**A blue and orange logo

Description automatically generated**

**DASC-5305-001-DATA VISUALIZATION ASSIGNMENT - 1**

**GROUP - 5**

**Jennithra Srinivasan - 1002206950**

**Mithran Pichaya Kumaran - 1002198775**

**Navya Manjunatha - 1002202642**

**Nivas Marudhavanan - 1002197280**

**SUBMISSION ON : 09/27/2024**

**A blue and orange logo

Description automatically generated**

**DEPARTMENT OF DATASCIENCE**

**Analysis of Terrorism Incidents in Various Countries - Group-5**

**Python**

**1. Introduction**

* **Dataset Description:**
  + The dataset contains detailed records of terrorism-related events occurring in various countries, spanning various years.
  + Key features include event identifiers, dates, location details, attack characteristics, perpetrator information, and casualty figures.
* **Tool Comparison:**

**R Libraries:**

1. **Amelia**: User-friendly for missing data imputation with effective visual diagnostics.
2. **corrplot**: Simple to create clear and informative correlation heatmaps.
3. **GGally**: Intuitive for generating pair plots; enhances ggplot2's capabilities.
4. **dplyr**: Excellent for data manipulation with clear syntax, integrates well with ggplot2.

**Python Libraries:**

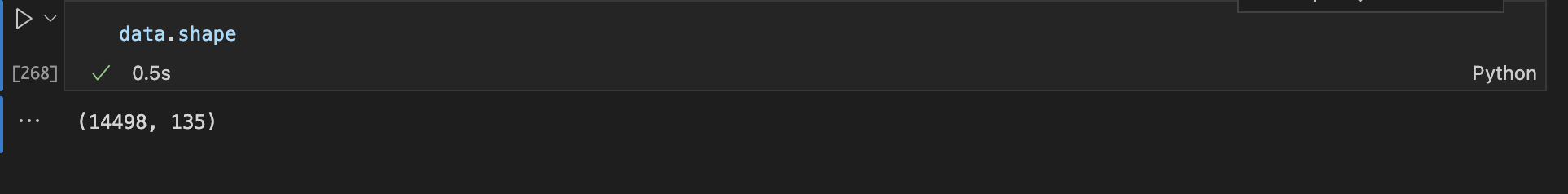
1. **pandas**: Versatile for data manipulation with numerous cleaning and transformation functions.
2. **seaborn**: Simplifies the creation of attractive statistical graphics; high-quality visualizations.
3. **matplotlib**: Foundational library for plotting with extensive customization options, though more complex to use.
4. **scipy.stats**: Offers a range of statistical tests; supports visualizations like Q-Q plots.
5. **LinearRegression**: Easy to implement for regression analysis, works well with visualization libraries.

2**. Data Overview – Dataset : region\_5.csv**

* **Key Features:**
  + **Date Information:** Year, month, and day of the event.
  + **Location:** Country and regional classification.
  + **Attack Characteristics:** Types of attacks and weapons used.
  + **Perpetrators:** Names of terrorist groups involved.
  + **Casualty Data:** Number of fatalities and injuries.

**3. Exploratory Data Analysis (EDA):**

The dataset comprises 14,498 records and 135 features, focusing on various incidents of terrorism from around the world. Each row represents an individual incident, while the columns contain a wide range of information pertinent to the event.



1. **Missing Values Analysis:**

In our analysis of the dataset, we conducted a thorough examination of missing values to understand the completeness of the data.

**Identification of Missing Values**: We identified which columns in the dataset contained missing values and quantified the extent of these missing entries.

1. **Calculation of Missing Value Percentages:** For each column with missing values we calculated the percentage of missing entries relative to the total number of records.
2. **Summary of Missing Values:** We compiled a summary that includes both the count and percentage of missing values for each affected column.
3. **Focus on Affected Columns:** By filtering the results to focus only on columns with missing values, we can effectively target our data cleaning efforts.
4. **Threshold Definition**: We established a threshold of 90% for missing values, meaning any column with more than 90% of its entries missing would be considered for removal from the dataset. This criterion helps retain only those features that are sufficiently populated and relevant for analysis.
5. **Identification of Columns to Drop**: We identified columns that exceeded the 90% threshold of missing values. By filtering the dataset, we focused on those columns that would not contribute significantly to our analysis due to the excessive amount of missing data.
6. **Creation of a Cleaned Dataset**: Using the identified columns, we created a new dataset, referred to as cleaned\_data, by dropping these high-missing-value columns. This process ensures that our analysis is based on a dataset that contains only relevant and meaningful features.
7. **Final Dataset Shape**: After dropping the columns, we evaluated the shape of the cleaned\_data to understand its new dimensions. This final dataset will serve as the foundation for our subsequent analysis, ensuring that we work with a more robust and reliable dataset.
8. **Data Cleaning Process**

Following the missing values analysis, the dataset underwent a cleaning process:

* **Columns Dropped:** Columns exceeding the defined threshold of missing data were removed from the dataset.
* **Current Shape:** The resultant cleaned dataset now comprises 14,498 rows and 71 columns, significantly reducing complexity while retaining critical information necessary for further analysis.

1. **Duplicate Entries**

A check for duplicate rows was conducted on the cleaned dataset. The result showed that there are no duplicate entries, indicated by an empty DataFrame.

A screen shot of a computer

Description automatically generated

1. **Column Types:**

Cleaned\_data Dataframe has

* Number of Object Columns: 26
* Number of Non-Object Columns: 45

1. **Visual Analysis of Numerical Columns:**

For each column with missing values and of type int64 or float64, the following visualizations were created:

* 1. **Histogram:** Displays the distribution of values in the column, helping to identify the frequency of data points and any skewness in the data.
  2. **Boxplot:** Provides a visual summary of the central tendency, spread, and potential outliers in the data.
  3. **Q-Q Plot:** Used to assess whether the distribution of the data follows a normal distribution. The plot compares the quantiles of the dataset against the quantiles of a theoretical normal distribution.

**Regression Line on Q-Q Plot:**

* 1. A linear regression model was fitted to the Q-Q plot points. The regression line (shown in green dashed format) helps in evaluating how well the sample quantiles align with the theoretical quantiles.
  2. This addition to the Q-Q plot provides further insight into the normality of the distribution, as deviations from the line suggest departures from normality.

PLOTS:

A graph with a number of points

Description automatically generated with medium confidence

A graph with a blue rectangle

Description automatically generated

1. **Key Findings and Actions Taken**
2. **Identification of Missing Values:**
   * Each column with missing values was examined, and the percentage of missing data was calculated and printed. The data type of each column was also noted for further analysis.
3. **Handling Numeric Columns:**
   * For columns with numeric data types (int64, float64):

**Columns Dropped**: Columns with more than 50% missing values were dropped from the dataset.

**Filling Missing Values**:

* For numeric columns with less than 50% missing values:
  + If the distribution was approximately normal (skewness between -1 and 1), missing values were filled with the mean.
  + For skewed distributions, missing values were filled with the median.

1. **Handling Categorical Columns:**

Missing values in categorical columns were filled with the mode (most frequent value) to retain data integrity and ensure no loss of significant information.

**OUTPUT:**

A computer screen with white text

Description automatically generated

1. **Identification of Categorical Columns:**
   * Each column in the cleaned dataset was assessed. For columns of type object, the number of unique values was calculated.
   * The unique counts for categorical columns were printed to facilitate understanding of their characteristics.
2. **Processing Categorical Columns:**
   * Columns with fewer than 6 unique values were flagged for further processing. For each of these columns:
   * The unique values were displayed to gain insight into the data.
   * The data type of the column was converted from object to category to optimize memory usage and improve performance in analysis.
3. **Handling Missing Values:**
   * For columns with missing values, the number of missing entries was checked. If present, the missing values were filled using the mode (the most frequently occurring value) to maintain data integrity.
4. **Visualization:**
   * A count plot was generated for each categorical column to visualize the distribution of unique values. This graphical representation helps to identify the frequency of each category and assess any imbalances in the data.

**PLOT:**

A green rectangular object with white text

Description automatically generated

1. **Features Analyzed for suicide analysis with region:**

The following features were selected for the analysis:

* Year of Attack (iyear): Represents the year in which the attack occurred.
* Month of Attack (imonth): Indicates the month of the attack.
* Day of Attack (iday): Specifies the day on which the attack took place.
* Country (country): Denotes the country where the attack occurred.
* Region (region): Indicates the broader geographical region of the attack.
* Type of Attack (attacktype1): Describes the category of the attack.
* Type of Weapon Used (weaptype1): Specifies the weapon type utilized in the attack.
* Number of Killed (nkill): Represents the total number of casualties in the attack.

**Scatter Plot Observations:**

* Each plot visualized the distribution of the target variable in relation to the respective feature.
* The scatter plots allowed for the identification of potential trends, clusters, or anomalies in the data.

A graph with blue dots

Description automatically generated

A blue graph with numbers and lines

Description automatically generated

A screenshot of a graph

Description automatically generated

1. **CORRELATION:**

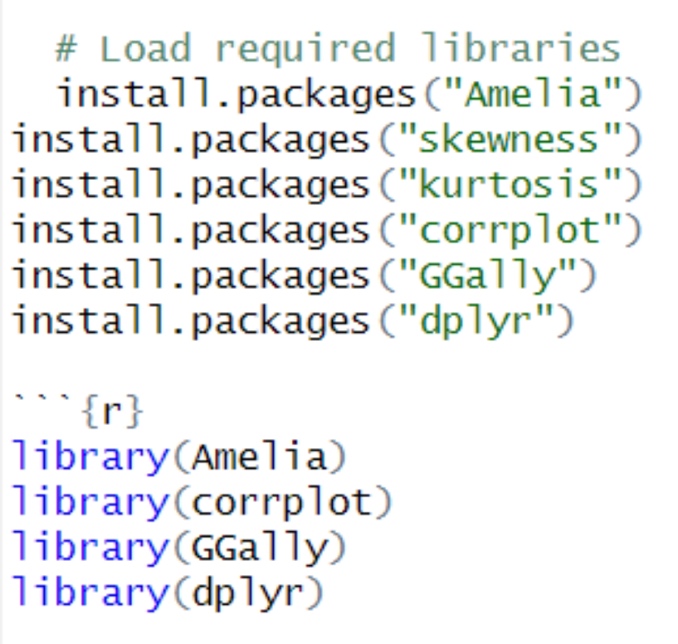
* **Correlation Matrix:** A correlation matrix was computed to quantify the relationships between pairs of features. This matrix ranges from -1 (perfect negative correlation) to +1 (perfect positive correlation).
* **Heatmap Visualization:** A heatmap was created to visualize the correlation values clearly.

A diagram of heat map

Description automatically generated

**R-PROGRAMMING**

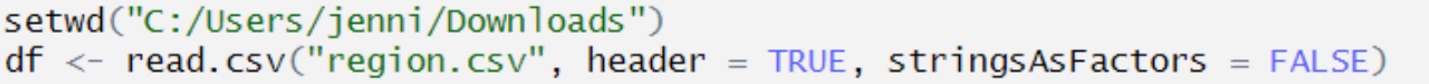
**LIBRARIES:**



Install and load necessary libraries for handling missing data, plotting correlations, performing data analysis, and manipulating data.

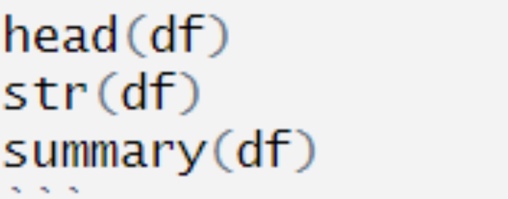
* **Amelia:** Handles missing values.
* **corrplot:** Visualizes correlation matrices.
* **GGally:** Provides enhanced visualizations, such as pair plots.
* **dplyr:** Data manipulation and grouping.

**LOADING DATA:**



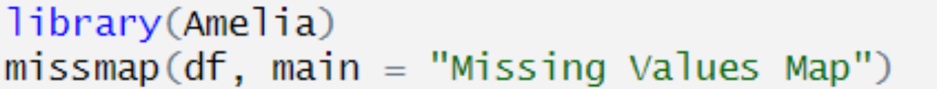
Setting the working directory and load the dataset from a CSV file. The stringsAsFactors = FALSE prevents automatic conversion of character columns to factors.

**INITITAL DATA INSPECTION:**

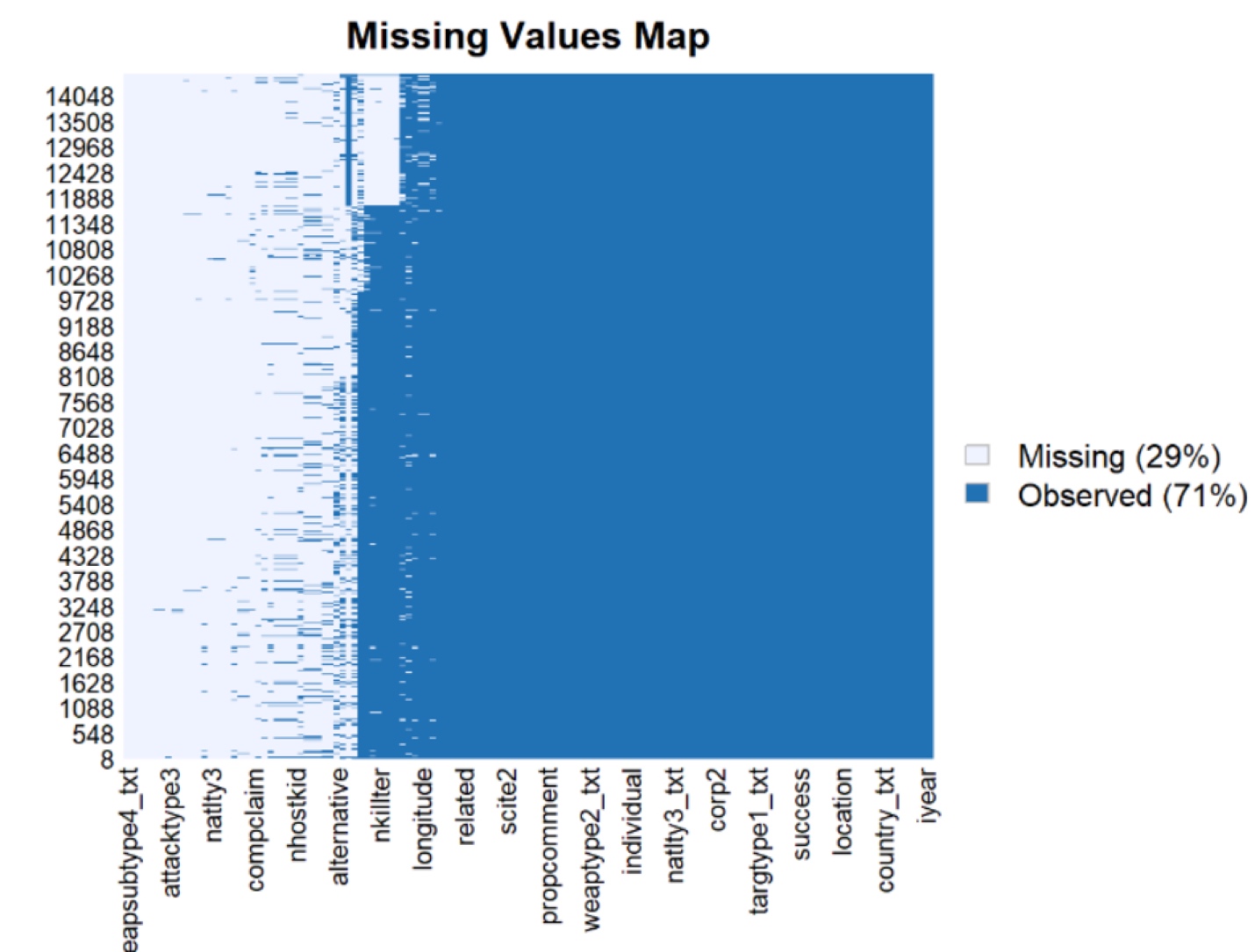
****

* **head():** Displays the first few rows.
* **str():** Shows the structure, data types, and column names
* **summary():** Provides descriptive statistics for each column.

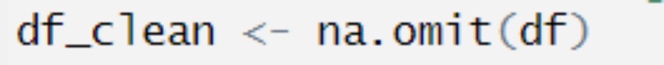
**MISSING DATA VISUALIZATION:**



* **missmap():** Creates a heatmap of missing values.
* **colSums(is.na()):** Counts missing values for each column.



**DATA CLEANING:**



Remove rows with any missing values.



Fill missing values in specific numeric columns using the mean or median, depending on the column distribution.

**DUPLICATES DETECTION:**



**Finding duplicates in the data.**



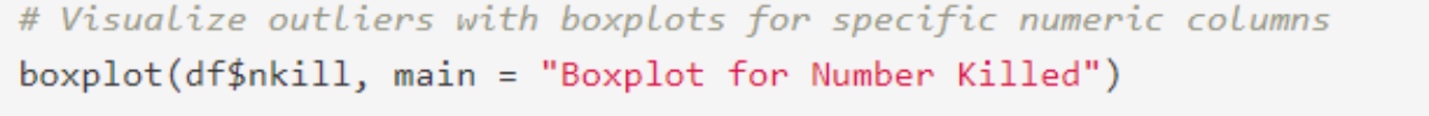
Removing those duplicates from the data.

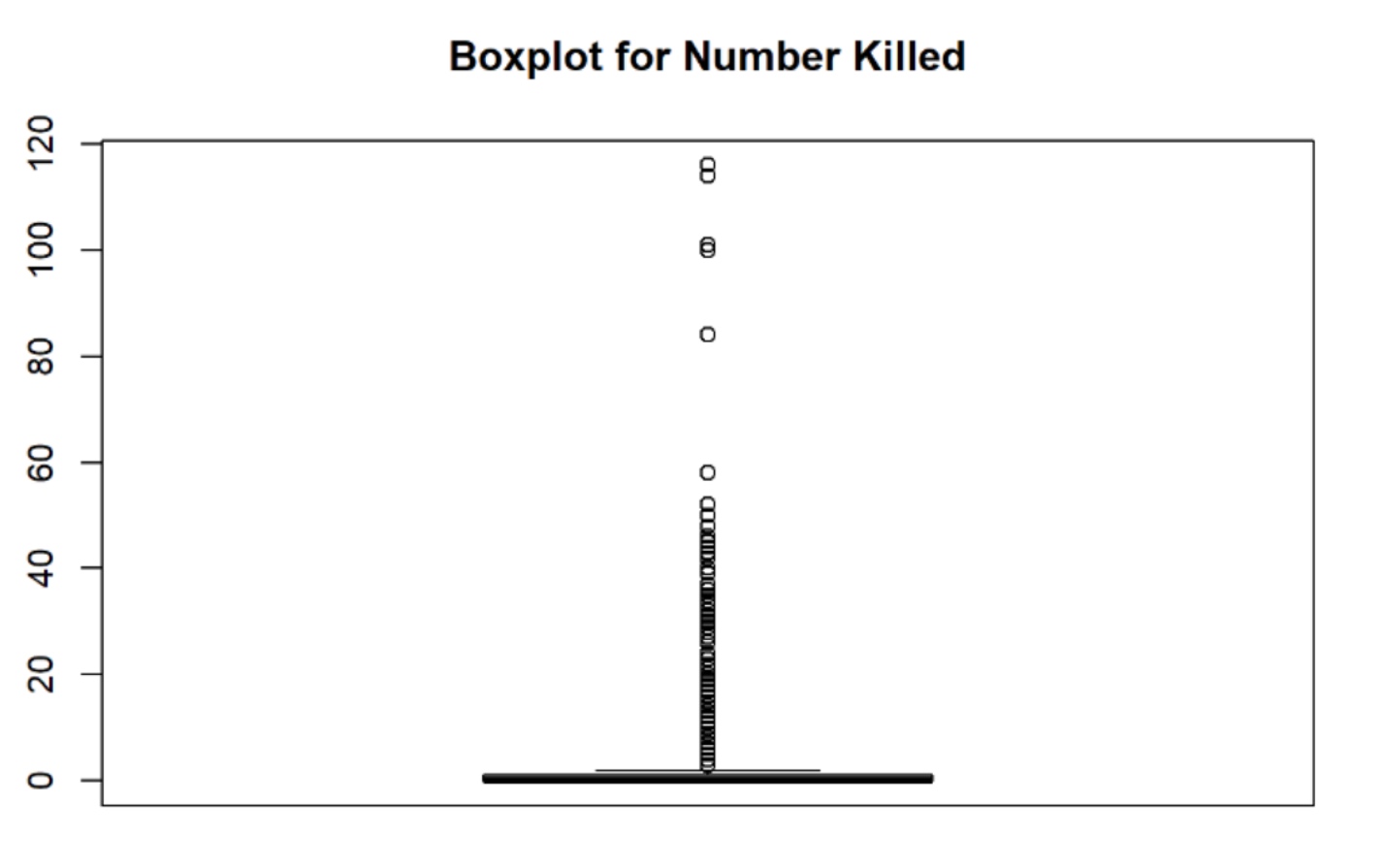
**OUTLIER DETECTION:**



Boxplots provide a visual representation of outliers based on the interquartile range (IQR).

**BOXPLOT FOR VISUALIZATING NUMBER KILLED:**



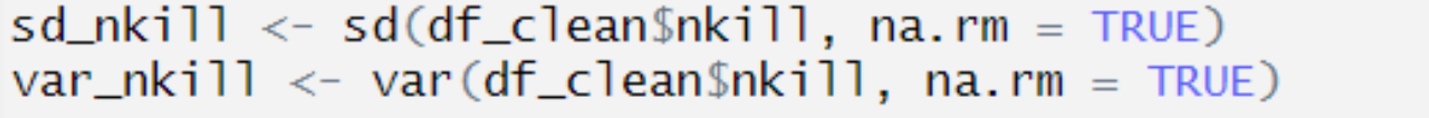


**BOXPLOT FOR NUMBER WOUNDED:**

A box plot for number wounded

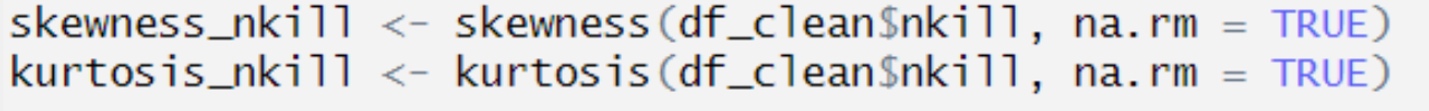
Description automatically generated

**DESCRIPTIVE STATISTICS:**



Calculate standard deviation and variance for key numeric columns like nkill and nwound.

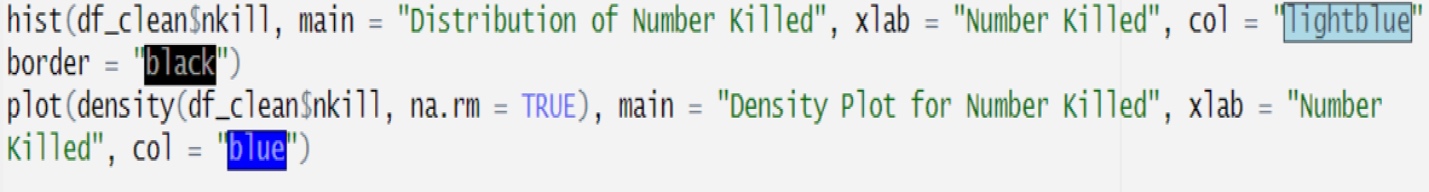
**SKEWNESS AND KURTOSIS:**



Measuring the skewness (asymmetry) and kurtosis (tailedness) of distributions to understand how data is distributed.

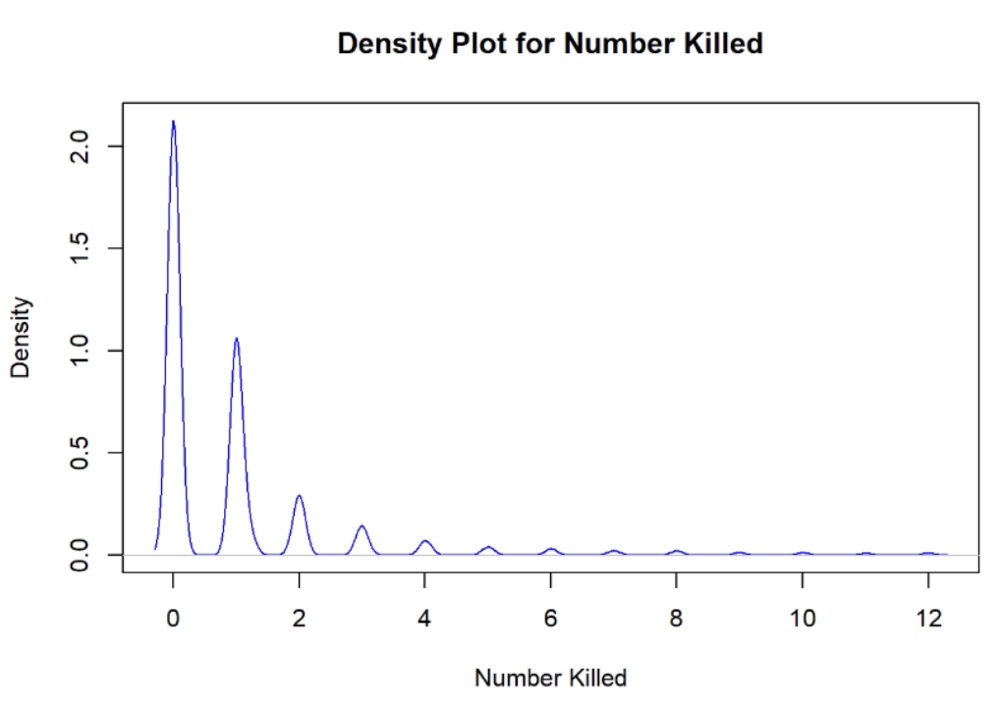
**DATA VISUALIZATION:**

**Histograms and Density Plots:**

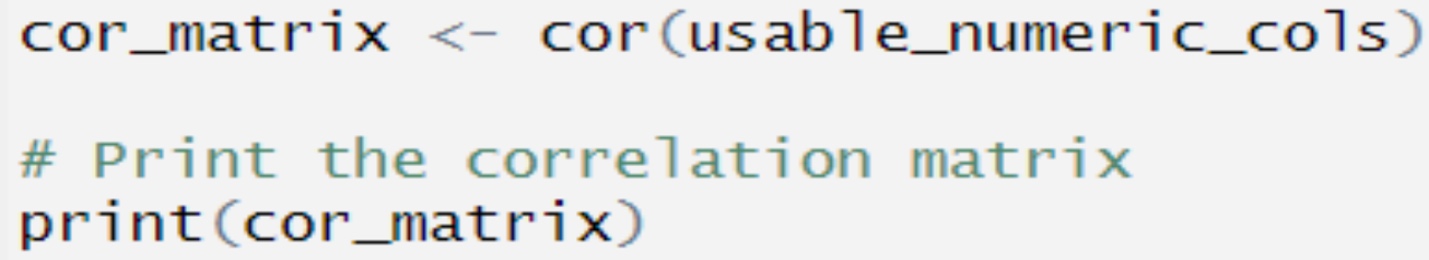


A graph of a number of killed

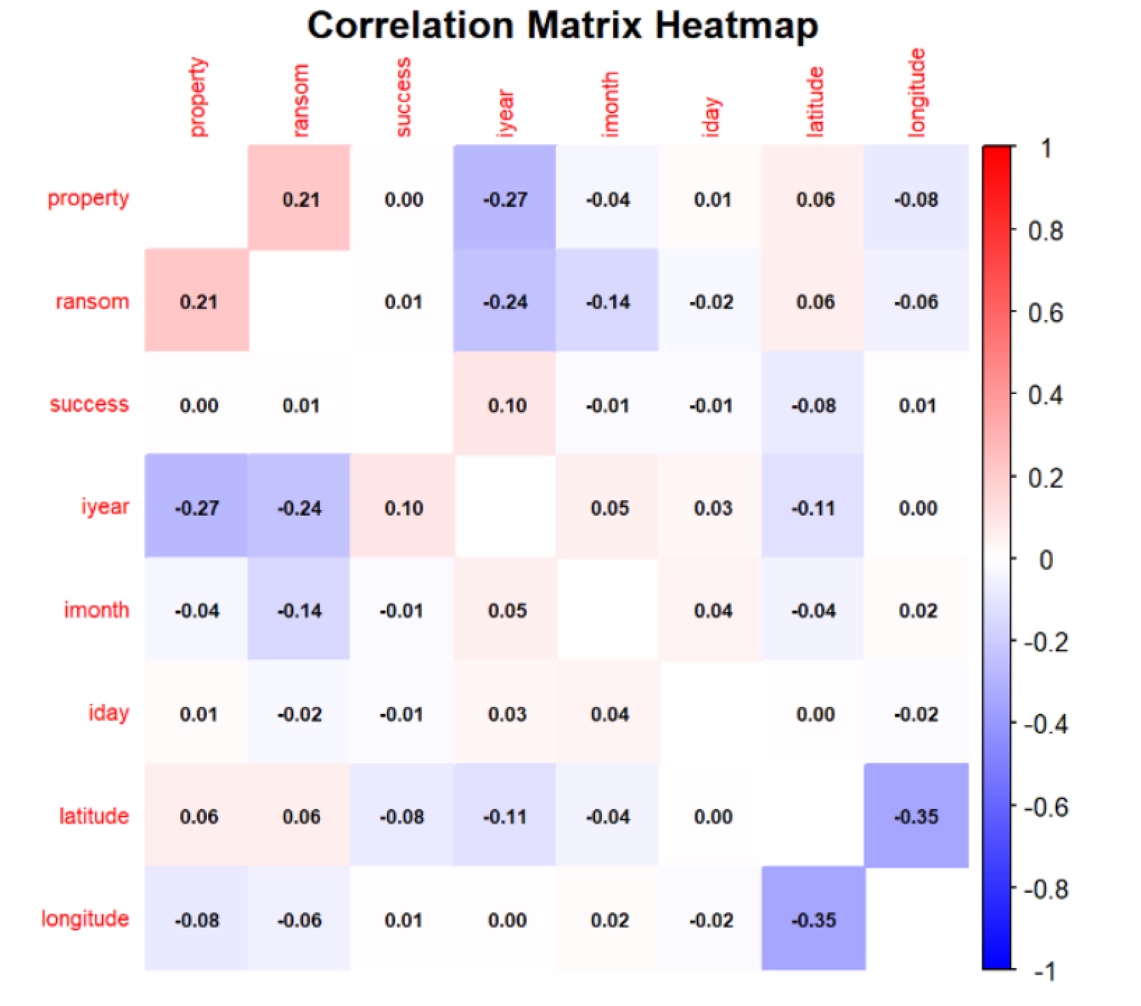
Description automatically generated



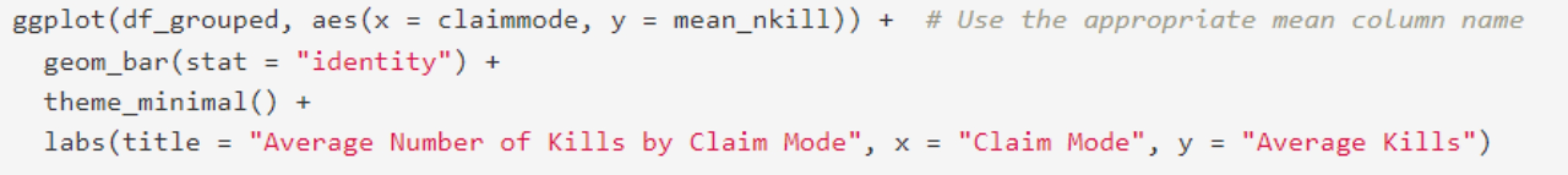
**Correlation Matrix:**

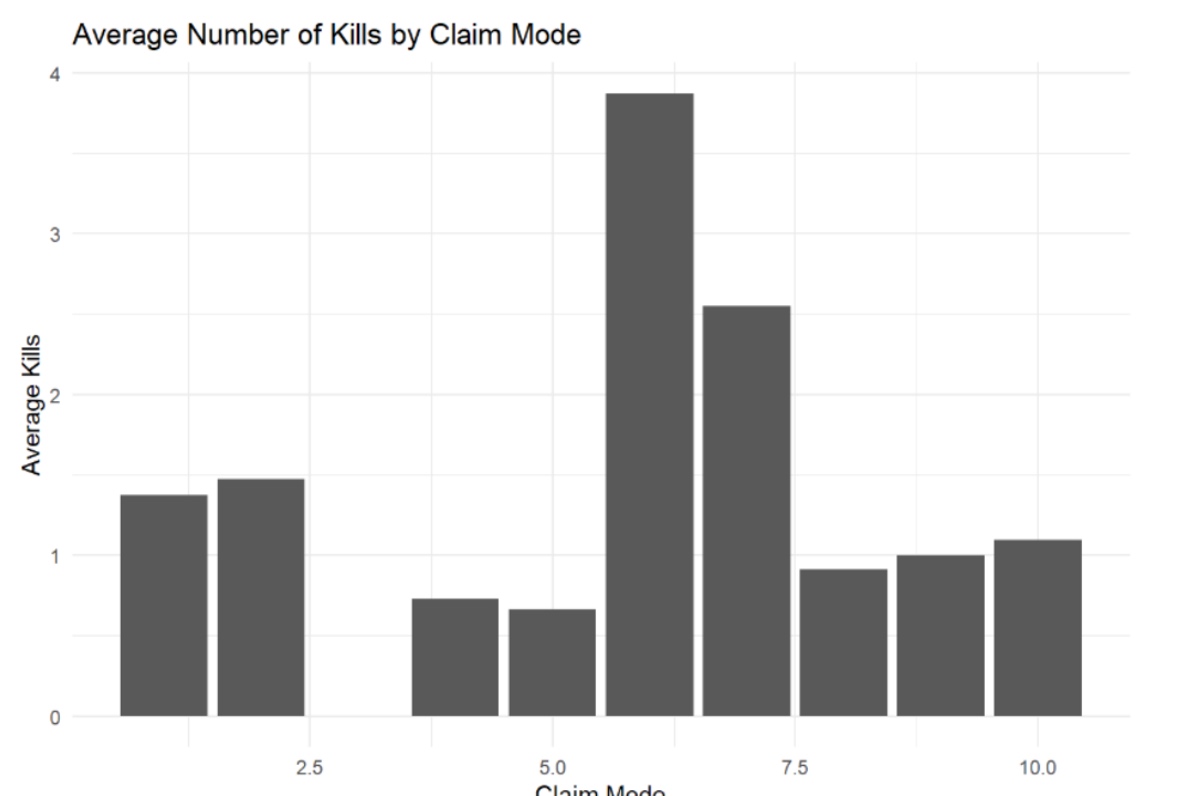


Calculate and visualize the correlation matrix of selected numeric columns using a heatmap.



**Bar Plot:**



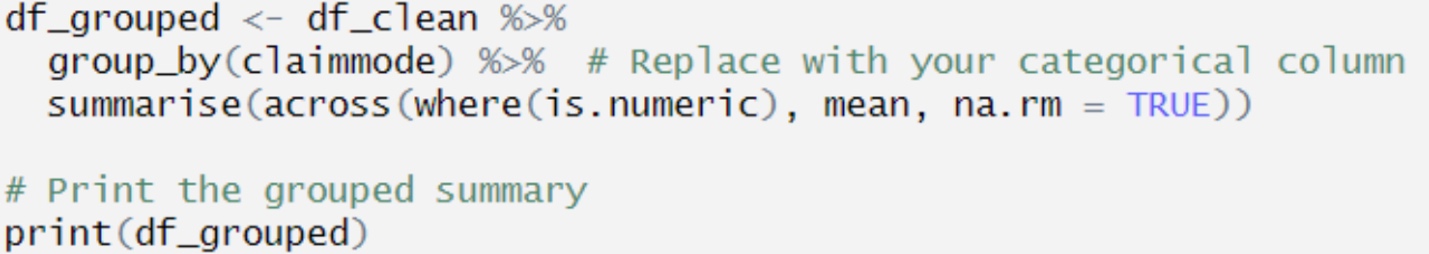


**Pair Plot:**



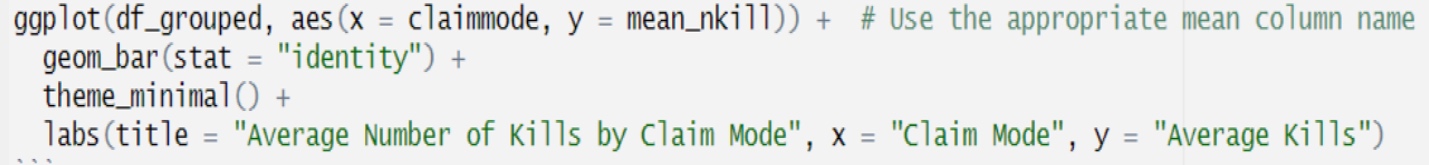
Creating pair plots (scatter plots with correlations) to explore relationships between numeric columns

**GROUPED DATA ANALYSIS:**



Group the dataset by a categorical variable (e.g., claimmode) and compute summary statistics (e.g., mean) for each group.

**PLOTTING GROUPED DATA:**



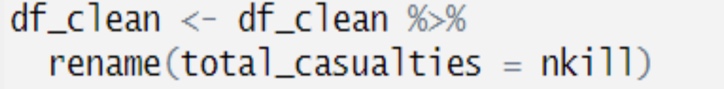
Creating bar plots to visualize the relationship between a categorical variable (e.g., claimmode) and the mean of numeric columns (e.g., nkill).

**SAVING CLEANED DATA:**



Saving the cleaned dataset as a CSV file for future use or sharing.

**HANDLING COLUMN NAMES AND RENAMING:**



Renaming columns for clarity or consistency (e.g., renaming nkill to total\_casualties).

**CONCLUSION:**

In conclusion, this project has encompassed a thorough analysis of a dataset, emphasizing the exploration of relationships among various features and the target variable. We employed a blend of data manipulation, exploration, and visualization techniques to extract meaningful insights from the data.

Both Python and R emerged as powerful tools for this analysis, each with unique strengths and weaknesses. Python is known for its versatility, ease of use, and strong integration capabilities, making it suitable for general programming and production applications. Its extensive libraries support tasks related to data manipulation, machine learning, and visualization.

In contrast, R excels in statistical analysis and data visualization, offering a robust suite of specialized packages. It is particularly valued in academic and research settings for its advanced statistical functionalities.

The selection between Python and R ultimately depends on project requirements, user expertise, and specific analytical tasks. Mastery of either language is invaluable in data science, enabling practitioners to uncover significant insights and enhance their analyses.