DataEng: Project Assignment 3

Data Integration

Assignment date: February 16
Due date: February 28, 2021 @10pm PT
Submit: assignment submission form

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://rbi.ddns.net/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.

D. Visualization

Aman developed a new visualization tool that allows you to view your bread crumb data and display it on a map. See Aman's description here. 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.

Submission

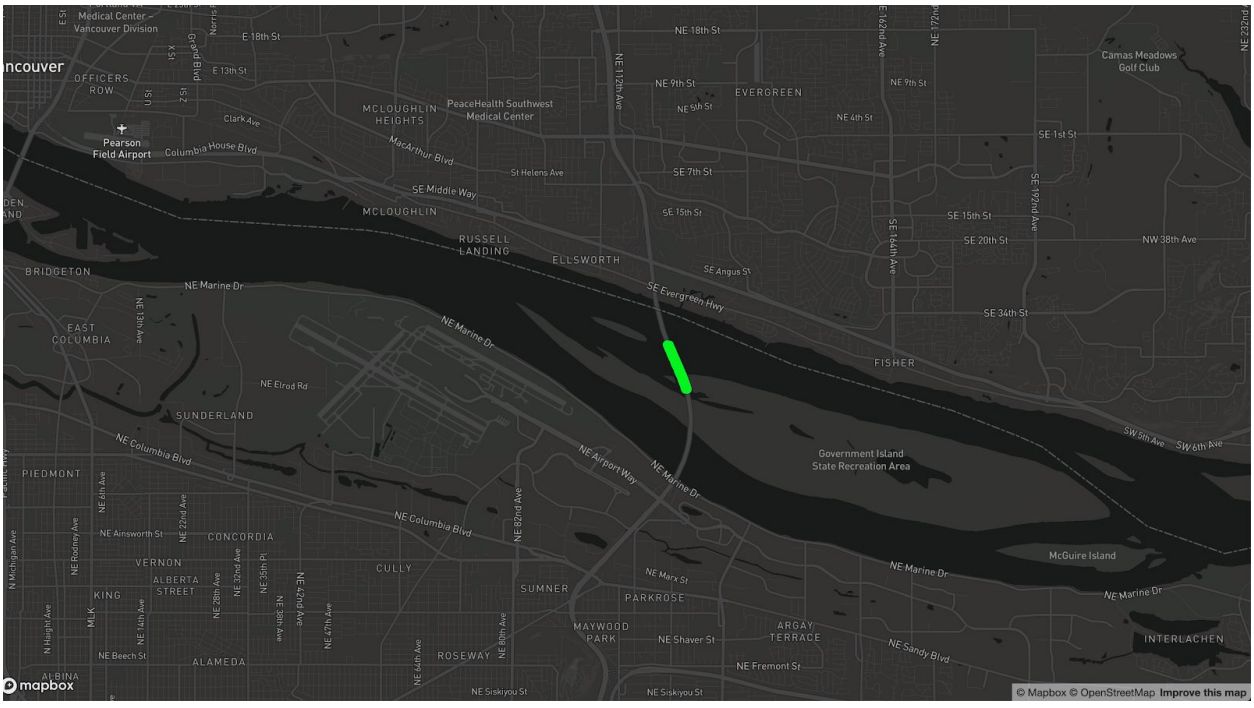
Make a copy of this document and update it to include the following visualizations. For each visualization extract from your database a list of {latitude, longitude, speed} tuples and then use the provided visualization code (see Section D above) to display bus speeds at all of the corresponding geographic 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.”

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!

```
SELECT *
FROM (SELECT A.tstamp, A.latitude, A.longitude, A.speed, B.route_id
    FROM breadcrumb AS A JOIN trip AS B
    ON A.trip_id = B.trip_id) AS C
WHERE tstamp::date = date '2020-10-17'
AND (latitude < 45.592404 AND longitude > -122.550711)
AND (latitude > 45.586158 AND longitude < -122.541270)
AND route_id = 65
```

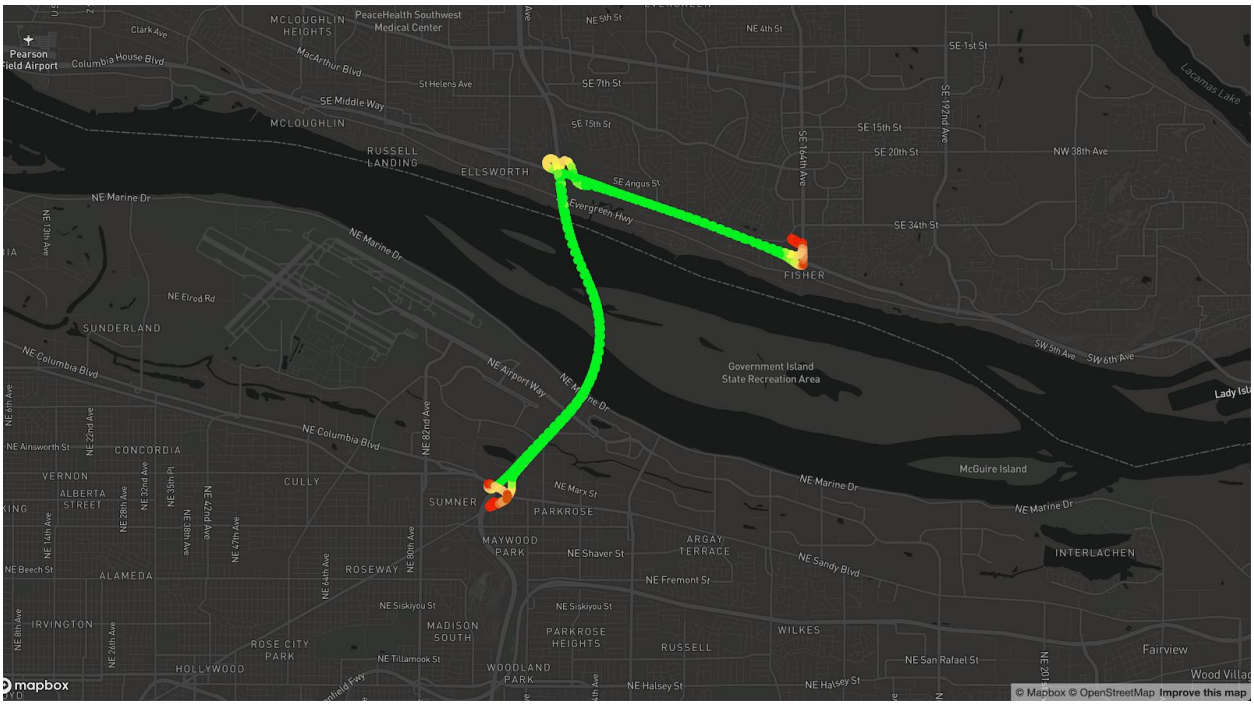
tstamp	latitude	longitude	speed	route_id
2020-10-17 08:44:51	45.587188	-122.544763	27	65
2020-10-17 08:35:27	45.592025	-122.54811	25	65
2020-10-17 08:35:32	45.590973	-122.547475	25	65
2020-10-17 08:35:37	45.589908	-122.546813	25	65
2020-10-17 08:35:47	45.58769	-122.545475	24	65
2020-10-17 08:35:52	45.586523	-122.544877	27	65
2020-10-17 09:05:11	45.592345	-122.54829	27	65
2020-10-17 09:05:16	45.591193	-122.547598	27	65
2020-10-17 09:05:21	45.590028	-122.546885	27	65
2020-10-17 09:05:26	45.58886	-122.546162	27	65
2020-10-17 09:05:31	45.58766	-122.54545	28	65
2020-10-17 09:05:36	45.586427	-122.544832	28	65
2020-10-17 09:15:01	45.586267	-122.544347	24	65
2020-10-17 09:15:06	45.58743	-122.544902	26	65
2020-10-17 09:15:11	45.588552	-122.545543	29	65



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

```
SELECT  *
FROM (SELECT A.trip_id,A.tstamp, A.latitude, A.longitude, A.speed, B.route_id, B.direction
    FROM breadcrumb AS A JOIN trip AS B
    ON A.trip_id = B.trip_id) AS C
WHERE tstamp::date = date '2020-10-16'
AND route_id = 65 AND direction='Out'
AND tstamp::time > '18:00:00' AND tstamp::time < '20:00:00'
```

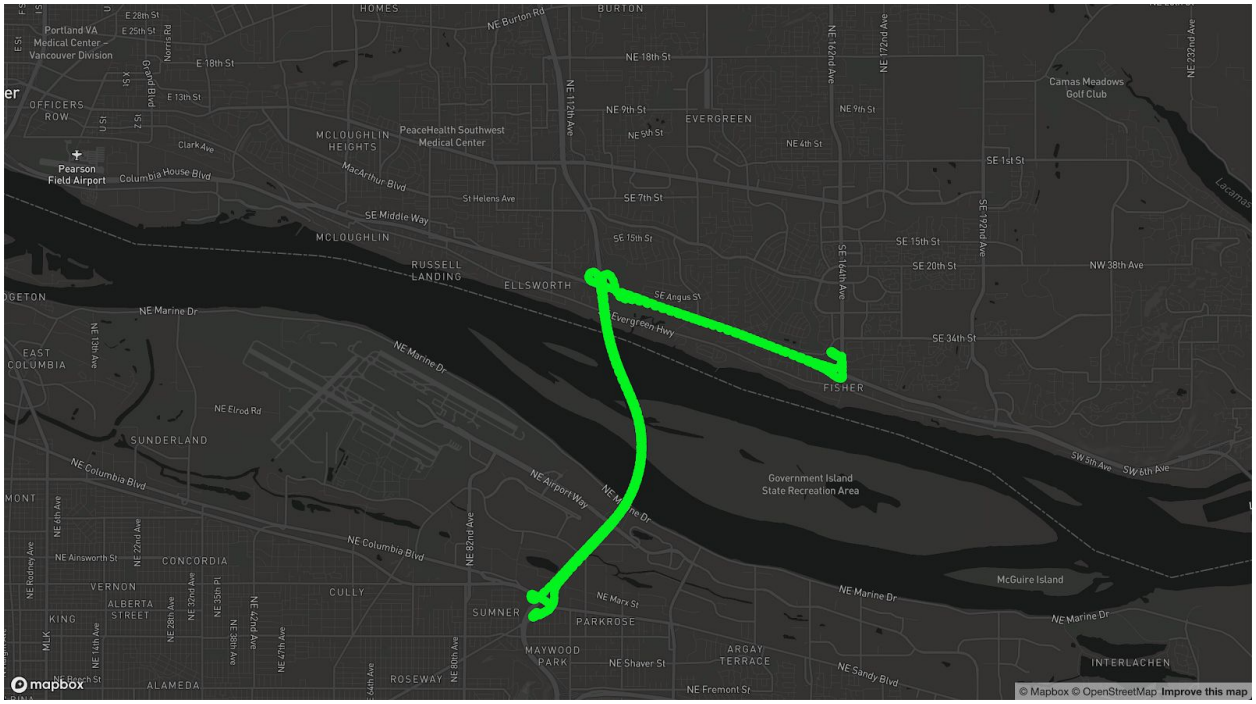
trip_id	tstamp	latitude	longitude	speed	route_id	direction
170556839	2020-10-16 18:32:22	45.598683	-122.52976	25	65	Out
170556839	2020-10-16 18:30:32	45.594195	-122.50433	0	65	Out
170556839	2020-10-16 18:30:37	45.59415	-122.504142	3	65	Out
170556839	2020-10-16 18:30:42	45.593803	-122.504065	7	65	Out
170556839	2020-10-16 18:30:47	45.5933	-122.50408	11	65	Out
170556839	2020-10-16 18:30:52	45.592792	-122.504348	12	65	Out
170556839	2020-10-16 18:30:57	45.592535	-122.505145	14	65	Out
170556839	2020-10-16 18:31:02	45.592683	-122.506128	15	65	Out
170556839	2020-10-16 18:31:07	45.592925	-122.507205	17	65	Out
170556839	2020-10-16 18:31:12	45.593195	-122.508423	20	65	Out
170556839	2020-10-16 18:31:17	45.593517	-122.509723	21	65	Out
170556839	2020-10-16 18:31:22	45.593847	-122.511125	23	65	Out
170556839	2020-10-16 18:31:27	45.594213	-122.512583	24	65	Out
170556839	2020-10-16 18:31:32	45.594602	-122.514085	24	65	Out
170556839	2020-10-16 18:31:37	45.595008	-122.51559	25	65	Out



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

```
SELECT *
FROM (SELECT A.tstamp, A.latitude, A.longitude, A.speed, B.route_id, B.direction
    FROM breadcrumb AS A JOIN trip AS B
    ON A.trip_id = B.trip_id) AS C
WHERE tstamp::date = date '2020-10-18'
AND route_id = 65 AND direction='Out'
AND tstamp::time > '09:00:00' AND tstamp::time < '11:00:00'
```

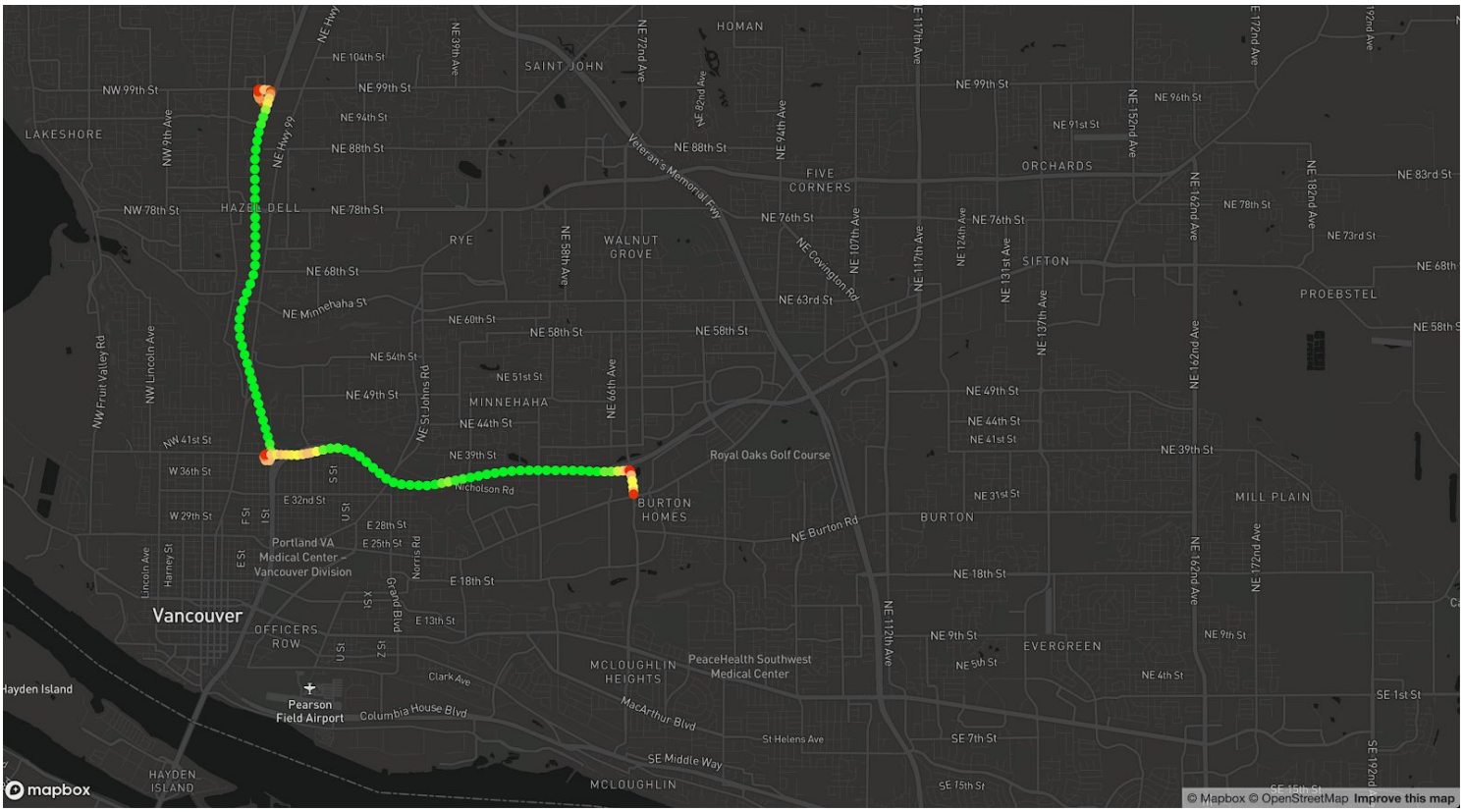
tstamp	latitude	longitude	speed	route_id	direction
2020-10-18 09:02:58	45.601627	-122.541828	26	65	Out
2020-10-18 09:00:13	45.594485	-122.5051	0	65	Out
2020-10-18 09:00:18	45.594468	-122.504995	1	65	Out
2020-10-18 09:00:23	45.59441	-122.504867	2	65	Out
2020-10-18 09:00:28	45.59431	-122.504675	3	65	Out
2020-10-18 09:00:33	45.594212	-122.504347	5	65	Out
2020-10-18 09:00:37	45.59413	-122.504115	5	65	Out
2020-10-18 09:00:43	45.593672	-122.504063	9	65	Out
2020-10-18 09:00:48	45.593102	-122.504133	12	65	Out
2020-10-18 09:00:53	45.592647	-122.504643	13	65	Out



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.

```
SELECT *
FROM breadcrumb AS A JOIN trip AS B
ON A.trip_id = B.trip_id
WHERE A.trip_id = (
    SELECT trip_id
    FROM breadcrumb
    GROUP BY trip_id
    ORDER BY (max(tstamp) - min(tstamp)) desc
    LIMIT 1
)
```

tstamp	latitude	longitude	direction	speed	trip_id	trip_id	route_id	vehicle_id	service_key	direction
2020-10-16 23:59:22	45.691092	-122.663318	0	0	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:05:33	45.650345	-122.649503	101	20	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:05:38	45.65003	-122.64824	110	20	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:05:43	45.649522	-122.647063	122	21	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:05:48	45.648867	-122.645942	130	22	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:05:53	45.648183	-122.644817	131	23	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:05:58	45.64751	-122.6437	131	22	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:06:03	45.646895	-122.642613	129	21	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:06:08	45.646452	-122.64137	117	21	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:06:13	45.646217	-122.640013	104	21	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:06:18	45.646148	-122.638573	94	22	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:06:23	45.646097	-122.63712	93	20	170528709	170528709	71	4036	Weekday	Out
2020-10-16 00:06:28	45.646075	-122.63571	91	21	170528709	170528709	71	4036	Weekday	Out



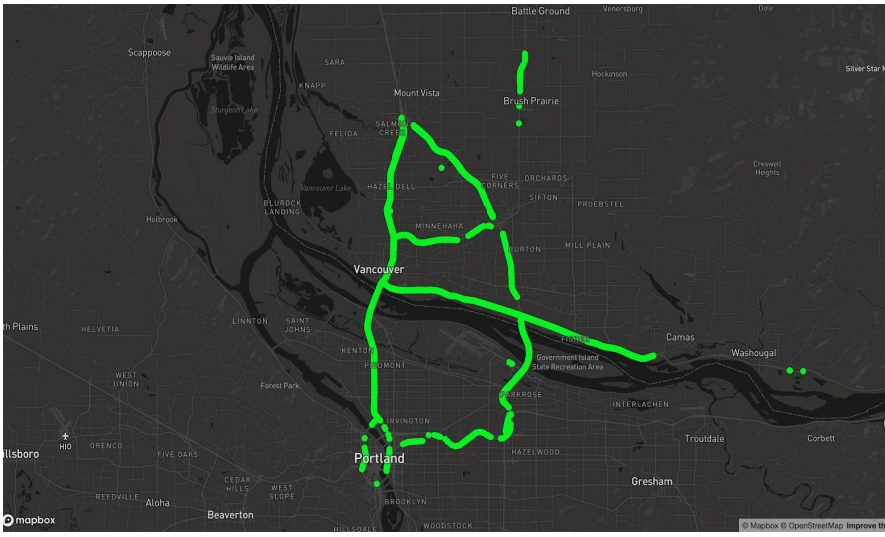
Visualization 5a, 5b, 5c, …. Three or more additional visualizations of your choice. Indicate why you chose each particular visualization.

5a. For a Weekday, show all locations (of all trips in the day) where the speed is above 25mph and below 30mph. Also for a Weekend day, find all locations for the same speed condition. Finally, provide a visual comparison between weekday and weekend speeds.

-> WeekDay : Friday (2020-10-16):

```
SELECT *
FROM breadcrumb
WHERE tstamp::date = date '2020-10-16'
AND speed > 25 AND speed < 30
```

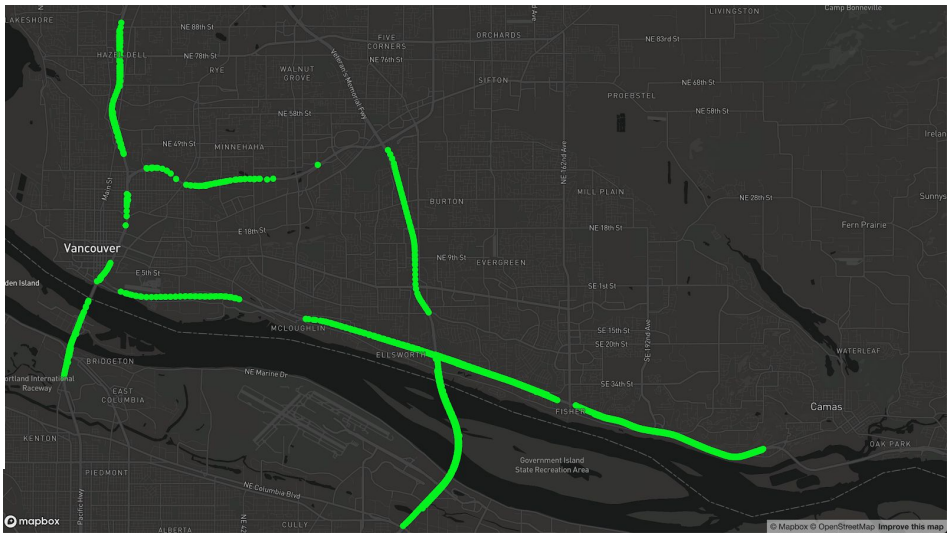
tstamp	latitude	longitude	direction	speed	trip_id
2020-10-16 15:12:32	45.597102	-122.684618	184	26	170472442
2020-10-16 15:12:47	45.593683	-122.683573	163	26	170472442
2020-10-16 15:13:37	45.582623	-122.678812	169	26	170472442
2020-10-16 15:13:42	45.581432	-122.678687	176	26	170472442
2020-10-16 15:13:47	45.58024	-122.678685	180	26	170472442
2020-10-16 15:23:41	45.524465	-122.665493	183	26	170472442
2020-10-16 16:01:12	45.653327	-122.662855	344	26	170472453
2020-10-16 16:01:22	45.655617	-122.663773	344	26	170472453



-> WeekEND : Saturday (2020-10-17):

```
SELECT *
FROM breadcrumb
WHERE tstamp::date = date '2020-10-17'
AND speed > 25 AND speed < 30
```

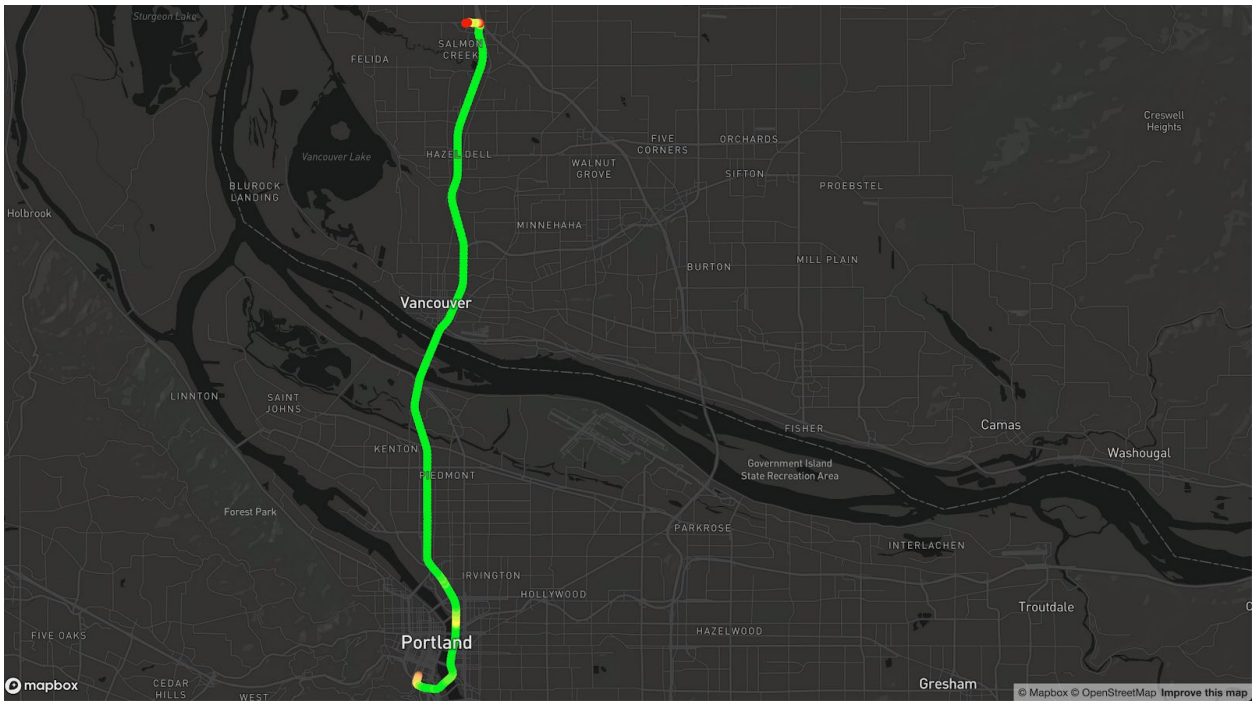
tstamp	latitude	longitude	direction	speed	trip_id
2020-10-17 06:58:32	45.606122	-122.562032	107	26	170570695
2020-10-17 06:58:42	45.605438	-122.558817	107	26	170570695
2020-10-17 06:58:47	45.605092	-122.557223	107	26	170570695
2020-10-17 11:12:02	45.756665	-122.547232	0	28	170570925
2020-10-17 14:10:32	45.73841	-122.551458	0	27	170571108
2020-10-17 14:10:37	45.739683	-122.551478	359	28	170571108
2020-10-17 14:10:42	45.740972	-122.551497	359	28	170571108
2020-10-17 14:10:47	45.742252	-122.551515	359	28	170571108
2020-10-17 14:10:52	45.743528	-122.551537	359	28	170571108
2020-10-17 14:10:57	45.744788	-122.551517	1	27	170571108
2020-10-17 14:11:02	45.746017	-122.551255	8	27	170571108
2020-10-17 14:11:12	45.748223	-122.55003	25	28	170571108
2020-10-17 08:44:49	45.589063	-122.494545	110	26	170543846



5b. In order to understand which is the longest yet fastest trip, query the database to find the longest trip by time that has an average speed > 20mph throughout the trip.

```
SELECT *
FROM breadcrumb
WHERE trip_id = (
    SELECT trip_id
    FROM breadcrumb
    GROUP BY trip_id
    HAVING avg(speed) > 20
    ORDER BY (max(tstamp) - min(tstamp)) desc
    LIMIT 1
)
```

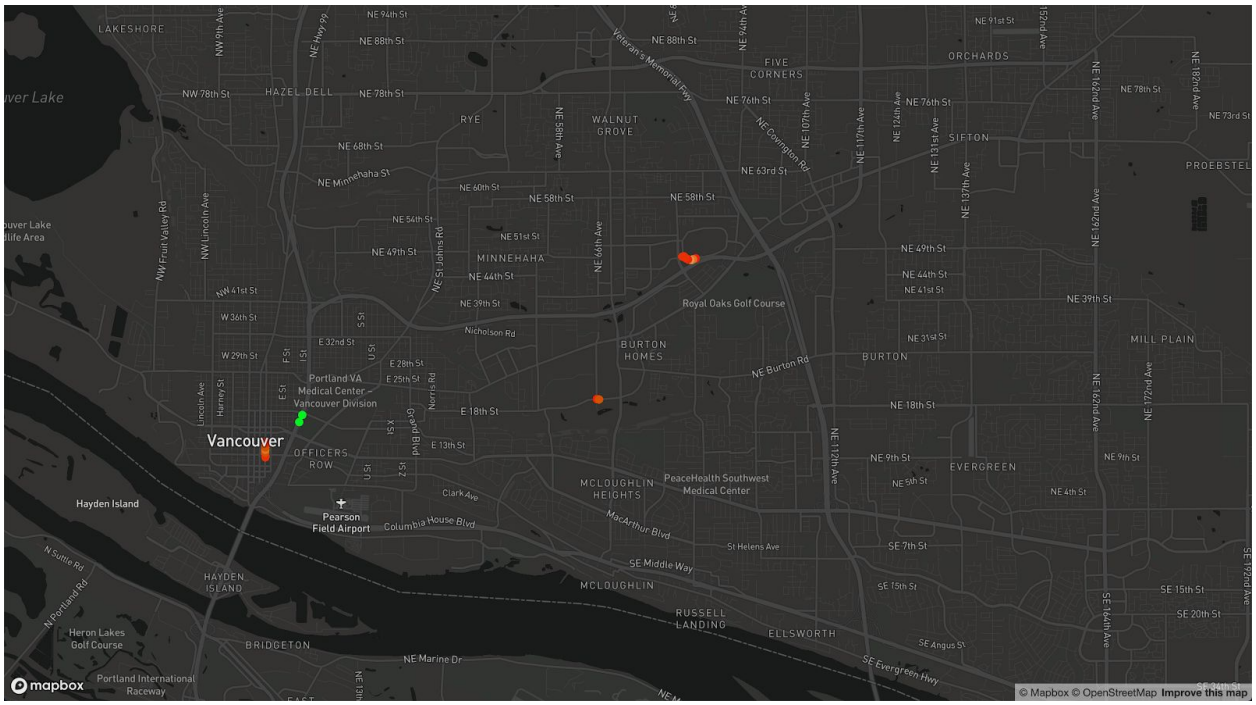
tstamp	latitude	longitude	direction	speed	trip_id
2020-10-16 05:47:34	45.509825	-122.682433	0	0	170472595
2020-10-16 05:47:39	45.509455	-122.682637	201	8	170472595
2020-10-16 05:47:44	45.509048	-122.682855	201	9	170472595
2020-10-16 05:47:49	45.508602	-122.68305	197	10	170472595
2020-10-16 05:47:54	45.50812	-122.683272	198	11	170472595
2020-10-16 05:47:59	45.507595	-122.683515	198	12	170472595
2020-10-16 05:48:04	45.507037	-122.683563	183	12	170472595
2020-10-16 05:48:09	45.506525	-122.683093	147	14	170472595
2020-10-16 05:48:14	45.506132	-122.682178	122	17	170472595
2020-10-16 05:48:19	45.505742	-122.681077	117	19	170472595
2020-10-16 05:48:24	45.505482	-122.679785	106	21	170472595
2020-10-16 05:48:28	45.50534	-122.678382	98	25	170472595



5c. Show trips on a weekday that have emitted only 10 or less sensor reading throughout their trip.

```
SELECT *
FROM breadcrumb
WHERE trip_id IN (
    SELECT trip_id
    FROM breadcrumb
    WHERE tstamp::date = date '2020-10-16'
    GROUP BY trip_id
    HAVING COUNT(*) <= 10)
```

tstamp	latitude	longitude	direction	speed	trip_id
2020-10-16 13:48:45	45.65599	-122.585032	0	0	170452147
2020-10-16 13:48:50	45.65587	-122.585355	242	6	170452147
2020-10-16 13:48:55	45.655723	-122.586027	253	11	170452147
2020-10-16 13:49:00	45.655753	-122.586717	274	10	170452147
2020-10-16 13:49:05	45.655845	-122.587063	291	5	170452147
2020-10-16 13:49:10	45.656002	-122.58735	308	5	170452147
2020-10-16 13:49:15	45.656202	-122.587605	318	5	170452147
2020-10-16 13:49:20	45.656158	-122.58794	260	5	170452147
2020-10-16 08:51:32	45.628762	-122.671367	0	0	170515717
2020-10-16 00:35:42	45.632288	-122.6644	0	0	170482303
2020-10-16 00:35:47	45.633285	-122.6638	23	23	170482303
2020-10-16 00:35:52	45.63427	-122.663202	23	23	170482303
2020-10-16 22:12:31	45.628172	-122.67055	0	0	170457824
2020-10-16 22:12:36	45.628407	-122.670518	5	5	170457824
2020-10-16 22:12:41	45.628537	-122.670517	1	2	170457824
2020-10-16 22:14:11	45.628663	-122.670545	351	0	170457824



Your Code

Provide a reference to the repository where you store your code. If you are keeping it private then share it with Bruce (bruce.irvin@gmail.com), David and Aman (github references TBD).

Git Repo - https://github.com/Yokeshthirumoorthi/DataEngineeringWithKakfa