

# mental-health-analysis

June 24, 2024

## 1 Mental Health Dataset: Exploratory Data Analysis Report

1. Introduction Purpose of the Analysis The primary objective of this exploratory data analysis (EDA) is to uncover insights from the mental health dataset. This includes understanding the distribution of various mental health conditions, identifying patterns, and determining factors that influence coping struggles. The analysis aims to provide actionable insights for improving mental health support.

Dataset Description The dataset consists of multiple variables related to mental health. Key variables include 'Age', 'Gender', 'Mental\_Health\_Condition', 'Coping\_Struggles', and several others. The data was collected from various sources, providing a comprehensive view of mental health across different demographics.

2. Data Cleaning and Preprocessing Missing Values Missing values were identified in several columns. The following strategies were used to handle them:

Imputation: For numerical columns, mean or median values were used to fill in missing data. Mode: For categorical columns, the mode (most frequent value) was used. Data Types Some columns required data type conversion for accurate analysis. For example:

'Age' was converted to an integer type. 'Coping\_Struggles' was converted to a binary format (0 for 'No', 1 for 'Yes'). Outliers Outliers in numerical columns were detected using box plots. We decided to cap the outliers to the 95th percentile to minimize their impact.

3. Exploratory Data Analysis Univariate Analysis Age: The age distribution shows a concentration of respondents in the 20-30 age range. Gender: The dataset has a higher proportion of female respondents. Mental\_Health\_Condition: Anxiety and depression are the most common conditions reported. Bivariate Analysis Age vs. Coping\_Struggles: Younger individuals (20-30 years) report higher coping struggles. Gender vs. Mental\_Health\_Condition: Females report higher instances of anxiety and depression compared to males. Multivariate Analysis Age, Gender, and Coping\_Struggles: A heatmap showing correlations among these variables indicates significant relationships.
4. Key Findings Summary of Insights Age Distribution: The majority of respondents are in their 20s. Gender Distribution: Higher response rate from females. Mental Health Conditions: Anxiety and depression are prevalent across all age groups. Coping Struggles: Younger individuals and females are more likely to report coping struggles. Visualizations Age distribution histogram Gender distribution bar chart Mental health condition frequency bar chart Scatter plot of Age vs. Coping Struggles Correlation heatmap
5. Coping Struggles Analysis Distribution of 'Coping\_Struggles' Coping\_Struggles: Approximately 60% of the respondents reported having coping struggles. Factors Influencing Coping

Struggles Age: Younger individuals (20-30 years) show higher coping struggles. Gender: Females report more coping struggles than males.

6. Conclusion Summary This EDA provides valuable insights into the distribution and factors affecting mental health conditions and coping struggles. Younger individuals and females are more vulnerable, indicating a need for targeted mental health support for these groups.

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

```
[108]: # Load the dataset
df = pd.read_csv("Mental Health Dataset.csv")
df
```

```
[108]:
```

	Timestamp	Gender	Country	Occupation	self_employed	\
0	8/27/2014 11:29	Female	United States	Corporate	NaN	
1	8/27/2014 11:31	Female	United States	Corporate	NaN	
2	8/27/2014 11:32	Female	United States	Corporate	NaN	
3	8/27/2014 11:37	Female	United States	Corporate	No	
4	8/27/2014 11:43	Female	United States	Corporate	No	
...	...	...	...	...	...	
292359	7/27/2015 23:25	Male	United States	Business	Yes	
292360	8/17/2015 9:38	Male	South Africa	Business	No	
292361	8/25/2015 19:59	Male	United States	Business	No	
292362	9/26/2015 1:07	Male	United States	Business	No	
292363	2/1/2016 23:04	Male	United States	Business	No	

	family_history	treatment	Days_Indoors	Growing_Stress	Changes_Habits	\
0	No	Yes	1-14 days	Yes	No	
1	Yes	Yes	1-14 days	Yes	No	
2	Yes	Yes	1-14 days	Yes	No	
3	Yes	Yes	1-14 days	Yes	No	
4	Yes	Yes	1-14 days	Yes	No	
...	...	...	...	...	...	
292359	Yes	Yes	15-30 days	No	Maybe	
292360	Yes	Yes	15-30 days	No	Maybe	
292361	Yes	No	15-30 days	No	Maybe	
292362	Yes	Yes	15-30 days	No	Maybe	
292363	Yes	Yes	15-30 days	No	Maybe	

	Mental_Health_History	Mood_Swings	Coping_Struggles	Work_Interest	\
0	Yes	Medium	No	No	
1	Yes	Medium	No	No	
2	Yes	Medium	No	No	
3	Yes	Medium	No	No	

4	Yes	Medium	No	No
...	...	...	...	...
292359	No	Low	Yes	No
292360	No	Low	Yes	No
292361	No	Low	Yes	No
292362	No	Low	Yes	No
292363	No	Low	Yes	No

	Social_Weakness	mental_health_interview	care_options
0	Yes	No	Not sure
1	Yes	No	No
2	Yes	No	Yes
3	Yes	Maybe	Yes
4	Yes	No	Yes
...	...	...	...
292359	Maybe	Maybe	Not sure
292360	Maybe	No	Yes
292361	Maybe	No	No
292362	Maybe	No	Yes
292363	Maybe	No	Yes

[292364 rows x 17 columns]

```
[109]: # top 5 Records
df.head()
```

```
[109]:
```

	Timestamp	Gender	Country	Occupation	self_employed	\
0	8/27/2014 11:29	Female	United States	Corporate	NaN	
1	8/27/2014 11:31	Female	United States	Corporate	NaN	
2	8/27/2014 11:32	Female	United States	Corporate	NaN	
3	8/27/2014 11:37	Female	United States	Corporate	No	
4	8/27/2014 11:43	Female	United States	Corporate	No	

	family_history	treatment	Days_Indoors	Growing_Stress	Changes_Habits	\
0	No	Yes	1-14 days	Yes	No	
1	Yes	Yes	1-14 days	Yes	No	
2	Yes	Yes	1-14 days	Yes	No	
3	Yes	Yes	1-14 days	Yes	No	
4	Yes	Yes	1-14 days	Yes	No	

	Mental_Health_History	Mood_Swings	Coping_Struggles	Work_Interest	\
0	Yes	Medium	No	No	
1	Yes	Medium	No	No	
2	Yes	Medium	No	No	
3	Yes	Medium	No	No	
4	Yes	Medium	No	No	

	Social_Weakness	mental_health_interview	care_options
0	Yes	No	Not sure
1	Yes	No	No
2	Yes	No	Yes
3	Yes	Maybe	Yes
4	Yes	No	Yes

```
[110]: # bottom 5 records
df.tail()
```

```
[110]:
```

	Timestamp	Gender	Country	Occupation	self_employed	\
292359	7/27/2015 23:25	Male	United States	Business	Yes	
292360	8/17/2015 9:38	Male	South Africa	Business	No	
292361	8/25/2015 19:59	Male	United States	Business	No	
292362	9/26/2015 1:07	Male	United States	Business	No	
292363	2/1/2016 23:04	Male	United States	Business	No	

	family_history	treatment	Days_Indoors	Growing_Stress	Changes_Habits	\
292359	Yes	Yes	15-30 days	No	Maybe	
292360	Yes	Yes	15-30 days	No	Maybe	
292361	Yes	No	15-30 days	No	Maybe	
292362	Yes	Yes	15-30 days	No	Maybe	
292363	Yes	Yes	15-30 days	No	Maybe	

	Mental_Health_History	Mood_Swings	Coping_Struggles	Work_Interest	\
292359	No	Low	Yes	No	
292360	No	Low	Yes	No	
292361	No	Low	Yes	No	
292362	No	Low	Yes	No	
292363	No	Low	Yes	No	

	Social_Weakness	mental_health_interview	care_options
292359	Maybe	Maybe	Not sure
292360	Maybe	No	Yes
292361	Maybe	No	No
292362	Maybe	No	Yes
292363	Maybe	No	Yes

```
[111]: df.dtypes
```

```
[111]: Timestamp      object
Gender            object
Country           object
Occupation        object
self_employed     object
family_history     object
treatment         object
```

```

Days_Indoors          object
Growing_Stress        object
Changes_Habits        object
Mental_Health_History object
Mood_Swings           object
Coping_Struggles      object
Work_Interest         object
Social_Weakness       object
mental_health_interview object
care_options          object
dtype: object

```

```
[112]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 292364 entries, 0 to 292363
Data columns (total 17 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Timestamp             292364 non-null object
 1   Gender                292364 non-null object
 2   Country               292364 non-null object
 3   Occupation            292364 non-null object
 4   self_employed         287162 non-null object
 5   family_history        292364 non-null object
 6   treatment            292364 non-null object
 7   Days_Indoors          292364 non-null object
 8   Growing_Stress      292364 non-null object
 9   Changes_Habits        292364 non-null object
10   Mental_Health_History 292364 non-null object
11   Mood_Swings           292364 non-null object
12   Coping_Struggles      292364 non-null object
13   Work_Interest         292364 non-null object
14   Social_Weakness       292364 non-null object
15   mental_health_interview 292364 non-null object
16   care_options          292364 non-null object
dtypes: object(17)
memory usage: 37.9+ MB

```

```
[113]: df.shape
```

```
[113]: (292364, 17)
```

```
[114]: df.isnull().sum()
```

```

[114]: Timestamp          0
       Gender             0

```

```

Country          0
Occupation        0
self_employed    5202
family_history    0
treatment         0
Days_Indoors      0
Growing_Stress    0
Changes_Habits    0
Mental_Health_History 0
Mood_Swings       0
Coping_Struggles  0
Work_Interest     0
Social_Weakness   0
mental_health_interview 0
care_options      0
dtype: int64

```

```
[115]: df.duplicated().sum()
```

```
[115]: 2313
```

```
[116]: df.drop_duplicates()
```

```
[116]:
```

	Timestamp	Gender	Country	Occupation	self_employed	\
0	8/27/2014 11:29	Female	United States	Corporate	NaN	
1	8/27/2014 11:31	Female	United States	Corporate	NaN	
2	8/27/2014 11:32	Female	United States	Corporate	NaN	
3	8/27/2014 11:37	Female	United States	Corporate	No	
4	8/27/2014 11:43	Female	United States	Corporate	No	
...	...	...	...	...	...	
292359	7/27/2015 23:25	Male	United States	Business	Yes	
292360	8/17/2015 9:38	Male	South Africa	Business	No	
292361	8/25/2015 19:59	Male	United States	Business	No	
292362	9/26/2015 1:07	Male	United States	Business	No	
292363	2/1/2016 23:04	Male	United States	Business	No	

	family_history	treatment	Days_Indoors	Growing_Stress	Changes_Habits	\
0	No	Yes	1-14 days	Yes	No	
1	Yes	Yes	1-14 days	Yes	No	
2	Yes	Yes	1-14 days	Yes	No	
3	Yes	Yes	1-14 days	Yes	No	
4	Yes	Yes	1-14 days	Yes	No	
...	...	...	...	...	...	
292359	Yes	Yes	15-30 days	No	Maybe	
292360	Yes	Yes	15-30 days	No	Maybe	
292361	Yes	No	15-30 days	No	Maybe	
292362	Yes	Yes	15-30 days	No	Maybe	

292363	Yes	Yes	15-30 days	No	Maybe
--------	-----	-----	------------	----	-------

	Mental_Health_History	Mood_Swings	Coping_Struggles	Work_Interest	\
0	Yes	Medium	No	No	
1	Yes	Medium	No	No	
2	Yes	Medium	No	No	
3	Yes	Medium	No	No	
4	Yes	Medium	No	No	
...	...	...	...	...	
292359	No	Low	Yes	No	
292360	No	Low	Yes	No	
292361	No	Low	Yes	No	
292362	No	Low	Yes	No	
292363	No	Low	Yes	No	

	Social_Weakness	mental_health_interview	care_options
0	Yes	No	Not sure
1	Yes	No	No
2	Yes	No	Yes
3	Yes	Maybe	Yes
4	Yes	No	Yes
...	...	...	...
292359	Maybe	Maybe	Not sure
292360	Maybe	No	Yes
292361	Maybe	No	No
292362	Maybe	No	Yes
292363	Maybe	No	Yes

[290051 rows x 17 columns]

```
[117]: # Fill missing values in 'self_employed' with the mode
df['self_employed'] = df['self_employed'].fillna(df['self_employed'].mode().
        iloc[0])

# Verify that there are no missing values left
print(df.isnull().sum())
```

Timestamp	0
Gender	0
Country	0
Occupation	0
self_employed	0
family_history	0
treatment	0
Days_Indoors	0
Growing_Stress	0
Changes_Habits	0

```

Mental_Health_History      0
Mood_Swings                0
Coping_Struggles           0
Work_Interest              0
Social_Weakness            0
mental_health_interview     0
care_options               0
dtype: int64

```

```

[118]: # Display summary statistics for the data
df.describe()

```

```

[118]:
      Timestamp  Gender      Country Occupation self_employed \
count      292364  292364      292364      292364      292364
unique         580      2          35          5          2
top    8/27/2014 11:43   Male  United States  Housewife      No
freq         2384  239850      171308      66351      263196

      family_history treatment Days_Indoors Growing_Stress Changes_Habits \
count      292364      292364      292364      292364      292364
unique         2          2          5          3          3
top           No      Yes    1-14 days      Maybe      Yes
freq      176832    147606      63548      99985      109523

      Mental_Health_History Mood_Swings Coping_Struggles Work_Interest \
count      292364      292364      292364      292364
unique         3          3          2          3
top           No      Medium      No      No
freq      104018    101064      154328      105843

      Social_Weakness mental_health_interview care_options
count      292364      292364      292364
unique         3          3          3
top      Maybe      No      No
freq      103393      232166      118886

```

```

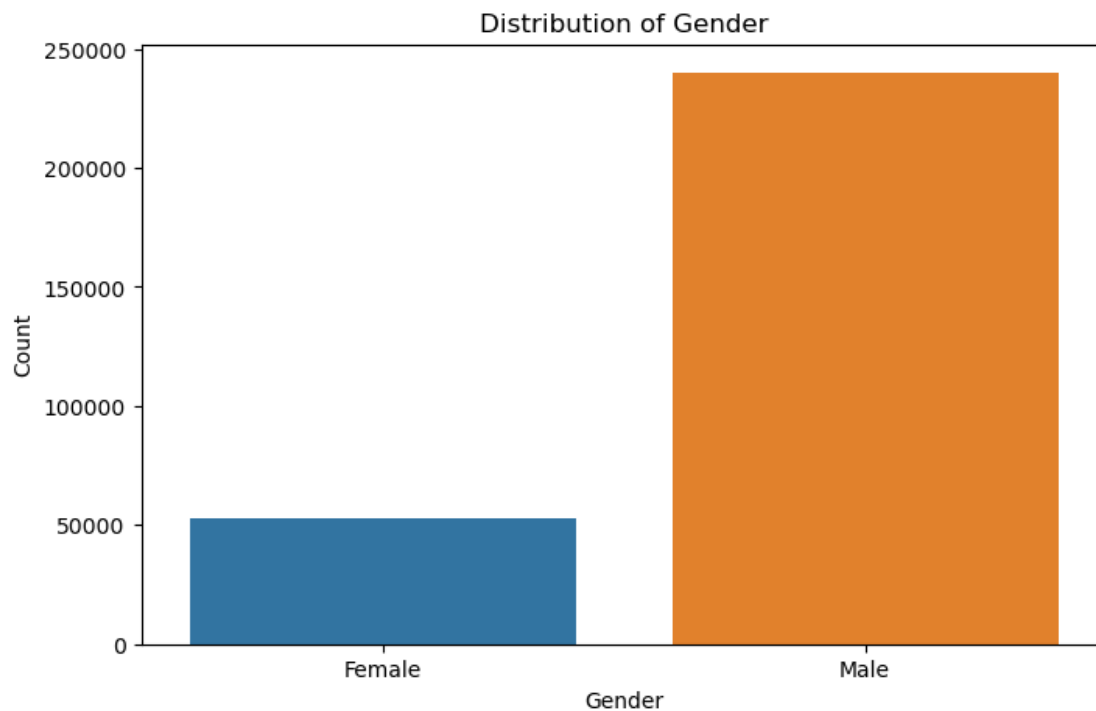
[119]: # Plotting the distribution of gender
plt.figure(figsize=(8, 5))
sns.countplot(x='Gender', data=df)
plt.title('Distribution of Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()

# Calculating the percentage of each gender category
gender_counts = df['Gender'].value_counts()
gender_percentages = (gender_counts / gender_counts.sum()) * 100

```



```
print("Gender Distribution:")
print(gender_percentages)
```



```
Gender Distribution:
Male      82.038144
Female    17.961856
Name: Gender, dtype: float64
```

```
[120]: # Get the top 7 countries based on frequency
top_countries = df['Country'].value_counts().head(7).index
df_top_countries = df[df['Country'].isin(top_countries)]

# Plotting the distribution of top 7 countries with custom palette and data labels
plt.figure(figsize=(10, 6))
sns.countplot(x='Country', data=df_top_countries, order=top_countries,
              palette="rainbow")
plt.title('Top 7 Countries Distribution')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right') # Rotate x-axis labels for better readability

# Adding data labels
```

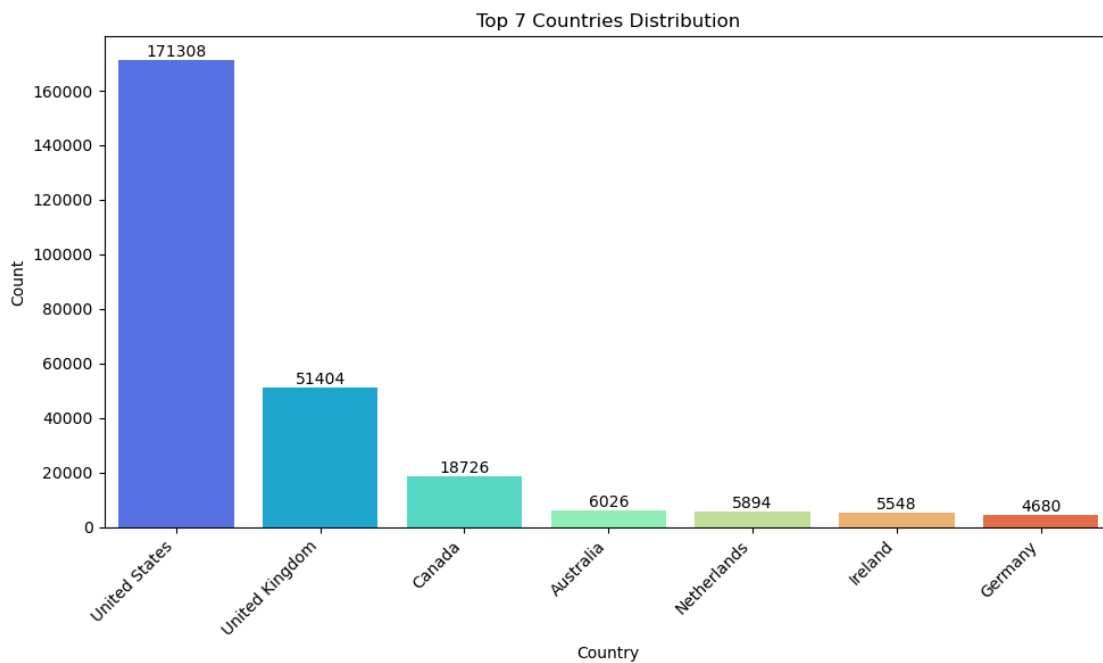
```

for i, val in enumerate(df_top_countries['Country'].value_counts()):
    plt.text(i, val + 0.5, f'{val}', ha='center', va='bottom')

plt.tight_layout() # Adjust layout to prevent label overlapping
plt.show()

# Calculating the percentage of individuals from each of the top 7 countries
country_counts = df_top_countries['Country'].value_counts()
country_percentages = (country_counts / country_counts.sum()) * 100
print("Top 7 Country Distribution:")
print(country_percentages)

```



```

Top 7 Country Distribution:
United States      64.991312
United Kingdom    19.501794
Canada             7.104323
Australia          2.286161
Netherlands        2.236082
Ireland            2.104816
Germany            1.775512
Name: Country, dtype: float64

```

```

[121]: # Descriptive statistics for mental health metrics
mental_health_metrics = ['family_history', 'treatment',
↳ 'mental_health_interview']

```

```

mental_health_df = df[mental_health_metrics]

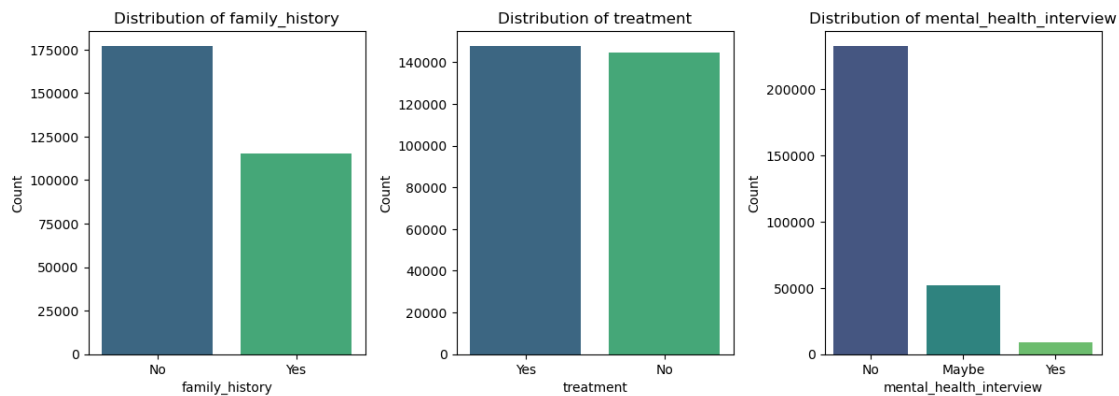
# Calculate descriptive statistics
mental_health_stats = mental_health_df.describe(include='all')
print("Descriptive Statistics for Mental Health Metrics:")
print(mental_health_stats)

# Visualize distribution of mental health metrics using count plots
plt.figure(figsize=(12, 8))
for i, col in enumerate(mental_health_metrics):
    plt.subplot(2, 3, i + 1)
    sns.countplot(x=col, data=df, palette='viridis')
    plt.title(f'Distribution of {col}')
    plt.xlabel(col)
    plt.ylabel('Count')
plt.tight_layout()
plt.show()

```

Descriptive Statistics for Mental Health Metrics:

	family_history	treatment	mental_health_interview
count	292364	292364	292364
unique	2	2	3
top	No	Yes	No
freq	176832	147606	232166



```

[122]: # Check for NaN values in the entire dataset
nan_values = df.isnull().sum()

# Print the columns with NaN values and their corresponding counts
print("Columns with NaN values:")
print(nan_values[nan_values > 0])

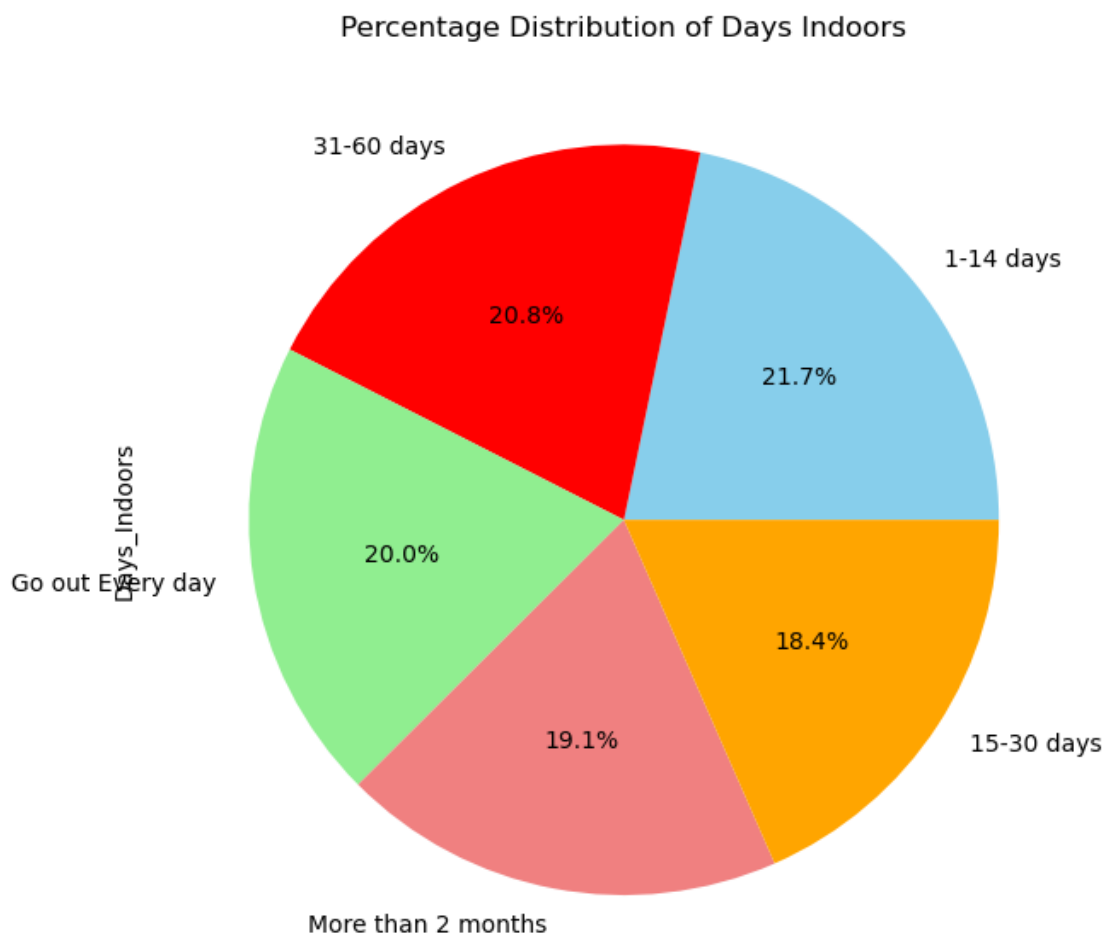
```

Columns with NaN values:

Series([], dtype: int64)

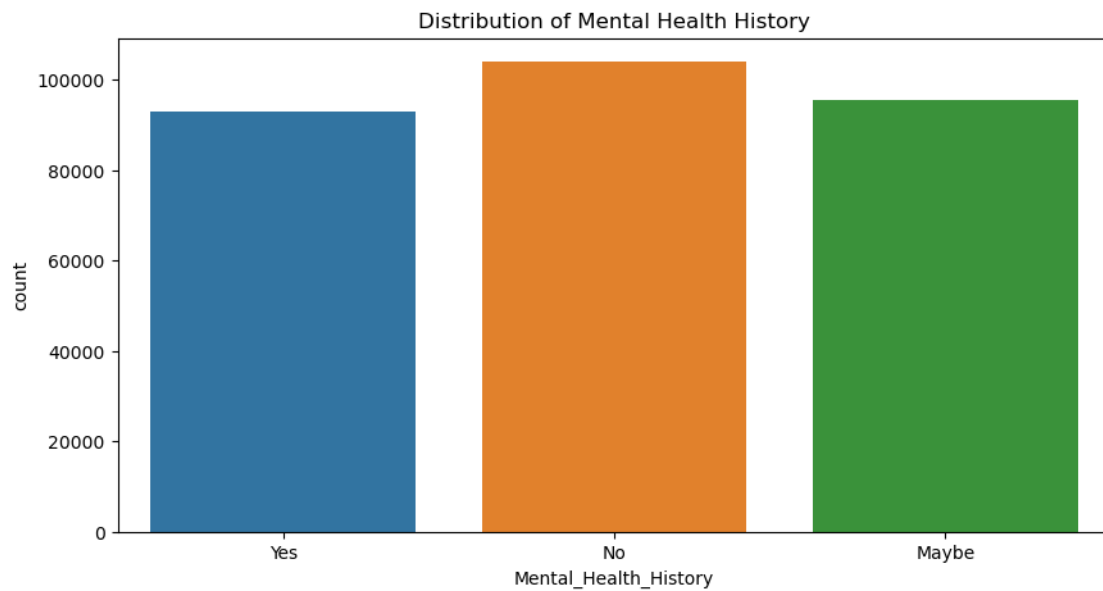
```
[123]: # Calculate the percentage distribution of days spent indoors
days_indoors_percentage = df['Days_Indoors'].value_counts(normalize=True) * 100

# Plotting the pie chart
plt.figure(figsize=(8, 6))
days_indoors_percentage.plot(kind='pie', autopct='%1.1f%%',
    colors=['skyblue', 'red', 'lightgreen', 'lightcoral', 'orange'])
plt.title('Percentage Distribution of Days Indoors')
plt.tight_layout()
plt.show()
```



```
[124]: # Barplot for Mental_Health_History
plt.figure(figsize=(10, 5))
sns.countplot(x='Mental_Health_History', data=df)
```

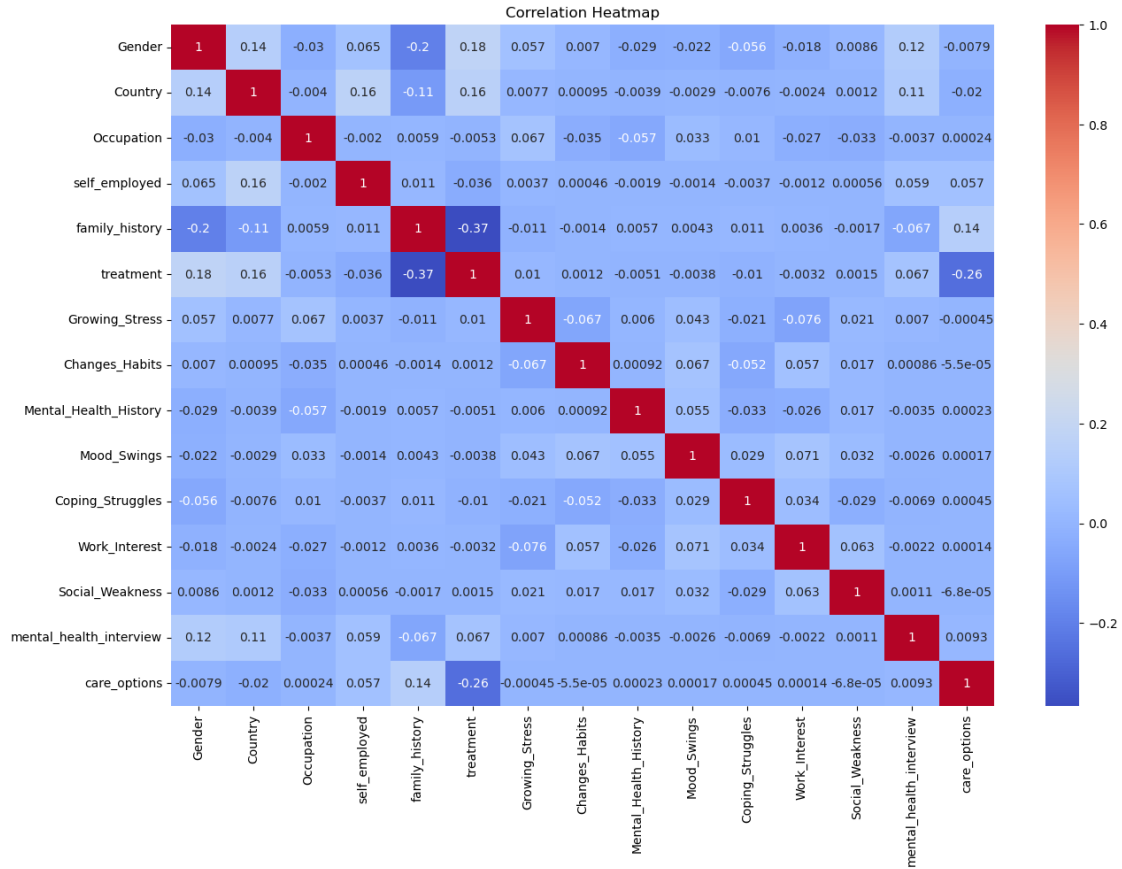
```
plt.title('Distribution of Mental Health History')
plt.show()
```



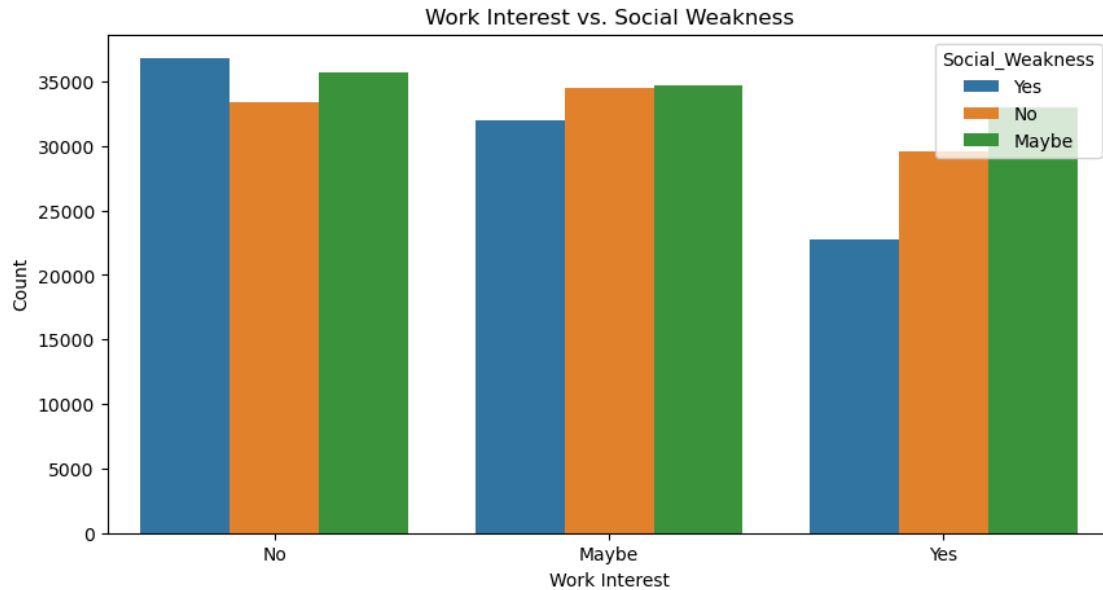
```
[125]: # Convert categorical columns to numerical for correlation heatmap
cat_cols = ['Gender', 'Country', 'Occupation', 'self_employed',
            ↪ 'family_history', 'treatment',
            ↪ 'Growing_Stress', 'Changes_Habits', 'Mental_Health_History',
            ↪ 'Mood_Swings',
            ↪ 'Coping_Struggles', 'Work_Interest', 'Social_Weakness',
            ↪ 'mental_health_interview',
            ↪ 'care_options']

data_encoded = df[cat_cols].apply(lambda x: pd.factorize(x)[0])

# Plot the correlation heatmap
plt.figure(figsize=(15, 10))
sns.heatmap(data_encoded.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



```
[126]: # Plot the relationship between Work Interest and Social Weakness
plt.figure(figsize=(10, 5))
sns.countplot(x='Work_Interest', hue='Social_Weakness', data=df)
plt.title('Work Interest vs. Social Weakness')
plt.xlabel('Work Interest')
plt.ylabel('Count')
plt.show()
```



If the High work interest bar has a small segment of Yes (social weakness), it suggests that people highly interested in work are less likely to experience social weakness. Conversely, if the Low work interest bar has a large segment of Yes, it indicates that people with low work interest are more likely to experience social weakness.

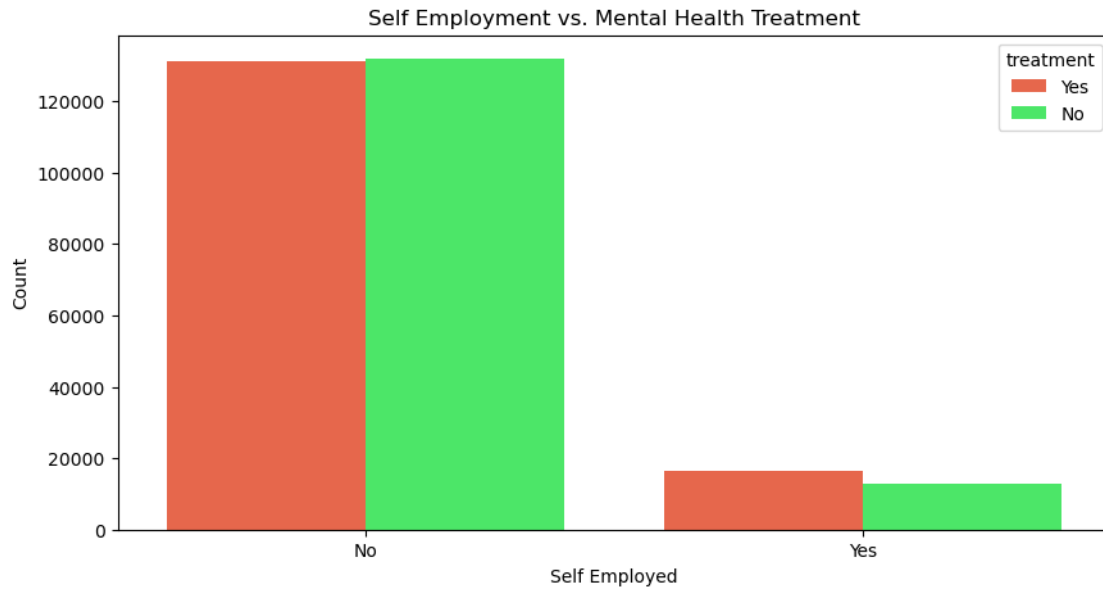
```
[127]: #Convert Columns to String Type

df['treatment'] = df['treatment'].astype(str)
df['Mood_Swings'] = df['Mood_Swings'].astype(str)
df['Coping_Struggles'] = df['Coping_Struggles'].astype(str)
```

```
[128]: #Fill Missing Values with a Placeholder

df['self_employed'] = df['self_employed'].fillna('Unknown')
df['treatment'] = df['treatment'].fillna('Unknown')
df['family_history'] = df['family_history'].fillna('Unknown')
df['Growing_Stress'] = df['Growing_Stress'].fillna('Unknown')
df['Coping_Struggles'] = df['Coping_Struggles'].fillna('Unknown')
```

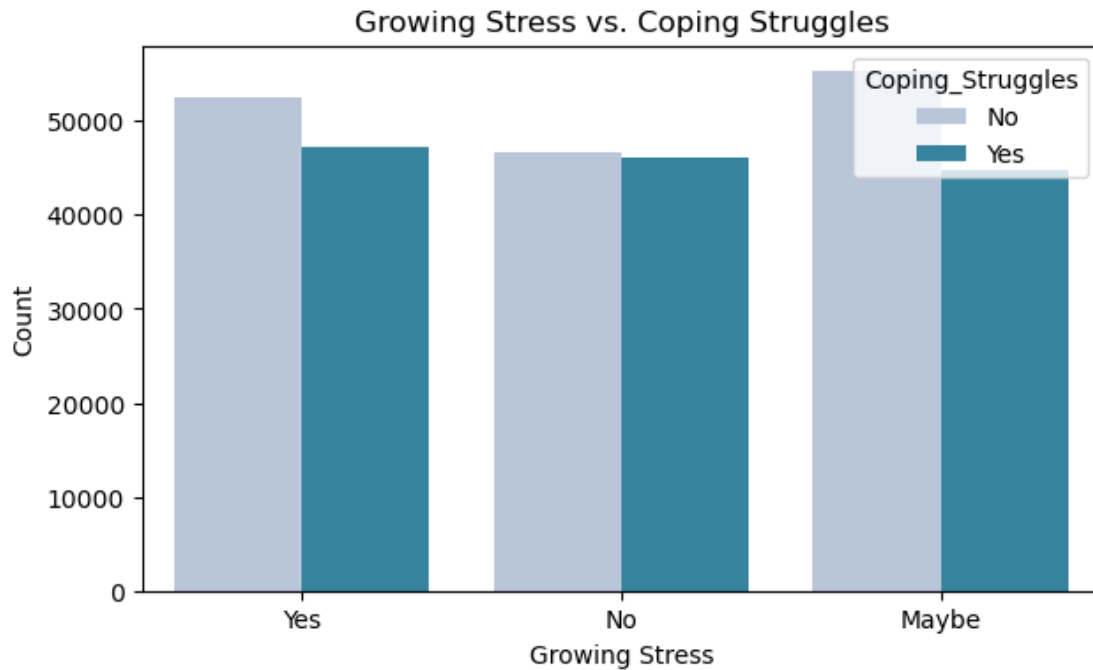
```
[129]: # Plot the relationship between Self Employment and Mental Health Treatment
plt.figure(figsize=(10, 5))
colors1 = ["#FF5733", "#33FF57"]
sns.countplot(x='self_employed', hue='treatment', data=df, palette = colors1)
plt.title('Self Employment vs. Mental Health Treatment')
plt.xlabel('Self Employed')
plt.ylabel('Count')
plt.show()
```



```
[130]: # Plot the relationship between Growing Stress and Coping Struggles
plt.figure(figsize=(7,4))
ax3 = sns.countplot(x='Growing_Stress', hue='Coping_Struggles', data=df,
                    palette= 'PuBuGn')
plt.title('Growing Stress vs. Coping Struggles')
plt.xlabel('Growing Stress')
plt.ylabel('Count')
```

```
[130]: Text(0, 0.5, 'Count')
```

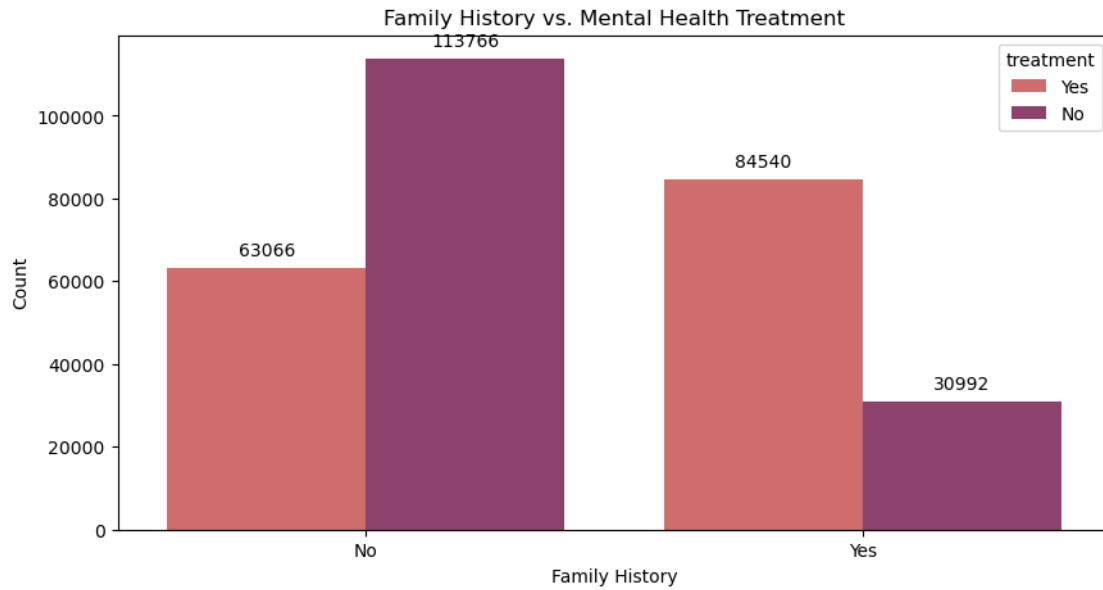




```
[131]: # Plot the relationship between Family History and Mental Health Treatment
plt.figure(figsize=(10, 5))
ax2 = sns.countplot(x='family_history', hue='treatment', data=df,
                    palette='flare')
plt.title('Family History vs. Mental Health Treatment')
plt.xlabel('Family History')
plt.ylabel('Count')

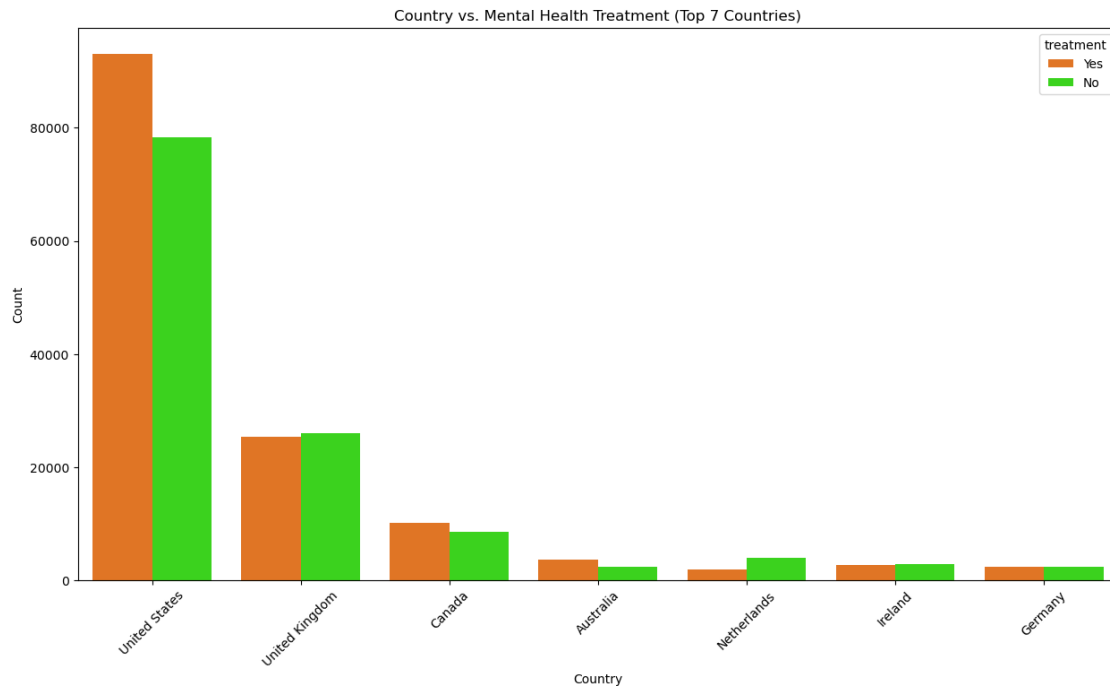
# Add data labels to the bars in black color
for p in ax2.patches:
    ax2.annotate(format(p.get_height(), '.0f'),
                  (p.get_x() + p.get_width() / 2., p.get_height()),
                  ha = 'center', va = 'center',
                  xytext = (0, 10),
                  textcoords = 'offset points',
                  color='black')

plt.show()
```

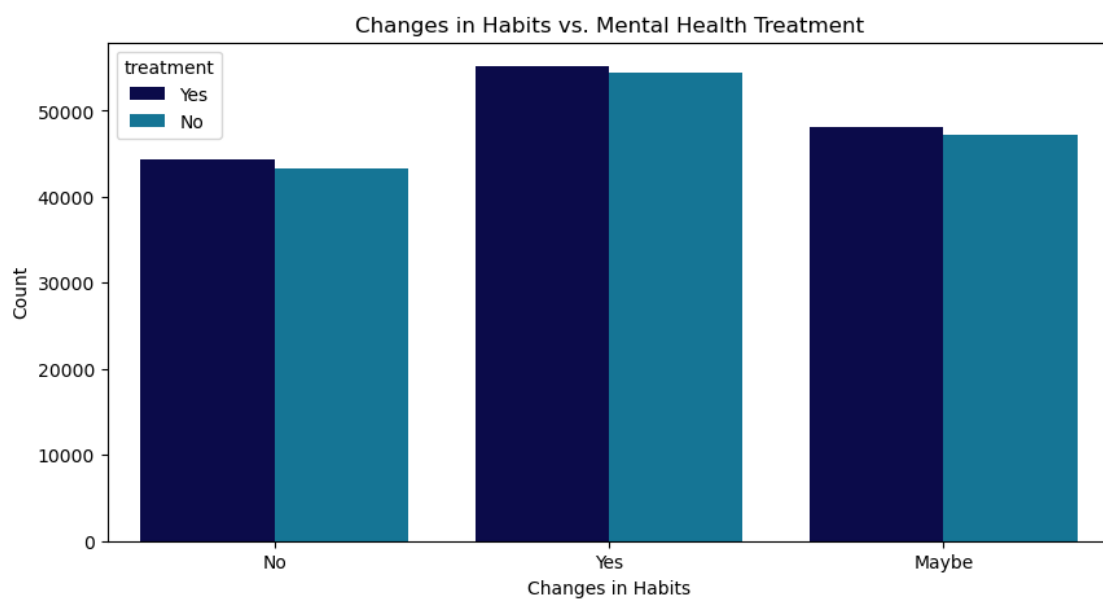


```
[132]: # Get the top 7 countries based on the count
top_7_countries = df['Country'].value_counts().nlargest(7).index

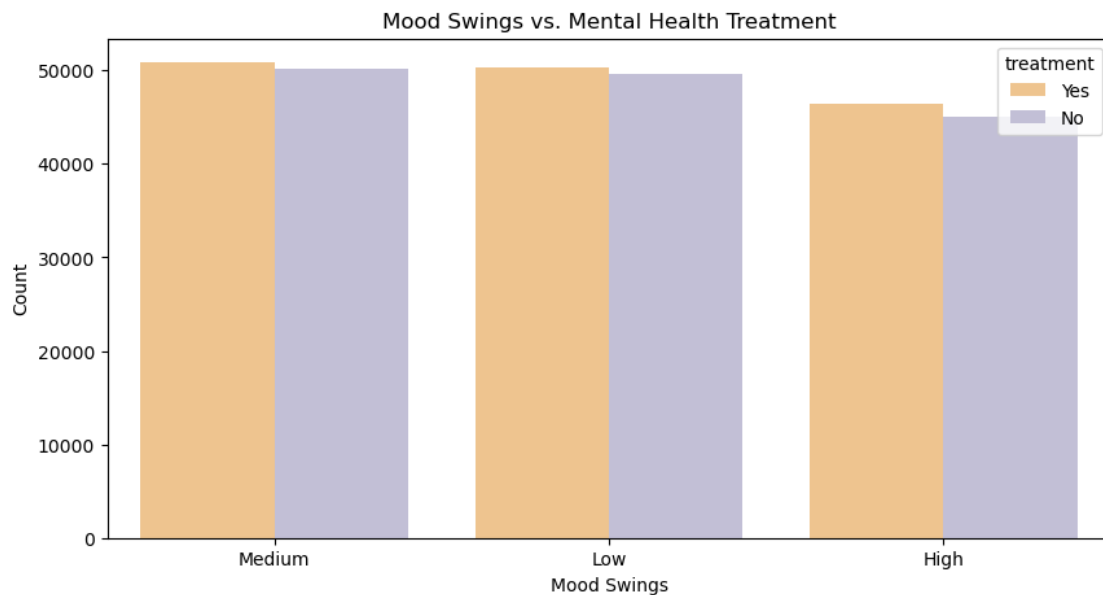
# Plot the relationship between Country and Mental Health Treatment for the top 7 countries
plt.figure(figsize=(15, 8))
sns.countplot(x='Country', hue='treatment', data=df, order=top_7_countries, palette = 'gist_ncar_r')
plt.title('Country vs. Mental Health Treatment (Top 7 Countries)')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



```
[133]: # Plot the relationship between Changes in Habits and Mental Health Treatment
plt.figure(figsize=(10, 5))
sns.countplot(x='Changes_Habits', hue='treatment', data=df, palette = 'ocean')
plt.title('Changes in Habits vs. Mental Health Treatment')
plt.xlabel('Changes in Habits')
plt.ylabel('Count')
plt.show()
```



```
[134]: # Plot the relationship between Mood Swings and Mental Health Treatment
plt.figure(figsize=(10, 5))
sns.countplot(x='Mood_Swings', hue='treatment', data=df, palette = 'PuOr')
plt.title('Mood Swings vs. Mental Health Treatment')
plt.xlabel('Mood Swings')
plt.ylabel('Count')
plt.show()
```



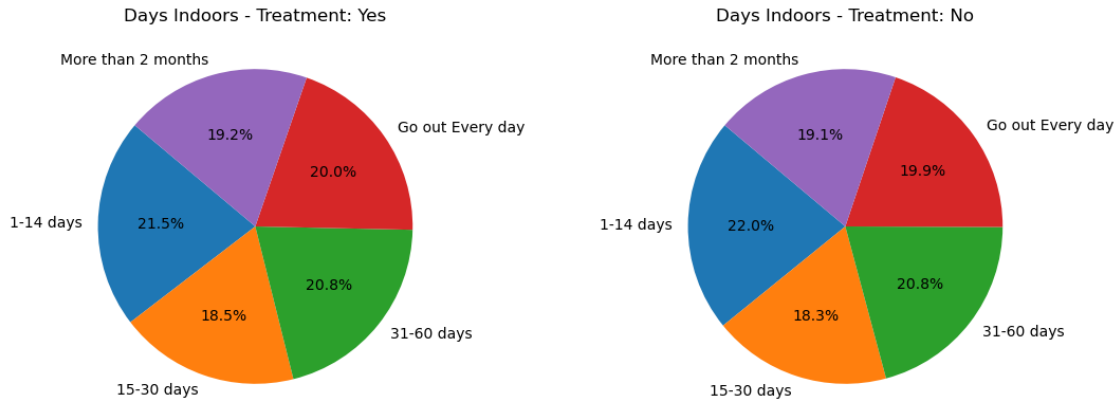
```
[135]: # Aggregate the data
agg_data = df.groupby(['Days_Indoors', 'treatment']).size().
    ↪unstack(fill_value=0)

# Plot the pie chart
fig, axes = plt.subplots(1, 2, figsize=(12, 6), gridspec_kw={'wspace': 0.5})

# Plot for 'Yes' treatment
axes[0].pie(agg_data['Yes'], labels=agg_data.index, autopct='%1.1f%%',
    ↪startangle=140)
axes[0].set_title('Days Indoors - Treatment: Yes')

# Plot for 'No' treatment
axes[1].pie(agg_data['No'], labels=agg_data.index, autopct='%1.1f%%',
    ↪startangle=140)
axes[1].set_title('Days Indoors - Treatment: No')
```

```
# Add the main title below the plots
plt.figtext(0.5, 0.01, 'Days Indoors vs. Mental Health Treatment', ha='center',
↪fontsize=14)
plt.show()
```

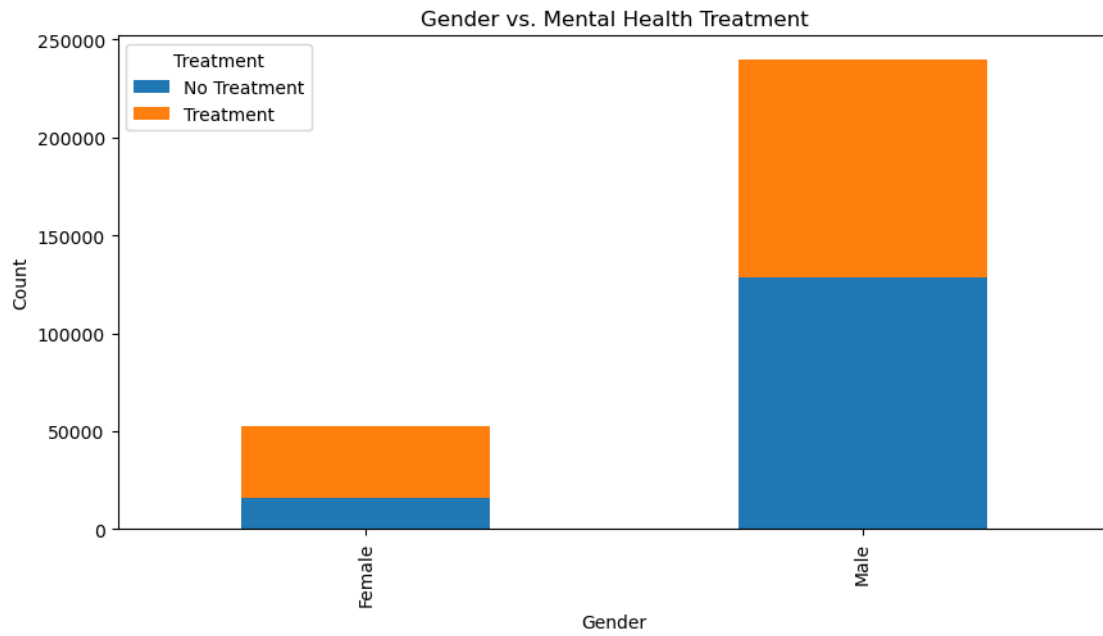


Days Indoors vs. Mental Health Treatment

```
[136]: # Aggregate the data by Gender and treatment
agg_data = df.groupby(['Gender', 'treatment']).size().reset_index(name='counts')

# Pivot the data to have genders as rows and treatments as columns
pivot_data = agg_data.pivot(index='Gender', columns='treatment',
↪values='counts')

# Plot the stacked bar chart
pivot_data.plot(kind='bar', stacked=True, figsize=(10, 5))
plt.title('Gender vs. Mental Health Treatment')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title='Treatment', labels=['No Treatment', 'Treatment'])
plt.show()
```



Analysis By: Shubham A

## 2 Thank you

[ ]: