Mental Health in Tech -

Project Type - Exploratory Data Analysis(EDA)

Contribution - Individual

Team Member 1 - Himanshu Arya

Project Summary -

This project explores the **Mental Health in Tech Survey (2014)** — a globally recognized dataset focused on assessing mental health prevalence, stigma, and employer support structures within the technology sector. With rising awareness around mental health in the workplace, particularly in high-pressure environments like tech companies and startups, this dataset offers an opportunity to examine real employee experiences and perceptions across regions, company sizes, and job structures.

The dataset contains responses to a detailed questionnaire that covered:

- Mental health history and treatment-seeking behavior
- Family history of mental illness
- Whether respondents work in tech or non-tech companies
- Presence of employer-provided mental health benefits
- Perceived openness to discuss mental health with coworkers and supervisors
- Ease of taking mental health leave
- Observed or experienced consequences for discussing mental health
- Comparison of attitudes toward physical vs mental health

I performed an in-depth **Exploratory Data Analysis (EDA)** using Python (Pandas, Seaborn, Matplotlib) to uncover insights and answer both **broad and specific questions**, such as:

- How do mental health concerns interfere with work productivity?
- Which factors predict whether someone seeks treatment?
- How comfortable are employees discussing mental health with their peers and supervisors?
- Does company size, remote work, or access to benefits influence mental health outcomes?
- Are there visible differences in mental health openness across countries?
- Is mental health taken as seriously as physical health in the workplace?

Throughout the analysis, I also paid close attention to data cleaning (especially age and gender normalization), feature engineering (company size grouping, binary encoding), and visual storytelling to communicate findings clearly and effectively.

This project is not just about code — it's about connecting data to real-world impact. The insights derived here can guide employers, HR departments, mental health advocates, and policy designers to **build more inclusive, transparent, and supportive work environments**.

By analyzing a real-world, socially impactful dataset, this project demonstrates the practical application of data analytics skills — from raw data to business-aligned insights — and showcases the role of data-driven decision-making in shaping healthier workplace cultures.

GitHub Link -

https://github.com/HiAr21/Mental-Health-in-Tech-EDA

Problem Statement

To explore how individuals in the tech industry perceive and manage mental health issues at work. We will analyze factors such as treatment-seeking behavior, workplace support, gender distribution, and openness to discuss mental health.

Define Your Business Objective?

The primary goal of this project is to help employers, HR teams, and policy designers in the tech industry better understand:

1. **Prevalence** of mental health challenges among employees
2. **Barriers** to seeking help (stigma, fear, lack of support)
3. **Cultural and structural factors** that influence treatment, disclosure, and trust
4. **Workplace readiness** — including benefits, openness, manager support, and anonymity

Ultimately, the objective is to provide data-driven recommendations to **improve mental health support systems**, **reduce stigma**, and create a workplace culture that prioritizes both physical and mental well-being.

General Guidelines : -

↳ 1 cell hidden

Let's Begin !

1. Know Your Data

Import Libraries

```
# Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

Dataset Loading

```
# Load Dataset
from google.colab import files
uploaded = files.upload()
```

Choose Files survey.csv

- **survey.csv**(text/csv) - 241306 bytes, last modified: 6/27/2025 - 100% done

```
Saving survey.csv to survey (2).csv
```

Dataset First View

```
# Dataset
df = pd.read_csv('survey.csv')

# Dataset First Look
df.head()
```

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	...	leave	mental_health_con
0	8/27/2014 11:29	37	Female	United States	IL	NaN	No	Yes	Often	25-Jun	...	Somewhat easy	
1	8/27/2014 11:29	44	M	United States	IN	NaN	No	No	Rarely	More than 1000	...	Don't know	
2	8/27/2014 11:29	32	Male	Canada	NaN	NaN	No	No	Rarely	25-Jun	...	Somewhat difficult	
3	8/27/2014 11:29	31	Male	United Kingdom	NaN	NaN	Yes	Yes	Often	26-100	...	Somewhat difficult	
4	8/27/2014 11:30	31	Male	United States	TX	NaN	No	No	Never	100-500	...	Don't know	

5 rows × 27 columns

```
# Dataset Rows & Columns count
(rows,cols) = df.shape
(rows,cols)
```

```
(1259, 27)
```

Dataset Information

```
# Dataset Info
df.info()
```

```
<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
```

Missing Values/Null Values

```
# Missing Values/Null Values Count
df.isnull().sum().sort_values(ascending=False).head(10)
```

	0
comments	1095
state	515
work_interfere	264
self_employed	18
Gender	0
Timestamp	0
Age	0
family_history	0
treatment	0
no_employees	0

dtype: int64

```
# Fill key categorical blanks
df['self_employed'] = df['self_employed'].fillna('No')
df['work_interfere'] = df['work_interfere'].fillna("Don't know")
```

2. Understanding Your Variables

```
# Dataset Columns
df.columns
```

```
Index(['Timestamp', 'Age', 'Gender', 'Country', 'state', 'self_employed',
       'family_history', 'treatment', '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', 'comments'],
      dtype='object')
```

```
# Dataset Describe
df.describe()
```

	Age
count	1.259000e+03
mean	7.942815e+07
std	2.818299e+09
min	-1.726000e+03
25%	2.700000e+01
50%	3.100000e+01
75%	3.600000e+01
max	1.000000e+11

Check Unique Values for each variable.

```
# Check Unique Values for each variable.
df.nunique()
```

	0
Timestamp	884
Age	53
Gender	46
Country	48
state	45
self_employed	2
family_history	2
treatment	2
work_interfere	5
no_employees	6
remote_work	2
tech_company	2
benefits	3
care_options	3
wellness_program	3
seek_help	3
anonymity	3
leave	5
mental_health_consequence	3
phys_health_consequence	3
coworkers	3
supervisor	3
mental_health_interview	3
phys_health_interview	3
mental_vs_physical	3
obs_consequence	2
comments	160

dtype: int64

3. ***Data Wrangling***

Data Wrangling Code

```
# Drop invalid ages
df = df[(df['Age']>=18) & (df['Age']<100)]
```

```
df['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, 53, 61, 72])
```

```
df['Gender'].value_counts()
```

Gender	count
Male	612
male	206
Female	122
M	116
female	62
F	38
m	34
f	15
Make	4
Male	3
Woman	3
Female	2
Female (trans)	2
Man	2
Cis Male	2
something kinda male?	1
Cis Female	1
Trans-female	1
Male-ish	1
woman	1
non-binary	1
Enby	1
Nah	1
fluid	1
queer/she/they	1
Male (CIS)	1
Mal	1
Agender	1
Androgyne	1
Genderqueer	1
male leaning androgynous	1
cis-female/femme	1
Trans woman	1
msle	1
Neuter	1
queer	1
Female (cis)	1
Mail	1
cis male	1
Malr	1
femail	1
Cis Man	1
ostensibly male, unsure what that really means	1

dtype: int64

```
# Clean Gender Column
def clean_gender(g):
  g=str(g).strip().lower()
  if 'female' in g or g in ['f','woman']:
    return 'Female'
  elif 'male' in g or g in ['m','man']:
    return 'Male'
  else:
    return 'Other'


df.loc[:, 'Gender'] = df['Gender'].fillna('Other').apply(clean_gender)
```

```
df['Gender'].value_counts()
```

Gender	count
Male	981
Female	249
Other	21

dtype: int64

```
# Datatypes
df['Timestamp'] = pd.to_datetime(df['Timestamp'])
```

```
/tmp/ipython-input-135-1846066471.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Timestamp'] = pd.to_datetime(df['Timestamp'])
```

```
# Drop irrelevant columns
df_country = df.copy()
df_country.drop(columns=['Timestamp','comments','state'],inplace=True)
```

```
# Number of employees column
df_country['no_employees'].value_counts()
```

no_employees	count
25-Jun	289
26-100	288
More than 1000	281
100-500	175
5-Jan	158
500-1000	60

dtype: int64

```
# Simplify company size
def clean_no_employees(x):
    if x in ['5-Jan']:
        return '1-5'
    elif x in ['25-Jun']:
        return '6-25'
    else:
        return x

df_country['no_employees'] = df_country['no_employees'].apply(clean_no_employees)
```

```
# Simplify company size
def simplify_company_size(x):
    if x in ['1-5', '6-25']:
        return 'Small'
    elif x in ['26-100', '100-500']:
        return 'Medium'
    else:
        return 'Large'

df_country['company_size'] = df_country['no_employees'].apply(simplify_company_size)
```

```
df_country.head()
```

	Age	Gender	Country	self_employed	family_history	treatment	work_interfere	no_employees	remote_work	tech_company	...	leave	mental_h
0	37	Female	United States	No	No	Yes	Often	6-25	No	Yes	...	Somewhat easy	
1	44	Male	United States	No	No	No	Rarely	More than 1000	No	No	...	Don't know	
2	32	Male	Canada	No	No	No	Rarely	6-25	No	Yes	...	Somewhat difficult	
3	31	Male	United Kingdom	No	Yes	Yes	Often	26-100	No	Yes	...	Somewhat difficult	
4	31	Male	United States	No	No	No	Never	100-500	Yes	Yes	...	Don't know	

5 rows × 25 columns

```
df_country.columns
```

```
Index(['Age', 'Gender', 'Country', 'self_employed', 'family_history',
       'treatment', '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', 'company_size'],
      dtype='object')
```

4. Data Vizualization, Storytelling & Experimenting with charts : Understand the relationships between variables

```
df_country.nunique()
```

	0
Age	45
Gender	3
Country	46
self_employed	2
family_history	2
treatment	2
work_interfere	5
no_employees	6
remote_work	2
tech_company	2
benefits	3
care_options	3
wellness_program	3
seek_help	3
anonymity	3
leave	5
mental_health_consequence	3
phys_health_consequence	3
coworkers	3
supervisor	3
mental_health_interview	3
phys_health_interview	3
mental_vs_physical	3
obs_consequence	2
company_size	3

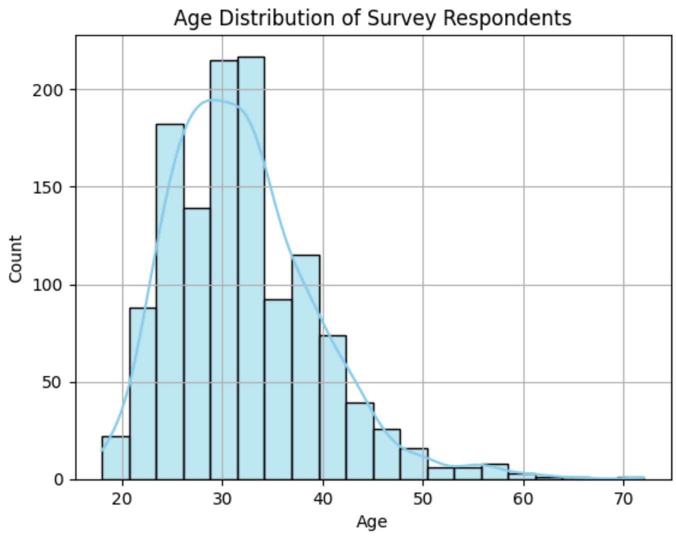
dtype: int64

1.Demographic Analysis

Survey Response Analysis

Chart - 1 : Age Distribution

```
# 1 - Age Distribution
sns.histplot(df_country['Age'], bins=20, kde=True, color='skyblue')
plt.title('Age Distribution of Survey Respondents')
plt.xlabel('Age')
plt.ylabel('Count')
plt.grid(True)
plt.show()
```



Why this chart?
To understand the dominant age group among respondents and potential generational patterns in mental health behavior.

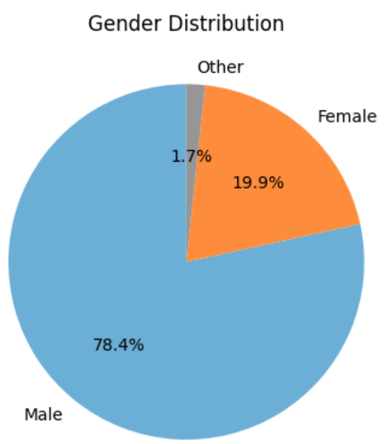
Insights:

- Most respondents are aged between **20–35**, the early-to-mid-career stage in tech.
- This age group is more likely to experience high pressure and rapid change.

Business Impact:
Wellness programs and messaging should be tailored for this core demographic.

Chart - 2 : Gender Breakdown

```
# 2 - Gender Breakdown

df['Gender'].value_counts().plot.pie(autopct='%1.1f%%', startangle=90, colors=['#6baed6','#fd8d3c','#969696'])
plt.title('Gender Distribution')
plt.ylabel('')
plt.show()
```



Why this chart?
To assess representation and evaluate gender-based mental health trends later.

Insights:

- Survey is heavily **male-dominated**, reflective of the tech industry at the time.
- Presence of "Other" gender category supports inclusive data collection.

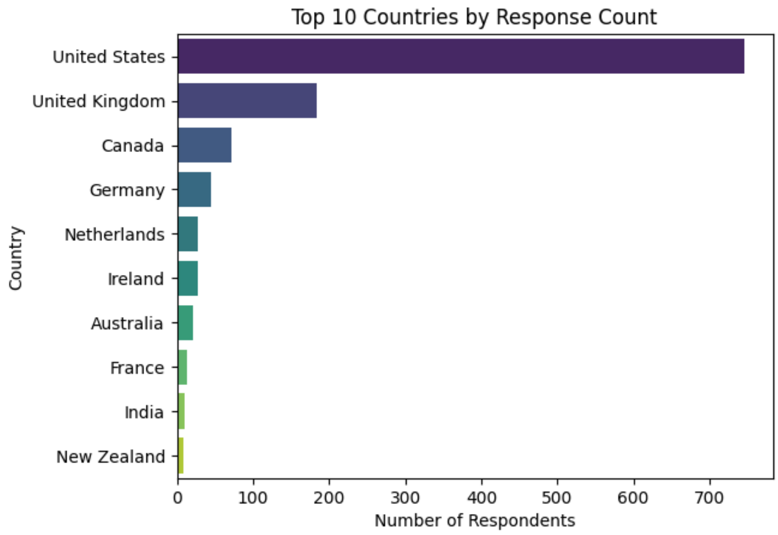
Business Impact:
Gender-based analysis is valid and could reveal unique patterns.

Chart - 3 : Country-Wise

```
# 3 - Country Wise response count

top_countries = df_country['Country'].value_counts().head(10)
```

```
sns.barplot(x=top_countries.values, y=top_countries.index, palette='viridis')
plt.title("Top 10 Countries by Response Count")
plt.xlabel("Number of Respondents")
plt.ylabel("Country")
plt.show()
```

```
/tmp/ipython-input-145-407372444.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for

  sns.barplot(x=top_countries.values, y=top_countries.index, palette='viridis')
```



Why this chart?
To see which countries dominate the sample and whether cross-country comparisons will be meaningful.

Insights:

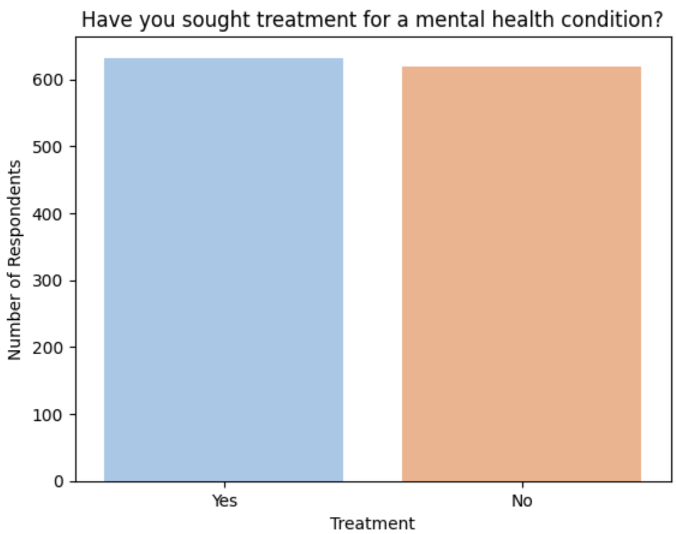
- Majority of responses are from the **United States**, followed by UK, Canada, and Germany.
- Country-level analysis will be reliable for these regions only.

2. Mental Health Experience

Chart - 4 : Treatment Count

```
# 4 - Treatment Count
sns.countplot(data=df_country, x='treatment', palette='pastel')
plt.title("Have you sought treatment for a mental health condition?")
plt.xlabel("Treatment")
plt.ylabel("Number of Respondents")
plt.show()
```

```
/tmp/ipython-input-146-152995800.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='treatment', palette='pastel')
```



Why this chart?
To assess how many people have actively sought help for mental health issues.

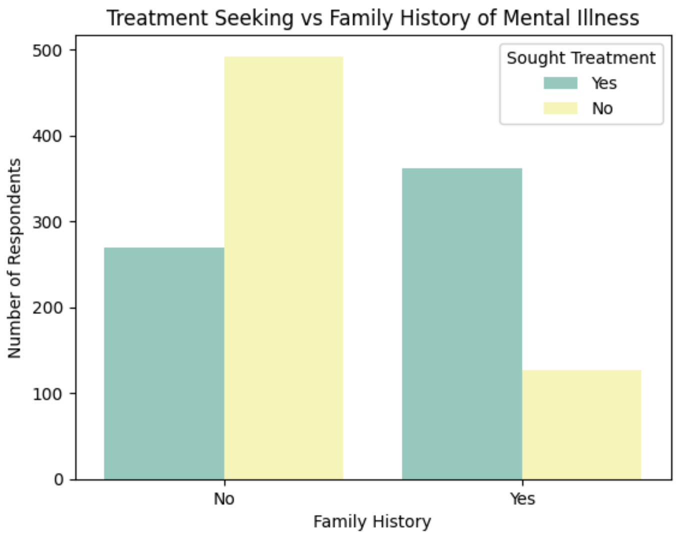
Insights:

- A significant number of respondents **have received treatment**, suggesting high awareness or prevalence.
- Still, many have **not sought treatment**, possibly due to stigma, cost, or employer barriers.

Business Impact:
Companies must address structural and cultural blocks to seeking help.

Chart - 5 : Treatment vs Family History

```
# 5 - Treatment vs Family history

sns.countplot(data=df_country, x='family_history', hue='treatment', palette='Set3')
plt.title("Treatment Seeking vs Family History of Mental Illness")
plt.xlabel("Family History")
plt.ylabel("Number of Respondents")
plt.legend(title="Sought Treatment")
plt.show()
```



Why this chart?
To explore if individuals with a family history are more likely to seek help.

Insights:

- Respondents with a **family history of mental illness** are more likely to seek treatment.
- This may reflect greater awareness or personal exposure to mental health systems.

Business Impact:
Mental health literacy and awareness campaigns can be targeted at those without family exposure.

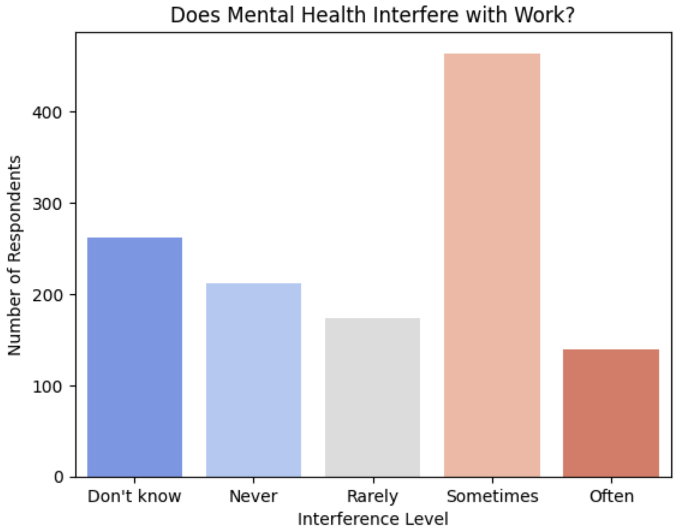
Chart - 6 : Work Interference

```
df_country['work_interfere'].unique()
```

```
array(['Often', 'Rarely', 'Never', 'Sometimes', "Don't know"],
      dtype=object)
```

```
# 6 - Work Interference Levels

sns.countplot(data=df_country, x='work_interfere', order=["Don't know",'Never', 'Rarely', 'Sometimes', 'Often'], palette='coolwarm')
plt.title("Does Mental Health Interfere with Work?")
plt.xlabel("Interference Level")
plt.ylabel("Number of Respondents")
plt.show()
```

```
/tmp/ipython-input-149-2356677880.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='work_interfere', order=["Don't know",'Never', 'Rarely', 'Sometimes', 'Often'], palette='coolwarm')
```



Why this chart?
To understand the impact of mental health on daily work performance.

Insights:

- A large number report **"Sometimes"** or **"Often"** interference, indicating productivity loss.
- Very few report "Never" — highlighting how common mental health struggles are in tech.

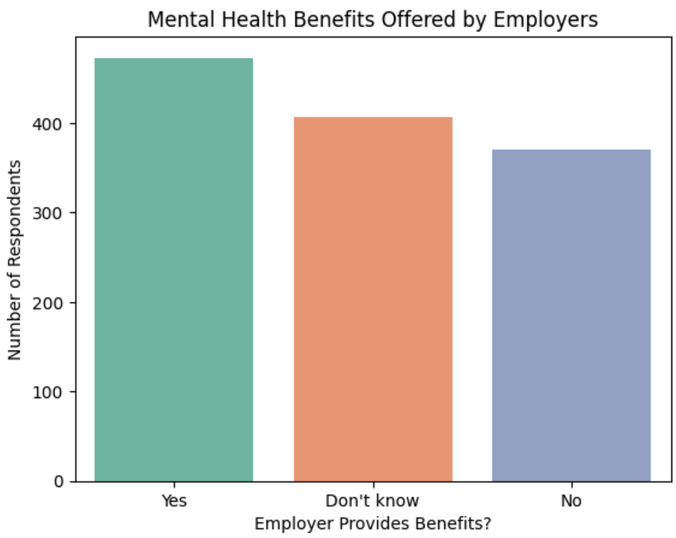
Business Impact:
Direct impact on productivity. Highlights need for flexible leave, counseling, or workload adjustments.

3. Workplace Support & Openness

Chart - 7 : mental health benefits

```
# 7 - Does your company provide mental health benefits?

sns.countplot(data=df_country, x='benefits', palette='Set2')
plt.title("Mental Health Benefits Offered by Employers")
plt.xlabel("Employer Provides Benefits?")
plt.ylabel("Number of Respondents")
plt.show()
```

```
/tmp/ipython-input-150-2378328457.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='benefits', palette='Set2')
```



Why this chart?
To assess whether companies offer mental health coverage and how widespread that support is.

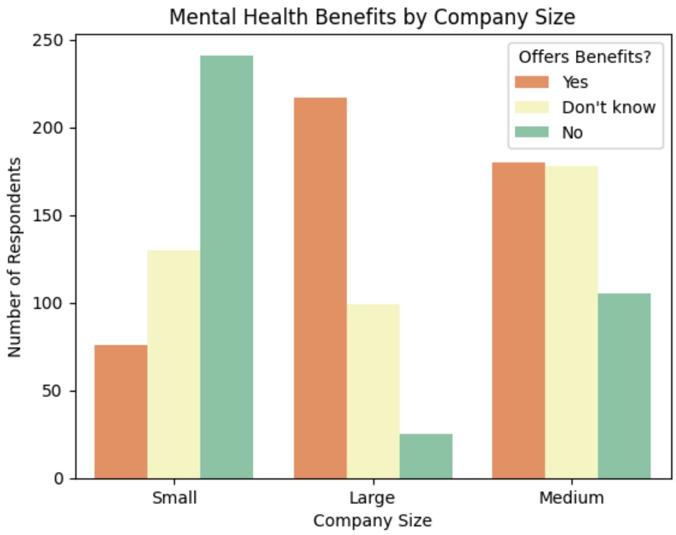
Insights:

- Many respondents answered **"Don't know"**, suggesting **lack of communication or visibility**.
- A large portion of companies still do not offer direct benefits.

Business Impact:
Improved communication of available benefits could increase their utilization and employee trust.

Chart - 8 : company size vs benefits

```
# 8 - Company Size vs Benefits Offered

sns.countplot(data=df_country, x='company_size', hue='benefits', palette='Spectral')
plt.title("Mental Health Benefits by Company Size")
plt.xlabel("Company Size")
plt.ylabel("Number of Respondents")
plt.legend(title="Offers Benefits?")
plt.show()
```



Why this chart?
To analyze if **larger companies** are more likely to offer mental health support.

Insights:

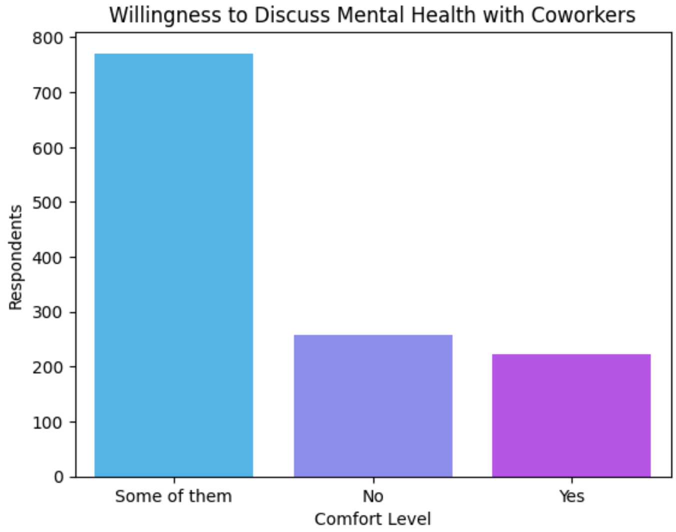
- **Larger companies (500+)** are more likely to provide mental health benefits.
- Smaller companies are either **unaware**, less equipped, or less invested in mental wellness policies.

Business Impact:
Startups and small tech teams may need **standardized frameworks or shared resources** to catch up.

Chart - 9 : general talk

```
# 9 - Openness : Talking to Coworkers

sns.countplot(data=df_country, x='coworkers', palette='cool')
plt.title("Willingness to Discuss Mental Health with Coworkers")
plt.xlabel("Comfort Level")
plt.ylabel("Respondents")
plt.show()
```

```
/tmp/ipython-input-152-1557361383.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='coworkers', palette='cool')
```



Why this chart?
To understand how open employees are to discussing mental health with peers.

Insights:

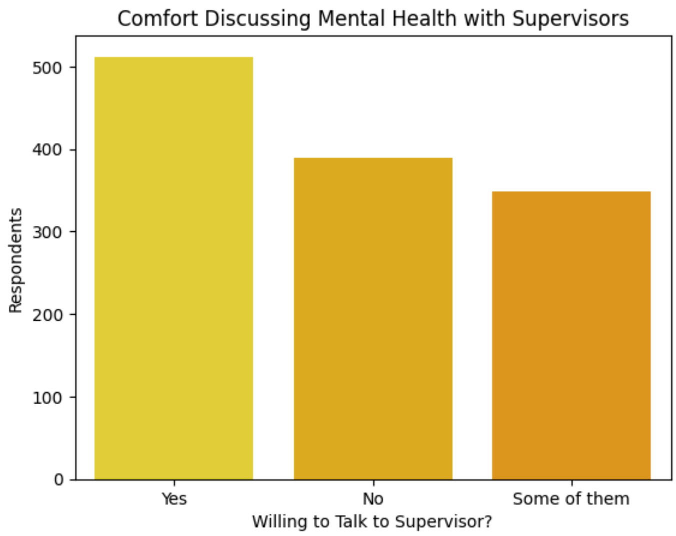
- The response is fairly balanced — not everyone is comfortable, indicating stigma or fear of judgment.

Business Impact:
Companies must work on **normalizing open discussion**, especially through mental health awareness campaigns and leadership modeling.

Chart - 10 : talk with supervisors

```
# 10 - Talking to Supervisors

sns.countplot(data=df_country, x='supervisor', palette='Wistia')
plt.title("Comfort Discussing Mental Health with Supervisors")
plt.xlabel("Willing to Talk to Supervisor?")
plt.ylabel("Respondents")
plt.show()
```

```
/tmp/ipython-input-153-2651794747.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='supervisor', palette='Wistia')
```



Why this chart?
Manager support plays a critical role in encouraging disclosure and accommodation.

Insights:

- Fewer respondents feel comfortable talking to supervisors than to coworkers.
- Supervisor attitudes directly affect willingness to seek help.

Business Impact:
Train managers to respond **empathetically and confidentially** to mental health disclosures.

```
# 10b - Talking to Supervisors - company size wise

sns.countplot(data=df_country, x='supervisor',hue='company_size', palette='Wistia')
plt.title("Comfort Discussing Mental Health with Supervisors")
plt.xlabel("Willing to Talk to Supervisor?")
plt.ylabel("Respondents")
plt.show()
```

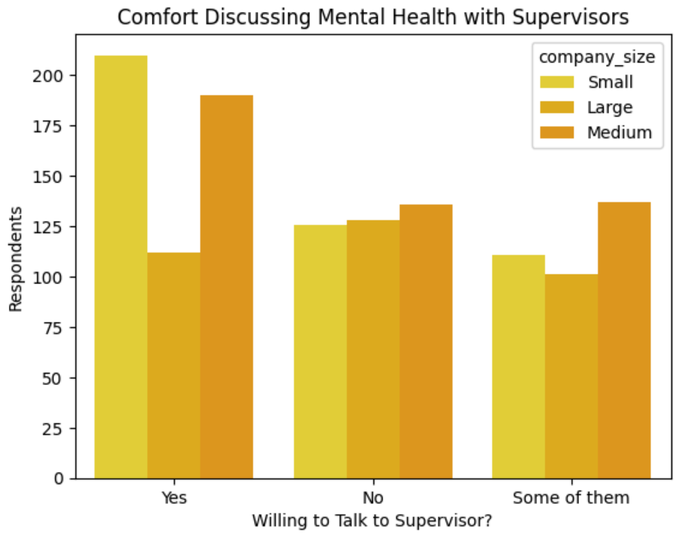
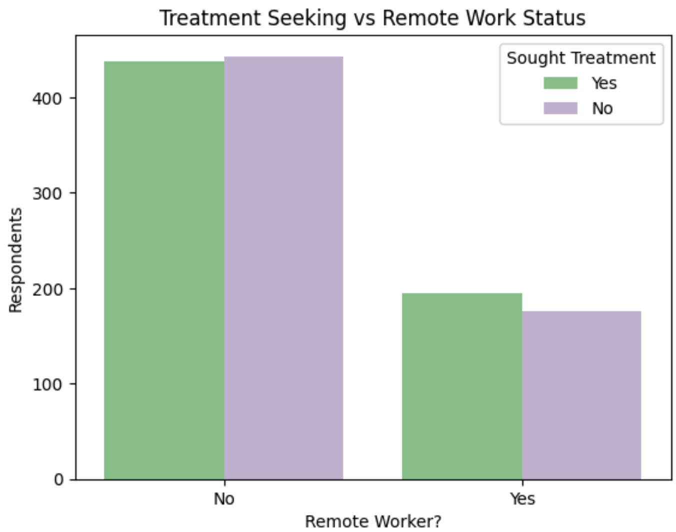


Chart - 11 : Remote work vs Treatment seeking

```
# 11 - Remote Work vs Treatment Seeking

sns.countplot(data=df_country, x='remote_work', hue='treatment', palette='Accent')
plt.title("Treatment Seeking vs Remote Work Status")
plt.xlabel("Remote Worker?")
plt.ylabel("Respondents")
plt.legend(title="Sought Treatment")
plt.show()
```



Why this chart?
To check if remote employees are more or less likely to seek treatment.

Insights:

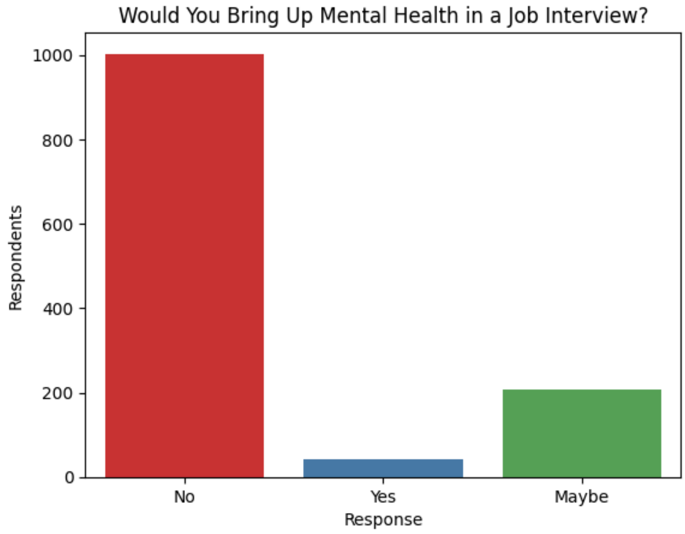
- No strong visible pattern, but some evidence that **remote workers** may be **slightly more open** to seeking help.

Business Impact:
Companies should ensure remote workers are equally covered and informed about available resources.

Chart - 12 : Talk in an interview

```
# 12 - Would you bring up a mental health issue in an interview?

sns.countplot(data=df_country, x='mental_health_interview', palette='Set1')
plt.title("Would You Bring Up Mental Health in a Job Interview?")
plt.xlabel("Response")
plt.ylabel("Respondents")
plt.show()
```

```
/tmp/ipython-input-156-1886967927.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='mental_health_interview', palette='Set1')
```



Why this chart?
To see how safe people feel discussing mental health when applying for a job.

Insights:

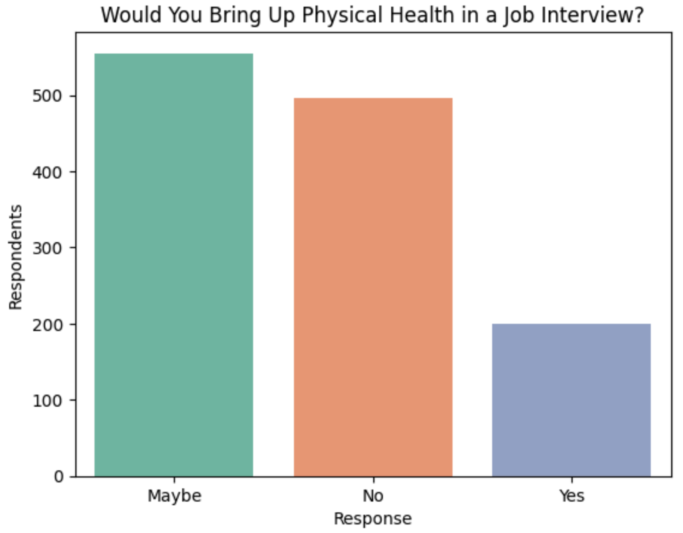
- Majority are **unwilling** to bring it up, even if it affects their work.
- Indicates a **strong perceived stigma** or risk of discrimination during hiring.

Business Impact:
Companies must make it clear through HR and public communication that **mental health history won't be used unfairly** in selection.

Chart - 13 : Physical vs Mental Health

```
# 13 - Would you bring up a physical health issue in an interview?

sns.countplot(data=df_country, x='phys_health_interview', palette='Set2')
plt.title("Would You Bring Up Physical Health in a Job Interview?")
plt.xlabel("Response")
plt.ylabel("Respondents")
plt.show()
```

```
/tmp/ipython-input-157-1384571048.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='phys_health_interview', palette='Set2')
```



Why this chart?
To compare perceptions around mental vs physical health disclosure.

Insights:

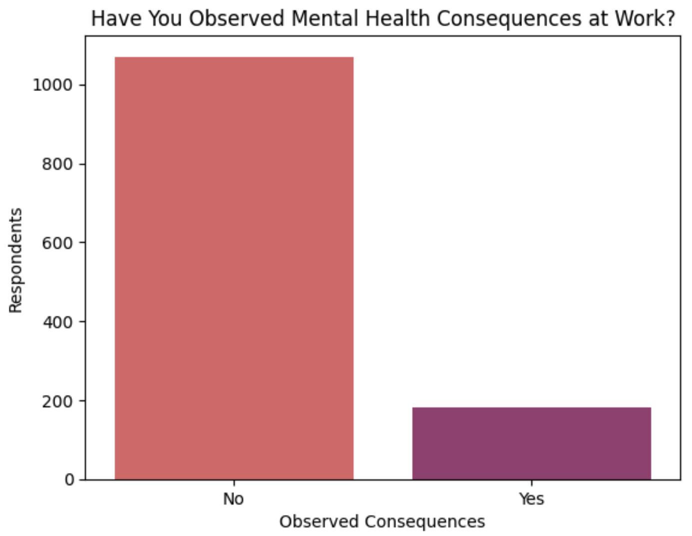
- More people are open to disclosing **physical health conditions**, confirming a **double standard** in how health is treated.

Business Impact:
Bridging the gap between mental and physical health policies can build trust and reduce fear.

Chart - 14 : Coorporate Consequences

```
# 14 - Have you observed negative consequences for others with mental health conditions?

sns.countplot(data=df_country, x='obs_consequence', palette='flare')
plt.title("Have You Observed Mental Health Consequences at Work?")
plt.xlabel("Observed Consequences")
plt.ylabel("Respondents")
plt.show()
```

```
/tmp/ipython-input-158-1288831179.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for

  sns.countplot(data=df_country, x='obs_consequence', palette='flare')
```



Why this chart?
To understand if stigma or punishment for mental health issues exists in the workplace.

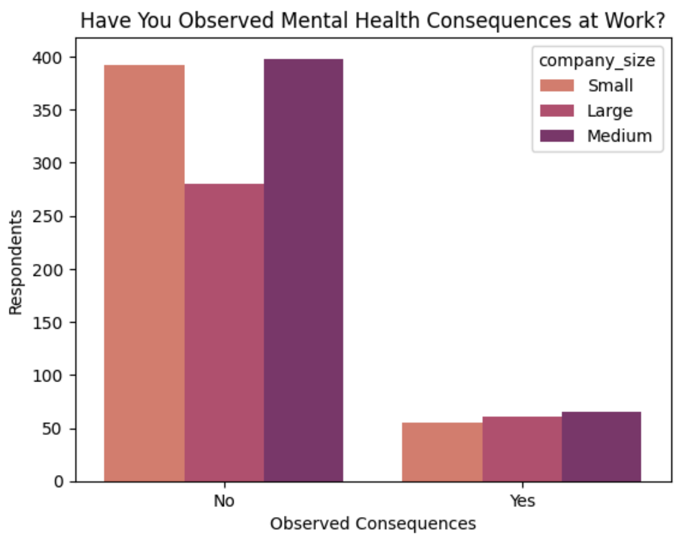
Insights:

- A few number of respondents have **witnessed negative outcomes** for others.
- Shows that even when formal policies exist, **culture may not align**.

Business Impact:

Cultural shifts and leadership modeling are essential to **back up policy with practice**.

```
sns.countplot(data=df_country, x='obs_consequence',hue='company_size', palette='flare')
plt.title("Have You Observed Mental Health Consequences at Work?")
plt.xlabel("Observed Consequences")
plt.ylabel("Respondents")
plt.show()
```



Others

Chart - 15 - Correlation Heatmap

```
# Correlation Heatmap visualization code

binary_df = df_country.copy()
binary_df['treatment'] = binary_df['treatment'].map({'Yes': 1, 'No': 0})
binary_df['family_history'] = binary_df['family_history'].map({'Yes': 1, 'No': 0})
binary_df['remote_work'] = binary_df['remote_work'].map({'Yes': 1, 'No': 0})

sns.heatmap(binary_df[['treatment', 'family_history', 'remote_work']].corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap (Binary Fields)")
plt.show()
```