Trip summaries

Don Li

12/06/2020

```
library( data.table )
```

Summarise trip trajectories

List of covariates:

- `crow_dist`: Distance as the crow flies (Haversine, km). Numeric.
- `path_dist`: Path distance (Haversine, km). Numeric.
- `path_dist2`: Path distance; longitude and latitude reversed (Haversine, km). Numeric.
- `timediff`: Observed arrival time. Numeric.
- `start_x`, `start_y`: Journey start longitudes and latitudes. Numeric.
- `end_x`, `end_y`: Journey end longitudes and latitudes. Numeric.
- `weekday`, `hour`: Day and hour that the trip started. Factor; Numeric.
- `rush_hour`: Whether trip started during rush hour. Factor.
- `mean_speed`, `var_speed`: Mean and variance of speed. Numeric.
- `sampling_rate`, `sampling_rate_var`: GPS sampling rate. Numeric.

Covariates to be joined later:

- `azure_dist`: Path distance from Azure Maps.
- `OSRM_dist`: Path distance from OSRM.
- `trip_start`, `trip_end`: A factor with levels `generic`, `CX`, `CY`, etc. These are landmarks. `generic` catches all other points.

```
source( "G:/azure_hackathon/data/Don2/trip_summary.R" )
source( "G:/azure_hackathon/data/Don2/distance_functions.R" )
load( "G:/azure_hackathon/datasets2/data_processing/all_data3_speed.RData" )
trip_summary = sumamrise_trips( all_data )
```

Combine external distances/times from OSRM and Azure Maps

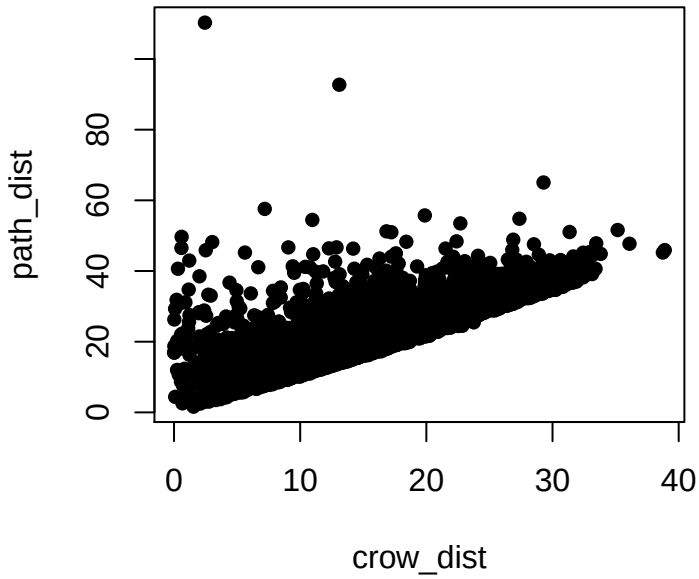
Join Azure and OSRM distances to the summaries.

```
load( "G:/azure_hackathon/datasets2/external_paths/external_dist.RData" )

trip_summary[ external_distance_summary,
    c("azure_dist", "OSRM_dist") :={
        list( i.azure_dist, i.OSMR_dist )
    },
    on = "trj_id"]

save( trip_summary,
    file = "G:/azure_hackathon/datasets2/trip_summary/trip_summary1_externaldist.RData" )
```

Loopy trips

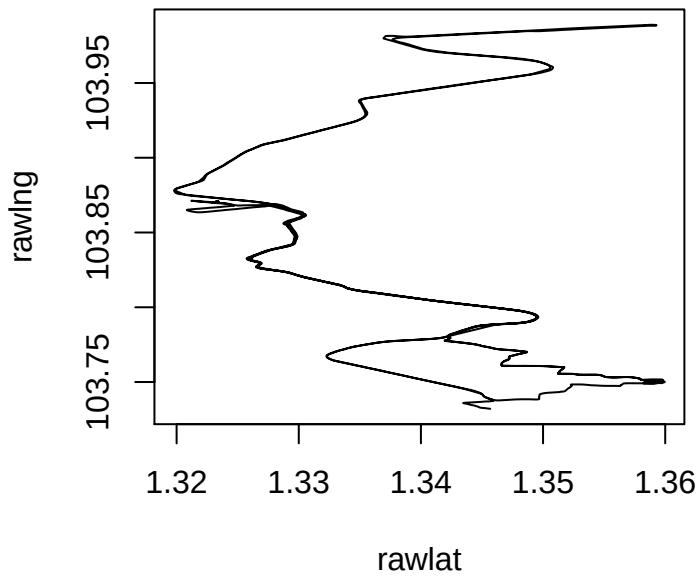
A trip where they just loop around the city. Obviously, we cannot predict the ETA of these kinds of trips using only the origin and the destination. There are some trips with very long path distances but short crow distances (top left).

```
load( "G:/azure_hackathon/datasets2/trip_summary/trip_summary1_externaldist.RData" )
load( "G:/azure_hackathon/datasets2/data_processing/all_data3_speed.RData" )

trip_summary[ , {
    plot( crow_dist, path_dist, pch = 16 )
} ]
```




```
## NULL
```

```
loopiest_trip = trip_summary[ which.max(path_dist - crow_dist) ]
all_data[ trj_id == loopiest_trip$trj_id, {
    plot( rawlat, rawlng, type = "l" )
    } ]
```




```
## NULL
```

I thought about taking them out. But these trips contain information about driver behaviour that deviates from the shortest path distance. So, I think it is best to keep them in.

Add landmark data

Using the results from our clustering exercise, we have landmarks for `trip_start` and `trip_end`. `generic` is the factor level where the trip does not start/end from one of our identified landmarks.

```
load( "G:/azure_hackathon/datasets2/landmarks/deriving_landmarks_dbscan.RData" )
add_landmarks( trip_summary, big_clusters )
```

Save the file

```
save( trip_summary,
    file = "G:/azure_hackathon/datasets2/trip_summary/trip_summary2_landmark.RData" )
```