In [16]:
```
import pandas as pd

# Load the dataset
df = pd.read_csv('Final Survey Data1.csv')
```

In [17]:
```
df
```

Out[17]:

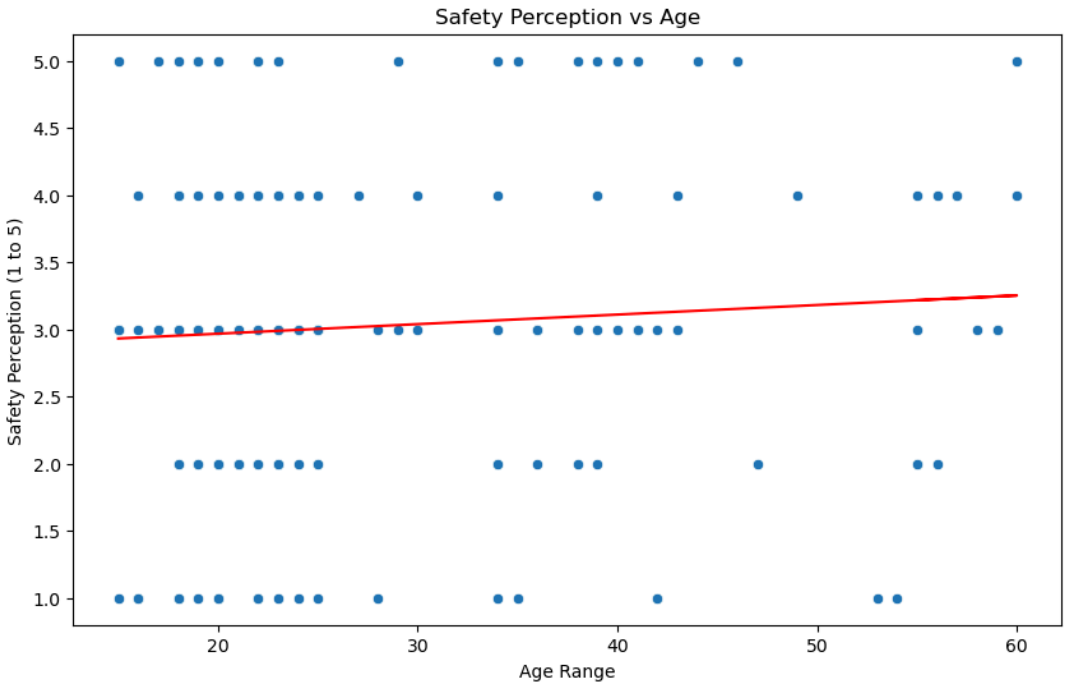
	What is your age range?	Which region do you belong to?	Gender	How often do you think women face harassment in public places?	Have you or someone you know experienced harassment or violence in public or private spaces?	On a scale of 1 to 5, how safe do you think public spaces are for women in your area?	Do you believe societal norms and gender stereotypes contribute to unsafe environments for women?	In y opini do cultu attitude y commu suppor hin wome safe
0	15	Urban	Female	Daily	No	3	Strongly agree	Hin
1	17	Urban	Female	Weekly	Yes	3	Agree	Neu
2	17	Semi-Urban	Male	Daily	Yes	3	Agree	Supp
3	17	Urban	Female	Daily	No	3	Agree	Supp
4	15	Urban	Female	Daily	Yes	3	Strongly agree	Neu
...	...	...	...	...	...	...	...	
168	56	Urban	Male	Daily	Yes	2	Agree	Hin
169	55	Urban	Female	Daily	No	2	Agree	Supp
170	57	Urban	Male	Rarely	No	4	Disagree	Supp
171	60	Urban	Female	Never	No	5	Agree	Supp
172	55	Urban	Male	Monthly	No	4	Neutral	Supp

173 rows × 27 columns

In [21]:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset

# Prepare the data
# Make sure to replace 'What_is_your_age_range' and the safety rating c
X = df[['What is your age range?']]  # Update this if the column name i
y = df['On a scale of 1 to 5, how safe do you think public spaces are f

# Create and fit the model
model = LinearRegression()
model.fit(X, y)

# Print the coefficient
print('Coefficient:', model.coef_)

# Visualization: Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=X.squeeze(), y=y)  # .squeeze() to convert to 1D arra
plt.plot(X, model.predict(X), color='red')  # Regression line
plt.title('Safety Perception vs Age')
plt.xlabel('Age Range')
plt.ylabel('Safety Perception (1 to 5)')
plt.show()
```

```
Coefficient: [0.00712216]
```

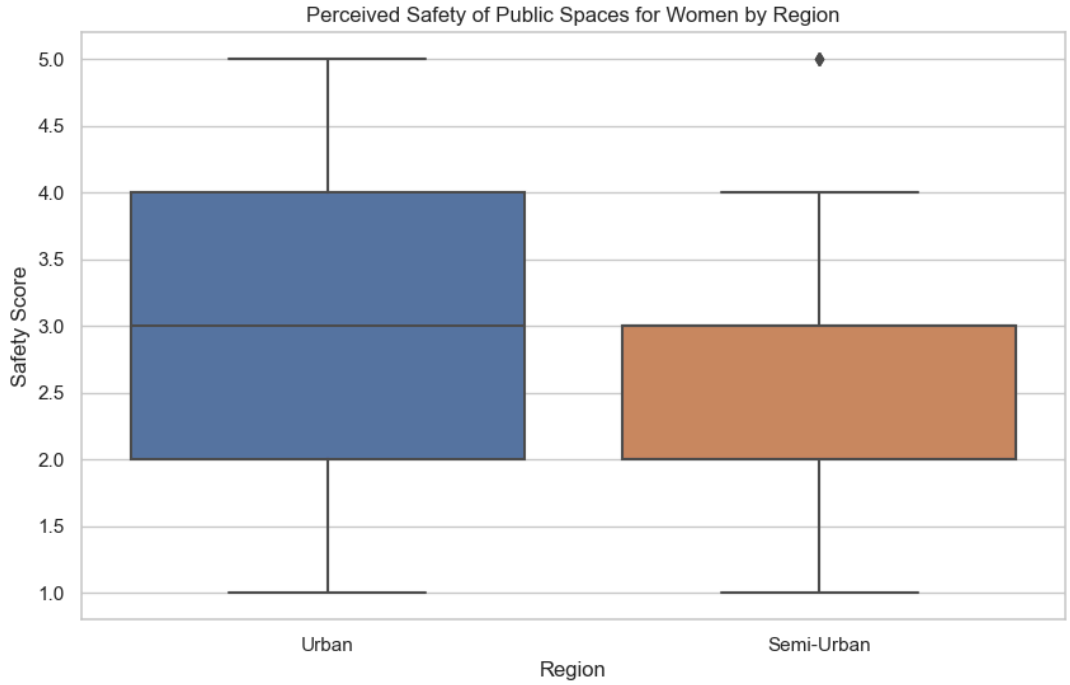


In [ ]:
```
#Since the P-value is greater than the common significance level of 0.0
```

In [28]:

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('Final Survey Data1.csv')

# Hypothesis: There is a significant difference in the perceived safety

# Extract relevant columns
df['Region'] = df['Which region do you belong to?']
df['Safety_Score'] = df['On a scale of 1 to 5, how safe do you think pu

# Filter data for Urban and Semi-Urban regions
urban_safety = df[df['Region'] == 'Urban']['Safety_Score'].dropna()
semi_urban_safety = df[df['Region'] == 'Semi-Urban']['Safety_Score'].dr

# Perform t-test
t_stat, p_value = stats.ttest_ind(urban_safety, semi_urban_safety)

# Print the results
print('T-statistic:', t_stat)
print('P-value:', p_value)

# Set the style of the visualization
sns.set(style="whitegrid")

# Create a box plot
plt.figure(figsize=(10, 6))
sns.boxplot(x='Region', y='Safety_Score', data=df[df['Region'].isin(['U

# Add title and labels
plt.title('Perceived Safety of Public Spaces for Women by Region')
plt.xlabel('Region')
plt.ylabel('Safety Score')

# Show the plot
plt.show()

# Display summary statistics
print("\
Summary Statistics:")
print(df[df['Region'].isin(['Urban', 'Semi-Urban'])].groupby('Region')[
```

```
T-statistic: 0.9019915670483174
P-value: 0.3683922504358804
```



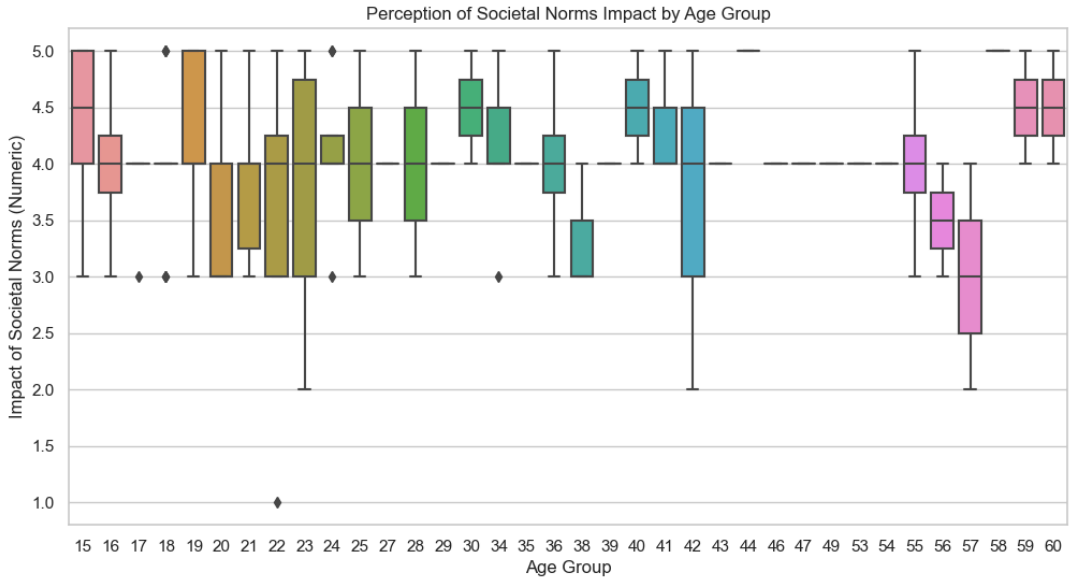

```
Summary Statistics:
            count      mean       std  min  25%  50%  75%  max
Region
Semi-Urban   25.0  2.840000  1.247664  1.0  2.0  3.0  3.0  5.0
Urban       140.0  3.064286  1.126596  1.0  2.0  3.0  4.0  5.0
```

In [ ]:
```
# high p-value suggests no significant difference in perceptions across
```

In [30]:

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

# Hypothesis: There is a significant difference in the perception of so

# Extract relevant columns
df['Age_Group'] = df['What is your age range?']
df['Societal_Norms_Impact'] = df['Do you believe societal norms and gen

# Map categorical responses to numerical values
df['Societal_Norms_Impact_Numeric'] = df['Societal_Norms_Impact'].map({
    'Strongly disagree': 1,
    'Disagree': 2,
    'Neutral': 3,
    'Agree': 4,
    'Strongly agree': 5
})

# Perform ANOVA with the numeric values
groups = df.groupby('Age_Group')['Societal_Norms_Impact_Numeric'].apply

# Perform ANOVA test
f_stat, p_value = stats.f_oneway(*groups)

# Print the results
print('F-statistic:', f_stat)
print('P-value:', p_value)

# Set the style of the visualization
sns.set(style="whitegrid")

# Create a box plot
plt.figure(figsize=(12, 6))
sns.boxplot(x='Age_Group', y='Societal_Norms_Impact_Numeric', data=df)

# Add title and labels
plt.title('Perception of Societal Norms Impact by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Impact of Societal Norms (Numeric)')

# Show the plot
plt.show()

# Display summary statistics
print("\
Summary Statistics:")
print(df.groupby('Age_Group')['Societal_Norms_Impact_Numeric'].describe
```

```
F-statistic: 0.6417961903510908
P-value: 0.9362318695103276
```



Summary Statistics:

Age_Group	count	mean	std	min	25%	50%	75%	max
15	6.0	4.333333	0.816497	3.0	4.00	4.5	5.00	5.0
16	4.0	4.000000	0.816497	3.0	3.75	4.0	4.25	5.0
17	6.0	3.833333	0.408248	3.0	4.00	4.0	4.00	4.0
18	16.0	4.000000	0.632456	3.0	4.00	4.0	4.00	5.0
19	14.0	4.285714	0.611250	3.0	4.00	4.0	5.00	5.0
20	11.0	3.636364	0.809040	3.0	3.00	3.0	4.00	5.0
21	10.0	3.800000	0.632456	3.0	3.25	4.0	4.00	5.0
22	16.0	3.750000	1.064581	1.0	3.00	4.0	4.25	5.0
23	14.0	3.857143	0.949262	2.0	3.00	4.0	4.75	5.0
24	8.0	4.125000	0.640870	3.0	4.00	4.0	4.25	5.0
25	11.0	4.000000	0.774597	3.0	3.50	4.0	4.50	5.0
27	1.0	4.000000	NaN	4.0	4.00	4.0	4.00	4.0
28	2.0	4.000000	1.414214	3.0	3.50	4.0	4.50	5.0
29	2.0	4.000000	0.000000	4.0	4.00	4.0	4.00	4.0
30	2.0	4.500000	0.707107	4.0	4.25	4.5	4.75	5.0
34	7.0	4.142857	0.690066	3.0	4.00	4.0	4.50	5.0
35	2.0	4.000000	0.000000	4.0	4.00	4.0	4.00	4.0
36	4.0	4.000000	0.816497	3.0	3.75	4.0	4.25	5.0
38	3.0	3.333333	0.577350	3.0	3.00	3.0	3.50	4.0
39	5.0	4.000000	0.000000	4.0	4.00	4.0	4.00	4.0
40	2.0	4.500000	0.707107	4.0	4.25	4.5	4.75	5.0
41	3.0	4.333333	0.577350	4.0	4.00	4.0	4.50	5.0
42	3.0	3.666667	1.527525	2.0	3.00	4.0	4.50	5.0
43	2.0	4.000000	0.000000	4.0	4.00	4.0	4.00	4.0
44	1.0	5.000000	NaN	5.0	5.00	5.0	5.00	5.0
46	1.0	4.000000	NaN	4.0	4.00	4.0	4.00	4.0
47	1.0	4.000000	NaN	4.0	4.00	4.0	4.00	4.0
49	1.0	4.000000	NaN	4.0	4.00	4.0	4.00	4.0
53	1.0	4.000000	NaN	4.0	4.00	4.0	4.00	4.0
54	1.0	4.000000	NaN	4.0	4.00	4.0	4.00	4.0
55	4.0	4.000000	0.816497	3.0	3.75	4.0	4.25	5.0
56	2.0	3.500000	0.707107	3.0	3.25	3.5	3.75	4.0
57	2.0	3.000000	1.414214	2.0	2.50	3.0	3.50	4.0
58	1.0	5.000000	NaN	5.0	5.00	5.0	5.00	5.0
59	2.0	4.500000	0.707107	4.0	4.25	4.5	4.75	5.0
60	2.0	4.500000	0.707107	4.0	4.25	4.5	4.75	5.0

In [36]:

```
# Map categorical responses to numerical values
harassment_mapping = {'Never': 1, 'Rarely': 2, 'Monthly': 3, 'Weekly':
df['Harassment Frequency Numeric'] = df['How often do you think women f

# Select the numeric columns for the F-test
safety_scores = df['On a scale of 1 to 5, how safe do you think public
harassment_scores_numeric = df['Harassment Frequency Numeric']

# Perform F-test
f_statistic, p_value = stats.f_oneway(safety_scores, harassment_scores_

# Display hypothesis and results
print("Hypothesis:")
print("H0: There is no significant difference between perceived safety
print("H1: There is a significant difference between perceived safety s
print(f"\
F-statistic: {f_statistic}")
print(f"p-value: {p_value}")

# Create and display a box plot
plt.figure(figsize=(10, 6))
data = [safety_scores, harassment_scores_numeric]
labels = ['Perceived Safety', 'Harassment Frequency']
plt.boxplot(data, labels=labels)
plt.title("Comparison of Perceived Safety and Harassment Frequency Scor
plt.ylabel("Score")
plt.show()

# Calculate and display mean scores
print("\
Mean scores:")
print(f"Perceived Safety: {safety_scores.mean():.2f}")
print(f"Harassment Frequency: {harassment_scores_numeric.mean():.2f}")
```

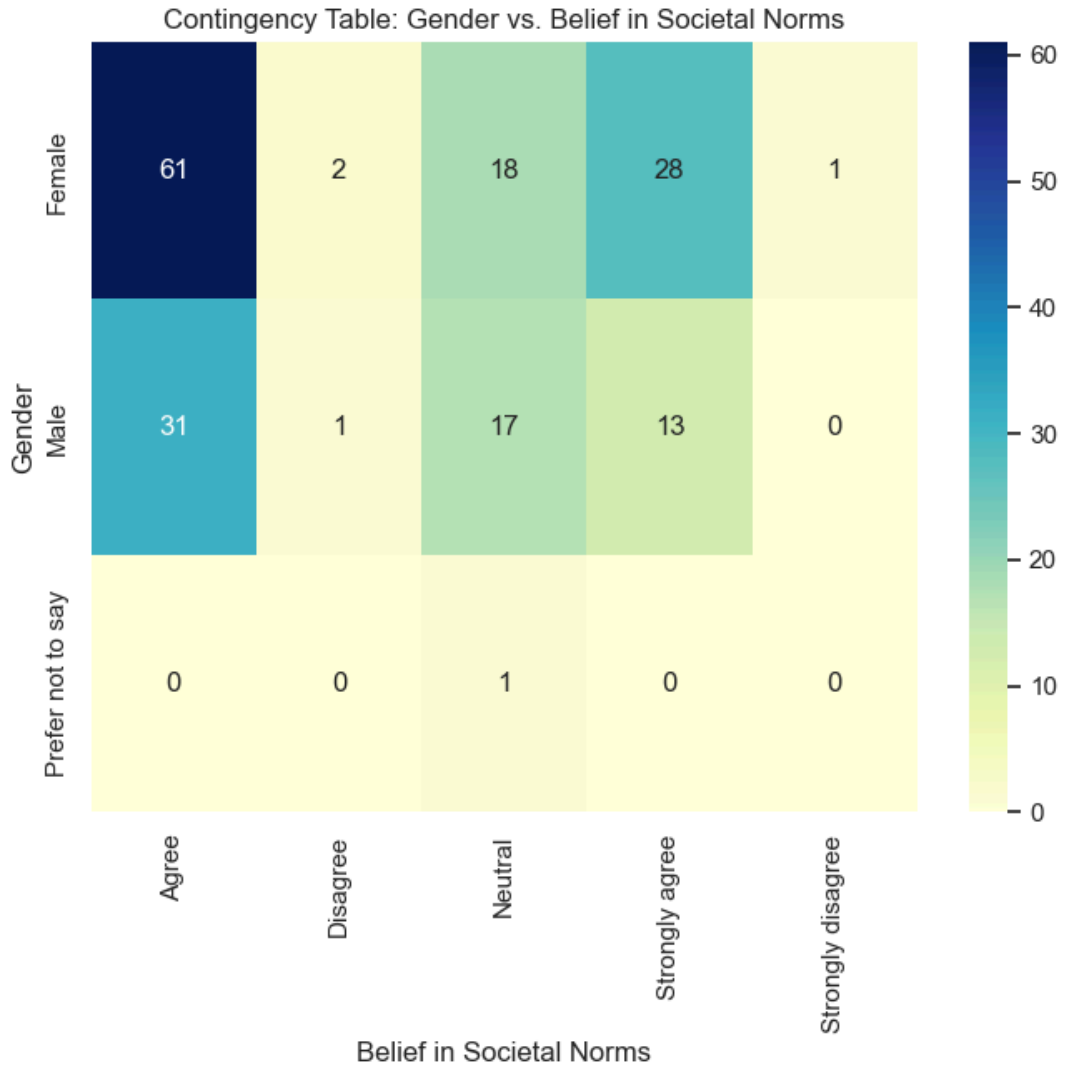
```
Hypothesis:
H0: There is no significant difference between perceived safety scores
and harassment frequency scores.
H1: There is a significant difference between perceived safety scores
and harassment frequency scores.
F-statistic: 56.80232181675223
p-value: 4.3090717367602903e-13
```


```
Mean scores:
Perceived Safety: 3.02
Harassment Frequency: 4.00
```

In [37]:

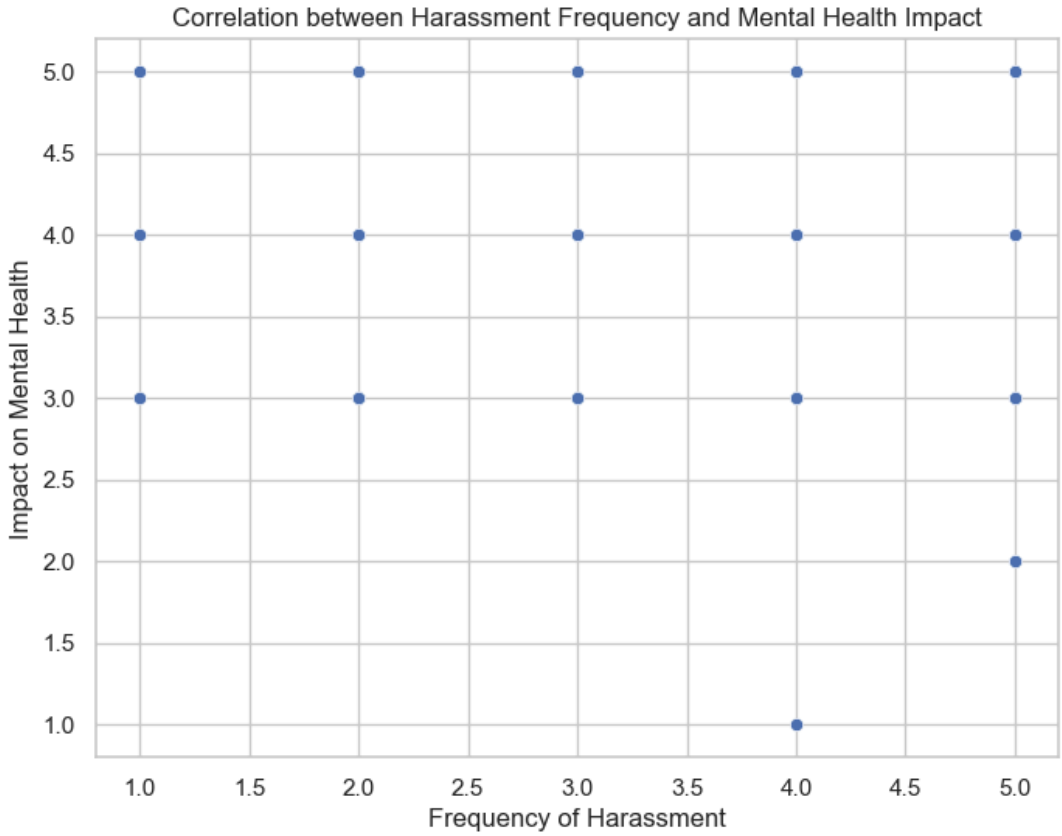
```
# Perform Chi-square test on two categorical columns
from scipy.stats import chi2_contingency

# Create a contingency table
contingency_table = pd.crosstab(df['Gender'], df['Do you believe societ

# Perform the Chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Display hypothesis and results
print("Hypothesis:")
print("H0: There is no association between gender and belief in societa
print("H1: There is an association between gender and belief in societa
print(f"\
Chi-square statistic: {chi2}")
print(f"p-value: {p}")

# Visualize the contingency table
plt.figure(figsize=(8, 6))
sns.heatmap(contingency_table, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Contingency Table: Gender vs. Belief in Societal Norms')
plt.xlabel('Belief in Societal Norms')
plt.ylabel('Gender')
plt.show()
```

```
Hypothesis:
H0: There is no association between gender and belief in societal norm
s contributing to unsafe environments.
H1: There is an association between gender and belief in societal norm
s contributing to unsafe environments.
Chi-square statistic: 7.291902234813499
p-value: 0.5054909865781201
```


Contingency Table: Gender vs. Belief in Societal Norms
Gender
Female
Male
Prefer not to say
61
2
18
28
1
31
1
17
13
0
0
0
1
0
0
Agree
Disagree
Neutral
Strongly agree
Strongly disagree
Belief in Societal Norms
60
50
40
30
20
10
0

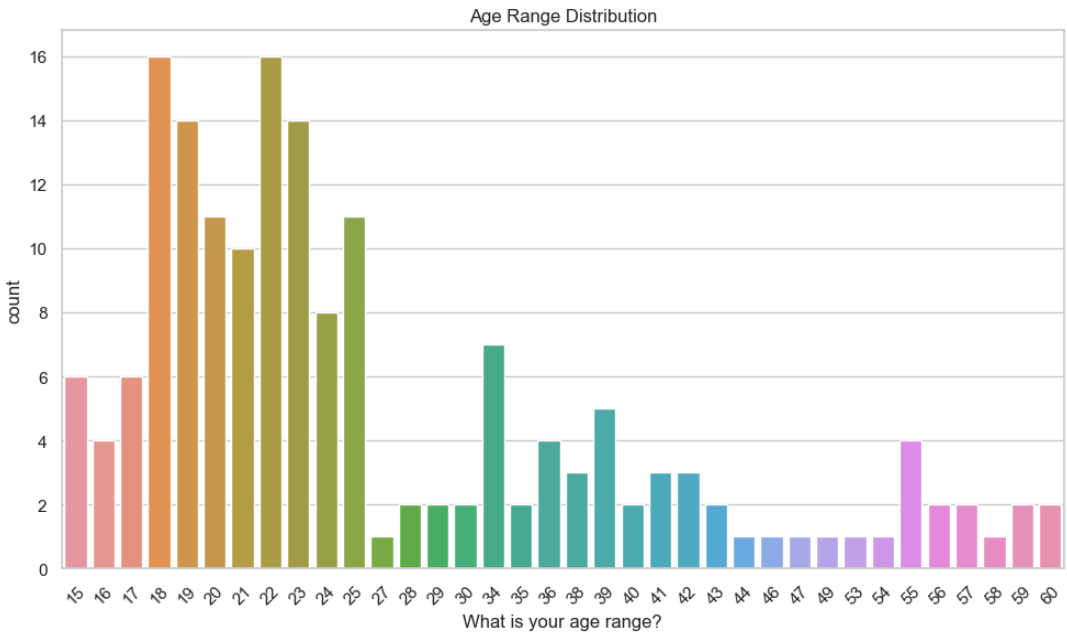
In [40]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import pearsonr



# Selecting relevant columns for Pearson's correlation
harassment_frequency_column = 'How often do you think women face harass
mental_health_impact_column = "Do you believe concerns about safety aff

# Dropping rows with NaN values in the selected columns
filtered_df = df[[harassment_frequency_column, mental_health_impact_col

# Converting categorical data to numerical for correlation
harassment_mapping = {'Daily': 5, 'Weekly': 4, 'Monthly': 3, 'Rarely':
mental_health_mapping = {'Strongly agree': 5, 'Agree': 4, 'Neutral': 3,

filtered_df[harassment_frequency_column] = filtered_df[harassment_frequ
filtered_df[mental_health_impact_column] = filtered_df[mental_health_im

# Calculating Pearson's correlation
correlation, p_value = pearsonr(filtered_df[harassment_frequency_column

# Plotting the correlation
plt.figure(figsize=(8, 6))
sns.scatterplot(x=filtered_df[harassment_frequency_column], y=filtered_
plt.title('Correlation between Harassment Frequency and Mental Health I
plt.xlabel('Frequency of Harassment')
plt.ylabel('Impact on Mental Health')
plt.grid(True)
plt.show()

# Displaying the correlation and p-value
print('Pearson Correlation Coefficient:', correlation)
print('P-value:', p_value)
```



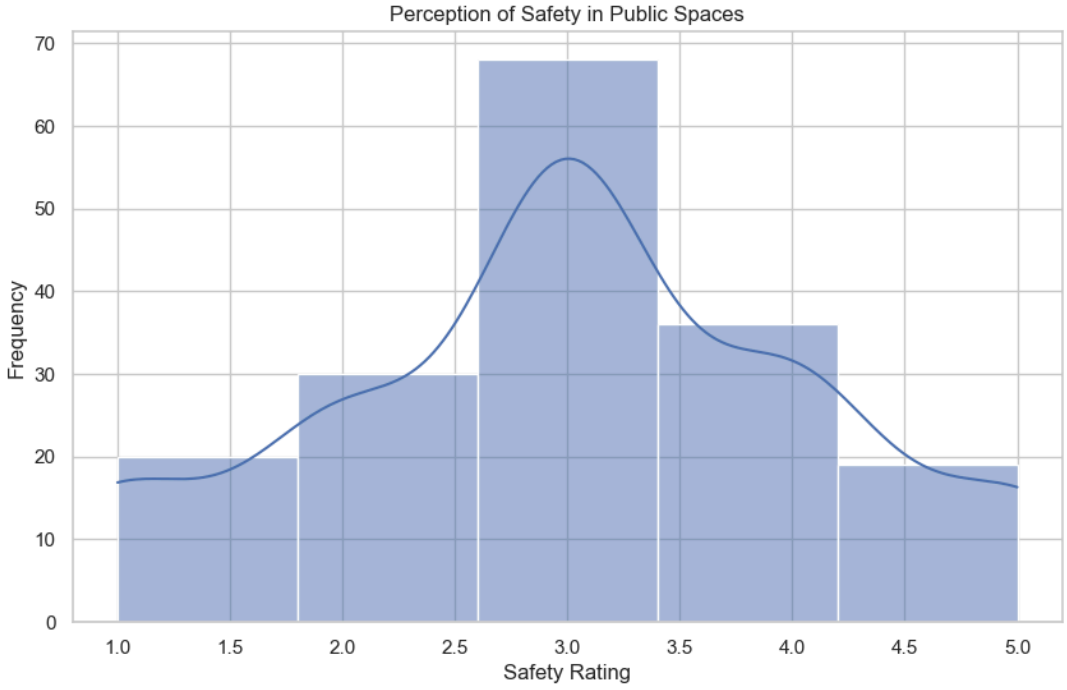

```
Pearson Correlation Coefficient: 0.08210798602909242
P-value: 0.2828463561419668
```

In [41]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='What is your age range?')
plt.title('Age Range Distribution')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [42]:

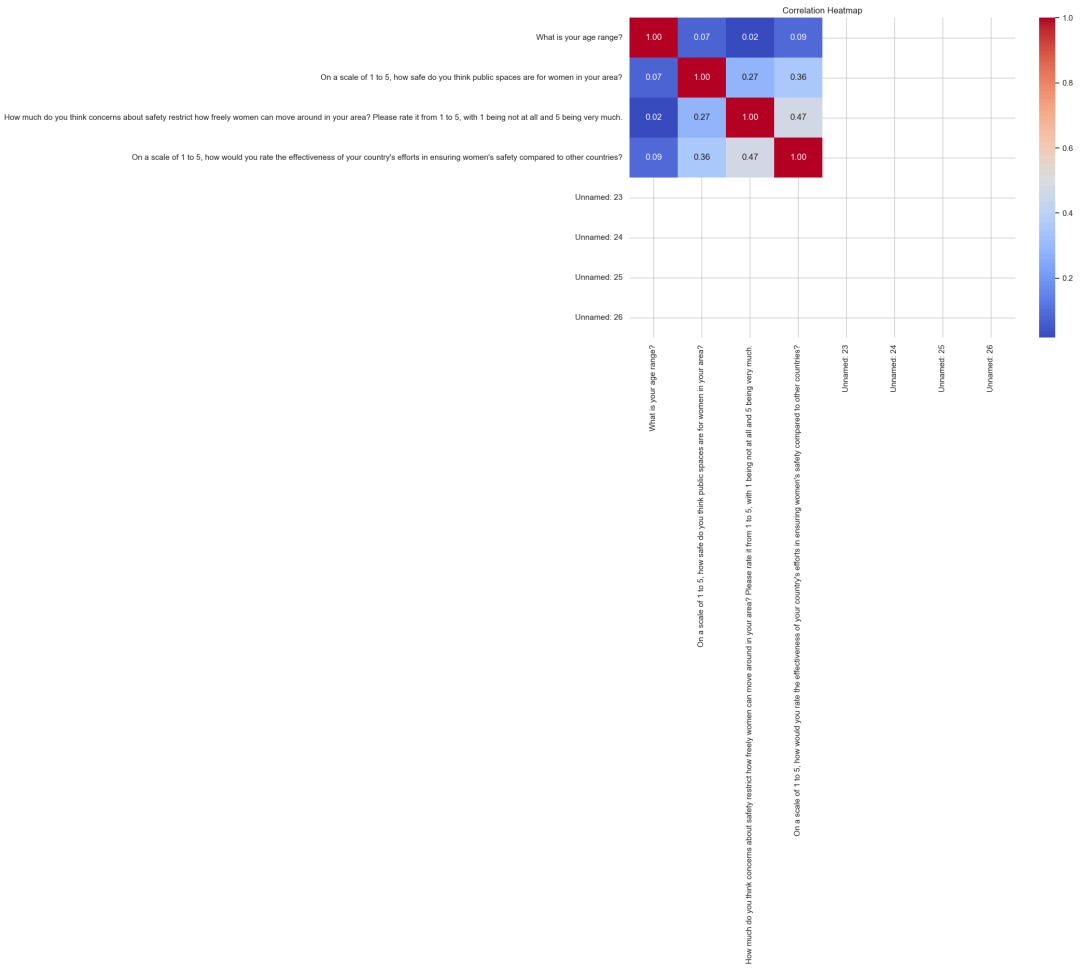
```
plt.figure(figsize=(10, 6))
sns.histplot(df['On a scale of 1 to 5, how safe do you think public spa
plt.title('Perception of Safety in Public Spaces')
plt.xlabel('Safety Rating')
plt.ylabel('Frequency')
plt.show()
```



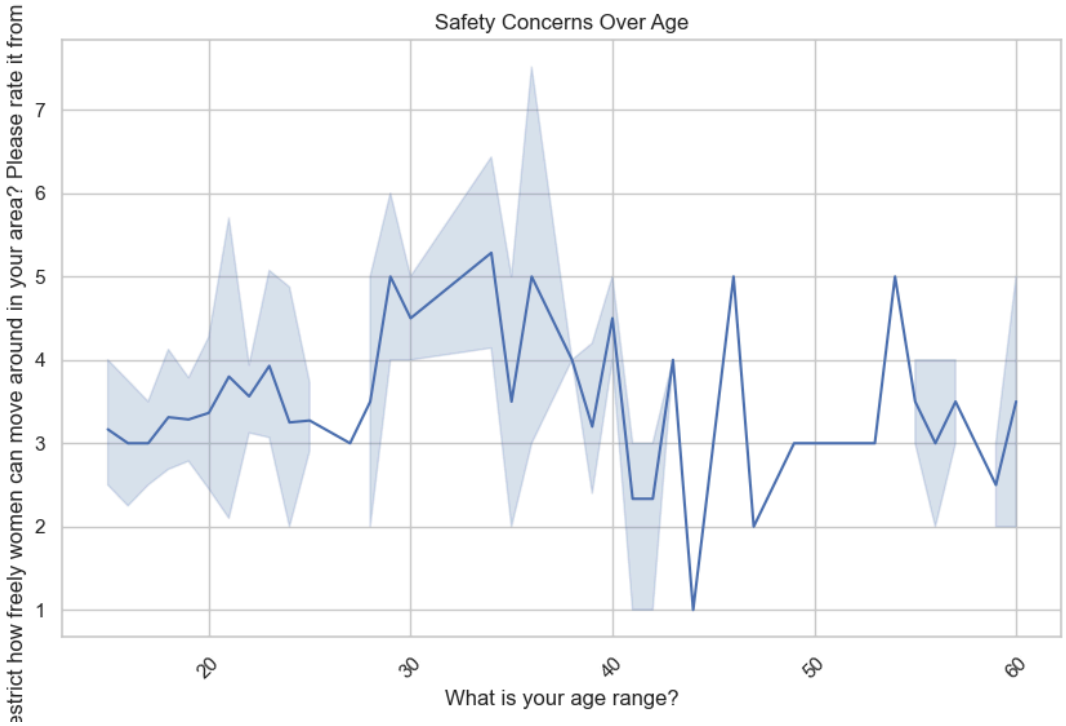
In [44]:
```
plt.figure(figsize=(12, 8))
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_67060\143386923.py:2: Futur
eWarning: The default value of numeric_only in DataFrame.corr is depre
cated. In a future version, it will default to False. Select only vali
d columns or specify the value of numeric_only to silence this warnin
g.
  correlation_matrix = df.corr()
```

Out[44]: Text(0.5, 1.0, 'Correlation Heatmap')



In [45]:
```
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x='What is your age range?', y='How much do you t
plt.title('Safety Concerns Over Age')
plt.xticks(rotation=45)
plt.show()
```



In [ ]:
```

```