# GIS in Sustainable Urban Planning and Management: A Global Perspective


Methodological demonstration for **Chapter 15 – *"Planning for Transit Oriented Development (TOD) using a TOD index"*** **(pp.267-280)**


- Yamini Jain Singh, Johannes Flacke, Mark Zuidgeest, Martin van Maarseveen

- Methodological demonstration by André Mano


**Disclaimer**



**How to use this document**

Most of the steps described are illustrated with screenshots. Bear in mind that what the screenshot depicts and what you see in your computer might differ slightly depending on the QGIS version you are using and the way your toolbars and add-ons are arranged. Along the text you will see different icons. The key for these icons is as follows:

Data or external resource to download;

A software action you are supposed to do;

Information specific about QGIS.

Additional or complementary scientific information;

An important concept which you may want keep in mind;

**[1]** An operation that is referenced in the flowchart of operations.

Additionally, for the sake of readability, the following style conventions are used:

- A reference to dataset or a layer uses `this style;`
- A QGIS command, or any clickable button is noted using **this style.**
- A QGIS menu or section is highlighted using *this style*.

At the end of the document, a diagram depicting the workflow described in these pages can be seen. It is advisable to look at it first and/or refer to it as you proceed.

**Outline**

In order to assess transit efficiency, for the case of Arnhem and Nijmegen City two distinct Transit Oriented Development (TOD) indexes are developed: the actual TOD index measuring TOD indexes around existing transit nodes (train stations) and the Potential TOD aimed at identifying areas that are potential locations for transit connectivity. The TOD indexes presented in this methodological documentation use less variables compared with what the book chapter describes due to restrictions related to data usage and distribution. However, the steps to take would essentially be the same regardless of the number of variables that are taken into account.

**Getting started**

Download the data; the data consists of the following files:

- `Arnhem-Nijmegen.qgs` – a QGIS project preloaded with the layers;
- `CBS_demographics_2007.shp` – Polygon features representing statistical units of demographic data ;
- `roads.shp` – Line features representing the road network;
- `land_use.shp` – Polygon features with land use attributes;
- `walkable_cyclable.shp` – Line features representing pedestrian paths and bycicle tracks;
- `railways.shp` – Line features representing railways;
- `Train_stations.shp` – Point features with train stations data;
- `Potential_tod.shp` – Grid with the potential TOD index of study area;
- `Expressions.txt` – text file with all the field calculator expressions.

Start QGIS and open `Arnhem-Nijmegen.qgis` (Figure 1).

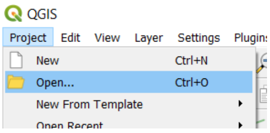



Figure 1 – opening a project

From the *Layers panel*, **right-click** on a layer and access the attribute table to examine it. Repeat the procedure for the other layers (Figure 2).

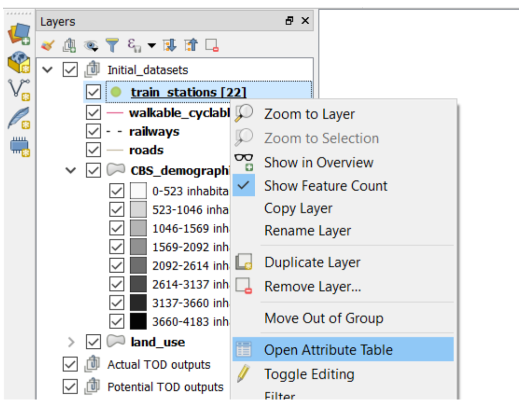


Figure 2 – assessing attribute table

**BEFORE YOU START**

Although there is nothing fundamentally different, from a methodological point of view, on what calculating the actual and potential TOD indexes concerns, they are different in their meaning. Therefore, for the sake of clarity we decided to split this exercise in two parts – each addressing one TOD type.

**FIRST PART – ACTUAL TOD**

**Create a buffer [1]**

For the Actual TOD, an area of 800m around each train station is taken as the window of analysis. This means all the subsequent analysis will consider this areas only

- From the *Processing toolbox*, filter by "buffer" to find the **Buffer** tool. Provide `train_stations` as **Input layer** set the **Distance** to 800 and provide an output name – we suggest `buffer_800`. Hit **Run** to execute (Figure 3).

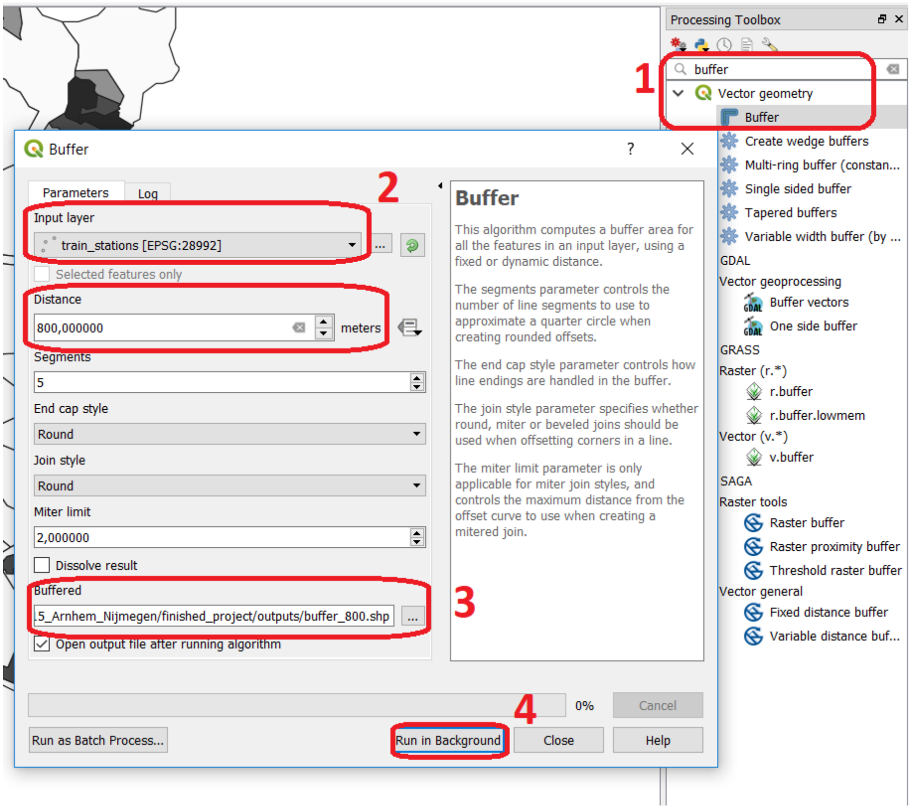



Figure 3 – Using the Buffer tool

Now that we have our window of analysis, we can proceed with the operations that will allow us to have the information necessary to extract the five variables that we need. These five variables are:

1. Diversity of land uses;
2. Mixness of land uses with respect to its residential or non-residential nature;
3. Population density;
4. Total length of walkable and cyclable roads;
5. Number of road intersections.

Please refer to the book chapter for details on the nature and logic of these indicators.

**Intersection [2]**

For the first two variables, we need to obtain the land use classes for each of our buffered areas. To collect that information we will use a simple intersect.

➤ From the *Processing toolbox*, filter by "intersection" to find the **intersection** tool. Provide `buffer_800` as **Input layer**, `land_use` as Insersection layer and provide an output name – we suggest `intersections`. Hit **Run** to execute (Figure 4).

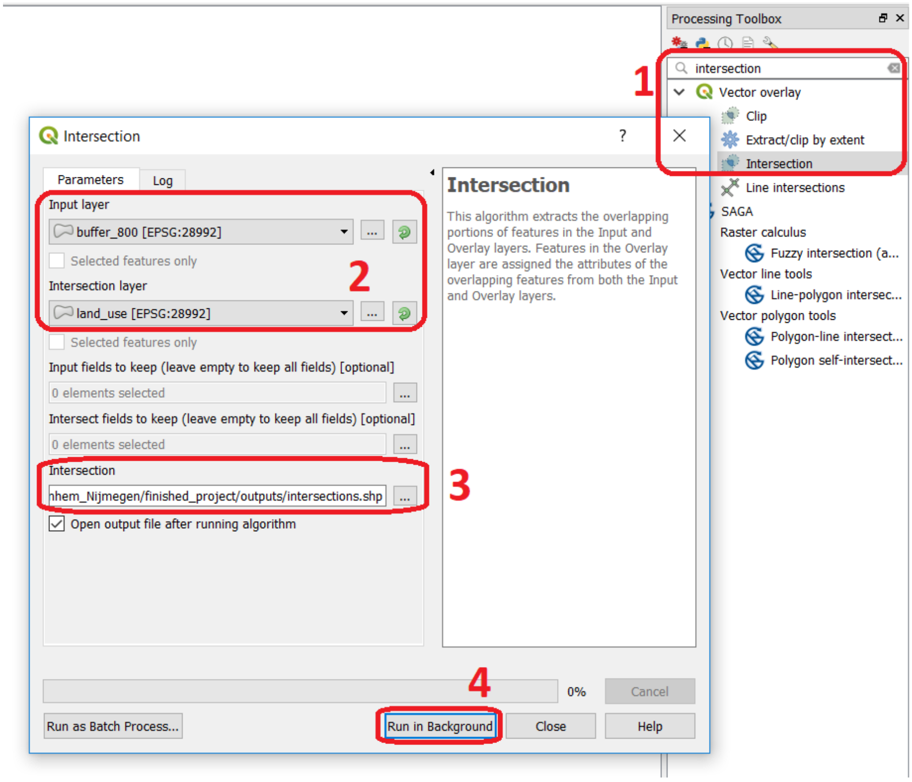



Figure 4 – Using the Intersection tool

**Field calculator [3]**

We will calculate the land use diversity as:

$$LU_d(i) = \frac{-\sum_i Q_{lui} \times \ln(Q_{lui})}{\ln(n)}$$

Given that:

$$Q_{lui} = \frac{S_{lui}}{S_i}$$

where:

$lui$ = land use class within the analysis window $i$
$S_{lui}$= Total area of the specific land use within the analysis window $i$
$S_i$ = Total area of the analysis window $i$
$n$ = Total number of land use classes within the analysis window $i$

- From the *Layers panel*, **right-click** on the layer `intersections` and choose the option **Properties**. From the **Source Fields** start editing mode by clicking on the **Toggle editing mode** button and then on the **Field Calculator** icon. In the *Field Calculator* widow choose **Create a new field** and provide "diver" as the **Output field name** and *Decimal number (real)* as the **Output field type** and enter the following expression:

```
((sum($area , group_by:="lu_type_en")/sum($area ,
group_by:="naam")) * (log10(sum($area ,
group_by:="lu_type_en")/sum($area , group_by:="naam")))) /
log10( count_distinct( "lu_type_en" ))
```

Press **OK** to dismiss the *Field Calculator* dialog and click on **Toggle editing mode** to make the newly created field permanent (Figure 5).

!! You will be asked if you want to save the edits to the layer, make sure you do!!

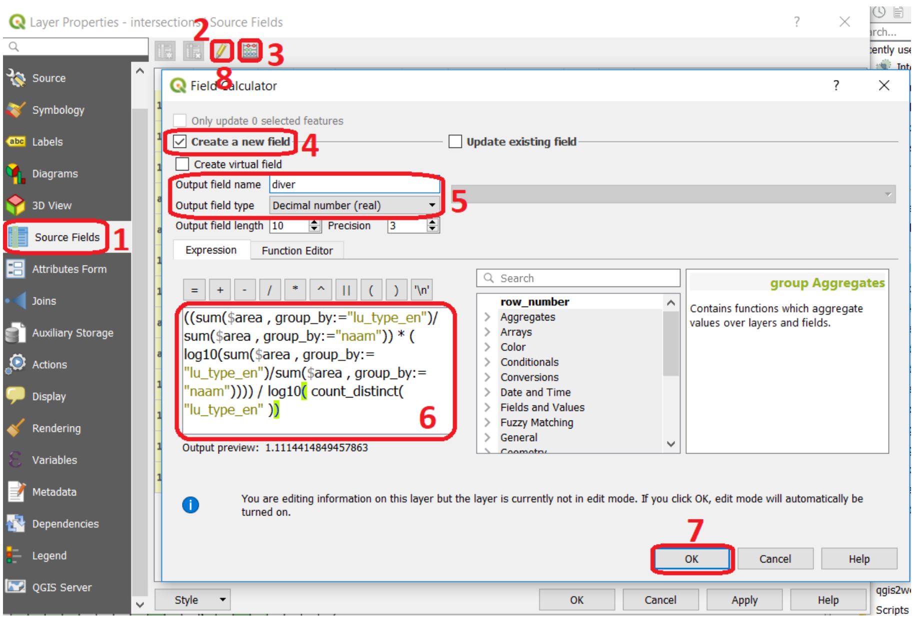



Figure 5 – Using the field calculator

What we just did was to add a new column to the attribute table of the `intersections` layer containing, for each feature, the values of the diversity of land values. Let's take a closer look at the expression we used to materialize the equations:

First we need to calculate $Q_{lui}$:

The sum | Of the areas | By land use | Dividing by | The total area of the analysis window (identified by the attribute "naam")

```
(sum($area,group_by:="lu_type_en")/sum($area, group_by:="naam"))
```

Then we multiply $Q_{lui}$:

```
*
```

By:

The natural log | Of $Q_{lui}$

```
(log10(sum($area,group_by:="lu_type_en")/sum($area , group_by:="naam")))
```

And finally we divide by

The natural log | Of the number of land use classes represented in the window of analysis

```
/ log10( count_distinct( "lu_type_en"))))
```

The above is executed for each feature of the layer `intersections`, however we are still missing the final part of the equation which is the $\sum$ of all the values in order to obtain the diversity value for the whole window of analysis i.e. (buffer)

Repeat the previous operation using the *Field Calculator*, but this time provide "total_div" as the **Output field name** and *Decimal number (real)* as the **Output field type** and use the following expression:

```
sum( "diver", group_by:="naam" )
```

Again, please note the subtlety of using the `group by` as the second argument of the `sum` function. This will ensure that the sum will be performed by train station/buffer (identified by the "`naam`" attribute)and not just a blunt sum of all the values. Consequently, you will have 22 distinct sum values, one per each station instead of a single value.

**Field calculator [3]**

Next, we will calculate the land use mixness as:

$$MI(i) = \frac{\sum_{\cap i} S_c}{\sum_{\cap i}(S_c + S_r)} \quad \forall i$$

Where:

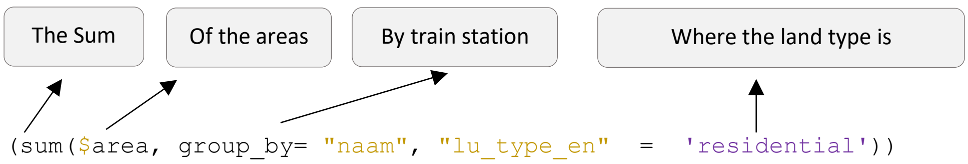
*i* = area of analysis
*Sc* = sum of the total area under non-residential urban land uses within *i*
*Sr* = sum of the total area under residential land use within *i*.

➤ Again using the *Field Calculator*, create a new field named "mix" as the **Output field name** and *Decimal number (real)* as the **Output field type** and use the following expression:

```
(sum($area, group_by= "naam", "lu_type_en" = 'residential')) /((sum($area, group_by= "naam", "lu_type_en" = 'residential' )+(sum($area, group_by= "naam", "lu_type_en" is not 'residential' )))
```

A new field was added to the attribute table of the `intersections` layer containing, for each feature, the values of the mixness of land values. Let's take a closer look at the expression we used to materialize the equation:

First, we calculate the total area, per station, of residential as the type of land use:



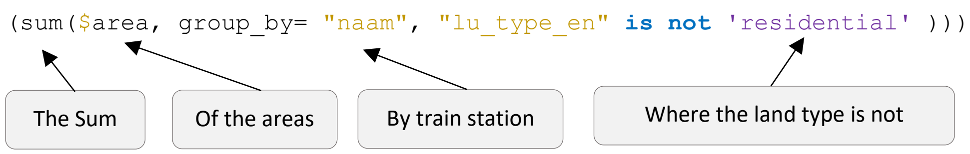

Here a third argument of the `sum` function is added: a 'where' condition which allow us to restrict the scope of the function only to those features that match the criteria, in this case the condition is `"lu_type_en" = 'residential'`

Then we divide

/

By:

```
((sum($area, group_by= "naam", "lu_type_en" = 'residential' )
+
```



- Create a new field using the *Field Calculator*. Provide “total_mix” as the **Output field name** and *Decimal number (real)* as the **Output field type** and use the following expression:

```
sum( "mix", group_by:="naam" )
```

- At this point, the `intersections` layer has more features than needed to proceed with the analysis. After calculating the sum of the values, we only have 22 distinct values under both “total_div” and “total_mix”. We will simplify the geometry and the attributes associated to it by dissolving the layer by station name.

**Dissolve fields [4]**

- From the *Processing toolbox*, filter by “dissolve” to find the **Dissolve** tool. Provide `intersections` as **Input layer**, `naam` as **Unique ID fields** and provide an output name – we suggest `dissolve`. Hit **Run** to execute (Figure 6).

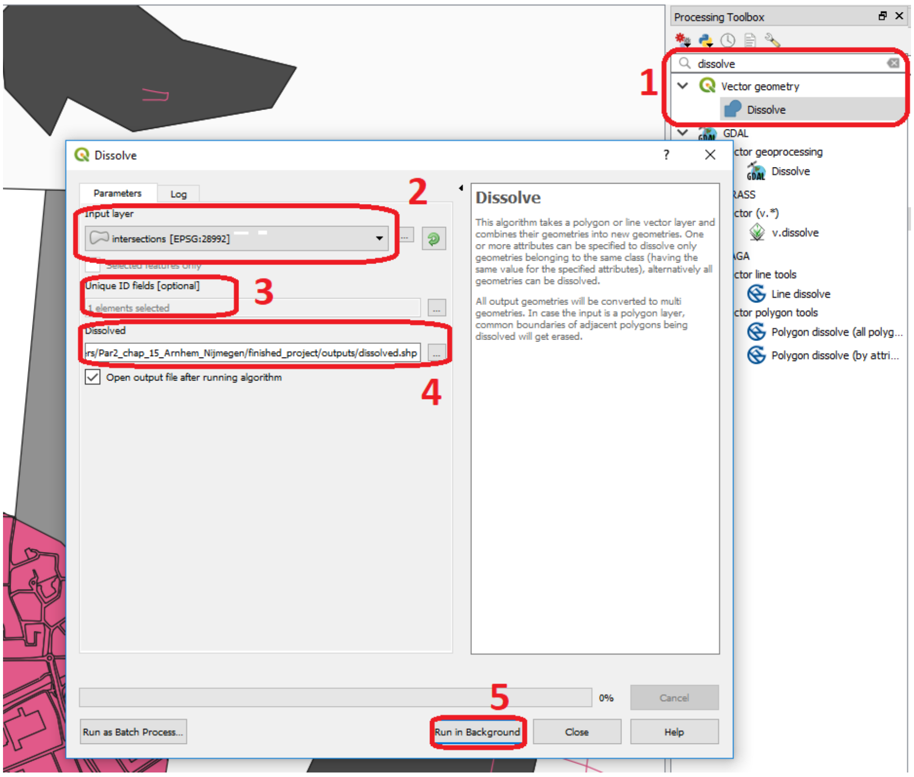



Figure 6 – Dissolve by attribute

- After the dissolve, you will have attributes, which are not meaningful. Even though they will not have any influence in the outcome of the next operations, It is advisable to delete those columns for the sake of data consistency.

**Drop fields (optional)**

- From the *Layers panel*, **right-click** on the layer `dissolve` and choose the option **Properties**. From the **Source Fields** start editing mode by clicking on the **Toggle editing**

**mode** button, mark the fields you want to delete, click on **Delete field** button and again on **Toggle editing mode** to save the changes. Hit **Ok** to dismiss the dialog (Figure 7).

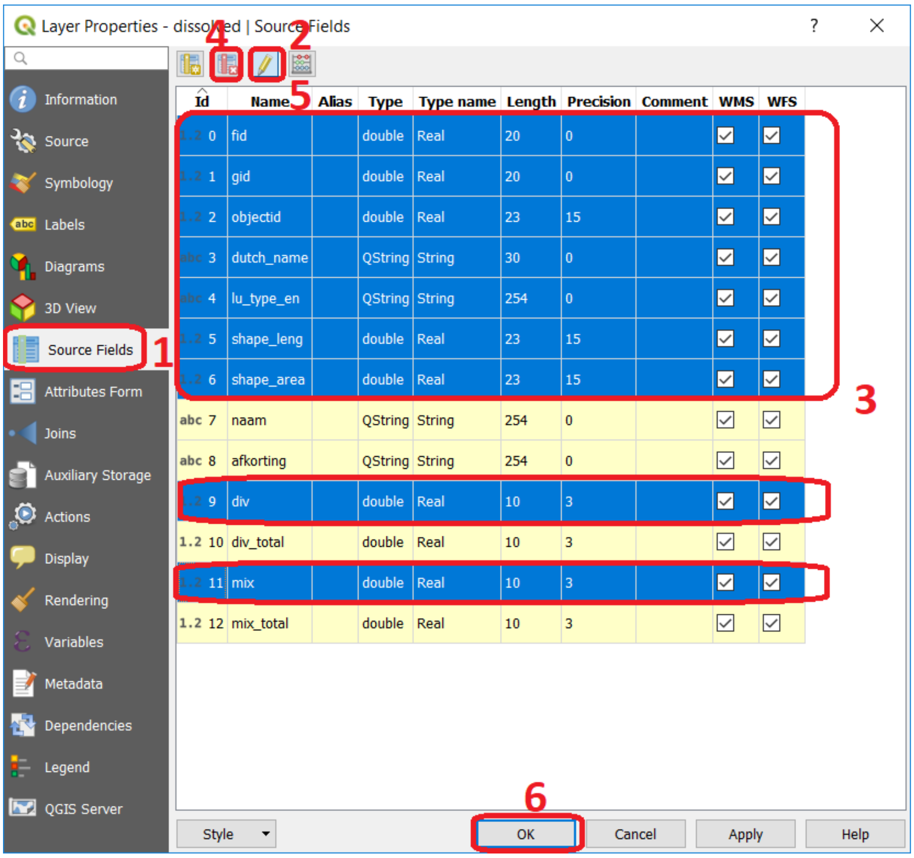


Figure 7 – Deleting fields

**Join by location [5]**

The third variable we need is be the population density.

Population density is aggregated by neighborhood; however, each window of analysis typically overlaps more than one neighborhood, which means that we cannot just take one of these values. To attenuate this problem we will take the value of the average population density of all the neighborhoods that overlap our window of analysis.

- From the *Processing toolbox*, filter by "join" to find the **Join attributes by location (summary)** tool. Provide `dissolve` as **Input layer**, and `cbs_demographics` as **Join layer.** In the **Fields to summarize** choose `POP_DNSTY` and in the **Summaries to calculate** choose *Mean*. Finally, mark the option **overlap** provide an output name – we suggest `pop_density`. Hit **Run** to execute (Figure 8).

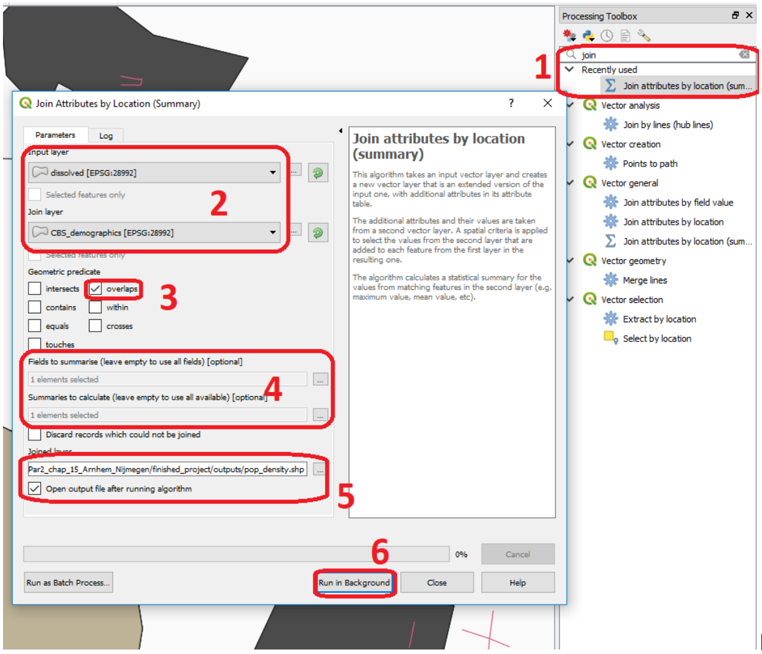



Figure 8 – Joining attributes by location

**Join by location [6]**

The fourth variable we need is the total length of walkable and cyclable roads.

- From the *Processing toolbox*, again use the **Join attributes by location (summary)** tool. Provide `pop_density` as **Input layer**, and `walkable_cyclable` as **Join layer.** In the **Fields to summarize** choose `lenght` and in the **Summaries to calculate** choose *sum*. Finally, mark the option **intersect** provide an output name – we suggest `walk_cycle`. Hit **Run** to execute (Figure 9).

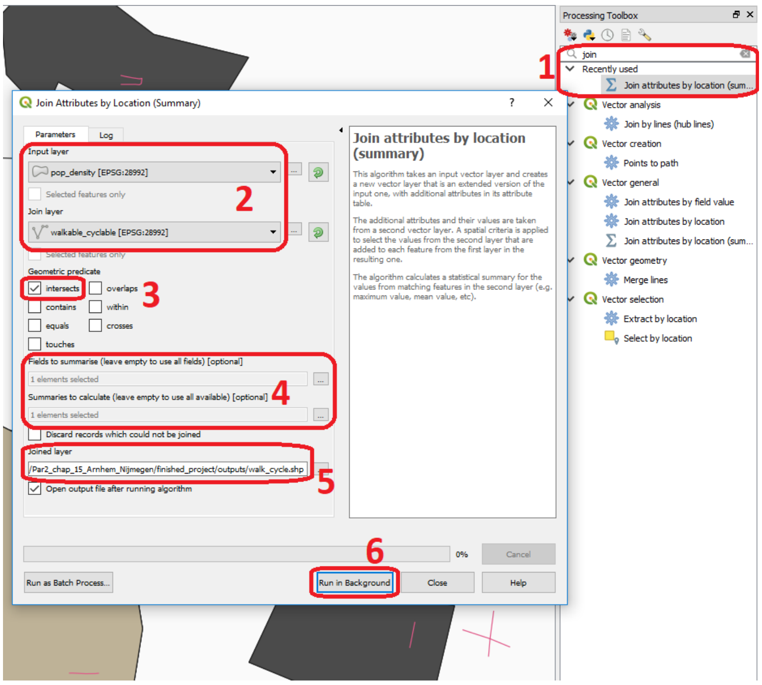



Figure 9 – Joining attributes by location

**Simplify geometries > drop nodes > Join by location [7]**

The last variable we need to calculate the actual TOD index is the number of road intersections. The approach taken here is to eliminate all the nodes of the `roads` layer that are not an intersection.

- From the *Processing toolbox*, filter by "simplify" to find the **Simplify** tool. Provide `roads` as **Input layer**, and choose *Distance (Douglas-Peucker)* as the **Simplification method**. Provide 500 as the **Tolerance** value and provide an output name – we suggest `roads_simplified`. Hit **Run** to execute (Figure 10).

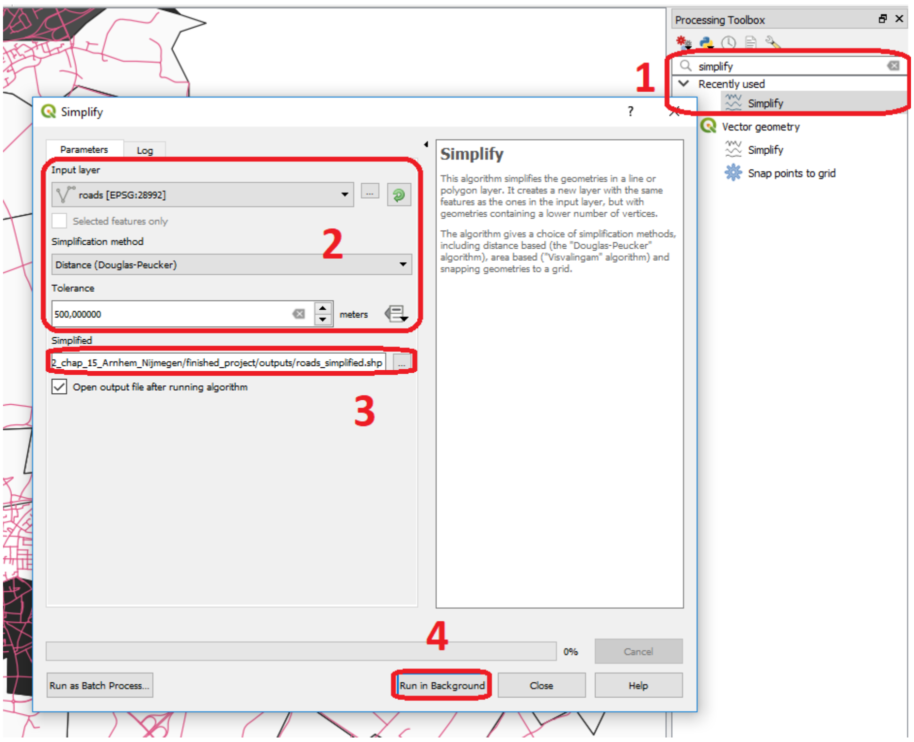



Figure 10 – simplifying (line) geometries

The simplification removes all the nodes in between intersections while preserving the location of the nodes that correspond to an intersection in the same location as before. Therefore, this operation preserves the topology of the network. The 500m threshold is simply a generous threshold that assumes that in a dense urban landscape hardly do we find a road that does not fork within 500m from the previous fork. Below, an illustration of this reasoning (Figure 11).

The following steps will consist of:

- Extracting the intersection nodes as point geometries;
- Add a count field to the attribute table of the point geometries layer;
- Use a **Join by location (summary)** to add the *sum* of the points count to the attribute table layer of the `walk_cycle`

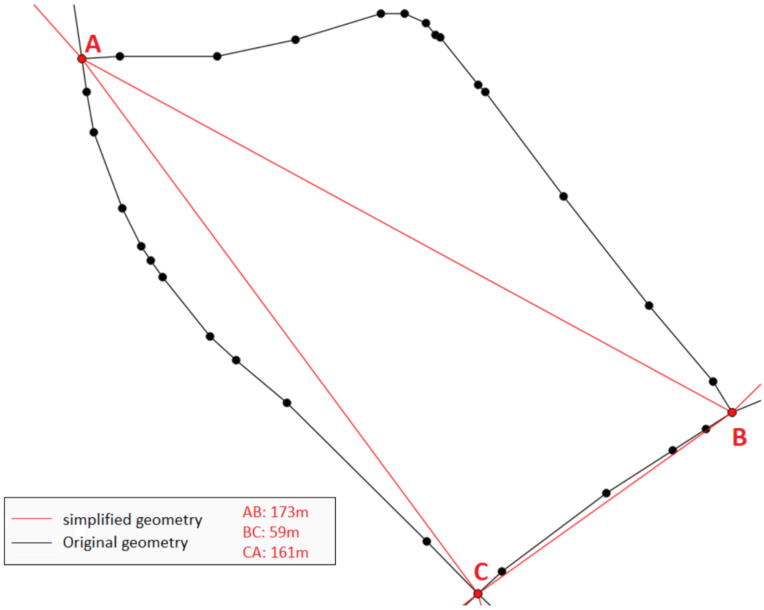



Figure 11 – Isolating network nodes that represent an intersection

- From the *Processing toolbox*, filter by “nodes” to find the **Extract nodes** tool. Provide `roads_simplified` as **Input layer** and provide an output name – we suggest `intersections`. Hit **Run** to execute (Figure 12).

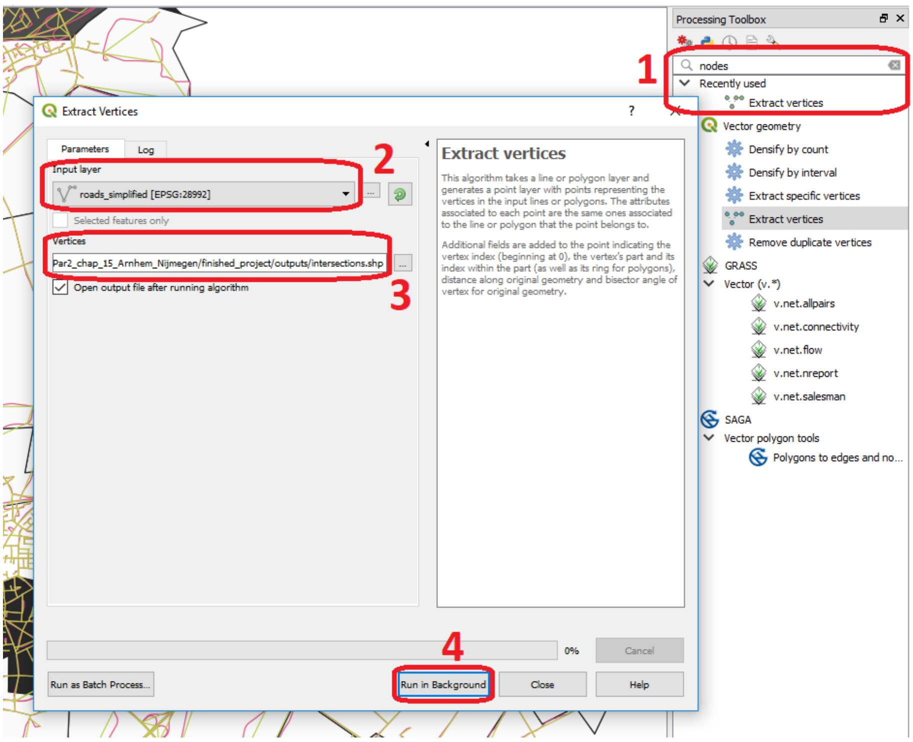



Figure 12 – Extracting nodes

- Use the **Field Calculator** to add a new field to the layer `intersections`. In the *Field Calculator* widow choose **Create a new field** and provide "count" as the **Output field name** and *Whole number (integer)* as the **Output field type** and enter '1' as the expression to be used(Figure 13):

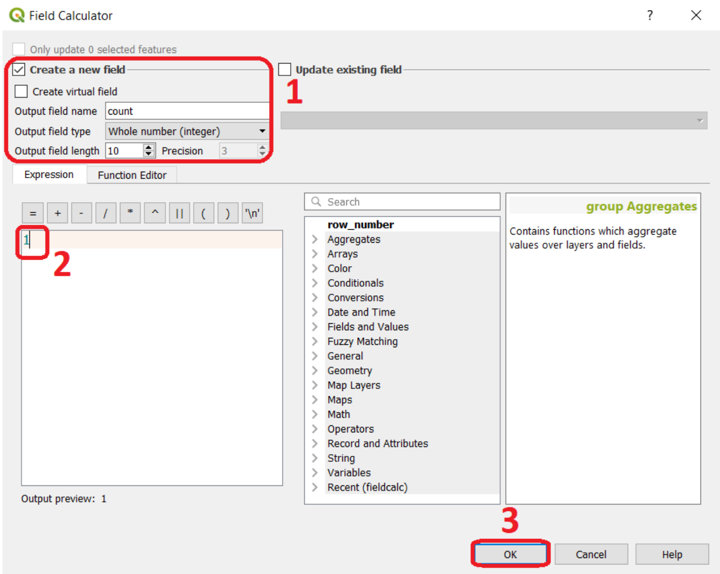



Figure 13 – Using the field calculator

- From the *Processing toolbox*, again use the **Join attributes by location (summary)** tool. Provide `walk_cycle` as **Input layer**, and `intersections` as **Join layer.** In the **Fields to summarize** choose `count` and in the **Summaries to calculate** choose *sum*. Finally, mark the option **intersect** provide an output name – we suggest `actual_tod`. Hit **Run** to execute (Figure 14).

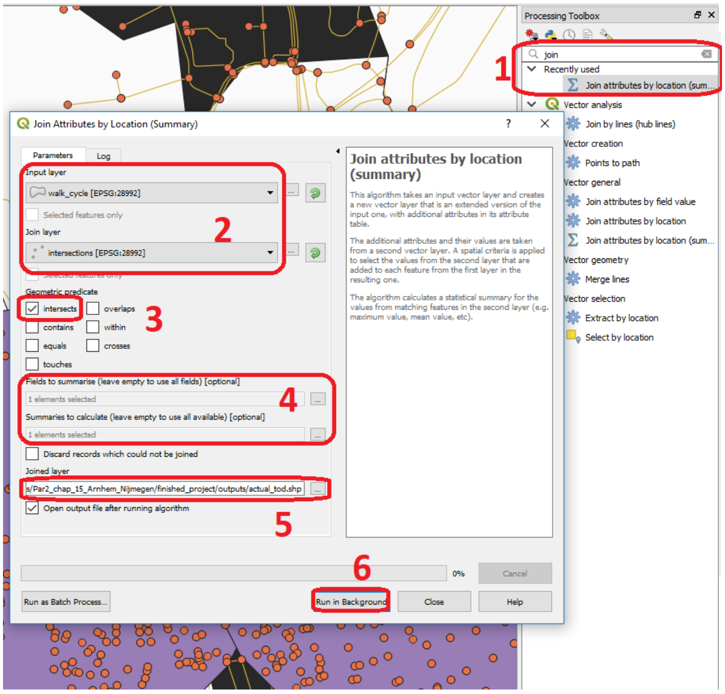
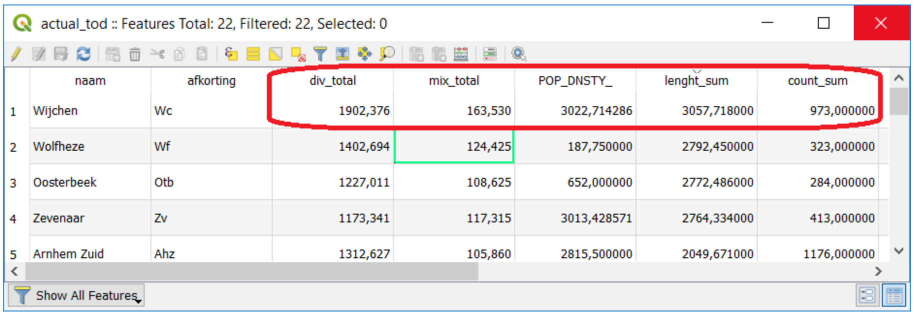



Figure 14 – Using the field calculator

We now have in the layer `actual_tod` all the variables we need to calculate the Actual TOD (figure 15).




| | naam | afkorting | div_total | mix_total | POP_DNSTY_ | lenght_sum | count_sum |
|---|---|---|---|---|---|---|---|
| 1 | Wijchen | Wc | 1902,376 | 163,530 | 3022,714286 | 3057,718000 | 973,000000 |
| 2 | Wolfheze | Wf | 1402,694 | 124,425 | 187,750000 | 2792,450000 | 323,000000 |
| 3 | Oosterbeek | Otb | 1227,011 | 108,625 | 652,000000 | 2772,486000 | 284,000000 |
| 4 | Zevenaar | Zv | 1173,341 | 117,315 | 3013,428571 | 2764,334000 | 413,000000 |
| 5 | Arnhem Zuid | Ahz | 1312,627 | 105,860 | 2815,500000 | 2049,671000 | 1176,000000 |

Figure 15 – Actual_tod attribute table

However these variables have different value ranges, which means that we cannot use them in a on a weighted average based multi-criteria analysis, which is what we need to do. To solve this problem we first have to normalize the data.

Normalizing refers to the process of rescaling the data so that the range of values covered by the data is the same making it comparable. In our case, a simple min-max normalization is used:

$$x' = \frac{x - average(x)}{(\max(x) - \min(x))}$$

This will rescale all our variables to range of 0 to 1.

**Field calculator [8]**

Use the **Field Calculator** to add new fields to the layer `actual_tod` according to the table below.

| Field name | Field type | Expression |
|---|---|---|
| div_norm | real | `("div_total" - minimum("div_total")) / (maximum("div_total") - minimum("div_total"))` |
| mix_norm | real | `("mix_total" - minimum("mix_total")) / (maximum("mix_total") - minimum("mix_total"))` |
| popd_norm | real | `("POP_DNSTY_" - minimum("POP_DNSTY_")) / (maximum("POP_DNSTY_") - minimum("POP_DNSTY_"))` |
| leng_norm | real | `("lenght_sum" - minimum("lenght_sum")) / (maximum("lenght_sum") - minimum("lenght_sum"))` |
| int_norm | real | `("count_sum" - minimum("count_sum")) / (maximum("count_sum") - minimum("count_sum"))` |

To get final TOD index for each station we just need to perform a weighted average.

Use the **Field Calculator** to add one last field to the layer `actual_tod`. Provide "tod_index" as the **Output field name**, *real* as **Output field type** and the expression:

```
"div_norm"*0.2+"mix_norm"*0.2+"popd_norm"*0.2+"leng_norm"
*0.2+"int_norm"*0.2
```

The `actual_tod` layer should end up with this final set of fields (Figure 16):

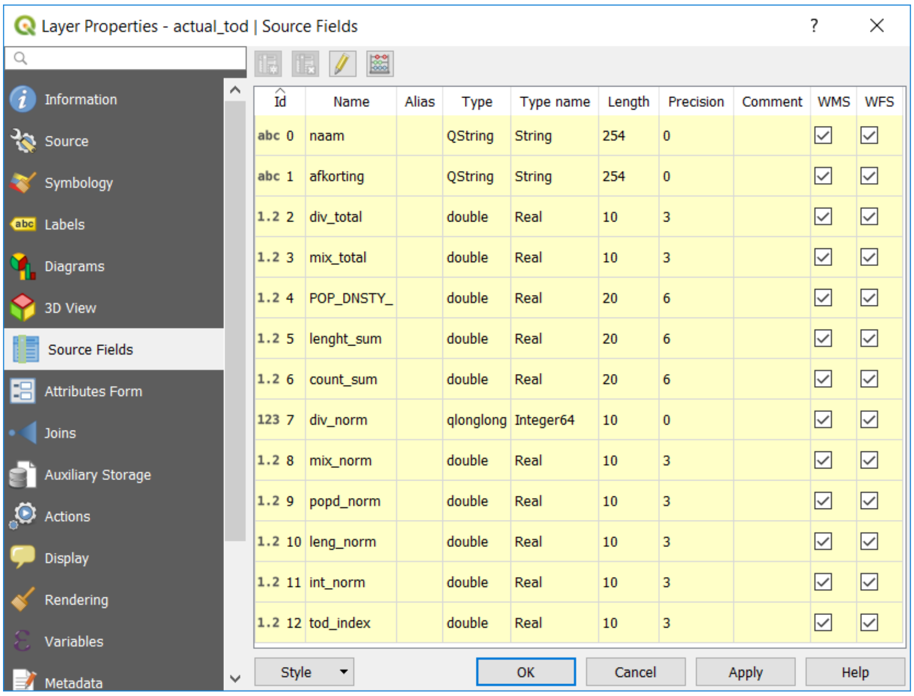



Figure 16 – Actual_tod field map

To visualize in a map the stations according to their tod_index, we have to style the data.

- From the *Layers panel*, **right-click** on the layer `actual_tod` and access the layer´s **Properties**. At any tab In the *Layer Properties* dialog click on the **Style** button, choose **Load Style** and point to the corresponding style file which is named "actual_tod". This style file is included in the data of this exercise, it can be found inside the folder "Styles". Finish by clicking on **Apply** (Figure 17)

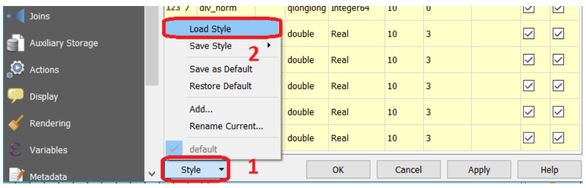



Figure 17 – Applying a style

Your map map should look like this (Figure 18). Depending on which layers you have active, and their order, the map you see might look a bit different. However, the `actual_tod` layer should be exactly what you see here.

If the `actual_tod` layer is not styled exactly as seen below, most likely you have given different field names along the process. Check if you have a "tod_index" and a "naam" in your attribute table, if You are missing one of them the style will not be applied.

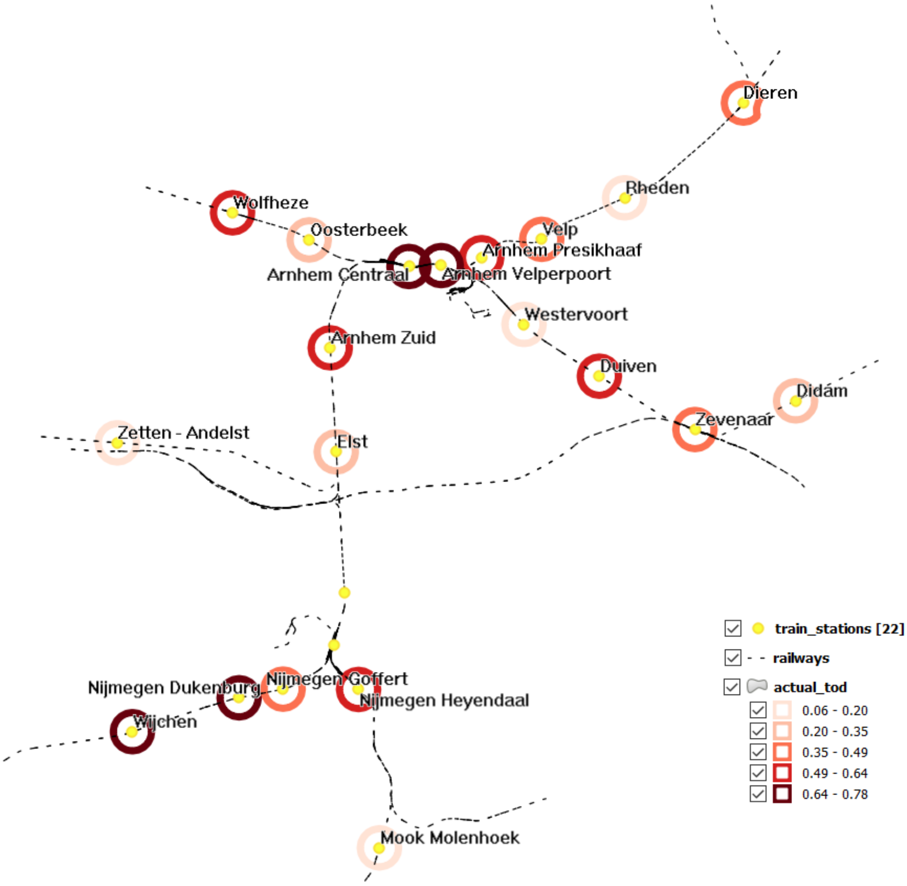



Figure 18 – Actual TOD index map

The style you applied essentially takes the "tod_index" field and builds a graduated symbol from it, grading it with warm colors from light to heavy. Finally, the name of the stations is used as label.

Please realize that the tod index value attributed to any given station can change substantially depending on how you weight the five criteria. The "tod_index" field was calculated assuming each of the criteria had exactly the same weight/influence (i.e 20%, hence the 0.2 multiplication factor used for all the criteria) and it was based on this calculation that the data was styled. Changing the weights attributed to the criteria could substantially change the tod index value of a station.

**SECOND PART – POTENTIAL TOD**

In the article, for the Potential TOD, a grid of 300X300m was covering the extent of the study area was used. For this demonstration, we simplified the procedure by using 800X800m grid instead. A larger grid allows faster computations time and does not compromise the main objective of demonstrating the methodology that was used.

On top of this 800X800m grid the same five variables used for the actual TOD index are used. Therefore the steps described in the previous pages are exactly the same, only difference is that instead of having the TOD Index being calculated into 22 window of analysis (i.e. made of a 800m buffer from  train stations) they have to be calculated for every 1240 cells that make up the 800X800m grid.

Because we don't see any added value in going through the same procedure all over again, we simply provide you with the final version of the grid. You just have to style it and visually compare it with the actual TOD.

- From the *Layers panel*, **right-click** on the layer `potential_tod` and access the layer´s **Properties**. At any tab In the *Layer Properties* dialog click on the **Style** button, choose **Load Style** and point to the corresponding style file which is named "potential_tod". This style file is included in the data of this exercise, it can be found inside the folder "Styles".  Finish by clicking on **Apply** (Figure 19)

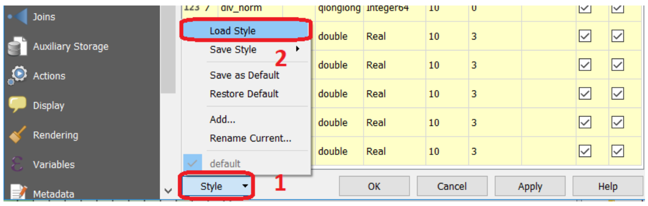



Figure 19 – Applying a style

After applying the style, depending on which layers are active, and their order, you should get a final representation of the actual and potential TOD indexes (Figure 20).

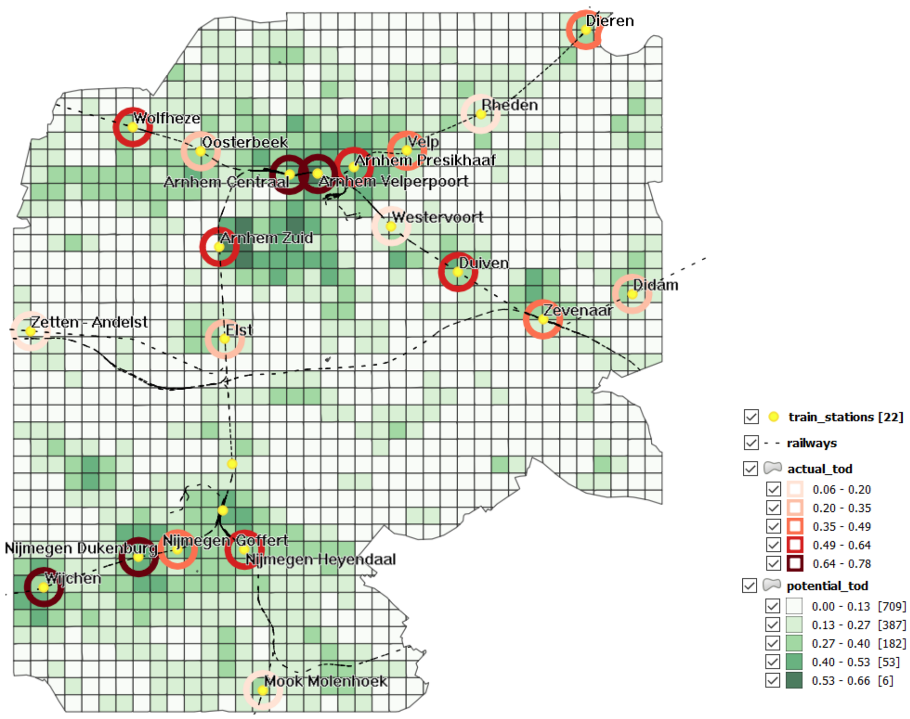



Figure 20 – actual_tod and potential TOD index map

## Flowchart of operations