**Objective**

For bootcamp, we aim to create an interactive web application that analyse the "Nuclear Explosions 1945-1998" dataset. The goal is to provide users with insightful visualizations and data exploration tools to understand the historical context, geographical distribution, and characteristics of nuclear explosions conducted during this period.

**Queries and Questions**

**1. Query 1:** *"Explore nuclear explosions by country and time period."*

* **User Parameters:**
  + **Country Selection:** Users can select one or multiple countries from a dropdown list (e.g., USA, USSR, UK, France, China).
  + **Time Period:** Users can specify a range of years using a slider or two numeric input fields for the start year and end year.

**2. Query 2:** *"Filter nuclear explosions based on yield and depth."*

* **User Parameters:**
  + **Yield Range (in kilotons):** Users can adjust a numerical range slider to set the minimum and maximum yield.
  + **Depth Selection:** Users can choose between above ground, underground, or all explosions using radio buttons.

**3. Query 3:** *"Analyze the purpose and type of nuclear explosions in specific regions."*

* **User Parameters:**
  + **Purpose Selection:** Users can select the purpose of detonation from options like COMBAT, TESTING, PNE (Peaceful Nuclear Explosions), etc., using a multi-select box.
  + **Region Selection:** Users can pick a region or deployment location from a dropdown list.
  + **Type of Deployment:** Users can choose the method of deployment (e.g., Tower, Air Drop) using checkboxes.

**Interactive Widgets from Streamlit**

* **Dropdown Menus:** For selecting countries, regions, and deployment types.
* **Range Sliders:** To specify year ranges and yield ranges.
* **Multi-select Boxes:** Allowing users to select multiple purposes of detonations.
* **Radio Buttons:** For choosing depth categories (above ground, underground, all).
* **Checkboxes:** For selecting multiple deployment types.

**Visual Presentation of Data**

**1. Charts and Graphs:**

* **Bar Charts:**
  + **Title:** "Number of Nuclear Explosions by Country"
  + **Description:** Displays the count of nuclear explosions conducted by the selected countries within the specified time frame.
  + **User Interaction:** Updates dynamically based on country and year range selections.
* **Line Graphs:**
  + **Title:** "Yield of Nuclear Explosions Over Time"
  + **Description:** Plots the yields of explosions over the selected time period, allowing users to observe trends.
  + **User Interaction:** Filters data based on yield range and displays lines for each selected country.
* **Pie Charts:**
  + **Title:** "Distribution of Explosion Purposes"
  + **Description:** Shows the proportion of different purposes for explosions in the selected region.
  + **User Interaction:** Updates when users select different purposes or regions.

**2. Maps:**

* **Interactive World Map:**
  + **Title:** "Geographical Distribution of Nuclear Explosions"
  + **Description:** Plots explosion sites on a world map using PyDeck or Folium.
  + **Features:**
    - **Markers:** Colored markers representing each explosion, with color intensity indicating yield.
    - **Tooltips:** Hovering over a marker reveals details like name, date, yield, depth, and purpose.
    - **Layers:** Options to toggle layers for different depths or purposes.

**3. Tables:**

* **Data Table:**
  + **Title:** "Detailed Explosion Data"
  + **Description:** Tabular display of the filtered dataset with sortable columns.
  + **User Interaction:** Users can sort by date, yield, depth, etc.

**User-Friendly Web Pages and Visualizations**

* **Layout and Navigation:**
  + **Sidebar:** Contains all user input controls, neatly organized and labeled.
  + **Main Page:** Displays the visualizations and updates them in real-time based on user inputs.
  + **Navigation Tabs:** Separate tabs for each query to organize content and improve user experience.
* **Labels and Titles:**
  + **Controls:** Each widget is accompanied by a clear label explaining its purpose.
  + **Visualizations:** All charts and maps have descriptive titles, and axes are labeled appropriately.
* **Legends and Explanations:**
  + **Charts and Maps:** Include legends that explain colors, marker sizes, and other visual cues.
  + **Help Tooltips:** Provide additional information when users hover over certain elements or labels.
* **Design Elements:**
  + **Color Scheme:** Use a consistent and accessible color palette throughout the app.
  + **Fonts:** Choose clean, legible fonts for readability.
  + **Images:** Incorporate relevant images or icons to enhance visual appeal.

**Navigation and User Interaction**

* **Interactive Updates:**
  + Visualizations refresh automatically as users adjust the input controls, providing immediate feedback.
* **Responsive Design:**
  + Ensure that the application is responsive and works well on various screen sizes and devices.
* **User Guidance:**
  + **Introduction Section:** Briefly explain the purpose of the app and how to use it.
  + **Instructions:** Provide clear instructions where necessary, such as how to select multiple options.
* **Error Handling:**
  + Display user-friendly error messages if no data is available for the selected parameters.

**Additional Features and Considerations**

* **Data Insights:**
  + Highlight key findings or interesting patterns observed from the data.
  + Include annotations on charts for significant events (e.g., major treaties or bans).
* **Data Download Option:**
  + Allow users to download the filtered dataset for their own analysis.
* **Performance Optimization:**
  + Efficiently handle data loading and filtering to ensure smooth user experience.

# Points to be covered

**Mapping the Design to Project Requirements**

To ensure that the project meets all the specified requirements, we'll align each component of the design with the corresponding requirements. Below is how the proposed design will incorporate the required Python features, Streamlit features, visualizations, and data analytics capabilities.

### **Python Features**

**[PY1] Function with Two or More Parameters (One with Default Value):**

* **Implementation:** Create a function filter\_data(country, year\_range, purpose='All') that filters the dataset based on the selected country, year range, and purpose (with 'All' as the default value).
* **Usage:** The function will be called at least twice—once using the default value for purpose, and once specifying a specific purpose selected by the user.

**[PY2] Function Returning More Than One Value:**

* **Implementation:** Develop a function calculate\_statistics(filtered\_data) that returns multiple statistics such as total number of explosions, average yield, and maximum yield from the filtered dataset.
* **Usage:** These statistics will be displayed on the dashboard to provide users with summary insights.

**[PY3] Error Checking with try/except:**

* **Implementation:** Implement error handling when reading the dataset or processing user inputs. For example, using try/except blocks to catch errors like missing files or invalid data types.
* **Usage:** Ensures the application doesn't crash and provides informative error messages to the user.

**[PY4] List Comprehension:**

* **Implementation:** Use a list comprehension to process data efficiently. For example, creating a list of explosion names that meet certain criteria:
* high\_yield\_explosions = [name for name, yield in zip(data['Data.Name'], data['Data.Yield.Upper']) if yield > 1000]
* **Usage:** This list can be used to highlight significant explosions in the visualizations or data tables.

**[PY5] Dictionary Access:**

* **Implementation:** Create a dictionary mapping countries to their total number of explosions or average yields:
* country\_explosion\_counts = data['WEAPON SOURCE COUNTRY'].value\_counts().to\_dict()
* **Usage:** Access keys and values to display this information in the application or use it in further calculations.

### **Streamlit Features**

**[ST1] Slider Widget:**

* **Usage:** Implement a range slider for users to select the time period (year range) they're interested in exploring.

**[ST2] Dropdown Menu:**

* **Usage:** Use dropdown menus for:
  + **Country Selection:** Allowing users to select one or multiple countries.
  + **Region Selection:** Users can pick a specific region or deployment location.

**[ST3] Multi-select Box:**

* **Usage:** Enable users to select multiple purposes of detonations (e.g., COMBAT, TESTING) to filter the data accordingly.

**[ST4] Customized Page Design Features:**

* **Implementation:**
  + **Sidebar:** Organize all input controls in a Streamlit sidebar for easy navigation.
  + **Fonts and Colors:** Customize the application's fonts and color schemes to enhance readability and visual appeal.
  + **Images:** Include relevant images, such as historical photos or icons related to nuclear testing.
  + **Navigation:** Use Streamlit's page layout options or a navigation bar to allow users to switch between different sections or queries.

### **Visualizations**

**[VIZ1] Bar Chart:**

* **Title:** "Number of Nuclear Explosions by Country"
* **Implementation:** Use Plotly or Matplotlib to create a bar chart that updates based on the selected countries and time period.
* **Features:** Include labels, legends, and tooltips for interactivity.

**[VIZ2] Line Graph:**

* **Title:** "Yield of Nuclear Explosions Over Time"
* **Implementation:** Plot the yield of explosions over the selected time period, with lines representing different countries or purposes.
* **Features:** Interactive hover information, zoom functionality, and clearly labeled axes.

**[VIZ3] Pie Chart:**

* **Title:** "Distribution of Explosion Purposes"
* **Implementation:** Display a pie chart showing the proportion of different purposes for the selected data subset.
* **Features:** Legends and hover text showing percentages and counts.

**[VIZ4] Data Table:**

* **Title:** "Detailed Explosion Data"
* **Implementation:** Present a sortable and filterable table using Streamlit's st.dataframe() or st.table() functions.
* **Features:** Users can sort columns and search within the table.

**[MAP] Detailed Map:**

* **Implementation:** Use PyDeck or Folium to create an interactive map plotting the locations of the explosions.
  + **Markers:** Represent explosions with markers whose size or color reflects the yield.
  + **Tooltips:** Show detailed information when users hover over or click on a marker (e.g., explosion name, date, yield).
  + **Layers:** Allow users to toggle layers to view different depths (above ground, underground).

### **Data Analytics Capabilities**

**[DA1] Clean or Manipulate Data (Required):**

* **Implementation:** Use lambda functions to clean or transform data. For example, converting depth values to categorize explosions as "Above Ground" or "Underground":
* data['Depth Category'] = data['Location.Coordinates.Depth'].apply(lambda x: 'Underground' if x > 0 else 'Above Ground')
* **Usage:** This new column can be used for filtering and visualization.

**[DA2] Sort Data:**

* **Implementation:** Allow users to sort the data table by yield, date, or any other column in ascending or descending order.

**[DA3] Find Top Largest or Smallest Values:**

* **Implementation:** Provide functionality to display the top N explosions by yield:
* top\_explosions = data.nlargest(N, 'Data.Yield.Upper')
* **Usage:** Highlight these in the data table or visualizations.

**[DA4] Filter Data by One Condition:**

* **Implementation:** Filter data based on a single condition, such as explosions conducted by a specific country:
* filtered\_data = data[data['WEAPON SOURCE COUNTRY'] == selected\_country]

**[DA5] Filter Data by Two or More Conditions with AND/OR:**

* **Implementation:** Apply multiple conditions to filter data:
* filtered\_data = data[
* (data['WEAPON SOURCE COUNTRY'].isin(selected\_countries)) &
* (data['Data.Yield.Upper'] >= min\_yield) &
* (data['Data.Yield.Upper'] <= max\_yield)
* ]

**[DA7] Add/Drop/Select/Create New Columns:**

* **Implementation:** Create new columns like 'Decade' to group data by decades:
* data['Decade'] = (data['Date.Year'] // 10) \* 10
* **Usage:** Use this column for grouping data in visualizations.

**[DA9] Add a New Column or Perform Calculations on DataFrame Columns:**

* **Implementation:** Calculate the average yield per country and add it as a new column:
* average\_yield = data.groupby('WEAPON SOURCE COUNTRY')['Data.Yield.Upper'].mean().reset\_index()
* **Usage:** Display this information in visualizations or data tables.

### **Query and Feature Alignment**

**Query 1:** "Explore nuclear explosions by country and time period."

* **Python Features:**
  + **[PY1]** Function with parameters for filtering data.
  + **[PY2]** Function returning statistics.
  + **[PY3]** Error handling when processing inputs.
* **Streamlit Features:**
  + **[ST1]** Year range slider.
  + **[ST2]** Country selection dropdown.
  + **[ST4]** Customized sidebar and navigation.
* **Visualizations:**
  + **[VIZ1]** Bar chart of explosions by country.
  + **[VIZ2]** Line graph of yield over time.
  + **[MAP]** Interactive map showing explosion sites.
* **Data Analytics:**
  + **[DA1]** Data cleaning with lambda functions.
  + **[DA4]** Filtering by country.
  + **[DA5]** Filtering by country and time period.
  + **[DA7]** Adding 'Decade' column.

**Query 2:** "Filter nuclear explosions based on yield and depth."

* **Python Features:**
  + **[PY4]** List comprehension to process yields.
* **Streamlit Features:**
  + **[ST1]** Yield range slider.
  + **[ST3]** Depth selection radio buttons.
* **Visualizations:**
  + **[VIZ2]** Line graph showing yield trends.
  + **[VIZ4]** Data table of filtered explosions.
* **Data Analytics:**
  + **[DA1]** Data manipulation to categorize depth.
  + **[DA3]** Finding top explosions by yield.
  + **[DA5]** Filtering by yield and depth.
  + **[DA9]** Calculations on yield data.

**Query 3:** "Analyze the purpose and type of nuclear explosions in specific regions."

* **Python Features:**
  + **[PY5]** Dictionary mapping purposes to counts.
* **Streamlit Features:**
  + **[ST2]** Region selection dropdown.
  + **[ST3]** Multi-select for purposes.
  + **[ST4]** Customized design elements.
* **Visualizations:**
  + **[VIZ1]** Bar chart of explosions by purpose.
  + **[VIZ3]** Pie chart of explosion purposes.
  + **[MAP]** Map with layers for different purposes.
* **Data Analytics:**
  + **[DA1]** Data manipulation for purposes.
  + **[DA5]** Filtering by purpose and type.
  + **[DA7]** Creating new columns if needed.

### **Additional Considerations**

* **Error Handling ([PY3]):** Throughout the application, ensure that all data operations and user inputs are wrapped with appropriate error handling to prevent crashes.
* **User Experience Enhancements:**
  + **[ST4]:** Consistent styling, clear labels, and intuitive navigation.
  + **Visual Aids:** Use legends, annotations, and tooltips to enhance understanding.
* **Performance Optimization:**
  + Efficient data loading and filtering to maintain responsiveness.

I hop this detailed mapping helps you understand how the proposed design aligns with the project requirements and demonstrates how each feature will be implemented in the application.