

The Spark Foundation

Data Science & Business Analytics Internship jan-2022

Author :- Bikash Barik

Task 4- Exploratory Data Analysis of a Global Terrorism

Importing all necessary libraries

```
In [3]: 1 !pip install folium  
2 # 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-packages (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-packages (from folium) (2.11.3)  
Requirement already satisfied: branca>=0.3.0 in c:\users\windows 10\anaconda3\lib\site-packages (from folium) (0.4.2)  
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\windows 10\anaconda3\lib\site-packages (from Jinja2>=2.9->folium) (1.1.1)  
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\windows 10\anaconda3\lib\site-packages (from requests->folium) (4.0.0)  
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\windows 10\anaconda3\lib\site-packages (from requests->folium) (1.26.4)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\windows 10\anaconda3\lib\site-packages (from requests->folium) (2020.12.5)  
Requirement already satisfied: idna<3,>=2.5 in c:\users\windows 10\anaconda3\lib\site-packages (from requests->folium) (2.10)
```

```
In [21]: 1 import pandas as pd
2 import matplotlib.patches as mpatches
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 import numpy as np
6 plt.style.use('fivethirtyeight')
7 import folium
8 import folium.plugins
9 from matplotlib import animation, rc
10 import io
11 import base64
12 from IPython.display import HTML, display
13 import warnings
14 warnings.filterwarnings('ignore')
15 import codecs
16
```

Reading csv file

```
In [5]: 1 terror = pd.read_csv("Desktop/globalterrorismdb_0718dist.csv", encoding = "I
2
```

```
In [6]: 1 terror.head(5)
```

```
Out[6]:
```

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

5 rows × 135 columns



```

In [10]: 1 # Which Region had the most terrorism attacks??
2 print('The region of',terror['region_txt'].value_counts(dropna=True, normali
3 print(' '))
4 # Name of the city, village, or town in which the incident occurred
5 print(terror['city'].value_counts(dropna=True, normalize=False, ascending=Fa
6 print(' '))
7 print('The most known city that had terror attacks was' ,terror['city'].valu
8 print(' '))
9 #Most notably used weapon
10 print('The most used weapon in terror attacks was' ,terror['weaptype1_txt'].
11 print(' '))
12 #most known country
13 print('The most known country with terror attacks was' ,terror['country_txt'
14 print(' '))
15 #1 = "Yes" The incident was a suicide attack. 0 = "No" There is no indicatio
16 s = terror['suicide'].value_counts(normalize=True).mul(100).round(1).astype(
17 print('Out of',terror['suicide'].value_counts(dropna=True, normalize=False,
18 print(' '))
19 print('The most preferred method of attack was',terror['attacktype1_txt'].va
20 print(' '))
21 t = terror['targtype1_txt'].value_counts(normalize=True).mul(100).round(1).a
22 print('The main targets of terrorists were',terror['targtype1_txt'].value_co

```

The region of Middle East & North Africa had the highest amount of Terrorist Attacks totalled 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

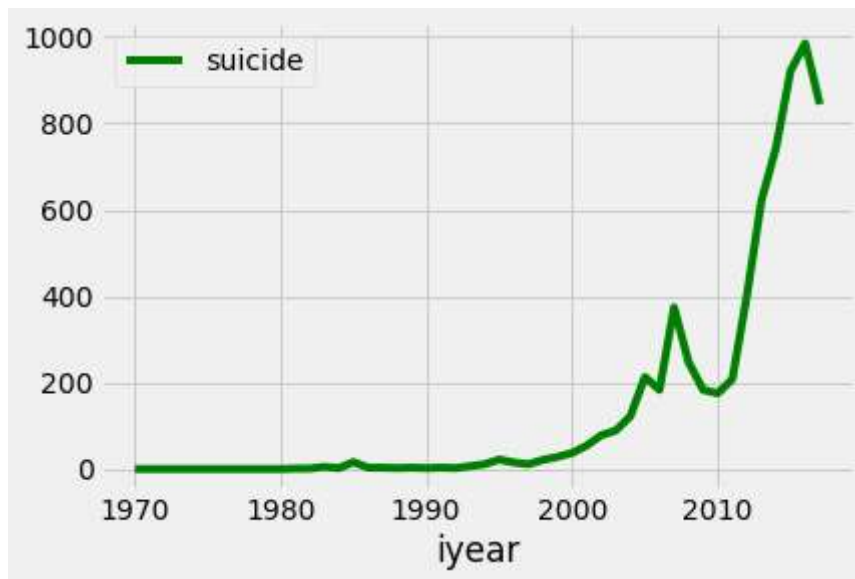
The most known country with terror attacks was Iraq totalled at 24636

Out of 181691 total attacks 3.7% were suicide attacks

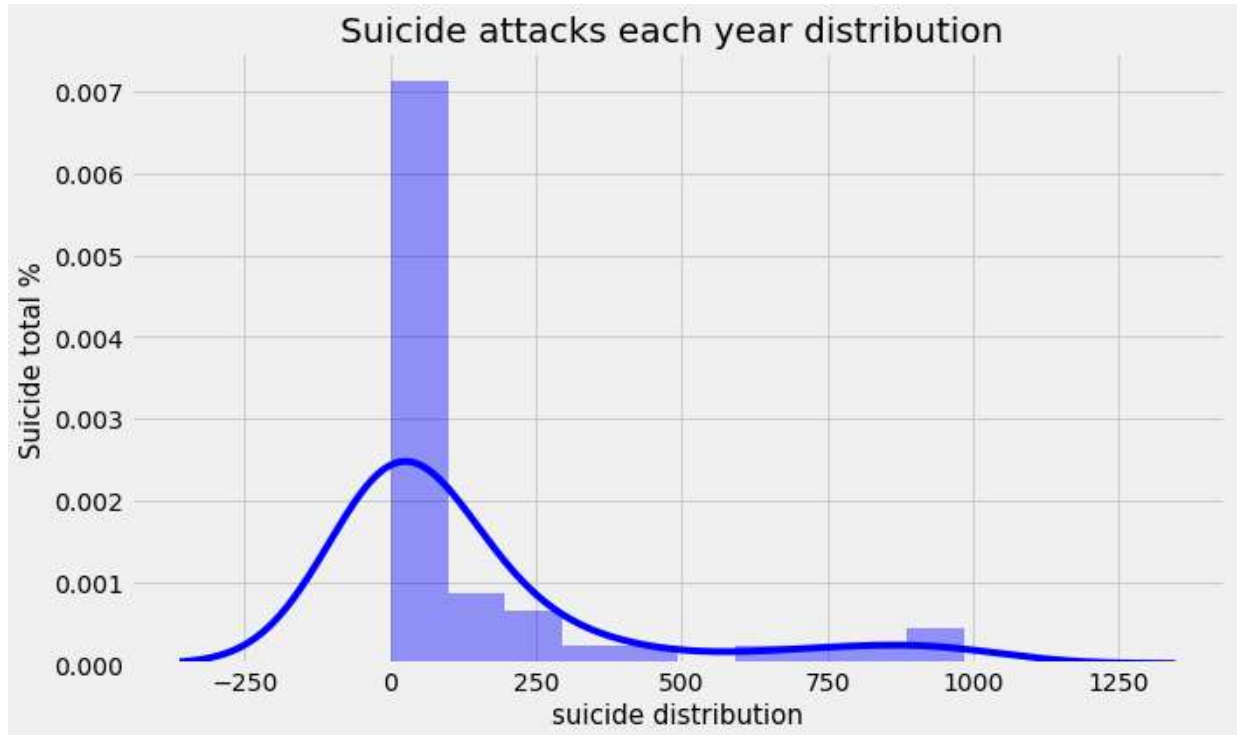
The most preferred method of attack was Bombing/Explosion totalling at 88255

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

```
In [44]: 1 #suicides per year  
2 #suicide attacks, both success's and fails  
3 suicides_by_year = terror[["iyear","suicide"]].groupby("iyear").aggregate(np  
4 suicides_by_year.plot(color = 'g');
```

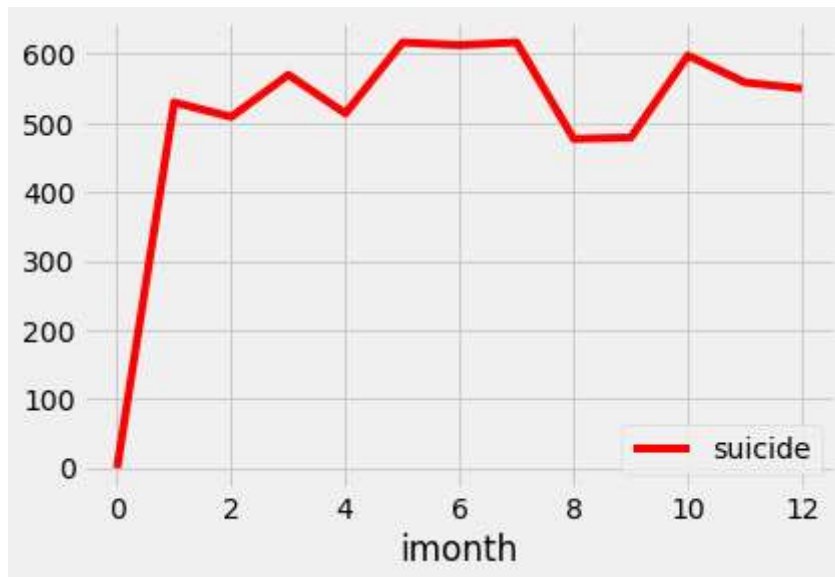


```
In [51]: 1 #suicide attacks each year distribution
2 plt.figure(figsize=(10,6))
3 fig = sns.distplot(terror[["iyear","suicide"]].groupby("iyear").aggregate(np
4 fig.set_xlabel("suicide distribution",size=15)
5 fig.set_ylabel("Suicide total %",size=15)
6 plt.title('Suicide attacks each year distribution ',size = 20)
7 plt.show()
```

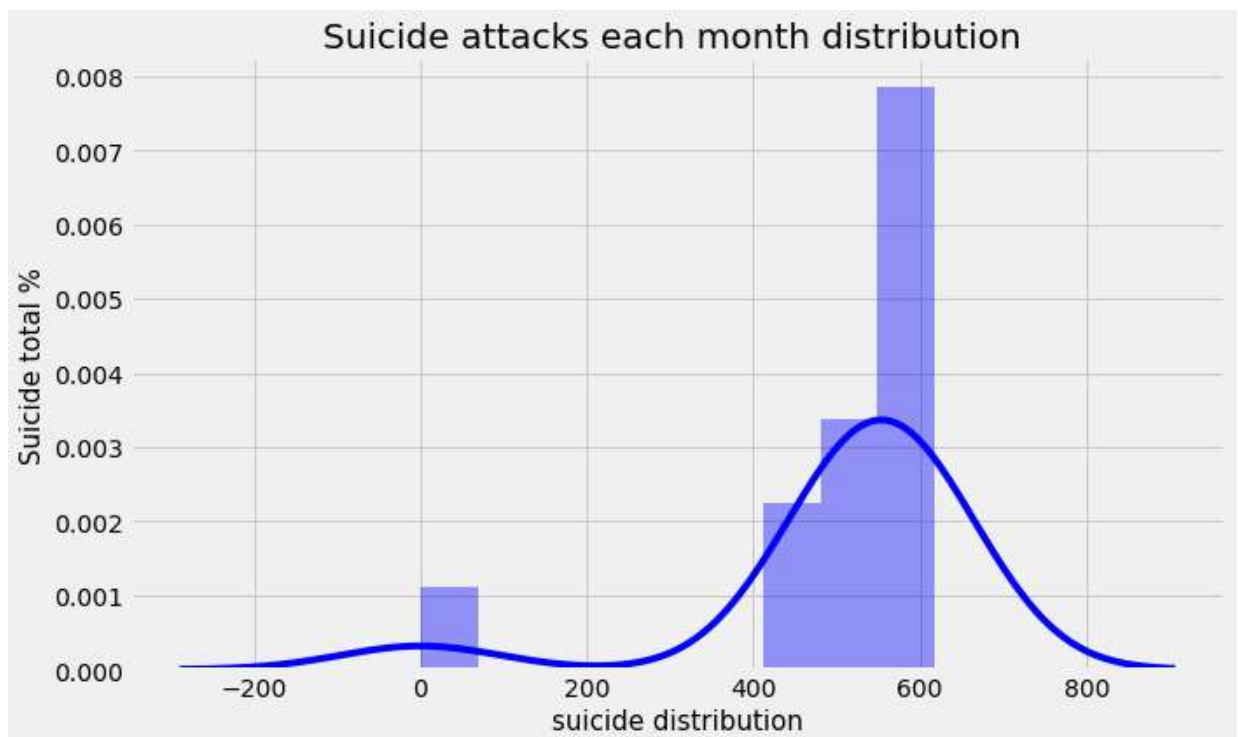


```
In [13]: 1 #suicide attacks, both success's and fails
2 suicides_by_year = terror[["iyear","suicide"]].groupby("iyear").aggregate(np
```

```
In [52]: 1 #suicides per month
2 #suicide attacks, both success's and fails
3 suicides_by_month = terror[["imonth","suicide"]].groupby("imonth").aggregate
4 suicides_by_month.plot(color = 'r');
```

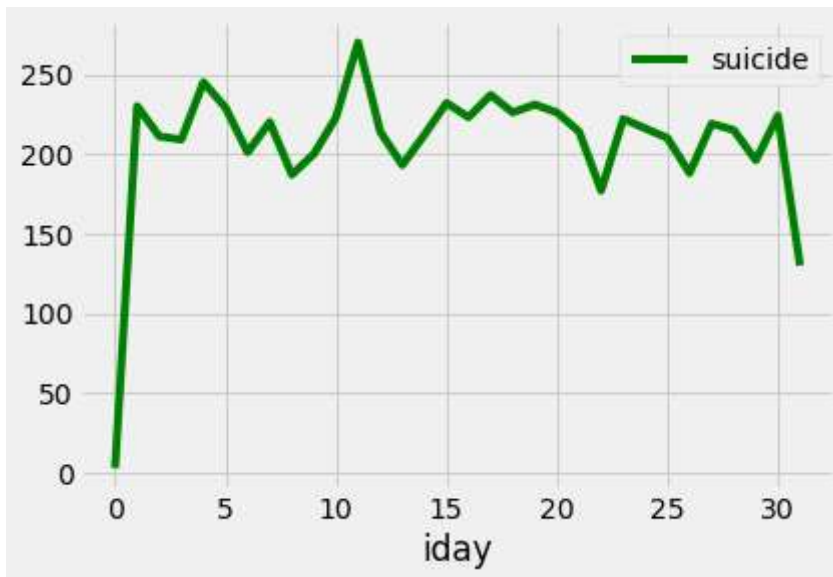


```
In [53]: 1 #suicide attacks distribution each month
2 plt.figure(figsize=(10,6))
3 fig = sns.distplot(terror[["imonth","suicide"]].groupby("imonth").aggregate(
4 fig.set_xlabel("suicide distribution",size=15)
5 fig.set_ylabel("Suicide total %",size=15)
6 plt.title('Suicide attacks each month distribution ',size = 20)
7 plt.show()
```



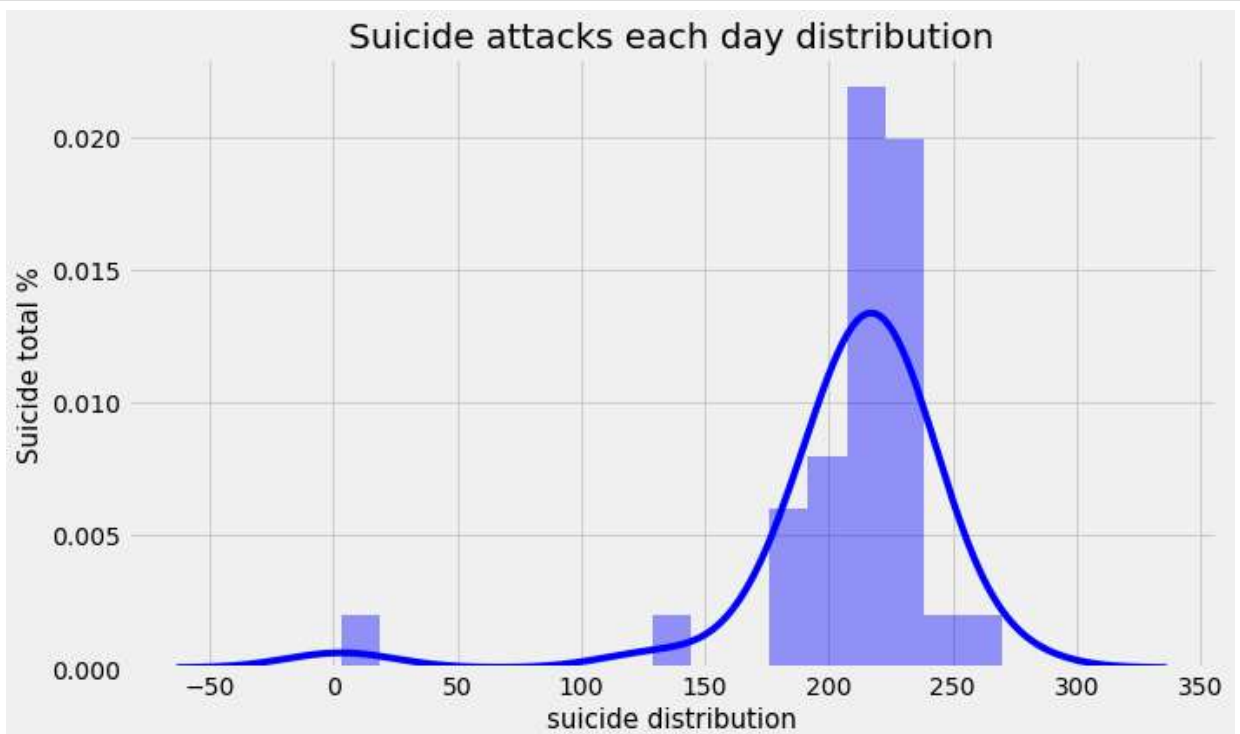
In [54]:

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

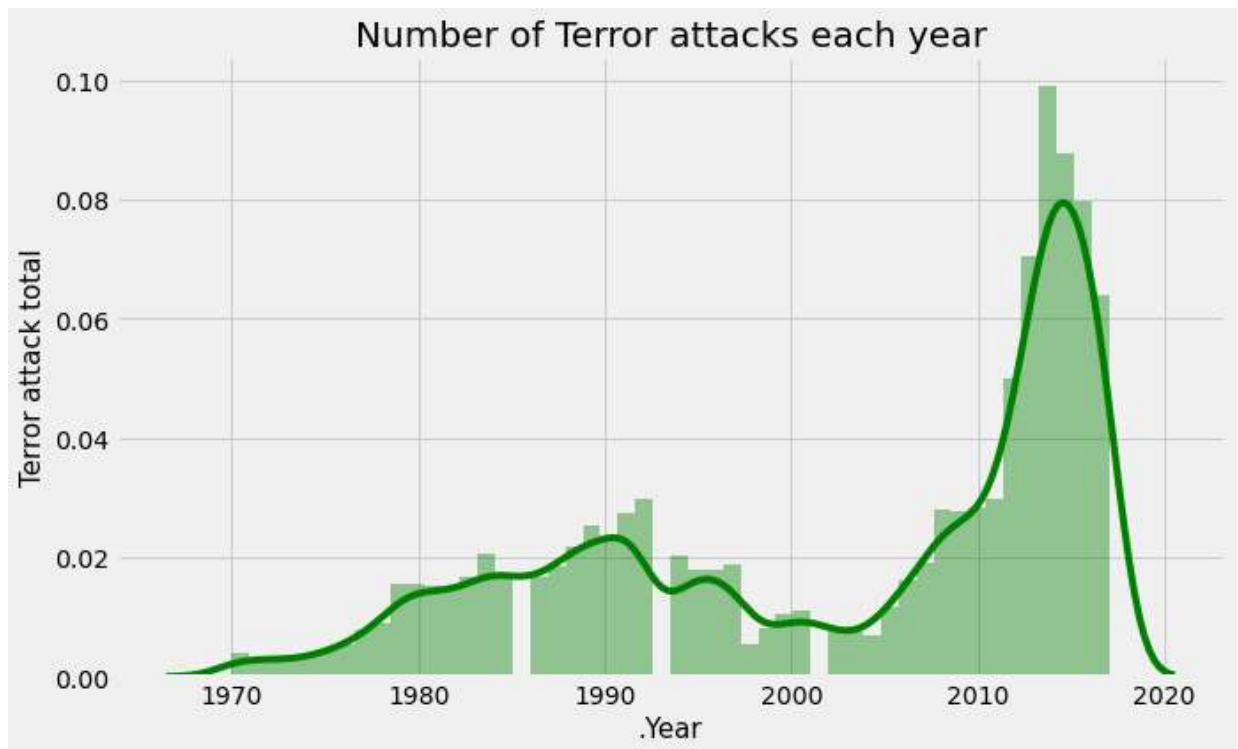


In [55]:

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



```
In [58]: 1 #Overall did the number of terror attacks increase?
2 #no. of terror attacks
3 plt.figure(figsize=(10,6))
4 fig = sns.distplot(terror["iyear"].values, color = 'g')
5 fig.set_xlabel(".Year",size=15)
6 fig.set_ylabel("Terror attack total",size=15)
7 plt.title('Number of Terror attacks each year',size = 20)
8 plt.show()
```



```
In [33]: 1 !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 (from plotly) (1.15.0)

Installing collected packages: tenacity, plotly

Successfully installed plotly-5.5.0 tenacity-8.0.1


```
In [34]: 1 #Can we visually see what parts of the world are most targetted? using plotly
2 import plotly.express as px
3
4 #to avoid lagging, only show the first 1k instead of the 180,000 data points
5 geog = terror.head(1000)
6 geog = geog[['latitude', 'longitude']]
7 fig = px.scatter_geo(geog, lat='latitude', lon='longitude')
8
9 fig.show()
```

```
In [35]: 1 #Can we visually narrow this down further?
2 geog1 = terror.head(1000)
```

```
In [60]: 1 fig = px.scatter_mapbox(geog1, lat='latitude', lon='longitude', hover_name="
2           color_discrete_sequence=["fuchsia"], zoom=1, height=
3 fig.update_layout(
4     mapbox_style="white-bg",
5     mapbox_layers=[
6         {
7             "below": 'traces',
8             "sourcetype": "raster",
9             "sourceattribution": "United States Geological Survey",
10            "source": [
11                "https://basemap.nationalmap.gov/arcgis/rest/services/USGSIm
12            ]
13        }
14    ])
15 fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
16 fig.show()
```

In [37]:

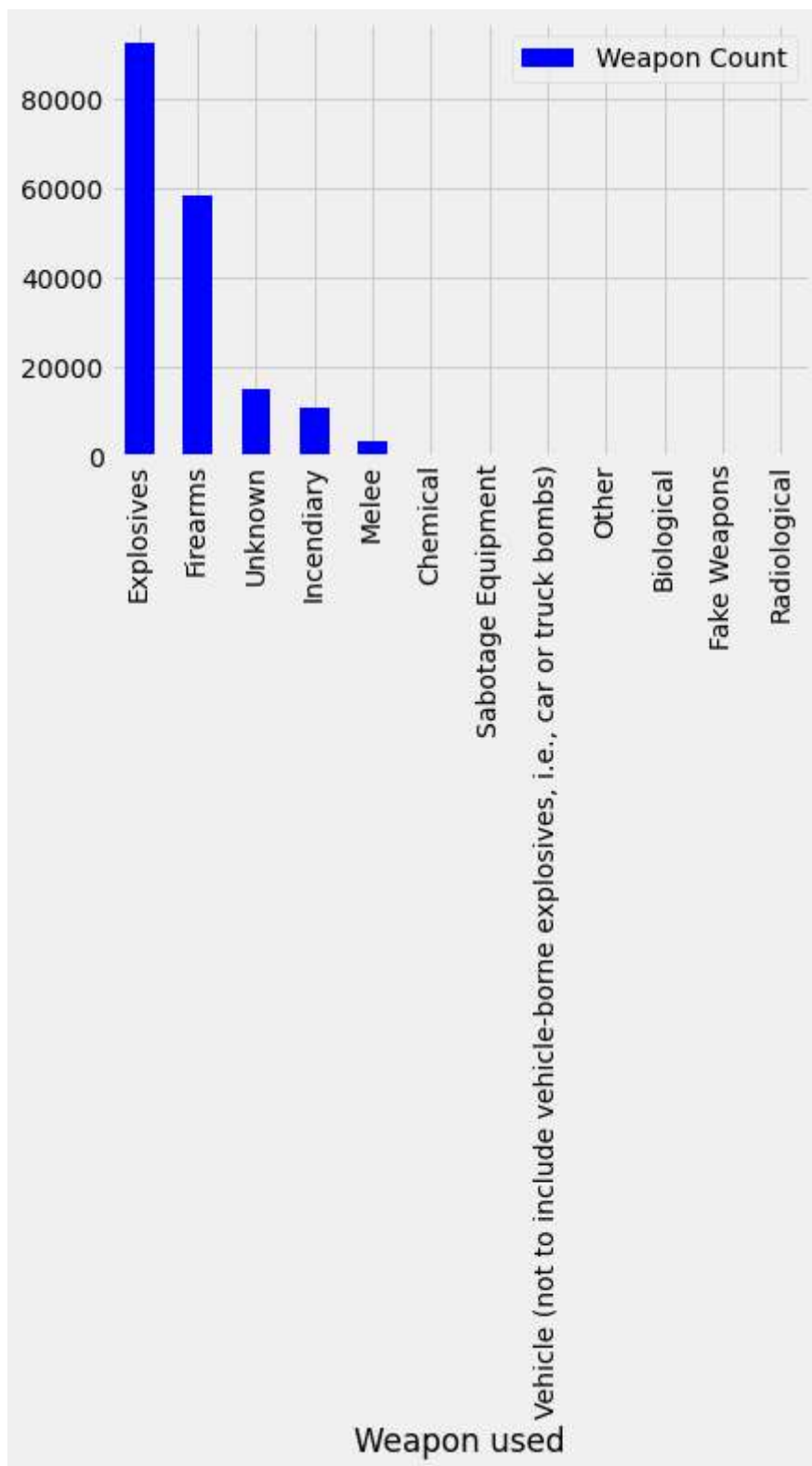
```
1 #Main Weapons:Explosives
2 from pandas import DataFrame
3
4
5 weapons = terror['weaptype1_txt'].value_counts()
6 weapons = DataFrame(weapons)
7
8 weapons.reset_index(level=0, inplace=True)
9 weapons.columns = ['Weapon used', 'Weapon Count']
10 weapons
```

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

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



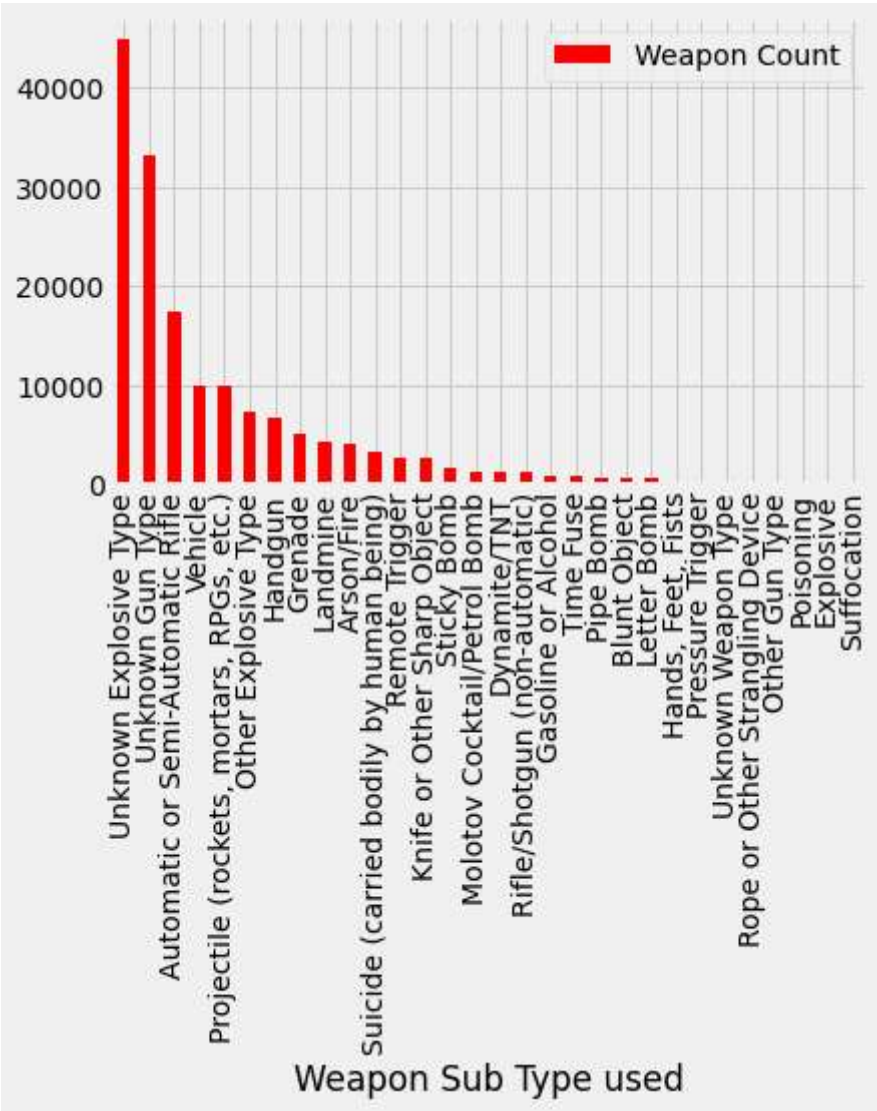

```
In [39]: 1 #Sub Weapon: unknown explosive type. What most terrorists most likely combin
2 subweapons = terror['weapsubtype1_txt'].value_counts()
3 subweapons = DataFrame(subweapons)
4
5 subweapons.reset_index(level=0, inplace=True)
6 subweapons.columns = ['Weapon Sub Type used', 'Weapon Count']
7 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]:

1 subweapons = subweapons.plot.bar(x='Weapon Sub Type used', y='Weapon Count',



In []: 1

In []: 1

In []: 1

In []: 1

In []: 1

In []: 1

In []: 1

In []: 1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1

In []:

1