T1-24-25-DAS 732-Data Visualization Assignment 2: Scientific and Information Visualization

Udayagiri Narayana Srimanth
IMT2022052
Udayagiri.Srimanth@iiitb.ac.in

Khadarabad Tahir Mohammed
IMT2022100
Khadarabad.Mohammed@iiitb.ac.in

Pradyun Devarakonda
IMT2022525
Devarakonda.Pradyun@iiitb.ac.in

***Index Terms*—gridMET Dataset, Heatwaves, CPAN Networks, Twitch Analytics, InfoVis, Streaming Trends**

I. Introduction

In this assignment, we explore two distinct visualization approaches: scientific visualization (SciVis) and information visualization (InfoVis), each applied to different datasets with unique analytical objectives.

In the SciVis component, we analyze meteorological variables to examine the record-breaking heatwave in the summer of 2023, which saw unprecedented temperature highs in the American Southwest. Using data such as maximum and minimum near-surface air temperature, surface downwelling solar radiation, mean vapor pressure deficit, energy release component, burning index, FM100, FM1000, wind speed, and wind direction, we created various visualizations including color maps, contour plots, and quiver plots. These visualizations provide insight into atmospheric conditions and correlations between variables, helping to analyze the dynamics of the extreme heatwave.

The InfoVis component includes three tasks that employ diverse visualization techniques to explore different datasets. First, node-link diagrams were created for the CPAN Explorer, a project analyzing relationships between developers and packages in the CPAN (Comprehensive Perl Archive Network) community, using data captured by Linkfluence in July 2009. Second, interactive parallel coordinate plots (PCP) and treemaps were generated for the Top Streamers on Twitch dataset, which includes detailed information about Twitch's top 1,000 streamers by follower count as of August 2020. These visualizations reveal patterns and trends within the dataset, highlighting key aspects of Twitch's streamer ecosystem.

II. Task division

A. SciVis

- Quiver Plots: Narayana
- Color Maps: Pradyun
- Contour Plots: Tahir

B. InfoVis

- Node Link Diagrams: Pradyun
- PCP: Narayana
- Treemaps: Tahir

III. SciVis Dataset

We've analyzed gridMET[1] data from May 1st 2023, to July 31st 2023. One key event in this time period is a heatwave in the Southwest of the US. The temperature in parts of Nevada, Colorado and New Mexico tied to their all time high. In parts of Arizona and Cayman Islands, highest ever night time temperatures in Phoenix Arizona which also had its record for longest time without falling below 90F/32.2C. There were several heat deaths too, including migrants in the US Mexican border.

The heatwave lasted from 1st July 2023 to 31st July 2023. The temperatures peaked around 16th

July. To analyze the meteorological conditions before and after the peak, we've plotted meteorological data obtained from the gridMET[1] dataset into Color, Contour, and Quiver plots.

A. Variables of Interest

These variables theoretically have a correlation with heatwaves:

1) tmmx (Maximum Near-Surface Air Temperature): The most direct indicator of a heatwave is elevated near-surface temperatures. Persistent high maximum temperatures are a primary characteristic of heatwaves.
2) tmmn (Minimum Near-Surface Air Temperature): Increased minimum temperatures, particularly overnight, are common during heatwaves and contribute to the lack of nighttime cooling. This can exacerbate heat stress and is often used to define the intensity of a heatwave.
3) srad (Surface Downwelling Solar Radiation): Increased solar radiation leads to surface warming and can contribute to higher daytime temperatures. High solar radiation can drive heatwaves by increasing the energy available for warming near-surface air.
4) vpd (Mean Vapor Pressure Deficit): VPD represents the difference between the amount of moisture in the air and how much it can hold. Higher VPD indicates drier air, which often occurs during heatwaves as the air becomes less saturated with moisture. High VPD can exacerbate the effects of heat by increasing evapotranspiration rates and drying out vegetation.
5) ERC and BI (Energy Release Component and Burning Index): These indices, used in fire risk assessment, are indirectly related to heatwaves because prolonged hot, dry conditions increase the risk of wildfire. Due to dry vegetation and increased fire potential, elevated ERC and BI values are often seen during and after heatwaves.
6) FM100 and FM1000 (100-hour and 1000-hour dead fuel moisture): Low fuel moisture levels correlate with hot and dry conditions often seen during heatwaves. Heatwaves can dry out vegetation, lowering dead fuel moisture levels, heightening wildfire risk.

Figure 1: Mean Precipitation(mm) from July 10th 2025 to July 25th 2025

Variable Name	Plot Type
vpd (Mean Vapor Pressure Deficit)	Contour Plot (Contour Fill)
pr (Precipitation)	Color Plot (Continuous, Sequential)
tmmn, tmmx (Air Temperature)	Color Plot (Continuous, Sequential)
ERC (Fire Risk Index)	Contour Plot (Marching Squares)
BI (Fire Risk Index)	Contour Plot (Contour Fill)
FM100, FM1000 (Dead Fuel Moisture)	Color Plot (Sequential)

Table I: Variable Names and Plot Types

B. Color Plots

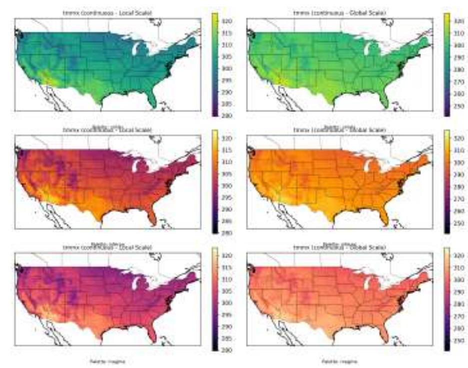
1) Color Plots for general analysis:

- Precipitation Due to July being Summer in the US and the effects of the heatwave, US was relatively dry(Figure.1). The color palette `Blues` was used to plot precipitation as it is generally considered an intuitive color palette for plotting precipitation data. For precipitation, where the data varies smoothly over a range, color plots are highly effective for showing continuous variation in intensity. The `Blues` colormap used in the code is a sequential colormap, which is suitable for representing data with a single range of values (e.g., from low to high precipitation).
- Relative Humidity During the time of the heatwave, there was low relative humidity.

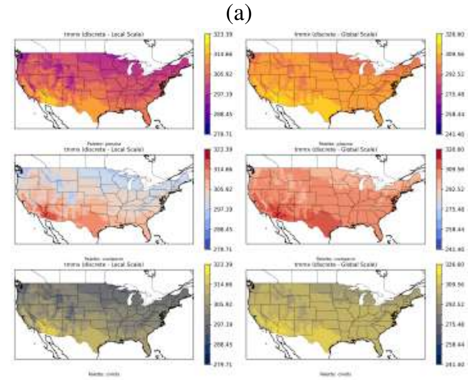
2) Comparison of color plots based on the variable and the type of color plot: We compared the 3 types of color plots: continuous, discrete, and logarithmic. Each plot depicts a type of plot for each variable. Within the plot, we plotted local and global scaling-based color plots and included 3 color palettes. We made plots with the mean of meteorological data from 10th July 2023 to 25th

July 2023, as we observed a peak during that time. We've analyzed 4 variables and plotted color plot comparisons for them:

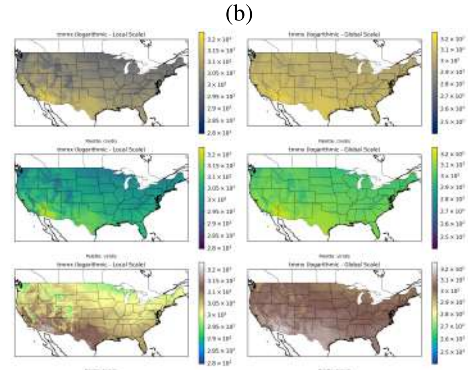
1) **tmmx (Maximum Near-Surface Air Temperature)** Due to the effects of the heatwave, the maximum near-surface air temperature was high in the Southwest, compared to other regions(Figure.2). For the maximum air temperature variable (tmmx), a continuous color plot is most suitable. The recommended color palettes are warm-toned, such as `inferno` or `plasma`(Figure.2). After the comparison, we decided to use the color palette `inferno`, sequential color map type, and parametric mapping with local scaling. Explanation: The maximum near-surface air temperature (tmmx) represents continuous, warm-toned temperature values. inferno is a perceptually uniform sequential colormap that goes from dark purple through orange to bright yellow, ideal for representing rising temperatures in a visually intuitive way (hotter temperatures are brighter). Sequential maps are suitable because temperature is a continuous variable, naturally ordered from low to high. Local scaling, where min and max are calculated from the dataset's range, allows better contrast and visualization for the specific region and time range. The plots in Figure 3 depict tmmx.
2) **tmmn (Minimum Near-Surface Air Temperature)** The minimum near-surface air temperature in the southwest was high, as a result of the heatwave(Figure 4). Minimum air temperature (tmmn) benefits from a continuous color plot. Gradual color palettes like `viridis` or `magma` are ideal (Figure 4).
After the comparison, we decided to use the color palette magma, sequential color map type, and parametric mapping with local scaling.
Explanation: The minimum near-surface air temperature (tmmn) represents continuous, cooler-toned temperature values. magma is a perceptually uniform sequential colormap that transitions from dark purple to lighter yellow,



(a)



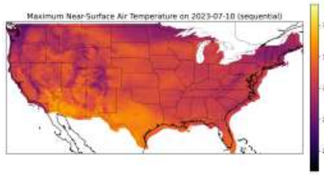
(b)



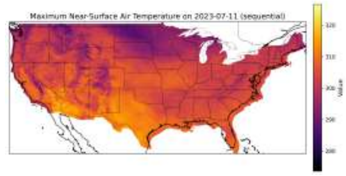
(c)

Figure 2: Plots for Maximum Near-Surface Air Temperature (tmmx) using different color scales, palettes, and color plot types. (a): Continuous, (b): Discrete, (c): Logarithmic. Left: Local Scaling, Right: Global Scaling

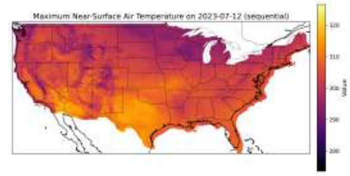
effectively representing increasing tempera-



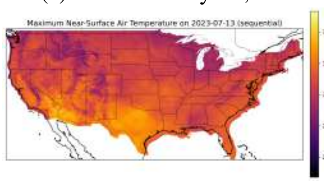
(a) tmmx on July 10, 2023



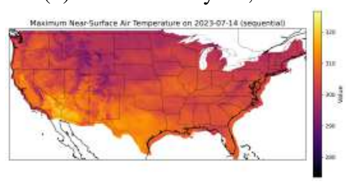
(b) tmmx on July 11, 2023



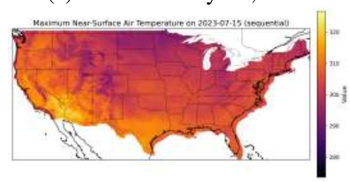
(c) tmmx on July 12, 2023



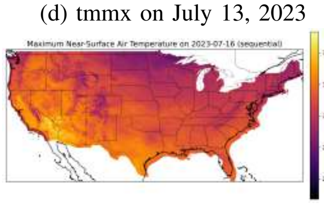
(d) tmmx on July 13, 2023



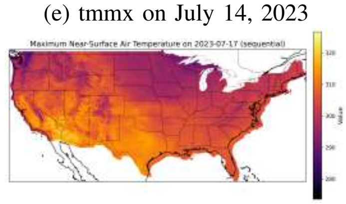
(e) tmmx on July 14, 2023



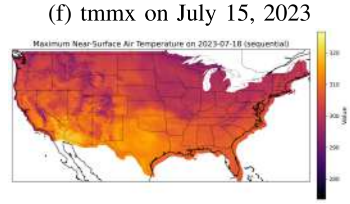
(f) tmmx on July 15, 2023



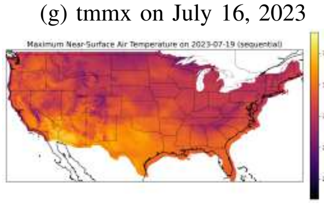
(g) tmmx on July 16, 2023



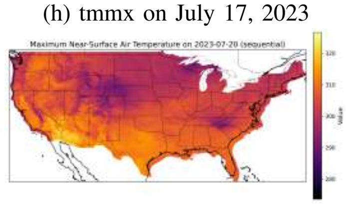
(h) tmmx on July 17, 2023



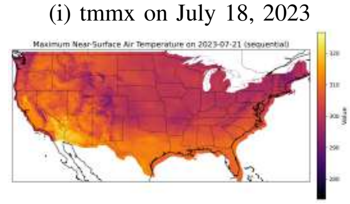
(i) tmmx on July 18, 2023



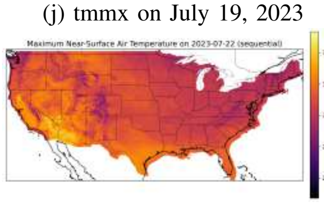
(j) tmmx on July 19, 2023



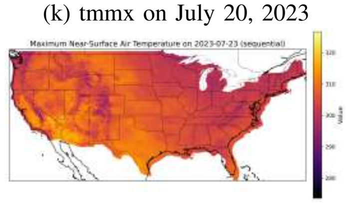
(k) tmmx on July 20, 2023



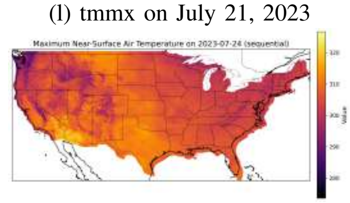
(l) tmmx on July 21, 2023



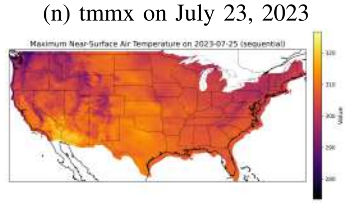
(m) tmmx on July 22, 2023



(n) tmmx on July 23, 2023



(o) tmmx on July 24, 2023

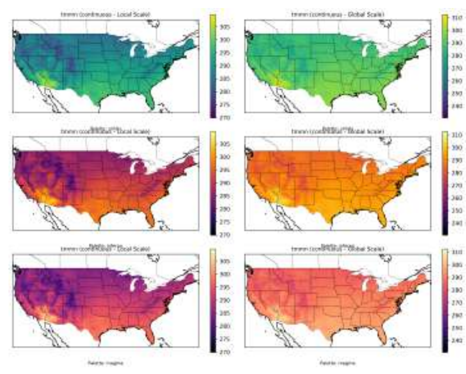


(p) tmmx on July 25, 2023

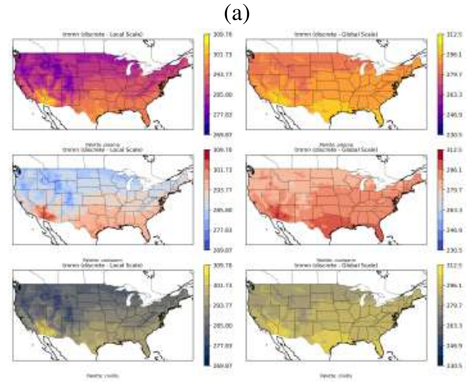
Figure 3: Maximum Near-Surface Air Temperature(tmmx) from July 10th to July 25th 2023

ture values while maintaining a distinct look from inferno, which is used for maximum temperatures. Sequential maps are appropriate here because temperature is a continuous, ordered variable. Local scaling, which calculates min and max from the dataset's specific range, enhances contrast, making it easier to distinguish temperature variations for the given region and time. The plots in Figure.5 depict tmmn.

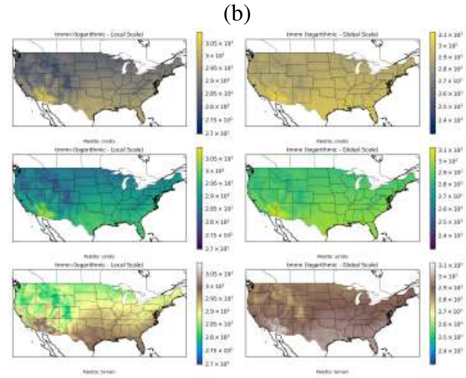
3) **FM100 (100-hour Dead Fuel Moisture)** As expected, the 100-hour Dead Fuel Moisture was low in the Southwest, compared to the rest of the US (Figure 6). For FM100, a discrete color plot is recommended, often with contrasting colors (Figure 6).
   After the comparison, we decided to use the color palette tab10, discrete color map type, and parametric mapping with local scaling.
   Explanation: FM100 represents the moisture content in fuels (such as vegetation) that impacts fire susceptibility. This data benefits from a discrete colormap as it allows clear categorical distinctions between moisture levels, which can be used for interpreting fire risk thresholds (e.g., dry, moist, and wet categories). The tab10 palette, with contrasting colors, enhances this categorization, making it visually easier to identify different moisture levels. Local scaling, where the data's min and max are computed from the specific dataset range, ensures effective contrast within the plotted region. The plot in Figure 7 depicts FM1000 on July 16th 2023.
4) **FM1000 (1000-hour Dead Fuel Moisture)** 1000-hour Dead Fuel Moisture also was low in the Southwest, compared to the rest of the US(Figure 8). For FM1000, a discrete or logarithmic color plot is suitable due to its wide range(Figure 8).
   After the comparison, we decided to use the color palette viridis, logarithmic (sequential) color map type, and parametric mapping with local scaling.
   Explanation: FM1000 reflects moisture content in larger, slow-drying fuels, which can



(a)



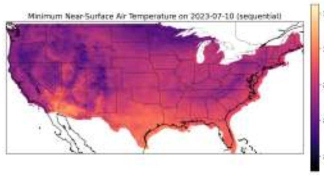
(b)



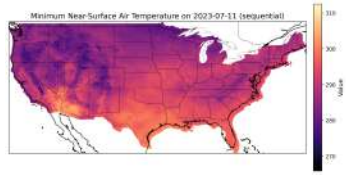
(c)

Figure 4: Plots for Minimum Near-Surface Air Temperature (tmmn) using different color scales, palettes, and color plot types. (a): Continuous, (b): Discrete, (c): Logarithmic. Left: Local Scaling, Right: Global Scaling

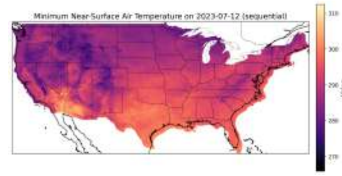
vary widely in value. The viridis colormap is



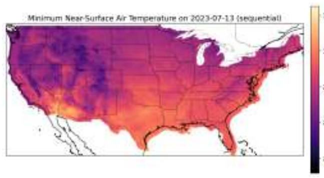
(a) tmmn on July 10, 2023



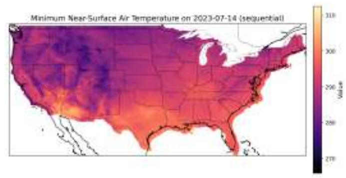
(b) tmmn on July 11, 2023



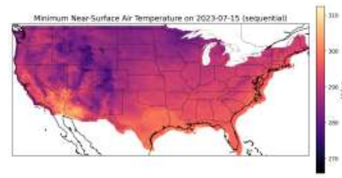
(c) tmmn on July 12, 2023



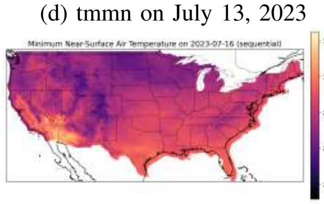
(d) tmmn on July 13, 2023



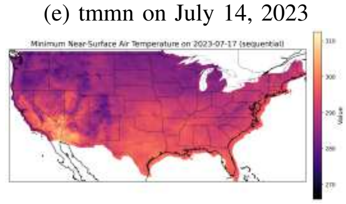
(e) tmmn on July 14, 2023



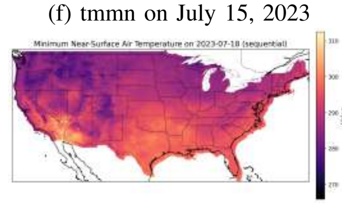
(f) tmmn on July 15, 2023



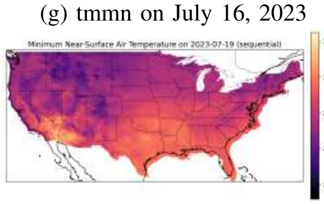
(g) tmmn on July 16, 2023



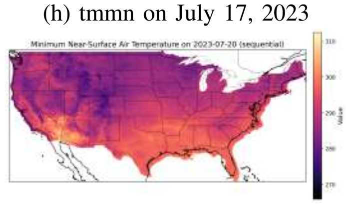
(h) tmmn on July 17, 2023



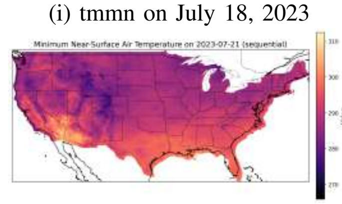
(i) tmmn on July 18, 2023



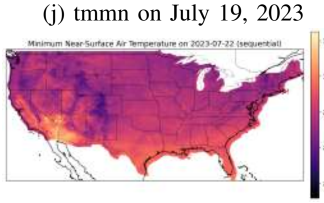
(j) tmmn on July 19, 2023



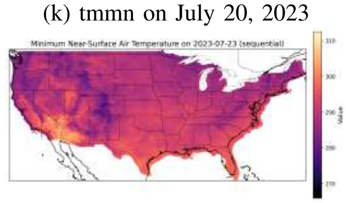
(k) tmmn on July 20, 2023



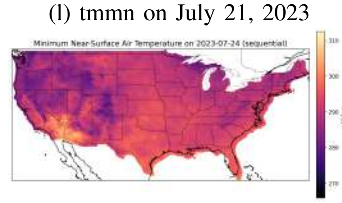
(l) tmmn on July 21, 2023



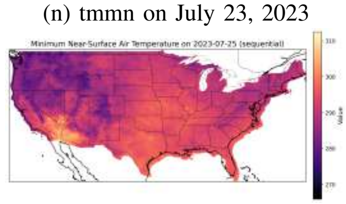
(m) tmmn on July 22, 2023



(n) tmmn on July 23, 2023

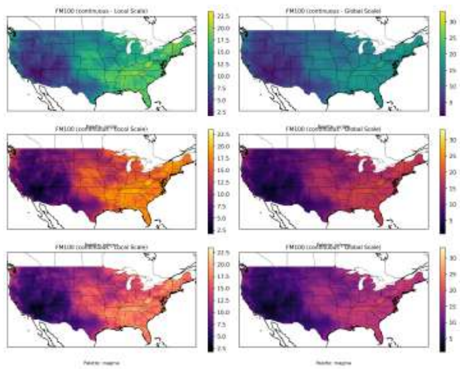


(o) tmmn on July 24, 2023

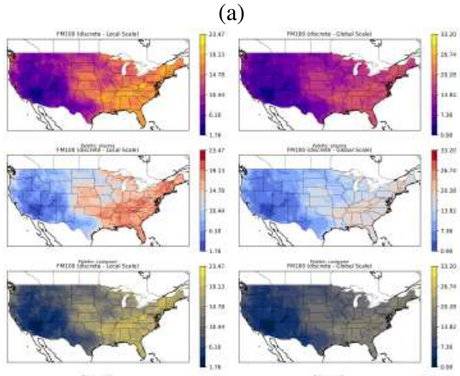


(p) tmmn on July 25, 2023

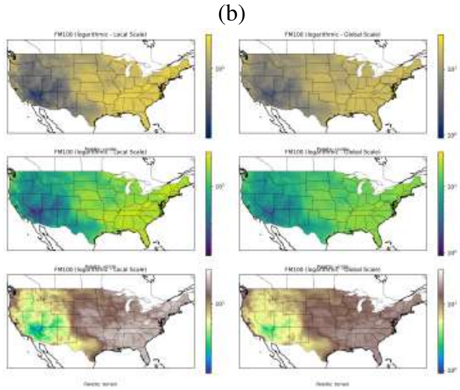
Figure 5: Minimum Near-Surface Air Temperature (tmmn) from July 10th to July 25th, 2023



(a)



(b)



(c)

Figure 6: Plots for 100-hour Dead Fuel Moisture (FM100) using different color scales, palettes, and color plot types. (a): Continuous, (b): Discrete, (c): Logarithmic. Left: Local Scaling, Right: Global Scaling

Figure 7: 100-hour Dead Fuel Moisture on 16th July 2023

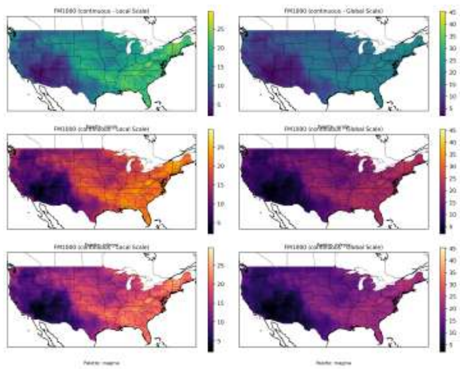
perceptually uniform, transitioning smoothly from dark purple through green to yellow, which is ideal for representing this broad range of moisture levels. A logarithmic color map type is chosen due to the wide range of FM1000 values, allowing better visualization of smaller values while retaining higher-value variations. Local scaling is applied to compute the min and max based on the dataset range, optimizing contrast and readability. The plot in Figure.9 depicts FM1000 on July 16th 2023.

C. Contour Plots

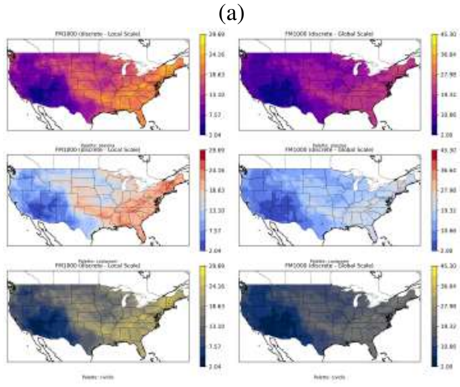
1) Mean Vapour Pressure Deficit: The mean vapor pressure deficit (VPD) can rise during heatwaves([8],[7]).
Explanation: As temperatures rise, the air's saturation vapor pressure increases faster than the humidity, causing VPD to increase exponentially. A 3°C increase in temperature can increase VPD by 45%([10]).

From these plots(Figure 10), we see that the worst-hit area (Southwest US) has had the highest mean vapor pressure deficit (vpd) during the heatwave, thus validating the proposition.

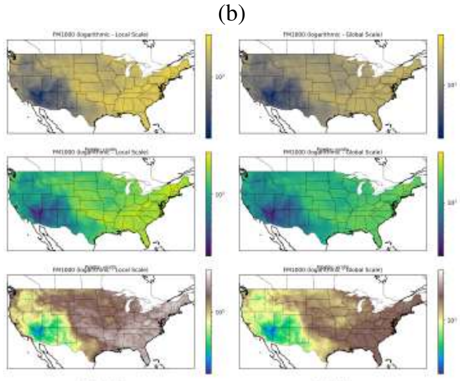
Temporal Trend of Vapor Pressure Deficit: As the animation progresses, we observe a clear pattern of increasing VPD across the U.S., especially prominent in the Southwest region. This indicates an intensification of the heatwave and drying effect, leading to a rising VPD over time. Higher VPD values are associated with lower humidity and higher temperatures, which align with the expected impacts of a prolonged heatwave. The Southwest consistently shows the highest VPD values throughout the animation, highlighting its susceptibility to dry con-



(a)



(b)



(c)

Figure 8: Plots for 1000-hour Dead Fuel Moisture (FM1000) using different color scales, palettes, and color plot types. (a): Continuous, (b): Discrete, (c): Logarithmic. Left: Local Scaling, Right: Global Scaling

Figure 9: 1000-hour Dead Fuel Moisture on 16th July 2023

ditions during the heatwave. The values diminish as we move towards the Northern and Eastern regions, where VPD is generally lower. This suggests that geographical factors, like regional climate patterns and elevation, might be influencing the moisture retention capacity of the air.

We used a contour fill algorithm for plotting VPD data as it is well-suited for creating smooth, continuous transitions across a spatial dataset, which is ideal for depicting VPD distributions. This smoothness is essential for interpreting gradual changes in atmospheric variables, like VPD, across large geographic regions.

2) Burning Index: Temperature Influence on Burning Index: We can observe that as the days progress the BI values increase over time, especially in regions like the Southwest U.S. This trend suggests that as temperatures rise, vegetation and ground fuels become drier, making them more susceptible to ignition. The higher temperatures likely enhance evaporation from soil and vegetation, resulting in drier fuels that contribute to elevated BI values, which indicate an increased potential for fire spread.

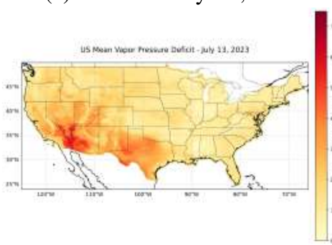
Progressive Increase in Burning Index Reflecting Fuel Dryness: The plots in Figure 11 show a gradual intensification in BI values, particularly in the Southwestern US. This progression suggests that as fuels continue to dry over consecutive days without precipitation, their flammability increases. This pattern highlights the cumulative impact of prolonged dry spells, where each day without moisture contributes to an incremental increase in fire spread potential.

We used the contour fill algorithm because Contour fill algorithms are better suited for smoothly varying continuous fields (e.g., indices(burning in-

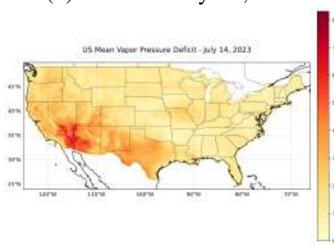
(a) VPD on July 10, 2023

(b) VPD on July 11, 2023

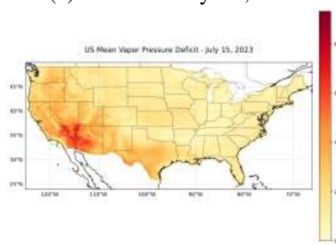
(c) VPD on July 12, 2023



(d) VPD on July 13, 2023



(e) VPD on July 14, 2023



(f) VPD on July 15, 2023

(g) VPD on July 16, 2023

(h) VPD on July 17, 2023

(i) VPD on July 18, 2023

(j) VPD on July 19, 2023

(k) VPD on July 20, 2023

(l) VPD on July 21, 2023

(m) VPD on July 22, 2023

(n) VPD on July 23, 2023

(o) VPD on July 24, 2023

(p) VPD on July 25, 2023

Figure 10: Mean Vapor Pressure Deficit from July 10th to July 25th 2023 measured in kPa

dex is an index)) and interpreting spatial patterns with a focus on smooth transitions and overall trends.

3) Energy Release Component: The Energy release component was also found to be high during the heatwave. The visualizations use the marching squares algorithm and an Inferno color palette(Figure 12). We chose the marching squares algorithm for ERC to experiment with the marching squares algorithm and due to the marching squares algorithm being preferred for a more boundary-focused analysis, such as isolating areas above or below a certain threshold.

The ERC is a measure of the potential energy available to a fire, based on the dryness of fuels and environmental conditions, with higher ERC values indicating more intense fire behavior.

Inference: Increasing Fuel Dryness Over Time: In the plots in Figure 12, ERC values increase in regions where the weather remains dry and hot over time, particularly in the Southwestern and Western U.S. This trend implies a drying of fuels, with higher ERC values representing an accumulation of potential fire energy in these areas. The prolonged absence of precipitation, combined with high temperatures, contributes to lower fuel moisture, thus increasing the energy available for potential fires.

Regional Patterns in ERC Values: The highest ERC values are concentrated in the Southwest and West, which are typically arid and semi-arid regions. These areas often experience low humidity and frequent dry conditions, contributing to consistently higher ERC values compared to regions with more stable, humid climates. The spatial pattern suggests that regional climate and ecosystem characteristics play a key role in ERC values, with arid landscapes naturally exhibiting higher fire potential.

D. Quiver Plot

1) Data Selection: We selected a sample of 9 days from July 14th to July 22nd from the gridMET[1] wind dataset to analyze heat wave patterns in the southwestern United States, particularly in California, Utah, Nevada, and Arizona. This period includes days of elevated temperatures, during which various atmospheric conditions intensify heat waves[9] . We used the columns ‘vs‘ and ‘th‘, representing wind speed and direction, to generate quiver plots.

2) Implementation: The following steps outline the data processing and quiver plot creation methods:

- **Data Preprocessing**: Using the `xarray` [11] library, we imported the gridMET[1] data, specifically ‘vs‘ (wind speed) and ‘th‘ (wind direction), and handled any missing values. The datasets were aligned by latitude, longitude, and time to ensure compatibility for plotting.
- **Quiver Plot Variations**:
  - **Proportional Vector Lengths**: In this type, vectors vary in length based on wind speed, and their orientation represents wind direction. This approach provides a direct visual understanding for both speed and direction.
  - **Fixed Vector Lengths with Colormap**: Here, vectors maintain a consistent length, with wind speed depicted by a color gradient, enhancing visibility of speed variations while preserving direction information.
- **Plot Generation**: For each day in the selected period, we generated quiver plots overlaid on the southwestern U.S. map. These visualizations highlight variations in wind patterns, allowing for an analysis of wind’s impact on heat wave conditions. Additionally, we created animated GIFs of daily quiver plots to visualize temporal changes.

3) Study of Wind Speed and Wind Direction: As shown in [3] and [2], wind speed have a positive correlation with the Heat wave intensity and winds have a study direction.

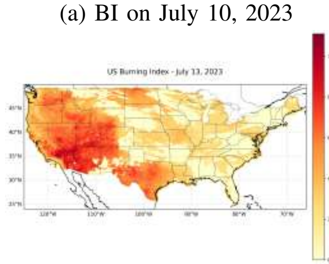
4) Experiments:

- **Region Selection**: We initially plotted the quiver plot for the whole America however, the transitions were not clear. Since the heat waves impact was concentrated in the southwestern states, we narrowed our focus to this region for more accurate visualizations.

(a) BI on July 10, 2023

(b) BI on July 11, 2023

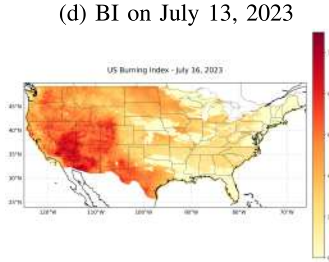
(c) BI on July 12, 2023



(d) BI on July 13, 2023

(e) BI on July 14, 2023

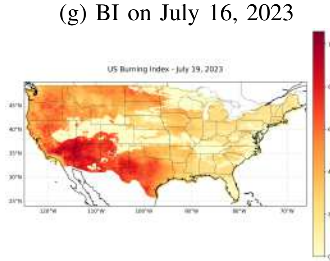
(f) BI on July 15, 2023



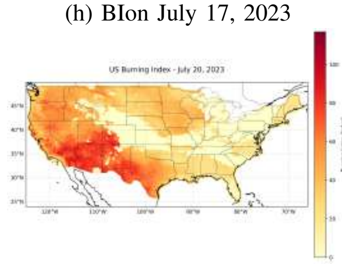
(g) BI on July 16, 2023

(h) BIon July 17, 2023

(i) BI on July 18, 2023

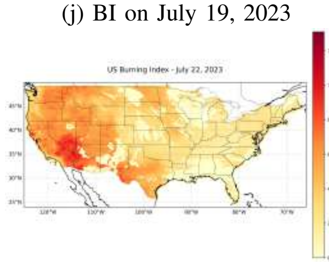


(j) BI on July 19, 2023



(k) BI on July 20, 2023

(l) BI on July 21, 2023



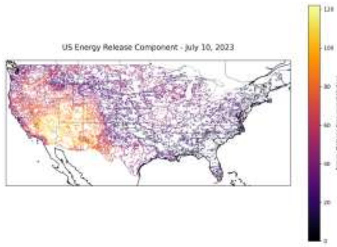
(m) BI on July 22, 2023

(n) BI on July 23, 2023

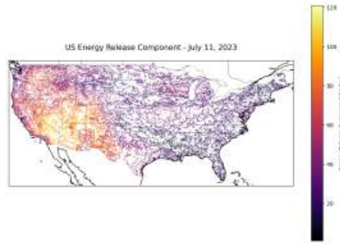
(o) BI on July 24, 2023

(p) BI on July 25, 2023

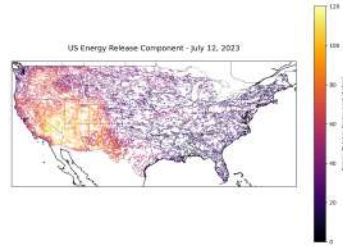
Figure 11: Burning Index from July 10th to July 25th 2023



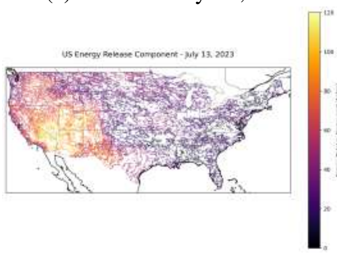
(a) ERC on July 10, 2023



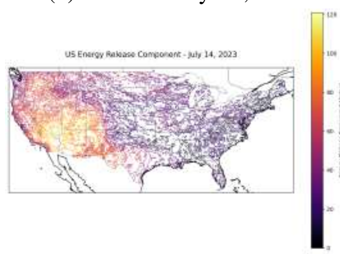
(b) ERC on July 11, 2023



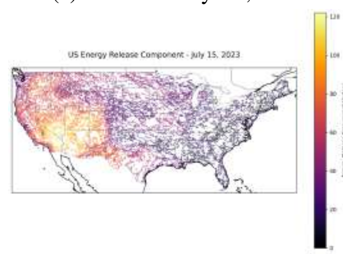
(c) ERC on July 12, 2023



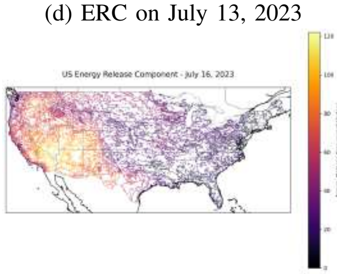
(d) ERC on July 13, 2023



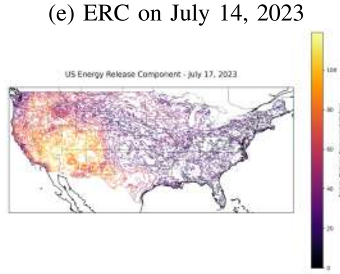
(e) ERC on July 14, 2023



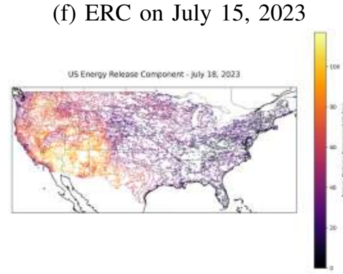
(f) ERC on July 15, 2023



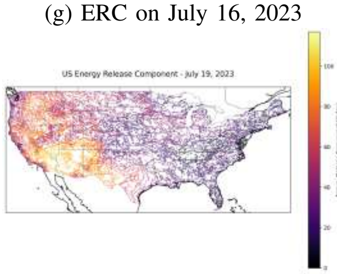
(g) ERC on July 16, 2023



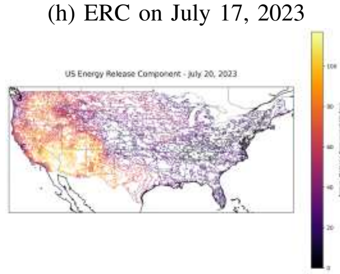
(h) ERC on July 17, 2023



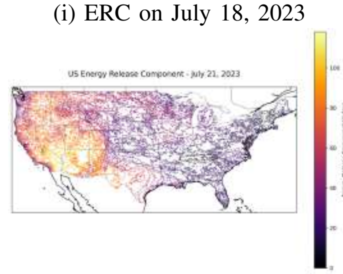
(i) ERC on July 18, 2023



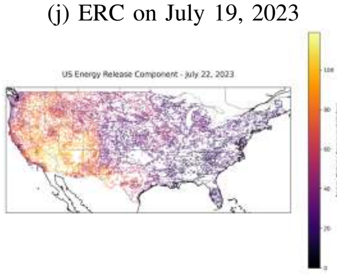
(j) ERC on July 19, 2023



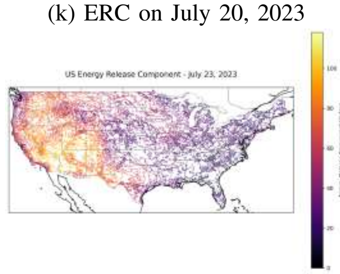
(k) ERC on July 20, 2023



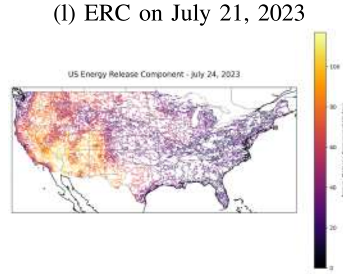
(l) ERC on July 21, 2023



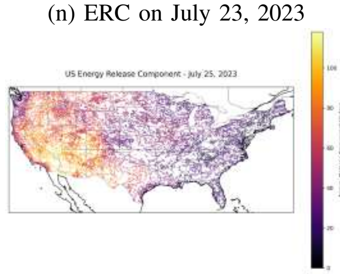
(m) ERC on July 22, 2023



(n) ERC on July 23, 2023



(o) ERC on July 24, 2023



(p) ERC on July 25, 2023

Figure 12: Energy Release Component(ERC) from July 10th to July 25th 2023, Contour plot using marching squares

- **Visualization Techniques**: We implemented two distinct approaches for visualizing wind speed and direction using quiver plots, as illustrated in the figures below
  - **Proportional Vector Lengths**: In this approach, vector lengths vary according to wind speed, providing a direct visual indication of wind magnitude. The vectors' orientation represents wind direction, allowing for an intuitive understanding of both speed and direction without relying on additional color coding. This method gives a natural representation of magnitude but may lead to overlap in regions of high wind density, as shown in 30. We adjusted parameters such as scale, head length, and vector width to enhance plot clarity.
  - **Fixed Vector Lengths with Colormap**: In this variation, all vectors are of equal length, with wind speed represented by color. The fixed-length vectors offer a clean and consistent look, while the color gradient provides a distinct indication of speed variations across the plot, as seen in 17. This approach helps avoid vector overlap in dense areas while maintaining direction information and highlighting speed differences effectively.
  - **Color Map Selection**: We tested several colormaps with varying color transitions, including Viridis, Blues, and Coolwarm, along with custom transitions designed to emphasize higher speeds. While Viridis and Blues offered smooth gradients, they lacked contrast for distinguishing higher wind speeds. After refining, we chose a custom Light-to-Dark colormap that shifts from light blue to red-orange around 5 m/s, ensuring clear visibility of speed changes. Additionally, we applied a black background, which enhanced the color contrast, making wind vectors and speed gradients stand out distinctly and providing a visually impactful display. This final configuration balanced clarity and contrast, yielding the best results for both contour and vector mapping.

5) Visualizations:

- **Visualizations using fixed length and variable length arrows**:
  In this section, we included both types of quiver plots to analyze the wind patterns related to heat wave intensity from July 14–22, 2023.

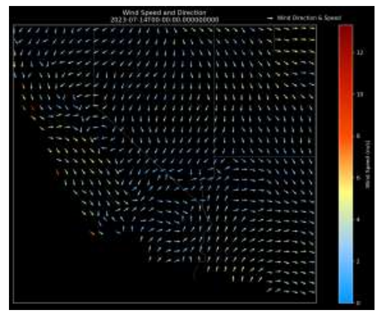


Figure 13: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-14.

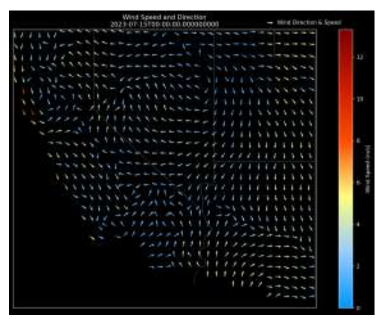


Figure 14: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-15.

6) Observations:

- **July 14, 2023**:Long arrows and orange-red arrows are visible along the California coast and some parts of Arizona, indicating high wind speeds associated with the onset of heat waves.

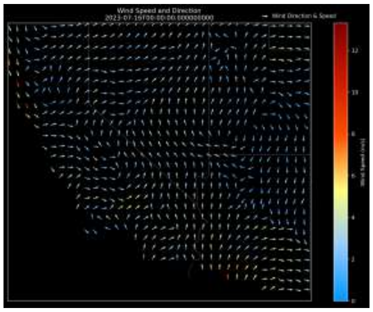


Figure 15: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-16.

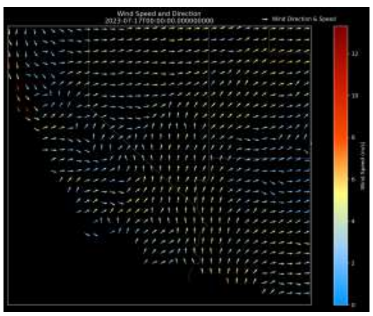


Figure 16: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-17.

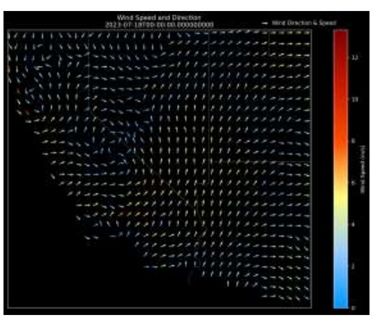


Figure 17: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-18.

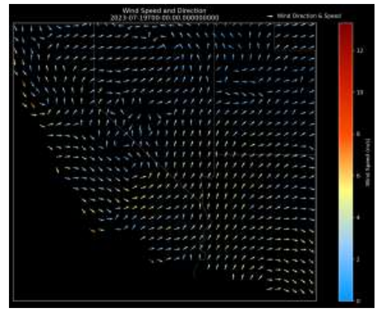


Figure 18: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-19.

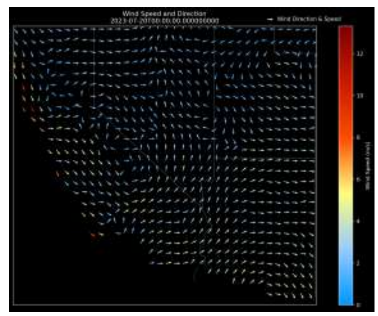


Figure 19: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-20.

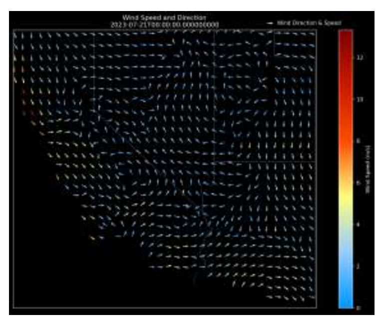


Figure 20: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-21.

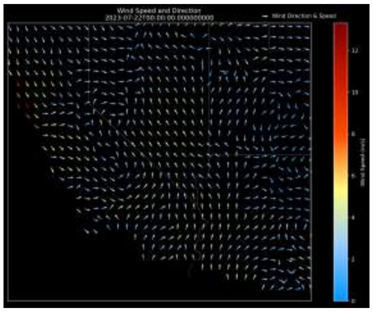


Figure 21: Plotting the quiver plot for wind direction and speed and using the custom color map to depict the magnitude of wind speed(in m/s) for 2023-07-22.

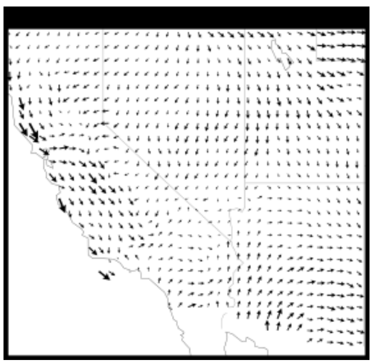


Figure 22: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-14

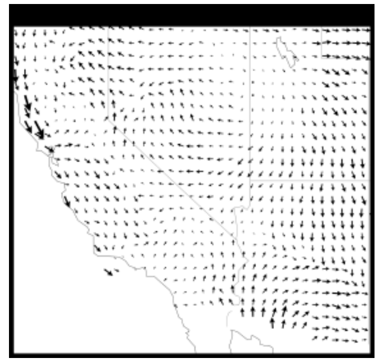


Figure 23: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-15

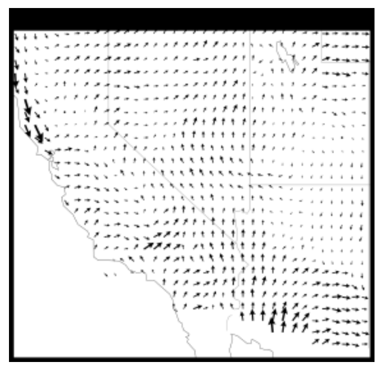


Figure 24: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-16

The inner regions of the map are primarily blue, representing lower wind speeds.

- **July 16, 2023**:Similar to July 14, long and orange-red arrows appear on the California coast and parts of Arizona. Compared to July 14, there is a noticeable increase in both the number and intensity of these high-speed arrows, indicating a rise in heat wave intensity.
- **July 16, 2023**:Similar to July 14, long and orange-red arrows appear on the California coast and parts of Arizona. Compared to July 14, there is a noticeable increase in both the number and intensity of these high-speed arrows, indicating a rise in heat wave intensity.
- **July 17–19, 2023**:An increasing number of long arrows penetrate deeper into the inner parts of the map, with more orange-red arrows indicating intensified wind speeds. The arrow directions for these high-speed quivers remain relatively steady, suggesting sustained wind patterns during the peak heat wave phase.
- **July 20–22, 2023**:The number of long and orange-red arrows begins to decrease, indicating a reduction in wind intensity. This decrease suggests that the heat wave's intensity is start-

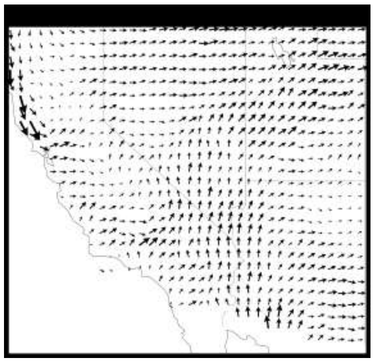


Figure 25: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-17

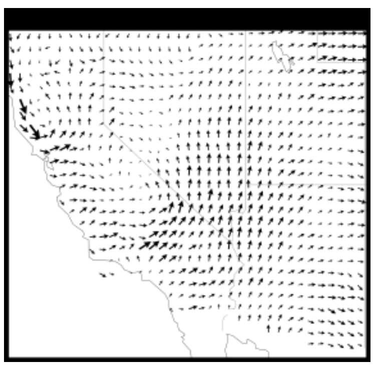


Figure 26: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-18

ing to diminish as the high-speed wind regions contract and show less persistence across the map.

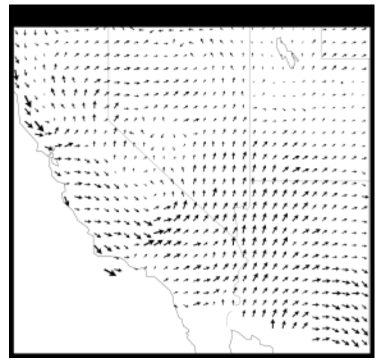


Figure 27: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-19

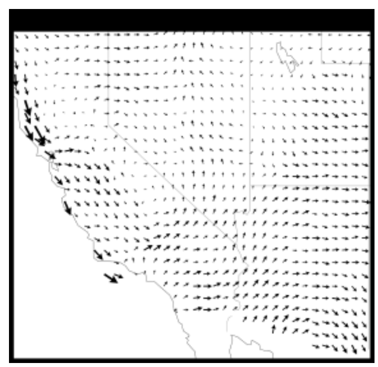


Figure 28: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-20

IV. InfoVis Dataset

A. Node Link Diagrams

1) The Dataset: **CPAN authors:** CPAN Explorer is a visualization project aiming at analyzing the relationships between the developers and the packages of the Perl language, known as the CPAN community. This snapshot was created by Linkfluence in July 2009. This file contains the network of developers, linked when they use the same Perl module.

The dataset has been visualized in Gephi by

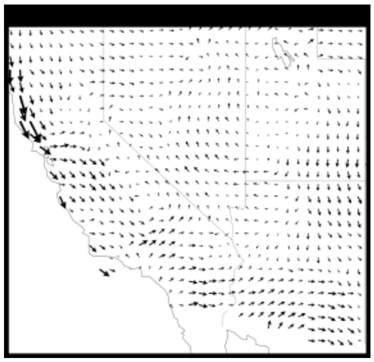


Figure 29: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-21

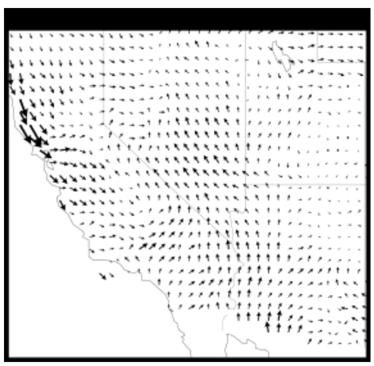


Figure 30: Plotting the quiver plot for wind direction and speed and modelling the length of the arrow to be proportional to the magnitude of the wind speed(in m/s) for 2023-07-22

Franck Cuny, a Perl hacker working at the French social media agency Linkfluence. He developed the CPAN Explorer web site, an interactive visualization to analyze relationships between developers and packages of CPAN (Comprehensive Perl Archive Network). The authors page shows relationships between developers inside CPAN: each developer is represented by a node with a size proportional to the number of modules the developer has released on CPAN. An edge between two developers is created when one developer uses a module from the other developer. The more uses of a developer's modules by other developers, the bigger the label.

2) Overview of Data: The initial node-link diagram of the dataset is shown in 31. The dataset has

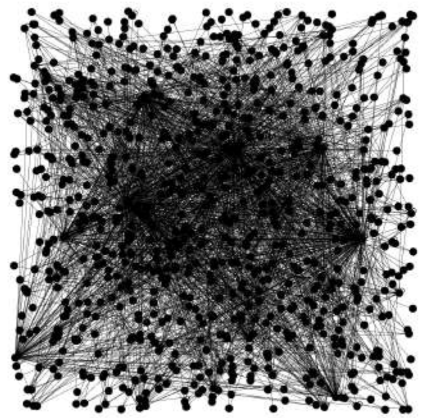


Figure 31: Initial Network

840 nodes and 2248 edges, and is a directed graph. Basic Statistics of the Graph

- Average Degree: Each node in the graph has an average degree of 2.676. The degree, out-degree, and in-degree distributions are shown in 32. The graph density is 0.003. The in-degree varies from 1 to 328, and the out-degree varies from 0 to 73.

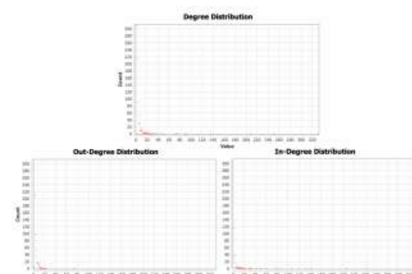


Figure 32: Degree distributions: Degree(top), out-degree(bottom left), and in-degree(bottom right)

- Graph Distance: The graph has a diameter 9 and an average path length 4.357882096069869.

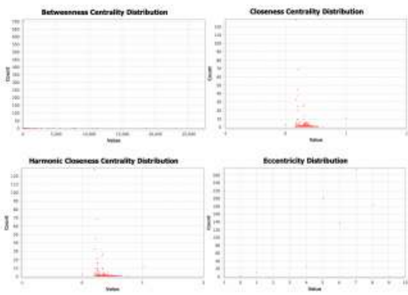


Figure 33: Miscellaneous distributions: Betweenness Centrality(top left), Closeness Centrality(top right), Harmonic Closeness Centrality(bottom left), and Eccentricity(bottom right)

- Connected Components: The graph has 2 weakly, and 786 strongly connected components.
- Average Clustering Coefficient: The graph has an average clustering coefficient of 0.233.
- Modularity and Modularity Classes:
  The Modularity of the graph(with the resolution set to 1.0) is 0.427, with the graph consisting of 11 communities. Modularity class size distribution is shown in 34.

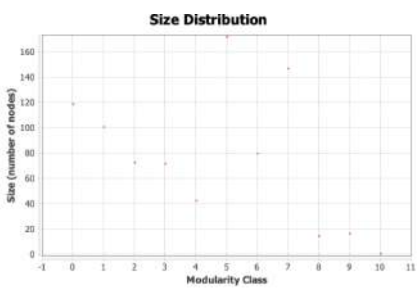


Figure 34: Modularity class size distribution

The network when each node is colored by modularity class(using the standard Gephi palette), produces 35.

We will stick to this partition(modularity class), as this is a good way to visualize communities.

3) Graph Layout Algorithms: Comparision: Fruchterman Reingold Layout The Fruchterman-Reingold layout is a force-directed algorithm that

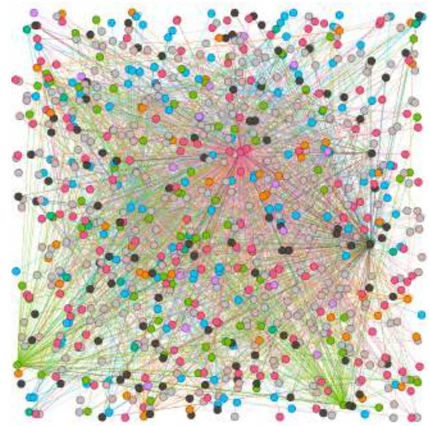


Figure 35: The Network, each node colored with respect to its modularity class

treats nodes as repelling objects (like charged particles) and edges as springs, trying to find an equilibrium where nodes are evenly distributed, and edges have roughly equal lengths. This layout works well for creating circular clusters and provides a balanced structure.

Running the Fruchterman Reingold graph layout algorithm on the network, colored with respect to modularity classes is shown in 36.

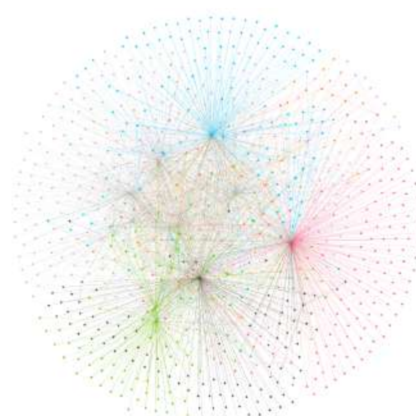


Figure 36: Fructerman Reingold on the network(colored with respect to modularity classes)

Now, we modified the visualization to a visualisation where each node's size and label proportional to in-degree (Figure 37) and out-degree (Figure 38). We colored all the edges #c0c0c0 to ensure a better visibility of nodes from here on.

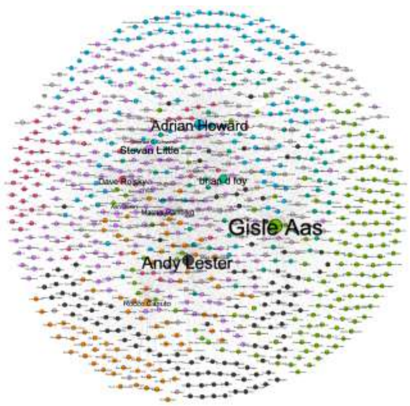


Figure 37: Fructerman Reingold on the network(colored with respect to modularity classes) with each node and label size proportional to in-degree

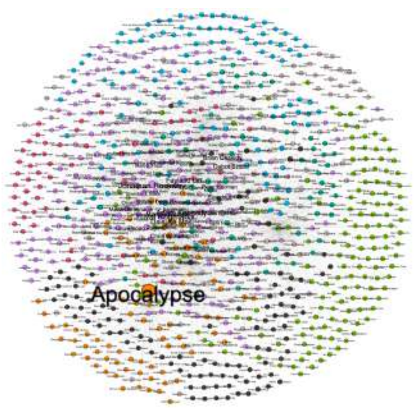


Figure 38: Fructerman Reingold on the network(colored with respect to modularity classes) with each node and label size proportional to out-degree

We've plotted the Betweenness Centrality(proportional to label size, Figure 39). We see that the authors Michael G Schwern, Steven Little, Yuval Kogman, Andy Lester, and Gisle Aasare some of the most influential ones.

Figure 39: Fructerman Reingold: nodes colored with modularity class, and the size of nodes and node labels is proportional to the betweenness centrality.

Yifan Hu Layout The Yifan Hu layout is another force-directed algorithm optimized for large graphs. It emphasizes both the structural hierarchy and modularity of the network. This layout generally spreads out the nodes more uniformly compared to Fruchterman-Reingold, creating a more open structure.

We've plotted the network in the Yifan Hu Layout and colored each node according to its modularity class(Figure 40). From Figure 40, we see that the Yifan Hu layout was better at depicting the clustering of different modularity classes.

From here on, we color all the edges #c0c0c0 for better visibility of nodes.

Plotting this network with each node's size and label proportional to in-degree shows that the network only has a few nodes of a large in-degree, while most of the others have smaller in-degrees(Figure 41).

Similar observations can be made when we plot the network with each node's size and label proportional to out-degree. Figure 42).

We plotted the network with node and label size proportional to between-ness centrality, and nodes colored according to modularity . The influential authors (Michael G Schclass (Fig 43).

Force Atlas Layout The Force Atlas layout is

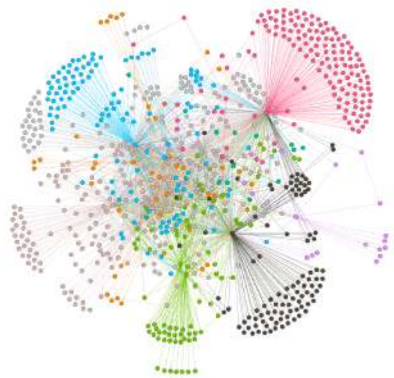


Figure 40: Yifan Hu Layout: nodes colored with respect to modularity class

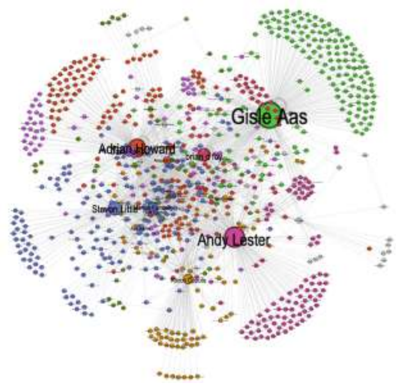


Figure 41: Yifan Hu Layout: nodes colored according to modularity class and each node's size and label are proportional to in-degree

specifically designed to show clusters and hierarchical structure. It pulls related nodes together tightly, forming dense clusters based on modularity and relatedness.

We've plotted the network in the Force Atlas layout and colored each node according to its modularity class(Figure 44).

In this layout, we see an outlier(in the top right corner). The outlier is Adam J. Kaplan(id: 2780).

From here on, we color all the edges #c0c0c0 for better visibility of nodes.

Plotting this network with each node's size and

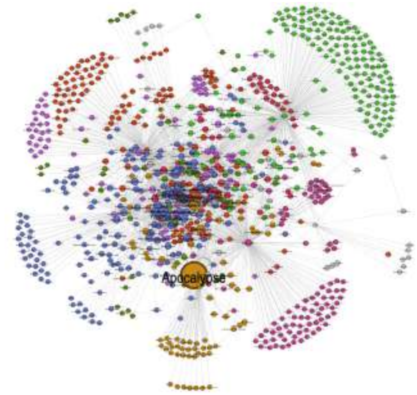


Figure 42: Yifan Hu Layout: nodes colored according to modularity class and each node's size and label are proportional to out-degree

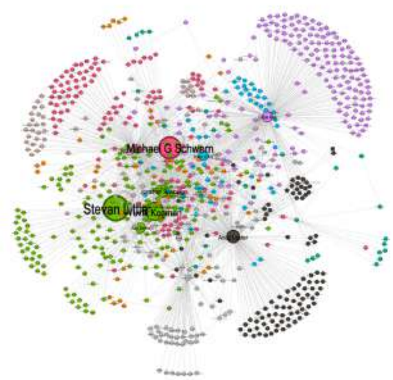


Figure 43: Yifan Hu Layout: node and label size proportional to between-ness centrality, and nodes colored according to modularity class

label proportional to in-degree using the Force Atlas layout also shows that the network only has a few nodes of a large in-degree, while most of the others have smaller in-degrees (Figure 45).

Similar observations can be made when we plot the network with each node's size and label proportional to out-degree using the Force Atlas layout (Figure 46).

We plotted the network with node and label size proportional to betweenness centrality, and nodes

Figure 44: Force Atlas Layout: nodes colored with respect to modularity class

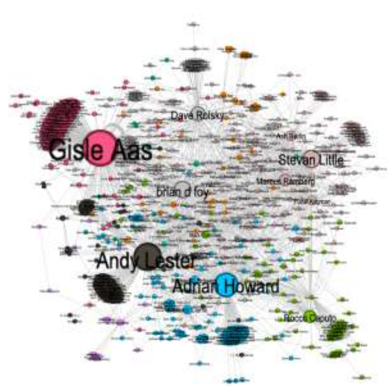


Figure 45: Force Atlas Layout: nodes colored according to modularity class and each node's size and label are proportional to in-degree

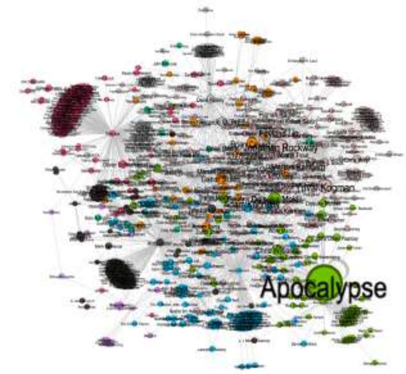


Figure 46: Force Atlas Layout: nodes colored according to modularity class and each node's size and label are proportional to out-degree

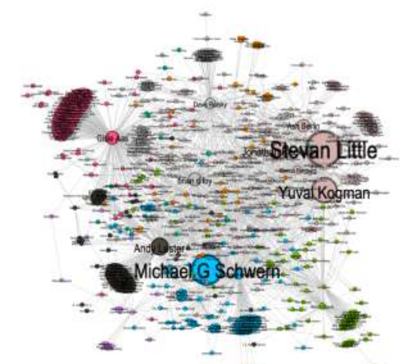


Figure 47: Force Atlas Layout: node and label size proportional to betweenness centrality, and nodes colored according to modularity class

colored according to modularity . The influential authors (Michael G Schclass (Fig 47).

4) Comparision:

1) Fruchterman Reingold Layout:
   We find that Fruchterman Reingold layout is effective for an overall view but can get overly dense in the center, making it difficult to see smaller structures within clusters.
   It is also hard to visualize communities using this layout, as there is no clear boundary between them.
2) Yifan Hu Layout: We find that it provides a better balance between separation and structure, making it suitable for visualizing both intra- and inter-cluster relationships with clearer boundaries.
3) Force Atlas Layout: We find that it is optimal for modularity and cluster visibility, though at the expense of individual detail clarity within clusters.

In the three layouts discussed, the Yifan Hu layout produced better visualizations due to the following reasons:

- Balanced Node Distribution and Readability:
  The Yifan Hu layout effectively spreads nodes out while maintaining a clear depiction of the network's modularity structure. Unlike more compact layouts (like Force Atlas) that may result in dense clusters, Yifan Hu's spacing provides enough room around each node, reducing visual clutter. This spacing makes it easier for viewers to distinguish individual nodes and edges, which is particularly helpful when analyzing complex or densely connected networks.
- Clear Modularity Clustering:
  The layout preserves modularity by forming distinct clusters for each community or class, enhancing the visual separation between groups. This distinct clustering helps in identifying boundaries between modularity classes, making the layout ideal for highlighting the overall structure of the network and the relationships between different communities.
- Efficient Edge Representation:
  The layout minimizes edge overlap without sacrificing structural information. This balance allows you to see not only intra-cluster connections (connections within clusters) but also inter-cluster connections (connections between clusters), both of which are important for understanding network structure. Reduced edge crossings enhance readability and prevent important connections from being obscured.
- Hierarchical Structure and Depth Perception:
  Yifan Hu's design subtly emphasizes hierarchical relationships within clusters, allowing nodes in the core or periphery of each cluster to be identified more easily. This hierarchical insight is often lost in layouts like Fruchterman-Reingold, where clusters are more uniformly circular, or Force Atlas, where nodes are packed tightly within clusters.
- Versatile and Scalable for Large Graphs:
  The Yifan Hu layout is specifically optimized for large networks, balancing computational efficiency with visual clarity. This scalability means that it can handle a substantial number of nodes and edges without significantly sacrificing readability, making it a good choice for complex datasets like the one you are working with.

5) Node Link Diagrams: Summary: After analyzing all the three layouts, we can conclude that the choice of layout depends on the focus:

- For broad structural understanding (e.g., interconnections and balance across the entire graph), the Fruchterman-Reingold or Yifan Hu layouts are beneficial.
- For detailed modularity analysis (e.g., distinguishing between closely connected clusters), the Force Atlas layout provides the clearest view of the community structure.

Each layout has trade-offs between density, clarity, and modularity focus, so the choice depends on the specific insights you aim to extract from the data.

For this dataset, overall, the Yifan Hu layout provides better visualizations, achieving a balance between clarity, spacing, and modularity structure.

B. Parallel Coordinate Plot

1.) **Dataset:** The dataset used is "Top Twitch Streamers" [4] . It contains data on Twitch's top 1000 streamers, ranked by follower count. The columns used as axes in the parallel coordinates plot are: Watch time, Stream Time, Peak viewers, Average viewers, Followers, Followers gained, Views gained, and Language.

2.) **Implementation:** The codebase is primarily developed in JavaScript, using Plotly.js [6] and PapaParse [5] to create a Parallel Coordinates Plot (PCP) for visualizing Twitch metrics. Displayed on an HTML page, the visualization can be viewed in a live server setup such as in VS Code. PapaParse loads and parses the CSV data, dynamically converting types for numerical fields and filtering out rows with null or empty values for a clean dataset. Once processed, Plotly.js is used to create the PCP by defining plot dimensions for each data column, mapping numerical values directly and converting categorical data (e.g., Language) into numeric labels via a 'languageMap'. A Viridis colormap is applied based on follower count, creating a color gradient that allows intuitive comparison across channels. The layout includes a title, customized margins, and a white background for readability, with a color bar legend for follower counts. The visualization is also made responsive and interactive.

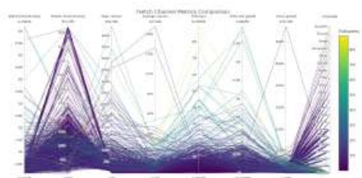


Figure 48: Parallel Coordinates plot for Twitch Streamers

3.) **User Interactions:**

- **Brushing:** Users can filter and highlight specific subsets of data points by left-clicking and dragging along any axis. By repeating this action on multiple axes, only traces that satisfy all selected conditions remain highlighted, showing an intersection of filtered data points. This enables users to explore complex patterns within the data by isolating points that meet multiple criteria, as demonstrated in Figure 49.

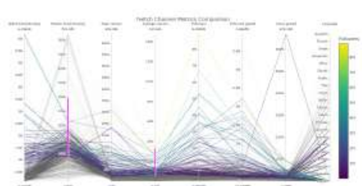


Figure 49: Example of brushing over multiple axes in the parallel coordinate plot

- **Axes-Reordering:** To rearrange an axis, the user only needs to left-click on its name and drag it to the left or right. Axes-Reordering allows for easier comparison between variables that are not adjacent, as comparing axes which are not adjacent can be challenging.

4.) **Inferences:** Possible inferences from Figure 48.
Follower Distribution: The dense cluster of violet lines indicates that the majority of the top 1000 Twitch streamers have follower counts on the lower end of the scale. This suggests that a substantial portion of these streamers have relatively fewer followers compared to the most popular channels, highlighting a distribution skewed towards smaller followings.
Watch Time vs. Stream Time: The plot shows that a few streamers achieve disproportionately high engagement through extended streaming hours, suggesting a gradual increase in watch time as stream length grows. However, this pattern is limited since we cannot get better insights of axes that are not adjacent to each other and we are unable to isolate specific categories so deeper insights would require interactive visual tools.
For instance, to understand the number of English-language streamers, we could apply a brushing filter to the "Language" axis for English shown in Figure 50.

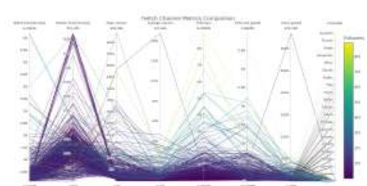


Figure 50: Parallel co-ordinate plot with brushing on English Language

If we further filter by followers (e.g., streamers with followers count between 1 million and 2 million followers) which is shown in Figure 51, we gain insights that aren't easily visible in a static parallel coordinate plot.
Understanding the relationship between "Views Gained" and "Stream Time" in this static view is challenging. By reordering the axes (as in Figure 52), we see little

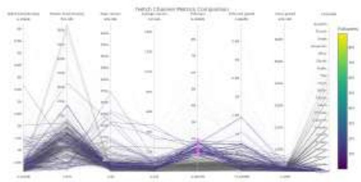


Figure 51: Parallel co-ordinate plot with brushing on English Language and Followers

to no significant correlation between these variables, indicating that simply increasing broadcast hours doesn't necessarily lead to higher viewership. This suggests that stream duration alone is insufficient for audience growth; other factors—such as content quality, audience interaction, and promotion—likely play a more critical role in attracting viewers.

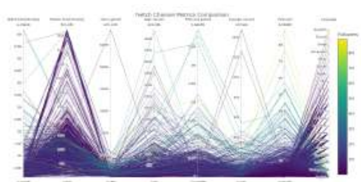


Figure 52: Parallel co-ordinate plot with axes-reordering on Views Gained

Another observation between stream time and followers gained.We can easily observe from Fig that even if stream time is more followers gained is less the trend for followers gained versus stream time would be downward. This suggests that streaming for longer durations does not necessarily correlate with an increase in followers.

In this manner Axes-reordering allows us to compare and understand the correlation between non-adjacent axes by making them adjacent through axis reordering. In a static parallel coordinates plot (PCP), comparing non-adjacent axes is challenging, but reordering helps overcome this limitation.

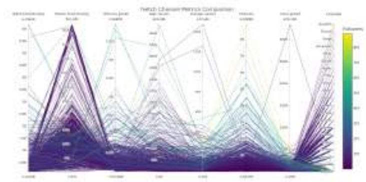


Figure 53: Parallel co-ordinate plot with axes-reordering on followers gained

C. Treemap

Treemaps are a powerful visualization tool for representing hierarchical data, offering a compact and intuitive way to display data relationships across multiple levels. By using nested rectangles, treemaps allow viewers to understand both the proportions and hierarchy within a dataset at a glance. This layout is particularly effective for large datasets with multiple categories and subcategories, as it maximizes the use of space and makes it easy to compare relative sizes. In our case, we're leveraging a treemap to visualize Twitch data by displaying several viewership and follower statistics in relation to some of the streamers' other categorical attributes. This approach provides insights into the audience distribution across different content types, enabling us to quickly identify popular languages and assess the impact of mature content or partnership status on viewership.

1. Dataset: We analyzed data and created treemap visualizations using the Top Streamers on Twitch dataset. This dataset includes information on Twitch's top 1000 streamers (based on follower count) as of August 2020. [4] It contains 11 columns:

1) **Channel Name**: Display name of the channel.
2) **Watch Time (Minutes)**: Total watch time on the channel for the year.
3) **Stream Time (Minutes)**: Total time streamed on the channel for the year.
4) **Peak Viewers**: Maximum number of viewers reached by a stream on the channel for the year.
5) **Average Viewers**: Average number of viewers across streams on the channel for the year.

6) **Followers**: Total number of followers for each channel.
7) **Followers Gained**: Number of followers gained by each channel in the year.
8) **Views Gained**: Number of views gained by each channel in the year.
9) **Partnered**: Indicates if the channel is a Twitch Partner.
10) **Mature**: Indicates if the channel is marked as mature content.
11) **Language**: Primary language in which the streamer's content is broadcast.

2. Data Preprocessing: The dataset was already very clean from Kaggle, requiring little to no preprocessing. Most columns were complete, with minimal missing data, and the values were well-structured for analysis. Since the data was so well-prepared, we were able to proceed directly with visualization and analysis without extensive cleaning steps. This allowed us to focus on meaningful insights from the data rather than data preparation.

3. Implementation: The implementation of the treemaps involves a Python script that reads in the dataset using pandas, a library that provides powerful data handling and manipulation capabilities. The script then uses Plotly, a visualization library, to generate an interactive treemap based on specific hierarchy levels from the dataset.

Once generated, the figure is saved in two formats: HTML and JSON. The HTML file allows for interactive viewing in any web browser, preserving the treemap's interactivity, while the JSON file is useful for programmatic access or further customization. This setup enables independent exploration of the treemap, allowing users to dive into specific data categories within the saved files.

3. User Interaction: The treemap visualization offers several interactive features that make it easy to explore and understand the hierarchical data:

- **Hover Information:** When hovering over different sections of the treemap, users can see detailed information, including the exact *Average viewers* for each category (*Language*, *Mature* content status, and *Partnered* status). This hover display provides a quick reference to values without needing to click. (See Figure 54)
- **Zoom-In and Drill-Down:** Clicking on a particular segment allows users to zoom into that specific category, displaying subcategories within the chosen hierarchy. This is useful for examining smaller segments or subcategories in more detail, such as drilling down by *Language* and then by *Mature* status. (See Figure 55)
- **Zoom-Out and Navigation:** Users can navigate back up the hierarchy by clicking on higher-level categories or labels, making it easy to explore different parts of the dataset without losing the overall context.
- **Dynamic Resizing:** As users zoom in and out of different categories, the sections of the treemap dynamically resize to fill the available space, optimizing the view for better readability.
- **Tooltips and Customization:** Users can customize the appearance of tooltips and the overall layout using Plotly's JSON format, giving additional flexibility in data exploration.

These features together provide an interactive and engaging way to visualize, navigate, and interpret hierarchical data, making it ideal for users who want to explore the dataset's structure and relationships in depth.

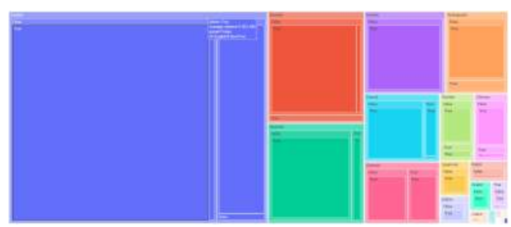


Figure 54: User Interaction: Hover Information, Tooltips and Customization

5. Experiments: Several experiments were conducted to determine the most effective approach for visualization of the Twitch Dataset.

- **Feature Selection:** Some features were examined to determine their impact on Twitch viewer patterns and to enhance the treemap's

Figure 55: User Interaction: Zoom-In and Drill-Down, Dynamic Resizing, Zoom-Out and Navigation

visual clarity. The features chosen included *language*, *maturity setting*, and *partnership status* as these categories were relevant for the hierarchical categorization of streamers.

- **Treemap Layout Algorithms:** Different algorithms were tested to optimize the data's layout within the treemap structure:
  - **Squarified:** Produced a balanced layout with nearly square blocks, facilitating easy visual comparison and readability, especially for datasets with a wide range of viewer counts (See Figure 56).
  - **Slice-and-Dice (Horizontal):** Displayed data in horizontal strips but posed challenges when comparing categories with thinner areas due to values being lower (See Figure 57).
  - **Slice-and-Dice (Vertical):** Used vertical strips to arrange data but had readability limitations similar to the horizontal method (See Figure 58).
- **Color Scheme:** We tested several color schemes, including Magma, Plasma, Viridis, and Cividis, (See Figures 59, 60, 61, 62) to determine the most effective for visualizing the Twitch dataset in a treemap. While Magma and Plasma offered high contrast, their intense saturation sometimes made it difficult to differentiate between closely packed data points. Cividis, designed for accessibility, lacked the visual sharpness needed to clearly distinguish between values.
  Viridis emerged as the best choice. Its gradient, transitioning from purple and blue to cyan and yellow, provided a stark contrast that helped us quickly perceive differences in data. The transition from darker to lighter shades in Viridis aligned well with the hierarchical structure of the treemap, making the data more intuitive and easier to understand at a glance.

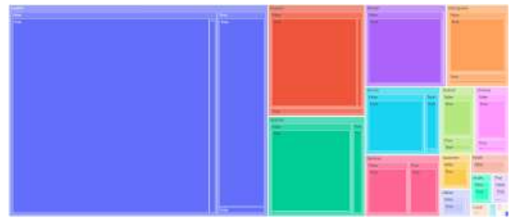


Figure 56: Squarified Algorithm applied to generate Treemap of Average Viewers by Language, Maturity, and Partnered Status

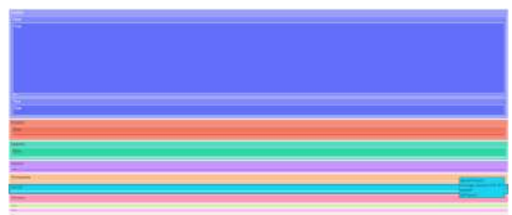


Figure 57: Slice-and-Dice (Horizontal) Algorithm applied to generate Treemap of Average Viewers by Language, Maturity, and Partnered Status

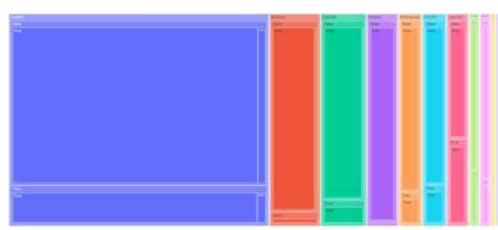


Figure 58: Slice-and-Dice (Vertical) Algorithm applied to generate Treemap of Average Viewers by Language, Maturity, and Partnered Status

Through a comparative analysis, the Squarified algorithm was determined to be the most effective for visualizing the data. Its balanced shape and

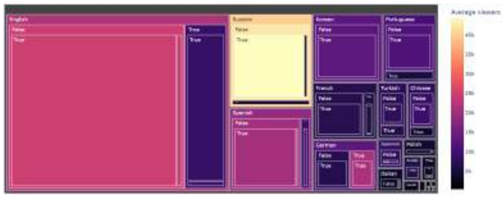


Figure 59: Squarified Algorithm applied to generate Treemap of Average Viewers by Language, Maturity, and Partnered Status using 'magma' colormap

Figure 60: Squarified Algorithm applied to generate Treemap of Average Viewers by Language, Maturity, and Partnered Status using 'plasma' colormap

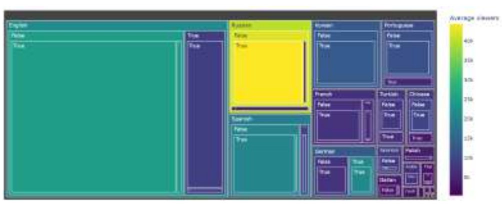


Figure 61: Squarified Algorithm applied to generate Treemap of Average Viewers by Language, Maturity, and Partnered Status using 'viridis' colormap

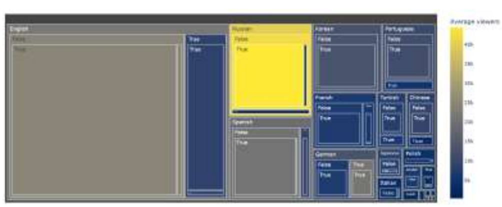


Figure 62: Squarified Algorithm applied to generate Treemap of Average Viewers by Language, Maturity, and Partnered Status using 'cividis' colormap

layout facilitated intuitive exploration of viewer patterns among Twitch streamers, providing a clear and immediate understanding of relative values. The combination of the Squarified algorithm with the Viridis color scale was selected as the optimal configuration, offering an accessible and coherent visualization of Twitch viewer data. This approach enhanced the clarity and interpretability of the treemap, enabling users to easily grasp trends and insights from the dataset.

6. Observations: Several observations were made from these visualizations, which facilitated the formulation of inferences and potential explanations for the underlying patterns in the data.

- **Visualization 1 - Average Viewers by Language, Maturity, and Partnered Status**
  The visualizations reveal that the highest overall average viewership is found among English-speaking, Non-Mature, Partnered streamers. (See Figure 61) Notably, Russian streamers, particularly in the top tier, exhibit significantly higher average viewership, with the average viewership of Russian streamers around 40,000, whereas the second highest, English streamers, average around 20,000. Furthermore, Mature German streamers appear to vastly outperform their Non-Mature German counterparts in terms of average viewership.
  Several factors may explain these trends on Twitch. The dominance of English-speaking streamers can be attributed to the global reach of the English language and its widespread adoption across various regions. The higher viewership among Russian streamers might reflect a concentrated fanbase, possibly due to cultural factors, regional preferences, or popular content creators who cater specifically to the Russian-speaking audience. The performance gap between Mature and Non-Mature German streamers could be influenced by the broader appeal of mature content, which tends to attract more engaged viewers, or it may reflect a higher level of professionalism and established fanbases among Mature streamers. Additionally, the effect of Partnership status can enhance visibility and credibility, leading

to higher average viewership for partnered streamers.

- **Visualization 2 - Watch Time by Maturity, and Language**
  The aggregate watch time in English dominates both the Mature and Non-Mature categories. (See Figure 63) In the Non-Mature category, English streamers lead in watch time, followed by Korean, Russian, Spanish, and Portuguese streamers. In the Mature category, English remains the highest in total watch time, but Portuguese streamers surpass their English counterparts in average watch time, with Portuguese streamers averaging around 3.6 billion minutes, while English streamers average approximately 800 million minutes.
  This disparity can be attributed to several factors on Twitch. For one, English-speaking content generally has a larger global audience, leading to higher overall aggregate viewership, particularly in the Non-Mature category where content is more widely accessible. However, the higher average watch time for Portuguese streamers in the Mature category could suggest a more dedicated and niche audience for mature content in the Portuguese-speaking community. Portuguese-speaking viewers may be more engaged with mature content, resulting in longer viewing sessions compared to English-speaking audiences. Additionally, the relatively smaller but highly engaged audience for Portuguese streamers may lead to more consistent and focused viewership, especially in mature content where niche interests tend to dominate. These differences highlight the varying dynamics in how audience engagement is shaped by language, content type, and cultural preferences on the platform.
- **Visualization 3 - Followers Gained by Language, Maturity, and Partnered Status**
  The aggregate followers gained in August 2020 is highest in English, with a total of 98 million, followed by Spanish with 38 million. (See Figure 64)However, while Spanish ranks second in total followers, the average followers gained per streamer is significantly higher in Spanish, with top Spanish streamers gaining an average of 1.6 million followers, compared to only around 680,000 for English streamers.
  Several factors contribute to this phenomenon. First, English streamers benefit from the largest global audience, leading to high total follower counts. However, this also results in greater competition among English-language streamers, which can limit the average follower gain per streamer. In contrast, Spanish-speaking streamers often operate in markets where Twitch is experiencing faster growth, particularly in Latin America and Spain. These regions have seen a rise in the gaming and streaming culture, with Spanish streamers standing out more due to lower competition and a highly engaged local audience. Moreover, Twitch's algorithms may promote localized content more effectively, boosting visibility for Spanish-language streamers. Additionally, cultural factors play a role, as Spanish-speaking viewers are often more loyal to streamers from their own language group, contributing to higher follower numbers per streamer.

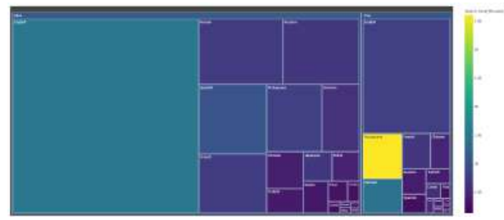


Figure 63: Squarified Algorithm applied to generate Treemap of Watch Time by Maturity, and Language using 'viridis' colormap

REFERENCES

1. Climatology Lab, gridMET: Gridded Meteorological Data, https://www.climatologylab.org/gridmet.html. Accessed: 2024-11-15.
2. Lau, N.-C., Nath, M.J.: A model study of heat waves over North America: Meteorological aspects and projections for the twenty-first century. Journal of Climate **25**(14), 4761–4784 (2012)

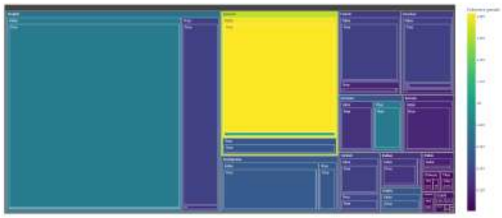


Figure 64: Squarified Algorithm applied to generate Treemap of Followers Gained by Language, Maturity, and Partnered Status using 'viridis' colormap

3. Li, D., Sun, T., Liu, M., Wang, L., Gao, Z.: Changes in wind speed under heat waves enhance urban heat islands in the Beijing metropolitan area. Journal of Applied Meteorology and Climatology **55**(11), 2369–2375 (2016)
4. Mishra, A.: Twitch Data, https://www.kaggle.com/datasets/aayushmishra1512/twitchdata (2024). Accessed: 2024-11-15.
5. PapaParse Contributors, PapaParse - The Fastest In-Browser CSV Parser for JavaScript, https://github.com/mholt/PapaParse (2024). Accessed: 2024-11-15.
6. Plotly Technologies Inc., Plotly.js - A JavaScript Graphing Library, https://github.com/plotly/plotly.js (2024). Accessed: 2024-11-15.
7. Schönbeck, L.C., Schuler, P., Lehmann, M.M., Mas, E., Mekarni, L., Pivovaroff, A.L., Turberg, P., Grossiord, C.: Increasing temperature and vapour pressure deficit lead to hydraulic damages in the absence of soil drought. Plant, Cell & Environment **45**(11), 3275–3289 (2022)
8. Swiss Federal Institute for Forest, S., WSL, L.R.: What is VPD?, (2023). https://vpdrought.wsl.ch/en/what-is-vpd/. Accessed: 2024-11-14.
9. Wikipedia contributors, 2023 Heat Waves, https://en.wikipedia.org/wiki/2023_heat_waves (2024). Accessed: 2024-11-15.
10. Will, R.E., Wilson, S.M., Zou, C.B., Hennessey, T.C.: Increased vapor pressure deficit due to higher temperature leads to greater transpiration and faster mortality during drought for tree seedlings common to the forest–grassland ecotone. New Phytologist **200**(2), 366–374 (2013)
11. Xarray Development Team, Xarray Documentation, https://docs.xarray.dev/en/stable/. Accessed: 2024-11-15.