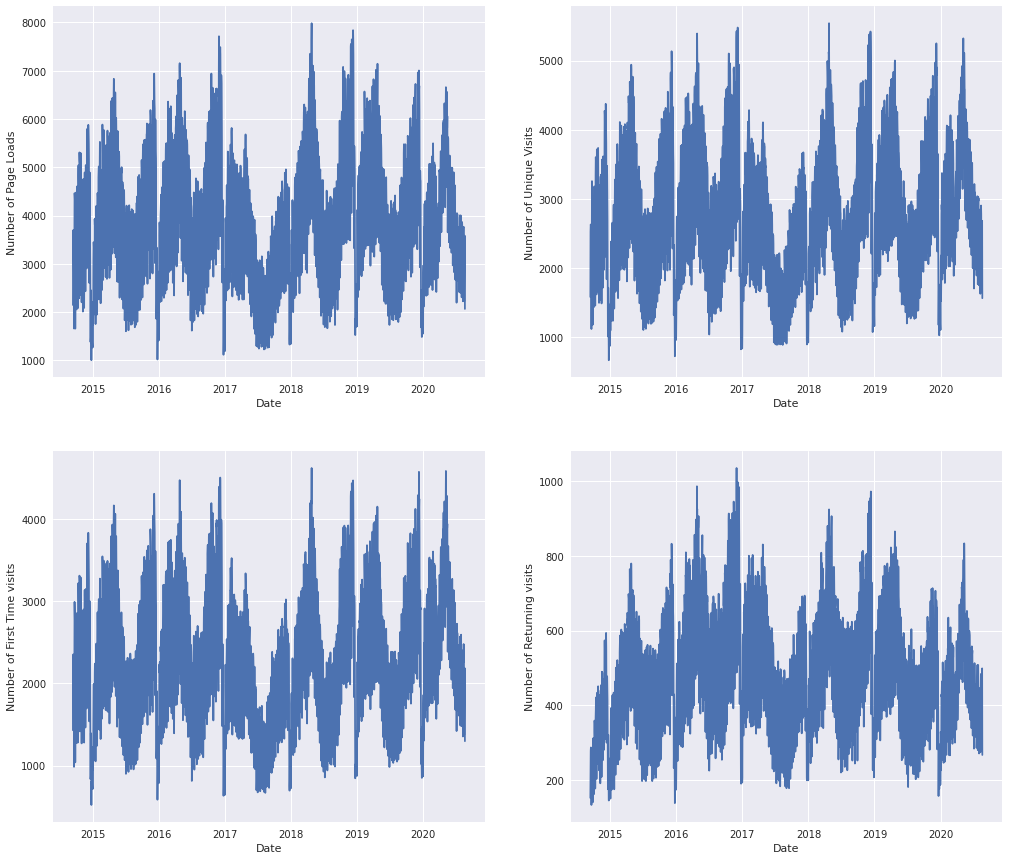
ax2[1].plot(data['Date'], data['Returning.Visits'])

ax2[1].set\_xlabel("Date")

ax2[1].set\_ylabel("Number of Returning visits")

figure.show()

**Output:**



**Program:**

def day\_wise\_EDA(day):

sun\_data = day\_grouped\_data.get\_group(day)

figure, ax = plt.subplots(2, 2, figsize=(17, 15))

plt.style.use('seaborn')

ax1 = ax[0]

ax2 = ax[1]

# Plot the Number of Page Loads with time

print("=================================================================={} ANALYSIS======================================================".format(day.upper()))

ax1[0].plot(sun\_data['Date'], sun\_data['Page.Loads'])

ax1[0].set\_xlabel("Date")

ax1[0].set\_ylabel("Number of Page Loads")

# Plot the Number of Unique Visits with time

ax1[1].plot(sun\_data['Date'], sun\_data['Unique.Visits'])

ax1[1].set\_xlabel("Date")

ax1[1].set\_ylabel("Number of Unique Visits")

# Plot the Number of First Time visits with time

ax2[0].plot(sun\_data['Date'], sun\_data['First.Time.Visits'])

ax2[0].set\_xlabel("Date")

ax2[0].set\_ylabel("Number of First Time visits")

# Plot the Number of Returning visits with time

ax2[1].plot(sun\_data['Date'], sun\_data['Returning.Visits'])

ax2[1].set\_xlabel("Date")

ax2[1].set\_ylabel("Number of Returning visits")

figure.show()

# Call the above function for every day

# 1. Sunday

day\_wise\_EDA('Sunday')

**Output:**

