Room Optimization for Stockton University NAMS:
Graphic Interpretation of Room Usage for Identification of Low Occupancy Locations

Background:

Overhearing my assistant director speak about percent usages of rooms in the USC, I thought it would be a good project to visually depict the numbers he made up in excel. In his work he only shows percentages but I wanted to take it a step further to answer these questions: At what times are they not being used? How many students are in that room? Would it be easier to place these values on building layouts to provide nested information?

Using the following coding languages and libraries, both learned from this project and previous knowledge, I was able to present my graphics in a website based format:

Previous knowledge: Languages: Python, HTML, CSS, R
Libraries: Scrapy(Python), Matplotlib(Python), Jupyter(Python)
Learned: Languages: Javascript
Libraries: D3.js(Javascript), Pandas(Python)

Procedure/Process:

I began by using Scrapy to download the information off of Stockton's website. The information included:

Class Name, Times, Section Number, Program, Days

Multiple attempts and fixing of the code had to occur here. I have used **Scrapy** intensively in the past so it wasn't much trouble but this is where the bulk of my work had occurred. It could be classified mostly as **Data Wrangling**.

The problem here was to fit the data that was coming off the HTML file to fit into a csv format. From there, there were "holes"(empty lines that when human read, it implies that the information is the same as the line above it), different encoding (utf-8 to ascii) and splitting data columns (8:00-9:00 to 8:00 in one column and 9:00 to another). These were fixed by writing code in **Python** to find empty lines and use previous lines to copy data, deleting abnormal characters and using split strings, respectively.

After scraping the website, the data had to further be manipulated using **R**. There were entries in the data that were one offs. These were classes that moved temporarily for a day to a computer lab. This data does not help since I cared mostly about permanent class locations. After removing those, I tried to continue to use **R** but I noticed I needed to go back to Python to use the **Matplotlib** libraries.

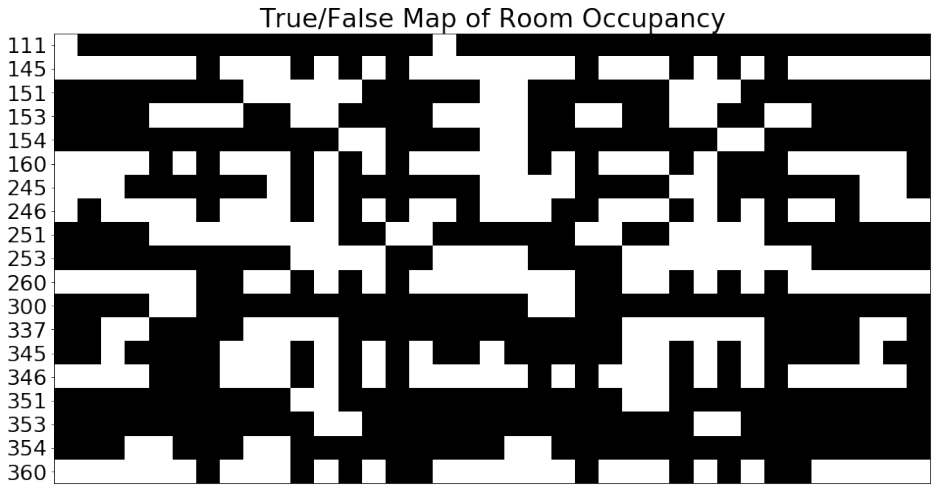
From here I had to sit back and decide what information can be plotted and how. My initial inspiration into data visualization was Tufte so I wanted to follow his principle of Data-Ink Ratio. I also wanted to mix in encouraging the eyes to compare data. I wanted to reader to explore the data and make their own assumptions, not me to push what conclusions I found in the data. I also wanted to practice visually appealing (novel) graphs against familiar graphs (bar graphs, layouts) as mentioned in Cairo's Wheel. So I came up with this solution:

- Break down the data in several ways depending on what the reader wanted to find out.
- Place all the information from simple to complex into the graph but not force the complex data on the reader. If they wanted more data, it was there for them to find but not a requirement to understand what was going on.

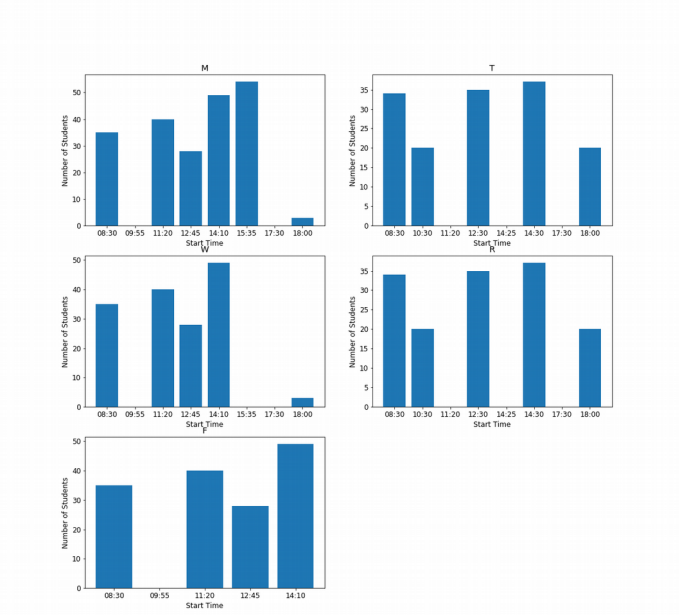
- Try different formats of displaying the data, introducing redundancy. That way if one method of displaying the data doesn't work, the other formats can supplement the others.

Following these guidelines (And recommendation from Professor Mick) I based my project around delivering my graphs in an HTML format on building layouts.

Initially I played around on **Jupyter** and **Matplotlib** and **Pandas** to create a grid graph of **True/False boxes** to denote if the room is in use or not. I had to create loops and empty dataframes to create a mapping of these. The problem with this was there was a lot of information on one graph (A lot of boxes that the reader had to see and then try to interpret what it was saying.) I kept this graph as a conceptual graph to show empty spaces but nothing more. Additional information was layer on this by adding a color scale to show what program (Chemistry, Biology, Physics…) uses that room. Again, not very multidimensional.

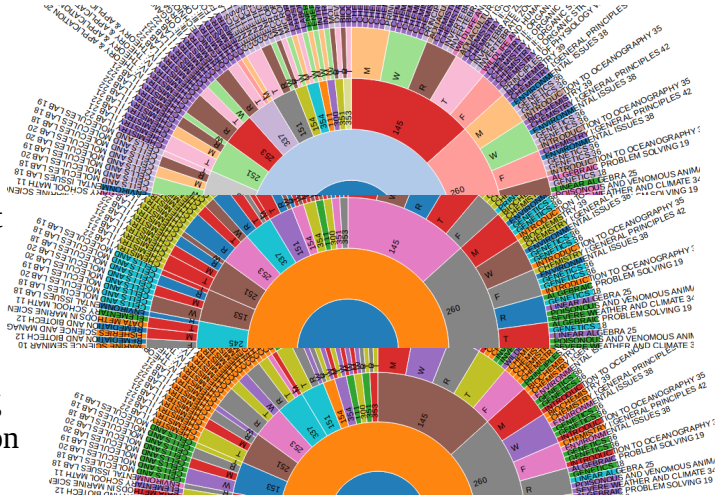


Upon receiving the building layouts I wanted to place some information on it. I wanted to label each room with its name, occupancy percent and a More.. button in case the reader wanted to see in depth why the room had that percentage. This is where I had to use **HTML** coding to place that information on there and **Javascript** to create functions that would dynamically create buttons and text based on what floor was chosen from a drop-down. I wrote a JSON file by hand (I would not recommend this and I will describe how to work around this later) to load the required information of every room, their respective pixel locations, and where the button would point to. I then wrote some code to create bar graphs of each day in one image of each room and saved each image.



In the class forums, d3.js was being mentioned and I decided to look at it. That was when I found a graphic that had the potential of nesting information. It was the sunburst. This required more planning as to how I wanted to nest the information. It was determined to break down to Building>Room>Day>Time-Number of students. The problem with this code was that the data that was fed into it was in JSON format but multileveled nested data. This could not be written by hand and there is no program or code to take my csv data and transform it to that. Therefore I had to write a code that created the JSON file for me. This process took a while but learning from that I noticed I could

write any CSV file to JSON now. I fed this into the sunburst code. Then came to the part of determining what colors to use since I wanted to focus on the program usage. I changed the code to use the 20 category color scale but that didn't work because it didn't allow much variation between the programs. Then I change it to 10 colors. This didn't allow much separation. I then separated the use of the scale to separate variables with the outermost ring using its own scale. This worked as I wanted to. I also changed the code to include the class name and number of students on the outermost ring sections. I tried loading all of Stockton's information but this caused lag issues since it is loading a lot of data. I kept it to USC usage and it looked fine. This ring clearly denotes which rooms are being used a lot compared the least used rooms.



Analysis:

True/False Map:

This map was created as a conceptual graph to speed up the process of seeing where the "problem" areas were. This reduces the eye-movement principle to a left-right movement since there is no x-axis to compare to. I believe this to be the simplest form I could have portrayed the information.

Bar Graphs:

These aren't as novel as the true/false maps but are easily readable and familiar to readers. The problem with these is that only so much data can fit on these before they become too cluttered. These bar graphs focus on what what the goal of this project was, to show lapses in room scheduling. This minimalism and focus is more geared towards Tufte's principles of show and not distort data, comparison vs description. It fails at Tufte's principles of high resolution. The bars in the graphic takes up most of the space without adding any value to it.

Sunburst:

These are quite novel and fit a lot of data efficiently, with more data being revealed as you click on the data. Using Cairo's wheel I can break down the graph to its major points (ones that I believe were strong in). It is .5 in density, .75 in multi-dimensionality, .9 in originality, 1 in figuration, 1 in functionality. This is what I strove to do with this graph. I did not want to do a complex graph where only select people would be able to interpret it. Every element in this graph serves a purpose. I also used a different way to portray information for the "wow" factor but it doesn't hinder what the graph is trying to show.

Conclusion:

Going forward another way I could display the data would be Plot.ly. It was an amazing way to display the graphics, especially since this project is meant to be deployed on a website. More research can be put into area visuals based on the amount of time the room is used. For this project I believe there is no one solution since the range of students or number of sections in each room can cause some data to be hidden in visualizations.

While creating the sunburst, loading all the school's data slowed it down significantly. If the sunburst proves to be a useful tool for readers I could write code to create sunbursts for every Wing/Building (eg. F-Wing, C-Wing, West-Quad, the upcoming USC2).

Lastly, in its current state, the process for creating these graphs is not fully automated. I have to launch scripts in progression and go into the CSV files and move and delete columns. I want to be able to run one script and have it display all the required graphs and information. This will allow the process to be quicker and easier to automate, especially to those not savvy with programming.