Visualizing Time Series Data

Terrorism Analysis and Finding Insights

Submitted by
Dipanwita Manna

Under the guidance of

Dr. Sylvester Fernandes,
Forsk Coding School

11th September 2020

Table Of Contents:

Introduction

This document reflects the collection and coding for the project's titled Terrorism Analysis and Finding Insights.
In recent years, analytical reporting has evolved into one of the world's most important business intelligence components, compelling companies to adapt their strategies based on powerful data-driven insights. So this project is totally based on Data Analysis and Data Visualization.
This web application is a tool to analyze and predict the incidents of terrorism across the world and India specifically. It uses a dataset of nearly **1,90,000 records** maintained by the "**National Consortium for the study of terrorism and responses to terrorism**" (START) from the year 1970 to 2018. As the Protective vehicles are very few with the Army and they are distributed uniformly across the area. Similarly, Explosive Detection Dogs (ED Dogs) are only less in the entire country. So this project can be used as a Predictive Analysis tool to find the trendline of each kind of Attack. This tool helps in finding the concentration of Attack type in the area which would help in the allocation of the resources. Visualizing the data gives clear patterns about the data and makes it easy for the analysis.
For Visual representations of terror attacks, maps, as well as charts, are used with the help the which the client can retrieve the data and use it as a predictive analysis tool. The goal was to be as transparent as possible regarding the data collection methodology, with a commitment to creating a highly comprehensive and consistent terrorism incident dataset of approximately 1,90,000 records.

For the interactive dashboard, a productive Python framework "Dash", along with a Python library "Plotly" has been used for frontend development.

Related Work

Time series data typically consist of a sequence of data taken at successive equally spaced time intervals. **Time series *analysis*** comprises methods for analyzing time-series data in order to extract meaningful statistics and other characteristics of the data. **Data visualization** is a crucial step and should be the first step before starting with time series modeling. It can be used as a model to predict future values based on previously observed values. It is important to know before modeling how the variables in the dataset are correlated, what the value ranges are, and how it is affected by the time period.

Steps are:

- Understanding clients requirements
- Analyzing the Data
- Filtering the Data
- Implementation of the requirements
- Visualizing the data by graphs maps and charts

My Work

Under the guidance and as per the suggestion of Forsk Coding School, I followed the **Agile Methodology**. Agile is an iterative approach for the completion of a project. Requirements, plans, and results were evaluated continuously by the team to have a natural mechanism for responding to change quickly.

Under this project, we were trained by Forsk Coding School and with they being PMC we carried out this project.

1. Firstly Filtering the data in accordance with clients' requirements. Under this, I was taught how to filter the data step by step.
2. Then by using the popular framework i.e Dash in Python, the web application is created and made the layout in it as per the requirement of the client.
3. Map representation in-dash and also about chart representation were used
4. Lastly, I applied minimal HTML and CSS for styling the website to make it look clean and good for the user.

You can able to see my work on the GitHub.com

Github Link: https://github.com/DipanwitaManna16/Terrorism_Analysis

My Project Demo link:
https://drive.google.com/file/d/1aBbE1GgG06glepWo0Ep7aRex22LUiji1/view?usp=sharing

The project is completely based on User Interface and Data Visualization techniques. After viewing this demo you will find that there are various things that I had done in my project as per the client requirement and meet all the requirements provided by them.

Implementation

For this project I used :

Pandas Library (for data analysis)

Dash Framework (for data visualization) :

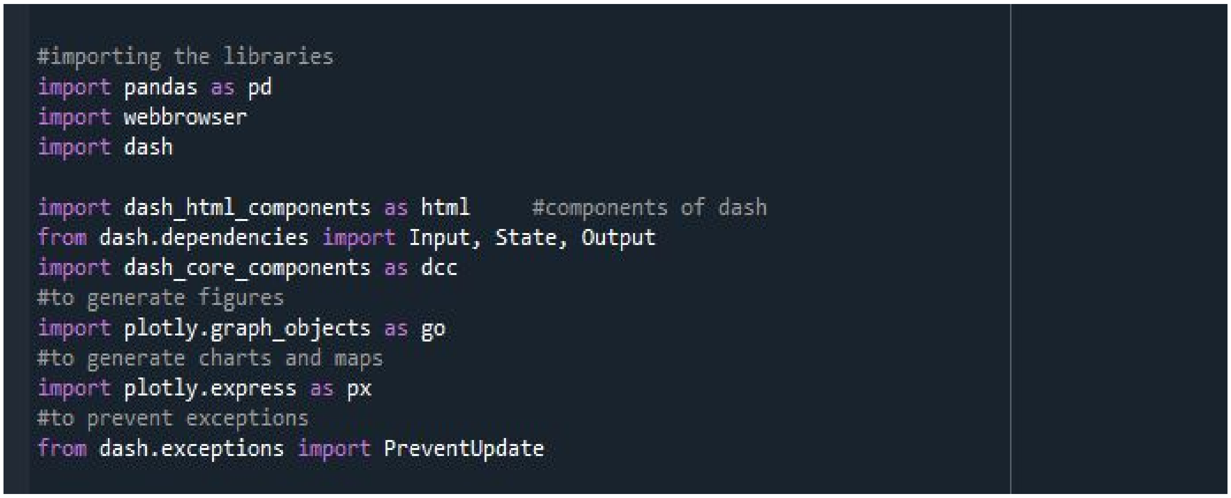
- Dash Html Components Library (enables us to use HTML & CSS in Python)
- Dash Core Component Library (allows us to build tabs, range slider, dropdown, etc.)
- Dash Dependencies Library (makes the data and the visualization interactive with the creation of callbacks)
- Dash Exceptions Handling.

Webbrowser Library

Plotly Libraries (Graphing Libraries)

- Plotly graph objects
- Plotly express

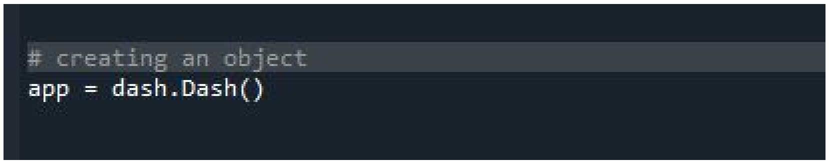
❑ **Importing all the libraries and packages**



```
#importing the libraries
import pandas as pd
import webbrowser
import dash

import dash_html_components as html     #components of dash
from dash.dependencies import Input, State, Output
import dash_core_components as dcc
#to generate figures
import plotly.graph_objects as go
#to generate charts and maps
import plotly.express as px
#to prevent exceptions
from dash.exceptions import PreventUpdate
```

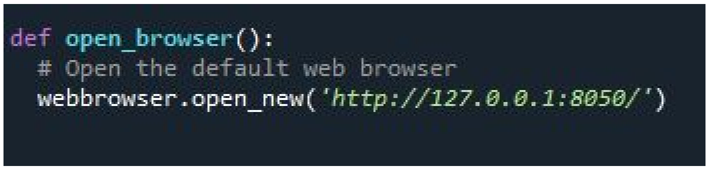
❑ **Creating a dash object**



```
# creating an object
app = dash.Dash()
```

❑ Loading the dataset and creating a month-list , date-list, region-list, country-list, state-list, city-list, and attack-type-list, all the variables being declared globally.

❑ **Opening a default web browser**

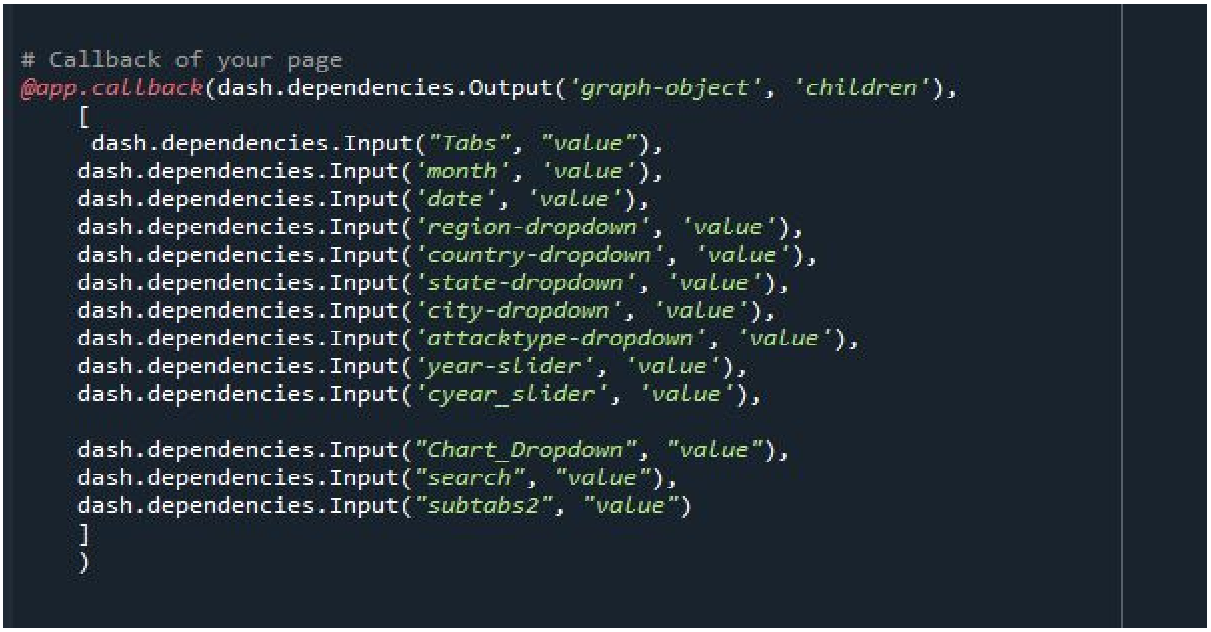


```
def open_browser():
  # Open the default web browser
  webbrowser.open_new('http://127.0.0.1:8050/')
```

❑ Creating a function for UI(user interface) which consists of HTML and dcc dependencies. This is the beauty of Dash that we don’t need external Html files all can be added in one page. Also added some CSS styling for the webpage.

❑ Tabs for Map-Tool and Chart-Tool, dropdowns for filtering the data, range-slider for selecting year range are added to the UI.

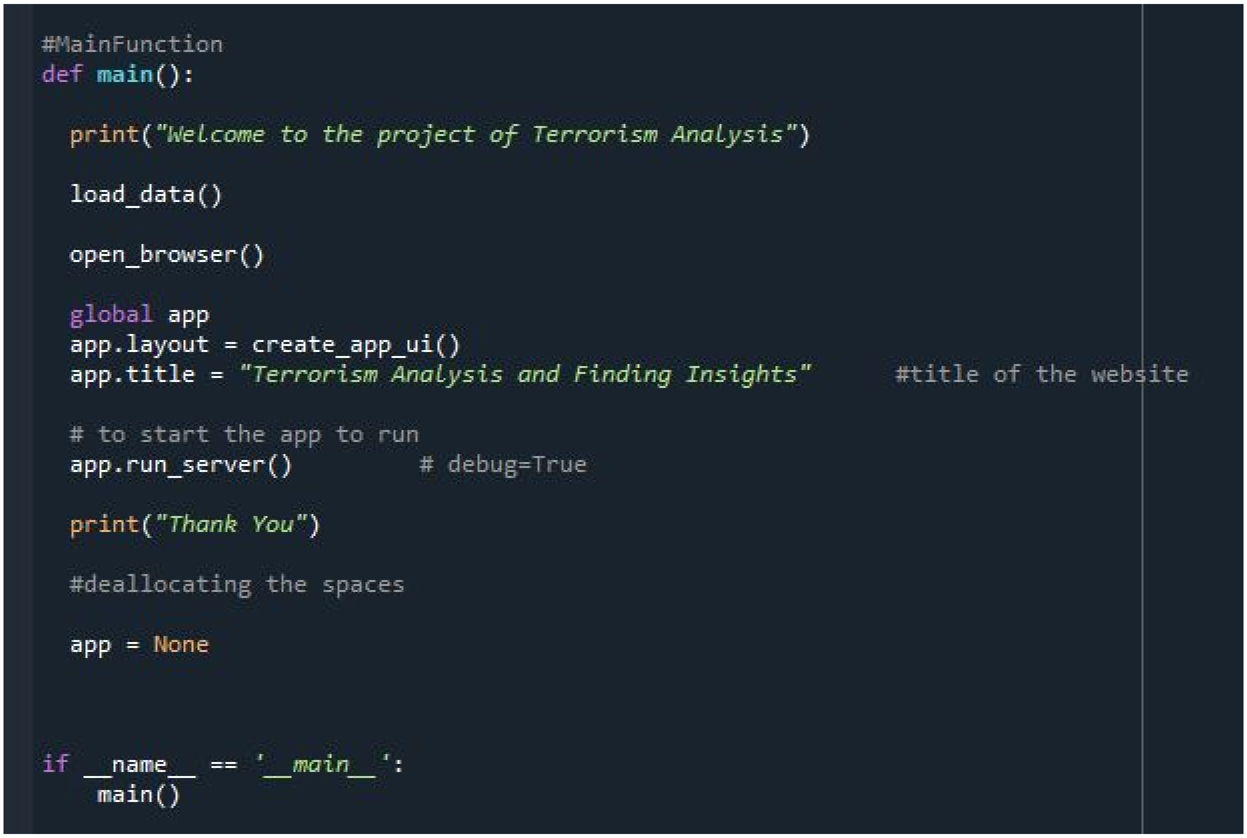
❑ To make the UI more responsive I have made as many callbacks as possible.



```
# Callback of your page
@app.callback(dash.dependencies.Output('graph-object', 'children'),
    [
     dash.dependencies.Input("Tabs", "value"),
    dash.dependencies.Input('month', 'value'),
    dash.dependencies.Input('date', 'value'),
    dash.dependencies.Input('region-dropdown', 'value'),
    dash.dependencies.Input('country-dropdown', 'value'),
    dash.dependencies.Input('state-dropdown', 'value'),
    dash.dependencies.Input('city-dropdown', 'value'),
    dash.dependencies.Input('attacktype-dropdown', 'value'),
    dash.dependencies.Input('year-slider', 'value'),
    dash.dependencies.Input('cyear_slider', 'value'),

    dash.dependencies.Input("Chart_Dropdown", "value"),
    dash.dependencies.Input("search", "value"),
    dash.dependencies.Input("subtabs2", "value")
    ]
    )
```

- Created Callbacks, a type of decorator for inter-relating the codes with the graphical representation. This concept was very much new to me and got to learn and implement in this live project. Whenever an input property changes, the function that the callback decorator wraps will get called automatically.
- We updated the graph according to the choice of the dropdown and made a connection between input data (the dropdown) and output data (the graph).
- As the client required to process the data separately for World and India on Map and Charts, I created two different sub-tabs for each tab i.e World and India. The India Map and India Chart have the data of the events that happened in India only.
- To know the year of the event happened. A Year Rangeslider is plotted in UI and linked it with the data, it will show the events that occurred on those years along with the places on the map.
- In the Chart section, an input box is created for knowing the type of event that took place in the particular year range and its count.
- Lastly, I have defined the main function for loading my data, creating my app layout, and to run my server.



```
#MainFunction
def main():

  print("Welcome to the project of Terrorism Analysis")

  load_data()

  open_browser()

  global app
  app.layout = create_app_ui()
  app.title = "Terrorism Analysis and Finding Insights"      #title of the website

  # to start the app to run
  app.run_server()         # debug=True

  print("Thank You")

  #deallocating the spaces

  app = None



if __name__ == '__main__':
    main()
```

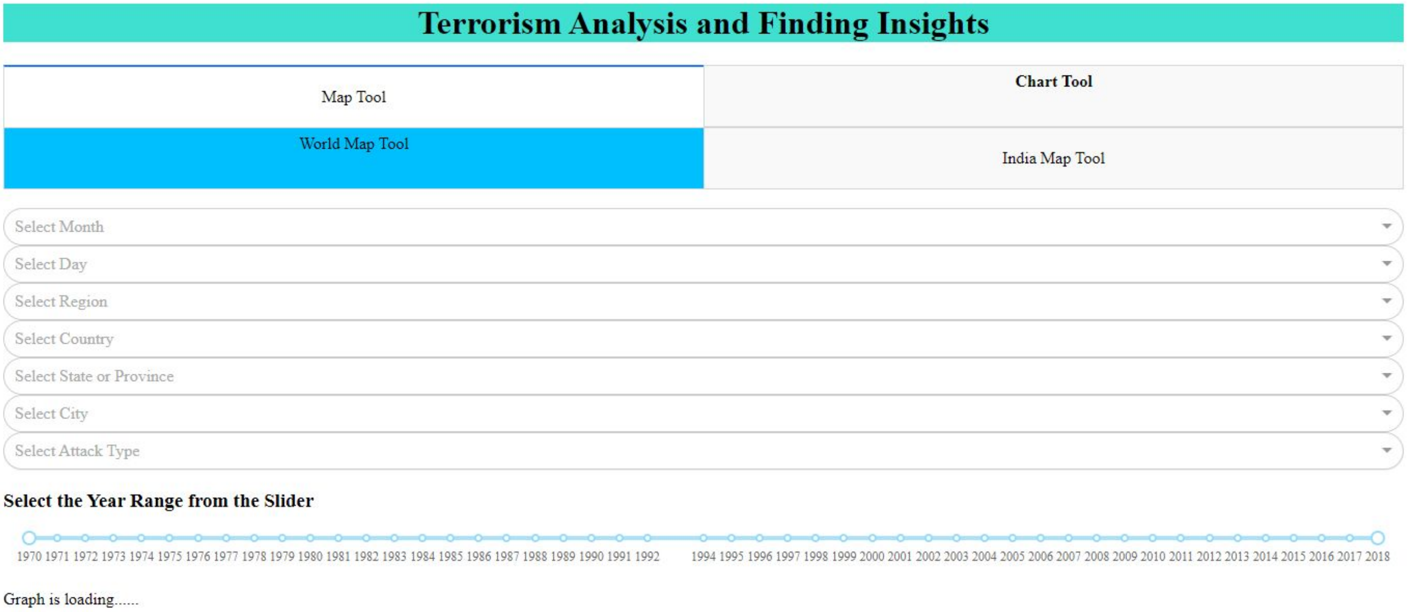
Results and Visualization

The webpage with Tabs and sub-tabs :

- ➢ **Main Tabs: Map Tool and Chart Tool**
- ➢ **Sub Tabs:**

  World Map Tool and India Map Tool

  World Chart Tool and India Chart Tool



Terrorism Analysis and Finding Insights

Map Tool | Chart Tool

World Map Tool | India Map Tool

Select Month

Select Day

Select Region

Select Country

Select State or Province

Select City

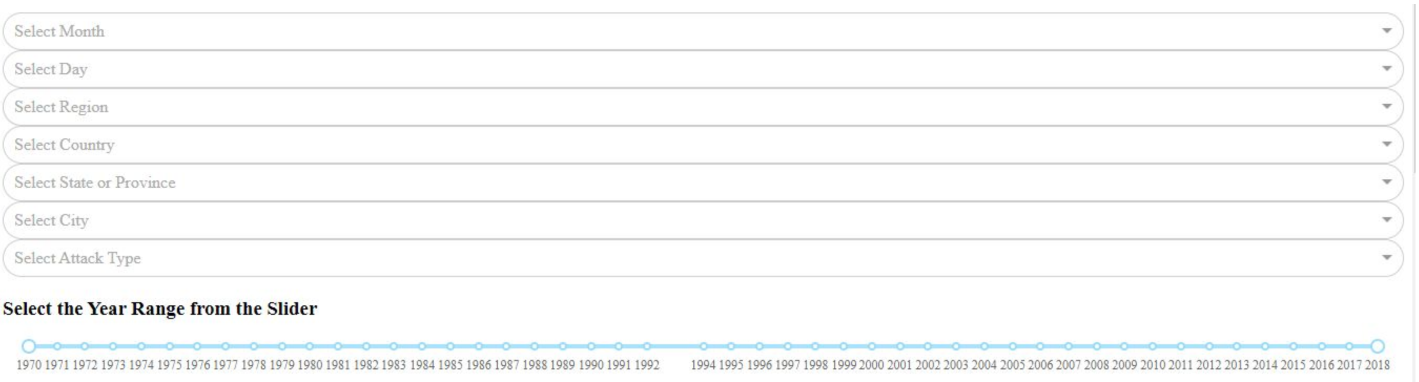
Select Attack Type

Select the Year Range from the Slider

1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018

Graph is loading......

The Dropdowns for filtering the data and Range Slider for selecting year-range :



Select Month

Select Day

Select Region

Select Country

Select State or Province

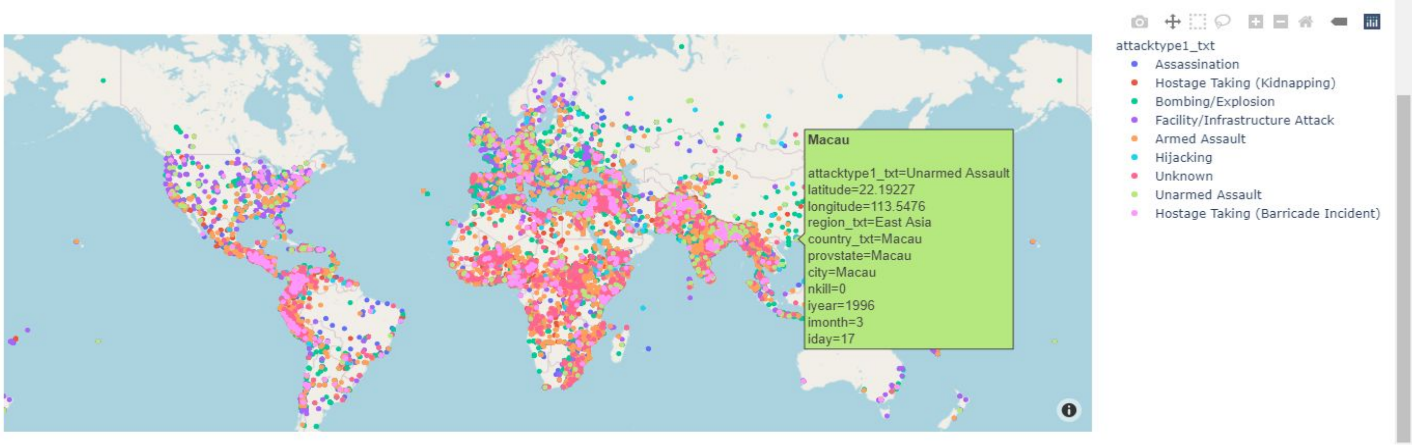
Select City

Select Attack Type

Select the Year Range from the Slider

1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018

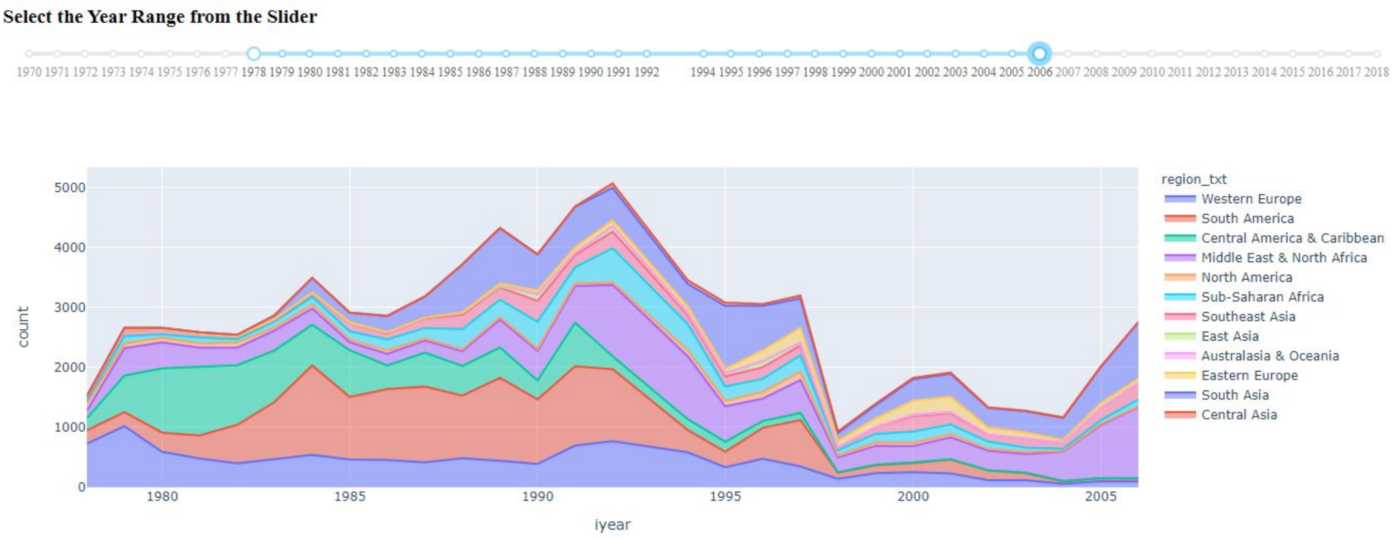
World Map having all the data before filtering according to the attack-type:



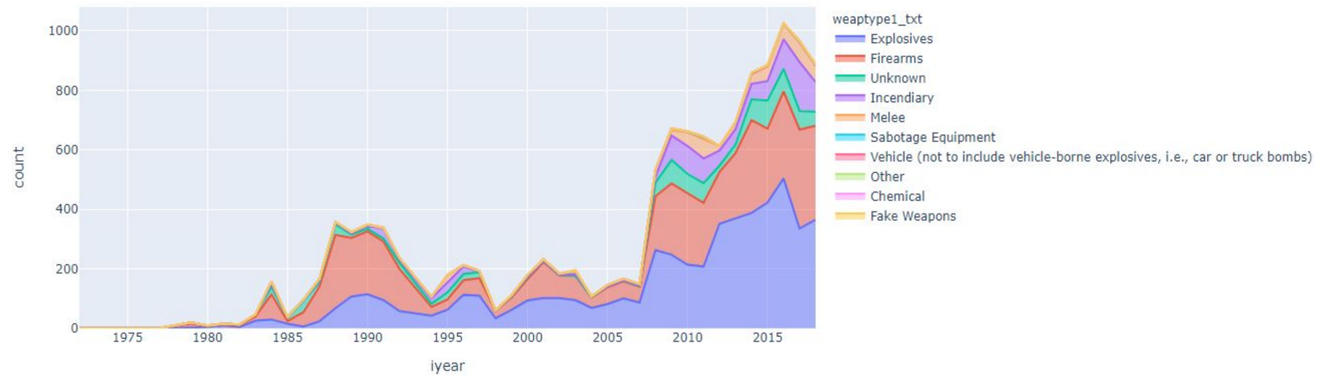
Macau

attacktype1_txt=Unarmed Assault
latitude=22.19227
longitude=113.5476
region_txt=East Asia
country_txt=Macau
provstate=Macau
city=Macau
nkill=0
iyear=1996
imonth=3
iday=17

World Chart Tool according to the regions within the year-range 1978 to 2006 :



India Chart Tool according to the weapon-type :



Software Improvements

Dash is a very popular newly emerging framework basically used for building Machine -Learning and Data Science projects, having functionalities of both the flask and reacts js where the major drawback of it being quite unclear with its concept of being open source but overall it is a great way for a Python developer to create interactive web apps without having to learn Javascript and Front End Web Development.

Python with pandas library and dash framework made it easier to access the large data into a graphical representation.

As suggested I used a different IDE that is spyder 4.1.5 for this project and the only issue I had was running my project on console and loading of a map. Because of big data, it was taking time for updation so used the F5 key for a quick run. Time to time update of spyder is required for a smooth run of big data codes. And this UI will have many other extensions with more dates in the coming years and we might need more libraries or a different approach to deal with bigger data.

Conclusion

From a beginner to now have a piece of in-depth knowledge in the fundamentals of Data Analysis and Data visualization libraries. It was quite a big learning period during the project session. Unfortunately, good code won't speak for itself. Even the most elegantly designed and well-written codebase that solves the most pressing problem in the world won't just get adopted on its own.

This is the end of my documentation review on my web app Terrorism analysis and Finding Insights. Successfully Completed the project according to the client requirement. We are now able to determine and analyze terrorism activities in particular areas at a particular time and so. It will help my client to have a better understanding of the trendline of each kind of Attack (Bombing, Assassination, etc). As the protective vehicles are less in numbers with the Army and are distributed uniformly across the area. Similarly, Explosive Detection Dogs (ED Dogs) are only less in the entire country. So now my tool will definitely help in finding the concentration of Attack type - Bombing in the area which would help in the allocation of the resources. Visualizing the data will give clear patterns about the data and makes it easy for them to analyze.

In the future, the application must be improved according to the data and the latest technology.

Acknowledgment

This project would not have been possible without the support and help of Dr. Sylvester Fernandes of Forsk Coding School. I am very much thankful to **Forsk Coding School** for their guidance and constant supervision as well as for providing necessary information regarding the project.

I am also grateful to the insightful comments and suggestions offered by my peers in the slack community for sharing information and somehow guiding and helping in error correction.

Thanks for all your encouragement!

REFERENCE

http://forskcodingschool.com/
http://dash.plotly.com/
https://www.youtube.com/watch?v=eusglTlW4OA&t=1002s
https://www.geeksforgeeks.org/

Biography

Hello,

My name is Dipanwita Manna. I am from Haldia, West Bengal. I am a final year Student currently pursuing B.tech on Computer Science and Technology Being a student from a Computer Science background I started developing my interest in Data Science and covered the basics of python under the guidance of Forsk Coding School. I am interested in Data Analysis and Data Visualization, love to work with data and find insights from it.

I am always eager to learn new technologies and have also attended workshops as well as training on Machine Learning, Deep Learning, and Data Science