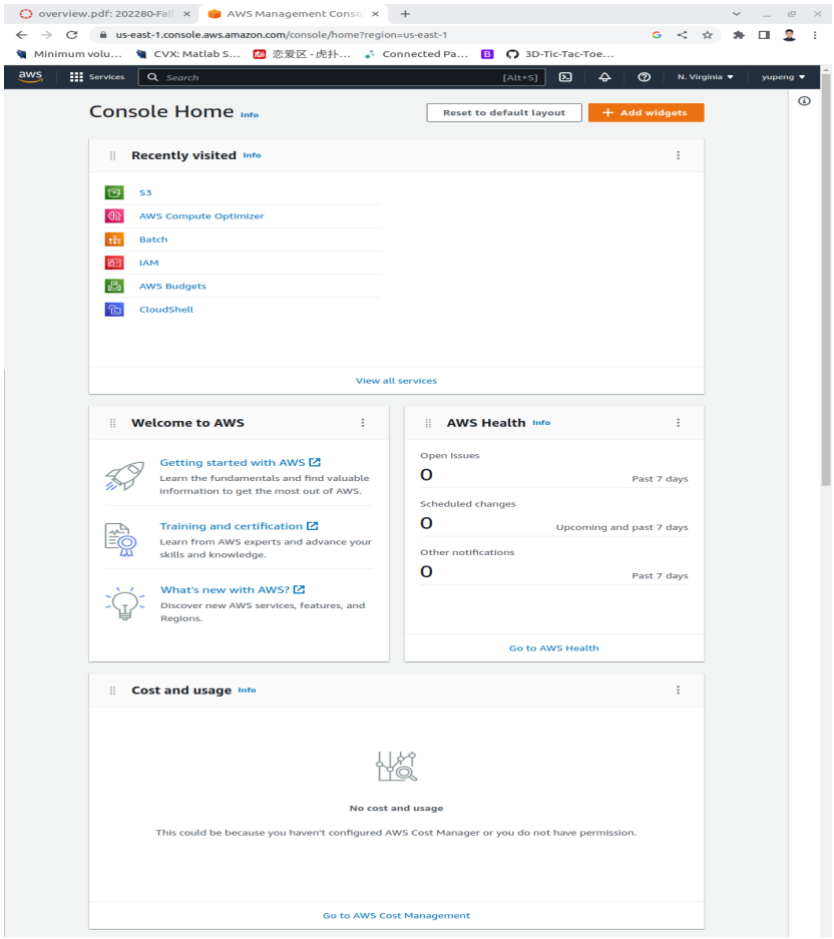
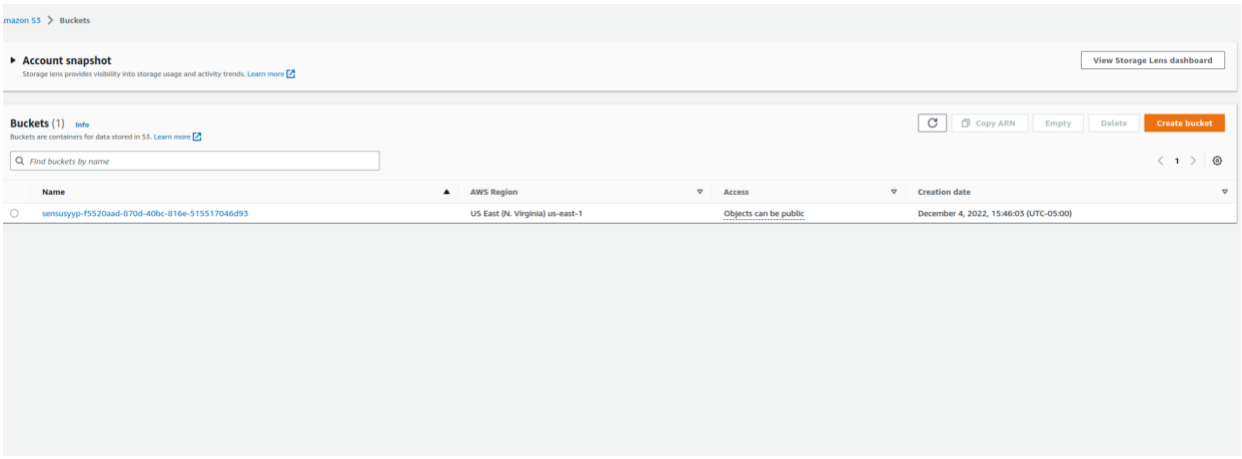
Final Report

1. Create the AWS bucket and share this with other group members: Amazon S3 bucket# sensusyyp-f5520aad-870d-40bc-816e-515517046d93

    a. We need to sign up for the AWS cloud service:

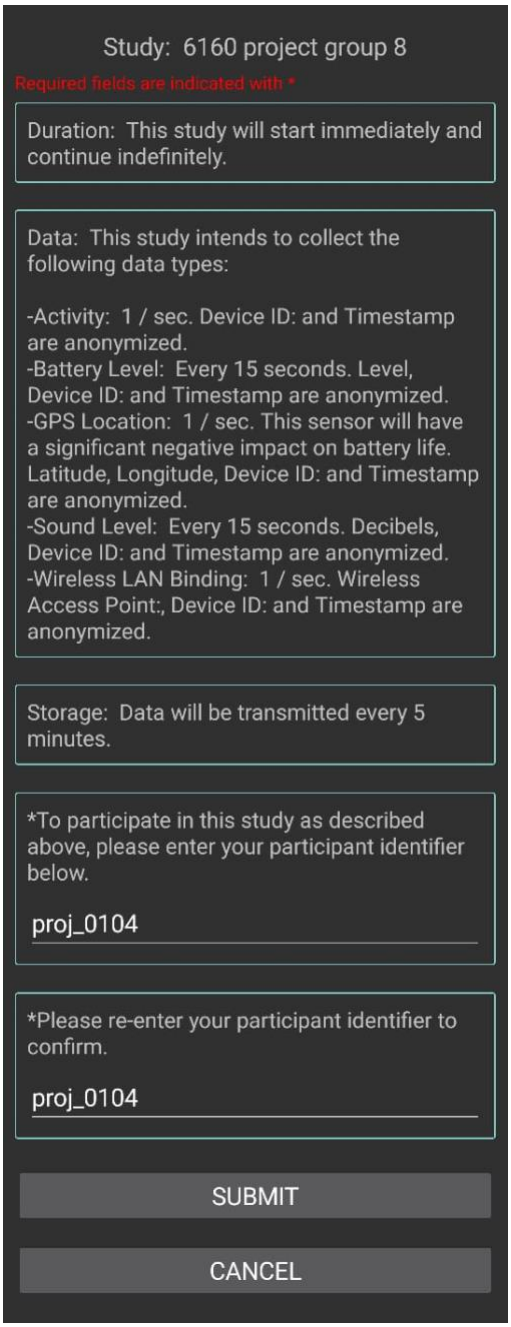


    b. Configure and create the bucket:

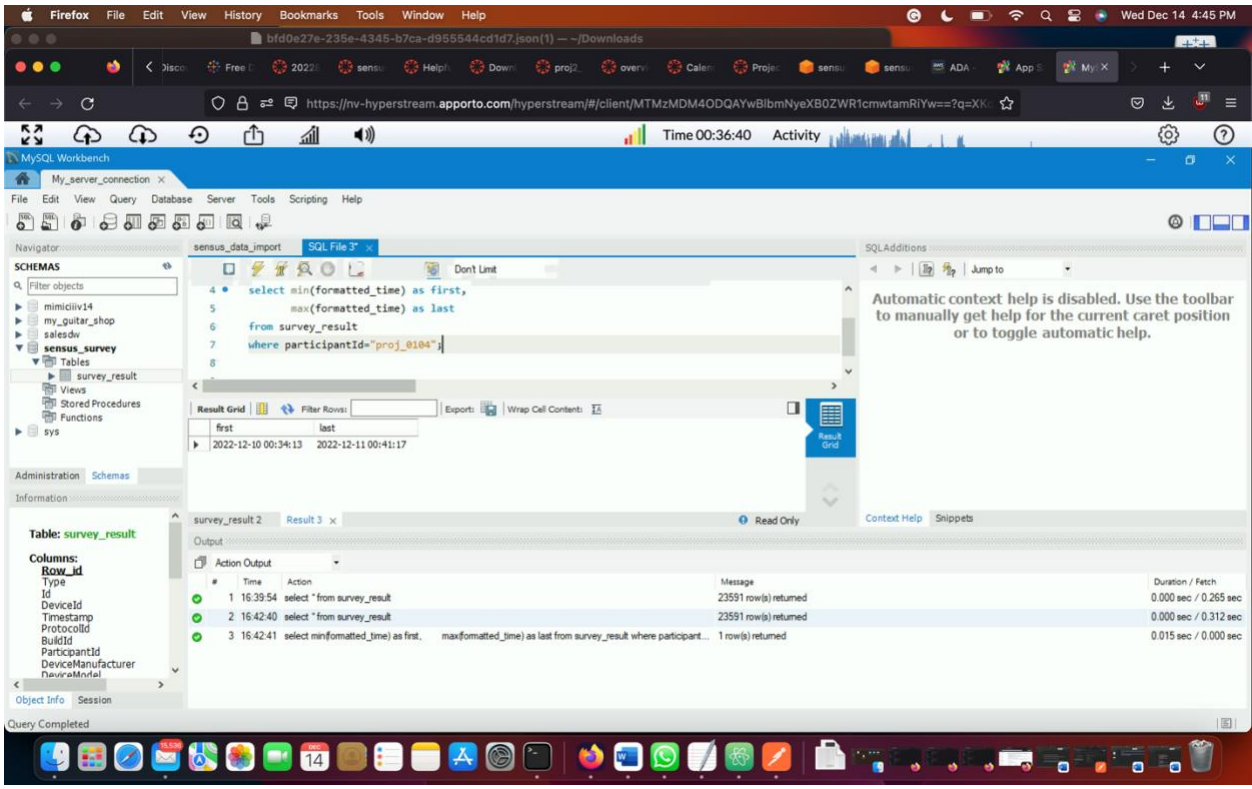


i. And our bucket name is: sensusyyp-f5520aad-870d-40bc-816e-515517046d93
ii. With the Sensus S3 IAM account: AKIAQPOLBEURN5OVLC6F:+izZe5ayLSK5+QHVIUcVT/tFJFqvhUUDb+EwamxG

2. Collect the data using following protocol (issue: can't set Gps location sampling at each 1 seconds and will talk about this in the later section):



3. Download the DATA from the bucket using the python script and pay attention to modifying the correct missing part. I will share my python script with this report.
4. Overall, of the study:
    a. The overall duration of this study is:
        SQL Query and result of First and Last Time Stamp:



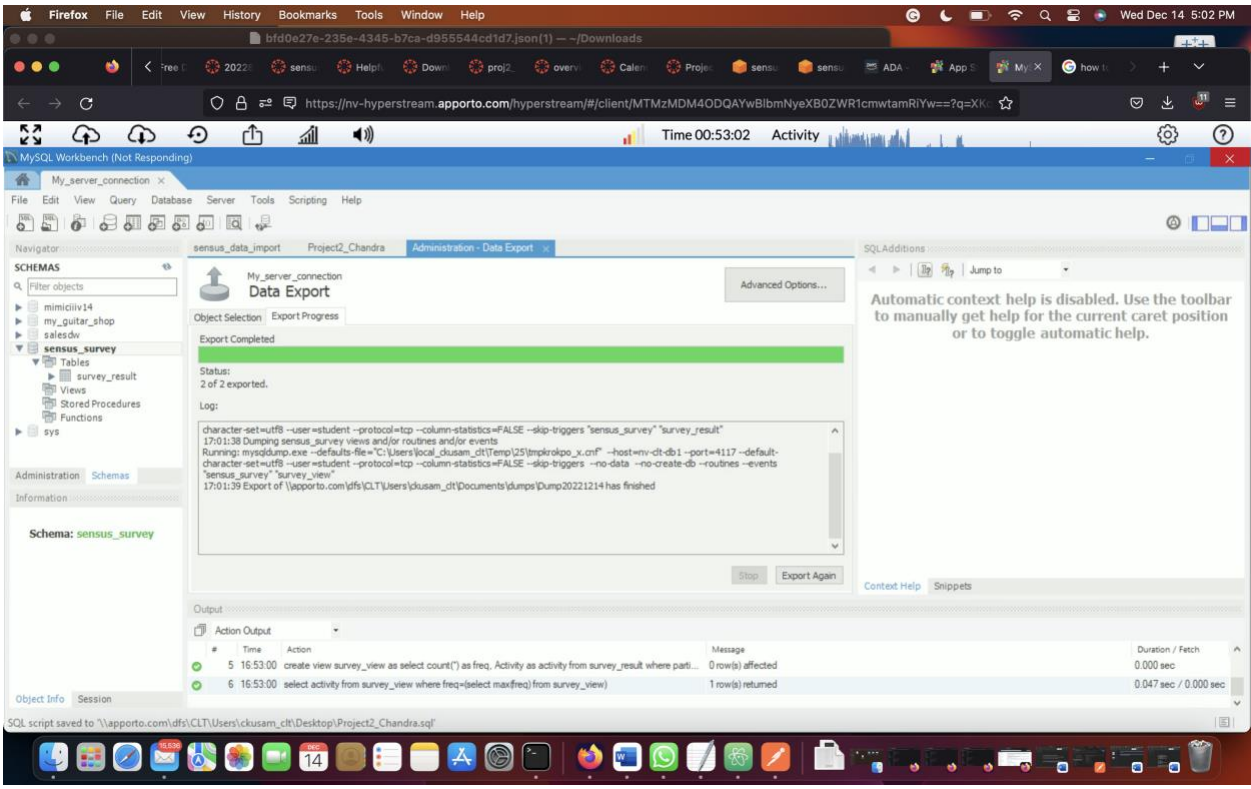
Python First and Last Time Stamp:

```
    print('The first timestamps of the record data is', data_pd.Formatted_time.min())
    print('The last timestamps of the record data is', data_pd.Formatted_time.max())
528]  ✓ 0.3s

··  The first timestamps of the record data is 2022-12-10 00:34:13
    The last timestamps of the record data is 2022-12-11 00:41:17
```

b. The total data size is 4.3 Mb for the output group8.csv file and 641.4 KB for the original dataset.

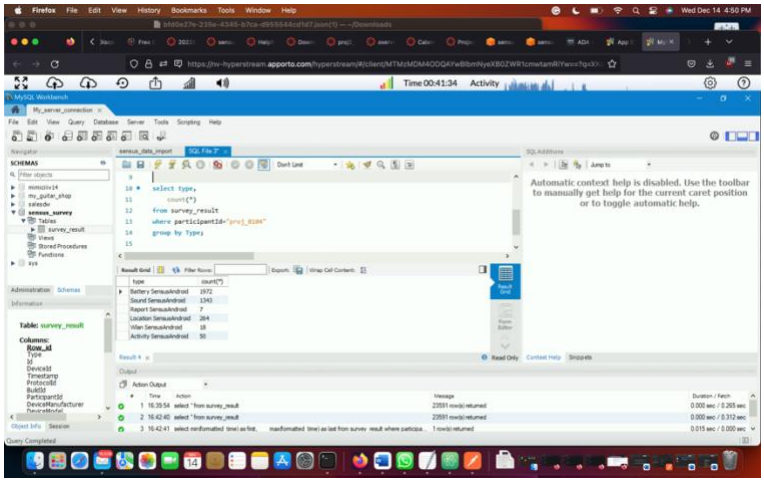
c. Contribution: Individual contribution worked helped with the creation of the report and importing the CSV file into MySQL workbench and helped to enter the participant Id previously we are not getting it and exported the SQL dump data.



5. Data summary:
    a. DeviceId is 0e5ae7d221cc8301, DeviceManufacturer is samsung, DeviceModel is m20lte, OperatingSystem is Android 29, and **participant number is ‘proj_0104’.**

6. Data Aggregation with python and SQL query and result:

    a. How many records are for each prob?



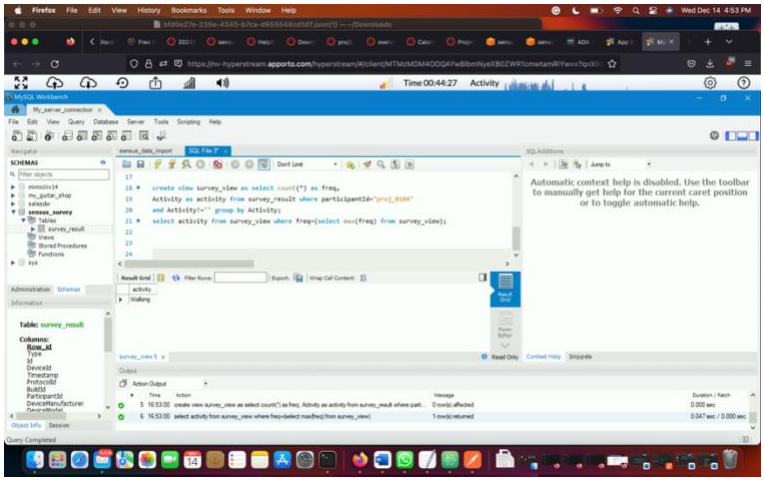
```
we have 3661 numbers of the record


we have collect the data based on the following type:
Battery SensusAndroid Sound SensusAndroid Report SensusAndroid Sensus.Heartbeat SensusAndroid Location SensusAndroid Wlan SensusAndroid Activity SensusAndroi

The detail number of records for each type is

Battery SensusAndroid             1972
Sound SensusAndroid               1343
Location SensusAndroid             264
Activity SensusAndroid              50
Wlan SensusAndroid                  18
Report SensusAndroid                 7
Sensus.Heartbeat SensusAndroid       7
Name: Type, dtype: int64
```

b. What is my favorite activity?



```
print('I have the following types of activity',activity.Activity.unique());
print('The detail frequency of each activity is');
print(activity.Activity.value_counts(1));
print('The most frequent activity is \'Walking\'');
print('it happens',activity.Activity.value_counts()[0],'times', 'and it takes',activity.Activity.value_counts(1)[0],'percentage');
#print('I am  a quite person')
```

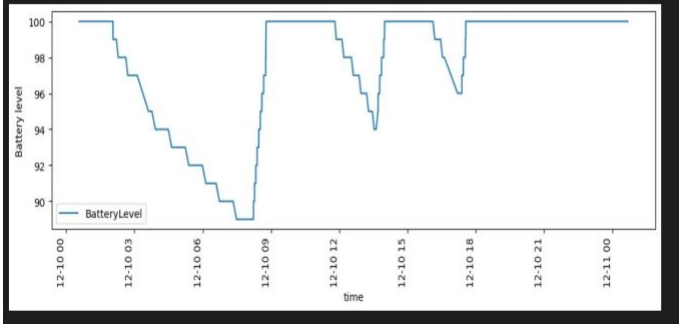
✓ 0.4s

```
I have the following types of activity ['Walking' 'OnFoot' 'Still' 'OnBicycle' 'Running' 'InVehicle' 'Unknown']
The detail frequency of each activity is
Walking      0.28
OnFoot       0.26
Still        0.22
OnBicycle    0.06
Running      0.06
InVehicle    0.06
Unknown      0.06
Name: Activity, dtype: float64
The most frequent activity is 'Walking'
it happens 14 times and it takes 0.28 percentage
```

c. Plot the Battery level history.

```
ax1 = Battery1.plot(x='Formatted_time',y='BatteryLevel',figsize=(12,4),rot=90); #rotate 90 degree for x label
ax1.set_ylabel('Battery level'); # set y label name
ax1.set_xlabel('time');   # set x label name
```

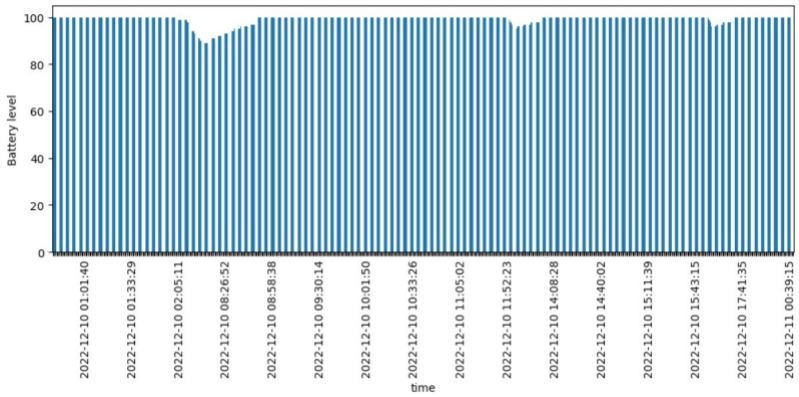
✓ 0.1s



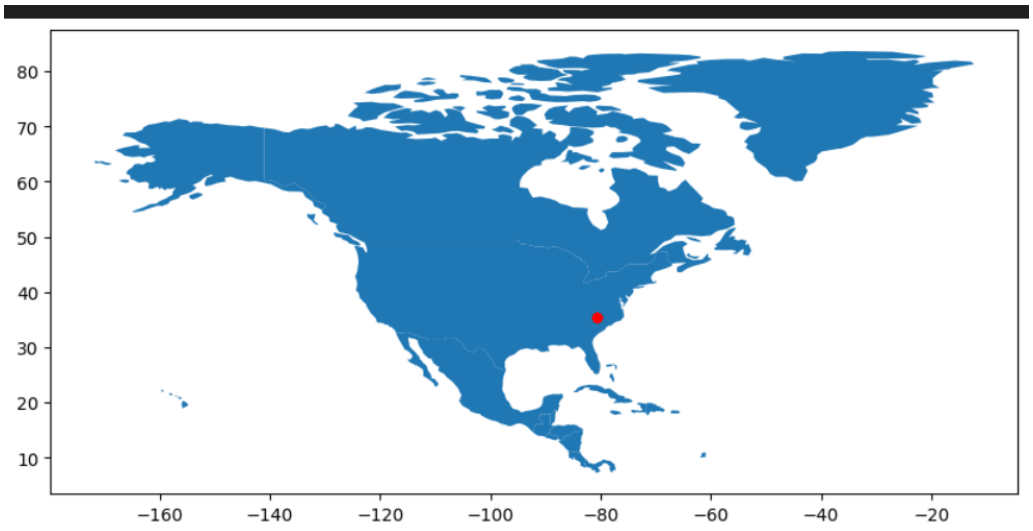

```
print("we have following columns for Battery data",Battery.columns,"we only need 'BatteryLevel' and 'Formatted_time'. Let's drop useless one ")
```

```
we have following columns for Battery data Index(['Row_id', 'Type', 'Id', 'DeviceId', 'Timestamp', 'ProtocolId',
       'BuildId', 'ParticipantId', 'DeviceManufacturer', 'DeviceModel',
       'OperatingSystem', 'TaggedEventId', 'TaggedEventTags',
       'SensingAgentStateDescription', 'LocalOffsetFromUTC', 'BatteryLevel',
       'Activity', 'Phase', 'State', 'Confidence', 'Latitude', 'Longitude',
       'LocationAccuracy', 'ProtocolName', 'Formatted_time'],
      dtype='object') we only need 'BatteryLevel' and 'Formatted_time'. Let's drop useless one
Here we get what we want Index(['BatteryLevel', 'Formatted_time'], dtype='object')
Here is the detailed information for Battery level
```

	BatteryLevel
count	1972.000000
mean	99.171400
std	2.140673
min	89.000000
25%	100.000000
50%	100.000000
75%	100.000000
max	100.000000



d. Let's see where are we.



7. Issues with mobile apps/cloud, whether they were resolved, and how (potentially how). While using sensusmobile even though given all permissions which were asked it doesn't catch the sound polling levels in some of our phones. Tried to solve this issue by deleting the existing study group and creating a new one and uninstalling the sensusmobile and installing it again.

Nothing seems to work for this issue. And if we take out the phone from charging it doesn’t submit the data every 5 minutes. To solve this issue, we need to keep our phone charging or disable the required charging option in the app.

8. Take Aways: 1. Using S3 Bucket, Writing SQL Queries and python visualization.