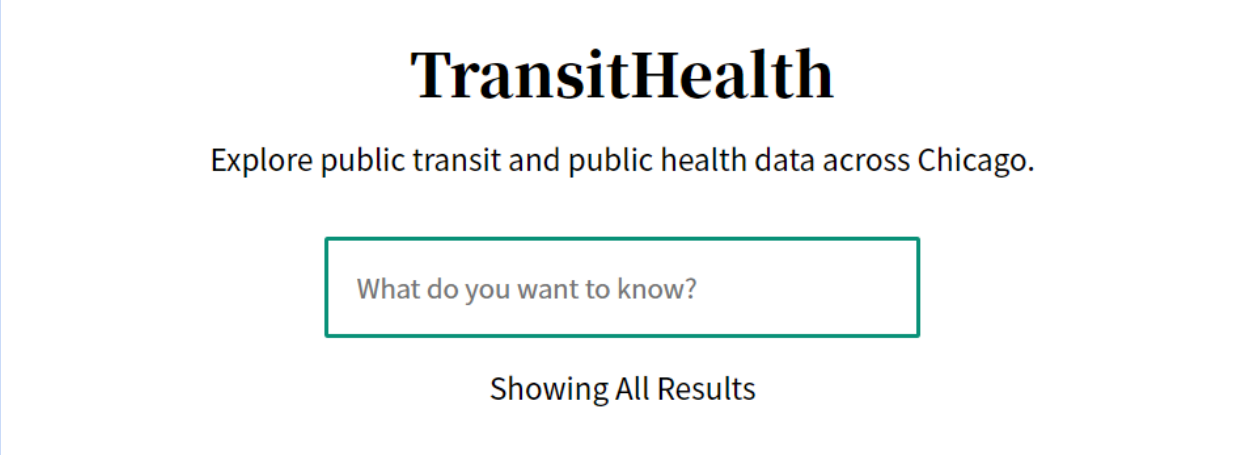
Introduction and Overview

During the Covid-19 pandemic, a research group I was a part of formed to build a web application that will give Chicago residents an idea of public transportation use by the numbers.

Main objective of the project was to inform local residents on how the pandemic has affected transit use. We felt a responsibility to understand the data and distribute it to highlight key observations. The group as a collective looked at multiple avenues of transportation in the city such as the metra train, bus, e-scooter, bike, and many more.

TransitHealth Web Application



Done by constructing datasets that take a deep dive into the specifics of both Chicago as a whole and individual areas within the city.

For this Project I was tasked with two jobs:

- Build a visualization of the Covid-19 cases and deaths by the week from March 2020 to August 2021
- Illustrate Electric Scooter use throughout the city of Chicago separated by the area

Data Collection and Manipulation

Data was collected from the Chicago Data Portal. It should be noted that according to the portal, covid cases are counted when a patient receives a

positive PCR test while in hospitalization.

CHICAGO DATA PORTAL

- Imported into a Cloud9 server where data was extracted and distributed into an offline pipeline
- Extracted data was then manipulated and transformed using the languages SQL and Python to create specified datasets

Dataset groupings:

1) Grouped by age group (0-17, 18-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80+)
2) Grouped by week (a week being Monday to Sunday)

```
select
    date_extract_y(lab_report_date) as year,
    date_extract_woy(lab_report_date) as week,
    sum(cases_age_0_17),
    sum(cases_age_18_29),
    sum(cases_age_30_39),
    sum(cases_age_40_49),
    sum(cases_age_50_59),
    sum(cases_age_60_69),
    sum(cases_age_70_79),
    sum(cases_age_80_),
    sum(deaths_0_17_yrs),
    sum(deaths_18_29_yrs),
    sum(deaths_30_39_yrs),
    sum(deaths_40_49_yrs),
    sum(deaths_50_59_yrs),
    sum(deaths_60_69_yrs),
    sum(deaths_70_79_yrs),
    sum(deaths_80_yrs),
    Min(lab_report_date) as week_start,
    Max(lab_report_date) as week_end
Where
    lab_report_date <= "2021-8-12"
Group by
    year,week
Order by
    year, week
```

Datasets were then loaded into an application programming interface (API)

Website

Code was implemented using Python to make metrics that are described in the API and then sent to the application.

```python
class Covid_CDD_Metric:
    """
    Metrics for community area data.
    """

    def __init__(self, con):
        self.con = con

    def cases_for_given_age(self, givenAge):
        """
        Returns the number of cases week by week from March 2020 - July 2021 given
        age (in "minAge_maxAge" format in following increments:
            0_17, 18_29, 30_39, 40_49, 50_59, 60_69, 70_79, 80_)
        """
        query = """
        SELECT
            week as date, cases_age_{givenAge} as value
        From
            Covid_cases_deaths_data
        """.format(givenAge = givenAge)
        print(query)
        cur = self.con.cursor()
        cur.execute(query)
        rows = rows_to_dicts(cur, cur.fetchall())
        return rows
```

The metrics are put onto the site using JavaScript to create a graphic user interface (GUI). This is also where labels for data are formed for easier user readability.

```javascript
    cases_for_given_age_0_17: {
      name: "Covid cases for ages 0-17 in the city of Chicago",
      units: "people",
      dataset: "Covid",
      description: "",
      format: Formatter.numberWithCommas,
      fullFormat: Formatter.numberWithCommas,
    },
    cases_for_given_age_18_29: {
      name: "Covid cases for ages 18-29 in the city of Chicago",
      units: "people",
      dataset: "Covid",
      description: "",
      format: Formatter.numberWithCommas,
      fullFormat: Formatter.numberWithCommas,
    },
```

Finished Product

Website Application URL: [TransitHealth]

Figure 1: Users can compare different age demographics on specified dates

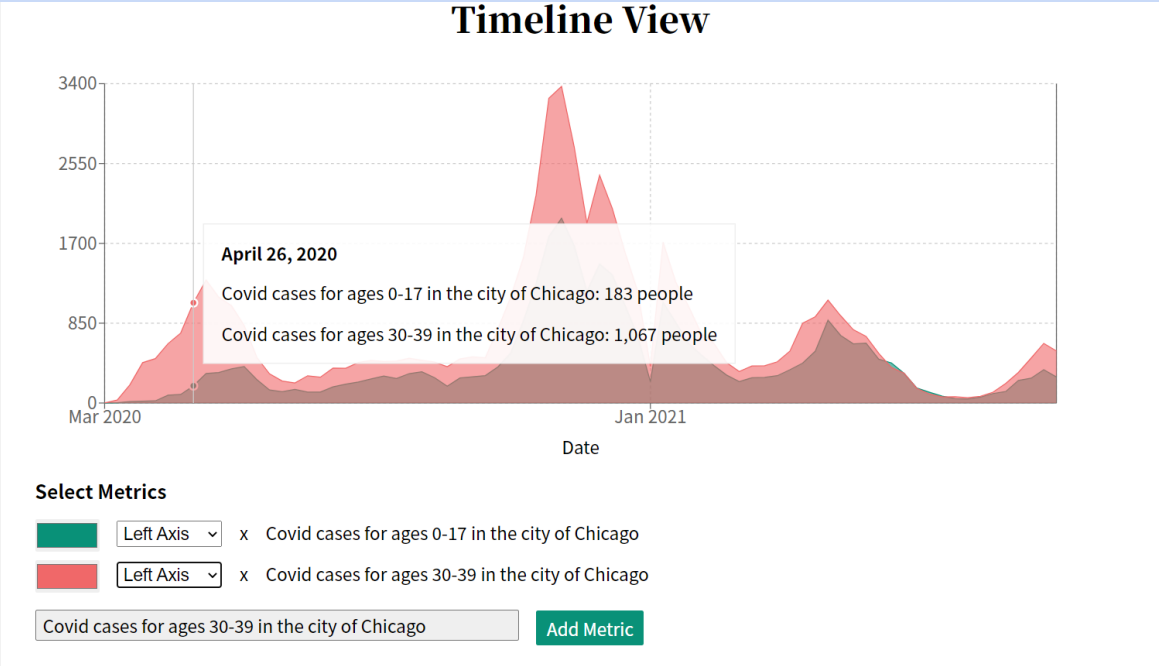
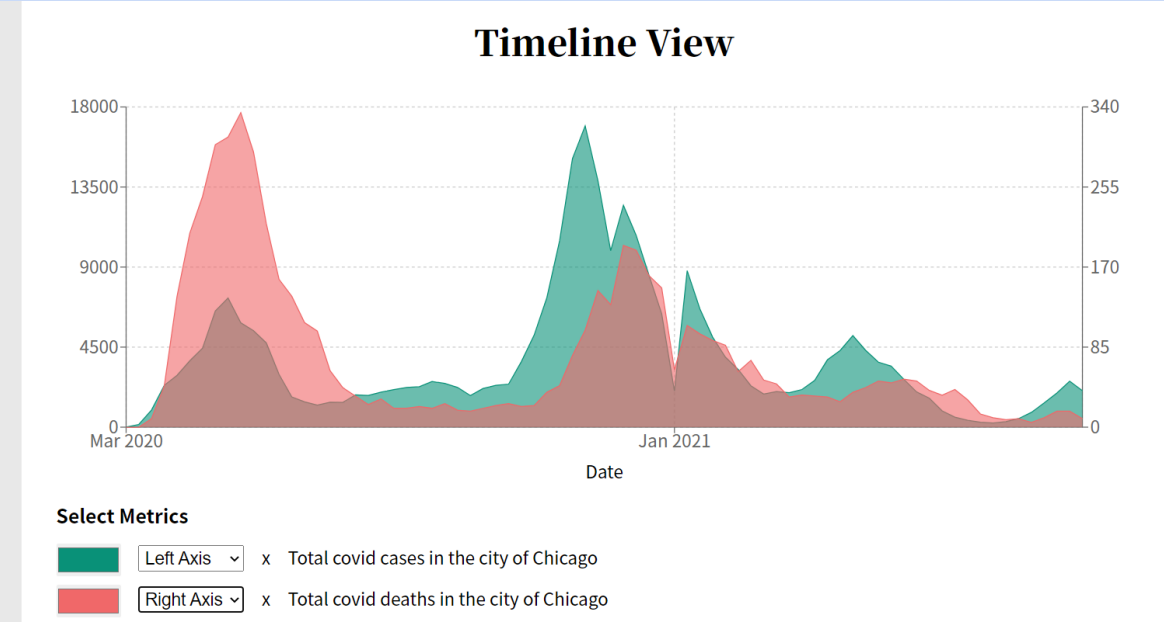


Figure 2: Users can draw patterns from data using multi-axis graphs



Escooter Use Data Collection

Electric scooters (also known as the E-Scooters) are a form of transportation that has recently expanded to public use provided by the city.

Similarly to the first project mentioned on this blog article, the Chicago Data Portal collected data regarding e-scooter use around the city. This time though, Chicago was divided into over 60 subsections where analytics can be examined. This in conjunction with other metrics made by my research partners can help illustrate different ideas.

Sample data table from the Chicago Data Portal:

Trip ID	Ste	End T...	Trip ...	Trip ...	Accu...	Start...	End ...	Start...	End ...	Start...
0b6cc3f2-...	06/15/20...	06/15/20...	688	3,921	152			31	31	LOWER W...
e081485f...	06/15/20...	06/15/20...	4,473	1,831	0			22	22	LOGAN S...
297b2b7...	06/15/20...	06/15/20...	2,078	383	152	17031833...	17031833...	28	28	NEAR WE...
4793f3c4-...	06/15/20...	06/15/20...	0	5	10			28	28	NEAR WE...
6a4a10f7-...	06/15/20...	06/15/20...	2,471	510	152			28	28	NEAR WE...
c590eb9b...	06/15/20...	06/15/20...	1,909	373	152	17031833...	17031833...	28	28	NEAR WE...
d81b3d7...	06/15/20...	06/15/20...	1,099	312	152			28	28	NEAR WE...
e092683a...	06/15/20...	06/15/20...	1,361	315	152			28	28	NEAR WE...
d5ec7eaf-...	06/15/20...	06/15/20...	1,073	299	152	17031833...	17031833...	28	28	NEAR WE...
11c798d1...	06/15/20...	06/15/20...	1,319	661	10			26	26	WEST GA...
9103d3cd...	06/15/20...	06/15/20...	128	145	0			24	24	WEST TO...
28df76c0-...	06/15/20...	06/15/20...	3,615	19	1					
6908a3f6...	06/15/20...	06/15/20...	53	900	152			31	31	LOWER W...

Escooter GUI Construction

Data was manipulated using SQL and Python, constructing sets of statistics that were desired to be graphed:

```
select
    start_community_area_number, end_community_area_number, count(trip_id), avg(trip_distance)
where
    start_time <= "2019-10-16"
group by
    start_community_area_number, end_community_area_number
```

```
# Convert community area numbers to integers
df["avg_trip_distance"] = df["avg_trip_distance"].astype(float)


#transform avg trip distance from meters to miles
df["avg_trip_distance_miles"] = df["avg_trip_distance"] / 1609.0

df.to_csv(args.output_file, index=False)

end = timer()
secs = end - start

print(f"Transformed {len(df)} records in {secs:.1f} secs.")
```

Notes About the Data:

- Data was collected for the year 2019
- Each scooter trip was given its own unique id
- Two of the three characteristics of a trip includes which area of Chicago was the e-scooter rented out from and where the e-scooter was located when returned
- The final characteristic describing a trip is the trip distance

Website

Controlled data was then loaded into a API to make metrics:

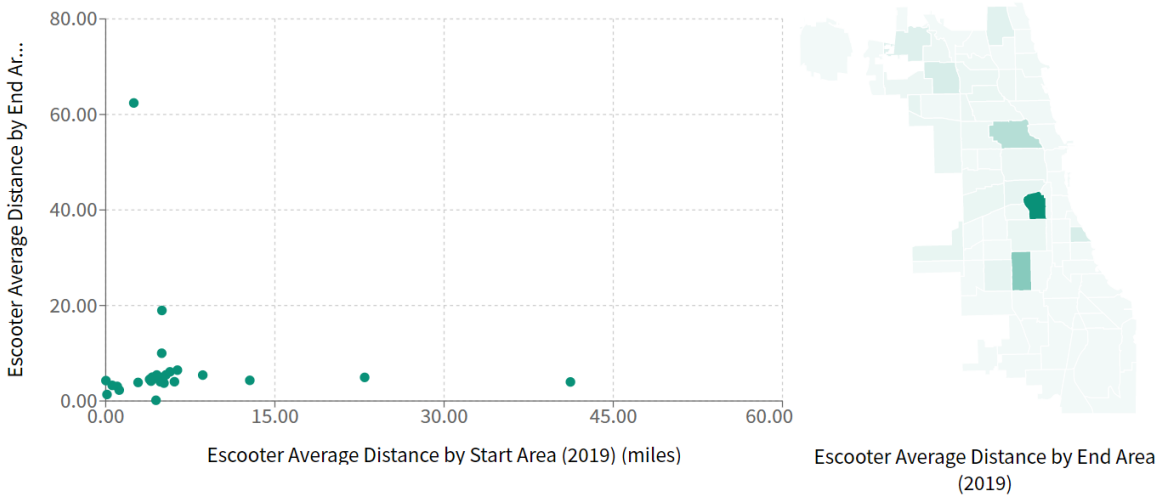
```
def number_of_trips_based_on_start_cn (self):
    """
    Returns the number of trips from all communities that had at least
    one ride start there
    """
    query = """
    SELECT
        start_community_area_number AS area_number, sum(count_trip_id) AS value
    From
        Escooter
    Group by
        start_community_area_number
    """.format()
    cur = self.con.cursor()
    cur.execute(query)
    rows = rows_to_dicts(cur, cur.fetchall())
    return rows
```

Finished Product

By Community Area

X Axis: Escooter Average Distance by Start Area (2019)

Y Axis: Escooter Average Distance by End Area (2019)

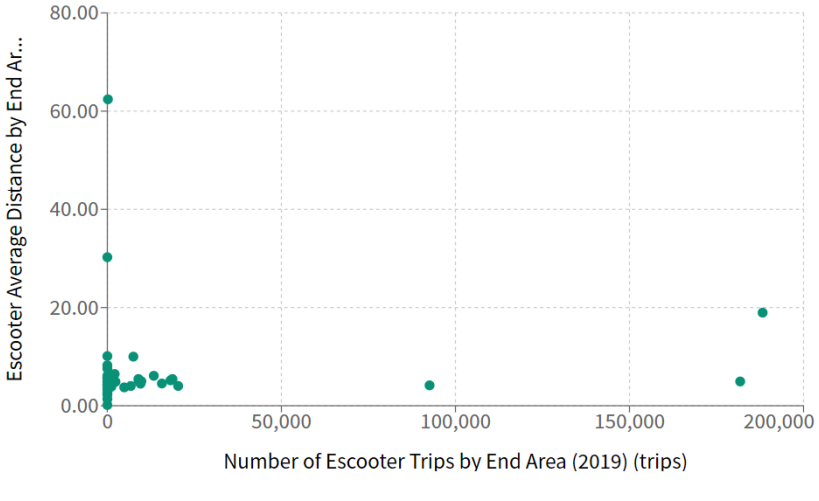


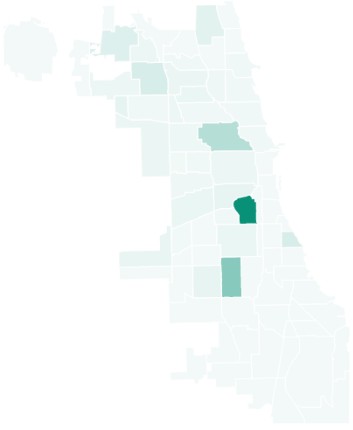
Escooter Average Distance by End Area (2019)

By Community Area

X Axis: Number of Escooter Trips by End Area (2019)

Y Axis: Escooter Average Distance by End Area (2019)





Escooter Average Distance by End Area (2019)

References

Check out the Official Transit Health Website:
https://scarletstudio.github.io/transithealth/

If you are curious at looking at my individual pull requests, take a look at this link:
https://github.com/scarletstudio/transithealth/pulls?q=is%3Apr+is%3Aclosed+author%3Atalemu