The Spark Foundation

Data Science & Business Analytics Internship jan-2022

Author :- Shamu Vishwakarma

Task 4- Exploratory Data Analysis of a Global Terrorism

Importing all necessary libraries

In [3]:
```
!pip install folium
# For visualizing geospatial data
```

```
Requirement already satisfied: folium in c:\users\windows 10\anaconda3\lib\site-packages
(0.12.1.post1)
Requirement already satisfied: requests in c:\users\windows 10\anaconda3\lib\site-packag
es (from folium) (2.25.1)
Requirement already satisfied: numpy in c:\users\windows 10\anaconda3\lib\site-packages
(from folium) (1.20.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\windows 10\anaconda3\lib\site-pac
kages (from folium) (2.11.3)
Requirement already satisfied: branca>=0.3.0 in c:\users\windows 10\anaconda3\lib\site-p
ackages (from folium) (0.4.2)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\windows 10\anaconda3\lib\sit
e-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\windows 10\anaconda3\lib\si
te-packages (from requests->folium) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\windows 10\anaconda3\li
b\site-packages (from requests->folium) (1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\windows 10\anaconda3\lib\s
ite-packages (from requests->folium) (2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in c:\users\windows 10\anaconda3\lib\site-pa
ckages (from requests->folium) (2.10)
```

In [21]:
```
import pandas as pd
import matplotlib.patches as mpatches
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('fivethirtyeight')
import folium
import folium.plugins
from matplotlib import animation,rc
import io
import base64
from IPython.display import HTML, display
import warnings
warnings.filterwarnings('ignore')
import codecs
```

Reading csv file

In [5]:
```
terror = pd.read_csv("Desktop/globalterrorismdb_0718dist.csv", encoding = "ISO-8859-1")
```

In [6]:
```
terror.head(5)
```

Out[6]:

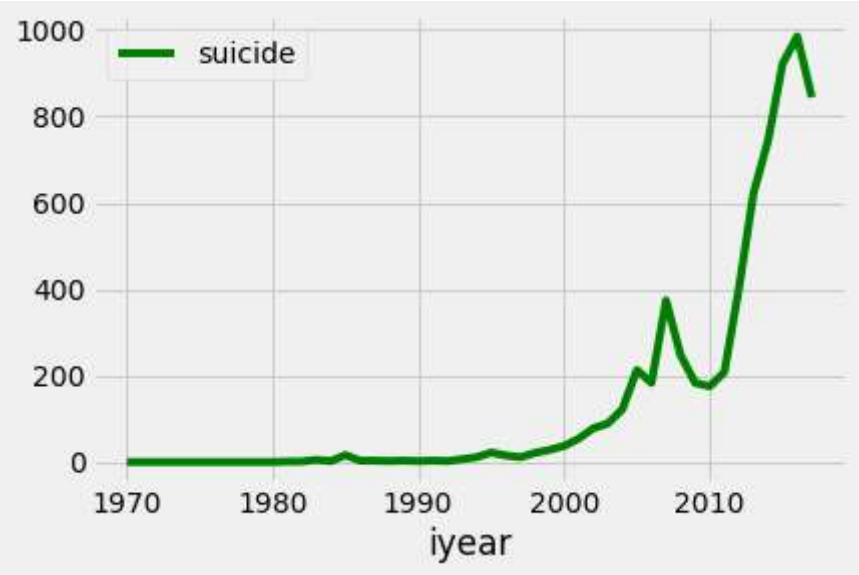
	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4

5 rows × 135 columns

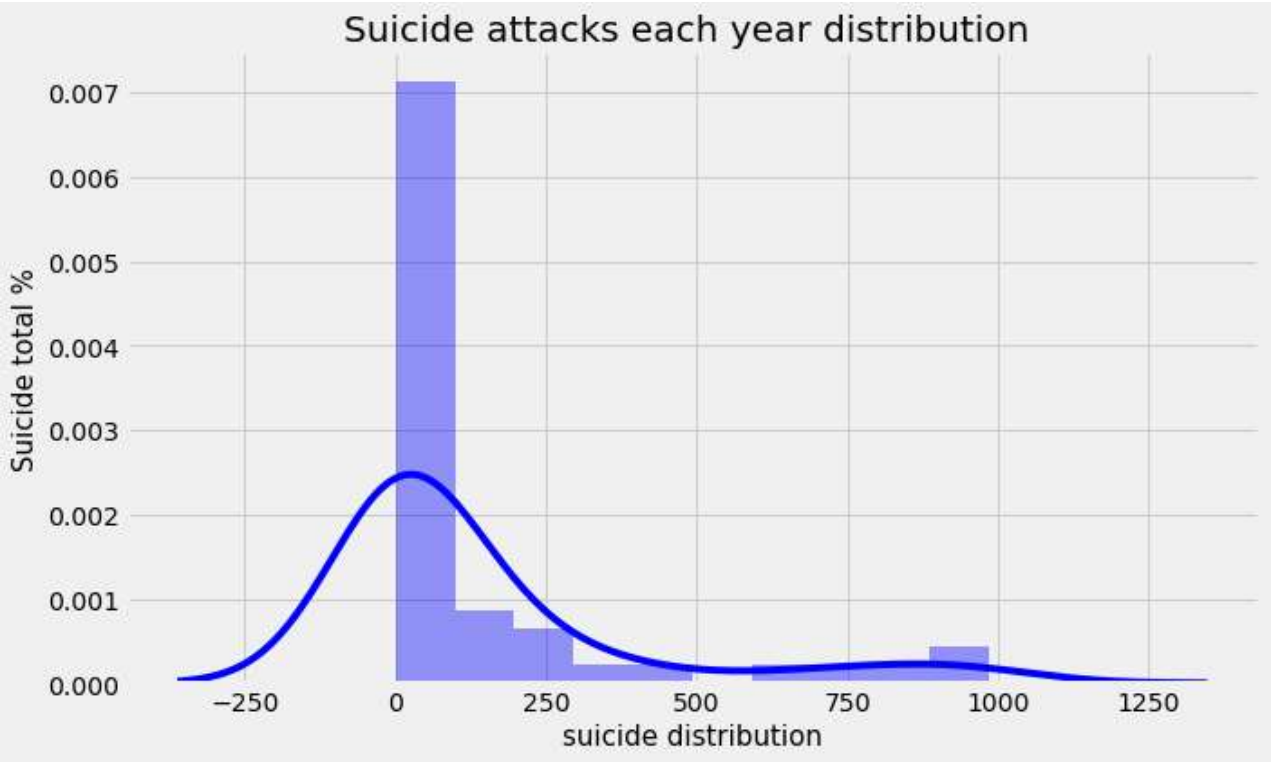
In [10]:
```
# Which Region had the most terrorism attacks??
print('The region of',terror['region_txt'].value_counts(dropna=True, normalize=False, a
print(' ')
# Name of the city, village, or town in which the incident occurred
print(terror['city'].value_counts(dropna=True, normalize=False, ascending=False).index[
print(' ')
print('The most known city that had terror attacks was' ,terror['city'].value_counts(dr
print(' ')
#Most notably used weapon
print('The most used weapon in terror attacks was' ,terror['weaptype1_txt'].value_count
print(' ')
#most known country
print('The most known country with terror attacks was' ,terror['country_txt'].value_cou
print(' ')
#1 = "Yes" The incident was a suicide attack. 0 = "No" There is no indication that the
s = terror['suicide'].value_counts(normalize=True).mul(100).round(1).astype(str) + '%'
print('Out of',terror['suicide'].value_counts(dropna=True, normalize=False, ascending=F
print(' ')
print('The most preferred method of attack was',terror['attacktype1_txt'].value_counts(
print(' ')
t = terror['targtype1_txt'].value_counts(normalize=True).mul(100).round(1).astype(str)
print('The main targets of terrorists were',terror['targtype1_txt'].value_counts(dropna
```

```
The region of Middle East & North Africa had the highest amount of Terrorist Attacks tot
alled at 50474

Unknown and Unnamed Cities consisting of Terrorist Attacks totalled at 9775

The most known city that had terror attacks was Baghdad

The most used weapon in terror attacks was Explosives totalled at 92426

The most known country with terror attacks was Iraq totalled at 24636

Out of 181691 total attacks 3.7% were suicide attacks

The most preferred method of attack was Bombing/Explosion totalling at 88255

The main targets of terrorists were Private Citizens & Property totalling at 23.9%
```

In [44]:
```
#suicides per year
#suicide attacks, both success's and fails
suicides_by_year = terror[["iyear","suicide"]].groupby("iyear").aggregate(np.sum)
suicides_by_year.plot(color = 'g');
```



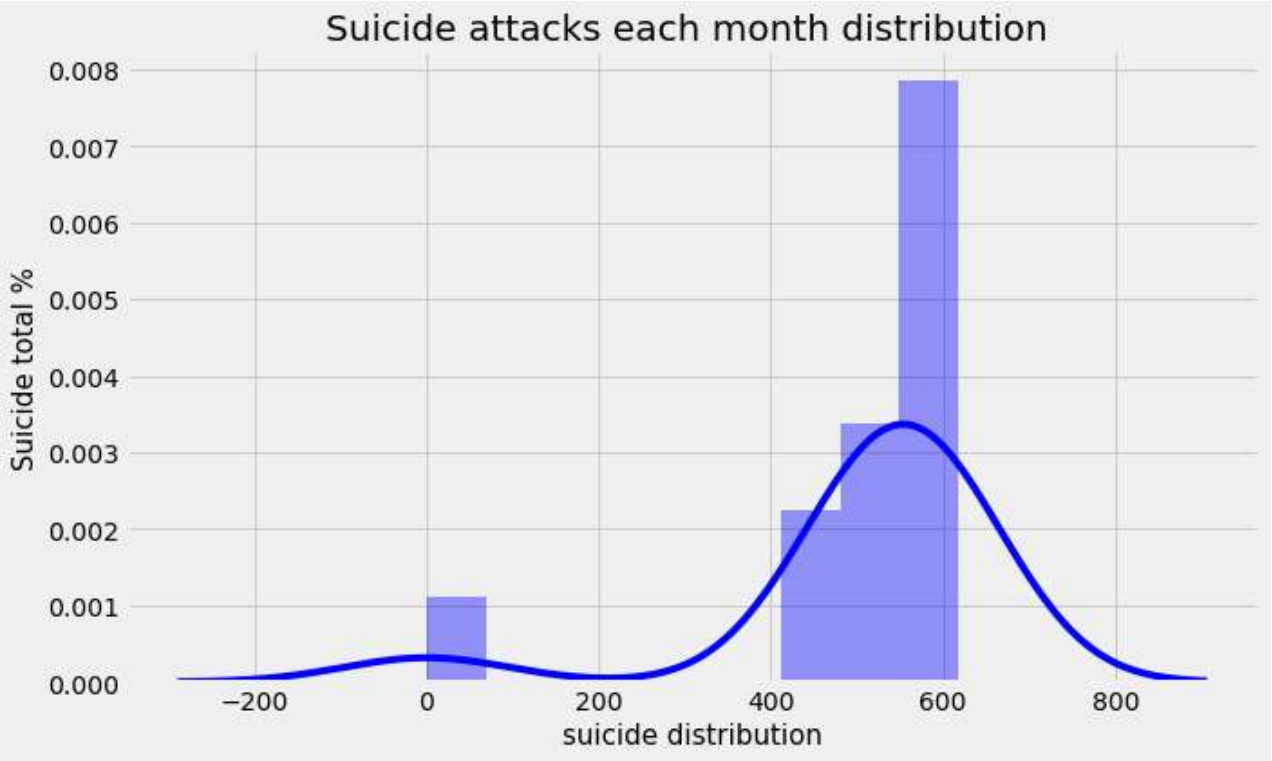
In [51]:
```
#suicide attacks each year distribution
plt.figure(figsize=(10,6))
fig = sns.distplot(terror[["iyear","suicide"]].groupby("iyear").aggregate(np.sum), colo
fig.set_xlabel("suicide distribution",size=15)
fig.set_ylabel("Suicide total %",size=15)
plt.title('Suicide attacks each year distribution ',size = 20)
plt.show()
```



In [13]:
```
#suicide attacks, both success's and fails
suicides_by_year = terror[["iyear","suicide"]].groupby("iyear").aggregate(np.sum)
```

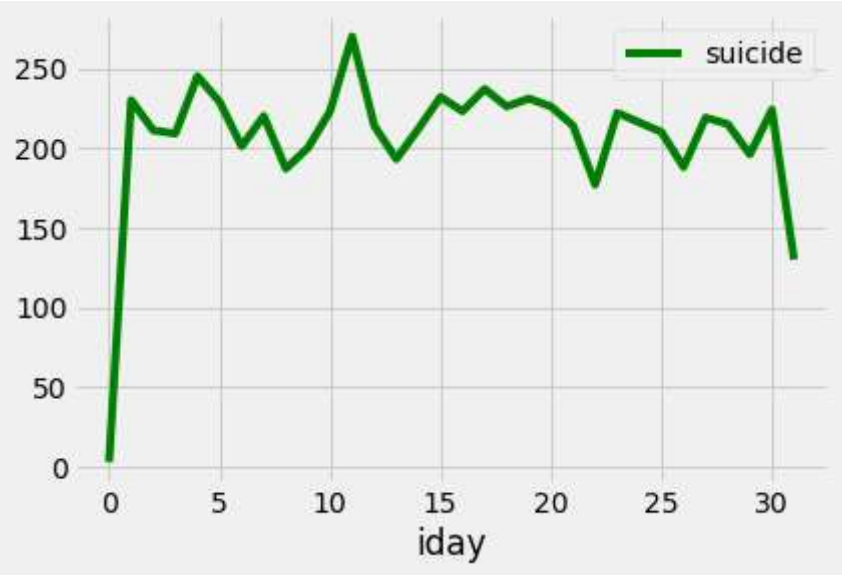
In [52]:
```
#suicides per month
#suicide attacks, both success's and fails
suicides_by_month = terror[["imonth","suicide"]].groupby("imonth").aggregate(np.sum)
suicides_by_month.plot(color = 'r');
```

In [53]:
```
#suicide attacks distribution each month
plt.figure(figsize=(10,6))
fig = sns.distplot(terror[["imonth","suicide"]].groupby("imonth").aggregate(np.sum), co
fig.set_xlabel("suicide distribution",size=15)
fig.set_ylabel("Suicide total %",size=15)
plt.title('Suicide attacks each month distribution ',size = 20)
plt.show()
```



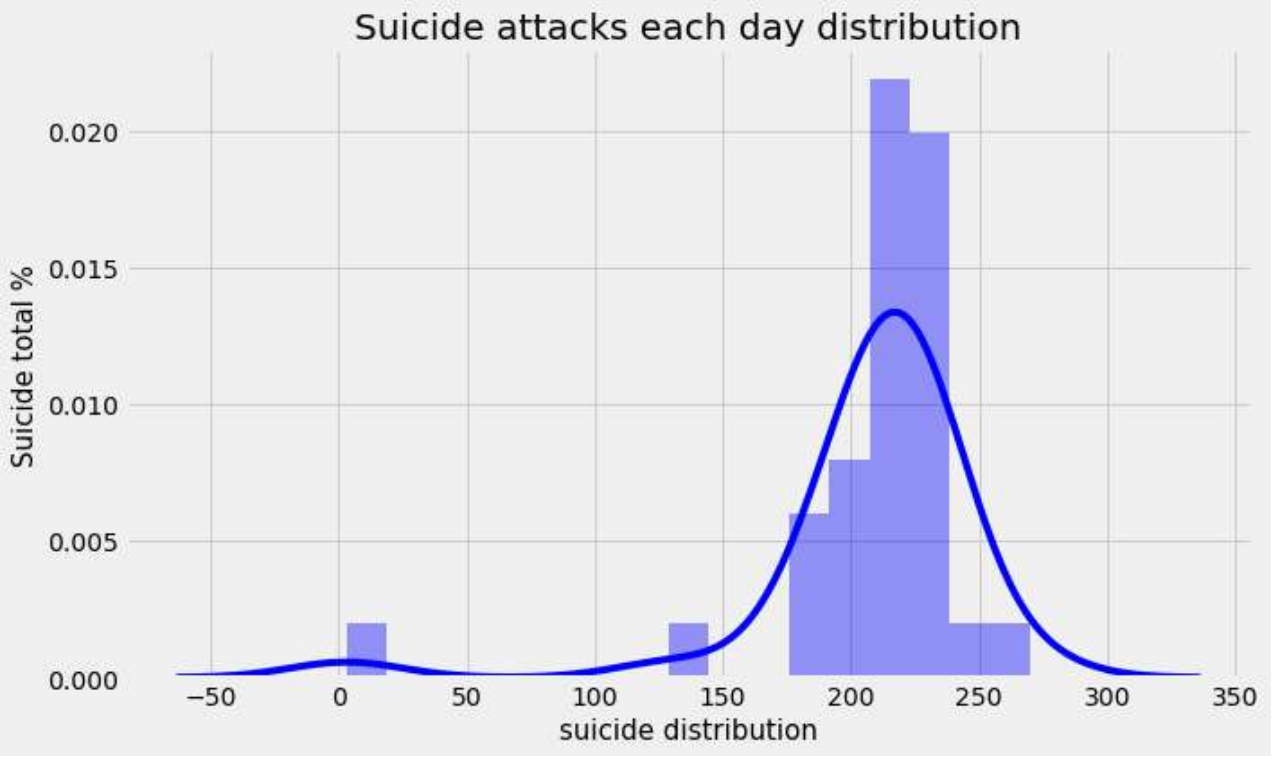
In [54]:

```
#suicides per day
#suicide attacks, both success's and fails
suicides_by_month = terror[["iday","suicide"]].groupby("iday").aggregate(np.sum)
suicides_by_month.plot(color = 'g');
```



In [55]:

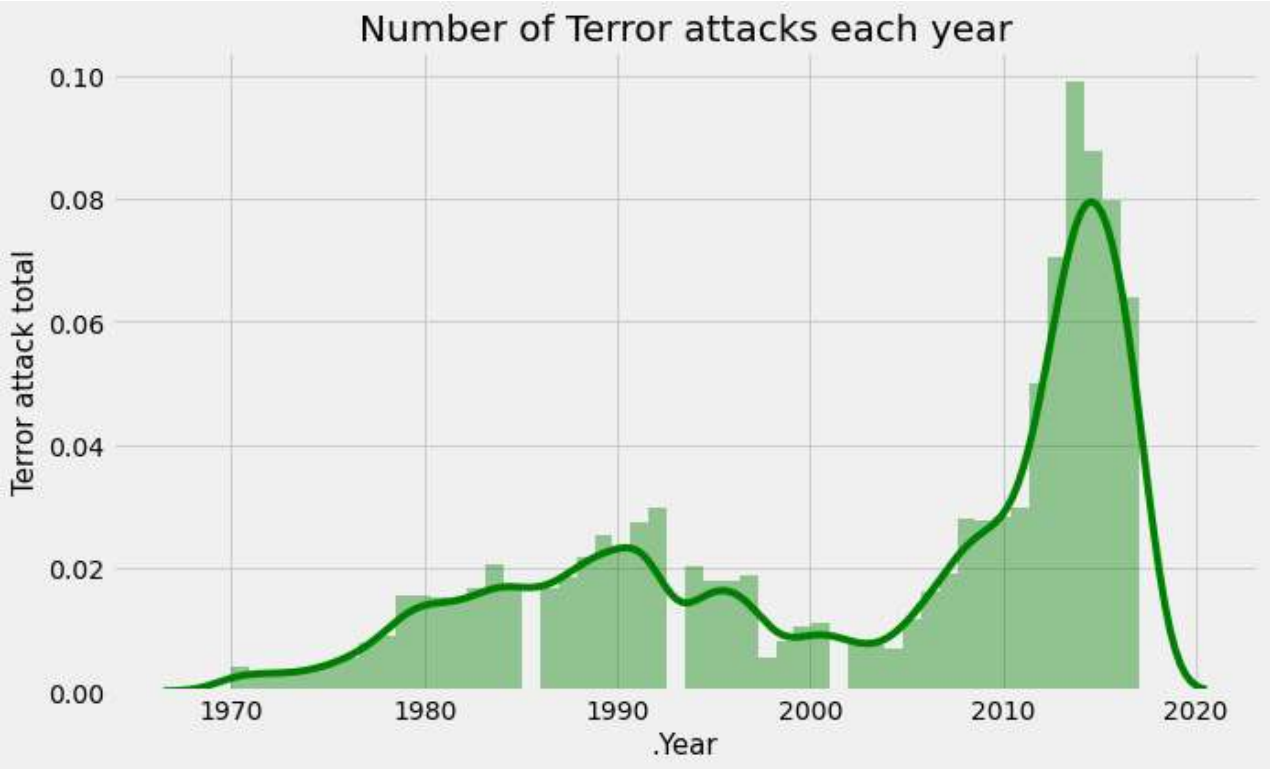
```
#suicides each day distribution
plt.figure(figsize=(10,6))
fig = sns.distplot(terror[["iday","suicide"]].groupby("iday").aggregate(np.sum), color
fig.set_xlabel("suicide distribution",size=15)
fig.set_ylabel("Suicide total %",size=15)
plt.title('Suicide attacks each day distribution',size = 20)
plt.show()
```



In [58]:

```
#Overall did the number of terror attacks increase?
#no. of terror attacks
```

```
plt.figure(figsize=(10,6))
fig = sns.distplot(terror["iyear"].values, color = 'g')
fig.set_xlabel(".Year",size=15)
fig.set_ylabel("Terror attack total",size=15)
plt.title('Number of Terror attacks each year',size = 20)
plt.show()
```



In [33]:
```
!Pip install plotly
```

```
Collecting plotly
  Downloading plotly-5.5.0-py2.py3-none-any.whl (26.5 MB)
Collecting tenacity>=6.2.0
  Downloading tenacity-8.0.1-py3-none-any.whl (24 kB)
Requirement already satisfied: six in c:\users\windows 10\anaconda3\lib\site-packages (f
rom plotly) (1.15.0)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.5.0 tenacity-8.0.1
```

In [34]:
```
#Can we visually see what parts of the world are most targetted? using plotly
import plotly.express as px

#to avoid lagging, only show the first 1k instead of the 180,000 data points.
geog = terror.head(1000)
geog = geog[['latitude','longitude']]
fig = px.scatter_geo(geog, lat='latitude', lon='longitude')

fig.show()
```

In [35]:
```
#Can we visually narrow this down further?
geog1 = terror.head(1000)
```

In [60]:
```
fig = px.scatter_mapbox(geog1, lat='latitude', lon='longitude', hover_name="city", hove
                        color_discrete_sequence=["fuchsia"], zoom=1, height=500)
fig.update_layout(
    mapbox_style="white-bg",
    mapbox_layers=[
        {
            "below": 'traces',
            "sourcetype": "raster",
            "sourceattribution": "United States Geological Survey",
            "source": [
                "https://basemap.nationalmap.gov/arcgis/rest/services/USGSImageryOnly/M
            ]
        }
      ])
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

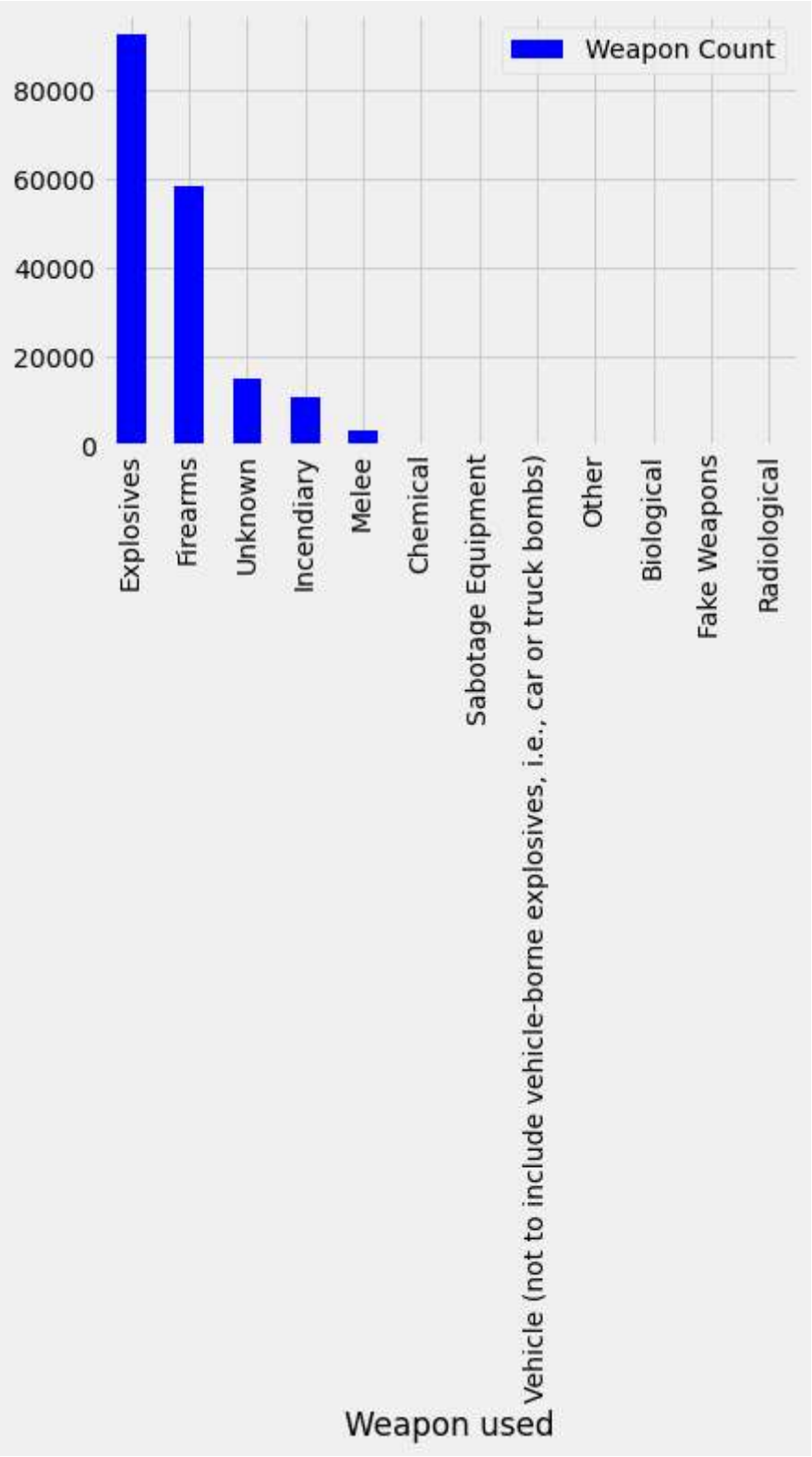
In [37]:
```
#Main Weapons:Explosives
from pandas import DataFrame


weapons = terror['weaptype1_txt'].value_counts()
weapons = DataFrame(weapons)

weapons.reset_index(level=0, inplace=True)
weapons.columns = ['Weapon used', 'Weapon Count']
weapons
```

Out[37]:

	Weapon used	Weapon Count
0	Explosives	92426
1	Firearms	58524
2	Unknown	15157
3	Incendiary	11135
4	Melee	3655
5	Chemical	321
6	Sabotage Equipment	141
7	Vehicle (not to include vehicle-borne explosiv...	136
8	Other	114
9	Biological	35
10	Fake Weapons	33
11	Radiological	14

In [61]:
```
weapons.plot.bar(x='Weapon used', y='Weapon Count', rot=90, color = 'b')
```

Out[61]: <AxesSubplot:xlabel='Weapon used'>



In [39]:

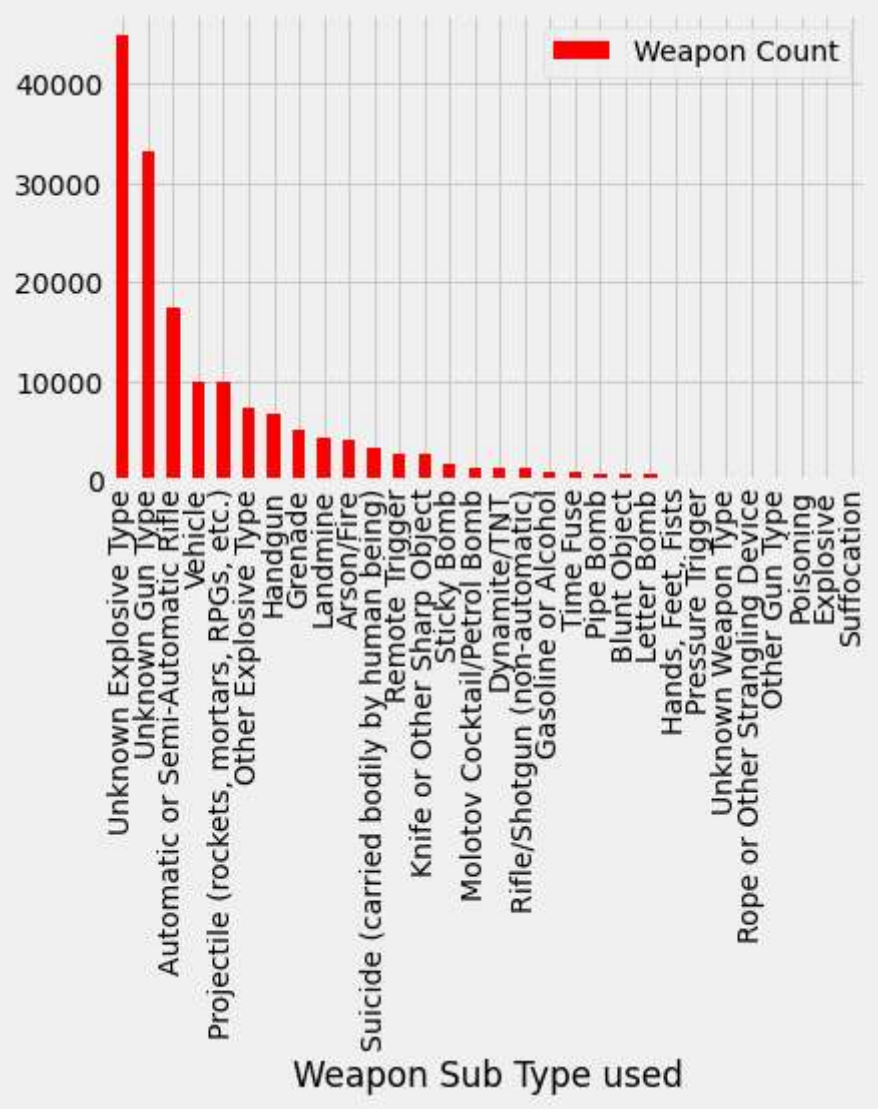
```
#Sub Weapon: unknown explosive type. What most terrorists most likely combine their fir
subweapons = terror['weapsubtype1_txt'].value_counts()
subweapons = DataFrame(subweapons)

subweapons.reset_index(level=0, inplace=True)
subweapons.columns = ['Weapon Sub Type used', 'Weapon Count']
subweapons
```

Out[39]:

	Weapon Sub Type used	Weapon Count
0	Unknown Explosive Type	44980
1	Unknown Gun Type	33137
2	Automatic or Semi-Automatic Rifle	17412
3	Vehicle	9900
4	Projectile (rockets, mortars, RPGs, etc.)	9848
5	Other Explosive Type	7304
6	Handgun	6704
7	Grenade	5167
8	Landmine	4251
9	Arson/Fire	4141
10	Suicide (carried bodily by human being)	3245
11	Remote Trigger	2719
12	Knife or Other Sharp Object	2585
13	Sticky Bomb	1594
14	Molotov Cocktail/Petrol Bomb	1239
15	Dynamite/TNT	1222
16	Rifle/Shotgun (non-automatic)	1175
17	Gasoline or Alcohol	844
18	Time Fuse	792
19	Pipe Bomb	625
20	Blunt Object	587
21	Letter Bomb	548
22	Hands, Feet, Fists	231
23	Pressure Trigger	219
24	Unknown Weapon Type	107
25	Rope or Other Strangling Device	103
26	Other Gun Type	86
27	Poisoning	83
28	Explosive	65
29	Suffocation	10

In [40]:

```
subweapons = subweapons.plot.bar(x='Weapon Sub Type used', y='Weapon Count', rot=90, co
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: