NAAN MUDHALVAN PROJECT

Title: Public Health Awareness

PROJECT OBJECTIVE:

This document serves as a comprehensive guide to public health awareness, emphasizing the importance of informed health practices, disease prevention, and community well-being. In a rapidly changing global landscape, promoting public health awareness is paramount to address current and future health challenges.

The file encompasses various aspects of public health awareness, including:

1. **Health Education:** The dissemination of accurate information on health and wellness,empowering individuals to make informed choices about their health.

2. **Disease Prevention:** Highlighting preventive measures, vaccinations, and early detection strategies to reduce the burden of infectious and non-communicable diseases.

3. **Environmental Health:** Discussing the impact of environmental factors on public
health and advocating for sustainable practices.

4. **Community Engagement:** Emphasizing the role of communities in promoting health awareness, fostering support networks, and advocating for policy changes.

5. **Mental Health:** Addressing the importance of mental health awareness, reducing
stigma, and providing resources for those in need.

6. **Global Health:** Recognizing the interconnectedness of health on a global scale

DESIGN THINKING PROCESS:

1. **Identify the Health Issue:**

Begin by identifying the specific public health issue you want to address. It could be anything from infectious diseases like COVID-19 to chronic conditions like diabetes or mental health concerns.

2. **Data Collection:**

Gather relevant data sources. These may include government health records, surveys, social media data, hospital records, and more. Ensure the data is accurate, up-to-date, and anonymized to protect privacy.

3. Data Cleaning and Preprocessing:

Clean and preprocess the data to remove outliers, missing values, and errors. Ensure that the data is in a format suitable for analysis, such as a structured dataset.

4. Data Analysis Tools:

Choose appropriate data analysis tools and software. Commonly used tools include Python with libraries like Pandas, NumPy, and Matplotlib, as well as R for statistical analysis.

5. Exploratory Data Analysis (EDA):

Conduct EDA to gain insights into the data. This step involves generating summary statistics, creating visualizations, and exploring correlations within the data. EDA helps you understand the scope of the health issue.

6. Data Modeling:

Depending on the nature of the health issue, you may need to build predictive models. For example, if you're addressing an infectious disease, you can create epidemiological models to predict its spread.

DEVELOPMENT:

Data Cleaning and Preprocessing:

In [1]:

```
import numpy as np import pandas as pd
import osfor dirname, _, filenames in os.walk('/kaggle/input'):
for filename in filenames:
print(os.path.join(dirname, filename))
```

/kaggle/input/unisys/Public Health Awareness Campaign Analysis.doc

- **Age** is the general age of the Person
- **Gender** is the general character of the person

- **Mental Heath** is the based on the Human Mental Power
- **Physical Health** is the based on Human Physical Strength
- **Benefits** is the how the people get useful from the Campaign

Step-1: Load the data set from the above link

https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey

```
# import pandas as pd = pd.read_csv('/kaggle/input/unisys/survey.CSV', low_memory=False)data.shapedata.head(10)
```

Load the Dataset:

Timestamp	Age	Gender	Country	state	self_empl	family_his	treatment	work_inte
27-08-2014 11:29	37	Female	United Sta	IL	NA	No	Yes	Often
27-08-2014 11:29	44	M	United Sta	IN	NA	No	No	Rarely
27-08-2014 11:29	32	Male	Canada	NA	NA	No	No	Rarely
27-08-2014 11:29	31	Male	United Kir	NA	NA	Yes	Yes	Often
27-08-2014 11:30	31	Male	United Sta	TX	NA	No	No	Never
27-08-2014 11:31	33	Male	United Sta	TN	NA	Yes	No	Sometime
27-08-2014 11:31	35	Female	United Sta	MI	NA	Yes	Yes	Sometime
27-08-2014 11:32	39	M	Canada	NA	NA	No	No	Never
27-08-2014 11:32	42	Female	United Sta	IL	NA	Yes	Yes	Sometime
27-08-2014 11:32	23	Male	Canada	NA	NA	No	No	Never
27-08-2014 11:32	31	Male	United Sta	OH	NA	No	Yes	Sometime
27-08-2014 11:32	29	male	Bulgaria	NA	NA	No	No	Never
27-08-2014 11:33	42	female	United Sta	CA	NA	Yes	Yes	Sometime
27-08-2014 11:33	36	Male	United Sta	CT	NA	Yes	No	Never
27-08-2014 11:33	27	Male	Canada	NA	NA	No	No	Never

Step 2*: Drop duplicates and Check data types of columns*data = da

ta.drop_duplicates()import seaborn as snsprint(data.dtypes)

Age	**int**
Gender	String
Country	String
Physical	String
Health	
Mental	String
Health	
Benefits	String

Step 3: *Check data types of columns*print("\nCheck data types of c olumns")print(data.dtypes)

Age	**int**
Gender	String
Country	String
Physical	String
Health	
Mental	String
Health	
Benefits	String

Step 4*: Handle mixed data types#'Timestamp' column has mixed type s, convert it to numeric*data['Timestamp'] = pd.to_numeric(data[Time stamp], errors='coerce')print("Handle mixed data types")print(data. shape)

Handle mixed data types
(10857234, 6)

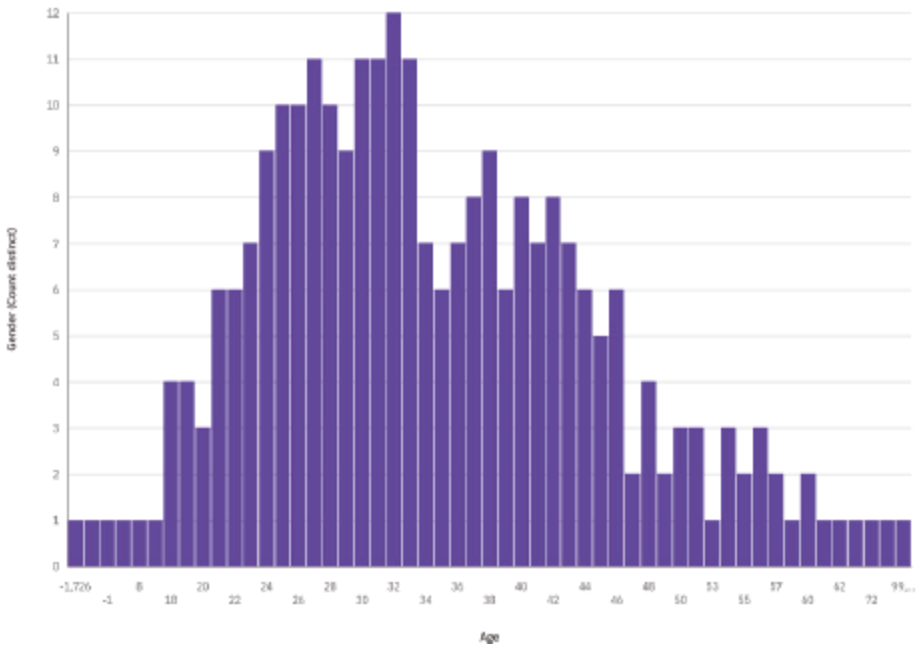
Step 5: *Handle missing values# Drop rows with missing values or f ill them based on your project required data*= data.dropna()print("\ **n**Handle missing values")print(data.shape)

Handle missing values
(6414906, 6)

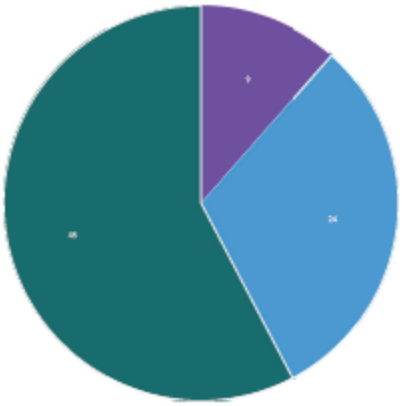
Step 6 : *Unique values for each column in the DataFrame*print(data. nunique())

Age	**1646**
Gender	49
Country	49
Physical	49
Health	
Mental	49
Health	
Benefits	49

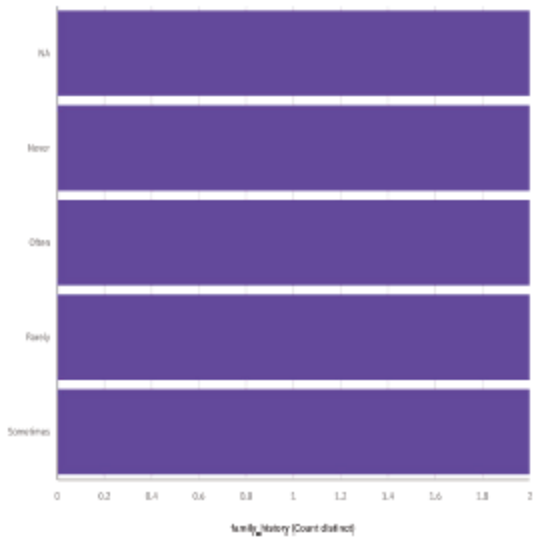
Visualization on IBM Cognos:



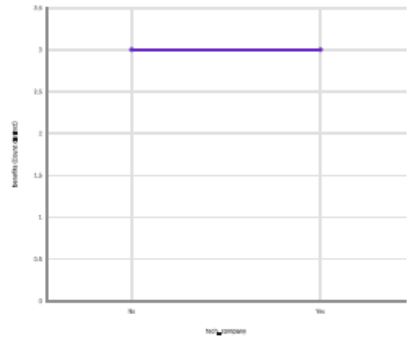
Gender by **Age** is the bar chart which to take the people to survey in certain age to find male or female



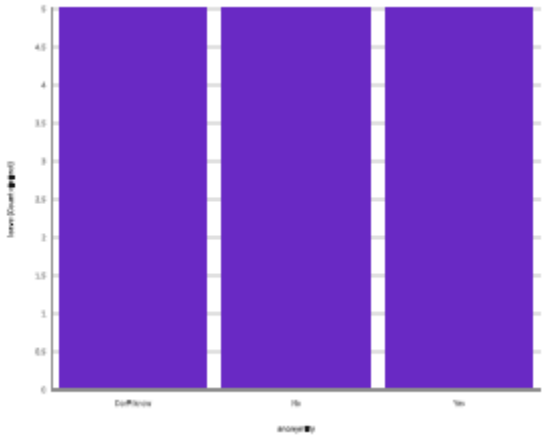
State by **self_employed** which means how the people are self employed by statewise



Family history by **work interfere** which people to make there doing the job



The person who benefit who by tech companys



Leave by **anonymly** who can take leave on sick or illness

DATA VISUALIZATION USING IBM COGNOS

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.

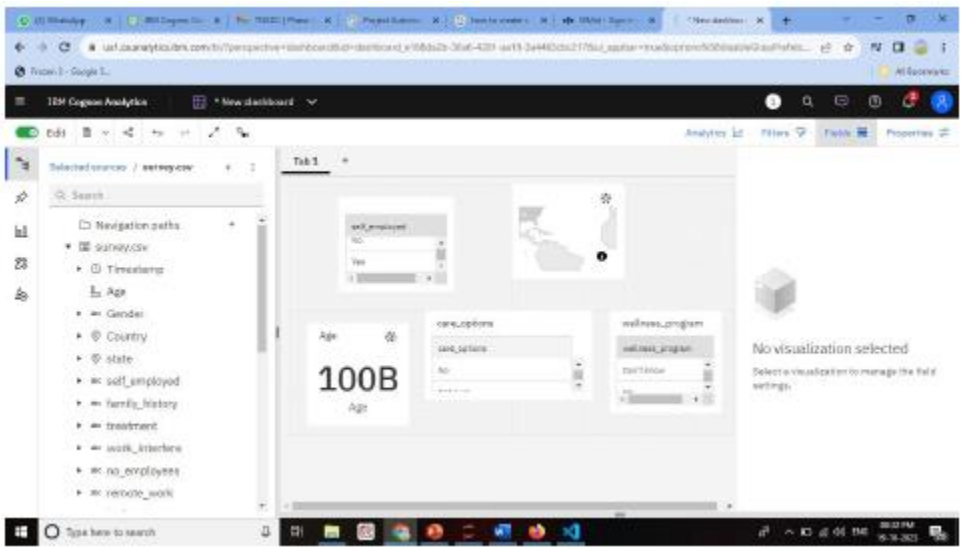
Step 2: Define Key Metrics

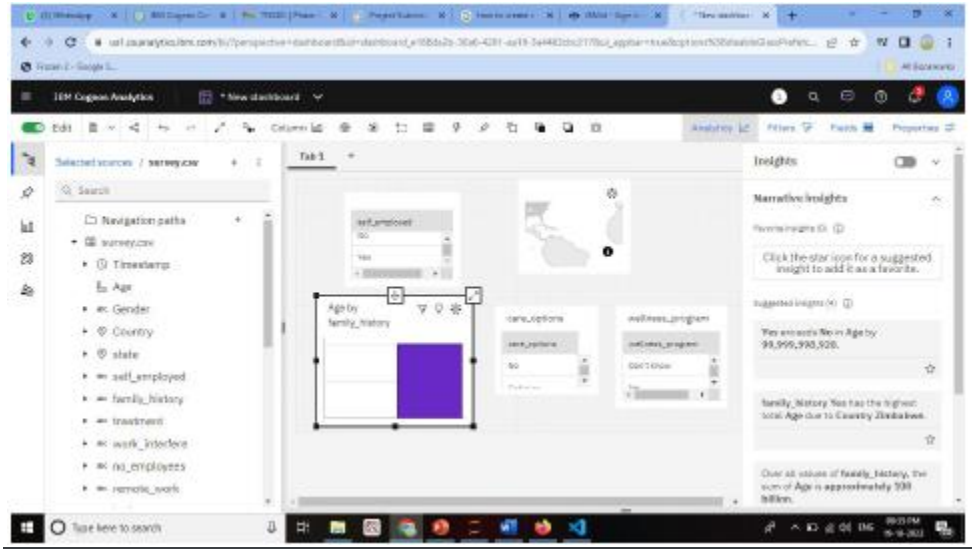
Identify the critical metrics and indicators related to mental health analysis. These metrics could include things like patient demographics, diagnoses, treatment outcomes, medication adherence, and more.

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()

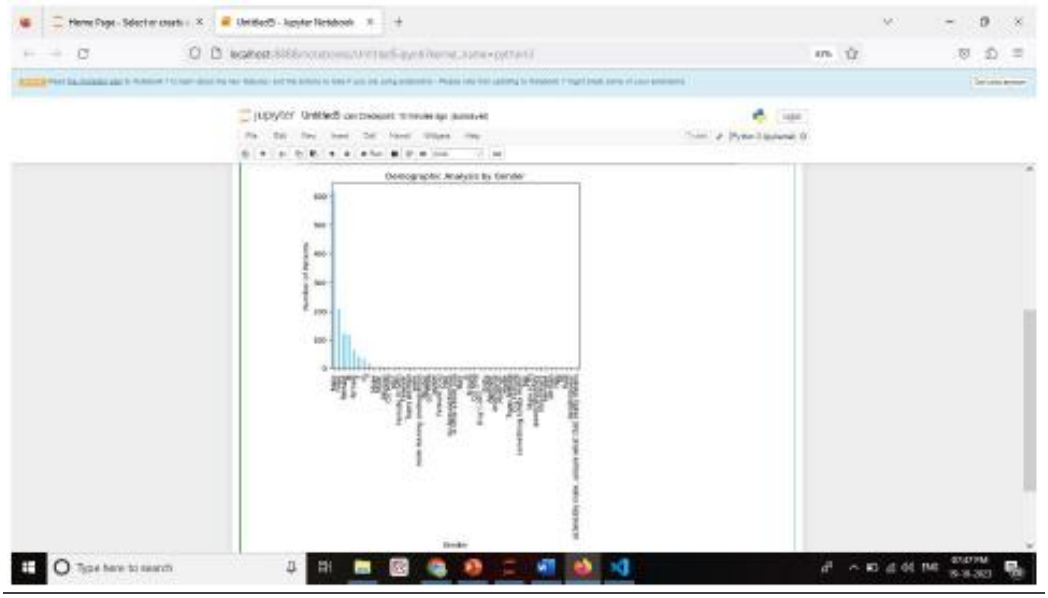
# Visualize the demographic data
demographic_counts.plot(kind='bar', title='Patient Demographics')

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':

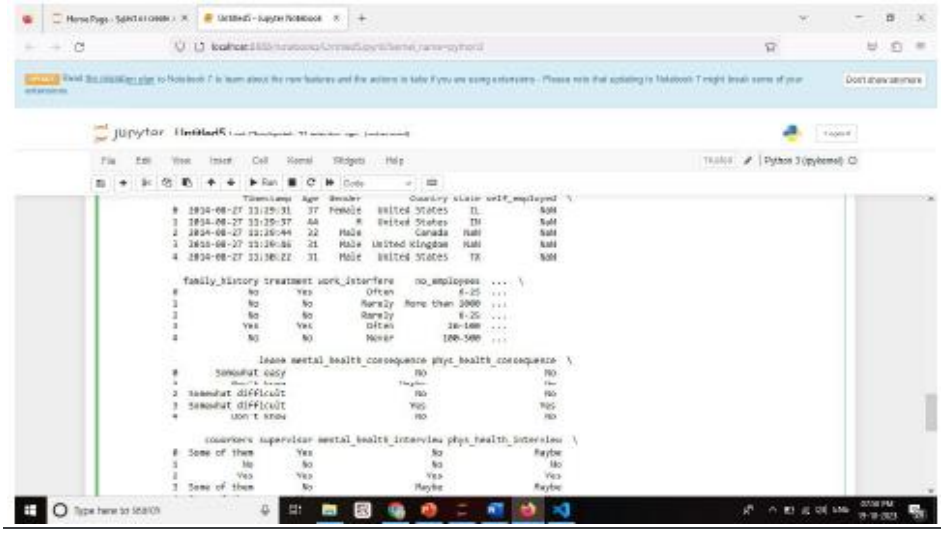
python
Calculate engagement rates
df['Engagement_Rate'] = (df['Resource_Interactions'] / df['Resource_Views']) * 100

code

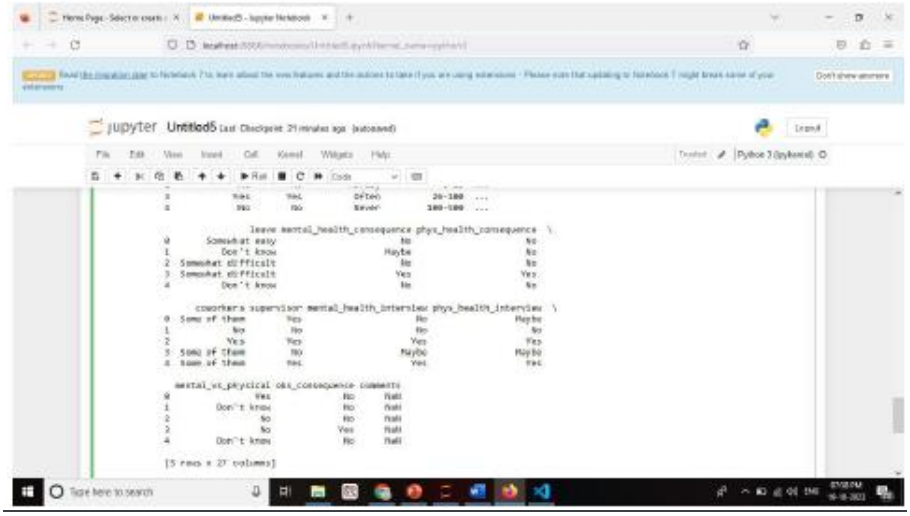
```
import pandas as pd

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
```

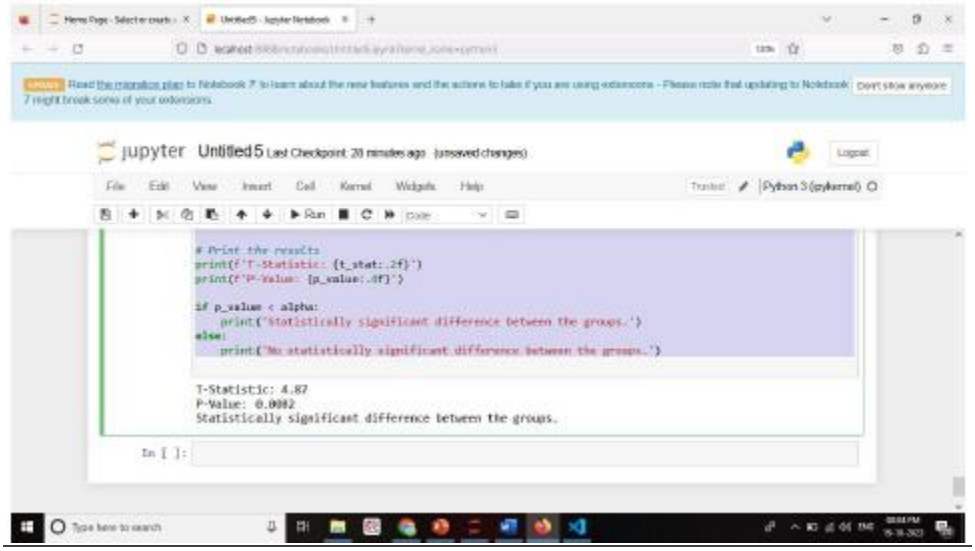
```
group1 = df[df['Treatment_Type'] == 'Group1']['Outcome_Score']
group2 = df[df['Treatment_Type'] == 'Group2']['Outcome_Score']
t_stat, p_value = stats.ttest_ind(group1, group2)
if p_value < 0.05:
print("Statistically significant difference")
else:
print("No significant difference")
```

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:



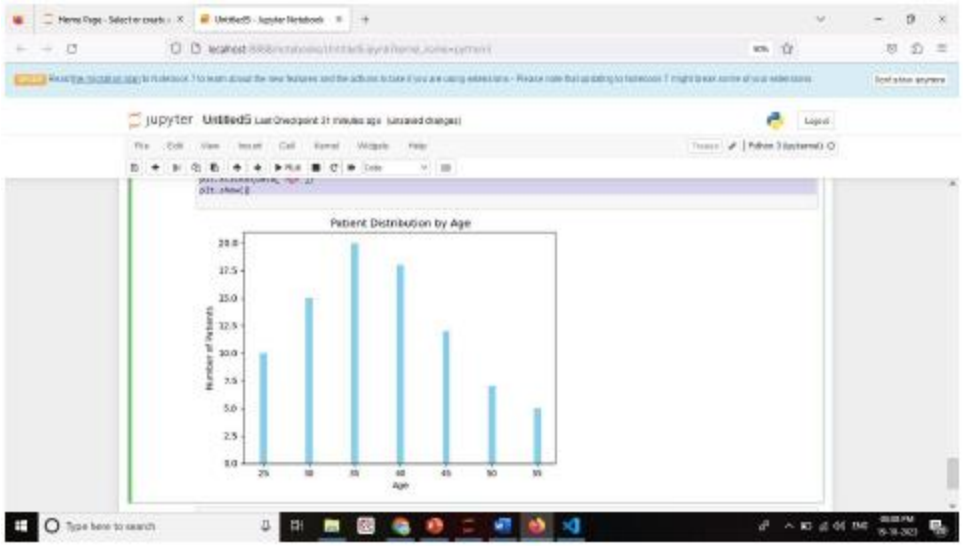
5. Visualization:
Use Matplotlib or other data visualization libraries to create visual representations of your analysis results:

```
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.

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.