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

conflict= pd.read_csv("fatalities.csv")
conflict
```

```
                                                    name date_of_event   age
\
0      'Abd a-Rahman Suleiman Muhammad Abu Daghash    2023-09-24  32.0

1              Usayed Farhan Muhammad 'Ali Abu 'Ali    2023-09-24  21.0

2                  'Abdallah 'Imad Sa'ed Abu Hassan    2023-09-22  16.0

3                  Durgham Muhammad Yihya al-Akhras    2023-09-20  19.0

4                      Raafat 'Omar Ahmad Khamaisah    2023-09-19  15.0

...                                            ...           ...   ...

11119                             Binyamin Herling    2000-10-19  64.0

11120                  Farid Musa 'Issa a-Nesasreh    2000-10-17  28.0

11121                             Hillel Lieberman    2000-10-07  36.0

11122                    Fahed Mustafa 'Odeh Baker    2000-10-07  21.0

11123                            Wichlav Zalsevsky    2000-10-02  24.0


       citizenship     event_location event_location_district  \
0      Palestinian     Nur Shams R.C.                 Tulkarm
1      Palestinian     Nur Shams R.C.                 Tulkarm
2      Palestinian           Kfar Dan                   Jenin
3      Palestinian  'Aqbat Jaber R.C.                 Jericho
4      Palestinian         Jenin R.C.                   Jenin
...            ...                ...                     ...
11119      Israeli             Nablus                  Nablus
11120  Palestinian         Beit Furik                  Nablus
11121      Israeli             Nablus                  Nablus
11122  Palestinian              Bidya                  Salfit
11123      Israeli              Masha                  Salfit

      event_location_region date_of_death gender
took_part_in_the_hostilities  \
0                 West Bank    2023-09-24      M
NaN
1                 West Bank    2023-09-24      M
```

```
NaN
2                 West Bank    2023-09-22      M
NaN
3                 West Bank    2023-09-20      M
NaN
4                 West Bank    2023-09-19      M
NaN
...                     ...           ...    ...
...
11119             West Bank    2000-10-19      M
Israelis
11120             West Bank    2000-10-17      M
Unknown
11121             West Bank    2000-10-07      M
Israelis
11122             West Bank    2000-10-07      M
No
11123             West Bank    2000-10-02      M
Israelis

      place_of_residence place_of_residence_district type_of_injury  \
0         Nur Shams R.C.                     Tulkarm        gunfire
1         Nur Shams R.C.                     Tulkarm        gunfire
2               al-Yamun                       Jenin        gunfire
3      'Aqbat Jaber R.C.                     Jericho        gunfire
4                  Jenin                       Jenin        gunfire
...                  ...                         ...            ...
11119            Kedumim                     Tulkarm        gunfire
11120         Beit Furik                      Nablus        gunfire
11121         Elon Moreh                      Nablus        gunfire
11122              Bidya                      Salfit        gunfire
11123             Ashdod                      Israel        gunfire

            ammunition                killed_by  \
0      live ammunition  Israeli security forces
1      live ammunition  Israeli security forces
2      live ammunition  Israeli security forces
3      live ammunition  Israeli security forces
4      live ammunition  Israeli security forces
...                ...                      ...
11119  live ammunition    Palestinian civilians
11120              NaN        Israeli civilians
11121  live ammunition    Palestinian civilians
11122              NaN        Israeli civilians
11123  live ammunition    Palestinian civilians

                                                   notes
0      Fatally shot by Israeli forces while standing ...
1      Fatally shot by Israeli forces while trying to...
2      Fatally shot by soldiers while firing at them ...
```

```
3      Shot in the head by Israeli forces while throw...
4      Wounded by soldiers’ gunfire after running awa...
...                                                 ...
11119                  Killed while hiking on Mt. Eival.
11120  Killed by a settler from Itamar while harvesti...
11121     His body was found a day after he disappeared.
11122   Killed by settlers who rioted in Biddya village.
11123                                                NaN

[11124 rows x 16 columns]
```

Data Preprocessing

```
conflict.head()
```

```
                                          name date_of_event   age  \
0  'Abd a-Rahman Suleiman Muhammad Abu Daghash    2023-09-24  32.0
1         Usayed Farhan Muhammad 'Ali Abu 'Ali    2023-09-24  21.0
2             'Abdallah 'Imad Sa'ed Abu Hassan    2023-09-22  16.0
3             Durgham Muhammad Yihya al-Akhras    2023-09-20  19.0
4                 Raafat 'Omar Ahmad Khamaisah    2023-09-19  15.0

   citizenship     event_location event_location_district  \
0  Palestinian     Nur Shams R.C.                 Tulkarm
1  Palestinian     Nur Shams R.C.                 Tulkarm
2  Palestinian           Kfar Dan                   Jenin
3  Palestinian  'Aqbat Jaber R.C.                 Jericho
4  Palestinian         Jenin R.C.                   Jenin

  event_location_region date_of_death gender
took_part_in_the_hostilities  \
0             West Bank    2023-09-24      M
NaN
1             West Bank    2023-09-24      M
NaN
2             West Bank    2023-09-22      M
NaN
3             West Bank    2023-09-20      M
NaN
4             West Bank    2023-09-19      M
NaN

  place_of_residence place_of_residence_district type_of_injury  \
0     Nur Shams R.C.                     Tulkarm        gunfire
1     Nur Shams R.C.                     Tulkarm        gunfire
2           al-Yamun                       Jenin        gunfire
3  'Aqbat Jaber R.C.                     Jericho        gunfire
```

```
4               Jenin                             Jenin       gunfire

        ammunition                killed_by  \
0  live ammunition  Israeli security forces
1  live ammunition  Israeli security forces
2  live ammunition  Israeli security forces
3  live ammunition  Israeli security forces
4  live ammunition  Israeli security forces

                                               notes
0  Fatally shot by Israeli forces while standing ...
1  Fatally shot by Israeli forces while trying to...
2  Fatally shot by soldiers while firing at them ...
3  Shot in the head by Israeli forces while throw...
4  Wounded by soldiers’ gunfire after running awa...
```

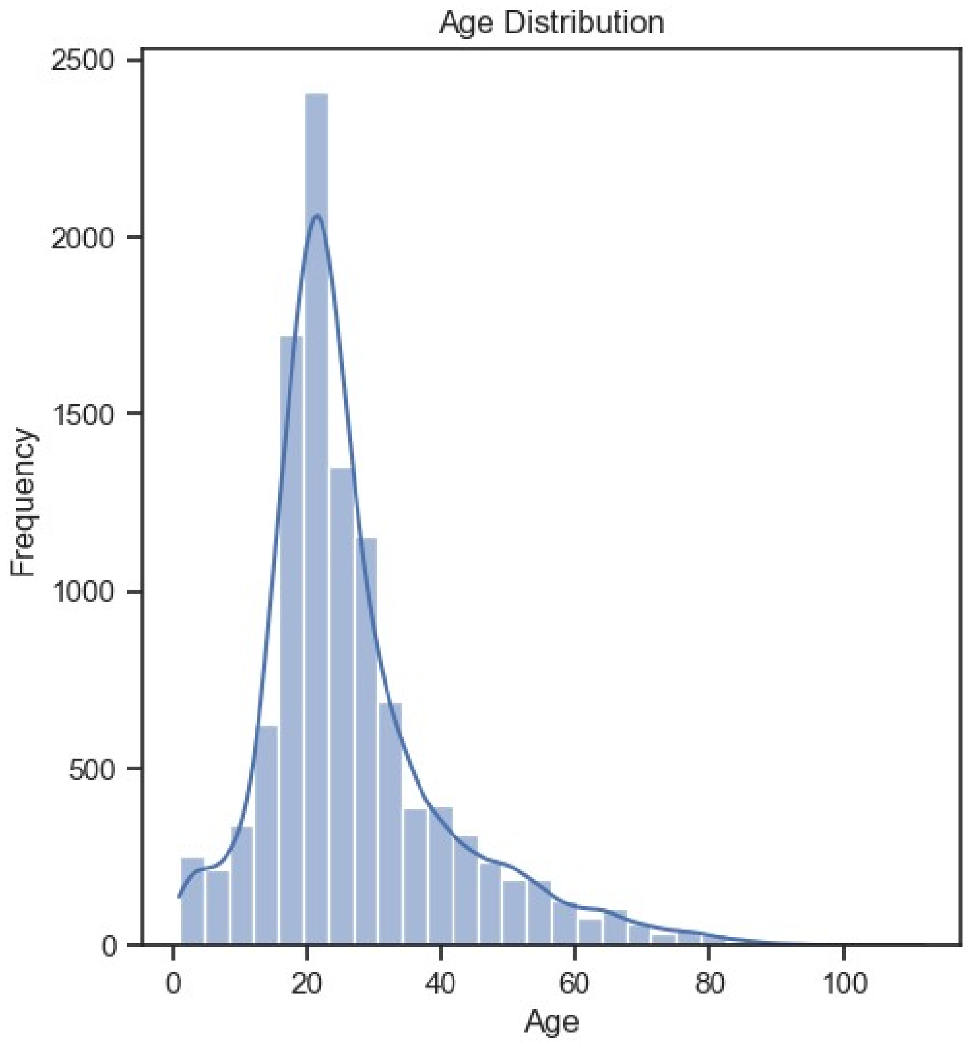
```
conflict.info()
```

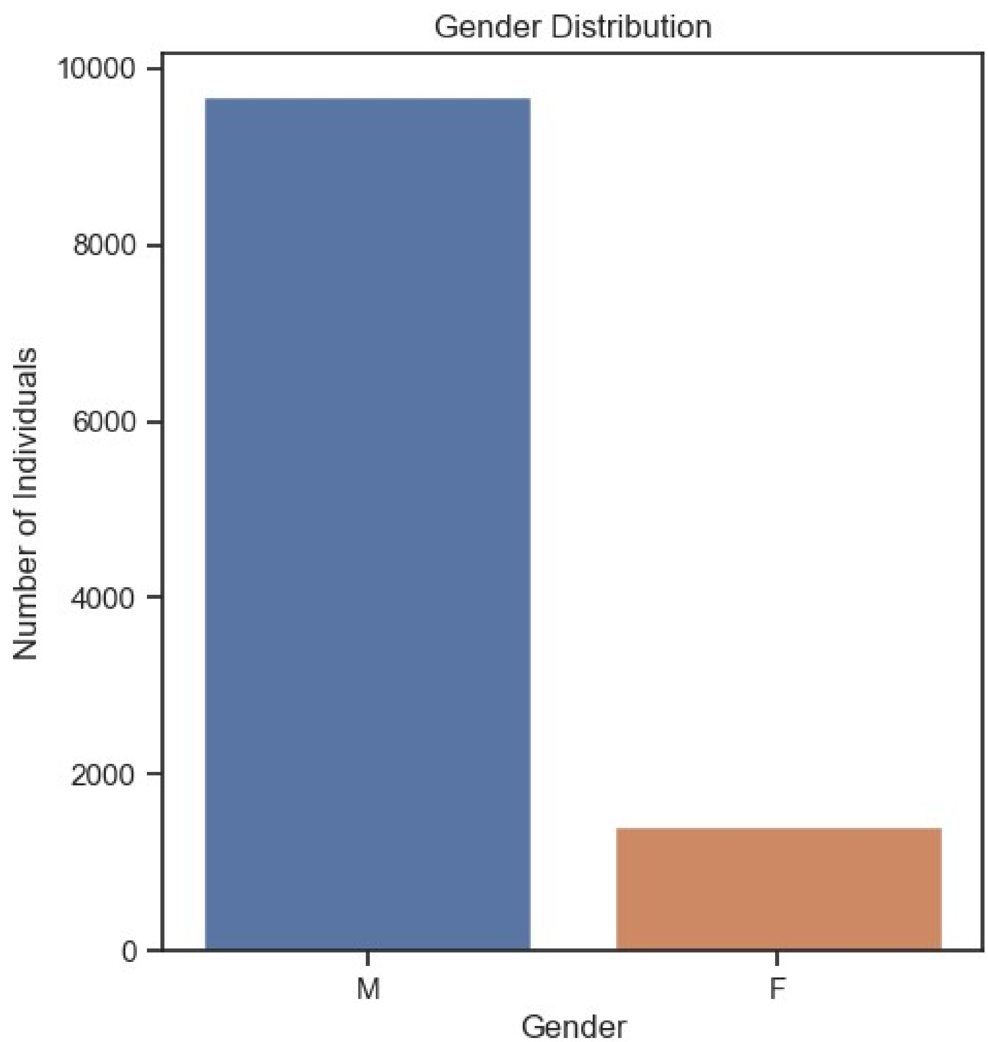
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11124 entries, 0 to 11123
Data columns (total 16 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   name                          11124 non-null  object
 1   date_of_event                 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
```

```
conflict.describe()
```

```
                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

#Checking Age column as Normal or not.
sns.set()
sns.set(style="ticks")
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
ax = sns.histplot(conflict['age'], bins=30,kde=True)    #Plotting
Histogram for age
ax.set(xlabel='Age', ylabel='Frequency', title='Age Distribution')


#Checking Skewness towards one gender
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 2)
ax= sns.countplot(x='gender', data=conflict)   # Plotting bar chart
for gender
ax.set(xlabel='Gender', ylabel='Number of Individuals', title='Gender
Distribution')
plt.show()
```


Age Distribution
Frequency
Age
0
500
1000
1500
2000
2500
0
20
40
60
80
100




```
#Filling Null values in Age and Gender Column
conflict['age'].fillna(conflict['age'].median(), inplace=True)
conflict['gender'].fillna(conflict['gender'].mode()[0], inplace=True)

conflict.isnull().sum()
```

```
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     1430
place_of_residence                 68
place_of_residence_district        68
```

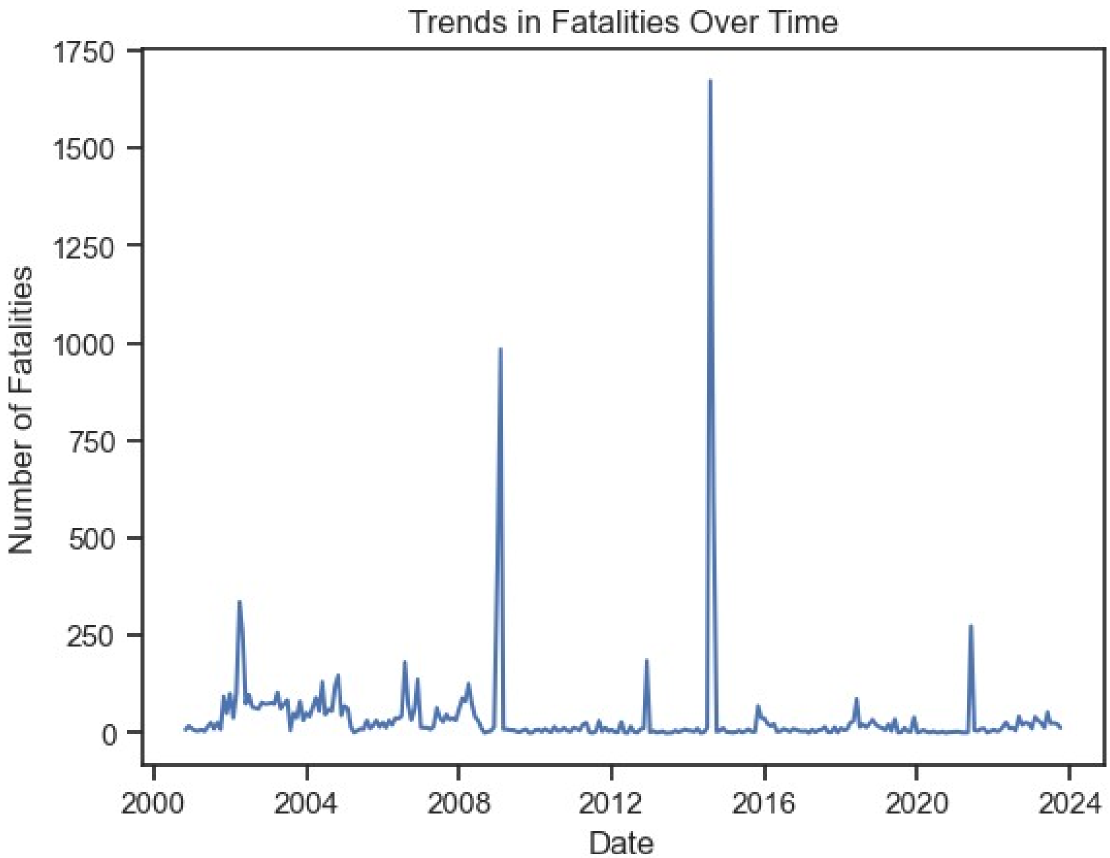
```
type_of_injury                      291
ammunition                         5253
killed_by                             0
notes                               280
dtype: int64
```

Q1- Identify the trends in Fatalities over Time

```
# Checking the range of dates (Fatalities)
print("\nRange of dates:")
print(f"Minimum Date: {conflict['date_of_event'].min()}")
print(f"Maximum Date: {conflict['date_of_event'].max()}")
```

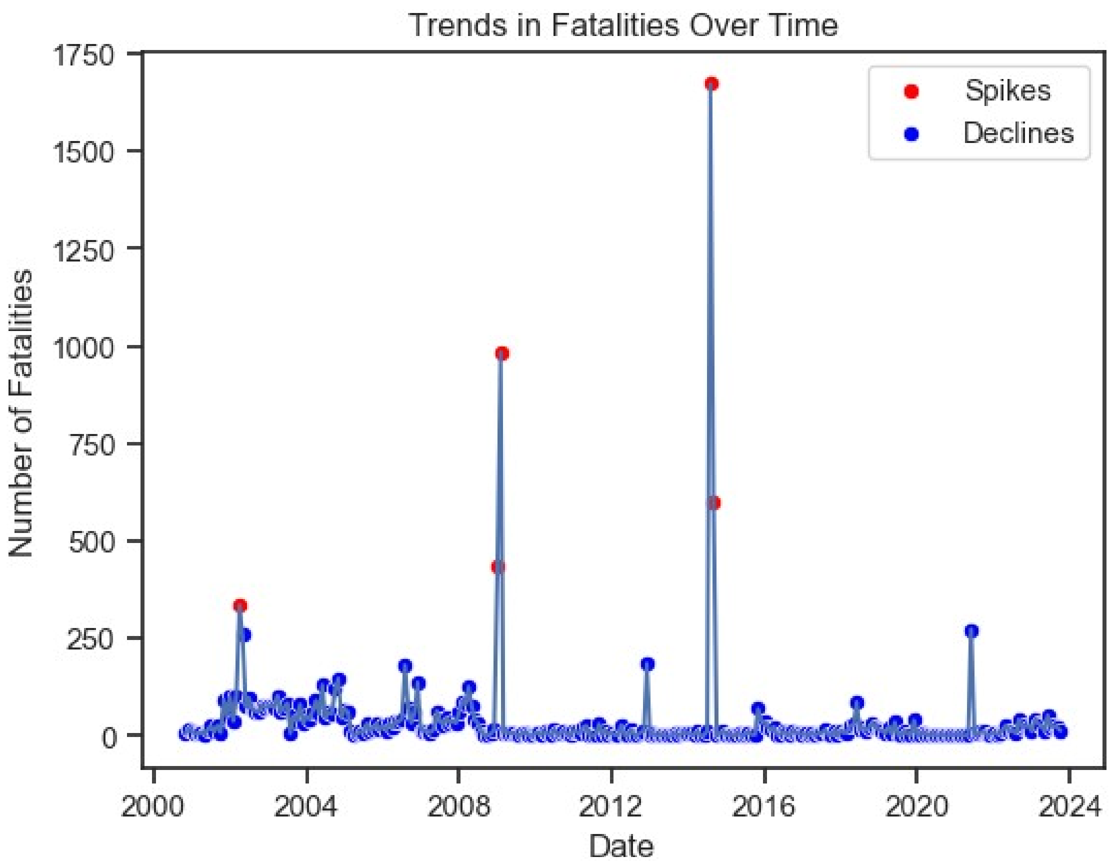
```
Range of dates:
Minimum Date: 2000-10-02
Maximum Date: 2023-09-24
```

```
#Converting to datetime format
conflict['date_of_event']= pd.to_datetime(conflict['date_of_event'])

#Time Series Analysis (On monthly basis)
time_Series= conflict.set_index('date_of_event')
['name'].resample('M').count()
ax= sns.lineplot(x=time_Series.index, y=time_Series)
ax.set(xlabel='Date', ylabel='Number of Fatalities', title='Trends in
Fatalities Over Time')
plt.show()
```



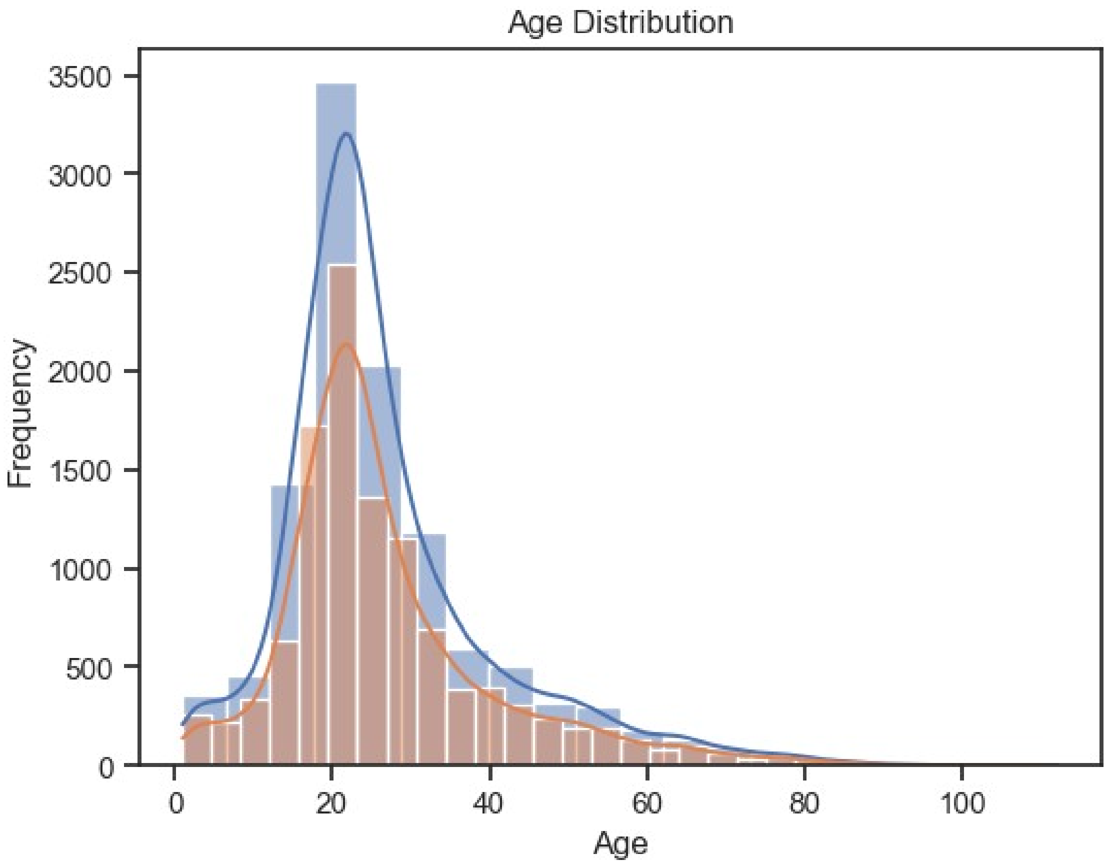

```
#Analysing the trend in Fatalities with Time series plot
time_Series = conflict.set_index('date_of_event')
['name'].resample('M').count()

#Calculating the threshold value for Spikes and Declines
mean_fatal = time_Series.mean()
std_fatal = time_Series.std()
threshold = mean_fatal + 2*std_fatal

ax= sns.lineplot(x=time_Series.index, y=time_Series)

#Scatterplot for Spikes
spikes = time_Series[time_Series > threshold]
sns.scatterplot(x=spikes.index, y=spikes, color='red', label='Spikes')

#Scatterplot for declines
declines = time_Series[time_Series < threshold]
sns.scatterplot(x=declines.index, y=declines, color='blue',
label='Declines')

ax.set(xlabel='Date', ylabel='Number of Fatalities', title='Trends in
Fatalities Over Time')
```

```
plt.legend()
plt.show()
```

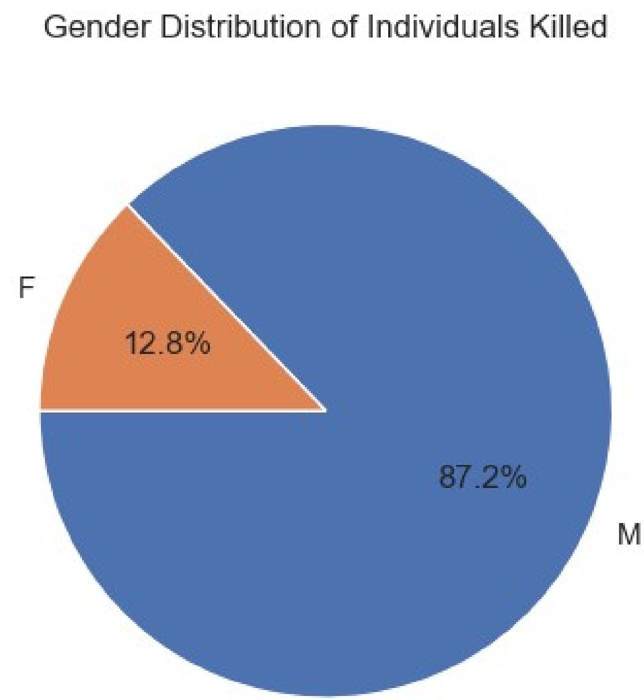


Q2-Examining age,gender,and citizenship of Individuals Killed

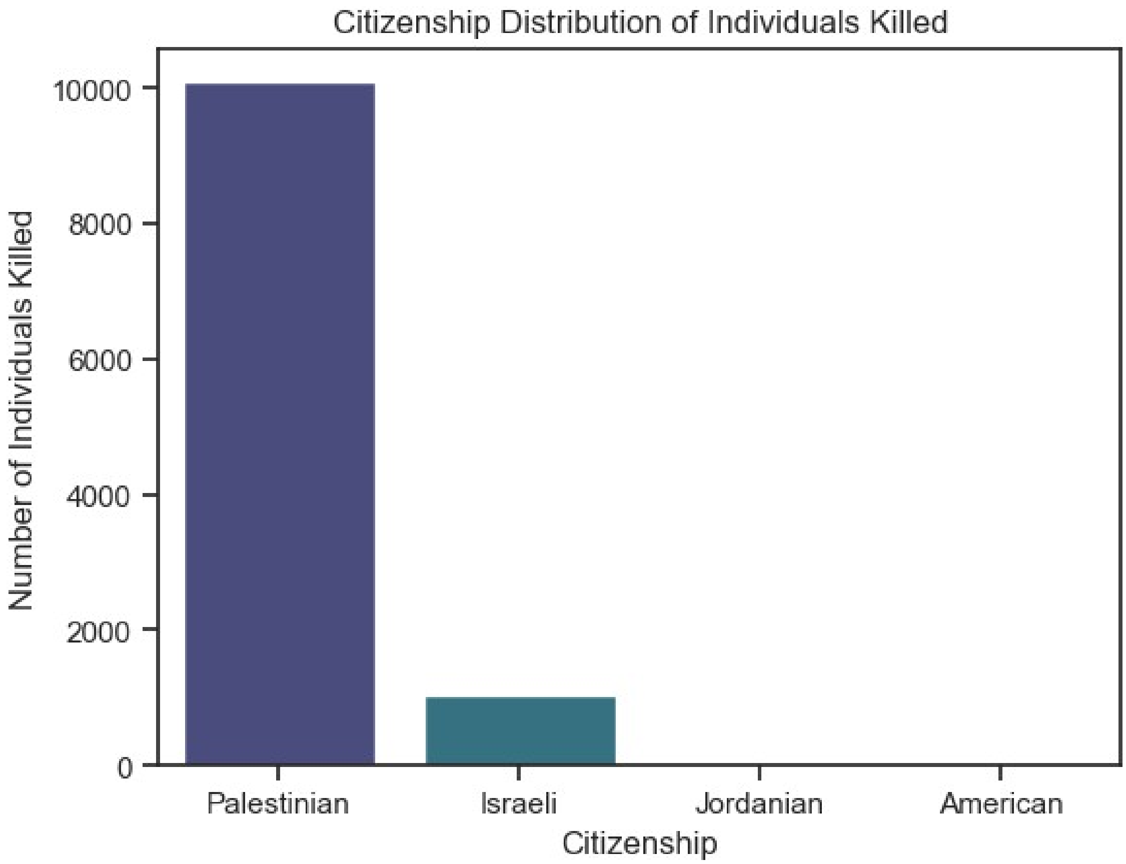
```
# Analysing the Age Distribution with Histogram
sns.histplot(conflict['age'], bins=20, kde=True)
ax = sns.histplot(conflict['age'], bins=30,kde=True)   #Plotting
Histogram for age
ax.set(xlabel='Age', ylabel='Frequency', title='Age Distribution')
plt.show()
```



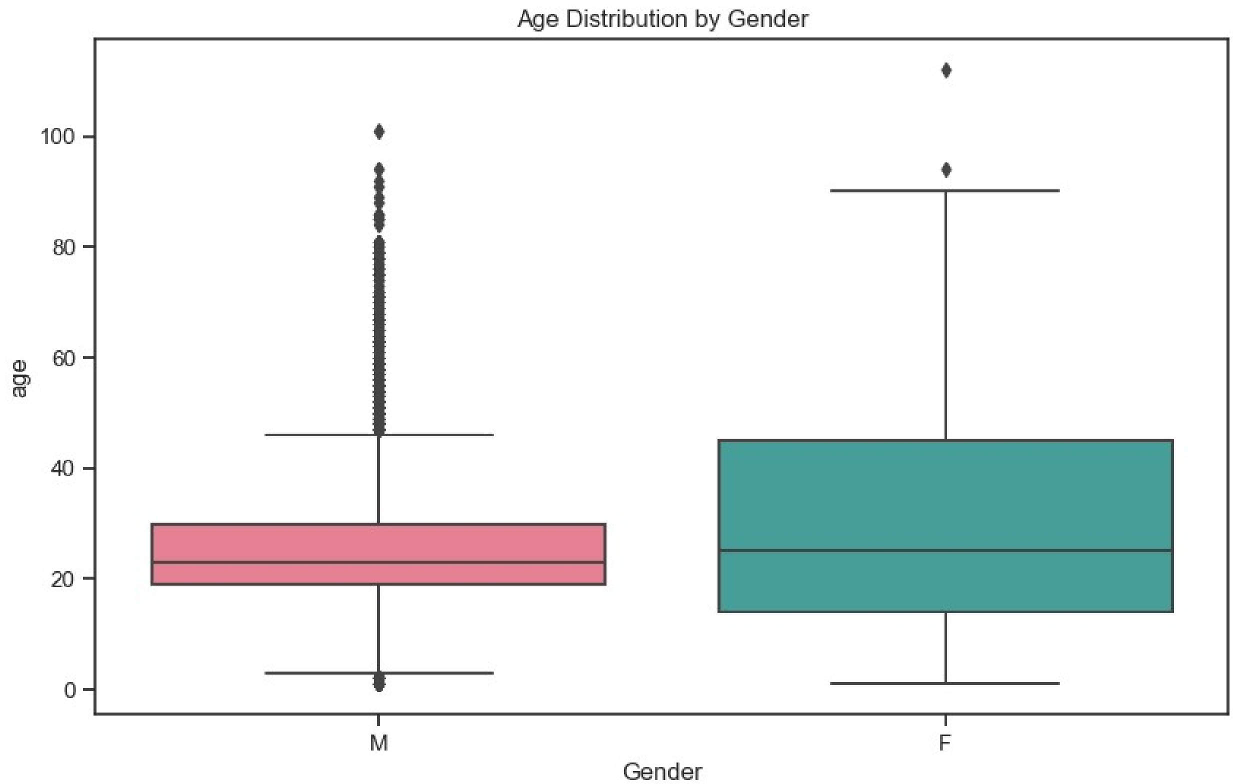

```
# Analysing Gender distribution with Pie chart
gender_counts = conflict['gender'].value_counts()
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%',
startangle=180)
plt.title('Gender Distribution of Individuals Killed')
plt.show()
```



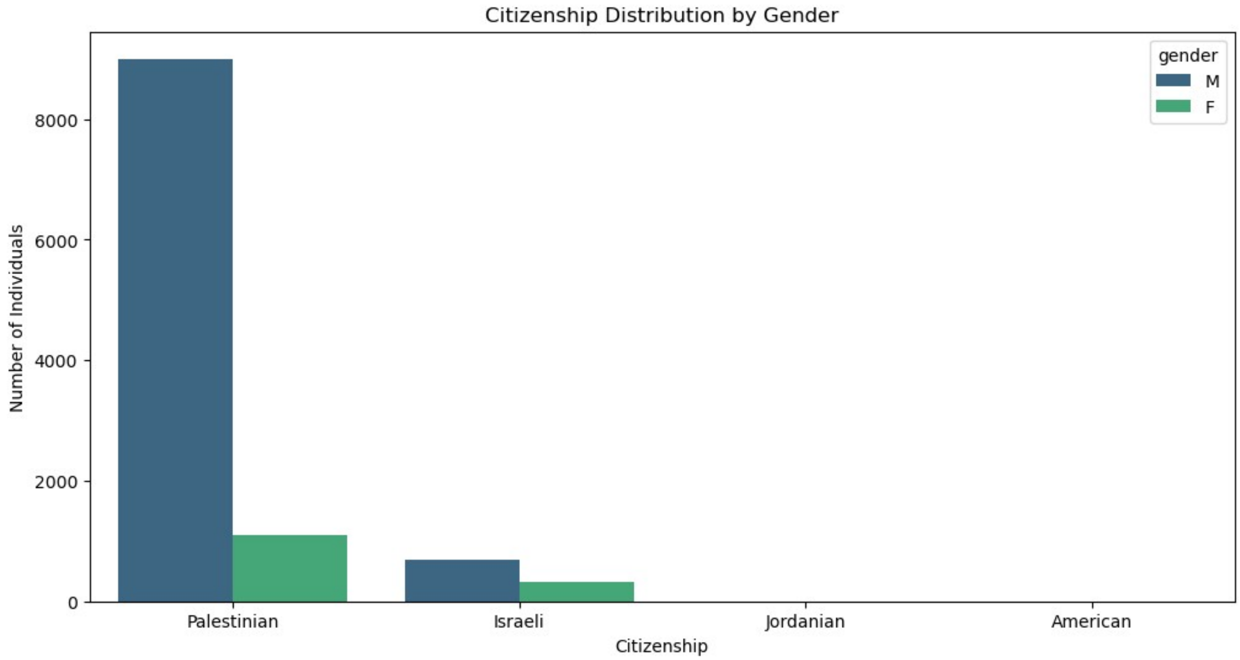

```
# Analysing Citizenship distribution with bar chart
citizenship_counts = conflict['citizenship'].value_counts()
ax = sns.barplot(x=citizenship_counts.index,
y=citizenship_counts.values, palette='viridis')
ax.set(xlabel='Citizenship', ylabel='Number of Individuals Killed',
       title='Citizenship Distribution of Individuals Killed')
plt.show()
```



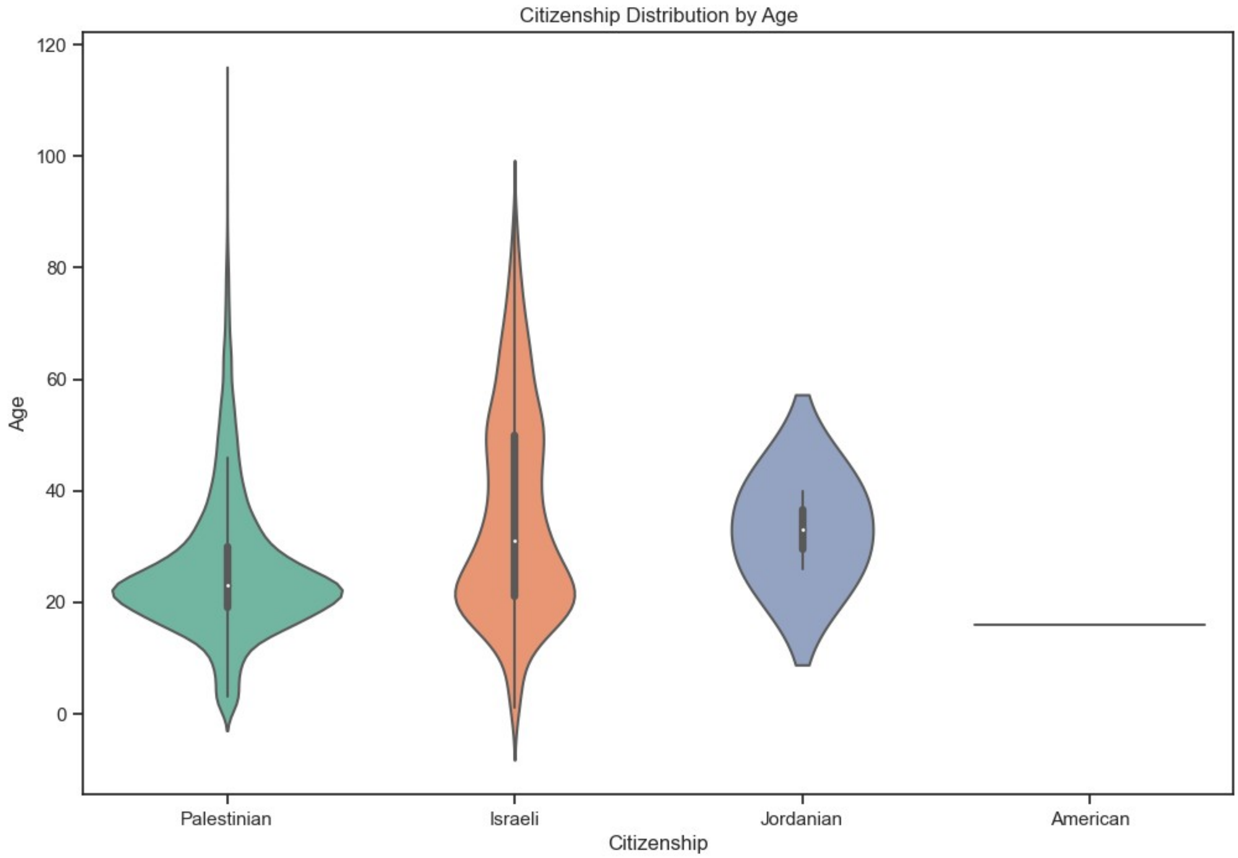

```
# Analyzing disparities using Boxplot for age distribution by gender
plt.figure(figsize=(10, 6))
ax = sns.boxplot(x='gender', y='age', data=conflict, palette='husl')
ax.set(xlabel='Gender', ylabel='age', title='Age Distribution by
Gender')
plt.show()
```




```
# Analyzing disparities using countplot for citizenship distribution
by gender
plt.figure(figsize=(12, 6))
ax = sns.countplot(x='citizenship', hue='gender', data=conflict,
palette='viridis')
ax.set(xlabel='Citizenship', ylabel='Number of Individuals',
title='Citizenship Distribution by Gender')
plt.show()
```




```
# Analyzing disparities using Violin plot for citizenship distribution
by age
plt.figure(figsize=(12, 8))
ax = sns.violinplot(x='citizenship', y='age', data=conflict,
palette='Set2')
ax.set(xlabel='Citizenship', ylabel='Age', title='Citizenship
Distribution by Age')
plt.show()
```



Q-3 Indentify Areas have Higher Levels of Violence

```
# Total counts Category wise in Region Column
conflict['event_location_region'].value_counts()
```

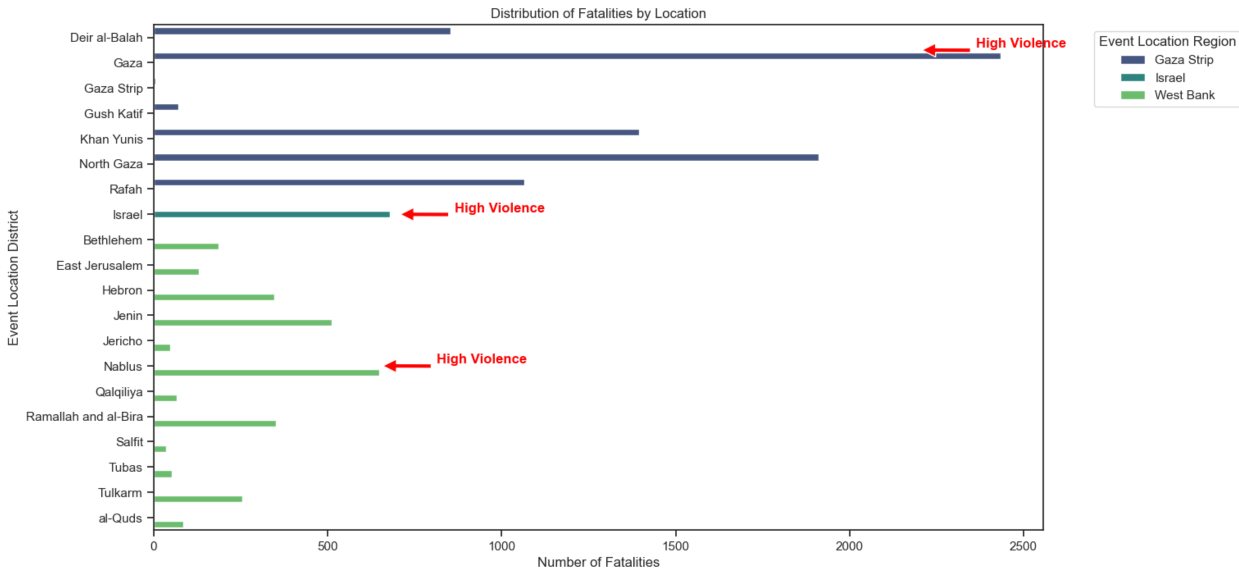
```
Gaza Strip    7733
West Bank     2712
Israel         679
Name: event_location_region, dtype: int64
```

```
# Total counts Category wise in District Column
conflict['event_location_district'].value_counts()
```

```
Gaza                        2435
North Gaza                  1910
Khan Yunis                  1394
Rafah                       1066
Deir al-Balah                854
Israel                       679
Nablus                       647
```

```
Jenin                       512
Ramallah and al-Bira        350
Hebron                      347
Tulkarm                     254
Bethlehem                   186
East Jerusalem              130
al-Quds                      85
Gush Katif                   70
Qalqiliya                    65
Tubas                        52
Jericho                      48
Salfit                       36
Gaza Strip                    4
Name: event_location_district, dtype: int64
```

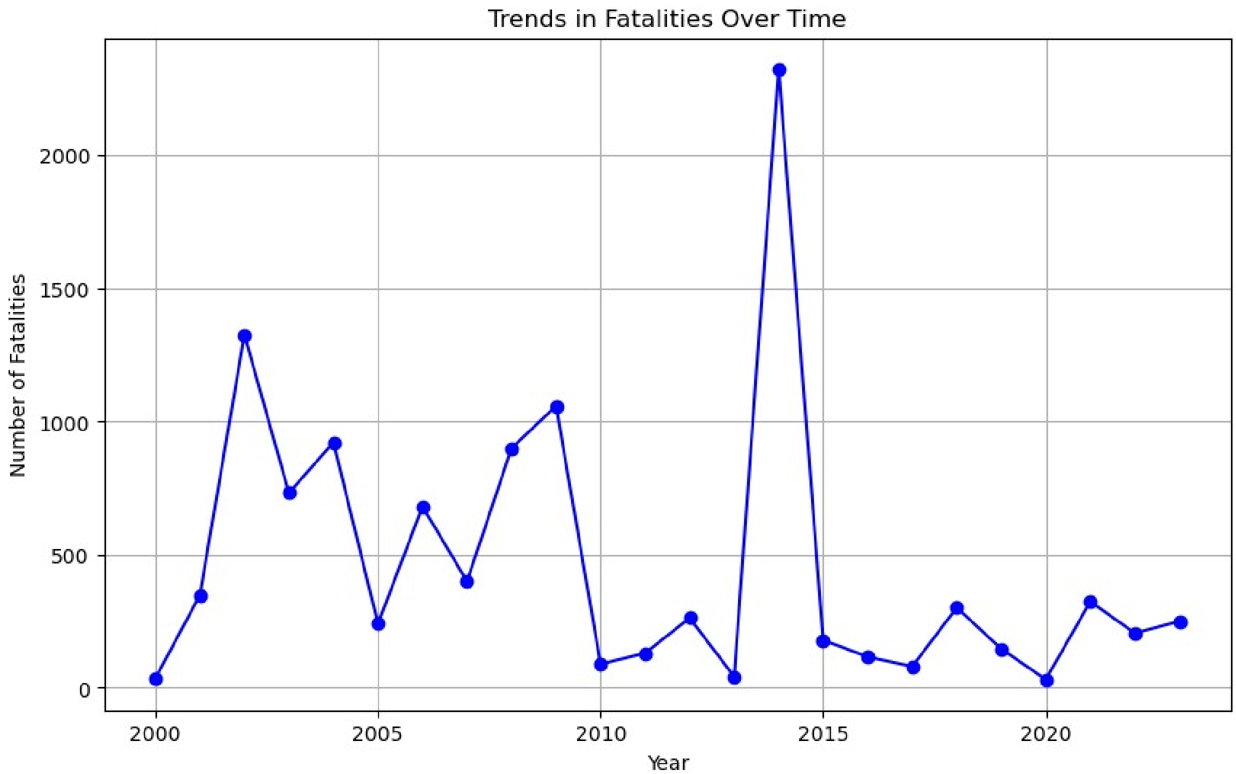
```
# Applying Group by on District and Region wise
location_counts = conflict.groupby(['event_location_region',
'event_location_district']).size().reset_index(name='fatalities_count'
)

# Plotting Bar chart for Fatalities Distribution
plt.figure(figsize=(14, 8))
ax = sns.barplot(x='fatalities_count', y='event_location_district',
hue='event_location_region', data=location_counts, dodge=True,
palette='viridis')
ax.set(xlabel='Number of Fatalities', ylabel='Event Location
District', title='Distribution of Fatalities by Location')
plt.legend(title='Event Location Region', bbox_to_anchor=(1.05, 1),
loc='upper left')


# Adding Indicators on Region where High Violence exist
annotations = [
    {'label': 'High Violence', 'xy': (2200, 0.5),'xytext': (50, 2)},
    {'label': 'High Violence', 'xy': (700, 7), 'xytext': (50, 2)},
    {'label': 'High Violence', 'xy': (650, 13), 'xytext': (50, 2)}
]

for annotation in annotations:
    plt.annotate(annotation['label'], xy=annotation['xy'],
xytext=annotation['xytext'], textcoords='offset points',
                 arrowprops=dict(facecolor='red', shrink=0.05),
color='red', fontsize=12, weight='bold')
plt.show()
```



Q 4

Group by year and calculate the sum of fatalities for each year

```
conflict['date_of_death'] = pd.to_datetime(conflict['date_of_death'],
errors='coerce')

fatalities_over_time =
conflict.groupby(conflict['date_of_death'].dt.year)['name'].count()

# Plotting the trends
plt.figure(figsize=(10, 6))
fatalities_over_time.plot(kind='line', marker='o', linestyle='-',
color='b')
plt.title('Trends in Fatalities Over Time')
plt.xlabel('Year')
plt.ylabel('Number of Fatalities')
plt.grid(True)
plt.show()
```

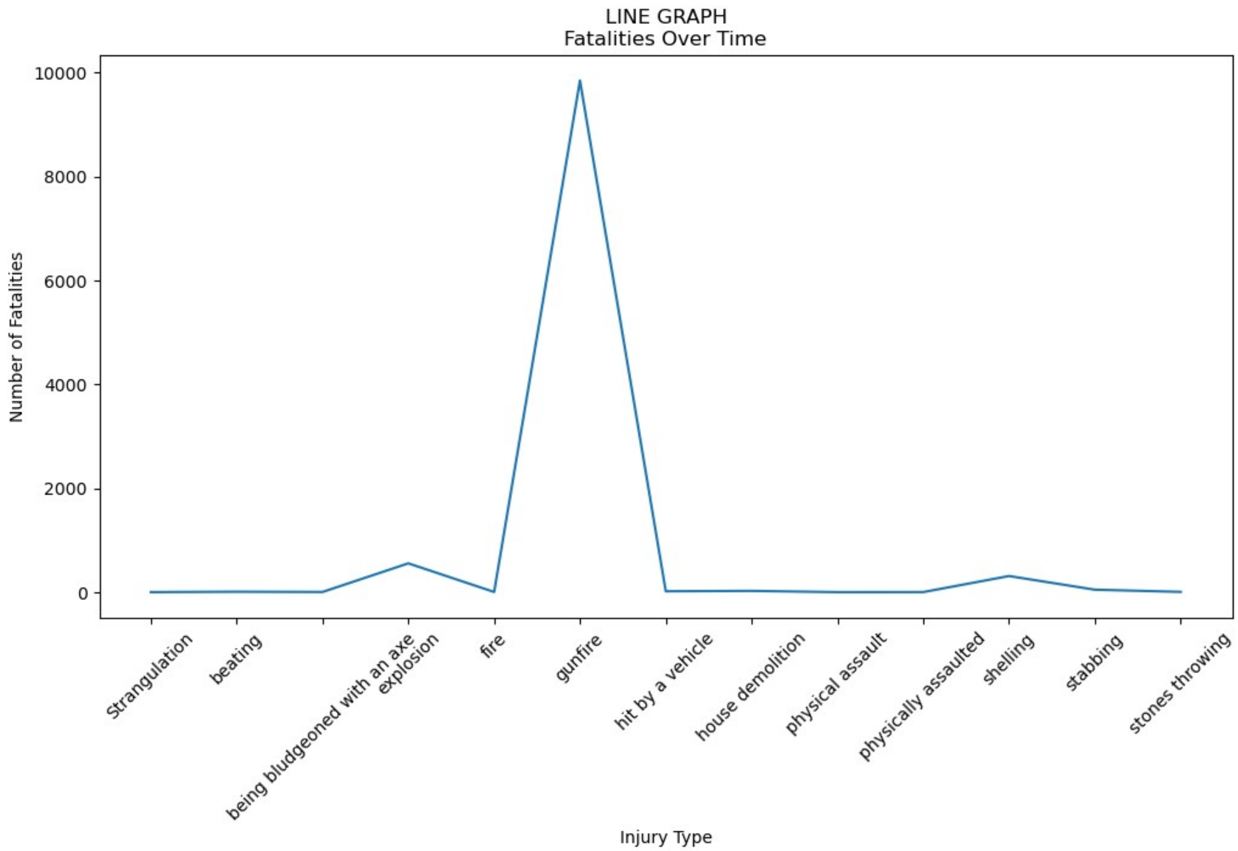


group the data by injuries

```
fatalities_by_injury =
conflict.groupby('type_of_injury').size().reset_index(name='fatalities
_count')
print(fatalities_by_injury)

plt.figure(figsize=(12, 6))
plt.plot(fatalities_by_injury['type_of_injury'],
fatalities_by_injury['fatalities_count'])
plt.title('\nLINE GRAPH\nFatalities Over Time')
plt.xlabel('Injury Type')
plt.ylabel('Number of Fatalities')
plt.xticks(rotation=45)
print(plt.show())
```

```
                  type_of_injury  fatalities_count
0                  Strangulation                 1
1                        beating                 9
2   being bludgeoned with an axe                 4
3                      explosion               555
4                           fire                 4
5                        gunfire              9849
6               hit by a vehicle                18
7               house demolition                25
8               physical assault                 1
```

```
9          physically assaulted                2
10                     shelling              311
11                     stabbing               48
12              stones throwing                6
```



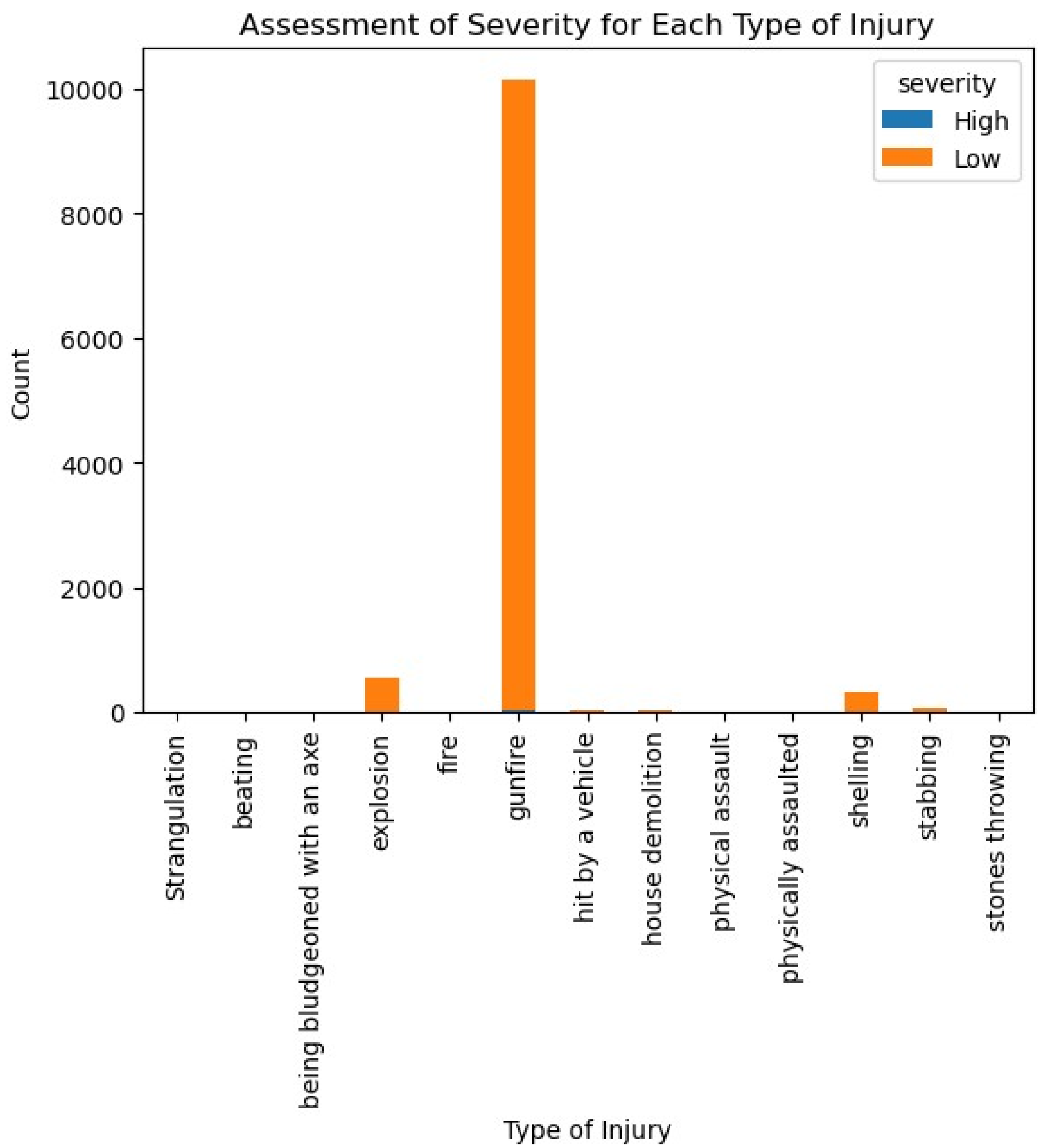

```
None
```

Severity distribution for each injury.

```
conflict['type_of_injury'].fillna(conflict['type_of_injury'].mode()
[0], inplace=True)
conflict['ammunition'].fillna(conflict['ammunition'].mode()[0],
inplace=True)
conflict['age'].fillna(conflict['age'].median(), inplace=True)
nan_count_age = conflict['age'].isnull().sum()
print("Number of NaN values in 'age' column after handling null
values:", nan_count_age)

conflict['severity'] = conflict['notes'].apply(lambda x: 'High' if
'serious' in str(x).lower() else 'Low')

common_injuries = conflict['type_of_injury'].value_counts()
```

```python
print("Most Common Types of Injuries:")
print(common_injuries)

plt.figure(figsize=(10, 6))
conflict.groupby('type_of_injury')
['severity'].value_counts().unstack().plot(kind='bar', stacked=True)
plt.title('Assessment of Severity for Each Type of Injury')
plt.xlabel('Type of Injury')
plt.ylabel('Count')
plt.show()
```

```
Number of NaN values in 'age' column after handling null values: 0
Most Common Types of Injuries:
gunfire                            10140
explosion                            555
shelling                             311
stabbing                              48
house demolition                      25
hit by a vehicle                      18
beating                                9
stones throwing                        6
being bludgeoned with an axe           4
fire                                   4
physically assaulted                   2
physical assault                       1
Strangulation                          1
Name: type_of_injury, dtype: int64

<Figure size 1000x600 with 0 Axes>
```



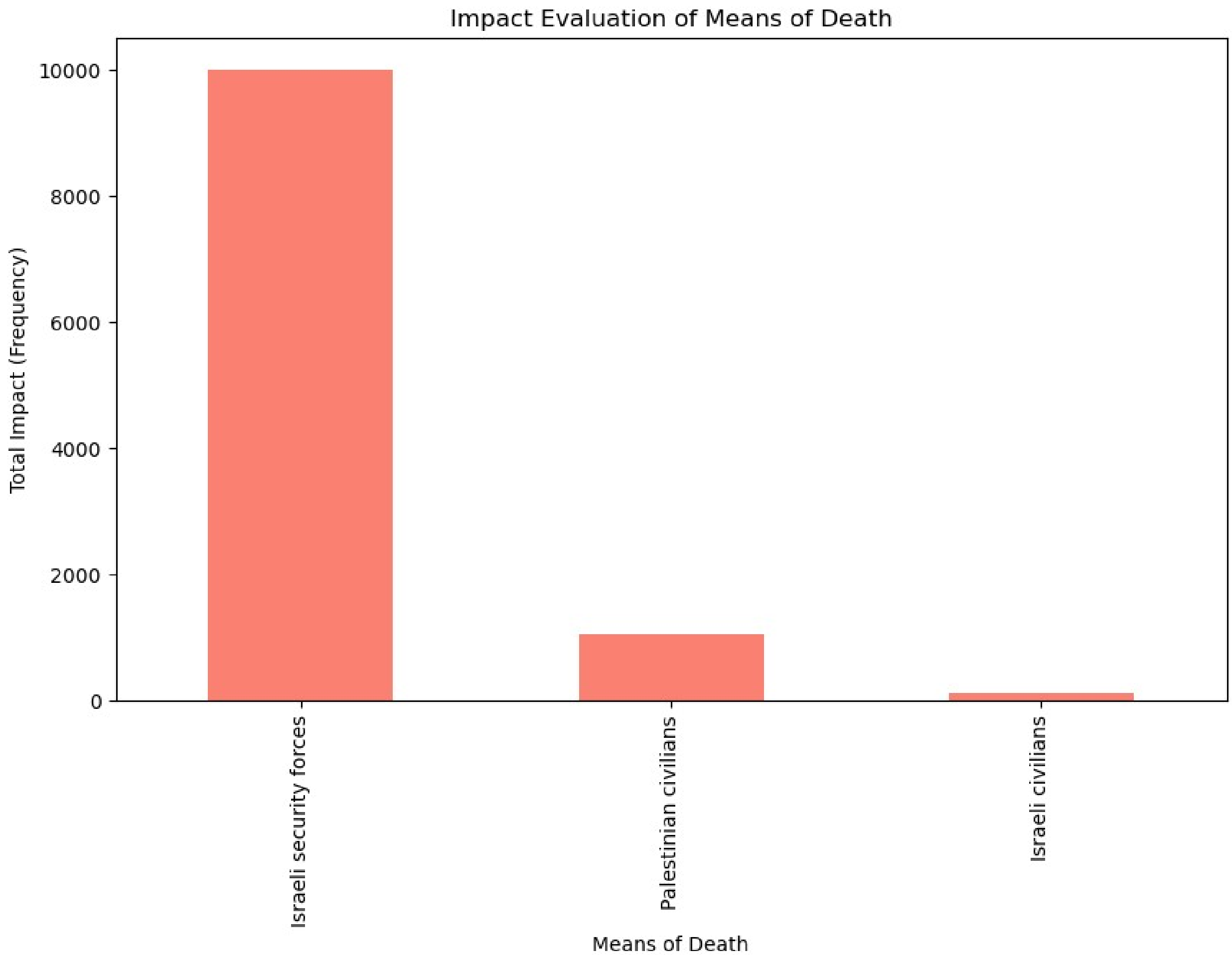
Visualizing impact of means of death

```
common_ammunition = conflict['ammunition'].value_counts()

print("Most Common Types of Ammunition:")
print(common_ammunition)

common_means_of_death = conflict['killed_by'].value_counts()

print("\nMost Common Means of Death:")
print(common_means_of_death)
```

```
impact_evaluation = conflict.groupby('killed_by')
['type_of_injury'].value_counts().unstack().fillna(0)
impact_evaluation['Total'] = impact_evaluation.sum(axis=1)
impact_evaluation = impact_evaluation.sort_values(by='Total',
ascending=False)


print("\nImpact Evaluation of Means of Death:")
print(impact_evaluation)

plt.figure(figsize=(10, 6))
impact_evaluation['Total'].plot(kind='bar', color='salmon')
plt.title('Impact Evaluation of Means of Death')
plt.xlabel('Means of Death')
plt.ylabel('Total Impact (Frequency)')
plt.show()
```

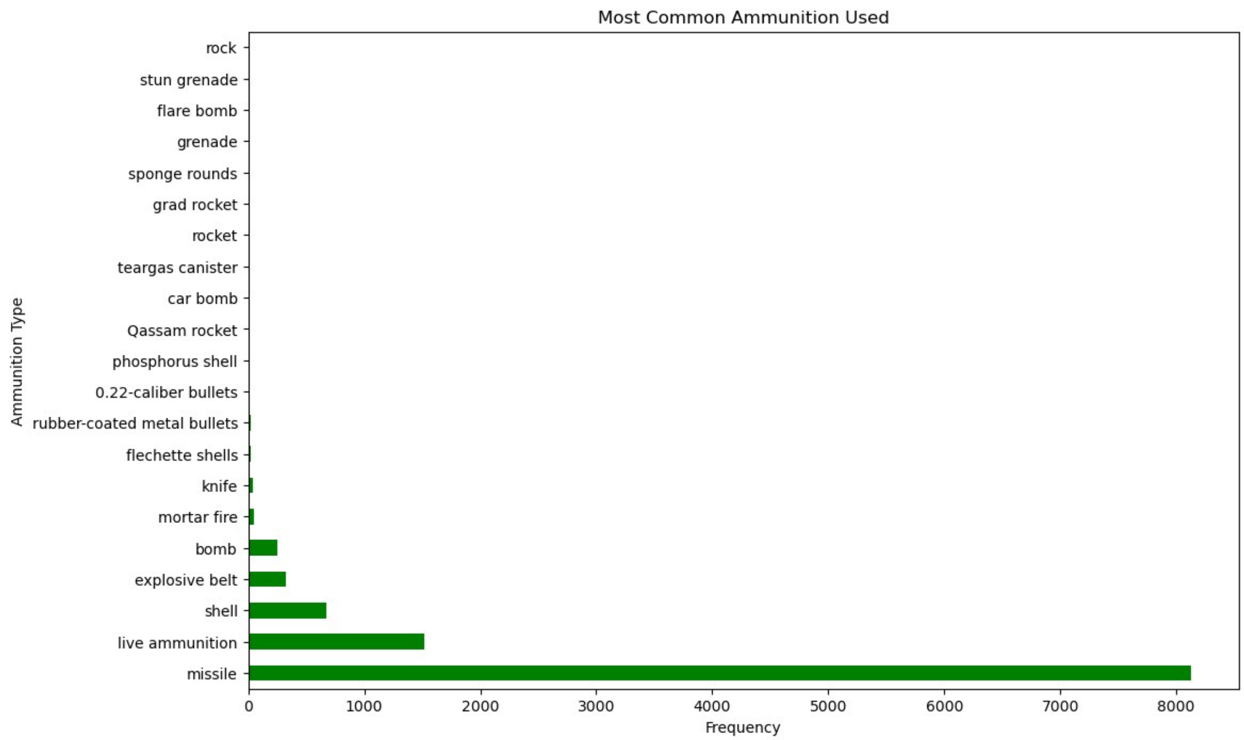
```
Most Common Types of Ammunition:
missile                         8130
live ammunition                 1514
shell                            675
explosive belt                   326
bomb                             249
mortar fire                       51
knife                             37
flechette shells                  22
rubber-coated metal bullets       19
0.22-caliber bullets              16
phosphorus shell                  16
Qassam rocket                     15
car bomb                          15
teargas canister                  13
rocket                            12
grad rocket                        7
sponge rounds                      2
grenade                            2
flare bomb                         1
stun grenade                       1
rock                               1
Name: ammunition, dtype: int64

Most Common Means of Death:
Israeli security forces    10000
Palestinian civilians       1028
Israeli civilians             96
Name: killed_by, dtype: int64

Impact Evaluation of Means of Death:
type_of_injury           Strangulation  beating  being bludgeoned with
an axe  \
```

```
killed_by

Israeli security forces              0.0      5.0
0.0
Palestinian civilians                1.0      4.0
4.0
Israeli civilians                    0.0      0.0
0.0

type_of_injury           explosion  fire  gunfire  hit by a vehicle  \
killed_by
Israeli security forces       47.0   1.0   9606.0               3.0
Palestinian civilians        508.0   0.0    448.0              14.0
Israeli civilians              0.0   3.0     86.0               1.0

type_of_injury           house demolition  physical assault  \
killed_by
Israeli security forces              25.0               0.0
Palestinian civilians                 0.0               1.0
Israeli civilians                     0.0               0.0

type_of_injury           physically assaulted  shelling  stabbing  \
killed_by
Israeli security forces                   1.0     311.0       1.0
Palestinian civilians                     1.0       0.0      43.0
Israeli civilians                         0.0       0.0       4.0

type_of_injury           stones throwing    Total
killed_by
Israeli security forces              0.0  10000.0
Palestinian civilians                4.0   1028.0
Israeli civilians                    2.0     96.0
```



Most Common Ammunition Used

```
plt.figure(figsize=(12, 8))
conflict['ammunition'].value_counts().plot(kind='barh', color='green')
plt.title('Most Common Ammunition Used')
plt.xlabel('Frequency')
plt.ylabel('Ammunition Type')
plt.show()
```

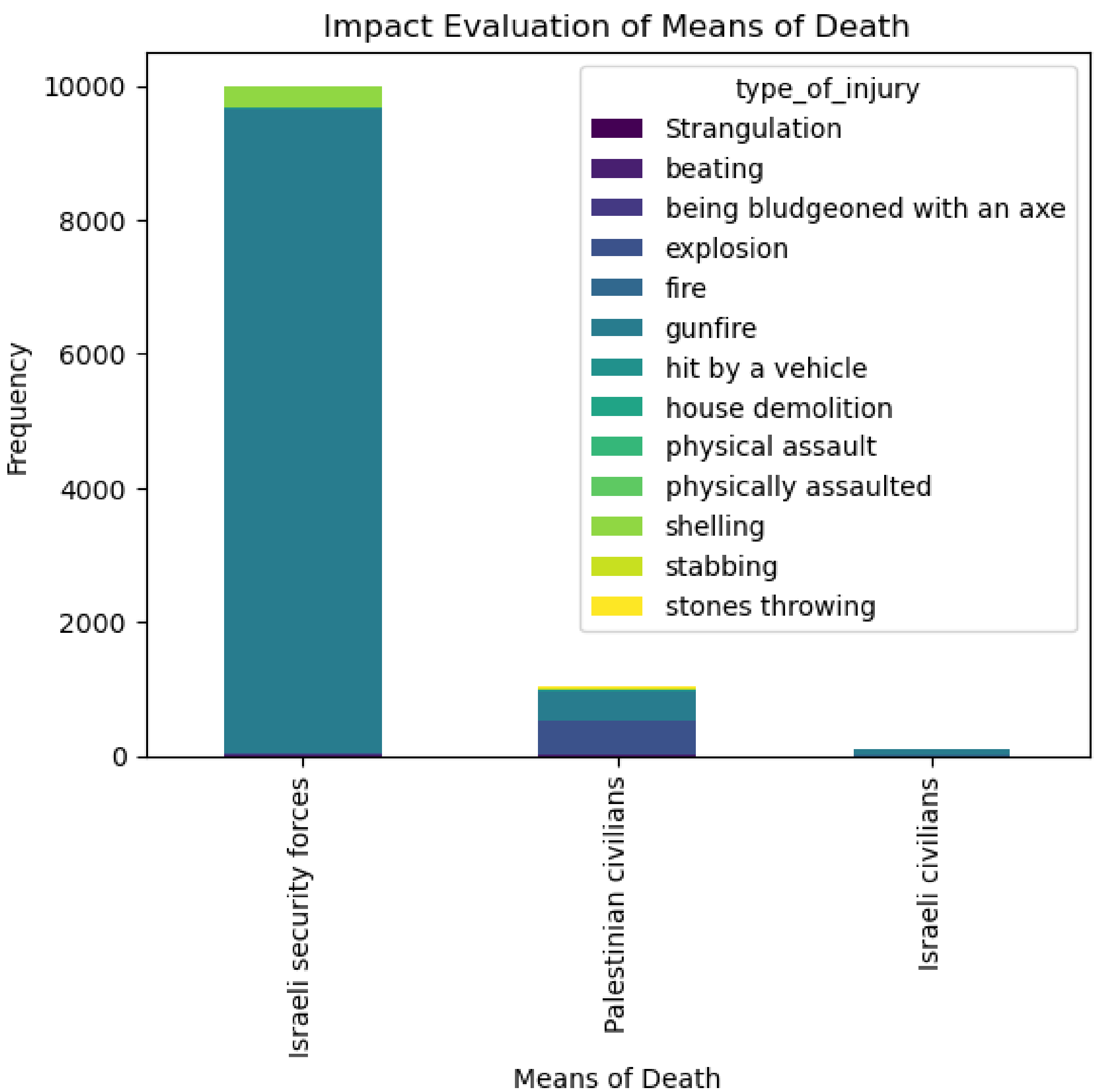


reason to use

```
 horizontal bar chart is used to display the most common ammunition
types, providing a clear comparison of their frequencies. The choice
of a horizontal bar chart allows for easy readability of ammunition
types on the y-axis with corresponding frequencies on the x-axis.
```

Impact Evaluation of Means of Death

```
plt.figure(figsize=(12, 8))
impact_evaluation.drop('Total', axis=1).plot(kind='bar', stacked=True,
colormap='viridis')
plt.title('Impact Evaluation of Means of Death')
plt.xlabel('Means of Death')
plt.ylabel('Frequency')
plt.show()

<Figure size 1200x800 with 0 Axes>
```



reason to use

```
A stacked bar chart is used to illustrate the distribution of
fatalities across different means of death, allowing a visual
comparison of the contribution of each category to the total impact.
The colormap 'viridis' enhances readability by providing a
perceptually uniform color scheme, aiding in better interpretation of
the data
```

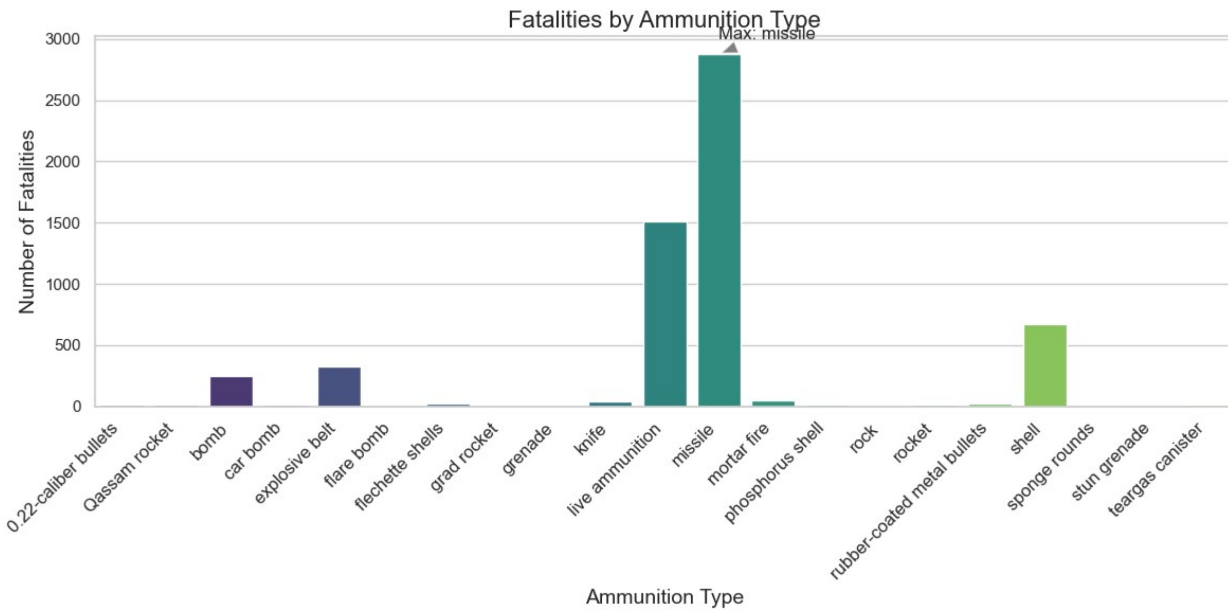
Question 5

Age Distribution of Victims

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

```
conflict = conflict.dropna(subset=['ammunition'])

fatalities_by_ammunition =
conflict.groupby('ammunition').size().reset_index(name='fatalities_cou
nt')

max_fatalities_ammunition =
fatalities_by_ammunition.loc[fatalities_by_ammunition['fatalities_coun
t'].idxmax()]['ammunition']
print("\nMost fatalities caused by ammunition:",
max_fatalities_ammunition)

sns.set(style="whitegrid", rc={"axes.titlesize": "xx-large",
"axes.labelsize": "x-large"})


plt.figure(figsize=(12, 6))
bar_plot = sns.barplot(x='ammunition', y='fatalities_count',
data=fatalities_by_ammunition, palette='viridis')
bar_plot.set_title('Fatalities by Ammunition Type', fontsize=16)
bar_plot.set_xlabel('Ammunition Type', fontsize=14)
bar_plot.set_ylabel('Number of Fatalities', fontsize=14)
bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation=45,
ha='right', fontsize=12)
bar_plot.yaxis.grid(True)

max_idx = fatalities_by_ammunition['fatalities_count'].idxmax()
bar_plot.annotate(f'Max: {max_fatalities_ammunition}',
                  xy=(max_idx,
fatalities_by_ammunition['fatalities_count'].max()),
                  xytext=(0, 10), textcoords='offset points',
                  arrowprops=dict(facecolor='black',
arrowstyle='wedge,tail_width=0.7', alpha=0.5))

plt.tight_layout()
plt.show()
```

```
Most fatalities caused by ammunition: missile
```



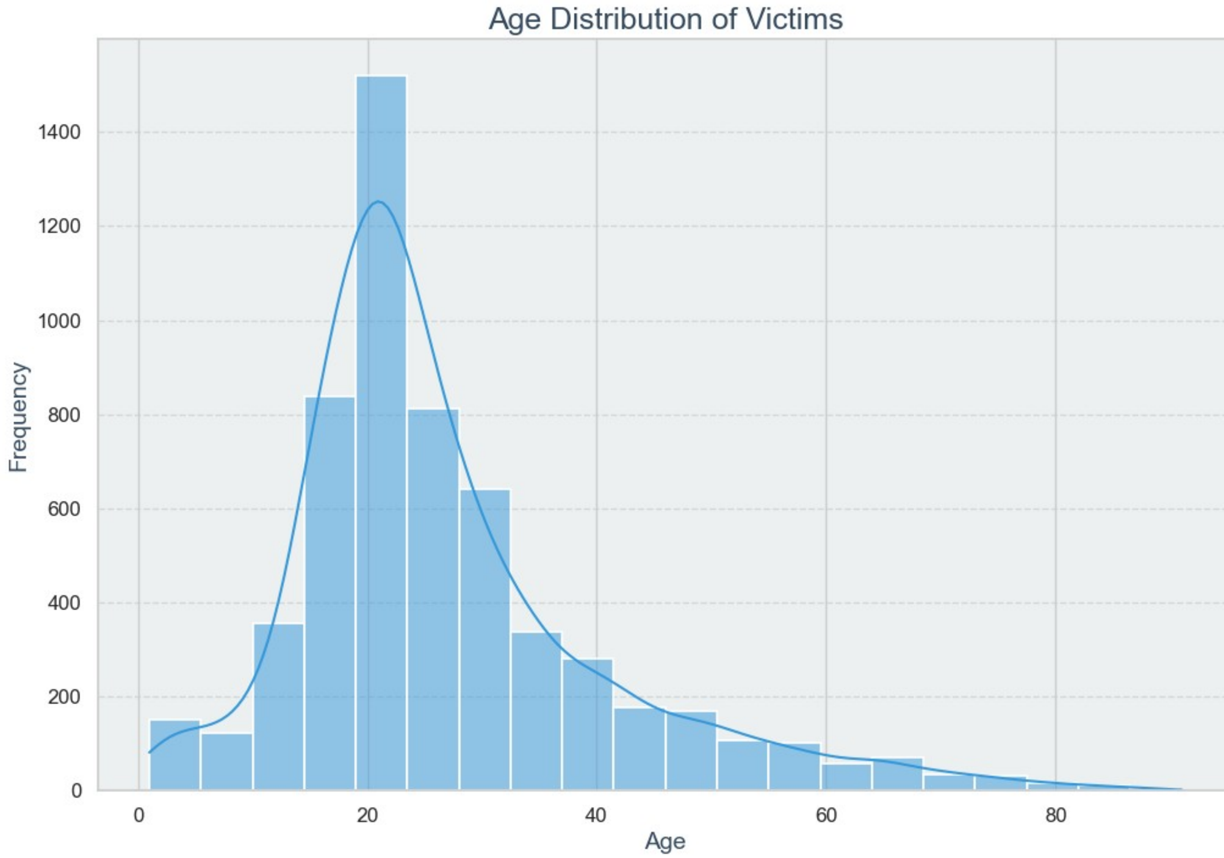
why used bar char in above question

```
A bar chart is used to visually represent the frequency of fatalities
associated with different ammunition types, providing a clear
comparison of fatality counts for each category. The code above
employs a bar chart to display the distribution of fatalities across
various ammunition types in the 'fatalities.csv' dataset.
```

Question 6

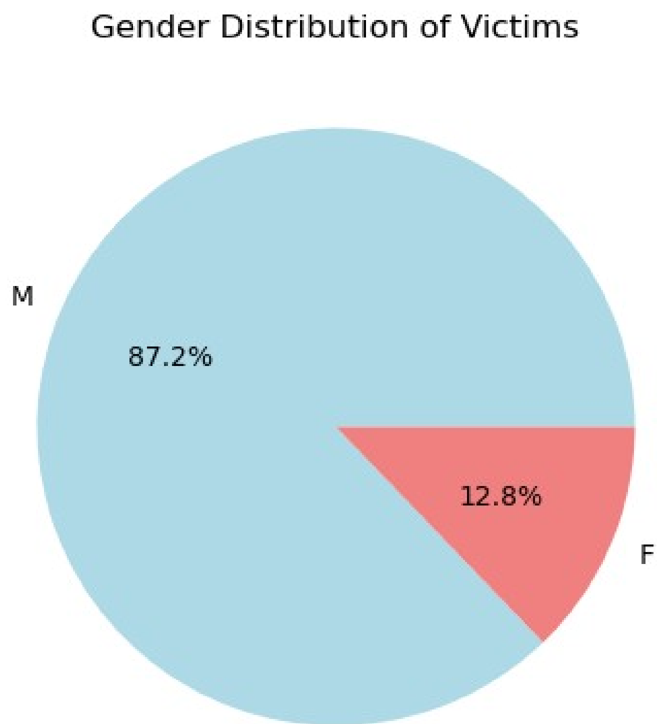
Age distribution of victims

```
sns.set_theme(style="whitegrid")
fig, ax = plt.subplots(figsize=(12, 8))
sns.histplot(data=conflict, x='age', bins=20, kde=True,
color='#3498db', edgecolor='white', linewidth=1.2)
ax.set_title('Age Distribution of Victims', fontsize=18,
color='#34495e')
ax.set_xlabel('Age', fontsize=14, color='#34495e')
ax.set_ylabel('Frequency', fontsize=14, color='#34495e')
ax.grid(axis='y', linestyle='--', alpha=0.7)
ax.tick_params(axis='both', which='major', labelsize=12,
color='#34495e')
ax.set_facecolor('#ecf0f1')
plt.show()
```



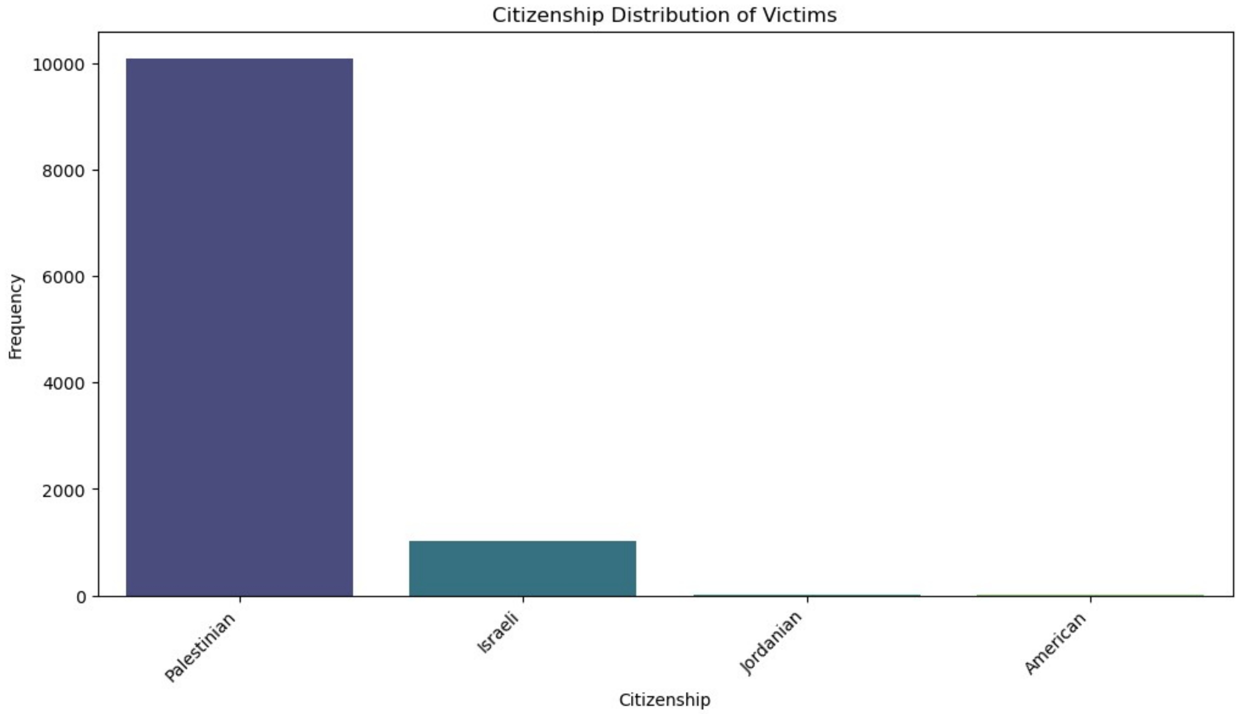
Gender distribution of victims

```
plt.figure(figsize=(8, 5))
conflict['gender'].value_counts().plot(kind='pie', autopct='%1.1f%%',
colors=['lightblue', 'lightcoral'])
plt.title('Gender Distribution of Victims')
plt.ylabel('')
plt.show()
```



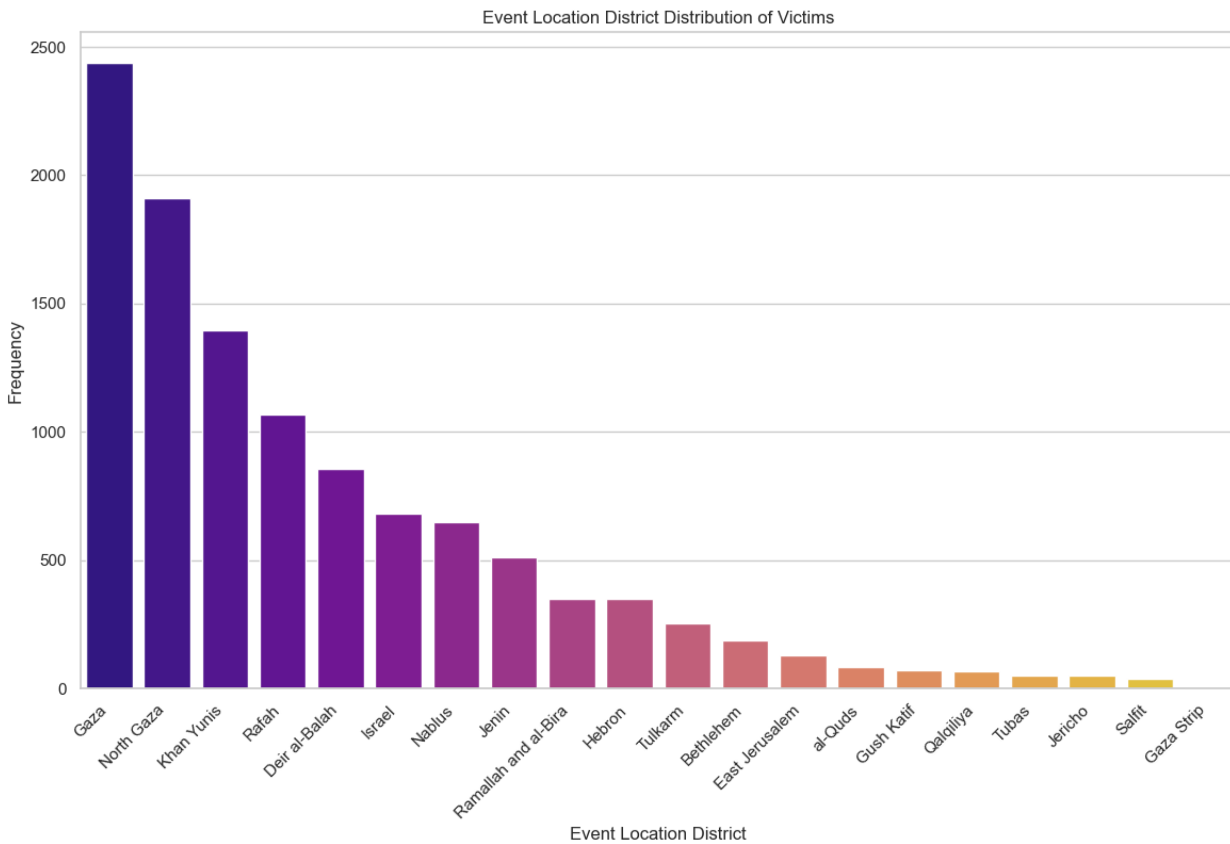
Citizenship Distribution

```
plt.figure(figsize=(12, 6))
sns.countplot(data=conflict, x='citizenship',
order=conflict['citizenship'].value_counts().index, palette='viridis')
plt.title('Citizenship Distribution of Victims')
plt.xlabel('Citizenship')
plt.ylabel('Frequency')
plt.xticks(rotation=45, ha='right')
plt.show()
```



Place of Residence Distribution

```
plt.figure(figsize=(14, 8))
sns.countplot(data=conflict, x='event_location_district',

order=conflict['event_location_district'].value_counts().index,
palette='plasma')
plt.title('Event Location District Distribution of Victims')
plt.xlabel('Event Location District')
plt.ylabel('Frequency')
plt.xticks(rotation=45, ha='right')
plt.show()
```



reason to use

```
# for gender distribution
A pie chart effectively illustrates the proportional distribution of
genders among victims, offering a clear comparison of the relative
frequencies of male and female victims in the dataset.

# for citizenship graph
A bar chart is suitable for visualizing citizenship distribution
because it provides a clear comparison of the frequency of victims for
each citizenship category, helping to identify the most prevalent
citizenships among the victims
```