Software Name:

TrajAnalytics / Data Loading Software

Version:

Local Version

Date:

July 16, 2018

Introduction:

TrajAnalytics / Data Loading Software allows user to upload and use their own GPS data to create TrajAnalytics database that can be visualized using local version of TrajAnalytics/ Visualization System. This software includes three packages as the following:

1. **Load GPS Data:** to load only GPS data.
2. **Load GPS Data with Road Network:** to load GPS data with road network data.
3. **Load GPS Data with Regions:** to load GPS data with regions boundary data.

Each package includes one Python code, one configuration file, and one or more sample data folders.

Software Usage Requirements:

1. This software includes Python codes, to be able to run them, you need to install any version of Python 2.7.
2. This software is based on some Python packages like os, sys, json, psycopg2, psycopg2.extensions, Tkinter, ttk, tkFileDialog, and tkMessageBox. Some of these packages come directly with Python 2.7, but you need to install the others. Installing Python packages is very easy by using pip installer program. For example, if you want to install psycopg2 package, you will write (pip install psycopg2) into your terminal (for Mac users) or your command line (for windows user). If you forget to install any of the required packages, the software will issue an import error to tell you to install the missed package, after installing the missed package you can run the software again.

3. This software will create TrajAnalytics database inside PostgreSQL9.5, so to run this software properly you need to install PostgreSQL version 9.5 or newer one.
4. Inside each package there is one configuration file called config.json to run the software properly, you need to set up the configuration parameters inside this file like the following:

{
"host": "localhost",
"dbname": "postgres",
"user": "**Your PostgreSQL user name**",
"password": "**Your PostgreSQL password**",
"TrajUserName": "**Your TrajAnalytics user name**",
"Trajdbname": "**Your TrajAnalytics database name**"
}

Note: You can only change the bold text. Please use only lower case letters and do not use any space or any other characters. See the example in Figure 1.

```
{
  "host": "localhost",
  "dbname": "postgres",
  "user": "postgres",
  "password": "user",
  "TrajUserName":"test",
  "Trajdbname":"mydatabase"
}
```

Figure 1: Example of config.json file.

Load GPS Data Package:

General Usage Notes:

1. This packages contains one Python code called Load_GPS_Data.py, one configuration file called config.json, and one sample data folder. Before running the code you need to set up the configuration parameters like

what is mentioned in step 4 from the section of the Software Usage Requirements.

2. You can use this code to upload your GPS trajectory data as CSV file to create TrajAnalytics database that can be used by the local version of TrajAnalytics/ Visualization System.
3. This software takes only the GPS trajectory data file as input.

Input and Output Format:

4. The **CSV** file of GPS trajectory data contains the sampling points of the trajectories. Each sampling point has the following dimensions: Trip ID, Date/Time, Latitude, Longitude, and Speed. Your GPS trajectory file must be exactly like the example in Figure 2. Please check **Data.csv** in the **Sample Data** folder of this package to make sure that your file has the accepted format.

```
tripid,pdatetime,latitude,longitude,speed
282,2013-07-01 02:58:14,41.152,-8.60586,2.92838383994976
282,2013-07-01 02:58:29,41.1521,-8.60592,21.2670026453991
282,2013-07-01 02:58:44,41.1522,-8.60697,36.3501445698443
282,2013-07-01 02:58:59,41.1509,-8.60751,22.9627538013043
282,2013-07-01 02:59:14,41.1504,-8.60844,30.465762877744
282,2013-07-01 02:59:29,41.1508,-8.60986,0.401892079728533
282,2013-07-01 02:59:44,41.1508,-8.60988,17.6540286680847
282,2013-07-01 02:59:59,41.1514,-8.60951,38.6371062879999
282,2013-07-01 03:00:14,41.1528,-8.61,43.5248440664887
282,2013-07-01 03:00:29,41.1544,-8.60958,35.6575161062409
282,2013-07-01 03:00:44,41.1557,-8.60917,56.2162611497052
282,2013-07-01 03:00:59,41.1578,-8.60895,51.0494566825696
282,2013-07-01 03:01:14,41.1596,-8.60809,98.2071114391681
282,2013-07-01 03:01:29,41.1631,-8.60658,58.9427859605568
282,2013-07-01 03:01:44,41.1653,-8.60632,58.7603489999198
282,2013-07-01 03:01:59,41.1675,-8.60644,56.9352403425285
282,2013-07-01 03:02:14,41.1696,-8.60594,59.0298257387252
282,2013-07-01 03:02:29,41.1713,-8.60406,65.5049442164509
282,2013-07-01 03:02:44,41.1711,-8.60081,67.145411090051
```

Figure 2: Example of CSV file of GPS trajectory data.

5. In Case you do not have your own data, you can use the **Data.csv** from the **Sample Data** folder.
6. The output of this software is trajAnalytics database in PostgreSQL database system. The name of this database will be like (**traja_Your TrajAnalytics database name _1_1_ Your TrajAnalytics user name**),

for example **traja_ mydatabse _1_1_ test**. This database can be used as input to the local version of TrajAnalytics /Visualization System. The TrajAnalytics database is spatial database that contains the spatial extension to implement spatial quires. In addition to that, this database contains two main tables: one table contains the GPS trajectory data and table called TD, that contains the trip information in the TrajAnalytics format.

Load GPS Data with Road Network Package:

General Usage Notes:

1. This packages contains one Python code called Load_GPS_Data_With_Road_Network.py, one configuration file called config.json, and sample data folder. Before running the code you need to set up the configuration parameters like what is mentioned in step 4 from the section of the Software Usage Requirements.
2. You can use this code to upload your GPS trajectory data as CSV file, and road network data as JSON file (GeoJSON format) to create TrajAnalytics database that can be used by the local version of TrajAnalytics/ Visualization System.
3. This software takes two files as input, the GPS trajectory data and the road network data.
4. The GPS trajectory data and the road network data must be related to each other. They must belong to same bounding box (same area). Your GPS trajectory data must be mapped to the roads of the uploaded road network before using this software.

Input and Output Format:

5. The **CSV** file of GPS trajectory data contains the sampling points of the trajectories. Each sampling point has the following dimensions: Trip ID, Date/Time, Latitude, Longitude, Speed, Road ID, and Road Type. Your GPS trajectory file must be exactly like the example in Figure 3. Please check **Data_RN.csv** in the **Sample Data** folder of this package to make sure that your file has the accepted format.
6. The JSON file of the road network data contains the road segments of the road network data. This file must be in **GeoJSON** format. For each road

segment must have information like in Figure 4. Please check **OSM_Roads_Car.json** in the **Sample Data** folder of this package to make sure that your file has the accepted format.

7. In Case you do not have your own data, you can use the **Data_RN.csv** and **OSM_Roads_Car.json** from the **Sample Data** folder.
8. The output of this software is trajAnalytics database in PostgreSQL database system. The name of this database will be like (**traja_Your TrajAnalytics database name _1_1_ Your TrajAnalytics user name**), for example **traja_ mydatabse _1_1_ test**. This database can be used as input to the local version of TrajAnalytics /Visualization System. The TrajAnalytics database is spatial database that contains the spatial extension to implement spatial quires. In addition to that, this database contains three tables: one table contains the GPS trajectory data, another table contains the road network data, and the TDS table that contains the trip information in the TrajAnalytics format.

```
tripid,pdatetime,latitude,longitude,speed,rid,highway
289,2013-07-01 02:56:54,41.1807,-8.62173,82.7422816227526,219537590,primary
289,2013-07-01 02:57:09,41.1799,-8.62571,66.7646596138376,224644820,primary
289,2013-07-01 02:57:24,41.1796,-8.62901,8.46593547303371,142699186,primary_link
289,2013-07-01 02:57:39,41.1795,-8.62941,2.81201269648338,569491087,primary
289,2013-07-01 02:57:54,41.1795,-8.62955,33.3303646095659,224644820,primary
289,2013-07-01 02:58:09,41.1783,-8.62909,22.6542311665647,13714843,secondary
289,2013-07-01 02:58:24,41.1784,-8.62797,41.9427549307907,274187613,residential
289,2013-07-01 02:58:39,41.179,-8.62604,40.8311446809905,274187613,residential
289,2013-07-01 02:58:54,41.1796,-8.62417,21.2092063910977,274187613,residential
289,2013-07-01 02:59:09,41.1789,-8.62367,17.0697076150275,149939905,residential
289,2013-07-01 02:59:24,41.1784,-8.6242,1.0043071155841,149939905,residential
289,2013-07-01 02:59:39,41.1784,-8.62425,0.200861423116822,149939905,residential
289,2013-07-01 02:59:54,41.1784,-8.62426,0.200861423116822,149939905,residential
289,2013-07-01 03:00:09,41.1784,-8.62425,0.200861423116822,149939905,residential
290,2013-07-01 03:25:33,41.178,-8.65543,112.016691289988,13258510,motorway
290,2013-07-01 03:25:48,41.1811,-8.65919,116.153191972258,13258526,motorway
293,2013-07-01 04:14:44,41.1784,-8.65579,94.3084080912742,13258526,motorway
293,2013-07-01 04:14:59,41.181,-8.65897,95.1269864801047,13258526,motorway
293,2013-07-01 04:15:14,41.1836,-8.66221,94.4411079575586,13258526,motorway
```

Figure 3: Example of CSV file of GPS trajectory data.

```
{
  "type": "FeatureCollection",
  "features":
  [
   {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            -8.6409147,
            41.1674513
          ],
          [
            -8.6408185,
            41.1673868
          ],
          [
            -8.6407269,
            41.1672831
          ],
          [
            -8.6406898,
            41.1671886
          ],
          [
            -8.6407025,
            41.167052
          ],
          [
            -8.6407815,
            41.166907
          ],
          [
            -8.6409089,
            41.1668122
          ]
        ]
      },
      "properties": {
        "lanes": "2",
        "oneway": "yes",
        "osm_id": 4256495,
        "highway": "motorway_link"
      }
    }
  ]
}
```


Figure 4: GeoJSON file contains information for one road segment.

Load GPS Data with Regions Package:

General Usage Notes:

1. This packages contains one Python code called Load_GPS_Data_With_Regions.py, one configuration file called config.json, and two sample data folders. Before running the code, you need to set up the configuration parameters like what is mentioned in step 4 from the section of the Software Usage Requirements.
2. You can use this code to upload your GPS trajectory data as CSV file, and regions boundary data as JSON file (GeoJSON format) to create TrajAnalytics database that can be used by the local version of TrajAnalytics/ Visualization System.
3. This software takes two files as input, the GPS trajectory data and the regions boundary data.
4. The GPS trajectory data and the regions data must be related to each other. They must belong to same bounding box (same area). Your GPS trajectory data must be mapped to the regions of the uploaded regions data before using this software.

Input and Output Format:

5. The **CSV** file of GPS trajectory data contains the sampling points of the trajectories. Each sampling point has the following dimensions: Trip ID, Date Time, Latitude, Longitude, Speed, and Region ID. Your GPS trajectory file must be exactly like the example in Figure 5. Please check **Data_RE.csv** in the **Sample Data ZipCode Regions** folder of this package to make sure that your file has the accepted format.
6. The JSON file of regions data contains the regions boundary. This file must be in **GeoJSON** format. For each region has information like in Figure 6. Please check **ZipRegions.json** in the **Sample Data ZipCode Regions** folder of this package to make sure that your file has the accepted format.
7. In Case you do not have your own data, you can use the **Data_RE.csv** and **ZipRegions.json** from the **Sample Data ZipCode Regions** folder or the files from the **Sample Data Grid Regions** folder
8. The output of this software is trajAnalytics database in PostgreSQL database system. The name of this database will be like (**traja_Your TrajAnalytics database name _1_1_ Your TrajAnalytics user name**),

for example **traja_ mydatabse _1_1_ test**. This database can be used as input to the local version of TrajAnalytics /Visualization System. The TrajAnalytics database is spatial database that contains the spatial extension to implement spatial quires. In addition to that, this database contains three main tables: one table contains the GPS trajectory data, another table contains the regions boundary data, and the TDR table that contains the trip information in the TrajAnalytics format.

```
tripid,pdatetime,latitude,longitude,speed,rid
35,2016-01-01 00:26:34,40.8487319946289,-73.9138107299805,0,1429
64,2016-01-01 00:39:15,40.8571853637695,-73.9043807983398,0,1429
237,2016-01-01 00:42:11,40.8486862182617,-73.91162109375,0,1429
286,2016-01-01 00:42:20,40.8491592407227,-73.9220199584961,0,1429
312,2016-01-01 00:43:22,40.8600082397461,-73.9098968505859,0,1429
320,2016-01-02 00:23:03,40.8451614379883,-73.9243545532227,0,1429
337,2016-01-01 00:29:03,40.8553161621094,-73.9036636352539,0,1429
535,2016-01-01 00:45:27,40.8521614074707,-73.9107360839844,0,1429
606,2016-01-01 00:45:59,40.8522491455078,-73.920295715332,0,1429
655,2016-01-01 00:45:26,40.8460464477539,-73.9119644165039,0,1429
854,2016-01-01 00:49:15,40.847110748291,-73.9203338623047,0,1429
56,2016-01-01 00:39:58,40.7737808227539,-73.9362411499023,0,1741
93,2016-01-01 00:39:55,40.7683868408203,-73.9245986938477,0,1741
100,2016-01-01 00:25:53,40.7666969299316,-73.9215621948242,0,1741
115,2016-01-01 00:40:51,40.7758674621582,-73.9290161132813,0,1741
166,2016-01-01 00:41:38,40.7704086303711,-73.9234924316406,0,1741
209,2016-01-01 00:41:32,40.775634765625,-73.9350814819336,0,1741
225,2016-01-01 00:41:54,40.7686614990234,-73.9270324707031,0,1741
260,2016-01-01 00:42:37,40.7694129943848,-73.9268112182617,0,1741
```

Figure 5: Example of CSV file of GPS trajectory data.

```
{
  "type": "FeatureCollection",
  "features":
  [
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              -8.68912,
              41.1233
            ],
            [
              -8.68912,
              41.1277951968863
            ],
            [
              -8.68315262813425,
              41.1277951968863
            ],
            [
              -8.68315262813425,
              41.1233
            ],
            [
              -8.68912,
              41.1233
            ]
          ]
        ]
      },
      "properties": {
        "r_id": 1
      }
    }
  ]
}
```


Figure 6: GeoJSON file contains information for one region.