PUBLIC TRANSPORT WITHIN CITIES: INDICATORS OF FREQUENCY AND SPEED

Hugo Poelman

AIM

This (draft) technical note describes the workflow used to assess frequency and speed of public transport networks inside urban areas. A previous analysis of timetable data on urban public transport has allowed us to develop indicators of accessibility to public transport, by relating the population distribution in urban areas with the location of stops and stations, and the frequency of services available at these stops[1]. These indicators still lacked information about the performance of the networks, especially in terms of speed and frequency of specific transport modes.

As spatial units of analysis, we will use the administrative cities, greater cities and the grid-based urban centres[2]. These will be combined with timetable data in order to develop aggregated indicators on the entire urban public transport network in the spatial units.

DATA PREPARATION

While most of the input data, especially on timetables and stops, are extracted from tables of the GTFS[3] model, they need some further preparation.

The GTFS **Stops** table contains latitude/longitude coordinates in decimal degrees (WGS84). These need to be converted into a point feature class and projected into an equal-area projection relevant for Europe [4], including the XY coordinates in meters. By means of a series of spatial joins, these stop points obtain additional attributes: the codes of cities, greater cities and urban centres in which they are located.

For further processing, the stop features and their attribute table are filtered: only those located in the spatial units under review (cities, greater cities or urban centres) will be used. Especially in cases where the timetable data cover an entire country, this filter results in a substantial reduction of the data volume, allowing for a much smoother processing.

Table **StopTimes** is derived from GTFS input data: it contains the complete set of times at each of the stops, including stop_id, departure time and arrival time, but also route_type (transport mode), trip_id and service_id (selection related to services active on a selected working day). To facilitate further calculations, time values have been converted into decimal numbers.

Table **Trips** contains the list of trips, including trip_id and route_id.

---

[1] *See: Poelman, H. and Dijkstra, L., 2014, Measuring access to public transport in cities and regions, Regional Working Paper, Directorate-General for Regional and Urban Policy, European Commission (forthcoming).*

[2] *For the definition of these concepts, see: Dijkstra, L. and Poelman, H., 2012, Cities in Europe, the new OECD-EC definition, European Commission, Brussels (http://ec.europa.eu/regional_policy/sources/docgener/focus/2012_01_city.pdf)*

[3] *For a description of this specification and its concepts, see: https://developers.google.com/transit/gtfs/.*

[4] *Lambert Azimuthal Equal Area projection (GCS ETRS 1989), EPSG:3035.*

WORKFLOW

This workflow will focus on all connections between a departure stop and the next arrival stop. These connections will be the basic units of analysis for which we will calculate travel time and travel speed.

In a first step of the process, Stops and StopTimes are combined to query the relevant departures and the relevant arrivals. This query selects all StopTimes in the chosen spatial units, departing between 6:00 and 20:00 on the selected working day.

```
CREATE TABLE WORK."TripsDepartures"n AS
   SELECT DISTINCT t1.URAU_CODE_2011 AS URAU_CODE_2011,
          t2.trip_id,
          t2.stop_id,
          t2.departure_time_dec,
          t2.stop_sequence,
          t2.route_type
      FROM WORK.STOPS t1
           INNER JOIN WORK.STOPTIMES t2 ON (t1.stop_id = t2.stop_id)
      WHERE t2.departure_time_dec >= 6 AND t2.departure_time_dec < 20
      ORDER BY t1.URAU_CODE_2011,
               t2.trip_id,
               t2.departure_time_dec;
```

In a similar way we select all arrivals in the relevant spatial units, by querying StopTimes with an arrival time after 05:59.

```
   CREATE TABLE WORK."TripsArrivals"n AS
   SELECT DISTINCT t1.URAU_CODE_2011 AS URAU_CODE_2011,
          t2.trip_id,
          t2.stop_id,
          t2.arrival_time_dec,
          t2.stop_sequence,
          t2.route_type
      FROM WORK.STOPS t1
           INNER JOIN WORK.STOPTIMES t2 ON (t1.stop_id = t2.stop_id)
      WHERE t2.arrival_time_dec >= 6
      ORDER BY t1.URAU_CODE_2011,
               t2.trip_id,
               t2.arrival_time_dec;
```

We will now combine the information of departures and arrivals. For each trip departure, all subsequent destination stops and their arrival time are queried (provided the arrival stops are located in the same spatial unit).

```
CREATE TABLE WORK."DepArrCities"n AS
   SELECT t1.URAU_CODE_2011,
          t1.trip_id,
          t1.departure_time_dec,
          t2.arrival_time_dec,
          t1.route_type,
          t1.stop_id AS 'dep_stop_id (stop_id)'n,
          t2.stop_id AS 'arr_stop_id (stop_id)'n,
          t1.stop_sequence AS 'dep_stop_seq (stop_sequence)'n,
          t2.stop_sequence AS 'arr_stop_seq (stop_sequence)'n
      FROM WORK.TRIPSDEPARTURES t1, WORK.TRIPSARRIVALS t2
      WHERE (t1.URAU_CODE_2011 = t2.URAU_CODE_2011 AND t1.trip_id = t2.trip_id) AND
t2.arrival_time_dec > t1.departure_time_dec
      ORDER BY t1.URAU_CODE_2011,
               t1.trip_id,
               t1.departure_time_dec,
               t2.arrival_time_dec;
```

From this query, the earliest arrival by trip and by departure stop is selected. This will help determining the first segment of the trip, once the trip started at the departure stop[5].

```
CREATE TABLE WORK."MinTripsSeg"n AS
   SELECT t2.URAU_CODE_2011,
          t2.trip_id,
          t2.route_type,
          t2.'dep_stop_id (stop_id)'n,
          t2.departure_time_dec,
          /* MIN_of_arrival_time_dec */
            (MIN(t2.arrival_time_dec)) AS MIN_of_arrival_time_dec
      FROM WORK.DEPARRCITIES t2
      GROUP BY t2.URAU_CODE_2011,
               t2.trip_id,
               t2.route_type,
               t2.'dep_stop_id (stop_id)'n,
               t2.departure_time_dec
      ORDER BY t2.URAU_CODE_2011,
               t2.trip_id,
               t2.departure_time_dec;
```

While the previous query contains the departure and arrival times of the first segment after departure, it does not yet inform about the exact location of the arrival stop. By combining the information from the two previous queries, we can determine which is the arrival stop corresponding to the first segment of each part of a trip. Hence, we can calculate the travel time for each of these segments.

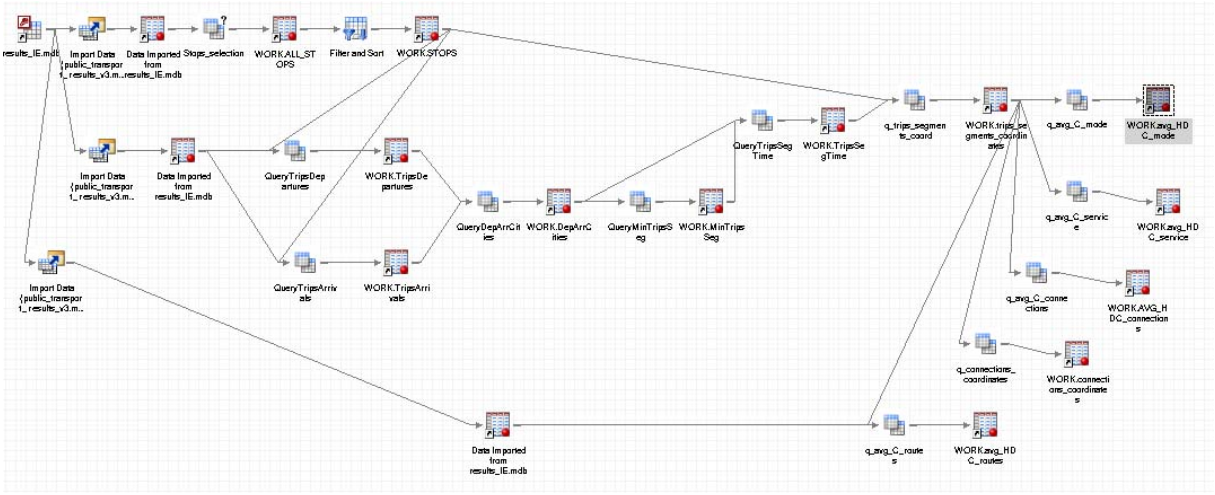
```
CREATE TABLE WORK."TripsSegTime"n AS
   SELECT t2.URAU_CODE_2011,
          t2.trip_id,
          t2.route_type,
          t2.'dep_stop_id (stop_id)'n,
          t2.departure_time_dec,
          t2.'arr_stop_id (stop_id)'n,
          t2.arrival_time_dec,
          /* TRAVEL_TIME_DEC */
            (t2.arrival_time_dec - t2.departure_time_dec) AS TRAVEL_TIME_DEC
      FROM WORK.MINTRIPSSEG t1, WORK.DEPARRCITIES t2
      WHERE (t1.URAU_CODE_2011 = t2.URAU_CODE_2011 AND t1.trip_id = t2.trip_id AND
t1.'dep_stop_id (stop_id)'n = t2.'dep_stop_id (stop_id)'n AND t1.departure_time_dec
= t2.departure_time_dec AND t1.MIN_of_arrival_time_dec = t2.arrival_time_dec)
      ORDER BY t2.URAU_CODE_2011,
               t2.route_type,
               t2.trip_id,
               t2.departure_time_dec;
```

We now combine this result with the information from the Stops point features. Using the XY coordinates of the departure stop and the arrival stop we calculate the Euclidian distance and the trip speed for each of the connections. We also create a unique identifier for the connection (CONNECT_ID) (based on the identifiers of the stops).

```
CREATE TABLE WORK."trips_segments_coordinates"n AS
       SELECT t2.URAU_CODE_2011,
          t2.route_type,
          t2.trip_id,
          t2.'dep_stop_id (stop_id)'n,
          t2.departure_time_dec,
          dep.POINT_X AS DEP_POINT_X,
          dep.POINT_Y AS DEP_POINT_Y,
          arr.POINT_X AS ARR_POINT_X,
```

[5] *According to the GTFS specification, this query should be superfluous: the specification contains the attribute "stop sequence", which allows tracking the trip from one stop to the next. Unfortunately, many GTFS datasets do not contain any values for the stop sequence attribute.*

```
          arr.POINT_Y AS ARR_POINT_Y,
          t2.'arr_stop_id (stop_id)'n,
          t2.arrival_time_dec,
          t2.TRAVEL_TIME_DEC,
          /* DISTANCE */
          (SQRT((dep.POINT_X-arr.POINT_X)**2+(dep.POINT_Y-arr.POINT_Y)**2)) AS
DISTANCE,
          /* CONNECT_ID */
          (CASE
            WHEN t2.'dep_stop_id (stop_id)'n LT t2.'arr_stop_id (stop_id)'n
            THEN  t2.'dep_stop_id (stop_id)'n || '_' || t2.'arr_stop_id (stop_id)'n
            ELSE t2.'arr_stop_id (stop_id)'n || '_' || t2.'dep_stop_id (stop_id)'n
           END) AS CONNECT_ID
      FROM WORK.TRIPSSEGTIME t2, WORK.STOPS dep, WORK.STOPS arr
      WHERE (t2.'dep_stop_id (stop_id)'n = dep.stop_id AND t2.'arr_stop_id
(stop_id)'n = arr.stop_id);
```



Workflow for the production of aggregated data by connection, transport mode, service, route and spatial unit.

AGGREGATED RESULTS

This result now allows the calculation of various aggregated values, especially the sum of Euclidian distances travelled, the sum of the travel times, and the average (Euclidian) speed of all trip segments.

Aggregation by spatial unit and transport mode:

```
CREATE TABLE WORK."avg_HDC_mode"n AS
   SELECT t1.URAU_CODE_2011 LABEL='',
          t1.route_type,
          /* SUM_of_DISTANCE */
            (SUM(t1.DISTANCE)) AS SUM_of_DISTANCE,
          /* SUM_of_TRAVEL_TIME_DEC */
            (SUM(t1.TRAVEL_TIME_DEC)) AS SUM_of_TRAVEL_TIME_DEC,
          /* AVG_EUCL_SPEED */
            (((SUM(t1.DISTANCE))/1000)/(SUM(t1.TRAVEL_TIME_DEC))) AS AVG_EUCL_SPEED
      FROM WORK.TRIPS_SEGMENTS_COORDINATES t1
      GROUP BY t1.URAU_CODE_2011,
               t1.route_type;
```

Aggregation by spatial unit and individual trip:

```
CREATE TABLE WORK."avg_HDC_service"n AS
   SELECT t1.URAU_CODE_2011 LABEL='',
          t1.route_type,
          t1.trip_id,
```

```
        /* SUM_of_DISTANCE */
          (SUM(t1.DISTANCE)) AS SUM_of_DISTANCE,
        /* SUM_of_TRAVEL_TIME_DEC */
          (SUM(t1.TRAVEL_TIME_DEC)) AS SUM_of_TRAVEL_TIME_DEC,
        /* AVG_EUCL_SPEED */
          (((SUM(t1.DISTANCE))/1000)/(SUM(t1.TRAVEL_TIME_DEC))) AS AVG_EUCL_SPEED
    FROM WORK.TRIPS_SEGMENTS_COORDINATES t1
    GROUP BY t1.URAU_CODE_2011,
             t1.route_type,
             t1.trip_id;
```

By spatial unit, connection (CONNECT_ID = segment between two stops) and transport mode:

```
CREATE TABLE WORK."AVG_HDC_connections"n AS
   SELECT t1.URAU_CODE_2011 LABEL='',
          t1.route_type,
          t1.CONNECT_ID,
          /* AVG_of_DISTANCE */
            (AVG(t1.DISTANCE)) AS AVG_of_DISTANCE,
          /* AVG_of_TRAVEL_TIME_DEC */
            (AVG(t1.TRAVEL_TIME_DEC)) AS AVG_of_TRAVEL_TIME_DEC,
          /* AVG_EUCL_SPEED */
            (((SUM(t1.DISTANCE))/1000)/(SUM(t1.TRAVEL_TIME_DEC))) AS AVG_EUCL_SPEED
      FROM WORK.TRIPS_SEGMENTS_COORDINATES t1
      GROUP BY t1.URAU_CODE_2011,
               t1.route_type,
               t1.CONNECT_ID;
```

Using the Trips table, we can also produce aggregates by spatial unit, transport mode and route, because that table contains the route identifier. The aggregation can be interesting in those cases where the original GTFS data contain a “shape” table, which can be used to represent the actual network geometry. Line features depict specific routes, allowing the representation of aggregates on route-specific maps. Unfortunately, most of the GTFS datasets available do not contain the "shape" data.

```
CREATE TABLE WORK."avg_HDC_routes"n AS
   SELECT t1.URAU_CODE_2011,
          t1.route_type,
          t2.route_id,
          /* SUM_of_TRAVEL_TIME_DEC */
            (SUM(t1.TRAVEL_TIME_DEC)) AS SUM_of_TRAVEL_TIME_DEC,
          /* SUM_of_DISTANCE */
            (SUM(t1.DISTANCE)) AS SUM_of_DISTANCE,
          /* AVG_EUCL_SP */
            (((SUM(t1.DISTANCE))/1000)/(SUM(t1.TRAVEL_TIME_DEC))) AS AVG_EUCL_SP
      FROM WORK.TRIPS_SEGMENTS_COORDINATES t1
           INNER JOIN WORK.TRIPS t2 ON (t1.trip_id = t2.trip_id)
      GROUP BY t1.URAU_CODE_2011,
               t1.route_type,
               t2.route_id;
```

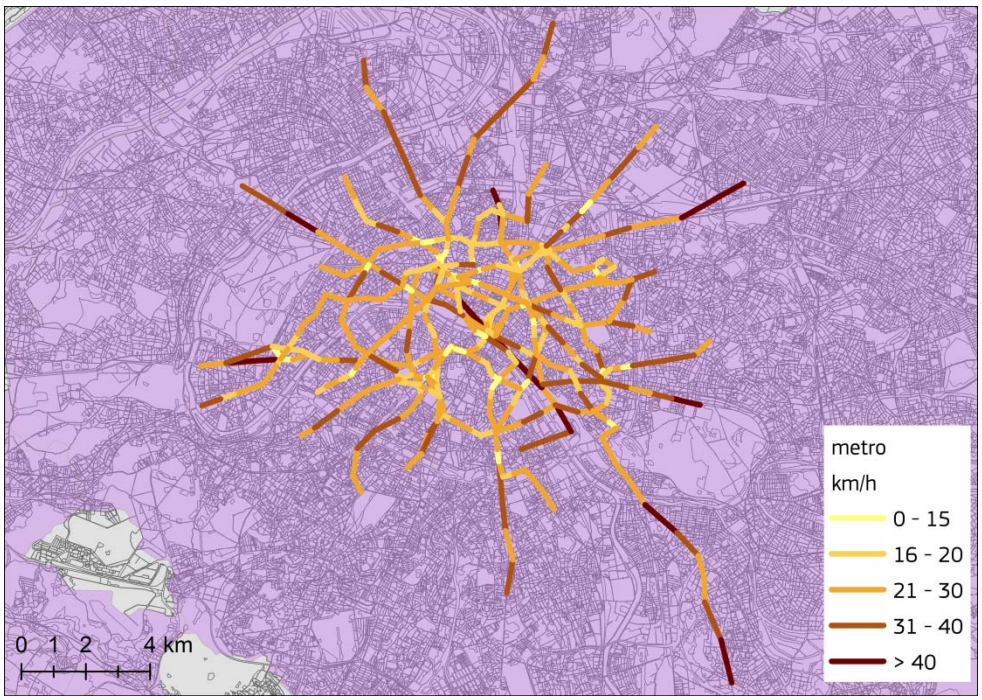
MAPPING THE RESULTS

Furthermore, we can create a table with the identifiers and the coordinates of all available connections between stops. If required, this table can be used to create line features enabling a cartographic representation of results.

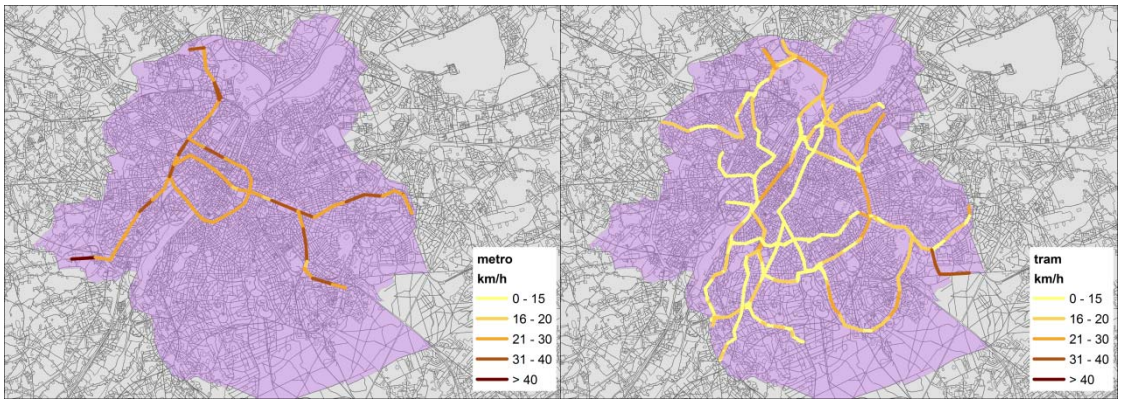
```
CREATE TABLE WORK."connections_coordinates"n AS
   SELECT DISTINCT t1.CONNECT_ID,
          t1.DEP_POINT_X,
          t1.DEP_POINT_Y,
          t1.ARR_POINT_X,
          t1.ARR_POINT_Y
      FROM WORK.TRIPS_SEGMENTS_COORDINATES t1
      ORDER BY t1.CONNECT_ID,
```

```
t1.DEP_POINT_X,
t1.DEP_POINT_Y;
```

Due to the absence of realistic network shapes in most of the datasets, the maps with the trip segment representation will be very schematic. In the case of elaborate networks with many lines and stops (e.g. most of the bus networks in medium-sized or bigger cities) this will result in cluttered maps. Examples show that more simple tram or metro networks can easily be mapped on the basis of the schematic connection lines.



Paris (greater city): metro network: average Euclidian speed by network segment.

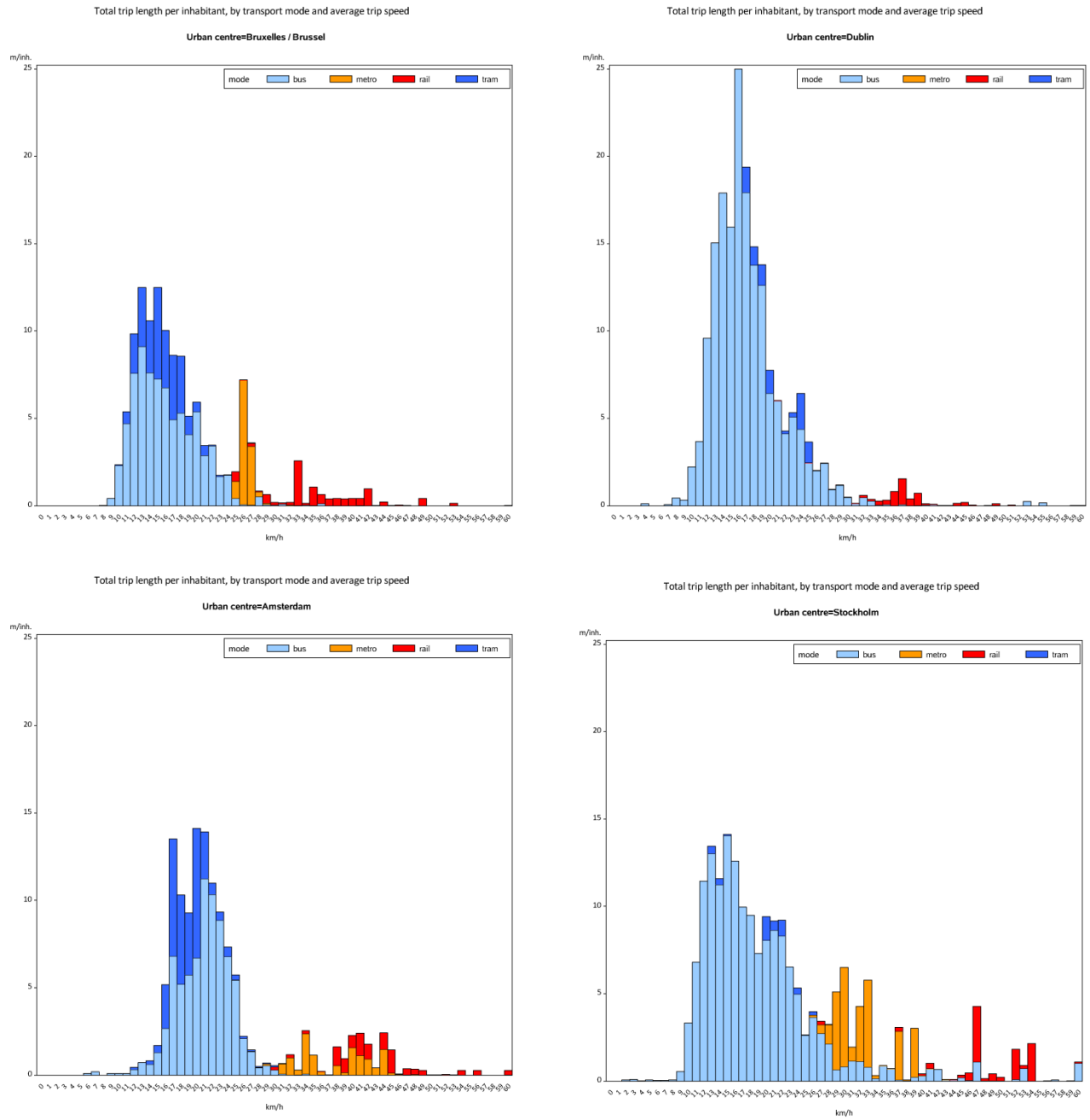


Brussels (city): Average Euclidian speed by segment of the metro and the tram network

CREATION OF SUMMARY OUTPUT BY SPATIAL UNIT (CITIES, GREATER CITIES, URBAN CENTRES)

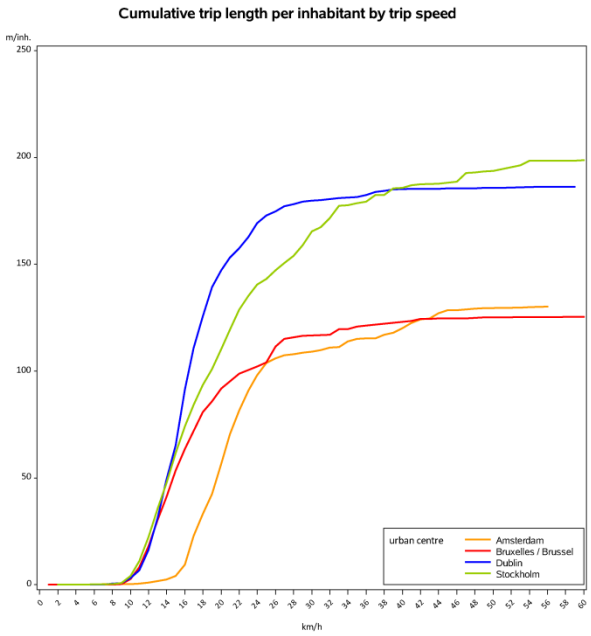
Based on the aggregated data by spatial unit, mode and trip we create a series of graphs depicting the distribution of trip length by transport mode and by trip speed. For that purpose, trips are aggregated by rounded (integer) values of trip speed. We also calculate the total trip length (all transport combined) by spatial unit. This allows us to calculate the share of trip length of a certain speed – by transport mode - as a

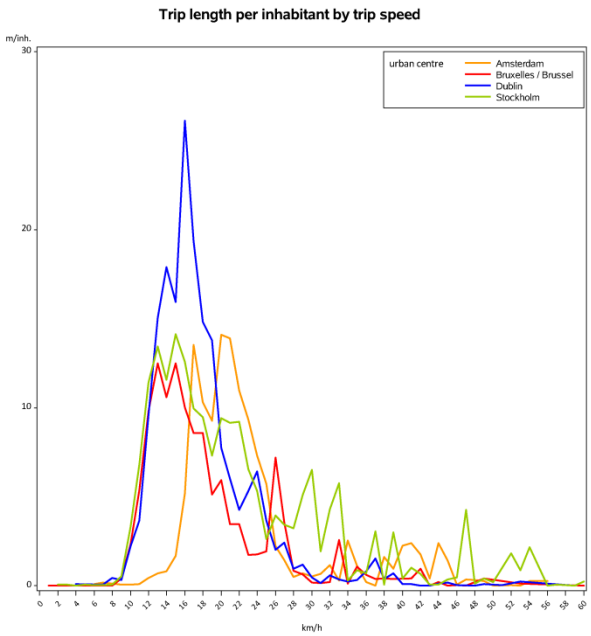
percentage of total trip length. Alternatively, trip length by speed can be represented as length/inhabitant (total population of the spatial unit).



Total trip length per inhabitant by transport mode and average Euclidian trip speed in selected urban centres.

Trip length aggregation by speed category and by spatial unit (without distinction by mode) allows for an easy calculation of (cumulative) length by inhabitant and a graphical comparison of the trip length and speed between selected spatial units.





These graphs illustrate that total trip length per inhabitant is substantially higher in Stockholm and Dublin than in Amsterdam or Brussels, whereas the average speed of the trips is highest in Amsterdam and Stockholm (the faster the slope of the cumulative graph rises, the slower the trips are).

Summary indicators can be calculated by spatial unit and transport mode. The initial process flow needed to be executed as many times as there are (GTFS) data sources available; i.e. in order to obtain results for all transport modes of a particular spatial unit, we may need to analyse several GTFS datasets (e.g. one for rail, another for bus and tram). Hence, the summary process aggregates all results by mode and spatial unit to obtain a final distribution of trip length, trip time and speed by spatial unit and transport mode.

The average trip speed and trip length data by spatial unit can be further cross-tabulated by transport mode. This results in a set of summary indicators by spatial unit:

- modal split of the length of all trips (%)

- length of all trips (by mode), by inhabitant (m/inh.)

- average trip speed by mode and total

In addition, summary indicators on access to public transport can be added[6], especially:

- share of population without easy access to public transport services (% or total population)

- population-weighted median number of departures an hour (within walking distance).

[6] *See footnote 1. These indicators can only be calculated if comprehensive data are available for all public transport modes, including mainline rail connections departing in the city under review. For that reason, the spatial coverage of speed/length indicators and of accessibility indicators is currently not the same.*