In [37]:
```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
```

In [3]:
```
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	ID	Sex	Age	Marital Status	Ethnic Group	Religion	Educational Level	Witnessed Violence	Type of Violence witnessed	Co Enf Re
0	1001	Female	31-45	Single	Ewe	Christian	Tertiary	Yes	Emotional Abuse	I
1	1002	Male	31-45	Married	Akan	Christian	Tertiary	Yes	Verbal Abuse	I
2	1003	Male	31-45	Single	Akan	Christian	Tertiary	Yes	Verbal Abuse,Physical Abuse	
3	1004	Male	18-30	Single	Ewe	Christian	Tertiary	Yes	Physical Abuse	I
4	1005	Female	31-45	Single	Akan	Christian	Tertiary	No	NaN	I

5 rows × 27 columns

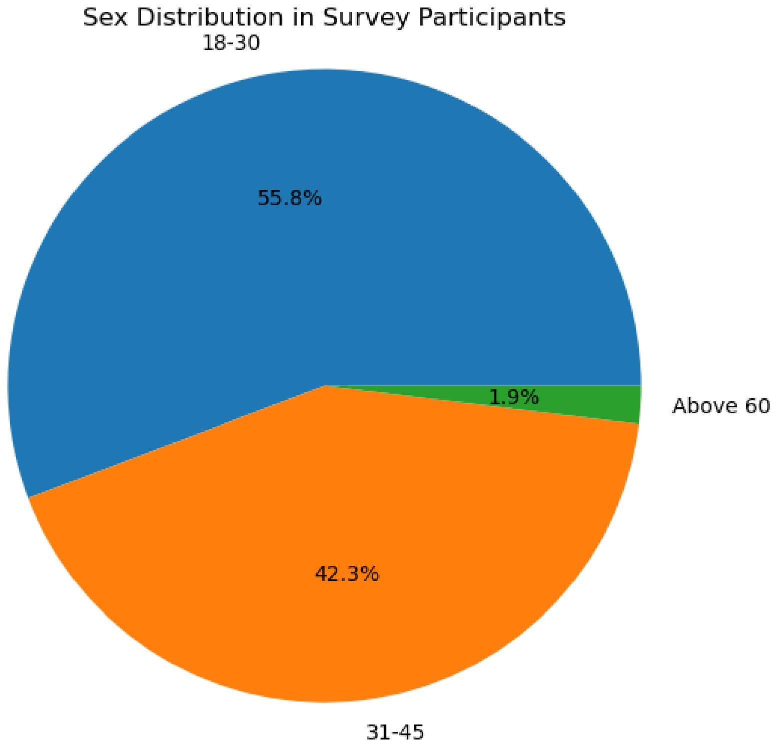
In [5]:
```
df.shape
```

Out[5]:
```
(52, 27)
```

In [6]:
```
df["Sex"].value_counts()
```

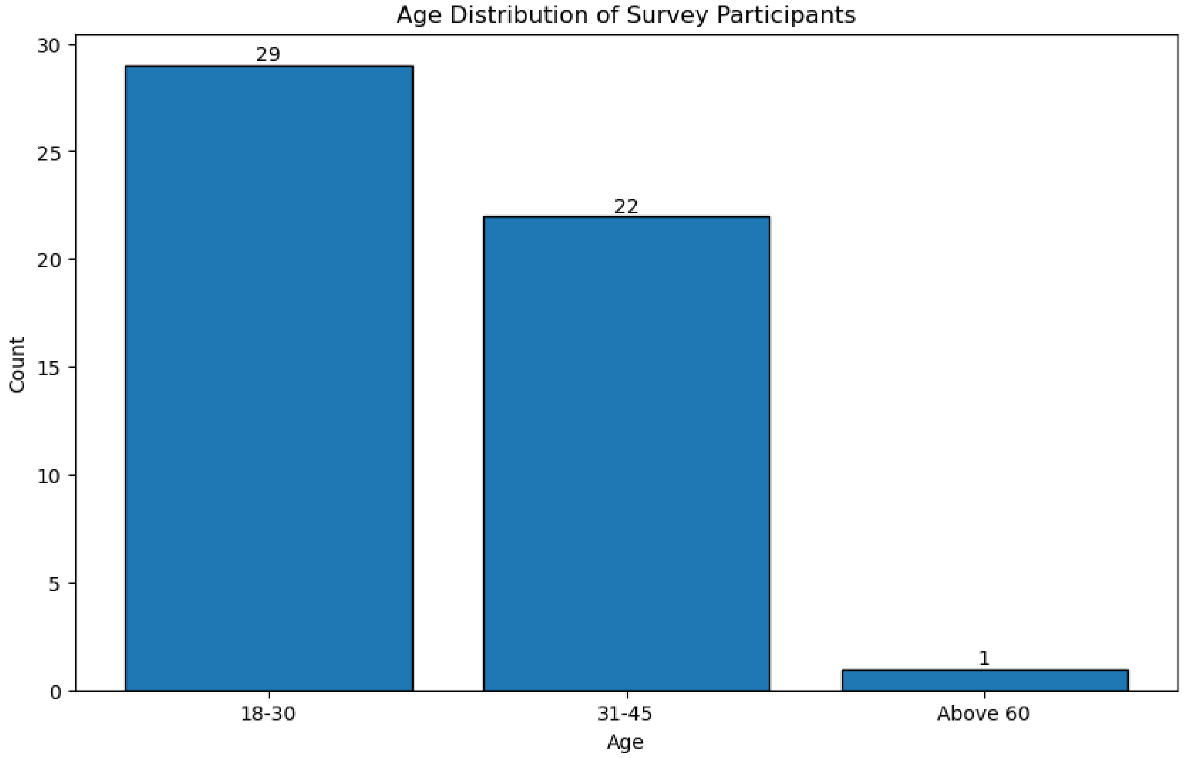
Out[6]:
```
Male      33
Female    19
Name: Sex, dtype: int64
```

In [7]:
```
age_count = df["Age"].value_counts()

plt.figure(figsize=(10, 6))
plt.pie(age_count, labels=age_count.index, autopct='%1.1f%%')
plt.title('Sex Distribution in Survey Participants')
plt.axis('equal')
plt.savefig('plot1.jpg', dpi=300)
plt.show()
age_count
```



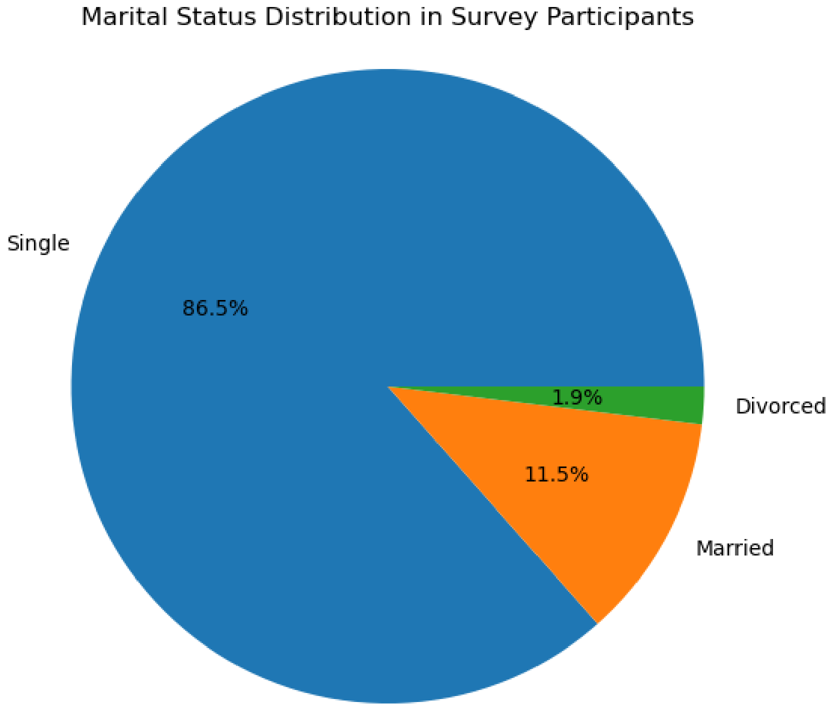
Out[7]:
```
18-30       29
31-45       22
Above 60     1
Name: Age, dtype: int64
```

In [8]:
```
age_data = df['Age']
age_counts = age_data.value_counts().sort_index()

plt.figure(figsize=(10, 6))
plt.bar(age_counts.index, age_counts.values, edgecolor='black')

# Adding count values on top of each bar
for i, count in enumerate(age_counts.values):
    plt.text(age_counts.index[i], count, str(count), ha='center', va='bottom

plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Age Distribution of Survey Participants')
plt.xticks()
plt.savefig('plot2.png', dpi=300)
plt.show()
```



In [9]:

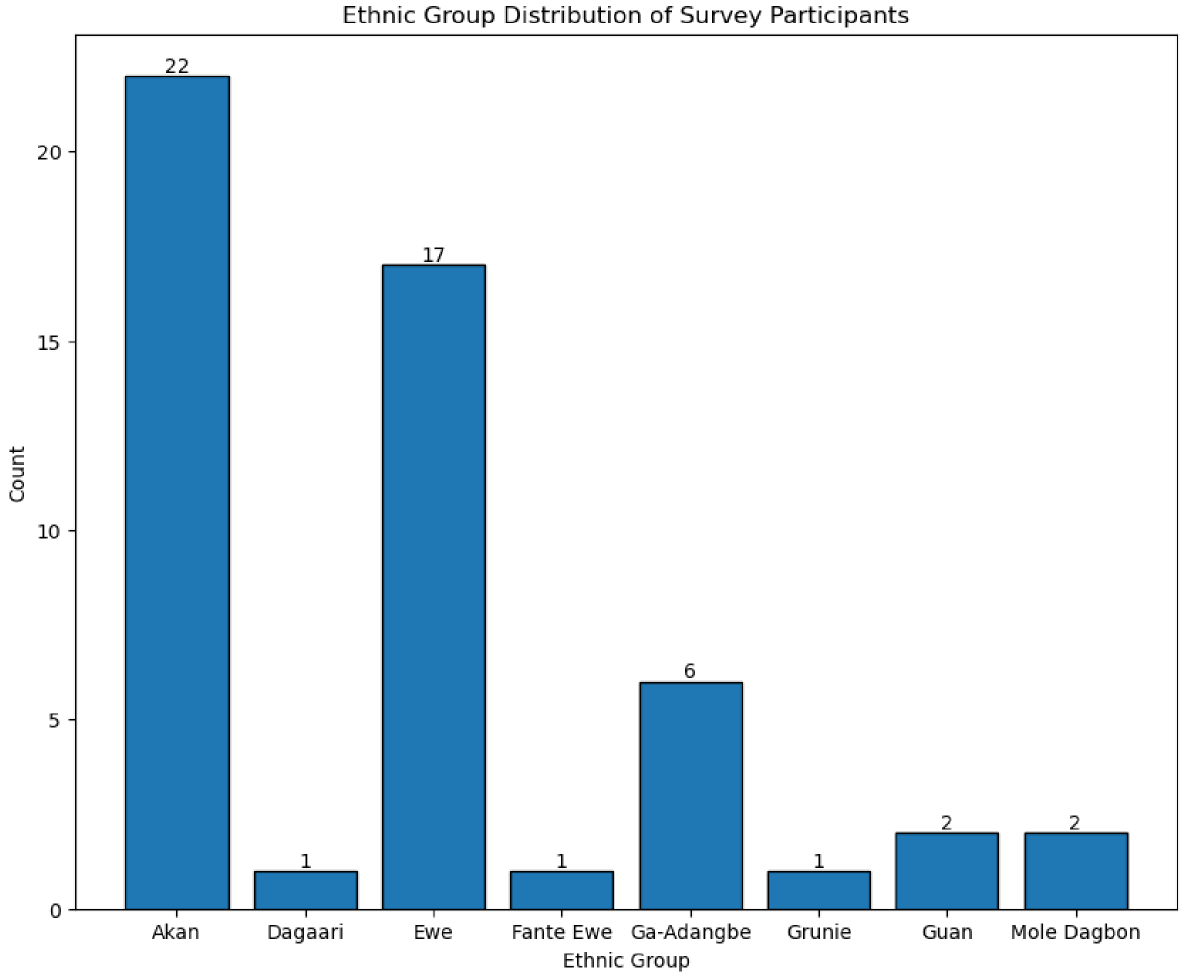
```
marital_count = df["Marital Status"].value_counts()

plt.figure(figsize=(10, 6))
plt.pie(marital_count, labels=marital_count.index, autopct='%1.1f%%')
plt.title('Marital Status Distribution in Survey Participants')
plt.axis('equal')
plt.savefig('plot3.png', dpi=300)
plt.show()
```



In [10]:

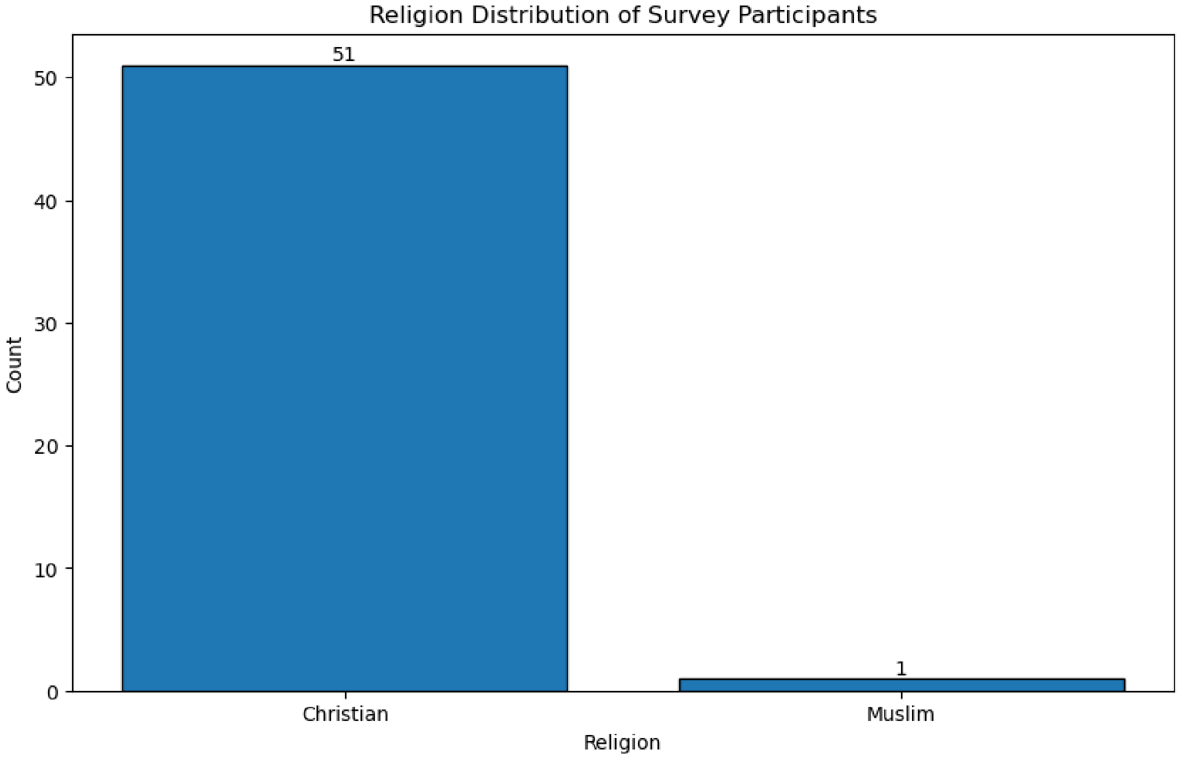
```
ethnic_data = df['Ethnic Group']
ethnic_counts = ethnic_data.value_counts().sort_index()

plt.figure(figsize=(10, 8))
plt.bar(ethnic_counts.index, ethnic_counts.values, edgecolor='black')
```

```
# Adding count values on top of each bar
for i, count in enumerate(ethnic_counts.values):
    plt.text(ethnic_counts.index[i], count, str(count), ha='center', va='bot

plt.xlabel('Ethnic Group')
plt.ylabel('Count')
plt.title('Ethnic Group Distribution of Survey Participants')
plt.xticks()
plt.savefig('plot4.png', dpi=300)
plt.show()
```



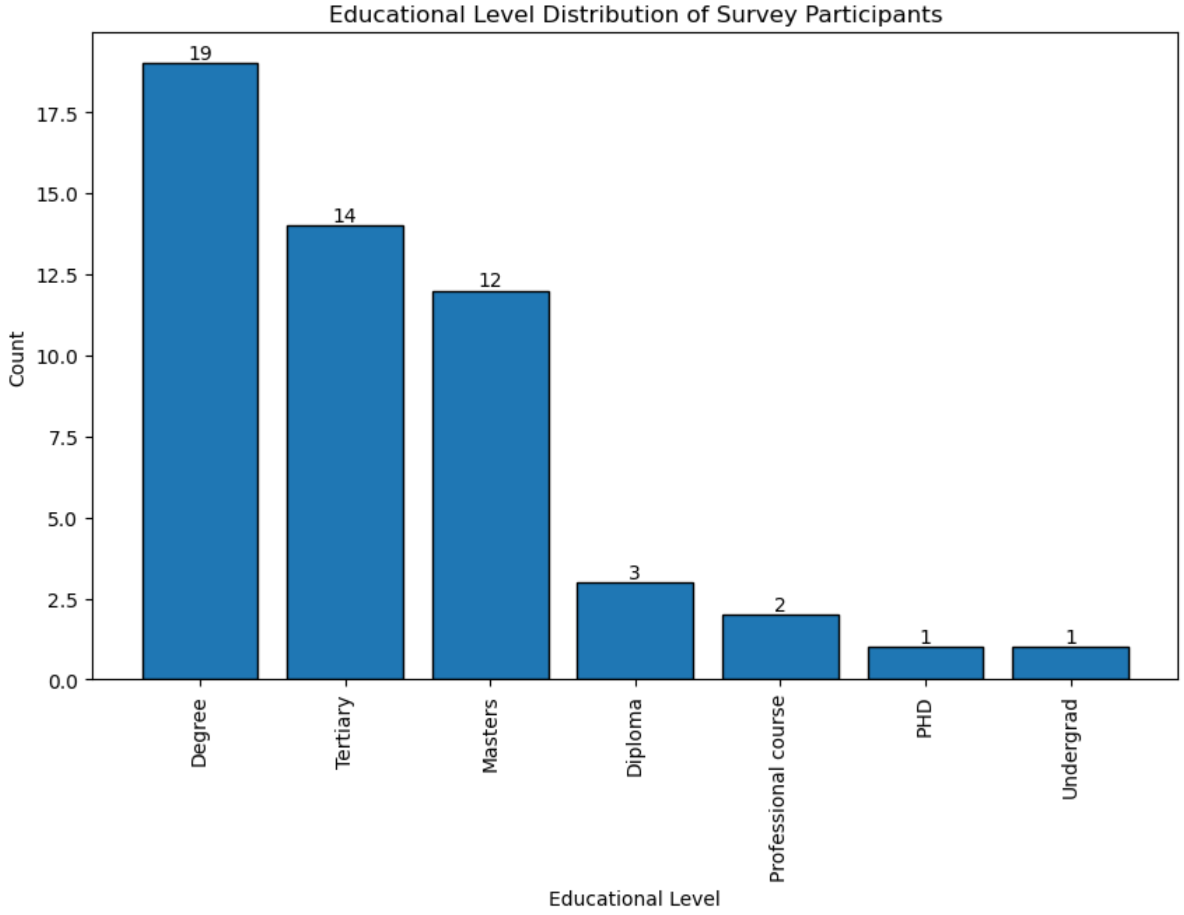
In [11]:

```
religion_data = df['Religion']
religion_counts = religion_data.value_counts()

plt.figure(figsize=(10, 6))
plt.bar(religion_counts.index, religion_counts.values, edgecolor='black')

# Adding count values on top of each bar
for i, count in enumerate(religion_counts.values):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.xlabel('Religion')
plt.ylabel('Count')
plt.title('Religion Distribution of Survey Participants')
plt.xticks()
plt.savefig('plot5.png', dpi=300)
plt.show()
```

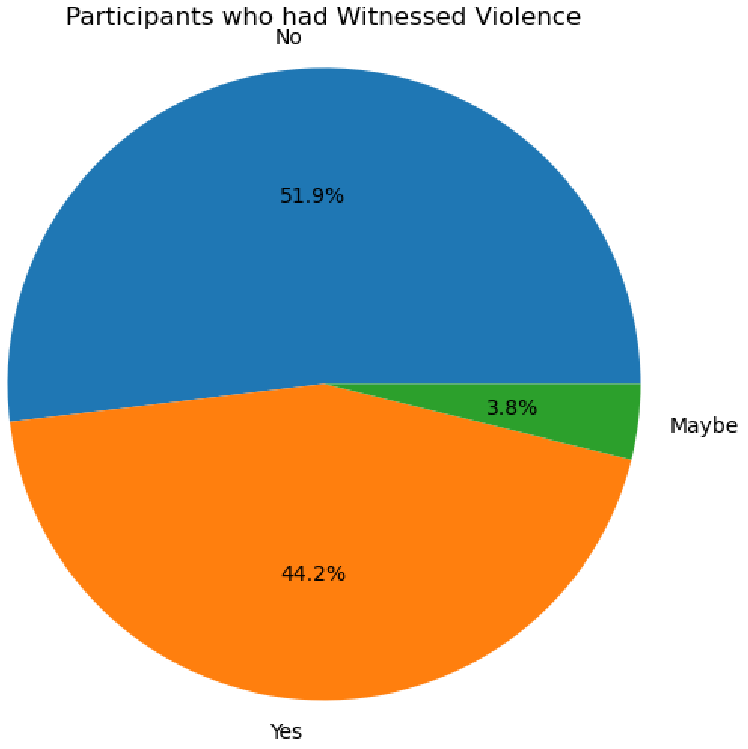


In [12]:

```
education_data = df['Educational Level']
education_counts = education_data.value_counts()

plt.figure(figsize=(10, 6))
plt.bar(education_counts.index, education_counts.values, edgecolor='black')

# Adding count values on top of each bar
for i, count in enumerate(education_counts.values):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.xlabel('Educational Level')
plt.ylabel('Count')
plt.title('Educational Level Distribution of Survey Participants')
plt.xticks(rotation=90)
plt.savefig('plot6.png', dpi=300)
plt.show()
```

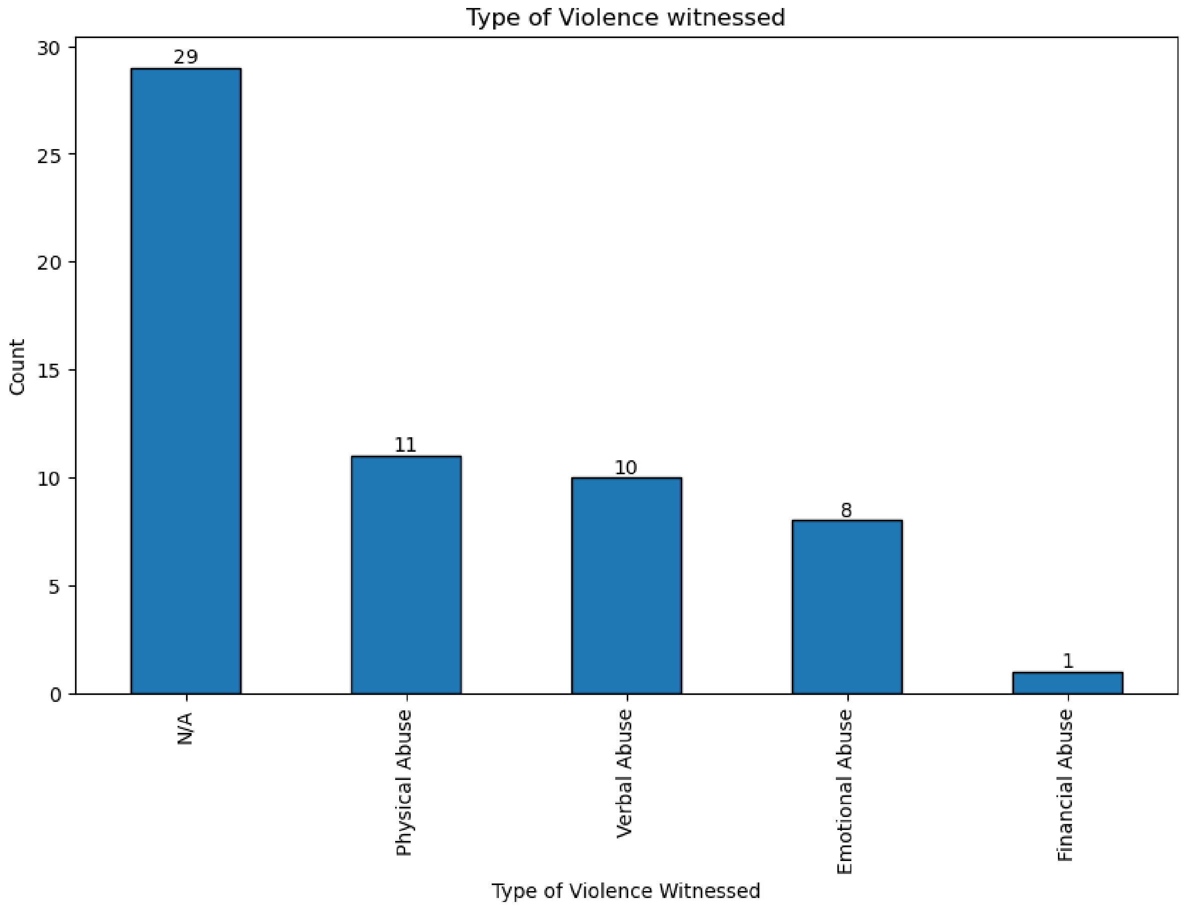

Educational Level Distribution of Survey Participants

In [13]:
```
witnessed_violence_data = df['Witnessed Violence']
witnessed_violence_counts = witnessed_violence_data.value_counts()

plt.figure(figsize=(10, 6))
plt.pie(witnessed_violence_counts, labels=witnessed_violence_counts.index, a
plt.title('Participants who had Witnessed Violence')
plt.axis('equal')
plt.savefig('plot7.png', dpi=300)
plt.show()
```


Participants who had Witnessed Violence

In [14]:

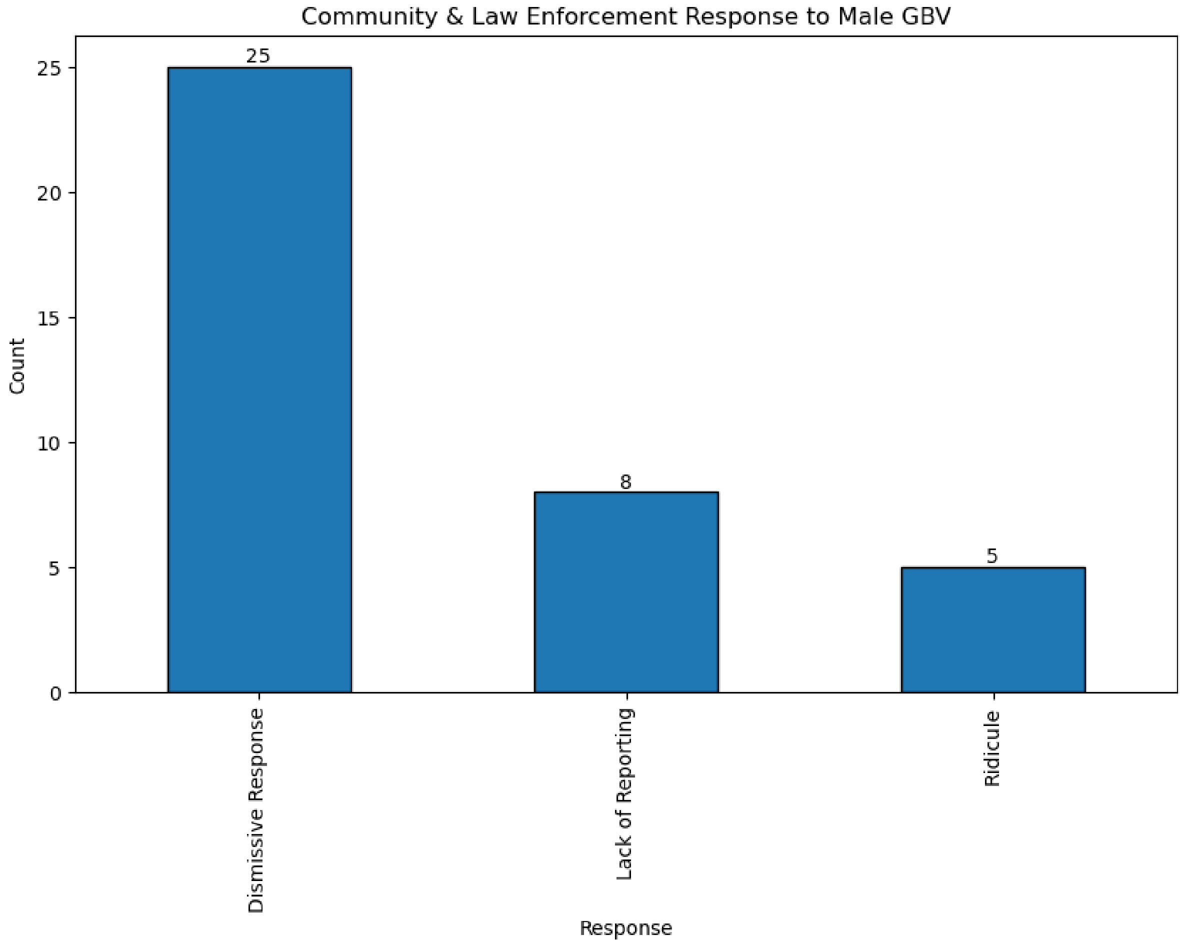
```
witnessed_violence = df['Type of Violence witnessed'].str.split(",", expand=
witnessed_violence_counts = witnessed_violence.stack().value_counts()

# Merge duplicate categories
witnessed_violence_counts['Physical Abuse'] += witnessed_violence_counts.pop

# Include N/A category in the counts
n_a_count = len(df[df['Type of Violence witnessed'].isna()])
witnessed_violence_counts['N/A'] = n_a_count

# Sort the counts in descending order
witnessed_violence_counts = witnessed_violence_counts.sort_values(ascending=

# Plot the bar graph
plt.figure(figsize=(10, 6))
witnessed_violence_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Type of Violence Witnessed')
plt.ylabel('Count')
plt.title('Type of Violence witnessed')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(witnessed_violence_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot8.png', dpi=300)
plt.show()
```



In [15]:

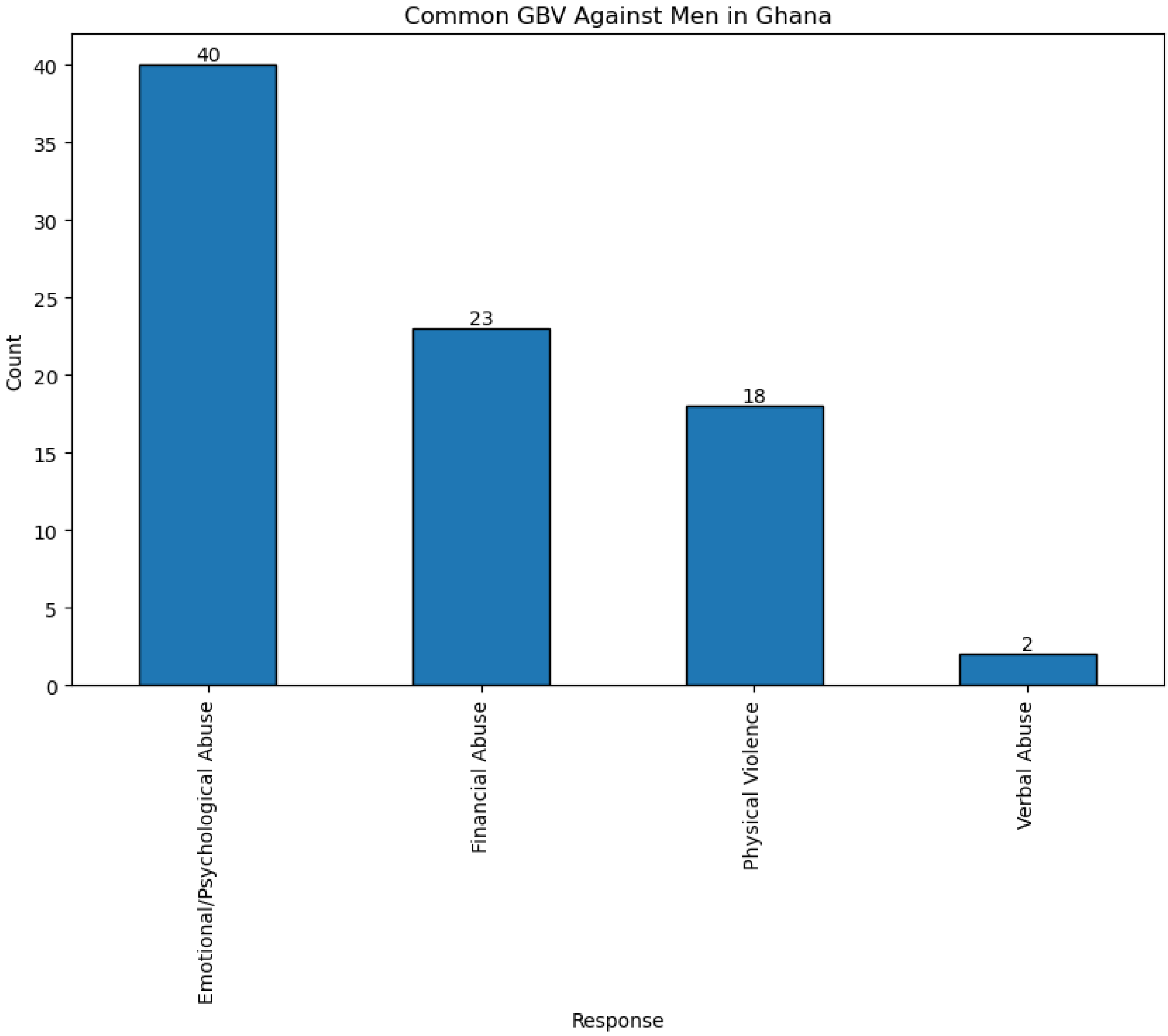
```
response_data = df['Community & Law Enforcement Response to Male GBV']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
```

```
response_counts = responses.value_counts()

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Community & Law Enforcement Response to Male GBV')
plt.xticks(rotation=90)

for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot9.png', dpi=300)
plt.show()
```



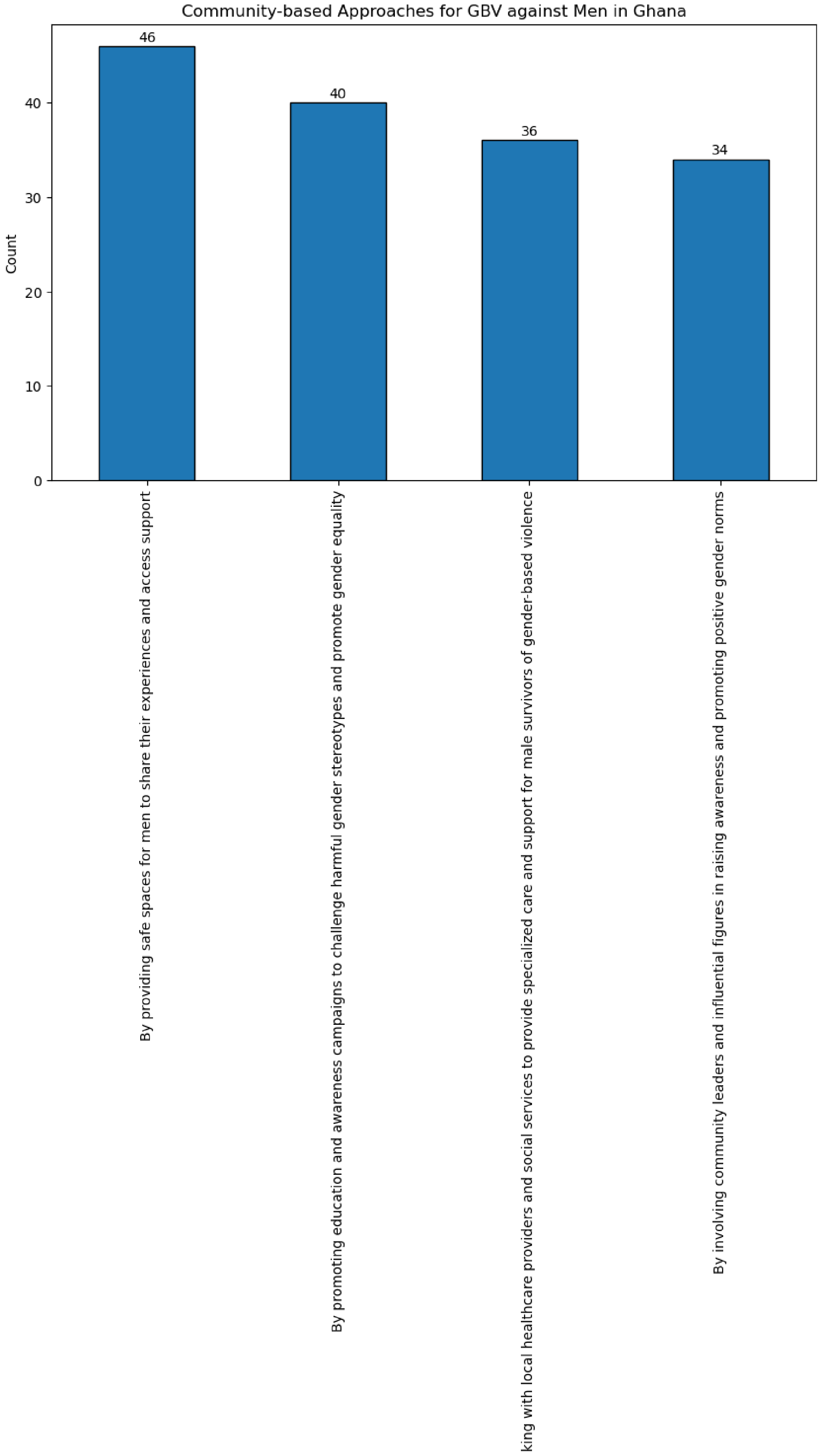
In [16]:

```
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")
response_data = df['Common GBV Against Men in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()

# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Common GBV Against Men in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot10.png', dpi=300)
plt.show()
```



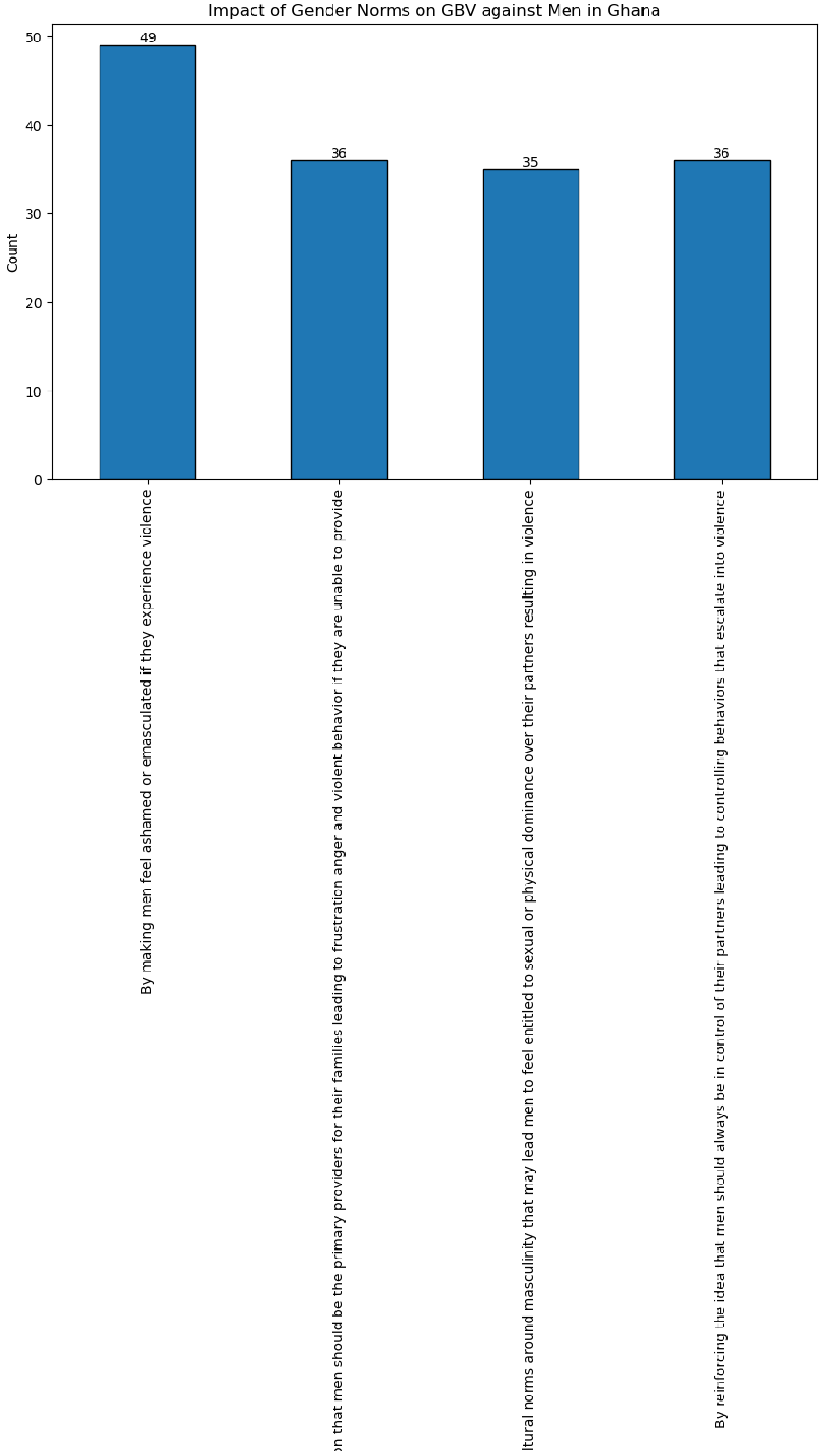
```
In [17]: response_data = df['Community-based Approaches for GBV against Men in Ghana'

         # Split the responses and stack them into a single column
         responses = response_data.str.split(",", expand=True).stack()

         # Count the occurrences of each response
         response_counts = responses.value_counts()

         # Merge all duplicate categories
         response_counts = response_counts.groupby(response_counts.index.str.strip())

         # Sort the counts in descending order
         response_counts = response_counts.sort_values(ascending=False)


         # Plot the bar graph
         plt.figure(figsize=(10, 6))
         response_counts.plot(kind='bar', edgecolor='black')
         plt.xlabel('Response')
         plt.ylabel('Count')
         plt.title('Community-based Approaches for GBV against Men in Ghana')
```

```
# Add count values on top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count + 0.5, str(count), ha='center')

plt.xticks(rotation=90)
plt.savefig('plot11.png', dpi=300)
plt.show()
```


Community-based Approaches for GBV against Men in Ghana
Count
0
10
20
30
40
46
40
36
34
By providing safe spaces for men to share their experiences and access support
By promoting education and awareness campaigns to challenge harmful gender stereotypes and promote gender equality
king with local healthcare providers and social services to provide specialized care and support for male survivors of gender-based violence
By involving community leaders and influential figures in raising awareness and promoting positive gender norms

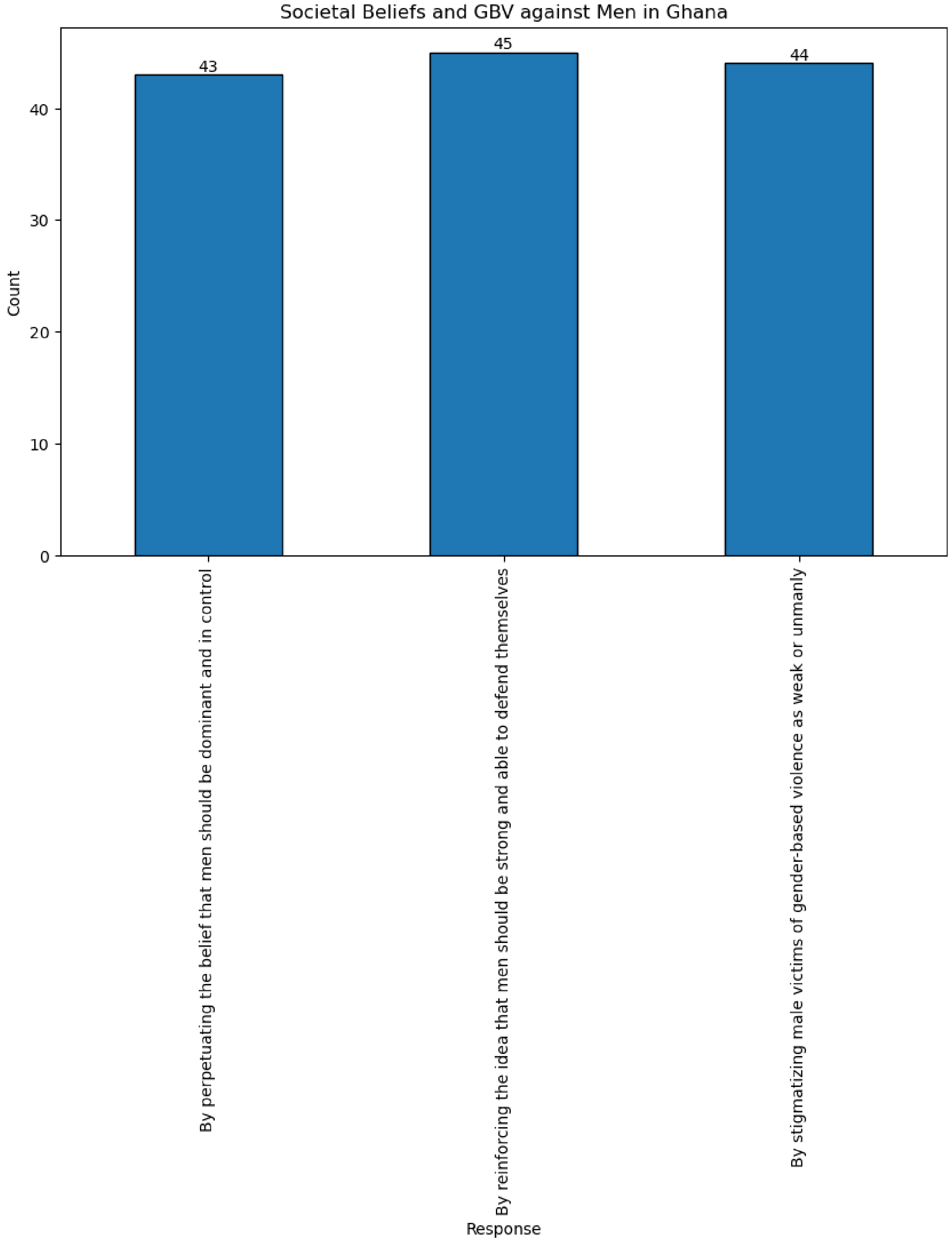
In [18]:

```python
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")
response_data = df['Impact of Gender Norms on GBV against Men in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()

# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Impact of Gender Norms on GBV against Men in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot12.png', dpi=300)
plt.show()
```


Impact of Gender Norms on GBV against Men in Ghana
Count
0
10
20
30
40
50
49
36
35
36
By making men feel ashamed or emasculated if they experience violence
›n that men should be the primary providers for their families leading to frustration anger and violent behavior if they are unable to provide
ltural norms around masculinity that may lead men to feel entitled to sexual or physical dominance over their partners resulting in violence
By reinforcing the idea that men should always be in control of their partners leading to controlling behaviors that escalate into violence

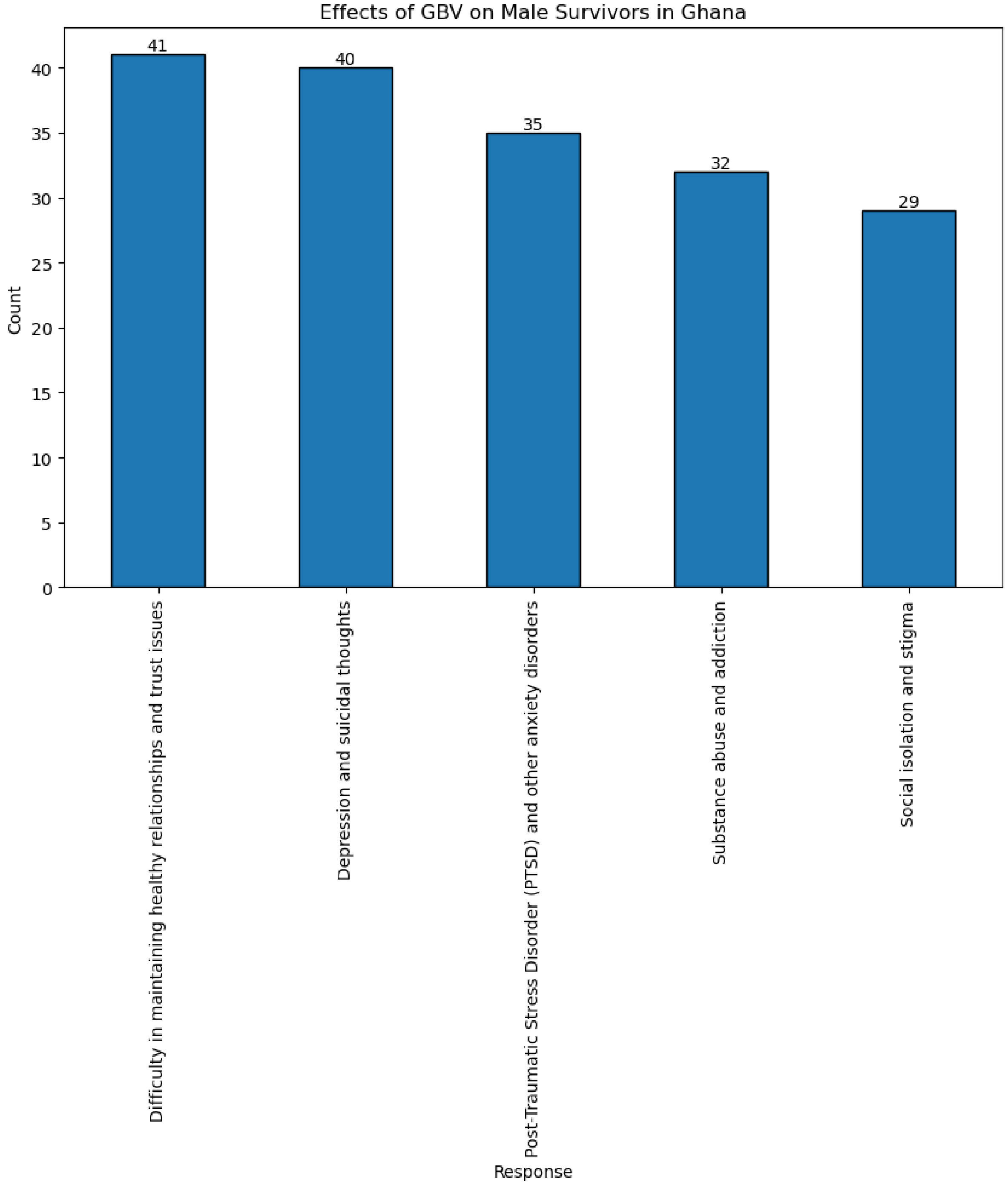
In [19]:

```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Societal Beliefs and GBV against Men in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()

# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Societal Beliefs and GBV against Men in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot13.png', dpi=300)
plt.show()
```



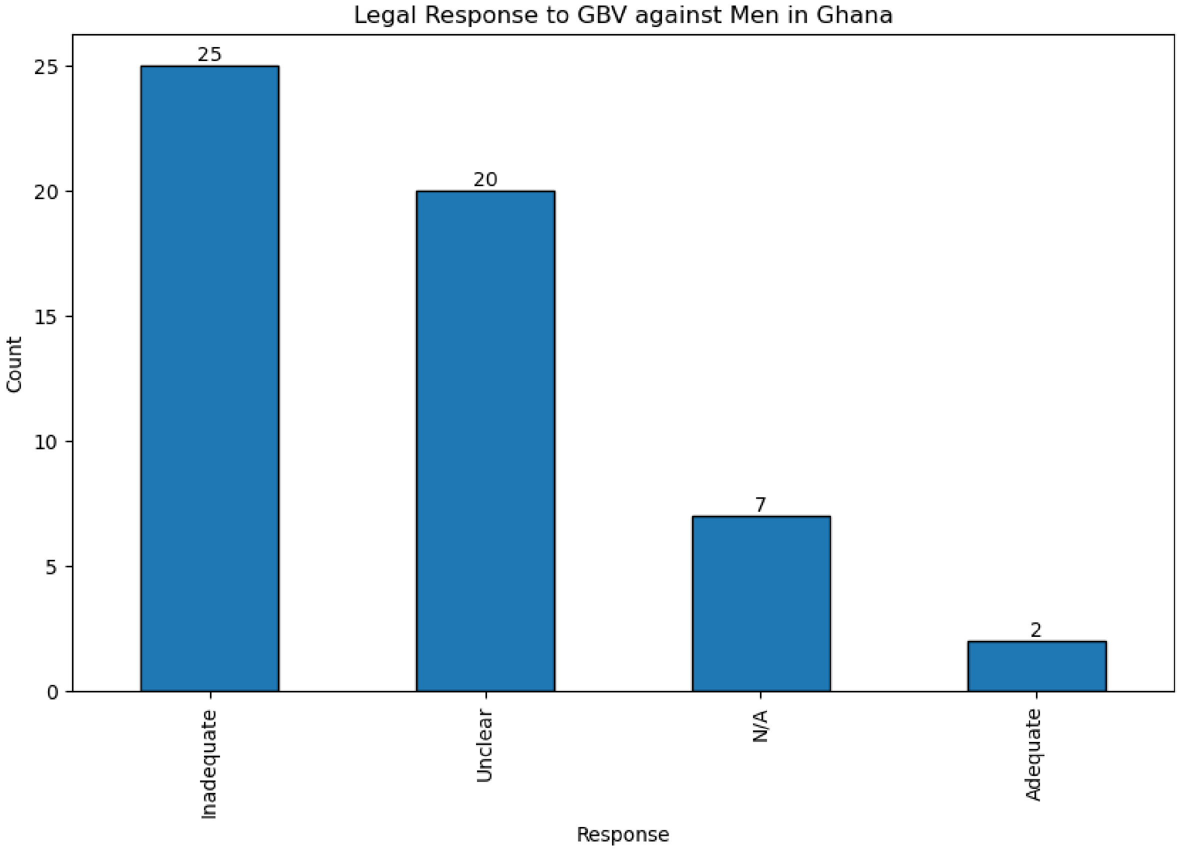
In [20]:
```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Effects of GBV on Male Survivors in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())
```

```
# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Effects of GBV on Male Survivors in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')


plt.savefig('plot14.png', dpi=300)
plt.show()
```



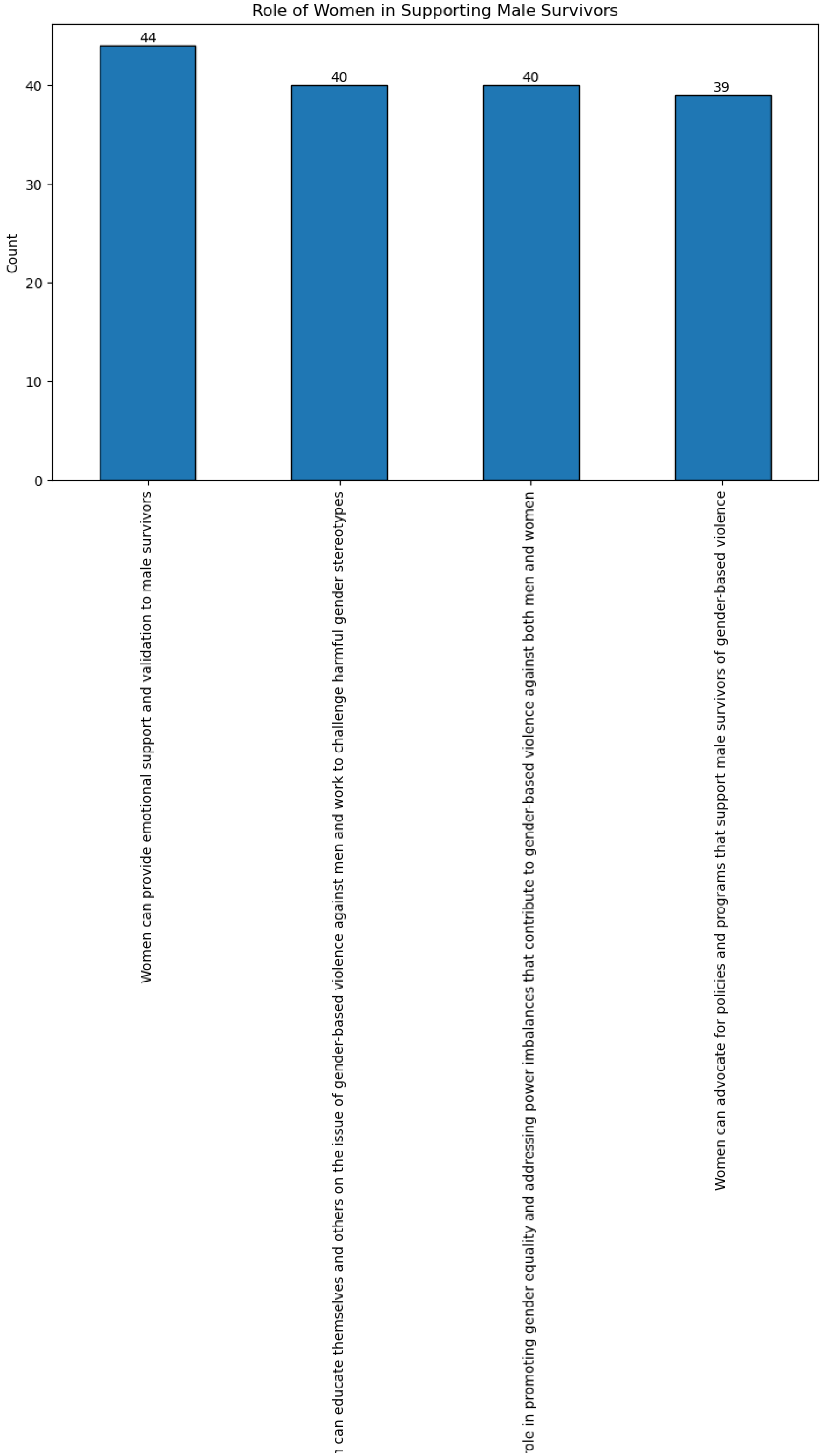
In [21]:

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

# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Legal Response to GBV against Men in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()

# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Include N/A category in the counts
n_a_count = df['Legal Response to GBV against Men in Ghana'].isna().sum()
response_counts['N/A'] = n_a_count

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Legal Response to GBV against Men in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot15.png', dpi=300)
plt.show()
```



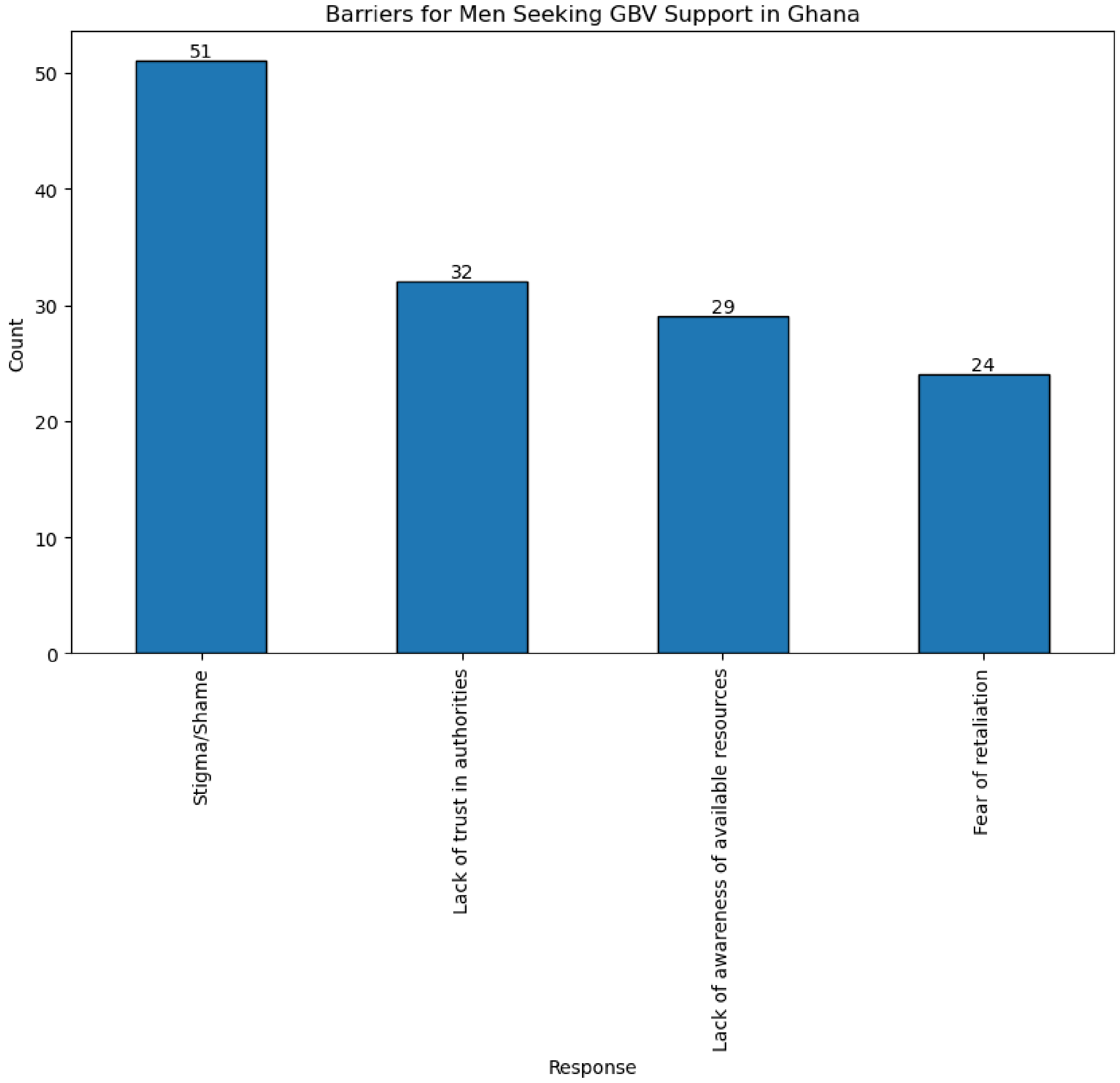
In [22]:

```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Role of Women in Supporting Male Survivors']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Role of Women in Supporting Male Survivors')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')



plt.savefig('plot16.png', dpi=300)
plt.show()
```


Role of Women in Supporting Male Survivors
Count
0
10
20
30
40
44
40
40
39
Women can provide emotional support and validation to male survivors
n can educate themselves and others on the issue of gender-based violence against men and work to challenge harmful gender stereotypes
role in promoting gender equality and addressing power imbalances that contribute to gender-based violence against both men and women
Women can advocate for policies and programs that support male survivors of gender-based violence

In [23]:

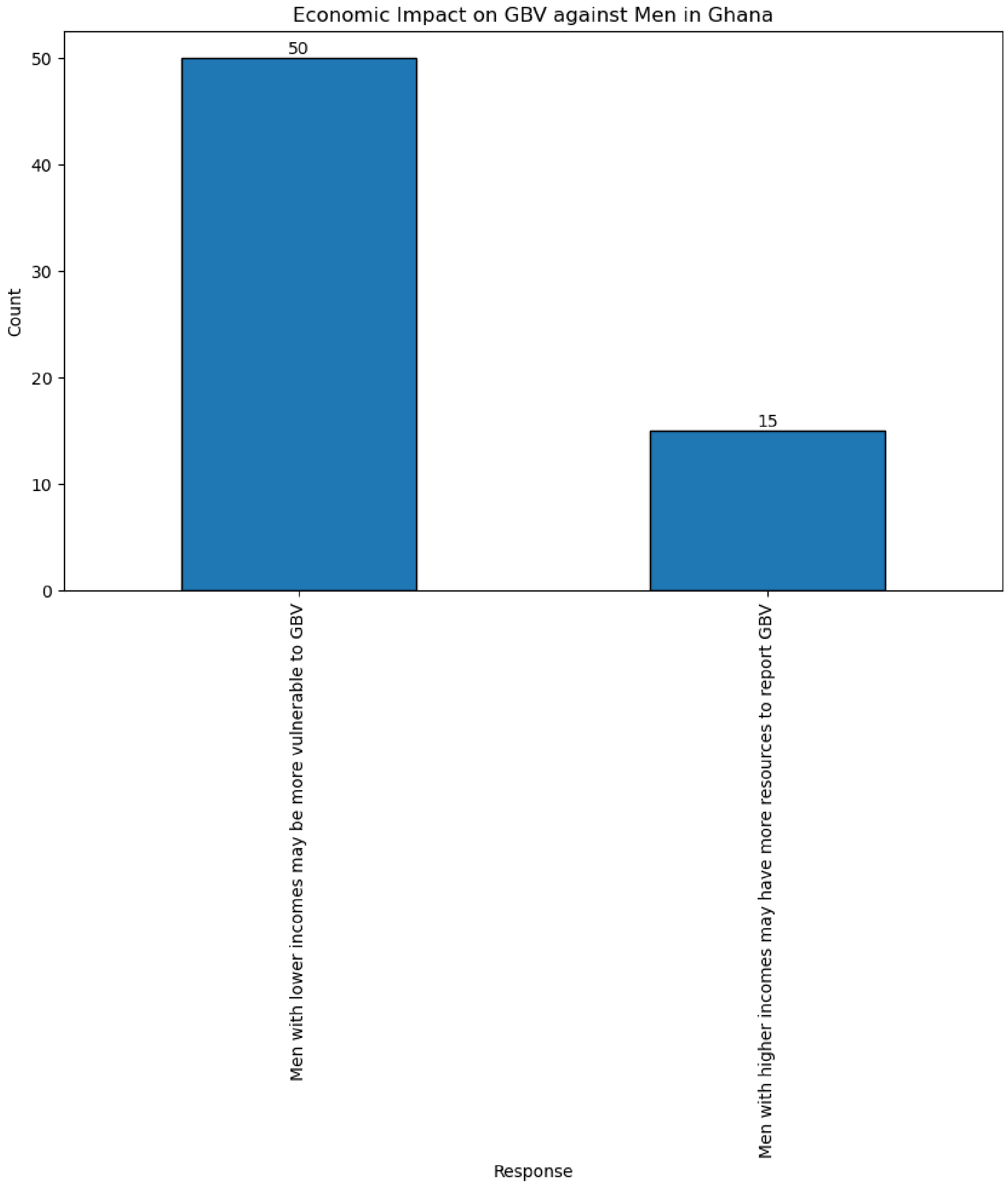
```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['\nBarriers for Men Seeking GBV Support in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Barriers for Men Seeking GBV Support in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot17.png', dpi=300)
plt.show()
```



In [24]:

```python
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Economic Impact on GBV against Men in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Economic Impact on GBV against Men in Ghana')
plt.xticks(rotation=90)
```

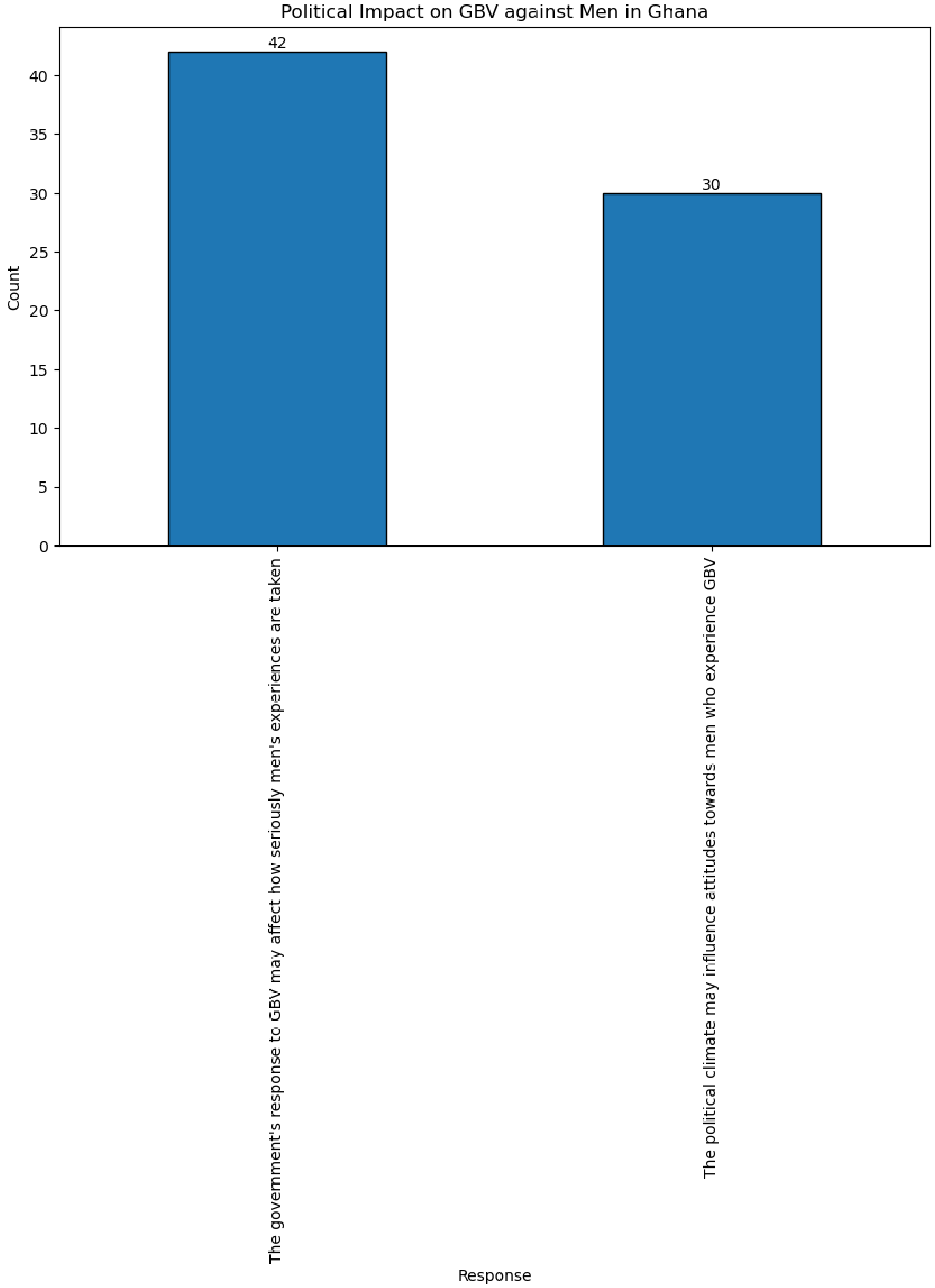
```
# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot18.png', dpi=300)
plt.show()
```



In [25]:

```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Political Impact on GBV against Men in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()
```

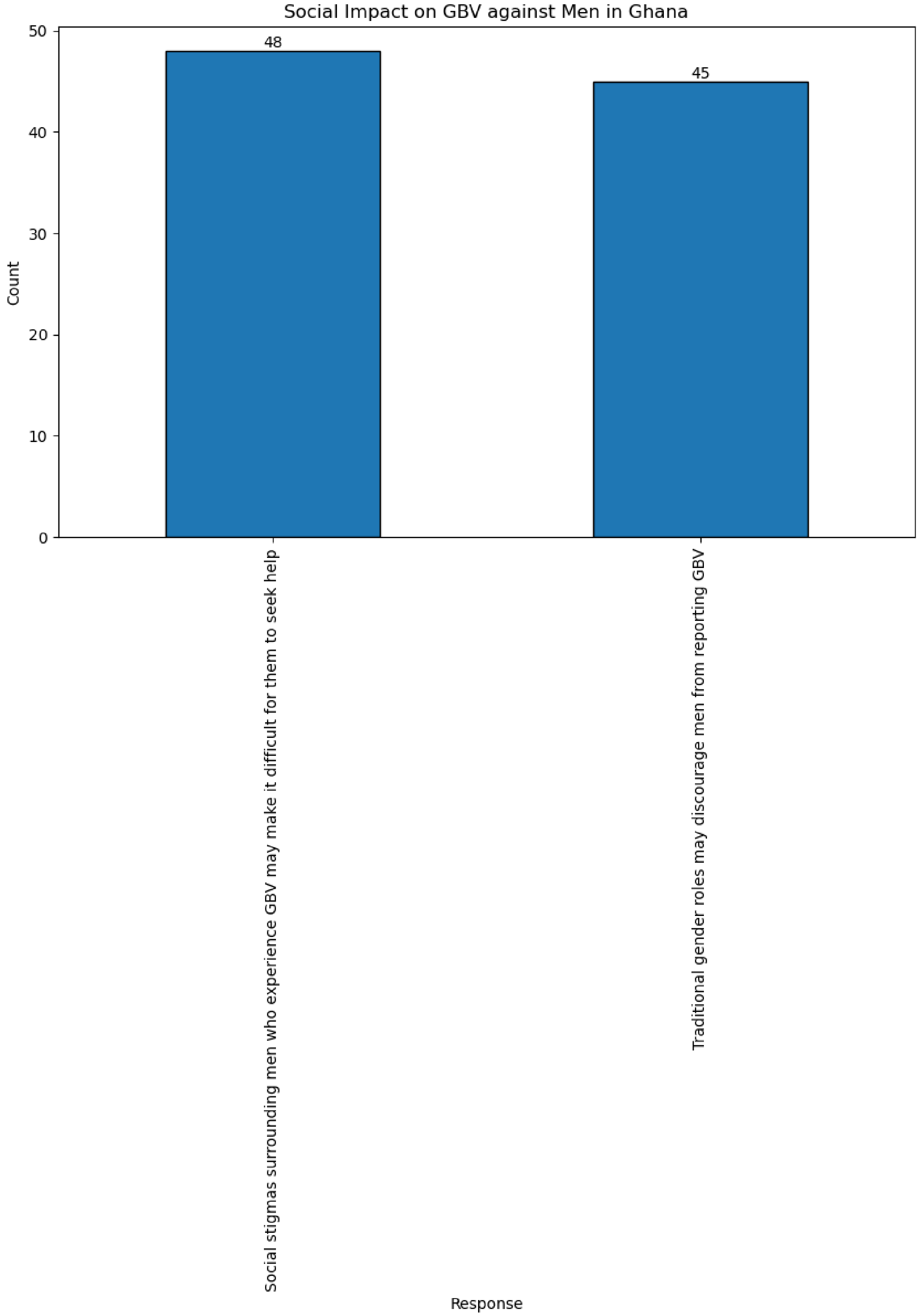
```python
# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Political Impact on GBV against Men in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')


plt.savefig('plot19.png', dpi=300)
plt.show()
```



In [26]:

```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Social Impact on GBV against Men in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()
```

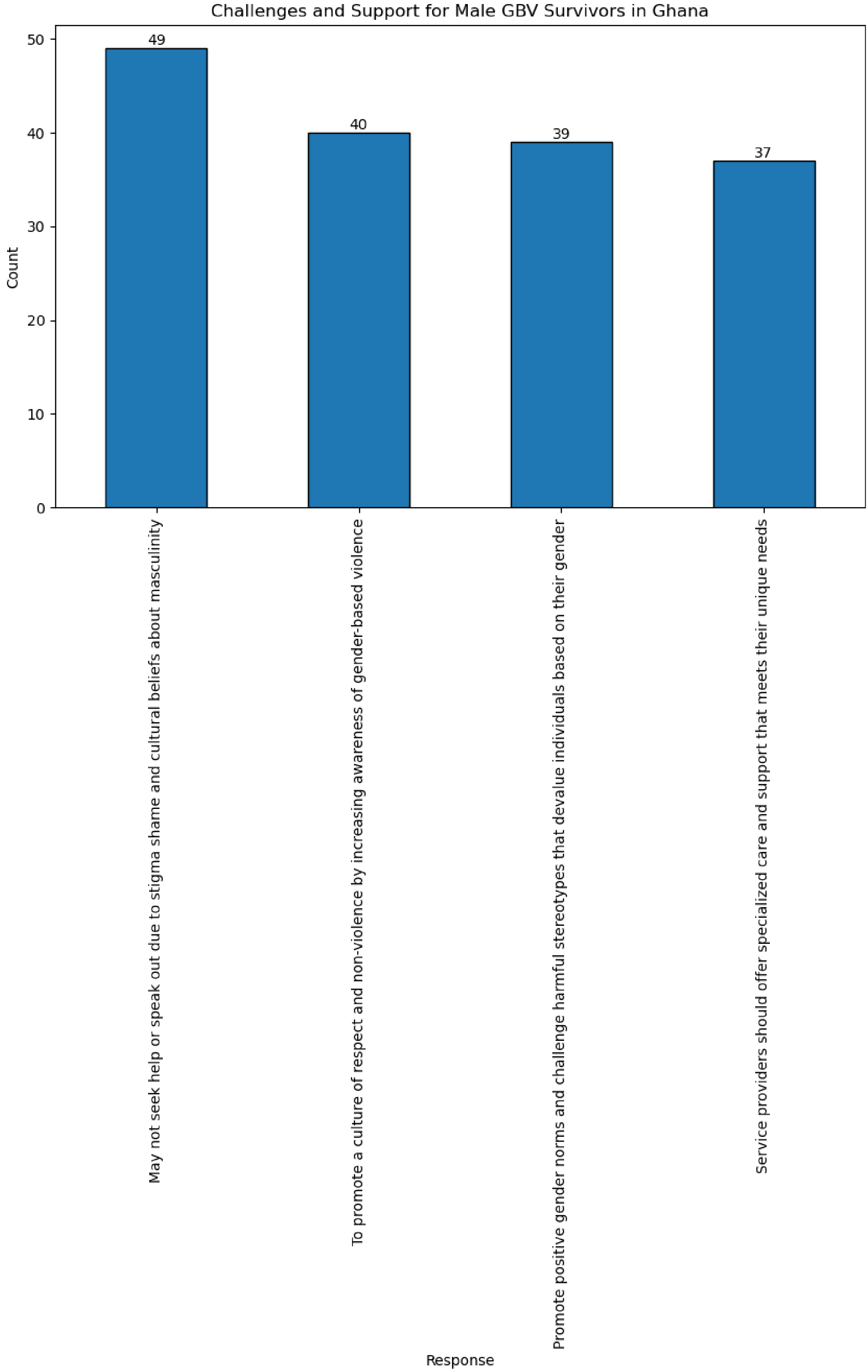
```python
# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Social Impact on GBV against Men in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot20.png', dpi=300)
plt.show()
```



In [27]:

```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Challenges and Support for Male GBV Survivors in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
```

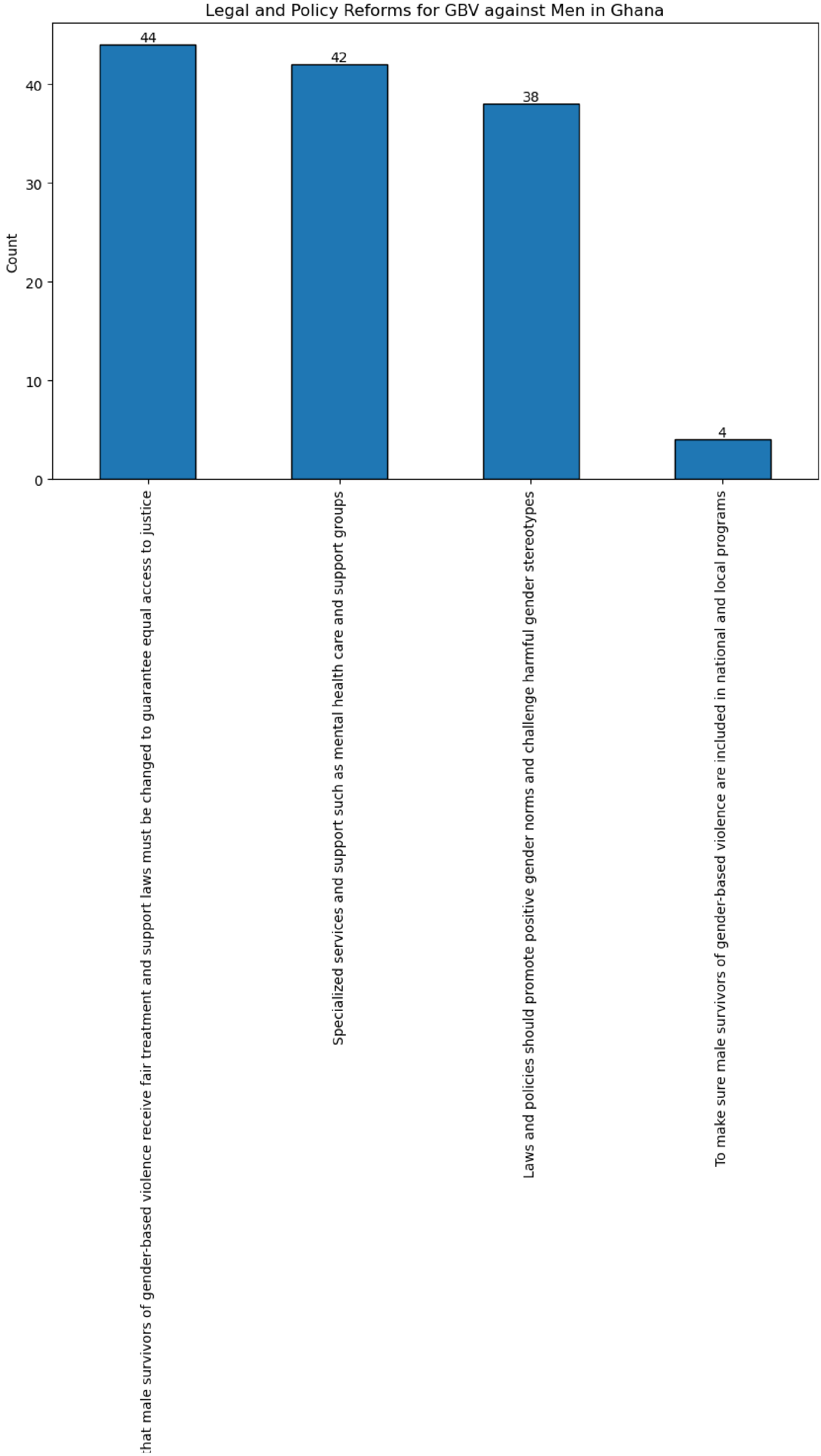
```
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Challenges and Support for Male GBV Survivors in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')


plt.savefig('plot21.png', dpi=300)
plt.show()
```



In [28]:

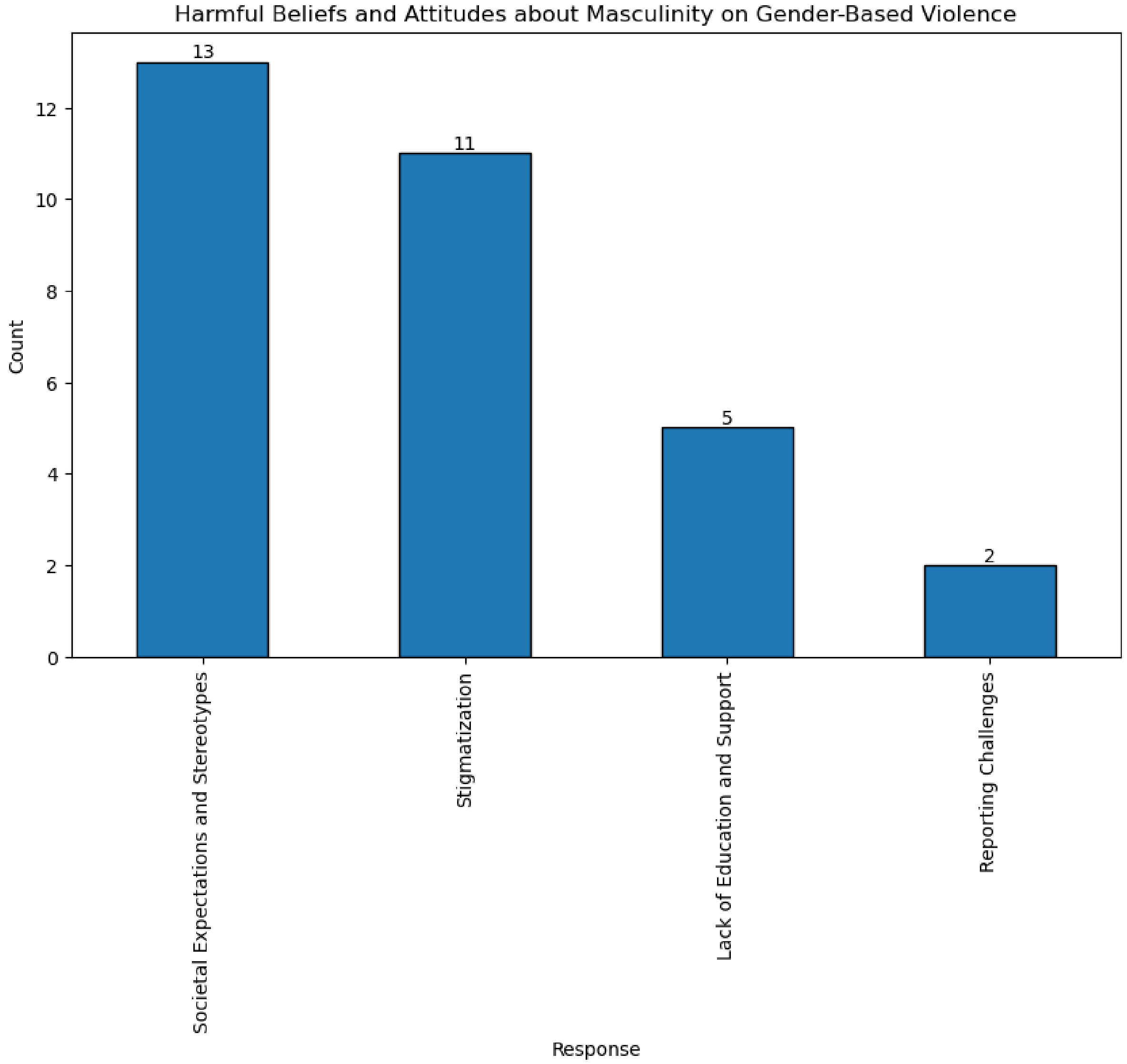
```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Legal and Policy Reforms for GBV against Men in Ghana']
```

```python

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Legal and Policy Reforms for GBV against Men in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot22.png', dpi=300)
plt.show()
```


Legal and Policy Reforms for GBV against Men in Ghana
Count
0
10
20
30
40
44
42
38
4
:hat male survivors of gender-based violence receive fair treatment and support laws must be changed to guarantee equal access to justice
Specialized services and support such as mental health care and support groups
Laws and policies should promote positive gender norms and challenge harmful gender stereotypes
To make sure male survivors of gender-based violence are included in national and local programs

In [29]:

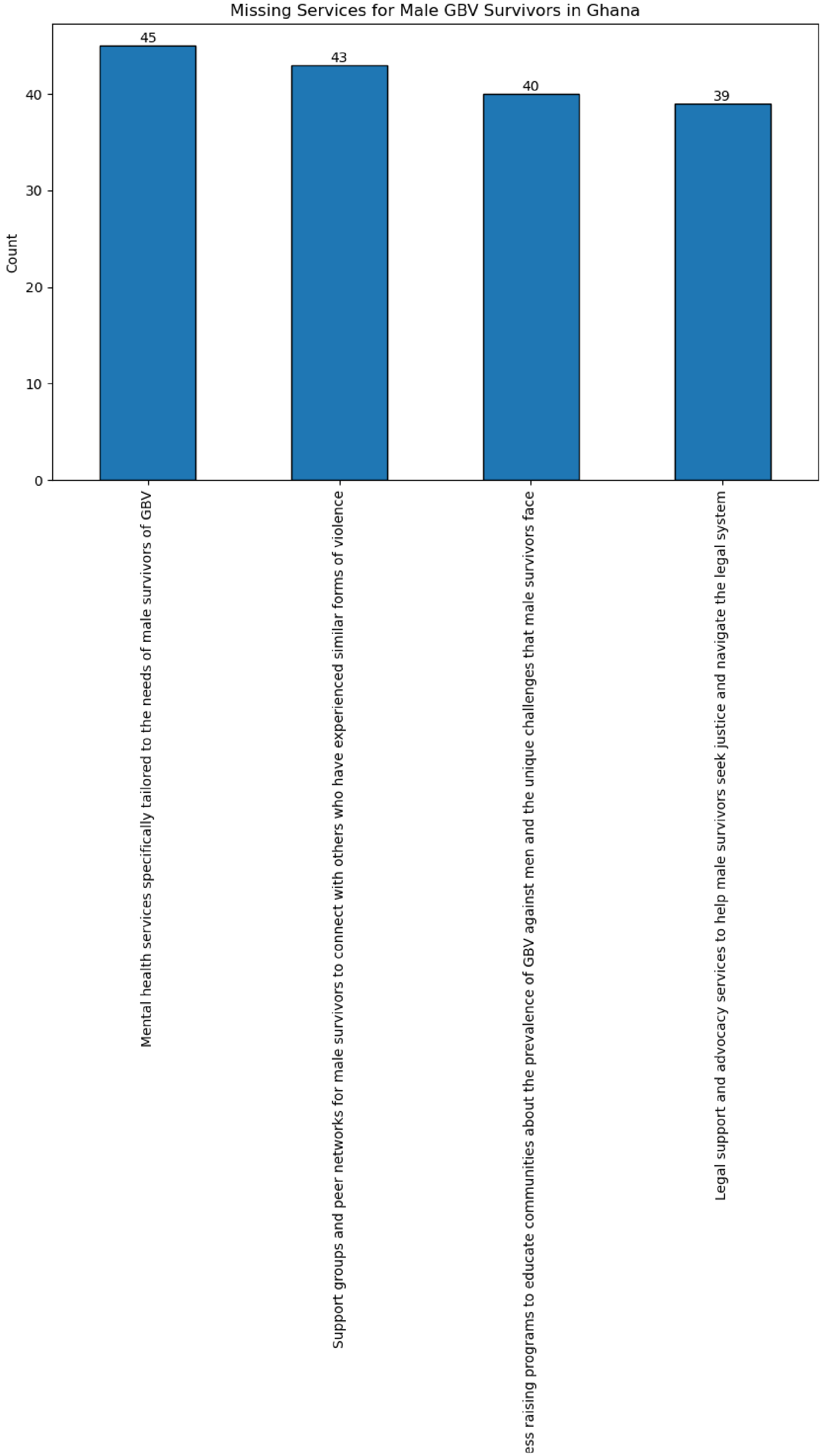
```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Harmful Beliefs and Attitudes about Masculinity on Gende

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Harmful Beliefs and Attitudes about Masculinity on Gender-Based V
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot23.png', dpi=300)
plt.show()
```



In [30]:

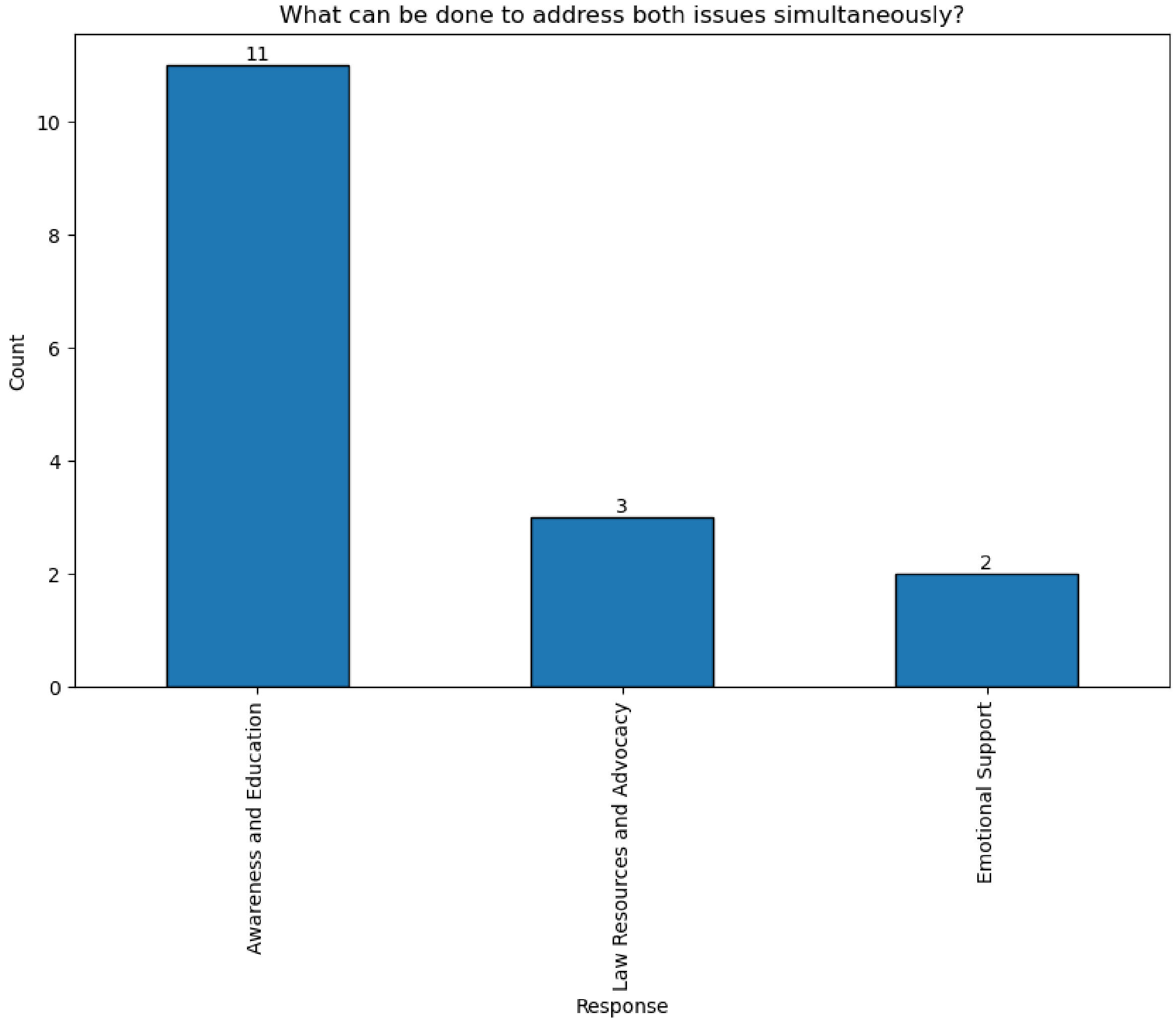
```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Missing Services for Male GBV Survivors in Ghana']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('Missing Services for Male GBV Survivors in Ghana')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
```

```python
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot24.png', dpi=300)
plt.show()
```


Missing Services for Male GBV Survivors in Ghana
Count
0
10
20
30
40
45
43
40
39
Mental health services specifically tailored to the needs of male survivors of GBV
Support groups and peer networks for male survivors to connect with others who have experienced similar forms of violence
ess raising programs to educate communities about the prevalence of GBV against men and the unique challenges that male survivors face
Legal support and advocacy services to help male survivors seek justice and navigate the legal system

In [31]:

```
# Read the data
df = pd.read_csv("GBV (Responses) - Form Responses 4.0.csv")

# Get the response data
response_data = df['Solutions to GBV Against Men']

# Split the responses and stack them into a single column
responses = response_data.str.split(",", expand=True).stack()

# Count the occurrences of each response
response_counts = responses.value_counts()


# Merge all duplicate categories
response_counts = response_counts.groupby(response_counts.index.str.strip())

# Sort the counts in descending order
response_counts = response_counts.sort_values(ascending=False)

# Plot the bar graph
plt.figure(figsize=(10, 6))
response_counts.plot(kind='bar', edgecolor='black')
plt.xlabel('Response')
plt.ylabel('Count')
plt.title('What can be done to address both issues simultaneously?')
plt.xticks(rotation=90)

# Add value counts at the top of each bar
for i, count in enumerate(response_counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.savefig('plot25.png', dpi=300)
plt.show()
```



What can be done to address both issues simultaneously?

```
In [32]: with PdfPages('plots1.pdf') as pdf:
             for filename in ['plot1.jpg', 'plot2.png', 'plot3.png', 'plot4.png', 'pl
                              'plot8.png','plot9.png','plot10.png','plot11.png','plot1
                              'plot15.png','plot16.png','plot17.png','plot18.png','plc
                              'plot22.png','plot23.png','plot24.png','plot25.png']:
                 fig = plt.figure()
                 img = plt.imread(filename)
                 plt.imshow(img)
                 plt.axis('off')
                 pdf.savefig(fig)
                 pdf.savefig(fig, dpi=300)
                 plt.close()
```

```
In [ ]:
```