DataEng S22: Project Assignment 3

Data Integration

Due date: May 22, 2022 @10pm PT

Submit at: https://forms.gle/9UKRF6aCvXGESjYt7

Congratulations! By now you have a working, end-to-end data pipeline. Unfortunately, it does not have enough data to properly implement our Data Scientist's visualization. To fill out information such as "route ID" you need to access another source of data and build a new pipeline to integrate it with your initial pipeline. Here are your steps:

A. access the stop event data
B. build a new pipeline for the stop event data
C. integrate the stop event data with the bread crumb data
D. testing

A. Stop Event Data

Access C-Tran "Stop Event" data at this URL: http://www.psudataeng.com:8000/getStopEvents As with the previous data source, this data set gives all C-Tran vehicle stop events for a single day of operation.

B. New Pipeline

Your job is to build a new pipeline that operates just like the previous one, including use of Kafka, automation, validation and loading.

C. Integrate Stop Events with Bread Crumbs

The two pipelines (BreadCrumb pipeline and StopEvent pipeline) must update the values in the Trip table such that all of the columns of both tables are filled correctly.

[5/20/2022] Alternatively, it would be OK to load the StopEvent data into a separate table and then use SQL views to integrate the two datasets.

D. Visualization

MapboxGL is a data visualization tool that allows you to view your bread crumb data and display it on a map. Your job is to integrate this tool with your database tables so that you can query the

breadcrumb and trip data in your database server, transform to geoJSON format and display the resulting map visualization. To get started, see this guide.

[5/20/2022] Alternatively, you may use an alternative visualization tool (such as folium) to create the required visualizations. We do not provide any guides for doing it, but you are free to do so if you prefer. The submitted visualizations must be equivalent or superior to the visualizations produced by the provided MapBoxGL based visualization tool.

Submission

Make a copy of this document and update it to include the following visualizations. For each visualization extract from your database a list of {latitude, longitude, speed} tuples and then use the provided visualization code (see Section D above) to display bus speeds at all of the corresponding geographic coordinates. So, for example, if you are asked to visualize a “trip”, then you must query your database to find all of the {latitude, longitude, speed} tuples for that trip, and then display a map showing the recorded/calculated bus speed at each {latitude,longitude} location.

No need to produce software that neatly displays trips, routes, dates, times, etc. onto the visualization itself. Instead, just paste a screen capture of the map-based speed visualization into your submission document and then include a text description of the contents of the visualization. For example, text like this: “Bus Speeds for all outbound trips of route 65 between 9am and 11am on Sunday October 32, 2020.”

Our database was stopped by GCP and we have raised a ticket for it. In the meantime, I created another database and used just today's data to do the visualization.

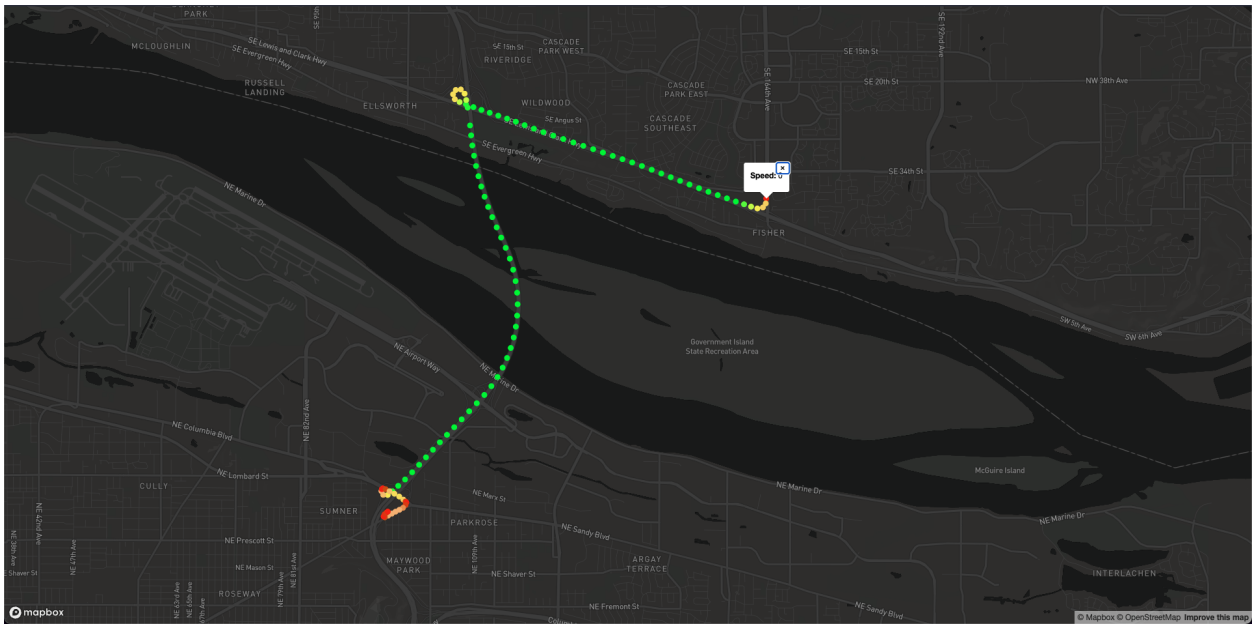
Visualization 1. A visualization of speeds for a single trip for any bus route that crosses the Glenn Jackson I-205 bridge. You choose the day, time and route for your selected trip. To find a trip that traverses this bridge, consider finding a trip that includes breadcrumb sensor points within this bounding box: [45.592404, -122.550711, 45.586158, -122.541270]. Any bus trip that includes breadcrumb points within that box either crosses the bridge or goes swimming in the Columbia river!

Query Used to find the trip:

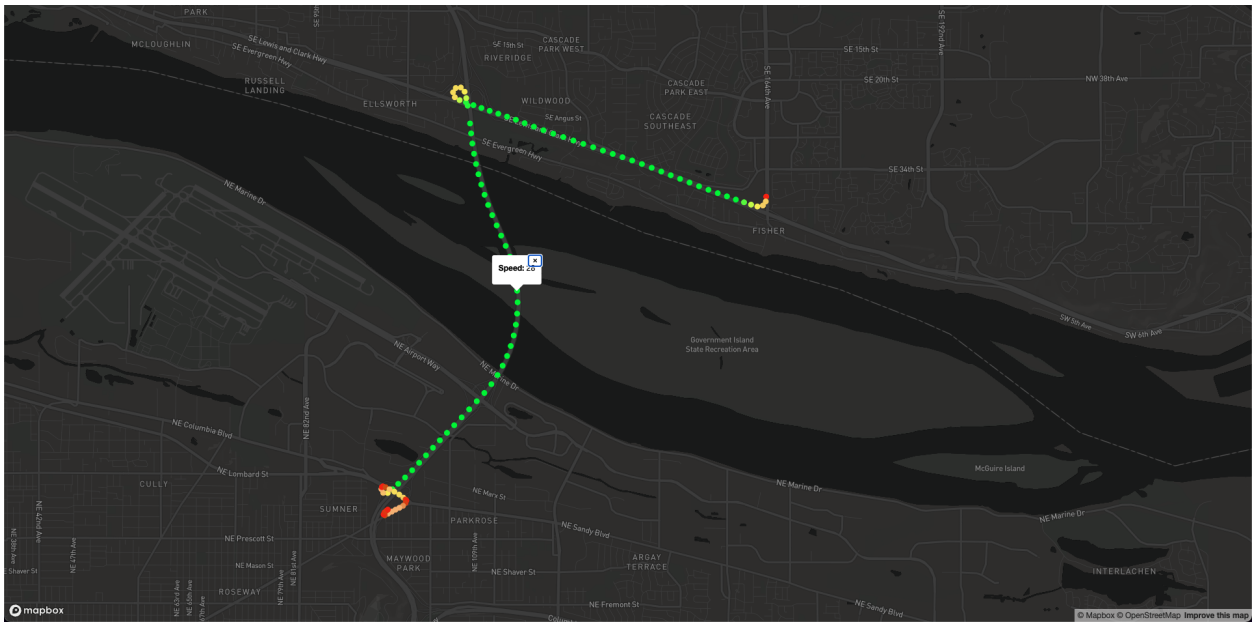
select * from breadcrumb where latitude >= 45.586158 and latitude <= 45.592404 and longitude <= -122.541270 and longitude >= -122.550711 limit 10;

select * from breadcrumb where trip_id = 171131323;

Trip Id : 171131323



Bus speeds Trip Id 171131323 (24th October 2020)



Bus speeds for Trip Id 171131323 (24th October 2020)

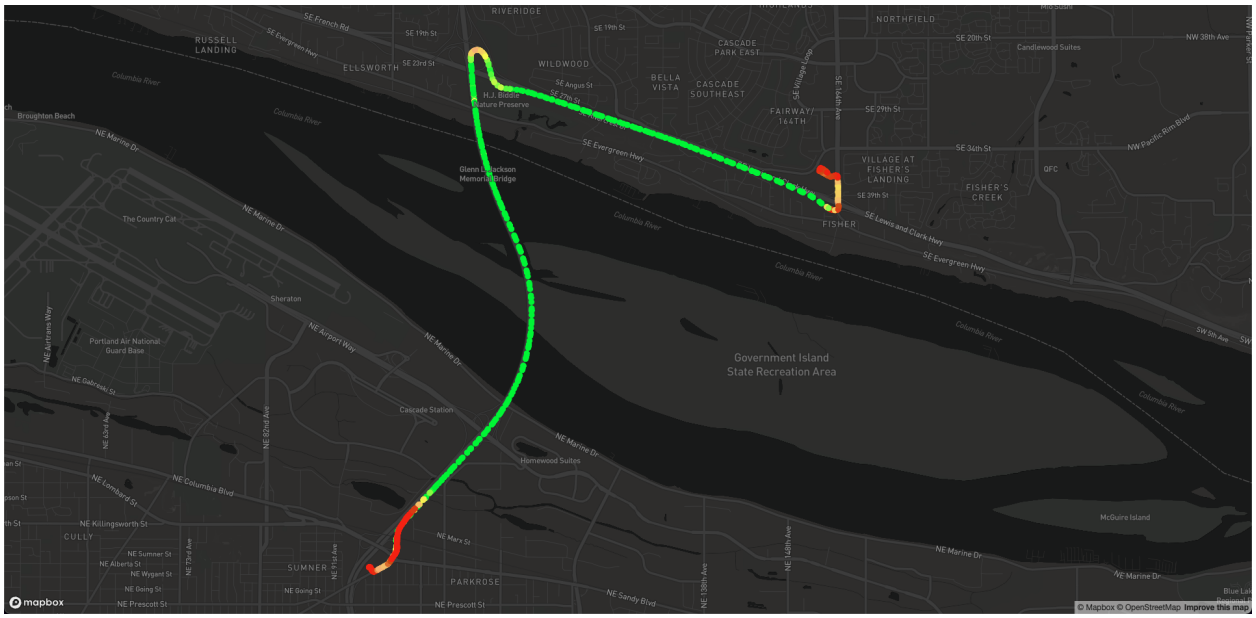
Visualization 2. All outbound trips that occurred on route 65 on any Friday (you choose which Friday) between the hours of 4pm and 6pm.

Query Used (Sunday Data Visualization)

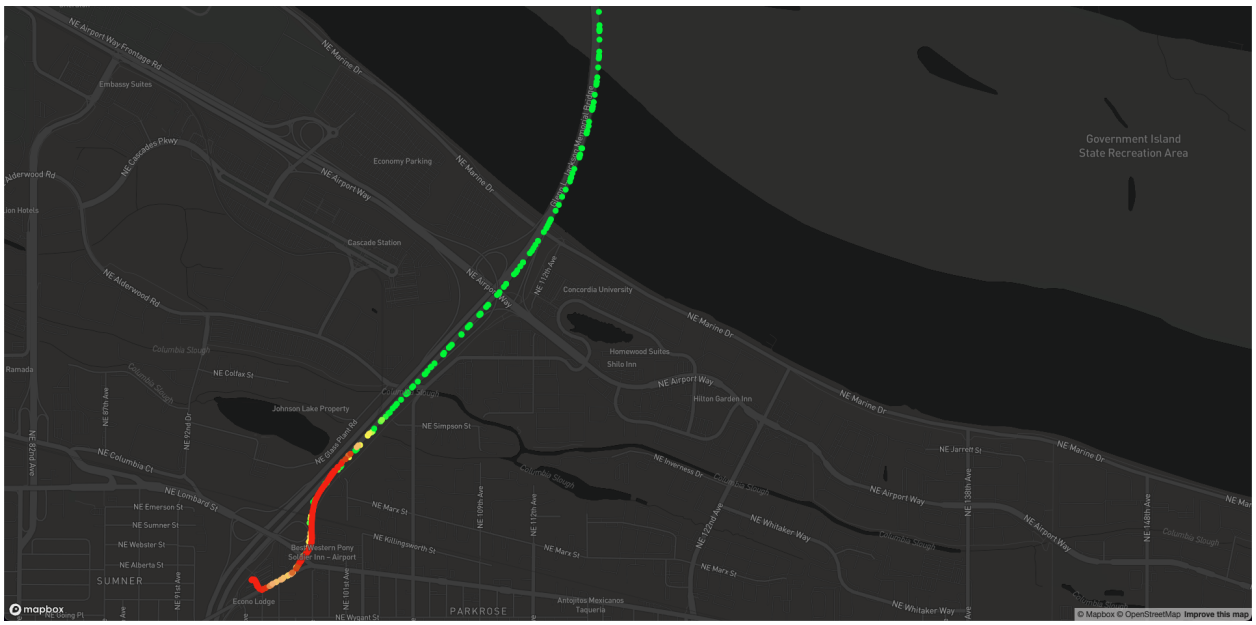
Route 65
Time 12pm to 2pm

Direction Out (0)

select b.longitude, b.latitude, b.speed from breadcrumb b, trimet_trips tt where extract(isodow from b.tstamp) = 7 and extract(hour from b.tstamp) >= 12
and extract(hour from b.tstamp) <= 14 and b.trip_id = tt.trip_id and tt.route_id = 65 and tt.direction = 0 order by b.tstamp asc;



Bus speeds for trips on route 65 on a Sunday (25th October) (no Friday data because of the reason explained above) between 12 pm and 2 pm for outgoing trips (Busy Time)



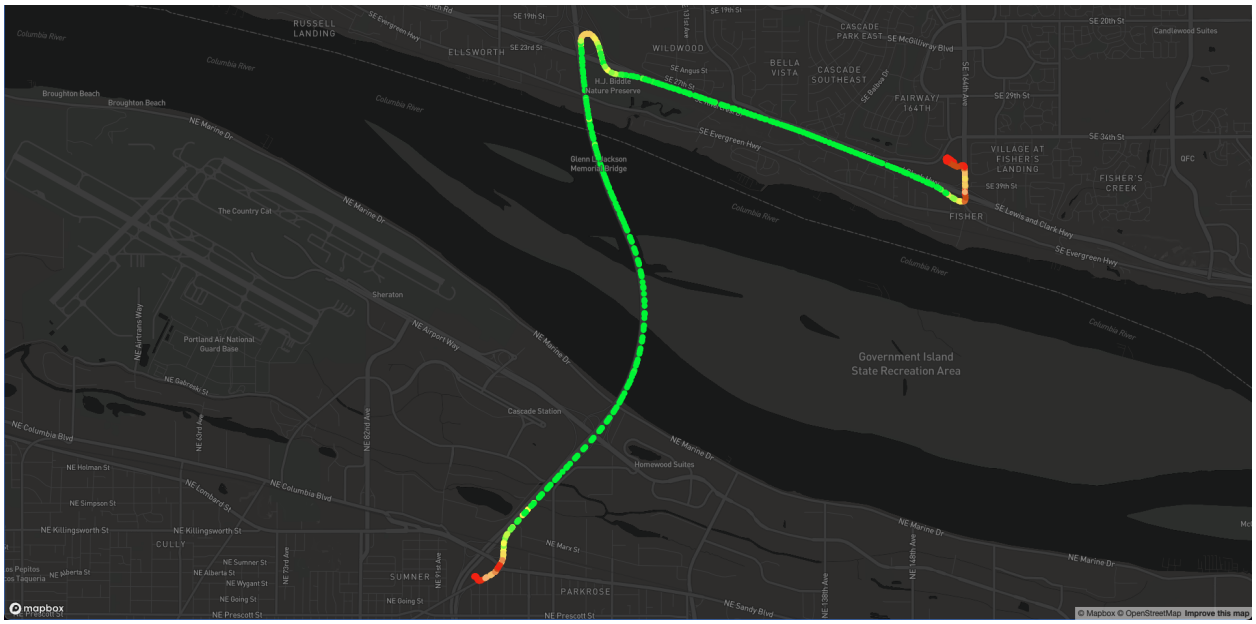
Bus speeds for trips on route 65 on a Sunday (25th October) (no Friday data because of the reason explained above) between 12 pm and 2 pm for outgoing trips (Busy Time)

Visualization 3. All outbound trips for route 65 on any Sunday morning (you choose which Sunday) between 9am and 11am.

Query Used (Sunday Data Visualization)

Route 65
Time 9am to 11am
Direction Out (0)

select b.longitude, b.latitude, b.speed from breadcrumb b, trimet_trips tt where extract(isodow from b.tstamp) = 7 and extract(hour from b.tstamp) >= 9
and extract(hour from b.tstamp) <= 11 and b.trip_id = tt.trip_id and tt.route_id = 65 and tt.direction = 0 order by b.tstamp asc;



Bus speeds for trips on route 65 on a Sunday(25th October) between 9 am and 11 am for outgoing trips (Not So Busy)

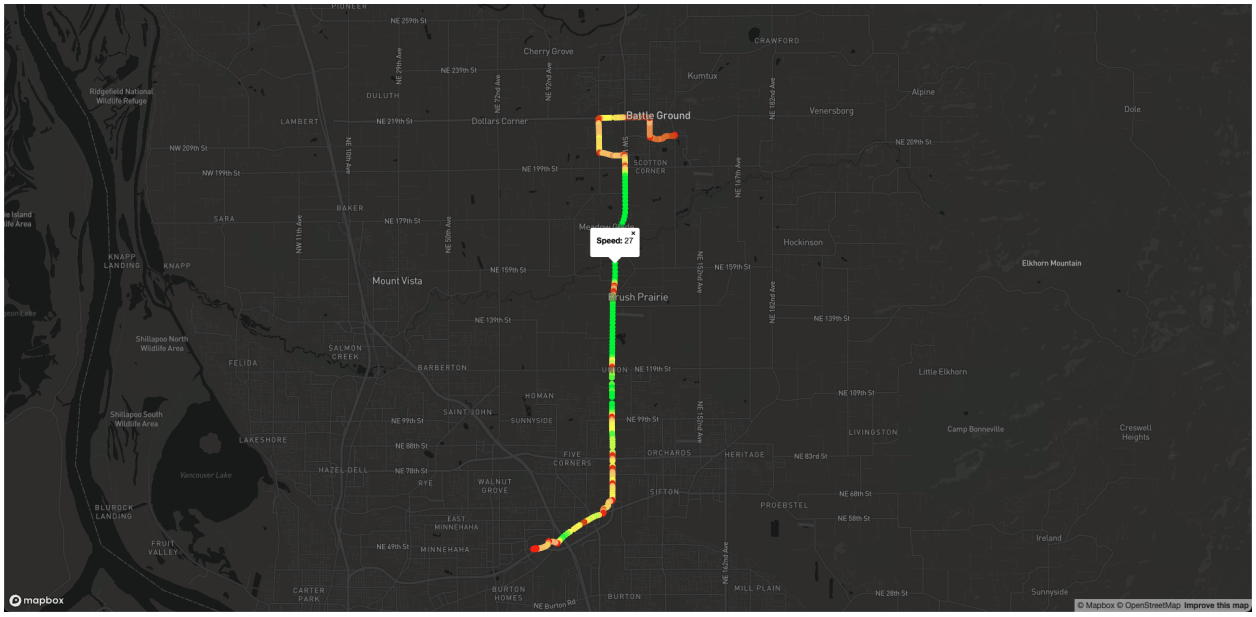
Bus speeds for trips on route 65 on a Sunday(25th October) between 9 am and 11 am for outgoing trips (Not So Busy)

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

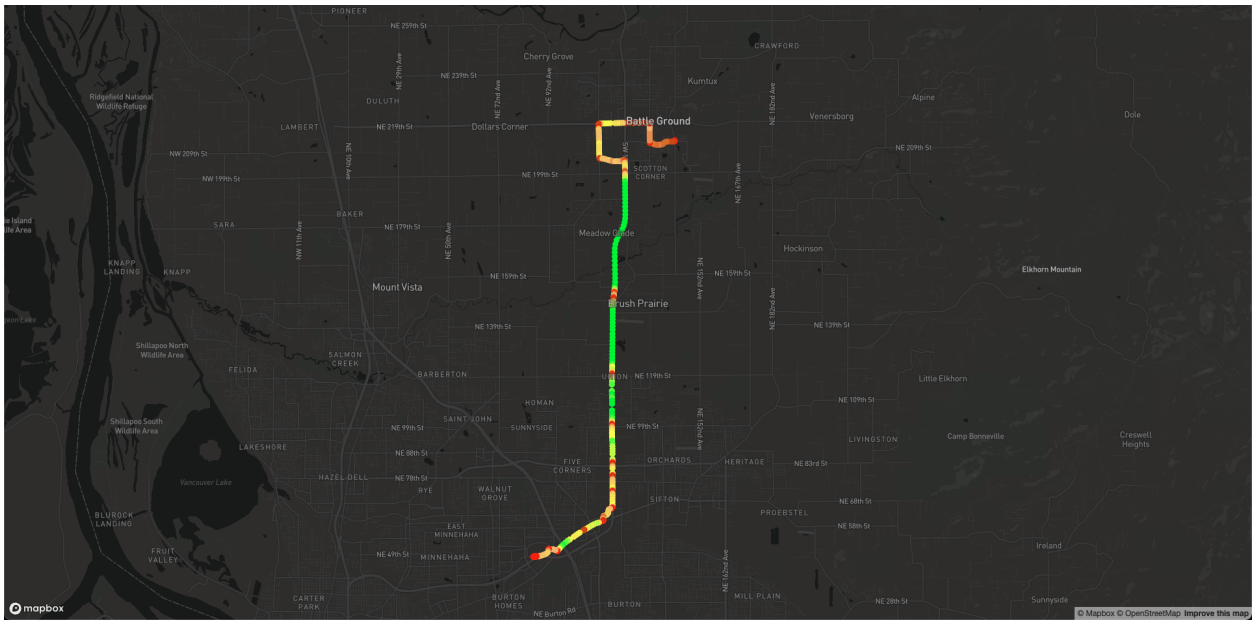
Date : 2020-10-25
Trip Id : 171152616
Route Id : 7

Query Used to Find the Longest Trip

select trip_id, max(tstamp) - min(tstamp) as duration from breadcrumb group by trip_id order by duration desc limit 2;



Bus speeds for Trip Id 171152616 (which is the longest trip) on October 25th 2020.



Bus speeds for Trip Id 171152616 (which is the longest trip) on October 25th 2020.

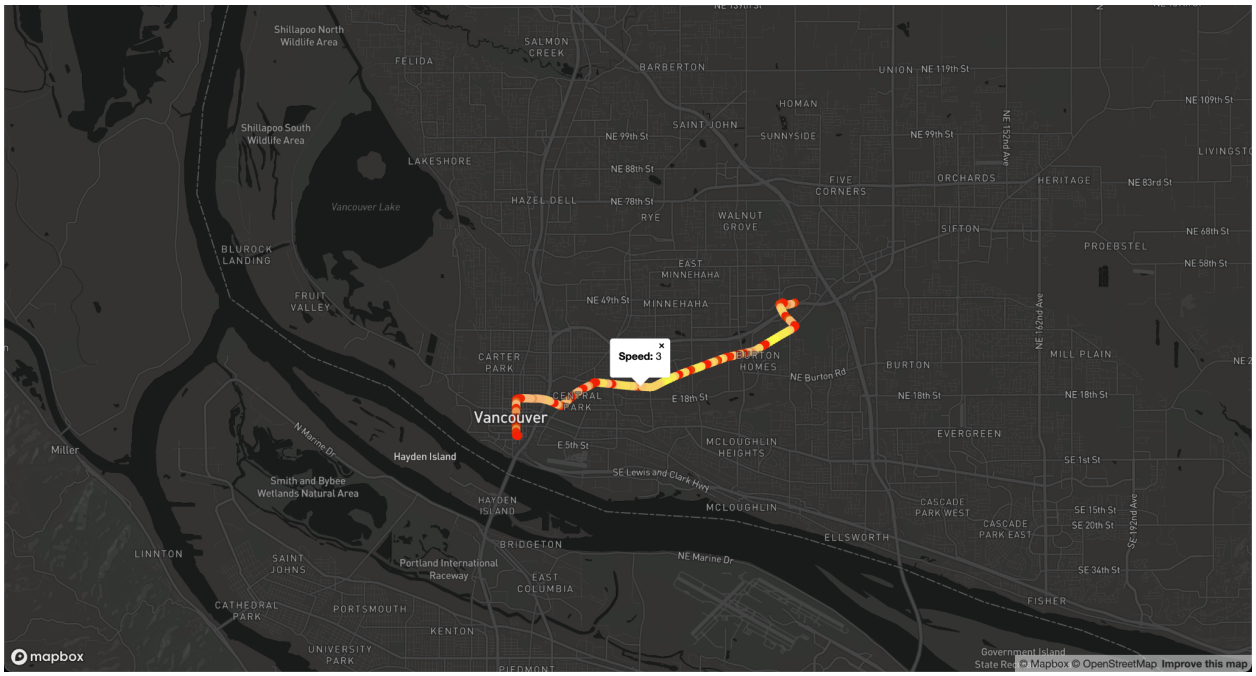
Visualization 5a, 5b, 5c, …. Three or more additional visualizations of your choice. Indicate why you chose each particular visualization.

5a. Visualize the route with the most number of trips.

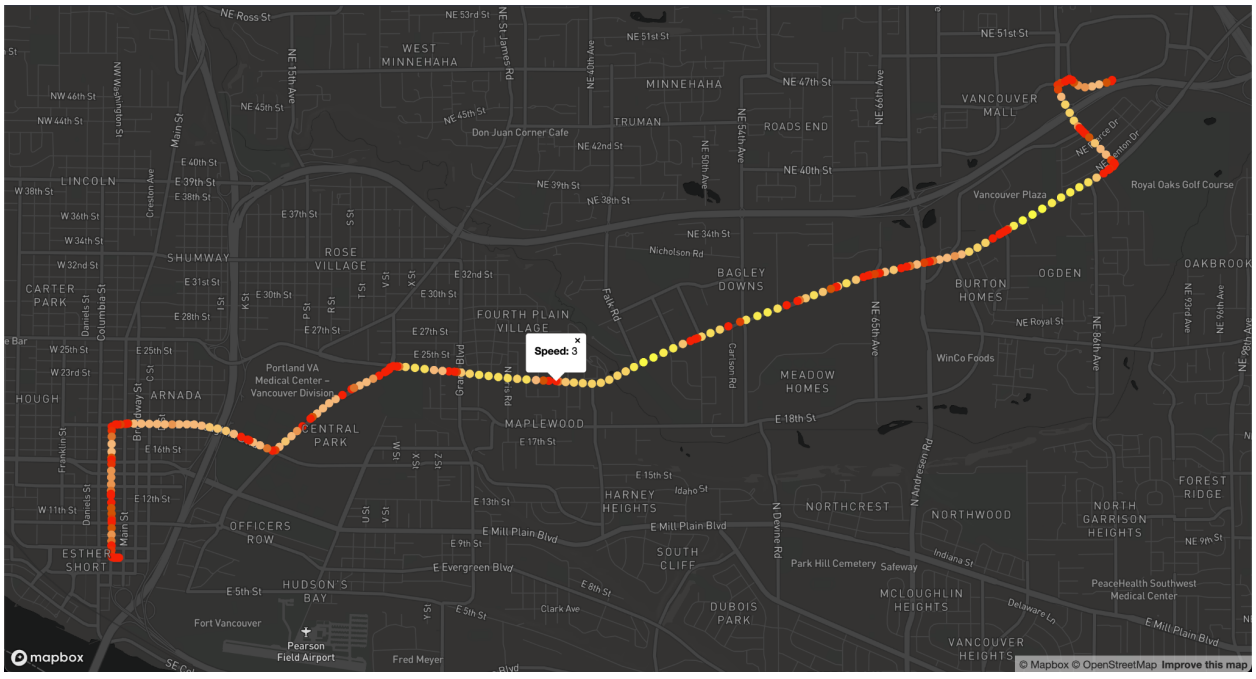
I wanted to visualize this because I wanted to see how far/near is the route to downtown which has the most number of trips. And as per the visualization, I can see that it goes right into downtown.

Trip Id : 171157278
Route 50
The number of trips on Sunday : 259 (up and down included)
Query to get trip count for routes

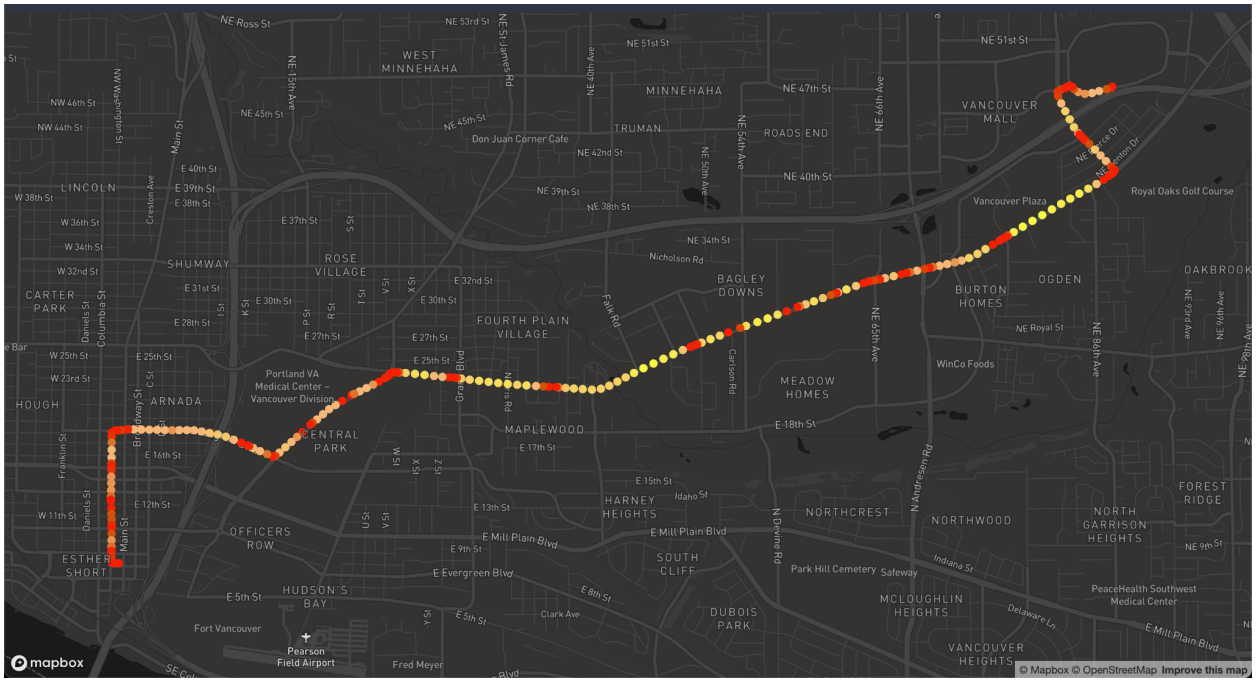
select route_id, count(*) total from trimet_trips group by route_id order by total desc;



Bus speeds for Trip 171157278 on Route 50 on October 25, 2022 (Sunday) (Most active route)



Bus speeds for Trip 171157278 on Route 50 on October 25, 2022 (Sunday) (Most active route)



Bus speeds for Trip 171157278 on Route 50 on October 25, 2022 (Sunday) (Most active route)

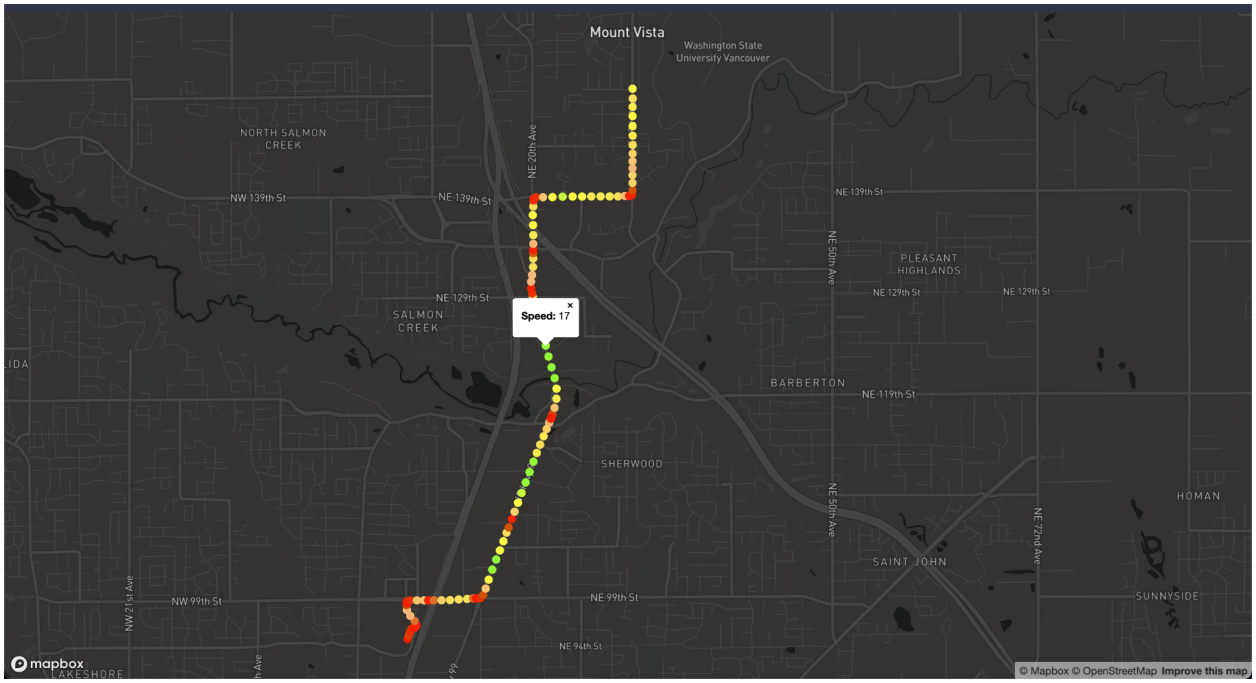
5b. Visualize the route with the least number of trips

Just like 5a, I wanted to visualize this to understand if the trip count had anything to do with the route being far away from the city center. And as per the visualization, it is!

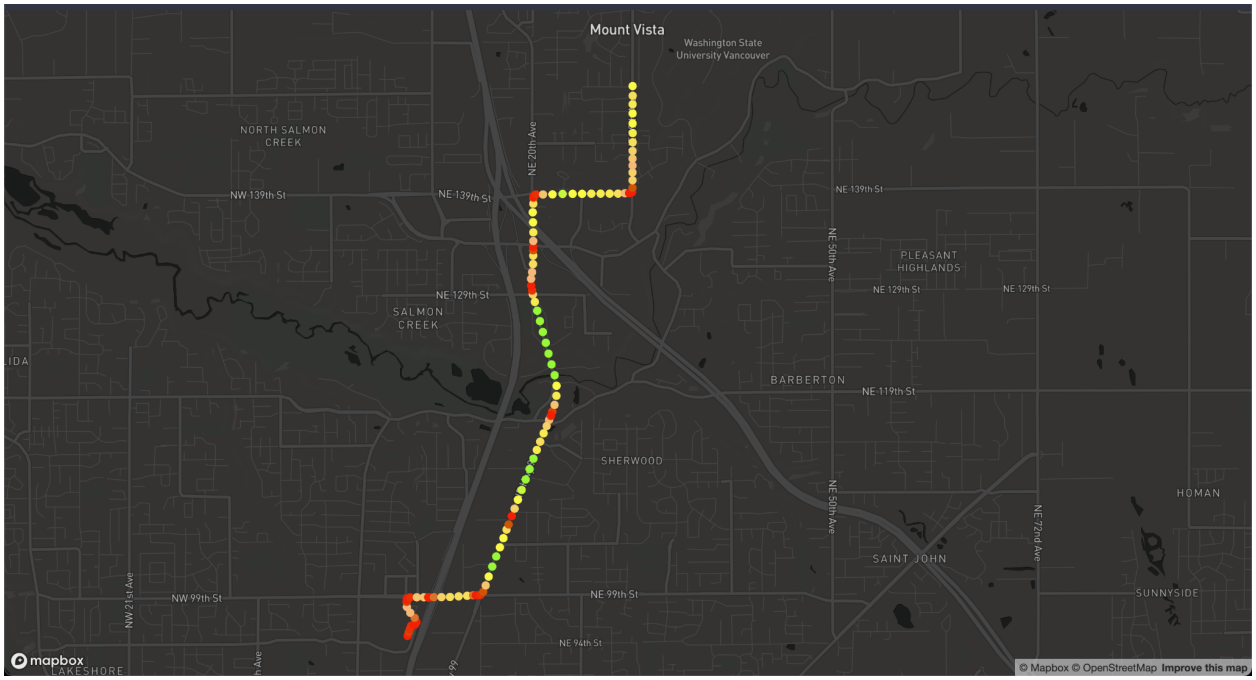
Trip Id : 171180372
Route 19
The number of trips on Sunday : 37 (up and down included)

Query to get trip count for routes :
select route_id, count(*) total from trimet_trips group by route_id order by total desc;

Bus speeds for Trip 171180372 on Route 19 on October 25, 2022 (Sunday) (Least active route)



Bus speeds for Trip 171180372 on Route 19 on October 25, 2022 (Sunday) (Least active route)



Bus speeds for Trip 171180372 on Route 19 on October 25, 2022 (Sunday) (Least active route)

5c. The route taken in a trip which has the highest speed

We wanted to visualize the route of trip which has the highest number of instances where the speed has crossed 90% of the max_speed.

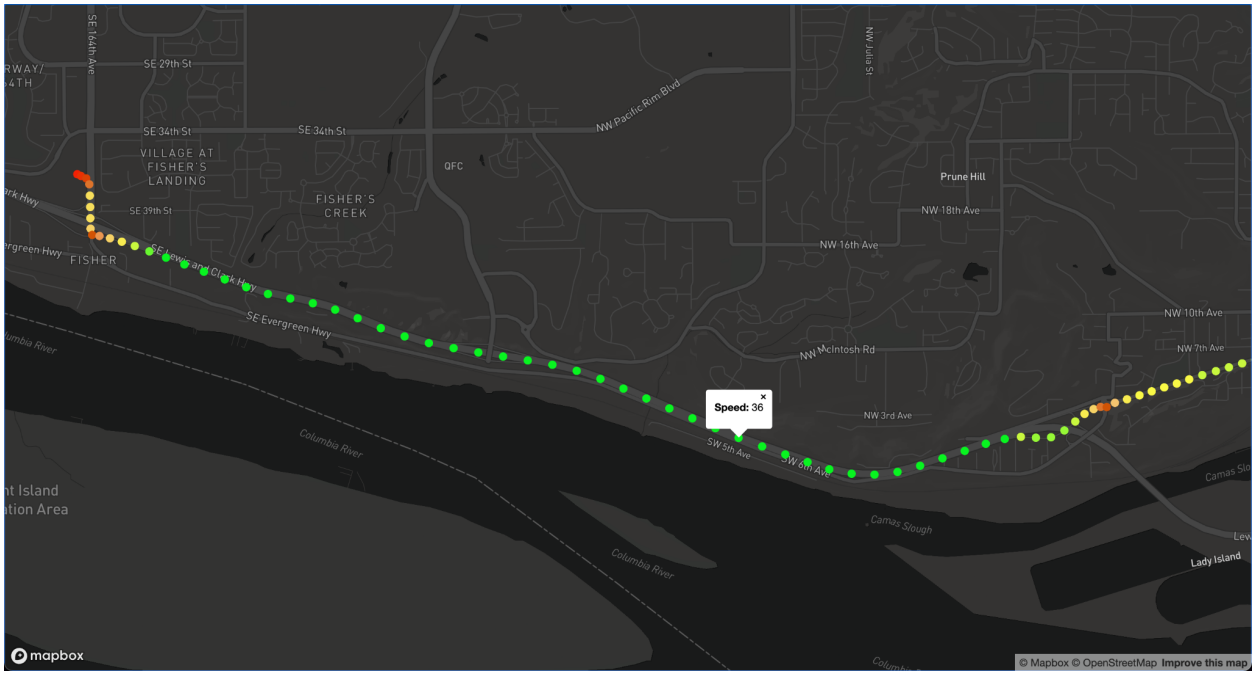
So basically, the trip which has a lot of indications of speeding.

To get the breadcrumb data we used the following query:

```
with trip_max as (
with max_speed as (select max(speed) as mspeed from breadcrumb)
select trip_id, count(*) as ct from breadcrumb, max_speed where speed > mspeed * 90/100 and
speed <= mspeed group by trip_id order by ct desc limit 1)
select b.longitude, b.latitude, b.speed from breadcrumb b, trip_max where b.trip_id =
trip_max.trip_id;
```

Trip Id 171178318

Bus speed visualization for a trip wth the highest number of near max speeds.



Bus speed visualization for a trip wth the highest number of near max speeds.

Your Code

Provide a reference to the repository where you store your code. If you are keeping it private then share it with Bruce and Genevieve.

The repo is at https://github.com/prabhumarappan/cs510-data-engineering/
The code for this assignment is at
https://github.com/prabhumarappan/cs510-data-engineering/tree/main/project-assignment3