
Public Health Awareness Campaigns Data Visualization

Title: Data Visualization for Public Health Awareness Campaigns Analysis

Project Description: The Data Visualization project aims to provide insightful and actionable visual representations of data related to public health awareness campaigns. Visualizations play a crucial role in simplifying complex data, highlighting trends, and facilitating decision-making. This project focuses on transforming cleaned campaign data into meaningful charts, graphs, and visual narratives that help stakeholders gain a deeper understanding of public health awareness efforts.

Project Objectives: Data Selection: Identify and select the most relevant attributes and metrics from the cleaned campaign dataset for visualization.

Visualization Design: Plan and design an array of visualizations, including bar charts, line charts, heatmaps, and more, to represent campaign data effectively.

Storytelling: Create a compelling and data-driven narrative that explains key campaign insights to both technical and non-technical audiences.

Interactive Dashboards: Develop interactive dashboards or reports to allow users to explore campaign data, filter information, and gain real-time insights.

Insight Extraction: Extract actionable insights and trends from the visualizations that can guide future public health awareness campaigns.

Feedback Incorporation: Collaborate with stakeholders and subject matter experts to incorporate feedback and refine visualizations.

Documentation: Maintain clear documentation of visualization methodologies, tools, and data sources for reproducibility.

Source:

Cleaned data from phase 3

Code:

```
Import pandas as pd
```

```
Import numpy as np
```

```
Import matplotlib.pyplot as plt
```

```
From sklearn.model_selection import train_test_split
```

```
From sklearn.linear_model import LogisticRegression
```

```
From sklearn.metrics import accuracy_score, classification_report
```

```
From sklearn.preprocessing import LabelEncoder
```

```
# Load CSV data into a Pandas DataFrame
```

```
Data = pd.read_csv('your_data.csv')
```

```
# Define the target variable and features
```

```
Target_column = 'treatment'
```

```
# Replace with specific target column
```

```
Features = ['Age', 'Gender', 'Country', 'family_history', 'work_interfere', 'no_employees', 'remote_work',  
            'tech_company', 'benefits', 'care_options', 'wellness_program', 'seek_help', 'anonymity', 'leave',  
            'mental_health_consequence', 'phys_health_consequence', 'coworkers', 'supervisor',  
            'mental_health_interview', 'phys_health_interview', 'mental_vs_physical', 'obs_consequence']
```

```
# Filter the DataFrame to include only the selected columns
```

```
Data = data[[target_column] + features]
```

```
# Handle categorical data by encoding it
```

```
le = LabelEncoder()
```

```
Data['Gender'] = le.fit_transform(data['Gender'])
```

```
Data['Country'] = le.fit_transform(data['Country'])
```

```
# Split the data into training and testing sets
```

```
X = data.drop(columns=[target_column])
```

```
Y = data[target_column]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Model building (example: Logistic Regression)
```

```
Model = LogisticRegression()
```

```
Model.fit(X_train, y_train)
```

```
# Make predictions
```

```
Y_pred = model.predict(X_test)
```

```
# Evaluation
```

```
Accuracy = accuracy_score(y_test, y_pred)
```

```
Print(f"Accuracy: {accuracy}")
```

```
Print(classification_report(y_test, y_pred))
```

```
# Data visualization (example: a histogram of 'Age')
```

```
Plt.hist(data['Age'], bins=20, color='blue', alpha=0.7)
```

```
Plt.xlabel('Age')
```

```
Plt.ylabel('Frequency')
```

```
Plt.title('Age Distribution')
```

```
Plt.show()
```