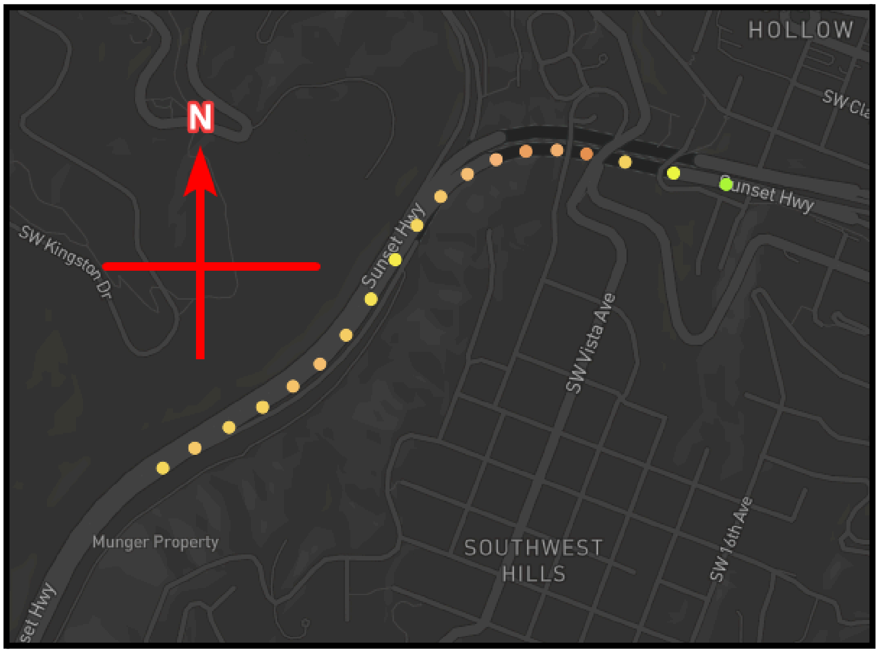
DataEng S24: Project Assignment 3

Data Integration

Submission

Here is a link to my code: https://github.com/NAlexH2/de-proj-cs510
You can find milestone02 in `‘milestone-submissions/milestone03’`

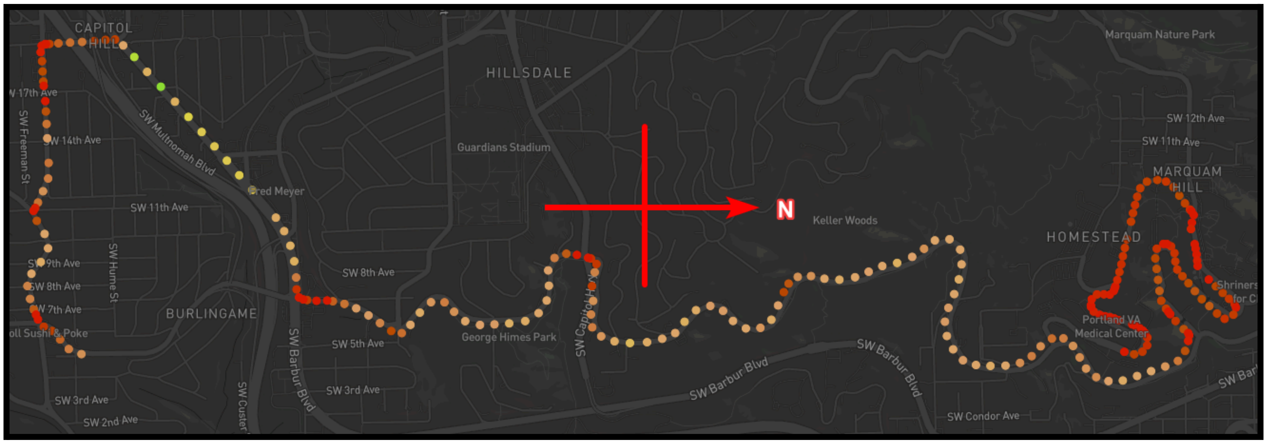
Visualization 1. A visualization of speeds for a single trip for any bus route that crosses the US-26 tunnel. You choose the day, time and route for your selected trip. To find a trip that traverses this tunnel, consider finding a trip that includes breadcrumb sensor points within this bounding box: [(45.506022, -122.711662), (45.516636, -122.700316)]. Any bus trip that includes breadcrumb points within that box either drove across the tunnel or teleported across!
The GPS coordinates are off by a little bit, so here’s my visualization of a bus traversing the whole tunnel. This is for 01/23/2023 between 4:12pm and 4:13pm with a minimum speed of 9 and a maximum speed of 17



Query:
SELECT longitude, latitude, speed FROM breadcrumb WHERE latitude BETWEEN 45.511002 AND 45.516153 AND longitude BETWEEN -122.705720 AND -122.692402 AND trip_id=243393633;

Visualization 2. All outbound trips that occurred on route 65 on any Friday (you choose which Friday) between the hours of 4pm and 6pm.

I added the compass because of the fact I rotated the map to make capturing the image easier.

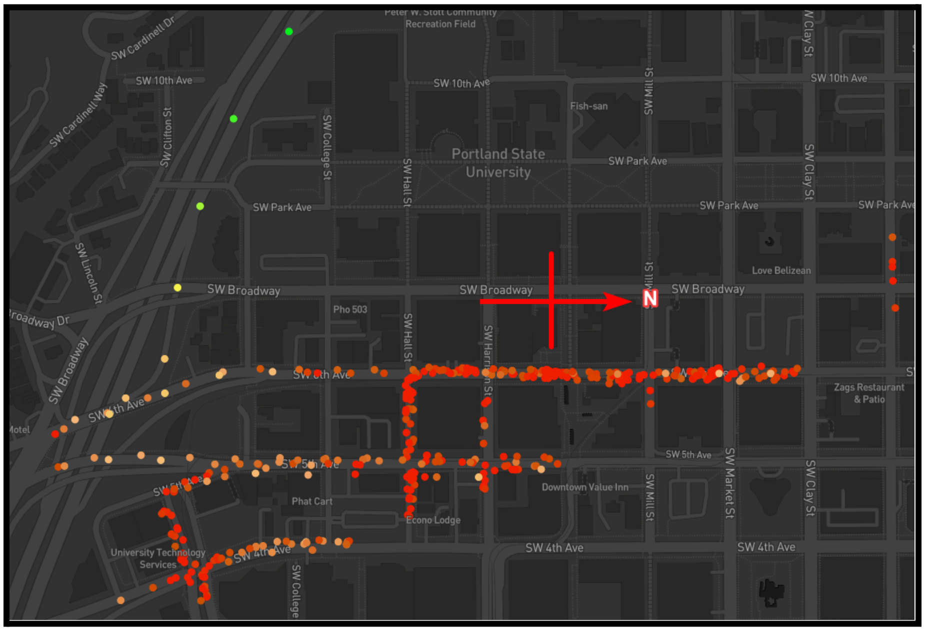


Query:

SELECT b.longitude, b.latitude, b.speed FROM breadcrumb b JOIN trip t ON b.trip_id = t.trip_id WHERE t.route_id = 65 AND b.tstamp >= '2023-01-20 16:00:00' AND b.tstamp < '2023-01-20 18:00:00' AND t.direction='Out';

Visualization 3. All trips that travel to and from PSU campus on any Sunday morning (you choose which Sunday) between 9am and 11am.

The query I am using is a bounding box around all of the PSU. I felt this would be the most appropriate way to represent the data, unfortunately though I'm not sure how to aggregate the data of all stops with a query that came through this bounding box.



Query:

```
SELECT b.longitude, b.latitude, b.speed FROM trip t JOIN breadcrumb b ON t.trip_id =
b.trip_id WHERE (b.latitude BETWEEN 45.507121 AND 45.514871) AND (b.longitude
BETWEEN -122.688456 AND -122.681560) AND b.tstamp >= '2023-01-22 09:00:00' AND
b.tstamp <= '2023-01-22 11:00:00';
```

Visualization 4. The longest (as measured by time) trip in your entire data set. Indicate the date, route #, and the trip ID of the trip along with a visualization showing the entire trip.

First I needed to find the longest trip, so I used the milestone 2 query I had and added a timestamp range onto it

```
SELECT trip_id, MAX(end_time - start_time) AS longest_trip_duration
FROM (
        SELECT trip_id, MIN(tstamp) AS start_time, MAX(tstamp) AS end_time
        FROM breadcrumb
        WHERE tstamp >= '2023-01-16' AND tstamp < '2023-01-24'
        GROUP BY trip_id
) AS trip_times
GROUP BY trip_id
ORDER BY longest_trip_duration DESC;
```

Which gave me the trip_id of ‘243309359’ with a duration of 03:27:05. I then used trip id to find out what the route was from the trip table. This particular route doesn’t have any data from the stop id api, so it’s route, day of week, and direction were all the defaults I selected.

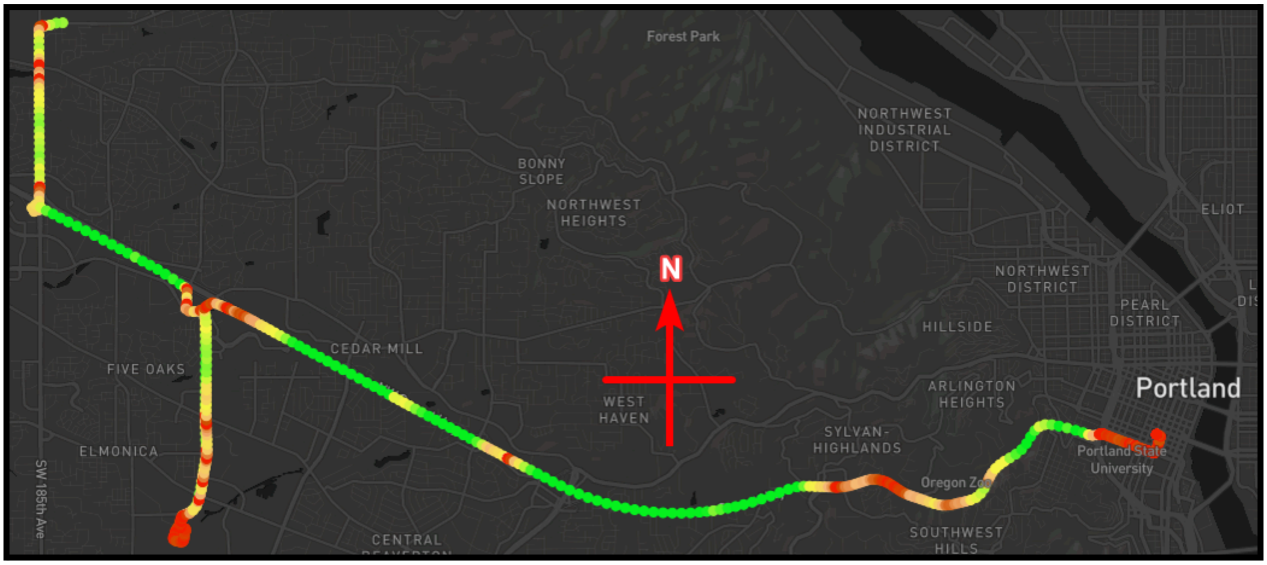
Finally, running this query:

```
SELECT * FROM breadcrumb WHERE trip_id=243309359;
```

netted me the run date of 01/23/2023. The day I’m actually writing this was the same day the data was collected and I triple double checked and this vehicle had *some* data, but not this specific trip_id.

Putting it all together was simple as using this query:

```
SELECT longitude, latitude, speed FROM breadcrumb WHERE trip_id=243309359;
```

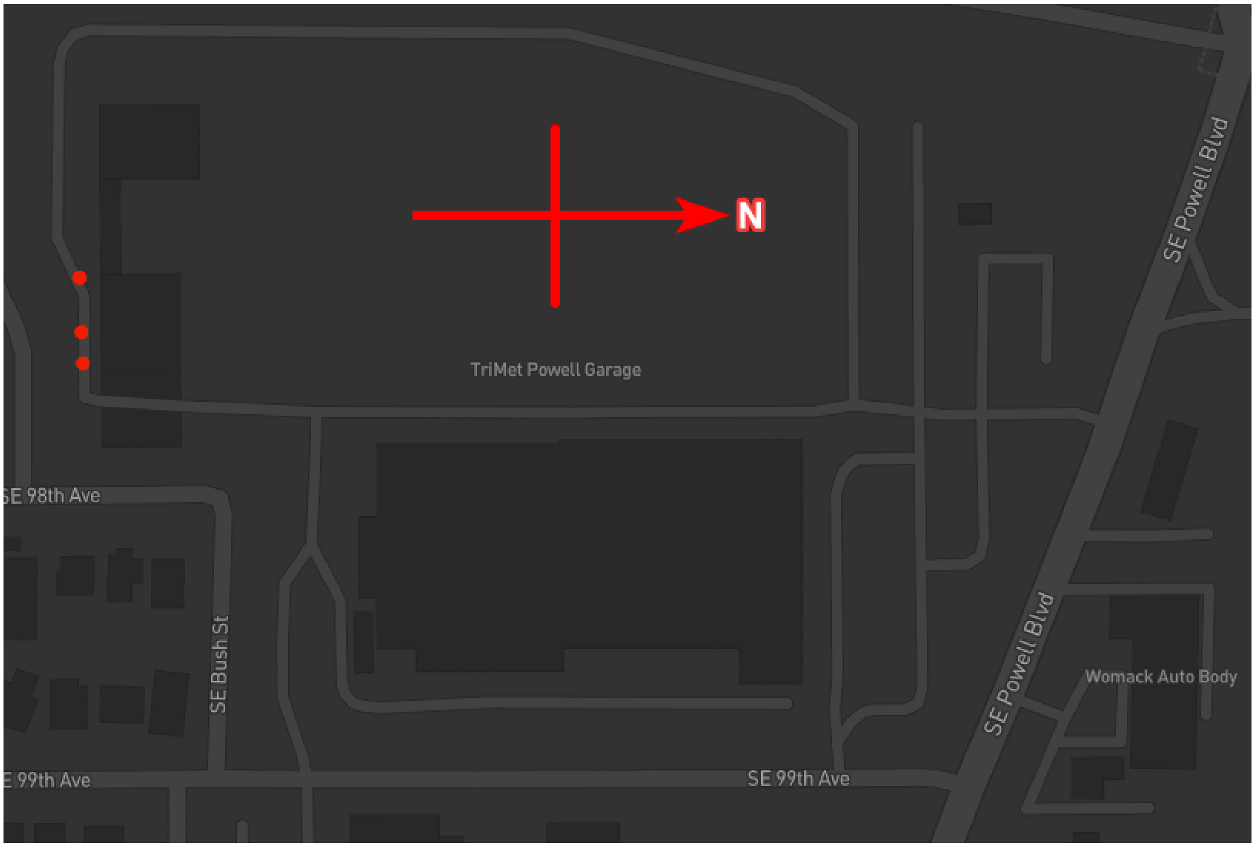


Visualization 5a, 5b, 5c, …. Three or more additional visualizations of your choice. Indicate why you chose each particular visualization.

Viz 5a: What was the shortest trip, because I'm curious to what that looks like. Using the longest trip query (in viz4), I ordered it by the length in ascending order, so the smallest was at the top. That gave me the trip_id of '241892771'. This is what it looks like. It seems as though recording might have been turned on for either testing reasons after repair, or by accident. As uninteresting as expected, but I still wanted to know.

Query:

SELECT longitude, latitude, speed FROM breadcrumb WHERE trip_id=241892771;



Viz 5b: Which trip on any day in the dataset was the busiest in the region around the airport. Identify that trip_id, day, and then display the data for that trip.
This one actually both is and isn't too surprising. This one trip_id has 811 records and takes place on 12/29/2022. What boggles the mind is how does it have 811 records? Lets display it and find out.

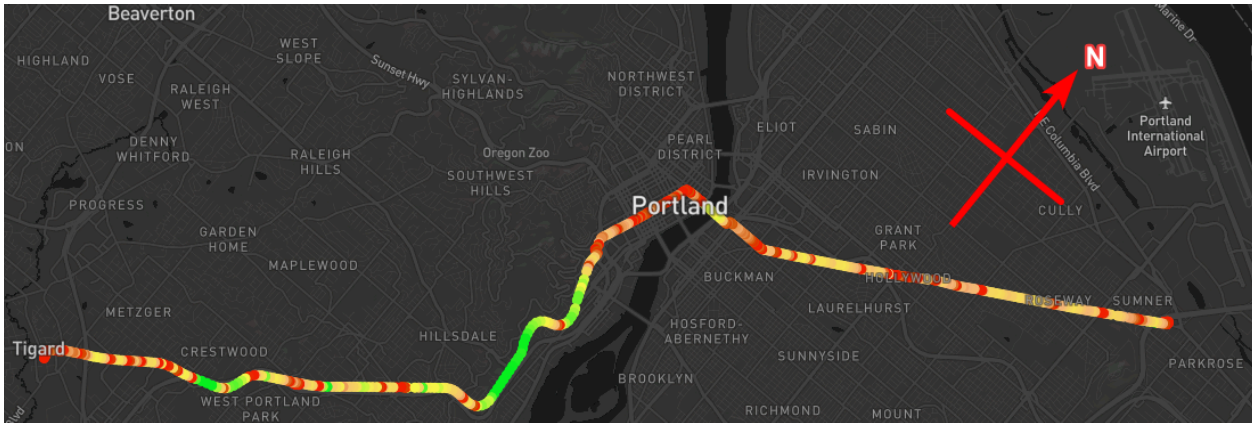
First the staging query:

```
SELECT trip_id, COUNT(*) AS num_trips, DATE_TRUNC('day', tstamp) AS trip_date
        FROM breadcrumb
        WHERE latitude BETWEEN 45.557301 AND 45.590433
        AND longitude BETWEEN -122.592736 AND -122.560334
        GROUP BY trip_id, trip_date
   ORDER BY num_trips DESC;
```

This gives the trip_id of '228310648'. Using the same simple query from 5a, we get this visualization.

Viz query:

```
SELECT longitude, latitude, speed FROM breadcrumb WHERE trip_id=228310648;
```



As it turns out, this starts at the BikeLink: Trimet Tigard lot and ends at the Parkrose / Sumner Transit Center. I've tried to take a better guess at why this is what it is, but it might just be one of the new long green buses that are driving around too.

Viz 5c: 01/22/2023 the Portland Trail Blazers had a home game against the LA Lakers. This was a Sunday and based on historical record extrapolation (because I couldn't find the actual record), home games on a Sunday typically start at 6pm. What is the shortest and longest trip that passed through there with a bounding box of [(45.526479, -122.661363), (45.537210, -122.671839)]. This value must be greater than 10 seconds. This essentially asks the question of how long was a trip within this bounding box, and what are those two trips which have the shortest trip in the bounding box, and the longest trip in the bounding box.

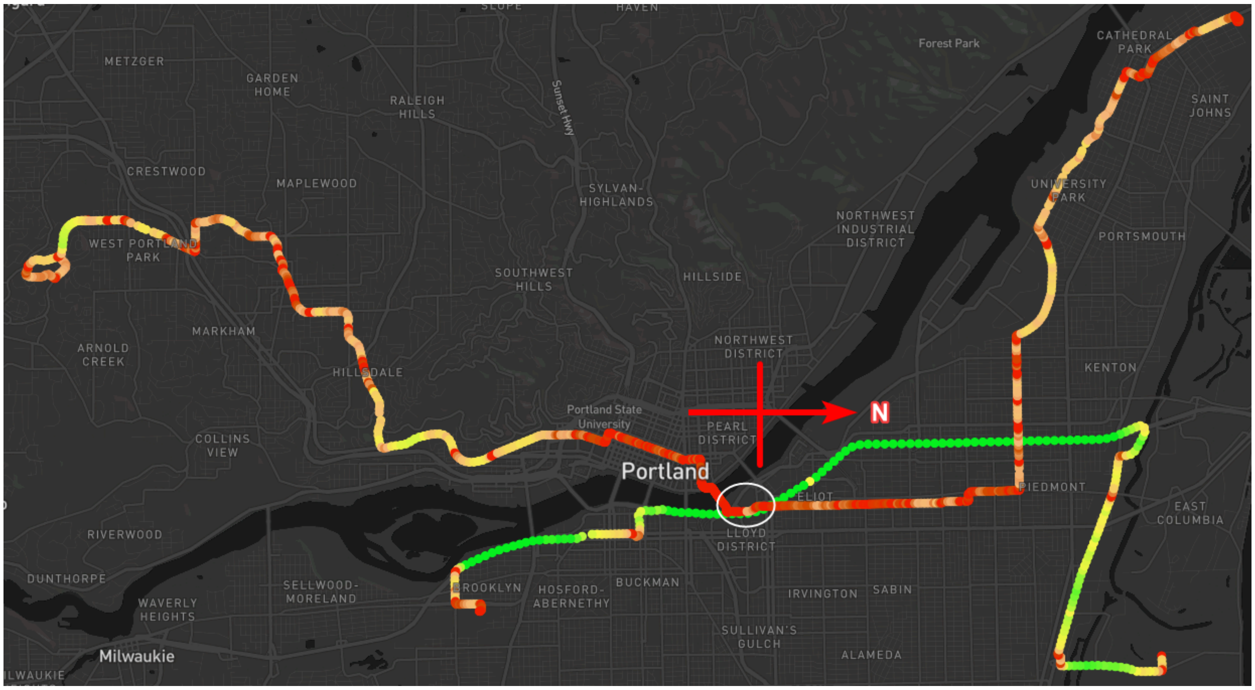
First I had to find the longest and shortest trips within the bounding box which resulted in two trip ids: 243004120with 45 seconds, and 243040258 with 962 seconds. These aren't entire trips, this is a singular trip_id event and how long it spent in the bounding box. I had the query start was early as 5pm but stop before 10pm.
Finding the shortest and longest trips is a simple single query which is ordered and allows me to jump from the top and bottom of the data with shift+G:

```
WITH trip_durations AS (
        SELECT trip_id, MIN(tstamp) AS start_time, MAX(tstamp) AS end_time,
EXTRACT(EPOCH FROM (MAX(tstamp) - MIN(tstamp))) AS duration_seconds
        FROM breadcrumb
        WHERE latitude BETWEEN 45.526479 AND 45.537210
        AND longitude BETWEEN -122.671839 AND -122.661363
        AND DATE_TRUNC('day', tstamp) = '2023-01-22'
        GROUP BY trip_id
)
SELECT trip_id, MIN(duration_seconds) AS trip_dur
FROM trip_durations
WHERE start_time::time >= '17:00:00' AND start_time::time <= '22:00:00'
GROUP BY trip_id
HAVING MIN(duration_seconds) > 10
ORDER BY trip_dur DESC;
```

Next I had to do a simple data collection of the two:

```
SELECT longitude, latitude, speed FROM breadcrumb WHERE trip_id=243004120 OR
trip_id=243040258;
```

Which produced the following visualization.



While it may not be immediately obvious based on the description above, you can see that the shortest trip_id was green and was able to go as fast as legally required, while the other was stuck behind traffic almost the entire time it was running. I've drawn a little white circle with the approximation of the bounding box I used. Both of these buses happen to be in the bounding box approximately the same time as well.

The northbound vehicle spent as little as 45 seconds in total within the bounding box while the other spent about 16 minutes in total within the bounding box. This one is pretty cool and I think tells you a lot about the preference of travel people had for the Portland Trail Blazers vs LA Lakers on 01/22/2023.

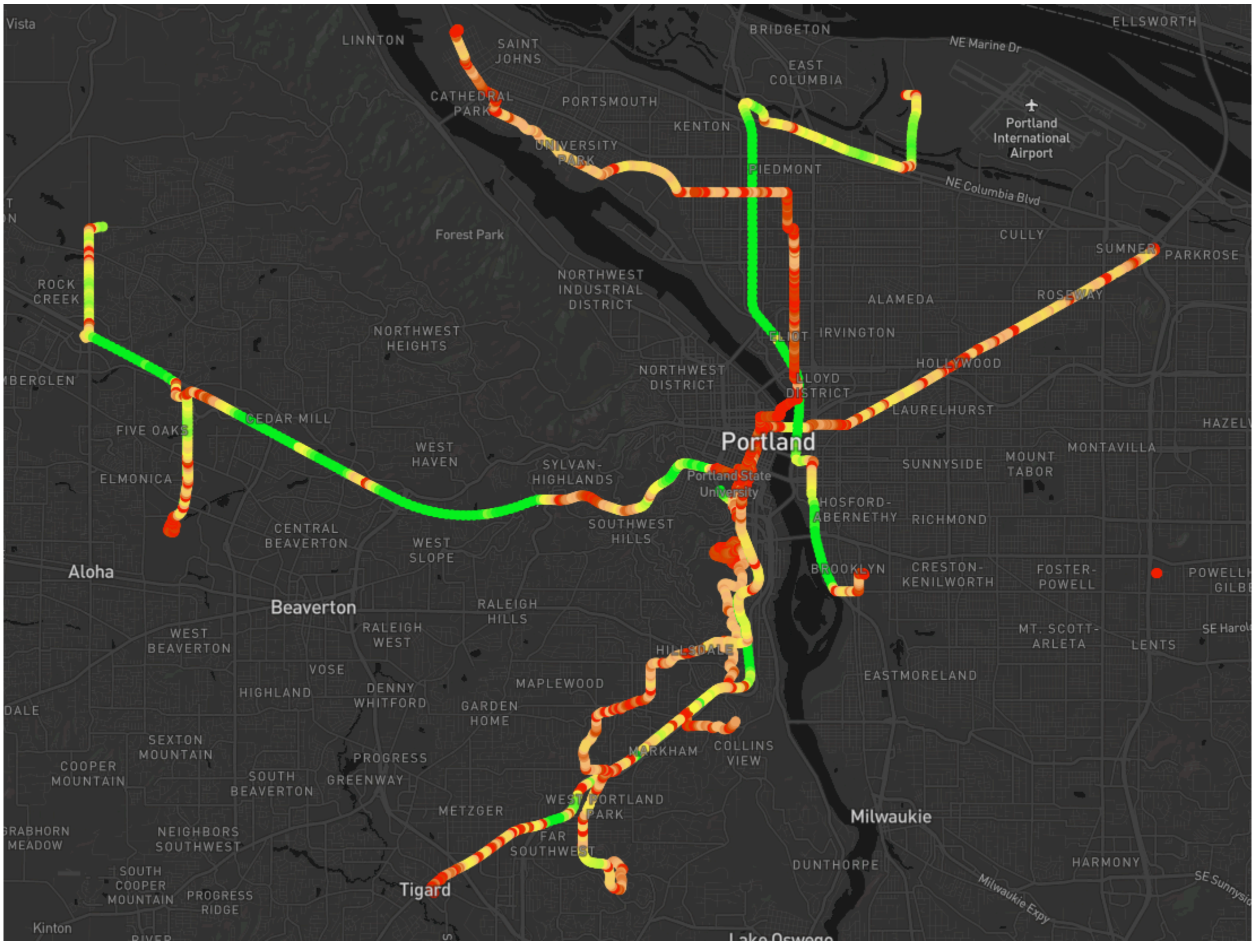
Your Code

Provide a reference to the repository where you store your python code. If you are keeping it private then share it with the Professor (rbi@pdx.edu or mina8@pdx.edu) and TA (vysali@pdx.edu).

Here is a link to my code: https://github.com/NAlexH2/de-proj-cs510
You can find milestone02 in `‘milestone-submissions/milestone03’`

There's one more image! Keep scrolling. It's just for fun more than anything.

Here’s one last bonus image. All the data I collected jumbled up and unlabeled. I realized I could just make my own long .geojson file with every record I get and show that too.


Vista
LINNTON
SAINT JOHNS
BRIDGETON
EAST COLUMBIA
ELLSWORTH
NE Marine Dr
CATHEDRAL PARK
PORTSMOUTH
KENTON
Portland International Airport
UNIVERSITY PARK
PIEDMONT
NE Columbia Blvd
Forest Park
CULLY
SUMNER
PARKROSE
ROCK CREEK
NORTHWEST INDUSTRIAL DISTRICT
ALAMEDA
ROSEWAY
NORTHWEST HEIGHTS
ELIOT
IRVINGTON
HOLLYWOOD
BERGLEN
NORTHWEST DISTRICT
LLOYD DISTRICT
FIVE OAKS
CEDAR MILL
LAURELHURST
Portland
HAZELW
WEST HAVEN
SYLVAN-HIGHLANDS
Portland State University
SUNNYSIDE
MOUNT TABOR
MONTAVILLA
ELMONICA
HOSFORD-ABERNETHY
RICHMOND
CENTRAL BEAVERTON
SOUTHWEST HILLS
WEST SLOPE
Aloha
BROOKLYN
CRESTON-KENILWORTH
FOSTER-POWELL
POWELL GILBE
Beaverton
RALEIGH HILLS
RALEIGH WEST
WEST BEAVERTON
MT. SCOTT-ARLETA
LENTS
SE Harol
HILLSDALE
VOSE
EASTMORELAND
HIGHLAND
DENNY WHITFORD
MAPLEWOOD
GARDEN HOME
DALE
SEXTON MOUNTAIN
MARKHAM
COLLINS VIEW
COOPER MOUNTAIN
PROGRESS
SOUTH BEAVERTON
GREENWAY
WEST PORTLAND PARK
METZGER
Milwaukie
GRABHORN MEADOW
NEIGHBORS SOUTHWEST
FAR SOUTHWEST
HARMONY
SOUTH COOPER MOUNTAIN
PROGRESS RIDGE
Tigard
DUNTHORPE
SE Sunnysid
Milwaukie Expy
Kinton
RIVER
Lake Oswego