PUBLIC HEALTH AWARENESS CAMPAIGN ANALYSIS

Analyzing a public health awareness campaign involves specific objectives and data collection strategies aimed at evaluating the effectiveness of the campaign in raising awareness and potentially changing public health behaviors. Here's how you can define analysis objectives and collect campaign data for a public health awareness campaign analysis

DEFINE ANALYSIS OBJECTIVES:

Awareness Measurement:Measure the level of awareness before and after the campaign to determine if the campaign effectively increased awareness of the health issue.**Behavior Change:** Assess whether the campaign had any impact on public health behaviors, such as vaccination rates, healthy eating, or exercise habits.Audience Reach:Determine how many people were reached by the campaign, both online and offline.

Collect Campaign Data:

Surveys and Questionnaires: Create and distribute surveys or questionnaires to assess changes in awareness and behaviors among the target audience. Include questions related to campaign recognition and impact.Social Media Metrics: If the campaign leveraged social media, collect data on engagement metrics, such as likes, shares, comments, and the number of followers or subscribers.Web Analytics: If there's an associated website or landing page, collect data on website traffic, user behavior, and conversion rates.

Analyze Data:

Quantitative Analysis: Analyze quantitative data collected from surveys, social media, website analytics, and healthcare data to measure changes in awareness and behaviors. Qualitative Analysis:Analyze qualitative data from focus groups and interviews to gain deeper insights into how the campaign influenced people's perceptions and actions. Comparative Analysis Compare pre-campaign and post-campaign data to determine the impact of the campaign

Interpret Results:

Awareness and Behavior Change: Interpret the data to assess whether the campaign achieved its objectives in terms of awareness and behavior change. Engagement: Analyze engagement data to understand which campaign elements were most effective in capturing the audience's attention. Cost-Effectiveness: Evaluate whether the campaign was cost-effective in achieving its goals.

Report and Presentation:

Prepare a comprehensive report with data analysis, findings, and recommendationsPresent the findings to stakeholders, including public health officials, campaign sponsors, and the general public if applicable. Use visual aids and clear language to convey the results and their implications

CONCLUTION:

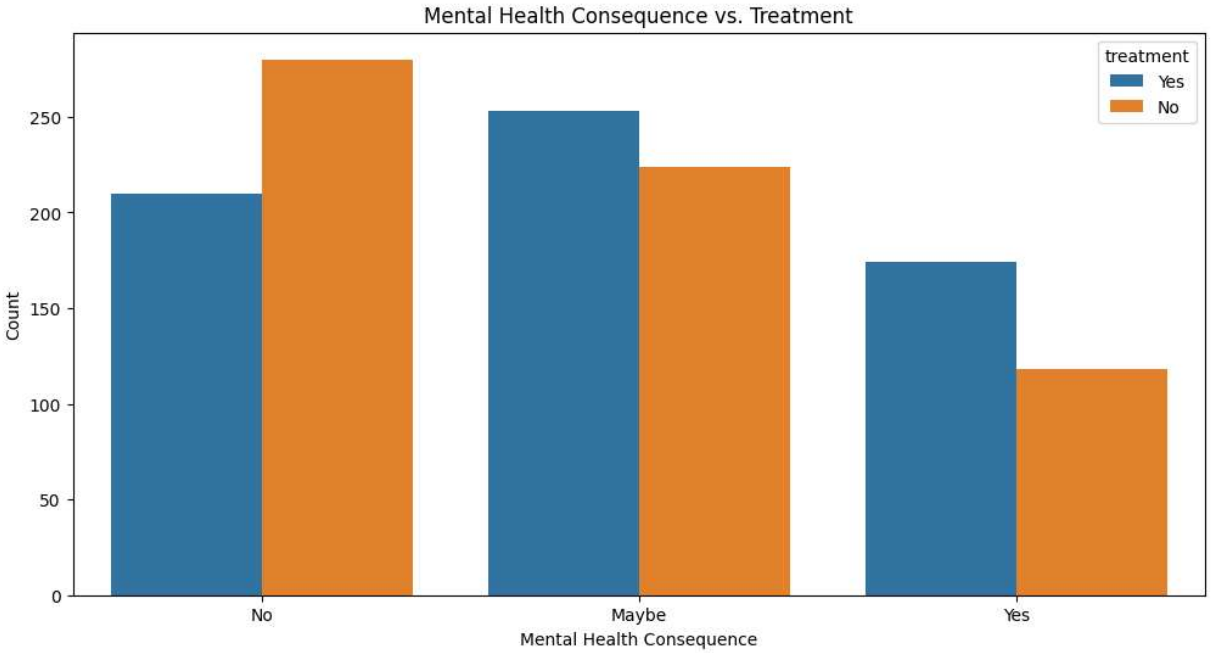
Remember that public health awareness campaigns are critical for promoting health and preventing diseases, so the analysis should provide actionable insights for improving future campaigns and better addressing public health issues.

In [176…

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('survey.csv')
print("Dataset Overview:")
print(data.head())
print(data.info())
plt.figure(figsize=(12, 6))
sns.countplot(data=data, x='mental_health_consequence', hue='treatment')
plt.title("Mental Health Consequence vs. Treatment")
plt.xlabel("Mental Health Consequence")
plt.ylabel("Count")
plt.show()
```

```
Dataset Overview:
             Timestamp  Age  Gender         Country state self_employed  \
0  2014-08-27 11:29:31   37  Female   United States    IL           NaN  \
1  2014-08-27 11:29:37   44       M   United States    IN           NaN
2  2014-08-27 11:29:44   32    Male          Canada   NaN           NaN
3  2014-08-27 11:29:46   31    Male  United Kingdom   NaN           NaN
4  2014-08-27 11:30:22   31    Male   United States    TX           NaN

  family_history treatment work_interfere    no_employees  ...
0             No       Yes          Often            6-25  ...  \
1             No        No         Rarely  More than 1000  ...
2             No        No         Rarely            6-25  ...
3            Yes       Yes          Often          26-100  ...
4             No        No          Never         100-500  ...

               leave mental_health_consequence phys_health_consequence
0       Somewhat easy                       No                      No  \
1          Don't know                    Maybe                      No
2  Somewhat difficult                       No                      No
3  Somewhat difficult                      Yes                     Yes
4          Don't know                       No                      No

      coworkers supervisor mental_health_interview phys_health_interview
0  Some of them        Yes                      No                Maybe  \
1           No         No                      No                   No
2          Yes        Yes                     Yes                  Yes
3  Some of them         No                   Maybe                Maybe
4  Some of them        Yes                     Yes                  Yes

  mental_vs_physical obs_consequence comments
0                Yes              No      NaN
1         Don't know              No      NaN
2                 No              No      NaN
3                 No             Yes      NaN
4         Don't know              No      NaN

[5 rows x 27 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Timestamp                  1259 non-null   object
 1   Age                        1259 non-null   int64
 2   Gender                     1259 non-null   object
 3   Country                    1259 non-null   object
 4   state                      744 non-null    object
 5   self_employed              1241 non-null   object
 6   family_history             1259 non-null   object
 7   treatment                  1259 non-null   object
 8   work_interfere             995 non-null    object
 9   no_employees               1259 non-null   object
 10  remote_work                1259 non-null   object
 11  tech_company               1259 non-null   object
 12  benefits                   1259 non-null   object
 13  care_options               1259 non-null   object
```

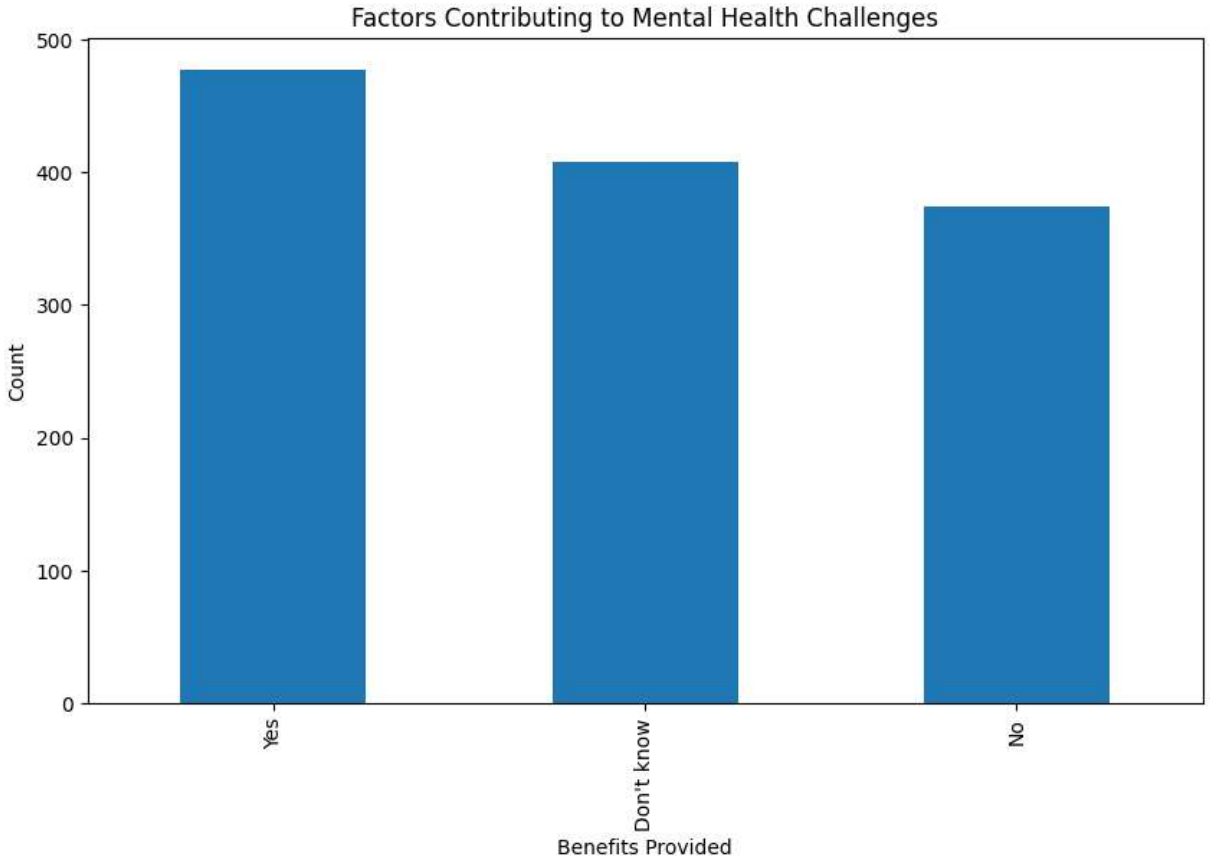
```
 14  wellness_program           1259 non-null   object
 15  seek_help                  1259 non-null   object
 16  anonymity                  1259 non-null   object
 17  leave                      1259 non-null   object
 18  mental_health_consequence  1259 non-null   object
 19  phys_health_consequence    1259 non-null   object
 20  coworkers                  1259 non-null   object
 21  supervisor                 1259 non-null   object
 22  mental_health_interview    1259 non-null   object
 23  phys_health_interview      1259 non-null   object
 24  mental_vs_physical         1259 non-null   object
 25  obs_consequence            1259 non-null   object
 26  comments                   164 non-null    object
dtypes: int64(1), object(26)
memory usage: 265.7+ KB
None
```



In [177…
```
mental_health_prevalence = df["treatment"].value_counts()
print("Prevalence of Mental Health Issues:")
print(mental_health_prevalence)
```

```
Prevalence of Mental Health Issues:
treatment
1    637
0    622
Name: count, dtype: int64
```

In [178…
```
plt.figure(figsize=(10, 6))
data['benefits'].value_counts().plot(kind='bar')
plt.title("Factors Contributing to Mental Health Challenges")
plt.xlabel("Benefits Provided")
plt.ylabel("Count")
plt.show()
```



In [179…

```
data.isnull().sum()
```

Out[179]:
```
Timestamp                      0
Age                            0
Gender                         0
Country                        0
state                        515
self_employed                 18
family_history                 0
treatment                      0
work_interfere               264
no_employees                   0
remote_work                    0
tech_company                   0
benefits                       0
care_options                   0
wellness_program               0
seek_help                      0
anonymity                      0
leave                          0
mental_health_consequence      0
phys_health_consequence        0
coworkers                      0
supervisor                     0
mental_health_interview        0
phys_health_interview          0
mental_vs_physical             0
obs_consequence                0
comments                    1095
dtype: int64
```

Encoding

In [180…
```
# Import label encoder
from sklearn import preprocessing

# label_encoder object knows
# how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
df['treatment']=label_encoder.fit_transform(data['treatment'])
```

In [181…
```
data['treatment'].value_counts()
```

Out[181]:
```
treatment
Yes    637
No     622
Name: count, dtype: int64
```

In [182…
```
data['Gender']=label_encoder.fit_transform(data['Gender'])
```

In [183…
```
data['Gender'].value_counts()
```

Out[183]:
```
Gender
20    615
39    206
10    121
16    116
35     62
8      38
37     34
33     15
18      4
21      3
30      3
5       2
25      2
13      2
11      2
28      1
41      1
40      1
27      1
31      1
45      1
12      1
17      1
32      1
0       1
24      1
44      1
34      1
6       1
15      1
7       1
1       1
3       1
23      1
38      1
29      1
4       1
47      1
19      1
22      1
46      1
42      1
9       1
48      1
26      1
2       1
36      1
14      1
43      1
Name: count, dtype: int64
```

In [184…
```
data['mental_health_consequence']=label_encoder.fit_transform(data['mental_health_c
```

In [185…
```
data['mental_health_consequence'].value_counts()
```

Out[185]:
```
mental_health_consequence
1    490
0    477
2    292
Name: count, dtype: int64
```

In [186…
```
data['family_history'].value_counts()
```

Out[186]:
```
family_history
No     767
Yes    492
Name: count, dtype: int64
```

In [187…
```
data['family_history']=label_encoder.fit_transform(data['family_history'])
```

In [188…
```
data['family_history'].value_counts()
```

Out[188]:
```
family_history
0    767
1    492
Name: count, dtype: int64
```

In [189…
```
data['phys_health_consequence'].value_counts()
```

Out[189]:
```
phys_health_consequence
No       925
Maybe    273
Yes       61
Name: count, dtype: int64
```

In [190…
```
data['phys_health_consequence']=label_encoder.fit_transform(data['phys_health_conse
```

In [191…
```
data['phys_health_consequence'].value_counts()
```

Out[191]:
```
phys_health_consequence
1    925
0    273
2     61
Name: count, dtype: int64
```

In [192…
```
data['mental_health_consequence'].value_counts()
```

Out[192]:
```
mental_health_consequence
1    490
0    477
2    292
Name: count, dtype: int64
```

In [193…
```
data["mental_health_interview"].value_counts()
```

Out[193]:
```
mental_health_interview
No       1008
Maybe     207
Yes        44
Name: count, dtype: int64
```

In [194…
```
data['supervisor'].value_counts()
```

Out[194]:
```
supervisor
Yes             516
No              393
Some of them    350
Name: count, dtype: int64
```

In [195…
```
data['supervisor']=label_encoder.fit_transform(data['supervisor'])
```

In [196…
```
data['supervisor'].value_counts()
```

Out[196]:
```
supervisor
2    516
0    393
1    350
Name: count, dtype: int64
```

In [197…
```
data['mental_health_interview'].value_counts()
```

Out[197]:
```
mental_health_interview
No       1008
Maybe     207
Yes        44
Name: count, dtype: int64
```

In [198…
```
data['mental_health_interview'].value_counts()
```

Out[198]:
```
mental_health_interview
No       1008
Maybe     207
Yes        44
Name: count, dtype: int64
```

In [199…
```
data['mental_health_interview']=label_encoder.fit_transform(data['mental_health_int
```

In [200…
```
data['mental_health_interview'].value_counts()
```

Out[200]:
```
mental_health_interview
1    1008
0     207
2      44
Name: count, dtype: int64
```

In [201…
```
data['phys_health_consequence'].value_counts()
```

Out[201]:
```
phys_health_consequence
1    925
0    273
2     61
Name: count, dtype: int64
```

In [202…
```
data['phys_health_consequence']=label_encoder.fit_transform(data['phys_health_conse
```

In [203…
```
data['mental_vs_physical'].value_counts()
```

Out[203]:
```
mental_vs_physical
Don't know    576
Yes           343
No            340
Name: count, dtype: int64
```

In [204…
```
data['mental_vs_physical']=label_encoder.fit_transform(data['mental_vs_physical'])
```

In [205…
```
data['mental_vs_physical'].value_counts()
```

Out[205]:
```
mental_vs_physical
0    576
2    343
1    340
Name: count, dtype: int64
```

In [206…
```
data['obs_consequence'].value_counts()
```

Out[206]:
```
obs_consequence
No     1075
Yes     184
Name: count, dtype: int64
```

In [207…
```
data['obs_consequence']=label_encoder.fit_transform(data['obs_consequence'])
```

In [208…
```
data['obs_consequence'].value_counts()
```

Out[208]:
```
obs_consequence
0    1075
1     184
Name: count, dtype: int64
```

In [209…
```
data['treatment'].value_counts()
```

Out[209]:
```
treatment
Yes    637
No     622
Name: count, dtype: int64
```

In [210…
```
data['treatment']=label_encoder.fit_transform(data['treatment'])
```

In [211…
```
data['treatment'].value_counts()
```

Out[211]:
```
treatment
1    637
0    622
Name: count, dtype: int64
```

In [212…
```
data['remote_work'].value_counts()
```

Out[212]:
```
remote_work
No     883
Yes    376
Name: count, dtype: int64
```

In [213…
```
data['remote_work']=label_encoder.fit_transform(data['remote_work'])
```

In [214…
```
data['remote_work'].value_counts()
```

Out[214]:
```
remote_work
0    883
1    376
Name: count, dtype: int64
```

In [215…
```
data['tech_company'].value_counts()
```

Out[215]:
```
tech_company
Yes    1031
No      228
Name: count, dtype: int64
```

In [216…
```
data['tech_company']=label_encoder.fit_transform(data['tech_company'])
```

In [217…
```
data['tech_company'].value_counts()
```

Out[217]:
```
tech_company
1    1031
0     228
Name: count, dtype: int64
```

In [218…
```
data['benefits'].value_counts()
```

Out[218]:
```
benefits
Yes           477
Don't know    408
No            374
Name: count, dtype: int64
```

In [219…
```
data['benefits']=label_encoder.fit_transform(data['benefits'])
```

In [220…
```
data['benefits'].value_counts()
```

Out[220]:
```
benefits
2    477
0    408
1    374
Name: count, dtype: int64
```

In [221…
```
data['coworkers'].value_counts()
```

Out[221]:
```
coworkers
Some of them    774
No              260
Yes             225
Name: count, dtype: int64
```

In [222…
```
data['coworkers']=label_encoder.fit_transform(data['coworkers'])
```

In [223…
```
data['coworkers'].value_counts()
```

Out[223]:
```
coworkers
1    774
0    260
2    225
Name: count, dtype: int64
```

In [224…
```
data['phys_health_interview'].value_counts()
```

Out[224]:
```
phys_health_interview
Maybe    557
No       500
Yes      202
Name: count, dtype: int64
```

In [225…
```
data['phys_health_interview']=label_encoder.fit_transform(data['phys_health_intervi
```

In [226…
```
data['phys_health_interview'].value_counts()
```

Out[226]:
```
phys_health_interview
0    557
1    500
2    202
Name: count, dtype: int64
```

In [227…
```
data =data.drop(['Country','comments','state','Timestamp'],axis=1)
```

In [228…
```
data.dropna(axis=0,inplace=True)
```

In [229…
```
data
```

Out[229]:

	Age	Gender	self_employed	family_history	treatment	work_interfere	no_employee
18	46	39	Yes	1	0	Sometimes	1-
20	29	20	No	1	1	Sometimes	100-50
21	31	39	Yes	0	0	Never	1-
22	46	20	No	0	1	Often	26-10
23	41	20	No	0	1	Never	More tha 100
...	...	...	...	...	...	...	
1252	29	39	No	1	1	Sometimes	100-50
1253	36	20	No	1	0	Rarely	More tha 100
1255	32	20	No	1	1	Often	26-10
1256	34	39	No	1	1	Sometimes	More tha 100
1258	25	20	No	1	1	Sometimes	26-10

977 rows × 23 columns

In [230…
```
data['self_employed'].value_counts()
```

Out[230]:
```
self_employed
No     852
Yes    125
Name: count, dtype: int64
```

In [231…
```
data['self_employed']=label_encoder.fit_transform(data['self_employed'])
```

In [232…
```
data['self_employed'].value_counts()
```

Out[232]:
```
self_employed
0    852
1    125
Name: count, dtype: int64
```

In [233…
```
data['work_interfere'].value_counts()
```

Out[233]:
```
work_interfere
Sometimes    458
Never        207
Rarely       170
Often        142
Name: count, dtype: int64
```

In [234…
```
data['work_interfere']=label_encoder.fit_transform(data['work_interfere'])
```

In [235…
```
data['work_interfere'].value_counts()
```

Out[235]:
```
work_interfere
3    458
0    207
2    170
1    142
Name: count, dtype: int64
```

In [236…
```
data['no_employees'].value_counts()
```

Out[236]:
```
no_employees
26-100            224
More than 1000    220
6-25              213
1-5               140
100-500           137
500-1000           43
Name: count, dtype: int64
```

In [237…
```
data['no_employees']=label_encoder.fit_transform(data['no_employees'])
```

In [238…
```
data['no_employees'].value_counts()
```

Out[238]:
```
no_employees
2    224
5    220
4    213
0    140
1    137
3     43
Name: count, dtype: int64
```

In [239…
```
data['anonymity'].value_counts()
```

Out[239]:
```
anonymity
Don't know    624
Yes           299
No             54
Name: count, dtype: int64
```

In [240…
```
data['anonymity']=label_encoder.fit_transform(data['anonymity'])
```

In [241…
```
data['anonymity'].value_counts()
```

Out[241]:
```
anonymity
0    624
2    299
1     54
Name: count, dtype: int64
```

In [242…
```
data['leave'].value_counts()
```

Out[242]:
```
leave
Don't know            413
Somewhat easy         210
Very easy             156
Somewhat difficult    108
Very difficult         90
Name: count, dtype: int64
```

In [243…
```
data['leave']=label_encoder.fit_transform(data['leave'])
```

In [244…
```
data['leave'].value_counts()
```

Out[244]:
```
leave
0    413
2    210
4    156
1    108
3     90
Name: count, dtype: int64
```

In [245…
```
data
```

Out[245]:

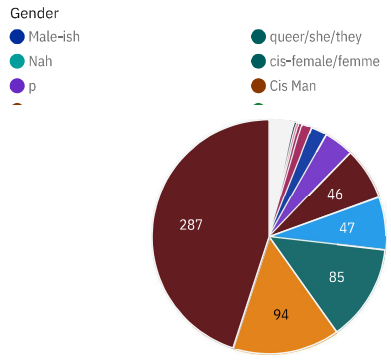
	Age	Gender	self_employed	family_history	treatment	work_interfere	no_employee
18	46	39	1	1	0	3	
20	29	20	0	1	1	3	
21	31	39	1	0	0	0	
22	46	20	0	0	1	1	
23	41	20	0	0	1	0	
…	…	…	…	…	…	…	
1252	29	39	0	1	1	3	
1253	36	20	0	1	0	2	
1255	32	20	0	1	1	1	
1256	34	39	0	1	1	3	
1258	25	20	0	1	1	3	

977 rows × 23 columns

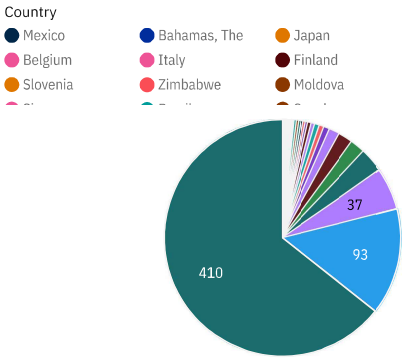
In [ ]:

Tab 1

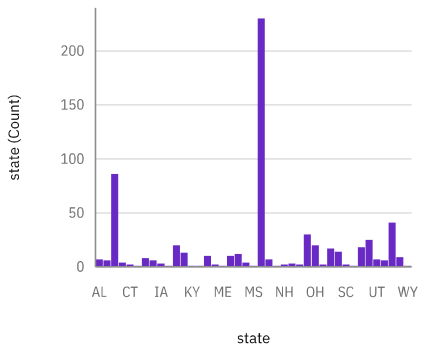
Gender by Gender



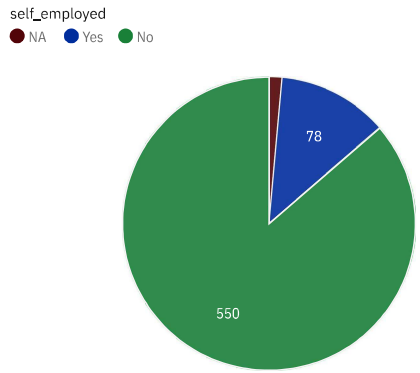
1 Country by Country



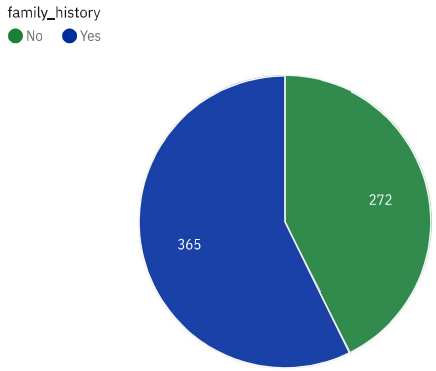
2 state by state 3



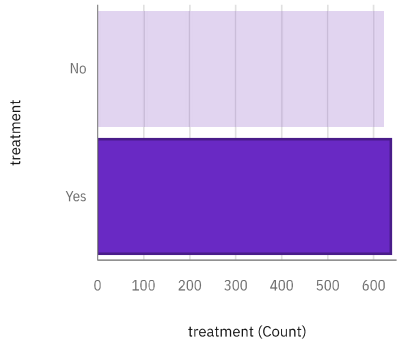
self_employed by self_employed 4



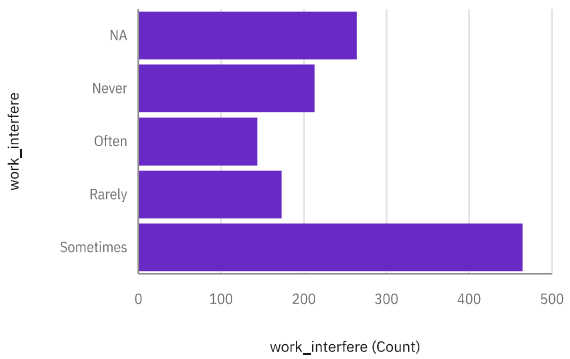
family_history by family_history 5



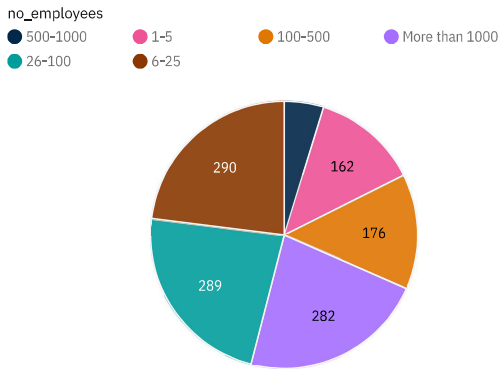
treatment by treatment 6



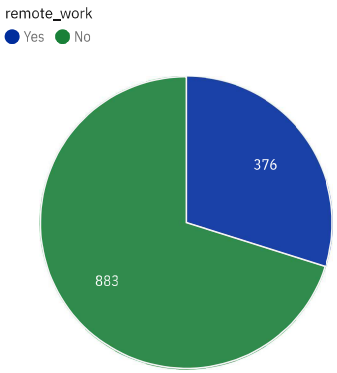
work_interfere by work_interfere 7



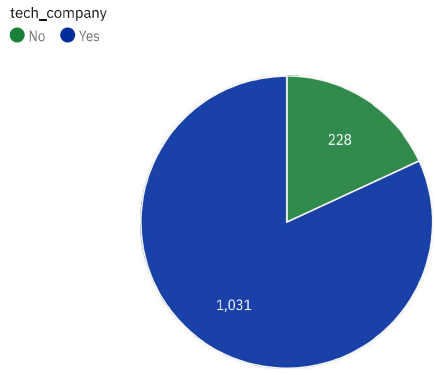
no_employees by no_employees 8



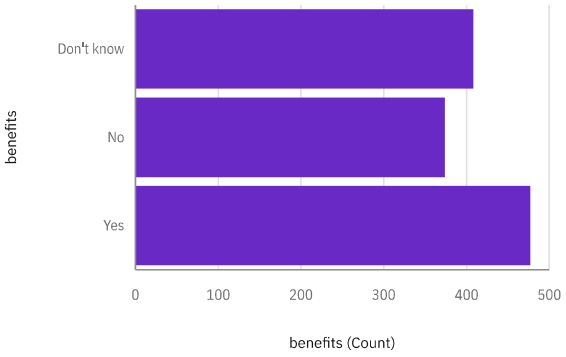
remote_work by remote_work 9



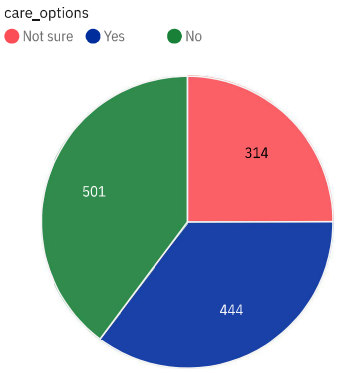
tech_company by tech_company 10



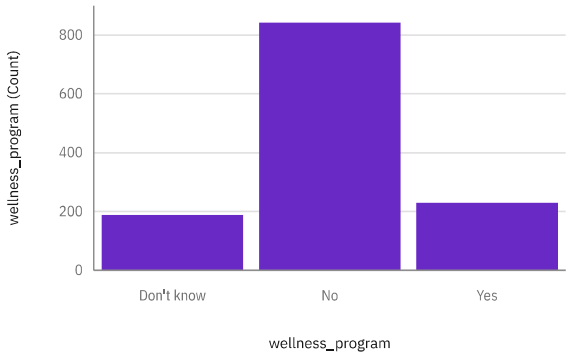
benefits by benefits 11



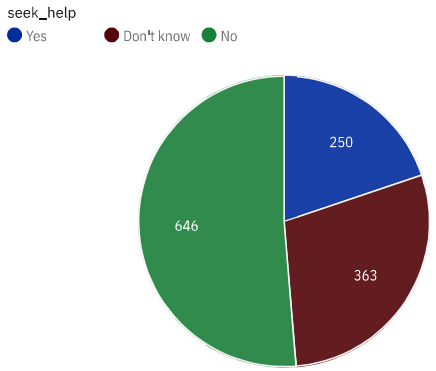
care_options by care_options 12



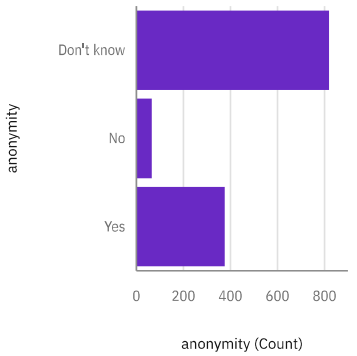
wellness_program by wellness_program 13



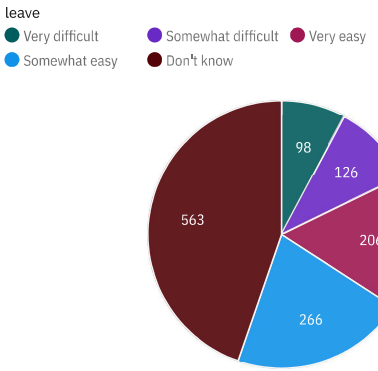
seek_help by seek_help 14



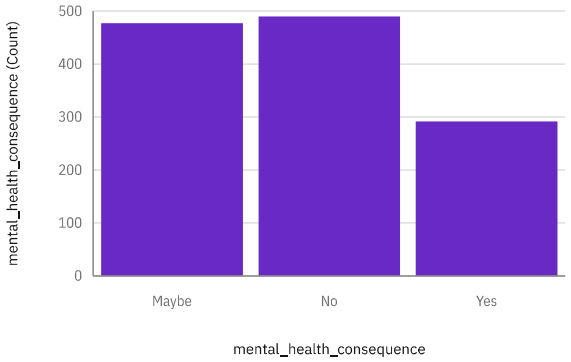
anonymity by anonymity 15



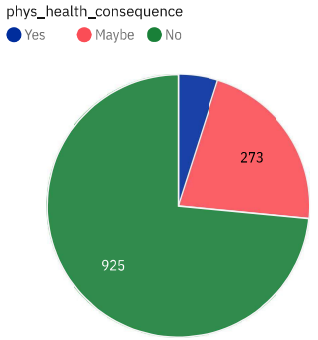
leave by leave 16



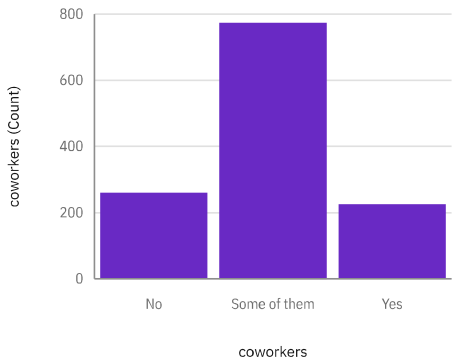
mental_health_consequence by mental_health_consequence 17



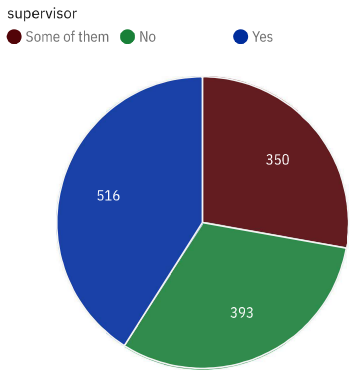
phys_health_consequence by phys_health_consequence 18



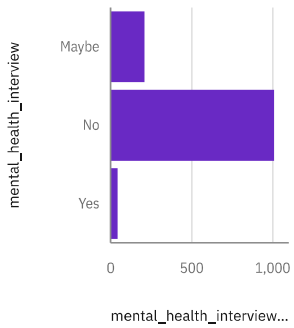
coworkers by coworkers 19



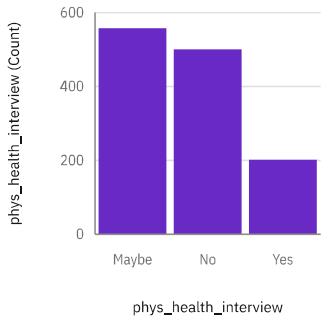
supervisor by supervisor 20



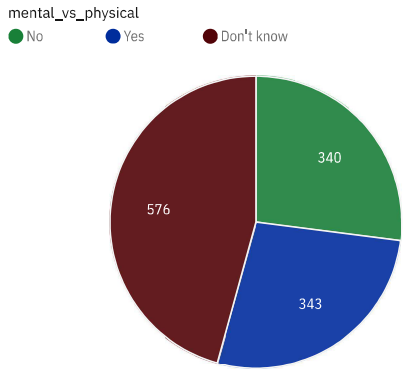
mental_health_interview by mental_health_interview 21



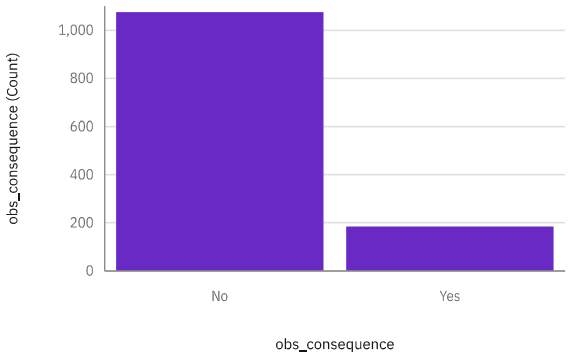
phys_health_interview by phys_health_interview 22



mental_vs_physical by mental_vs_physical 23



obs_consequence by obs_consequence 24



Filter(s) applied to the visualization(s):

Widget 1
Gender Includes: A little about you, Agender, All, Androgyne, Cis Female, Cis Male, Cis Man, Enby, F, Femake, Female, Female (cis), Female (trans), Genderqueer, Guy (-ish) ^_^, M, Mail, Make, Mal, Male, Male (CIS), Male-ish, Malr, Man, Nah, Neuter, Trans woman, Trans-female, Woman, cis male, cis-female/femme, f, femail, female, fluid, m, maile, male, male leaning androgynous, msle, non-binary, ostensibly male, unsure what that really means, p, queer, queer/she/they, something kinda male?, woman
treatment Includes: Yes

Widget 2
Country Includes: Australia, Austria, Bahamas, The, Belgium, Bosnia and Herzegovina, Brazil, Bulgaria, China, Canada, Colombia, Costa Rica, Croatia, Czech Republic, Denmark, Finland, France, Georgia, Germany, Greece, Hungary, India, Ireland, Israel, Italy, Japan, Latvia, Mexico, Moldova, Netherlands, New Zealand, Nigeria, Norway, Philippines, Poland, Portugal, Romania, Russia, Singapore, Slovenia, South Africa, Spain, Sweden, Switzerland, Thailand, United Kingdom, Uruguay, Zimbabwe, United States
treatment Includes: Yes

Widget 3
state Includes: AL, AZ, CA, CO, CT, DC, FL, GA, IA, ID, IL, IN, KS, KY, LA, MA, MD, ME, MI, MN, MO, MS, NA, NC, NE, NH, NJ, NM, NV, NY, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VA, VT, WA, WI, WV, WY
treatment Includes: Yes

Widget 4
self_employed Includes: NA, No, Yes
treatment Includes: Yes

Widget 5
family_history Includes: No, Yes
treatment Includes: Yes

Widget 6
treatment Includes: No, Yes
treatment Includes: Yes

Widget 7
work_interfere Includes: NA, Never, Often, Rarely, Sometimes

Widget 8
no_employees Includes: 1-5, 100-500, 26-100, 500-1000, 6-25, More than 1000

Widget 9
remote_work Includes: No, Yes

Widget 10
tech_company Includes: No, Yes

Widget 11
benefits Includes: Don't know, No, Yes

Widget 12
care_options Includes: Not sure, No, Yes

Widget 13

wellness_program Includes: Don't know, No, Yes

Widget 14
seek_help Includes: Don't know, No, Yes

Widget 15
anonymity Includes: Don't know, No, Yes

Widget 16
leave Includes: Don't know, Somewhat difficult, Somewhat easy, Very difficult, Very easy

Widget 17
mental_health_consequence Includes: Maybe, No, Yes

Widget 18
phys_health_consequence Includes: No, Maybe, Yes

Widget 19
coworkers Includes: No, Some of them, Yes

Widget 20
supervisor Includes: No, Some of them, Yes

Widget 21
mental_health_interview Includes: Maybe, No, Yes

Widget 22
phys_health_interview Includes: Maybe, No, Yes

Widget 23
mental_vs_physical Includes: Don't know, No, Yes

Widget 24
obs_consequence Includes: No, Yes