Akash Prasad Mishra

Global Terrorism Exploratory Data Analysis

Import Necessaries libraries

In [3]:
```
import pandas as pd
import plotly.express as px
import numpy as np
from PIL import Image
```

In [326…
```
# Load Dataset
Dataset = pd.read_csv("D:\Data_sample\globalterrorismdb_0718dist.csv",encoding="lat
```

```
C:\Users\mishr\AppData\Local\Temp\ipykernel_17520\2001912989.py:2: DtypeWarning:

Columns (4,6,31,33,61,62,63,76,79,90,92,94,96,114,115,121) have mixed types. Specify
dtype option on import or set low_memory=False.
```

In [15]:
```
# Pritn Top 4 Rows

Dataset.head(4)
```

Out[15]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country
0	197000000001	1970	7	2	NaN	0	NaN	58	Domin Repu
1	197000000002	1970	0	0	NaN	0	NaN	130	Me
2	197001000001	1970	1	0	NaN	0	NaN	160	Philipp
3	197001000002	1970	1	0	NaN	0	NaN	78	Gre

4 rows × 135 columns

In [32]:
```
# Check Columns

Dataset.columns
```

Out[32]:
```
Index(['eventid', 'iyear', 'imonth', 'iday', 'approxdate', 'extended',
       'resolution', 'country', 'country_txt', 'region',
       ...
       'addnotes', 'scite1', 'scite2', 'scite3', 'dbsource', 'INT_LOG',
       'INT_IDEO', 'INT_MISC', 'INT_ANY', 'related'],
      dtype='object', length=135)
```

In [19]:
```
# Check Data Types of each columns

Dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181691 entries, 0 to 181690
Columns: 135 entries, eventid to related
dtypes: float64(55), int64(22), object(58)
memory usage: 187.1+ MB
```

In [20]:
```
Dataset.head(4)
```

Out[20]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country
0	197000000001	1970	7	2	NaN	0	NaN	58	Domin Repu
1	197000000002	1970	0	0	NaN	0	NaN	130	Me
2	197001000001	1970	1	0	NaN	0	NaN	160	Philipp
3	197001000002	1970	1	0	NaN	0	NaN	78	Gre

4 rows × 135 columns

In [35]:
```
#  Check Nan Values

Dataset.isna().sum()
```

Out[35]:
```
eventid            0
iyear              0
imonth             0
iday               0
approxdate    172452
               ...
INT_LOG            0
INT_IDEO           0
INT_MISC           0
INT_ANY            0
related       156653
Length: 135, dtype: int64
```

In [ ]:
```
# Now We will drop column if there over 30% Nan values
```

In [337…
```
def DropNa(Dataset):

    List_col = Dataset.columns

    index = 0

    Total_Values = len(Dataset[List_col[0]])

    for i in Dataset.isna().sum().values:

        if((i/Total_Values)*100 >30):
```

```
            Dataset =  Dataset.drop([List_col[index]],axis=1)

        index+=1

    return Dataset
```

In [338…

```
# Call Dropna
Dataset = DropNa(Dataset)
```

Year-wise Crimes Globally

In [340…

```
# use Year_dict keys and values in dict. Top10US
YearViseGlob ={'year':Dataset['iyear'].value_counts().keys(),'Crimes':Dataset['iyea

# Create bar chart with Plotly Express
fig = px.bar(YearViseGlob, x='year', y='Crimes', height = 500, title='Year-vise  Cr

# URL of a image
image_url = im

# set color to all mar
colors = ['#FCF8F3'] * len(YearViseGlob['year'])

# Open Image
open_image = Image.open(r"C:\Users\mishr\Downloads\download (5).jpg")

# Add image as background
fig.add_layout_image(
    dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,  # Adjust width
        sizey=2,  # Adjust height
        sizing="stretch",
        opacity=0.5,
        layer="below"
    )
)

fig.update_layout(paper_bgcolor = '#17153B',title_font_color="#F9F5F6",plot_bgcolor
                  ,font_size=13)



# Update layout to ensure the image covers the full background
fig.update_layout(
    template="plotly_white",
    images=[dict(
        source=image_url,
        xref="paper",
        yref="paper",
```

```
        x=0,
        y=1,
        sizex=1,
        sizey=2,
        sizing="stretch",
        opacity=0.4,
        layer="below"
    )],
    margin=dict(l=0, r=0, t=50, b=0))

# Update text
fig.update_traces(textfont_size=12, textangle = 0, textposition='outside',cliponaxi

# Show chart

fig.show()
```

In [39]:
```
# Dataset After Droping some Columns
Dataset.columns
```

Out[39]:
```
Index(['eventid', 'iyear', 'imonth', 'iday', 'extended', 'country',
       'country_txt', 'region', 'region_txt', 'provstate', 'city', 'latitude',
       'longitude', 'specificity', 'vicinity', 'crit1', 'crit2', 'crit3',
       'doubtterr', 'multiple', 'success', 'suicide', 'attacktype1',
       'attacktype1_txt', 'targtype1', 'targtype1_txt', 'targsubtype1',
       'targsubtype1_txt', 'corp1', 'target1', 'natlty1', 'natlty1_txt',
       'gname', 'guncertain1', 'individual', 'weaptype1', 'weaptype1_txt',
       'weapsubtype1', 'weapsubtype1_txt', 'nkill', 'nwound', 'property',
       'ishostkid', 'dbsource', 'INT_LOG', 'INT_IDEO', 'INT_MISC', 'INT_ANY'],
      dtype='object')
```

Country With crimes

In [90]:

Out[90]:
```
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
```

In [158…
```
# use Year_dict keys and values in dict. Top10US
YearViseGlob ={'Country':Dataset['country_txt'].value_counts().keys()[:10],'Crimes'

# Create bar chart with Plotly Express
fig = px.bar(YearViseGlob, y='Country', x='Crimes', height = 500, title='Top 10 Cou

# URL of a image
image_url = im

# set color to all mar
colors = ['#FF6969'] * len(YearViseGlob['Country'])



# Open Image
open_image = Image.open(r"C:\Users\mishr\Downloads\download (5).jpg")

# Add image as background
fig.add_layout_image(
    dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,  # Adjust width
        sizey=2,  # Adjust height
```

```
        sizing="stretch",
        opacity=0.5,
        layer="below"
    )
)

fig.update_layout(paper_bgcolor = '#17153B',title_font_color="#F9F5F6",plot_bgcolor
                  ,font_size=13)



# Update layout to ensure the image covers the full background
fig.update_layout(
    template="plotly_white",
    images=[dict(
        source=image_url,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,
        sizey=2,
        sizing="stretch",
        opacity=0.4,
        layer="below"
    )],
    margin=dict(l=0, r=0, t=50, b=0))

# Update text
fig.update_traces(textfont_size=12, textangle = 0, textposition='outside',cliponaxi

# Show chart

fig.show()
```

Crimes trend in india

In [155…
```
India_YearWise = Dataset[Dataset['country_txt']=='India']['iyear'].value_counts()
```

In [166…
```
# India_YearWise = {'year':India_YearWise.keys(),"crime":India_YearWise.values}


# Show The revenue trend  year vise using line plot
fig = px.scatter(India_YearWise.values[::-1],text=India_YearWise.values[::-1],color
                     "value": "Total crime (in K)",
                     "index": "Rate",
    "color":"year"
                 })

# Update the style that apllied on line (text font,size etc.)
fig.update_traces(textfont_size=12,cliponaxis=False,textfont=dict(color='red'),line

# Visualize The line chart

fig.show()
```

Types Of Crime Trend in india Over the Years

In [ ]:
```
CrimeType = Dataset[Dataset['country_txt']=='India'][['attacktype1_txt','iyear']]

CrimeType =  CrimeType.groupby(['iyear','attacktype1_txt']).size().reset_index(name
```

In [223…
```
# Show The revenue trend  year vise using line plot
fig = px.line(CrimeType,x='iyear',y='count',color = 'attacktype1_txt',title='Crime
                    "iyear": "Year",
                    "count": "Total",
    'attacktype1_txt':'Type of Crime'
                },width=1100, height=400)


# set color to all mar
colors = ['#FF6969'] * len(CrimeType['iyear'])



# Open Image
open_image = Image.open(r"C:\Users\mishr\Downloads\Sweden Makes It LEGAL for Jihadi

# Add image as background
```

```
fig.add_layout_image(
    dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,  # Adjust width
        sizey=2,  # Adjust height
        sizing="stretch",
        opacity=0.5,
        layer="below"
    )
)

fig.update_layout(paper_bgcolor = '#17153B',title_font_color="#F9F5F6",plot_bgcolor
                  ,font_size=13)



# Update layout to ensure the image covers the full background
fig.update_layout(
    template="plotly_white",
    images=[dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,
        sizey=2,
        sizing="stretch",
        opacity=0.7,
        layer="below"
    )],
    margin=dict(l=0, r=0, t=80, b=10))

# Update text
# fig.update_traces(textfont_size=12, textangle = 0, textposition='outside',clipona

# Show chart

fig.show()
```

Types Of Crime Trend in Worlds Over the Years

In [352…

```
CrimeType = Dataset[['attacktype1_txt','iyear']]

CrimeType =  CrimeType.groupby(['iyear','attacktype1_txt']).size().reset_index(name
```

In [353…

```
# Show The revenue trend  year vise using line plot
fig = px.line(CrimeType,x='iyear',y='count',color = 'attacktype1_txt',title='Crime
                      "iyear": "Year",
                      "count": "Total",
    'attacktype1_txt':'Type of Crime'
                  })


# set color to all mar
colors = ['#FF6969'] * len(CrimeType['iyear'])



# Open Image
open_image = Image.open(r"C:\Users\mishr\Downloads\Sweden Makes It LEGAL for Jihadi

# Add image as background
fig.add_layout_image(
    dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
```

```
        sizex=1,  # Adjust width
        sizey=2,  # Adjust height
        sizing="stretch",
        opacity=0.5,
        layer="below"
    )
)

fig.update_layout(paper_bgcolor = '#17153B',title_font_color="#F9F5F6",plot_bgcolor
                  ,font_size=13)



# Update layout to ensure the image covers the full background
fig.update_layout(
    template="plotly_white",
    images=[dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,
        sizey=2,
        sizing="stretch",
        opacity=0.4,
        layer="below"
    )],
    margin=dict(l=0, r=0, t=50, b=0))

# Update text
# fig.update_traces(textfont_size=12, textangle = 0, textposition='outside',clipona

# Show chart

fig.show()
```

Suicide Cases in india

In [350…

```
CrimeType = Dataset[Dataset['country_txt']=='India'][['suicide','iyear']]

CrimeType = CrimeType[CrimeType['suicide']==1]

# Aggregate the Column

CrimeType =  CrimeType.groupby(['iyear','suicide']).size().reset_index(name='count'
```

In [351…

```
# Show The revenue trend  year vise using line plot
fig = px.line(CrimeType[['iyear','count']],x='iyear',y='count',text='count',title='
                     "iyear": "Year",
                     "count": "Total",
    'attacktype1_txt':'Type of Crime'
                  })


# set color to all mar
colors = ['#FF6969'] * len(CrimeType['iyear'])
```

```
# Open Image
open_image = Image.open(r"C:\Users\mishr\Downloads\wp2782547-suicide-wallpaper.jpg"

# Add image as background
fig.add_layout_image(
    dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,  # Adjust width
        sizey=2,  # Adjust height
        sizing="stretch",
        opacity=0.5,
        layer="below"
    )
)

fig.update_layout(paper_bgcolor = '#17153B',title_font_color="#F9F5F6",plot_bgcolor
                  ,font_size=13)



# Update layout to ensure the image covers the full background
fig.update_layout(
    template="plotly_white",
    images=[dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,
        sizey=2,
        sizing="stretch",
        opacity=0.4,
        layer="below"
    )],
    margin=dict(l=0, r=0, t=50, b=0))

# Update text
fig.update_traces(textposition="bottom center")

# Show chart

fig.show()
```

Frequecy trends Of different types of crimes in different country

In [312…

```
# Aggregate the Column

CrimeType = Dataset[['country_txt','attacktype1_txt','iyear']].groupby(['attacktype
```

In [313…

```
# Show The revenue trend  year vise using line plot
fig = px.scatter(CrimeType,x='country_txt',y='count',size='count',color = 'attackty
                     "country_txt": "Country",
                     "count": "crime frequency",
                 },width=1100, height=500)


# set color to all mar
colors = ['#FF6969'] * len(CrimeType['country_txt'])



# Open Image
open_image = Image.open(r"C:\Users\mishr\Downloads\Sweden Makes It LEGAL for Jihadi

# Add image as background
fig.add_layout_image(
```

```
    dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,  # Adjust width
        sizey=2,  # Adjust height
        sizing="stretch",
        opacity=0.5,
        layer="below"
    )
)

fig.update_layout(paper_bgcolor = '#17153B',title_font_color="#F9F5F6",plot_bgcolor
                  ,font_size=13)



# Update layout to ensure the image covers the full background
fig.update_layout(
    template="plotly_white",
    images=[dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,
        sizey=2,
        sizing="stretch",
        opacity=0.4,
        layer="below"
    )],
    margin=dict(l=10, r=10, t=50, b=0))

# Update text
# fig.update_traces(textfont_size=12, textangle = 0, textposition='outside',clipona

# Show chart

fig.show()
```

Region with Most people kills in different crimes

In [342…
```
# Calculate mean of nkill column
Dataset['nkill'] = Dataset['nkill'].fillna(Dataset['nkill'].mean())
```

In [343…
```
region_vise = Dataset[['region_txt','nkill']]
```

In [346…
```
# Aggregate the Column

CrimeType = region_vise.groupby(['region_txt']).size().reset_index(name="Kills")
```

In [348…
```
#

# Create bar chart with Plotly Express
fig = px.bar(CrimeType, y='Kills', x='region_txt', height = 500, title='People kill



# set color to all mar
colors = ['#FF6969'] * len(CrimeType['region_txt'])
```

```python
# Open Image
open_image = Image.open(r"C:\Users\mishr\Downloads\download (5).jpg")

# Add image as background
fig.add_layout_image(
    dict(
        source=open_image,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,  # Adjust width
        sizey=2,  # Adjust height
        sizing="stretch",
        opacity=0.5,
        layer="below"
    )
)

fig.update_layout(paper_bgcolor = '#17153B',title_font_color="#F9F5F6",plot_bgcolor
                  ,font_size=13)



# Update layout to ensure the image covers the full background
fig.update_layout(
    template="plotly_white",
    images=[dict(
        source=image_url,
        xref="paper",
        yref="paper",
        x=0,
        y=1,
        sizex=1,
        sizey=2,
        sizing="stretch",
        opacity=0.4,
        layer="below"
    )],
    margin=dict(l=0, r=0, t=50, b=0))

# Update text
fig.update_traces(textfont_size=12, textangle = 0, textposition='outside',cliponaxi

# Show chart

fig.show()
```

In [ ]: