

According To Analysis The GTD we found that

- Afghanistan in 2015 was the highest country in by attack counts 1928
- 2014 was the highest number of killed by 44490
- 1971 was the lowest number of killed by 173
- Western Europe was the highest region in attack Type and Australasia & Oceania was the lowest region
- Highest target nationality is from Iraq
- 2014 highest number of killed persons and 1971 was the lowest
- Explosive weapon Type was the most weapon Type in Number of Killed
- bombing/explosion is the Highest Attack Type by number of killed 88255
- Number of Successful operations is 161245 , Failed operations is 20025

Numbers of Attack Through Years And Countries :

Jupyter

Last Checkpoint: 2 days ago

File

Edit

View

Run

Kernel

Settings

Help

Python 3 (ipykernel)

Trusted

205 rows × 3 columns

dtype: float64

```
[113]: grouped_data=data.groupby(['country','Year']).size().reset_index(name='attack_count')

[115]: maxyear=grouped_data.loc[grouped_data.groupby('country')['attack_count'].idxmax()]
maxyear

[115]:
```

	country	Year	attack_count
29	Afghanistan	2015	1928
38	Albania	1997	41
60	Algeria	1997	344
81	Andorra	1974	1
94	Angola	1990	205
...
3681	Yemen	2014	763
3700	Yugoslavia	1998	46
3714	Zaire	1996	13
3727	Zambia	1989	18
3735	Zimbabwe	1980	14

205 rows × 3 columns

```
[79]: grouped_data = data.groupby('Year')['killed'].sum().reset_index()
```

Year that Have Maximum and Minimum Number of Attack

Jupyter DataPreparation Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

205 rows x 3 columns

```
[79]: grouped_data = data.groupby('Year')['killed'].sum().reset_index()
year_with_highest_killed = grouped_data.loc[grouped_data['killed'].idxmax()]
print("Year with highest number of killed:")
print(year_with_highest_killed)

Year with highest number of killed:
Year      2014.0
Killed    44490.0
Name: 43, dtype: float64

[119]: year_with_lowest_killed = grouped_data.loc[grouped_data['killed'].idxmin()]
print("\nYear with lowest number of killed:")
print(year_with_lowest_killed)

Year with lowest number of killed:
Year      1971.0
Killed     173.0
Name: 1, dtype: float64
```

Attacks Numbers Through Regions

Jupyter DataPreparation Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
Killed      173.0
Name: 1, dtype: float64

[81]: attackcountbyregion=data.groupby(['region']).size().reset_index(name='attack_count')
attackcountbyregion
```

	region	attack_count
0	Australasia & Oceania	282
1	Central America & Caribbean	10344
2	Central Asia	563
3	East Asia	802
4	Eastern Europe	5144
5	Middle East & North Africa	50474
6	North America	3456
7	South America	18978
8	South Asia	44974
9	Southeast Asia	12485
10	Sub-Saharan Africa	17550
11	Western Europe	16639

Most Nationalities that have been Killed

Jupyter DataPreparation Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

```
[83]: Highest_Targetnationality = data['Target_Nationality'].value_counts().reset_index()
Highest_Targetnationality
```

```
[83]:
```

	Target_Nationality	count
0	Iraq	24113
1	Pakistan	13900
2	India	12098
3	Afghanistan	10931
4	Colombia	7922
...
210	People's Republic of the Congo	1
211	Saba (Netherlands Antilles)	1
212	Tuvalu	1
213	Western Sahara	1
214	Marshall Islands	1

215 rows x 2 columns

Number of Killed Through Years

Killed Number Through Years

```
[86]: grouped_data=data.groupby(['Year']).size().reset_index(name='Killed')
```

```
[88]: Year_with_high_victims=grouped_data.loc[grouped_data.groupby('Year')['Killed'].idxmax()]
```

```
[90]: KilledNumbers = Year_with_high_victims.sort_values(by='Killed', ascending=False)
KilledNumbers
```

```
[90]:
```

	Year	Killed
43	2014	16903
44	2015	14965
45	2016	13587
42	2013	12036
46	2017	10900
41	2012	8522
40	2011	5076
22	1992	5071
39	2010	4826
37	2008	4805
38	2009	4721
21	1991	4683

Number of Attacks According To Weapon Type

```
[92]: number_of_attack_according_to_weapontype = data.groupby('Weapon_Type').size().reset_index(name='Killed').sort_values(by='Killed',ascending=False)
number_of_attack_according_to_weapontype
```

```
[92]:
```

	Weapon_Type	Killed
9851	Explosive	20925
8785	Automatic firearm	12419
13431	Pistol	4472
11736	Incendiary	4153
10347	Firearm	4053
...
6828	An explosive device weighing 15 kilograms and ...	1
6829	An explosive device weighing 15 kilograms hidd...	1
6830	An explosive device weighing 15 kilograms plac...	1
6831	An explosive device weighing 15 kilograms that...	1
19147	wrenches	1

19148 rows x 2 columns

Numbers of Attacks According To Attack Types

Jupyter DataPreparation Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

JupyterLab Python 3 (ipykernel)

```
[94]: number_of_attack_according_to_attacktype = data.groupby('Attack_Type').size().reset_index(name='Killed').sort_values(by='Killed',ascending=False)
number_of_attack_according_to_attacktype
```

```
[94]:
```

	Attack_Type	Killed
2	bombing/explosion	88255
0	armed assault	42669
1	assassination	19312
6	hostage taking (kidnapping)	11158
3	facility/infrastructure attack	10356
8	unknown	7276
7	unarmed assault	1015
5	hostage taking (barricade incident)	991
4	hijacking	659

Number of Successful and Failed Operations

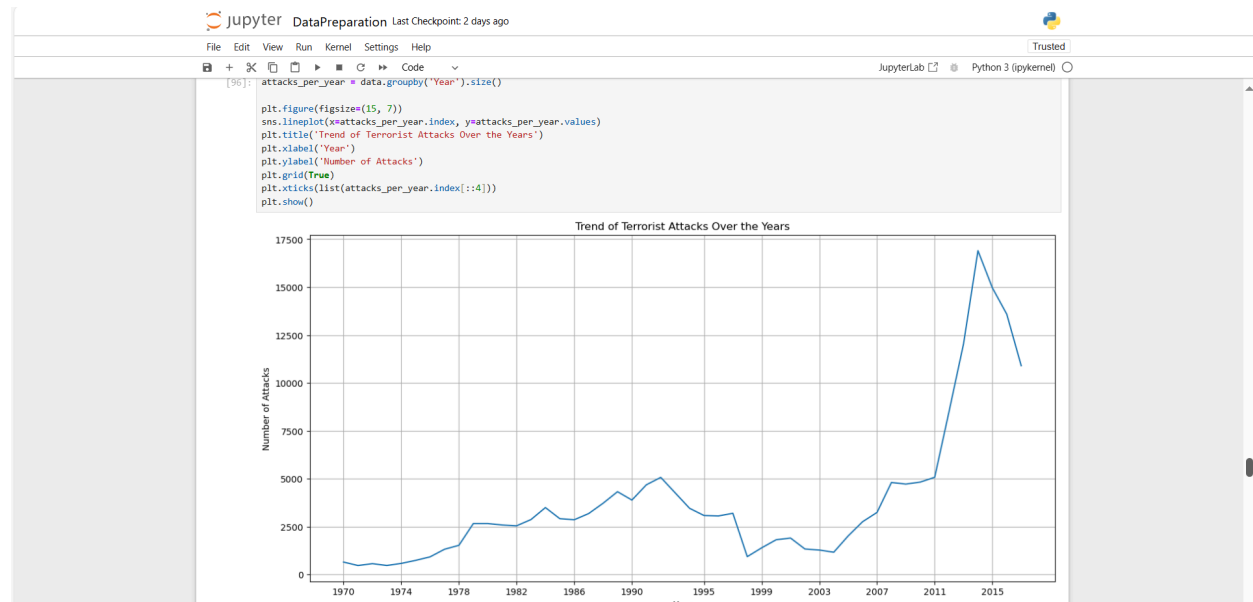
Number of operations failed and succsed

```
[111]: successful_and_failed_operations = data.groupby('successful_operation')['incident_place'].count().reset_index(name='operations')
successful_and_failed_operations
```

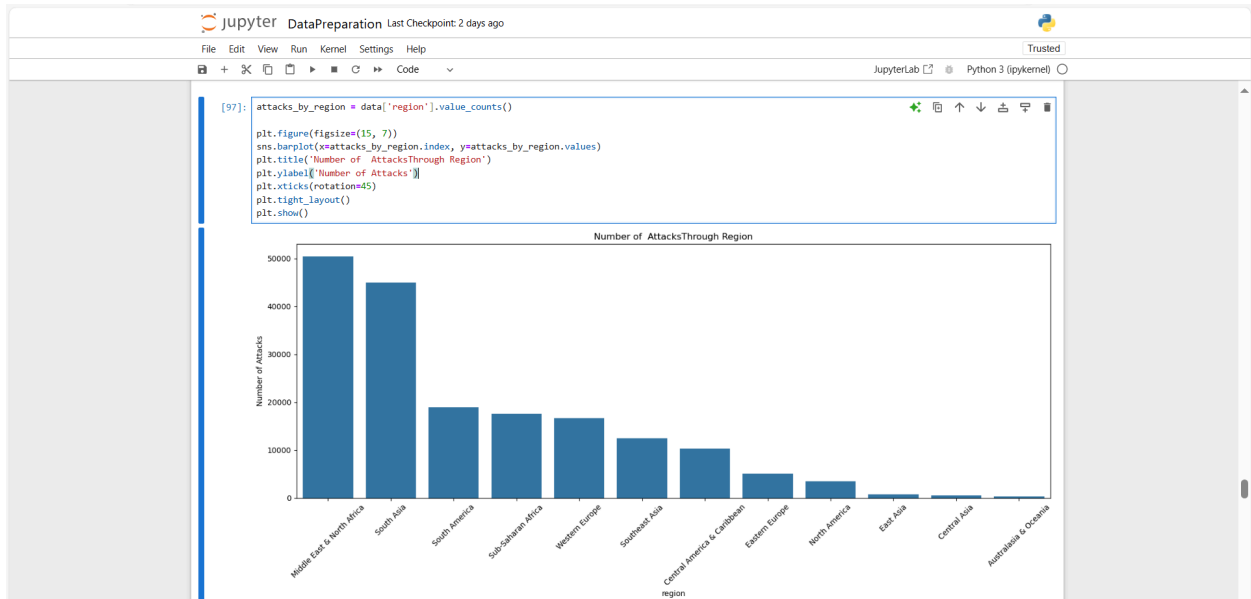
```
[111]:
```

	successful_operation	operations
0	0	20025
1	1	161245

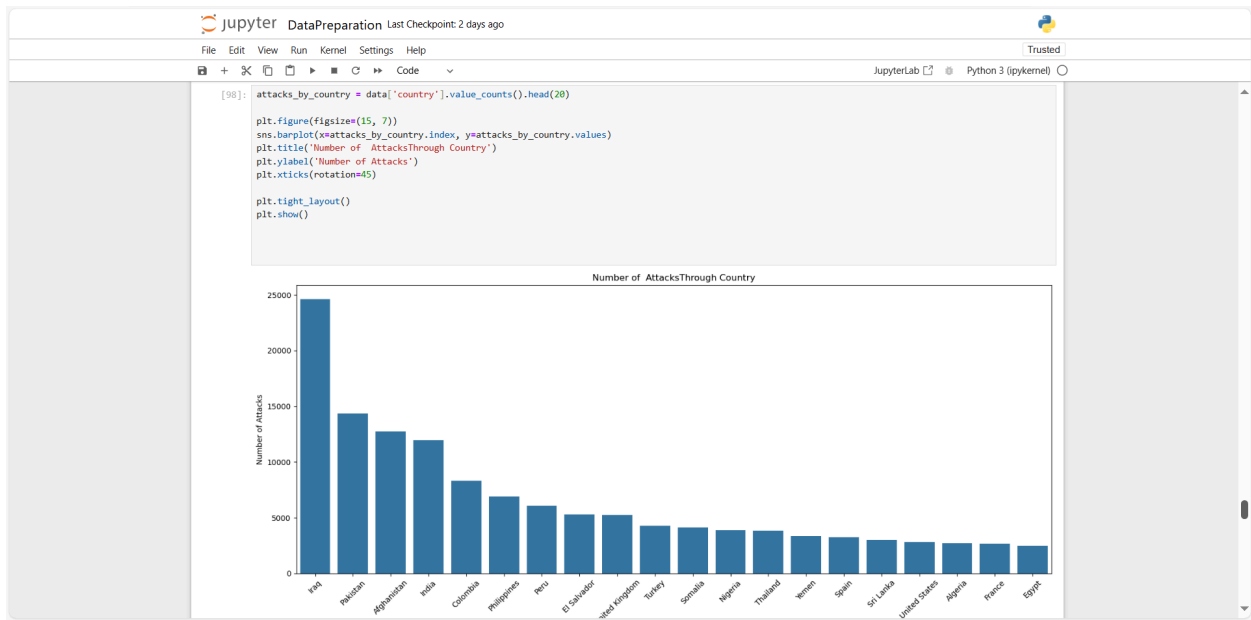
Line plot showing the Trend of Terrorist Attacks Over the Years



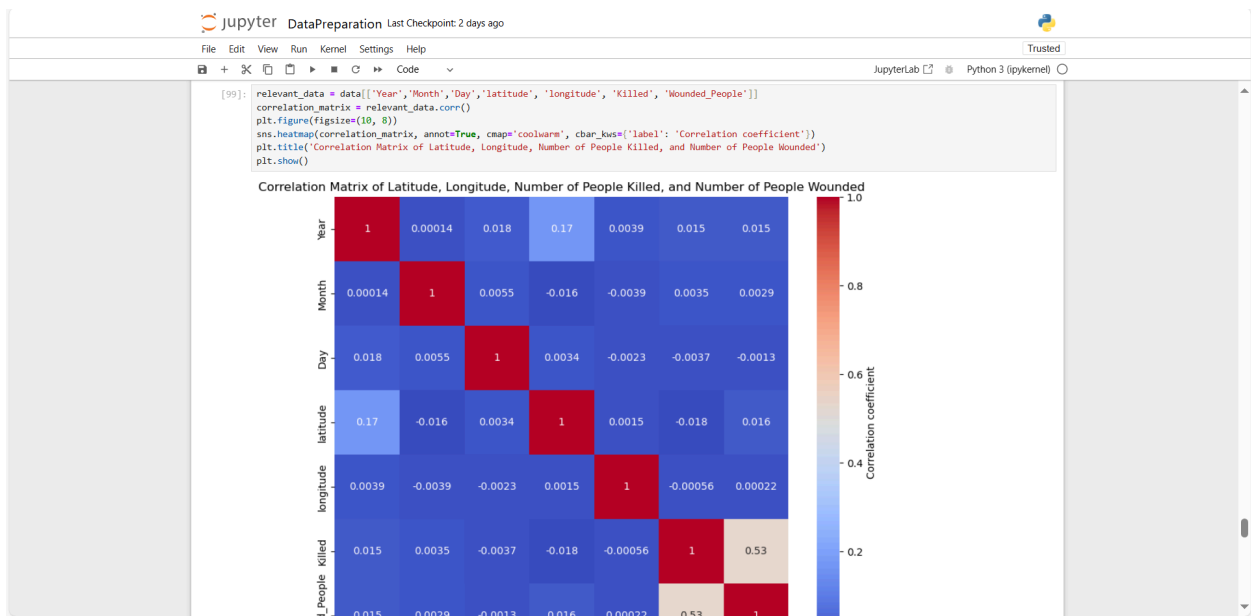
Bar plot showing the Number of AttacksThrough Region



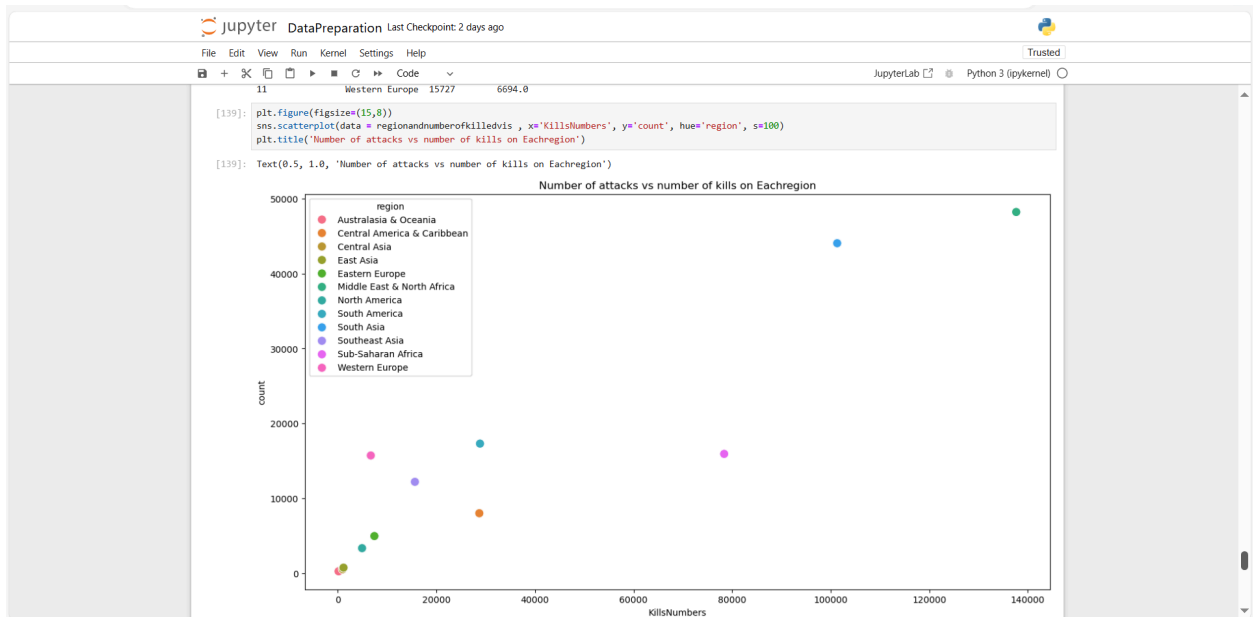
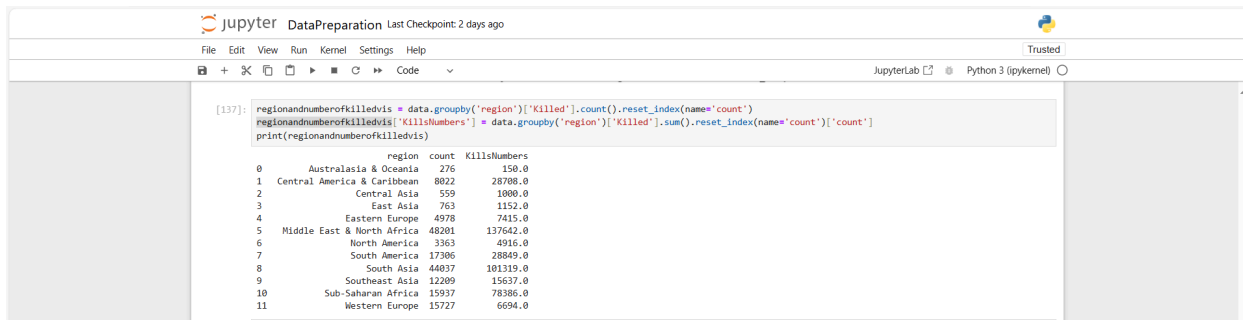
Bar plot showing Numbers of Attack through countries



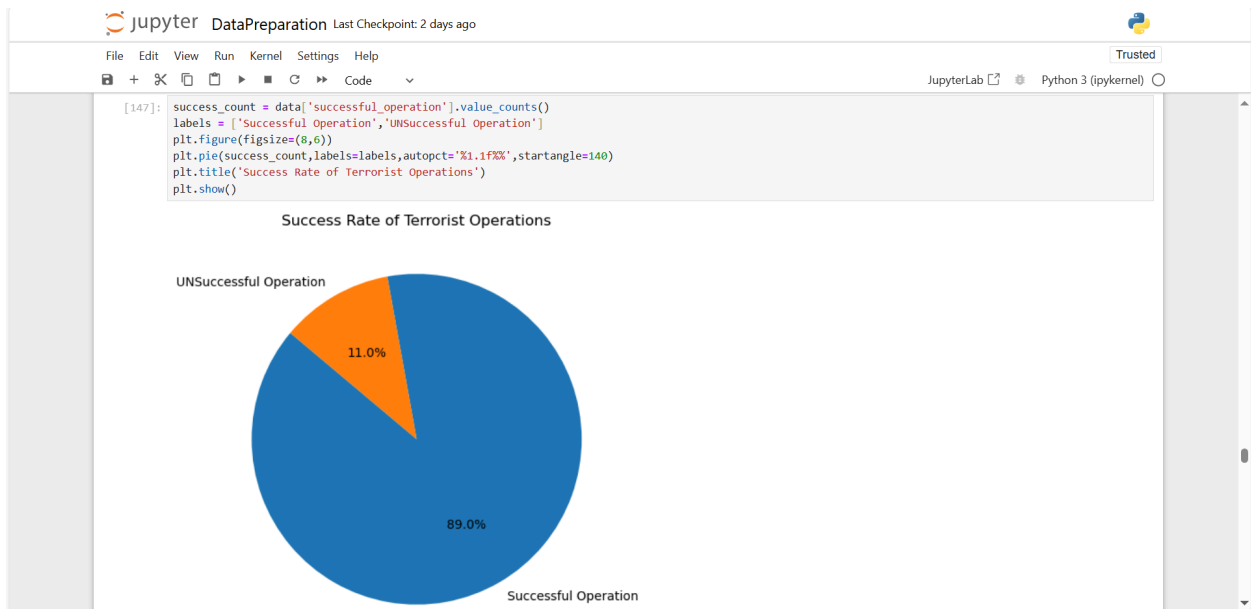
Correlation Matrix



ScatterPlot showing Numbers of killed persons through region



Pie Chart showing number of successful and unsuccessful operations



Scatterplot showing number of attacks and numbers by weapon type

