Description of calculations

This document describes the different spatial analyses and calculations performed to transform source data into the data model of the KPO data system. Generally, the input refers to data tables in the source data or tables from the KPO data model. The output refers to tables of the KPO data model, that are used in the end for the synthesis maps and the interactive system. The parameters are optional arguments used in the calculation that influence the outcome. Details of the calculations are in the SQL script files indicated, saved in the scripts folder of the project repository.

0. Data model preparation

Extract the geographic features and relevant attributes from the source data into the layers of the KPO datasysteem data model. Pre-process the 9292 GTFS timetable data for analysis.

Input: Various (see data sources)

Parameters: Province Noord Holland boundary, MRA boundary

Output: Various (see data model)

Script: prepare_pilot_data.sql; prepare_network_data.sql

1. Woonscenarios

The difference in the number of households in WLO scenario regions, between the current situation (2014) and future scenarios (e.g. WLO 2040 Hoog, WLO 2040 Laag). Also the percent change in households and the household density, based on the area of the region.

Input: WLO 2040 scenarios, WLO 2014 data

Parameters: –

Output: Woonscenarios

Script: prepare_pilot_data.sql

2. Street Isochrones from public transport stops

Create travel isochrones from public transport stops. Isochrones are polygons delimiting areas within a maximum travel time along the street network from the public transport stops.

The polygons are generated from buffers around the streets segments that are below the predefined cutoff travel distance/time.

Input: TOP10NL wegdeel, GTFS stops

Parameters: travel mode (bicycle, walk), origin (stations), cutoff distance or time (10 minutes), buffer size (100m)

Output: Isochronen

Script: analyse_isochrones.sql; prepare_pilot_data.sql

3. OV Halte Frequencies

The frequency of public transport services on individual public transport stops for different transport modes, at different times of day, and for different types of train service. The operation extracts the total number of services for the selected parameters and then calculates the average hourly rate, i.e. number of services per hour.

Input: GTFS stops, GTFS stop times, GTFS trips, GTFS routes, GTFS calendar dates

Parameters: modaliteit (train, metro, tram, bus, ferry), time period (ochtendspits, dal uren, avondspits), train type (high speed, intercity, sprinter)

Output: OV haltes

Script: calculate_ov_frequency.sql

4. Dichtstbijzijnde Station

Sets the name of the nearest train station to a given housing scenario region, indicating if the region is within walking or cycling distance of the train station, if it intersects the respective isochrone. If not, identifies the station that is nearest to the region, based on the nearest fiets isochrone. When more than one station isochrone is intersected, or if they are at the same distance, these are ordered by highest frequency of high speed, intercity and sprinter trains.

Input: Woonscenarios, Isochronen, OV haltes

Parameters: isochroon modaliteit (fiets, loop), train types, halte frequentie (avondspits)

Output: Woonscenarios

Script: calculate_scenarios.sql

5. Overzicht woonscenarios

Calculates of the number of households in relation to the invloedsgebieden van knooppunten, for each housing scenario and TOD beleidsniveau. This summary indicates the total number of households in the Province for the scenario, and the number of households that is within walking distance, within cycling distance and outside the invloedsgebied of the knooppunten. A different calculation is made for different TOD levels (0%, 50%, 100%). For 0% TOD level the households of the regions are added in groups based on the 'within walking distance' and 'within cycling distance' attribute, and those outside. For 50% TOD level, half of the households outside the influence of stations are added to the 'within cycling distance' total. And for 100% TOD level all households outside the influence of stations are added to the 'within cycling distance' total. In this case there will be 0 households outside the influence of stations.

Input: Woonscenarios

Parameters: woonscenario, TOD beleidsniveau

Output: Overzicht woonscenarios

Script: calculate_scenarios.sql

6. Knooppuntenscenarios

Calculates the total number of households associated with each transit node (train station), based on the different housing scenarios and TOD policy levels (0%, 50% and 100%) combinations. This calculation uses the nearest station attribute of the housing scenarios, and sums for each station all the households of the scenario. For 0% TOD level, only thre households within walking or cycling distance of the station are considered. For 50% TOD level, half of the households that are nearest to the station but outside the area of influence are added to it. And for 100% TOD level, all the households that are nearest to the station are added. As well as the total, we also calculate the percent change in households in relation to the current situation.

The percent change in households of a scenario is then multiplied by the percent of local residents using the station (defined as the percent of walking and cycling as 'voortransport'). This gives the percent change in station usage. This factor is used to estimate the different station characteristics for the scenario, namely number of in- and uit stappers (train and BTM), usage of bike parking spaces, and usage of Park and Ride parking spaces.

Input: Woonscenarios, Kenmerken knooppunten

Parameters: woonscenario, TOD beleidsniveau

Output: Knooppuntenscenarios

Script: calculate_scenarios.sql

7. Ruimtelijke intensiteit

The sum of the number of residents, workers and students per 100m grid cell. The number of residents is given by CBS with the grid cell. The number of workers is the sum of jobs in all activities of the LISA data set. The number of students is estimated based on the size of the education establishments in the LISA data set, using a ratio of 10 students per worker. The Open DUO data set is not used because in several cases the education establishment has an extremely large number of students (tens of thousands) that must represent the total number of an organisation or group of schools, and not on the specific site.

Input: CBS vierkant 100m, LISA 2016

Parameters: LISA codes (education)

Output: Ruimtelijke kenmerken

Script: prepare_pilot_data.sql

8. Fysieke dichtheid

The built density per 100m grid cell is based on the floor space index (FSI) value provided by PBL. The FSI (also known as floor area ratio, https://en.wikipedia.org/wiki/Floor_area_ratio) gives the total built up area of every floor, divided by the area of the ground, including built and open space. For the calculation we only consider built blocks with an area of 100m2 or more. The physical density on a grid cell is the sum of the FSI of all the building blocks intersecting the cell weighted (multiplied) by the area of each part, divided by the total area of the cell.

The result is a decimal value where 1.0 corresponds to 1 built floor of 1 hectare (the area of a cell). Considering open space (including street width and unbuilt block interior), this should be equivalent of an urban area with buildings with 2 or 3 floors. Similarly, a value of 2.0 would be equivalent to an area with 5 to 6 floors.

Input: CBS vierkant 100m, PBL bouwvlak FSI

Parameters: bouwvlak > 100 m2

Output: Ruimtelijke kenmerken

Script: prepare_pilot_data.sql

9. OV Bereikbaarheidsniveau

The OV bereikbaarheid of each CBS 100m grid cell is based on the same principles of the Transport for London (TfL) PTAL methodology (https://tfl.gov.uk/cdn/static/cms/documents/connectivity-assessment-guide.pdf).

This measures the diversity and frequency of all public transport services at a location, giving a greater weight to rail than to other public transport modes. The PTAL calculation takes into account the frequency of different public transport routes available within reach during morning peak hours. The only adaptation of the TfL methodology to the Netherlands is the the distance to stops from a location. In this case we consider the same distances as the walk and cycle isochrones from ov stops: bus and tram stops within 400m (5 minutes walk), metro stops within 800m (10 minutes walk), train stations within 3000m (10 minutes cycle). The result is the OV bereikbaarheidsniveau and the OV bereikbaarheidsindex. The index is a numeric value of accessibility, while the level is a scale of 8 classes for predefined index values, based on the original PTAL implementation in London:

1. 1a - Very poor (0.01 tot 2.5)
2. 1b - Very poor (2.5 tot 5)
3. 2 - Poor (5 tot 10)
4. 3 - Moderate (10 tot 15)
5. 4 - Good (15 tot 20)
6. 5 - Very Good (20 tot 25)
7. 6a - Excellent (25 tot 40)
8. 6b - Excellent (40 of meer)
9.

Input: CBS vierkant 100m, GTFS stops, GTFS stop times, GTFS trips, GTFS routes, GTFS calendar dates, Isochronen

Parameters: Isochrone distance, halte modaliteit, train service type, time of day

Output: Ruimtelijke kenmerken

Script: calculate_ptal.sql

10. Onderbenut bereikbare locaties

Selects the 100m grid cells of the Ruimtelijke kenmerken layer that have a low level of use and a high level of accessibility. The level of use can be defined the number of households, the intensity (residents + workers + students), the built density (FSI) or the average property value (WOZ). The user sets the maximum level of use and the minimum level of accessibility to be considered. The level of use is based on 6 classes:

	Huishoudens	**Intensiteit**	**Dichtheid**	**WOZ**
Laag	Minder 10	Minder 50	Minder 0.1	Minder 150k
	10 - 20	50 - 100	0.1 - 0.4	150k - 200k
	20 - 40	100 - 200	0.4 - 0.7	200k - 300k
	40 - 60	200 - 400	0.7 - 1.0	300k - 500k
	60 - 80	400 - 600	1.0 - 1.5	500k - 750k
Hoog	80 - 100	600 - 800	1.5 - 2	750k - 1000k

Input: Ruimtelijke kenmerken, water surface

Parameters: maximum use, minimum accessibility

Output: Ruimtelijke kenmerken (subset)

Script: KPO plugin (calculated on the fly)

11. Ontwikkellocaties kenmerken

The identification, number of planned housing units, and density for each plan location (Plancapaciteit, Kantoorleegstanden). Calculate the mean number of households, intensity, density, property value, and OV Bereikbaarheidsniveau based on the 100m grid cells contained in the plan's surface.

Input: Regional agreement zones, Plancapaciteit, Kantooreegstanden, OV Bereikbaarheidsniveau

Parameters: huising plan type

Output: Ontwikkellocaties

Script: prepare_pilot_data.sql

12. Overzicht Ontwikkellocaties

The total number of planned housing units that are inside and outside desirable locations (underused and accessible, as defined by the user, see calculation 10). If a housing plan contains grid cells of desirable locations, its housing units are considered to be located in a desirable location (grid cell).

Input: Ontwikkellocaties

Parameters: -

Output: Overzicht ontwikkellocaties

Script: KPO plugin (calculated on the fly)

13. OV Isochronen van knooppunten

Create isochrones from the knooppunten along the public transport routes of bus, tram and metro, using the mean travel time between stops and the available routes during avondspits. The isochrones are for 10 minutes using public transport, plus a maximum of 5 minutes walk at the destination (400m isochrone). OV destinations within a 10 minute walk (800m isochrone) of the knooppunt are not considered.

Input: Knooppunten, GTFS stops, GTFS stop times, GTFS trips, GTFS routes, GTFS calendar dates, TOP10NL wegdeel

Parameters: knooppunt, travel mode, cutoff time, time of day, buffer size

Output: Isochrones

Script: analyse_isochrones.sql

14. Invloedsgebied overlap

Identify zones in the area of influence of several train stations, defined by the bike isochrones. For each 100m grid cell inside more than one isochrone calculate the aantal van knooppunten and the knooppunten namen, merging the identical cells into overlap zones. For each zone calculate the inwoner dichtheid and the intensity, as a sum of the residents and intensity of each cells divided by the number of cells (ha).

Input: Ruimtelijke kenmerken, Isochronen

Parameters: -

Output: Invloedsgebied overlap

Script: prepare_pilot_data.sql

15. Fietsroutes

Identify the individual bike routes that cross the invloedsgebied overlap zones, and that start or end at a train station. Select the relevant network links from the Fiets Telweek survey, calculating the route intensity per link. These are aggregated to form route geometries.

Input: Fiets TelWeek netwerk, Fiets TelWeek routes, Invloedsgebied overlap, Knooppunten
Parameters: -
Output: Fietsroutes
Script: analyse_routes.sql

16. OV routes

Create the individual public transport routes for bus, tram and metro that stop at a train station. Aggregate the links between stops that make up a uniquely named route (e.g. bus 37 or tram 3). Calculate the route frequency (services per hour).
Input: GTFS routes, GTFS stop times, GTFS trips, GTFS stops
Parameters: -
Output: OV routes
Script: calculate_ov_frequency.sql

17. Identify ov routes at locations

Identify the ov routes that serve the different locations and indicate if they are within walking isochrone from a station. Use a buffer to select the OV routes that have stops within those locations.
Input: Magneten, Belangrijke locaties, OV routes
Parameters: buffer (400m)
Output: Magneten, Belangrijke locaties
Script: prepare_pilot_data.sql