Exploratory Data Analysis

Spatiotemporal Fuel Consumption Forecasting

YAP, Zhi Yun

Supervised by: Mr. Ahmad, Dr. Tristan, Mr. MingYang, Prof. Pan Hui

Work Summary

- Extracted POI data points from OpenStreetMap Overpass API
- Append additional attributes to existing dataset
  - CurSpeed – vehicle speed in current time period
  - POI – point-of-interest in Shenzhen
  - IsWeekday – whether current day is a weekday
- Analyzed trend and distribution (temporal, spatial, semantic) in the dataset
- Completed preliminary data cleansing
  - Removed erroneous data with faulty speed information
  - Removed data out of the Shenzhen city
- Propose sliding interval for sampling
- Questions-to-ask

Overview of the dataset

- The dataset is broadly divided into 3 subsets – Trajectory data, Trip data and Edge data

Trajectory data		**Trip data**		**Edge data**	
# trajectory	: 18,182,397	# trip	: 4,833,563	# edge	: 51,130
# vehicle	: 3,364	# starting date	: 82	# starting node	: 35,245
# day	: 35	# stopping date	: 81	# ending node	: 35,989
# edge	: 43,085	Earliest start time	: 00:00:01	# highway type	: 11
# starting node	: 30,593	Latest stop time	: 23:59:59		
# ending node	: 31,143				
# trip	: 218,006				
# trip (weekday)	: 156,127				
# trip (weekend)	: 62,773				

Temporal view: Speed pattern

- First, compute speed of each trajectory at current time using formula:

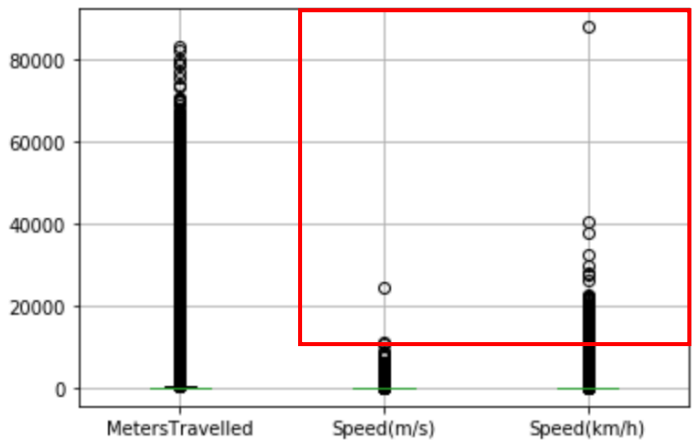
$$v_r(i) = \frac{dist(l_{pi}, l_{pi+1})}{t_{pi+1} - t_{pi}}$$

where v_r denotes speed at time t_{pi} and $dist$ is the great-circle distance between GPS locations l_{pi} and l_{pi+1} (i.e., longitude and latitude)

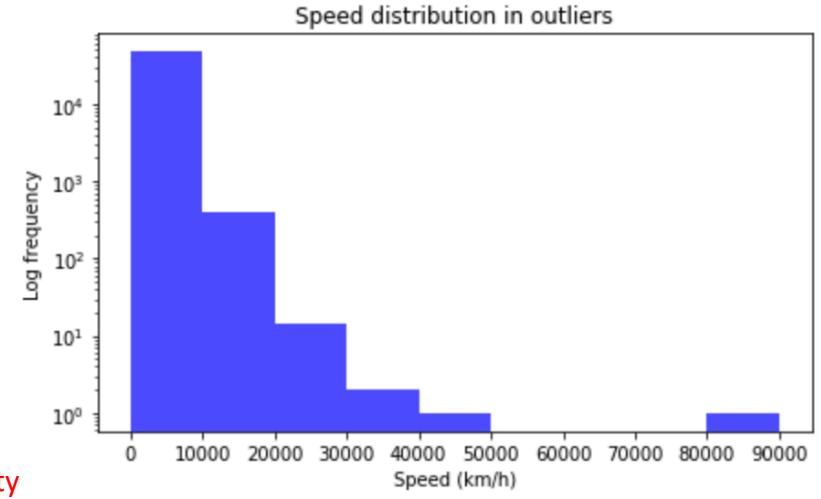
- Identified outliers and erroneous data with > 5000km/h

	ObjectID	TripID	StartNode	EndNode	Timestamp
18060355	589210	215702	29981	18978	2016-07-05 17:22:05
18060356	589210	215702	29981	18978	2016-07-05 19:35:55
18060357	589210	215702	18978	18891	2016-07-05 21:49:45
...	...	...	...	...	...
18060414	589210	215702	11272	13641	2016-07-11 04:58:24
18060415	589210	215702	4344	4997	2016-07-11 07:12:14

Delayed update appears to be faulty

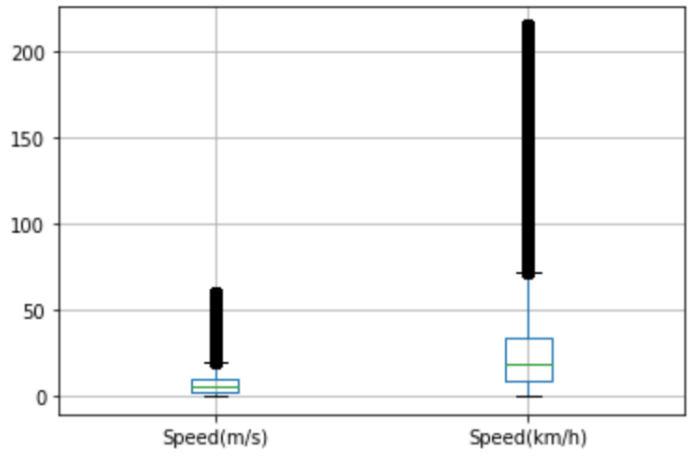


Unreasonable speed appears to be faulty

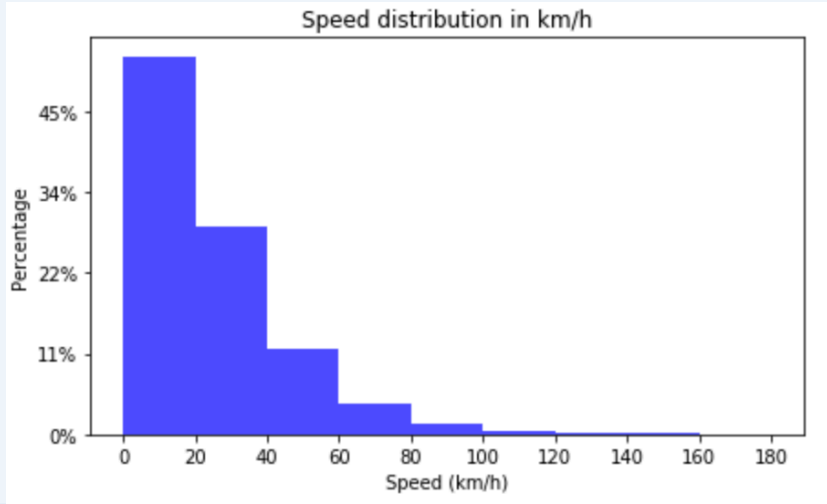


Temporal view: Speed pattern

- Remove 1.95% erroneous data with `SecondsTravelled` equals to 0
- Filter out trip with only 1 trajectory data
- Remove outliers with speed(km/h) > 0.9973 quantile
  - This gets rid of data with speed larger than 215km/h

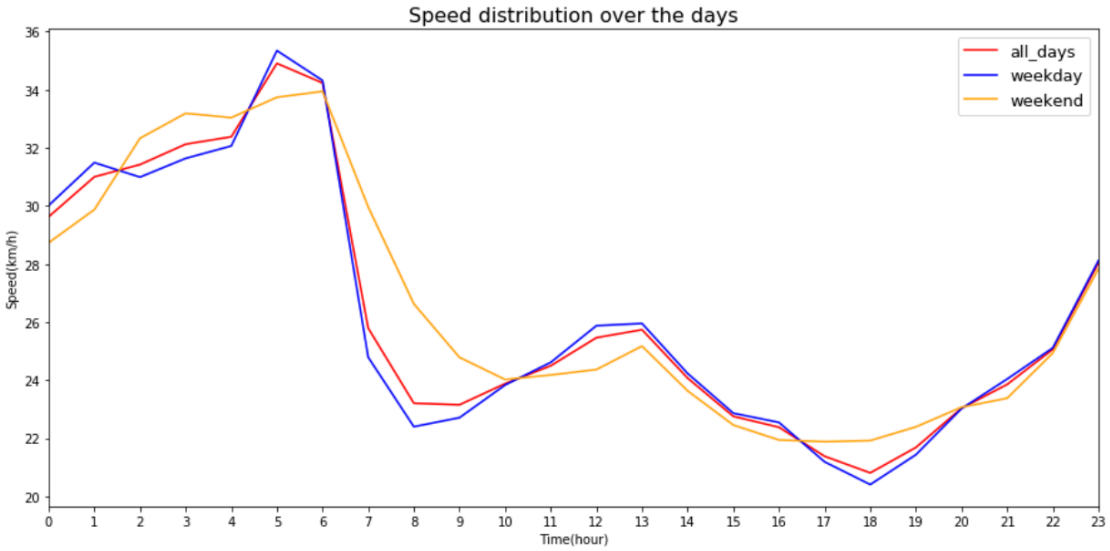


- Avg. : 24.2. km/h
- Min. : 0.001 km/h
- Max. : 215 km/h
- 25% : 8.69 km/h
- 50% : 16.7 km/h
- 75% : 33.7 km/h



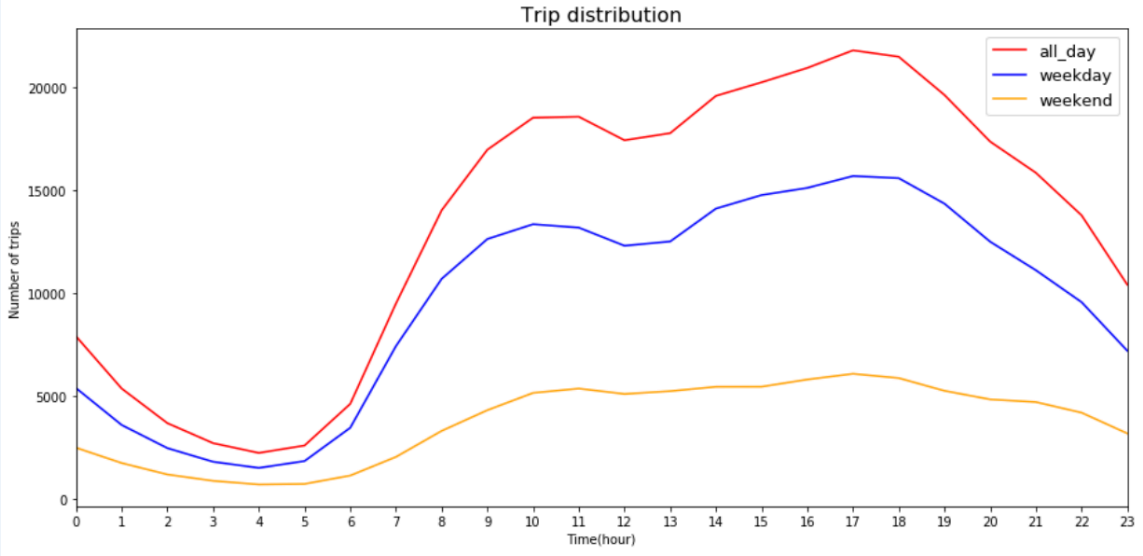
- Over 45% of trajectory data has vehicle speed [0, 20] km/h
- This is because >73% trajectory data is collected during weekdays
- Traffic congestion is much worse during weekdays (to be shown in next slide)

Temporal view: Speed pattern throughout the day



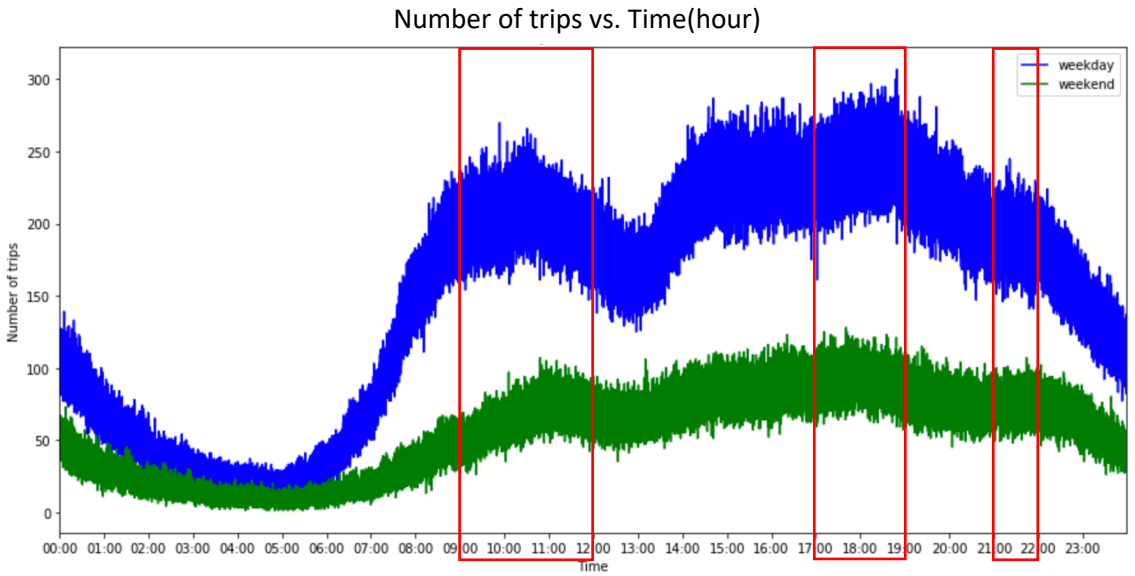
- Larger fluctuation in speed pattern during weekdays as compared to weekend

Comparing speed distribution to trip distribution

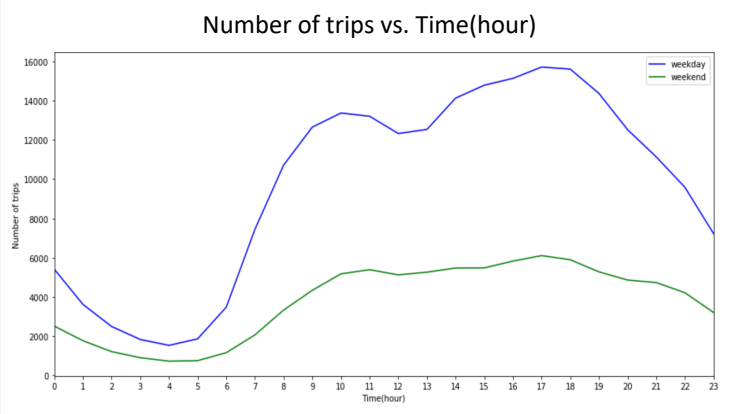


- As intuitive thinking, overall speed distribution has an inverse relationship with the overall trip distribution through the day

Temporal view: Traffic distribution throughout the day



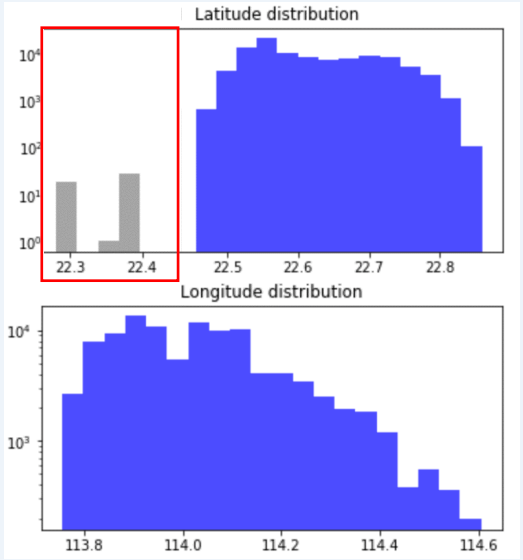
- Similar to speed pattern, traffic during weekdays have more greater fluctuation
- Overall similar traffic pattern on weekdays and weekend, both showing peaks at 3 different time periods
  - 0900 – 1200 morning
  - 1700 – 1900 evening
  - 2100 – 2200 night



Added new binary attribute `IsWeekday` to help capture this trend

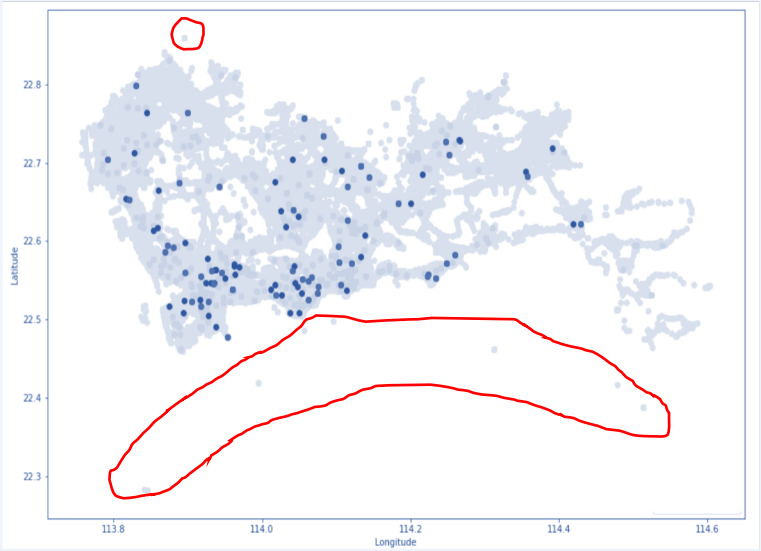
Spatial view: Geographic distribution

Plot location distribution



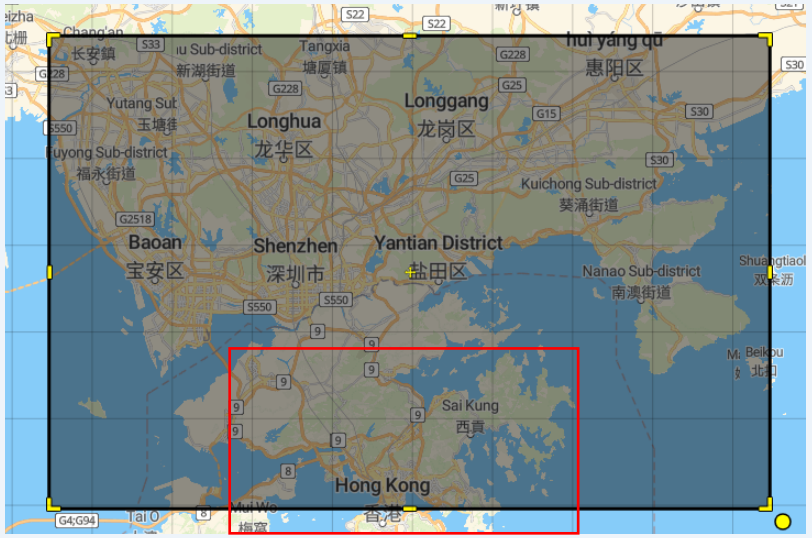
- Latitude coordinate smaller than 22.4 appears to be the outliers

Visualize GPS coordinates on 2D map



- Outliers previously identified (bounded in red) appear to be disconnected from the main urban network

Locate bounding box on Google Map

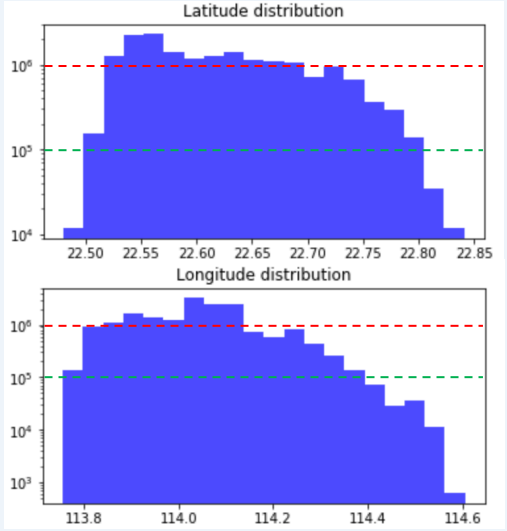


- Outliers previously identified appear to be out of Shenzhen city

Spatial view: Geographic distribution

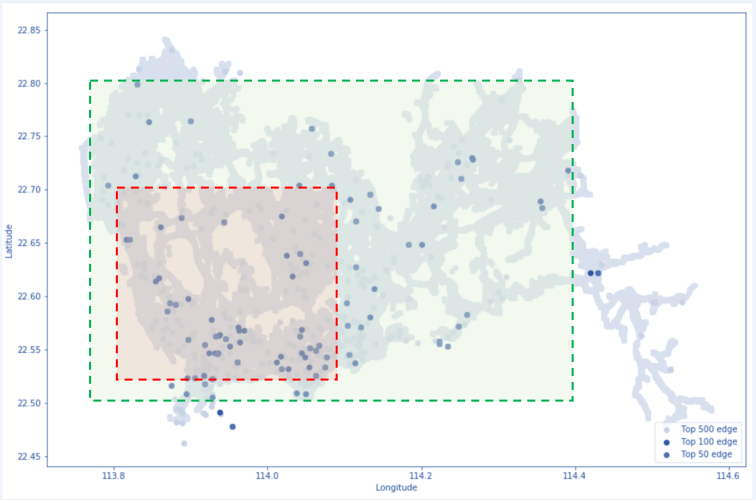
After removing outliers in latitude distribution

Plot location distribution



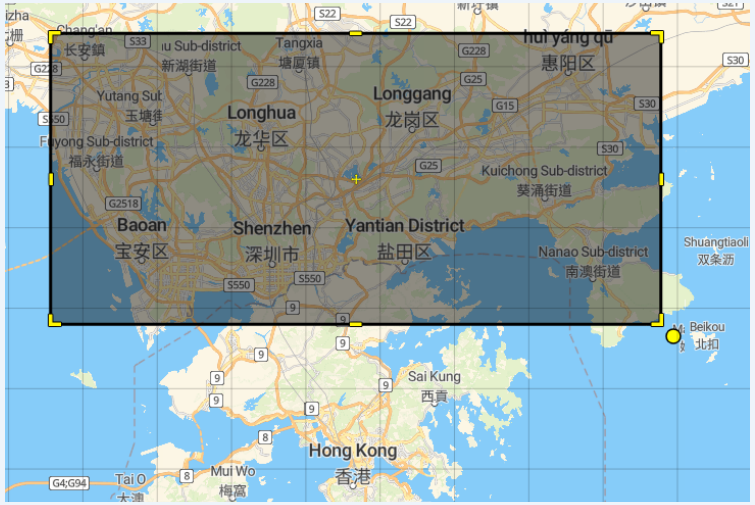
	≥ 10^5	≥ 10^6
Lat	[22.50, 22.80]	[22.52, 22,70]
Lon	[113.8, 114.4]	[113.8, 114.1]

Visualize GPS coordinates on 2D map



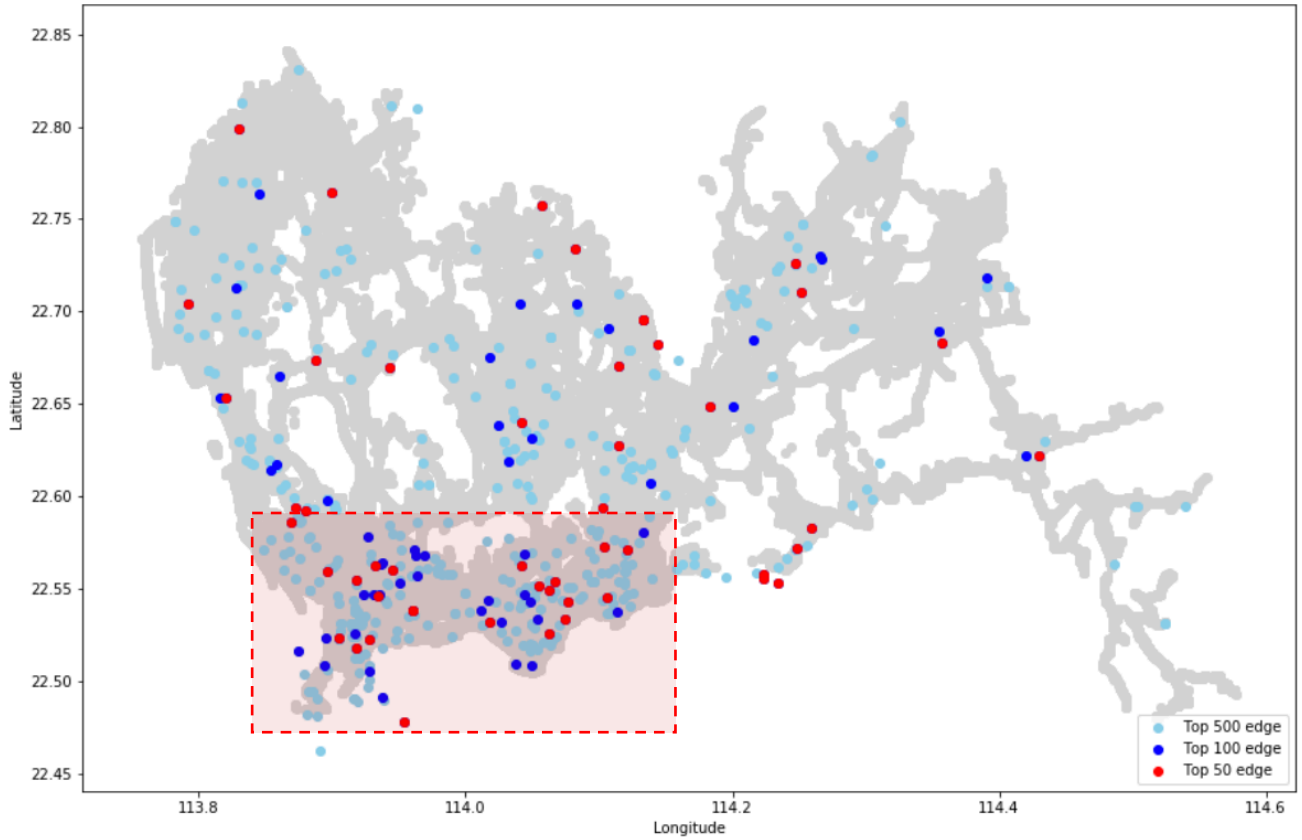
- > 52% GPS coordinates are captured by the red box (southern west)
- > 65% GPS coordinates are captured by the green box

Locate bounding box on Google Map



- After removing the outliers, all the remaining trajectory data are within Shenzhen city

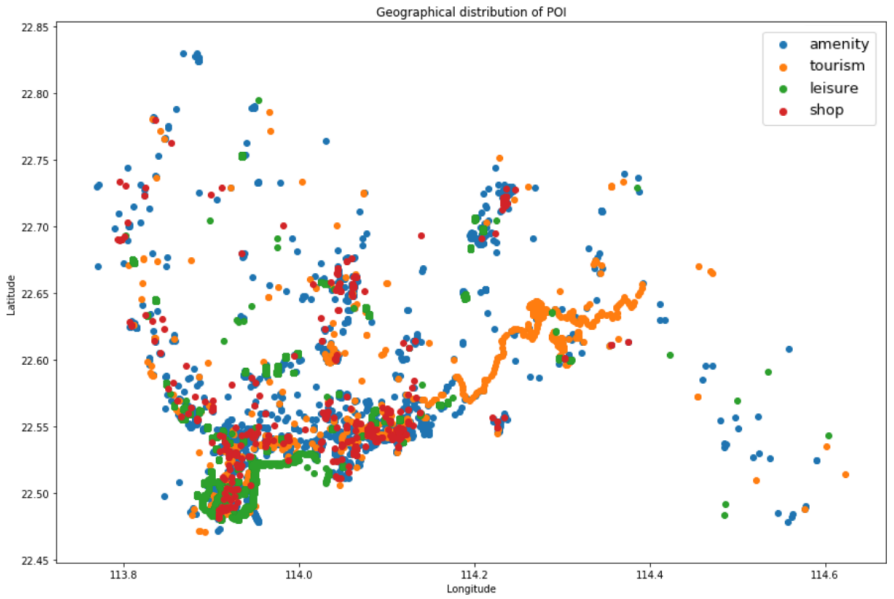
Spatial view: Geographical distribution

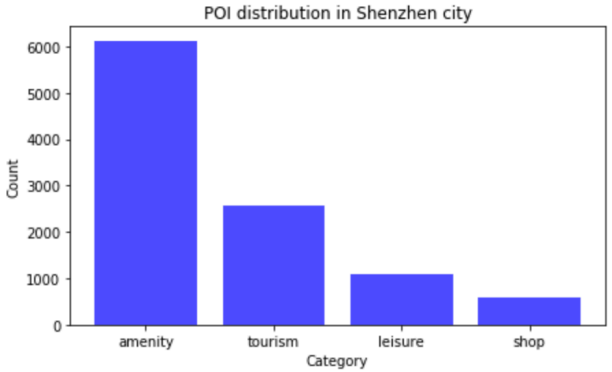


- Of the 42,997 edges in the road network, **top 500, 100 and 50** edges with most vehicle trajectories are highlighted
- **Southern west** of Shenzhen city appears to be busiest region with large cluster of top 500, 100, and 50 edges

Semantic view: POI information from OpenStreetMap

- Get all coordinates and names of POI (amenity, tourism, leisure, and shop) in Shenzhen from Overpass API
- Each category has over 100 tags – fine-grained POI definition
- In total 10,356 POI data points extracted





Sample geojson data from Overpass API


```
"features": [
  {
    "type": "Feature",
    "properties": {
      "@id": "node/277045518",
      "amenity": "fuel"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        114.0431003,
        22.6009762
      ]
    },
    "id": "node/277045518"
  },
  {
    "type": "Feature",
    "properties": {
      "@id": "node/277501742",
      "amenity": "school",
      "created_by": "Potlatch 0.9c",
      "name": "Tsinghua Experimental School",
      "name:zh": "桃源居中澳实验学校"
    },
```


Longitude, Latitude

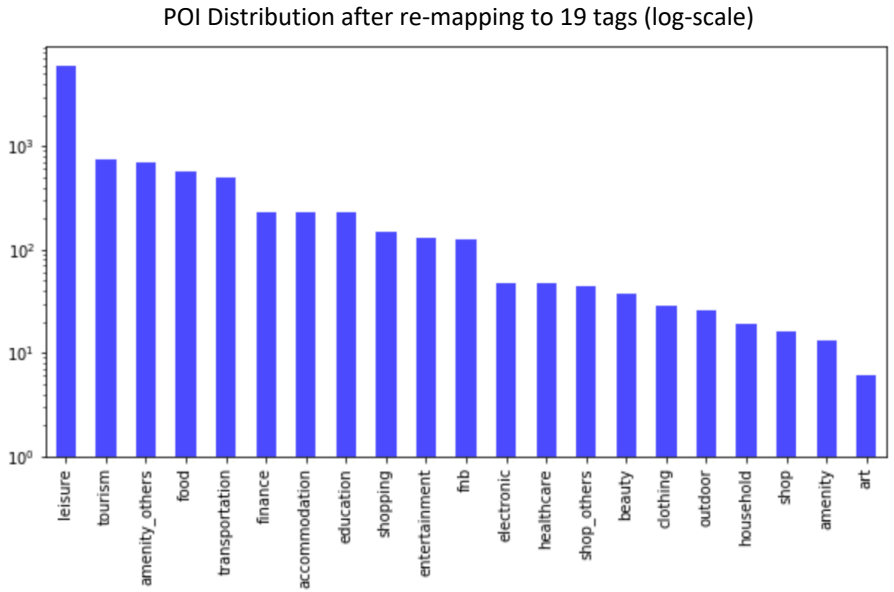
Category: tag

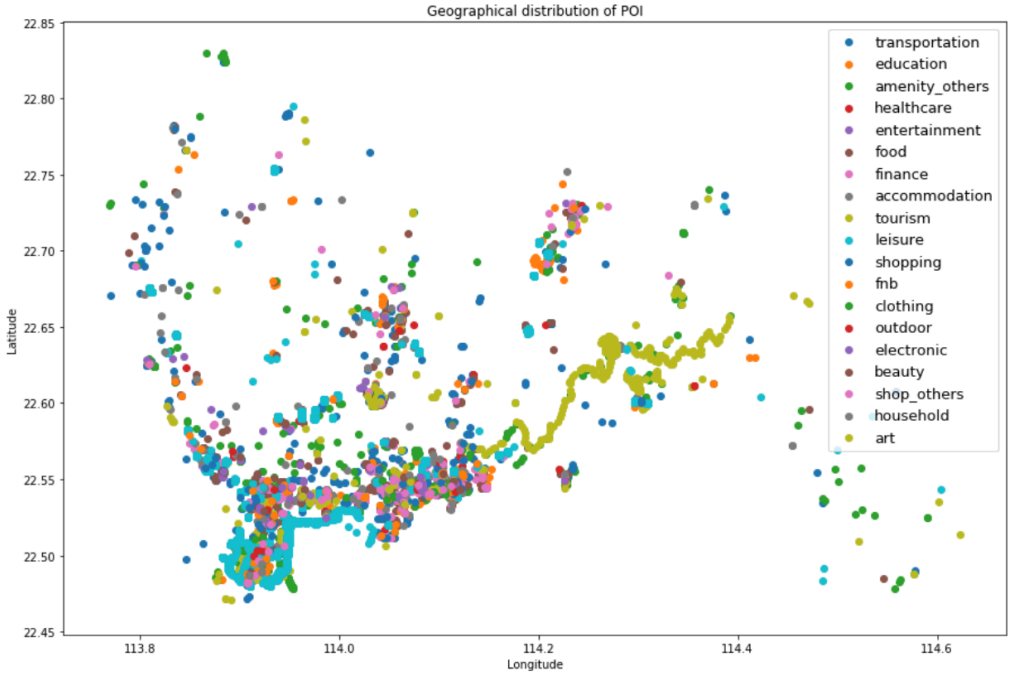
Name

Semantic view: POI information from OpenStreetMap

- Re-map 168 tags into 19 new tags to present data with appropriate granularity
- Leisure category has far more GPS locations in Shenzhen due to larger recreational area

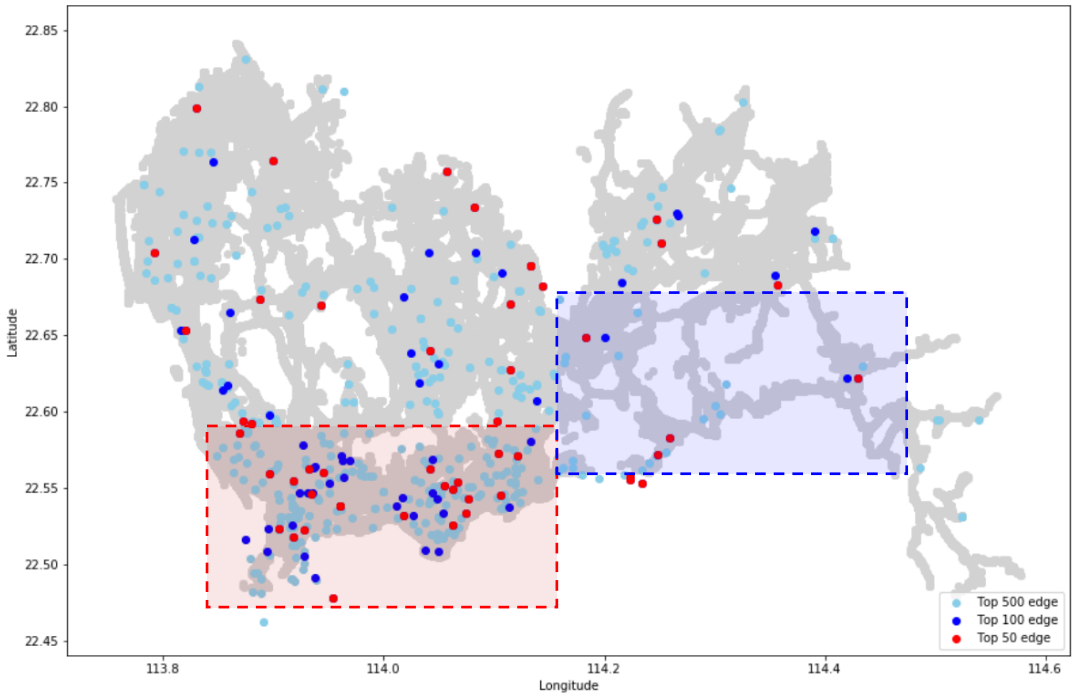
art	education	food	shop_others
amenity_others	electronic	healthcare	shopping
accommodation	entertainment	household	tourism
beauty	finance	leisure	transportation
clothing	fnb	outdoor	

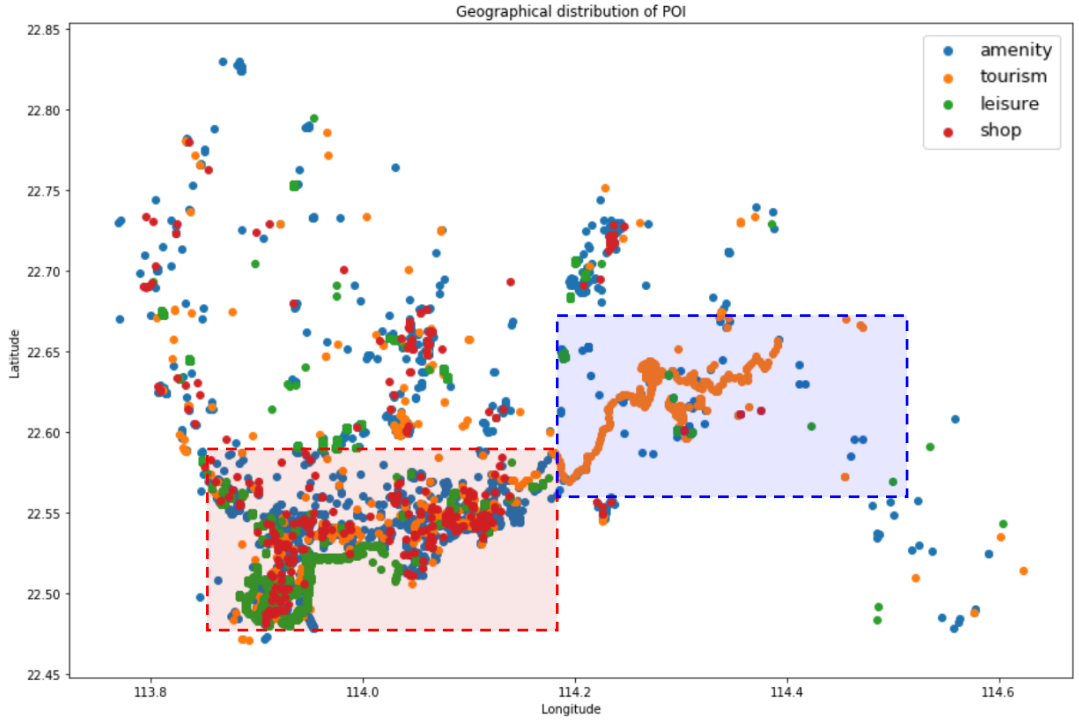




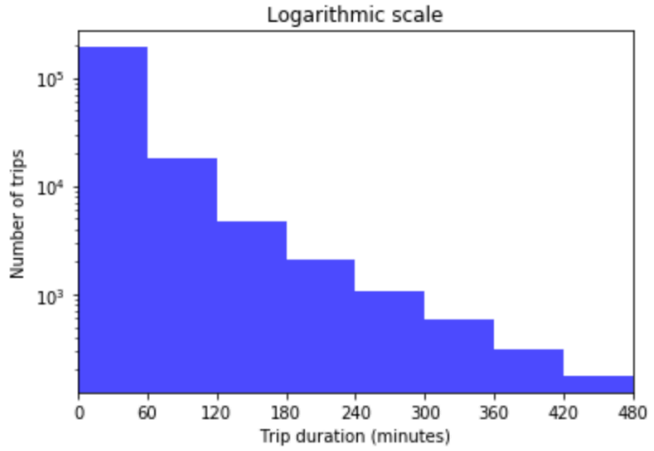
Relationship between traffic pattern & POI distribution

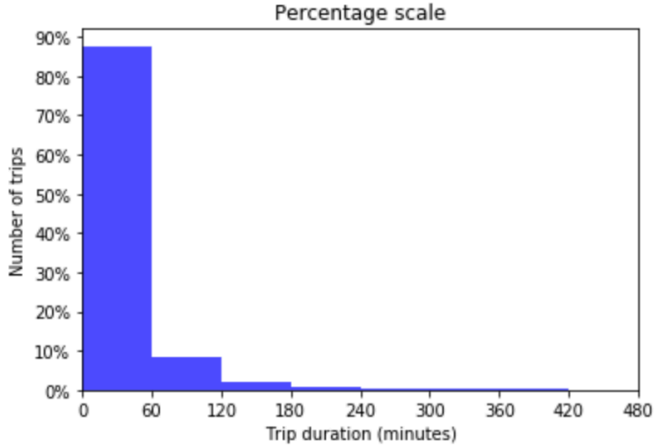
- Regions with POI cluster(s) have more busy edges
- Tourism spot (blue box) has a weak relationship with the traffic pattern
- Southern west, the busiest region (red box) has most shop and leisure spots

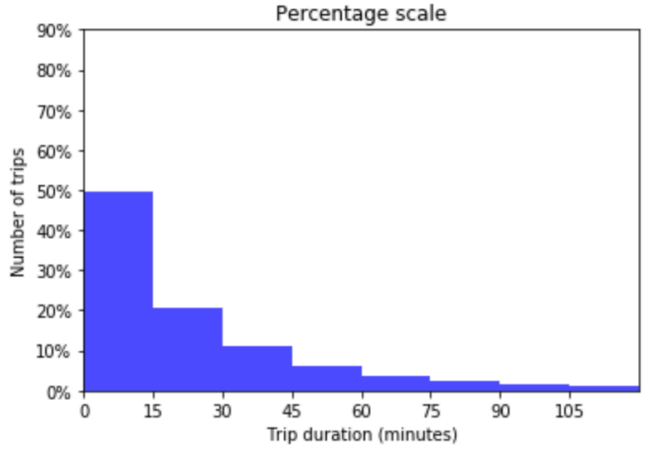




Determine sliding interval from trip duration pattern





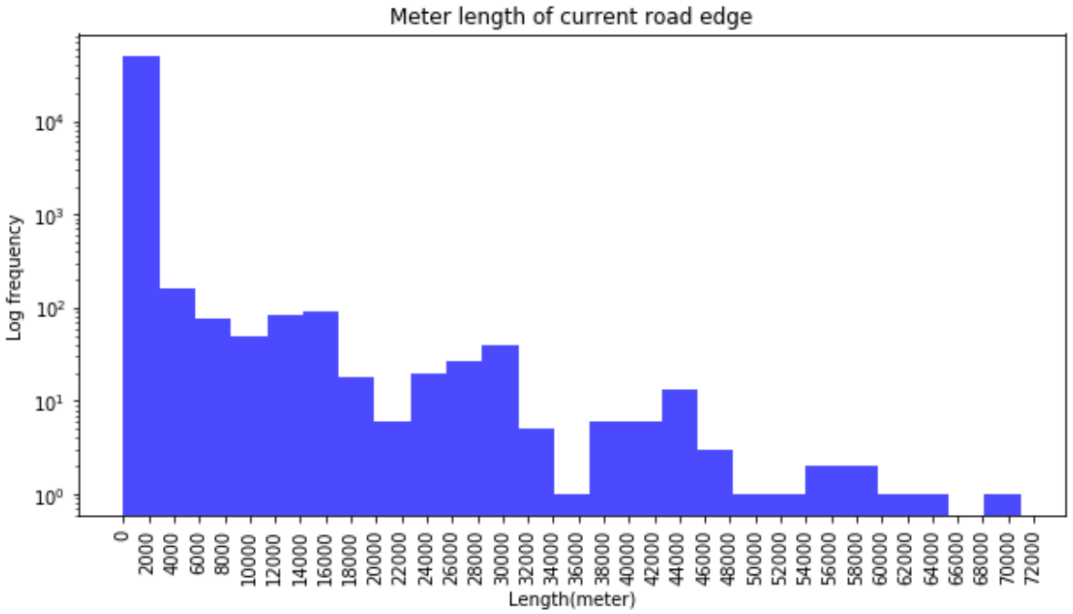


- Nearly 90% of the trip data has duration length of 0-60 min.
- ~50% of the trips are within 15 min.

Propose:

- Use time interval of **5-minutes** as the window for input data sampling
- Multi-step prediction for time intervals of 5, 10, and 15 minutes

Road length distribution of current edges in the network



- Edges defined in the current road network have great variability in length
- For this project, edges and nodes need to be redefined based on speed, traffic, POI features and geographical connectivity

To-do:

- Research on existing algorithm and techniques to transform the edges and nodes defined to *supersegment* (as mentioned in proposed methodology)

Plan for following weeks

1. Match POI data points to trajectory data

2. Research on algorithm to re-define nodes and edges for graph construction

3. Graph that captures the road network structure to be ready by the end of October

Question

1. Index for trip data (trip_sz.csv) does not appear to be the foreign key of TripID in trajectory data (traj_sz.csv)
   - Starting time does not match

```
In [114]: 1 traj_df[traj_df['TripID']==46419][:3]
```

Out[114]:

	ObjectID	Lon	Lat	StartMileage	Date	TripID	EdgeID	Highway	IsWe
3287425	181669	114.057760	22.545838	68873.350000	2016-07-06 18:20:57	46419	43192	tertiary_link	
3287426	181669	114.057582	22.543052	68873.565508	2016-07-06 18:21:26	46419	2830	trunk_link	
3287427	181669	114.053530	22.543150	68873.781015	2016-07-06 18:21:55	46419	18758	trunk_link	

```
In [112]: 1 trip_df[trip_df['TripID']==46419]
```

Out[112]:

	TripID	ObjectID	StartTime	StartLon	StartLat	StopTime	StopLon	StopLat	TravelM
46419	46419	121430	2016-07-01 13:16:26	114.225873	22.71149	2016-07-01 14:21:05	113.959133	22.563538	

2. Since fuel consumption data is only available at trip level, how can we calculate fuel consumed at each timestamp?
3. How is edge defined in current dataset? Is it defined based on map extracted from OpenStreetMap?