The War for Ukraine

SMU Data Analytics Bootcamp

Project 3 Group 4 | 30JAN2023

Group Members: Sajid Anjum, Raymond Bell, Garrett Kidd, Japhet Mwamba, and Brenna Wallace

Contents:

Section I – Introduction and Motivation

In 2014, Russia, under Putin, conquered the Crimea peninsula of Ukraine. Since then, Russian separatists have seized control of the Luhansk and Donetsk regions in Eastern Ukraine, the same regions in which rich oil deposits were discovered in 2010. The world has long been wary of Putin's grand imperialistic designs, and when he started amassing troops on the Ukrainian border in late 2021, most analysts were predicting a ground invasion of Ukraine. On February 24th, 2022, Putin announced the commencement of a "special military operation" whose main goal seemed to be the toppling of Zelensky's Ukrainian regime within 72 hours. 340 days later, the war still rages.

Since Russia is a military superpower who also happens to be the primary energy supplier to Western Europe, the Russian invasion of Ukraine is an issue with tremendous international ramifications, not the least of which could be the resurrection of the specter of World War II. In this project, our goal is to analyze a data set that tracks all the different war incidents in Ukraine over the past year to see if we can find some interesting patterns and insights.

Commented [RB1]: which

Section II – Research Questions

1. What is the relationship between the number of war incidents and the number of fatalities? How does this vary over time?
2. Who instigated most of the battles? The instigator of a particular war incident is a subjective decision. Does this decision vary with the news source?
3. What does the distribution of the location of war incidents tell us?

Commented [RB2]: subjective?

Section III – Data Cleaning/Wrangling/Processing/Filtering

Initial Transformations

We used three datasets. Our primary dataset was a .csv file from the website https://acleddata.com/.com. We navigated to their export tool (https://acleddata.com/data-export-tool/) and selected all conflict data from 01/01/22 to 01/12/23 for the country of Ukraine. Our resulting dataset was a .csv file with 28 columns and 54,817 rows. This was too much data four our website to handle, so we selected a random sample of 1000 rows and trimmed the dataset to 17 columns we considered relevant. We used a Python Jupyter Notebook to do these initial transformations. This is the list of columns we decided to keep:

```
['data_id', 'event_date', 'year', 'event_type', 'sub_event_type',
       'actor1', 'admin1', 'admin2', 'admin3', 'location', 'latitude',
       'longitude', 'source', 'notes', 'fatalities', 'timestamp', 'source2']
```

We converted the .csv file to a .json in Python and uploaded it to our data folder as '`ukraine_war_data.json`'.

The other two datasets that we used were geoJSONs taken from the website https://data.humdata.org/dataset/geoboundaries-admin-boundaries-for-ukraine?

You can find these in our data folder as '`geoBoundaries-UKR-ADM1.geojson`' and '`geoBoundaries-UKR-ADM2.geojson`'

This website is run by The Humanitarian Data Exchange Group and has files available for download that represent the oblast (state) and raion (county) level administrative regions of Ukraine. We used these files only for our map to generate the boundaries of the regions on our map.

Commented [RB3]: region?

Commented [SA4R3]: Raion is a Ukrainian word, just like oblast. It would roughly translate to county.

Data Cleaning:

We needed to delete duplicate rows. However, the problem was sometimes the exact same event was reported by two different sources with a different 'data_id' column. Moreover, sometimes the exact same event was reported by the exact same source twice, with two different 'actor1' or primary instigators. Since double counting events, especially events with large numbers of fatalities, would affect our data analysis, we used:

```
df1001 = df1000.drop_duplicates(subset=['fatalities', 'latitude', 'longitude',
'timestamp'])
```

This allowed us to only drop events which had the same values in the four columns above. There were 36 such rows, out of 1000, which wasn't bad. This meant that some of our data about 'sources' and 'actor1' would not be exact, but we felt this was preferable to having bad data about fatalities and location. As we will see later, the data about news sources and primary instigators was quite political and subjective.

Commented [RB5]: This allowed us to drop events which had the same values as the four columns above? Recommend restating this sentence.

The second thing we did was to make a new column called 'source2'. There were 108 different news sources reporting 1000 events, and that was too many to do a meaningful analysis. Thus, we combined all sources that reported 10 or fewer events into the value 'other' and we were left with only 14 different news source values.

Export:

We used the `.to_json()` method to export our data as .json file, and we used the d3 library to read the .json file into Javascript.

Section IV – Website

Html, Layout, colors, design decisions:

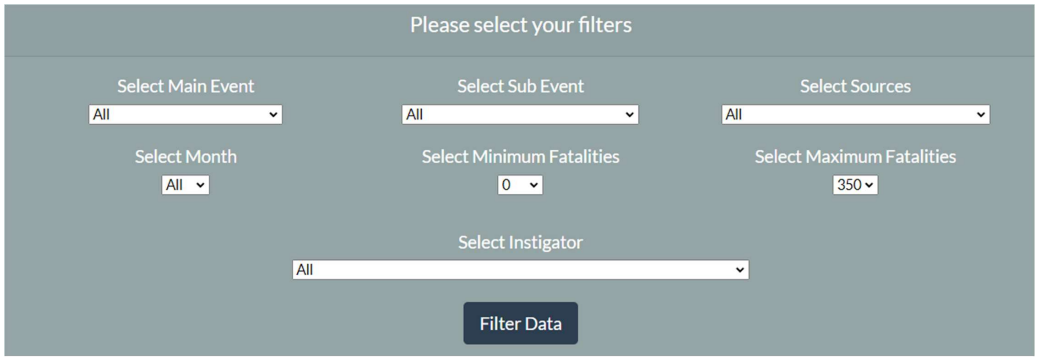
The layout of our website consisted of a home page with a navigation bar that allowed us to link to six additional webpages. The navigation bar was a template theme from bootstrap flatly, and the six additional pages were:

Sunburst Graph, Other Graphs, Map, About Us, Data Tables, and References. The two graphs pages were dropdowns in the "Dashboards" menu. The Data Tables and References pages were dropdowns in the "Data" menu.

Since the topic was a grave one, we tried to choose somber greys as the main theme of our website. The main colors that we chose to break up the theme were the blue and gold of the Ukrainian flag, and a seagreen hue that contrasted well with the remaining colors. In our map, we stuck to bold, contrasting primary and other common colors for our markers to make the markers easy to distinguish from the map and from each other. The markers were a function of the number of fatalities for an additional level of detail. We chose the background colors of the administrative layers to be shades of grey and green-grey in keeping with the theme of our website.

Filter Menu:

Our graphs and map were able to filter data by the following categories:



We used `$d3.select().node().value` to read in the value of the filters and applied the `data.filter()` method to filter the data by the selected value. In order to populate the menus, we used `.foreach()` to add the values of the "event_type" to a variable designed as a set, so only unique values would be taken. Then, we converted the set to an array using the `Array.from()` method. Then, we used `d3.select().append().text()` to populate the various menus.

Populating the "Sub Event" menu was tricky as the sub events depended on the main event. Thus, the values of "Sub Event" automatically changed when a "Main Event" was selected. If the Main Event was selected after the Sub Event, the Sub Event defaulted back to "All".

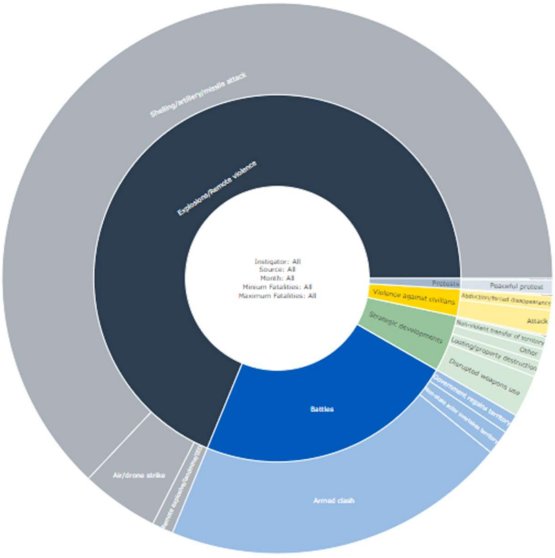
After all the filters were set, clicking the "Filter Data" button caused the filter to change the data displayed on the graphs and the map. We made sure to display an error message if the maximum number of fatalities was greater than the minimum number.

The filter menu for the sunburst chart did not include "Main Event" and "Sub Event" because the sunburst chart automatically filters for those events.

Sunburst Graph:

We used the plotly library to make our sunburst graph. The inner ring of our sunburst was the main event and the outer ring was the sub event. We used a FOR loop to collect the value counts of events and subevents into a dictionary, and then used another FOR loop to convert that dictionary into an array compatible with the needs of the plotly sunburst graph.

The center of the sunburst is a list of all the filters that are being used to generate the sunburst.



Map:

We used the leaflet library to create a map of all the war incidents in Ukraine. The shape of the markers were circles. The colors corresponded to the type of incident while the size of the circle was a function of the number of fatalities. When clicked, the marker showed information about the specific incident. We used `L.circle()` to create the markers, `L.layerGroup()` to create a layer of markers, and a `L.control.layers()` to turn the events layer on and off.

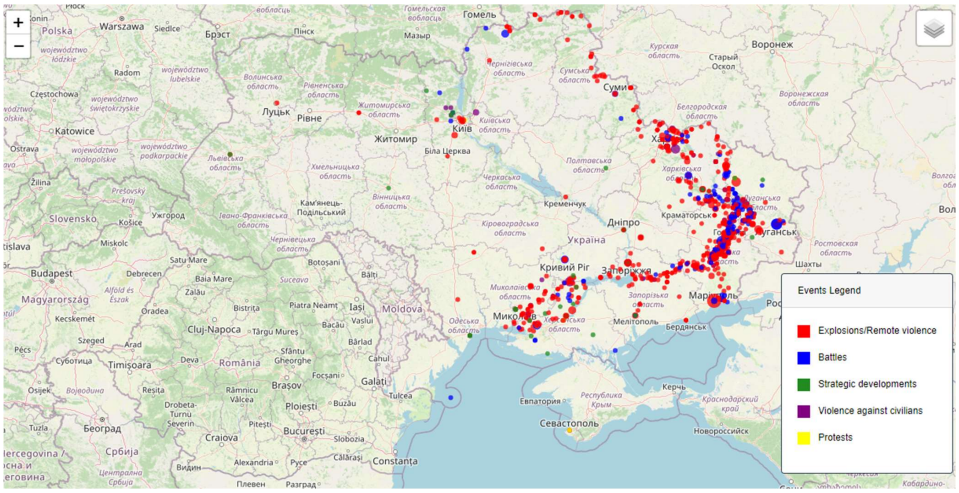
We also added a legend in the bottom right corner of this map. We used `L.DomUtil()` to add the html that was used to create the legend to the map.

The leaflet plot had three base layers—Street, Topography, and Satellite. It also had five overlays—Events, Admin Regions 1, Admin Regions 2, Heat Map, and Marker Cluster. The events layer was used to turn the war incident markers off and on. The admin regions 1 showed the boundaries of Ukrainian oblasts (states) and the admin regions 2 showed the boundaries of Ukrainian raions (counties). Both layers changed color when the mouse hovered over them and when clicked, zoomed in on the regions. We used the `L.geoJSON` function from the leaflet library and the `mouseover` attribute of the `onEachFeature()` attribute of this function to create these two layers.

The heat layer and marker cluster layers were standard plugins of the leaflet library. Once we downloaded the appropriate files, the following methods generate those layers:

`L.heatLayer()` for the heat layer

`L.marker.addLayer()` for the marker clusters



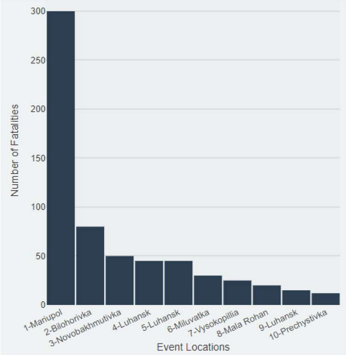
Dashboard:

Line graph:

We used the highcharts library to create this line graph. We extracted the month from the `event_date` column using the `.split()` method, and used `.filter()` and `.length` attributes to obtain the y-axis values for the number of incidents and an additional FOR LOOP to obtain the number of fatalities. The line graph had time (in months) on the y-axis and two traces—the number of war events and the number of fatalities.

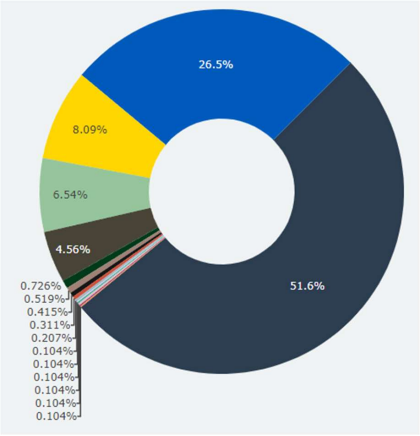
Bar graph

We used the plotly library to create a bar graph to represent the ten incidents with the highest number of fatalities. The `.filter()` and `.slice()` methods were used to create the labels and the values for the graph. The labels for the bar graph were the values of the `location` column in our data set. However, since plotly automatically combines the data from two separate events at the same location into one event, we added a rank number in front of each location to separate the incidents.



Donut chart

We created a plotly donut chart to display the primary instigators for our dataset. We filtered our data by instigators, collected the value count of each instigator, and used those as “labels” and “values” for our donut chart.



Data Table:

The only datatable we uploaded was our main Ukraine_war_data.json table. However, we trimmed this table to the eleven most relevant columns and renamed the columns to make the names more user friendly. Then, we used the `DataTables()` method from the `datatables.net` plugin to display the tables in a pretty format.

About:

We created a simple page of the pictures and names of the five individuals who were responsible for creating and delivering this project.

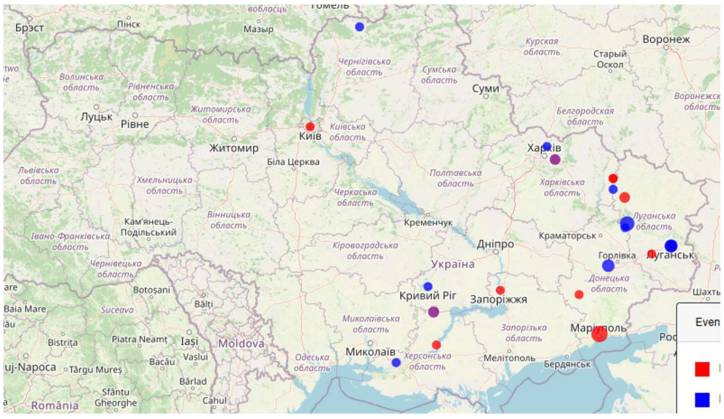
References:

Our references page was a static, informative webpage that references where we got the data of this project from and further reading.

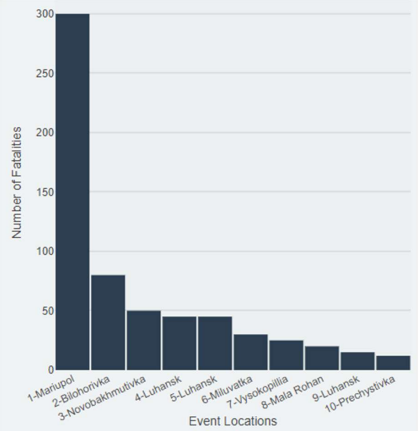
Section V -- Conclusions and Future Work:

1. What is the relationship between the number of war incidents and the number of fatalities? How does this vary over time?

We needed three visualizations to answer this question. The first one was our line chart. We see that the number of war incidents is fairly constant with a sharp uptick in September corresponding to the time that Putin sent reinforcements to Ukraine. However, the number of fatalities seems to be a bumpy chart. The map, when filtered for events with ten or more fatalities, gives us a clue as to what is going on.

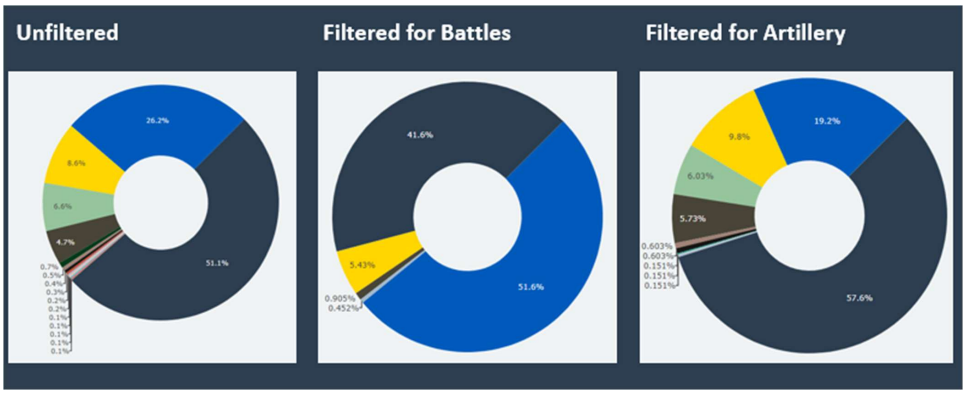


Map filtered for war incidents with ten or more fatalities



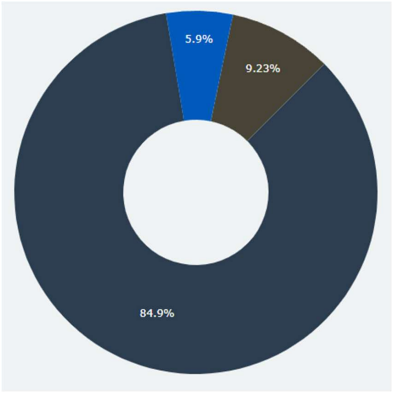
It turns out that only a handful of incidents have more than ten fatalities, but many of these events have many times more than ten fatalities. The big spike in fatalities in March 2022 is the result of the bombing of the city of Mariupol during which 300 people were killed. We can also see this data in the bar chart with the list of top ten most fatal war incidents as the months of the most deadly incidents correspond to the spike on the line chart. Thus, we can conclude that the fatalities and war events are quite related and are fairly constant aside from a few incidents that are very deadly.

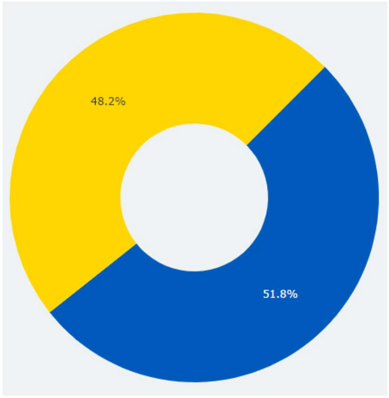
2. Who instigated most of the battles? The instigator of a particular war incident is a subject decision. Does this decision vary with the news source?



The dark grey represents the Military Forces of Russia, the gold represents the United Armed Forces of Novorossiya (Russian separatists), and the blue represents the Military Forces of Ukraine. It is very clear that the majority of incidents and the shelling was begun by Russia. However, the Ukrainians were responsible for starting most of the battles. We also decided to see what would happen if we filtered by News Source.

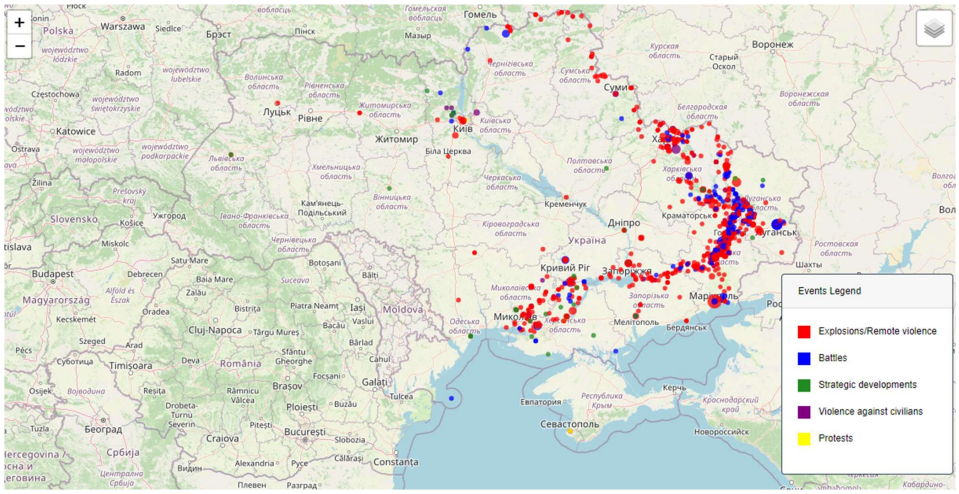
The Ministry of Defense of Ukraine reported 391 of the 1000 total incidents, and it is a pro-Ukraine source. However, the DPR Armed Forces Press Service reported 126 of the 1000 total incidents and it is a pro-Russian source. The respective donut charts, filtered for Explosions/Remote Violence, are the following. The colors are the same as above. The dark brown represents the Russian Airforce.



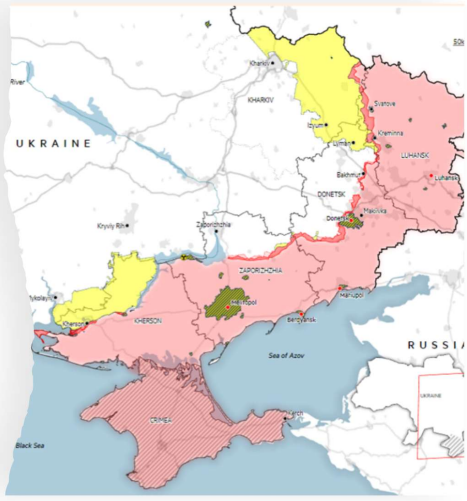


The Ukrainian news source holds Ukraine responsible for only 6% of explosions, while DPR holds Ukraine responsible for 52% of all explosions. Thus, we may conclude that information about primary instigators may be highly biased.

3. What does the distribution of the location of war incidents tell us?



The overwhelming majority of incidents are concentrated in a narrow band stretching from Kherson Oblast to Kharkiv Oblast in the North. Below is a map from the Financial Times showing Russian controlled parts of Ukraine. Thus, we may conclude that most of the battles are fought at the line separating Russian-controlled and Ukrainian-controlled parts of Ukraine.



Limitations:

- Our principal limitation was the size of the data set. We feel that our code can handle some more data points, but we will need to make changes to our map and perhaps our graphs to make sure that the data can be displayed in a comprehensible manner.
- Our filters do not allow us to select multiple events, sub-events, instigators, or sources. It would be nice if we could select multiple events, or perhaps even display multiple events as different trendlines on the same graph so we may compare them.

Future Work:

- We would like to be able to connect our website to a live API that will update information based on live events. We would also like our map to show which nation is controlling which territory, and which territories are under dispute.
- We would also like to integrate a database and a backend server into the website because our website is not built to import and export hundreds of thousands of data points.