Homework Assignment 1

Data Science Visualization and Exploratory Data Analysis (EDA)

1. Dataset Description and Objectives

The dataset used in this analysis was based on terror attacks in region_4, which includes records of terror attacks across different countries and years in that region, with attributes such as success rates, weapon types, attack types, target types, and regional trends. The main objectives of the EDA are:

- Cleaning and preprocessing the dataset by handling missing values, duplicates and negative values.

- Exploring correlations among numerical attributes.

- Visualizing trends of successful and failed attacks over time.

- Comparing different visualization tools for EDA in terms of usability and output quality.
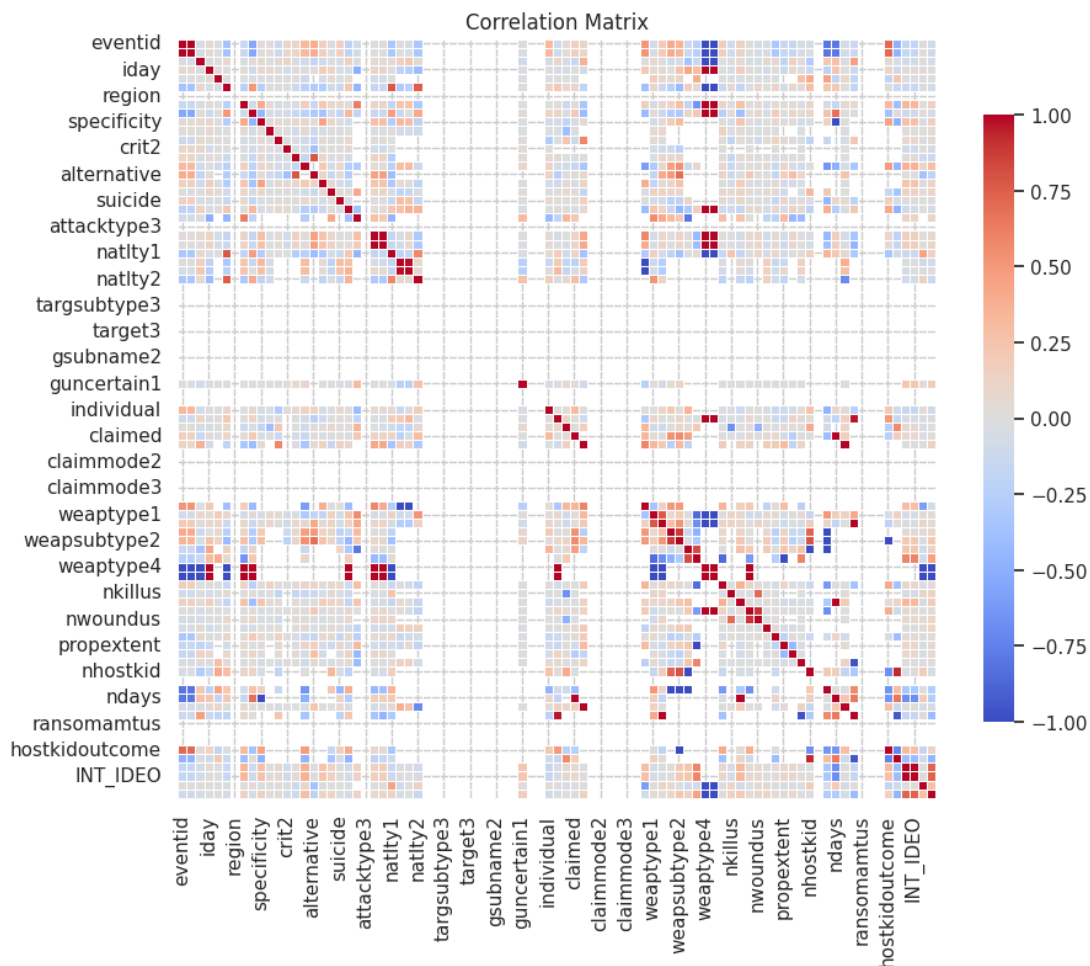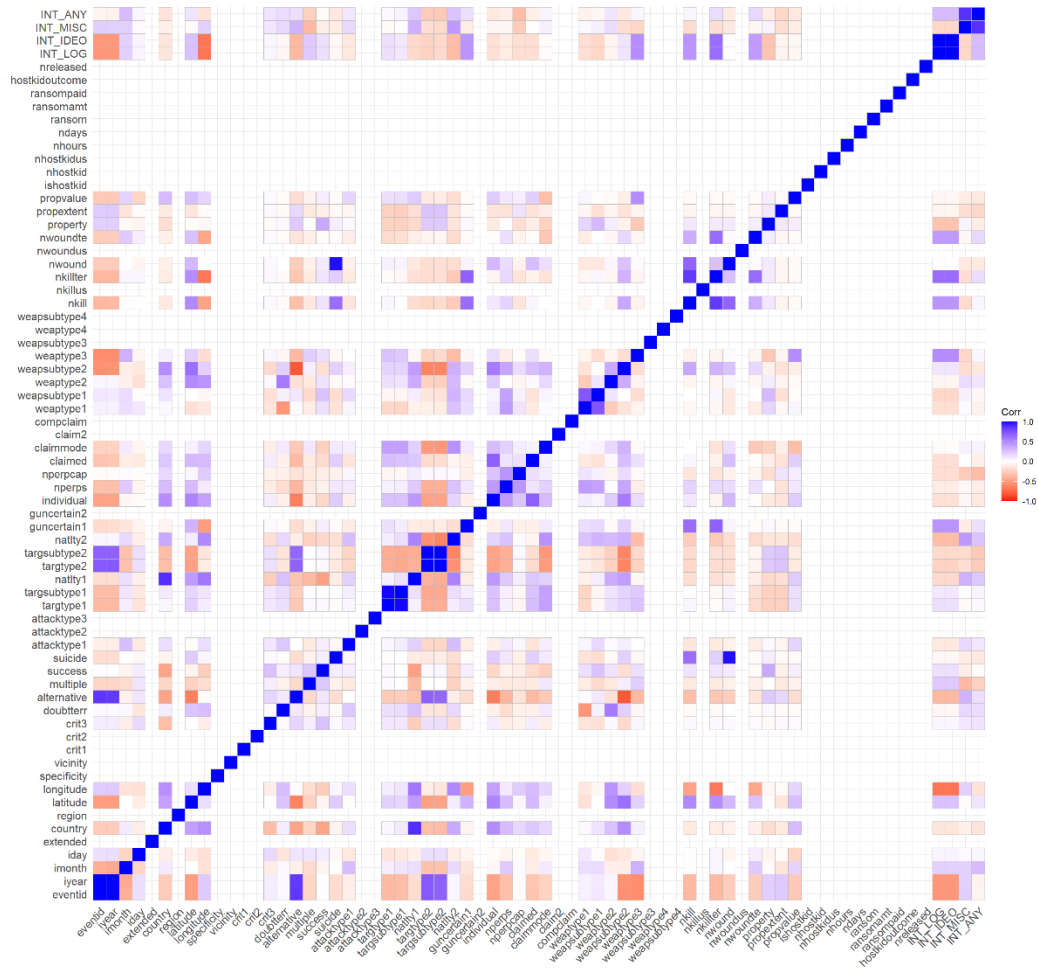

Figure 1

Figure 2

Fig.1(python) and Fig.2(R) the correlation matrix have been represented without any preprocessing steps to find the relation between the features and success. For clarity and quick insights, the R-generated (Fig-2) correlation matrix is better due to its strong diagonal and well-distributed color patterns, but we can barely see any relation of the features with respect to success. Even after some preprocessing, we can barely see any relation between the features to success.
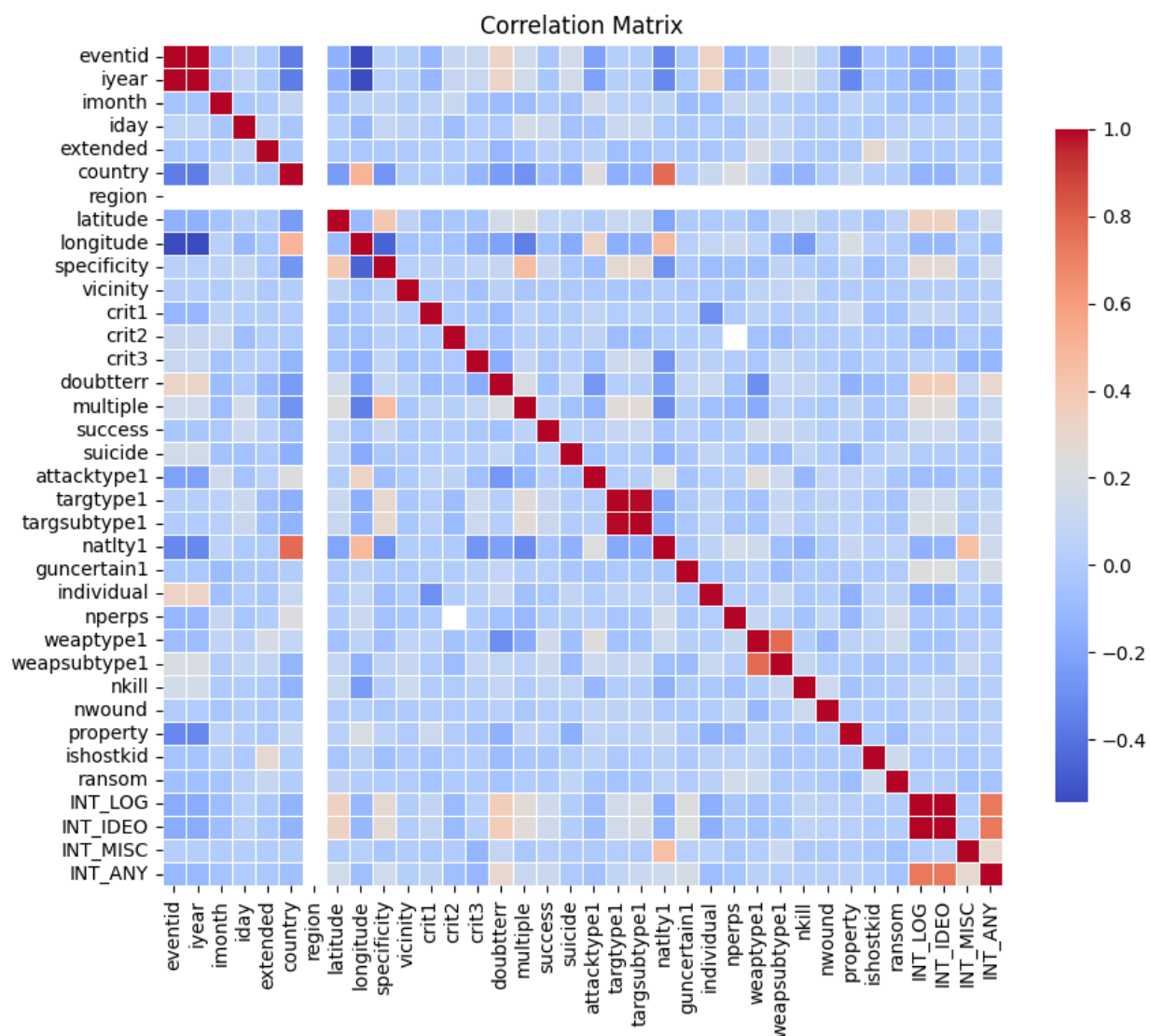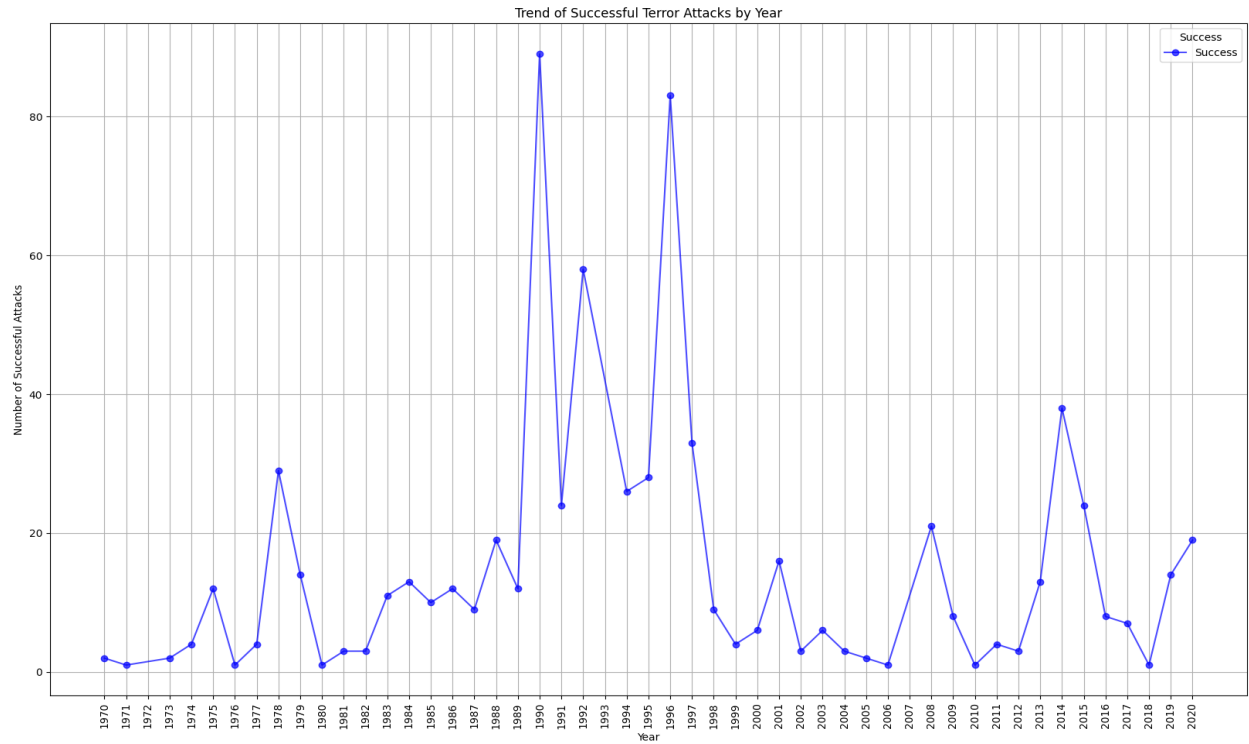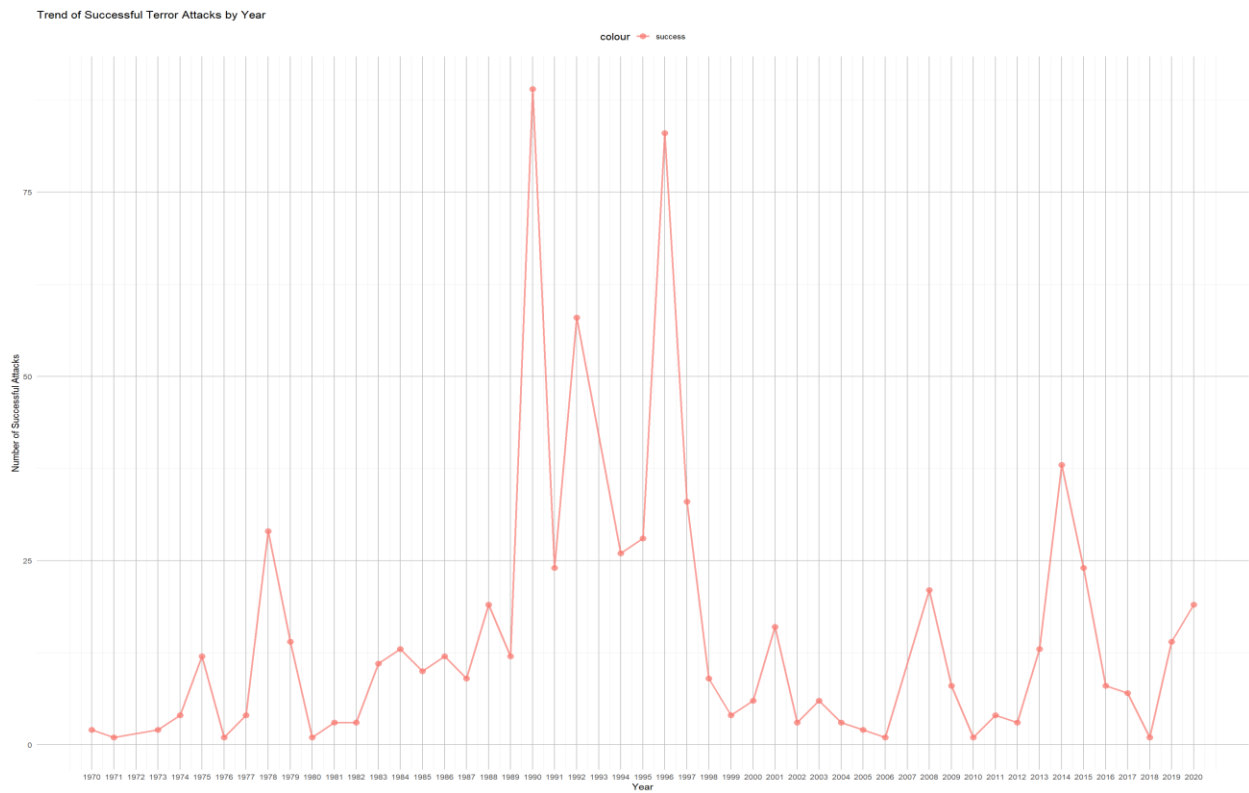
Figure 3

2. Comparison of Visualization Tools

EDA was performed using multiple tools, including R (ggplot2, ggcorrplot, plotly) and Python (Matplotlib, Seaborn, Plotly in Google Colab). These tools were compared based on:
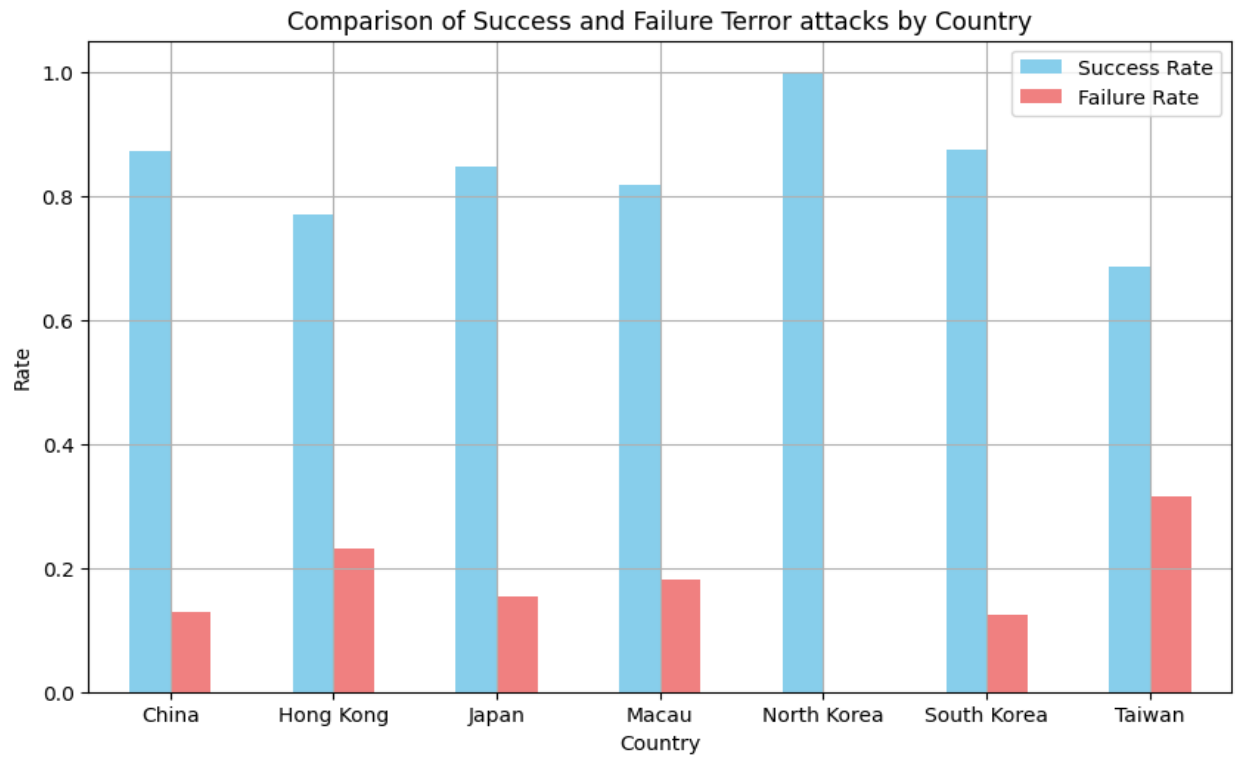
- **Visualization Quality**: The clarity and effectiveness of graphical representations.

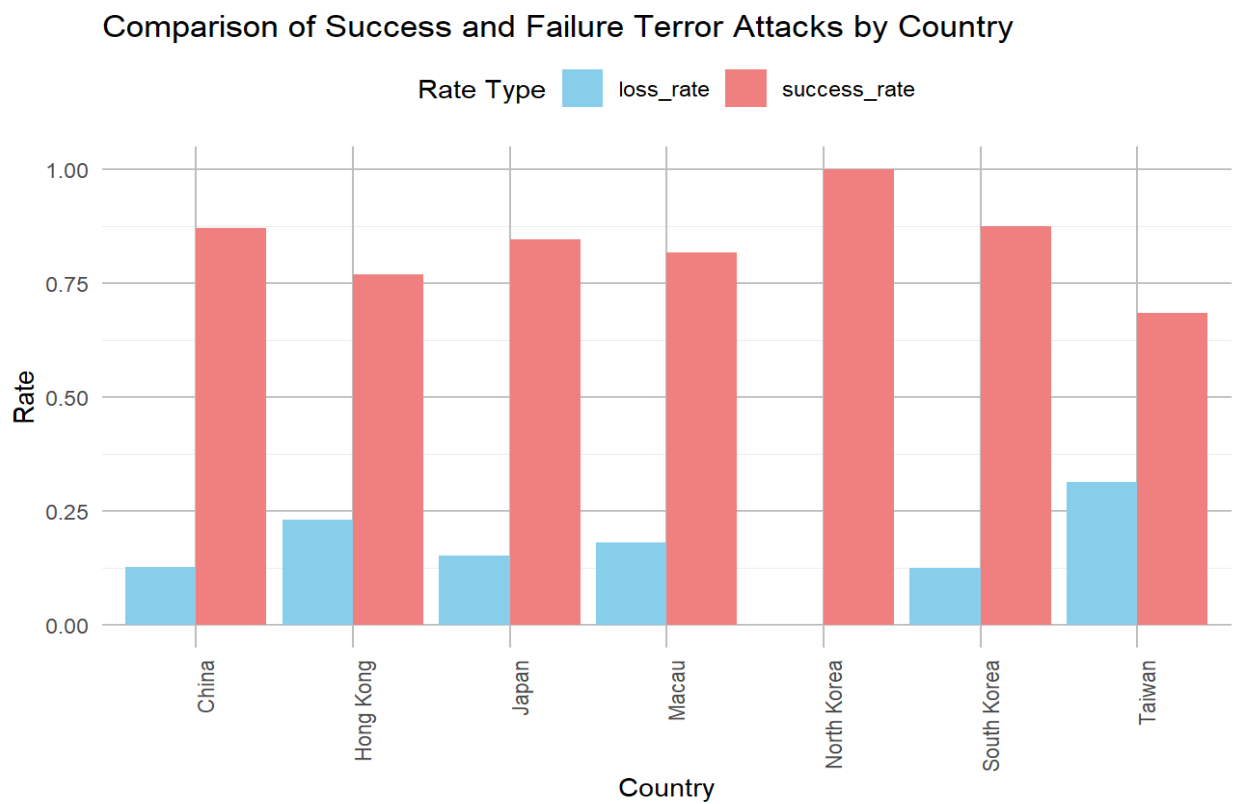- **Performance**: Speed and efficiency in handling large datasets.
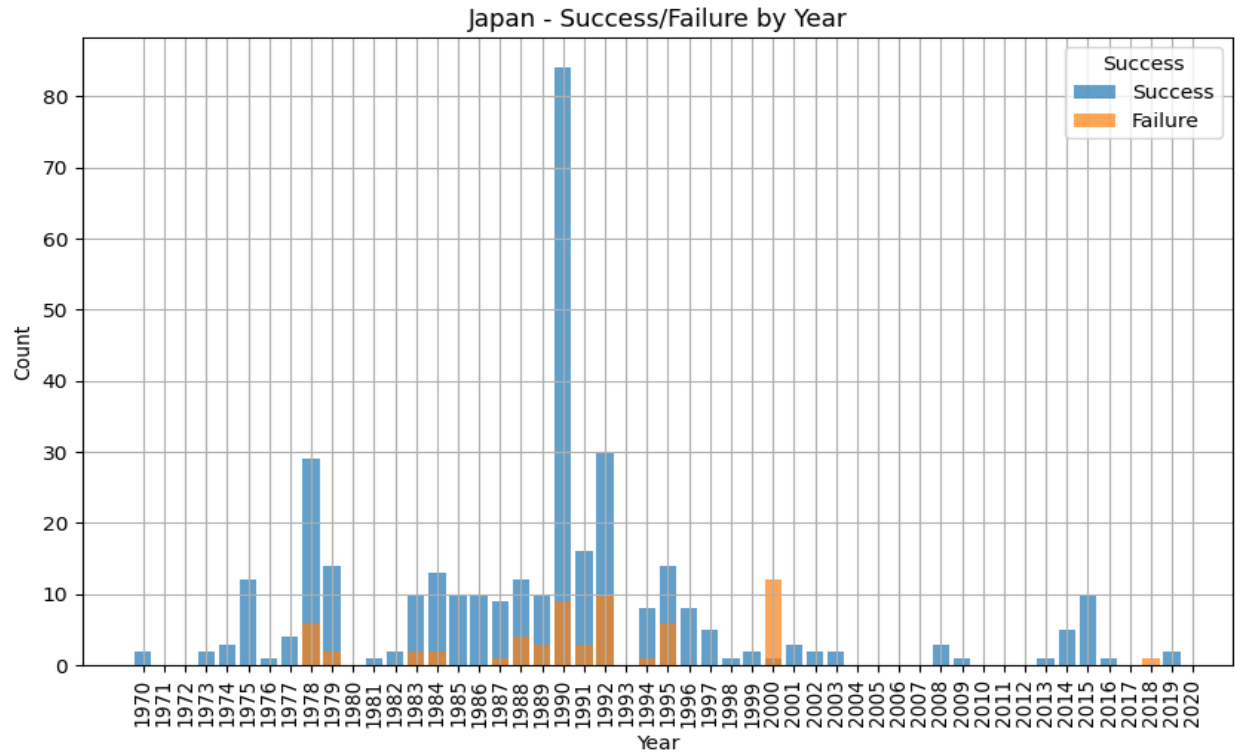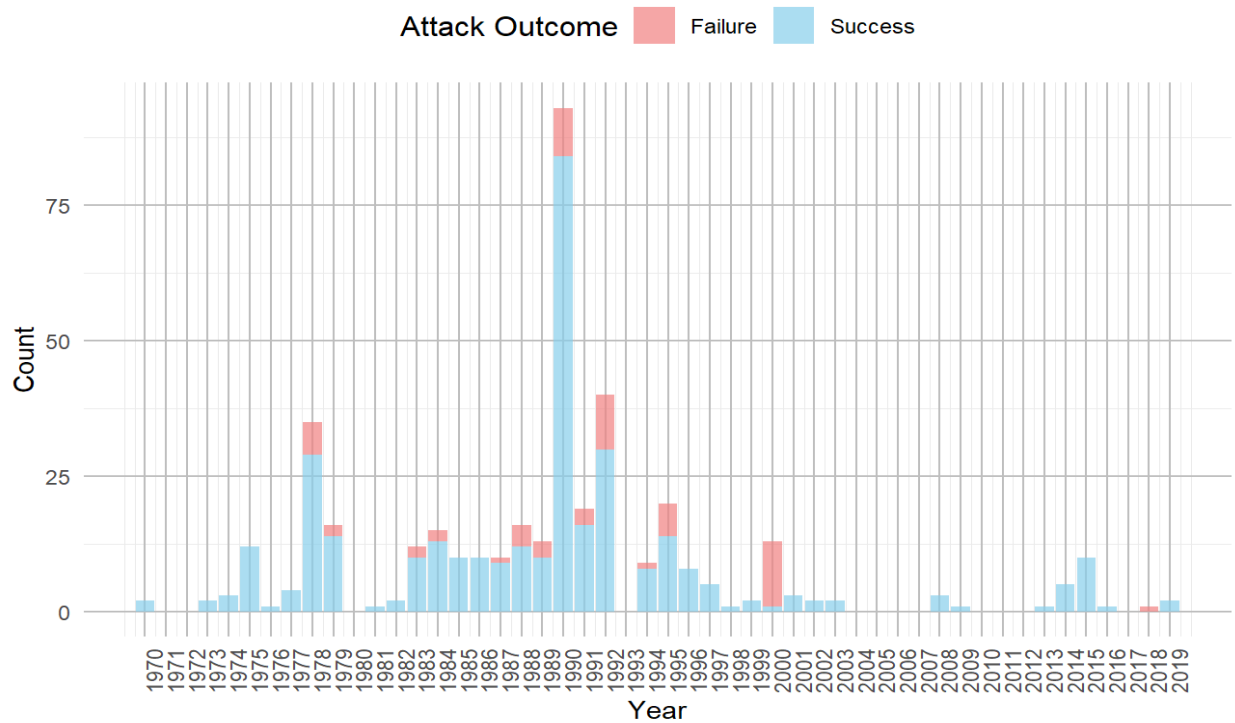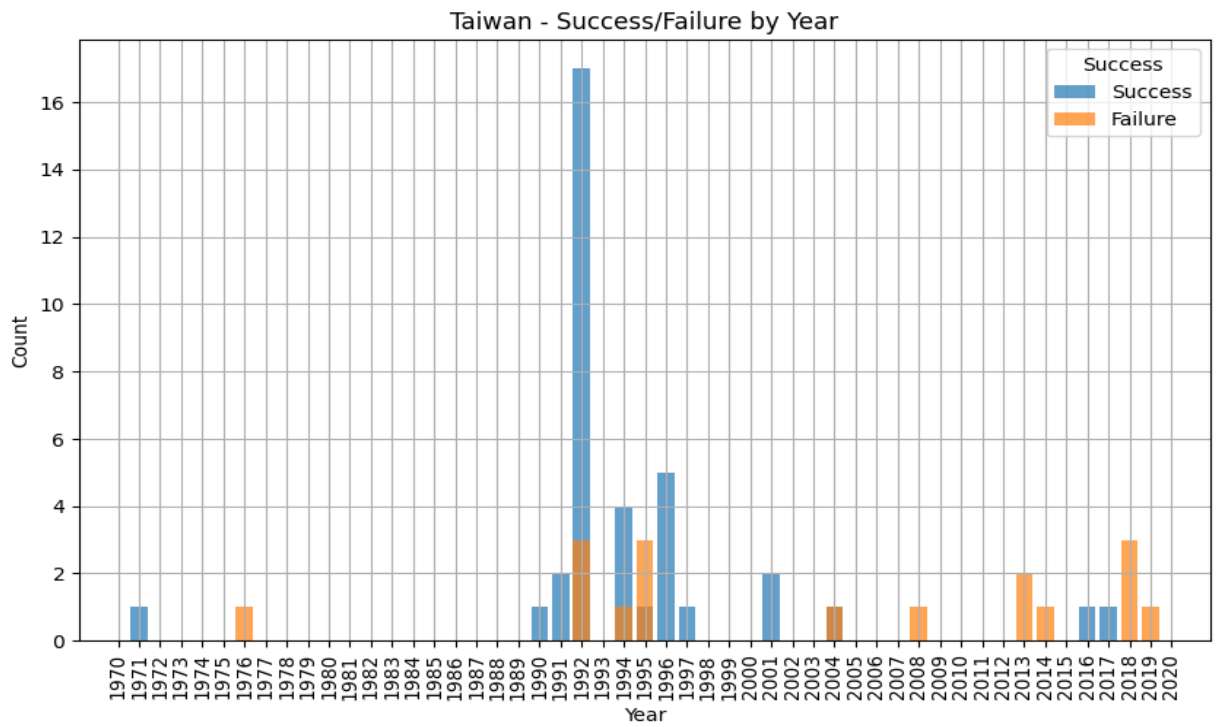
Python I

R I

Comparison of Success and Failure Terror attacks by Country

Python II

Comparison of Success and Failure Terror Attacks by Country

R II

Python III

R III

Taiwan - Success/Failure by Year

P

Python IV

Taiwan - Success/Failure by Year

R IV

China - Success/Failure by Year
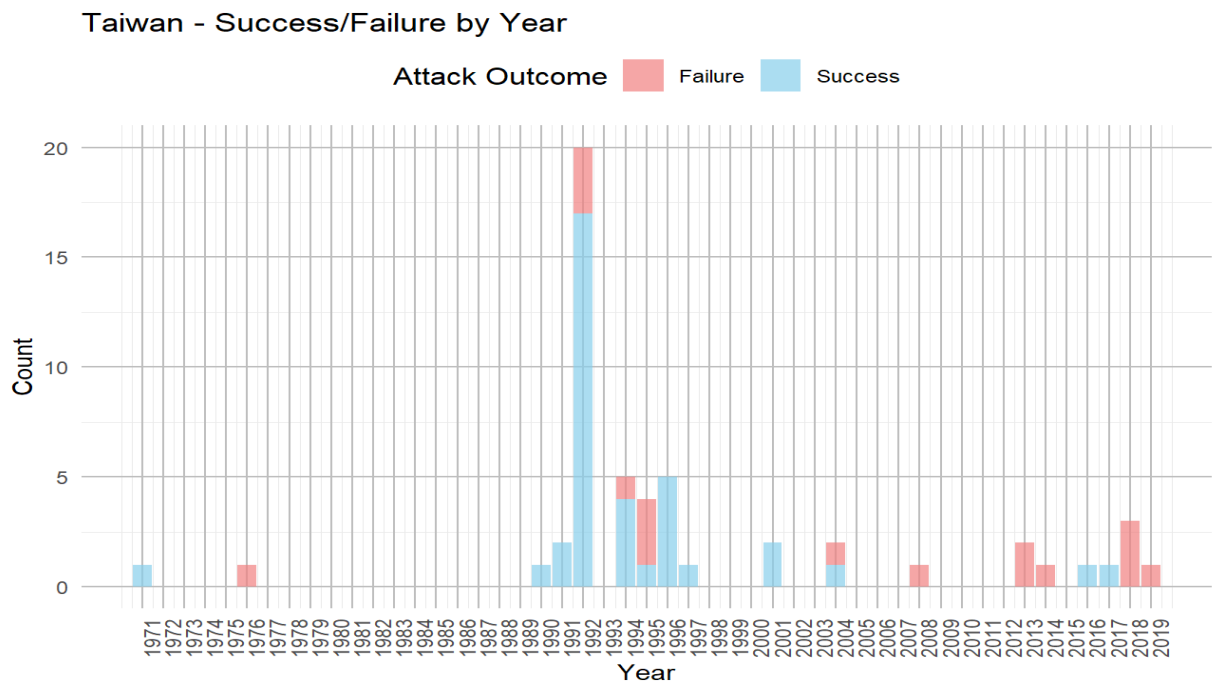
Python V

China - Success/Failure by Year
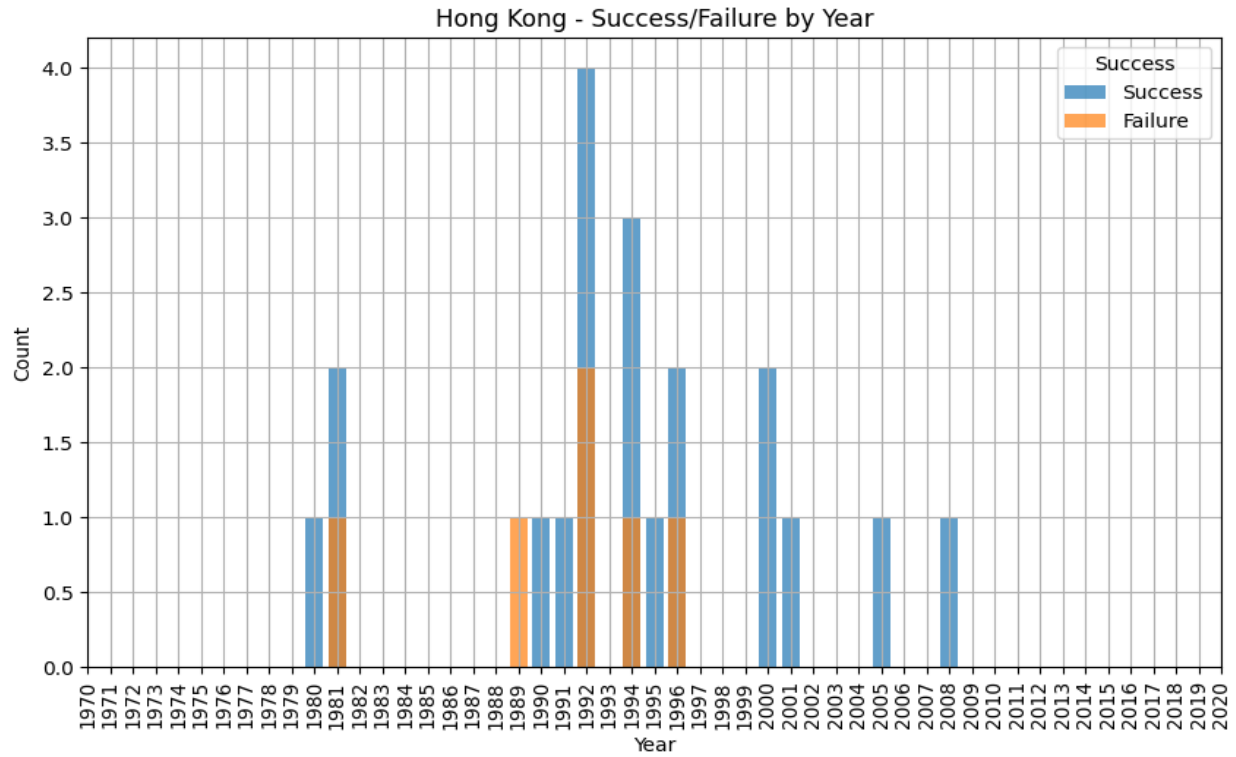
R V

Python VI

R VI

Python VII

R VII

North Korea - Success/Failure by Year
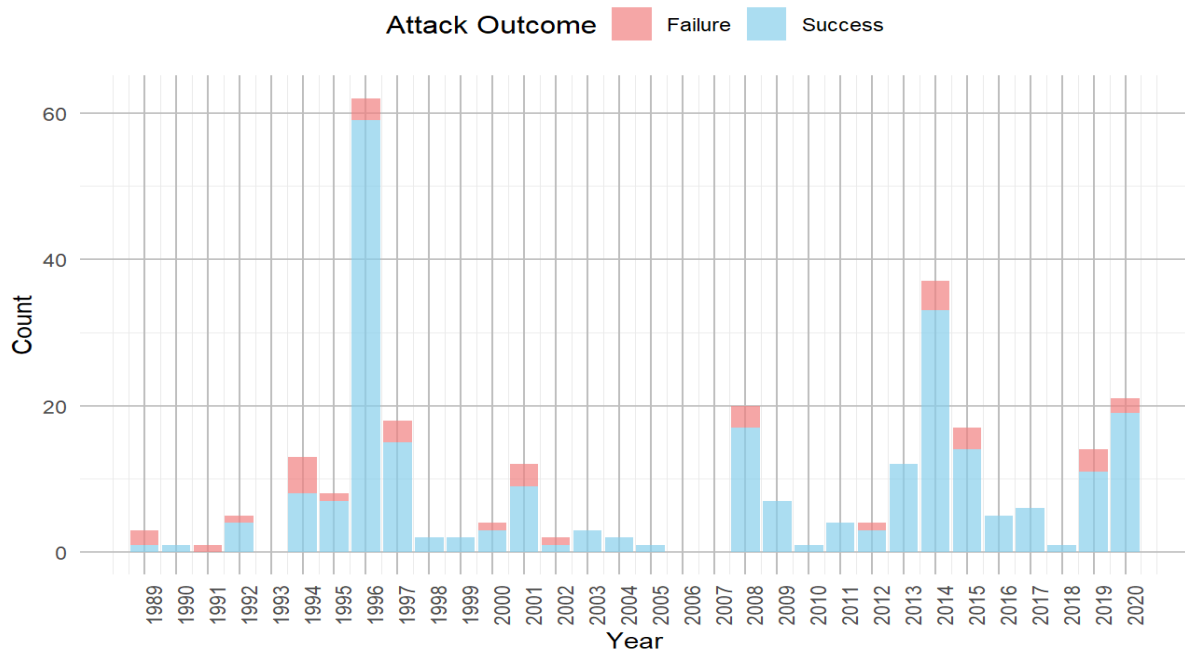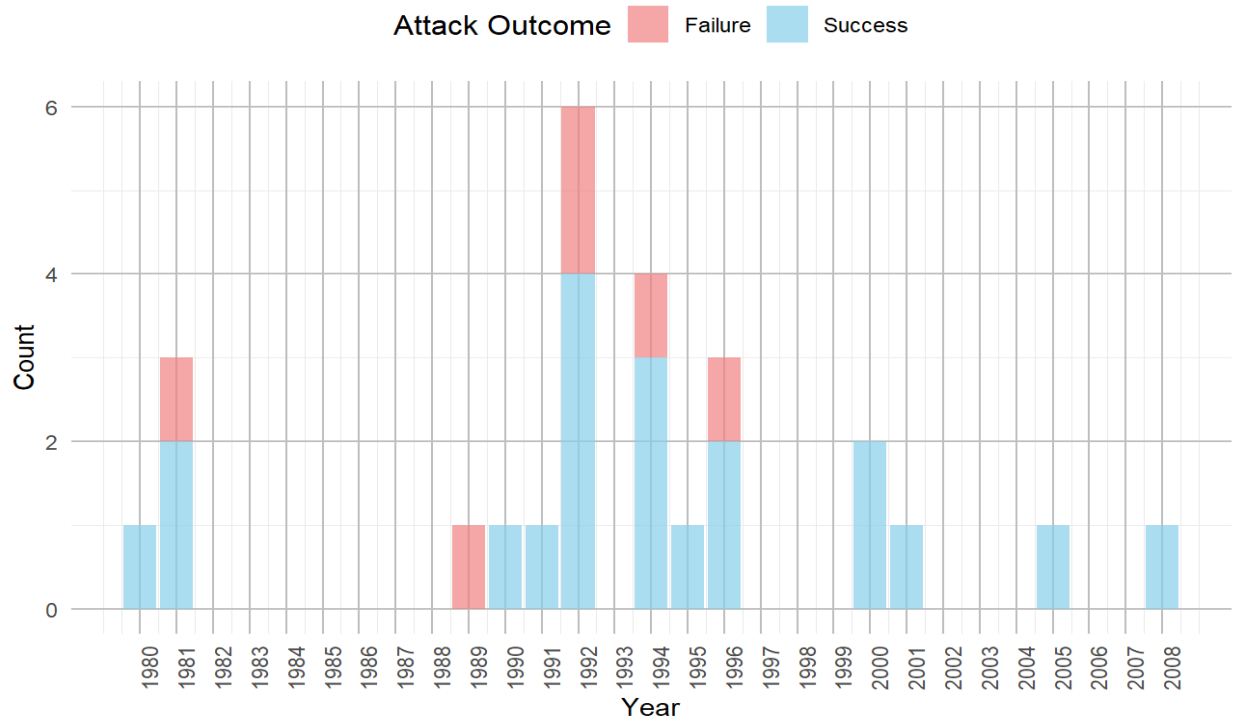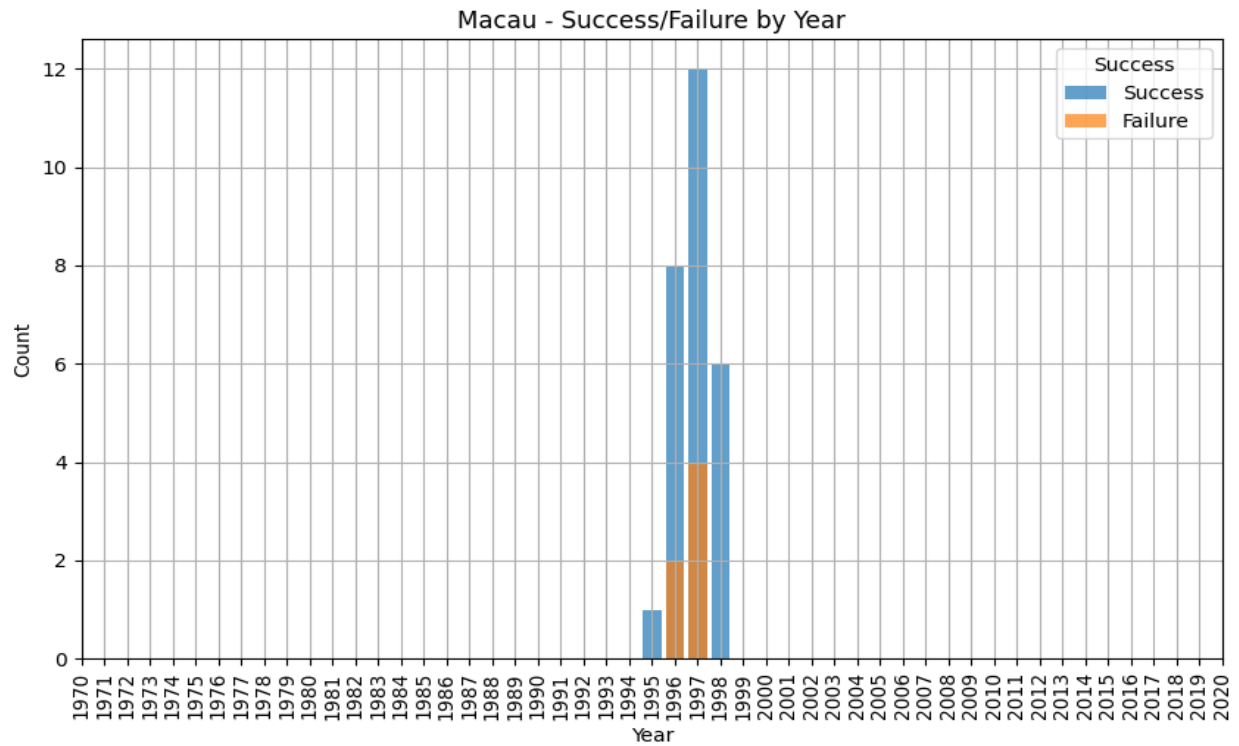
Python VIII
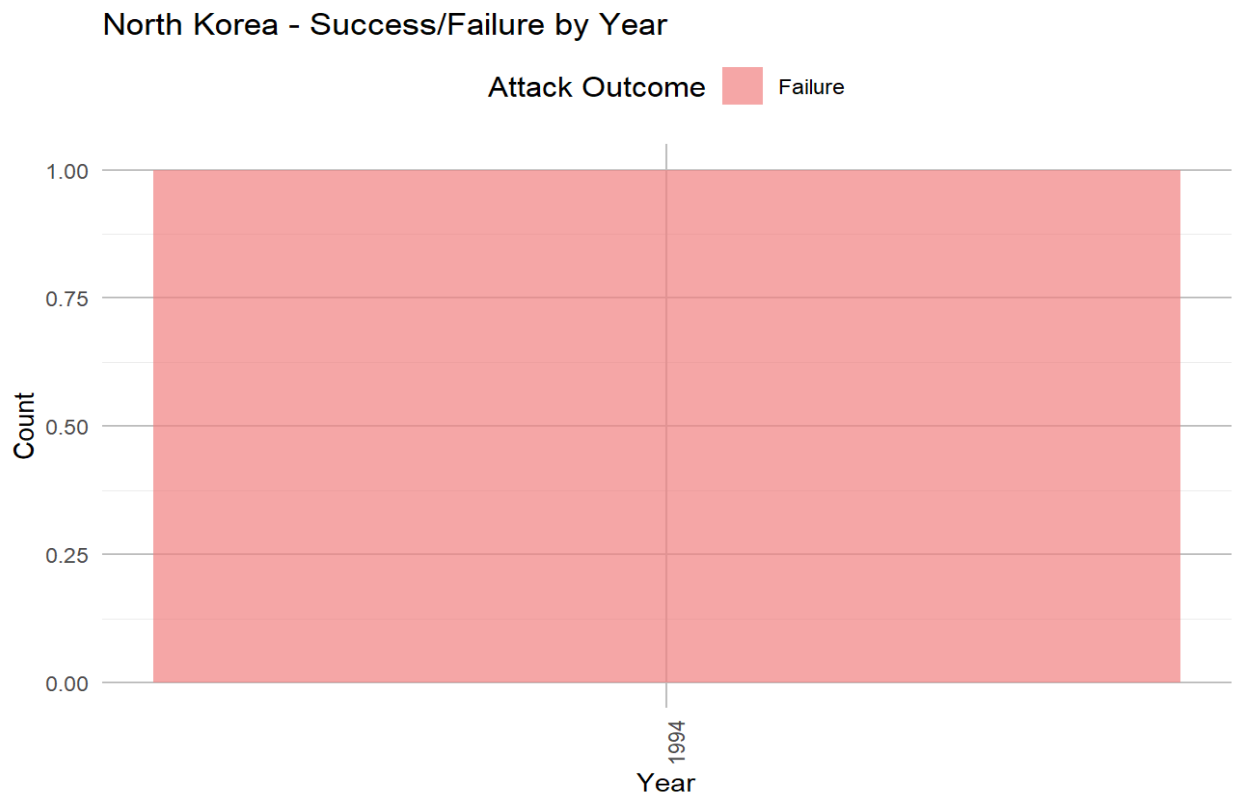
North Korea - Success/Failure by Year

R VIII
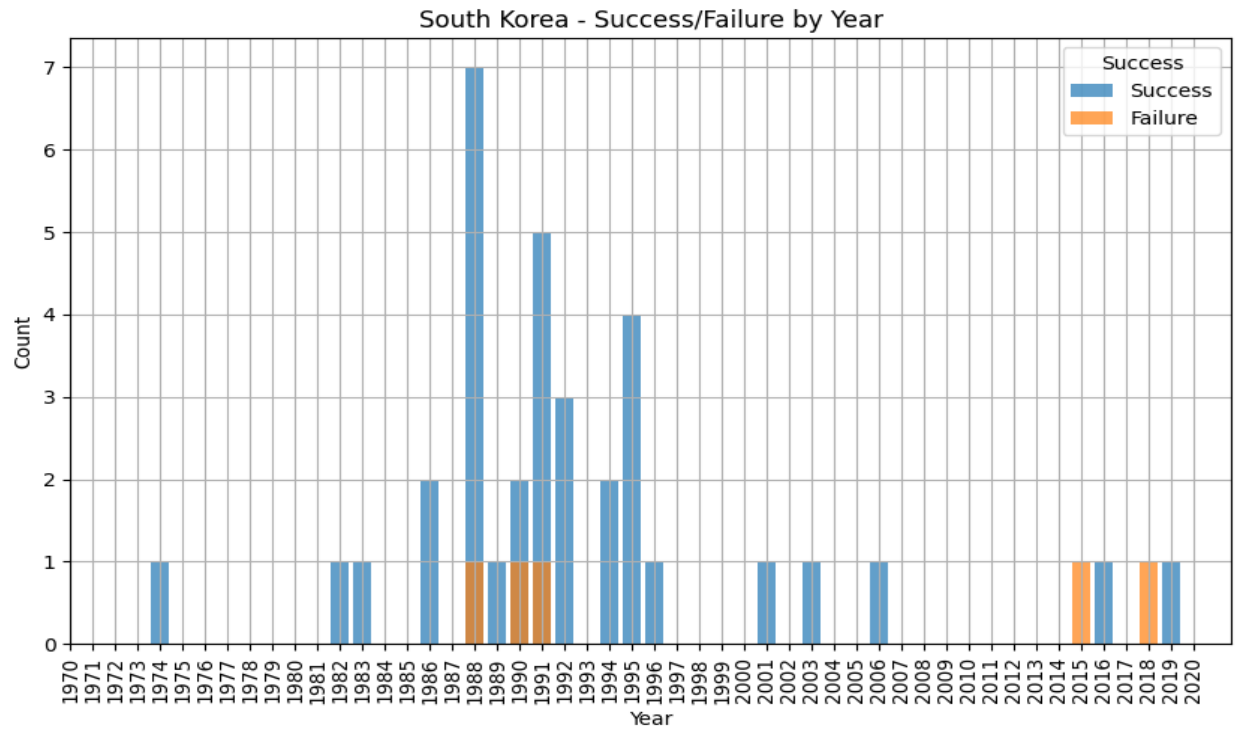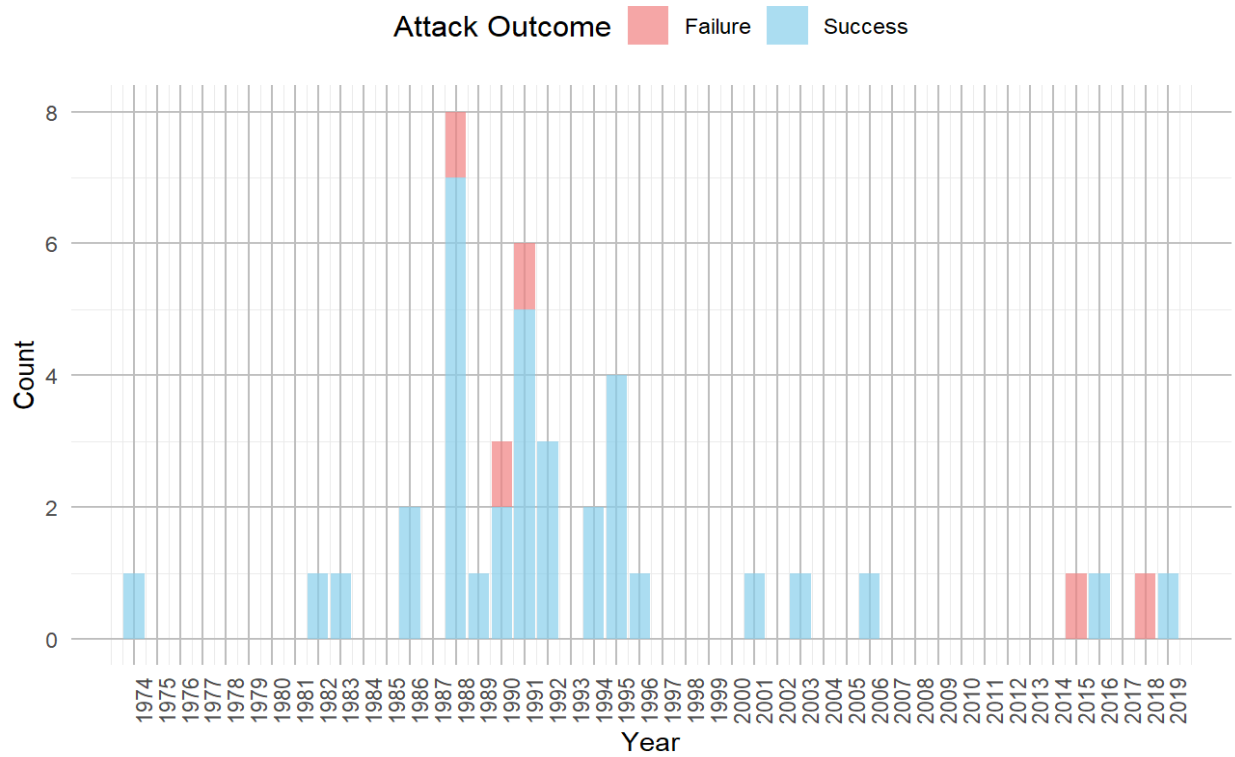
Python IX

R IX

Terror Attacks by Country
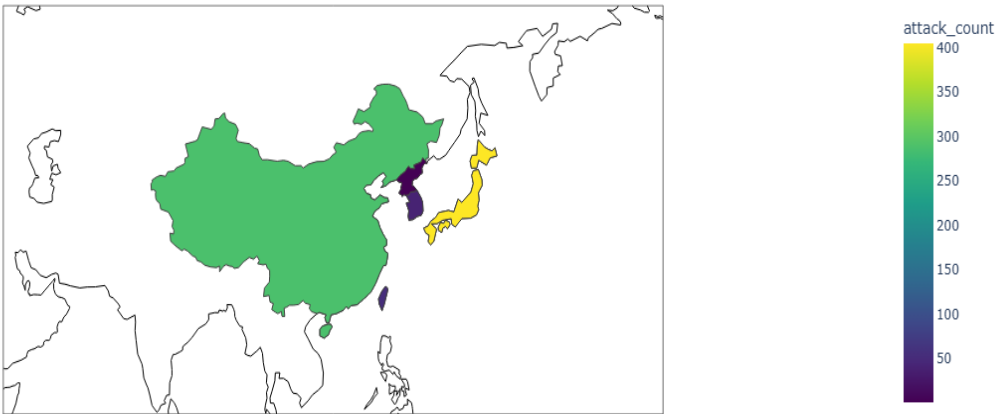
Python X

Terror Attacks by Country

R X

Treemap of Attacks by Target Type

R was unsuccessful to display a Treemap using plotly library.

The left side is generated through R and the right through Python.

Tool	Visualization Quality	Performance
R(ggplot2)	High	Efficient
R(plotly)	Interactive & High	Moderate
Python(matplotlib)	Moderate	Fast
Python(searborn)	High	Fast
Python(plotly)	Interactive & High	Moderate

3. Visualizations and Interpretations

3.1 Time-Series Analysis of Successful Attacks

- A line chart created using ggplot2 (R) displayed the trend of successful attacks over the years.

- In Python (Seaborn, Matplotlib), similar trends were observed with some stylistic differences in the plots.

- plotly (both R and Python) allowed interactive exploration, enhancing insights.

3.2 Stacked Bar Chart for Attack Success/Failure

Using ggplot2 in R and seaborn in Python, a stacked bar chart compared the number of successful and failed attacks yearly.

3.3 Choropleth Map of Attacks by Country

- plotly (R and Python) was used to create an interactive world map showing the number of attacks by country.

- Interactive elements improved the usability compared to static maps in matplotlib.

3.4 Attack Success Rate by Weapon Type

- A bar chart (ggplot2 and seaborn) displayed the success rate of different weapon types.

- Both R and Python provided similar insights, but Python's seaborn offered slightly more aesthetic appeal.

4. Key Insights and Tool Strengths/Weaknesses

Key Insights:

- There is a significant increase in successful attacks over time, with peaks in specific years.

- Some countries experience higher attack frequencies, as shown in the choropleth map.

- Weapon type and attack type play a critical role in determining attack success.

- The success rate of suicide attacks was visualized using a pie chart, revealing a high failure rate.

Strengths & Weaknesses of Tools:

- ggplot2 (R): Strong for static visualizations but lacks interactivity.

- seaborn (Python): Excellent aesthetics and ease of use for EDA.

- plotly (R & Python): Best for interactive analysis, but computationally expensive.

- Matplotlib (Python): Simple but requires extensive customization for high-quality visuals.

Conclusion

Both R and Python offer powerful visualization capabilities. For static EDA, ggplot2 and seaborn perform well, while plotly is best suited for interactive and dynamic exploration. The choice of tool depends on the specific requirements of the analysis and the user's familiarity with coding environments.