6030 Process Book

Richard Garcia, Ashton Sobeck

Table of Contents:

Overview and Motivation

Provide an overview of the project goals and the motivation for it. Consider that this will be read by people who did not see your project proposal

We were motivated in relating our 6030 project and problem to state-of-the-art research, more specifically in map visualizations where we could present data on a layer over a map. We found an article on the IEEE Transactions on Visualization and Computer Graphics titled "UrbanMotion: Visual Analysis of Metropolitan-Scale Sparse Trajectories" that served as motivation for finding our dataset and replicating as similar as we could how the paper presented their data. We will present a more detailed description of the article in our Related Work section.

By visualizing a dataset collected by the city of Melbourne, Australia that shows the number of pedestrians identified by sensors in different areas of Melbourne. We would like to present a visualization website that can be easily utilized by any user and display changes in pedestrian movement on a map visual of Melbourne to understand pedestrian density which can potentially lead to optimization of infrastructure in Melbourne.

Related Work

We knew we wanted our project to focus on presenting a map visualization with data on top. While searching for a research paper that could give us inspiration, we happened upon a paper titled, "[UrbanMotion: Visual Analysis of Metropolitan-Scale Sparse Trajectories]()" in IEEE Transactions on Visualization and Computer Graphics.

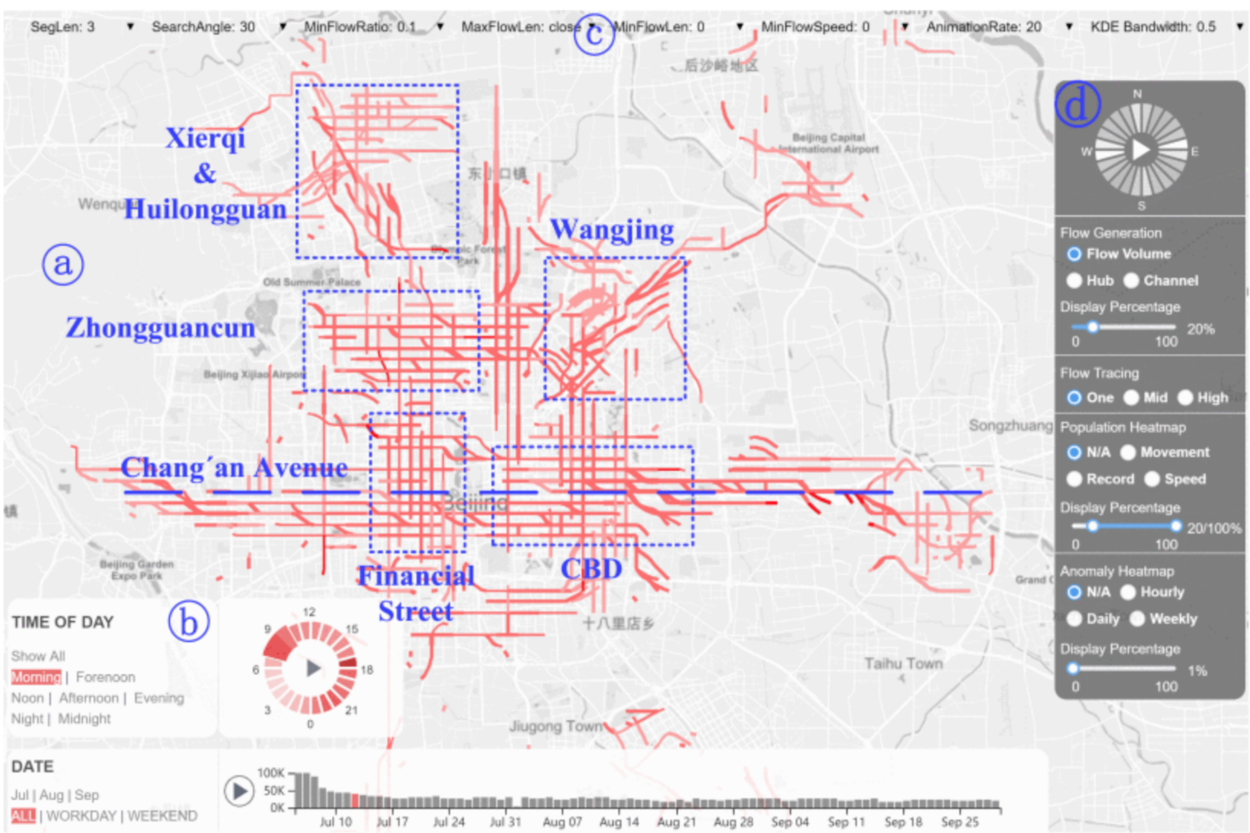

Figure 1. UrbanMotion Dashboard. Showcasing pedestrian routes and trajectories in Beijing from 7am - 10am on July 12th, 2020.

The main summary of the research article uses smart device data in three major cities in China (Beijing, Tianjin, and Tangshan) to create a dashboard called UrbanMotion (Figure 1) that visualizes multi-directional population flows of each city. This visualization was found helpful in “visually analyzing population movement in modern cities and having the potential to help in real-world applications such as “traffic optimization, urban planning, and business site configuration.”

We took inspiration from this article to formulate the problems we want to solve using a dataset we found of pedestrian movement in Melbourne, Australia.

Questions

The questions are answering by presenting our visualization of pedestrian movement is Melbourne, Australia are these:

1. **How can the knowledge of massive human movement in Melbourne contribute to optimizing infrastructures such as energy, traffic, and urban planning?**
2. **Is pedestrian movement influenced by a deviation in weather (e.g. rain, extreme heat) or a natural disaster (e.g. earthquake, flood)?**

We came up with these during our first milestone when we found the research article that gave us the inspiration for our project. The researchers in the article wanted their UrbanMotion dashboard to give insights in decision making when it came to optimizing infrastructure. The article also has a figure where they compare pedestrian trajectories and routes of a day with a rainstorm to a normal day. From this information we recontextualized their goals and observations into the questions above that can be applied to our Melbourne Pedestrian dataset.

Data

Source, scraping method, cleanup, etc.

We found our dataset from the Data is Plural Google Sheet. This dataset shows the number of pedestrians identified by sensors in different areas of Melbourne, Australia. The data is provided by the city of Melbourne, and is actively being updated. Currently, there are 3.86 million rows of data for this dataset. The data starts in 2009, and is updated every month. Each row is a total hourly sensor count of pedestrians Click here to access the dataset.

The city of Melbourne gives clear explanations of what column in the dataset represents. The dataset has 13 columns. In each row, there is a unique ID, the date and time of the reading in the format: `dd/mm/yyyy hh:mm:ss`, the year of the sensor reading, the month, the day of the year in number format, the day of the week, the time of day in 24 hour format, the sensor ID, the sensor name, the hourly counts for that sensor, and the latitude and longitude location of the sensor.

The web interface for the dataset gave us multiple options to download the data. We used the Socrata Open Data API (SODA) that provides programmatic access to this dataset including the ability to filter, query, and aggregate data. We also used the dataset's own filtering interface on the website that hosted it. The data would be exported as a csv.

There were multiple ways in which we had to clean up the data. First our dataset did not include the longitude and latitude coordinates corresponding to each pedestrian sensor. We were able to find a small dataset that the city Melbourne posted where it showed the same dataset as the one we are using, but with coordinate information. We joined the coordinate columns, which were the longitude column, latitude column, and a column called location where the coordinates were combined in this format: (x,y), to the dataset we were using by using the pandas library and joining on sensor id, which each dataset had. Second, we also removed the time at the end of the column titled date_time in order to just use the data for our data form which allows a user to filter by date on the visualization. Third, the date output from using a date html form is in this format: yyyy-mm-dd while the date_time column is in the format: dd/mm/yyyy. To fix this, we split the strings, and compared each of the individual values within the strings to their respective values (mm in date html compared to mm in the date_time column, etc). In the hourly_count columns, the counts sometimes had commas in the numbers if they were over 999. To fix this, we used a replace() call on the value that held the hourly_count to remove the comma and convert it from a string to a number.

Exploratory Data Analysis

What visualizations did you use to initially look at your data? What insights did you gain? How did these insights inform your design?

We could not replicate the UrbanMotion Dashboard from the paper because of one specific reason. Our dataset of pedestrian density in Melbourne is collected by using a sensor placed on a specific street that picks up foot traffic when pedestrians pass through the sensor area. The dataset used by the paper utilized each pedestrian's smart device to track their trajectory as they walk through certain areas. Because our dataset did not provide a continuous stream of data where we could just plot trajectories of pedestrians we have to adjust our visual to just show a density count (through color) on that specific street in a static position on a visual.

Because we started the project with the idea to present a map visualization that juxtaposed our data on another layer on top of the map we didn't utilize many visualizations to initially look at our data before we decided on a certain visual. After Dr. Iuricich read our project proposal, he recommended we utilize the deck.gl framework instead of d3. He stated the deck.gl was more intuitive for mapping data onto a map, which is something we wanted to do in our case. We started researching deck.gl and all the possible layers we could put on top of our Melbourne map. We happened upon the HexagonLayer visualization layer that deck.gl offered and immediately knew it would be the best way to visualize what it is we would like to display with our dataset. By using the HexagonLayer we could showcase the density of a specific street in Melbourne in a manner that did not make the visualization convoluted. Our thoughts were only confirmed once we actually created our visual using the HexagonLayer. There are no changes from our HexagonLayer visualization to other visualizations or vice versa. The HexagonLayer visualization was the first major visualization we used for our data and we didn't change it. Due to its built-in aggregation, expanding our filter to allow for longer time ranges was trivial due to how the HexagonLayer processes data. By default, the HexagonLayer will take whatever data points are returned and aggregate them by sum. Knowing this, parameterizing the starting and ending hours was the method in which we implemented our aggregation.

Design Evolution

What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles you learned in the course. Did you deviate from your proposal?

We knew that we needed to accurately correlate our final visualization with the paper that we found at the beginning of the semester, while recognizing that the data between the two projects were different. As a group, we came to the conclusion that we would need a map that showed the difference of densities on each street that we had a data point for in order to answer our questions. We started designing the website by creating sketches that we would reference as we started development. Keeping the user in mind was one of our primary goals as we did not want to have to explain many portions of the visualization through text.

Figure 2. Initial Sketch. Our initial sketch of our visualization. The visualization from the paper we found is used as a placeholder for the actual visualization.

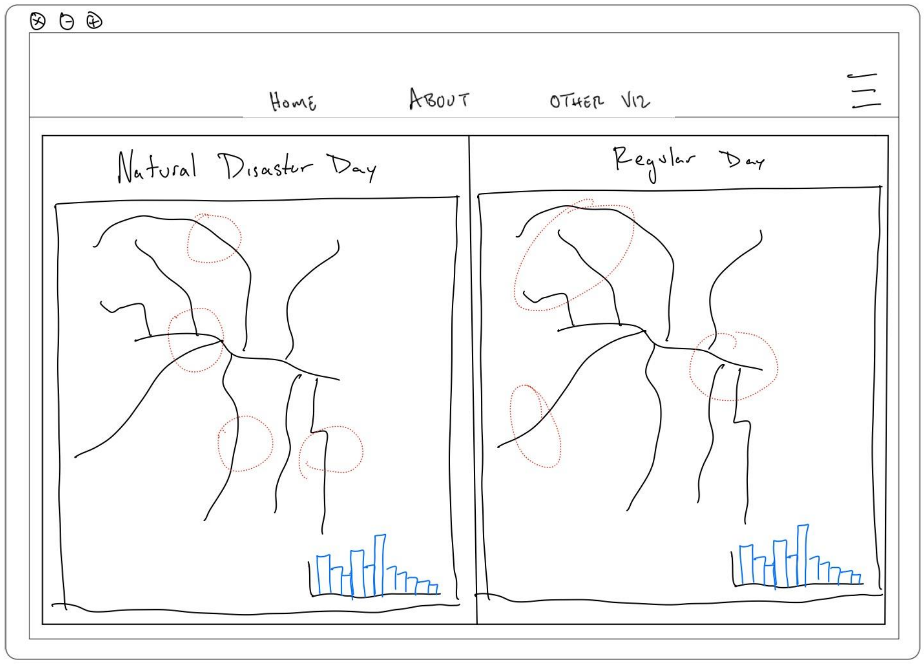

Figure 3. Initial Natural Disaster Sketch. Our initial sketch of our second visualization answering the question of how natural disasters affect pedestrian density.

After our initial sketches, we focused on development. Utilizing deck.gl as the main library that handled most of our data, we focused on learning how to first get the map on the screen with points plotted. Once determining that the hexagon layer was the best layer for us to use, we went through and cleaned up our data, and started plotting our data points. From there, we went about further drilling down into the data through interaction. We wanted our interaction to be simple to understand and perceive.

We chose a barchart through an on hover event that showed the entire day's worth of data. We chose this method of visualization due to the barchart's ability to display trends very easily. Throughout our development our barchart successfully showed the trends throughout the day of the density of a street in Melbourne.

We chose the deck.gl layer HexagonLayer as the visualization of our data because it used the marks of the hexagon bins to show the street and the channel of hue to show how many pedestrians were at the street during the time period selected.

On our first iteration of implementing interaction, our tooltip's barchart took up a great deal of space on the screen, used different and matching colors in the bars, and disappeared once the user had stopped hovering over the hexagonal pillar. Once Dr. Iuricich gave feedback on our interactions at that point, we went back to the drawing board on how we should approach the interaction within our visualization. With the feedback given of our prototype, we were determined to improve and create a final product that satisfied our questions in an elegant manner.

Dr. Iuricich also gave helpful feedback on the presentation of our visualization. He suggested that the hexagonal pillars should be shown from a top down perspective instead of an angled one in order to show all hexagons without one towering over another.

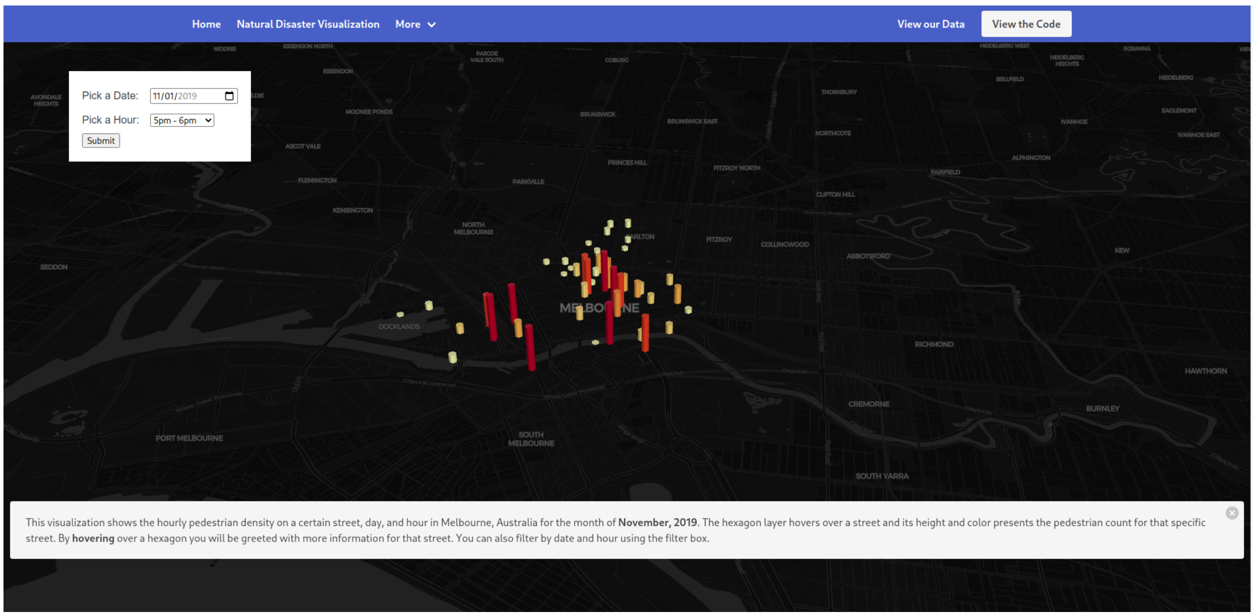

Figure 4. Initial visualization on our webpage.

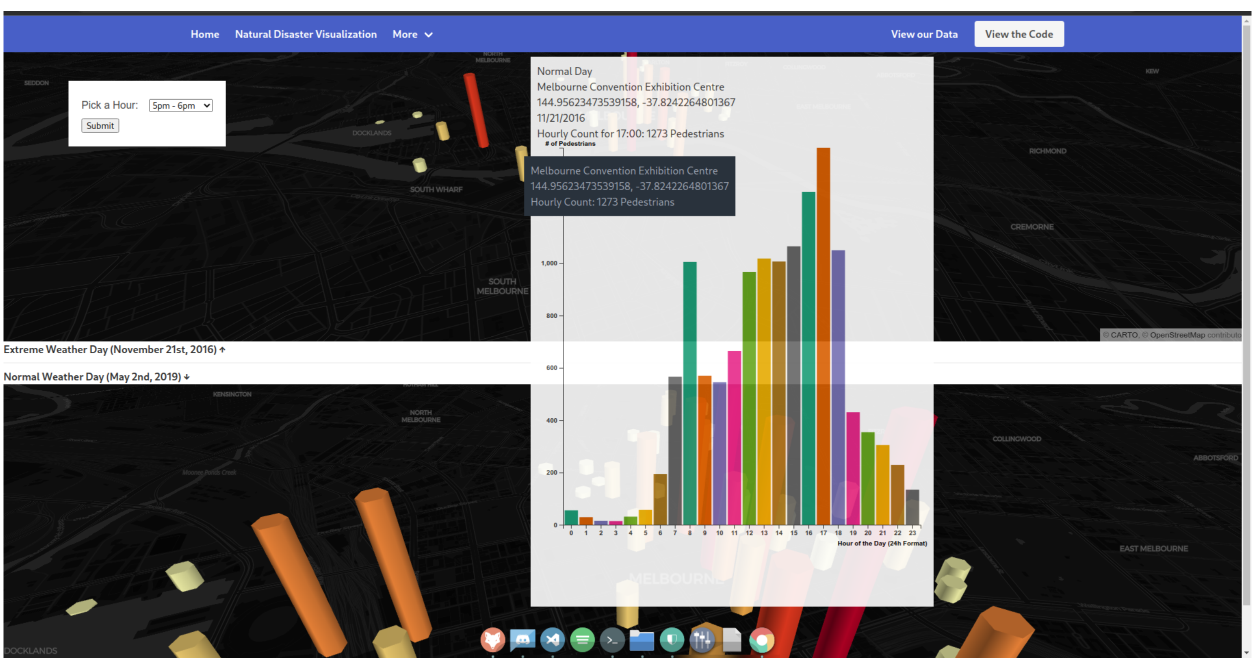

Figure 4. Initial natural disaster visualization on our webpage. Also showcases the tooltip for each hexagonal pillar.

From the feedback that we received, we determined that the tooltip that was shown by hovering over a hexagon should be more concise. With that in mind, we only show the user what street they are hovering over and the count of pedestrians during the time frame selected. Now, when a user clicks on a hexagon, a bar chart will appear in the upper left hand corner of the map. The color of the bars now uniformly match the color of the navigation bar instead of being random colors for each bar. When another hexagon is clicked, another bar chart will appear under the first one so that the user can compare the two street densities during the day that is selected. The hexagons that are selected get a colored circle plotted on them that will be the same color as the border on the bar chart plotted. The coordinating colors signify to the user which graph is for which so that they do not have to remember what they have selected previously. We chose only to allow two selections at a time in order to not flood the screen with barcharts and too much information.

We chose deck.gl's ScatterPlot layer to show the circles on the hexagons. We chose the circle mark to show the user that the hexagon had been clicked. The ScatterPlot layer allows us to choose the exact hue that the circle will be which we needed to be able to do in order to correctly show what hexagon has been selected.

We updated the Y-axis scale to dynamically change to the highest of the two hexagons selected. For example, if the highest amount of pedestrians detected for a certain street, x, was 300 and another street, y, had a high of 2000 pedestrians detected in an hour, if x was clicked first and then y, x's y-axis and bar height would dynamically change once y was clicked in order not to lie to the user and accurately allow for comparison between x and y.

For our natural disaster comparison visualization, we also changed the tooltip to be an on click event that puts the comparison between the normal day and extreme weather day in the upper left hand corner of the screen. When a hexagon is selected on either visualization, a colored circle will be plotted on the corresponding hexagon on both maps so that the user can compare easily where the streets are located on each map if the two maps are not in the same exact position. We utilized the same correlating circle and border color on the bars to signify that the hexagon clicked is the one being represented in the bar graph.

We chose the color of the barcharts in the main visualization to match the navigation bar on the top of the screen. The hue of the bars do not signify anything in this particular visualization, so having it be a uniform color with another element of our website worked perfectly. The border of the bars corresponds to the color of the circle plotted on the hexagon on the map. We chose a lilac and blue color to represent the first and second selected hexagons respectively. The two colors chosen showed up on every hue of the hexagons and we believed that the colors were easily distinguishable on every single hexagon.

In the natural disaster visualization, the hue of the bars in the two barcharts in the tooltip differ. We chose an orange hue for the extreme weather day and the same blue as the main visualization for the normal weather day. In the case of our natural disaster visualization, the hue of the bar did signify which day it was and we thought that the different colors signify that the days are different.

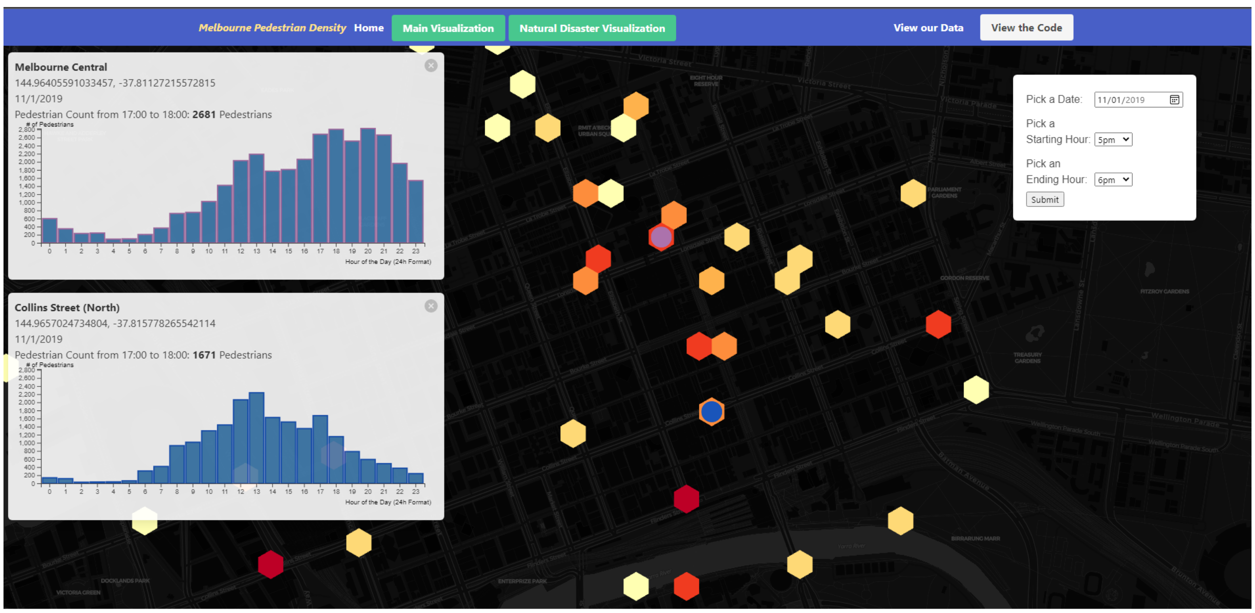

Figure 5. The final iteration of our main visualization. Border colors match the circles on the clicked hexagons on the barcharts.

Figure 6. The final iteration of our natural disaster visualization. The tooltip is in the top left corner and selected hexagons show up on both maps.

Overall, we are very proud of our final visualization. We believe that we answer both of the questions that we set out to answer at the beginning of the project. On top of that, we have related our visualization successfully to the research that we found.

We did differ a slight bit from our proposal. Instead of an on hover tooltip, we changed it to an on click tooltip that stayed in the top left hand corner of the screen until it was overwritten or deleted. We did change the hues of the bars in the tooltip. The maps did not change, but we altered the angle at which the hexagon bins were presented. We believe that all of the differences from the prototype positively enhanced the experience that the user has while interacting with our visualizations.

Implementation

Main Visualization

This visualization shows the hourly pedestrian density on a certain street, day, and hour in Melbourne, Australia for the month of November, 2019. The hexagon layer hovers over a street and its color presents the pedestrian count for that specific street. By hovering over a hexagon you will be greeted with the street name, coordinates of that street sensor and the count of pedestrians for that time range (figure 1).

Figure 1. Hovering over Hexagon

You can also filter by date and hour range using the filter box on the map. You can pick any day in the month November, 2019. You can choose any time range by picking a starting hour and an ending hour (figure 2).

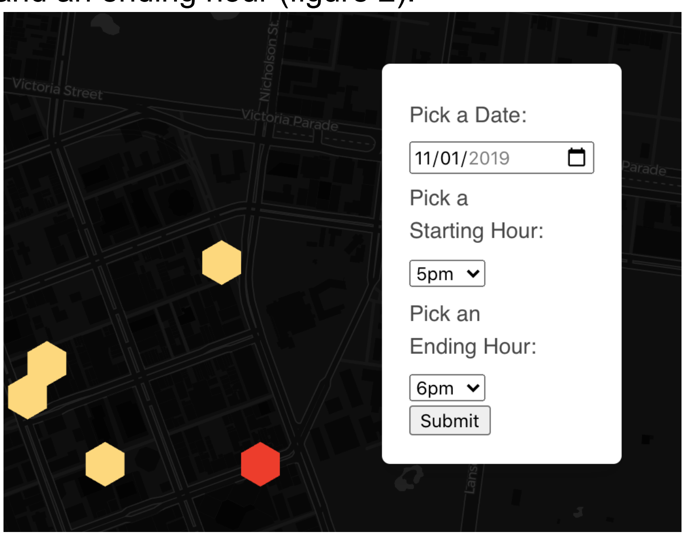

Figure 2. Filter Box on Visual

You can click on a hexagon and a tooltip will appear on the left of the screen showing the street name, coordinates of that street sensor, the date, the pedestrian count for the time range, and a barchart that shows the pedestrian count for each hour of the day. The x-axis presents the time with the labels 0-23 where 0 = midnight-1am, 1 = 1am-2am, 2 = 2am-3am, ..., and 23 = 11pm-midnight If you click on another hexagon, that hexagon's tooltip will show below the tooltip you first clicked Each hexagon you click and its corresponding tooltip will have the same color to let you know when tooltip corresponds to which hexagon. For example you can see in the figure below that the two hexagons I clicked have a colored circle and that color is the border of the bars in the barchart for that hexagon. Y-axes for the barcharts will be resized according to scale in the comparison (figure 3).

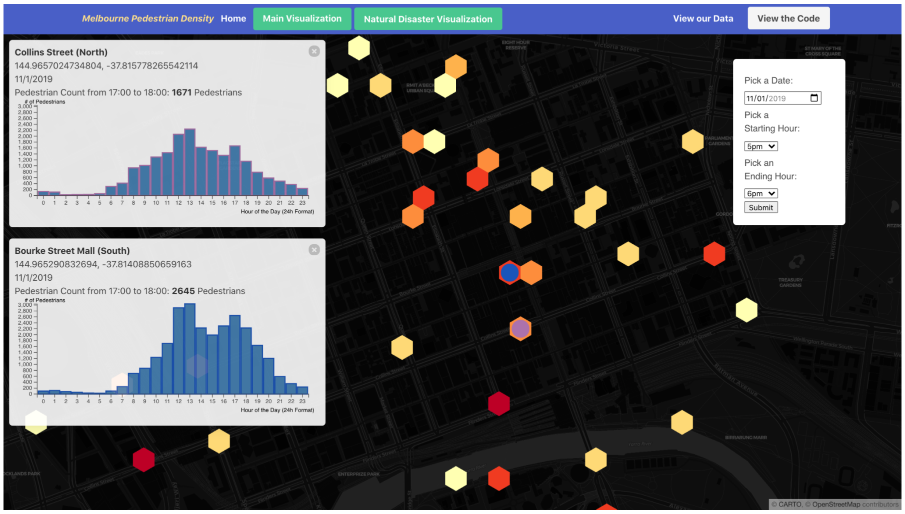

Figure 3: Tooltip on Visual

Natural Disaster Visualization

This visualization functions exactly the same as the main visualization with some changes. We are presenting both dates we are comparing as side by side visualizations. The top visual concerns the extreme weather day and the bottom visual shows the normal weather day. We are also utilizing the same filter box (excluding the option to filter by date) and hover information box as the main visualization (figure 4).

Figure 4. Natural Disaster Visualization

The tooltip popup is the same as the main visualization except that when you click a hexagon in either visual it will show the barchart of that hexagon you click as well as the bar chart of the equivalent hexagon in the other visual, if it exists. Each selected hexagon will have a colored circle on top of it to show that it was clicked (figure 5).

Figure 5: Natural Disaster Tooltip

Evaluation

What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?

There were multiple insights gathered from the visualizations. In the main visualization, a broad level insight is that the certain streets such as the Princes Bridge, The Arts Centre, Elizabeth Street, Bourke Street Mall, Melbourne Convention Exhibition Centre, remained the most busy streets at any given day in the November 2019 date averaging about 21000 plus pedestrians from 8 am to 11 pm. Another insight was that most pedestrian density occurred from around 7 am to 11 pm and there were less pedestrians walking in the early morning hours. The streets with sensors to the north of Victoria Street around Lincoln Square and Argyle Square had an overall lower density of pedestrians compared to streets in downtown Melbourne at any given day and time. The streets by the bridges pedestrians need to cross to make it into South Melbourne from downtown Melbourne always had a high pedestrian count at any given day and time. There were no changes when comparing the density of pedestrians on streets on a weekday compared to the weekend.

The natural disaster visualization, where we compared pedestrian density of a normal weather day (May 2nd, 2019) to an extreme weather day (November 21st, 2016) where an asthma thunderstorm formed around Melbourne killing 9 and hospitalizing hundreds showed us a new insight of our dataset. We compared the pedestrian density of both days for the time range of 5 pm - 10 pm. The reasoning behind this range is because the asthma thunderstorm started at 5:30 pm. We saw that most streets located in the center of Melbourne had less pedestrian traffic during the extreme weather day compared to the normal weather day. More specifically about 1000-2000 pedestrians less. For example the street Bourke Street Mall had a 10354 pedestrian count on the normal day compared to the extreme weather day which had a count 8219 pedestrians.

Overall, our visualizations answered both our questions, **1) How can the knowledge of massive human movement in Melbourne contribute to optimizing infrastructures such as energy, traffic, and urban planning? and 2) Is pedestrian movement influenced by a deviation in weather (e.g. rain, extreme heat) or a natural disaster (e.g. earthquake, flood)?** Our first visualization allowed us to answer the first question by presenting a clear picture of pedestrian movement in Melbourne. This visualization tool can be used by lawmakers and company officials when deciding on how to optimize infrastructure in Melbourne by seeing which streets are the most dense and improving the infrastructure surrounding that street. The natural disaster visualization allowed us to answer our second question because we saw that on the natural disaster day of November 21th, 2016 from 5:30 pm (when the asthma thunderstorm started) to 10 pm there were overall less pedestrians walking on normal busy streets when compared to a normal weather day (May 2nd, 2019).

Our visualizations worked as intended and we are confident in saying that we were able to accomplish the visualization we aspired to create in the beginning of the project. We believe that our visualizations are easy for any user to understand and manipulate. We also believe that our visualizations give any user the ability to formulate trends, see patterns, and watch a story unfold from the filtering options the visualization has.

We thought of one way to improve our visualization presentation. Instead of showing aggregation of pedestrian counts by just changing the color of the hexagon for a date range that the user chooses from the filtering dropdown menu, we would like to use some sort of play button for a range a user picks that will change the colors of the hexagons as the time picked goes by. For example let’s say the user wants to see the pedestrian change from 12 pm to 2pm. Once they choose that range, they press play and the colors of the hexagon will change as it shows the counts from 12 pm - 1pm and the transition to 1 pm - 2 pm.