# The typology of access and frequency of public transport departures

# Workflow description

This document provides a technical description of the workflow for the creation of a typology of access and frequency of public transport departures. This description accompanies a working paper presenting the typology and its results[[1]](#footnote-1).

This document is accompanied by a series of Python scripts. Running these scripts requires the installation of ArcGIS Desktop 10.x (or ArcGIS Pro 2.x) with their Python component. They can be run in a Python IDLE window after changing the variables (workspace and input dataset paths, field names, …) defined at the beginning of the code. They could also be adapted to be used with an ArcGIS user created ArcToolbox interface.

The analysis focuses on public transport services provided during weekdays. From the GTFS stop\_times tables we select all departures between 6:00 and 20:00 (14 hours long). The actual calendar day taken into account depends on the availability of timetable schedules in the input datasets. Each of the input datasets refers to its own period of validity. Within that period, we detect the day with the maximum number of departures. As service frequencies will be typically the highest during weekdays selecting the day with the maximum number of departures avoids public holidays, weekend days or school holiday periods. For the selected day the stop\_times table is combined with the stops table to allocate the total number of departures between 6:00 and 20:00 at each of the stops. GTFS stops tables contain XY coordinates. These are used to create a point dataset representing all stops.

The methodology uses following geographical datasets:

* a point dataset of stops with, for each point, the sum of departures between 6:00 and 20:00
* a street network dataset needed to produce service areas (catchment areas), adapted for pedestrians
* a polygon dataset with the population data, as detailed as possible
* a polygon dataset with the zones for which we want results (for instance administrative units, urban centres, functional urban areas, ...)

All geographical datasets must have the same geographic coordinate system.

Each of the following steps refers to a python script. The scripts are numbered in accordance with these steps.

### Step 1: Classify the stops into two groups of transport modes

We distinguish two groups of transport modes. Group 1 covers bus and tram. Group 2 includes metro, mainline rail and suburban rail. We created this distinction to take into account the differences in operational speed of the vehicles.

The script *01\_Share\_Stops\_Into\_2\_Groups.py* divides a point feature dataset with all transport modes into two output feature datasets: one for bus/tram, the other for metro/rail, because each group of transport mode is processed independently before their results are combined.

The criterion to distinguish the modes of transport is the value of the "route\_type" item from the General Transit Feed Specification.[[2]](#footnote-2)

For bus/tram: 0, 3, 4, 5, 6, 7, 11, 200, 700, 800, 900, 1000, 1200, 1300, 1400

For metro/rail: 1, 2, 100, 300, 400, 500, 600

Input: Zones\_all\_stops\_pt

Output: Zones\_bus\_tram\_pt; Zones\_metro\_rail\_pt

### Step 2: Create a point feature dataset of clustered stops

The script *02\_Cluster\_Stops.py* uses the output of the previous script to create a point feature dataset of clustered stops. It must be run for each group of transport modes: for bus/tram and for metro/rail.

All stops located within 50 m from each other are considered as a single cluster of stops. Each cluster is represented by a single point, located at the centre of the clustered stops. The output feature dataset contains all original points which do not have any neighbouring points within a radius of 50 meters, in addition to the new clustered points.

If a stop location represents a cluster of nearby stops, the departures of all clustered locations are summed and allocated to the cluster. The average number of departures per hour is then calculated for each of the (clustered) stops.

Input: Zones\_bus\_tram\_pt; Zones\_metro\_rail\_pt

Output: Zones\_bus\_tram\_pt\_clustered; Zones\_metro\_rail\_pt\_Clustered

### Step 3: Create service areas around the stops

The script *03\_Make\_Service\_Areas.py* creates service areas around the clustered stops using a pedestrian network. Service areas represent the spatial distribution of the average frequencies of departures between 6:00 and 20:00 accessible within a walking distance around stops (or clusters of stops). Service areas of 500 meters walking are created around bus and tram stops, while 1 km of walking distance is used around metro and rail stops.

A road network dataset adapted for pedestrians is thus needed, as well as an application able to produce the output polygons. The road network has a field which provides the number of minutes needed to walk through a network segment (Shape\_Length\*60/5000, assuming a walking speed of 5km/h).

In the ArcGIS Desktop application, parameters are: Meters, TRAVEL\_TO, 500 (or 1000), SIMPLE\_POLYS, NO\_MERGE, DISKS, NO\_LINES, OVERLAP, NO\_SPLIT, ALLOW\_UTURNS, TRIM\_POLYS (100 Meters).

Input: Zones\_bus\_tram\_pt\_clustered; Zones\_metro\_rail\_pt\_Clustered; Zones\_Pedestrian\_Network

Output: Zones\_bus\_tram\_SA; Zones\_metro\_rail\_SA

### Step 4: Reclassify the average number of departures of the service areas into 4 classes

The script *04\_Reclassify\_Service\_Areas.py* converts the service areas produced with the previous script into polygons in which the average number of departures are reclassified in four classes.

The tool must be run for each group of transport modes.

Within each of the groups of transport modes, service areas tend to partly overlap each other. The script first calculates the geometric union of the service areas but, in the overlapping areas it keeps the polygon with the highest value, as we assume that the stop with the most frequent departures is the most probable choice between the stops nearby. This produces polygon datasets used in step 6.

Frequencies are then reclassified into four classes recorded in a field FREQ\_CLASS\_BUS\_TRAM or FREQ\_CLASS\_METRO\_RAIL. This produces polygon datasets used in step 5. The results by group of transport mode can be used independently if an analysis is required of the frequencies of each group of transport modes separately.

|  |  |
| --- | --- |
| **Classes** | **Frequency for rail and metro / bus and tram** |
| 0 | Outside service areas |
| 1 | Less than 4 departures an hour |
| 2 | >= 4 and < 10 departures an hour |
| 3 | More than 10 departures an hour |

*Table 1: Frequency classes of departures by group of transport modes*

Input: Zones\_bus\_tram\_SA; Zones\_metro\_rail\_SA

Output: Zones\_bus\_tram\_SA\_Reclass; Zones\_metro\_rail\_SA\_Reclass; Zones\_bus\_tram\_Serv\_Freq\_4classes; Zones\_metro\_rail\_Serv\_Freq\_4classes

### Step 5: Create a polygon dataset combining transport modes, establishing a typology of 5 categories of frequencies

The script *05\_Combine\_Transport\_Modes\_and\_Make\_Serv\_Freq\_Typology.py* combines the dataset with the 4 frequency classes of the transport mode bus/tram with the one of metro/rail, producing a set of areas containing the combination of the 4x4 possible frequency classes, i.e. a matrix of 16 possible classes. It finally dissolves the intersected areas, grouping these 16 classes to obtain a final typology with five categories of frequencies: no access, low, medium, high, very high.

|  |  |
| --- | --- |
| **Category** | **Combinations of frequency classes of the bus/tram mode with the metro/rail mode** |
| 0 = No access | 0 and 0 (no easily accessible departures by none of the modes, i.e. areas at more than 500 meters from any bus or tram stop and at more than 1 km from any metro or train stop) |
| 1 = Low | 0-1 or 1-0 or 1-1 (less than 4 departures an hour for at least one group of modes, but no access to more than 4 departures an hour) |
| 2 = Medium | At least a 2 and no 3 (access to between 4 and 10 departures an hour for at least one group of modes, but no access to more than 10 departures an hour) |
| 3 = High | At least and maximum one 3 (access to more than 10 departures for one group of modes, but not for both) |
| 4 = Very high | 3 and 3 (access to more than 10 departures an hour for both groups of modes) |

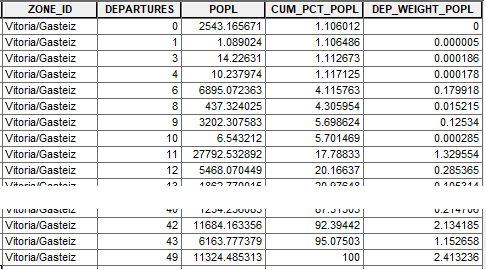
*Table 2: Typology of services frequencies*

Input: Zones\_bus\_tram\_Serv\_Freq\_4classes; Zones\_metro\_rail\_Serv\_Freq\_4classes

Output: Zones\_Serv\_Freq\_Typology

### Step 6: Create a table with the distribution of population by service frequency

The script *06\_Calculate\_Popl\_Distrib\_By\_Service\_Frequency.py* creates a table which makes the relationship between the frequency of departures and the cumulative distribution of the population.



This tool combines the reclassified service areas datasets of both transport modes (made in step 4). Where the two groups overlap, we know the maximum number of departures by bus and tram and the maximum by metro and train. The tool calculates the sum of both maxima.

This result is intersected by the areas containing the population figures. Hence we obtain the geographical distribution of population relative to the total number of departures available within walking distance. This distribution is finally summarised in a table where population is summed by the number of departures and by zone of interest (e.g. an urban centre, a city, a Functional Urban Area,…).

Input: Zones\_bus\_tram\_SA\_Reclass; Zones\_metro\_rail\_SA\_Reclass; Zones\_population\_pl; Zones\_pl

Output: Zones\_Popl\_Distrib\_by\_Serv\_Freq\_tbl

The frequency distribution may be skewed and can contain outlier values. Therefore, we also derive the population-weighted median number of departures an hour from the frequency table.

### Step 7: Create a table with the distribution of population by typology of frequency

The polygon dataset combining transport modes with a typology of 5 categories of frequencies (made in step 5) is intersected with the areas containing the population figures. From the intersected areas, the script *07\_Calculate\_Popl\_Distrib\_By\_Typology.py* produces a table with the sum of the population, grouped by category of service frequency (values 0-4) and by zone of interest.



Three other fields derived from the table produced in step 6 are added in the final table. They give the total population of the zones of interest, the sum of population located inside service areas and the sum of population-weighted median number of departures an hour.

Input: Zones\_Serv\_Freq\_Typology; Zones\_population\_pl; Zones\_pl; Zones\_Popl\_Distrib\_by\_Serv\_Freq\_tbl

Output: Zones\_Popl\_Distrib\_by\_Typology\_tbl

1. Poelman, H., Dijkstra, L. and Ackermans, L., 2020, How many people can you reach by public transport, bicycle or on foot in European cities?  
   Measuring urban accessibility for low-carbon modes. Working paper, EC DG Regional and Urban Policy, <https://ec.europa.eu/regional_policy/en/information/publications/working-papers/2020/low-carbon-urban-accessibility>; see pages 8-15 and 35-36. [↑](#footnote-ref-1)
2. See <https://developers.google.com/transit/gtfs/reference> and <https://developers.google.com/transit/gtfs/reference/extended-route-types> [↑](#footnote-ref-2)