Philippe Naviaux and Shammu Meyyappan
Professor Itay Hen
DSCI 510
December 15 2025

LA Hospital Finder

Our project is called 'LA Hospital Finder', a collaboration between Philippe Naviaux (pnaviaux@usc.edu, #3619671286) and Shammu Meyyappan (meyyappa@usc.edu, #4085790637). The goal of this project was to consolidate information on hospitals for residents of Los Angeles. First, we used web scraping to obtain the specific services offered and insurance providers accepted for hospitals across LA. We then created a tool allowing users to find a specific hospital based on their desired service or insurer. Lastly, we analyzed the relationship between the number of services offered and insurance providers accepted using a correlation coefficient.

We began the data sourcing process by conducting some background research. Using a feature layer provided by GeoHub LA City detailing all the hospitals in LA County, we selected only the hospitals located within the city limits of LA using a SQL query in ArcGIS Pro, providing us with a list of 22 hospitals whose websites we obtained via a Google search. We then wrote the get_data.py file utilizing the Beautiful Soup library and HTML parsing to obtain the listed services/insurers based on user input regarding the keywords signifying the beginning and end of each list that should be scraped before adding everything to a CSV. However, 5 of these websites had to be eliminated from our pool due to encryption/scraper protection we encountered, and 10 others had to be eliminated due to them not listing the specific insurance providers they accepted on their website. Lastly, both USC hospitals used the same website despite serving two separate locations, so the same website was used for both, leaving us with a total of 8 data samples collected (from mlkch.org, ladowntownmc.com, sch-hollywood.com, lach-la.com, uclahealth.org, keckmedicine.org, and dignityhealth.org) and saved to a CSV.

The cleaning process involved reiterating over each website's lists to only accept the text (using `get_text(strip=True)`) located between two specified keywords (indicating the beginning and ending of the list). This way, other text from the webpage can be avoided. The respective CSVs are updated accordingly.

For our first visualization, we created two bar charts listing the hospitals and the number of services offered and insurance providers accepted. This was achieved by using matplotlib with pyplot to visualize the data stored in the CSVs. The data from the CSV was appended to lists, which were summed to get the total number of insurancers and services. A tertiary list also provided the names of the hospitals for the bar chart.

Our next task was the creation of an analysis tool allowing users to find matching hospitals based on their desired service or insurer. The data from the CSVs was appended to lists for insurers and services. Then, we wrote a function for searching these lists for the name of a specific service or insurer that the user inputs, adding any that matched to a new 'found' list,

which is printed to the user. If no matches are found, then the user is informed of this via a print statement as well.

Our next visualization is a map of the various hospital locations. Websites were scraped again this time for the hospital address, which was added to the respective CSV as a new column. These CSVs were then brought into ArcGIS Pro where a ModelBuilder model was created and output as a .py script (util.py). This script geocodes the addresses to locations on the map and then displays them, one using ArcGIS Pro and another in a Jupyter notebook (Figures 1 & 2).

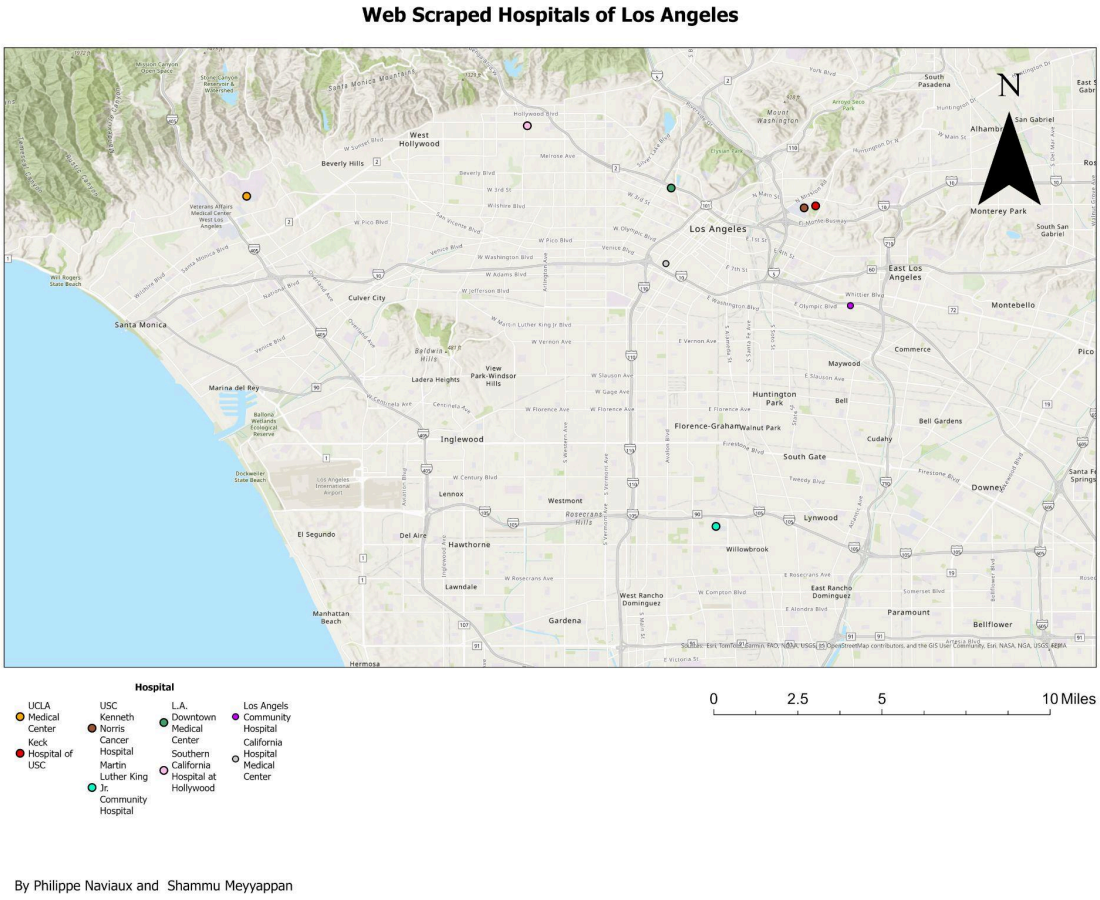

Figure 1: ArcGIS Pro Map Visualization of Hospitals

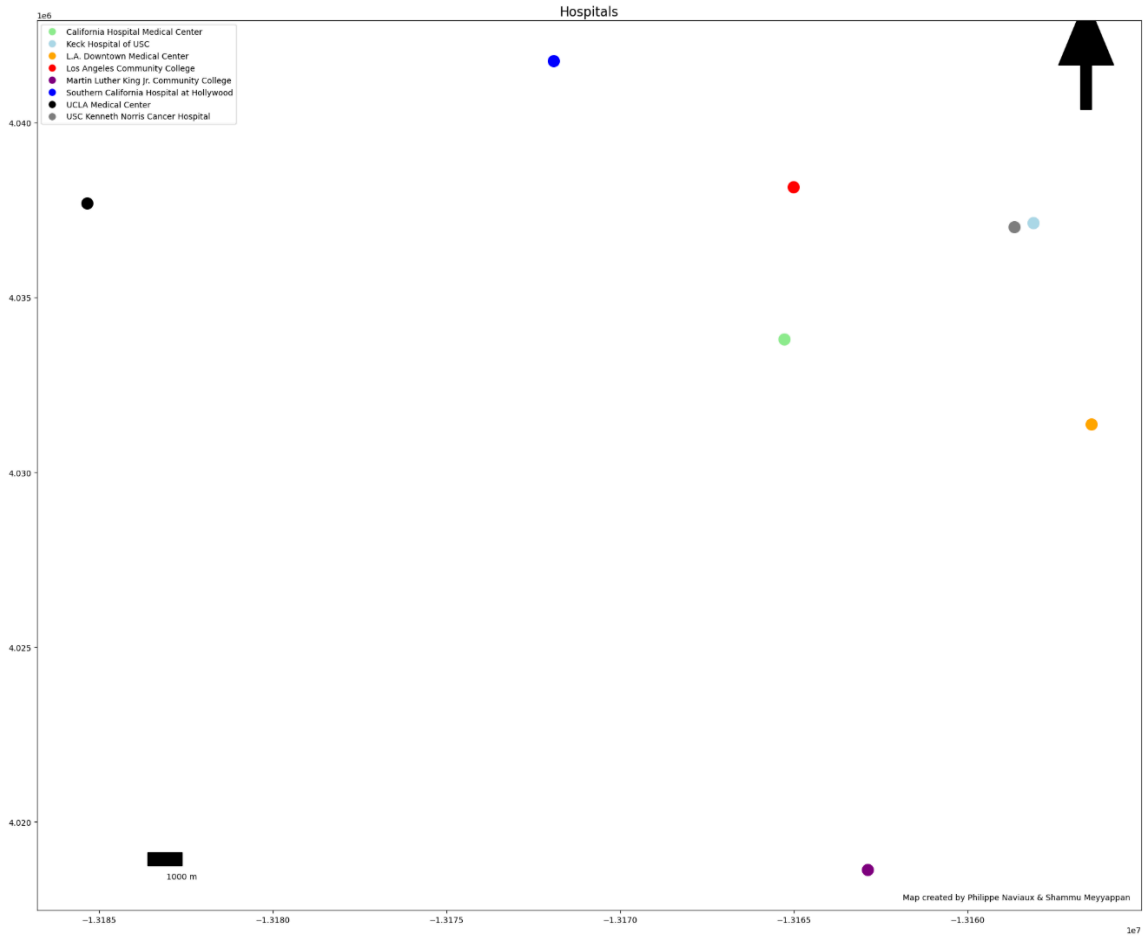

Figure 2: Jupyter Notebook Map Visualization of Hospitals

For our analysis portion, a correlation coefficient was run to see if there was a correlation between the number of services offered and the number of insurance providers accepted across the various hospitals included in our study. In order to run this analysis, all the CSVs were stored as lists. Then, a list of lists for both insurers and services was created. Then, a correlation coefficient for the insurers and services was run, showing a correlation coefficient of 0.82. The scatter plot shows where each hospital lies between insurance accepted and services offered. The figure below shows a positive correlation between the number of services offered and insurance providers accepted (Figure 3).

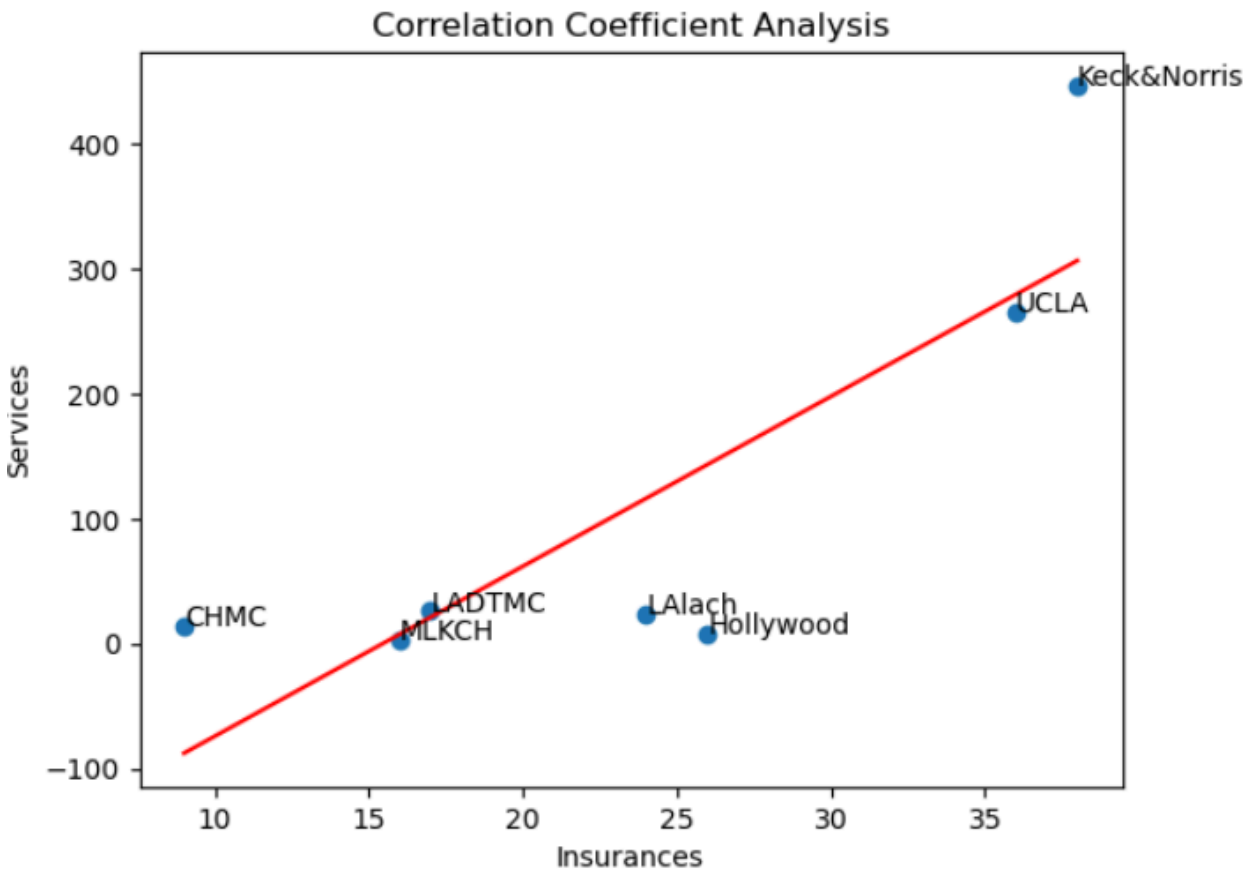

Figure 3: Correlation Coefficient Analysis

We hypothesized that as the number of services offered by a hospital increased, the number of accepted insurance providers would increase as well. This was backed up by our analysis, as seen by the positive correlation coefficient. Although we cannot assess causality from a purely correlatory study, one conclusion that may be drawn is that larger hospitals offer more services, and larger hospitals are more likely to be covered by more insurers. Whereas hospitals that specialize in fewer services might struggle to be covered by as many insurers. But we cannot accept this conclusion as true with our limited sample size.

The first challenge we faced was from websites being encrypted or failing to list their accepted insurance providers or services offered. The only workaround we could manage with our current skillset was to focus on the data we could actually parse with Beautiful Soup, as we had practiced in class, and discard the rest. We then struggled to script a filterable map visualization, and instead opted to create a text-based filter in which the user inputs their insurance provider or desired service, and a list of matching hospitals is output.

However, the biggest challenge our group faced was in creating a repository for the first time. We did our scripting in Jupyter notebooks and struggled to convert them into standalone .py scripts, causing some of our code to be a bit messier than we would have liked. In the future, we will plan out our code by outlining functions that can be called repeatedly called rather than copy-and-pasting code and changing variables as one might in a Jupyter notebook. We then

struggled to collaborate via GitHub due to our relative inexperience with the platform. Although we failed to meet every listed criterion, we hope the effort displayed and results showcased in our visualizations.ipynb file can award us with partial credit where applicable.

Given more time and experience, we would expand the capabilities of our HTML parsing script so that it can search through all the pages within a website to locate the ones that list the services offered and insurance providers accepted. We could then make use of the Google Places API to query for all the hospital websites within a region, so the user doesn't have to find them themselves. We would also improve the reasoning of our algorithm so it can identify where the list is located within a webpage without having specific keywords inputted, potentially by hardcoding specific insurer/service names to look for. We would also attempt to add a visual element to our text-based filter so that users can visualize where each of their matched hospitals is located within the city. This project could then even be repurposed/expanded beyond the city limits of LA to other cities/countries across the globe.