

## Phase 4: Development part 2

**Title:** Continue building the analysis by creating visualization using IBM cognos and integrating code for data analysis.

### Introduction:

Creating visualizations and integrating code for data analysis using a mental health in tech survey dataset can be a valuable way to gain insights and communicate your findings. In this example, I'll provide a step-by-step guide using Python and popular data analysis libraries such as Pandas, Matplotlib, and Seaborn. We'll assume you have your dataset in a CSV file.

### Dataset:

To perform create visualizations, you can use publicly available mental health in Tech Survey

datasets. One commonly used dataset is the “Mental Health in Tech Survey” from the UCI Machine Learning

Repository, which contains hourly health data from various locations. Here's how to access and

work with it:

#### 1. Import Necessary Libraries:

Before you start, make sure you have Python and these libraries installed. You can install them using pip.

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

#### 2. Load the Dataset:

Load your dataset into a Pandas DataFrame. You might need to adjust the path and column names based on your dataset.

```
df = pd.read_csv("mental_health_survey.csv")
```

#### 3. Explore the Data: Get a quick overview of dataset.

```
print(df.head())
```

```
print(df.describe())
```

```
print(df.isnull().sum())
```

#### **4. Data Cleaning and Preprocessing:**

Handle missing values and format the data as needed.

```
df.fillna(df.mean(), inplace=True)
```

#### **5. Visualize the Data:**

Create various visualizations to gain insights.

```
sns.histplot(df["Age"], kde=True)
```

```
plt.title("Distribution of Age")
```

```
plt.xlabel("Age")
```

```
plt.ylabel("Count")
```

```
plt.show()
```

```
sns.countplot(data=df, x="Gender")
```

```
plt.title("Gender Distribution")
```

```
plt.xlabel("Gender")
```

```
plt.ylabel("Count")
```

```
plt.xticks(rotation=45)
```

```
plt.show()
```

```
correlation_matrix = df.corr()
```

```
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
```

```
plt.title("Correlation Heatmap")
```

```
plt.show()
```

```
sns.boxplot(data=df, x="WorkInterfere", y="Age")
```

```
plt.title("Work Interference vs. Age")
```

```
plt.xlabel("Work Interference")
```

```
plt.ylabel("Age")
```

```
plt.xticks(rotation=45)
```

```
plt.show()
```

**Output:**



- 6. Interpret the Visualizations:** Analyze the visualizations to draw insights. For example, you can see how age and work interference are related or the distribution of gender in the dataset.
- 7. Further Analysis:** Depending on your research questions, you can perform more advanced analyses like regression, clustering, or sentiment analysis, and create visualizations accordingly.
- 8. Communicate Your Findings:** Document your analysis and findings in a report or presentation, and use the visualizations to support your conclusions.

Remember to replace "mental\_health\_survey.csv" with the actual file path and adjust the column names and data preprocessing steps according to your dataset's structure and specific analysis goals.

assess the success of your marketing or advertising efforts. You can use a variety of visualization techniques to represent this data effectively. Here are some ideas on how to visualize these aspects of a campaign: Visualizing campaign reach, awareness levels, and impact metrics is essential to

### 1. **Bar Charts:**

- Create a bar chart to compare the reach of your campaign across different channels or time periods.
- Use different colors to represent various awareness levels (e.g., low, medium, high).
- Show the impact metrics as different bar lengths, with the height of the bars indicating the level of impact.

### 2. **Line Charts:**

- Use line charts to illustrate the growth in reach and awareness levels over time.
- Include multiple lines to show different impact metrics, allowing you to track their progression over the campaign's duration.

### 3. **Pie Charts:**

- Create a pie chart to visualize the distribution of reach across different demographics or target segments.
- Use multiple pie charts to display the awareness levels and impact metrics for each segment.

### 4. **Heatmaps:**

- Build a heatmap that displays the intensity of campaign impact in various geographic regions.
- Use different colors to represent the reach, awareness, and impact levels.

### 5. **Bubble Charts:**

- Use bubble charts to represent reach, awareness, and impact metrics simultaneously.
- The size of the bubbles can represent the reach, and the color or position can indicate awareness levels and impact metrics.

### 6. **Funnel Chart:**

- Create a funnel chart to illustrate the conversion of reach into awareness and, ultimately, into impact.
- This can help you identify where in the funnel you're losing potential impact.

### 7. **Word Clouds:**

- Use word clouds to visualize the most commonly associated words or themes related to your campaign.
- The size of words in the cloud can indicate their frequency or impact.

### 8. **Geospatial Maps:**

- Employ maps to display the geographical reach and impact of your campaign.
- Use different colors or markers to show awareness and impact levels in specific regions.

### 9. **Stacked Area Charts:**

- Show the cumulative reach and awareness levels over time using a stacked area chart.
- Different layers of the chart can represent various impact metrics.

## 10. **Dashboard or Scorecard:**

- Create a comprehensive dashboard or scorecard that combines various visualizations to provide an overview of reach, awareness, and impact metrics.

Remember to label your visualizations clearly and use a consistent color scheme to ensure that your audience can easily interpret the data. The choice of visualization technique depends on the specific data you have and the story you want to convey about your campaign's reach, awareness, and impact.

I can certainly guide you through the process of performing advanced data analysis on a mental health in Tech survey dataset. However, you'll need to have the dataset at hand, and the specific steps may vary depending on the tools and programming languages you are using. Below, I'll outline a general process for conducting engagement rate calculation, demographic analysis, and statistical tests on your dataset.

### **Data Preparation:**

Import your dataset into a data analysis tool or programming environment such as Python (using libraries like Pandas), R, or any other software you prefer.

Examine the dataset to understand its structure and the variables it contains.

### **Engagement Rate Calculation:**

Define what you mean by "engagement rate." For mental health surveys, this might be the percentage of respondents who engaged with mental health resources, attended workshops, or took advantage of related programs.

Calculate the engagement rate by counting the number of engaged respondents and dividing it by the total number of respondents.

Example in Python:

### **Python code**

```
engagement_rate = (engaged_count / total_count) * 100
```

### **Demographic Analysis:**

Identify the demographic variables in your dataset (e.g., age, gender, ethnicity). Group and filter your data based on these demographic variables to gain insights. Visualize demographic information using charts and graphs (e.g., bar plots, pie charts, histograms).

Example in Python (using Pandas and Matplotlib):

### **Python code**

```
gender_counts = df['Gender'].value_counts()
gender_counts.plot(kind='bar')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```

### **Statistical Tests:**

Determine the research questions you want to answer using statistical tests. For example, you might want to test if there are significant differences in engagement rates between different demographic groups.

Choose the appropriate statistical test (t-test, chi-squared test, ANOVA, etc.) based on your research questions and the type of data (categorical or continuous).

Perform the statistical test using a statistical software or library.

Example in Python (t-test):

### **Python code**

```
from scipy.stats import ttest_ind
group1 = df[df['Group'] == 'Group A']['Engagement']
group2 = df[df['Group'] == 'Group B']['Engagement']
t_stat, p_value = ttest_ind(group1, group2)
```

```
if p_value < 0.05:  
    print("There is a significant difference in engagement between the groups.")  
else:  
    print("There is no significant difference in engagement between the groups.")
```

Remember to replace 'Gender,' 'Group,' 'Engagement,' and other variables with the actual column names and values from your dataset. Also, make sure your dataset is properly cleaned and formatted before conducting any analysis.