TASK 3 LGMVIP - Jupyter Notebo

localhost:8888/notebooks/TASK%203%20LGMVIP.ipynb

jupyter TASK 3 LGMVIP Last Checkpoint: 2 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Markdown

LETS GROW MORE INTERNSHIP DATA SCEINCE

DEC BATCH TASK 3

ANURAG JADHAV

TASK 3 - Image to Pencil Sketch with Python

Importing Libraries

In [15]:
```
import cv2
import matplotlib.pyplot as plt
```

Uploading the image

In [22]:
```
image = cv2.imread(r"C:\Users\anura\Downloads\IMG_20211003_110625.jpg")
```

Converting the image to RGB format

In [23]:
```
image2 = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image2)
plt.axis(False)
plt.show()
```

TASK 3 LGMVIP - Jupyter Notebo

localhost:8888/notebooks/TASK%203%20LGMVIP.ipynb

jupyter TASK 3 LGMVIP Last Checkpoint: 2 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Markdown

Importing Libraries

In [15]:
```
import cv2
import matplotlib.pyplot as plt
```

Uploading the image

In [22]:
```
image = cv2.imread(r"C:\Users\anura\Downloads\IMG_20211003_110625.jpg")
```

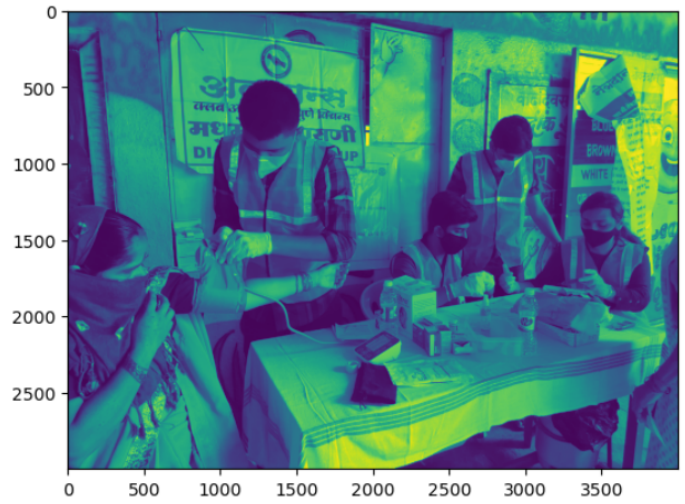
Converting the image to RGB format

In [23]:
```
image2 = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image2)
plt.axis(False)
plt.show()
```

Converting the image to greyscale image

In [24]:
```
grey_scale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.imshow(grey_scale_image)
```

Out[24]:
```
<matplotlib.image.AxesImage at 0x28c822d6550>
```



Inverting the image

In [25]:
```
inverted_image = cv2.bitwise_not(grey_scale_image)
plt.imshow(inverted_image)
```

Out[25]:
```
<matplotlib.image.AxesImage at 0x28c823169a0>
```

TASK 3 LGMVIP - Jupyter Notebo

localhost:8888/notebooks/TASK%203%20LGMVIP.ipynb

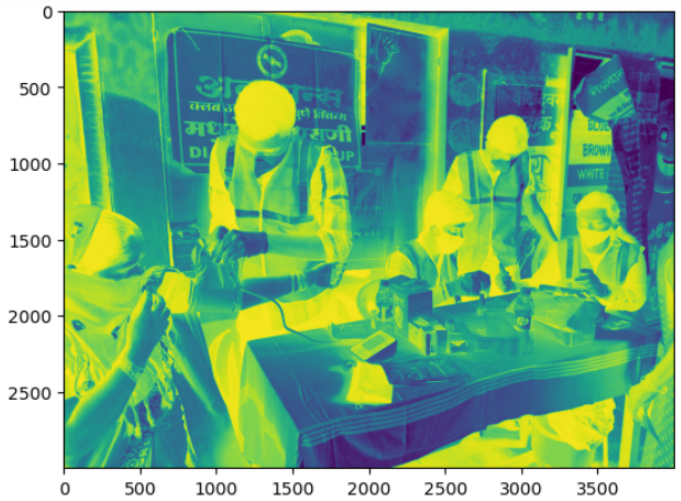
jupyter TASK 3 LGMVIP Last Checkpoint: 2 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

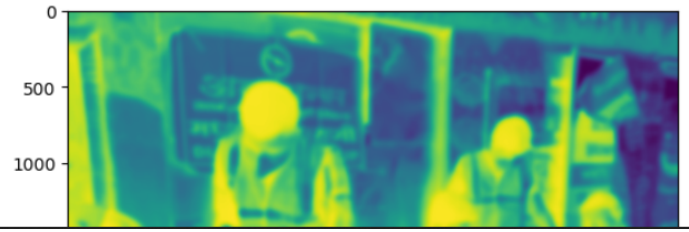
Run Markdown

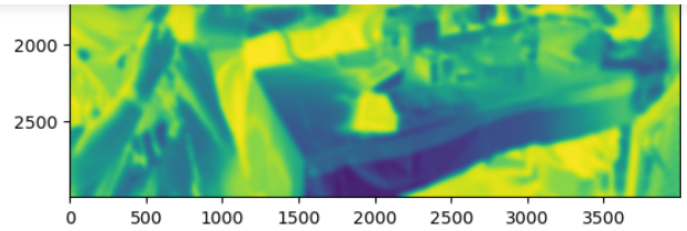


Blurring the image

In [26]:
```
blurred_image = cv2.GaussianBlur(inverted_image, (111,111), 0)
plt.imshow(blurred_image)
```

Out[26]:
```
<matplotlib.image.AxesImage at 0x28c82388e20>
```

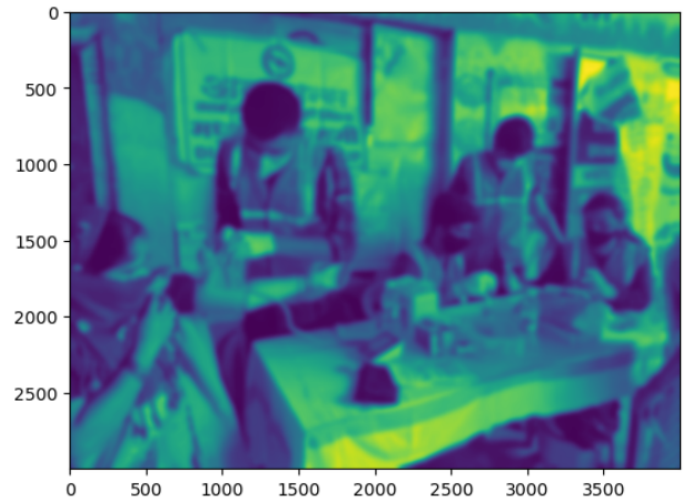




Inverting the blurred image

In [27]:
```
inverted_blur_image = cv2.bitwise_not(blurred_image)
plt.imshow(inverted_blur_image)
```

Out[27]:
```
<matplotlib.image.AxesImage at 0x28c823f1940>
```



TASK 3 LGMVIP - Jupyter Notebo

localhost:8888/notebooks/TASK%203%20LGMVIP.ipynb

jupyter TASK 3 LGMVIP Last Checkpoint: 2 minutes ago (autosaved)

Logout

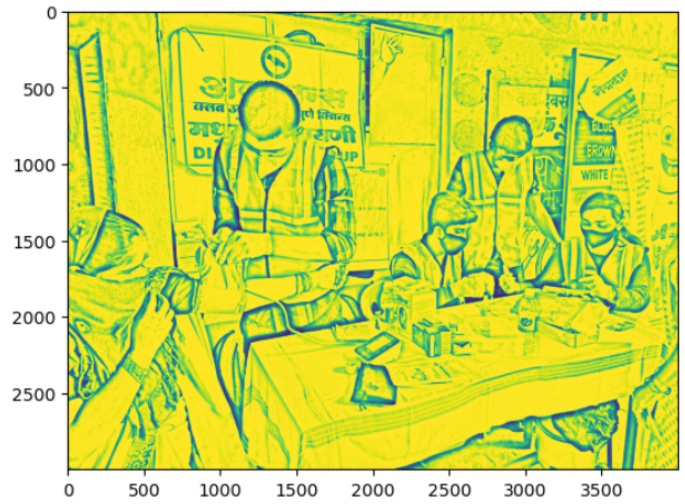
File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Markdown

In [28]:
```
sketch_image = cv2.divide(grey_scale_image, inverted_blur_image, scale=256.0)
plt.imshow(sketch_image)
```

Out[28]: <matplotlib.image.AxesImage at 0x28c825df5e0>



Saving the image

In [29]:
```
cv2.imwrite("sketch.png", sketch_image)
```

Out[29]: True

Converting the image back to BRG format

In [30]:
```
plt.figure(figsize = (10,10))
image3 = cv2.cvtColor(sketch_image, cv2.COLOR_RGB2BGR)
plt.imshow(image3)
```

TASK 3 LGMVIP - Jupyter Notebo

localhost:8888/notebooks/TASK%203%20LGMVIP.ipynb

jupyter TASK 3 LGMVIP Last Checkpoint: 2 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Markdown

```
plt.axis(False)
plt.show()
```

In [ ]:

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 5 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

LETS GROW MORE INTERNSHIP DATA SCEINCE

DEC BATCH TASK 4

ANURAG JADHAV

TASK 4 - Exploratory Data Analysis on Dataset - Terrorism

Importing Libraries

In [6]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

Upload dataset

In [16]:
```
data = pd.read_csv(r"C:\Users\anura\Downloads\globalterrorismdb_0718dist.csv",encoding = 'ISO-8859-1')
```

Displaying the first five rows of the dataset

In [37]:
```
data.head()
```

Out[37]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	...	addnotes	scite1	scite2	scite3	dbsource	INT_LOG
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS	0
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS	0

3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGIS	-9
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGIS	-9

5 rows × 135 columns

Displaying the last five rows of the dataset

In [18]:
```
data.tail()
```

Out[18]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	...	addnotes	scite1	scite2	scite3
181686	201712310022	2017	12	31	NaN	0	NaN	182	Somalia	11	...	NaN	"Somalia: Al-Shabaab Militants Attack Army Che...	"Highlights: Somalia Daily Media Highlights 2 ...	"Highlights: Somalia Daily Media Highlights 1 ...
181687	201712310029	2017	12	31	NaN	0	NaN	200	Syria	10	...	NaN	"Putin's 'victory' in Syria has turned into a ...	"Two Russian soldiers killed at Hmeymim base i...	"Two Russian servicemen killed in Syria mortar...
181688	201712310030	2017	12	31	NaN	0	NaN	160	Philippines	5	...	NaN	"Maguindanao clashes trap tribe members," Phil...	NaN	NaN
181689	201712310031	2017	12	31	NaN	0	NaN	92	India	6	...	NaN	"Trader escapes grenade attack in Imphal," Bus...	NaN	NaN
181690	201712310032	2017	12	31	NaN	0	NaN	160	Philippines	5	...	NaN	"Security tightened in Cotabato following IED ...	"Security tightened in Cotabato City," Manila ...	NaN

5 rows × 135 columns

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 5 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

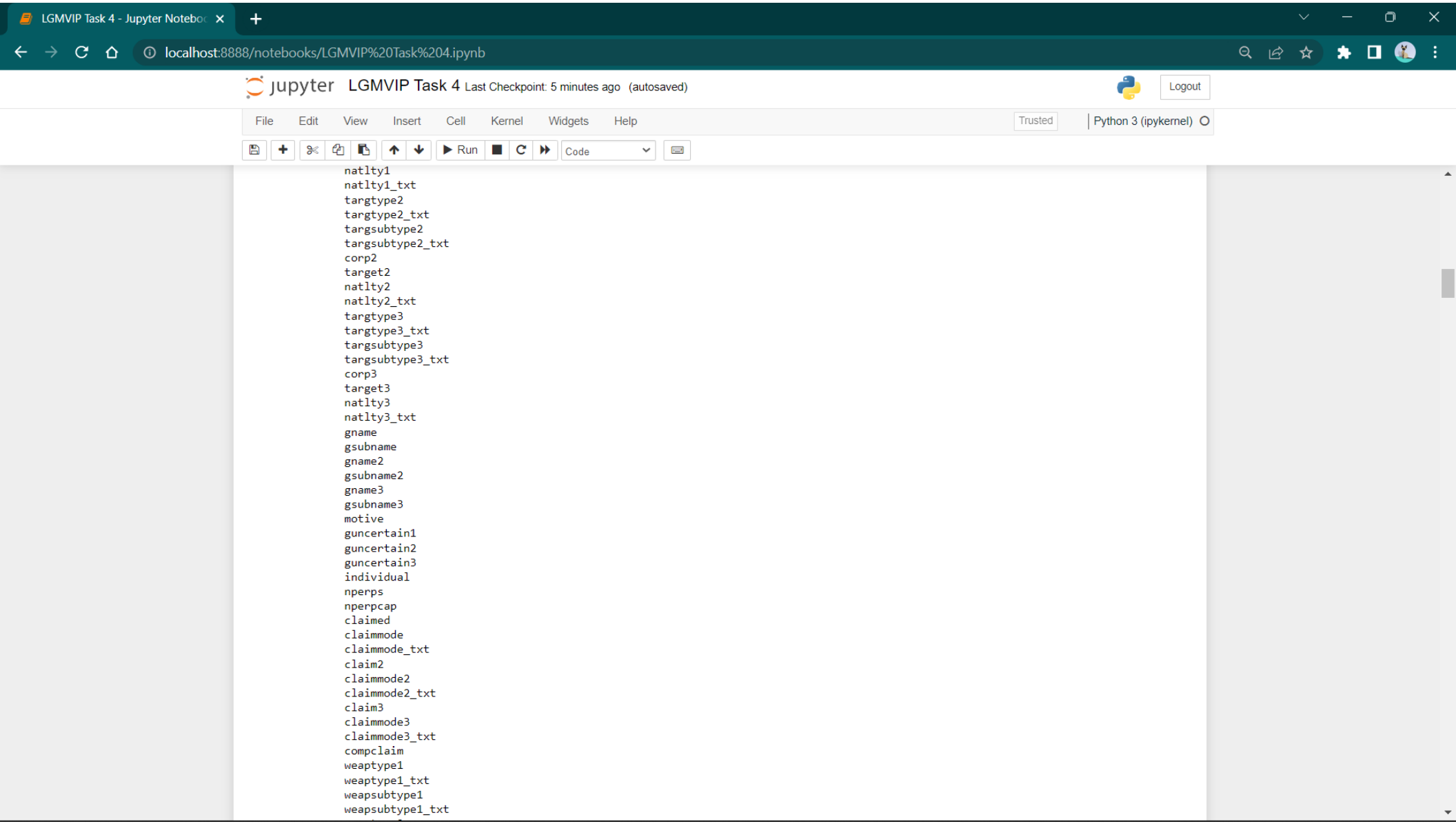
Trusted Python 3 (ipykernel)

Run Code

Displaying all the features present in the dataset

In [19]:
```
for column in data.columns:
  print(column)
```

```
eventid
iyear
imonth
iday
approxdate
extended
resolution
country
country_txt
region
region_txt
provstate
city
latitude
longitude
specificity
vicinity
location
summary
crit1
crit2
crit3
doubtterr
alternative
alternative_txt
multiple
success
suicide
attacktype1
attacktype1_txt
attacktype2
attacktype2_txt
attacktype3
attacktype3_txt
targtype1
targtype1_txt
targsubtype1
targsubtype1_txt
corp1
target1
```


LGMVIP Task 4 - Jupyter Notebo
localhost:8888/notebooks/LGMVIP%20Task%204.ipynb
jupyter LGMVIP Task 4 Last Checkpoint: 5 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
natlty1
natlty1_txt
targtype2
targtype2_txt
targsubtype2
targsubtype2_txt
corp2
target2
natlty2
natlty2_txt
targtype3
targtype3_txt
targsubtype3
targsubtype3_txt
corp3
target3
natlty3
natlty3_txt
gname
gsubname
gname2
gsubname2
gname3
gsubname3
motive
guncertain1
guncertain2
guncertain3
individual
nperps
nperpcap
claimed
claimmode
claimmode_txt
claim2
claimmode2
claimmode2_txt
claim3
claimmode3
claimmode3_txt
compclaim
weaptype1
weaptype1_txt
weapsubtype1
weapsubtype1_txt

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 5 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

Run Code

```
weapsubtype2
weapsubtype2_txt
weaptype3
weaptype3_txt
weapsubtype3
weapsubtype3_txt
weaptype4
weaptype4_txt
weapsubtype4
weapsubtype4_txt
weapdetail
nkill
nkillus
nkillter
nwound
nwoundus
nwoundte
property
propextent
propextent_txt
propvalue
propcomment
ishostkid
nhostkid
nhostkidus
nhours
ndays
divert
kidhijcountry
ransom
ransomamt
ransomamtus
ransompaid
ransompaidus
ransomnote
hostkidoutcome
hostkidoutcome_txt
nreleased
addnotes
scite1
scite2
scite3
dbsource
INT_LOG
INT_IDEO
INT_MISC
```

```
INT_MISC
INT_ANY
related
```

Displaying the shape and features of the dataset

In [20]:
```
data.shape
```

Out[20]:
```
(181691, 135)
```

Note:

Here I have selected the most relevant features of the Dataset instead of working on all the features these are:

In [24]:
```
df = data[["iyear", "imonth", "iday", "country_txt", "provstate", "region_txt",
           "longitude", "latitude", "attacktype1_txt", "target1", "nkill", "nwound",
           "summary", "gname", "targtype1_txt", "weaptype1_txt", "motive"]]
```

In [25]:
```
df.head()
```

Out[25]:

	iyear	imonth	iday	country_txt	provstate	region_txt	longitude	latitude	attacktype1_txt	target1	nkill	nwound	summary	gname	targtype1_
0	1970	7	2	Dominican Republic	NaN	Central America & Caribbean	-69.951164	18.456792	Assassination	Julio Guzman	1.0	0.0	NaN	MANO-D	Priv Citizen Prop
1	1970	0	0	Mexico	Federal	North America	-99.086624	19.371887	Hostage Taking (Kidnapping)	Nadine Chaval, daughter	0.0	0.0	NaN	23rd of September Communist League	Governm (Diploma
2	1970	1	0	Philippines	Tarlac	Southeast Asia	120.599741	15.478598	Assassination	Employee	1.0	0.0	NaN	Unknown	Journalist Me
3	1970	1	0	Greece	Attica	Western Europe	23.762728	37.997490	Bombing/Explosion	U.S. Embassy	NaN	NaN	NaN	Unknown	Governm (Diploma
4	1970	1	0	Japan	Fukouka	East Asia	130.396361	33.580412	Facility/Infrastructure Attack	U.S. Consulate	NaN	NaN	NaN	Unknown	Governm (Diploma

Renaming the columns of the dataset

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 5 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

Renaming the columns of the dataset

In [26]:
```
df.rename(columns={'iyear':'Year','imonth':'Month','iday':'Day','country_txt':'Country','provstate':'State',
                   'region_txt':'Region','longitude':'Longitude', 'latitude':'Latitude', 'attacktype1_txt':'AttackType',
                   'target1':'Target','nkill':'Killed','nwound':'Wounded','summary':'Summary','gname':'Group',
                   'targtype1_txt':'Target_type', 'weaptype1_txt':'Weapon_type','motive':'Motive'},inplace=True)
df.head()
```

Out[26]:

	Year	Month	Day	Country	State	Region	Longitude	Latitude	AttackType	Target	Killed	Wounded	Summary	Group	Target_type
0	1970	7	2	Dominican Republic	NaN	Central America & Caribbean	-69.951164	18.456792	Assassination	Julio Guzman	1.0	0.0	NaN	MANO-D	Private Citizens & Property
1	1970	0	0	Mexico	Federal	North America	-99.086624	19.371887	Hostage Taking (Kidnapping)	Nadine Chaval, daughter	0.0	0.0	NaN	23rd of September Communist League	Government (Diplomatic
2	1970	1	0	Philippines	Tarlac	Southeast Asia	120.599741	15.478598	Assassination	Employee	1.0	0.0	NaN	Unknown	Journalists & Media
3	1970	1	0	Greece	Attica	Western Europe	23.762728	37.997490	Bombing/Explosion	U.S. Embassy	NaN	NaN	NaN	Unknown	Government (Diplomatic
4	1970	1	0	Japan	Fukouka	East Asia	130.396361	33.580412	Facility/Infrastructure Attack	U.S. Consulate	NaN	NaN	NaN	Unknown	Government (Diplomatic

Displaying the shape and info of the data

In [27]:
```
df.shape
```

Out[27]: (181691, 17)

In [28]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181691 entries, 0 to 181690
Data columns (total 17 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Year         181691 non-null  int64
 1   Month        181691 non-null  int64
 2   Day          181691 non-null  int64
```

```
 3   Country      181691 non-null  object
 4   State        181270 non-null  object
 5   Region       181691 non-null  object
 6   Longitude    177134 non-null  float64
 7   Latitude     177135 non-null  float64
 8   AttackType   181691 non-null  object
 9   Target       181055 non-null  object
 10  Killed       171378 non-null  float64
 11  Wounded      165380 non-null  float64
 12  Summary      115562 non-null  object
 13  Group        181691 non-null  object
 14  Target_type  181691 non-null  object
 15  Weapon_type  181691 non-null  object
 16  Motive       50561 non-null   object
dtypes: float64(4), int64(3), object(10)
memory usage: 23.6+ MB
```

Checking for the duplicate values

In [29]:
```
duplicate_values = data.duplicated()
duplicate_values.value_counts()
```

Out[29]:
```
False    181691
dtype: int64
```

In [30]:
```
df.drop_duplicates(subset =None, keep='first', inplace = True)
df
```

Finding out the number of unique values

In [31]:
```
df.nunique()
```

Out[31]:
```
Year            47
Month           13
Day             32
Country        205
State         2855
Region          12
Longitude    48039
Latitude     48322
```

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 5 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

Run Code

```
Longitude       48039
Latitude        48322
AttackType          9
Target          86006
Killed            205
Wounded           238
Summary        112492
Group            3537
Target_type        22
Weapon_type        12
Motive          14490
dtype: int64
```

Checking for null values

In [40]:
```
df.isnull().sum()
```

Out[40]:
```
Year               0
Month              0
Day                0
Country            0
State            421
Region             0
Longitude       3977
Latitude        3976
AttackType         0
Target           544
Killed          9412
Wounded        15246
Summary        59038
Group              0
Target_type        0
Weapon_type        0
Motive        122224
dtype: int64
```

In [41]:
```
df['Killed'].fillna(0, inplace = True)
df['Wounded'].fillna(0, inplace = True)
```

In [42]:
```
df.isnull().sum()
```

Out[42]:
```
Year               0
Month              0
```

```
Day                0
Country            0
State            421
Region             0
Longitude       3977
Latitude        3976
AttackType         0
Target           544
Killed             0
Wounded            0
Summary        59038
Group              0
Target_type        0
Weapon_type        0
Motive        122224
dtype: int64
```

Insights Found

In [45]:
```
print("Country with the most attacks:",df['Country'].value_counts().idxmax())
print("City with the most attacks:",df['State'].value_counts().idxmax())
print("Region with the most attacks:",df['Region'].value_counts().idxmax())
print("Year with the most attacks:",df['Year'].value_counts().idxmax())
print("Month with the most attacks:",df['Month'].value_counts().idxmax())
print("Group with the most attacks:",df['Group'].value_counts().index[1])
print("Most Attack Types:",df['AttackType'].value_counts().idxmax())
```

```
Country with the most attacks: Iraq
City with the most attacks: Baghdad
Region with the most attacks: Middle East & North Africa
Year with the most attacks: 2014
Month with the most attacks: 5
Group with the most attacks: Taliban
Most Attack Types: Bombing/Explosion
```

Description of the data

In [46]:
```
df.describe(include = 'all')
```

Out[46]:

	Year	Month	Day	Country	State	Region	Longitude	Latitude	AttackType	Target	Killed	
count	172129.000000	172129.000000	172129.000000	172129	171708	172129	1.681520e+05	168153.000000	172129	171585	172129.000000	172

Description of the data

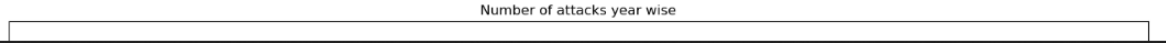
In [46]:
```
df.describe(include = 'all')
```

Out[46]:

	Year	Month	Day	Country	State	Region	Longitude	Latitude	AttackType	Target	Killed	
count	172129.000000	172129.000000	172129.000000	172129	171708	172129	1.681520e+05	168153.000000	172129	171585	172129.000000	172
unique	NaN	NaN	NaN	205	2855	12	NaN	NaN	9	86006	NaN	
top	NaN	NaN	NaN	Iraq	Baghdad	Middle East & North Africa	NaN	NaN	Bombing/Explosion	Civilians	NaN	
freq	NaN	NaN	NaN	23456	7625	48714	NaN	NaN	82126	6369	NaN	
mean	2003.056051	6.470060	15.518431	NaN	NaN	NaN	-4.830590e+02	23.779955	NaN	NaN	2.371553	
std	13.189553	3.391747	8.814150	NaN	NaN	NaN	2.101771e+05	18.295387	NaN	NaN	11.512104	
min	1970.000000	0.000000	0.000000	NaN	NaN	NaN	-8.618590e+07	-53.154613	NaN	NaN	0.000000	
25%	1991.000000	4.000000	8.000000	NaN	NaN	NaN	7.417291e+00	11.840929	NaN	NaN	0.000000	
50%	2009.000000	6.000000	15.000000	NaN	NaN	NaN	4.352619e+01	31.528200	NaN	NaN	0.000000	
75%	2014.000000	9.000000	23.000000	NaN	NaN	NaN	6.896761e+01	34.666667	NaN	NaN	2.000000	
max	2017.000000	12.000000	31.000000	NaN	NaN	NaN	1.793667e+02	74.633553	NaN	NaN	1570.000000	8

Data Visualisation

In [47]:
```
x_year = df['Year'].unique()
y_count_years = df['Year'].value_counts(dropna = False).sort_index()
plt.figure(figsize = (18,8))
sns.barplot(x = x_year, y = y_count_years, palette = 'rocket_r')
plt.xticks(rotation = 45)
plt.xlabel('Attack Year')
plt.ylabel('Number of Attacks each year')
plt.title('Number of attacks year wise')
plt.show()
```



LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 6 minutes ago (autosaved)

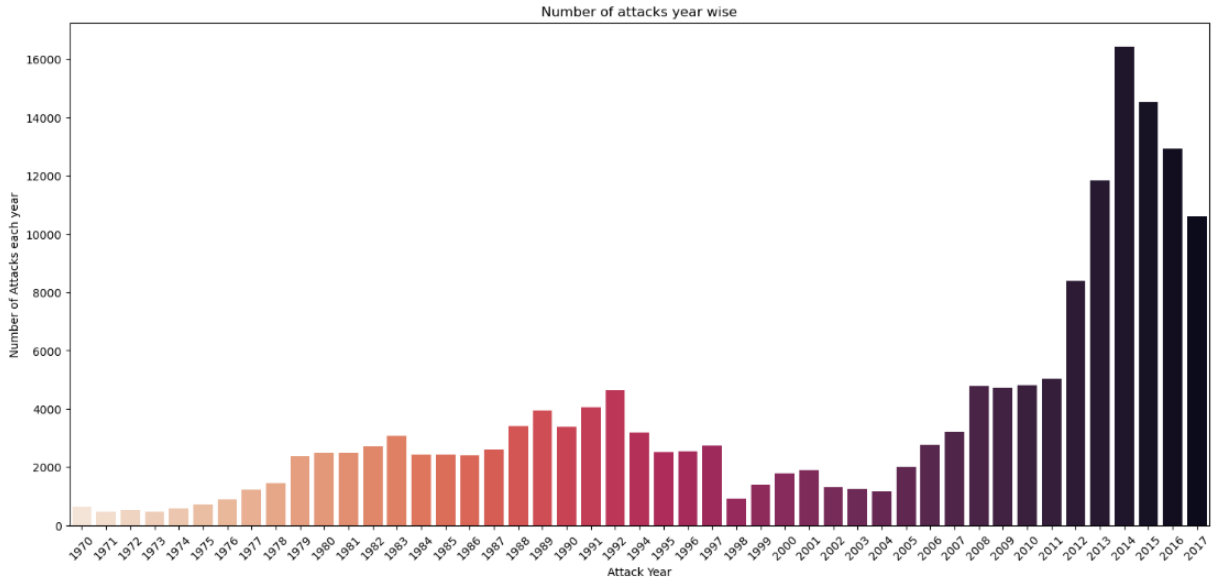
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [47]:
```
x_year = df['Year'].unique()
y_count_years = df['Year'].value_counts(dropna = False).sort_index()
plt.figure(figsize = (18,8))
sns.barplot(x = x_year, y = y_count_years, palette = 'rocket_r')
plt.xticks(rotation = 45)
plt.xlabel('Attack Year')
plt.ylabel('Number of Attacks each year')
plt.title('Number of attacks year wise')
plt.show()
```



Observations:

Maximum number of attacks took place in the year 2014 and The number of attacks have been gradually decreasing since 2014

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 6 minutes ago (autosaved)

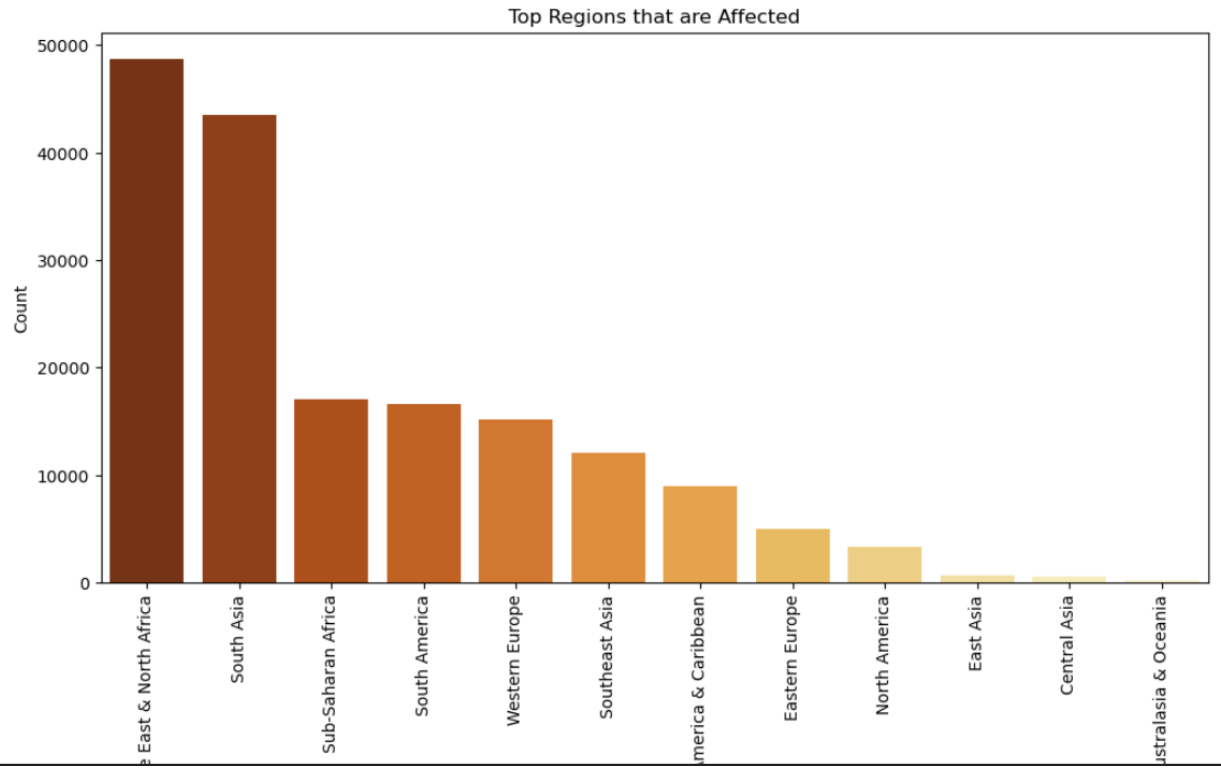
Logout

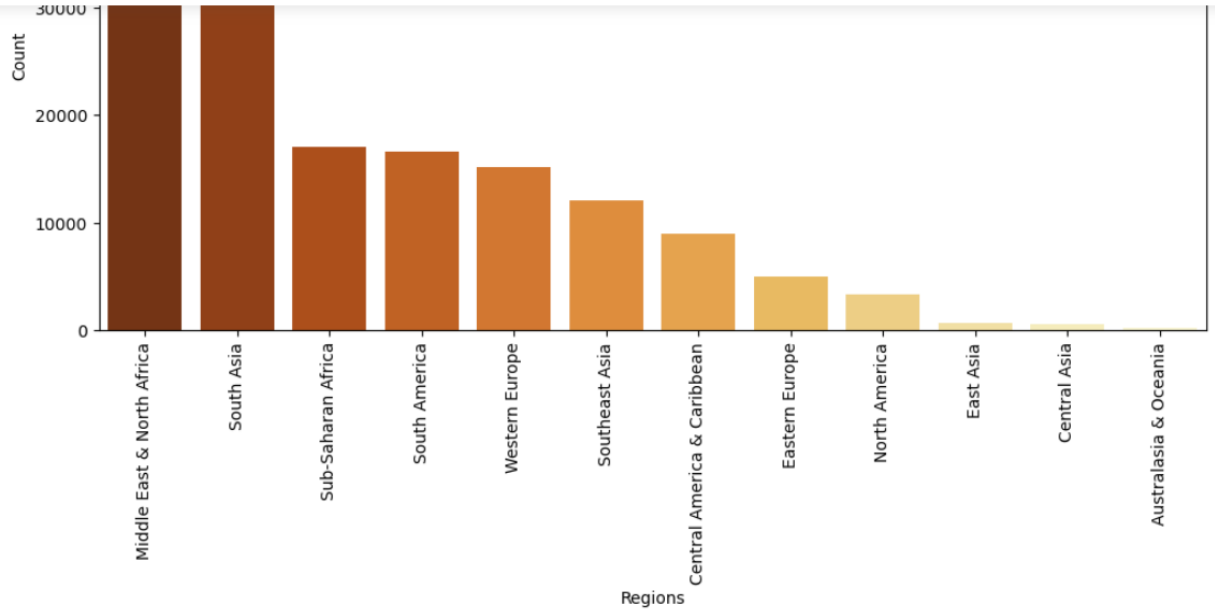
File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

Run Code

In [49]:
```
plt.subplots(figsize=(12,6))
sns.barplot(df['Region'].value_counts().index,df['Region'].value_counts().values,palette='YlOrBr_r')
plt.title('Top Regions that are Affected')
plt.xlabel('Regions')
plt.ylabel('Count')
plt.xticks(rotation= 90)
plt.show()
```





Observations:

Middle East & North America is the most affected region. The most peaceful region is Australia & Oceania with minimum number of terrorist attacks.

In [50]:
```
plt.figure(figsize=(12,6))
sns.barplot(x=df['AttackType'].value_counts().index,y = df['AttackType'].value_counts().values, palette = 'Greens_r')
plt.title('Most Common Attack Type')
plt.xlabel('Attack Type')
plt.xticks(rotation= 90)
plt.ylabel('Count')
plt.show()
```

Most Common Attack Type

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 6 minutes ago (autosaved)

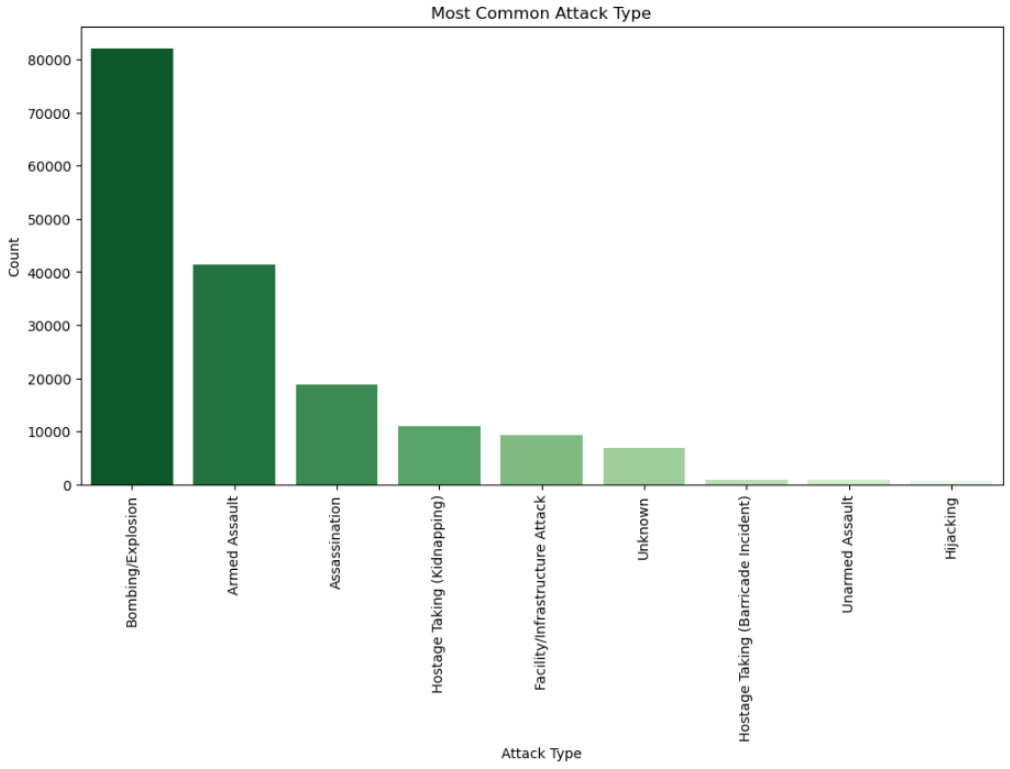
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [50]:
```
plt.figure(figsize=(12,6))
sns.barplot(x=df['AttackType'].value_counts().index,y = df['AttackType'].value_counts().values, palette = 'Greens_r')
plt.title('Most Common Attack Type')
plt.xlabel('Attack Type')
plt.xticks(rotation= 90)
plt.ylabel('Count')
plt.show()
```

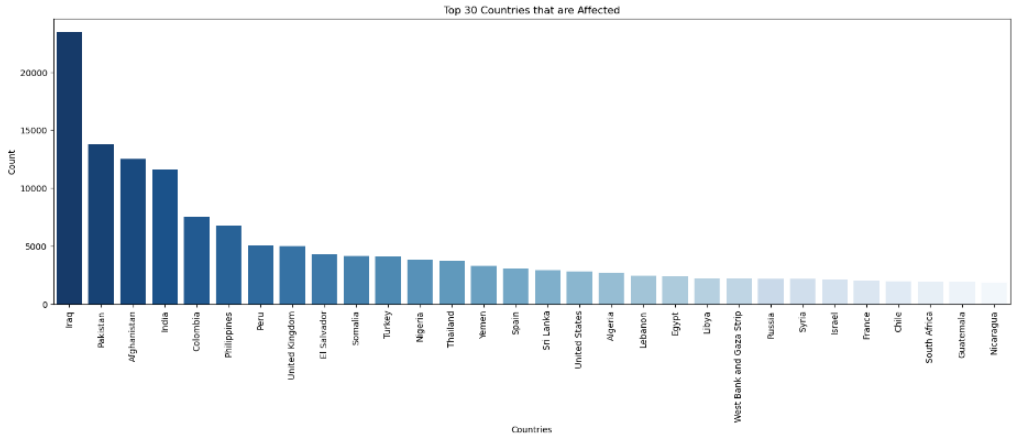


Observations:

Observations:

The most common attack type is Bombing/Explosion. The least common is hijacking.

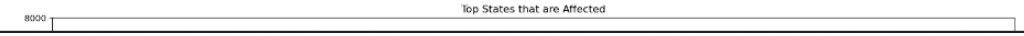
In [51]:
```
plt.subplots(figsize=(20,6))
sns.barplot(df['Country'].value_counts()[:30].index,df['Country'].value_counts()[:30].values,palette='Blues_r')
plt.title('Top 30 Countries that are Affected')
plt.xlabel('Countries')
plt.ylabel('Count')
plt.xticks(rotation= 90)
plt.show()
```



Observations:

The country that is most affected by these attacks is Iraq, followed by Pakistan and Afghanistan.

In [52]:
```
plt.subplots(figsize=(20,6))
sns.barplot(df['State'].value_counts()[:25].index,df['State'].value_counts()[:25].values,palette='viridis')
plt.title('Top States that are Affected')
plt.xlabel('States')
plt.ylabel('Count')
plt.xticks(rotation= 90)
plt.show()
```



LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

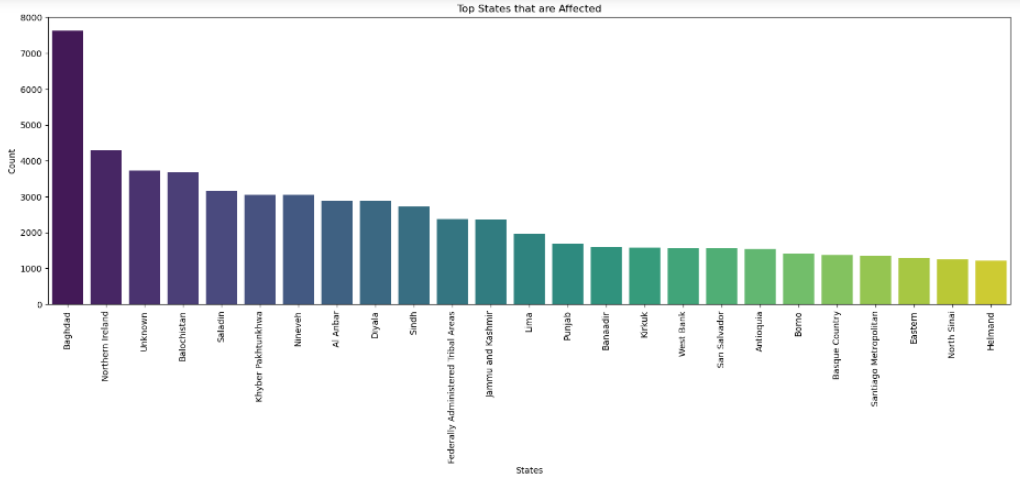
jupyter LGMVIP Task 4 Last Checkpoint: 6 minutes ago (autosaved)

Logout

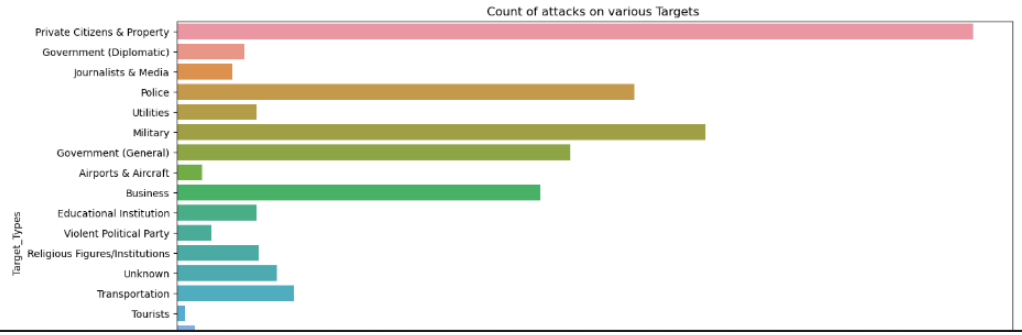
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code



In [53]:
```
plt.figure(figsize=(15,8))
sns.countplot(y=df['Target_type'])
plt.title('Count of attacks on various Targets ')
plt.xlabel('Counts')
plt.ylabel('Target_Types')
plt.show()
```



LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 7 minutes ago (autosaved)

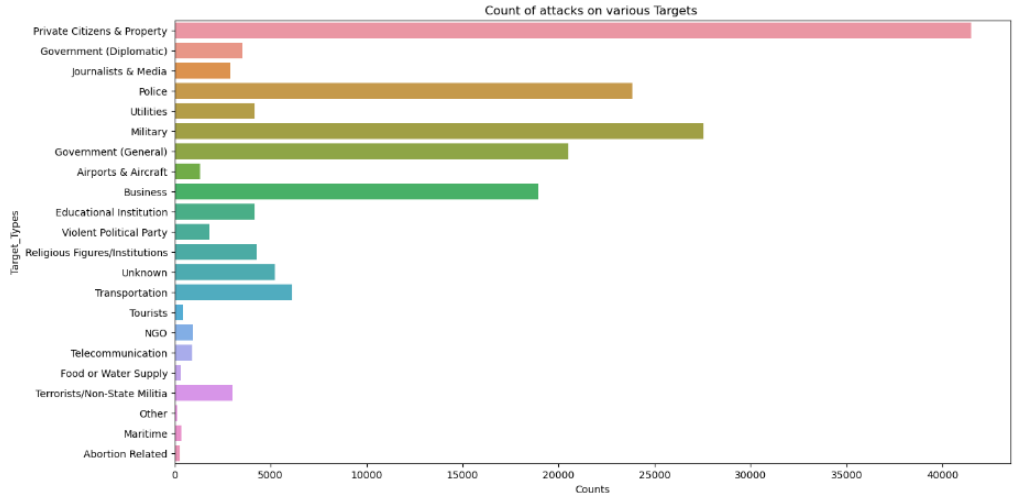
Logout

File Edit View Insert Cell Kernel Widgets Help

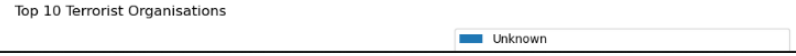
Trusted | Python 3 (ipykernel)

Run Code

In [53]:
```
plt.figure(figsize=(15,8))
sns.countplot(y=df['Target_type'])
plt.title('Count of attacks on various Targets ')
plt.xlabel('Counts')
plt.ylabel('Target_Types')
plt.show()
```



In [54]:
```
plt.figure(figsize = (10, 10))
plt.pie(df['Group'].value_counts()[:10].values,labels=df['Group'].value_counts()[:10].index,autopct='%1.2f%%'  )
plt.title('Top 10 Terrorist Organisations')
plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.show()
```



LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 7 minutes ago (autosaved)

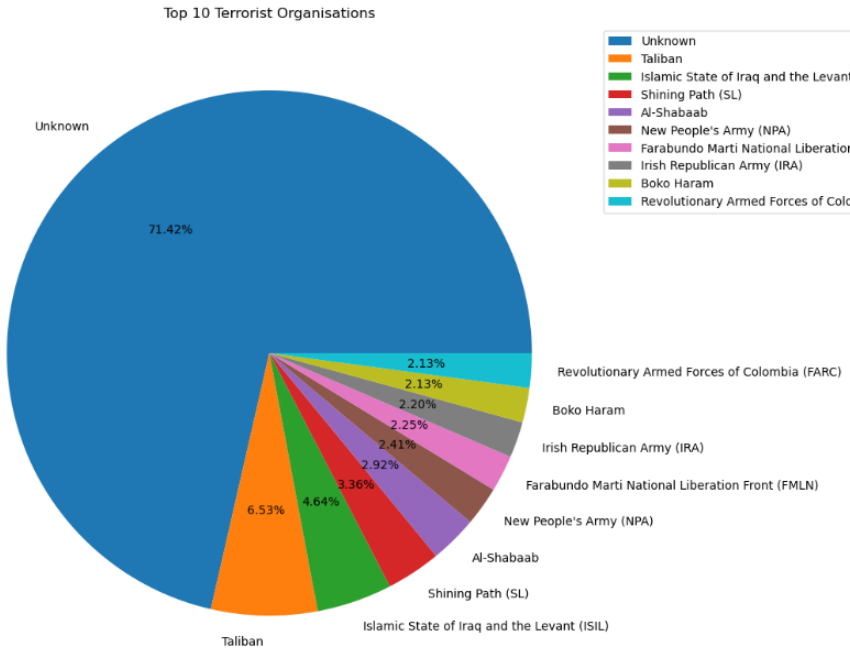
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [54]:
```
plt.figure(figsize = (10, 10))
plt.pie(df['Group'].value_counts()[:10].values,labels=df['Group'].value_counts()[:10].index,autopct='%1.2f%%'  )
plt.title('Top 10 Terrorist Organisations')
plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.show()
```



In [55]:
```
Casualities = df.groupby(['Country'], dropna = False)['Killed','Wounded'].sum()
Casualities['Casualities'] = Casualities.Killed + Casualities.Wounded
Casualities_sorted = Casualities.sort_values('Casualities', ascending = False)
Casualities_sorted['Country'] = Casualities_sorted.index
```

In [56]:
```
fig=plt.figure(figsize=(20,7))
sns.barplot(x = Casualities_sorted['Country'][:20], y = Casualities_sorted['Casualities'][:20], palette='viridis')
plt.xlabel("Countries ")
```

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 7 minutes ago (autosaved)

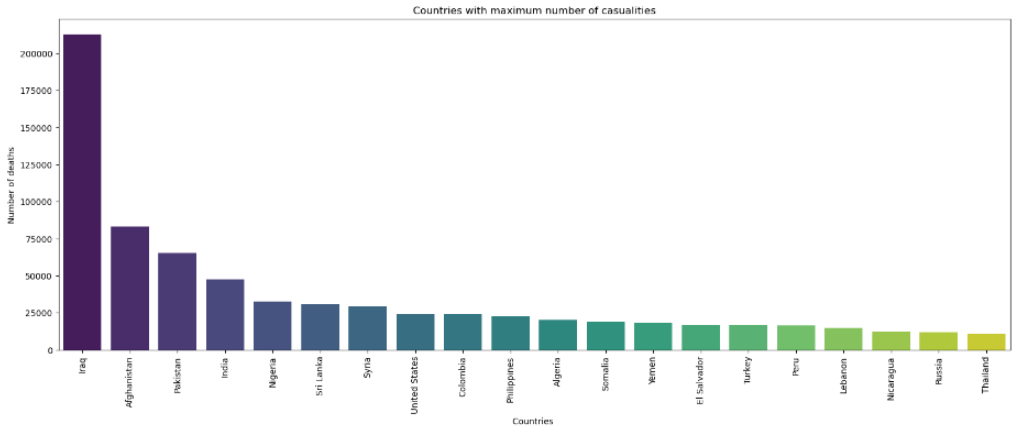
Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [55]:
```
Casualities = df.groupby(['Country'], dropna = False)['Killed','Wounded'].sum()
Casualities['Casualities'] = Casualities.Killed + Casualities.Wounded
Casualities_sorted = Casualities.sort_values('Casualities', ascending = False)
Casualities_sorted['Country'] = Casualities_sorted.index
```

In [56]:
```
fig=plt.figure(figsize=(20,7))
sns.barplot(x = Casualities_sorted['Country'][:20], y = Casualities_sorted['Casualities'][:20], palette='viridis')
plt.xlabel("Countries ")
plt.ylabel("Number of deaths")
plt.xticks(rotation = 90)
plt.title("Countries with maximum number of casualities")
plt.show()
```



In [57]:
```
Death_each_year = df.groupby(['Year'], dropna = False)['Killed'].sum()
fig=plt.figure(figsize=(20,7))
sns.set_style("darkgrid")
Death_each_year.plot(kind = 'bar', colormap = "Set2")
plt.xlabel("Years", fontsize = 12)
plt.ylabel("Number of deaths",fontsize = 12)
plt.xticks(rotation = 90)
plt.title("Number of Deaths each year")
plt.show()
```

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 7 minutes ago (autosaved)

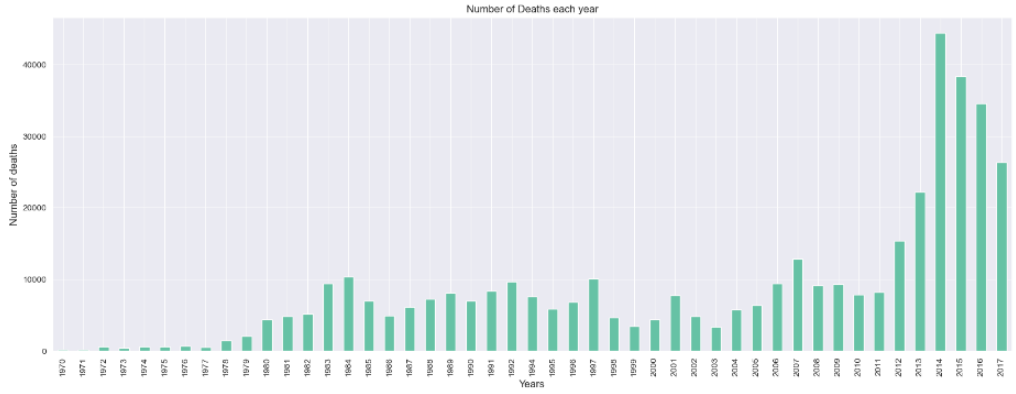
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [57]:
```
Death_each_year = df.groupby(['Year'], dropna = False)['Killed'].sum()
fig=plt.figure(figsize=(20,7))
sns.set_style("darkgrid")
Death_each_year.plot(kind = 'bar', colormap = "Set2")
plt.xlabel("Years", fontsize = 12)
plt.ylabel("Number of deaths",fontsize = 12)
plt.xticks(rotation = 90)
plt.title("Number of Deaths each year")
plt.show()
```



Observations:

The maximum number of casualties were in the year of 2014 aftr whic hthe deATHS have increased significantly

In [59]:
```
Attacks_2014 = df[df.Year == 2014]
fig=plt.figure(figsize=(17,7))
sns.barplot(x = Attacks_2014['Country'].value_counts()[:20].index, y = Attacks_2014['Country'].value_counts()[:20].values, palett
plt.xlabel("Countries ")
plt.ylabel("Number of attacks")
plt.xticks(rotation = 90)
plt.title("Most Attacked Countries in 2014")
plt.show()
```

LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

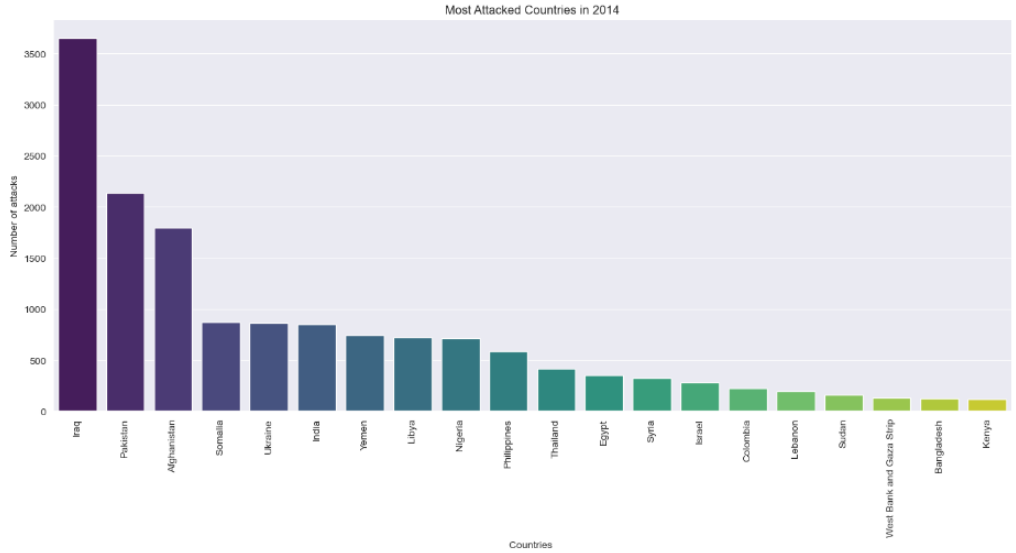
jupyter LGMVIP Task 4 Last Checkpoint: 7 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

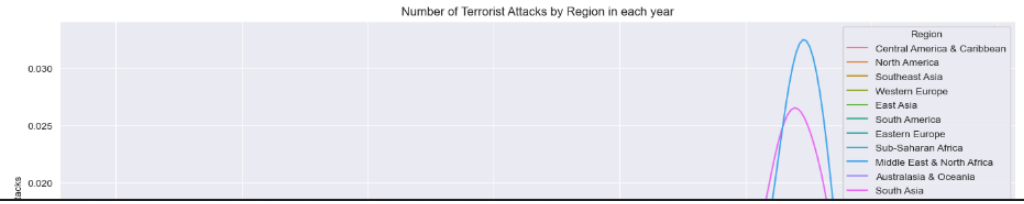
Trusted | Python 3 (ipykernel)

Run Code



In [61]:
```
fig=plt.figure(figsize=(17,7))
sns.set_style("darkgrid")
sns.kdeplot(df['Year'], hue = df['Region'])
plt.xlabel("Years")
plt.ylabel("Number of attacks")
plt.title("Number of Terrorist Attacks by Region in each year")
```

Out[61]: Text(0.5, 1.0, 'Number of Terrorist Attacks by Region in each year')



LGMVIP Task 4 - Jupyter Notebo

localhost:8888/notebooks/LGMVIP%20Task%204.ipynb

jupyter LGMVIP Task 4 Last Checkpoint: 7 minutes ago (autosaved)

Logout

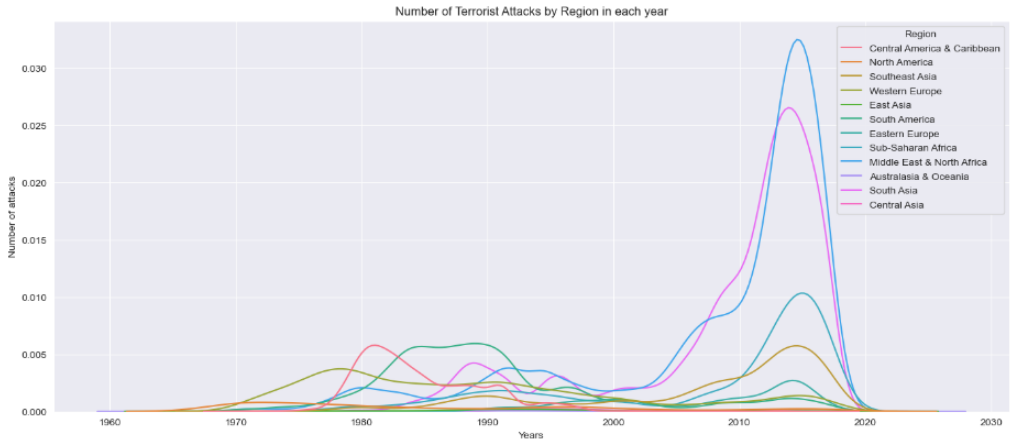
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [61]:
```
fig=plt.figure(figsize=(17,7))
sns.set_style("darkgrid")
sns.kdeplot(df['Year'], hue = df['Region'])
plt.xlabel("Years")
plt.ylabel("Number of attacks")
plt.title("Number of Terrorist Attacks by Region in each year")
```

Out[61]: Text(0.5, 1.0, 'Number of Terrorist Attacks by Region in each year')



CONCLUSION/RESULTS

```
The insights derived from the Data Analysis on Global Terrorism are:
1.Taliban is the most active terrorist organisation after 2012.

2.There are 205 Countries from 12 Regions covered in the dataset terrorist attack data in 3 years from 1970 to 2017. Overall
181691 terrorist attacks are recorded here which caused about 935737.0 Casualities Consisted of 411868.0 Kills and 523869.0
Wounded.

3.The most peaceful region is Australia & Oceania with minimum number of terrorist attacks.
```

In [ ]: