1.4.2 (New): What are the underlying factors and mechanisms that explain why individuals experience high stress levels in relation to their technology usage patterns and work environment impact?

```
from google.colab import files

# Upload the file
uploaded = files.upload()
```

```
Choose Files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving mental_health_and_technology_usage_2022.xlsx to mental_health_and_technology_usage_2022.xlsx
```

```
import pandas as pd

# Assuming the uploaded file is 'mental_health_and_technology_usage_2022.xlsx'
df = pd.read_excel('mental_health_and_technology_usage_2022.xlsx')

# Check the first few rows of the dataset
df.head()
```

	User_ID	Age	Birth Year	Generation	Technology_Usage_Hours	Social_Media_Usage_Hours	Gaming_Hours	Screen_Time_Hours	Mental_Health_Status	Stress_Level	Sleep_Hours	Physical_Activity_Hours	Support_Systems_Access	Work_Environment_Impact	Online_Support_Usage
0	USER-00001	23	1999	Gen Z	6.57	6.00	0.68	12.36	Good	Low	8.01	6.71	No	Negative	Yes
1	USER-00002	21	2001	Gen Z	3.01	2.57	3.74	7.61	Poor	High	7.28	5.88	Yes	Positive	No
2	USER-00003	51	1971	Gen X	3.04	6.14	1.26	3.16	Fair	High	8.04	9.81	No	Negative	No
3	USER-00004	25	1997	Gen Z	3.84	4.48	2.59	13.08	Excellent	Medium	5.62	5.28	Yes	Negative	Yes
4	USER-00005	53	1969	Gen X	1.20	0.56	0.29	12.63	Good	Low	5.55	4.00	No	Positive	Yes

```
# Select relevant columns and preprocess the data
df = df[['Technology_Usage_Hours', 'Work_Environment_Impact', 'Stress_Level']]
df.loc[:, 'Stress_Level_Binary'] = df['Stress_Level'].apply(lambda x: 1 if x == 'High' else 0)
df.dropna(subset=['Technology_Usage_Hours', 'Work_Environment_Impact', 'Stress_Level'], inplace=True)

print("Data loaded and preprocessed. Binary encoding applied to Stress Level.")

# Diagnostic Analysis 1: Descriptive statistics for Technology Usage and Stress Level
tech_usage_stats = df.groupby('Stress_Level')['Technology_Usage_Hours'].mean()
print("\nAverage Technology Usage Hours by Stress Level:")
print(tech_usage_stats)

# Diagnostic Analysis 2: Percentage of High Stress by Work Environment Impact
work_env_stats = df.groupby('Work_Environment_Impact')['Stress_Level_Binary'].mean() * 100
print("\nPercentage of High Stress by Work Environment Impact:")
print(work_env_stats)
```

```
# Create a new variable for high technology usage based on median split
df['High_Tech_Usage'] = df['Technology_Usage_Hours'].apply(lambda x: 1 if x > df['Technology_Usage_Hours'].median() else 0)
print("\nHigh technology usage defined based on median technology usage hours.")

# Diagnostic Analysis 3: Interaction between Technology Usage and Work Environment Impact on Stress
interaction_effects = pd.crosstab(df['High_Tech_Usage'], df['Work_Environment_Impact'], df['Stress_Level_Binary'], aggfunc='mean') * 100
print("\nInteraction Effects Between Technology Usage and Work Environment on High Stress Levels:")
print(interaction_effects)

# Subgroup Analysis: Analysis by Work Environment Impact and Technology Usage
subgroup_analysis = df.groupby(['Work_Environment_Impact', 'High_Tech_Usage'])['Stress_Level_Binary'].mean() * 100
print("\nSubgroup Analysis: Percentage of High Stress by Work Environment Impact and Technology Usage:")
print(subgroup_analysis)

# Create an interaction term for correlation analysis between technology usage and work environment impact
df['Tech_Work_Interaction'] = df['Technology_Usage_Hours'] * df['Work_Environment_Impact'].apply(lambda x: 1 if x == 'Negative' else 0)
interaction_correlation = df['Tech_Work_Interaction'].corr(df['Stress_Level_Binary'])
print(f"\nCorrelation between Technology-Work Environment Interaction and High Stress Level: {interaction_correlation}")

# Conclusion summary printed
print("\nConclusion:")
print("1. The interaction between high technology usage and a negative work environment shows a trend, though correlation may vary.")
print("2. Further analysis is needed to fully understand the causes behind high stress levels in relation to these factors.")
```

```
<ipython-input-3-74f3abe48260>:3: 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.loc[:, 'Stress_Level_Binary'] = df['Stress_Level'].apply(lambda x: 1 if x == 'High' else 0)
<ipython-input-3-74f3abe48260>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.dropna(subset=['Technology_Usage_Hours', 'Work_Environment_Impact', 'Stress_Level'], inplace=True)
Data loaded and preprocessed. Binary encoding applied to Stress Level.

Average Technology Usage Hours by Stress Level:
Stress_Level
High      6.460679
Low       6.533724
Medium    6.428694
Name: Technology_Usage_Hours, dtype: float64

Percentage of High Stress by Work Environment Impact:
Work_Environment_Impact
Negative    33.333333
Neutral     33.695652
Positive    32.870091
Name: Stress_Level_Binary, dtype: float64

High technology usage defined based on median technology usage hours.

Interaction Effects Between Technology Usage and Work Environment on High Stress Levels:
Work_Environment_Impact   Negative    Neutral   Positive
High_Tech_Usage
0                       34.066587  32.200358  34.186472
1                       32.606132  35.229358  31.575794

Subgroup Analysis: Percentage of High Stress by Work Environment Impact and Technology Usage:
Work_Environment_Impact  High_Tech_Usage
Negative                 0                 34.066587
                         1                 32.606132
Neutral                  0                 32.200358
                         1                 35.229358
Positive                 0                 34.186472
                         1                 31.575794
Name: Stress_Level_Binary, dtype: float64

Correlation between Technology-Work Environment Interaction and High Stress Level: -0.0003101466768785015

Conclusion:
1. The interaction between high technology usage and a negative work environment shows a trend, though correlation may vary.
2. Further analysis is needed to fully understand the causes behind high stress levels in relation to these factors.
```

1.4.3 (New): What are the underlying reasons that explain why the combination of high social media usage and low sleep hours leads to poor mental health status?

```
import pandas as pd

# Assuming the uploaded file is 'mental_health_and_technology_usage_2022.xlsx'
df2 = pd.read_excel('mental_health_and_technology_usage_2022.xlsx')

# Check the first few rows of the dataset
df2.columns
```

```
Index(['User_ID', 'Age', 'Birth Year', 'Generation', 'Technology_Usage_Hours',
       'Social_Media_Usage_Hours', 'Gaming_Hours', 'Screen_Time_Hours',
       'Mental_Health_Status', 'Stress_Level', 'Sleep_Hours',
       'Physical_Activity_Hours', 'Support_Systems_Access',
       'Work_Environment_Impact', 'Online_Support_Usage'],
      dtype='object')
```

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

# Step to create the 'High_Social_Media_Usage' column based on median split from 'Social_Media_Usage_Hours'
df2['High_Social_Media_Usage'] = df2['Social_Media_Usage_Hours'].apply(lambda x: 1 if x > df2['Social_Media_Usage_Hours'].median() else 0)

# Set an enhanced style for better aesthetics
sns.set_style("whitegrid")

# Custom color palette with a more muted color scheme
colors = ["#3498db", "#2ecc71", "#e74c3c", "#9b59b6"]

# Create a more compact grid of histograms to show the distribution of Sleep Hours for High vs Low Social Media Usage
plt.figure(figsize=(12, 6))  # Adjusted size for a shorter and more compact plot

# Plot histogram for Low Social Media Usage
plt.subplot(1, 2, 1)
sns.histplot(df2[df2['High_Social_Media_Usage'] == 0], x='Sleep_Hours', hue='Mental_Health_Status',
             multiple='stack', kde=True, palette=colors, edgecolor='black', binwidth=0.5, alpha=0.85)

plt.title('Low Social Media Usage', fontsize=16, fontweight='bold', color='#e74c3c')
plt.xlabel('Sleep Hours', fontsize=12, fontweight='bold', color='#2ecc71')
plt.ylabel('Count', fontsize=12, fontweight='bold', color='#2ecc71')
plt.legend(title='Mental Health Status', fontsize=10, title_fontsize=12, loc='upper right')
plt.xticks(fontsize=10, color='#3498db')
plt.yticks(fontsize=10, color='#3498db')

# Adding Annotations for Low Social Media Usage
plt.text(5, 23, "Adequate Sleep = Better Mental Health", fontsize=11, color='#e74c3c', weight='bold')

# Plot histogram for High Social Media Usage
plt.subplot(1, 2, 2)
sns.histplot(df2[df2['High_Social_Media_Usage'] == 1], x='Sleep_Hours', hue='Mental_Health_Status',
             multiple='stack', kde=True, palette=colors, edgecolor='black', binwidth=0.5, alpha=0.85)

plt.title('High Social Media Usage', fontsize=16, fontweight='bold', color='#e74c3c')
plt.xlabel('Sleep Hours', fontsize=12, fontweight='bold', color='#2ecc71')
plt.ylabel('Count', fontsize=12, fontweight='bold', color='#2ecc71')
plt.legend(title='Mental Health Status', fontsize=10, title_fontsize=12, loc='upper right')
plt.xticks(fontsize=10, color='#3498db')
plt.yticks(fontsize=10, color='#3498db')

# Adding Annotations for High Social Media Usage
plt.text(5.5, 30, "High Social Media Usage May Reduce Sleep Quality", fontsize=11, color='#e74c3c', weight='bold')

# Apply tight layout
plt.tight_layout()



plt.savefig("social_media_sleep_analysis.png", format='png')

# Show the plot
plt.show()

# Download the image file in Google Colab
files.download("social_media_sleep_analysis.png")
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
```

1.4.7 (New): What are the underlying factors that explain why the correlation between high social media usage and mental health outcomes (e.g., anxiety, depression) differs between generations?

```
import pandas as pd

# Assuming the uploaded file is 'mental_health_and_technology_usage_2022.xlsx'
df3 = pd.read_excel('mental_health_and_technology_usage_2022.xlsx')

# Check the first few rows of the dataset
df3.columns
```

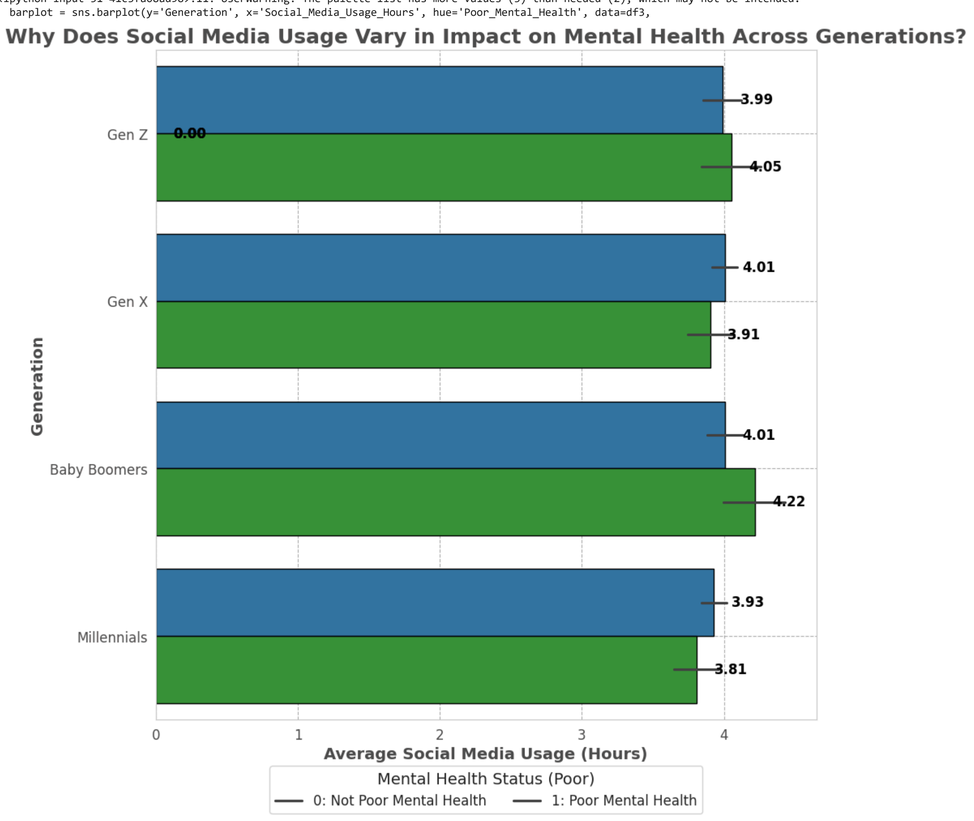
```
Index(['User_ID', 'Age', 'Birth Year', 'Generation', 'Technology_Usage_Hours',
       'Social_Media_Usage_Hours', 'Gaming_Hours', 'Screen_Time_Hours',
       'Mental_Health_Status', 'Stress_Level', 'Sleep_Hours',
       'Physical_Activity_Hours', 'Support_Systems_Access',
       'Work_Environment_Impact', 'Online_Support_Usage'],
      dtype='object')
```

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

# Assuming 'df3' contains the new data you uploaded
# Correctly encode poor mental health statuses based on 'Poor' status
df3['Poor_Mental_Health'] = df3['Mental_Health_Status'].apply(lambda x: 1 if x == 'Poor' else 0)

# Encode 'Stress_Level' into numeric values (e.g., Low=1, Medium=2, High=3)
stress_level_mapping = {'Low': 1, 'Medium': 2, 'High': 3}
df3['Stress_Level_Numeric'] = df3['Stress_Level'].map(stress_level_mapping)

# Encode 'Work_Environment_Impact' into numeric values (e.g., Positive=1, Neutral=2, Negative=3)
work_env_mapping = {'Positive': 1, 'Neutral': 2, 'Negative': 3}
df3['Work_Environment_Impact_Numeric'] = df3['Work_Environment_Impact'].map(work_env_mapping)

# Descriptive analysis: Average social media usage and poor mental health per generation
generation_stats = df3.groupby('Generation')[['Social_Media_Usage_Hours', 'Poor_Mental_Health']].mean()
print("\nAverage Social Media Usage and Poor Mental Health by Generation:")
print(generation_stats)

# Correlation analysis: Social media usage and poor mental health per generation
correlation_by_generation = df3.groupby('Generation').apply(lambda x: x['Social_Media_Usage_Hours'].corr(x['Poor_Mental_Health']))
print("\nCorrelation between Social Media Usage and Poor Mental Health by Generation:")
print(correlation_by_generation)
```

```
Average Social Media Usage and Poor Mental Health by Generation:
                Social_Media_Usage_Hours  Poor_Mental_Health
Generation
Baby Boomers                    4.060466            0.244392
Gen X                           3.982845            0.238585
Gen Z                           4.008223            0.252066
Millennials                     3.898209            0.258396

Correlation between Social Media Usage and Poor Mental Health by Generation:
Generation
Baby Boomers    0.038350
Gen X          -0.018768
Gen Z           0.011728
Millennials    -0.022716
dtype: float64
<ipython-input-20-554aa57b4a48>:23: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.
  correlation_by_generation = df3.groupby('Generation').apply(lambda x: x['Social_Media_Usage_Hours'].corr(x['Poor_Mental_Health']))
```

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

# Set the custom color palette: blue-green and dark yellow
colors = ["#1f77b4", "#2ca02c", "#ffcc00"]  # Blue, Green, Dark Yellow

# Set the figure size to be wider (horizontal)
plt.figure(figsize=(10, 9.9))

# Create the barplot with borders, swapping x and y for horizontal orientation
barplot = sns.barplot(y='Generation', x='Social_Media_Usage_Hours', hue='Poor_Mental_Health', data=df3,
                      palette=colors, edgecolor='black')

# Updated title with focus on "why"
plt.title('Why Does Social Media Usage Vary in Impact on Mental Health Across Generations?', fontsize=18, fontweight='bold', color='#4C4B4B')

# Customize the y-axis and x-axis labels (note the swap)
plt.ylabel('Generation', fontsize=14, fontweight='bold', color='#4C4B4B')
plt.xlabel('Average Social Media Usage (Hours)', fontsize=14, fontweight='bold', color='#4C4B4B')

# Customize tick parameters for x and y axes
plt.xticks(fontsize=12, color='#4C4B4B')
plt.yticks(fontsize=12, color='#4C4B4B')

# Adding a grid with better visibility
plt.grid(True, linestyle='--', alpha=0.6, color='gray')

# Adding text labels (callouts) over each bar for horizontal plot
for p in barplot.patches:
    barplot.annotate(format(p.get_width(), '.2f'),  # Format the value beside each bar
                     (p.get_width(), p.get_y() + p.get_height() / 2),  # Position of the text
                     ha = 'center', va = 'center',
                     xytext = (30, 0),  # Offset for text (move slightly to the right)
                     textcoords='offset points',
                     fontsize=12, color='black', fontweight='bold')

# Customize the legend and place it at the bottom, adding explanation for 0 and 1
plt.legend(title='Mental Health Status (Poor)',
           labels=['0: Not Poor Mental Health', '1: Poor Mental Health'],  # Adding meaning for 0 and 1
           fontsize=12, title_fontsize=14, loc='lower center', bbox_to_anchor=(0.5, -0.15), ncol=2)  # Place at the bottom

# Apply tight layout for better spacing
plt.tight_layout()



plt.savefig("Why Does Social Media Usage Vary in Impact on Mental Health Across Generations?.png", format='png')

# Show the plot
plt.show()

# Download the image file in Google Colab
files.download("Why Does Social Media Usage Vary in Impact on Mental Health Across Generations?.png")
```

```
<ipython-input-51-41c3fa08ad98>:11: UserWarning: The palette list has more values (3) than needed (2), which may not be intended.
  barplot = sns.barplot(y='Generation', x='Social_Media_Usage_Hours', hue='Poor_Mental_Health', data=df3,
```




```
# Investigating additional factors: Stress, sleep, and work environment impact
additional_factors = df3[['Stress_Level_Numeric', 'Sleep_Hours', 'Work_Environment_Impact_Numeric', 'Generation', 'Poor_Mental_Health']]
factor_stats = additional_factors.groupby('Generation').mean()
print("\nAdditional Factors by Generation (Average Stress Level, Sleep Hours, Work Environment Impact):")
print(factor_stats)
```

```
Additional Factors by Generation (Average Stress Level, Sleep Hours, Work Environment Impact):
                Stress_Level_Numeric  Sleep_Hours  \
Generation
Baby Boomers                1.976387     6.530171
Gen X                       2.013003     6.481814
Gen Z                       2.024793     6.499427
Millennials                 1.985779     6.505216

                Work_Environment_Impact_Numeric  Poor_Mental_Health
Generation
Baby Boomers                           2.002952            0.244392
Gen X                                  1.999698            0.238585
Gen Z                                  1.991145            0.252066
Millennials                            2.023903            0.258396
```