# grid2demand

Gird2demand is an open-source trip generation and distribution tool for teaching transportation planning and applications. It generates zone-to-zone travel demand based on alphanumeric grid zones. Users can obtain zone-to-zone travel demand with a few lines of python code based on OpenStreetMap and OSM2GMNS.

## Contents

- **Introduction**
- **What is grid2demand?**
- **Quick Start**

For the python source code and sample network files, readers can visit the project homepage at ASU Trans+AI Lab Github (https://github.com/asu-trans-ai-lab/grid2demand).

## I. Introduction and Background Knowledge

Trip generation and trip distribution are the first 2 steps in the larger context of the 4-step process in transportation planning. The standard four steps are briefly described below.

- Trip Generation: Estimate how many trips entering or leaving a zone/traffic-analysis-zone (TAZ)
- Trip Distribution: Estimate how many trips from each zone/TAZ end in all zones/TAZs
- Mode Choice: Estimate which travel-method is used (e.g., vehicle, transit, walk)
- Traffic Assignment: Distribute vehicles/traffic flow to different paths during travel

Trip generation is a procedure that uses socioeconomic data (e.g., household size, income, etc.) to estimate the number of person trips for a modeled time period (e.g., daily, peak hour) at a Traffic Analysis Zone (TAZ) level. A person trip involves a single person leaving from an origin and arriving at a single destination, and each trip has a classification/purpose, e.g., based on classification such as home-based (HB) or non-home-based (NHB). The typical purposes include work (HBW), shopping (HBS), school (HBSc), other (HBO).

In the four-step process, there are two typical methods used to predict trips based on attributes:

- Trip rate method based on regression equations
- Cross-classification using category-based trip rates

After estimating the total number of trips produced, the trips are often separated by different purposes (e.g., HBW, HBO, NHB).

An alternative approach to modeling trips is to model tours, which can be thought of as a series of linked trips. Tours are typically used in Activity-Based Models (ABM), where daily travel activities are generated based on activity patterns for households.

## Productions and Attractions

In trip-based transportation planning, for a home-based trip, a production is related to the home end/location, while an attraction is related to non-home end/location. For a non-home-based trip, a production is related to the origin location, and an attraction is related to the destination location. Entering and leaving trips should balance - if a person leaves a zone, they should also return; if a person enters a zone, they should also leave.

For example, if a person travels from home to work and then from work to home on a certain day, then there are 2 home-based work trip productions are generated at the home TAZ, and two attractions related at his or her work location.

## Estimate Trip Productions/Attractions Using Trip Rates

Productions are typically modeled as a function of population and/or number of households, as well as income levels or auto ownerships. Other explanatory variables might be used, such as the number of workers, but we need to make sure explanatory variables are often not interrelated and correlated with each other.

Attractions are often modeled as a function of the number of households and/or number of employees, where employment may be broken down by different types (e.g., retail, office, service, and other). Again, other explanatory variables can also be used, such as commercial floor space or CBD (Central Business District) variables, but the same checks for correlation between variables should be utilized. Attractions tend to be more difficult to measure/estimate, and we tend to have less trust in these estimates. For more information, users can read NCHRP Report 365: "Travel Estimation Techniques, CH 3 trip generation and NCHRP Report 716: Travel Demand Forecasting: Parameters and Techniques, CH 4.4 Trip Generation.

## Accessibility

In transportation planning, accessibility is first defined as the potential of opportunities for traveler interaction. Typically, accessibility captures the extent of the attractiveness of each potential destination and some researchers represent accessibility as the amount of activity potential reachable within a given travel time or distance from an origin location.

One of the goals of transportation system construction and management is to improve individuals' accessibility or the ease of reaching desired activities, destinations, and services. On the other hand, many transportation network design models instead focus on maximizing individuals' mobility or the ease of movement within the network.

In general, quantitative accessibility measures describe how many destinations can be reached how easily from a particular zone. For more information, users can check https://tfresource.org/topics/Accessibility.html.

## Trip distribution

There are a variety of trip distribution formulations. Among recent travel models, two formulations dominate: the gravity model and the destination choice model.

For each OD pair, a typical gravity model is applied to calculate zone-to-zone demand volume. The gravity model allocates trips roughly in proportion to the number of productions at the production end, roughly in proportion to the number of attractions at the attraction end, and roughly in proportion to a measure of proximity (often called a "friction factor") of the two zones. A gravity model maybe "singly-constrained" or

"doubly-constrained". For more information, please visit
https://tfresource.org/topics/Trip_distribution.html.

For each OD pair, a typical gravity model is applied to calculate zone-to-zone demand volume.

$$T_{ij} = P_i \cdot \frac{A_j \cdot F_{ij} \cdot K_{ij}}{\sum_j (A_j \cdot F_{ij} \cdot K_{ij})}$$

$$F_{ij} = a(d_{ij})^b e^{c(d_{ij})}$$

where $T_{ij}$ is total trips from zone $i$ to zone $j$; $P_i, A_j$ are productions in zone $i$ and attractions in zone $j$, respectively; $F_{ij}$ is the friction factor for travel from zone $i$ to zone $j$; $K_{ij}$ is the correction factor for travel from zone $i$ to zone $j$, equal to 1 by default; $d_{ij}$ is the accessibility from zone $i$ to zone $j$; parameter $a, b, c$ are the friction factor coefficients, of which the default values under three typical trip purposes are listed in the following table.

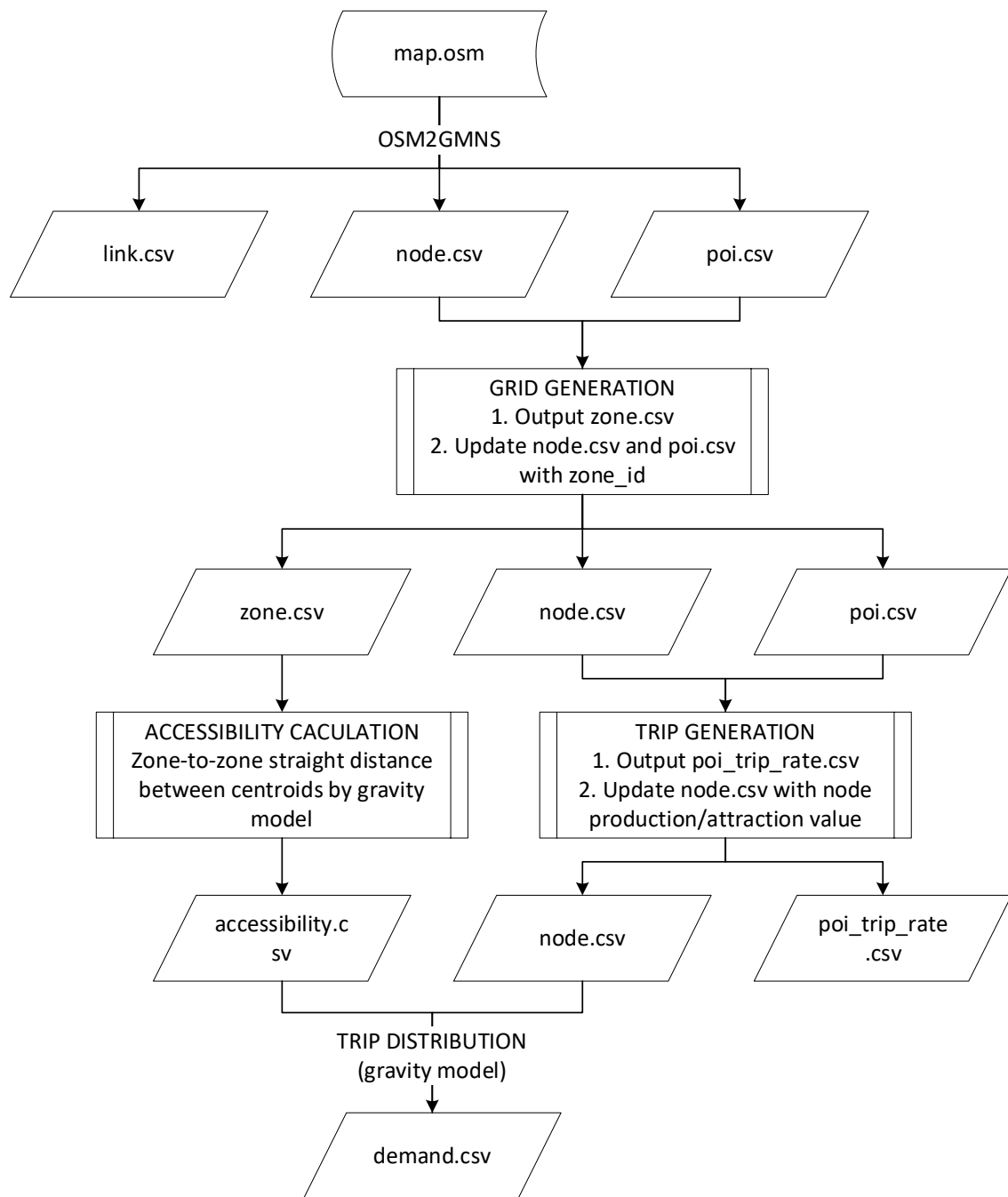| Trip purpose | a | b | c |
|---|---|---|---|
| HBW | 28507 | -0.02 | -0.123 |
| HBO | 139173 | -1.285 | -0.094 |
| NHB | 219113 | -1.332 | -0.1 |

## II. What is grid2demand?

Grid2demand is a quick trip generation and distribution tool based on the trip generation and trip distribution methods of the standard 4-step travel model in transportation planning applications. To create a simple coordinate system, the area of interest is partitioned into an alphanumeric grid (also known as atlas grid), in which each cell is identified by a combination of a letter and a number. The trip generation step is performed at the Point of Interest (POI) node level, using ITE trip generation tables (https://www.ite.org/technical-resources/topics/trip-and-parking-generation/trip-generation-10th-edition-formats/) or other trip rate references. The trip distribution is carried out using a typical gravity model. Data flow chart are illustrated in the following table and figure.

# Description of Data Files

| Step | Process | Input File or Parameter | Output File | Method |
|---|---|---|---|---|
| 0 | Network files preparation | map file from OpenStreetMap | *node.csv, link.csv, poi.csv* | Osm2gmns tool |
| 1 | Input files reading | *node.csv, poi.csv* | | |
| 2 | Zone generation and Grid partition | Number of blocks or grid scales in meter with latitude of the area of interest (optional) | *poi.csv* (update with zone id) | Alphanumeric grid |
| 3 | Trip Generation | *poi_trip_rate.csv* (optional), trip purpose | *poi_trip_rate.csv*, *node.csv* (update with zone id and demand values) | Trip rate method |
| 4 | Accessibility calculation | *accessibility.csv* (optional), latitude of the area of interest | *accessibility.csv* | Simple straight distance between zone centroids |
| 5 | Trip Distribution | Trip purpose, friction factor coefficients | *demand.csv, zone,csv* | Gravity model |
| 6 | Visualization | | QGIS or NEXTA | |

# Framework flowchart of grid2demand

```mermaid
map.osm
   │ OSM2GMNS
   ├──────────┬──────────┐
   ▼          ▼          ▼
link.csv   node.csv    poi.csv
              │          │
              └────┬─────┘
                   ▼
        GRID GENERATION
        1. Output zone.csv
        2. Update node.csv and poi.csv
           with zone_id
```

**GRID GENERATION**
1. Output zone.csv
2. Update node.csv and poi.csv with zone_id

zone.csv        node.csv        poi.csv

**ACCESSIBILITY CACULATION**
Zone-to-zone straight distance between centroids by gravity model

**TRIP GENERATION**
1. Output poi_trip_rate.csv
2. Update node.csv with node production/attraction value

accessibility.csv       node.csv        poi_trip_rate.csv

**TRIP DISTRIBUTION**
(gravity model)

demand.csv

For the entire package, the input files include the network files in GMNS format (*node.csv, link.csv*) as well as *poi.csv*, generated by the OSM2GMNS tool.

Users can download a default *poi_trip_rate.csv* from https://github.com/asu-trans-ai-lab/grid2demand/blob/main/examples/data_folder/poi_trip_rate.csv and apply further adjustments based on local traffic conditions.

The final output files include *zone.csv, accessibility.csv*, and *demand.csv* for zone-to-zone OD demand matrix. Accordingly, *node.csv* and *poi.csv* are updated with zone information.

## Grid partition and zone creation

To facilitate hierarchical and multi-resolution spatial computing, grid cells are used to aggregate trips to traffic analysis zones, while standard TAZs are typically defined based on census tracts. The user can specify the number of zones per row and per column or the cell width and height of each grid cell (in km or miles) for the area of interest. To maintain a consistent mapping, we use a fractional value in terms the degree at different latitudes to represent different lengths on a flat surface. That is, a value of 0.01 longitudinal degree at latitude 60 degree is equivalent to 0.558 km on a flat surface. Thus, a user can provide a latitude value of the area of interest. The closest latitude in the following table is selected to calculate the longitudinal length.

| Latitude | City | Degree-equivalent distance |
|---|---|---|
| 60° | Saint Petersburg | 55.80 km |
| 45° | Bordeaux | 69.47 km |
| 30° | New Orleans | 96.49 km |
| 0° | Quito | 111.3 km |

## Accessibility and distance computing

Accessibility is measured by zone-to-zone straight-line distance according to zone centroid coordinates. A more advanced version will be provided in the future to use the shortest path algorithm for computing end-to-end driving or multimodal travelling distance and costs.

## Trip generation

To enable detailed modeling of trip generation from park lots and buildings, different types of POI nodes are specifically covered in file *poi.csv*, extracted from the original *OSM files*. The user can supply for more information in poi.csv in case of missing values. The trip generation process used in grid2demand has the following 3 sub-steps.

1) For each node, the amount of produced or attracted traffic is computed based on underlying trip purpose and POI type, defined in *poi_trip_rate.csv*.

2) Update the field of production and attraction for each POI or boundary node in *node*.csv.

3) For each zone, its total production and attraction values can be calculated as the sum of node-based values across all nodes with the corresponding zone id.

 A sample *poi_trip_rate* table is listed below.

| poi_type_id | building | unit_of_measure | production_rate1 | attraction_rate1 | production_rate2 | attraction_rate2 | production_rate3 | attraction_rate3 |
|---|---|---|---|---|---|---|---|---|
| 1 | roof | 1,000 Sq. Ft. GFA | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 2 | public | 1,000 Sq. Ft. GFA | 4.79 | 0.1 | 4.79 | 0.1 | 4.79 | 0.1 |
| 3 | dormitory | 1,000 Sq. Ft. GFA | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 4 | yes | 1,000 Sq. Ft. GFA | 1.15 | 1.15 | 1.15 | 1.15 | 1.15 | 1.15 |
| 5 | university | 1,000 Sq. Ft. GFA | 1.17 | 0.1 | 1.17 | 0.1 | 1.17 | 0.1 |
| 6 | convenience | 1,000 Sq. Ft. GFA | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 7 | office | 1,000 Sq. Ft. GFA | 2.04 | 0.1 | 2.04 | 0.1 | 2.04 | 0.1 |

## III. Quick start

We will use the University of Maryland, College Park as an example to illustrate how to use grid2demand.

**Step 1: Installation**

You can install the latest release of grid2demand at PyPI via pip:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
pip install grid2demand
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

After running the command above, the grid2demand package along with three required dependency packages (Shapely, pandas) will be installed on your computer (if they have not been installed yet).

**Step 2: Determine the boundary of interest and download .osm file from OpenStreetMap**

(1) Adjust the map to the location of interest and click on the "Export" button on the top.
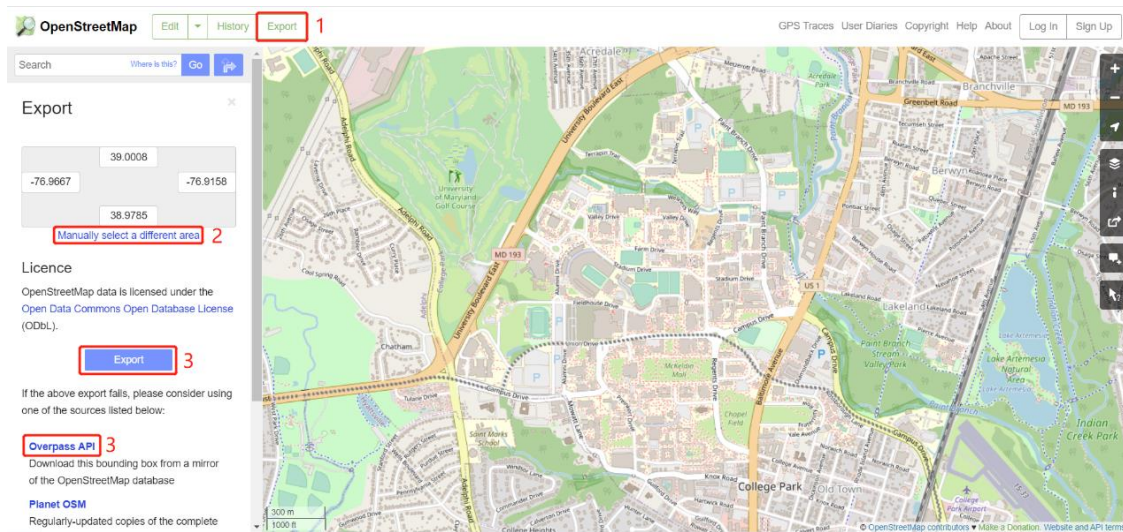
(2) Obtain the latitude and longitude coordinates (users can "manually select a different area").

(3) Click on the "Export" button found in the middle of the navigator to download an OSM data file.

(4) For a very large area of interest, users need to click the link of "Overpass API" to obtain a map file.



**Step 2: Execute OSM2GMNS to get network files in GMNS format**

Open the Python IDE such as Pycharm for a typical configuration. Then, use OSM2GMNS to convert *map. osm* file in OSM format into a network file in GMNS format.

Notes: User guide for osm2gmns can be found at https://osm2gmns.readthedocs.io/en/latest/.

```python
import osm2gmns as og
net = og.getNetFromOSMFile(r'map.osm',network_type=('railway','aeroway','auto','walk','bike'),\
                    POIs=True,default_lanes=True,default_speed=True)
og.connectPOIWithNet(net)
og.outputNetToCSV(net, output_folder)
```

Please note that *poi.csv* might have different degrees of missing information. Please supply additional accurate POI type information if needed.

**Step 3: Execute grid2demand Python code**

**(1) Import the package and read input network data**

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```
import grid2demand as gd
gd.ReadNetworkFile("./data_folder")
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**(2) Partition network into grid cells**

A user can customize the number of grid cells by setting "number_of_x_blocks" and "number_of_y_blocks". On the other hand, a user can customize cell's width and height in terms of a degree of longitude and latitude by setting "cell_width" and "cell_height". By default, "cell_width" and "cell_height" are set as the length on a flat surface under a specific latitude corresponding to the degree of 0.006 (equivalent to 400 meters or 0.25 miles at latitude = 45degree).

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```
gd.NetworkPartition(number_of_x_blocks=None, number_of_y_blocks=None,
cell_width=500, cell_height=500, latitude=30)
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**(3) Obtain production/attraction rates of each land use type with a specific trip purpose**

A user can customize *poi_trip_rate.csv* by adding an external file folder location according to different trip purposes. By default, the trip purpose is set as purpose 1.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```
gd.GetPoiTripRate(trip_rate_folder = None, trip_purpose = 1)
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**(4) Compute production/attraction value of each node according to POI type**

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```
gd.GetNodeDemand()
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**(5) Calculate zone-to-zone accessibility matrix by centroid-to-centroid straight-line distance**

A user needs to input the latitude value of the area of interest. The script will match the closest latitude to calculate the longitudinal length in kilometer. The degree of 30 is selected as the default. Also, a user can customize the accessibility matrix by setting the external folder of file *accessibility.csv*.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

gd.ProduceAccessMatrix(latitude=30,accessibility_folder=None)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**(6) Apply gravity model to perform trip distribution**

A user needs to input the trip purpose and the friction factor coefficients. The default values of HBW, HBO and NHB are described above.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

gd.RunGravityModel(trip_purpose=3,a=None,b=None,c=None)

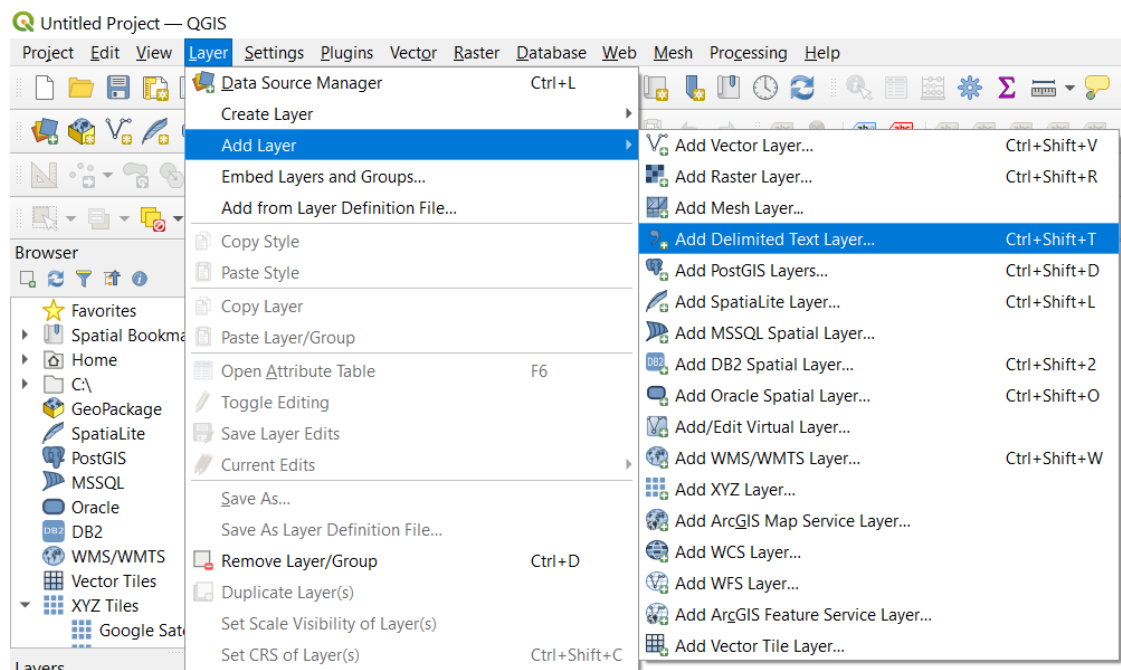~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

One can configure the working dictionary in the Python IDE (e.g., Pycharm), before executing grid2demand to obtain zone-to-zone demand, with generated four output files highlighted in blue below. The output files will be saved under the same folder of the input files.
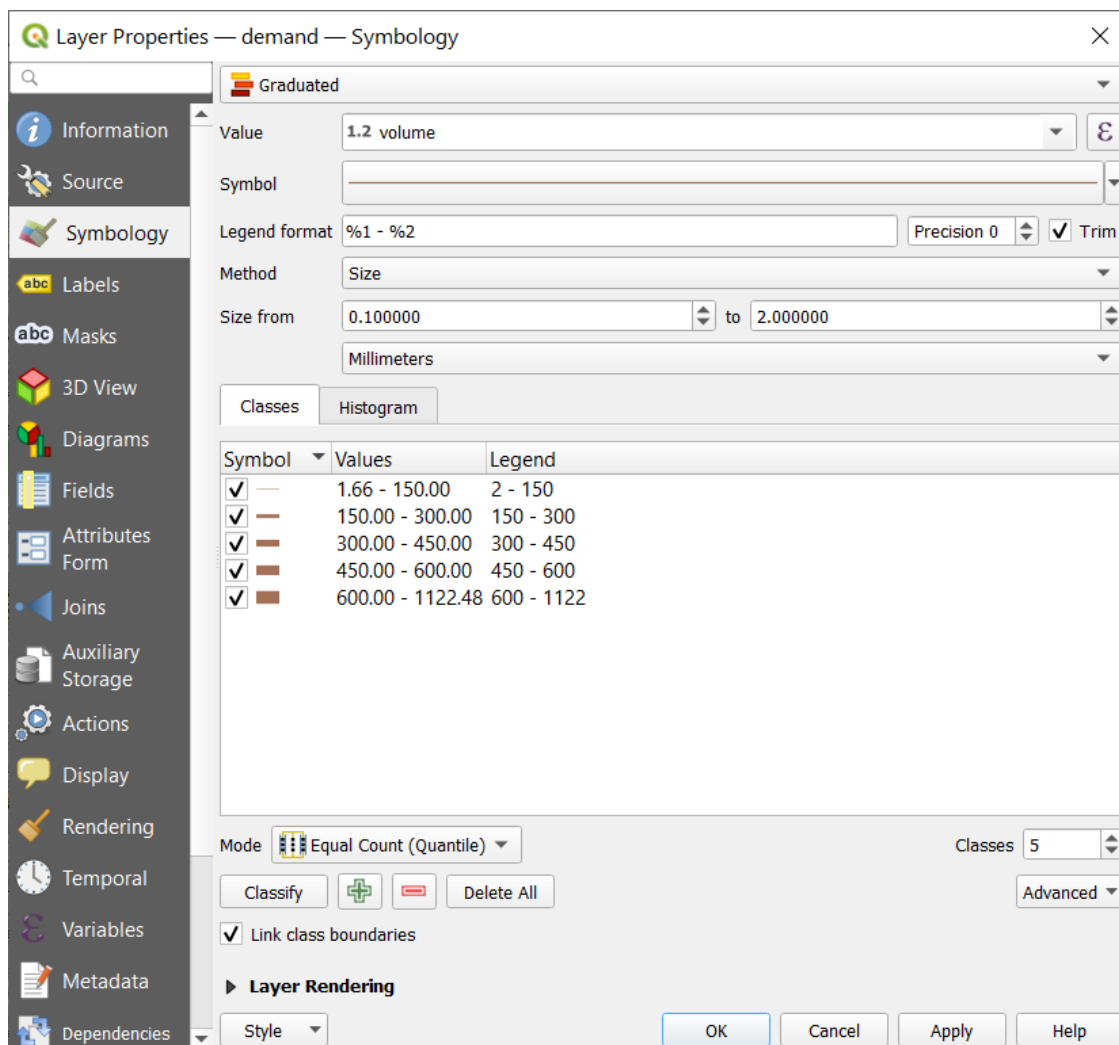
| | | | |
|---|---|---|---|
| accessibility | 12/27/2020 8:54 PM | Microsoft Excel Co… | 277 KB |
| demand | 12/27/2020 8:54 PM | Microsoft Excel Co… | 322 KB |
| node | 12/27/2020 8:54 PM | Microsoft Excel Co… | 667 KB |
| poi | 12/27/2020 8:54 PM | Microsoft Excel Co… | 944 KB |
| poi_trip_rate | 12/27/2020 8:54 PM | Microsoft Excel Co… | 2 KB |
| zone | 12/27/2020 8:54 PM | Microsoft Excel Co… | 8 KB |
| link | 12/25/2020 10:54 AM | Microsoft Excel Co… | 2,965 KB |

**Step 4: Visualization in QGIS**

Open QGIS and add Delimited Text Layer to load the demand.csv file (with geometry info).

Then open the "Properties" window of the demand layer. Set the symbology as graduated symbols by size.

The zone-to-zone demand volume can be visualized with a base map.