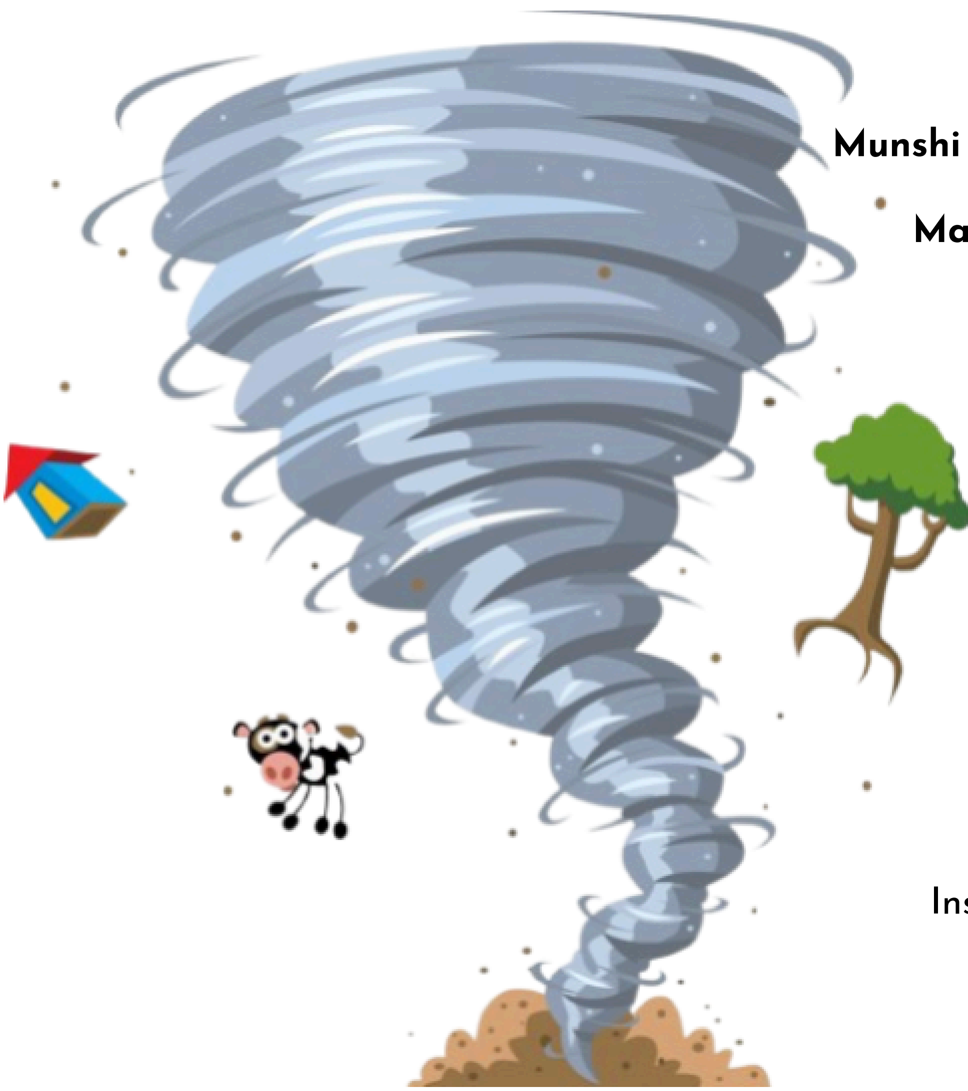
Final Project Report for

CS 5040/6040: Scientific Visualization for Data Science

Fury in Motion: A 70+Year Analysis of U.S. Tornado Activity



Submitted by:
Munshi Saifuzzaman - A02454509
Sam Elliss - A02387348
Matthew Larsen - A02095262

Instructor: **Dr. Steve Petruzza**

April 19, 2025

Project Goals

General goals:

1. Display tornado data in an engaging and informative format.
    a. Include as much data as possible, preferably in several different forms, to communicate more effectively.
2. Cover a large time period to discover trends over time.

Scientific goals:

Develop an interactive visualization tool that maps historical tornado events based on their length, fatalities, injuries, and width, enabling users to analyze tornado severity, track their geographical paths, and understand patterns in tornado occurrences.

1. Munshi's goal: As part of this project, I will focus on visualizing the top 10 to top 50 most impactful tornadoes in U.S. history, ranked by various severity metrics such as path length, fatalities, and width. The goal is to enable users to explore tornado intensity and damage potential through interactive comparisons. Each tornado will be visualized along with its magnitude, and a dynamic slider will allow filtering tornadoes within a specified metric range—e.g., from shortest to longest path when ranked by length. This interactive module is designed to support the investigation of scientific questions such as:
    a. Does a tornado with the highest magnitude or length always correspond to the highest fatality count or overall damage?
    b. What were the environmental conditions (e.g., wind speed, temperature, surface pressure) when a specific tornado event occurred and along its trajectory?
    To address the latter question, I will integrate **historical weather data using the Open-Meteo Weather API**, which provides hourly and daily meteorological parameters. This contextual layer will enhance our understanding of the conditions that potentially amplified tornado severity, offering a more comprehensive view of tornado impact.
2. Sam's goal: Include additional data in the dataset from weather stations to display more information that could be relevant to tornado formations. This will likely manifest as a different variation of a map-style visualization. In general, focus on covering tornado distribution over time.
    a. In implementation this is explored through a map dotted with weather station icons, each of which has recorded data month-to-month. The visualization will allow for investigation of specific time periods, and what the weather conditions were like across the US for that particular month. Information like elevation, snow, precipitation, and temperature will allow for weather quirks like droughts or hot flashes to be more obvious and their impact on tornadoes to be explored.

Questions like "why were there so many tornadoes during this time period?" and general trends of that sort.

b. Additionally, include a more traditional plot to explore how tornado frequency has changed over time. Include as much interactivity as makes sense to show different parts of the data, like magnitude.

3. Matthew's goal: Investigate if there is any correlation between county population sizes and the damage caused by historical tornadoes in that county. For the scope of this project, damage was measured by the number of injuries and fatalities reported for a given tornado. County populations were determined using the United States Census Bureau's county population estimates for 2023.
   a. This goal was achieved through the creation of an interactive choropleth map which displayed county population sizes in one of seven shades of blue with a geo-referenced stacked bar chart indicating the number of injuries and fatalities caused by each tornado in our dataset. Blue dots were used instead of red or yellow bars to represent tornadoes which had no reported fatalities or injuries.

Background

We mainly used the US Tornado Dataset covering tornadoes from 1950-2023, sourced from here. This dataset is a single CSV file covering the date, location, magnitude, injuries and fatalities, and end location. This worked as a solid foundation for the work we completed on this project, supplemented by other datasets when necessary for more information.

For **Sam's section**, to focus more on other meteorological factors involved, we pulled weather station data from NOAA in a variety of locations where tornadoes are common. These databases contained the same essential information like precipitation, temperature, and location; but varied quite a bit on the specific details they included. This necessitated some data cleaning to properly render stuff with matplotlib, but was manageable. Adding more weather station datasets would be very easy with the current modular design.

Munshi's section:

In addition to prior meteorological and demographic explorations, I focused on integrating atmospheric context directly into tornado path visualizations. To accomplish this, I used the Open-Meteo API to fetch historical weather variables—including temperature, precipitation, wind speed, and humidity—based on each tornado's start and end coordinates. This enriched the dataset with real-world environmental signals, enabling a deeper investigation into how weather conditions may influence tornado severity metrics.

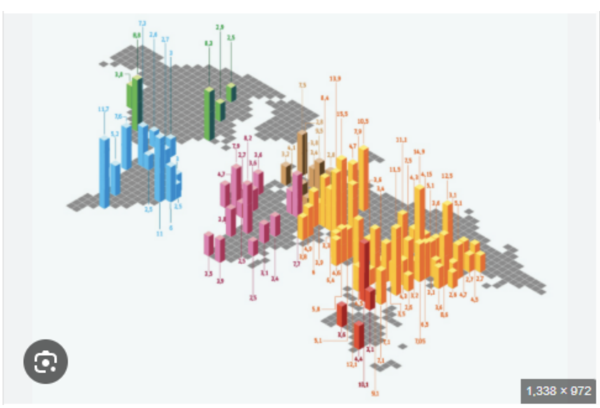
To guide this work, I consulted meteorological sources from NOAA, the C2ES climate portal, and various academic summaries that describe the relationship between atmospheric instability (e.g., surface heating, wind shear) and tornado dynamics. These sources reinforced the relevance of investigating variables like surface temperature, precipitation, and wind gusts. They also inspired hypothesis-driven analyses, such as whether higher temperatures correlate with longer tornado paths, or whether wind gusts relate to fatality counts.

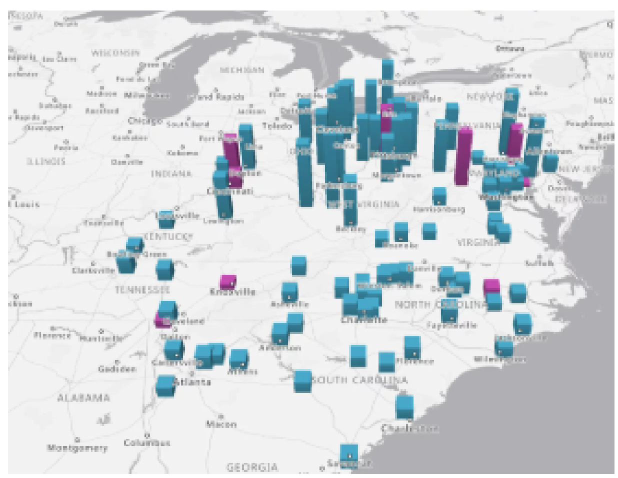
Additionally, I reviewed recent visual analytics approaches that integrate geospatial and temporal weather data with extreme weather events. This helped inform both the scientific framing of our questions and the design of interactive features, such as popups displaying weather metrics at each tornado marker, radar-style comparison views, and correlation matrix validation. These integrations positioned the project to not only visualize tornado paths but also interpret their broader atmospheric context.

Much of my implementation also involved exploring how to structure and optimize API-driven scientific workflows. Since Open-Meteo offered programmatic access to historical weather datasets, I examined its limitations and caching strategies to handle data retrieval at scale. This experience deepened my understanding of reproducible scientific pipelines where external data enrichment plays a critical role in hypothesis formation and visualization.

For **Matthew's section**, we pulled data from the United States Census Bureau regarding county population data and county lines. Each county's identification consists of a 2-digit state Federal Information Processing Series (FIPS) code and a 3-digit county FIPS code. These FIPS codes allowed us to merge the population data and county data into a single geojson file.

For additional background, we consulted similar data visualizations done with maps, since our data is very location-focused. There's a lot of different ways to visualize geographic data and interact with it, and looking at a variety of implementations helped give us ideas. Here's a few examples:





left and right

Project Description

We used the data described above in the background section to back our project. Overall our project is three different maps that we use to investigate three main questions about tornadoes in the US, as described in the Overview and Goals section.

Throughout the project a few new questions arose. For **Sam's section**, the exact meteorological factors to look into were defined pretty late into it, since the weather stations had so much data. So, the question of "what do we actually want to investigate here?" came up. Ended up deciding to focus on temperature and precipitation since those seemed most relevant. So, the new question for that section was something like "How has the distribution of tornadoes changed over time, and are temperature or precipitation obviously effecting that?" I further decided to focus on time-based analysis since my teammates were focusing on other areas, and created the second plot to cover simple questions like "Are there more tornadoes in recent years than further ago?"

Munshi's section: Weather-Driven Scientific Exploration

In my portion of the project, I extended the core tornado dataset by integrating historical atmospheric data using the Open-Meteo API. This enabled us to explore a series of scientific questions that go beyond structural attributes and into environmental and comparative dimensions of tornado behavior. My work involved developing interactive visualizations and analytical modules to answer five key questions:

1. **How do tornado characteristics—such as length, width, fatalities, and injuries—distribute spatially across the U.S., and what is the severity along their approximate paths?**
   To explore this, I built an interactive Folium-based map that interpolates tornado paths and visualizes them by EF rating, with start/end markers containing detailed weather popups. This allowed users to observe regional severity patterns, EF rating progression, and contextual damage clusters. We found that the most severe tornadoes (by fatalities or width) tend to cluster in the Southeast and Midwest, with visual distinctions between long-track and high-impact events.

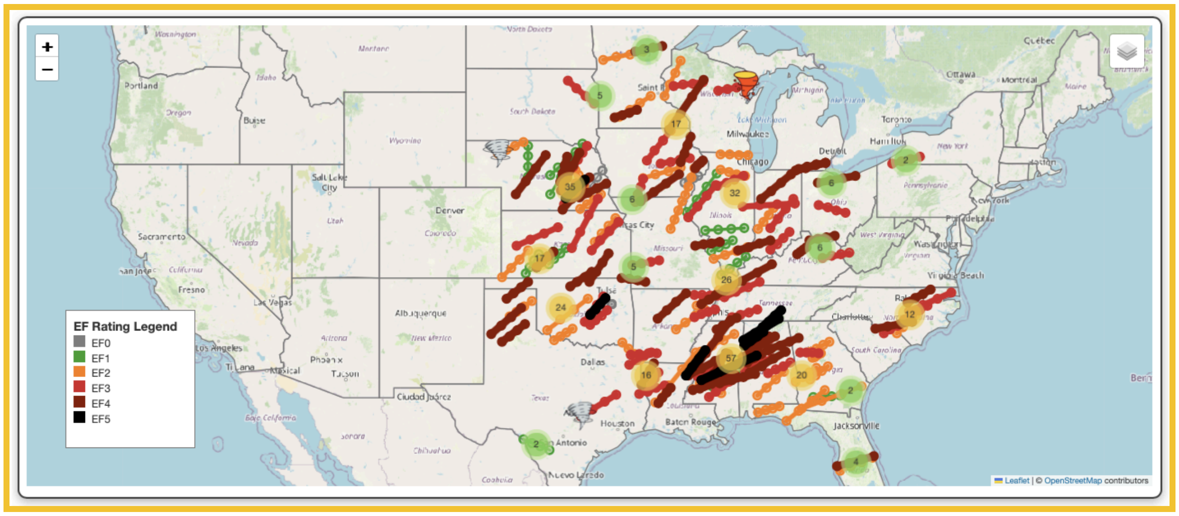


Figure: Top 150 Tornadoes by Length

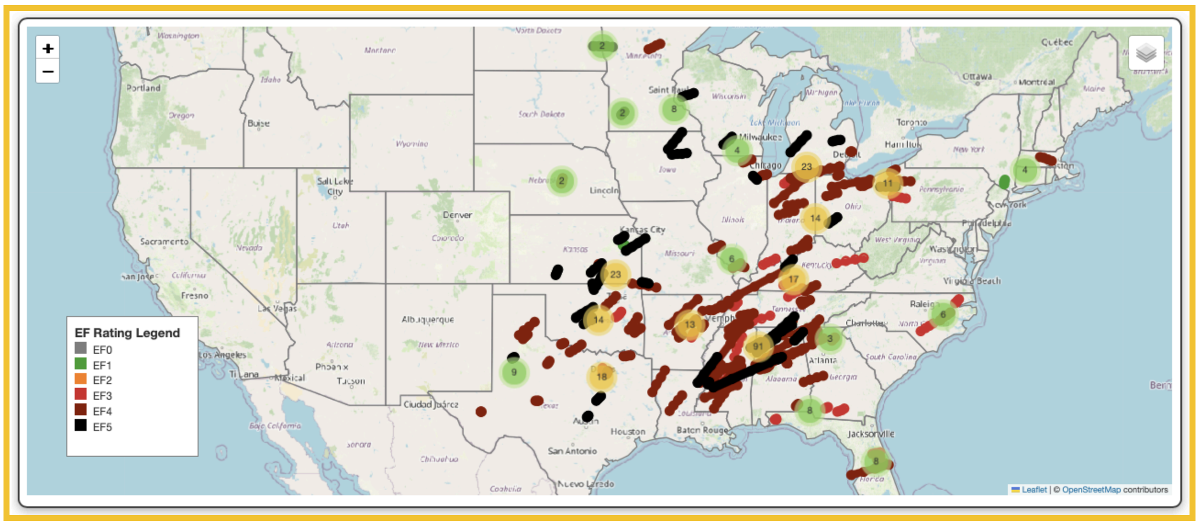


Figure: Top 150 Tornadoes by Width

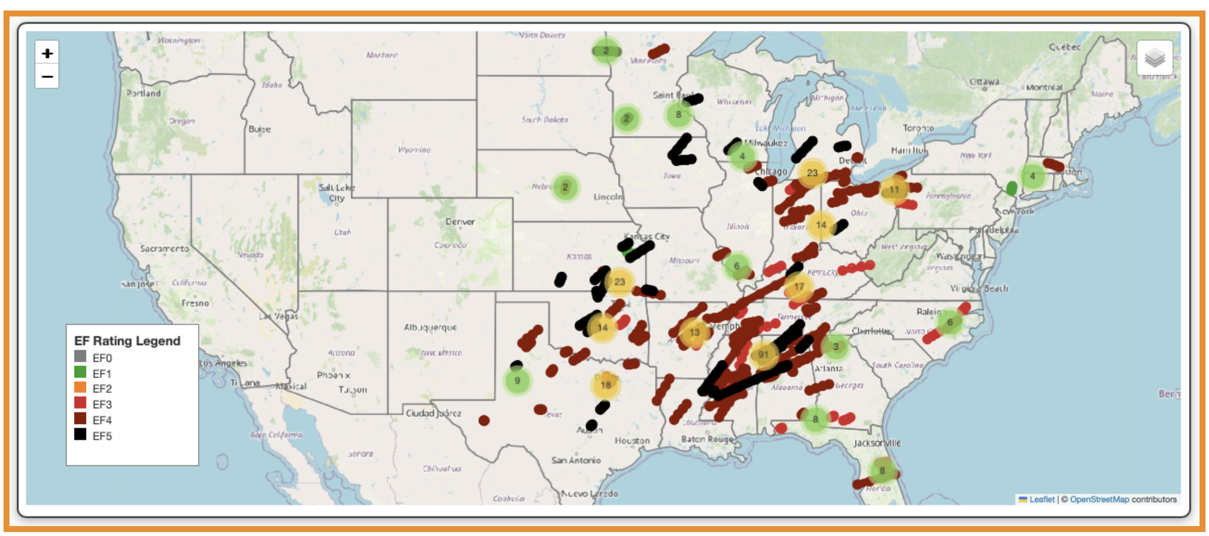


Figure: Top 150 Tornadoes by Fatalities

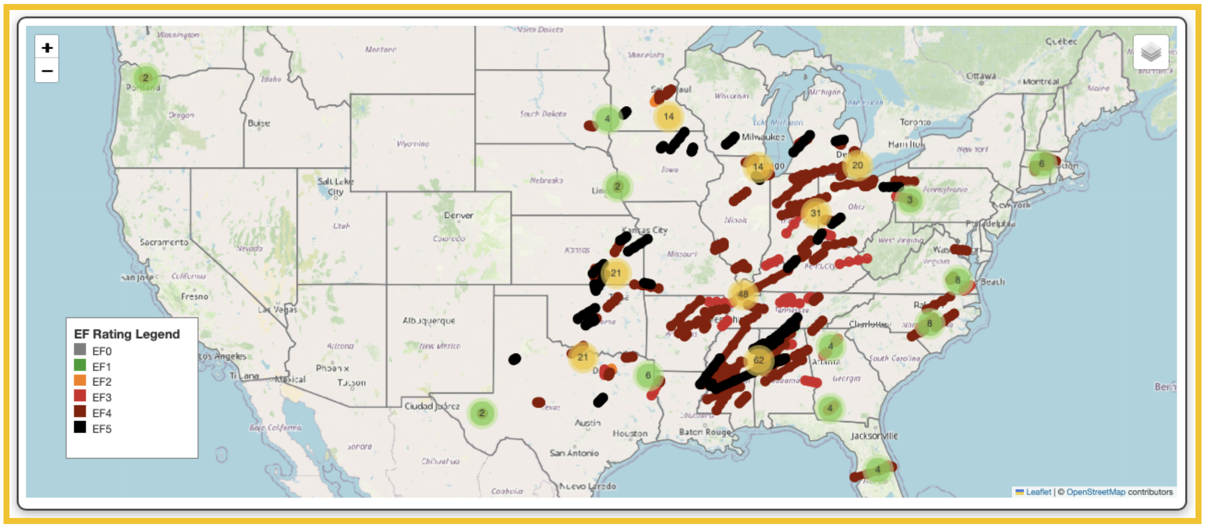
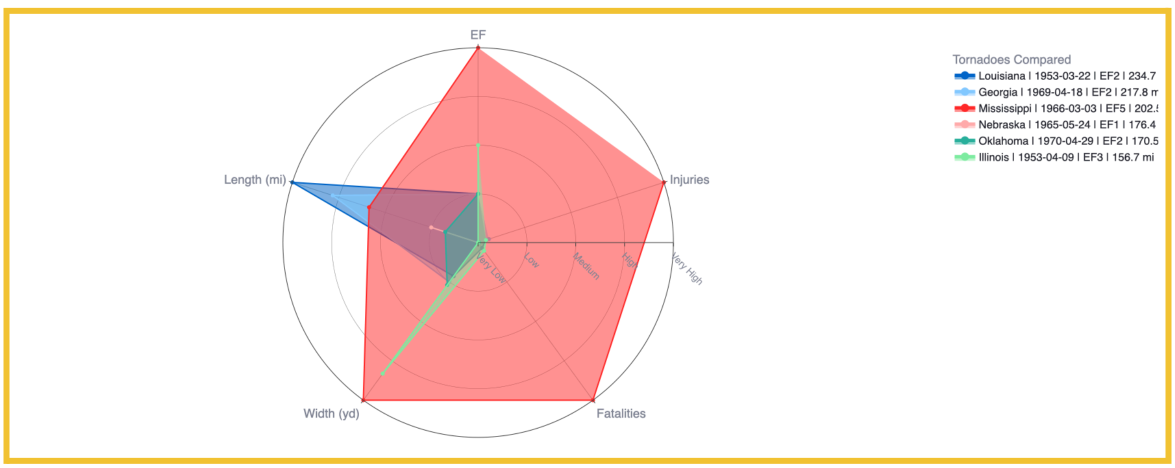


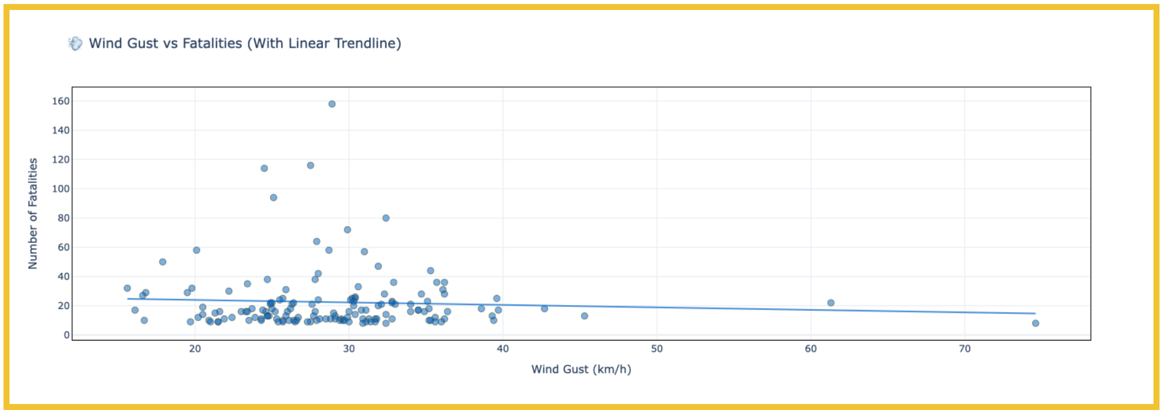
Figure: Top 150 Tornadoes by Injuries

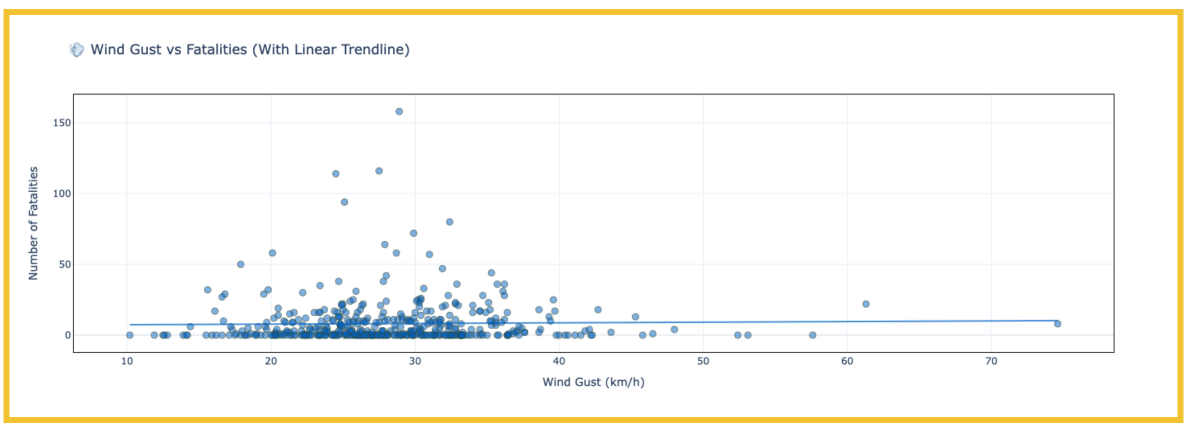
2. **How do different tornadoes compare across key metrics such as EF Rating, Length, Width, Fatalities, and Injuries?**
   To facilitate comparison, I implemented a radar-style spider chart that normalizes tornadoes across all five severity dimensions. This revealed outliers—like tornadoes with high human impact but low EF rating—and allowed users to visually contrast long, narrow tornadoes with short, wide, or deadly ones. A geographic radar view also lets users compare selected tornadoes spatially.

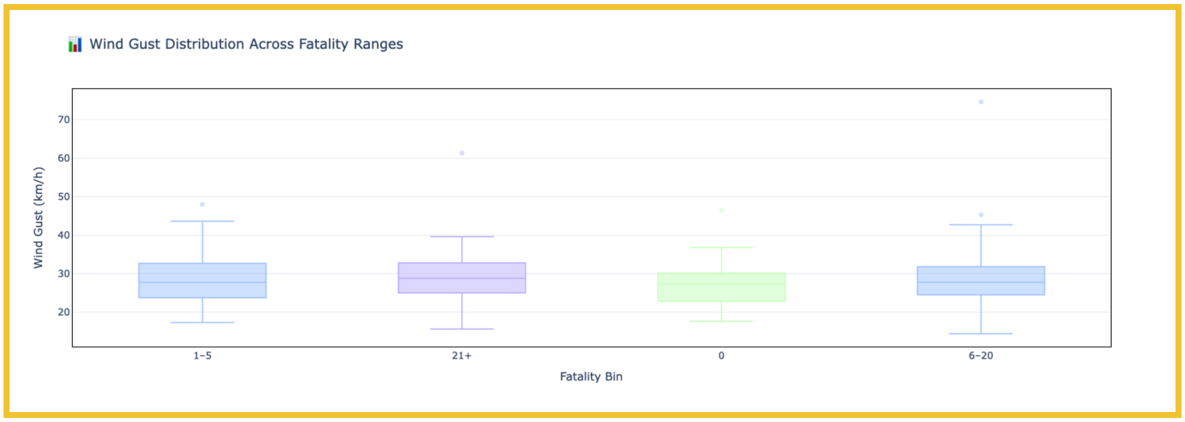


3. **Do higher surface wind gusts correlate with more fatalities?**
   Hypothesis: Tornadoes forming on days with intense surface winds would lead to greater damage and casualties ([NOAA](#) and [Wikipedia](#)).
   Finding: Across both the top-150 fatal tornadoes and the broader 457-tornado dataset (unique top 150 from all metrics), scatter and box plots showed no strong correlation. This suggests that human impact is more context-driven than driven by wind strength alone.

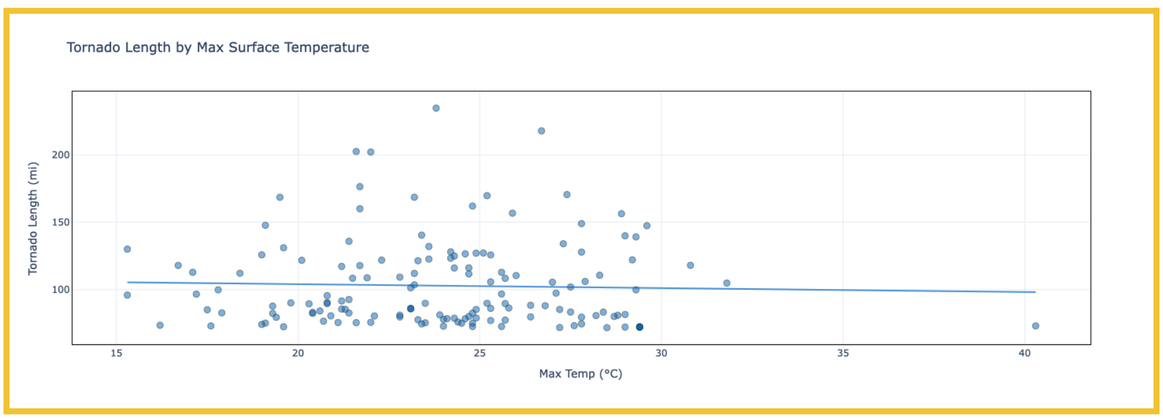
Below the first two figures are for Top 150 and the last two figures are for Top 457.

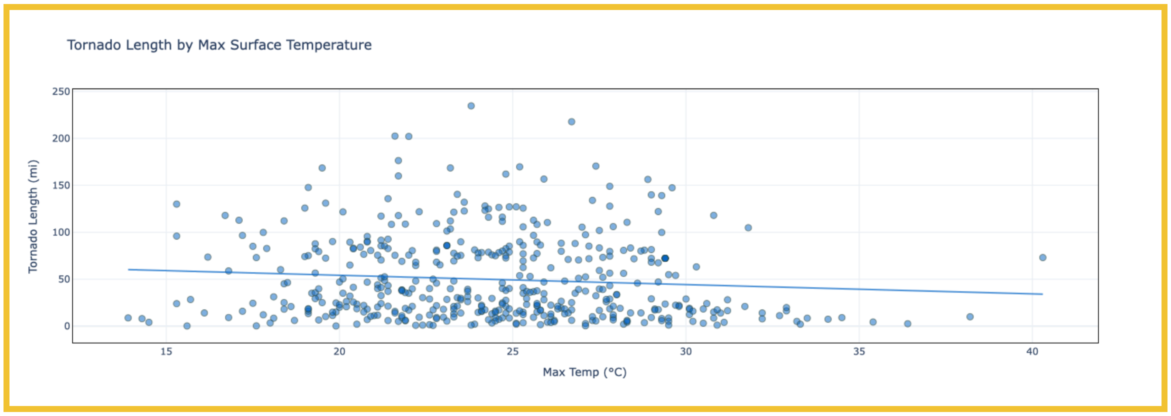


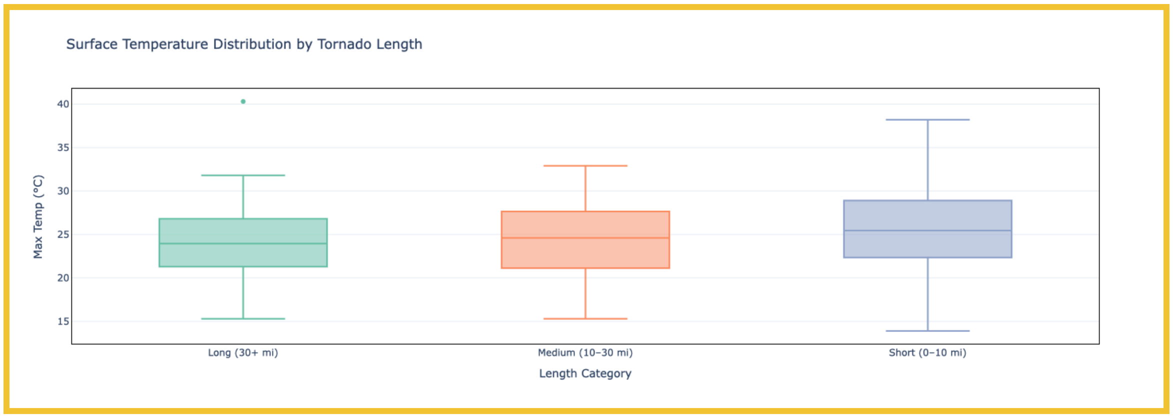




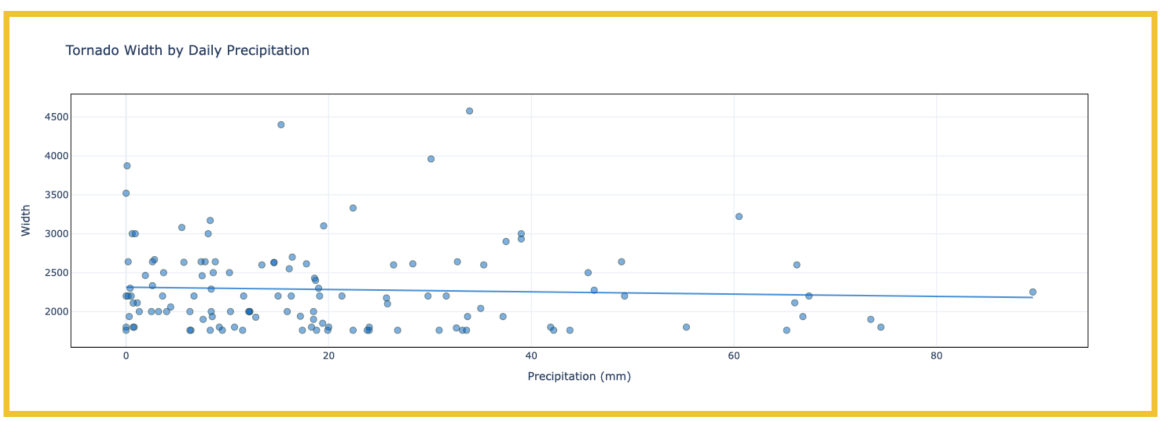
4. **Do higher surface temperatures correlate with longer tornado paths?**
Hypothesis: Warmer surface temperatures, linked to atmospheric instability, would support sustained tornado tracks (NOAA, C2ES, and NIU Newsroom).
Finding: Unexpectedly, long-track tornadoes occurred frequently on moderate-temperature days. Visual trends revealed a weak or even slightly negative correlation, highlighting the need to account for additional factors like wind shear or storm organization. **Below the first two figures are for Top 150 and the last two figures are for Top 457.**

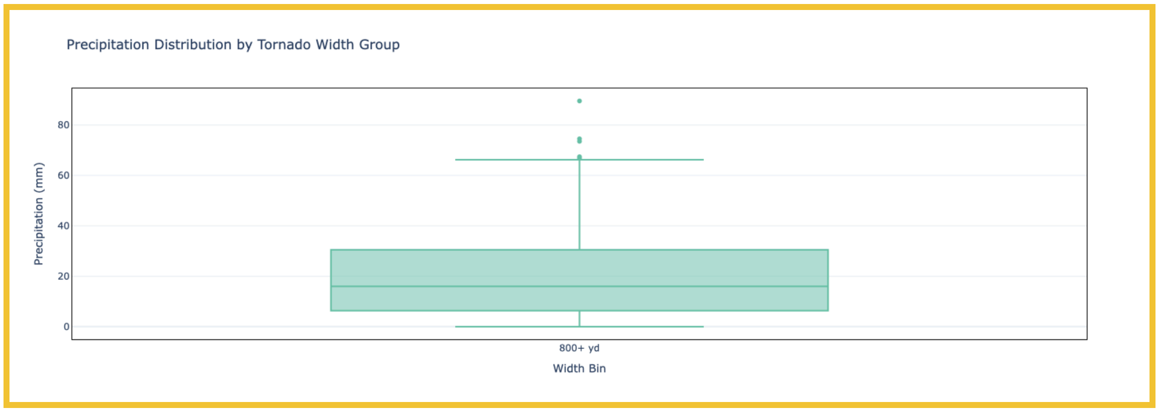


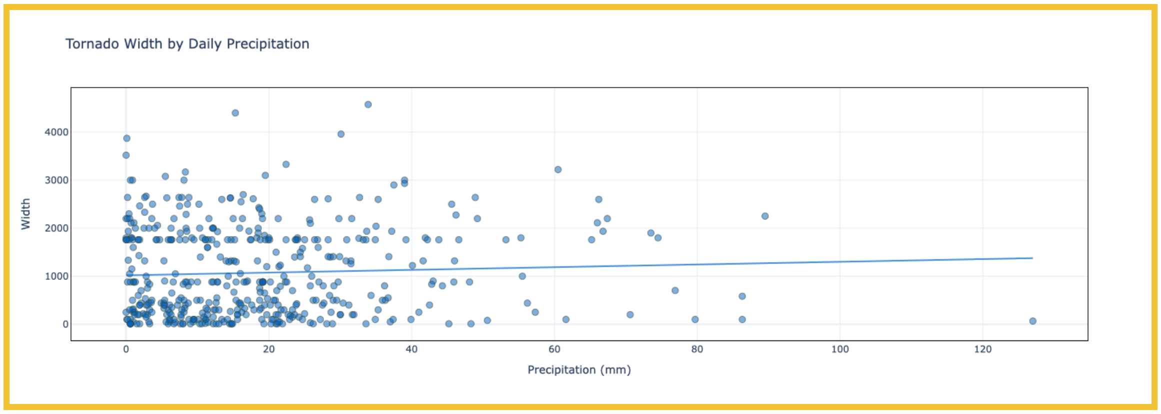


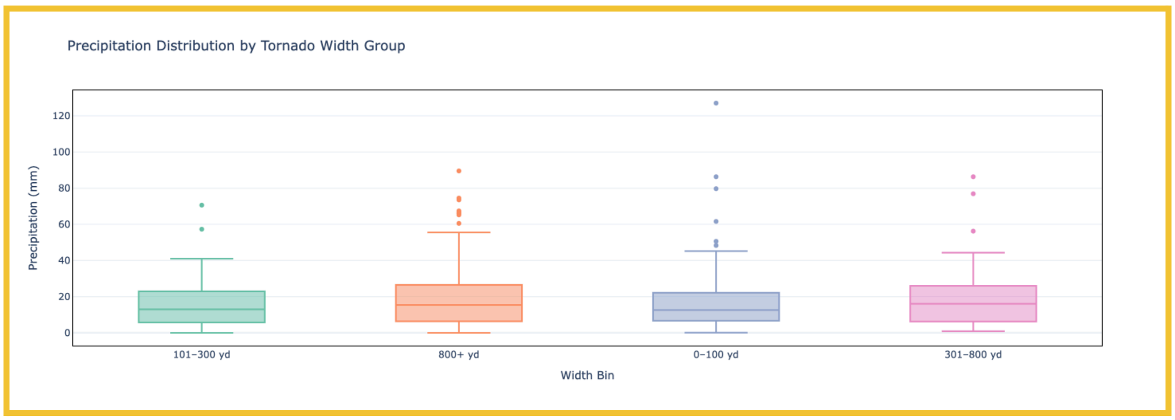


5. **Do tornadoes on rainy days tend to have wider paths?**
Hypothesis: Precipitation may signal convective intensity and thus wider tornado footprints (Joshua et al. and weather.gov).
Finding: While some wide tornadoes occurred during high rainfall, no consistent trend was found. Box plots showed overlapping distributions across width categories, suggesting precipitation alone is not a predictor of tornado width. **Below the first two figures are for Top 150 and the last two figures are for Top 457.**

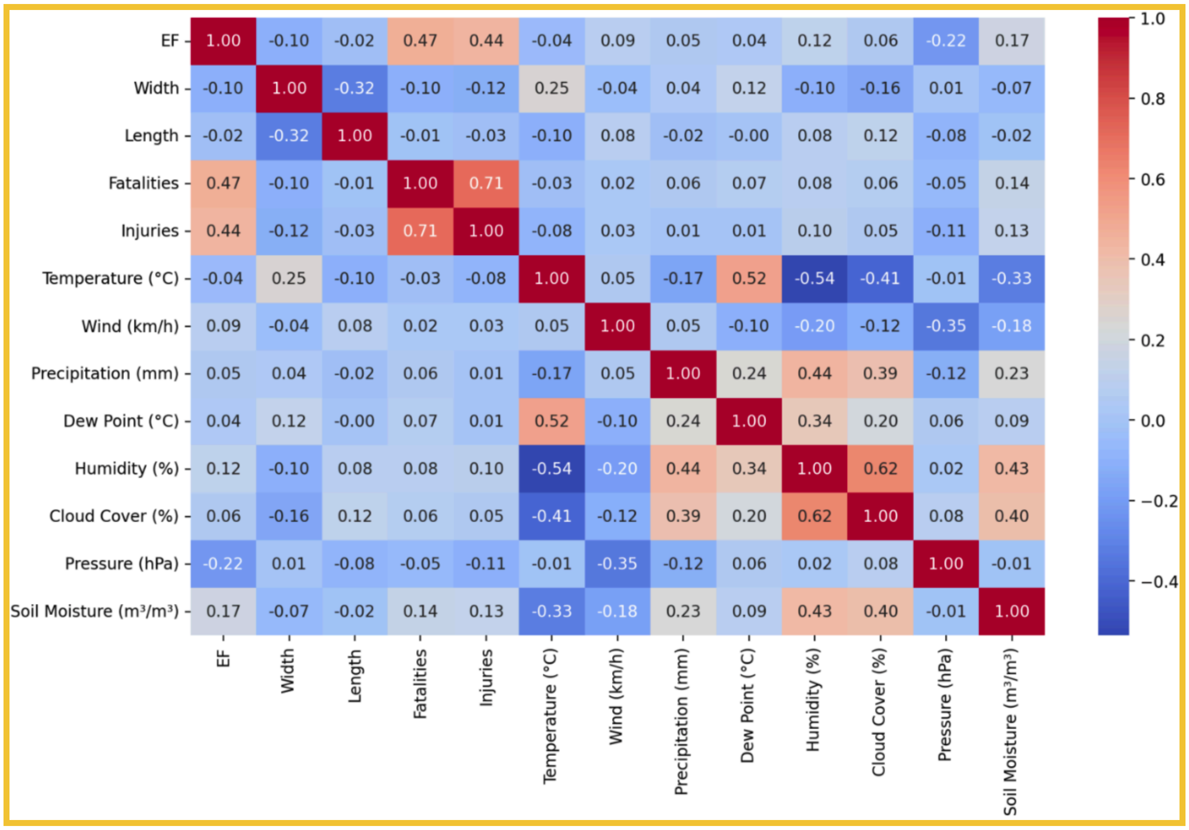








To validate these findings at scale, I **constructed a correlation matrix comparing all tornado and weather variables**. This confirmed strong relationships between EF rating and casualty metrics, while weather variables like wind, precipitation, and humidity showed weak or inconsistent associations—reinforcing the conclusion that tornado severity is multifactorial.



These evolving scientific questions shaped my contribution to both the analytical framing and user interface design. I developed interactive map layers, weather-enriched popups, and multi-metric comparison charts to bridge data-driven insights with intuitive exploration tools.

Implementation Details

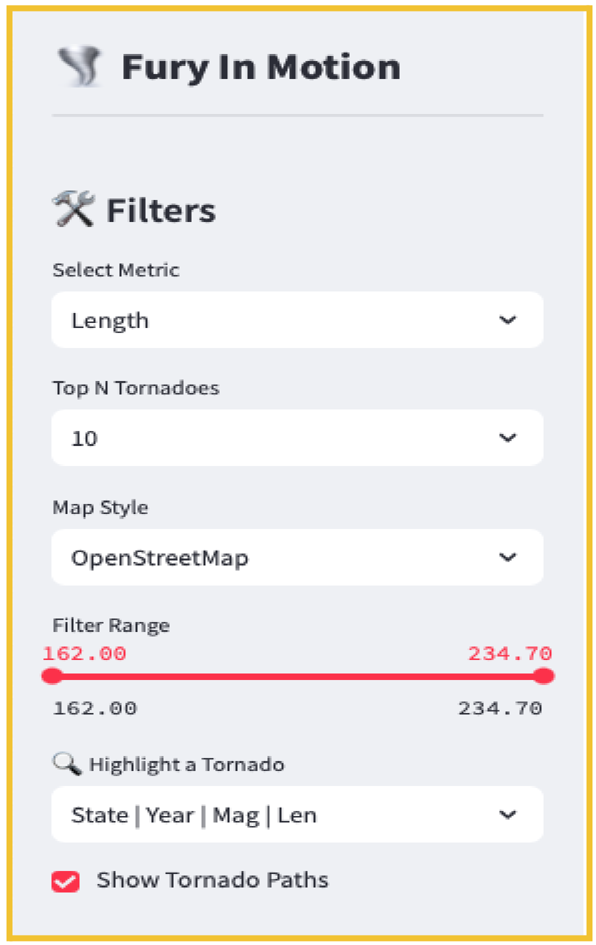
The project was implemented using Python, leveraging a modular architecture to ensure flexibility and scalability. We used **Streamlit** to build an interactive, browser-based frontend, which allowed users to dynamically explore tornado severity across various metrics. The core visualization engine was powered by **Folium** for geospatial rendering and **Plotly** for interactive scatter plots, box plots, and radar-style comparison charts.

To facilitate reproducible analysis and fast rendering, the dataset was preprocessed and augmented through multiple steps:

- A **prefetching script** was created to fetch historical weather data (e.g., temperature, precipitation, wind gusts, and humidity) using the **Open-Meteo API** for 457 unique tornadoes. These included the top 150 tornadoes each by length, width, fatalities, and injuries. The results were cached locally as a weather_cache.json file to avoid redundant API calls and ensure stability during visualization.

- The **NOAA SPC tornado dataset** was cleaned, enriched with weather attributes, and then linked to geographical coordinates for visualization.
- A modular folder structure was followed:
  - utils/ handled reusable data-loading, geo-coordination, and weather utilities.
  - components/ contained visualization modules, including folium_map_render.py, folium_radar_map.py, and scientific exploration scripts under science_questions/.
  - cache/ stored precomputed weather data.
  - data/ housed static files such as the tornado dataset and GeoJSON county boundaries.

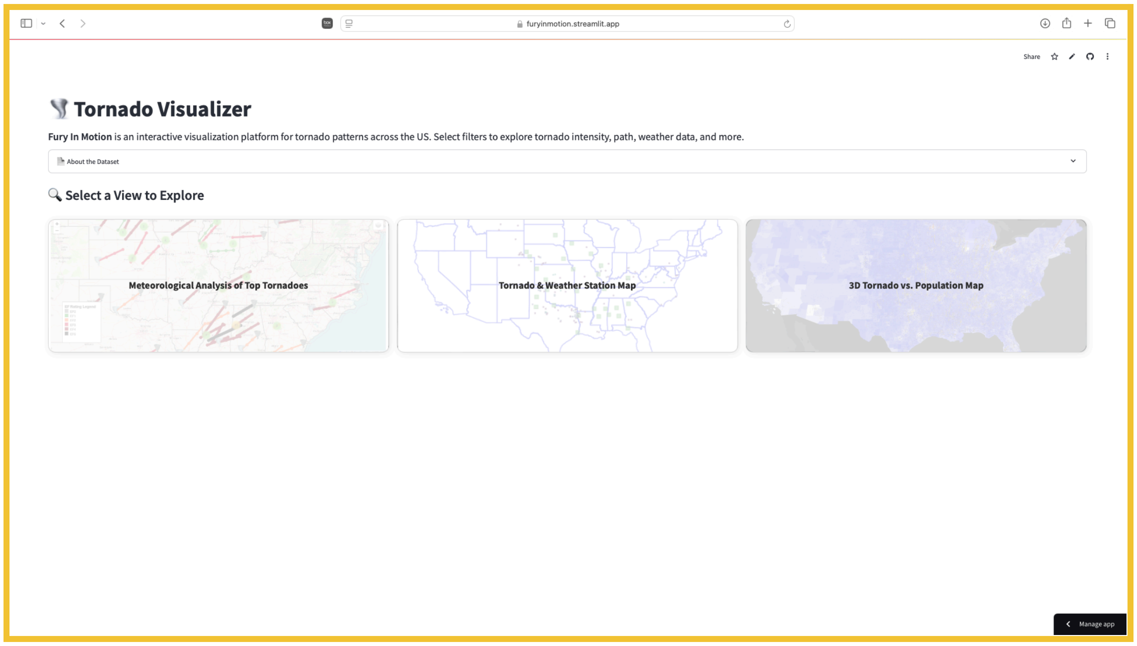
The application includes a **filter sidebar** (see the right Figure) where users can select metrics (e.g., width, fatalities), top-N tornado thresholds, custom map styles, and dynamic range sliders to filter by severity. Users can also highlight individual tornadoes and toggle segmented path displays. The interface dynamically updates the map based on selected filters.



Two main visualization modes were developed:

1. **Top-N Tornado Explorer** – Visualizes tornado paths colored by EF rating, along with weather summaries embedded at start/end points.
2. **Scientific Question Explorer** – Allows statistical investigation of hypotheses such as *"Does surface temperature correlate with tornado length?"* or *"Are high wind gusts linked to fatalities?"* using visual encodings like scatter and box plots.

Lastly, a **dashboard** was created to consolidate the platform, featuring a landing page with tab-based navigation and dataset summaries. The user can seamlessly switch between task views, access metadata, and explore results with minimal friction.



Sam:

- The main map visualization is implemented by putting the main dataset into a matplotlib map with GeoPandas to display all the tornadoes. This is whittled down to the tornadoes from a certain month and year that are selected by the user with an Interact intslider widget. Tornadoes can also be selected by magnitude.
- Weather station data is represented as squares on the map that the user can hover over for information on that month's weather in that area. If there isn't any data, the hover-over won't display anything but the program keeps running. Sometimes there's not data since the weather station time periods are somewhat erratic and sometimes missing data entries.
- The secondary plot is a simple breakdown of the dataset into sums of tornadoes per year. The magnitude selector is the same as with the main map.

Matthew:

- For the county coloring, I went through several iterations exploring whether it would make more sense to use a gradient color scheme or whether it made more sense to do some kind of binning. I ended up performing a logarithmic transformation on the population data and creating seven equal-width bins. I liked the result because it helped to bring some contrast to the visualization but gave enough distinction between more and less populous counties. A good indication is when looking at Utah, Salt Lake County has a different color than Utah or Davis County. Both of those vary drastically from more populous cities like New York or L.A., and less populous counties like Cache or Franklin County.

- As for the county lines themselves, I was able to use a TIGER Line shapefile provided by the US Census Bureau which contained a specific geometry for each county. However, using the full geometry overwhelmed my jupyter notebook's memory, so I had to simplify the geometry. This was done using a built-in GeoJSON function called simplify which allowed me to set a tolerance of one kilometer.
- Finally, the numbers of fatalities and injuries are represented using a geo-referenced stacked bar graph. The bar's center for each tornado is drawn at the tornado's beginning latitude and longitude. Pydeck did not offer a true stacked bar graph layer, so I actually overlaid two column layers one on top of the other with the radius of the fatalities column being slightly smaller than that of the injuries column. Additionally, the height of the fatalities column was increased by the number of injuries in order to maintain the appearance of a stacked column.

What We've Learned

Sam:

- Working with data is usually harder than it initially sounds. A lot of the time, your first attempt at a visualization bugs out, or doesn't communicate the point well, or is just plain cluttered. It's a good idea to keep iterating and trying new things.
- Matplotlib and Jupyter Notebook in general has a *lot* of functionality! It's a great tool.
- For all the benefits of an interactive visualization, sometimes a good graph is still the best option. It's important to really analyze what you're trying to communicate and then form a visualization that best fits that, rather than go in with an initial idea you refuse to change.
    - Poor documentation is really, really annoying. For some of the weather station datasets it is very unclear what the columns are actually recording, random months are missing, some data fields are recorded for only half of the time period, etc. Lots of problems. Really makes me appreciate a well-managed dataset.
- Tornado frequency has shifted over time, but actually hasn't increased as much as I thought it would have.
- Tornado activity is heavily clustered in the spring/summer months, with hardly any tornadoes happening over the winter.

Munshi:

1. Working on the scientific side of the visualization, I learned how important it is to validate assumptions with data—many of our initial hypotheses (such as "higher temperatures lead to longer tornadoes" or "stronger wind gusts cause more fatalities") did not hold up under analysis. This reminded me how often meteorological impacts are multifactorial and context-driven rather than being dictated by a single variable.

2. Integrating weather data from external APIs was technically rewarding but also introduced challenges, such as request limits and variability in weather records. Building a caching mechanism helped me appreciate the need for efficient, reproducible data

pipelines in scientific workflows.

3. Designing interactive elements—like the EF-colored tornado paths, weather-infused markers, and radar-style comparison tools—taught me how interactivity can drive deeper user engagement with complex spatial data. However, I also saw how too many features can overwhelm the interface, so I learned to prioritize clarity in both layout and chart design.

4. Finally, this project taught me how powerful correlation matrices and small-multiple visualizations can be when trying to draw broader conclusions. They helped verify that our top-N patterns weren't just anecdotal—they were statistically consistent across the full dataset.

Matthew:

- Learned about geojson formatting and how shapes such as county lines can be drawn on a map using lat/lon coordinate pairs. Also learned about some neat functions to simplify the geometry of a polygon in order to reduce geojson file sizes.
- Learned how to associate other data with lat/lon coordinates so that it could be visualized on a map.
- Sometimes having a visual representation of where a piece of data is referring to can be more helpful than just having the data tied to the name of a place.
- Also, layers on a map provide a very interesting way of providing multiple dimensions of data simultaneously to a viewer.
- Finally, it does seem that more populous areas are more prone to injuries or fatalities caused by tornadoes, however, the vast majority of tornadoes do not cause any damage in terms of human injuries or fatalities.

Differences Between Final and Initial Plan

Munshi:

- Initially planned to visualize top tornadoes based on length, width, fatalities and injuries and try to get a scientific output. Later thought of adding historical data to see whether any weather conditions changed during the tornado events. Later thought of a way of interactively comparing two or more tornadoes shown on the map.

Sam:

- Originally I wanted to do a lot more with many different meteorological factors that could influence tornadoes, but the actual implementation of my visualization was harder to do than I thought it would be and I had to scale back my ambitions a bit. I tried to round out my section by producing another plot investigating trends in tornado frequency over time, and playing to Jupyter notebook's interactive strengths. My initial plan also had a lot less weather stations, but as I investigated the tornado data more thoroughly I decided to add more.

- Overall my initial plan was pretty unfocused and I refined my goals a lot as I developed the visualizations, eventually focusing on looking at how the data changed over time since my two teammates were looking at other areas.

Matthew:

- I had originally intended to investigate some insurance datasets in order to determine the financial impacts of tornadoes. However, I ended up investigating more about whether the size of a population had an effect on the damage the tornado caused. Additionally, I had intended to display population in some kind of heatmap based on city populations, but found county populations to be accurate enough for my needs without being overly large.

Project Evaluation

Overall:

Overall, our project was successful in achieving its goals of visualizing U.S. tornado data in a meaningful, engaging, and scientifically grounded way. Each team member brought a unique perspective to the project—ranging from spatial impact analysis and population overlays to environmental correlation studies—enabling us to explore tornado behavior across multiple dimensions.

A core strength of the project lies in its modular design and ability to communicate complex data through interactive, map-based visualizations. The platform supports layered exploration of tornado severity using EF-colored path renderings, weather-enriched popups, radar-style comparisons, and demographic overlays. By combining different data sources—NOAA SPC tornado records, Open-Meteo weather APIs, Census population data, and NOAA weather stations—we enriched the raw dataset with context that added interpretability to both spatial and statistical trends.

From a scientific standpoint, the project delivered a mix of confirmatory insights and surprising null results. While some hypotheses, such as the correlation between EF rating and casualty metrics, held strong across datasets, others—like the influence of wind gusts or precipitation on impact severity—proved weaker than expected. Rather than detracting from the project, these findings emphasized the importance of data-driven validation and highlighted tornado severity as a multifactorial outcome.

The project also faced challenges, including dataset inconsistencies, visualization complexity, and limitations of third-party libraries (e.g., lack of built-in stacking for bar charts, API rate limits). However, we addressed these through preprocessing strategies, visual design iterations, and thoughtful simplification.

In the end, the platform provides users with a flexible, layered interface for exploring one of the most destructive natural phenomena in the U.S. While there's still room for extending interactivity and incorporating new disaster types or damage datasets, the current system successfully bridges scientific inquiry with accessible visualization—a core goal of any scientific visualization project.

Munshi:

From my perspective, the project was quite successful in achieving both our visualization and scientific goals. I set out to investigate whether real-world weather conditions had measurable effects on tornado severity metrics, and was able to integrate historical atmospheric data for 457 significant tornadoes across multiple impact categories. This enriched the dataset meaningfully and allowed for hypothesis-driven analysis supported by interactive, user-friendly visualizations.

One of the major strengths of my contribution was the ability to bridge meteorological context with spatial tornado dynamics. Tools like the radar comparison chart, EF-segmented path visualizer, and weather-enhanced map markers helped translate complex data into actionable insight. I believe the correlation matrix was another high-impact addition, providing a statistical foundation to support or reject our visual hypotheses.

That said, there were some limitations. The biggest weakness was that many of the weather variables I expected to be predictive (like wind gusts or surface temperature) turned out to have little to no correlation with impact metrics like fatalities or path length. This wasn't a flaw in implementation, but it did challenge the scientific narrative I initially hoped to build. Also, integrating weather data via the Open-Meteo API introduced occasional inconsistencies due to missing values or temporal resolution gaps, and required extra preprocessing to ensure consistency across tornado records.

Despite these limitations, I consider my portion of the project a success in terms of both analytical rigor and visual communication. It demonstrated how enriched geospatial data can uncover surprising insights—and just as importantly, reveal when expected patterns don't exist. These findings contributed a unique layer to the team's broader narrative and reinforced the value of combining scientific questioning with thoughtful visualization design.

Sam:

- I think my section was pretty successful. My main visualization allows a viewer to look at the distribution of tornadoes on a very intuitive map display, and freely change the year and month. This is great for tornadoes in particular since are much more frequent in certain months of the year, so being able to smoothly go from May 1990 to May 1991 and so on allows a viewer to better observe long-term trends in similar seasons.

- The weather station implementation helps point out particular meteorological conditions in certain regions for the given month, which is great for context. It's based on a hover-over interaction, which is intuitive, but doesn't give the full rundown on the landscape. That feels like a bit of a weakness. If I were to do this project again, I would try to either find denser data (maybe satellite climate analysis?) or produce some sort of color map over the map itself based on the weather station values. I briefly considered that for this project but it was too complex to implement in the timeframe with my skills.
- The additional plot adds a simple way to view how the amount of tornadoes can vary quite a bit year-to-year. Honestly, the amount it varies makes me wonder if there are longer-term patterns to it all, and I'd love to investigate more, but our dataset only goes back to 1950.

Matthew:

- I was very happy with the visualization I ended up with. It is able to display multiple dimensions of data simultaneously at a glance. However, it could feel a bit overwhelming and uninteresting after a while, because it does not present the user with many interaction capabilities. I did try to limit the overwhelming feeling a user might experience by displaying tornadoes causing no injuries/fatalities with blue dots so they would blend in, but I would have liked to include sliders or filters similar to Munshi's visualizations in order to present the user with more interactive capabilities. Things that would help them to narrow the focus of the visualization to specific years, tornado strengths, or damage levels, or improve the visual quality for their needs by adjusting the size of the bars and the opacity of the map layer.
- In terms of answering my research questions, I felt like the visualization really measured up to the task. Users are able to quickly identify areas of high/low population or of particular interest to themselves. They can tell at a glance if the area has seen a high or low number of tornadoes and where the most deadly ones struck. And users are able to hover individual tornadoes or counties to get more information about them, such as the year the tornado struck or the name and population of the county.
- If I were to do it again, I would like to include more of what I had initially set out to display, which is financial damages caused by a tornado. But I also think that we could expand our dataset with relative ease to include other natural disaster data such as tropical storms, hurricanes, typhoons, and wildfires.

Work Distribution

We approached this project as a collaborative effort with a shared goal: to uncover diverse scientific insights from U.S. tornado data through visual exploration. Each of us focused on different angles, which allowed us to maximize the depth of our findings while maintaining individual ownership of our analysis.

- **Sam** focused on temporal and meteorological trends, exploring how tornado patterns have changed over time and whether variables like temperature or precipitation had any observable effect on their frequency or distribution.
- **Munshi** integrated historical weather data from the Open-Meteo API, aiming to understand whether atmospheric conditions such as surface temperature, wind gusts, and precipitation correlate with tornado severity metrics like path length, fatalities, and injuries.
- **Matthew** worked on geographic and demographic insights by incorporating U.S. Census data and geoJSON county boundaries, enabling the team to visualize tornado impact in relation to population density and spatial regions.

In terms of implementation, each of us developed separate interactive maps or visual modules tailored to our scientific questions. These individual components were later unified into a dashboard-style interface, where users can switch between views and comparisons seamlessly.

For the final report and presentation, we divided responsibilities evenly. Each team member authored sections related to their own contributions, while the shared portions—such as the background, tools used, and overall evaluation—were written collaboratively. This structure allowed us to maintain both consistency and individuality across the project deliverables.

Softwares Used in This Project

This structure allowed us to maintain both consistency and individuality across the project deliverables. Our project is built on Python and the relevant packages such as: Streamlit, matplotlib, folium, numpy, folium.plugin, plotly, plotly.graph_objects, plotly.express, pandas, seaborn, geopandas, shapely, ssl, certifi, urllib.request, pydeck, json, tqdm, Dict, Tuple.

How to Reproduce the work

1. Clone this GitHub repository using:
   git clone https://github.com/munshisaifuzzaman/Fury-In-Motion.git
2. Now go to the main project directory and run the following:
   Streamlit run app.py
3. The rest will be guided by the output.

Code Repository

GitHub Link, Project Live Link

Additional Comments

"The storm never lasts forever. Bad weeks come and go.

You can't control the wind—but you can control your sail."