DAP 2

12/6/2024

Nasser Alshaya, Mario Venegas, Daniel Avila

Final Project

Final Project Writeup

This work analyses the use of public transportation within Hyde Park in Chicago. As a large proportion of the people in Hyde Park are students of the University of Chicago, this analysis uses Divvy Bike and CTA bus data to define recommendations around changes the use of services by the University given the preferences and demands of students during different times of the year. For the analysis, the team used two main sources of data:

- CTA Ridership, Daily Boarding Totals: This dataset shows systemwide boardings for both bus and rail services provided by CTA, dating back to 2001.
- Divvy Trip History Data: Divvy offers quarterly and monthly datasets of every trip taken since 2018.

This data had to be cleaned and limited to (a.) the time period of interest (2018-2023, which were selected based on the availability of data from Divvy), and (b.) the neighborhood of interest, Hyde Park. As part of our analysis, after the data was cleaned, the team made graphs to visualize our output. Finally, the team made a dashboard that enables policy makers to visualize this information for particular timeframes. The steps followed for this are enumerated below. Each number corresponds to a chunk in the coding section of the document.

1. As mentioned above, the data base from the CTA includes information dating back to 2001, and for every bus route of the CTA in Chicago. The coding shows the steps to limit the information to bus routes 171 and 172 - which operate only in Hyde Park, and to observations corresponding to 2018 onwards.
2. After the step above, the CTA data included information for each day from 2018 to 2023. However, due to the fact that the analysis seeks to determine the demand by time of the year, the team decided to group the data by year and month.
3. This code block creates a list that captures the average number of rides on a monthly basis across all years from the previous observations_per_month variable. We then create a data frame that matches the months with the average monthly observations, and plot this into a bar chart. Finally, we use the same data to create a heatmap, however the heatmap is created where the X axis is the month, and the Y axis is a year, which means that each cell is a unique month of each year. By

observing the year-month, we can clearly visualize the seasonal pattern of ridership in the data on an annual basis.

4. The information provided by Divvy regarding every ride taken in Chicago was uploaded quarterly from 2018-2020 and monthly from 2021-2023. In addition, the names of the columns included in the data bases changed at some point. This chunk explores the differences between each data base, monthly dissemination and quarterly dissemination.
5. In addition, the Divvy data includes information for every trip made in Chicago city limits. The data had to be cleaned to include information for Divvy docking stations in Hyde Park only. To identify these docking stations the team used the following data bases:
    - chicago_areas. This data base from the Chicago City Government includes geometric coordinates for each area in Chicago.
    - divvy_stations. This shape data base from the City of Chicago includes the geographic coordinates and names of every Divvy docking station in the city.

The general steps followed to identify the docking stations in Hyde Park using this information were the following:

- Convert the geometry column in chicago_areas to shapely geometry to make it compatible with the station_gdf, mwhich contained the geographic information of divvy docking stations.
- Limit geographic location to Hyde Park.
- Perform spatial joint to print the name of the Divvy Stations in Hyde Park.

6. Once the names of the stations in Hyde Park were identified, the team used them to limit the observations in the Divvy data sets to trips that started and ended in Hyde Park. This makes the analysis more comparable to the use of routes 171 and 172, which only operate inside Hyde Park. Since the information was provided in data sets with different column format, the team had to do three different scraping codes from a folder containing all data bases uploaded by Divvy. The code cleans each of the data sets from Divvy so they only include trips that started and ended in docking stations identified within Hyde Park. Once all the data bases were cleaned, they were concatenated into a single file. Chunks 6.1, 6.2, and 6.3 show the process described above for the three column formats identified.
7. This code block is simply naming the datasets to variables so that they can be referenced in preparation for joining into one large dataset.

8. We first create dictionaries in preparation for renaming the columns such that we can concatenate the data frames together on a column by column basis. Lines 445 - 450 rename the columns such that all columns across the three datasets are the same, and line 450 creates the singular concatenated dataset. Lines 452 through 455 are for data cleaning purposes, while line 458 groups the data by station and date, naming the new column "ride_count". Ride_count will be essential in the following visualizations and analysis.
9. Line 467 is creating a new column titled "year" within the concatenated dataset based on the year of the "year_month" column. We then break the concatenated dataset into two variables, one that contains data pre 2020 and another that contains data post 2020 to highlight any effects that may have occurred due to Covid. Additionally, it is worth noting that instruction was fully remote for 2020-2021, which further influenced our decision to create two new graphs for our visualizations. After breaking the data into pre and post covid, we create two lasagna plots to demonstrate Divvy ridership on a year_month and per station basis.
10. In this code block we attempt to generate a average monthly Divvy ridership to compare to the lasagna plots above. We want to see how an aggregated analysis from 2017 to 2024 demonstrate ****average**** monthly ridership in contrast to the lasagna plots, which show ridership on a monthly basis for each station. We group the data by year month, and take the mean of the ride_count column, saving that information in the newly created "avg_ride_count" column. Since the monthly data is saved numerically, we use the months dictionary to map the numeric values via a lambda function to the names of the month, ensuring to save the data in the column as a string. We then further groupby the month column to create a bar chart of the average monthly Divvy utilization. The benefit of visualizing the data with the lasagna plots in comparison to the bar chart is that we are able to get extra information from the lasagna plots, such as which stations are most active and when, which is not visible in the bar plot.
11. This code block creates a file that is essential to plotting data in the dashboard. Finally, we also created a multipage dashboard, one page for CTA rides visualizing aggregate rides from 2018 - 2023, and a page for Divvy bikes visualizing aggregate rides from 2017 - 2023.

We want to note that the dashboard was pushed separately to the GitHub repository for this project. These dashboards can be accessed directly in the "dashboard.py" file.

Link to Google Drive with Data