Building a Public Health Awareness Campaign Analysis

Defining Analysis Objectives

Before getting started , let's have a look at the objectives of our public health awareness campaign analysis. By setting clear goals, we can focus our efforts and extract maximum value from the collected data.

1. To Identify the area of Improvement in the Public health Awareness Campaigns

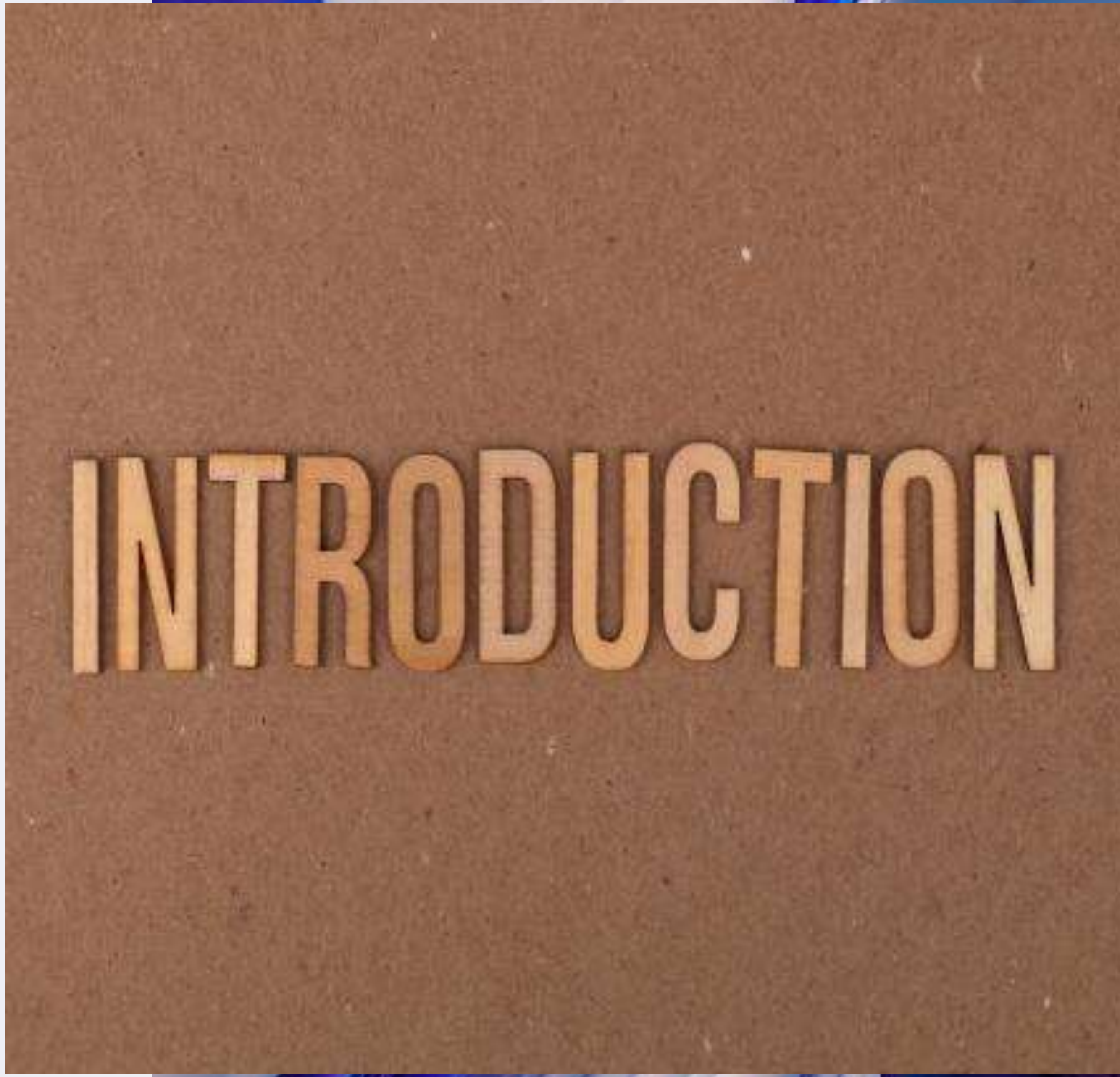
2. Analyze and Explore the survey data set to get accurate results

3. Explore new and hidden pattern exist
4. in survey dataset

4. To identify the relationship between variables and data points of mental health predication

Introduction

Welcome to the world of public health awareness campaigns! Infact health is everyone's right but unfortunately many are not able to get awareness about the health and take care of themselves. Main aim of conducting these campaigns are to provide essential awareness and prevent themselves from upcoming diseases to lead a healthy life.In this presentation, we'll dive into the analysis of these campaigns done through IBM Cognos data analytics and explore their impact of public health awareness campaigns on society. Get ready for an insightful journey!



Loading the dataset

Before embarking on any analysis, it's crucial to load the public health awareness campaign dataset. By properly preparing the data, we can ensure reliable results and meaningful insights.

jupyter Untitled1 Last Checkpoint: 8 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
[1]: import pandas as pd
     import numpy as np
     import seaborn as sns
     import matplotlib.pyplot as plt
```

```
[16]: print("Loading The Dataset\n")
      ds = pd.read_csv('C:\\Users\\CSE-LAB1-S08\\Desktop\\survey.csv')
      ds.head()
```

```
Loading The Dataset
```

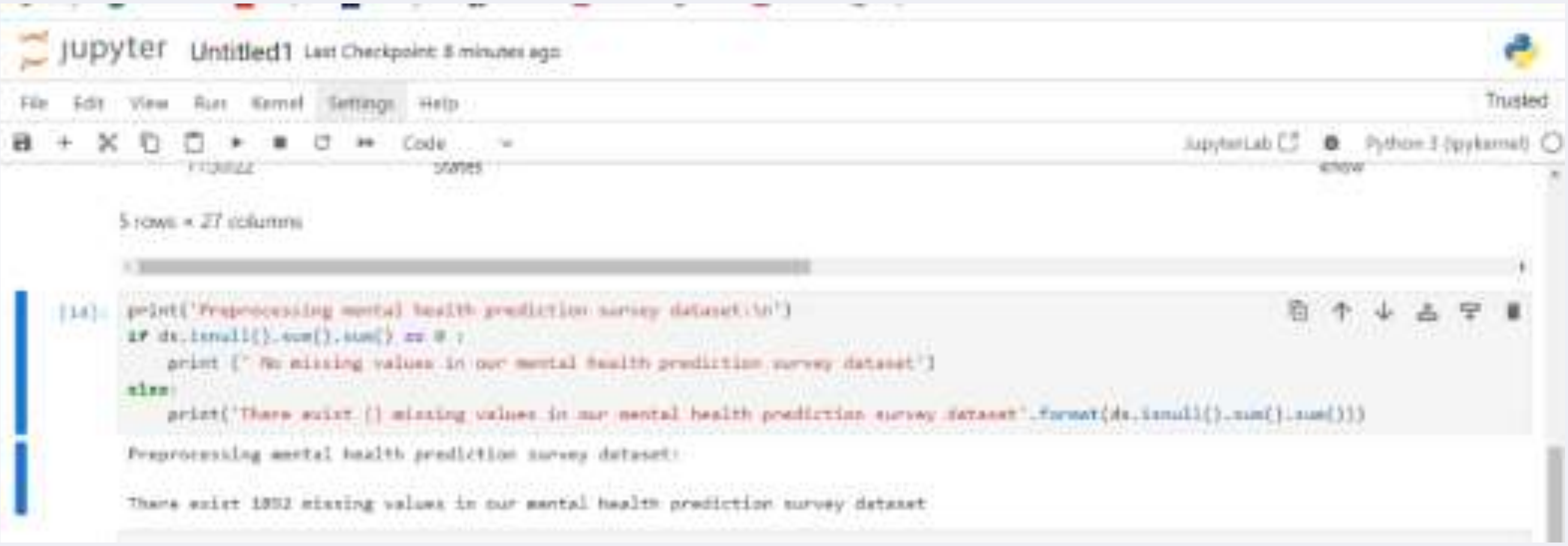
[16]:

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	...	leave	mental_health_consequence	phys_
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN	No	Yes	Often	6-25	...	Somewhat easy	No	
1	2014-08-27 11:29:37	44	M	United States	IN	NaN	No	No	Rarely	More than 1000	...	Don't know	Maybe	
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN	No	No	Rarely	6-25	...	Somewhat difficult	No	
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN	Yes	Yes	Often	26-100	...	Somewhat difficult	Yes	
4	2014-08-27 11:30:22	31	Male	United States	TX	NaN	No	No	Never	100-500	...	Don't know	No	

5 rows × 27 columns

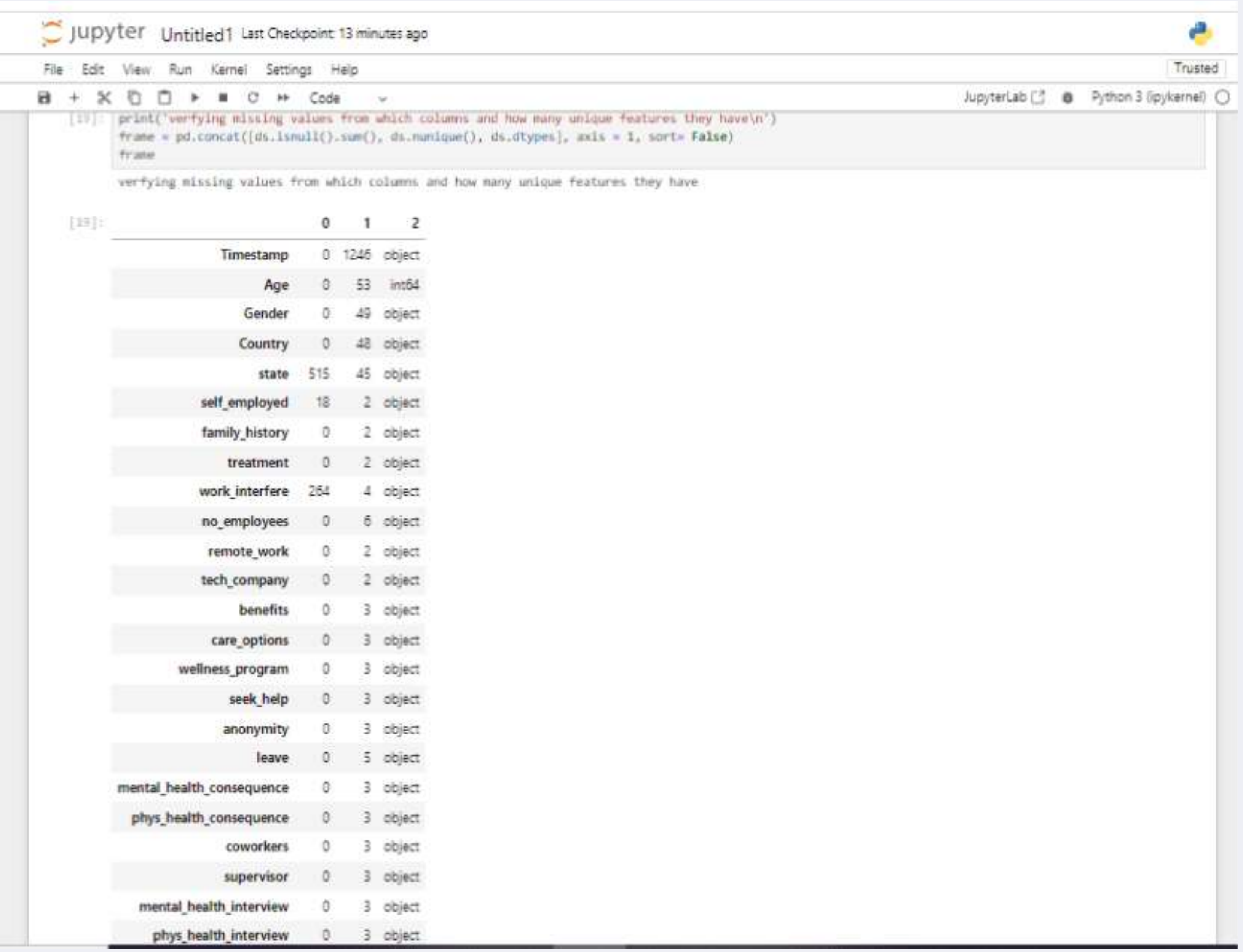
Preprocessing the dataset

Data preprocessing is a fundamental stage in data mining that involves transforming raw data into a usable format for a machine learning model



Loading...

Preprocessing the dataset


jupyter Untitled1 Last Checkpoint: 13 minutes ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[19]:
```
print('verfying missing values from which columns and how many unique features they have\n')
frame = pd.concat([ds.isnull().sum(), ds.nunique(), ds.dtypes], axis = 1, sort= False)
frame
```

```
verfying missing values from which columns and how many unique features they have
```

[19]:

	0	1	2
Timestamp	0	1246	object
Age	0	53	int64
Gender	0	49	object
Country	0	48	object
state	515	45	object
self_employed	18	2	object
family_history	0	2	object
treatment	0	2	object
work_interfere	264	4	object
no_employees	0	6	object
remote_work	0	2	object
tech_company	0	2	object
benefits	0	3	object
care_options	0	3	object
wellness_program	0	3	object
seek_help	0	3	object
anonymity	0	3	object
leave	0	5	object
mental_health_consequence	0	3	object
phys_health_consequence	0	3	object
coworkers	0	3	object
supervisor	0	3	object
mental_health_interview	0	3	object
phys_health_interview	0	3	object

jupyter Untitled1 Last Checkpoint: 20 days ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[37]:

```
from sklearn.preprocessing import MaxAbsScaler, StandardScaler

data['Age'] = MaxAbsScaler().fit_transform(data[['Age']])
data['Country'] = StandardScaler().fit_transform(data[['Country']])
data['work_interfere'] = StandardScaler().fit_transform(data[['work_interfere']])
data['no_employees'] = StandardScaler().fit_transform(data[['no_employees']])
data['leave'] = StandardScaler().fit_transform(data[['leave']])

data.describe()
```

[37]:

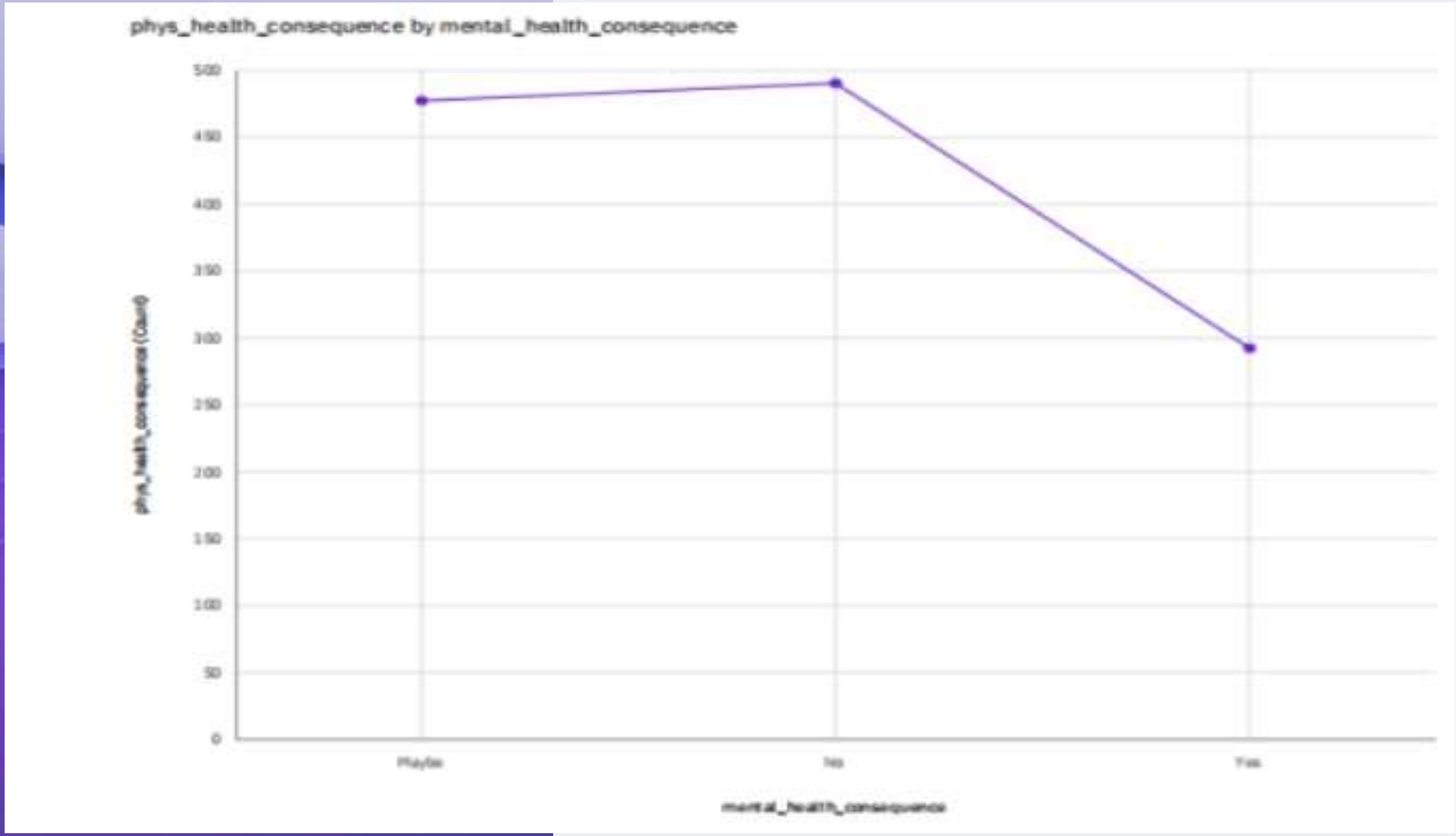
	Age	Gender	Country	self_employed	family_history	treatment	work_interfere	no_employees	remote_work	tech_company	...	anonymity
count	1250.000000	1250.00000	1.250000e+03	1250.000000	1250.000000	1250.000000	1.250000e+03	1.250000e+03	1250.000000	1250.000000	...	1250.000000
mean	0.444778	0.81760	3.979039e-17	0.114400	0.390400	0.504800	-8.242296e-17	-1.705303e-17	0.298400	0.820000	...	0.648000
std	0.102557	0.42388	1.000400e+00	0.318424	0.488035	0.500177	1.000400e+00	1.000400e+00	0.457739	0.384341	...	0.909482
min	0.069444	0.00000	-2.835244e+00	0.000000	0.000000	0.000000	-1.790702e+00	-1.603187e+00	0.000000	0.000000	...	0.000000
25%	0.375000	1.00000	3.156273e-01	0.000000	0.000000	0.000000	-9.731601e-01	-1.027826e+00	0.000000	1.000000	...	0.000000
50%	0.430556	1.00000	5.406895e-01	0.000000	0.000000	1.000000	4.819238e-01	1.228972e-01	0.000000	1.000000	...	0.000000
75%	0.500000	1.00000	3.406895e-01	0.000000	1.000000	1.000000	4.819238e-01	6.982587e-01	1.000000	1.000000	...	2.000000
max	1.000000	2.00000	6.157103e-01	1.000000	1.000000	1.000000	1.209466e+00	1.273620e+00	1.000000	1.000000	...	2.000000

8 rows × 24 columns

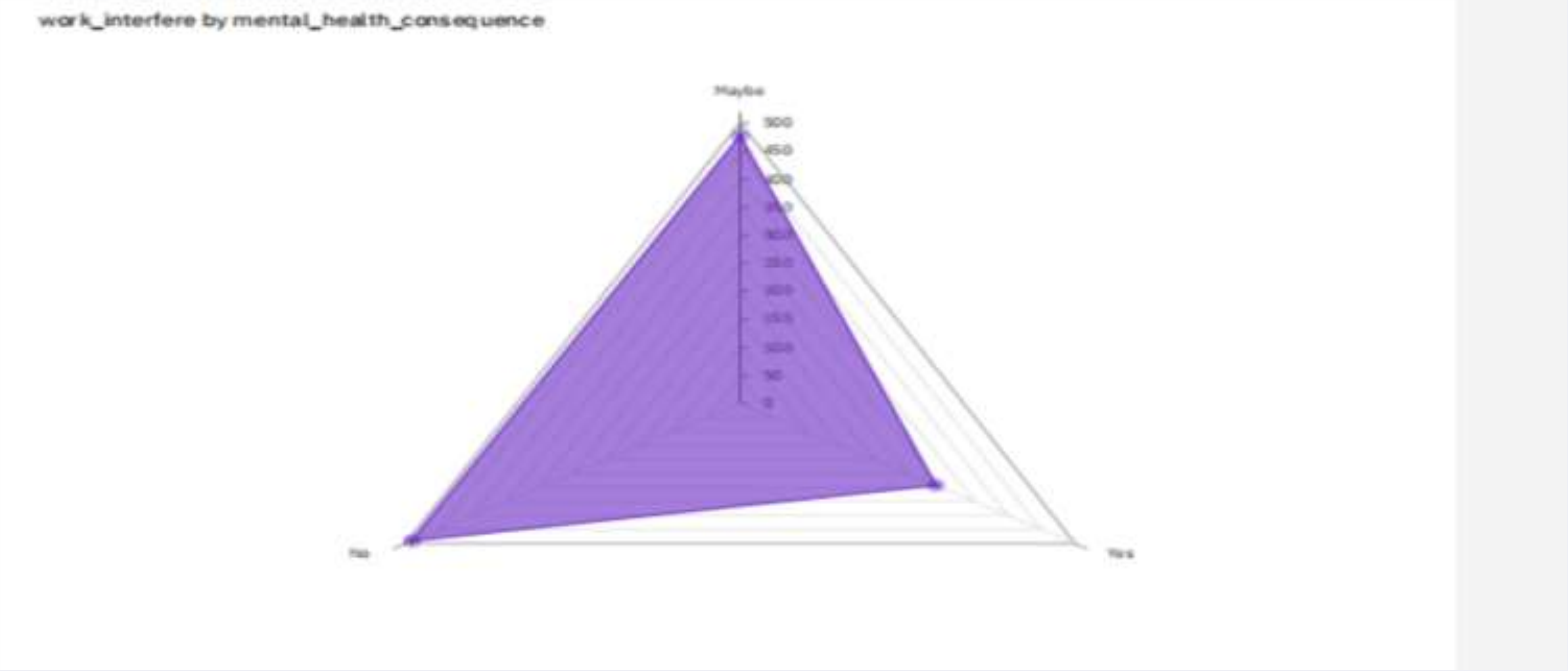
Building the Analysis with IBM Cognos

To visualize the public health awareness campaign data effectively, we'll leverage the power of IBM Cognos. This powerful tool will enable us to create compelling visualizations that highlight key trends and patterns.

Physical health consequence by Mental Health Consequence



Work Interfere by Mental health consequence

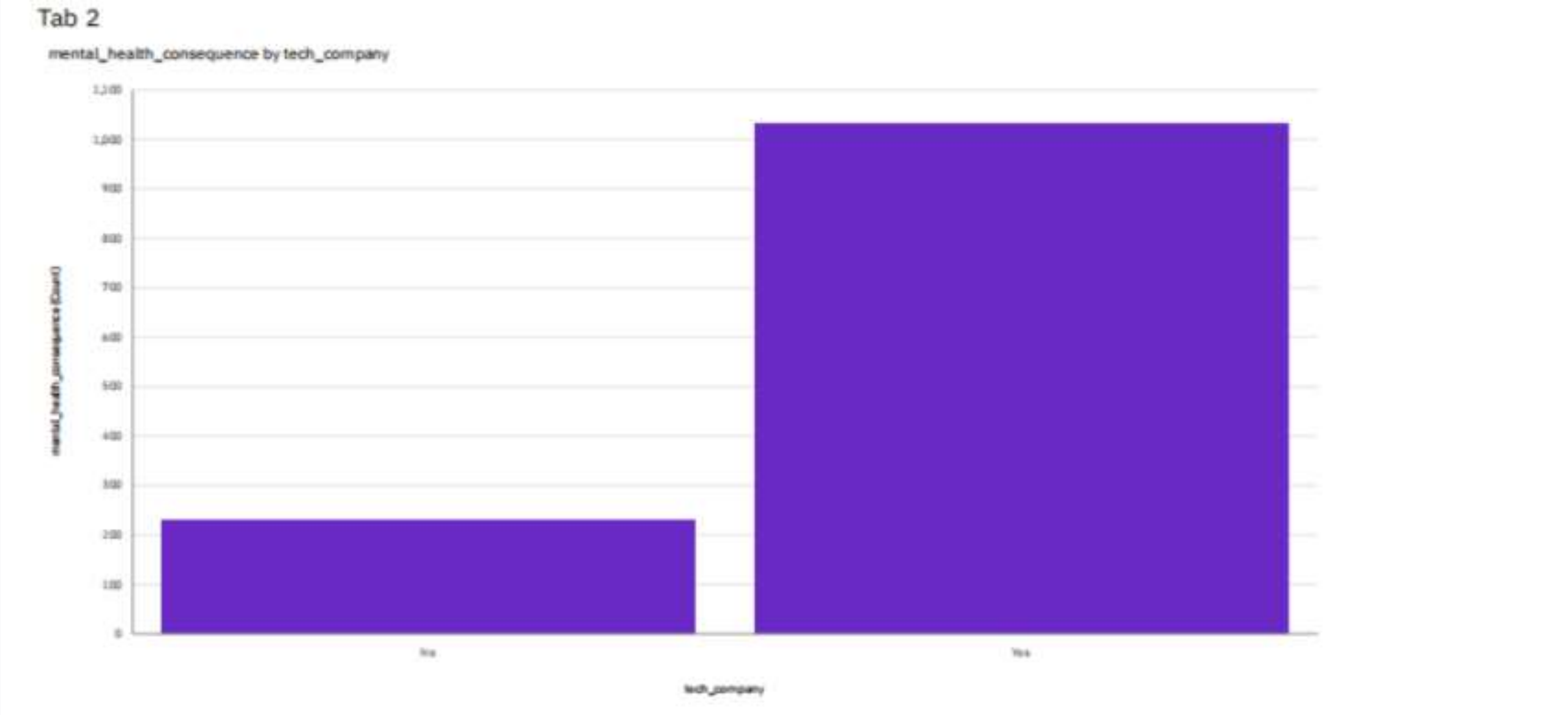


Working on Treatment Column

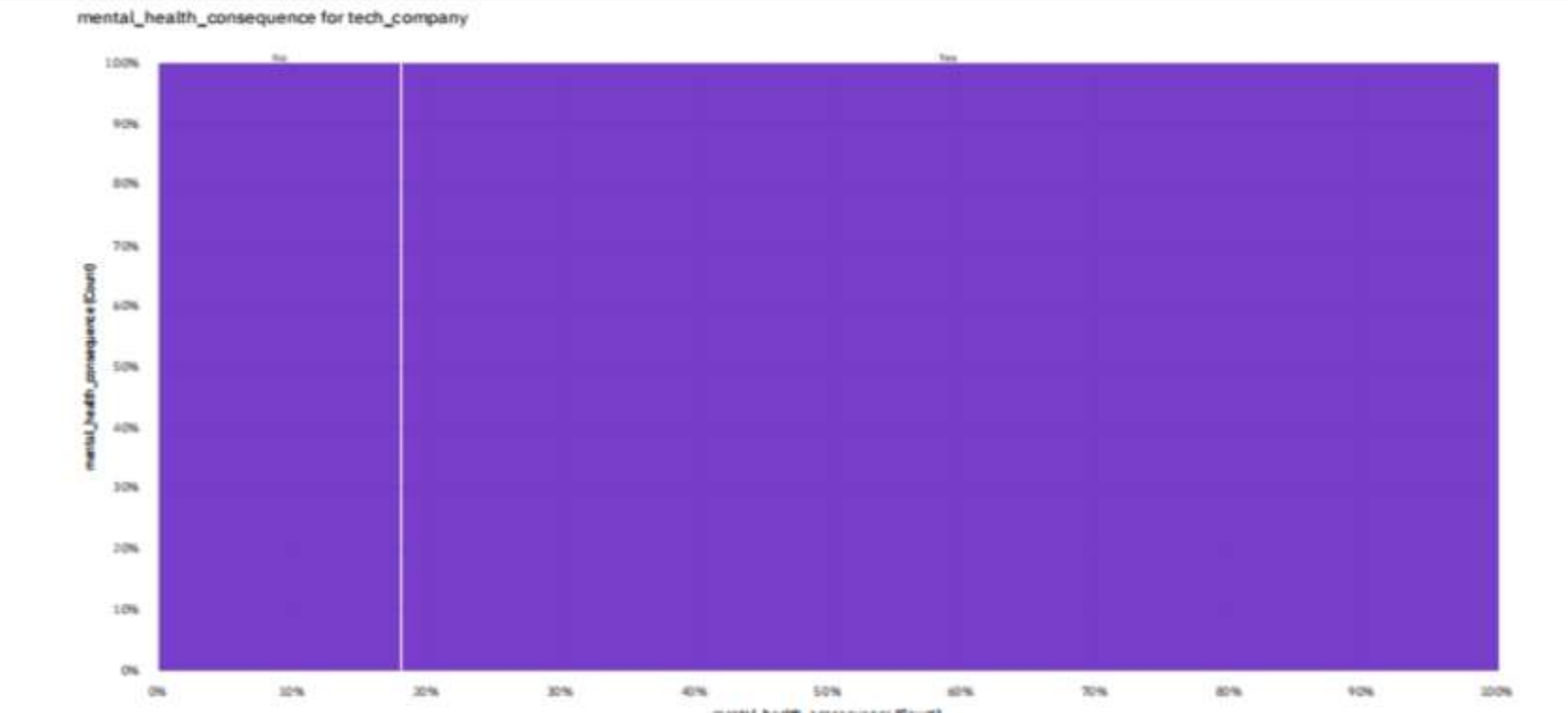
```
[38]: from sklearn.model_selection import train_test_split

      #I wanna work on 'treatment' column.
      X = data.drop(columns = ['treatment'])
      y = data['treatment']

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

      print(X_train.shape, y_train.shape)
      print('-'*30)
      print(X_test.shape, y_test.shape)
      print('_'*30)
```

```
(937, 23) (937,)
------------------------------
(313, 23) (313,)
```

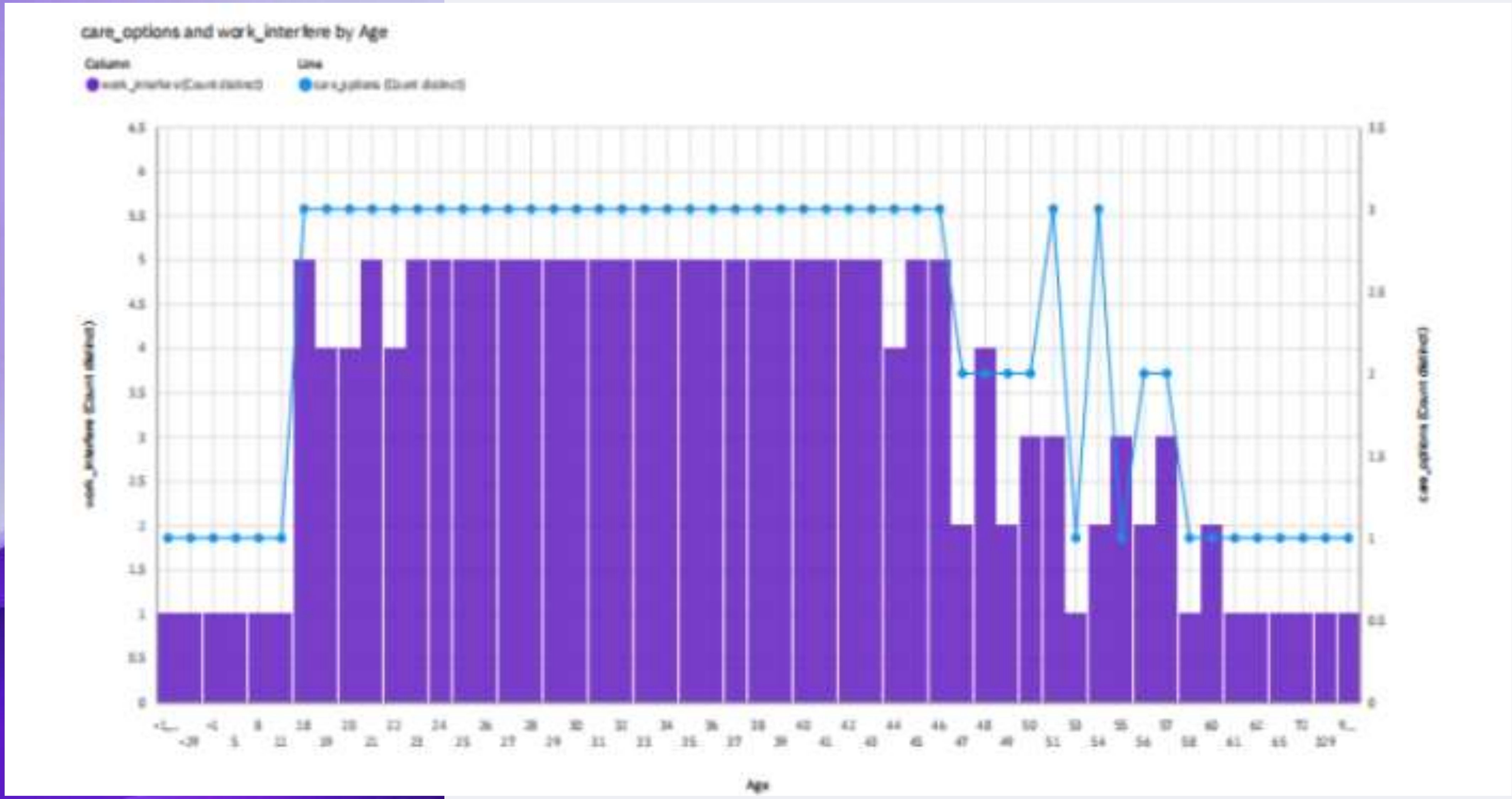
Mental Health Consequence by Tech Company



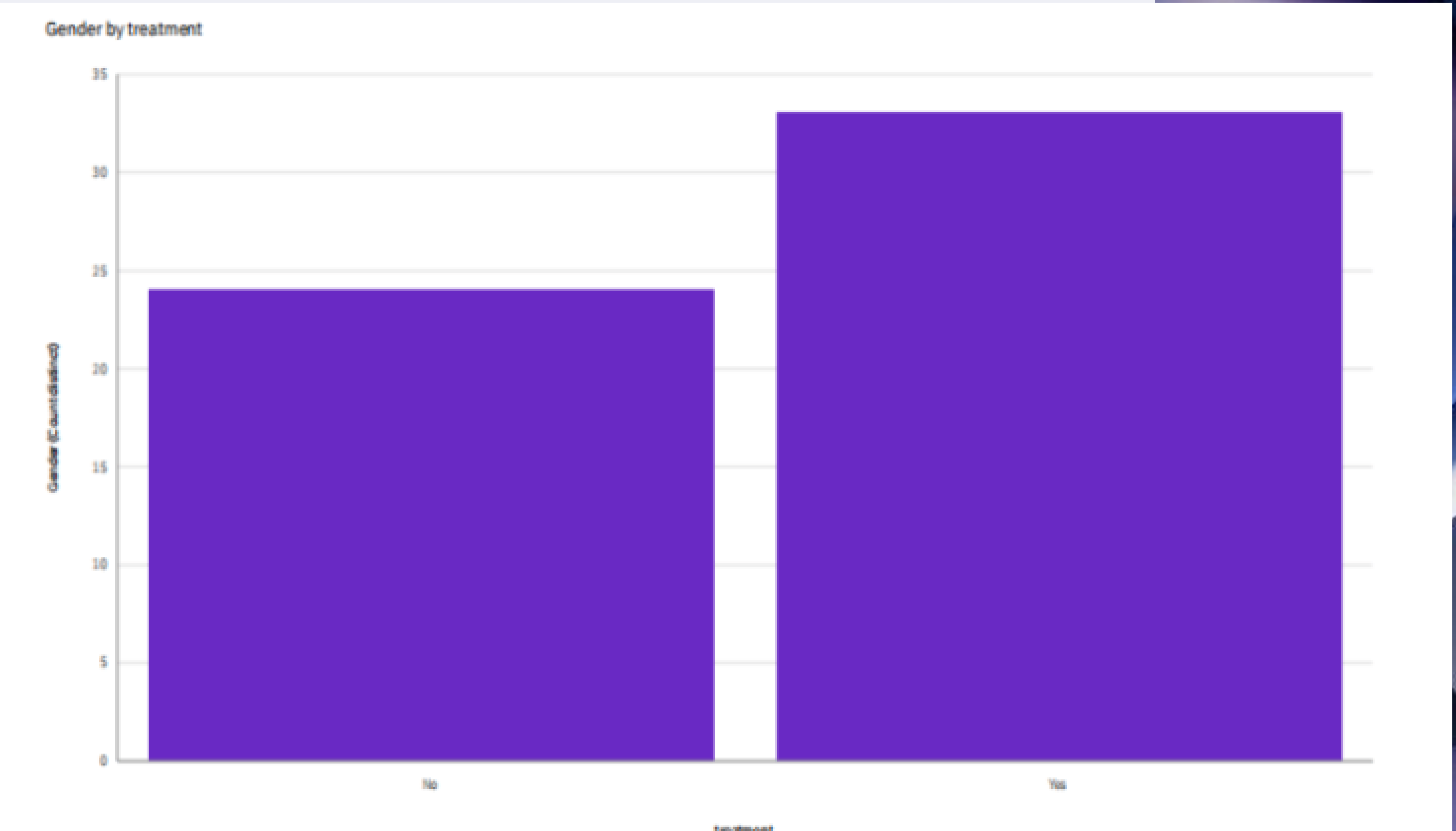
Mental Health Consequences for Tech Company



Care Options and Work Interfere by Age



Gender By Treatment



Data Collection and Cleaning

No analysis is complete without reliable data. We'll gather campaign data from the share source and meticulously clean and process it. Our aim is to ensure quality and accuracy, providing a solid foundation for our analysis.

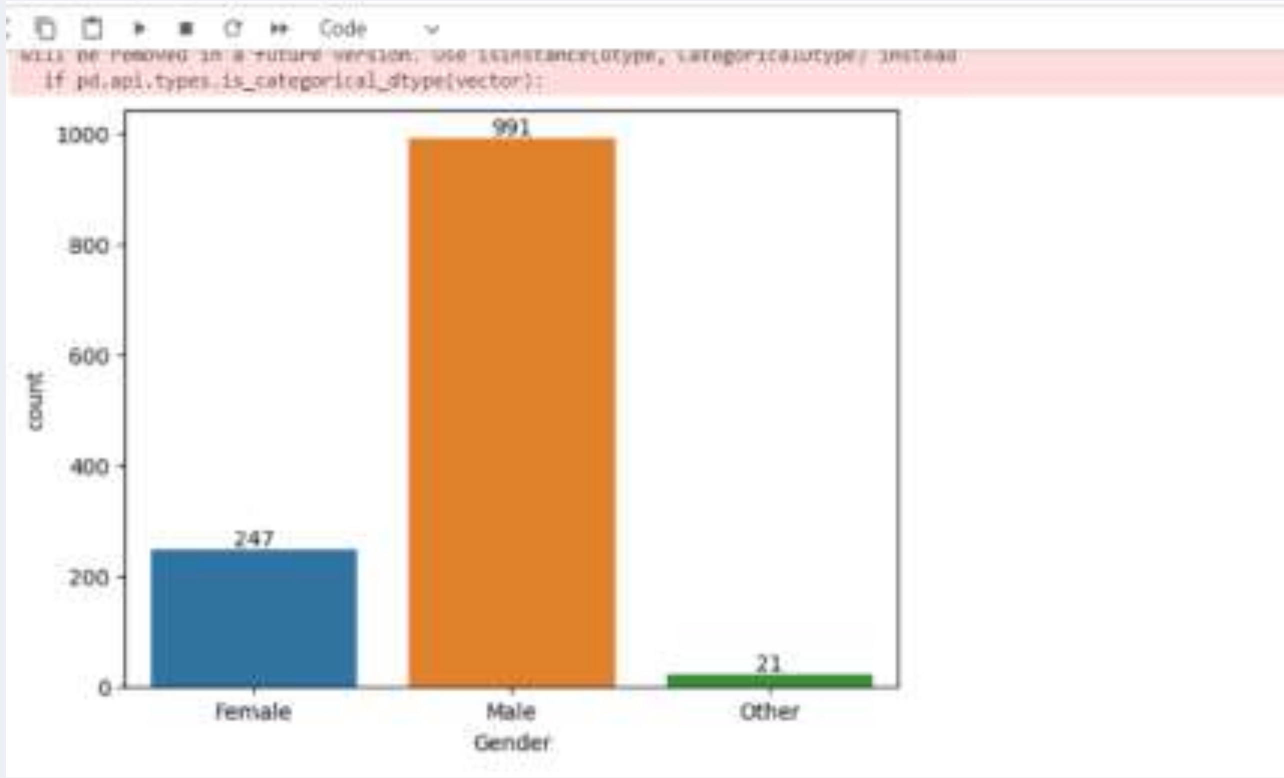
Data is collected from survey dataset provided by Ibm data analytics link. Finding the unique categories of gender in order to remove the unwanted categories

```
[22]: print(data['Gender'].unique())
print('')
print('-'*75)
print('')
#Check number of unique data too.
print('number of unique Gender in our dataset is :', data['Gender'].nunique())

['Female' 'M' 'Male' 'male' 'female' 'm' 'Male-ish' 'maile' 'Trans-female'
 'Cis Female' 'F' 'something kinda male?' 'Cis Male' 'Woman' 'f' 'Mal'
 'Male (CIS)' 'queer/she/they' 'non-binary' 'Femake' 'woman' 'Make' 'Nah'
 'All' 'Enby' 'fluid' 'Genderqueer' 'Female ' 'Androgyne' 'Agender'
 'cis-female/femme' 'Guy (-ish) ^_^' 'male leaning androgynous' 'Male '
 'Man' 'Trans woman' 'msle' 'Neuter' 'Female (trans)' 'queer'
 'Female (cis)' 'Mail' 'cis male' 'A little about you' 'Malr' 'p' 'femail'
 'Cis Man' 'ostensibly male, unsure what that really means']

---------------------------------------------------------------------------
```

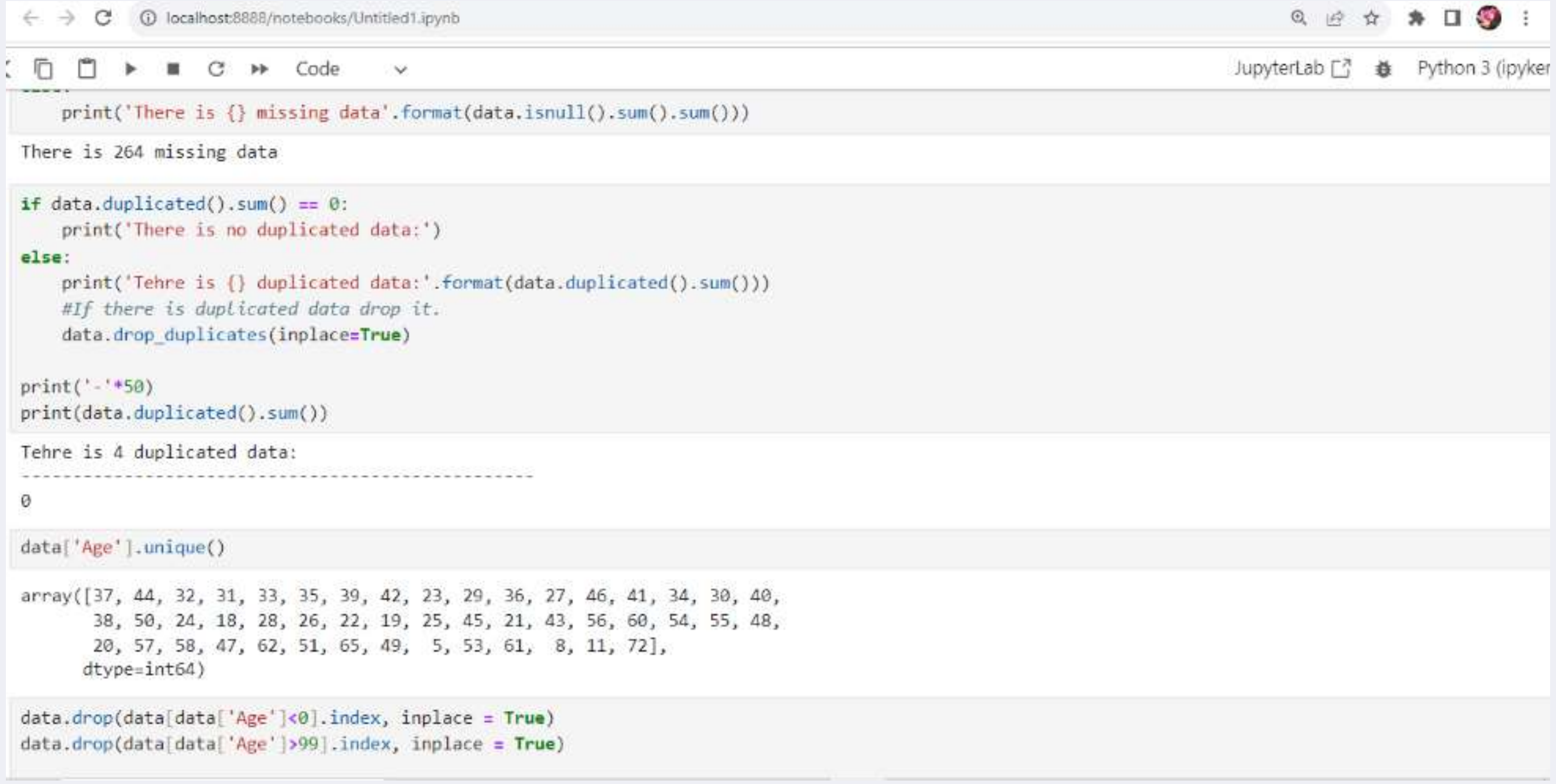
Now cleaning:

```
t View Run Kernel Settings Help
Code                                    JupyterLab    Python 3 (ipyk

number of unique Gender in our dataset is : 49

data['Gender'].replace(['Male ', 'male', 'M', 'm', 'Male', 'Cis Male',
                     'Man', 'cis male', 'Mail', 'Male-ish', 'Male (CIS)',
                      'Cis Man', 'msle', 'Malr', 'Mal', 'maile', 'Make',], 'Male', inplace = True)

data['Gender'].replace(['Female ', 'female', 'F', 'f', 'Woman', 'Female',
                     'femail', 'Cis Female', 'cis-female/femme', 'Femake', 'Female (cis)',
                     'woman',], 'Female', inplace = True)

data["Gender"].replace(['Female (trans)', 'queer/she/they', 'non-binary',
                     'fluid', 'queer', 'Androgyne', 'Trans-female', 'male leaning androgynous',
                      'Agender', 'A little about you', 'Nah', 'All',
                      'ostensibly male, unsure what that really means',
                      'Genderqueer', 'Enby', 'p', 'Neuter', 'something kinda male?',
                      'Guy (-ish) ^_^', 'Trans woman',], 'Other', inplace = True)

print(data['Gender'].unique())

['Female' 'Male' 'Other']

ax = sns.countplot(data=data, x='Gender');
ax.bar_label(ax.containers[0]);
```

After cleansing Insight can be obtained as



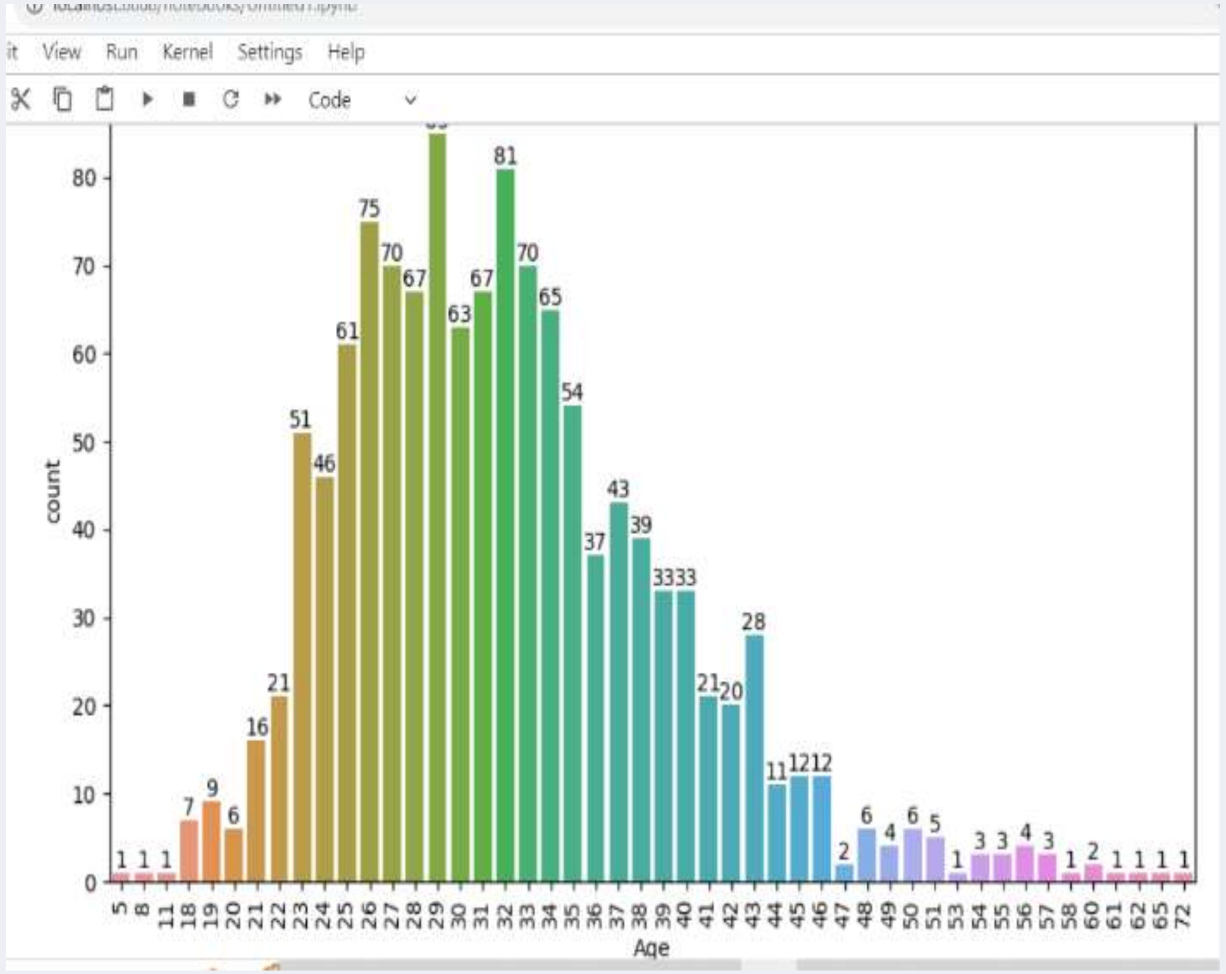
Checking for the missing and duplicate data

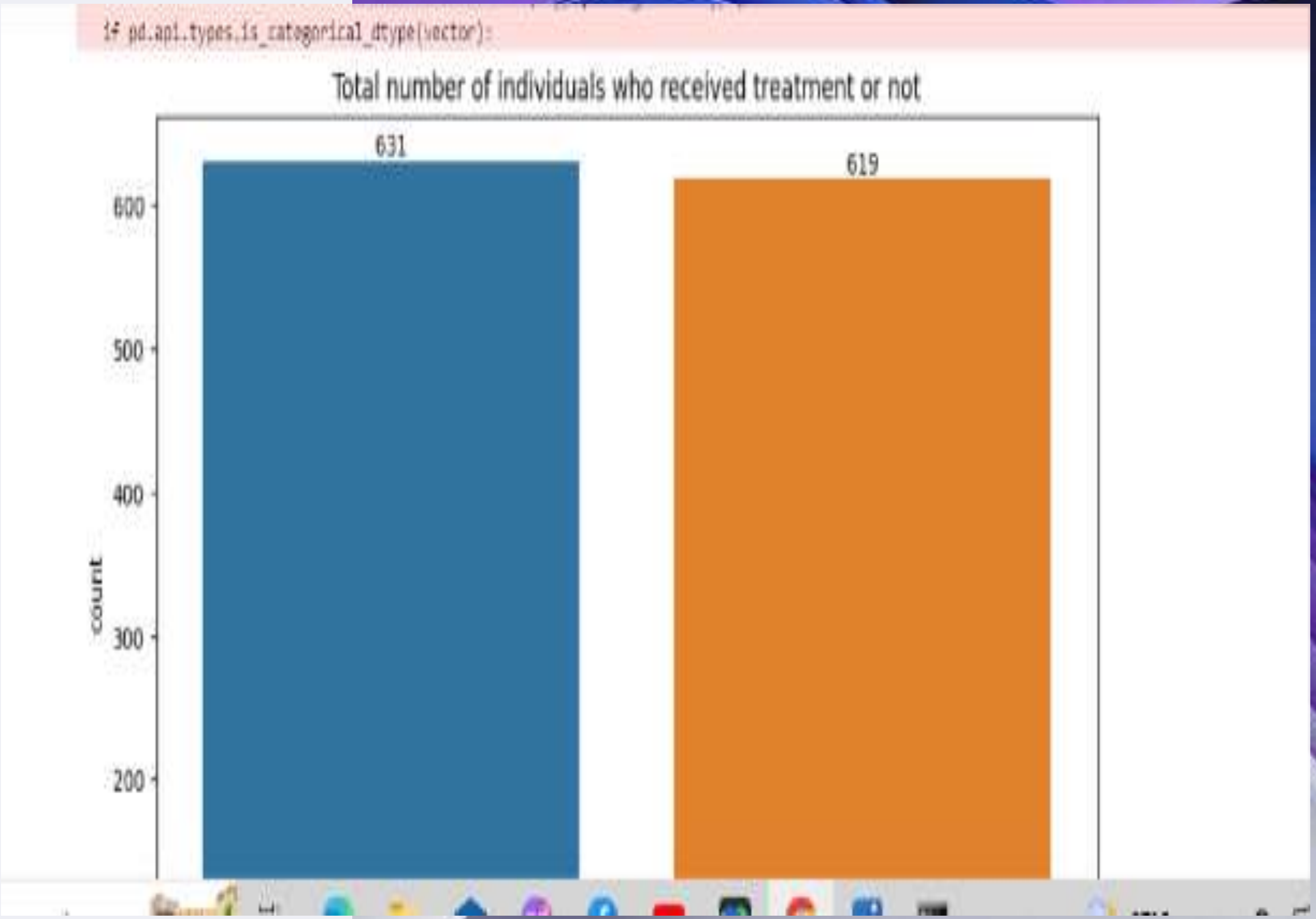


```
localhost:8888/notebooks/Untitled1.ipynb
Code                                             JupyterLab    Python 3 (ipyker

    print('There is {} missing data'.format(data.isnull().sum().sum()))

There is 264 missing data

if data.duplicated().sum() == 0:
    print('There is no duplicated data:')
else:
    print('Tehre is {} duplicated data:'.format(data.duplicated().sum()))
    #If there is duplicated data drop it.
    data.drop_duplicates(inplace=True)

print('-'*50)
print(data.duplicated().sum())

Tehre is 4 duplicated data:
--------------------------------------------------
0

data['Age'].unique()

array([37, 44, 32, 31, 33, 35, 39, 42, 23, 29, 36, 27, 46, 41, 34, 30, 40,
       38, 50, 24, 18, 28, 26, 22, 19, 25, 45, 21, 43, 56, 60, 54, 55, 48,
       20, 57, 58, 47, 62, 51, 65, 49,  5, 53, 61,  8, 11, 72],
      dtype=int64)

data.drop(data[data['Age']<0].index, inplace = True)
data.drop(data[data['Age']>99].index, inplace = True)
```

Working on the error data to ressolve it

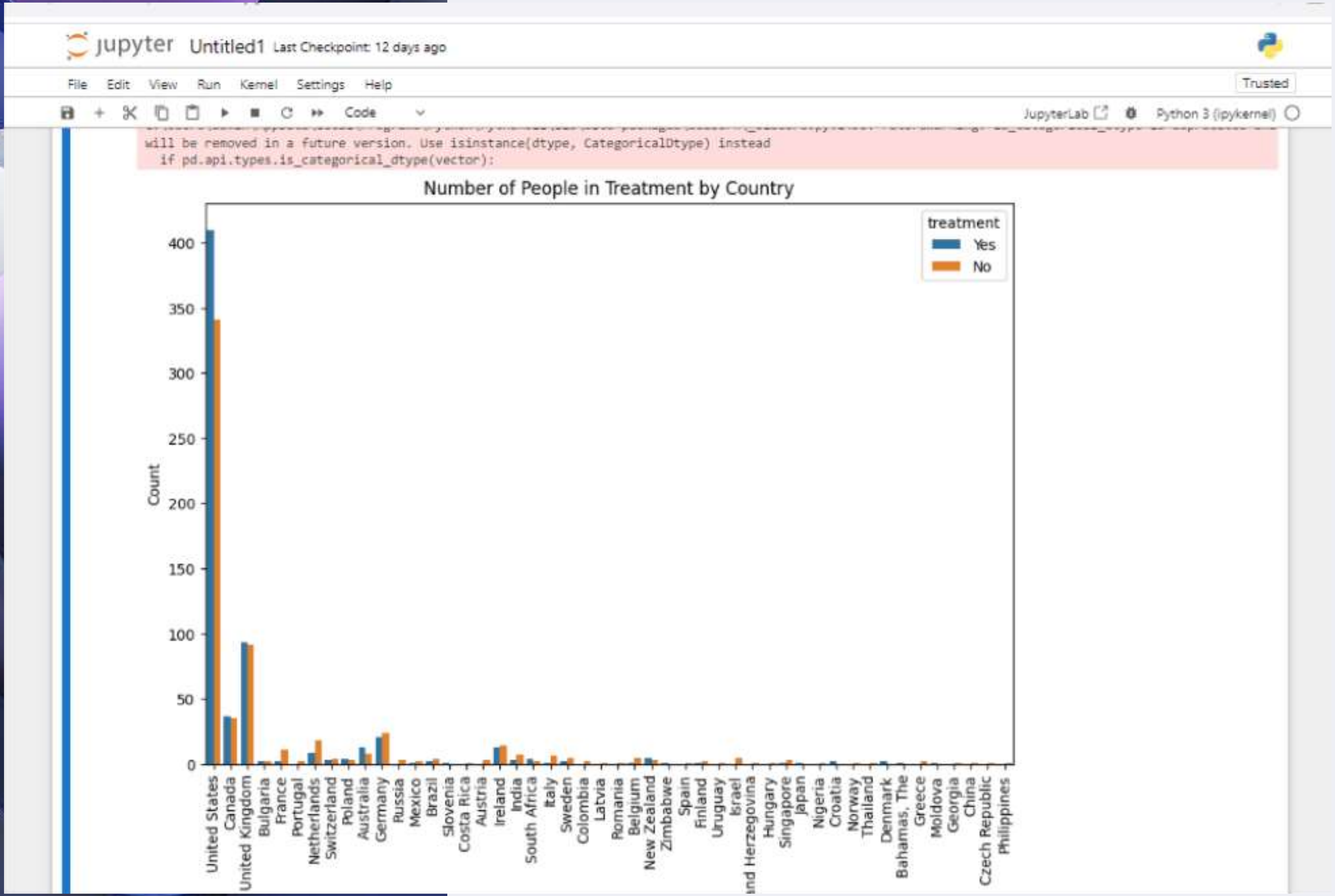
After resolving insights obtained can be





Exploring Mental Health Prediction Data

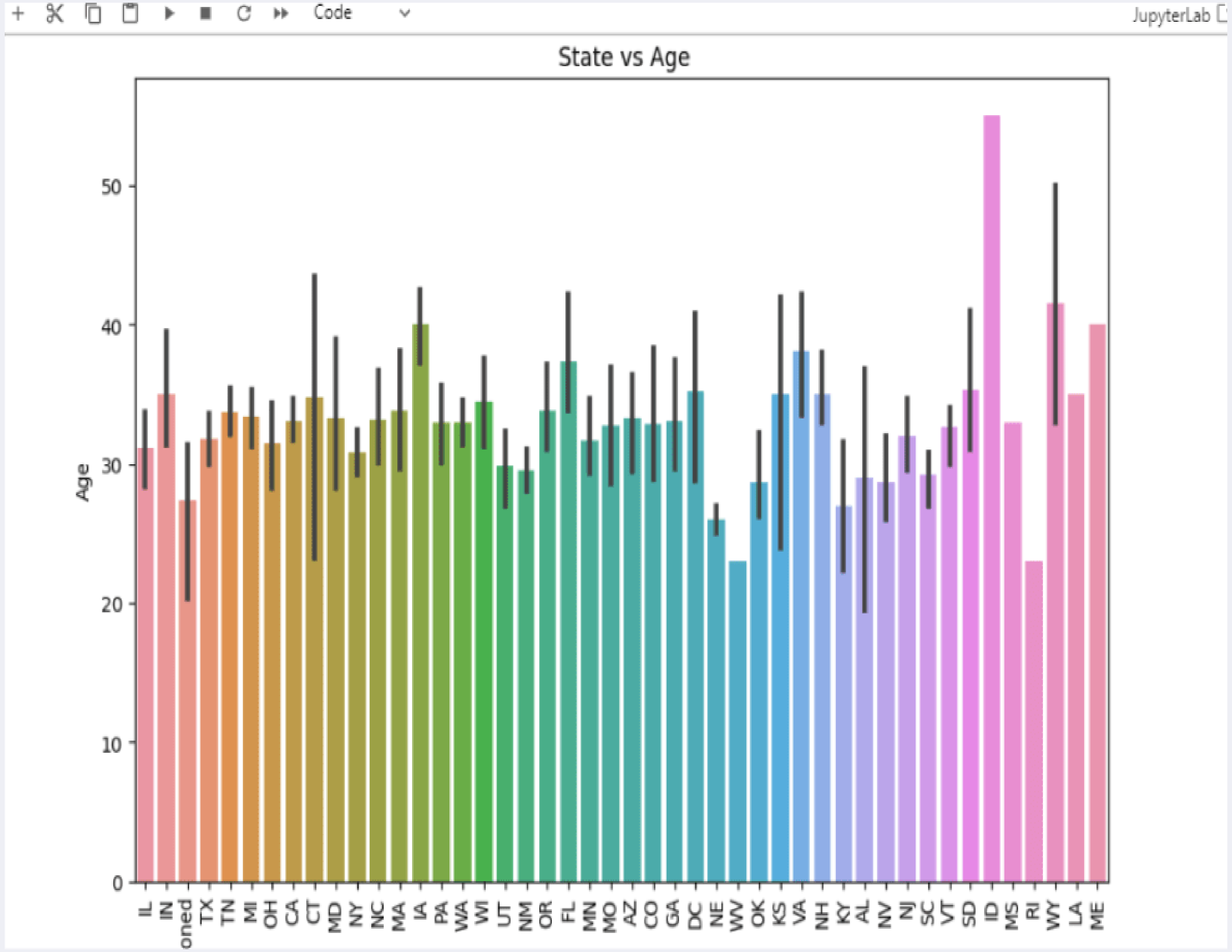
In addition to the public health awareness campaign analysis, we'll look into mental health prediction datasets. By combining these two domains, we can uncover fascinating insights and potential correlations.


jupyter Untitled1 Last Checkpoint: 12 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
Number of People in Treatment by Country
treatment
Yes
No
Count
400
350
300
250
200
150
100
50
0
United States
Canada
United Kingdom
Bulgaria
France
Portugal
Netherlands
Switzerland
Poland
Australia
Germany
Russia
Mexico
Brazil
Slovenia
Costa Rica
Austria
Ireland
India
South Africa
Italy
Sweden
Colombia
Latvia
Romania
Belgium
New Zealand
Zimbabwe
Spain
Finland
Uruguay
Israel
Hungary
Singapore
Japan
Nigeria
Croatia
Norway
Thailand
Denmark
Bahamas, The
Greece
Moldova
Georgia
China
Czech Republic
Philippines

Analyzing Mental Health Prediction Data

By applying advanced analytical techniques to the mental health prediction datasets, we'll unravel key findings and observations. Prepare to be surprised by the interplay between public health awareness and mental health.

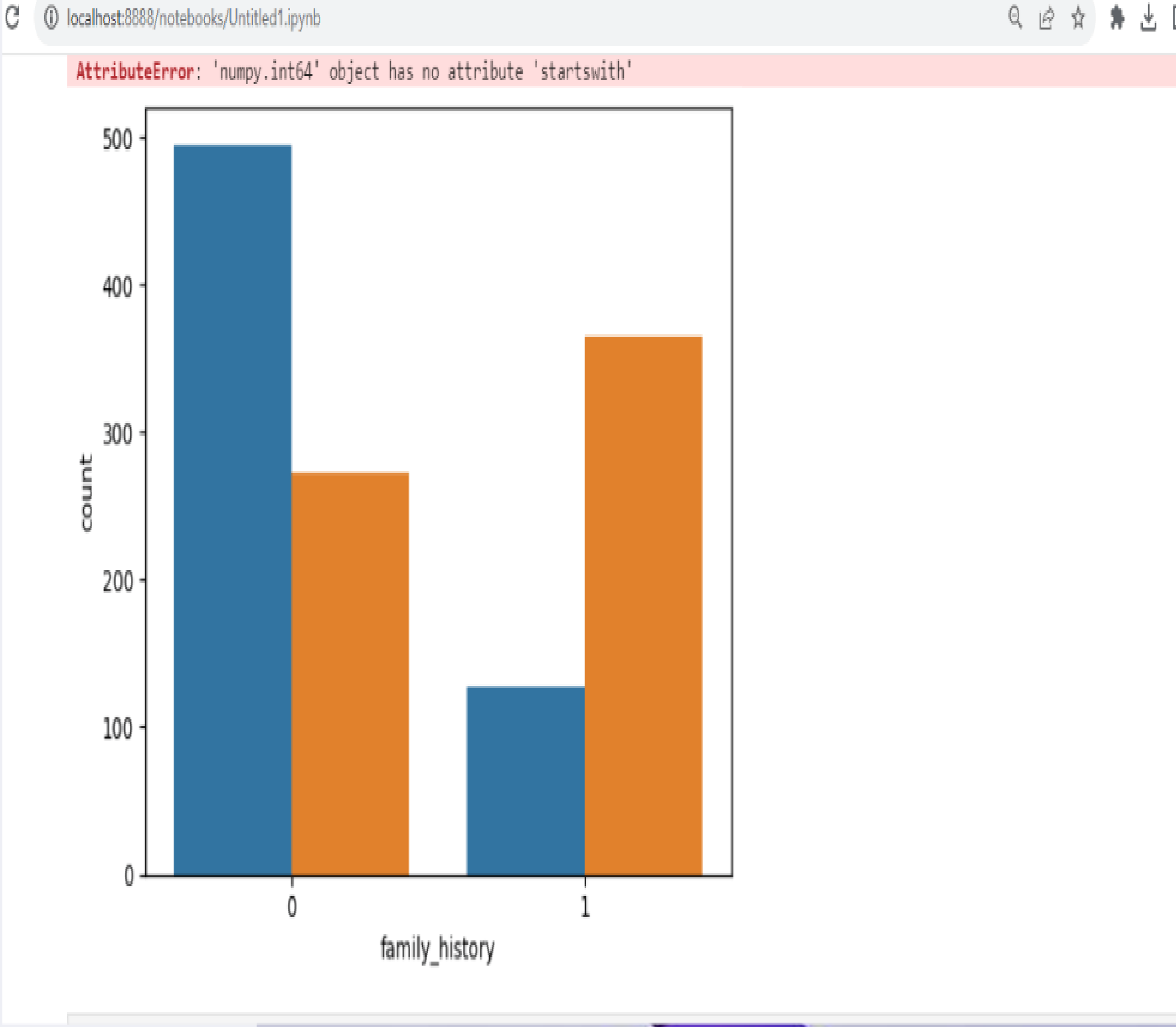
Analysing State Vs Age



Improving Public Health Awareness

Our analysis of the mental health prediction dataset will shed light on areas where public health awareness can be further enhanced. We'll explore innovative approaches to make a lasting impact on society.

Insights which manifest improvement in public health awareness

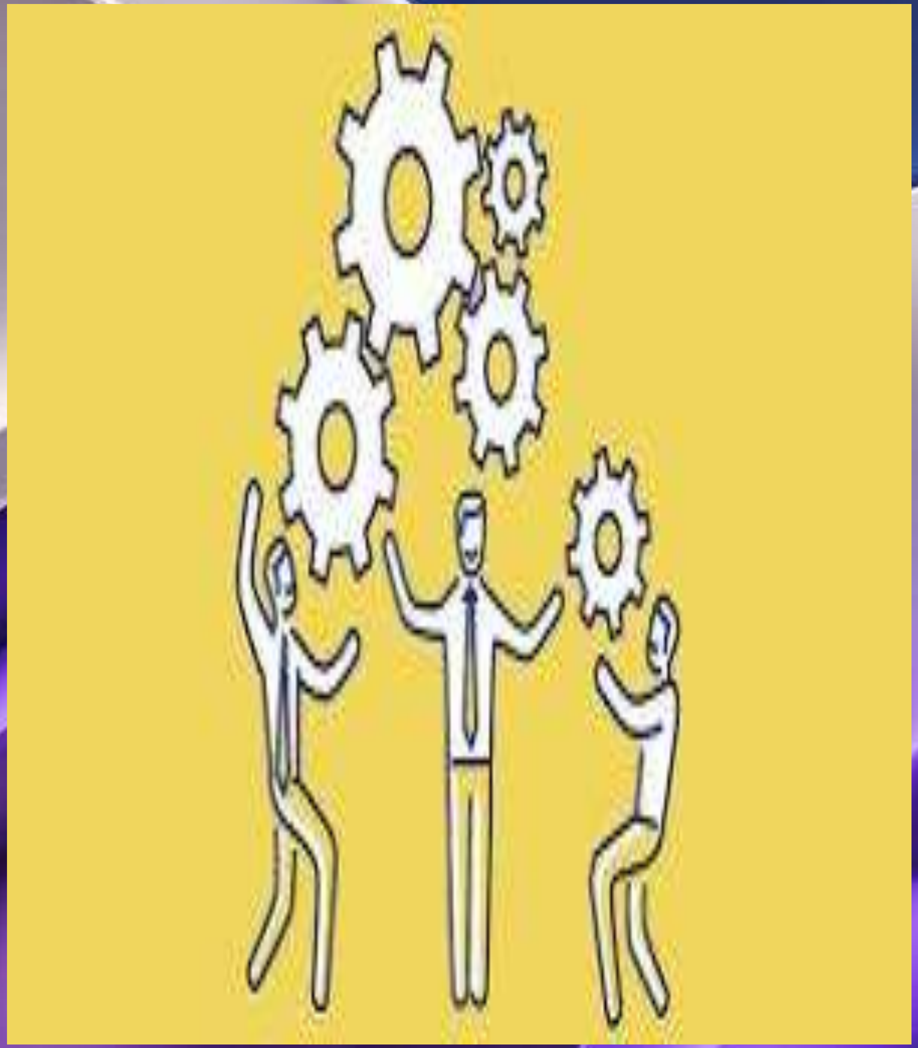


Accuracy of the Dataset

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.tree import DecisionTreeClassifier as DT

[40]: steps_rfc = [('Scaler', StandardScaler()),
                  ('clf', RFC(n_estimators = 40))]



clf_rfc = Pipeline(steps=steps_rfc)


clf_rfc.fit(X_train, y_train)

y_pred_rfc = clf_rfc.predict(X_test)
print('RFC accuracy: ', accuracy_score(y_true=y_test, y_pred=y_pred_rfc)*100)

RFC accuracy:  77.9552715654952l

[41]: steps_knn = [('scaler', StandardScaler()),
                  ('clf', KNN(n_neighbors = 5))]



clf_knn = Pipeline(steps=steps_knn)
```

The Importance of Public Health Awareness

Public health awareness plays a vital role in society, influencing behaviors, promoting well-being, and preventing diseases. Discover the significance and far-reaching impact of these campaigns.



As we wrap up our journey through the realm of public health awareness campaign analysis, reflect upon the profound impact these campaigns have on individuals and society as a whole. Let's continue championing public health awareness!