Assignment1, CMPT826

Step 1: Filter

- Seyedeh Mina Mousavifar
- 11279515
- sem311

Filtering users based on battery record

Obtaining battery record count per user

In [1]:
```
import pandas as pd

# reading data from pickle object
battery_data = pd.read_pickle('data/battery.pkl')

# counting number of battery information per user
battery_info = battery_data.groupby(['user_id']).size().reset_index(name
='record_count')

# finding number of users before filtering
all_users, _ = battery_info.shape
print('There is a total of', all_users, 'users.')
```

```
There is a total of 108 users.
```

Filtering users with less than 50% battery records.

Maximum battery record is $(60/5) * 24 * 30 = 8640$.

In [2]:
```
# calculating filtering cutoff
cutoff_percentage = 0.5
max_battery_info = (60 / 5) * 24 * 30
battery_cutoff = cutoff_percentage * max_battery_info

# filtering users with less than 50%
battery_info_50 = battery_info.loc[battery_info['record_count'] > batter
y_cutoff]
users_filter_50, _ = battery_info_50.shape

# preserving only users with more than 50% battery record
battery_data_50 = pd.merge(left=battery_data, right=battery_info_50,
                           left_on='user_id', right_on='user_id')

print(all_users - users_filter_50, 'users are removed in 50% threshold.'
)
```

```
65 users are removed in 50% threshold.
```

Filtering users with less than 75% battery records.

In [3]:
```
# calculating filtering cutoff
cutoff_percentage = 0.75
battery_cutoff = cutoff_percentage * max_battery_info

# filtering users with less than 75%
battery_info_75 = battery_info.loc[battery_info['record_count'] > batter
y_cutoff]
users_filter_75, _ = battery_info_75.shape

# preserving only users with more than 75% battery record
battery_data_75 = pd.merge(left=battery_data, right=battery_info_75,
                           left_on='user_id', right_on='user_id')

print(all_users - users_filter_75, 'users are removed in 75% threshold.'
)

# saving data frame as pickle object for future use
battery_data_50.to_pickle('data/battery_50.pkl')
battery_data_75.to_pickle('data/battery_75.pkl')
```

```
86 users are removed in 75% threshold.
```

> There is a total of 108 users, 65 users are removed in 50% threshold, and 86 users are removed in 75% threshold.

Filtering GPS records

Obtaining gps data

In [4]:
```
# reading data from pickle object
gps_data = pd.read_pickle('data/gps.pkl')

# finding number of all records
records_num, _ = gps_data.shape
print('There is a total of', records_num, 'GPS records.')
```

```
There is a total of 8592409 GPS records.
```

Filtering records with more than 100m accuracy.

In [5]:
```
# filtering accuracy more than 100
gps_data = gps_data.loc[gps_data['accu'] < 100]
gps_filter_accu, _ = gps_data.shape

print(records_num - gps_filter_accu,
      'records are removed for accuracies more than 100m.')
```

```
219886 records are removed for accuracies more than 100m.
```

Filtering records outside of latitude range.

Desired latitude is between (52.058367, 52.214608)

In [6]:
```
# outside latitude range
gps_data = gps_data.loc[gps_data['lat'] > 52.058366]
gps_filter_lat_low, _ = gps_data.shape

print(gps_filter_accu - gps_filter_lat_low,
      'records are removed for latitudes less than 52.058367')

gps_data = gps_data.loc[gps_data['lat'] < 52.214609]
gps_filter_lat_high, _ = gps_data.shape

print(gps_filter_lat_low - gps_filter_lat_high,
      'records are removed for latitudes more than 52.214608')
```

```
233849 records are removed for latitudes less than 52.058367
179810 records are removed for latitudes more than 52.214608
```

Filtering records outside of longitude range.

Desired longitude is between (-106.7649138128, -106.52225318)

In [7]:
```
# outside longitude range
gps_data = gps_data[gps_data['lon'] > -106.7649138128]
gps_filter_lon_low, _ = gps_data.shape

print(gps_filter_lat_high - gps_filter_lon_low,
      'records are removed for longitudes less -106.7649138128')

gps_data = gps_data.loc[gps_data['lon'] < -106.52225319]
gps_filter_lon_high, _ = gps_data.shape

print(gps_filter_lon_low - gps_filter_lon_high,
      'records are removed for latitudes more than -106.52225318')
```

```
867 records are removed for longitudes less -106.7649138128
1424 records are removed for latitudes more than -106.52225318
```

In [8]:
```
records_num, _ = gps_data.shape

print('GPS data size after filtering is', records_num, 'records.')
```

```
GPS data size after filtering is 7956573 records.
```

There are 8,592,409 records in GPS table. 635,836 records is being filtered after the filtering for GPS, so 7,956,573 records remain for GPS data.

Filtering Saskatoon data based on battery records

Obtaining saskatoon data with less than 100m accuracy and users with more than 50% battery records

In [10]:
```
# retrieving users with more than 50% battery info
user_battery = pd.read_pickle('data/battery_50.pkl')

records_num, _ = gps_data.shape

# creating dataframe for filtering Saskatoon data for preferred users
good_50_user_id = user_battery.user_id.unique()
gps_data_better50 = gps_data[gps_data.user_id.isin(good_50_user_id)]

gps_filter_50_battery, _ = gps_data_better50.shape
print(records_num - gps_filter_50_battery,
      'records are removed for users with more than 50% battery record
s.')
```

```
2280492 records are removed for users with more than 50% battery record
s.
```

In [11]:
```
# retrieving users with more than 75% battery info
user_battery = pd.read_pickle('data/battery_75.pkl')

records_num, _ = gps_data.shape

# creating dataframe for filtering Saskatoon data for preferred users
good_75_user_id = user_battery.user_id.unique()
gps_data_better75 = gps_data[gps_data.user_id.isin(good_75_user_id)]

gps_filter_75_battery, _ = gps_data_better75.shape
print(records_num - gps_filter_75_battery,
      'records are removed for users with more than 75% battery record
s.')
```

```
4624080 records are removed for users with more than 75% battery record
s.
```

2280492 records are removed for users with more than 50% battery records, and 4624080 records are removed for users with more than 75% battery records.

Applying filterings one after the other

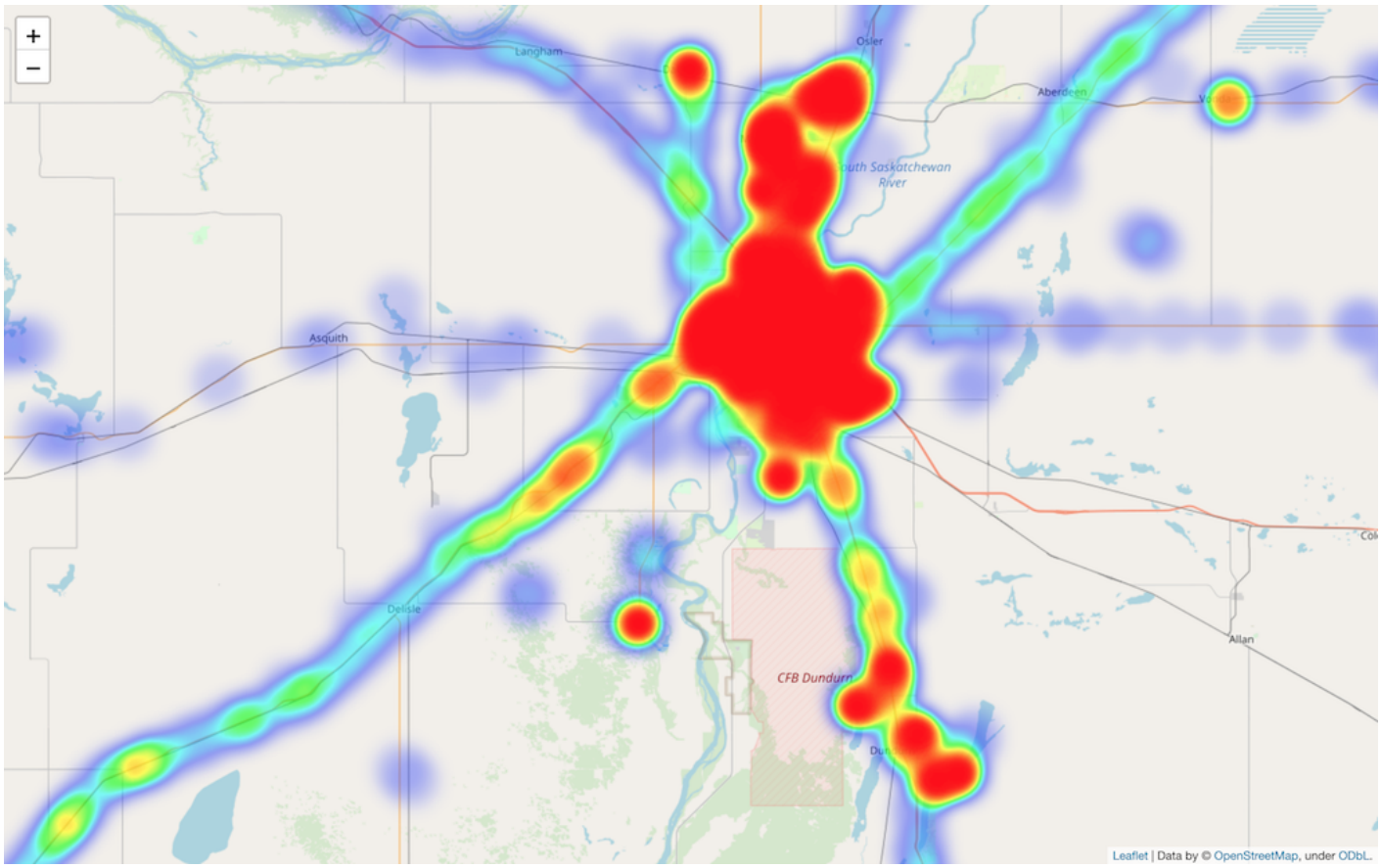
Filtering method	Records removed
Users with less than 50% battery records	2,280,492
Users with less than 75% battery records	4,624,080
Accuracy filtering	219,886
Latitude lower bound	233,849
Latitude upper bound	179,810
Longitude lower bound	867
Longitude upper bound	1424

Heatmap plotting

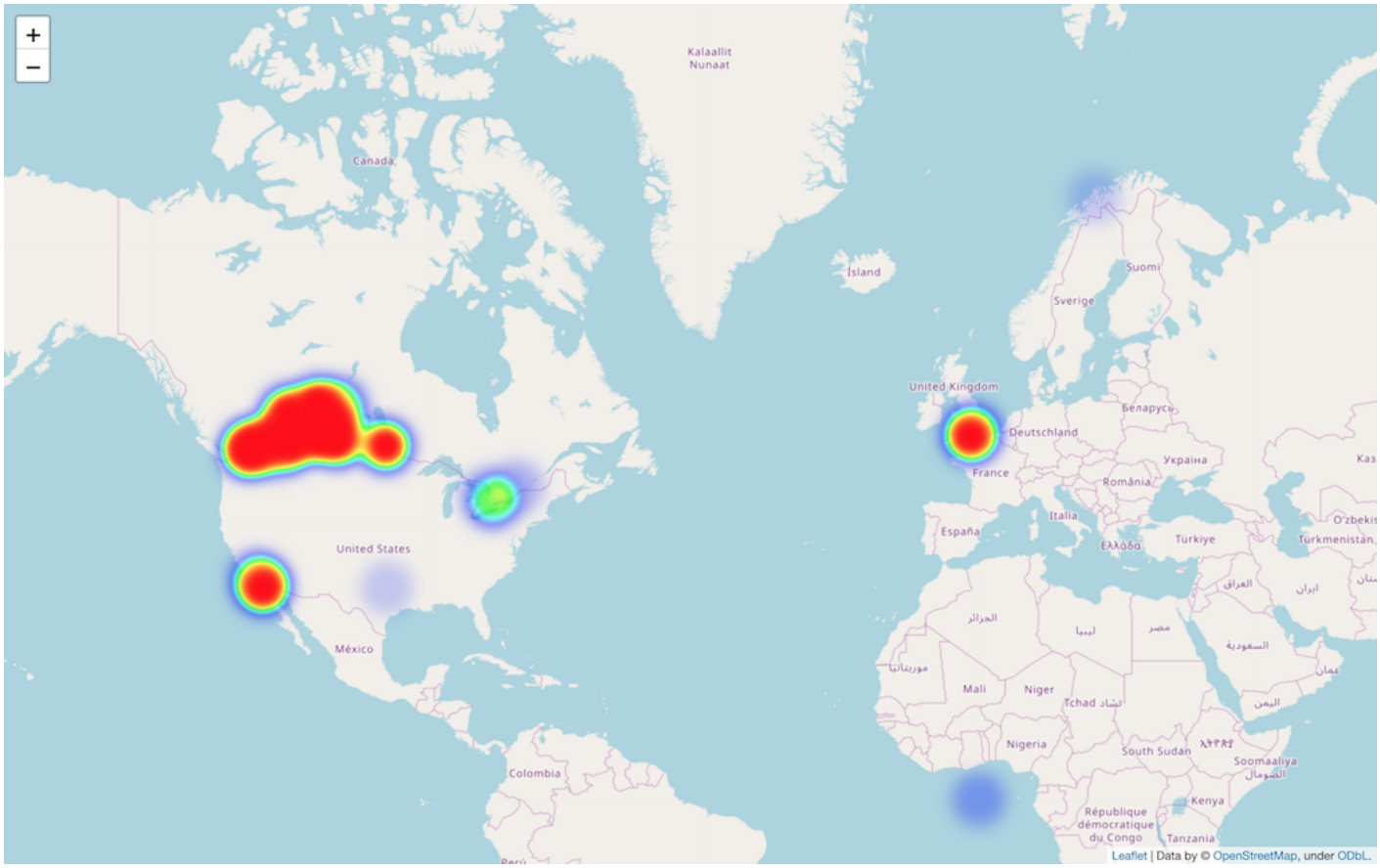
Plotting heatmap for all data

In [ ]:
```
import folium
from folium.plugins import HeatMap

# plotting heatmap for all records
gps_data = pd.read_pickle('data/gps.pkl')

# creating map
hmap_data = folium.Map(location=[52.058367, -106.7649138128])

hm_wide = HeatMap(list(zip(gps_data.lat.values, gps_data.lon.values, )),
min_opacity=0.2)

hmap_data.add_child(hm_wide)

hmap_data.save(os.path.join('maps', 'all_records_heatmap.html'))
```

Heatmap of GPS record of all data viewing Saskatoon



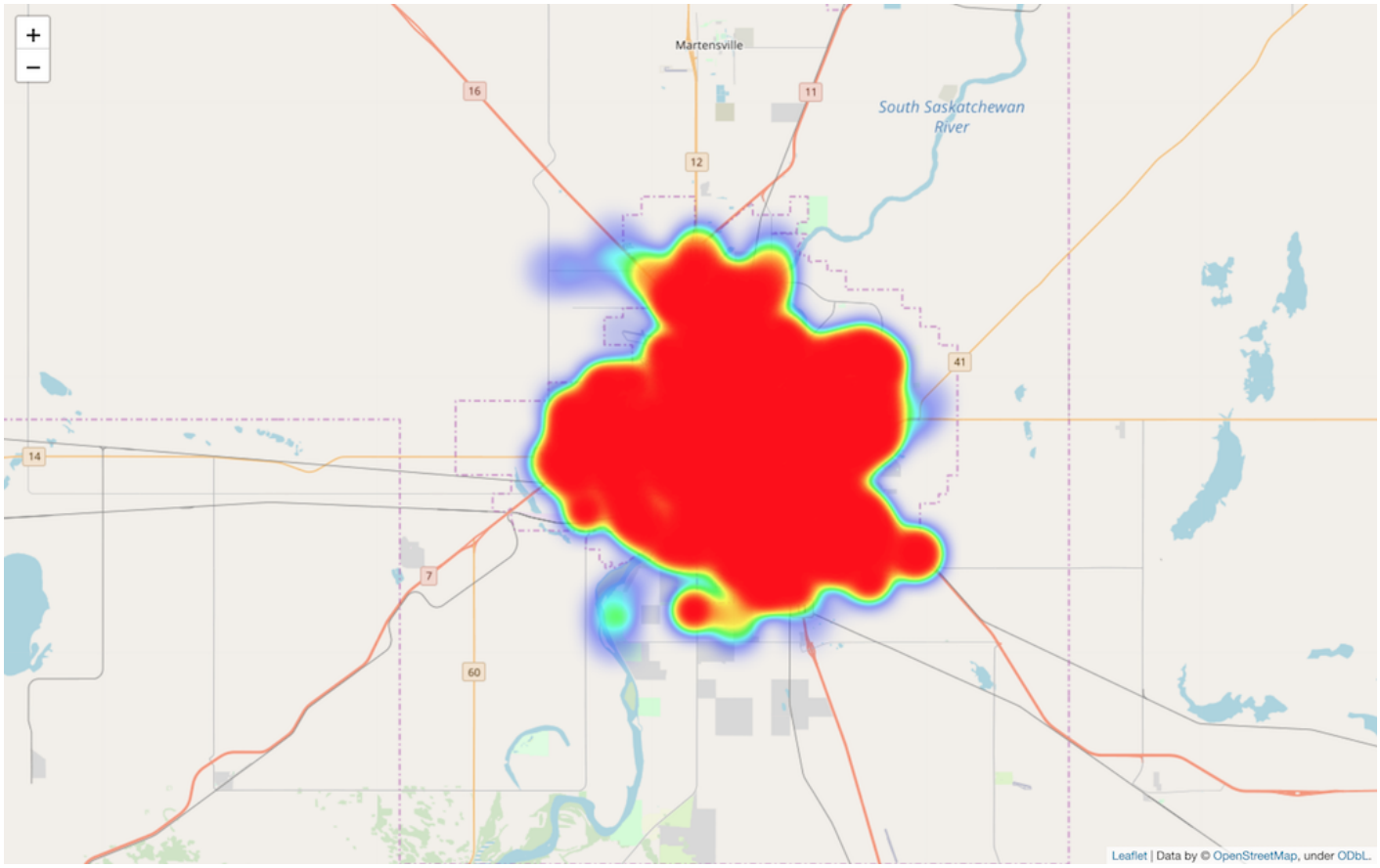
Heatmap of GPS record of all data viewing World



Plotting heatmap for filtered data

In [ ]:
```
# plotting heatmap for filtered records
gps_data = pd.read_pickle('data/gps_filter.pkl')

# creating map
hmap_data = folium.Map(location=[52.058367, -106.7649138128])

hm_wide = HeatMap(list(zip(gps_data.lat.values, gps_data.lon.values)), m
in_opacity=0.2)

hmap_data.add_child(hm_wide)

hmap_data.save(os.path.join('maps', 'filtered_records_heatmap.html'))
```

Heatmap of GPS record of filtered data viewing Saskatoon



Are there any suspicious locations?

> As we can see, there are records in locations such as other parts of Canada, the US, and the UK. These might be users who travelled during the term, primarily via airplane, that we don't see any trail of their vacation path. Furthermore, there are some suspicious points in the Atlantic ocean near Nigeria and the Norwegian sea, which are incorrect data.

How would you have filtered them if you had not filtered for Saskatoon?

> One approach to filter these points rather than longitude is to define ranges of latitude and longitude, and count number of records in these ranges and remove records with less than a certain count threshold in their specified range.