jupyter Fatalities in the Israeli-Palestinian 2000 - 2023 Last Checkpoint: 3 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [184]:
```
import numpy as np
import pandas as pd
```

In [185]:
```
data=pd.read_csv('fatalities_isr_pse_conflict_2000_to_2023.csv')
```

In [186]:
```
data.head()
```

Out[186]:

	name	date_of_event	age	citizenship	event_location	event_location_district	event_location_region	date_of_death	gender	took_part_in_the_hostilities
0	'Abd a-Rahman Suleiman Muhammad Abu Daghash	2023-09-24	32.0	Palestinian	Nur Shams R.C.	Tulkarm	West Bank	2023-09-24	M	NaN
1	Usayed Farhan Muhammad 'Ali Abu 'Ali	2023-09-24	21.0	Palestinian	Nur Shams R.C.	Tulkarm	West Bank	2023-09-24	M	NaN
2	'Abdallah 'Imad Sa'ed Abu Hassan	2023-09-22	16.0	Palestinian	Kfar Dan	Jenin	West Bank	2023-09-22	M	NaN
3	Durgham Muhammad Yihya al-Akhras	2023-09-20	19.0	Palestinian	'Aqbat Jaber R.C.	Jericho	West Bank	2023-09-20	M	NaN
4	Raafat 'Omar Ahmad Khamaisah	2023-09-19	15.0	Palestinian	Jenin R.C.	Jenin	West Bank	2023-09-19	M	NaN

Data Cleaning and Statistics

In [187]:
```
1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11124 entries, 0 to 11123
Data columns (total 16 columns):
 #  Column                    Non-Null Count  Dtype
---  ------                   --------------  -----
 0  name                     11124 non-null  object
 1  date_of_event            11124 non-null  object
 2  age                      10995 non-null  float64
 3  citizenship              11124 non-null  object
 4  event_location           11124 non-null  object
 5  event_location_district  11124 non-null  object
 6  event_location_region    11124 non-null  object
 7  date_of_death            11124 non-null  object
 8  gender                   11104 non-null  object
 9  took_part_in_the_hostilities  9694 non-null  object
 10 place_of_residence       11056 non-null  object
 11 place_of_residence_district  11056 non-null  object
 12 type_of_injury           10833 non-null  object
 13 ammunition               5871 non-null  object
 14 killed_by                11124 non-null  object
 15 notes                    10844 non-null  object
dtypes: float64(1), object(15)
memory usage: 1.4+ MB
```

In [188]:
```
1 data.describe()
```

Out[188]:

	age
count	10995.000000
mean	26.745703
std	13.780548
min	1.000000
25%	19.000000
50%	23.000000
75%	31.000000
max	112.000000

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [189]:
```
data.isna().sum()  #CHECKING NULL VALUES
```

Out[189]:
```
name                          0
date_of_event                 0
age                         129
citizenship                   0
event_location                0
event_location_district       0
event_location_region         0
date_of_death                 0
gender                       20
took_part_in_the_hostilities  1430
place_of_residence           68
place_of_residence_district   68
type_of_injury               291
ammunition                  5253
killed_by                     0
notes                       280
dtype: int64
```

In [190]:
```
data.drop('notes', axis=1, inplace=True)
```

In [191]:
```
#REPLACING NULL VALUES

data['age']=data['age'].fillna(data['age'].mean())
data['gender']=data['gender'].fillna(data['gender'].mode()[0])
data['type_of_injury']=data['type_of_injury'].fillna(data['type_of_injury'].mode()[0])
data['took_part_in_the_hostilities']=data['took_part_in_the_hostilities'].fillna(data['took_part_in_the_hostilities'].mode()[0])
data['place_of_residence']=data['place_of_residence'].fillna(data['place_of_residence'].mode()[0])
data['place_of_residence_district']=data['place_of_residence_district'].fillna(data['place_of_residence_district'].mode()[0])
```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [192]:
```
data.isna().sum()
```

Out[192]:
```
name                          0
date_of_event                 0
age                           0
citizenship                   0
event_location                0
event_location_district       0
event_location_region         0
date_of_death                 0
gender                        0
took_part_in_the_hostilities  0
place_of_residence            0
place_of_residence_district   0
type_of_injury                0
ammunition                 5253
killed_by                     0
dtype: int64
```

In [193]:
```
gender_counts = data['gender'].value_counts()        #Genderwise Count
gender_counts
```

Out[193]:
```
M    9701
F    1423
Name: gender, dtype: int64
```

In [194]:
```
#Mean_Age

# Calculate the mean age
mean_age = data['age'].mean()

# Print the mean age
print(f"Mean Age: {mean_age:.2f}")

```

```
Mean Age: 26.75
```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

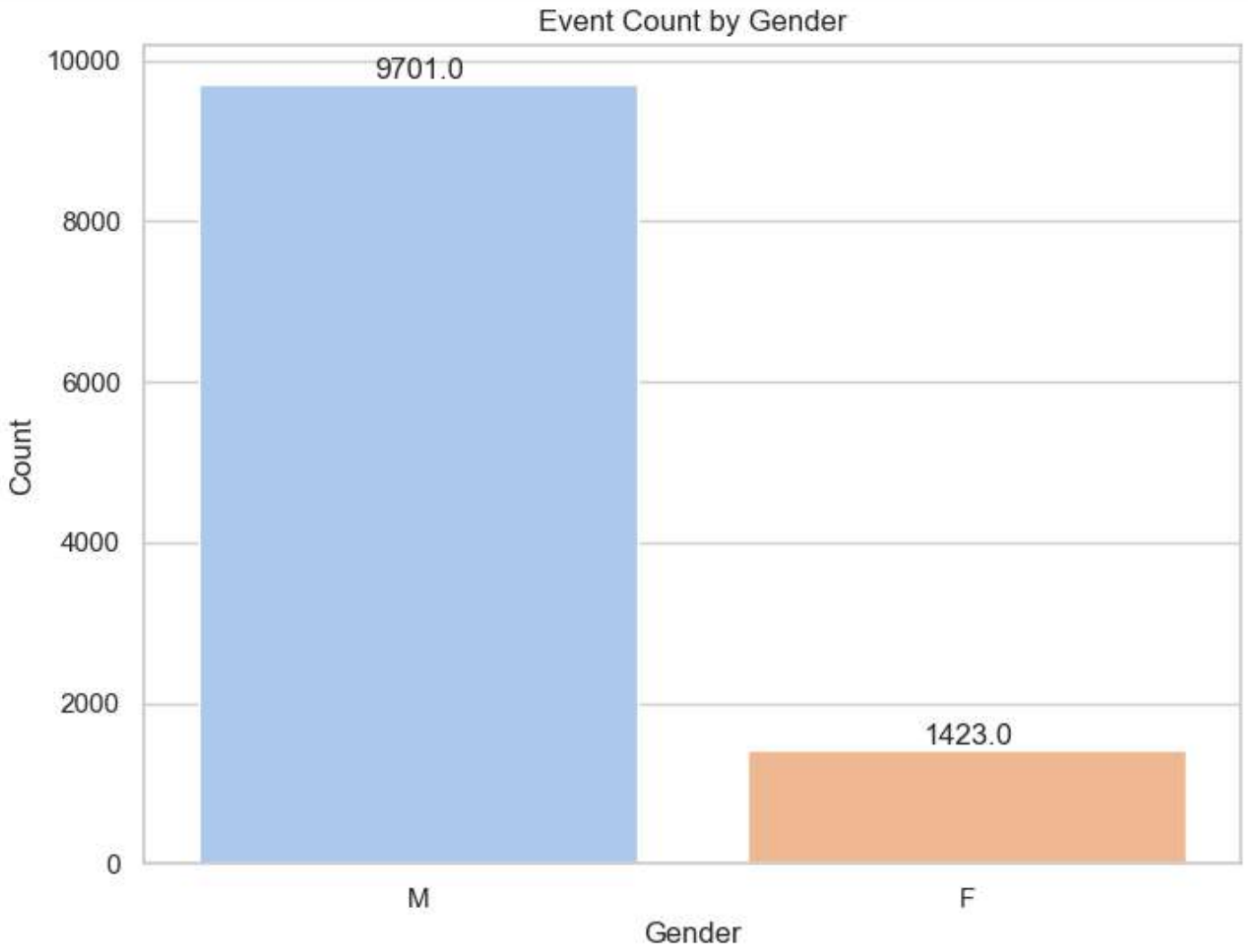
Data Visualization

In [195]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(color_codes=True)
%matplotlib inline
```

In [196]:

```
#  bar chart for genderwise event count

plt.figure(figsize=(8, 6))
sns.set(style="whitegrid")
ax = sns.countplot(data=data, x='gender', palette='pastel')

# Annotate the count on top of each bar
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='bottom')

plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Event Count by Gender')
plt.show()

```



Inference

87% are Male and 13% are female died in the Fatalities.

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

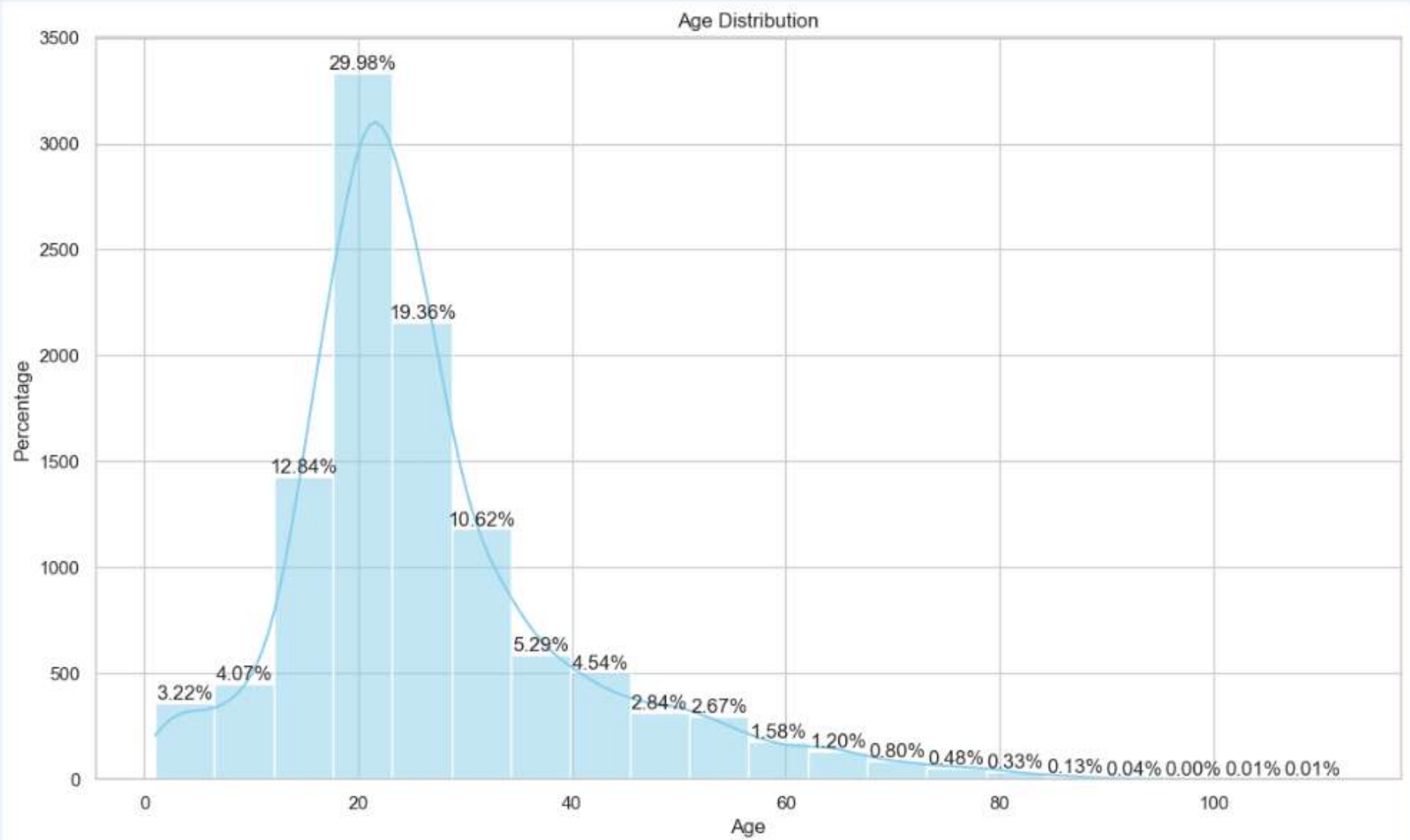
Run Code

In [197]:

```
#Histogram for Age Distribution

#plt.figure(figsize=(8, 6))
#sns.histplot(data=data, x='age', bins=20, kde=True, color='skyblue')
#plt.xlabel('Age')
#plt.ylabel('Frequency')
#plt.title('Age Distribution')
#plt.show()

#OR

# Create the histogram
plt.figure(figsize=(14, 8))
sns.set_style("whitegrid")
ax = sns.histplot(data=data, x='age', bins=20, kde=True, color='skyblue')

plt.xlabel('Age')
plt.ylabel('Percentage')
plt.title('Age Distribution')

# Annotate each bar with the percentage
total_count = len(data['age'])
for p in ax.patches:
    percentage = (p.get_height() / total_count) * 100
    ax.annotate(f'{percentage:.2f}%', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='bottom')

plt.show()


```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

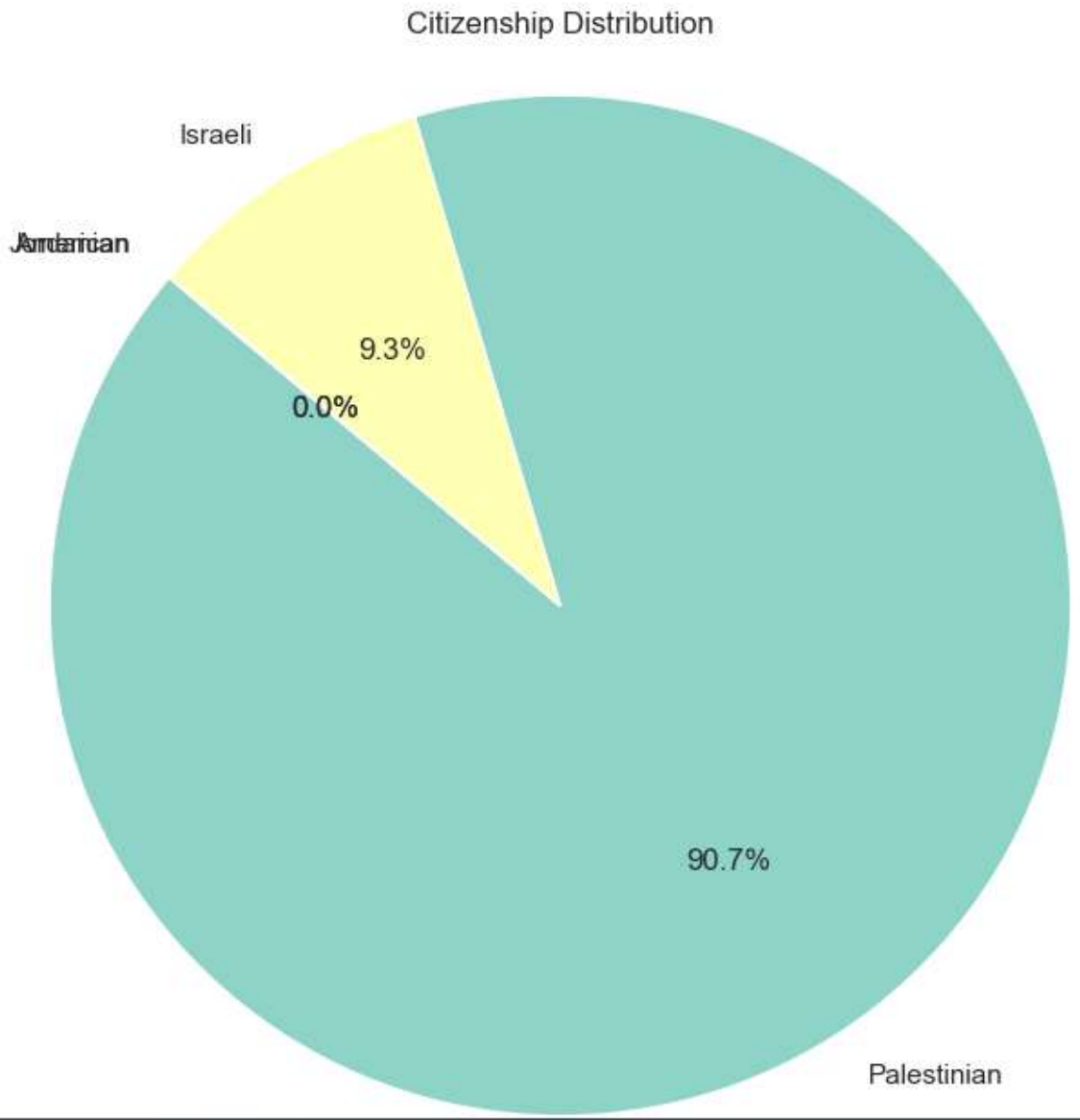


File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [198]:

```
#Pie Chart for Citizenship Distribution

citizenship_counts = data['citizenship'].value_counts()
plt.figure(figsize=(8, 8))
plt.pie(citizenship_counts, labels=citizenship_counts.index, autopct='%1.1f%%', startangle=140, colors=sns.color_palette('Set3'))
plt.axis('equal')
plt.title('Citizenship Distribution')
plt.show()
```



File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

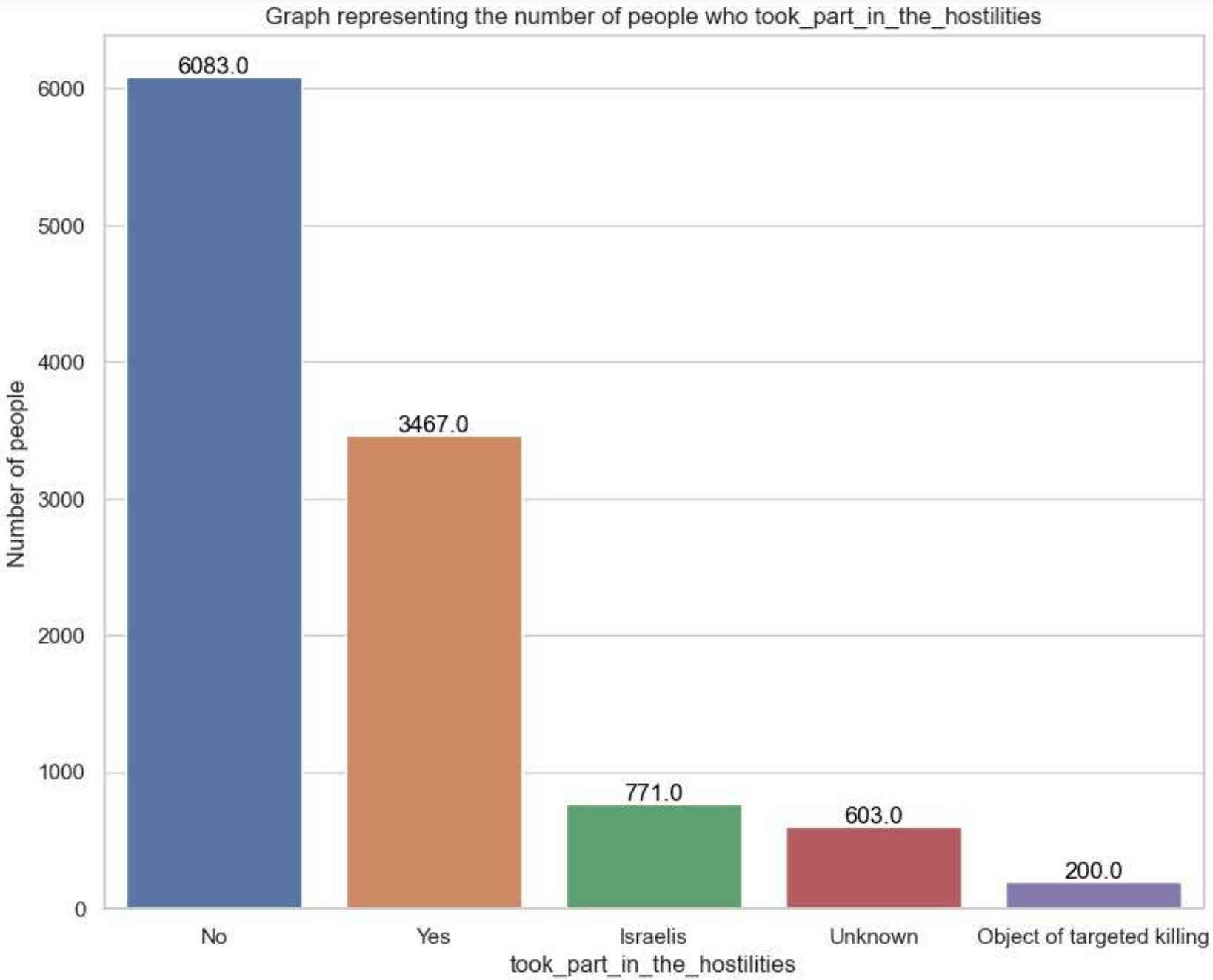
Run Code

In [213]:
```
American_Citicen_Count = data[data['citizenship'] == 'American']['citizenship'].count()
Jordanian_Citizen_Count = data[data['citizenship'] == 'Jordanian']['citizenship'].count()

print("American_Citizen_Count:", American_count)
print("Jordanian_Citizen_Countt:", Jordanian_count)
```

```
American_Citizen_Count: 1
Jordanian_Citizen_Countt: 2
```

In [200]:
```
took_part_in_the_hostilities = data.took_part_in_the_hostilities.value_counts().reset_index()
took_part_in_the_hostilities.columns = ['took_part_in_the_hostilities','count']
```

In [214]:
```
# Histogram for the 'took_part_in_the_hostilities' column

#plt.figure(figsize=(10, 6))
#sns.histplot(data['took_part_in_the_hostilities'])
#plt.title('Hositilities Distribution')
#plt.ylabel('Frequency')
#plt.show()
#for p in ax.patches:
  #  ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
    #          ha='center', va='center', fontsize=12, color='black', xytext=(0, 5), textcoords='offset points')
#plt.show()

#OR

plt.figure(figsize = (10,8))
ax = sns.barplot(x = 'took_part_in_the_hostilities',y = 'count',data = took_part_in_the_hostilities)
plt.xlabel('took_part_in_the_hostilities')
plt.ylabel('Number of people')
plt.title('Graph representing the number of people who took_part_in_the_hostilities')
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', fontsize=12, color='black', xytext=(0, 5), textcoords='offset points')
plt.show()
```



31% are taking part in the Hostilities 54.7% are not taking part in the Hostilities

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

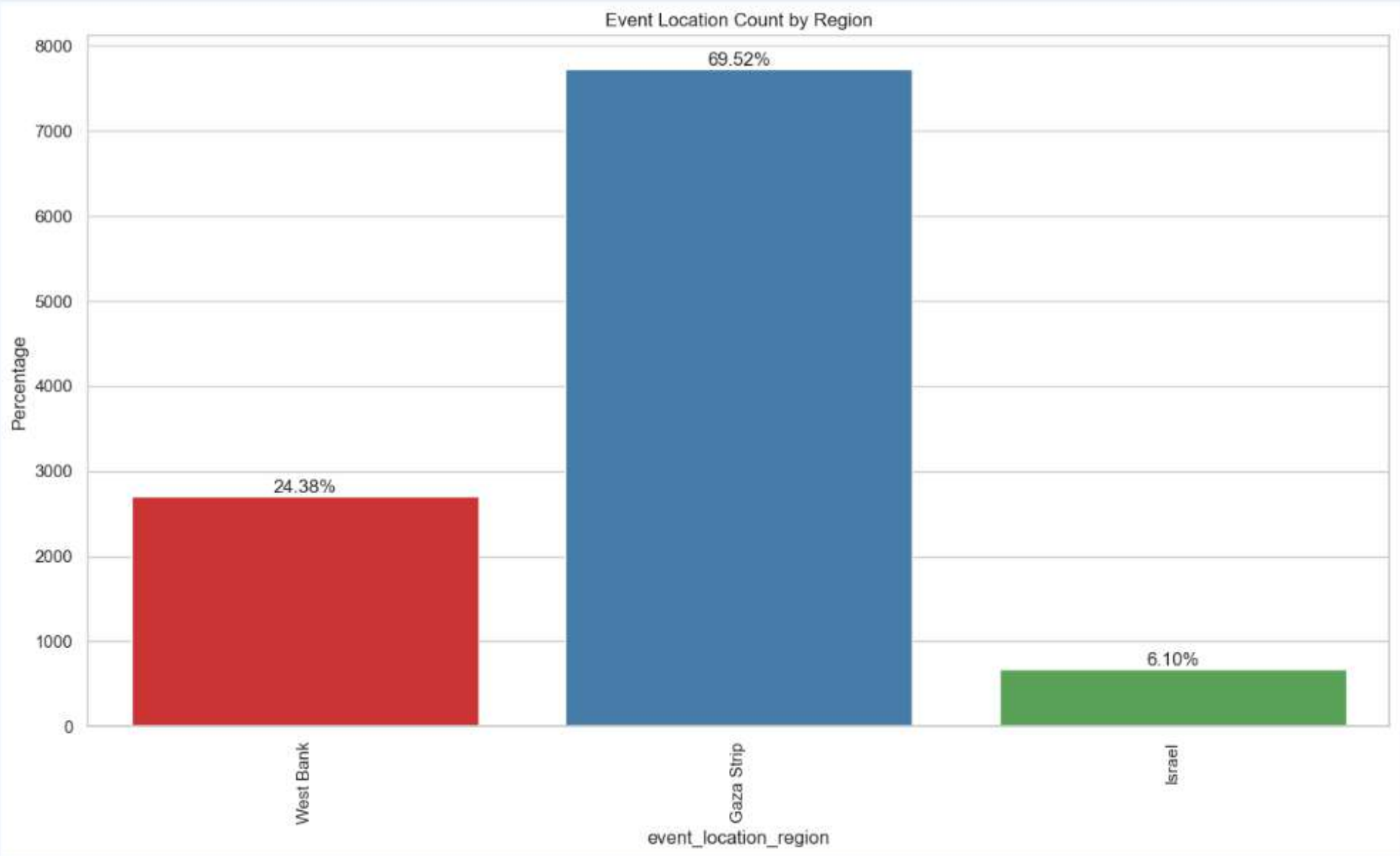
Run Code

In [216]:

```
#Event Location Count by Region

# Create the count plot
plt.figure(figsize=(15, 8))
sns.set_style("whitegrid")
ax = sns.countplot(data=data, x='event_location_region', palette='Set1')

plt.xticks(rotation=90)
plt.ylabel('Percentage')
plt.title('Event Location Count by Region')

# Annotate each bar with the percentage
total_count = len(data['event_location_region'])
for p in ax.patches:
    percentage = (p.get_height() / total_count) * 100
    ax.annotate(f'{percentage:.2f}%', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='bottom')

plt.show()

```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code



Inference

We can say Gaza city face the most number of causalties

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

Time Series Analysis

In [203]:

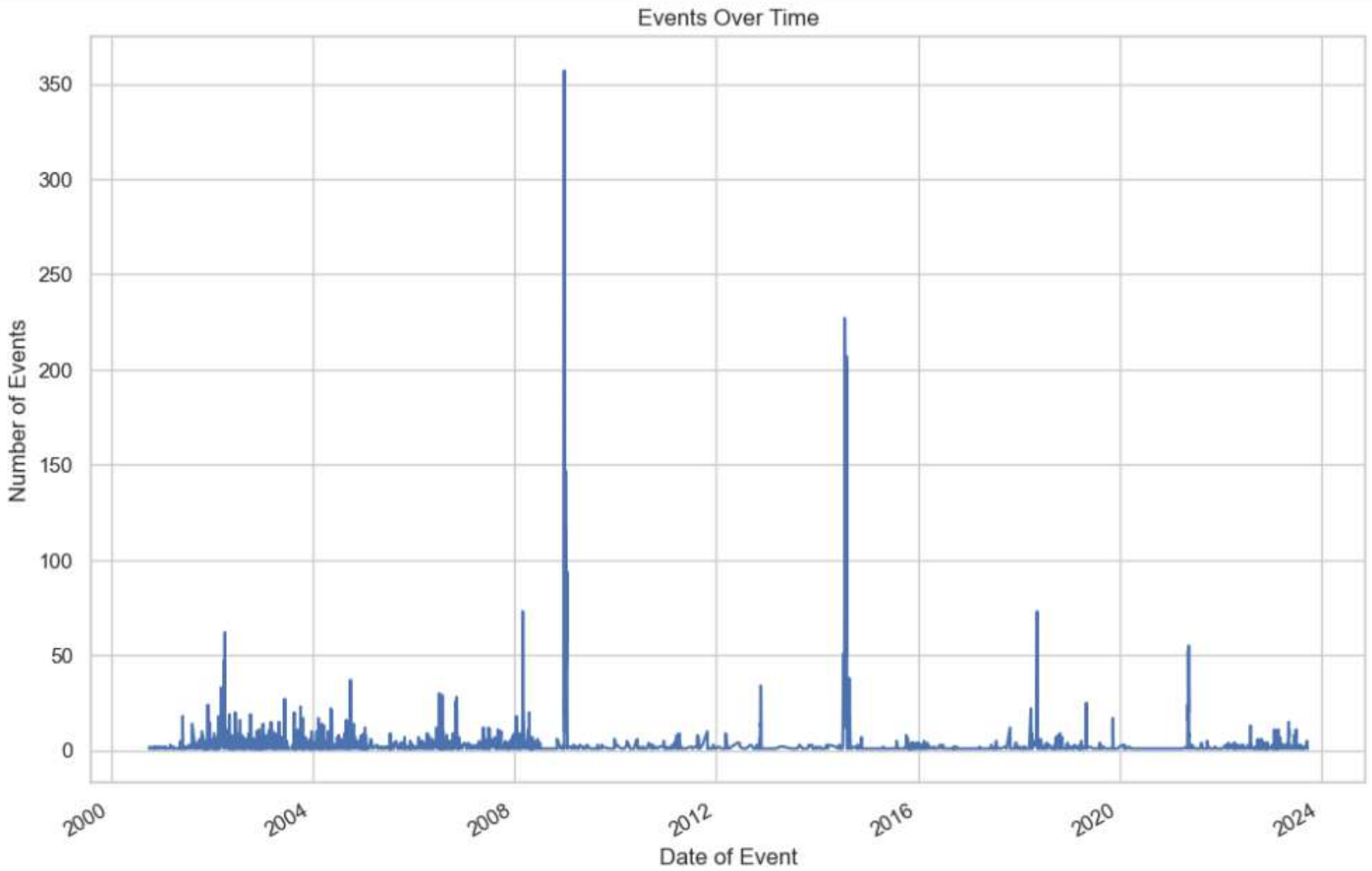
```
# Converting date columns to datetime

data['date_of_event'] = pd.to_datetime(data['date_of_event'])
data['date_of_death'] = pd.to_datetime(data['date_of_death'])
```

In [204]:

```
#Calculating the number of events per day

events_per_day = data.groupby('date_of_event').size()

plt.figure(figsize=(12, 8))
events_per_day.plot()
plt.title('Events Over Time')
plt.xlabel('Date of Event')
plt.ylabel('Number of Events')
plt.show()
```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code



File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

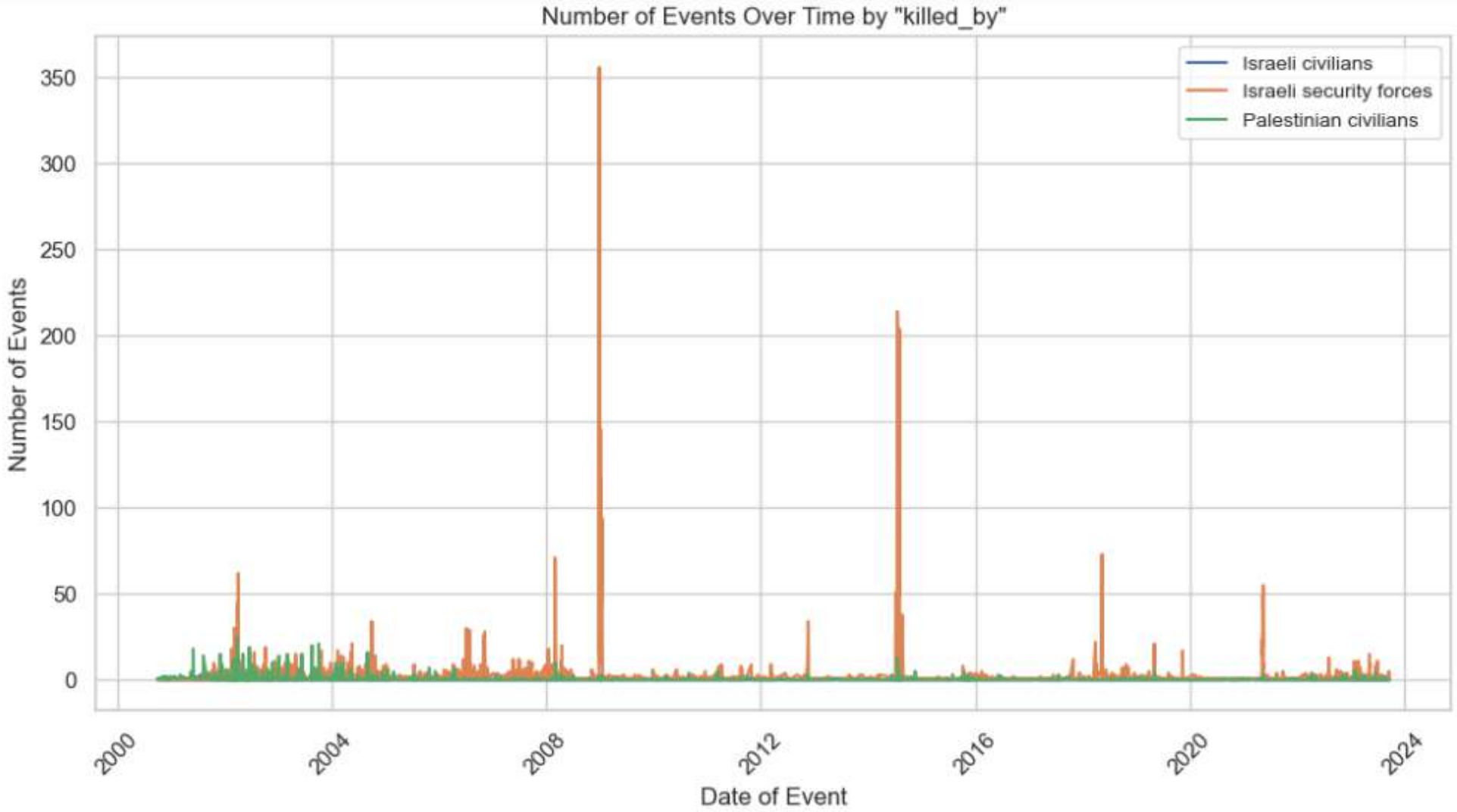
Run Code

In [205]:

```
# Grouping the 'date_of_event' and 'killed_by' column and calculating
#the count of events for each combination

events_by_killed_by = data.groupby(['date_of_event', 'killed_by']).size().unstack().fillna(0)

plt.figure(figsize=(12, 6))
for column in events_by_killed_by.columns:
    plt.plot(events_by_killed_by.index, events_by_killed_by[column], label=column)

plt.title('Number of Events Over Time by "killed_by"')
plt.xlabel('Date of Event')
plt.ylabel('Number of Events')
plt.legend(loc='upper right', fontsize='small')
plt.xticks(rotation=45)
plt.show()
```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code



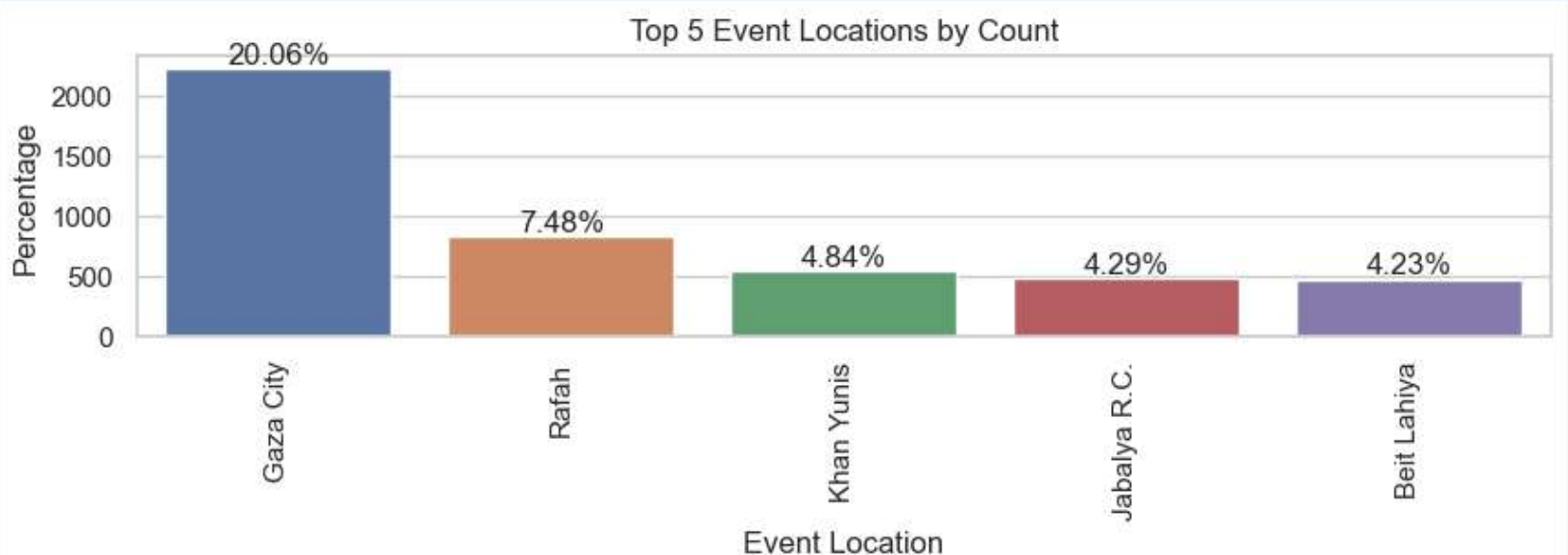
File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

Geological Analysis

In [237]:

```
# Get the top 5 event locations by count
top5_event_locations = data['event_location'].value_counts().head(5)

# Calculate the percentage
total_count = len(data['event_location'])
percentage = (top5_event_locations / total_count) * 100

# Create the bar plot
plt.figure(figsize=(10, 2))
sns.set_style("whitegrid")
ax = sns.barplot(x=top5_event_locations.index, y=top5_event_locations.values)

plt.title('Top 5 Event Locations by Count')
plt.xlabel('Event Location')
plt.ylabel('Percentage')

# Annotate each bar with the percentage
for i, p in enumerate(ax.patches):
    ax.annotate(f'{percentage[i]:.2f}%', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='bottom')

plt.xticks(rotation=90)
plt.show()

```

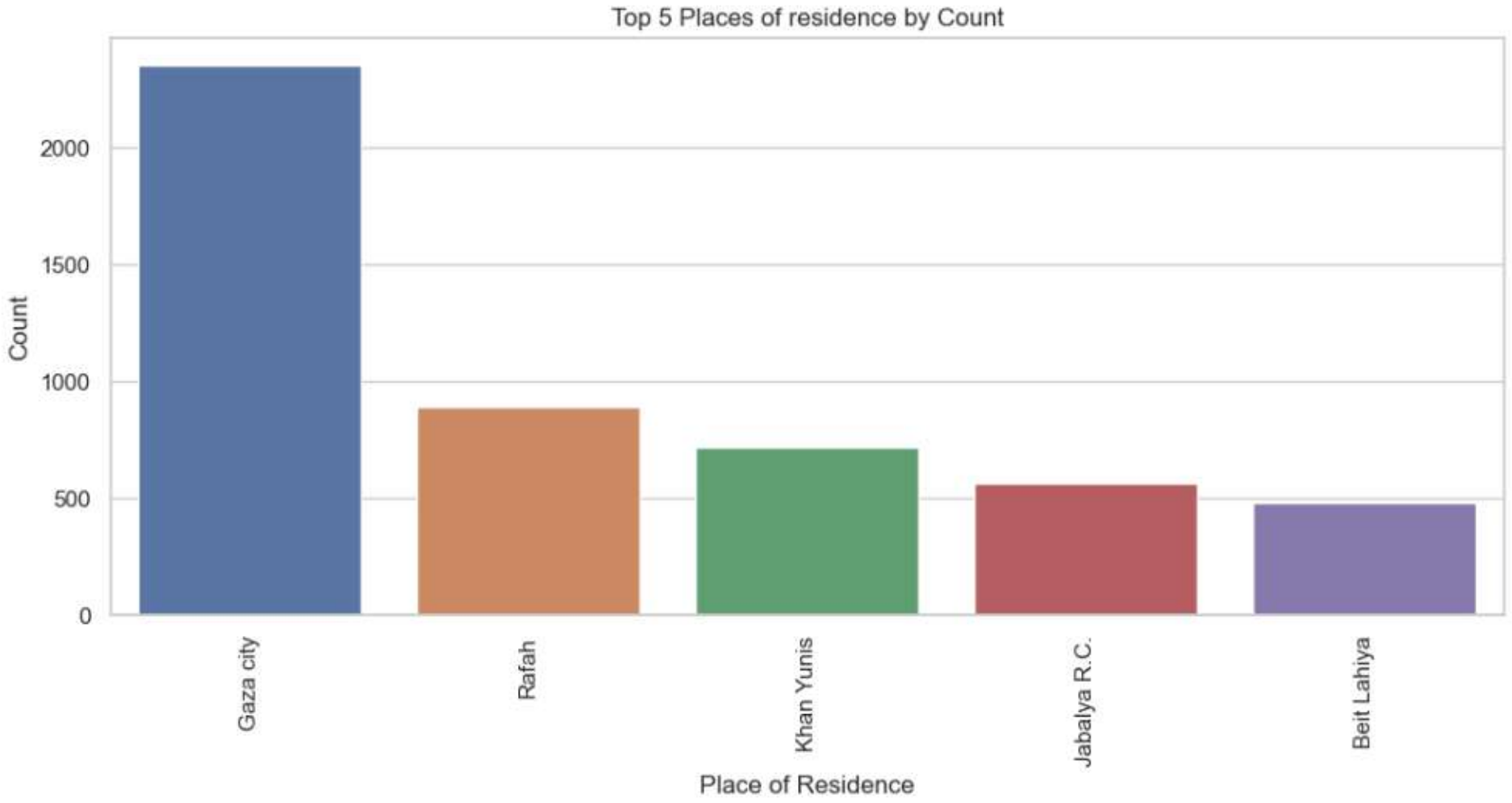


File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [238]:

```
#Top 5 place of residence by count

top5_place_of_residence = data['place_of_residence'].value_counts().head(5)

#Bar plot for the top 5 event locations
plt.figure(figsize=(12, 5))
sns.barplot(x=top5_place_of_residence.index, y=top5_place_of_residence.values)
plt.title('Top 5 Places of residence by Count')
plt.xlabel('Place of Residence')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```

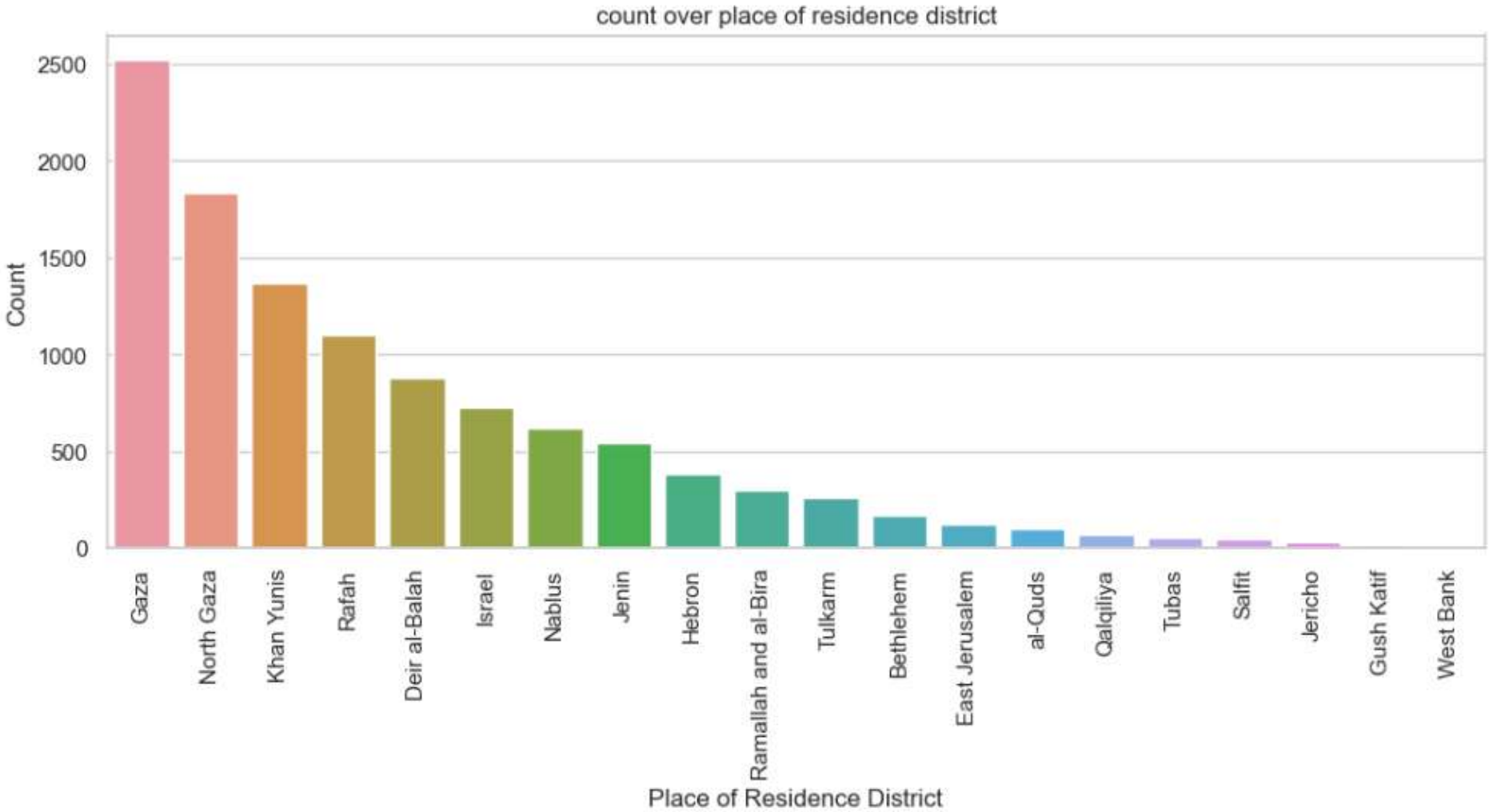


File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Markdown

In [242]:

```
#places of residence  district by count

place_of_residence_district = data['place_of_residence_district'].value_counts()

plt.figure(figsize=(12, 4.5))
sns.barplot(x=place_of_residence_district.index, y=place_of_residence_district.values)
plt.title('count over place of residence district')
plt.xlabel('Place of Residence District')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



We can say Gaza city face the most number of causalties

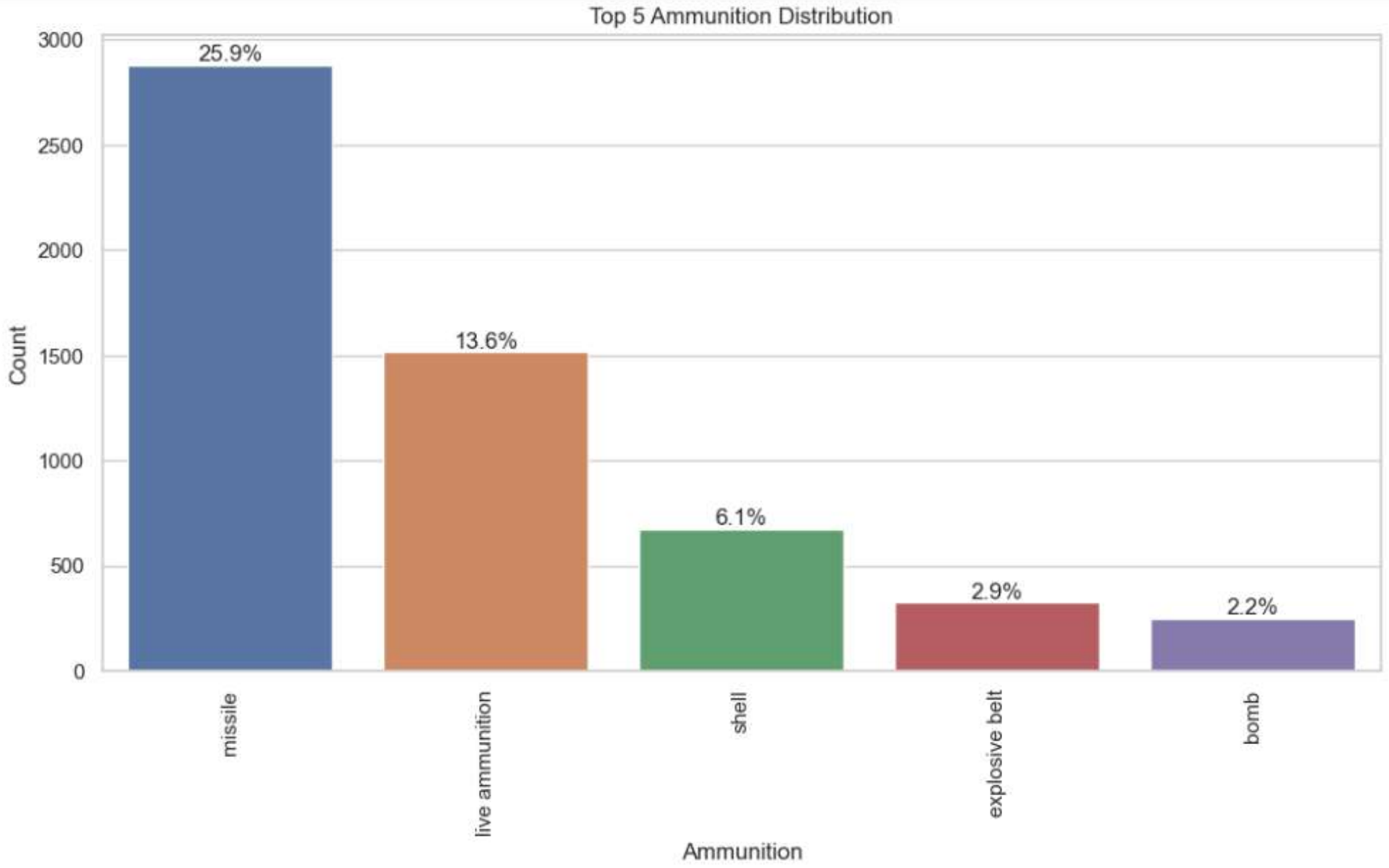
Injuries and Ammunation

In [209]:

```
top5_ammunition_counts = data['ammunition'].value_counts().head(5)

plt.figure(figsize=(12, 6))
ax = sns.barplot(x=top5_ammunition_counts.index, y=top5_ammunition_counts.values)
plt.title('Top 5 Ammunition Distribution')
plt.xlabel('Ammunition')
plt.ylabel('Count')
plt.xticks(rotation=90)

# Add percentage labels above the bars
total = len(data['ammunition'])  # Total number of data points

for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height() / total)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    ax.annotate(percentage, (x, y), ha='center', va='bottom')

plt.show()
```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Markdown

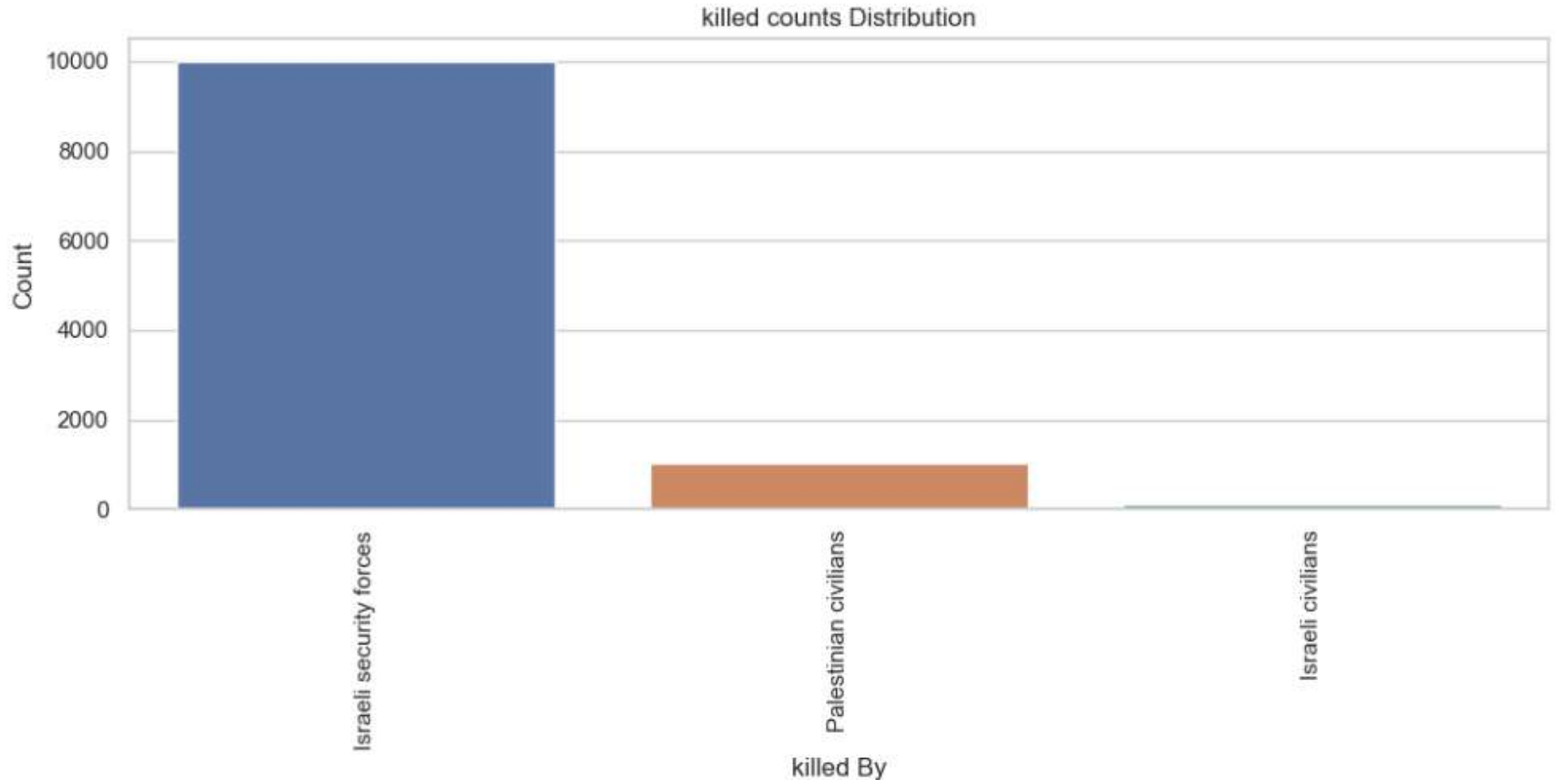


File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [243]:

```
#Killed Count  vs Killed By

killed_by_counts = data['killed_by'].value_counts()
plt.figure(figsize=(12, 4))
sns.barplot(x=killed_by_counts.index, y=killed_by_counts.values)
plt.title('killed counts Distribution')
plt.xlabel('killed By')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

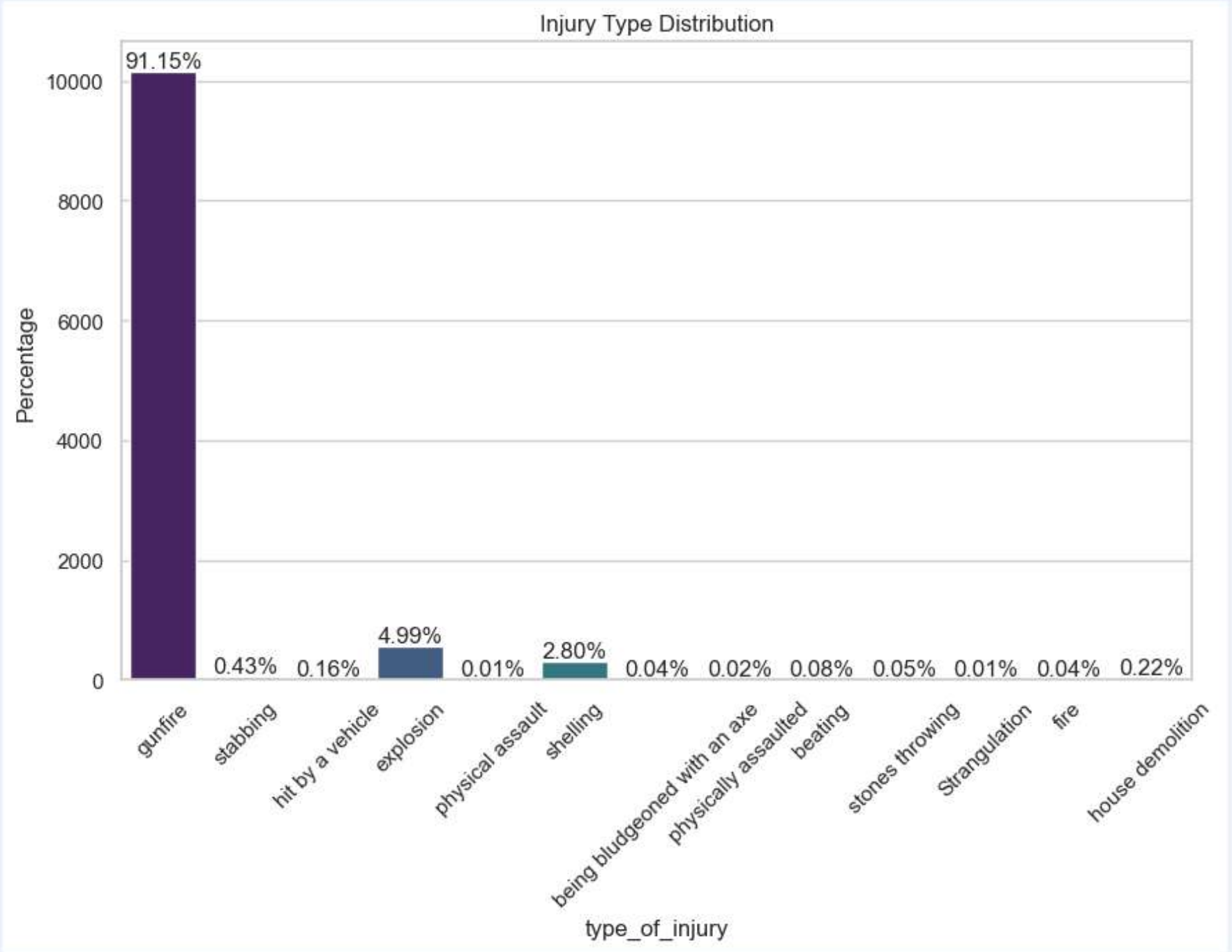
Run Code

In [211]:

```
#Count of  Injuries by Type

plt.figure(figsize=(10, 6))
sns.set_style("whitegrid")
ax = sns.countplot(data=data, x='type_of_injury', palette='viridis')

plt.xticks(rotation=45)
plt.ylabel('Percentage')
plt.title('Injury Type Distribution')

# Annotate each bar with the percentage
total_count = len(data['type_of_injury'])  # Assuming 'type_of_injury' is the column representing injury types
for p in ax.patches:
    percentage = (p.get_height() / total_count) * 100
    ax.annotate(f'{percentage:.2f}%', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='bottom')

plt.show()
```

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

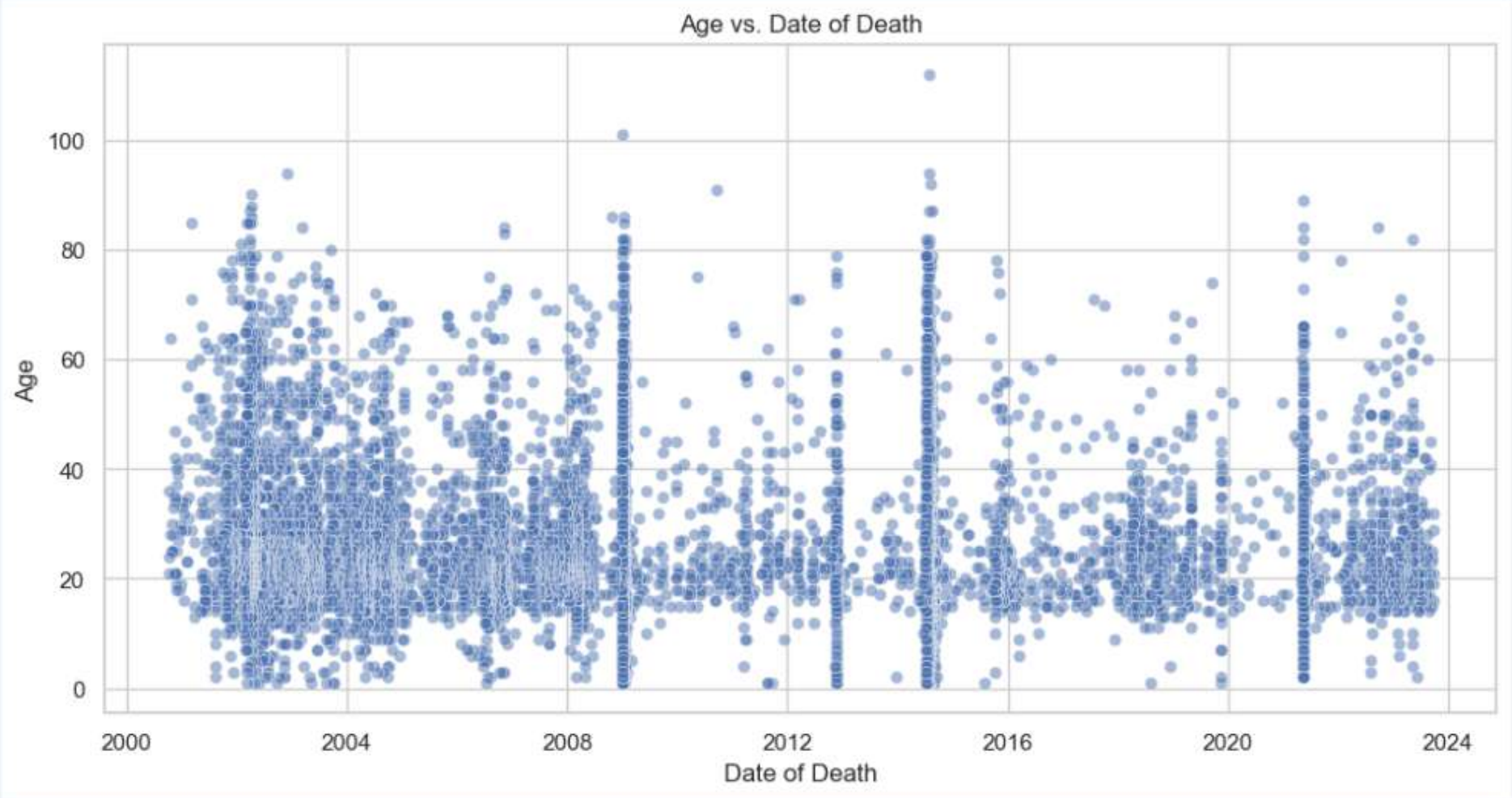
Run Code



File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [245]:
```
#Scatter Plot for Age Vs Date of Death
plt.figure(figsize=(12, 5.8))
sns.scatterplot(data=data, x='date_of_death', y='age', alpha=0.5, color='b')
plt.xlabel('Date of Death')
plt.ylabel('Age')
plt.title('Age vs. Date of Death')
plt.show()
```



Inference

Based on the graph, it is evident that individuals in the age group 18 - 40 experienced the highest number of casualties during war from 2000 to 2023. This suggests that this age group was disproportionately affected by the conflict.