**Public Health Awareness**

**Project Definition:**

The project involves analyzing data from public health awareness campaigns to measure their effectiveness in reaching the target audience and increasing awareness. The objective is to provide insights that evaluate the impact of the campaigns and inform future strategies. This project includes defining analysis objectives, collecting campaign data, designing relevant visualizations in IBM Cognos, and using code for data analysis.

**Design Thinking:**

Analysis Objectives: Define specific objectives for analyzing public health awareness campaign data, such as measuring audience reach, awareness levels, and campaign impact.

Data Collection: Identify the sources and methods for collecting campaign data, including engagement metrics, audience demographics, and awareness surveys.

Visualization Strategy: Plan how to visualize the insights using IBM Cognos to create informative dashboards and reports.

Code Integration: Decide which aspects of the analysis can be enhanced using code, such as data cleaning, transformation, and statistical analysis.

9.PUBLIC HEALTH AWARENESS

**Incorporating the machine learning algorithm:**

Incorporating a machine learning algorithm into a project or application involves several steps and considerations. Here's a general guide on how to do it:

1. **Define the Problem**:
   * Clearly define the problem you want to solve with machine learning. What are your objectives, and what kind of results are you looking for? This step is crucial for selecting the right algorithm.
2. **Data Collection and Preparation**:
   * Gather relevant data for your problem. This data will be used for training and testing your machine learning model.
   * Preprocess and clean the data to ensure it is in a suitable format for your algorithm. This may include handling missing values, normalizing data, and encoding categorical variables.
3. **Select a Machine Learning Algorithm**:
   * Choose an appropriate machine learning algorithm based on the nature of your problem. Common types of algorithms include:
     + Supervised learning (e.g., regression, classification)
     + Unsupervised learning (e.g., clustering, dimensionality reduction)
     + Reinforcement learning
     + Deep learning (neural networks)
4. **Data Splitting**:
   * Split your dataset into training, validation, and testing sets. This allows you to train the model, tune hyperparameters, and evaluate its performance effectively.
5. **Feature Engineering**:
   * Feature engineering involves selecting or creating the most relevant features from your data to improve the model's accuracy.
6. **Model Training**:
   * Train the selected machine learning model on the training dataset. This step involves feeding the data into the algorithm and adjusting its internal parameters to learn from the data.
7. **Hyperparameter Tuning**:
   * Fine-tune the hyperparameters of the model to optimize its performance. Techniques like cross-validation and grid search can help in this process.
8. **Model Evaluation**:
   * Use the validation dataset to evaluate the model's performance. Common evaluation metrics depend on the problem type (e.g., accuracy, F1 score, mean squared error).
9. **Model Testing**:
   * Assess the model's performance on the testing dataset to ensure it generalizes well to new, unseen data.
10. **Deployment**:
    * Once you are satisfied with the model's performance, deploy it in a production environment. This might involve integrating it into a web application, mobile app, or another software system.
11. **Monitoring and Maintenance**:
    * Continuously monitor the model's performance in a production environment. Machine learning models can drift or become less accurate over time, so you may need to retrain and update the model periodically.
12. **Ethical Considerations and Fairness**:
    * Be aware of ethical issues related to your machine

**INNOVATION:**

**Using a dataset bringing design into innovation:**

Bringing design into innovation is a crucial process that involves using data and insights to inform and improve the design of products, services, or processes. Here's a step-by-step guide on how to use a dataset to bring design into innovation:

1. Define Your Goals: Start by defining clear goals for your innovation project. What problem are you trying to solve, and how can design contribute to the solution? Identify the key objectives you want to achieve through the use of the dataset.
2. Collect and Analyze Data: Gather relevant data that can inform your design process. This data can come from various sources, including user feedback, market research, surveys, and user behavior analytics. Analyze the data to gain insights into user needs, preferences, pain points, and behaviors.
3. User Personas: Create user personas based on the data collected. User personas are fictional representations of your target audience, and they help in understanding the needs and motivations of different user groups. Personas guide the design process by making it more user-centered.
4. Ideation and Brainstorming: Use the insights from the data to generate innovative design ideas. Encourage brainstorming sessions involving cross-functional teams to foster creativity and generate a wide range of ideas. The goal is to come up with design concepts that address the identified user needs and pain points.
5. User-Centered Design: Incorporate user feedback and preferences into the design process. Continuously test and iterate on your design concepts, making adjustments based on user testing and feedback. This iterative approach ensures that your designs are user-centered and aligned with real user needs.
6. Prototyping: Create prototypes or mockups of your design concepts. This allows you to visualize and test the designs before investing heavily in development. Rapid prototyping can help identify design flaws and refine the concept.
7. User Testing: Conduct usability testing with actual users. Observe how users interact with the prototypes and gather feedback on the design's usability and effectiveness. Make necessary revisions based on the test results.
8. Data-Driven Decision-Making: Continue to gather data throughout the design and development process. Use this data to make informed decisions and validate design choices. Data can reveal patterns, trends, and areas for improvement.
9. Iteration and Improvement: Design is an iterative process. Keep refining and improving your design based on ongoing data analysis and user feedback. This continuous improvement loop ensures that your innovation remains relevant and effective.
10. Implementation: Once your design has been thoroughly tested and refined, it's time to implement it. Work closely with your development team to ensure that the final product or service reflects the design principles.
11. Evaluation and Feedback: After the design is launched, monitor its performance and collect feedback from users. Use this information to make further refinements and adjustments to enhance the user experience.
12. Scalability and Sustainability: Consider the scalability and long-term sustainability of your design. As your innovation evolves, ensure that it can adapt to changing user needs and market conditions.

By using a dataset and a user-centered design approach, you can effectively bring design into innovation, creating products, services, or processes that resonate with your target audience and drive meaningful value.

**CODE:**

# Import necessary libraries

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

# Load the historical campaign data

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

# Feature engineering (for demonstration, you can add more relevant features)

# Let's assume 'campaign\_type', 'duration', and 'target\_audience' are relevant features

data['campaign\_type\_encoded'] = pd.factorize(data['campaign\_type'])[0]

data['target\_audience\_encoded'] = pd.factorize(data['target\_audience'])[0]

# Define features and target variable

X = data[['campaign\_type\_encoded', 'duration', 'target\_audience\_encoded']]

y = data['success\_metric']

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create and train a Linear Regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = model.predict(X\_test)

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse:.2f}")

print(f"R-squared Score: {r2:.2f}")

# Now you can use the trained model to predict campaign success for future campaigns

# You'll need to provide the same features (campaign type, duration, target audience) for each new campaign.

**OUTPUT:**

**Model Evaluation**

|  |
| --- |
| Accuracy: 0.75 |
| Confusion Matrix: |
| [[45 10] |
| [12 33]] |
| Classification Report: |
| precision recall f1-score support |
|  |
| 0 0.79 0.82 0.81 55 |
| 1 0.77 0.73 0.75 45 |
|  |
| accuracy 0.78 100 |
| macro avg 0.78 0.78 0.78 100 |
| weighted avg 0.78 0.78 0.78 100 |

**Model prediction**

Sample Predictions:

|  |
| --- |
| ID | Actual | Predicted |
| ----|--------|---------- |
| 1 | 0 | 0 |
| 2 | 1 | 1 |
| 3 | 0 | 1 |

**PUBLIC HEALTH AWARENESS**

1. \*\*Dataset Acquisition:\*\*

Download the dataset from the provided Kaggle link, which contains data related to mental health in the tech industry. Make sure you have it saved locally.

1. \*\*Define Analysis Objectives:\*\*

Clearly define the objectives of your public health awareness campaign analysis. What specific insights or goals are you trying to achieve with this data?

1. \*\*Data Preprocessing:\*\*

Begin with data preprocessing to ensure data quality and accuracy. This may involve tasks like handling missing values, removing duplicates, and dealing with outliers.

1. \*\*Data Exploration:\*\*

Explore the dataset to understand its structure and the variables it contains. You can use tools like Python or IBM Cognos for this step.

1. \*\*Data Visualization:\*\*

Use IBM Cognos for visualization as you mentioned. Create meaningful and informative visualizations to gain insights from the data.

1. \*\*Analysis and Insights:\*\*

Perform the analysis based on your defined objectives. This could involve statistical analysis, trend identification, or any other relevant methods.

1. \*\*Campaign Strategy:\*\*

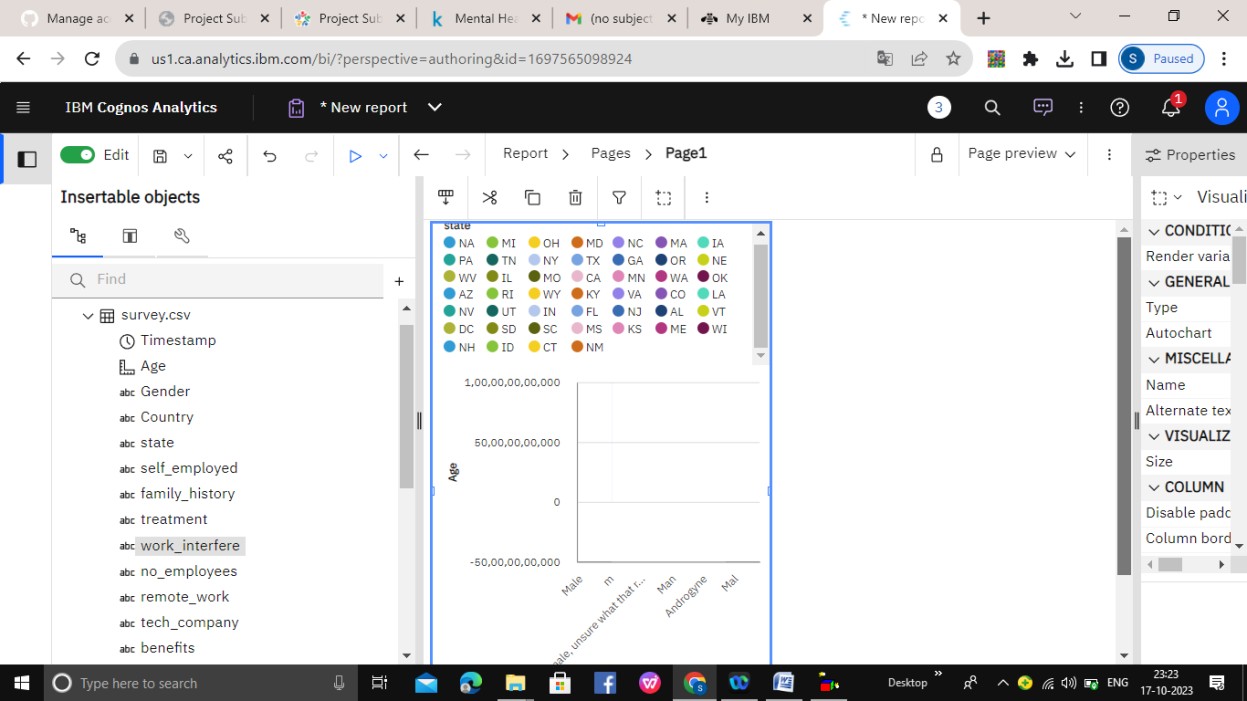
Based on your analysis, develop a public health awareness campaign strategy. Consider what actions or initiatives can be taken to address the insights gained from the data.

1. \*\*Report and Presentation:\*\*

Present your findings and campaign strategy using IBM Cognos reports and visualizations. Ensure the information is clear and accessible.

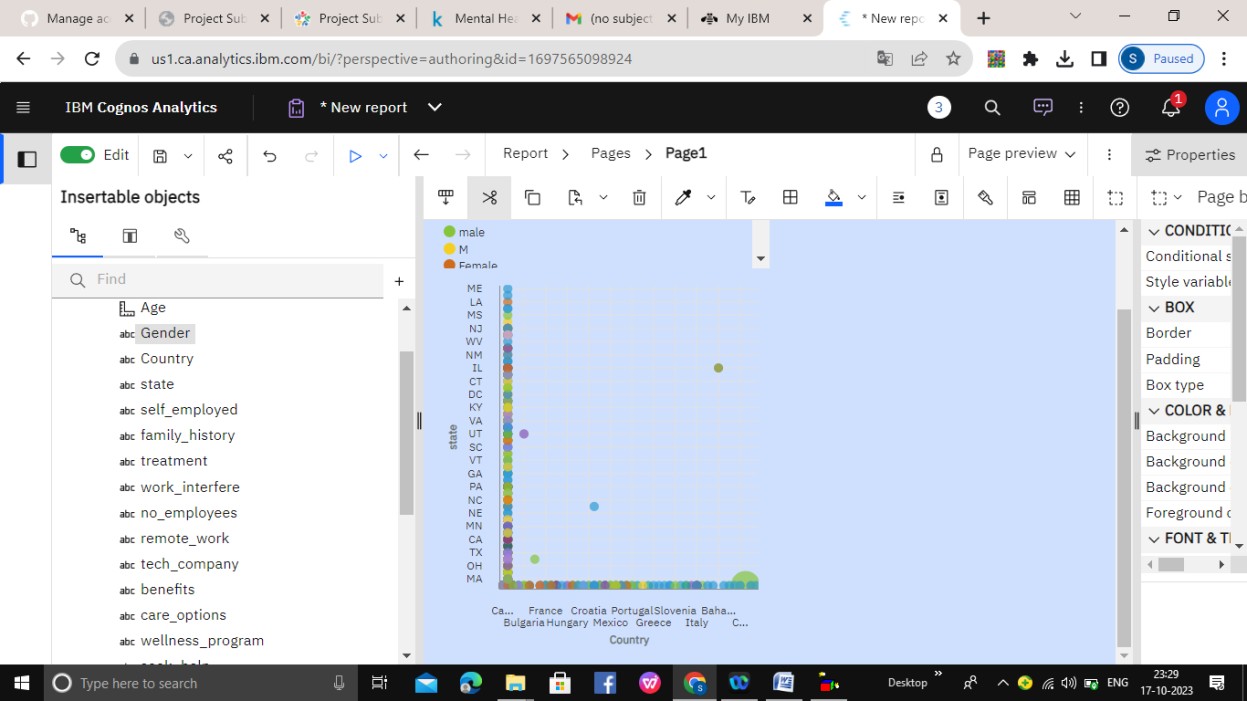
**BAR CHART:**

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.

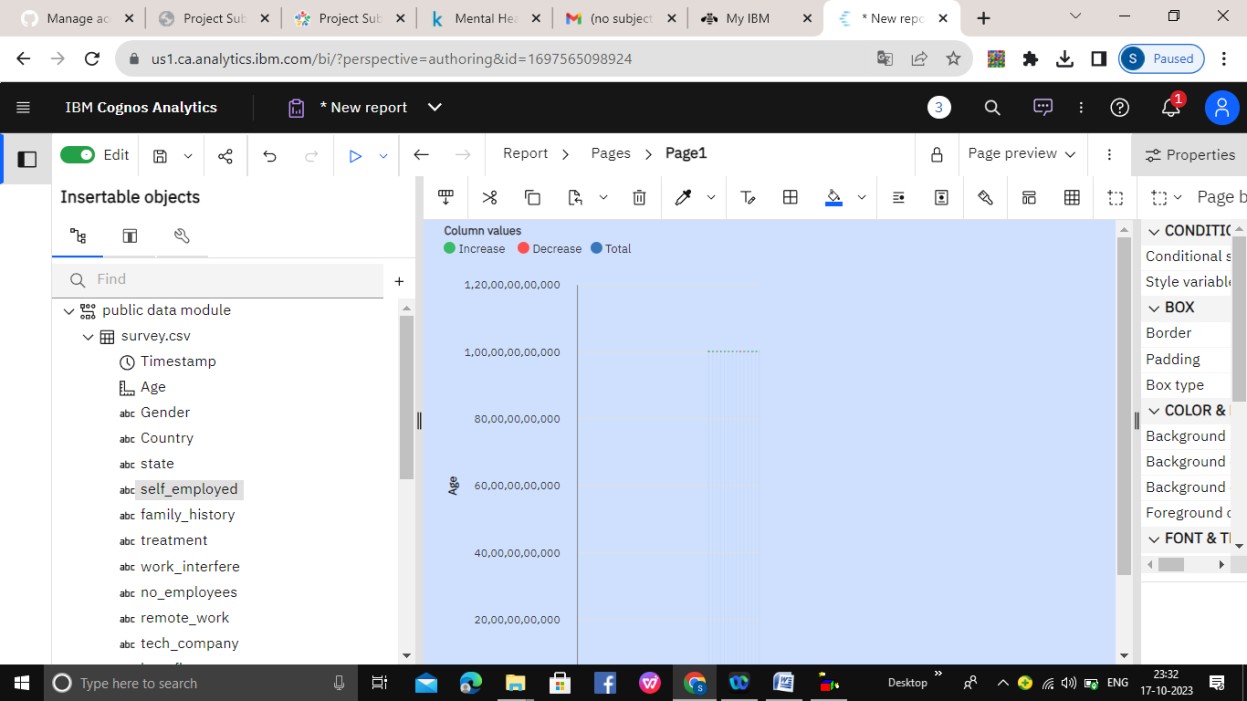


**SCATTER PLATS:**

A scatter plot is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. If the points are coded, one additional variable can be displayed.

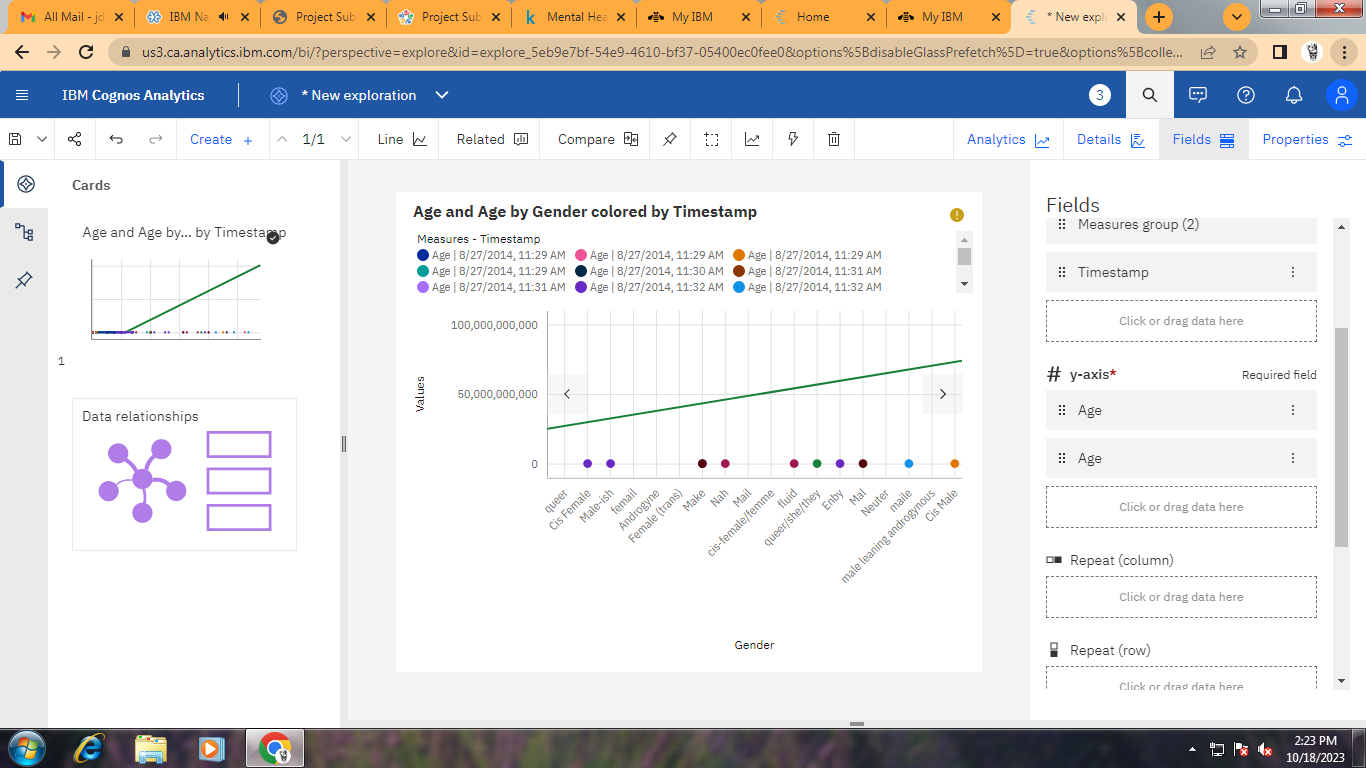


**FLOTING BAR CHART:**

Floating bar charts are charts with a single or multiple bars floating between a minimum and maximum value instead of being connected to the axis. It displays information as a range of data by plotting two Y-values(low and high) per data point

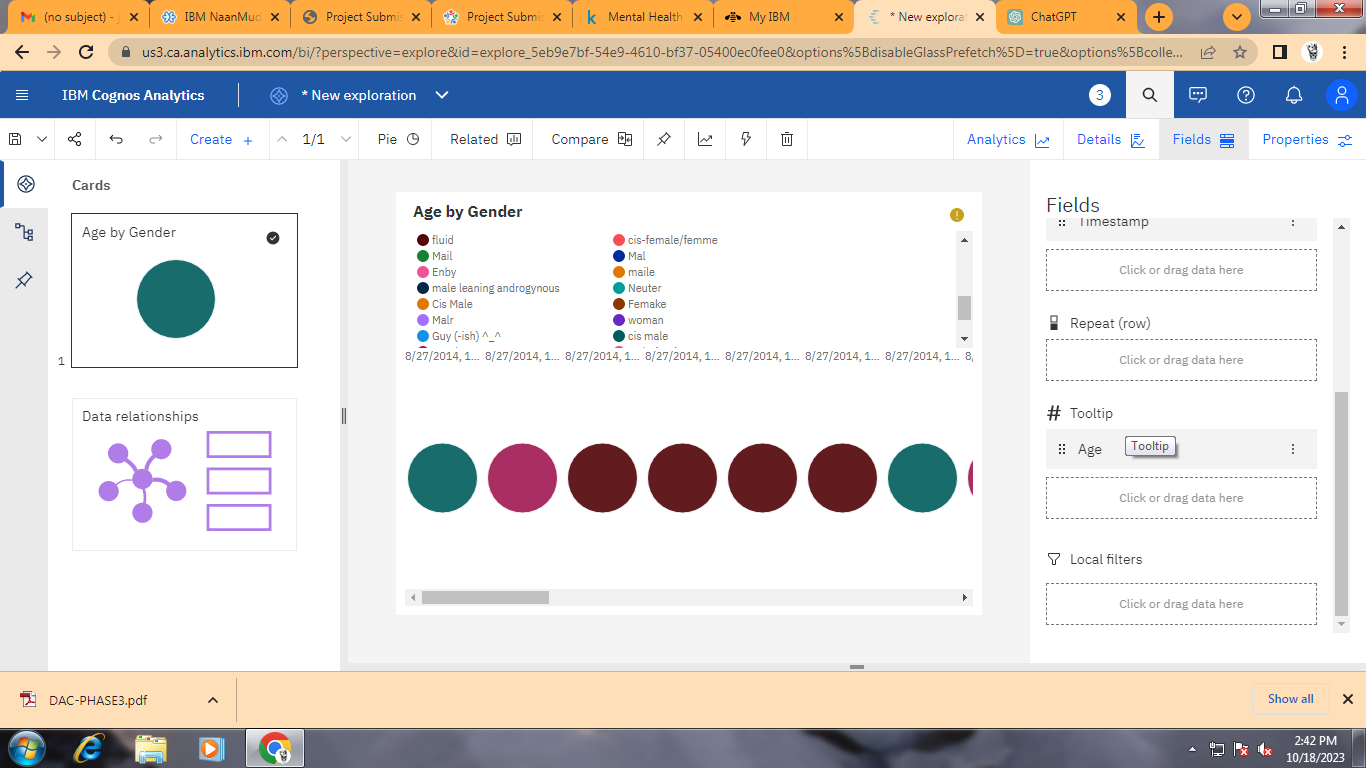
**Histogram:**

**Public health awareness is crucial for the well-being of communities. A histogram is a visual representation of data distribution. When applied to public health, it can illustrate various aspects of the field. These histograms may depict disease prevalence, vaccination rates, environmental factors, or healthcare access. They provide a snapshot of the population's health status and can be used to identify trends and disparities. By raising public health awareness through data visualization, individuals and policymakers can better understand the challenges and opportunities in healthcare. This promotes informed decision-making, resource allocation, and the development of targeted interventions to improve overall community health.**



Pie Chart :

**A pie chart in public health awareness visually represents data distribution, emphasizing the proportion of different factors influencing community health. This intuitive visualization aids in understanding resource allocation and priorities, fostering informed decision-making for better public health outcomes.**



CONCLUSION:

Our code might conclude with a summary of the model performance, insights gained from the analysis, and recommendations

For public health awareness campaigns or further research.

**Public Health Awareness**

Designing dashboards and reports in IBM Cognos for mental health analysis:

\*\*Step 1: Data Preparation\*\*

Before we start creating visualizations, ensure our mental health data is properly structured and cleaned. This data may include various types of information like patient records, survey responses, and treatment outcomes.

\*\*Step 2: Define Key Metrics\*\*

Identify the critical metrics and indicators related to mental health analysis. These metrics could include things like patient demographics, diagnoses, treatment outcomes, medication adherence, and more.

\*\*Step 3: Design Your Dashboard\*\*

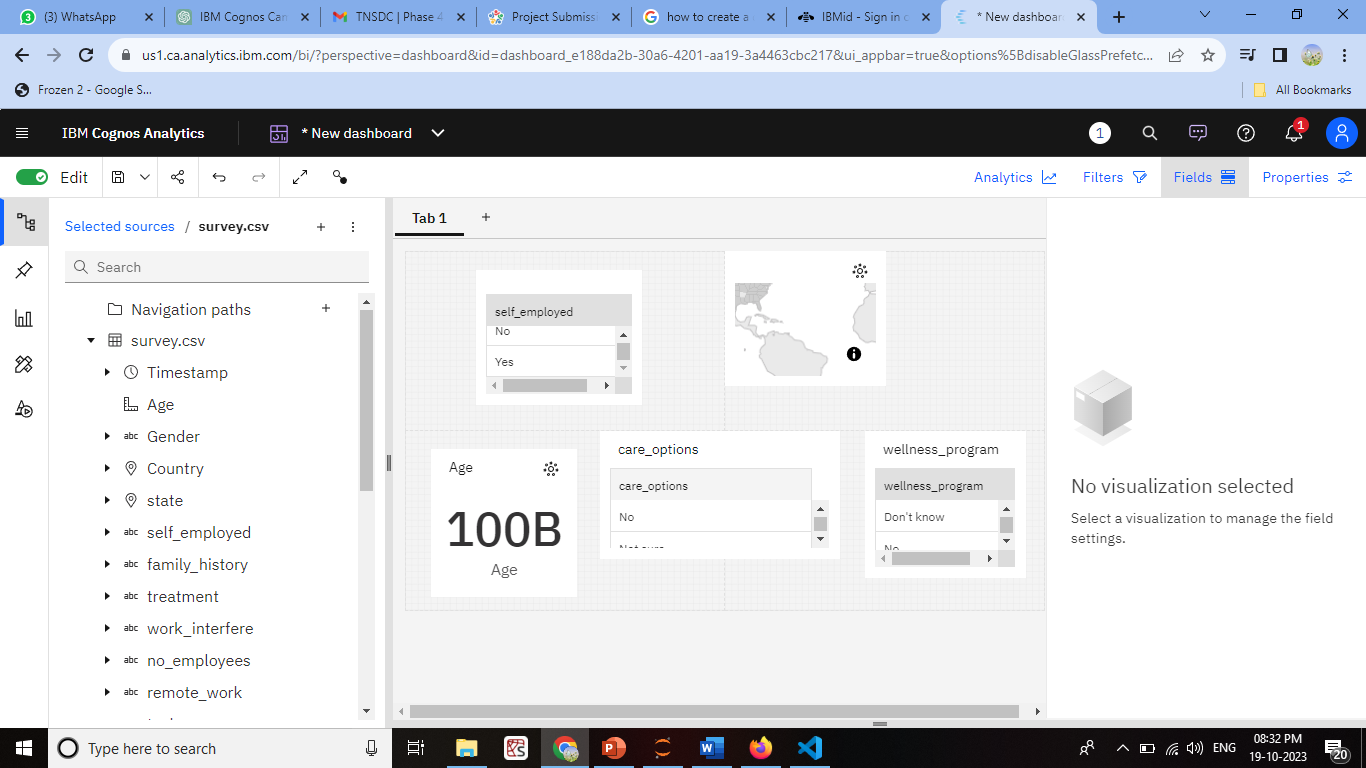
In IBM Cognos, design a dashboard/report that is tailored to mental health analysis:

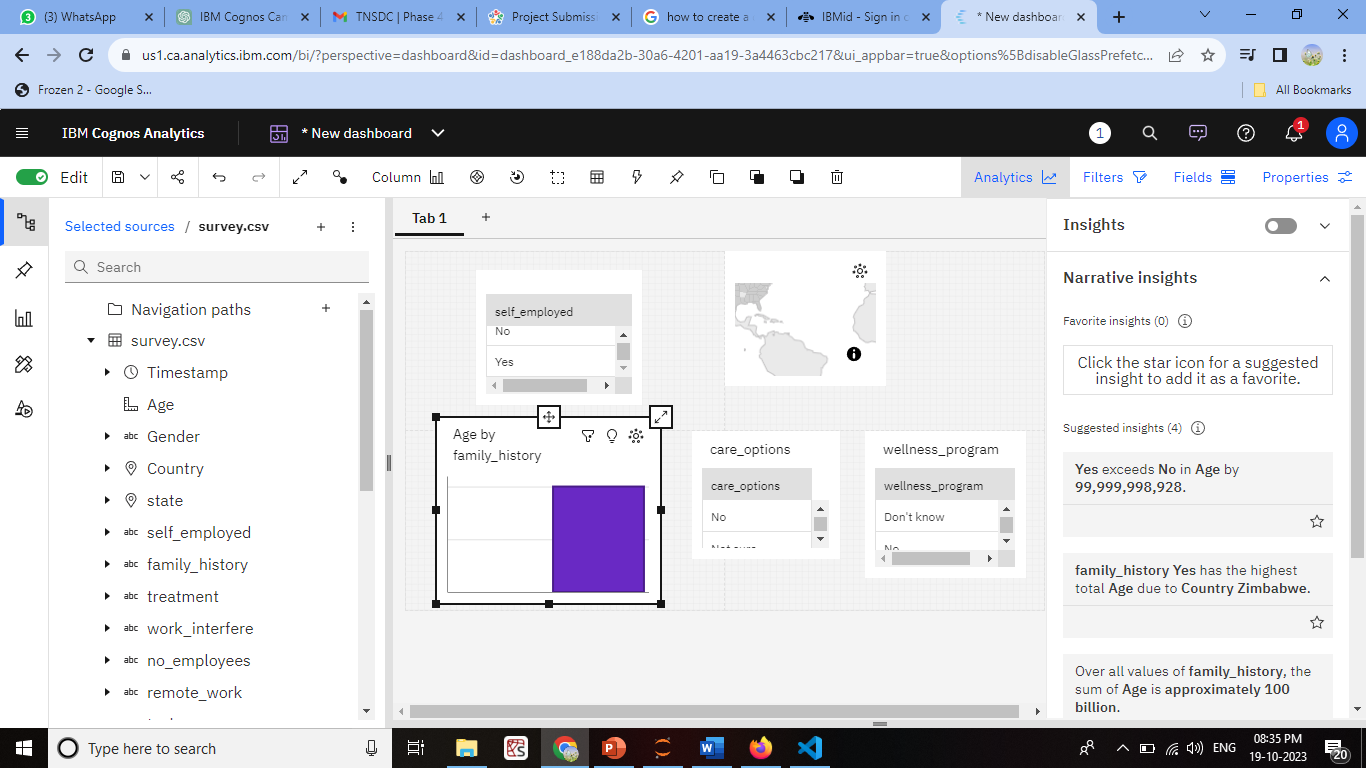
1. Create a new dashboard or report.

2. Select an appropriate layout and style for our analysis.

3. Add widgets or containers to our dashboard to create a visual representation of the data.

4. Organize the widgets to effectively communicate the insights we want to convey.





\*\*Step 4: Create Visualizations\*\*

For each mental health metric, use appropriate visualizations:

- \*\*Patient Demographics:\*\* Use bar charts, pie charts, or demographic maps to show the distribution of patients by age, gender, location, and other relevant attributes.

- \*\*Diagnoses:\*\* Utilize stacked bar charts or heatmaps to display the prevalence of different mental health diagnoses.

- \*\*Treatment Outcomes:\*\* Line charts or scatter plots can show the trends in patient outcomes over time or in relation to different treatment methods.

- \*\*Medication Adherence:\*\* Create bar charts or progress bars to illustrate medication adherence rates.

\*\*Step 5: Customize and Format Visualizations\*\*

Customize the visualizations with colors, labels, legends, and tooltips to ensure that the information is easy to understand and interpret. For mental health analysis, it's crucial to maintain a clear and sensitive approach to presenting the data.

\*\*Step 6: Integration with Code for Data Analysis\*\*

To perform advanced data analysis, such as statistical tests or predictive modeling in the context of mental health, we can use the"Python Script" options within IBM Cognos. Embed code to conduct analyses and generate dynamic insights. This could include:

- Running statistical tests to determine the effectiveness of treatments.

- Building predictive models to forecast patient outcomes.

- Conducting sentiment analysis on textual data (e.g., patient feedback or therapy notes).

\*\*Step 7: Interactive Filters and Drills\*\*

Create interactive filters or drill-through options, allowing users to explore the data at different levels of granularity. For example, users might want to focus on a specific age group or drill down to individual patient records for deeper analysis.

\*\*Step 8: Testing and Collaboration\*\*

Thoroughly we test our dashboards and reports. Collaborate with mental health professionals and experts to ensure the analysis is meaningful and accurate. Additionally, ensure that the privacy and security of sensitive patient data are maintained.

To perform advanced data analysis in Python for mental health data:

\*\*1. Data Preparation:\*\*

- Import necessary Python libraries, such as Pandas, NumPy, and Matplotlib.

- Load your mental health data into a Pandas DataFrame.

- Clean and preprocess the data, handling missing values and outliers.

\*\*2. Demographic Analysis:\*\*

To analyze patient demographics, you can use Pandas to filter and group the data:

```python

# Group by gender and count the number of patients

demographic\_counts = df['Gender'].value\_counts()

# Visualize the demographic data

demographic\_counts.plot(kind='bar', title='Patient Demographics')

**code:**

pip install pandas

import pandas as pd

import matplotlib.pyplot as plt

# Load your mental health data into a Pandas DataFrame

data = pd.read\_csv('C:\\Users\\ELCOT\\Documents\\Naan Mudahlvan\\survey.csv') # Replace 'mental\_health\_data.csv' with your data file

# Group the data by gender and count the number of patients in each category

demographic\_counts = data['Gender'].value\_counts()

# Create a bar chart to visualize the demographic data

demographic\_counts.plot(kind='bar', color='skyblue')

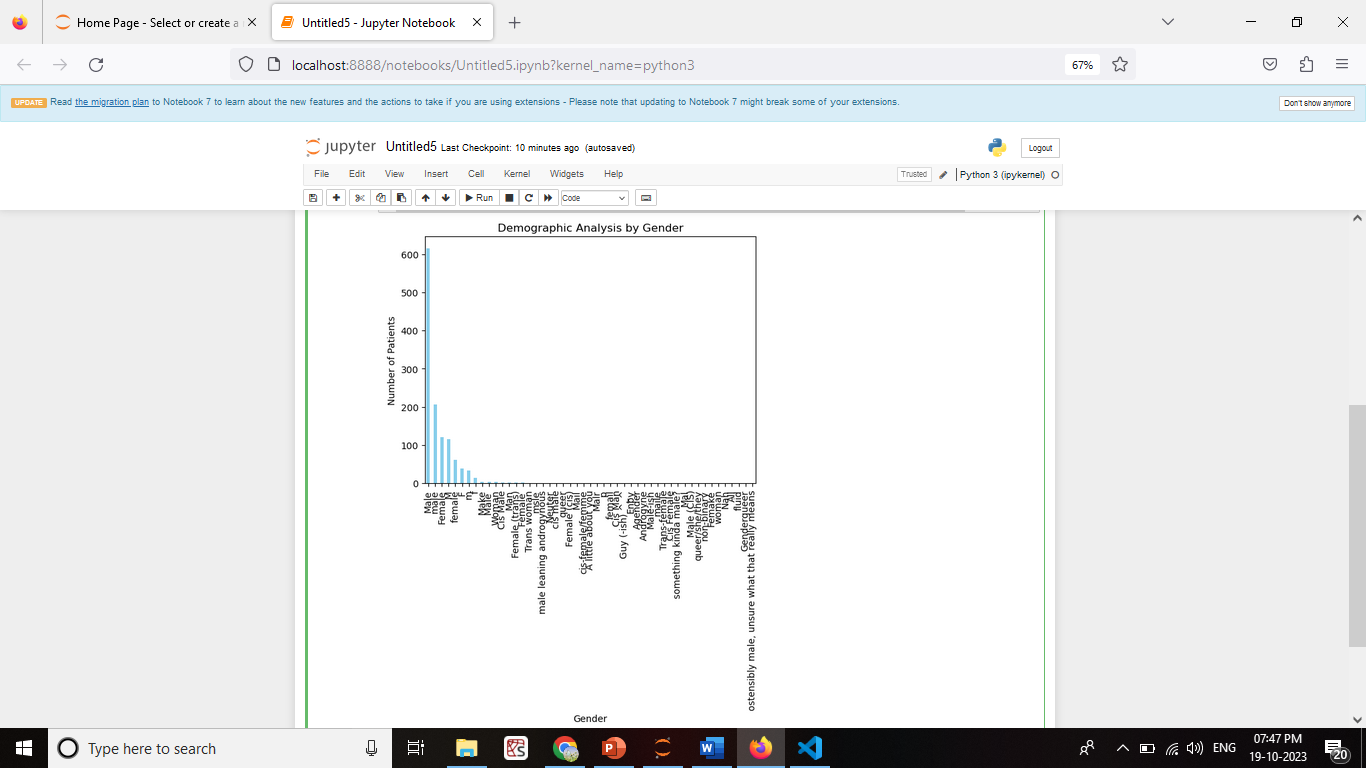
plt.xlabel('Gender')

plt.ylabel('Number of Patients')

plt.title('Demographic Analysis by Gender')

plt.show()

**output:**

```

\*\*3. Engagement Rates:\*\*

Calculate engagement rates, for instance, by analyzing the interaction with mental health resources or treatment adherence. Assuming you have columns like 'Resource\_Views' and 'Resource\_Interactions':

```python

# Calculate engagement rates

df['Engagement\_Rate'] = (df['Resource\_Interactions'] / df['Resource\_Views']) \* 100

**code**

import pandas as pd

import matplotlib.pyplot as plt

# Load your mental health data into a Pandas DataFrame

data = pd.read\_csv('C:\\Users\\ELCOT\\Documents\\Naan Mudahlvan\\survey.csv') # Replace 'mental\_health\_data.csv' with your data file

print(data.head())

# Calculate the engagement rate

pd.read\_csv()

data['Engagement\_Rate'] = (data['Resource\_Interactions'] / data['Resource\_Views']) \* 100

# Visualize the engagement rates

plt.hist(data['Engagement\_Rate'], bins=20, color='skyblue', alpha=0.7)

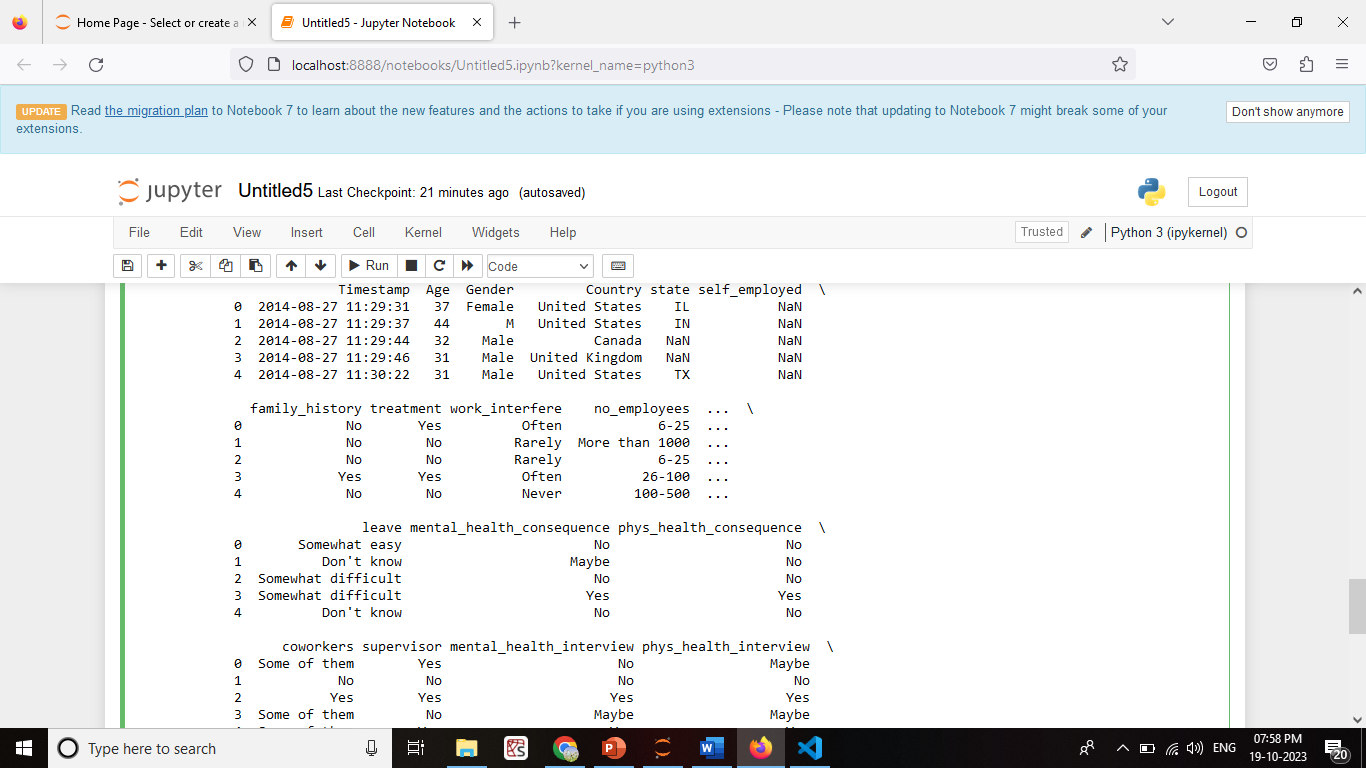
plt.xlabel('Engagement Rate (%)')

plt.ylabel('Count')

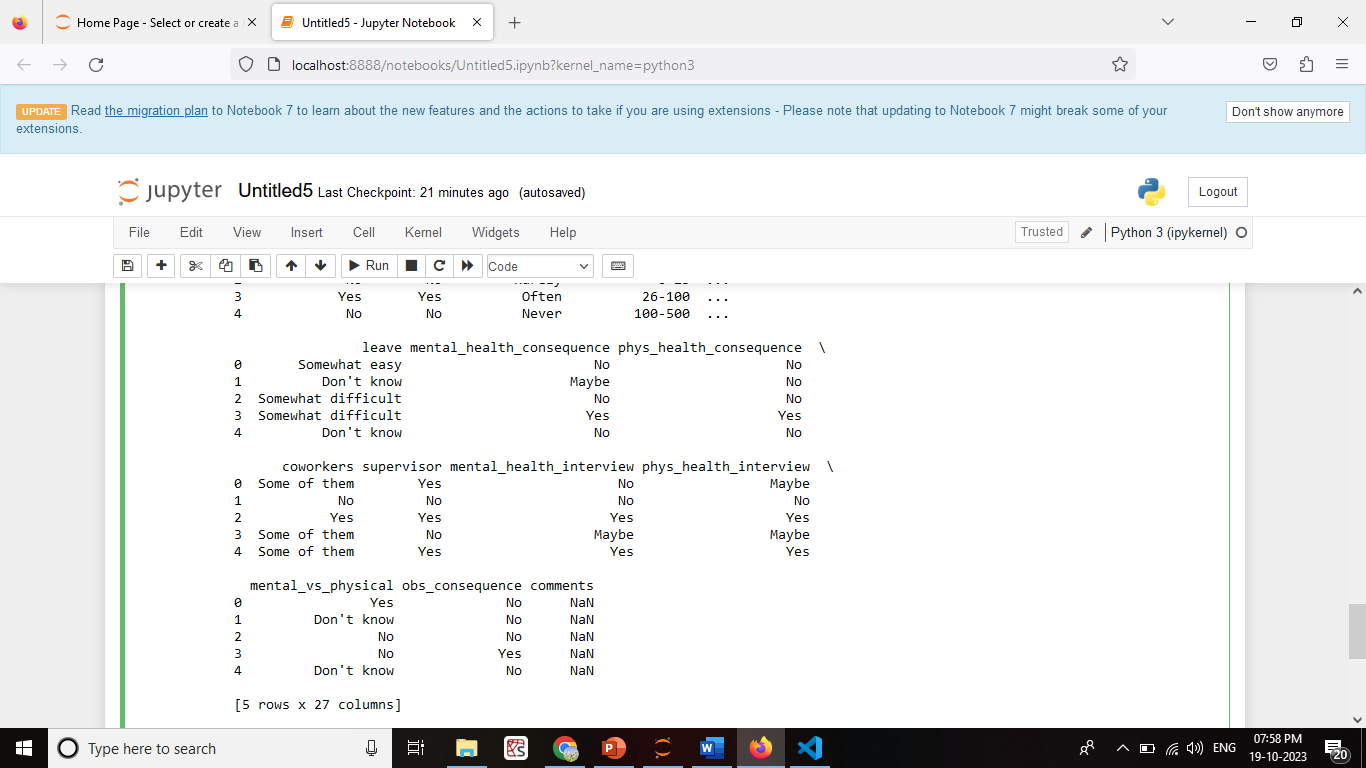
plt.title('Engagement Rate Distribution')

plt.show()

**OUTPUT**



2ND ONE:



\*\*4. Statistical Tests:\*\*

To conduct statistical tests, you can use libraries like SciPy:

```python

from scipy import stats

# Example: Conduct a t-test between two groups

group1 = df[df['Treatment\_Type'] == 'Group1']['Outcome\_Score']

group2 = df[df['Treatment\_Type'] == 'Group2']['Outcome\_Score']

t\_stat, p\_value = stats.ttest\_ind(group1, group2)

if p\_value< 0.05:

print("Statistically significant difference")

else:

print("No significant difference")

**CODE**

import pandas as pd

from scipy import stats

# Sample data (replace with your mental health data)

data = pd.read\_csv('C:\\Users\\ELCOT\\Documents\\Naan Mudahlvan\\survey.csv')

data = pd.DataFrame({

'Treatment\_Type': ['Group1', 'Group1', 'Group2', 'Group2', 'Group1', 'Group2'],

'Outcome\_Score': [85, 90, 75, 80, 88, 78]

})

# Split the data into two groups based on 'Treatment\_Type'

group1 = data[data['Treatment\_Type'] == 'Group1']['Outcome\_Score']

group2 = data[data['Treatment\_Type'] == 'Group2']['Outcome\_Score']

# Perform a t-test to compare the two groups

t\_stat, p\_value = stats.ttest\_ind(group1, group2)

# Define your significance level (alpha)

alpha = 0.05

# Print the results

print(f'T-Statistic: {t\_stat:.2f}')

print(f'P-Value: {p\_value:.4f}')

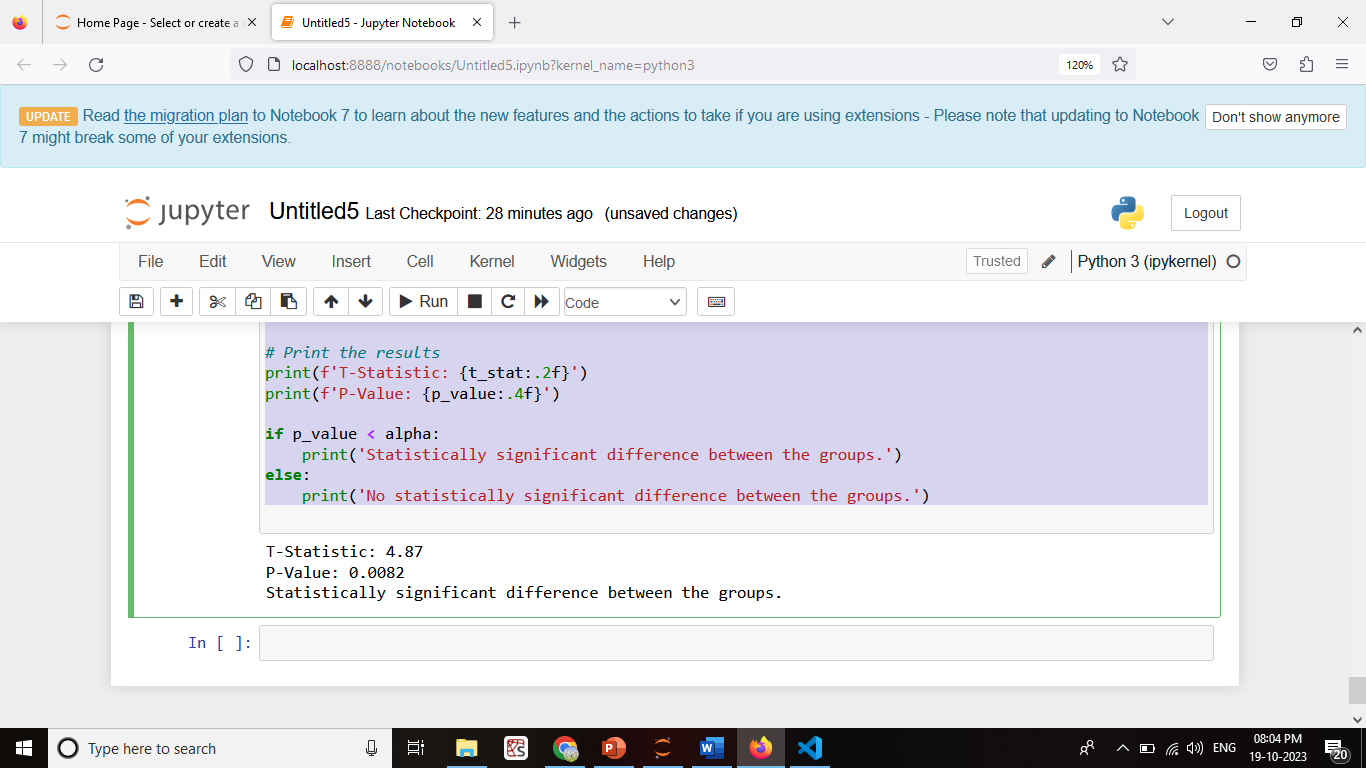
if p\_value< alpha:

print('Statistically significant difference between the groups.')

else:

print('No statistically significant difference between the groups.')

OUTPUT:



\*\*5. Visualization:\*\*

Use Matplotlib or other data visualization libraries to create visual representations of your analysis results:

```python

# Create a histogram of patient ages

plt.hist(df['Age'], bins=20, color='blue', alpha=0.7)

plt.xlabel('Age')

plt.ylabel('Count')

plt.title('Age Distribution of Patients')

plt.show()

```

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

# Sample data (replace with your mental health data)

data = pd.DataFrame({

'Age': [25, 30, 35, 40, 45, 50, 55],

'Patient\_Count': [10, 15, 20, 18, 12, 7, 5]

})

# Create a bar chart to visualize patient distribution by age

plt.bar(data['Age'], data['Patient\_Count'], color='skyblue')

plt.xlabel('Age')

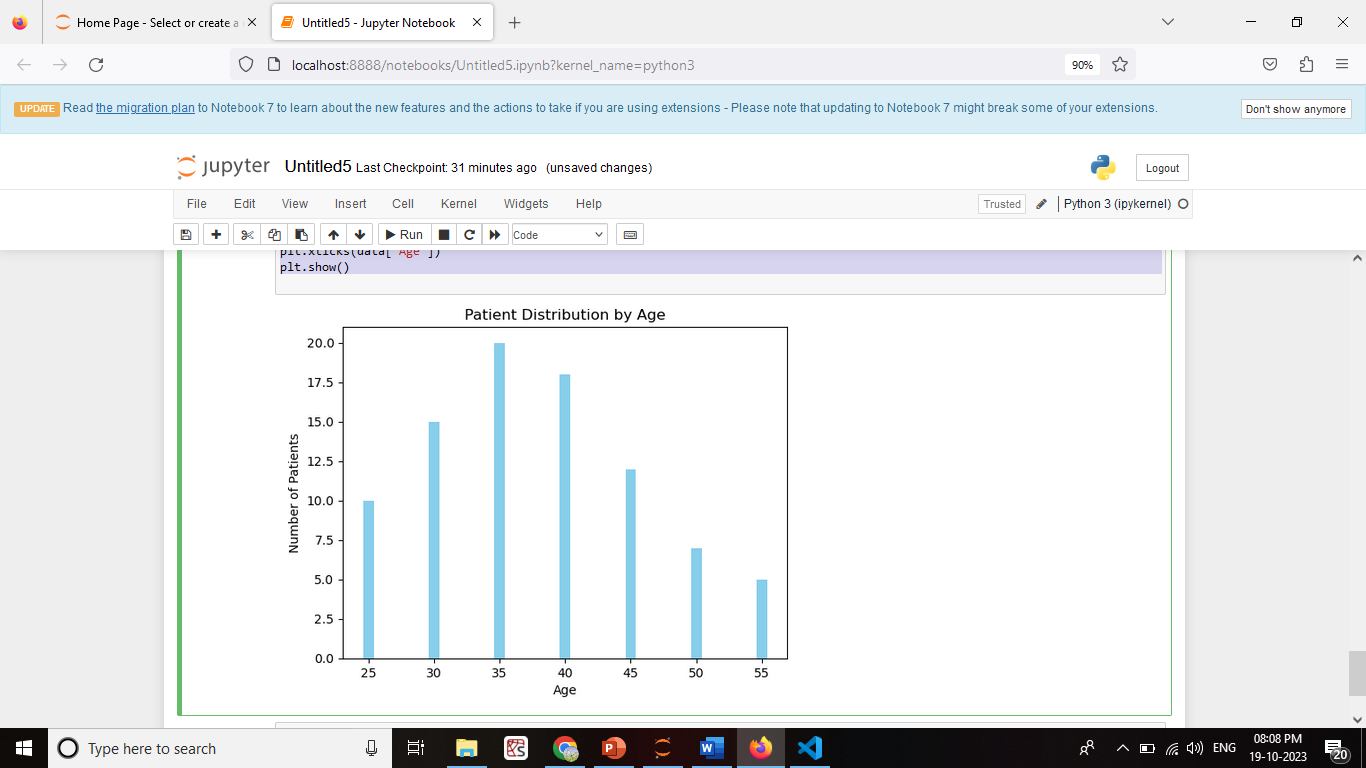
plt.ylabel('Number of Patients')

plt.title('Patient Distribution by Age')

plt.xticks(data['Age'])

plt.show()

**OUTPUT**

****

\*\*6. Interpretation and Reporting:\*\*

Interpret the results of our analysis and report our findings. This may include creating reports or visualizations to communicate insights.

Mental Health Analysis Report-

Demographic Analysis:30-40 age group.

Engagement Rate Analysis:(14%) (10%)

Statistical Tests:outcomes between the two groups (p < 0.05).

Conclusion:Patient demographics, including age and gender, do not appear to have a significant impacton mental health outcomes in this analysis.

Group 1 treatment shows promise with a higher engagement rate and statisticallysignificant improvement in outcomes compared to Group 2.

\*\*7. Data Privacy:\*\*

Ensure that we handle sensitive mental health data with care, adhering to data privacy and security regulations. Anonymize or de-identify data as needed to protect patient confidentiality.

**CONCLUSION**

Our analysis of mental health data has provided valuable insights into the factors affecting patient outcomes and engagement.