```
import pandas as pd
```

```
df=pd.read_csv("/content/globalterrorismdb_0718dist.tar.bz2",compression="bz2")
```

```
<ipython-input-2-b7b18a7931ce>:1: DtypeWarning: Columns (5,7,32,34,62,63,64,77,80,91,93,95,97,115,116,122) have mixed types. Specify dty
  df=pd.read_csv("/content/globalterrorismdb_0718dist.tar.bz2",compression="bz2")
```

```
df.head()
```

	Unnamed: 0	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	co
0	0	197000000001	1970	7	2	NaN	0	NaN	58	
1	1	197000000002	1970	0	0	NaN	0	NaN	130	
2	2	197001000001	1970	1	0	NaN	0	NaN	160	
3	3	197001000002	1970	1	0	NaN	0	NaN	78	
4	4	197001000003	1970	1	0	NaN	0	NaN	101	

```
df.columns
```

```
Index(['Unnamed: 0', 'eventid', 'iyear', 'imonth', 'iday', 'approxdate',
       'extended', 'resolution', 'country', 'country_txt',
       ...
       'addnotes', 'scite1', 'scite2', 'scite3', 'dbsource', 'INT_LOG',
       'INT_IDEO', 'INT_MISC', 'INT_ANY', 'related'],
      dtype='object', length=136)
```

```
print(df.columns.tolist())
```

Saved successfully!

```
r', 'imonth', 'iday', 'approxdate', 'extended', 'resolution', 'country', 'country_txt', 'region', 'regior
```

```
pd.set_option('display.max_columns',None)
```

```
df
```

	Unnamed: 0	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_txt	pro
0	0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	Central America & Caribbean	
1	1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	North America	
2	2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	Southeast Asia	
3	3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	Western Europe	
4	4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	East Asia	F
...	...	...	...	...	...	...	...	...	...	...	...	...	
181686	181686	201712310022	2017	12	31	NaN	0	NaN	182	Somalia	11	Sub-Saharan Africa	S

```
selected_column=[
"success",
"suicide",
"attacktype1",
"attacktype1_txt",
"targtype1_txt",
"targsubtype1_txt",
"target1",
"natlty1_txt",
"gname",
"gsubname",
"nperps",
"weaptype1_txt",
"nkillus"
]
```

Saved successfully!

```
len(selected_column)
```

```
15
```

```
# df_selected=df.loc[:,selected_column]
```

q1: How has the number of terrorist activities changed over the years? Are there certain regions where this trend is different from the global averages?

```
df.head()
```

	Unnamed: 0	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_txt	provstate	
0	0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	Central America & Caribbean	NaN	D

```
df["country_txt"].value_counts()
```

```
Iraq                    24636
Pakistan                14368
Afghanistan             12731
India                   11960
Colombia                 8306
                        ...
International               1
Wallis and Futuna           1
South Vietnam               1
Andorra                     1
Antigua and Barbuda         1
Name: country_txt, Length: 205, dtype: int64
```

```
import seaborn as sns
```

```
df2=df.groupby(['iyear'],as_index=False ).count()[['eventid']]  # Here we learn about slice select
```

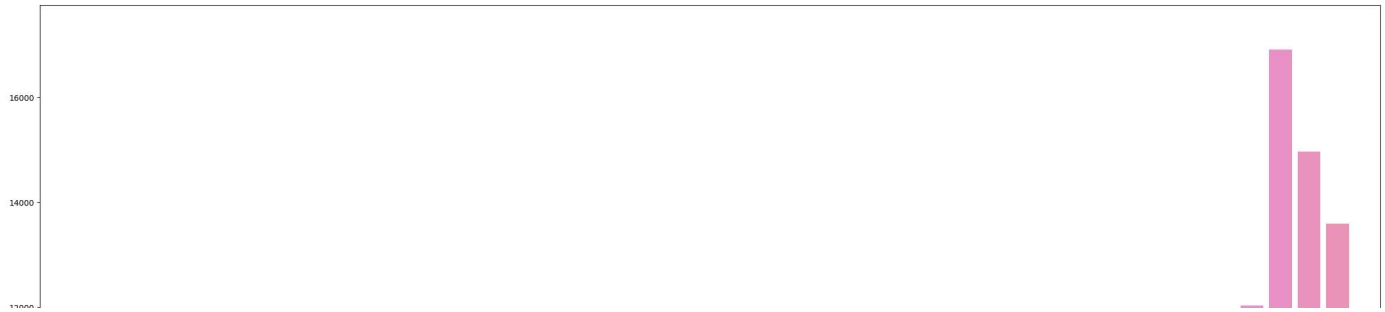
```
<ipython-input-14-d9f64c3b9201>:1: PerformanceWarning: DataFrame is highly fragmented.  This is usually the result of calling `frame.ins
  df2=df.groupby(['iyear'],as_index=False ).count()[['eventid']]  # Here we learn about slice select
```

```
year=df['iyear'].unique()
```

```
# print(year)
year_count=df["iyear"].value_counts(dropna = False).sort_index()
import matplotlib.pyplot as plt
%matplotlib inline
```

Saved successfully!

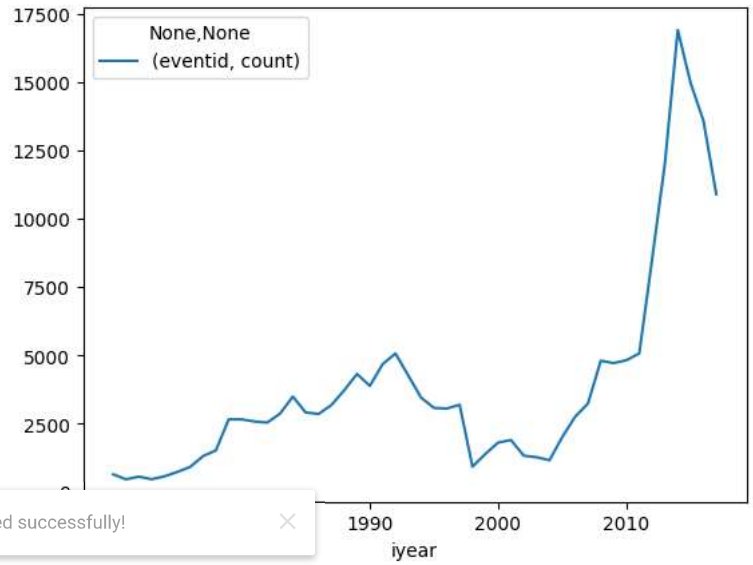
```
<Axes: ylabel='iyear'>
```




```
df4=df.groupby(['iyear'],as_index=True).agg({'eventid':['count']})
```

```
df4.plot()
```

```
<Axes: xlabel='iyear'>
```



Saved successfully!

```
df.head(2)
```

	Unnamed: 0	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_txt	provstate	
0	0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	Central America & Caribbean	NaN	Dc
1	1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	North America	Federal	I

```
df['region_txt'].unique()
```

```
array(['Central America & Caribbean', 'North America', 'Southeast Asia',
       'Western Europe', 'East Asia', 'South America', 'Eastern Europe',
       'Sub-Saharan Africa', 'Middle East & North Africa',
       'Australasia & Oceania', 'South Asia', 'Central Asia'],
      dtype=object)
```

```
df5=df[['region_txt','iyear','eventid','nkill']].groupby(['region_txt','iyear']).agg({'eventid':'count','nkill':'sum'})
```

```
pd.set_option('display.max_columns',120)
```

```
df5.index.levels[0].unique()
```

```
Index(['Australasia & Oceania', 'Central America & Caribbean', 'Central Asia',
       'East Asia', 'Eastern Europe', 'Middle East & North Africa',
       'North America', 'South America', 'South Asia', 'Southeast Asia',
       'Sub-Saharan Africa', 'Western Europe'],
      dtype='object', name='region_txt')
```

```
df5
```

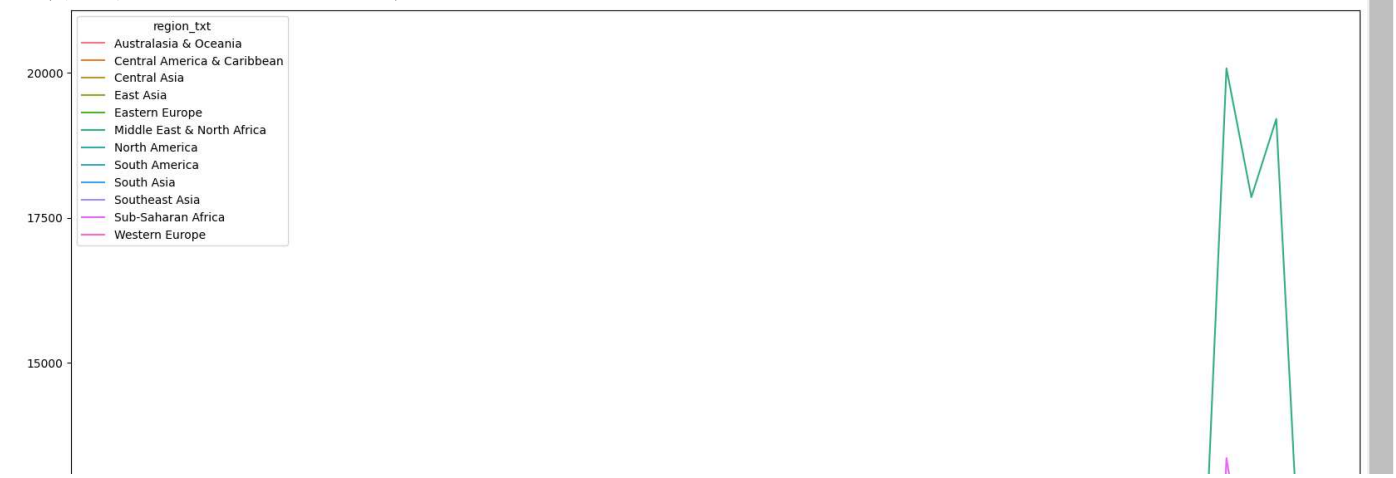
region_txt	iyear	eventid	nkill
Australasia & Oceania	1970	1	0.0
	1971	1	0.0
	1972	8	1.0
	1973	1	0.0
	1974	1	0.0
...	...	...	...
Western Europe	2013	261	7.0
	2014	215	5.0
	2015	333	171.0
	2016	273	238.0
	2017	291	83.0

529 rows × 2 columns

Saved successfully!

```
,y='nkill',hue='region_txt')
plt.xlabel("Year")
plt.ylabel("No of terrorist attacks")
```

```
Text(0, 0.5, 'No of terrorist attacks')
```




```
df5.unstack(level=0)
```

Saved successfully!

Is the number of incidents and the number of casualties correlated? Can you spot any irregularities or outliers?

```
df.drop_duplicates()
```

	Unnamed: 0	eventid	iyear	imonth	iday	approxdate	extended	resolution
0	0	197000000001	1970	7	2	NaN	0	NaN
1	1	197000000002	1970	0	0	NaN	0	NaN
2	2	197001000001	1970	1	0	NaN	0	NaN
3	3	197001000002	1970	1	0	NaN	0	NaN
4	4	197001000003	1970	1	0	NaN	0	NaN
...	...	...	...	...	...	...	...	...
181686	181686	201712310022	2017	12	31	NaN	0	NaN
181687	181687	201712310029	2017	12	31	NaN	0	NaN
181688	181688	201712310030	2017	12	31	NaN	0	NaN
181689	181689	201712310031	2017	12	31	NaN	0	NaN
181690	181690	201712310032	2017	12	31	NaN	0	NaN

181691 rows × 136 columns

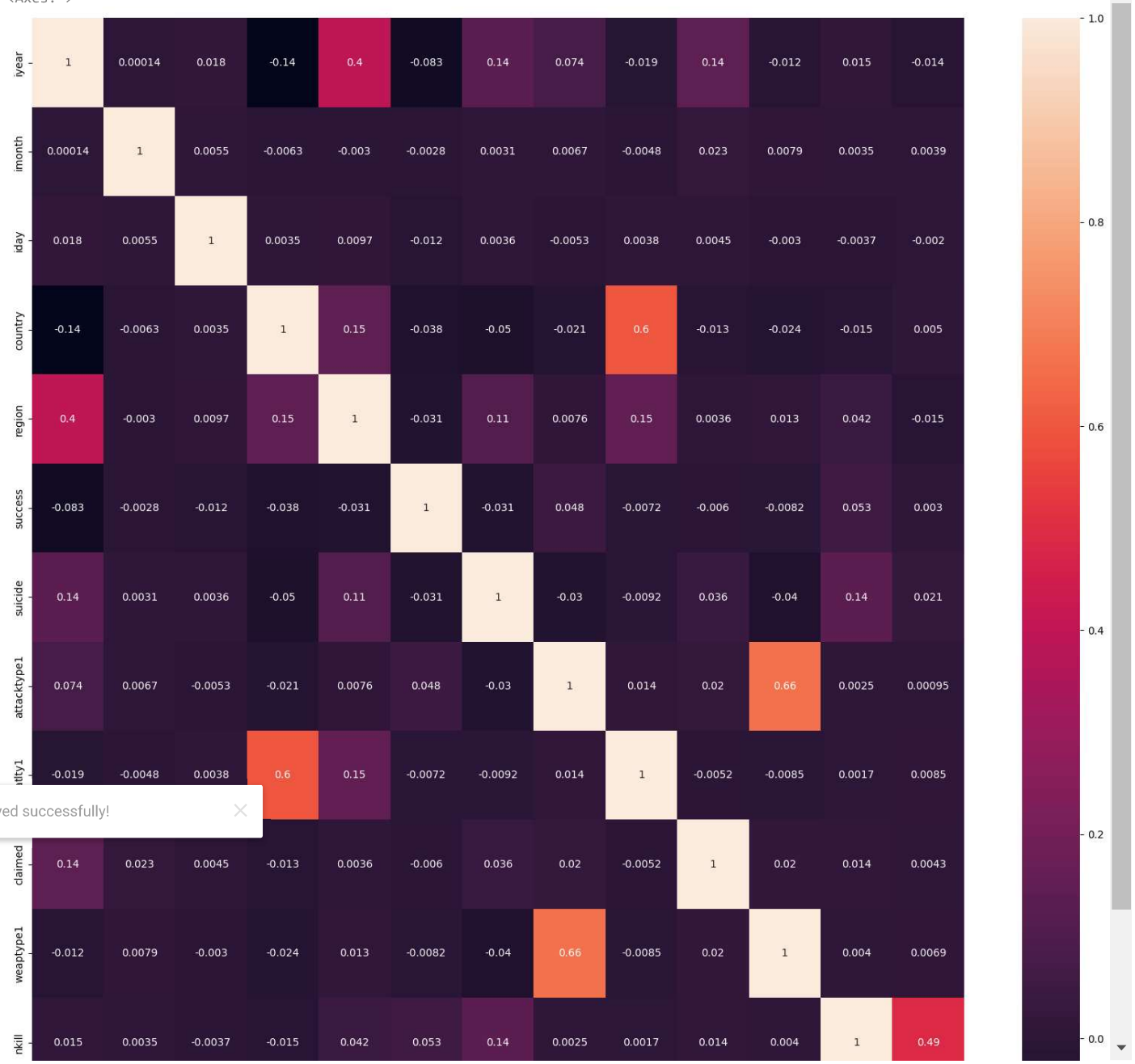
Saved successfully!

```
columns_of_interest = df[['iyear', 'imonth', 'iday', 'country', 'region', 'success', 'suicide', 'attacktype1', 'natlty1', 'c
df_corr = df[['iyear', 'imonth', 'iday', 'country', 'region', 'success', 'suicide', 'attacktype1', 'natlty1', 'claimed', 'we

plt.figure(figsize=(20,20))
correlation_matrix = columns_of_interest.corr()
# print(correlation_matrix)

# Create a heatmap
sns.heatmap(correlation_matrix,annot=True)
```

<Axes: >



Saved successfully!

NOW WE ARE GOINF TO FIND OUTLIER IN THE "NKILL" COLUMN

```
df.describe()
```

	Unnamed: 0	eventid	iyear	imonth	iday	extended	country	region	latitude
count	181691.000000	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	177135.000000
mean	90845.000000	2.002705e+11	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	23.498343
std	52449.818217	1.325957e+09	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	18.569242
min	0.000000	1.970000e+11	1970.000000	0.000000	0.000000	0.000000	4.000000	1.000000	-53.154613
25%	45422.500000	1.991021e+11	1991.000000	4.000000	8.000000	0.000000	78.000000	5.000000	11.510046
50%	90845.000000	2.009022e+11	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000	31.467463
75%	136267.500000	2.014081e+11	2014.000000	9.000000	23.000000	0.000000	160.000000	10.000000	34.685087
max	181690.000000	2.017123e+11	2017.000000	12.000000	31.000000	1.000000	1004.000000	12.000000	74.633553

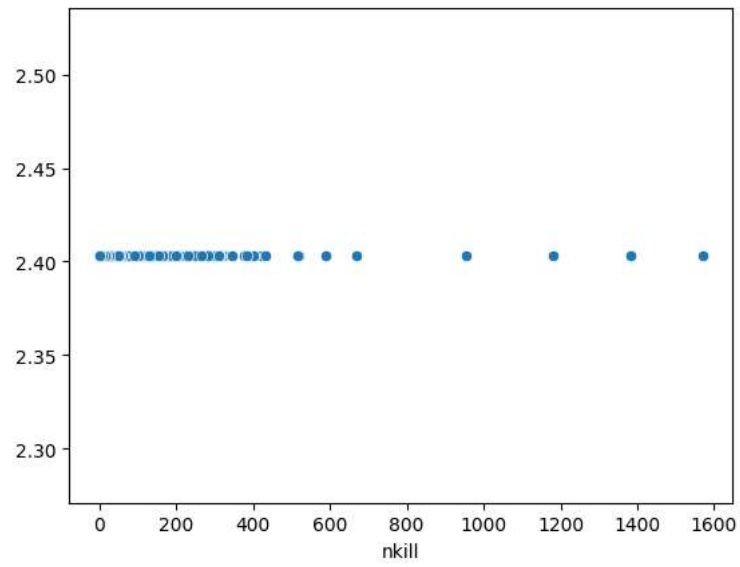
```
df['nkill'].describe()
```

```
count    171378.000000
mean          2.403272
std          11.545741
min           0.000000
25%           0.000000
50%           0.000000
75%           2.000000
max        1570.000000
```

Saved successfully!

```
sns.scatterplot(data=df,x=df['nkill'],y=df['nkill'].mean())
```

```
<Axes: xlabel='nkill'>
```




```
def find_outliers_IQR(df):

   q1=df.quantile(0.25)

   q3=df.quantile(0.75)
```

```
    IQR=q3-q1

    outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]

    return outliers

outlier=find_outliers_IQR(df['nkill'])



def drop_outlier(df):
  q1=df.quantile(0.25)
  q3=df.quantile(0.75)
  iqr=q3-q1
  non_outlier=df[~((df<(q1-1.5*iqr)) | (df>(q3+1.5*iqr)))]

  return non_outlier

df['nkill']=drop_outlier(df['nkill'])


df['nkill'].dropna()
```

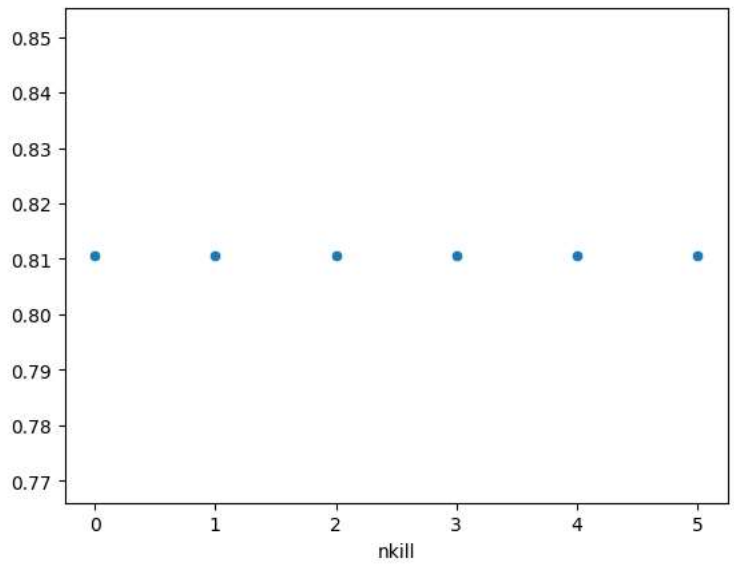
```
0         1.0
1         0.0
2         1.0
5         0.0
6         0.0
         ...
181686    1.0
181687    2.0
181688    0.0
181689    0.0
181690    0.0
Name: nkill, Length: 155136, dtype: float64
```

Saved successfully!

```
5.0
```

```
sns.scatterplot(data=df,x=df['nkill'],y=df['nkill'].mean())
```

```
<Axes: xlabel='nkill'>
```




```
df['nkill'].describe()
```

```
count    155136.000000
mean          0.810598
std           1.215948
```

```
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max           5.000000
Name: nkill, dtype: float64
```

What are the most common methods of attacks? Does it differ in various regions or in time?

```
df[['attacktype1','attacktype1_txt' ]]
df['attacktype1_txt'].value_counts().head(3)
```

```
Bombing/Explosion    88255
Armed Assault        42669
Assassination        19312
Name: attacktype1_txt, dtype: int64
```

```
q3=df.groupby(['attacktype1_txt','region_txt'],as_index=True).agg({'eventid':'count'})
q3
```

attacktype1_txt	region_txt	eventid
Armed Assault	Australasia & Oceania	51
	Central America & Caribbean	4361
	Central Asia	116
	East Asia	117
	Eastern Europe	1274
...	...	...
	merica	754
	South Asia	2021
	Southeast Asia	433
	Sub-Saharan Africa	1355
	Western Europe	265

108 rows × 1 columns

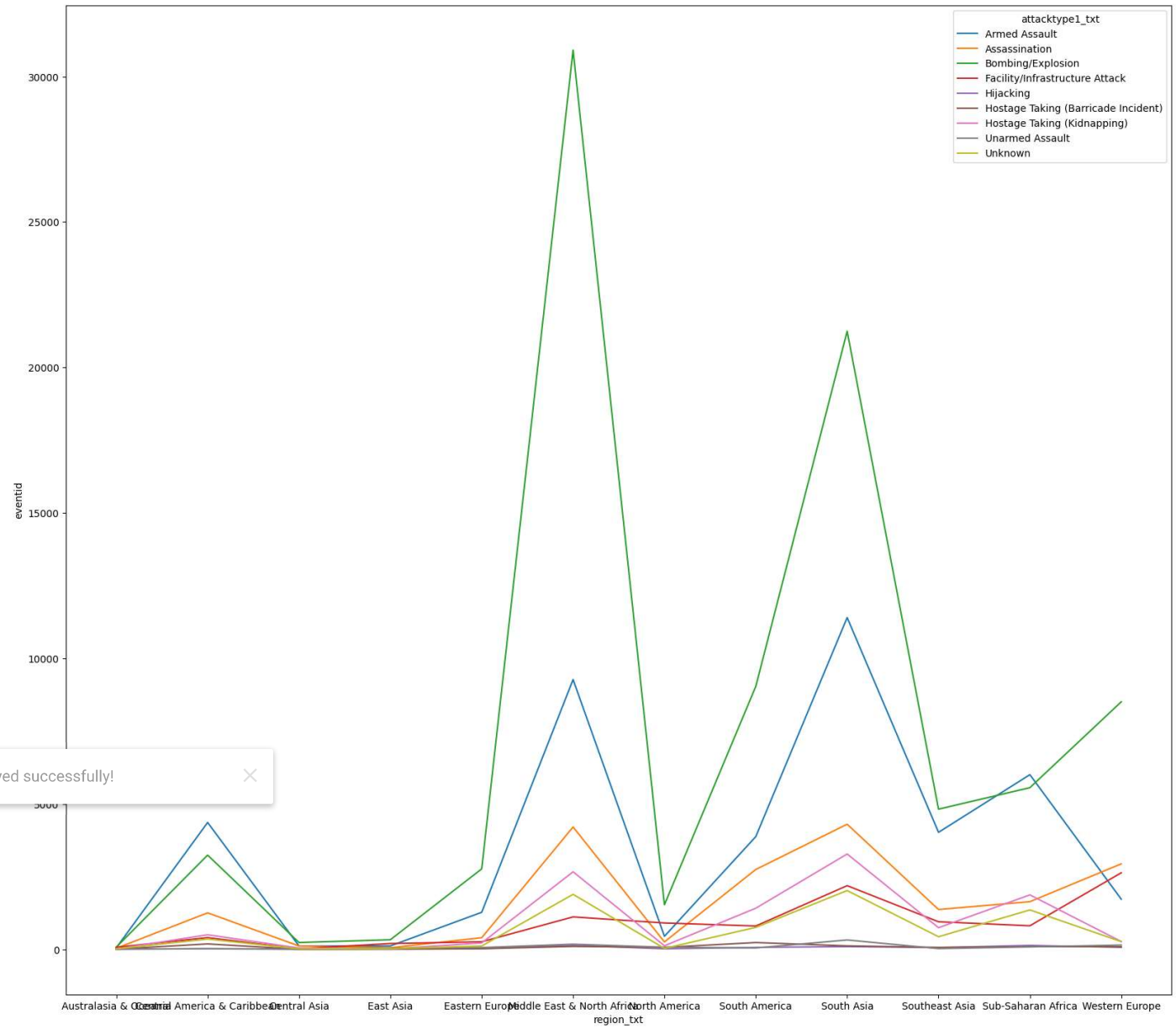
Saved successfully!

```
q3.unstack(level=-1)
```

	eventid											
region_txt	Australasia & Oceania	Central America & Caribbean	Central Asia	East Asia	Eastern Europe	Middle East & North Africa	North America	South America	South Asia	Southeast Asia	Sub-Saharan Africa	Western Europe
attacktype1_txt												
Armed Assault	51	4361	116	117	1274	9273	448	3875	11404	4022	6004	1724
Assassination	36	1254	115	55	400	4206	255	2745	4301	1369	1638	2938
Bombing/Explosion	75	3239	235	330	2766	30908	1534	9039	21246	4818	5557	8508
Facility/Infrastructure Attack	71	403	20	200	260	1115	906	803	2189	948	810	2631
Hijacking	3	26	8	18	26	138	18	67	93	59	136	67
Hostage Taking (Barricade Incident)	6	187	2	3	21	100	67	234	120	67	95	89
Hostage Taking (Kidnapping)	13	501	45	14	220	2666	123	1414	3277	744	1872	269

```
plt.figure(figsize=(20,18))
sns.lineplot(data=q3,x='region_txt',y='eventid',hue='attacktype1_txt')
```

```
<Axes: xlabel='region_txt', ylabel='eventid'>
```



Saved successfully!

```
q31=df.groupby(["iyear","attacktype1_txt"]).agg({'eventid':'count'})
```

```
q31
```

		eventid
iyear	**attacktype1_txt**	
1970	**Armed Assault**	61
	Assassination	22
	Bombing/Explosion	333
	Facility/Infrastructure Attack	174
	Hijacking	11
...	**...**	...
2017	**Hijacking**	58
	Hostage Taking (Barricade Incident)	82
	Hostage Taking (Kidnapping)	878
	Unarmed Assault	101
	Unknown	825

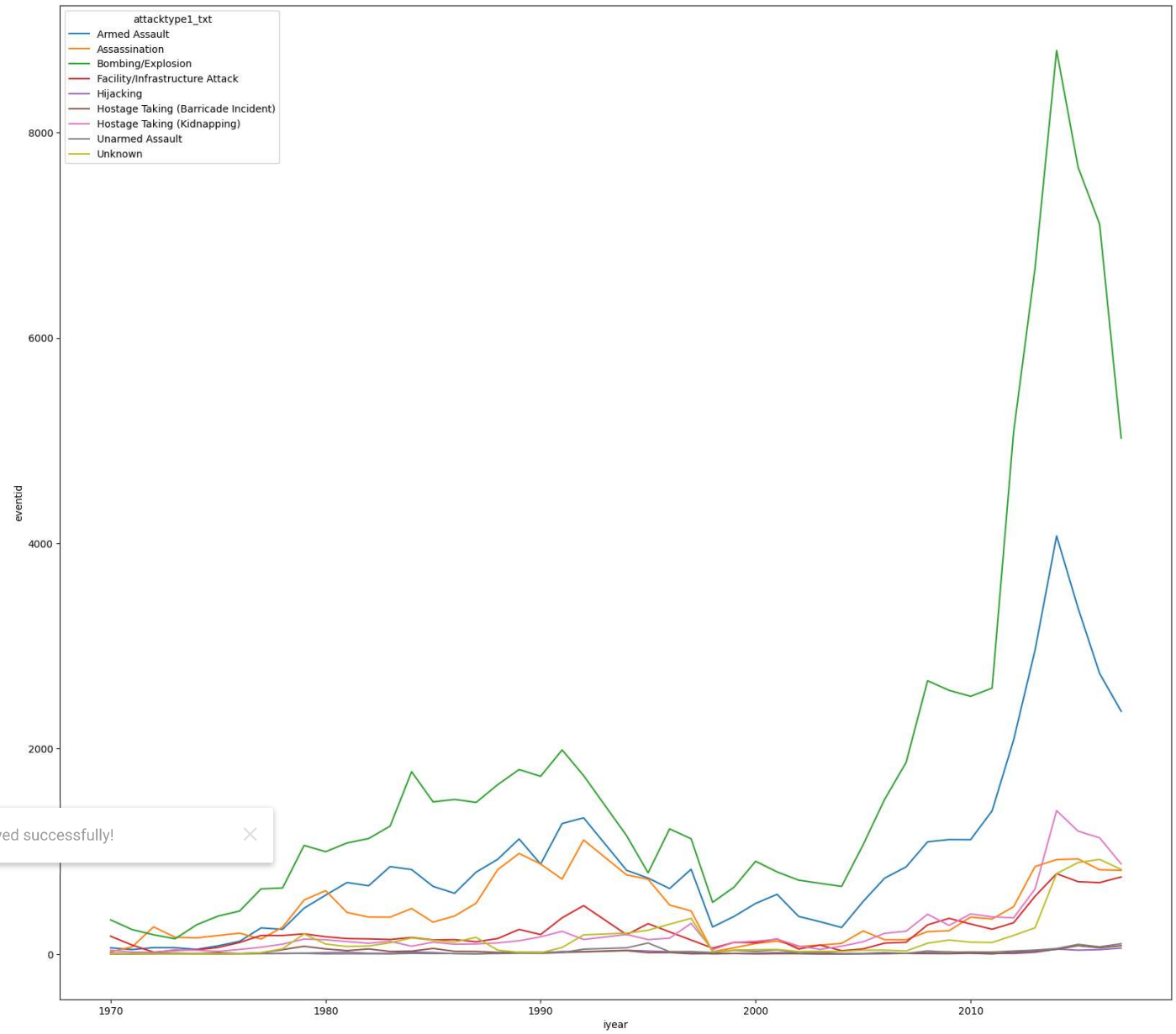
416 rows × 1 columns

```
q31.unstack(level=0)
```

	eventid																
iyear	**1970**	**1971**	**1972**	**1973**	**1974**	**1975**	**1976**	**1977**	**1978**	**1979**	**1980**	**1981**	**1982**	**1983**	**1984**	**1985**	**1986**
attacktype1_txt																	
Armed Assault	61.0	44.0	63.0	62.0	46.0	81.0	124.0	255.0	241.0	447.0	574.0	697.0	665.0	852.0	823.0	659.0	592.0
[illegible]	[illegible]	[illegible].0	265.0	164.0	158.0	181.0	204.0	146.0	263.0	526.0	618.0	405.0	361.0	360.0	443.0	311.0	371.0
Bombing/Explosion	333.0	239.0	188.0	149.0	285.0	370.0	419.0	635.0	644.0	1058.0	997.0	1082.0	1125.0	1246.0	1776.0	1482.0	1506.0
Facility/Infrastructure Attack	174.0	88.0	19.0	36.0	42.0	64.0	113.0	182.0	181.0	197.0	169.0	151.0	148.0	142.0	162.0	138.0	141.0
Hijacking	11.0	6.0	12.0	8.0	3.0	1.0	4.0	7.0	NaN	9.0	14.0	18.0	8.0	8.0	21.0	15.0	4.0
Hostage Taking (Barricade Incident)	3.0	1.0	4.0	7.0	5.0	13.0	6.0	13.0	43.0	76.0	50.0	34.0	50.0	26.0	29.0	55.0	27.0
Hostage Taking (Kidnapping)	38.0	20.0	16.0	43.0	37.0	27.0	45.0	67.0	97.0	146.0	140.0	122.0	105.0	124.0	77.0	116.0	96.0
Unarmed Assault	3.0	NaN	NaN	3.0	4.0	NaN	3.0	NaN	5.0	7.0	2.0	3.0	3.0	2.0	7.0	5.0	6.0
Unknown	6.0	3.0	1.0	1.0	1.0	3.0	5.0	14.0	52.0	196.0	98.0	74.0	79.0	110.0	157.0	134.0	117.0


```
plt.figure(figsize=(20,18))
sns.lineplot(data=q31,x='iyear',y='eventid',hue='attacktype1_txt')
```

```
<Axes: xlabel='iyear', ylabel='eventid'>
```



Saved successfully!

Plot the locations of attacks on a map to visualize their regional spread

```
locations_df = df[['iyear', 'country_txt', 'region_txt', 'latitude', 'longitude', 'weaptype1_txt']]
locations_df.head(10)
```