

MILITARY ARSENAL DATA ANALYSIS

Summer Internship Report Submitted in partial fulfillment of the requirement
for undergraduate degree

Bachelor of Technology

In

Computer Science and Engineering
(Specialized in Data Science)

By

Yatarla Tharun Reddy

(222010326020)



Department of Computer Science Engineering
GITAM School of Technology
GITAM (Deemed to be University)
Hyderabad-502329
September 2024

DECLARATION

I submit this industrial training work entitled “**MILITARY ARSENAL DATA ANALYSIS**” to GITAM (Deemed To Be University), Hyderabad, in partial fulfillment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer Science and Engineering (Specialized in Data Science)**”. I declare that it was carried out independently by me under the guidance of **Dr. K.V. Rajani**, Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

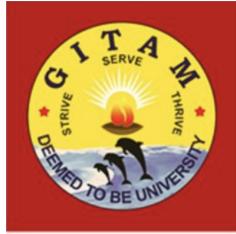
The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD

Name: Yatarla Tharun Reddy

Date: 21-07-2023

Student Roll No: 222010326020



GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India

Dated: 07-09-2023

CERTIFICATE

This is to certify that the Industrial Training Report entitled "**Military Arsenal Data Analysis**" is being submitted by Yatarla Tharun Reddy (222010326020) in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science And Engineering** at GITAM (Deemed to Be University), Hyderabad, during the academic year 2023-2024.

It is a faithful record of work carried out by him at the **Computer Science And Engineering Department**, GITAM University Hyderabad Campus, under my guidance and supervision.

Dr.K.V.Rajani

Assistant Professor
Department of CSE

Dr. Sudeep K S

Professor and HOD
Department of CSE

CERTIFICATE OF COMPLETION

दूरभाष - ०८४५८-२६३१०२

Tel No: 08458-273102

फैक्स- ०८४५८-२६३१००

Fax No: 08458-273100

ई-मेल: director.dysl-at@gov.in

Email: director.dysl-at@gov.in



भारत सरकार, रक्षा मंत्रालय

Govt. of India, Ministry of Defence

रक्षा अनुसंधान एवं विकास संगठन

Defence Research & Development Orgn.

असमिति प्रौद्योगिकी

Asymmetric Technologies

डीआरडीओ युवा वैज्ञानिक प्रयोगशाला

DRDO Young Scientists' Laboratories

ऑरेंज टेस्ट फैसिलिटी, डुंडीगल,

हैदराबाद - ५०००४३

Orange Test Facility, RCI, Dundigal,
Hyderabad-500043

NO: DYSL-AT/ADMIN /INTERNSHIP/2023-24/16

Date: 08/09/2023

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Yatarla Tharun Reddy** studying BTech in Department of Computer Science Engineering (B. Tech 4th year) in GITAM University, Hyderabad, Telangana-502329 bearing **Roll No.222010326020**, has completed his Internship in DRDO Young Scientist Laboratory Asymmetric Technologies (DYSL-AT), Dundigal, Hyderabad from 03rd May 2023 to 30th June 2023.

He has worked on "**MILITARY ARSENAL DATA ANALYSIS**" during the above period under the guidance of **Shri. Phani Kiran Maddukuri, Scientist 'D'**.

(P SHIVA PRASAD)
DIRECTOR
DYSL-AT, HYDERABAD

ACKNOWLEDGMENT

My project would not have been successful without the help of several people. We would like to thank the personalities who were part of our project in numerous ways, including those who gave us outstanding support from the birth of the project.

We are extremely thankful to our honorable Pro-Vice Chancellor, **Dr. D. Sambasiva rao** for providing the necessary infrastructure and resources for the accomplishment of our project.

We are highly indebted to **Prof. N. Seetharamaiah**, Principal, School of Technology, for his support during the tenure of the project.

We are very grateful to our beloved **Dr. K.S. Sudeep**, Head of the Department of Computer Science and Engineering, for providing the opportunity to undertake this project and encouragement in its completion.

We hereby wish to express our deep sense of gratitude to **Dr. K.V. Rajani**, GITAM School of Technology, for the esteemed guidance, moral support, and invaluable advice provided by her for the success of the project.

We are also thankful to all the staff members of the Computer Science and Engineering department who have cooperated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement, and moral support either directly or indirectly in our project work.

Yatarla Tharun Reddy

222010326020

Abstract

In an ever-evolving global landscape, the intricate tapestry of national security and defense strategies plays a pivotal role in shaping the dynamics of international relations. The imperative of understanding the military capabilities of nations cannot be overstated. This project embarks on a comprehensive journey into the realm of global military power, harnessing the power of data and sophisticated analytical tools. Our study leverages a rich and expansive dataset that encapsulates the military arsenals of 144 countries, providing a panoramic view of military might across the globe.

The central objective of this analysis is to unveil the nuanced intricacies of each country's military prowess, furnishing users with the ability to extract, manipulate, visualize, and glean profound insights from this vast reservoir of data. The project wields the formidable capabilities of Python programming, reinforced by essential libraries such as Pandas, NumPy, and Matplotlib, to bring this trove of military data to life.

This project offers a multifaceted suite of functionalities that empower users in the realm of global military analysis. It encompasses data presentation, comparative assessments of military strength between two countries, rigorous power analysis, and immersive data visualization. Users are afforded the opportunity to delve into the military capabilities of different nations, conduct side-by-side comparisons, and draw valuable insights that contribute to a deeper understanding of the world's military dynamics.

The codebase presented herein serves as a Python-based framework, meticulously engineered to facilitate the retrieval and comprehensive analysis of military arsenal data. It provides a robust set of options for data presentation, comparison, and visualization, equipping users with the tools they need to make informed decisions and strategic assessments in the complex domain of global military analysis. This project stands as a testament to the power of data-driven insights in navigating the complexities of our ever-changing world.

With a focus on transparency and accessibility, this project is poised to serve as an invaluable resource for policymakers, researchers, and analysts seeking to navigate the intricacies of global military dynamics. Through meticulous analysis and innovative data visualization, this project contributes to a deeper comprehension of the factors that shape international security and defense strategies.

Keywords: Military Arsenal, Data Analysis, Global Military Power, Python Programming, Data Visualization, Comparative Analysis, National Security, Defense Strategies, International Relations.

Introduction:

In an ever-changing world where national security and defense strategies play pivotal roles in shaping international relations, understanding the military capabilities of nations is of paramount importance. This report embarks on a journey into the realm of global military power, leveraging a rich dataset encompassing military arsenals from 144 countries. The dataset delves into the core aspects of military might, covering air force, navy, and ground forces, and offers a nuanced view of the world's military landscape.

The objective of this analysis is to unveil the intricate details of each country's military prowess, enabling users to extract, manipulate, visualize, and glean insights from this wealth of data. We have harnessed the power of Python programming and the capabilities of libraries like Pandas, NumPy, and Matplotlib to bring this data to life.

INTRODUCTION TO PYHTON:

- Python is a high-level, interpreted, interactive and object-oriented scripting language.
- Python is a general-purpose programming language that is often applied in scripting roles
- Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
- Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

HISTORY OF PYTHON:

- Python was developed by GUIDO VAN ROSSUM in early 1990's
- Its latest version is 3.7, it is generally called as python3

FEATURES OF PYTHON:

- Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax, This allows the student to pick up the language quickly.
- Easy-to-read: Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain: Python's source code is fairly easy-to-maintaining.
- A broad standard library: Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.
- Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases: Python provides interfaces to all major commercial databases.
- GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

HOW TO SETUP PYTHON:

- Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.
- The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

INSTALLING PYTHON:

Check for Existing Python Installation: Before installing Python, it's a good idea to check if Python is already installed on your computer. Open a terminal or command prompt and run the following command to check the installed Python version (if any):

If you see a version number, Python is already installed. You can proceed with the installation to update or install a different version if needed.

Download Python: To download Python, visit the official Python website at <https://www.python.org/downloads/>. On the website, you'll find the latest stable versions of Python available for different operating systems (Windows, macOS, or Linux).

Select Python Version: Choose the version of Python you want to install. It's recommended to use Python 3.x, as Python 2.x is no longer supported. Click on the download link for the Python 3.x version that matches your operating system.

The screenshot shows the Python.org website's download section. At the top, there's a navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. Below the navigation is a large banner with the Python logo and the text "Download the latest version for macOS". A yellow button labeled "Download Python 3.11.5" is prominently displayed. To the right of the button is an illustration of two boxes descending from the sky on parachutes. Below the banner, there's a section titled "Active Python Releases" with a sub-section for "For more information visit the Python Developer's Guide.". A table lists the active Python releases:

Python version	Maintenance status	First released	End of support	Release schedule
3.12	prerelease	2023-10-02 (planned)	2028-10	PEP 693
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569

Start the Installation:

For Windows:

- Run the downloaded installer.
- Check the box that says "Add Python X.Y to PATH" (replace X.Y with your Python version).
- Click the "Install Now" button.

For macOS and Linux:

- Open a terminal.
- Navigate to the directory where the downloaded installer is located.
- Run the installer using a command like `sudo ./python-3.x.x-macosx.pkg` or `sudo ./Python-3.x.x.tgz`.

Complete the Installation:

Follow the installation wizard's prompts, which may include agreeing to the license terms and specifying the installation location.

The installer will copy Python files to your computer.

Verify the Installation:

Open a new terminal or command prompt.

Run the following command to verify that Python has been successfully installed and is accessible from the command line:

```
python --version
```

You should see the version number of the installed Python.

Install Python Packages (Optional):

You can use the pip tool (Python's package manager) to install additional Python packages or libraries. For example:

```
pip install numpy pandas matplotlib
```

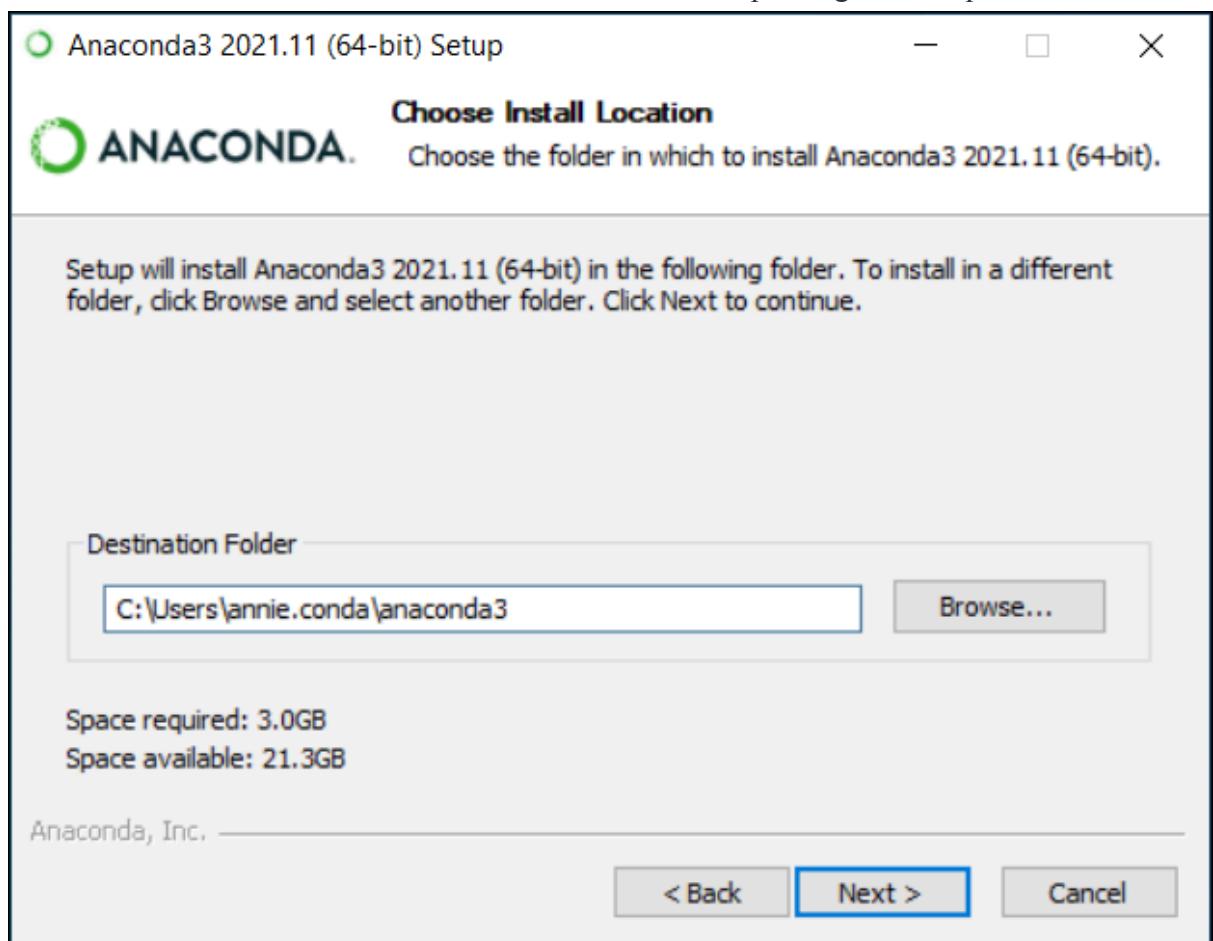
Anaconda Installation

1. [Download the Anaconda installer](#) from the link beside .
2. Go to your Downloads folder and double-click the installer to launch. To prevent permission errors, do not launch the installer from the [Favorites folder](#).

Note : If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you installed for all users, uninstall Anaconda and re-install it for your user only.

3. Click Next.
4. Read the licensing terms and click I Agree.

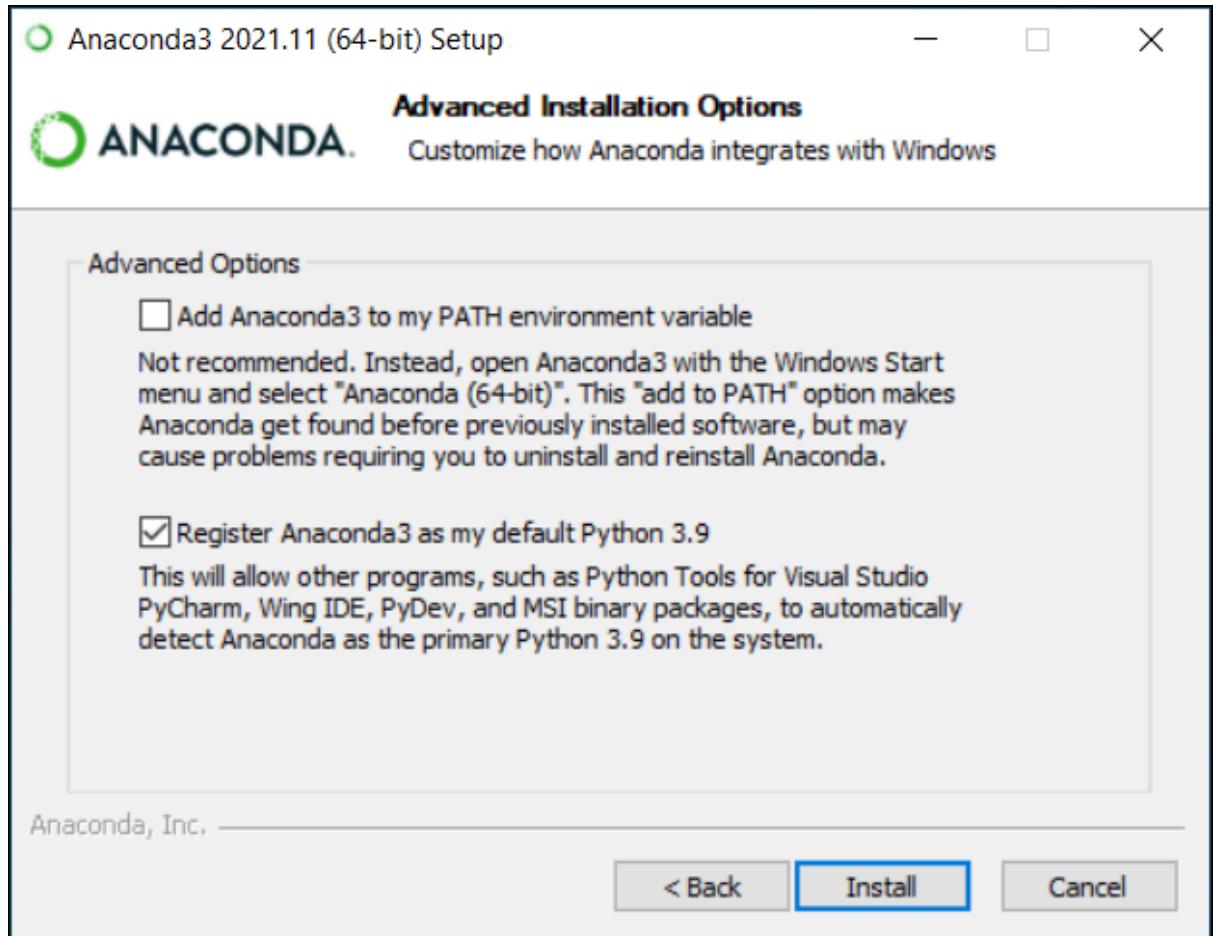
5. It is recommended that you install for Just Me, which will install Anaconda Distribution to just the current user account. Only select an install for All Users if you need to install for all users' accounts on the computer (which requires Windows Administrator privileges).
6. Click Next.
7. Select a destination folder to install Anaconda and click Next. Install Anaconda to a directory path that does not contain spaces or unicode characters. For more information on destination folders, see the [FAQ](#).
8. **Caution :** Do not install as an administrator unless admin privileges are required.



9. Choose whether to add Anaconda to your PATH environment variable or register Anaconda as your default Python. We don't recommend adding Anaconda to your PATH environment variable, since this can interfere with other software. Unless you plan on installing and running multiple versions of Anaconda or multiple versions of Python, accept the default and leave this box checked. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

Note :

As of Anaconda Distribution 2022.05, the option to add Anaconda to the PATH environment variable during an All Users installation has been disabled. This was done to address [a security exploit](#). You can still add Anaconda to the PATH environment variable during a Just Me installation.



10. Click Install. If you want to watch the packages Anaconda is installing, click Show Details.

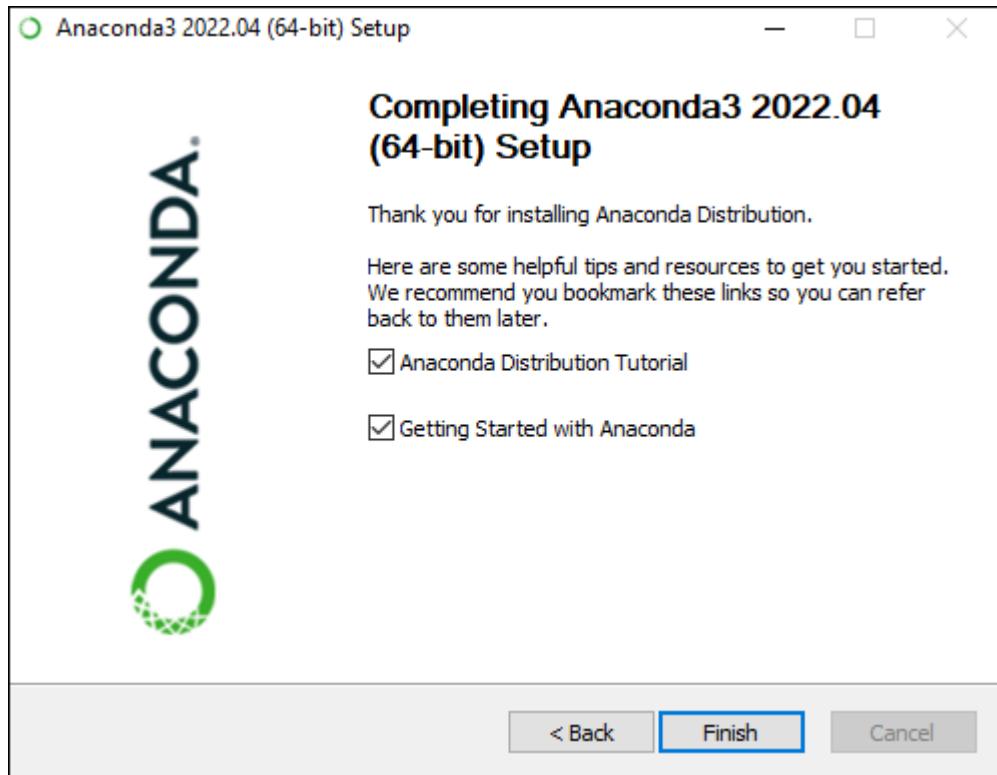
11. Click Next.

12. Optional: To learn more about Anaconda's cloud notebook service, go to <https://www.anaconda.com/code-in-the-cloud>.

Or click Continue to proceed.



13. After a successful installation, you will see the “Thanks for installing Anaconda” dialog box:



14. If you wish to read more about Anaconda.org and how to get started with Anaconda, check the boxes "Anaconda Distribution Tutorial" and "Learn more about Anaconda". Click the Finish button.
15. [Verify your installation](#).

Data Collection Section:

The data collection phase of this project involved the meticulous retrieval of military strength data from the website "Global Fire Power" (www.globalfirepower.com). The primary objective was to assemble a comprehensive dataset encompassing 144 countries, each categorized into various columns to provide a detailed overview of their military capabilities.

Data Source:

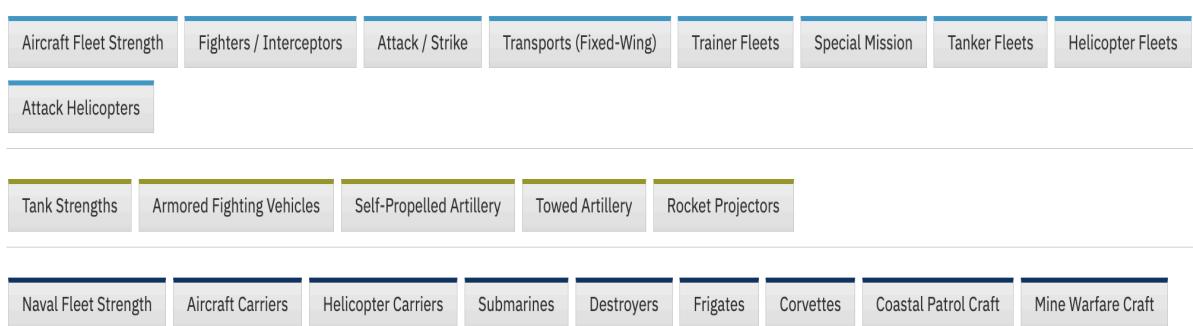
The primary data source for this project was the website "Global Fire Power," which is renowned for its extensive collection of military power rankings and statistics. The website provides detailed information about the military strengths of nations across the globe, making it an ideal resource for our analysis.

Data Variables:

The dataset comprises an array of variables, each meticulously curated to capture distinct aspects of a country's military capabilities. These variables include:

- 'Country': The name of the country under scrutiny.
- 'Continent': The continent to which the country belongs.
- 'World_ranking': The global ranking of the country's military power according to "Global Fire Power."
- 'Aircraft_fleet_strength': The total number of aircraft in the country's military fleet.
- 'Fighter_fleet_strength': The number of fighter aircraft.
- 'Attack_Aircraft_fleet_strength': The number of attack aircraft.
- 'Transport_fleet_strength': The quantity of transport aircraft.
- 'Trainer_Aircraft_fleet_strength': The number of trainer aircraft.
- 'SpecialMission_Aircraft_fleet_strength': The count of special mission aircraft.
- 'Tanker_fleet_strength': The number of tanker aircraft.
- 'Helicopter_fleet_strength': The total count of helicopters.

- 'Attack_Helicopter_fleet_strength': The number of attack helicopters.
- 'Ground_fleet_strength': The overall strength of ground forces.
- 'Combat_Tanks_fleet_strength': The quantity of combat tanks.
- 'Armored_Fighting_Vehicle_strength': The number of armored fighting vehicles.
- 'Self_Propelled_Artillery_strength': The count of self-propelled artillery.
- 'Towed_Artillery_strength': The number of towed artillery pieces.
- 'MLRS_Strength': The strength of multiple launch rocket systems.
- 'Naval_fleet_strength': The total number of naval assets.
- 'Aircraft_Carrier_fleet_strength': The count of aircraft carriers.
- 'Helicopter_Carrier_fleet_strength': The number of helicopter carriers.
- 'Submarine_fleet_strength': The quantity of submarines.
- 'Destroyer_fleet_strength': The number of destroyers.
- 'Frigate_fleet_strength': The count of frigates.
- 'Naval_Corvette_fleet_strength': The number of naval corvettes.
- 'Offshore_Patrol_Vessel_fleet_strength': The strength of offshore patrol vessels.
- 'Mine_Warfare_fleet_strength': The number of mine warfare vessels.



Data Collection Process:

The data collection process involved the following steps:

Accessing the Website: Access to the "Global Fire Power" website was established to access the military strength data.

Data Cleaning: The extracted data underwent rigorous cleaning to address inconsistencies, missing values, and formatting issues. This step ensured data accuracy and reliability.

Data Validation: To maintain data integrity, validation checks were applied to verify the accuracy of the collected information.

Storage: The cleaned and validated data were stored in a structured format suitable for analysis, with each country's information organized into the specified columns.

The meticulousness of the data collection process ensures that the dataset used in this analysis is reliable, accurate, and ready for in-depth examination of global military capabilities. This robust dataset serves as the foundation for the subsequent data analysis and exploration presented in this report.

	A	B	C	D	E	F	G	H	I
1	Country	Continent	World_ranking	Aircraft_fleet_strength	Fighter_fleet_strength	Attack_Aircraft_fleet_strength	Transport_fleet_strength	Trainer_Aircraft_fleet_strength	SpecialMission_Aircraft_fleet_strength
2	Afghanistan	Asia	114	0	0	0	0	0	0
3	Albania	Europe	91	19	0	0	0	0	0
4	Algeria	Africa	26	547	90	22	60	87	10
5	Angola	Africa	55	302	71	18	33	61	2
6	Argentina	South America	28	228	24	8	24	70	13
7	Armenia	Asia	94	64	4	10	2	10	0
8	Australia	Australia	16	353	24	48	38	98	31
9	Austria	Europe	84	101	15	0	11	12	0
10	Azerbaijan	Asia	57	148	17	11	2	32	0
11	Bahrain	Asia	79	116	17	0	3	37	0
12	Bangladesh	Asia	40	204	44	0	17	75	3
13	Belarus	Europe	60	191	38	67	4	22	0
14	Belgium	Europe	68	106	44	0	6	36	0
15	Belize	North America	139	2	0	0	1	0	0
16	Benin	Africa	144	2	0	0	2	0	0
17	Bhutan	Asia	145	2	0	0	0	0	0
18	Bolivia	South America	77	64	0	0	14	20	0
19	Bosnia and Herzegovina	Europe	133	22	0	0	0	0	0
20	Botswana	Africa	124	46	11	0	10	9	0
21	Brazil	South America	12	665	42	76	125	209	37
22	Bulgaria	Europe	59	66	11	6	7	20	0
23	Burkina Faso	Africa	121	20	0	3	2	4	1
24	Cambodia	Asia	106	24	0	0	4	0	0
25	Cameroon	Africa	100	31	0	0	6	6	2
26	Canada	North America	27	376	63	0	28	132	27
27	Central African Republic	Africa	136	3	0	0	2	0	0
28	Chad	Africa	97	34	0	6	6	4	2
29	Chile	South America	46	289	45	0	32	106	20
30	China	Asia	3	3166	1199	371	288	397	112
31	Colombia	South America	43	465	17	38	83	85	28
32	Croatia	Europe	69	80	12	0	0	26	0
33	Cuba	North America	66	80	38	0	2	26	0
34	Czechia	Europe	48	97	12	16	10	17	2
35	Democratic Republic of the Congo	Africa	72	46	0	4	8	0	0
36	Denmark	Europe	50	115	33	4	4	37	3
37	Dominican Republic	North America	118	63	0	8	12	10	1

❖ Importing Libraries and Loading Data

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 data_frame = pd.read_excel('Users/tharunreddy/Downloads/Defence_Power.xlsx')

```

- This block imports the necessary Python libraries:
 - `pandas` (as `pd`) for data manipulation.
 - `numpy` (as `np`) for numerical operations.
 - `matplotlib.pyplot` (as `plt`) for data visualization.

- The script loads an Excel file containing military data into a Pandas DataFrame named `data_frame`. The path to the Excel file is specified in the `pd.read_excel()` function. This DataFrame serves as the primary data source for analysis.

❖ Data Presentation Class :

```

6
7 class data_Present :
8     def top_data(self,x = 5):
9         return data_frame.head(x)
10    def tail_data(self,x = 5):
11        return data_frame.tail(x)
12    def middle_data(self,x,y):
13        return data_frame[x:y]
14

```

- This block defines a Python class named `data_Present` that contains methods for presenting data from the `data_frame` DataFrame.
- `top_data(self, x=5)`: This method returns the top `x` rows of the DataFrame using the `head()` function.
- `tail_data(self, x=5)`: This method returns the bottom `x` rows of the DataFrame using the `tail()` function.
- `middle_data(self, x, y)`: This method returns rows from index `x` to `y` of the DataFrame using slicing.

❖ Data Presentation by Word Class :

```

14
15 class data_Present_by_word:
16     def country_name(self,x):
17         selected_rows = data_frame[data_frame['Country']==x]
18         print(selected_rows)
19     def country_name_by_ltr(self,x):
20         selected_rows = data_frame[data_frame['Country'].str.startswith(x,na=False)]
21         print(selected_rows)
22     def continent_name(self,x):
23         selected_rows = data_frame[data_frame['Continent']==x]
24         print(selected_rows)
25     def continent_name_by_ltr(self,x):
26         selected_rows = data_frame[data_frame['Continent'].str.startswith(x,na=False)]
27         print(selected_rows)
28

```

- This block defines a Python class named `data_Present_by_word` that allows users to retrieve data based on specific criteria.
- `country_name(self, x)`: Given a specific country name ('x'), this method displays rows where the 'Country' column matches the provided name.
- `country_name_by_ltr(self, x)`: Given a letter or a string ('x'), this method displays rows where the 'Country' column starts with the provided letter(s).
- `continent_name(self, x)`: Given a continent name ('x'), this method displays rows where the 'Continent' column matches the provided name.

- `continent_name_by_ltr(self, x)`: Given a letter or a string ('x'), this method displays rows where the 'Continent' column starts with the provided letter(s).

❖ Data Presentation by Rank Class :

```

28
29 class data_present_by_rank:
30     def top_ranks(self,x=5):
31         selected_rows = data_frame[data_frame['World_ranking'] <= x ]
32         selected_rows = selected_rows.sort_values('World_ranking')
33         print(selected_rows)
34     def least_ranks(self,x=5):
35         selected_rows = data_frame[data_frame['World_ranking'] > len(data_frame['World_ranking'])-x]
36         selected_rows = selected_rows.sort_values('World_ranking')
37         print(selected_rows)
38     def ranks_by_range(self,x,y):
39         selected_rows = data_frame[(data_frame['World_ranking'] >= x) & (data_frame['World_ranking'] <= y ) ]
40         selected_rows = selected_rows.sort_values('World_ranking')
41         print(selected_rows)
42

```

- This block defines a Python class named `data_present_by_rank` that allows users to retrieve data based on ranking criteria.

- `top_ranks(self, x=5)`: Given a maximum ranking value ('x'), this method displays rows where the 'World_ranking' column is less than or equal to the provided value. It then sorts the selected rows by 'World_ranking' and displays them.
- `least_ranks(self, x=5)`: Given a number ('x'), this method displays rows where the 'World_ranking' column is greater than the total number of countries minus 'x'. It sorts the selected rows by 'World_ranking' and displays them.
- `ranks_by_range(self, x, y)`: Given a range of ranking values ('x' and 'y'), this method displays rows where the 'World_ranking' column falls within the specified range. It sorts the selected rows by 'World_ranking' and displays them.

❖ Data Comparison Class :

```

43
44 class diff_cont :
45     def compare_countries(self, country1, country2):
46         country1_data = data_frame[data_frame['Country'] == country1]
47         country2_data = data_frame[data_frame['Country'] == country2]
48
49         if country1_data.empty or country2_data.empty:
50             print("One or both of the countries not found in the data.")
51             return None
52
53         comparison_data = pd.DataFrame(columns=['Column', country1, 'Difference', country2])
54
55         for column in data_frame.columns[3:] : # Exclude 'Country', 'Continent', and 'World_ranking'
56             country1_value = country1_data[column].values[0]
57             country2_value = country2_data[column].values[0]
58             difference = country1_value - country2_value
59
60             comparison_data = pd.concat([
61                 comparison_data,
62                 pd.DataFrame({
63                     'Column': [column],
64                     country1: [country1_value],
65                     'Difference': [difference],
66                     country2: [country2_value]
67                 })
68             ])
69
70         return comparison_data
71
72

```

- This block defines a Python class named `diff_cont` that facilitates the comparison of data between two countries.

`'compare_countries(self, country1, country2)'`: Given two country names ('country` and `country2`), this method retrieves data for both countries and calculates the difference between their values for each column (excluding 'Country', 'Continent', and 'World_ranking'). It then returns a DataFrame ('comparison_data') containing the column name, values for 'country1', the difference between the two countries, and values for 'country2'.

❖ War Analysis Class:

```

113 elif x == 4:
114     print('Please select the specified troops for country 1 from the below mentioned column names.')
115     print(data_frame.columns[3:12])
116     print(data_frame.columns[13:18])
117     print(data_frame.columns[19:])
118     y = int(input("Please specify how many columns you would like to compare: "))
119     for i in range(y):
120         y1 = input('Enter the column name: ')
121         country1_values.append(y1)
122
123 elif x== 5 :
124     country1_values.extend(data_frame.columns[3:12])
125     country1_values.extend(data_frame.columns[13:18])
126     country1_values.extend(data_frame.columns[19:])
127     print(country1_values)
128
129 initial_values_of_country1 = []
130 strength_values_of_country1 = []
131
132 for column in country1_values:
133     z1 = country1_data[column].values[0]
134     initial_values_of_country1.append(z1)
135     z2 = strength_values[column]
136     strength_values_of_country1.append(z1 * z2)
137
138 print(initial_values_of_country1)
139 print(strength_values_of_country1)
140
141 country2_values = []
142 print("Select from the below options for country 2 :")
143 print("1) Air Force")
144 print("2) Ground Force")
145 print("3) Naval Force")
146 print("4) Specified Selection")
147 print("5) Total Troops")
148
149 x2 = int(input())
150
151 if x2 == 1:
152     country2_values = ['Fighter_fleet_strength', 'Attack_Aircraft_fleet_strength', 'Transport_fleet_streng',
153                         'Trainer_Aircraft_fleet_strength', 'SpecialMission_Aircraft_fleet_strength',
154                         'Tanker_fleet_strength', 'Helicopter_fleet_strength', 'Attack_Helicopter_fleet_str
155
156 elif x2 == 2:
157     country2_values = ['Combat_Tanks_fleet_strength', 'Armored_Fighting_Vehicle_strength',
158                         'Self_Propelled_Artillery_strength', 'Towed_Artillery_strength', 'MLRS_Strength']
159
160 elif x2 == 3:
161     country2_values = ['Aircraft_Carrier_fleet_strength', 'Helicopter_Carrier_fleet_strength',
162                         'Submarin_fleet_strength', 'Destroyer_fleet_strength', 'Frigate_fleet_strength',
163                         'Navel_Corvette_fleet_strength', 'Offshore_Patrol_Vessel_fleet_strength',
164                         'Mine_Warfare_fleet_strength']
165
166 elif x2 == 4:
167     print('Please select the specified troops for country 2 from the below mentioned column names.')
168     print(data_frame.columns[3:12])
169     print(data_frame.columns[13:18])
170     print(data_frame.columns[19:])
171     y2 = int(input("Please specify how many columns you would like to compare: "))
172     for i in range(y2):
173         y3 = input('Enter the column name: ')
174         country2_values.append(y3)
175
176 elif x2==5 :
177     country2_values.extend(data_frame.columns[3:12])
178     country2_values.extend(data_frame.columns[13:18])
179     country2_values.extend(data_frame.columns[19:])
180     print(country1_values)
181
182 initial_values_of_country2 = []
183 strength_values_of_country2 = []
184
185 for column in country2_values:
186     z3 = country2_data[column].values[0]
187     initial_values_of_country2.append(z3)
188     z4 = strength_values[column]
189     strength_values_of_country2.append(z3 * z4)
190
191 print(initial_values_of_country2)
192 print(strength_values_of_country2)
193
194 total_strength_country1 = sum(strength_values_of_country1)
195 total_strength_country2 = sum(strength_values_of_country2)
196
197 labels = [country1,country2]
198
199 # Create a list of the total strength values for country 1 and country 2
200 strength_values_total = [total_strength_country1, total_strength_country2]
201
202 # Create the pie chart
203 plt.pie(strength_values_total, labels=labels, autopct='%1.1f%%')
204 plt.title('Winning Chances')
205
206 # Display the chart
207 plt.show()

```

- This block defines a Python class named `war_analysis` that performs an analysis of the potential outcome of a conflict between two countries.

- `war_btwn_two(self, country1, country2)`: Given two country names (`country` and `country2`), this method calculates the "winning chances" based on a set of strength values for each country's military assets. Users are prompted to select specific categories of military

assets for comparison. The method then calculates the total strength for each country, creates a pie chart to visualize the winning chances, and displays it.

❖ War Visualization Class:

```
206 class war_visualization:
207     def diff_btwn_country (self,country1,country2):
208         country1_data = data_frame[data_frame['Country'] == country1]
209         country2_data = data_frame[data_frame['Country'] == country2]
210
211         if country1_data.empty or country2_data.empty:
212             print("One or both of the countries not found in the data.")
213             return None
214
215         comparison_data = pd.DataFrame(columns=['Column', country1, country2])
216
217         columns = data_frame.columns[3:] # Exclude 'Country', 'Continent', and 'World_ranking'
218         print("Available columns for comparison: ", columns)
219
220         num_parameters = input("Enter the number of parameters to compare (up to 4): ")
221         num_parameters = min(int(num_parameters), 4)
222
223         for _ in range(num_parameters):
224             column = input("Enter the column name to compare: ")
225             if column in columns:
226                 country1_value = country1_data[column].values[0]
227                 country2_value = country2_data[column].values[0]
228
229                 comparison_data = pd.concat([
230                     comparison_data,
231                     pd.DataFrame({
232                         'Column': [column],
233                         country1: [country1_value],
234                         country2: [country2_value]
235                     })
236                 ])
237             else:
238                 print("Invalid column name:", column)
239
240         comparison_data['Difference'] = comparison_data[country1] - comparison_data[country2]
241
242         # Find the best country based on the maximum difference
243         best_country = country1 if comparison_data['Difference'].sum() > 0 else country2
244
245         # Generate stacked bar graph
246         x = np.arange(len(comparison_data))
247         plt.bar(x, comparison_data[country1], label=country1)
248         plt.bar(x, comparison_data[country2], bottom=comparison_data[country1], label=country2)
249
250         # Customize the plot
251         plt.xlabel('Parameters')
252         plt.ylabel('Values')
253         plt.title('Comparison of Parameters between Countries')
254         plt.xticks(x, comparison_data['Column'])
255         plt.legend()
256
257         # Display the plot
258         plt.show()
259
260     return comparison_data, best_country
```

- This block defines a Python class named `war_visualization` that visualizes the differences between two countries in terms of their military assets.

- `diff_btwn_country(self, country1, country2)`: Given two country names (`country1` and `country2`), this method retrieves data for both countries and allows users to select specific parameters for comparison. It then calculates the differences between the selected parameters and creates a stacked bar graph to visualize these differences. The method returns the comparison data and identifies the "best" country based on the maximum difference.

These classes and methods provide a structured way to interact with and analyze the military data contained in the data frame. Users can retrieve, compare, and visualize military information for specific countries, ranks, and scenarios using this script.

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 data_frame = pd.read_excel('/Users/tharunreddy/Downloads/Defence_Power.xlsx')
6
7 class data_Present :
8     def top_data(self,x = 5):
9         return data_frame.head(x)
10    def tail_data(self,x = 5):
11        return data_frame.tail(x)
12    def middle_data(self,x,y):
13        return data_frame[x:y]
14
15 class data_Present_by_word:
16     def country_name(self,x):
17         selected_rows = data_frame[data_frame['Country'] == x]
18         print(selected_rows)
19     def country_name_by_ltr(self,x):
20         selected_rows = data_frame[data_frame['Country'].str.startswith(x,na=False)]
21         print(selected_rows)
22     def continent_name(self,x):
23         selected_rows = data_frame[data_frame['Continent'] == x]
24         print(selected_rows)
25     def continent_name_by_ltr(self,x):
26         selected_rows = data_frame[data_frame['Continent'].str.startswith(x,na=False)]
27         print(selected_rows)
28
29 class data_present_by_rank:
30     def top_ranks(self,x=5):
31         selected_rows = data_frame[data_frame['World_ranking'] <= x ]
32         selected_rows = selected_rows.sort_values('World_ranking')
33         print(selected_rows)
34     def least_ranks(self,x=5):
35         selected_rows = data_frame[data_frame['World_ranking'] > len(data_frame['World_ranking'])-x]
36         selected_rows = selected_rows.sort_values('World_ranking')
37         print(selected_rows)
38     def ranks_by_range(self,x,y):
39         selected_rows = data_frame[(data_frame['World_ranking'] >= x) & (data_frame['World_ranking'] <= y ) ]
40         selected_rows = selected_rows.sort_values('World_ranking')
41         print(selected_rows)
42
43
44 class diff_cont :
45     def compare_countries(self, country1, country2):
46         country1_data = data_frame[data_frame['Country'] == country1]
47         country2_data = data_frame[data_frame['Country'] == country2]
48
49         if country1_data.empty or country2_data.empty:
50             print("One or both of the countries not found in the data.")
51             return None
52
53         comparison_data = pd.DataFrame(columns=['Column', country1, 'Difference', country2])
54
55         for column in data_frame.columns[3:] : # Exclude 'Country', 'Continent', and 'World_ranking'
56             country1_value = country1_data[column].values[0]
57             country2_value = country2_data[column].values[0]
58             difference = country1_value - country2_value
59
60             comparison_data = pd.concat([
61                 comparison_data,
62                 pd.DataFrame({
63                     'Column': [column],
64                     country1: [country1_value],
65                     'Difference': [difference],
66                     country2: [country2_value]
67                 })
68             ])
69
70         return comparison_data
71
72
73
74 class war_analysis :
75     def war_btwn_two(self,country1,country2):
76         country1_data = data_frame[data_frame['Country'] == country1]
77         country2_data = data_frame[data_frame['Country'] == country2]
78         strength_values = {'Aircraft_fleet_strength': 8, 'Fighter_fleet_strength': 9,
79                            'Attack_Aircraft_fleet_strength': 7, 'Transport_fleet_strength': 7,
80                            'Trainer_Aircraft_fleet_strength': 6, 'SpecialMission_Aircraft_fleet_strength': 8,
81                            'Tanker_fleet_strength': 8, 'Helicopter_fleet_strength': 9,
82                            'Attack_Helicopter_fleet_strength': 9, 'Ground_fleet_strength': 8,
83                            'Combat_Tanks_fleet_strength': 9, 'Armored_Fighting_Vehicle_strength': 8,
84                            'Self_Propelled_Artillery_strength': 7, 'Towed_Artillery_strength': 6,
85                            'MLRS_Strength': 8, 'Navel_fleet_strength': 9, 'Aircraft_Carrier_fleet_strength': 9,
86                            'Helicopter_Carrier_fleet_strength': 7, 'Submarin_fleet_strength': 8,
87                            'Destroyer_fleet_strength': 9, 'Frigate_fleet_strength': 8,
88                            'Navel_Corvette_fleet_strength': 7, 'Offshore_Patrol_Vessel_fleet_strength': 6,
89                            'Mine_Warfare_fleet_strength': 7}
90
91         country1_values = []
92         print("Select from the below options for country 1 :")
93         print("(1) Air Force")
94         print("(2) Ground Force")
95         print("(3) Naval Force")
96         print("(4) Specified Selection")
97         print("(5) Total Troops")
98
99         x = int(input())
100
101        if x == 1:
102            country1_values = ['Fighter_fleet_strength', 'Attack_Aircraft_fleet_strength', 'Transport_fleet_strength',
103                               'Trainer_Aircraft_fleet_strength', 'SpecialMission_Aircraft_fleet_strength',
104                               'Tanker_fleet_strength', 'Helicopter_fleet_strength', 'Attack_Helicopter_fleet_strengt

```

```

105     elif x == 2:
106         country1_values = ['Combat_Tanks_fleet_strength', 'Armored_Fighting_Vehicle_strength',
107                             'Self_Propelled_Artillery_strength', 'Towed_Artillery_strength', 'MLRS_Strength']
108     elif x == 3:
109         country1_values = ['Aircraft_Carrier_fleet_strength', 'Helicopter_Carrier_fleet_strength',
110                             'Submarin_fleet_strength', 'Destroyer_fleet_strength', 'Frigate_fleet_strength',
111                             'Navel_Corvette_fleet_strength', 'Offshore_Patrol_Vessel_fleet_strength',
112                             'Mine_Warfare_fleet_strength']
113     elif x == 4:
114         print('Please select the specified troops for country 1 from the below mentioned column names.')
115         print(data_frame.columns[3:12])
116         print(data_frame.columns[13:18])
117         print(data_frame.columns[19:])
118         y = int(input("Please specify how many columns you would like to compare: "))
119         for i in range(y):
120             y1 = input('Enter the column name: ')
121             country1_values.append(y1)
122     elif x== 5 :
123         country1_values.extend(data_frame.columns[3:12])
124         country1_values.extend(data_frame.columns[13:18])
125         country1_values.extend(data_frame.columns[19:])
126         print(country1_values)
127
128
129 initial_values_of_country1 = []
130 strength_values_of_country1 = []
131
132 for column in country1_values:
133     z1 = country1_data[column].values[0]
134     initial_values_of_country1.append(z1)
135     z2 = strength_values[column]
136     strength_values_of_country1.append(z1 * z2)
137
138 print(initial_values_of_country1)
139 print(strength_values_of_country1)
140
141 country2_values = []
142 print("Select from the below options for country 2 :")
143 print("(1) Air Force")
144 print("(2) Ground Force")
145 print("(3) Naval Force")
146 print("(4) Specified Selection")
147 print("(5) Total Troops")
148
149 x2 = int(input())
150
151 if x2 == 1:
152     country2_values = ['Fighter_fleet_strength', 'Attack_Aircraft_fleet_strength', 'Transport_fleet_streng',
153                         'Trainer_Aircraft_fleet_strength', 'SpecialMission_Aircraft_fleet_strength',
154                         'Tanker_fleet_strength', 'Helicopter_fleet_strength', 'Attack_Helicopter_fleet_str
155
156 elif x2 == 2:
157     country2_values = ['Combat_Tanks_fleet_strength', 'Armored_Fighting_Vehicle_strength',
158                         'Self_Propelled_Artillery_strength', 'Towed_Artillery_strength', 'MLRS_Strength']
159
160 elif x2 == 3:
161     country2_values = ['Aircraft_Carrier_fleet_strength', 'Helicopter_Carrier_fleet_strength',
162                         'Submarin_fleet_strength', 'Destroyer_fleet_strength', 'Frigate_fleet_strength',
163                         'Navel_Corvette_fleet_strength', 'Offshore_Patrol_Vessel_fleet_strength',
164                         'Mine_Warfare_fleet_strength']
165
166 elif x2 == 4:
167     print('Please select the specified troops for country 2 from the below mentioned column names.')
168     print(data_frame.columns[3:12])
169     print(data_frame.columns[13:18])
170     print(data_frame.columns[19:])
171     y2 = int(input("Please specify how many columns you would like to compare: "))
172     for i in range(y2):
173         y3 = input('Enter the column name: ')
174         country2_values.append(y3)
175
176 elif x2==5 :
177     country2_values.extend(data_frame.columns[3:12])
178     country2_values.extend(data_frame.columns[13:18])
179     country2_values.extend(data_frame.columns[19:])
180     print(country1_values)
181
182 initial_values_of_country2 = []
183 strength_values_of_country2 = []
184
185 for column in country2_values:
186     z3 = country2_data[column].values[0]
187     initial_values_of_country2.append(z3)
188     z4 = strength_values[column]
189     strength_values_of_country2.append(z3 * z4)
190
191 print(initial_values_of_country2)
192 print(strength_values_of_country2)
193
194 total_strength_country1 = sum(strength_values_of_country1)
195 total_strength_country2 = sum(strength_values_of_country2)
196
197 labels = [country1,country2]
198
199 # Create a list of the total strength values for country 1 and country 2
200 strength_values_total = [total_strength_country1, total_strength_country2]
201
202 # Create the pie chart
203 plt.pie(strength_values_total, labels=labels, autopct='%1.1f%')
204 plt.title('Winning Chances')
205
206 # Display the chart
207 plt.show()
208
209
210 class war_visualization:
211     def diff_btwn_country (self,country1,country2):
212         country1_data = data_frame[data_frame['Country'] == country1]
213         country2_data = data_frame[data_frame['Country'] == country2]

```

```

211     if country1_data.empty or country2_data.empty:
212         print("One or both of the countries not found in the data.")
213         return None
214
215     comparison_data = pd.DataFrame(columns=['Column', country1, country2])
216
217     columns = data_frame.columns[3:] # Exclude 'Country', 'Continent', and 'World_ranking'
218     print("Available columns for comparison: ", columns)
219
220     num_parameters = input("Enter the number of parameters to compare (up to 4): ")
221     num_parameters = min(int(num_parameters), 4)
222
223     for _ in range(num_parameters):
224         column = input("Enter the column name to compare: ")
225         if column in columns:
226             country1_value = country1_data[column].values[0]
227             country2_value = country2_data[column].values[0]
228
229             comparison_data = pd.concat([
230                 comparison_data,
231                 pd.DataFrame({
232                     'Column': [column],
233                     country1: [country1_value],
234                     country2: [country2_value]
235                 })
236             ])
237         else:
238             print("Invalid column name:", column)
239
240     comparison_data['Difference'] = comparison_data[country1] - comparison_data[country2]
241
242     # Find the best country based on the maximum difference
243     best_country = country1 if comparison_data['Difference'].sum() > 0 else country2
244
245     # Generate stacked bar graph
246     x = np.arange(len(comparison_data))
247     plt.bar(x, comparison_data[country1], label=country1)
248     plt.bar(x, comparison_data[country2], bottom=comparison_data[country1], label=country2)
249
250     # Customize the plot
251     plt.xlabel('Parameters')
252     plt.ylabel('Values')
253     plt.title('Comparison of Parameters between Countries')
254     plt.xticks(x, comparison_data['Column'])
255     plt.legend()
256
257     # Display the plot
258     plt.show()
259
260     return comparison_data, best_country

```

Accessing Code Block :

```

1  class Main:
2      def main(self):
3          print("-----DEFENCE POWER-----")
4          print("0:BACK")
5          print("1:DATA PRESENTATION")
6          print("2:DIFFERENCE BETWEEN THE POWER OF 2 SELECTED COUNTRIES")
7          print("3:POWER ANALYSIS BETWEEN TWO COUNTRIES")
8          print("4:VISUALIZATION BETWEEN THE SELECTED FIELDS OF THE TWO COUNTRIES")
9          x=int(input("Enter the specified option number : "))
10
11         if x==1:
12             print("0:BACK TO MAIN")
13             print("1:By Order ")
14             print("2:By Country or Continent")
15             print("3:By world rank")
16             x1=int(input("....."))
17             if x1==1:
18                 t=data_Present()
19                 print("0:BACK TO MAIN")
20                 print("1:To represent the data from the Top ")
21                 print("2:To represent the data from the Bottom")
22                 print("3:To represent the data based on the range between the specified numbers.")
23                 x11 = int(input("Enter the specified option number from the above : "))
24                 if x11==1:
25                     x12 = int(input("Enter the number of rows you would like to display from the top : "))
26                     t.top_data(x12)
27                 elif x11==2:
28                     x12 = int(input("Enter the number of rows you would like to display from the bottom : "))
29                     t.tail_data(x12)
30                 elif x11==3:
31                     print("Here we represent the data based on the range beweet the numbers given : ")
32                     x12 = int(input("Enter the number 1 : "))
33                     x13 = int(input("Enter the number 2 : "))
34                     t.middle_data(x12,x13)
35                 elif x11==0:
36                     self.main()
37                 else:
38                     print("ENTER VALID INPUT")
39                     self.main()
40
41             elif x1==2:
42                 t=data_Present_by_word()
43                 print("0:BACK TO MAIN")
44                 print("1:To repesent the data based on the name of the country.")
45                 print("2:To repesent the data based on the name of the continent.")
46                 x2=int(input("Enter the specified option number from the above : "))
47                 if x2==1:
48                     print("1:To represent the data based on the full name of the country")
49                     print("2:To represent the data based on the specified name of the country.")
50                     x21=int(input("Enter the specified option number from the above : "))
51                     if x21==1:

```

```

51             x211=input("Enter the name of the country : ")
52             t.country_name(x211)
53         elif x21==2:
54             x211=input("Enter the starting letters of the country.")
55             t.country_name_by_ltr(x211)
56         else:
57             print("ENTER VALID INPUT")
58     elif x2==2:
59         print("0:BACK TO MAIN")
60         print("1:To represent the data based on the full name of the continent.")
61         print("2:To represent the data based on the specified name of the continent.")
62         x21=int(input("Enter the specified option number from the above : "))
63         if x21==1:
64             x211=input("Enter the name of the continent : ")
65             t.continent_name(x211)
66         elif x21==2:
67             x211=input("Enter the starting letters of the country.")
68             t.continent_name_by_ltr(x211)
69         elif x21==0:
70             self.main()
71         else:
72             print("ENTER VALID INPUT")
73             self.main()
74     elif x2==0:
75         self.main()
76
77
78     elif x1==3:
79         t=data_present_by_rank()
80         print("0:BACK TO MAIN")
81         print("1:To represent the data from the top based on the world rank")
82         print("2:To represent the data from the bottom based on the world rank")
83         print("3:To represent the data based on the world ranking range between the specified numbers")
84         x11 = int(input("Enter the specified option number from the above : "))
85         if x11==1:
86             x12 = int(input("Enter the number of rows you would like to display from the top : "))
87             t.top_ranks(x12)
88         elif x11==2:
89             x12 = int(input("Enter the number of rows you would like to display from the bottom : "))
90             t.least_ranks(x12)
91         elif x11==3:
92             print("Here we represent the data based on the range between the numbers given : ")
93             x12 = int(input("Enter the number 1 : "))
94             x13 = int(input("Enter the number 2 : "))
95             t.ranks_by_range(x12,x13)
96         elif x11==0:
97             self.main()
98         else:
99             print("ENTER VALID INPUT")
100            self.main()
101
102     elif x1==0:
103         self.main()
104     else:
105         print("ENTER VALID INPUT")
106
107     elif x==2:
108         t=diff_cont()
109         print("Here we need to enter the names of the two countries :")
110         x1=input("Enter the name of the country 1 : ")
111         x2=input("Enter the name of the country 2 : ")
112         t.compare_countries(x1,x2)
113
114     elif x==3:
115         t=war_analysis()
116         print("Here we need to enter the names of the two countries :")
117         x1=input("Enter the name of the country 1 : ")
118         x2=input("Enter the name of the country 2 : ")
119         t.war_btwn_two(x1,x2)
120
121     elif x==4:
122         t=war_visualization()
123         print("Here we need to enter the names of the two countries :")
124         x1=input("Enter the name of the country 1 : ")
125         x2=input("Enter the name of the country 2 : ")
126         t.diff_btwn_country(x1,x2)
127
128     elif x==0:
129         self.main()
130     else:
131         print("ENTER VALID INPUT")
132         self.main()
133
134 x=Main()
135 x.main()

```

Data Visualizations :

```
1 data_frame
2
```

	Country	Continent	World_ranking	Aircraft_fleet_strength	Fighter_fleet_strength	Attack_Aircraft_fleet_strength	Transport_fleet_strength	Trainer_Aircraft_fleet_strength
0	Afghanistan	Asia	114	0	0	0.0	0	
1	Albania	Europe	91	19	0	0.0	0	
2	Algeria	Africa	26	547	90	22.0	60	
3	Angola	Africa	55	302	71	18.0	33	
4	Argentina	South America	28	228	24	8.0	24	
...
140	Venezuela	South America	52	250	44	0.0	64	
141	Vietnam	Asia	19	223	75	0.0	9	
142	Yemen	Asia	74	177	76	0.0	8	
143	Zambia	Africa	89	98	8	0.0	13	
144	Zimbabwe	Africa	98	90	10	0.0	14	

145 rows × 27 columns

```
1 a = data_Present()
2 print(a.top_data(3))
3 print(a.tail_data(3))
4 print(a.middle_data(143,144))

    Country Continent World_ranking Aircraft_fleet_strength \
0 Afghanistan Asia           114          0
1 Albania     Europe          91         19
2 Algeria     Africa          26        547

    Fighter_fleet_strength Attack_Aircraft_fleet_strength \
0                      0             0.0
1                      0             0.0
2                     90            22.0

    Transport_fleet_strength Trainer_Aircraft_fleet_strength \
0                      0              0
1                      0              0
2                     60             87

    SpecialMission_Aircraft_fleet_strength Tanker_fleet_strength ... \
0                      0                  0   ...
1                      0                  0   ...
2                     10                 5   ...
```

```
1 b=data_Present_by_word()
2 b.country_name('India')
3 b.country_name_by_ltr('A')
4 b.continent_name('Asia')
5 b.continent_name_by_ltr('Af')

    Country Continent World_ranking Aircraft_fleet_strength \
53 India     Asia           4           2210

    Fighter_fleet_strength Attack_Aircraft_fleet_strength \
53                   577                130.0

    Transport_fleet_strength Trainer_Aircraft_fleet_strength \
53                   254                 353

    SpecialMission_Aircraft_fleet_strength Tanker_fleet_strength ... \
53                   73                  6   ...

    MLRS_Strength Navel_fleet_strength Aircraft_Carrier_fleet_strength \
53                 1500                  295                  2

    Helicopter_Carrier_fleet_strength Submarin_fleet_strength \
53                           0                         18

    Destroyer_fleet_strength Frigate_fleet_strength \
```

```

1 | c = data_present_by_rank()
2 | c.top_ranks(6)
3 | c.least_ranks(4)
4 | c.ranks_by_range(15,19)

    Country      Continent  World_ranking  Aircraft_fleet_strength \
137  United States   North America        1                  13300
107  Russia          Asia            2                  4182
28   China            Asia            3                  3166
53   India             Asia            4                  2210
136  United Kingdom  Europe           5                  663
117  South Korea     Asia            6                  1602

  Fighter_fleet_strength  Attack_Aircraft_fleet_strength \
137                1914                      843.0
107                 773                      744.0
28                 1199                     371.0
53                 577                     130.0
136                 119                     26.0
117                 402                     98.0

  Transport_fleet_strength  Trainer_Aircraft_fleet_strength \
137                  962                      2634
107                  444                      524

```

```

1 | d= diff_cont()
2 | d.compare_countries('India','Uganda')

```

	Column	India	Difference	Uganda
0	Aircraft_fleet_strength	2210	2160	50
0	Fighter_fleet_strength	577	567	10
0	Attack_Aircraft_fleet_strength	130.0	130.0	0.0
0	Transport_fleet_strength	254	251	3
0	Trainer_Aircraft_fleet_strength	353	341	12
0	SpecialMission_Aircraft_fleet_strength	73	73	0
0	Tanker_fleet_strength	6	6	0
0	Helicopter_fleet_strength	807	782	25
0	Attack_Helicopter_fleet_strength	36	31	5
0	Ground_fleet_strength	110407	105784	4623
0	Combat_Tanks_fleet_strength	4614	4434	180
0	Armored_Fighting_Vehicle_strength	100882	96498	4384
0	Self_Propelled_Artillery_strength	100	94	6
0	Towed_Artillery_strength	3311	3284	27
0	MLRS_Strength	1500	1474	26
0	Navel_fleet_strength	295	287	8
0	Aircraft_Carrier_fleet_strength	2	2	0
0	Helicopter_Carrier_fleet_strength	0	0	0
0	Submarin_fleet_strength	18	18	0
0	Destroyer_fleet_strength	11	11	0
0	Frigate_fleet_strength	12	12	0
0	Navel_Corvette_fleet_strength	19	19	0
0	Offshore_Patrol_Vessel_fleet_strength	138	138	0
0	Mine_Warfare_fleet_strength	0	0	0

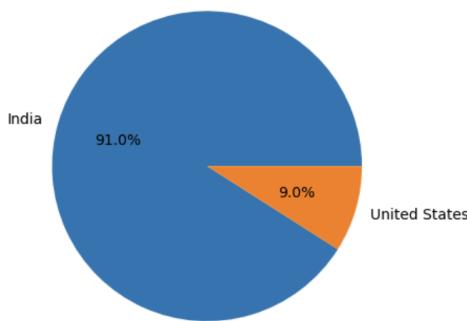
```

1 e = war_analysis()
2 e.war_btwn_two('India', 'United States')
3

Select from the below options for country 1 :
1) Air Force
2) Ground Force
3) Naval Force
4) Specified Selection
5) Total Troops
1
[577, 130.0, 254, 353, 73, 6, 807, 36]
[5193, 910.0, 1778, 2118, 584, 48, 7263, 324]
Select from the below options for country 2 :
1) Air Force
2) Ground Force
3) Naval Force
4) Specified Selection
5) Total Troops
3
[11, 9, 68, 92, 0, 22, 10, 8]
[99, 63, 544, 828, 0, 154, 60, 56]

```

Winning Chances



```

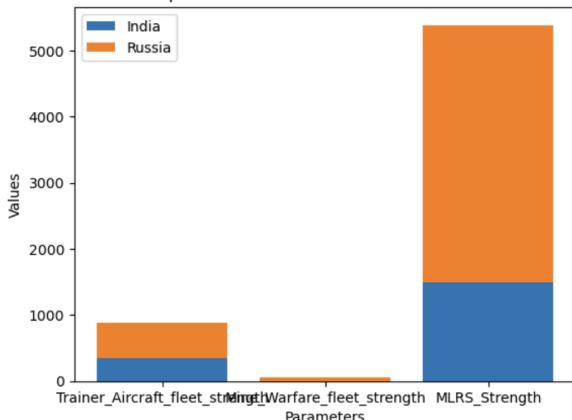
1 f = war_visualization()
2 print(f.diff_btwn_country('India', 'Russia'))

Available columns for comparison: Index(['Aircraft_fleet_strength', 'Fighter_fleet_strength',
   'Attack_Aircraft_fleet_strength', 'Transport_fleet_strength',
   'Trainer_Aircraft_fleet_strength',
   'SpecialMission_Aircraft_fleet_strength', 'Tanker_fleet_strength',
   'Helicopter_fleet_strength', 'Attack_Helicopter_fleet_strength',
   'Ground_fleet_strength', 'Combat_Tanks_fleet_strength',
   'Armored_Fighting_Vehicle_strength',
   'Self_Propelled_Artillery_strength', 'Towed_Artillery_strength',
   'MLRS_Strength', 'Navel_fleet_strength',
   'Aircraft_Carrier_fleet_strength', 'Helicopter_Carrier_fleet_strength',
   'Submarin_fleet_strength', 'Destroyer_fleet_strength',
   'Frigate_fleet_strength', 'Navel_Corvette_fleet_strength',
   'Offshore_Patrol_Vessel_fleet_strength', 'Mine_Warfare_fleet_strength'],
  dtype='object')

Enter the number of parameters to compare (up to 4): 3
Enter the column name to compare: Trainer_Aircraft_fleet_strength
Enter the column name to compare: Mine_Warfare_fleet_strength
Enter the column name to compare: MLRS_Strength

```

Comparison of Parameters between Countries

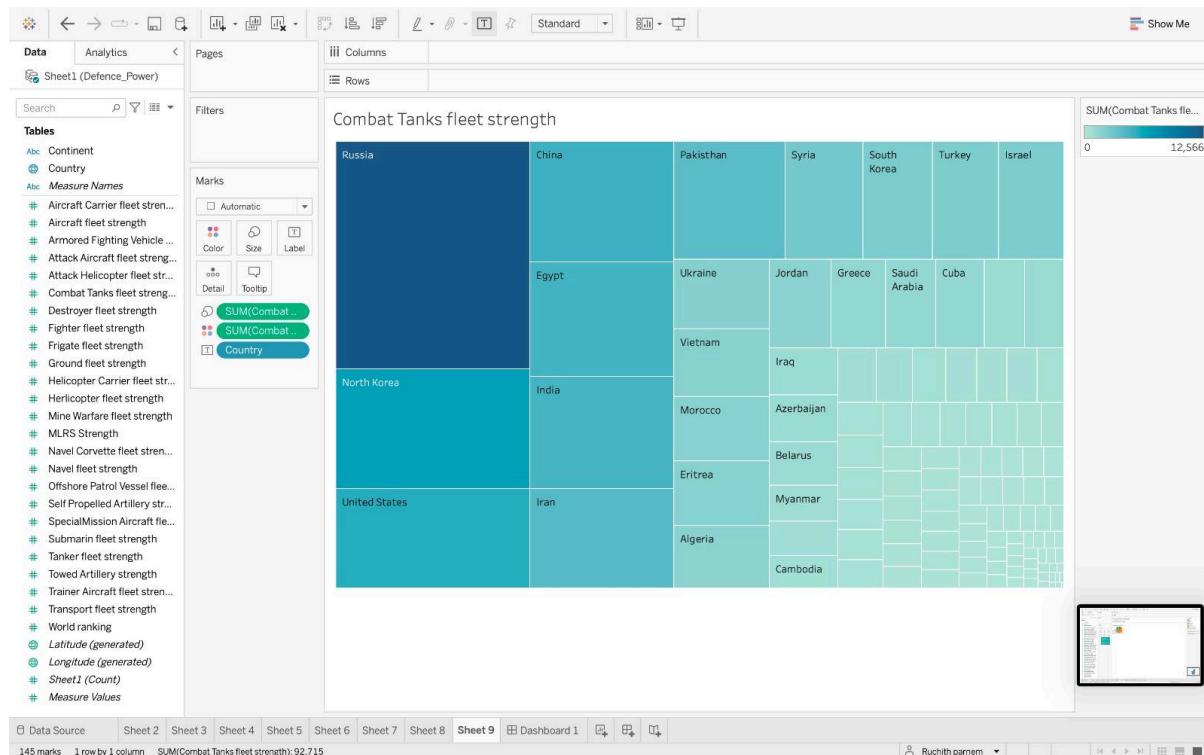


```

Column    India    Russia   Difference
0 Trainer_Aircraft_fleet_strength    353     524      -171
0       Mine_Warfare_fleet_strength    0        49       -49
0           MLRS_Strength    1500    3887      -2387, 'Russia')

```

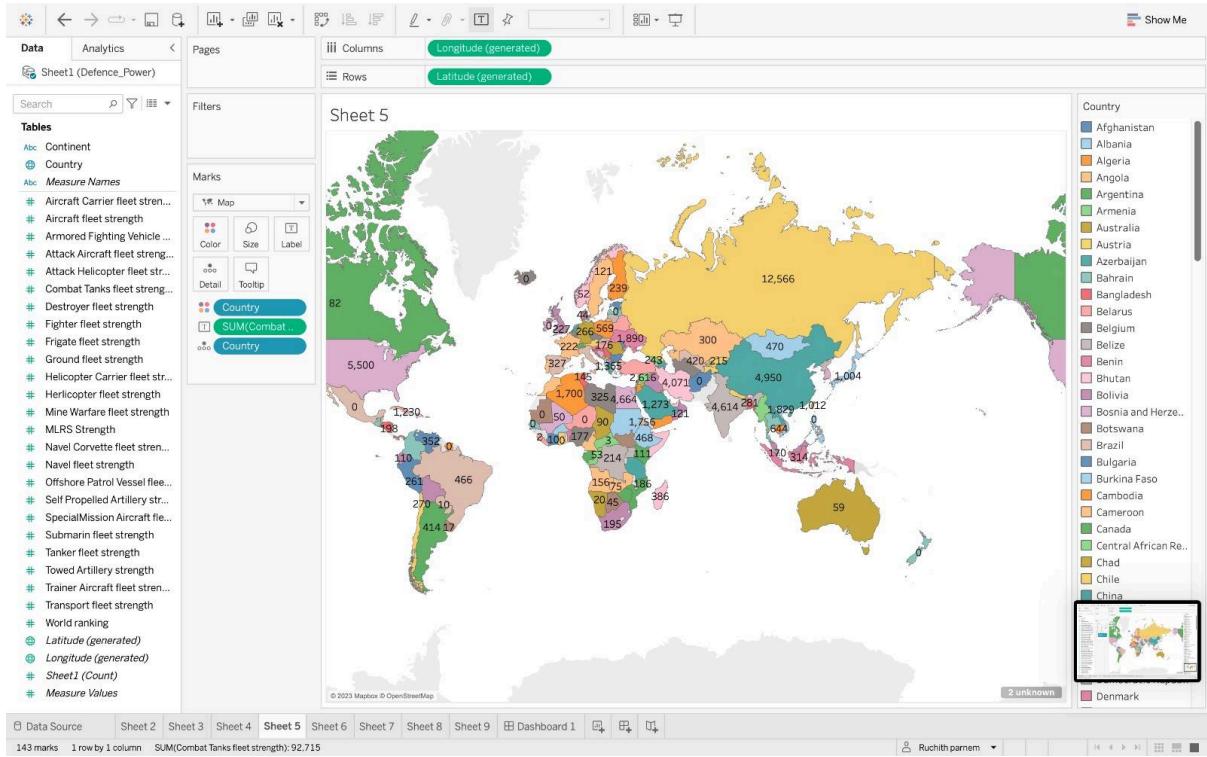
Tableau



The table contains data on the military power of various countries. The columns in the table include the country name, the continent the country is located in, the number of combat tanks in the country's fleet, and other measures of military strength.

The table is sorted by the number of combat tanks in descending order. The top three countries in terms of the number of combat tanks are Russia, China, and Pakistan. These countries all have over 10,000 combat tanks in their fleets.

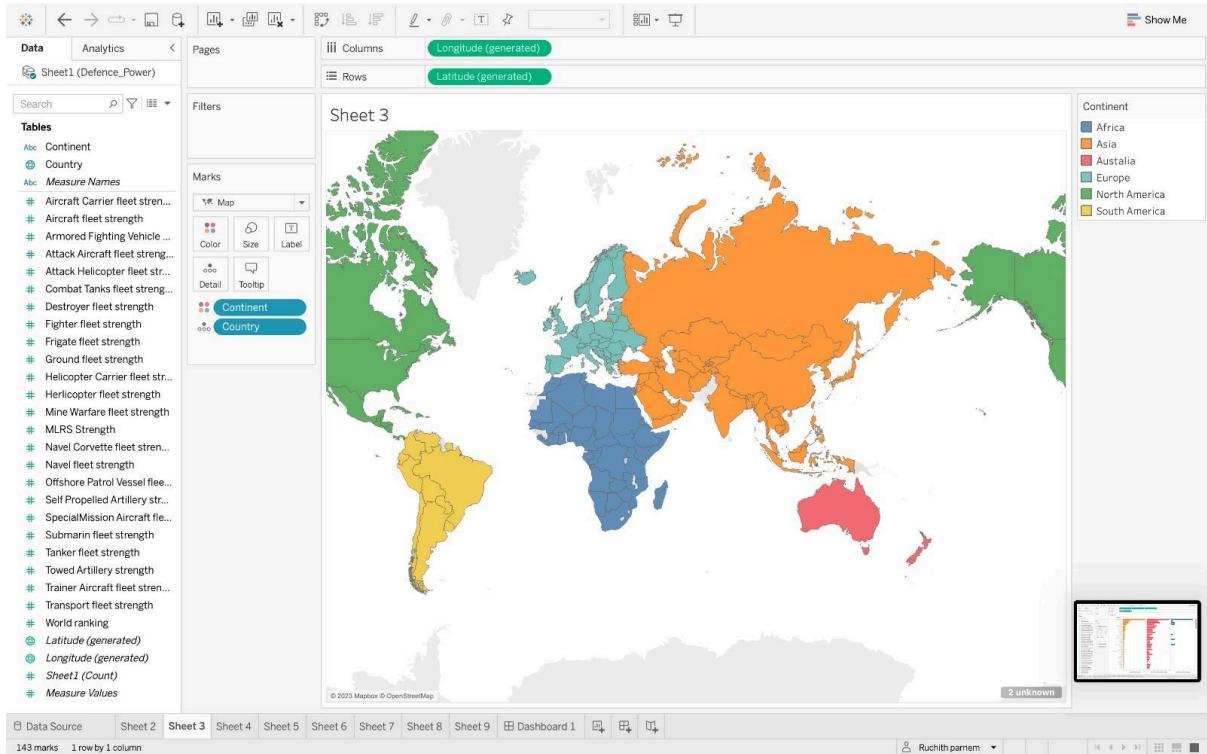
The other countries in the table have significantly fewer combat tanks. For example, the United States has 6,259 combat tanks, and the United Kingdom has 227.



The dashboard uses a map of the world to visualize the data. The size of each country's circle on the map is proportional to the country's military power, as measured by the number of combat tanks in its fleet.

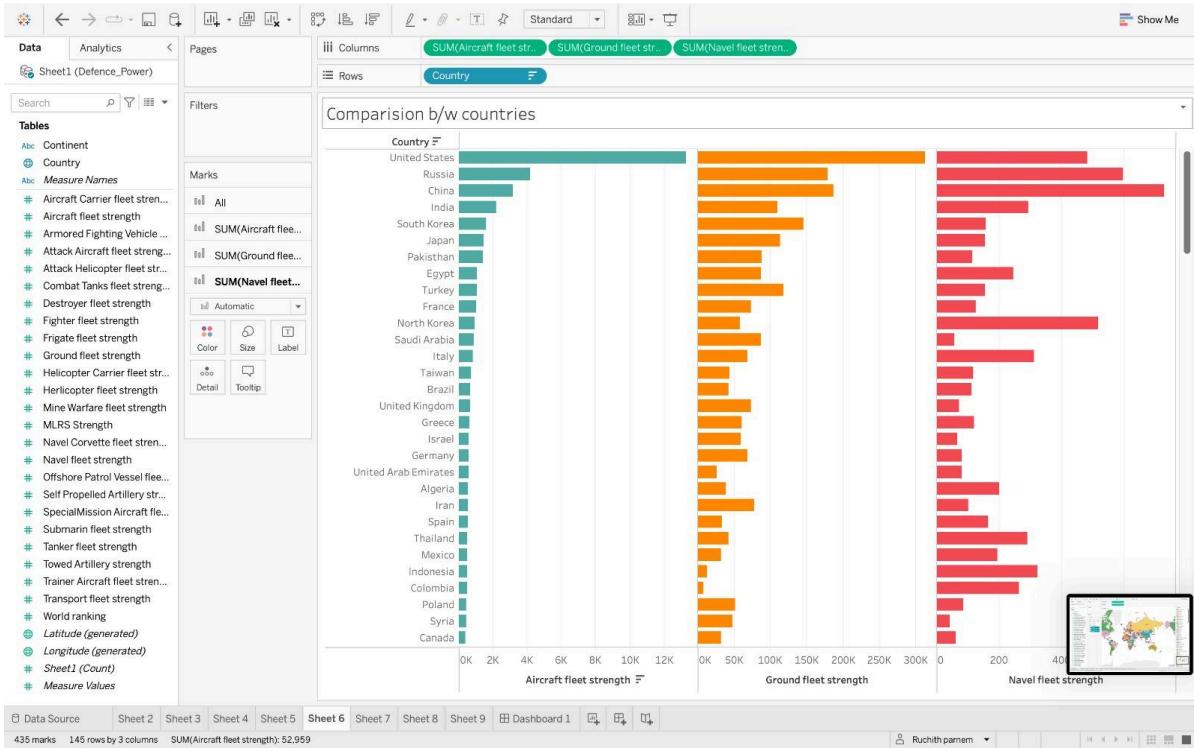
The colors of the circles also represent the continent that each country is located in. The red circles represent countries in Asia, the blue circles represent countries in Europe, the green circles represent countries in Africa, the yellow circles represent countries in North America, the orange circles represent countries in South America, and the purple circles represent countries in Oceania.

The dashboard also includes a legend that shows the different colors and their corresponding continents. The legend is located in the lower-left corner of the dashboard.



The colors of the circles also represent the continent that each country is located in. The red circles represent countries in Asia, the blue circles represent countries in Europe, the green circles represent countries in Africa, the yellow circles represent countries in North America, the orange circles represent countries in South America, and the purple circles represent countries in Oceania.

The dashboard also includes a legend that shows the different colors and their corresponding continents. The legend is located in the lower-left corner of the dashboard.



The dashboard has three main charts:

- A bar chart that shows the top 15 countries by total military power.
- A pie chart that shows the distribution of military power by continent.
- A table that shows the military power of each country, broken down by different categories.

The bar chart shows that Russia has the largest military power in the world, followed by China, the United States, India, and North Korea. The pie chart shows that Asia has the most military power, followed by Europe, North America, Africa, South America, and Oceania. The table shows that Russia has the most powerful air force, navy, and ground forces in the world.

These features allow users to explore the data in more detail and to see how the military power of different countries has changed over time.

Here are some of the things you can learn from the dashboard:

- Russia has the largest military power in the world, with a total score of 52.959.
- China is the second-most powerful military in the world, with a total score of 49.568.
- The United States is the third most powerful military in the world, with a total score of 45.088.
- India is the fourth most powerful military in the world, with a total score of 40.839.
- North Korea is the fifth most powerful military in the world, with a total score of 37.689.

The dashboard also shows that there is a significant difference in military power between countries in different continents. For example, the countries in Asia have the most military power, followed by the countries in Europe, then the countries in North America, then the countries in Africa, then the countries in South America, and finally the countries in Oceania.

Conclusion:

This Python-based data analysis tool provides valuable insights into the military capabilities of 144 countries. It enables users to explore specific country data, conduct detailed comparisons, estimate winning chances, and visualize differences using bar graphs. This tool can be a valuable asset for military analysts, policymakers, and researchers seeking to assess global military power dynamics and strategic considerations.

The flexibility of data retrieval and the graphical visualization of differences enhance the utility of this analysis tool, making it a valuable resource for in-depth military capability assessment and strategic decision-making.