RaspberryPi Trip Analysis

Purpose:

To report RaspberryPi's First Trip

Contents:

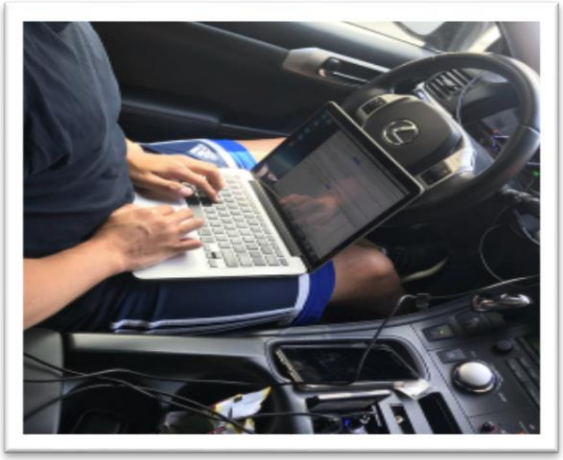
1. Setting up the RP3

1-1. physically setting up RP

It takes a maximum of 15 minutes

For setupn, we need a power source and two USB port (monitor and RP device), then we are successfully able to operate RaspberryPi in the car

Setup requires the camera to be fixed during the trip. Thus, it was taped under the CarVi Device.



Using the Hotspot, we were able to access wifi to connect to RP, AWS, and laptops

1-2. RP code settings

It takes approximately 10 minutes to boot RP, connect hotspot and start GPS. If RP boots correctly the red light will be on.

Running a created python file to have RP run the gps commands. Once you run the file you need to wait until the green light is flashing on GPS kits. Normally 7-8 trials are needed.

```
1  import os
2  os.system("sudo killall gpsd")
3  os.system("sudo cat /dev/ttyUSB0")
4  os.system("sudo gpsd /dev/ttyUSB0 -F /var/run/gpsd.sock")
5
6  os.system("cgps")
```

its functions are explained below:

rebootgps.py : gps rebooting every time you turn on RP3

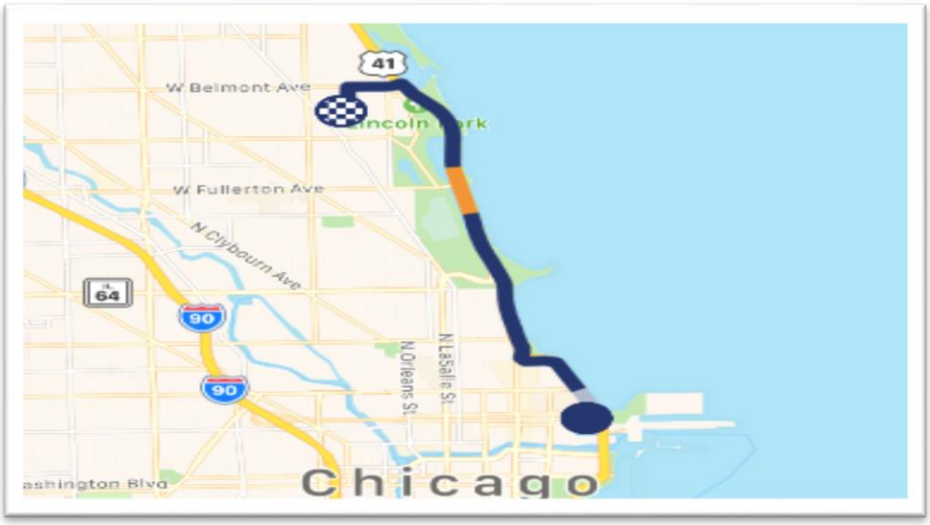
sudo kill all gpsd : Every time you run RP, you need to run this code to kill all gps signals
sudo cat /dev/ttyUSB0 & sudo gpsd : rebooting gps to capture the signals

This setup works under a strong gps and wifi signal, but takes longer under a weak signal.

For this trip, the entire setup process took about 35min.

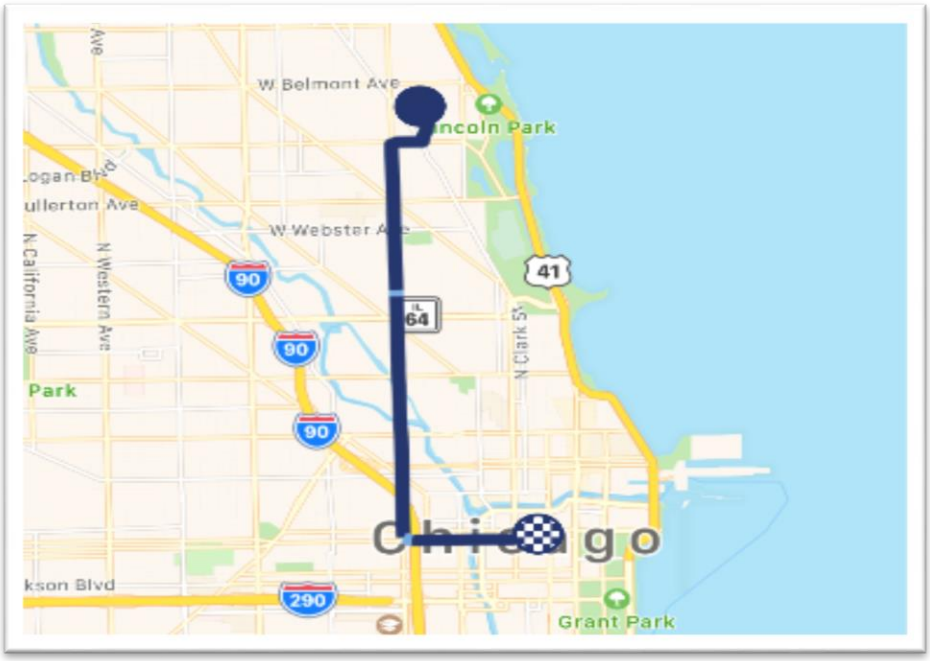
2. Two Trips

2-1. *First trip*: Streeterville to Lincoln Park



Hotspot lost the signal often and the lake could be a factor of the weak signals as well. Therefore, we could not get full signals during the trip. We needed to restart running RP for this reason. Unfortunately, we could NOT collect the valuable data.

2-2. *Second trip*: Lincoln Park to Loop



We were able to capture a better GPS signal, but GPS was still not detected correctly. RP automatically restarted it several times (5-7 times). No reason was found for this. However, we were able to collect 785 seconds of trip data (about 1/3 proportion of total trip). The GPS signal strength was better, but no consistently strong GPS signal could be obtained. To collect the data efficiently, a strong and stable GPS signal is needed.

3. RaspberryPi's Data Attributes

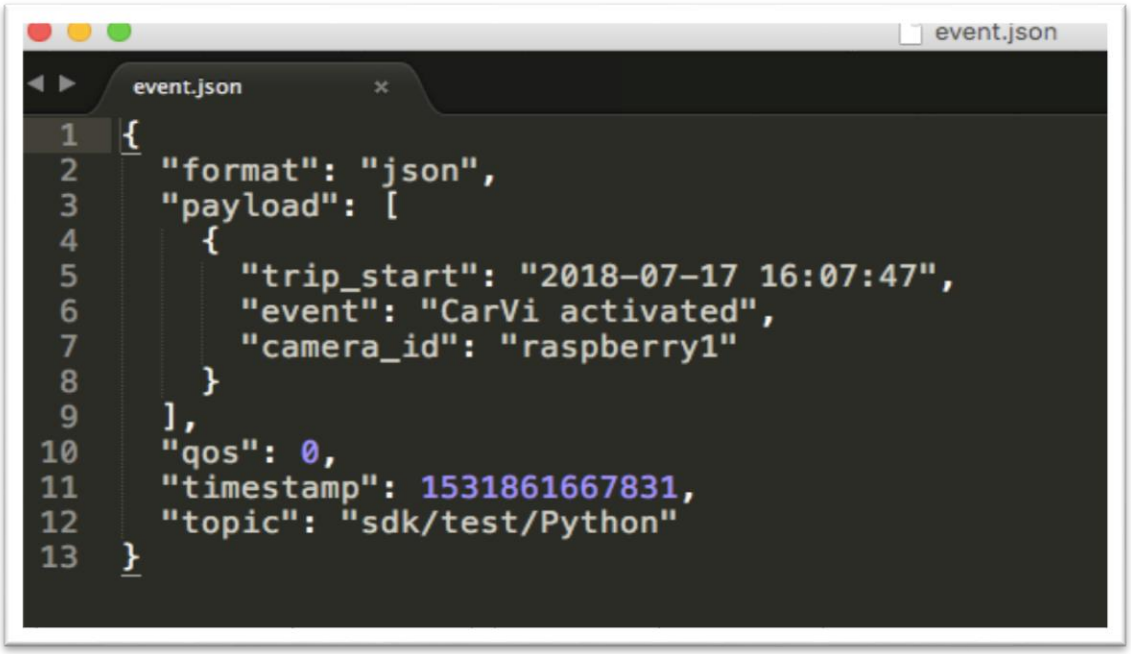
: 26 attributes are collected

Trip Start*: camera_id, event, trip_start,*
Location: *'NEW_location', 'location'*
GeoFencing : 'geo1', 'geo2', 'geo3', 'geo4', 'geo5', 'geo6'
Sensor: 'ax', 'ay', 'az', 'gx', 'gy', 'gz', 'mx', 'my', 'mz',
Others : satellites, emergencyCall, distance, speed, time_stamp, hdop

4. Data Outputs

4-1. Trip Start Attributes

'camera_id', 'event', 'trip_start'
: Trip Start, Event, and **Camera Id** attributes start to be recorded
At the point when RP3 starts to run basicPubsub.py, it records three attributes of "trip_start", "event", and "camera_id".



4-2. Two Location Options

4-2-1. RP's original GPS output: latitude and longitude with wrong decimal points
This is the raw gps data running on RP by commanding 'cgps'.

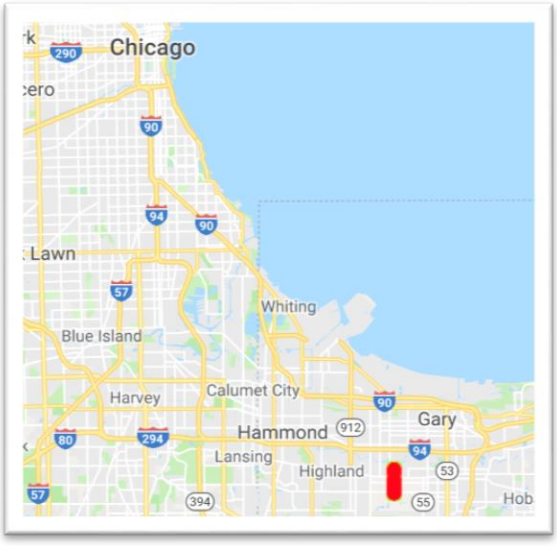
```
GPRMC,233055.00,A,4152.96575,N,08737.63550,W,0.689,,170718,,,A*6B
GPVTG,,T,,M,0.689,N,1.276,K,A*26
GPGGA,233055.00,4152.96575,N,08737.63550,W,1,04,4.61,410.7,M,-33.8,M,,*68
GPGSA,A,2,05,13,15,29,,,,,,,,,4.71,4.61,1.00*0B
GPGSV,3,1,12,02,26,107,05,05,46,053,27,07,01,035,,13,52,118,24*72
GPGSV,3,2,12,15,41,177,30,16,05,327,,20,09,250,,21,28,299,*7B
GPGSV,3,3,12,25,04,221,,26,10,302,,29,68,240,07,30,04,062,*7B
GPGLL,4152.96575,N,08737.63550,W,233055.00,A,A*7D
```

4-2-2. *'NEW_location'* **utilized by function** *vs 'location'* **utilized by module**
There are two different way to catch latitude and longitude. We named them as 'NEW_location' and 'location' to compare which way would be more accurate. The differences are shown below.

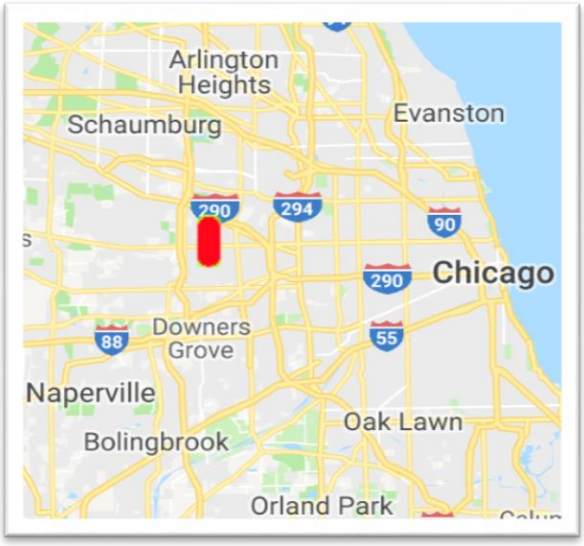
NEW_location : loca_format() is the function that you changed the decimal point correctly
location : *data.latitude* and *data.longitude* is a module that changes the decimal point correctly

```
message['aa_NEW_location'] = loca_format(data.lat) + "," + loca_format(data.lon)
message['aa_location'] = str(round(data.latitude,4)) + "," + str(round(data.longitude))
```

4-2-3. Outputs comparison *'NEW_location' vs 'location'*
Both methods output GPS locations incorrectly during *Second Trip (Lincoln park to Loop)*



NEW_location



location

4-3. Multiple Geo Fencing Areas

'geoZone1New', 'geoZone1', 'geoZone2New', 'geoZone2', 'geoZone3New', 'geoZone3'
Geo fencing functions outputs boolean values of True or False.
True: when the device is in the geo polygon area that we set up
False: when the device is NOT in the geo polygon area that we set up

As reported in 4-2, RP incorrectly collected GPS data.
If there is NO signal, 'location' values are (0.0, 0.0).
We observed (0.0, 0.0) for 'location' although 'satellites' value is either 03 or 04.

```
check = trip_query_df.a6_satellites[trip_query_df.aa_NEW_location == '0.0,0.0']
check.value_counts()
```

```
00    51
03    15
04     1
```

We only collected 67 True values in zone1.
Since we never got ideal gps values, zone2 and zone3 recorded False values.

However, we confirmed that multiple geo fencing functions were working during the trip.
Zone1 = original point (in case of RP cannot detect the gps signal)

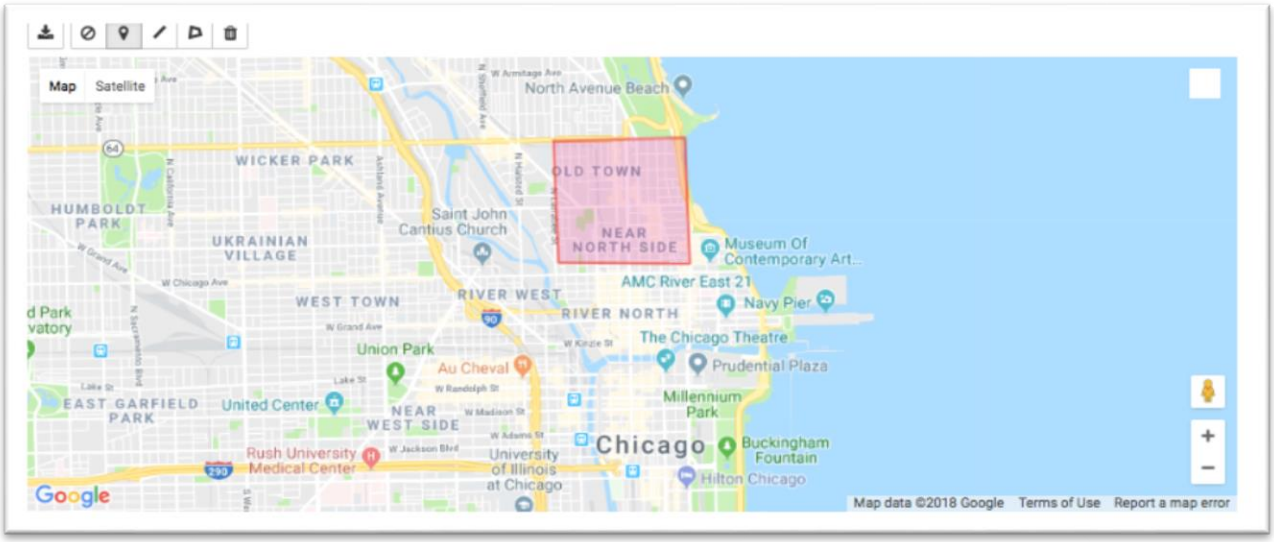

```
New_location

False    718
True      67
Name: aaN_geo, dtype: int64
```

```
location

False    718
True      67
Name: aa_geo, dtype: int64
```

Zone2 = old town area



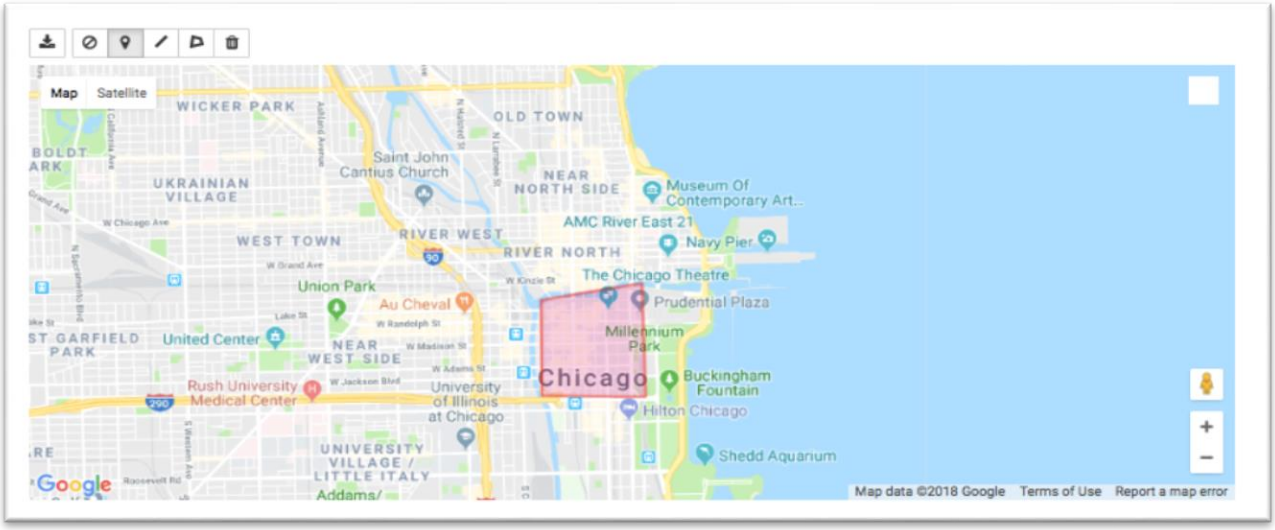

```
New_location

False    785
Name: abN_geo, dtype: int64
```

```
location

False    785
Name: ab_geo, dtype: int64
```

Zone3 = loop area




```
New_location

False    785
Name: acN_geo, dtype: int64
```

```
location

False    785
Name: ac_geo, dtype: int64
```

4-4. Sensor data

'ax', 'ay', 'az'

```
trip_query_df[['ax','ay','az']].head()
```

	ax	ay	az
0	[0.8298439979553223, 0.8317959904670715, 0.829...	[-0.4096759855747223, -0.4084559977054596, -0....	[0.33647599816322327, 0.3306199908256531, 0.33...
1	[0.8303319811820984, 0.8337479829788208, 0.833...	[-0.41333597898483276, -0.407723993062973, -0....	[0.33598798513412476, 0.3291559815406799, 0.3...
2	[0.831063985824585, 0.8308199644088745, 0.8308...	[-0.4101639986038208, -0.41235998272895813, -0...	[0.33086398243904114, 0.33671998977661133, 0.3...
3	[0.8291119933128357, 0.8332599997520447, 0.832...	[-0.40918800234794617, -0.40918800234794617, -...	[0.3352559804916382, 0.3291559815406799, 0.32...
4	[0.8325279951095581, 0.827892005443573, 0.8286...	[-0.4089439809322357, -0.41040799021720886, -0...	[0.33013200759887695, 0.33476799726486206, 0.3...

'gx', 'gy', 'gz'

```
trip_query_df[['gx','gy','gz']].head()
```

	gx	gy	gz
0	[199.61848394785468, 199.66932787603628, 199.6...	[68.16535621861895, 68.17921467244491, 68.1792...	[24.416541461941364, 24.43439728397907, 24.434...
1	[199.8872518916947, 199.91143076135654, 199.93...	[67.89818653904473, 67.92085593690001, 67.8939...	[24.81562941514521, 24.819141839925887, 24.825...
2	[199.62129798579272, 199.59487881408313, 199.5...	[68.01466175496097, 68.03212654866869, 68.0321...	[24.616392797055937, 24.611411881603704, 24.61...
3	[199.3399624938799, 199.25786362005636, 199.25...	[67.95362035434856, 67.95398918456374, 67.9539...	[24.56825533133298, 24.574267605349885, 24.574...
4	[199.203331389723, 199.10609681669868, 199.106...	[68.12732572532032, 68.11436202627564, 68.1143...	[24.593349446343606, 24.594373974719108, 24.59...

'mx', 'my', 'mz'

```
trip_query_df[['mx','my','mz']].head()
```

	mx	my	mz
0	[9.040674209594727, 9.255881309509277, 9.29552...	[-22.803131103515625, -22.62689971923828, -22....	[-3.7159624099731445, -3.687964916229248, -3.3...
1	[9.494452476501465, 9.495133399963379, 9.49513...	[-22.537275314331055, -22.711387634277344, -22...	[-2.975538969039917, -2.4842240810394287, -2.4...
2	[9.320876121520996, 8.910761833190918, 8.91076...	[-22.449220657348633, -22.61026954650879, -22....	[-2.750853300946045, -2.8598599433898926, -2....
3	[9.087240219116211, 9.331130981445312, 9.31570...	[-22.745315551757812, -22.59786605834961, -22....	[-2.4923858642578125, -2.3885891437530518, -2....
4	[8.68591594696045, 8.839469909667969, 8.839469...	[-22.968549728393555, -22.9056453704834, -22.9...	[-2.5740697383880615, -2.9967551231384277, -2....

We did not expect to obtain more than 5 values / second of sensor data. From this test we confirmed that each sensor data captured 7 to 10 values / second. We can conclude that RP's sensor works outside better than RP's indoor sensor.

The sensor worked as desired.

4-5. other attributes

satellites

```
satellites

06    463
05    135
07    107
00     51
03     17
04     12
```

emergencyCall

```
emergencyCall

False    785
```

distance(meter)

```
distance

0          0.000000
1          0.000000
2          0.000000
3          0.000000
4          0.000000
5          0.000000
6          0.000000
7          0.000000
8          0.000000
9          0.000000
10         0.000000
11         0.000000
12         0.000000
13         0.000000
14         0.000000
15         0.000000
16         0.000000
17         0.000000
18         0.000000
19         0.000000
20         0.000000
21         0.000000
22         0.000000
23         0.000000
24         0.000000
25         0.000000
26         0.000000
27         0.000000
28         0.000000
29         0.000000
           ...
755     1806.476111
756     1806.715000
757     1806.975278
758     1807.239444
759     1807.647500
760     1808.076667
761     1808.523056
762     1808.892500
763     1809.371944
764     1809.853611
765     1810.337500
766     1810.861944
767     1811.328611
768     1811.644167
769     1811.807500
770     1812.034722
771     1812.211667
772     1812.408611
773     1812.736944
774     1813.518889
775     1814.650556
776     1816.105278
777     1817.927500
778     1819.932222
779     1821.840278
780     1823.575556
781     1825.123611
782     1826.573333
783     1855.395556
784     1858.356944
```

speed

```
speed

0        None
1        None
2        None
3        None
4        None
5        None
6        None
7        None
8        None
9        None
10       None
11       None
12       None
13       None
14       None
15       None
16       None
17       None
18       None
19       None
20       None
21       None
22       None
23       None
24       None
25       None
26       None
27       None
28       None
29       None
         ...
755     0.461
756      0.86
757     0.937
758     0.951
759     1.469
760     1.545
761     1.607
762      1.33
763     1.726
764     1.734
765     1.742
766     1.888
767      1.68
768     1.136
769     0.588
770     0.818
771     0.637
772     0.709
773     1.182
774     2.815
775     4.074
776     5.237
777      6.56
778     7.217
779     6.869
780     6.247
781     5.573
782     5.219
783    10.376
784    10.661
```

time_stamp

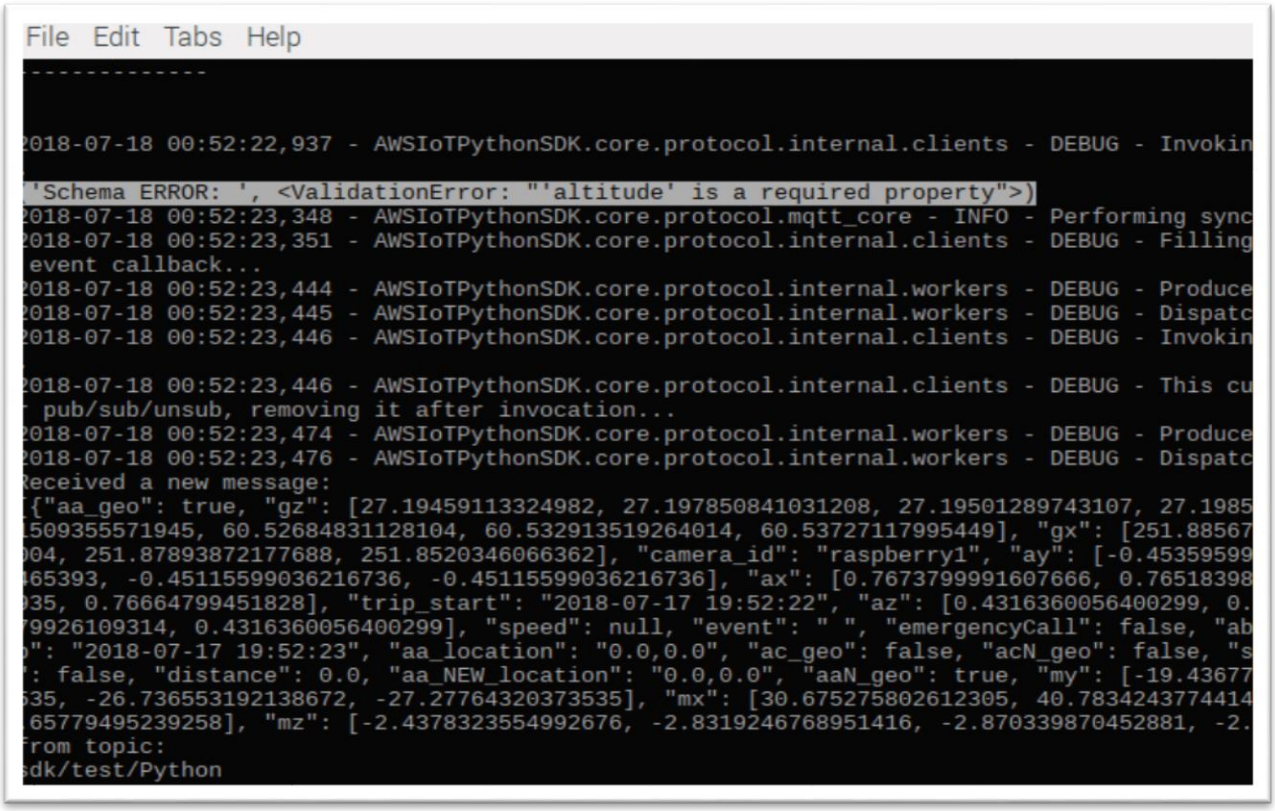
```
time_stamp

0      2018-07-17 16:52:13
1      2018-07-17 16:52:14
2      2018-07-17 16:52:15
3      2018-07-17 16:52:16
4      2018-07-17 16:52:17
5      2018-07-17 16:52:18
6      2018-07-17 16:52:19
7      2018-07-17 16:52:20
8      2018-07-17 16:52:21
9      2018-07-17 16:52:22
10     2018-07-17 16:52:23
11     2018-07-17 16:52:24
12     2018-07-17 16:52:25
13     2018-07-17 16:52:26
14     2018-07-17 16:52:27
15     2018-07-17 16:52:28
16     2018-07-17 16:52:29
17     2018-07-17 16:52:30
18     2018-07-17 16:52:31
19     2018-07-17 16:52:32
20     2018-07-17 16:52:33
21     2018-07-17 16:52:34
22     2018-07-17 16:52:35
23     2018-07-17 16:52:36
24     2018-07-17 16:52:37
25     2018-07-17 16:52:38
26     2018-07-17 16:52:39
27     2018-07-17 16:52:40
28     2018-07-17 16:52:41
29     2018-07-17 16:52:42
                  ...
755    2018-07-17 17:10:24
756    2018-07-17 17:10:25
757    2018-07-17 17:10:26
758    2018-07-17 17:10:27
759    2018-07-17 17:10:28
760    2018-07-17 17:10:29
761    2018-07-17 17:10:30
762    2018-07-17 17:10:31
763    2018-07-17 17:10:32
764    2018-07-17 17:10:33
765    2018-07-17 17:10:34
766    2018-07-17 17:10:35
767    2018-07-17 17:10:36
768    2018-07-17 17:10:37
769    2018-07-17 17:10:38
770    2018-07-17 17:10:39
771    2018-07-17 17:10:40
772    2018-07-17 17:10:41
773    2018-07-17 17:10:42
774    2018-07-17 17:10:43
775    2018-07-17 17:10:44
776    2018-07-17 17:10:45
777    2018-07-17 17:10:46
778    2018-07-17 17:10:47
779    2018-07-17 17:10:48
780    2018-07-17 17:10:49
781    2018-07-17 17:10:50
782    2018-07-17 17:10:51
783    2018-07-17 17:11:01
784    2018-07-17 17:11:02
```

All data here worked well, but we noticed that unit for distance is in meters and unit for speed is in miles. Therefore, we are going to update them.

5. jsonschema

We intentionally added the *altitude* attribute to check if our json schema works well. During our trip, we saw an error message about *altitude* as desired. We confirmed that the jsonschema validation works well. Therefore, we will get rid of the attribute.



6. Limitation

Week gps and Wifi signals
Huge time expense to setup RP
RP's Power off issues is because of the lower power voltage of the USB port (*raspberrypi.stackexchange.com*)

7. Future works

Reformatting sensor data to [min, avg, max]
Adding more 'event' cases
Updating units in distance and speed
Applying Job and VOD
Researching on heading attribute (we got values of EWNS)
Updating AWS lambda (our data preprocessing conducts on basicPubsub.py due to week wifi)