Tutorial: Museums and Heritage within Walking Distance of the Station

PRO Module: GIS & GEODATA – Network Analysis

Goal:	Analyse which museums, churches and cultural locations in Breda or Tilburg are within walking (or cycling) distance of the train station
Skills:	Geofabrik data loading, walk/cycle network filter, Service Area, Convex Hull, Difference, Count, Graduated symbolisation, Shortest Path layer to point, Print Layout
Time:	approx. 2 hours
Software:	QGIS 3.x | Geofabrik Noord-Brabant download | PDOK municipality boundary

Introduction: Is Heritage Accessible for Everyone?

Museums, churches, monuments and civic buildings are public facilities – but are they truly accessible for visitors without a car? Tourists and day-trippers arriving by train walk or cycle from the station to their destination. The question is: which cultural locations can they reach on foot, and how long does it take?

In this tutorial you analyse which cultural locations in Breda or Tilburg are realistically within walking distance of the central station. You use real OSM data from Geofabrik and the network analysis tools in QGIS.

Central research question:
"Which museums, churches and cultural locations in Breda (or Tilburg) are within walking distance of the train station, and how accessible are they compared to each other?"

PART 1: Data Download and Loading (25 min)

Exercise 1: Download Geofabrik Noord-Brabant

Step 1: Download the file

1. Go to: download.geofabrik.de/europe/netherlands/noord-brabant
2. Click on noord-brabant-latest-free.shp.zip
3. Save to your working folder, e.g. C:\GIS\Heritage\ and extract the ZIP

Step 2: Identify the files you need

After extracting you will see many files. You only need four:

File	Contents	Use
gis_osm_roads_free_1.shp	All roads and paths	Build walk/cycle network
gis_osm_transport_free_1.shp	Transport points (stations, stops)	Extract train station
gis_osm_pois_free_1.shp	Points of Interest	Museums, churches, monuments
gis_osm_buildings_a_free_1.shp	Building footprints (polygons)	Analysis 2: walking time to buildings

Why separate files for station and museums?
Geofabrik stores transport locations in a separate file: gis_osm_transport_free_1.shp. Train stations are NOT in the POI file. Museums, churches and monuments are in the POI file.

Exercise 2: Load the basemap and set the project CRS

Load the Positron basemap first – it helps you orientate yourself while loading and checking data. Then set the project coordinate system so all network measurements work in metres.

Step 1: Load the CartoDB Positron basemap

4. Check that QuickMapServices is installed: Plugins → Manage and Install Plugins → search **QuickMapServices** → Install if not present
5. Web → QuickMapServices → **CartoDB** → **CartoDB Positron**
6. The basemap is added at the bottom of the layers panel

Step 2: Set the project CRS to EPSG:28992

Geofabrik data is delivered in WGS84 (EPSG:4326, coordinates in degrees). Network analysis tools measure distances in the CRS units. On WGS84 the Service Area would measure in degrees rather than metres – all distances would be wrong. Switch to the Dutch national grid RD New (EPSG:28992) which works in metres.

7. Click the CRS indicator in the bottom-right corner of QGIS (shows e.g. EPSG:4326)
8. Search **28992** → select **RD New** → OK

Let op:
Always set CRS: EPSG:28992 on every export.

On every Save Features As step in this tutorial: set the CRS field explicitly to EPSG:28992. This ensures all network tools measure correctly in metres.

Exercise 3: Load the Geofabrik files

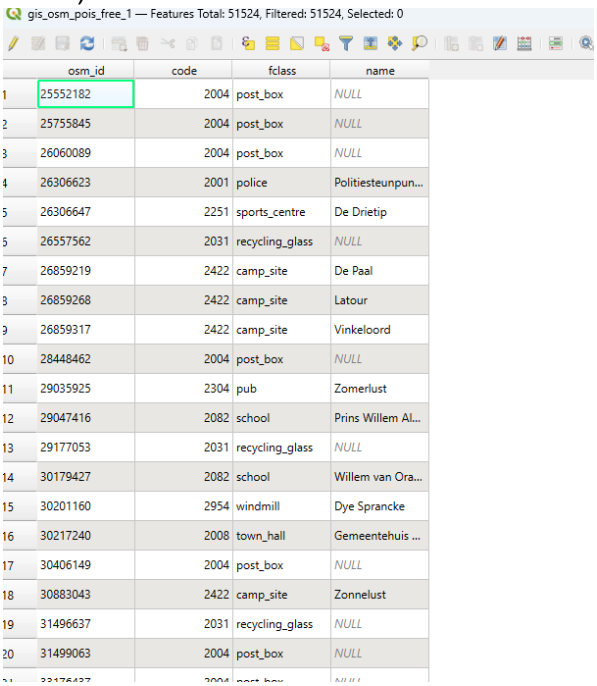
Step 1: Load all four files

9. Open QGIS and create a new project (**Project** → **New**)
10. Layer → Add Layer → **Add Vector Layer**
11. Select **gis_osm_roads_free_1.shp** → Open
12. Repeat for **gis_osm_transport_free_1.shp**
13. Repeat for **gis_osm_pois_free_1.shp**
14. Repeat for **gis_osm_buildings_a_free_1.shp**

Let op:
Noord-Brabant is a large file. Loading may take 30–60 seconds. Zoom in to Breda or Tilburg afterwards to speed up the map.

Step 2: Check the POI data

15. Open the attribute table of **gis_osm_pois_free_1** (right-click → Open Attribute Table)



	osm_id	code	fclass	name
1	25552182	2004	post_box	NULL
2	25755845	2004	post_box	NULL
3	26060089	2004	post_box	NULL
4	26306623	2001	police	Politiesteunpun...
5	26306647	2251	sports_centre	De Drietip
6	26557562	2031	recycling_glass	NULL
7	26859219	2422	camp_site	De Paal
8	26859268	2422	camp_site	Latour
9	26859317	2422	camp_site	Vinkeloord
10	28448462	2004	post_box	NULL
11	29035925	2304	pub	Zomerlust
12	29047416	2082	school	Prins Willem Al...
13	29177053	2031	recycling_glass	NULL
14	30179427	2082	school	Willem van Ora...
15	30201160	2954	windmill	Dye Sprancke
16	30217240	2008	town_hall	Gemeentehuis ...
17	30406149	2004	post_box	NULL
18	30883043	2422	camp_site	Zonnelust
19	31496637	2031	recycling_glass	NULL
20	31499063	2004	post_box	NULL

16. Find the field fclass and sort on it – do you see museum, church, townhall, memorial?

Step 3: Check the transport data

17. Open the attribute table of **gis_osm_transport_free_1**
18. Find the field fclass and sort on it – do you see railway_station?

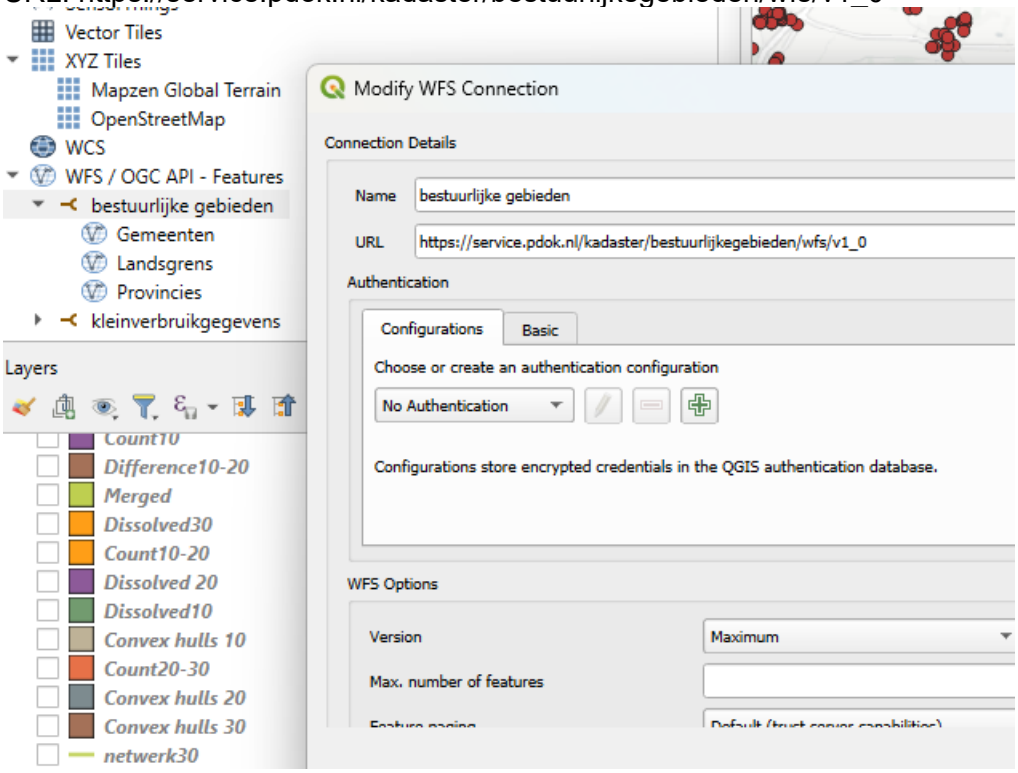
Verification:
Transport file: fclass contains railway_station and tram_stop. POI file: fclass contains museum, church, townhall, memorial.

Exercise 4: Get the municipality boundary via PDOK

You will clip the large Noord-Brabant files down to your municipality. First get the boundary.

Step 1: Load the municipalities layer via WFS

19. Layer → Add Layer → **Add WFS Layer**
20. Click New and fill in:
    - Name: PDOK Municipalities
    - URL: https://service.pdok.nl/kadaster/bestuurlijkegebieden/wfs/v1_0



21. Click **Connect**
22. Select the layer **Gemeenten** → Add → Close

Step 2: Filter and export your municipality

23. Right-click the municipalities layer → **Filter**

Enter one of the following (choose your city):

```
"naam" = 'Breda'
```

— or —

```
"naam" = 'Tilburg'
```

24. Click OK
25. Right-click → Export → **Save Features As** → GeoPackage → CRS: EPSG:28992 → save as **municipality.gpkg**

> **Verification:**
> Zoom to the municipality layer (right-click → Zoom to Layer). Can you see the boundary of your city?

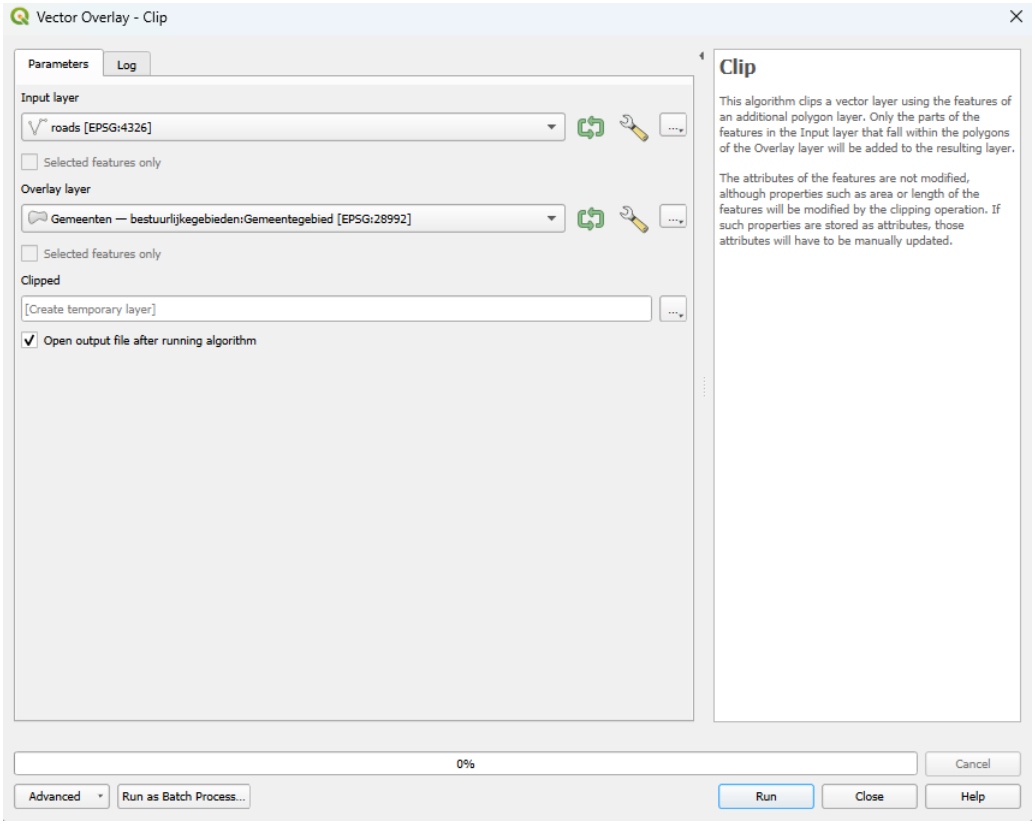
Exercise 5: Clip all data to the municipality

Clip all four Geofabrik files to your municipality. This makes every subsequent step much faster. Remember to set CRS: EPSG:28992 on each export.

Step 1: Clip roads

26. Processing → **Clip**
- Input layer: gis_osm_roads_free_1
- Overlay layer: municipality.gpkg
- Output: roads_clipped.gpkg | CRS: EPSG:28992

27. Run



Step 2: Clip transport

28. Processing → **Clip**
- Input layer: gis_osm_transport_free_1
- Overlay layer: municipality.gpkg
- Output: transport_clipped.gpkg | CRS: EPSG:28992

29. Run

Step 3: Clip POIs

30. Processing → **Clip**
- Input layer: gis_osm_pois_free_1
- Overlay layer: municipality.gpkg
- Output: pois_clipped.gpkg | CRS: EPSG:28992

31. Run

Step 4: Clip buildings

32. Processing → **Clip**
- Input layer: gis_osm_buildings_a_free_1
- Overlay layer: municipality.gpkg
- Output: buildings_clipped.gpkg | CRS: EPSG:28992

33. Run

Verification:
roads_clipped.gpkg: approx. 30,000–60,000 road segments
transport_clipped.gpkg: approx. 20–100 transport points
pois_clipped.gpkg: approx. 500–1,500 points
buildings_clipped.gpkg: tens of thousands of building footprints

Exercise 6: Extract train station and cultural locations

Extract two separate layers: the train station from the transport file, and the cultural destinations from the POI file.

Which fclass values are relevant?

Transport file (transport_clipped.gpkg):

fclass value	What is it?	Include?
railway_station	Train station	Yes – starting point
tram_stop, bus_stop, subway_entrance	Public transport stops	No

POI file (pois_clipped.gpkg):

fclass value	What is it?	Include?
museum	Museum	Yes
church	Church – often landmark buildings	Yes
townhall	Town hall / civic building	Yes
memorial	Monument / memorial	Yes
monument	Monument	Yes
castle	Castle	Yes
ruins	Ruins	Yes
artwork	Public artwork	Optional
tram_stop, bus_stop	Transport stops	No

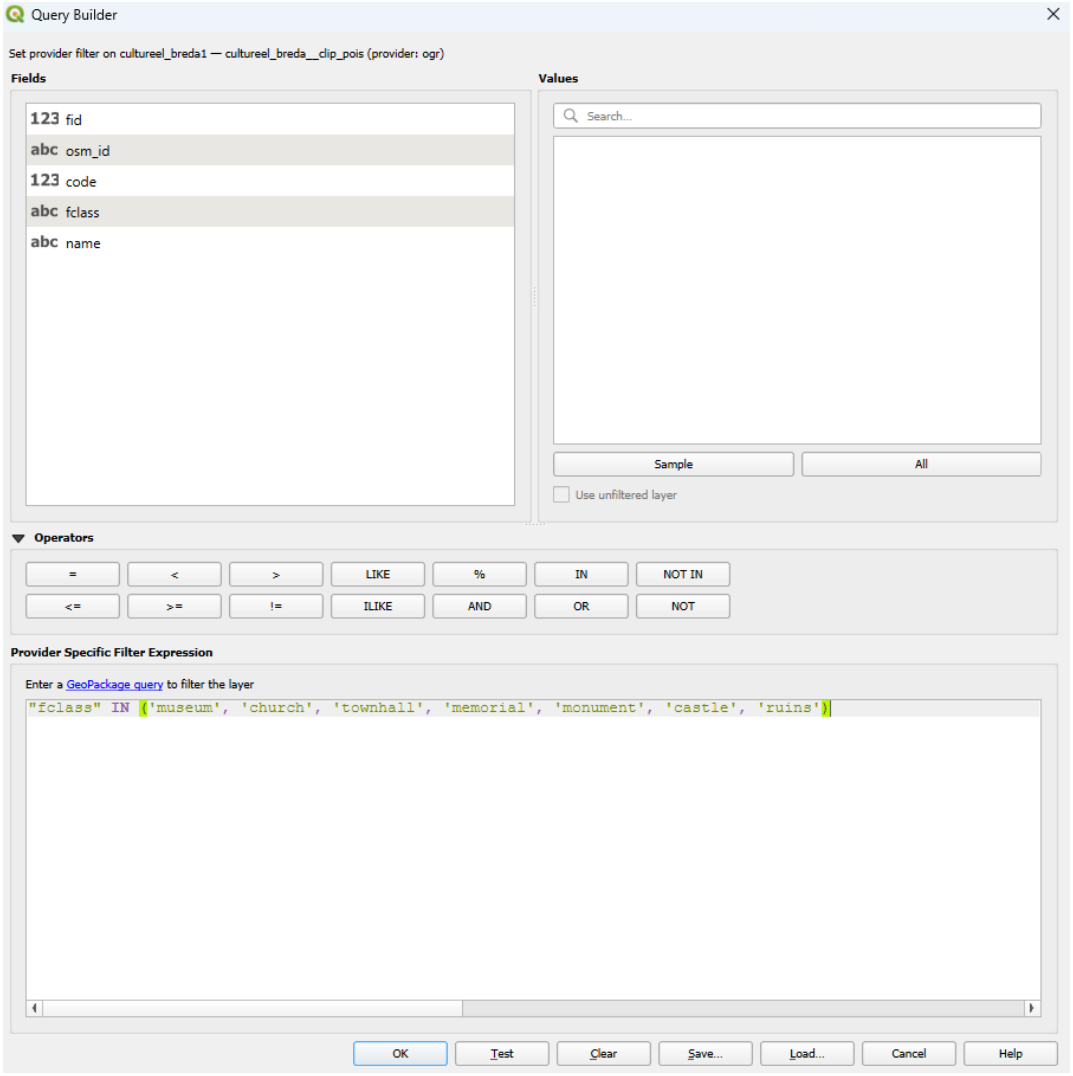
Let op:
Not all fclass values appear in every city. Always check which values are present in your dataset before filtering.

Step 1: Extract the train station

34. Right-click **transport_clipped.gpkg** → Filter
    ```
    "fclass" = 'railway_station'
    ```
35. Click OK
36. Export → Save Features As → GeoPackage → CRS: EPSG:28992 → save as **station.gpkg**
37. Remove the filter afterwards (right-click → Filter → clear expression → OK)

Step 2: Extract cultural locations

38. Right-click **pois_clipped.gpkg** → Filter
    ```
    "fclass" IN ('museum', 'church', 'townhall', 'memorial', 'monument', 'castle', 'ruins')
    ```



39. Click OK
40. Export → Save Features As → GeoPackage → CRS: EPSG:28992 → save as **cultural.gpkg**

41. Remove the filter

Verification:
station.gpkg: 1–3 features (Breda CS + possibly Breda Prinsenbeek)
cultural.gpkg: 20–100 features. Open the attribute table – do you see recognisable names such as “Breda’s Museum”, “Grote Kerk” or “Stadhuis”?

PART 2: Build the Network and Choose Your Mode (15 min)

Choose your mode: Option A or Option B
Option A – Walking (public transport + walking): Speed: 80 m/min.
Option B – Cycling (public transport + OV-bike): Speed: 250 m/min (= 15 km/h).
The network is almost identical for both options – only the distances and a small filter difference change.

	Option A – Walking	Option B – Cycling
Speed	80 m/min	250 m/min
10 minutes	800 m	2,500 m
20 minutes	1,600 m	5,000 m
30 minutes	2,400 m	7,500 m
Network file	walk_network.gpkg	cycle_network.gpkg
Extra excluded roads	–	footway, pedestrian, steps

Exercise 7: Build the network for your chosen mode

Option A – Walk network:

42. Right-click **roads_clipped.gpkg** → **Filter**
```
"fclass" NOT IN ('motorway', 'motorway_link', 'trunk', 'trunk_link')
```
43. OK → Export → Save Features As → GeoPackage → CRS: EPSG:28992 → **walk_network.gpkg**

Option B – Cycle network:

44. Right-click **roads_clipped.gpkg** → **Filter**
```
"fclass" NOT IN ('motorway', 'motorway_link', 'trunk', 'trunk_link',
'footway', 'pedestrian', 'steps')
```
45. OK → Export → Save Features As → GeoPackage → CRS: EPSG:28992 → **cycle_network.gpkg**

Let op:
Filter vs. Export: A filter is temporary and disappears when QGIS closes. Always export as a permanent GeoPackage so network analysis tools can use it.

Verification:
Zoom to the city edge – have the motorways A27/A16 disappeared from the layer? For Option B: have the footpaths also disappeared?

Analysis 1: Accessibility Zones per Station (35 min)

You calculate accessibility zones for 10, 20 and 30 minutes from all stations simultaneously. Each zone gets a separate polygon per station. Then you create exclusive rings and count how many cultural locations fall within each ring.

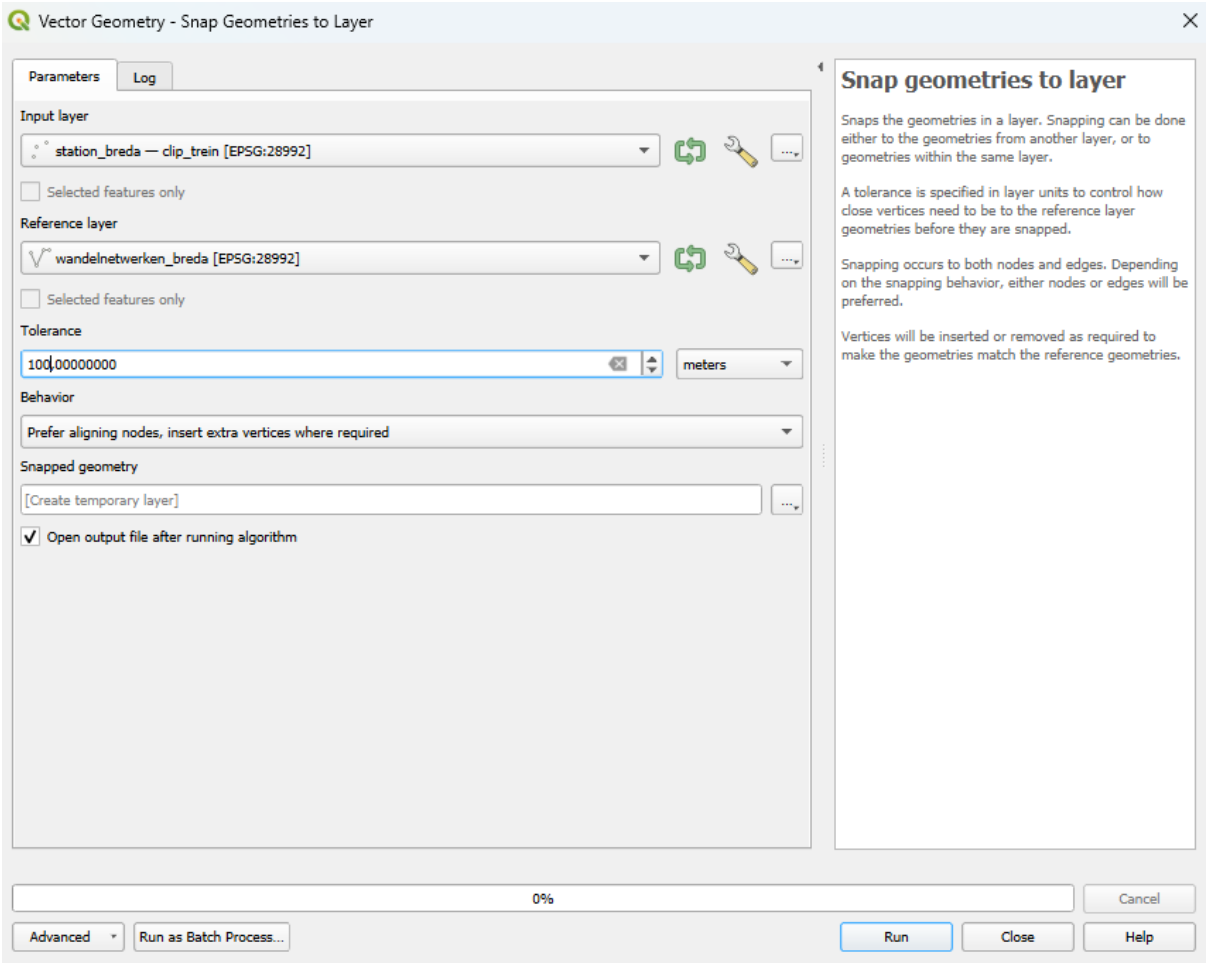
Use the distances that match your chosen mode
Option A – Walking: use 800 m, 1,600 m and 2,400 m as Travel cost.
Option B – Cycling: use 2,500 m, 5,000 m and 7,500 m as Travel cost.
Use the matching network: walk_network.gpkg (A) or cycle_network.gpkg (B).

Exercise 8: Snap stations to the network and check manually

Stations sometimes sit slightly outside the road network. If the start point is too far from a road, the Service Area finds no route. Snap geometries to layer moves each station point to the nearest point on the network. Note: the algorithm always picks the closest line – that is not always the correct one. Check each point manually afterwards.

Step 1: Automatic snapping

46. Processing Toolbox → search: **Snap geometries to layer**
    - Input layer: station.gpkg
    - **Reference layer:** walk_network.gpkg (Option A) or cycle_network.gpkg (Option B)
    - Tolerance: 100 (metres)
    - Behavior: Prefer closest point, insert extra vertices where required
    - Output: CRS: EPSG:28992 → station_snapped.gpkg
47. Run



Step 2: Check each station visually

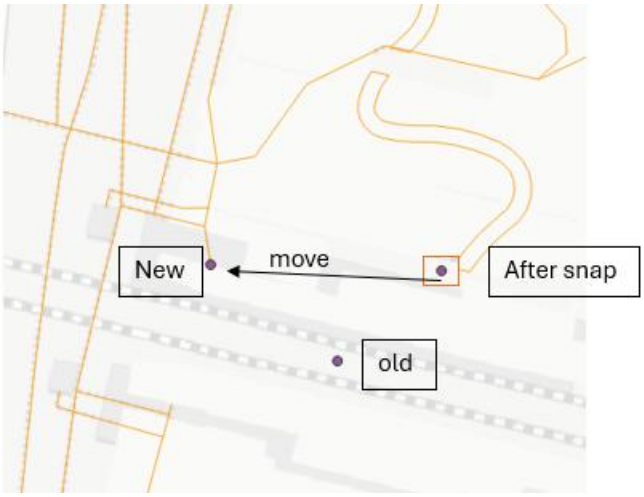
Zoom in to each station in station_snapped.gpkg and check which road the point has snapped to:

- Is the point on a road suitable for pedestrians – not a motorway or industrial access road?
- Is the point near the station entrance?
- Compare the snapped point with the satellite basemap or the OSM background.

Let op:
Tilburg Reeshof – check this station carefully.
At Tilburg Reeshof the algorithm sometimes snaps to a traffic road instead of a pedestrian street. Check that the point is on a residential street or footpath near the station entrance. If not, correct it manually in Step 3.

Step 3: Manual correction if the point is on the wrong road

48. Right-click **station_snapped.gpkg** → Toggle Editing (pencil icon on)
49. Press **V** to activate the Vertex Tool
50. Click the point you want to move – a red marker appears
51. Drag the point to the correct road: a footpath or residential street near the station entrance
52. Toggle Editing off → click **Save** in the dialog



Verification:
Overlay station_snapped.gpkg on walk_network.gpkg. Every point must visibly sit on a coloured line (= walkable road). No point should be floating in empty space or on a motorway.

Exercise 9: Service Area and zones per station

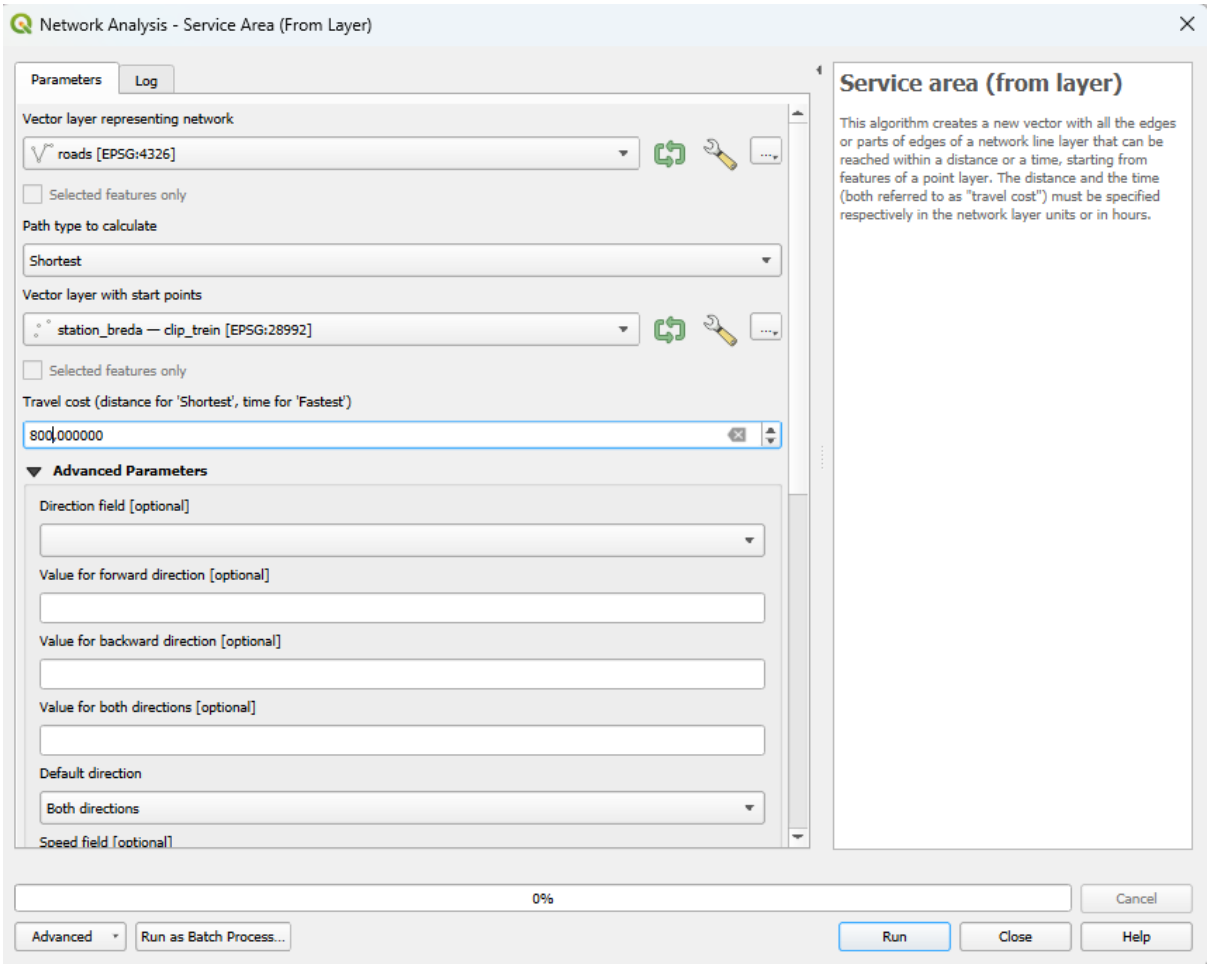
Calculate the reachable road segments from all stations simultaneously. Then convert the lines to polygons using Convex Hull – QGIS automatically creates a separate polygon per station.

Step 1: Calculate the Service Area (repeat for all three distances)

53. Processing Toolbox → **Service area (from layer)**

- **Vector layer representing network:** walk_network.gpkg (A) or cycle_network.gpkg (B)
- Path type: Shortest
- Vector layer with start points: station_snapped.gpkg
- **Travel cost Option A (walking):** 800 → sa_lines_10min.gpkg | 1600 → sa_lines_20min.gpkg | 2400 → sa_lines_30min.gpkg
- **Travel cost Option B (cycling):** 2500 → sa_lines_10min.gpkg | 5000 → sa_lines_20min.gpkg | 7500 → sa_lines_30min.gpkg

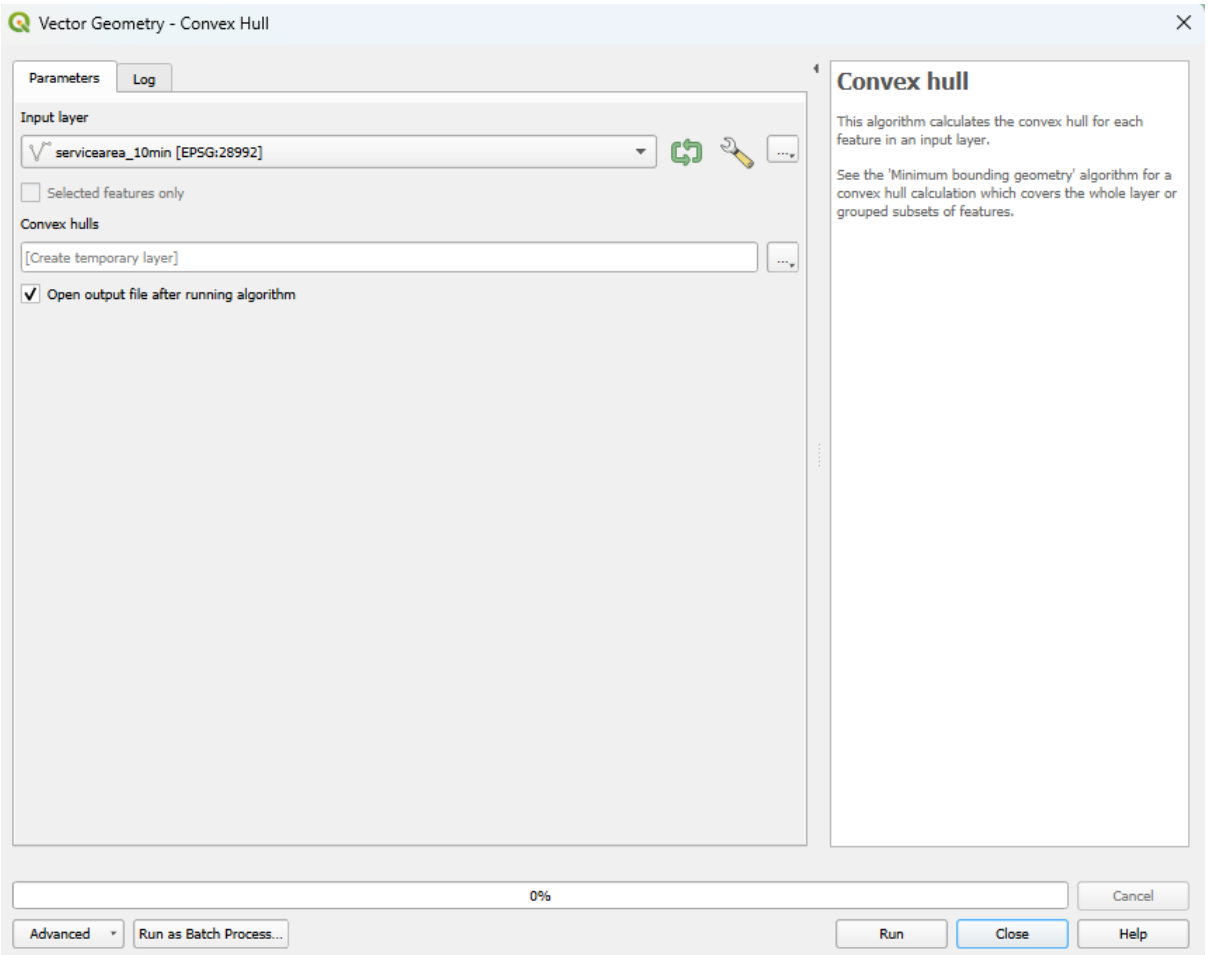
54. Run three times with the matching distances



Step 2: Convex Hull per station (repeat for all three distances)

55. Processing Toolbox → **Convex hull**
- Input layer: sa_lines_10min.gpkg
- Output: zone_10min.gpkg

56. Repeat for sa_lines_20min.gpkg → zone_20min.gpkg and sa_lines_30min.gpkg → zone_30min.gpkg



Verification:
Open the attribute table of zone_10min.gpkg. Are there 5 rows – one per station? Each station now has its own accessibility zone polygon.

Exercise 10: Difference – create exclusive rings per station

You have three layers: zone_10min, zone_20min and zone_30min, each with five polygons (one per station). By applying Difference directly to these layers you get three ring layers, each still with five separate features. Count Points in Polygon then counts per station individually.

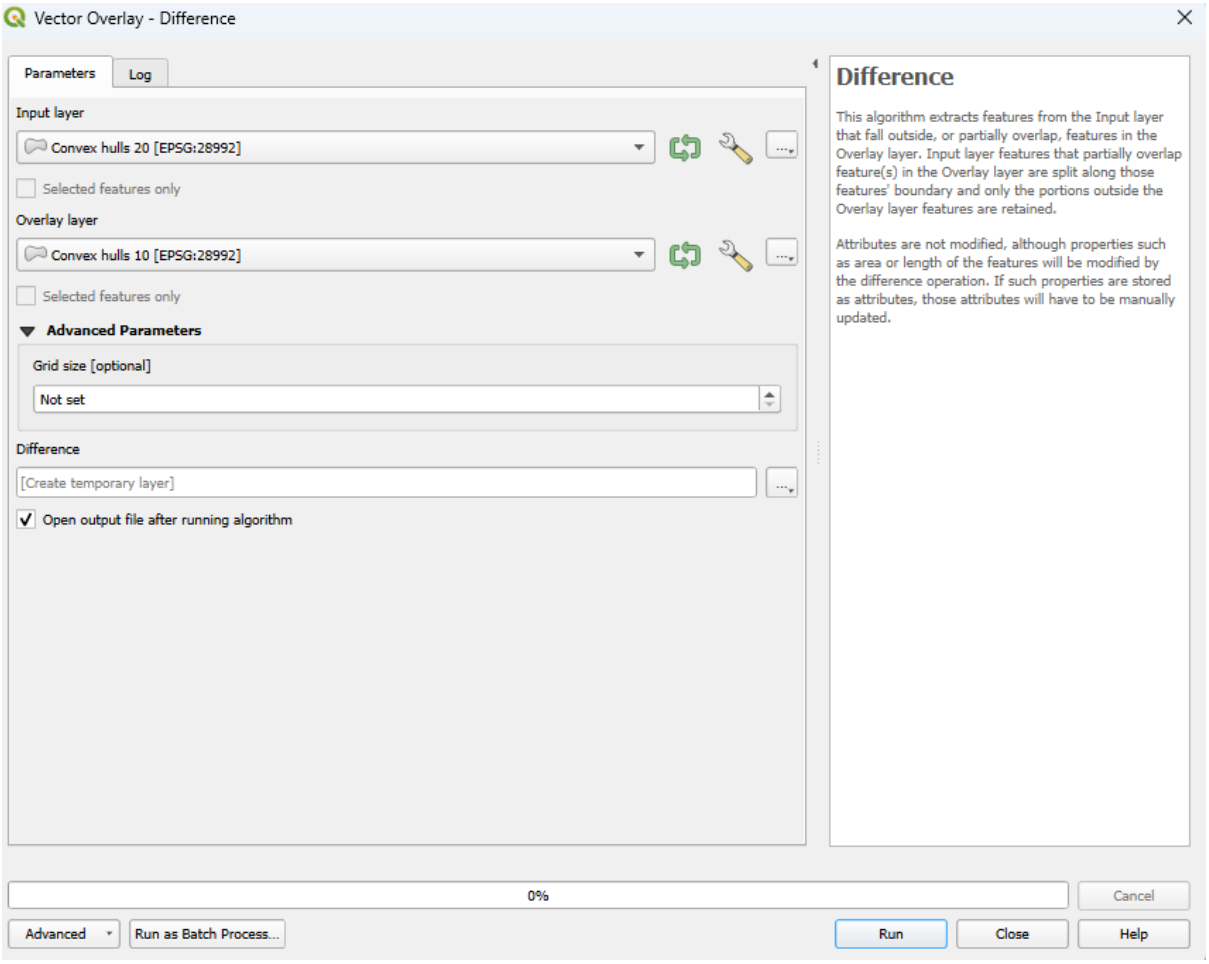
What are the three rings?
ring_0_10min: = zone_10min used directly – already done, one polygon per station
ring_10_20min: = zone_20min minus zone_10min – the area newly reachable between 10 and 20 min, per station
ring_20_30min: = zone_30min minus zone_20min – the area newly reachable between 20 and 30 min, per station

Step 1: Create ring_10_20min

57. Processing Toolbox → **Difference**
- Input layer: zone_20min.gpkg
- Overlay layer: zone_10min.gpkg

- Output: ring_10_20min.gpkg

58. Run



Step 2: Create ring_20_30min

59. Processing Toolbox → **Difference**

- Input layer: zone_30min.gpkg
- Overlay layer: zone_20min.gpkg
- Output: ring_20_30min.gpkg

60. Run

The inner ring is already available: zone_10min.gpkg. Use it directly as ring_0_10min in the next step.

Verification:
Open ring_10_20min.gpkg on the map. Overlay zone_10min.gpkg. The ring must not overlap the inner zone – every gap in ring_10_20min is exactly where zone_10min sits.

Exercise 11: Count and visualise per ring

Step 1: Count Points in Polygon per ring

61. Processing Toolbox → **Count points in polygon**

- Polygons: zone_10min.gpkg | Points: cultural.gpkg → Output: ring_0_10min_count.gpkg

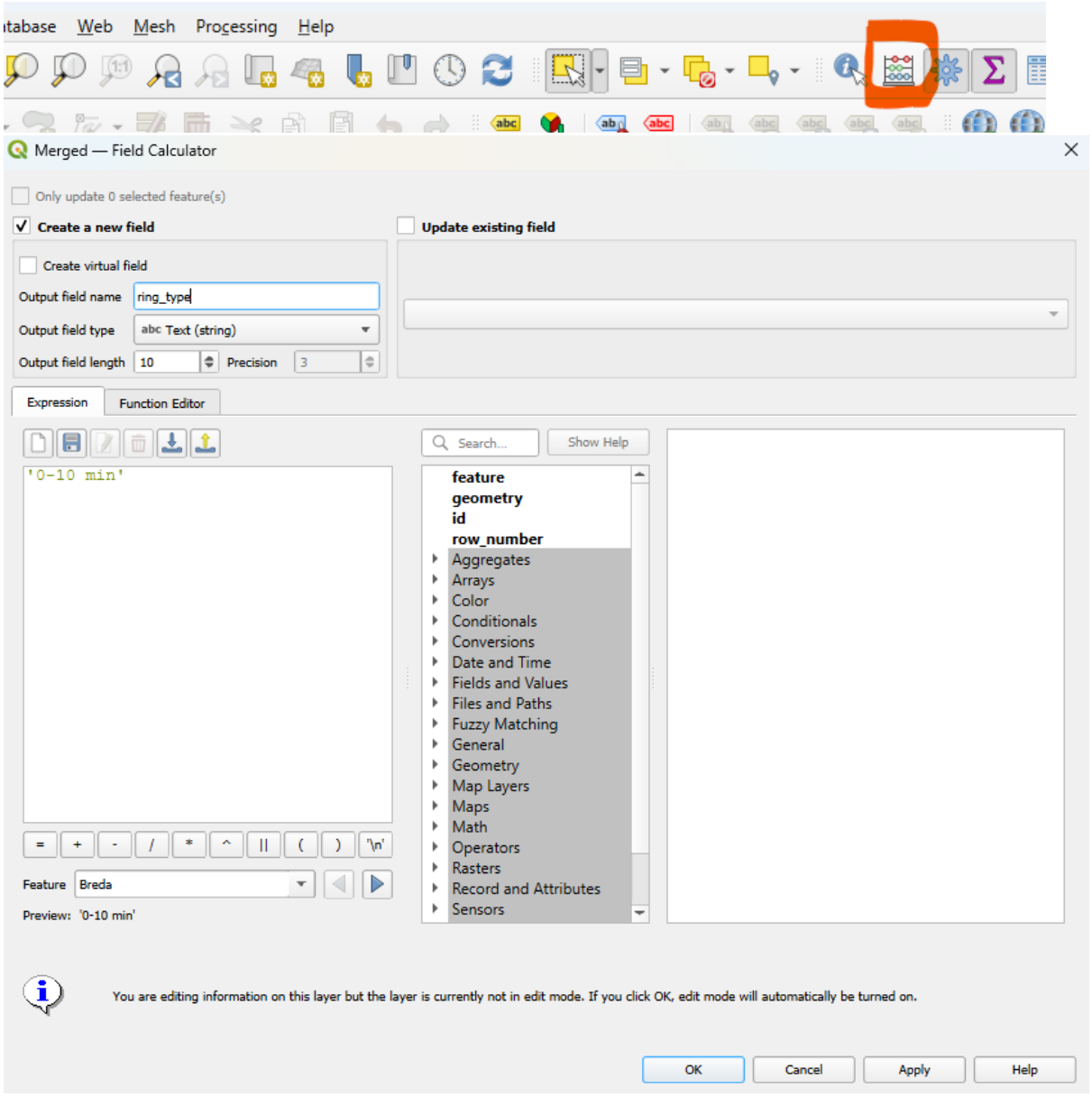
- Repeat: ring_10_20min.gpkg → ring_10_20min_count.gpkg
- Repeat: ring_20_30min.gpkg → ring_20_30min_count.gpkg

Each output layer has five rows with the field NUMPOINTS – the number of cultural locations per station per ring.

Step 2: Add a ring_type field and merge all rings into one layer

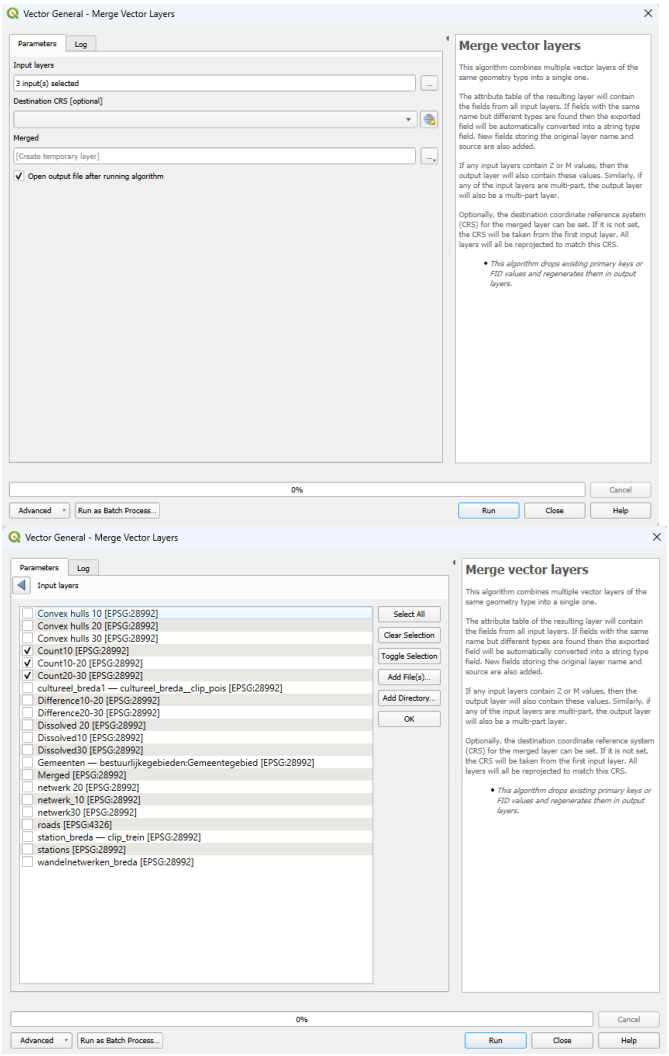
Adding a ring_type field before merging lets you identify which ring each feature belongs to after the merge.

62. Open **ring_0_10min_count.gpkg** → click the Field Calculator icon (ε) in the toolbar at the top of QGIS
    - New field: ring_type | Type: Text (string) | Expression: '0-10 min'
    - OK → Save
63. Repeat for ring_10_20min_count.gpkg → expression: '10-20 min'
64. Repeat for ring_20_30min_count.gpkg → expression: '20-30 min'



65. Processing Toolbox → **Merge vector layers**
- Input layers: ring_0_10min_count.gpkg, ring_10_20min_count.gpkg, ring_20_30min_count.gpkg
- Output: all_rings.gpkg

66. Run



Verification:
The attribute table of all_rings.gpkg has 15 rows (3 rings × 5 stations). Both NUMPOINTS and ring_type columns are present.

Step 3: Graduated classification on NUMPOINTS

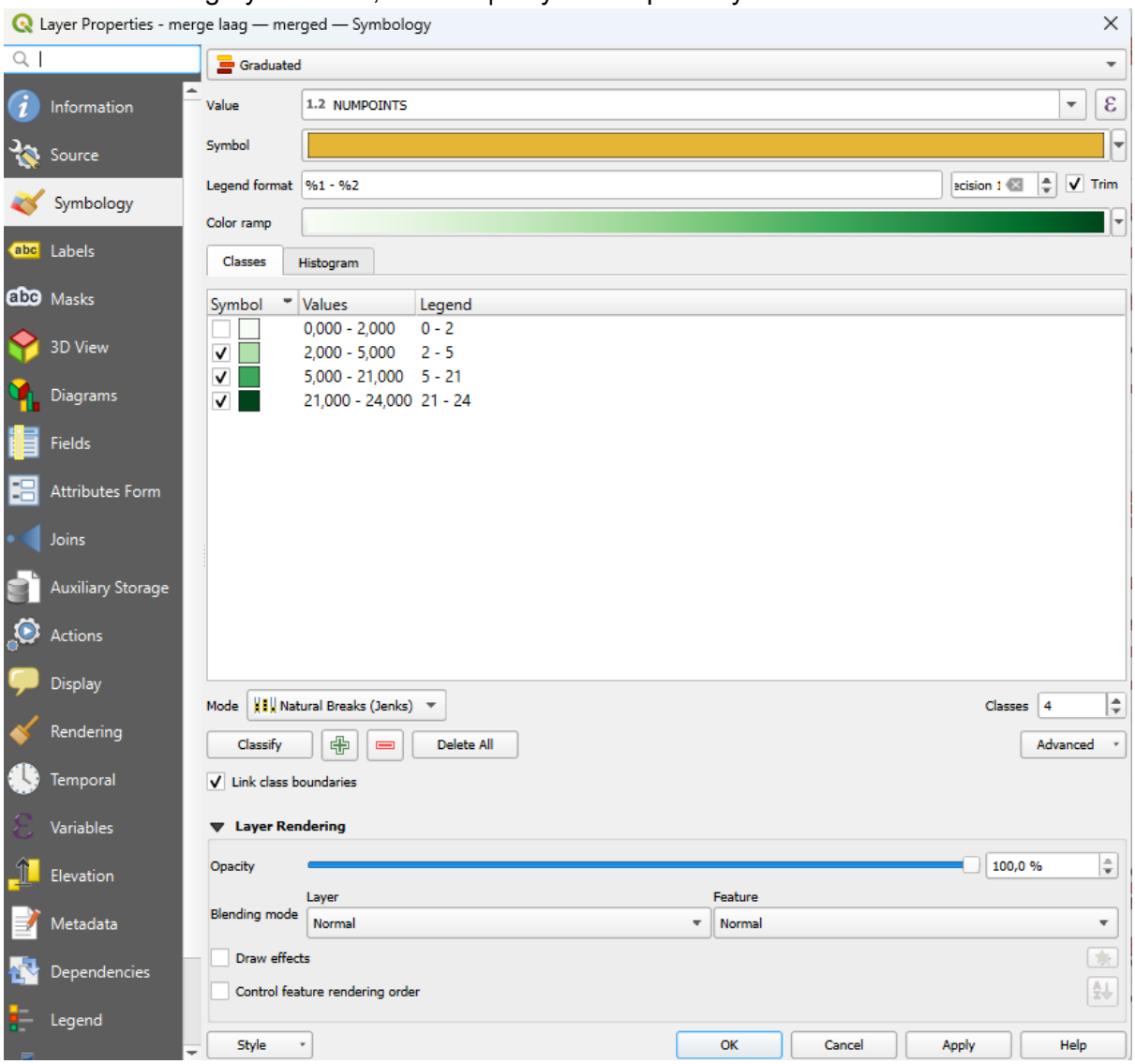
67. Right-click **all_rings.gpkg** → Properties → Symbology
68. Type: Graduated | Value: NUMPOINTS
69. Mode: Natural Breaks (Jenks) | Classes: 4 – click Classify

QGIS automatically determines the class boundaries from your data distribution. With this dataset the classes will typically come out around 0–2, 2–5, 5–21 and 21–24.

Class (Natural Breaks)	Colour	Meaning

0 – 2	Light green (lowest shade in Greens ramp)	Few or no locations
2 – 5	Mid green	Moderate
5 – 21	Dark green	Many
21 – 24	Very dark green (highest shade)	Very many

70. Colour ramp: click the ramp dropdown next to Color ramp and choose Greens
71. Stroke: dark grey #424242, 0.3mm | Layer transparency: 20%



Step 4: Labels

72. Right-click **all_rings.gpkg** → Properties → Labels → Single labels

- Value: click the epsilon button (ε) next to the Value field and enter:
  `"ring_type" || '\n' || "NUMPOINTS" || ' loc.'`
- Text size: 5pt
- Buffer: tick Draw text buffer (Buffer tab) – white outline around text so labels remain readable on dark green

- Placement: choose Offset from centroid – also tick Allow placing labels outside of polygons so small polygons still get a label

73. OK

> **Tip:**
> **Quick Field Calculator:** Click the Field Calculator icon (ε) directly in the toolbar at the top of the QGIS window – you do not need to open the attribute table separately.

Layer order: station_snapped.gpkg, cultural.gpkg, all_rings.gpkg, CartoDB Positron.

> **What do you see?**
> All 15 rings in one layer coloured by count. Light green = few, dark green = many. Each label shows the time band and the number of locations. You can now directly compare which station and which ring reveals the most heritage.

Exercise 12: Print Layout for Analysis 1

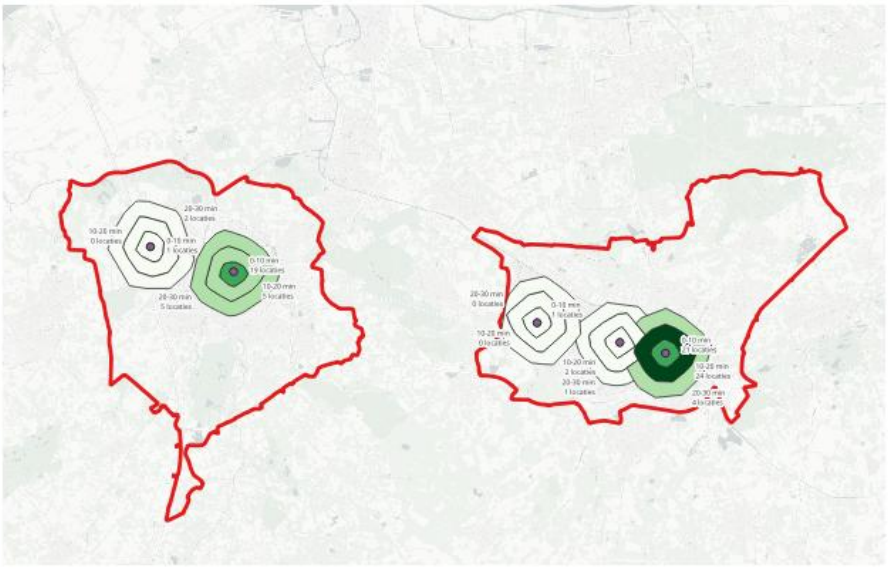
Create an overview map on A4 showing both cities at a scale where Tilburg and Breda fit together.

Step 1: Create a new layout

74. Project → New Print Layout → name it **Analysis1_Accessibility**
75. Layout → Page Properties → Size: A4 | Orientation: Landscape

Step 2: Add the map at scale 1:200,000

76. Add Item → **Add Map** → draw a rectangle filling approx. 70% of the page
77. In the Item Properties panel: **Scale** → enter **200000**
78. Click the map and use Move item content to centre Breda and Tilburg



Tip:
At 1:200,000 both cities fit on one A4 landscape page. Use the Move item content button (hand icon) to adjust the map position without changing the scale.

Step 3: Legend showing only visible items

79. Add Item → **Add Legend**

- Item Properties → untick: Auto update
- **Only show items inside linked maps** → tick this – only layers actually visible in the map frame appear in the legend
- Remove processing layers you do not want in the legend using the minus button

Step 4: Other elements

- Add Item → Add Scale Bar → unit: km
- Add Item → Add North Arrow
- Add Item → Add Label → title: **Cultural Heritage within Walking Distance of the Station – Breda & Tilburg**

Step 5: Export

80. Layout → **Export as PDF** → resolution 300 dpi → filename: **analysis1_accessibility.pdf**

Analysis 2: Walking Time to Buildings from the Station (20 min)

In Analysis 1 you calculated zones from the station outwards. Now you zoom in to one city and calculate the exact walking route from each cultural location to the station. You use Shortest path (layer to point) with the cultural location points as the source layer and the station as the single destination point.

Exercise 13: Clip cultural locations to one municipality

In Analysis 1 you analysed all stations together across both cities. In Analysis 2 you zoom in to one city and calculate the exact walking route from each cultural location to its nearest station. Start by clipping cultural.gpkg to just one municipality.

1. Processing → **Clip**
- Input layer: cultural.gpkg
- Overlay layer: municipality.gpkg – or load a new WFS filter for just Tilburg or Breda
- Output: cultural_tilburg.gpkg | CRS: EPSG:28992 (or cultural_breda.gpkg)
2. Run

Tip:
If your municipality.gpkg already covers only one city, use it directly. If it covers both, apply a filter first: right-click → Filter → "naam" = 'Tilburg' – then use the filtered layer as the overlay.

Verification:
Open the attribute table of cultural_tilburg.gpkg. Are all features located within Tilburg? The number of features should be less than in cultural.gpkg.

Exercise 14: Shortest Path – layer to point

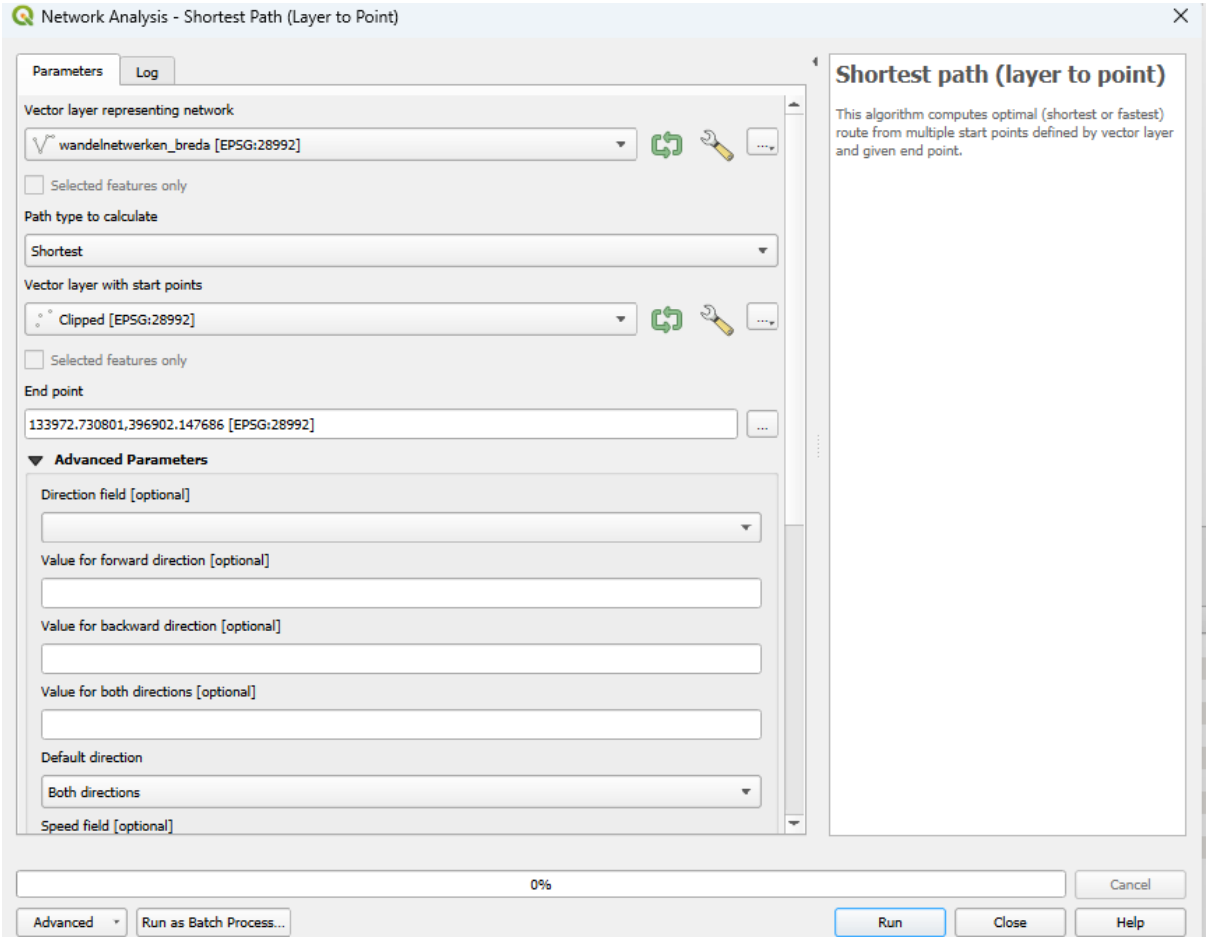
This tool calculates the shortest network path from every cultural location point to a single destination – in this case the train station. Each cultural location gets a line with a cost field showing the distance in metres along the network.

3. Processing Toolbox → **Shortest path (layer to point)**

Parameters:

- **Vector layer representing network:** walk_network.gpkg (Option A) or cycle_network.gpkg (Option B)
- Path type: Shortest
- Vector layer with start points: cultural_tilburg.gpkg
- End point: click on the Tilburg station on the map (use station_snapped.gpkg as reference)
- Output: routes_to_station.gpkg | CRS: EPSG:28992

4. Run



Verification:
Open the attribute table of routes_to_station.gpkg. Is the field cost present with values in metres? The number of rows should match the number of cultural locations in cultural_tilburg.gpkg.

Exercise 15: Calculate walking time and visualise

Step 1: Calculate walking time in minutes

5. Click the Field Calculator icon (ε) in the toolbar at the top of QGIS

Make sure routes_to_station.gpkg is the active layer. Create a new field:

- Field name: walking_time_min
- Output field type: Decimal number (double)

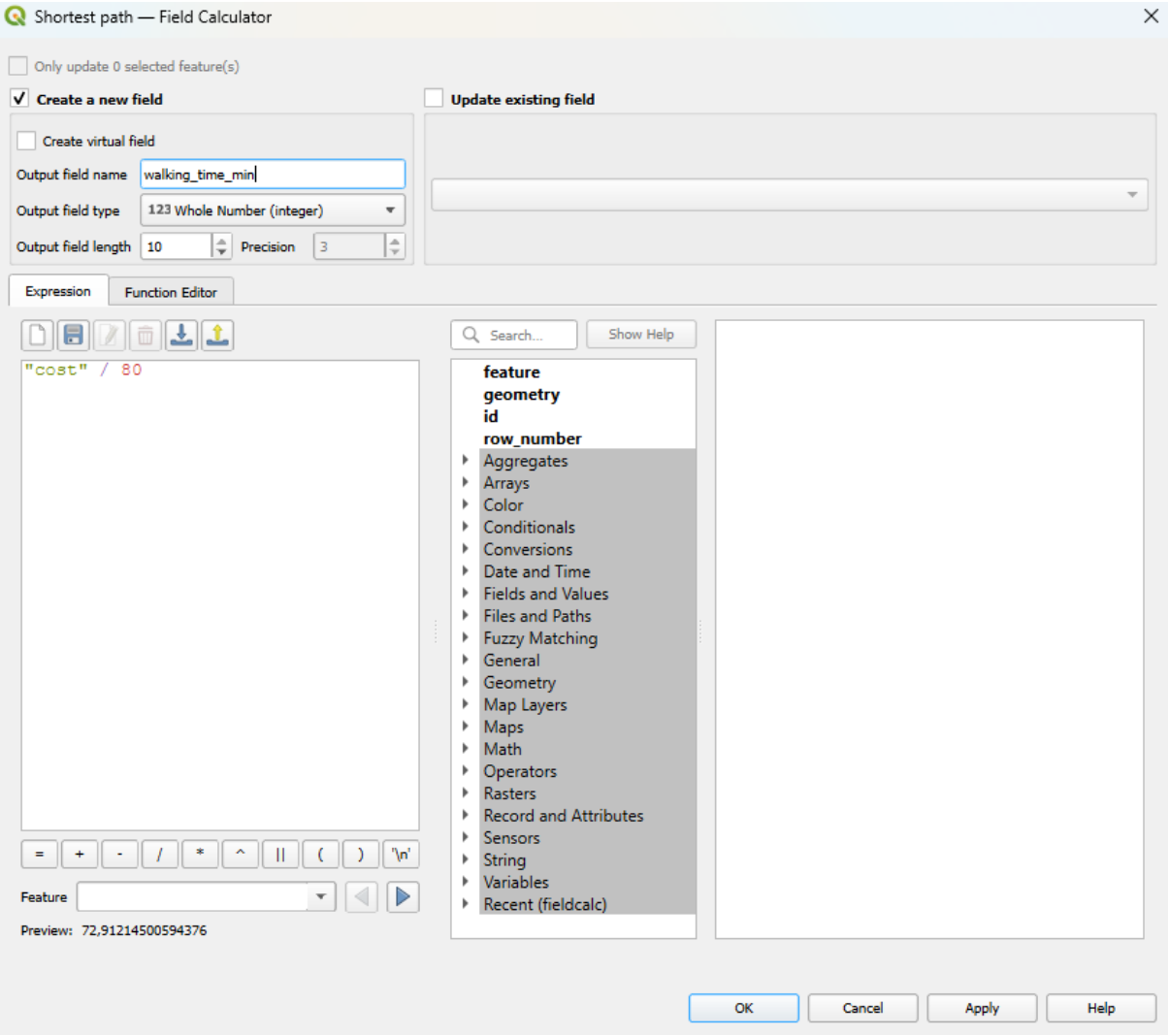
Expression:

```
"cost" / 80
```

For Option B (cycling) use:

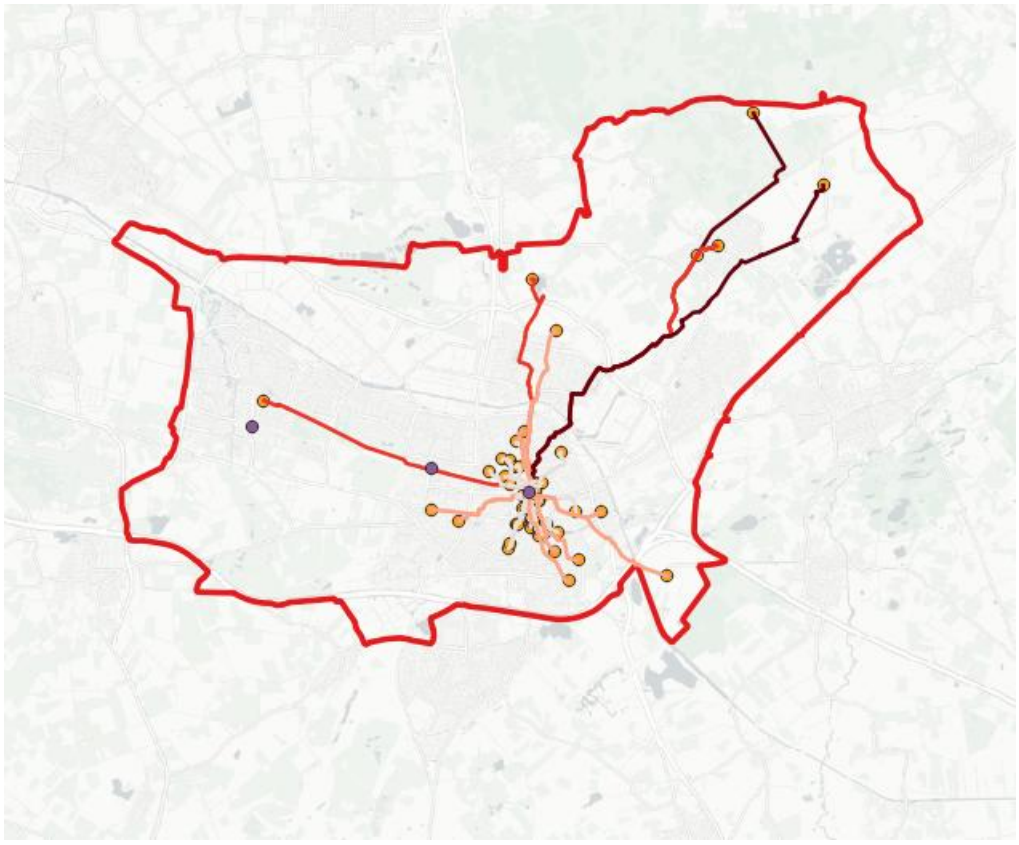
```
"cost" / 250
```

6. OK → Save edits



Step 2: Graduated classification on walking_time_min

7. Right-click **routes_to_station.gpkg** → Properties → Symbology
8. Type: Graduated | Value: walking_time_min
9. Mode: Natural Breaks (Jenks) | Classes: 4 – click Classify
10. Colour ramp: choose a green → red ramp (green = close to station, red = far)
11. Stroke: none or very thin | Layer transparency: 10%



What do you see?
Each line connects a cultural location to the station via the road network. Lines close to the station are green, lines further away turn red. You can now directly read off which cultural locations are easy to reach on foot and which require a longer walk.

Exercise 16: Print Layout for Analysis 2

Same setup as Exercise 12. Create a map focused on one city showing the walking routes coloured by travel time.

12. Project → New Print Layout → name it **Analysis2_WalkingTime**
- A4 landscape | Scale: 200,000
- Add Map → centre on Tilburg (or Breda)
- Add Legend → tick Only show items inside linked maps
- Add Scale Bar (km), North Arrow
- Title: Walking Time from Cultural Locations to Station – Tilburg

13. Export as PDF: **analysis2_walking_time.pdf** – 300 dpi

Troubleshooting

Problem: Service Area returns empty output

- Check that walk_network.gpkg is saved as a permanent GeoPackage, not a temporary filtered layer
- Use station_snapped.gpkg – these points are guaranteed to be on the network

Problem: Count Points in Polygon returns 0 everywhere

- Check that cultural.gpkg and the zone polygons have the same CRS (both EPSG:28992)
- Zoom in and visually check whether the points actually lie inside the polygon

Problem: fclass values are missing from my data

- Processing → List Unique Values on the fclass field to see which values are present
- Adjust the filter expression in Exercise 6 to match the values actually in your data

Problem: Shortest Path (layer to point) is slow or returns no routes

- Check that the end point (station) is close to the road network. Use station_snapped.gpkg to click the correct point.
- If cultural_tilburg.gpkg has many features, this tool may take up to a minute. This is normal.

Reflection Questions

Answer briefly:

- How many cultural locations are accessible within 10, 20 and 30 minutes? What strikes you about the distribution?
- Are locations clustered near the station or spread across the city?
- If you did Option B: how much more heritage is reachable by bicycle compared to walking? What does that say about the role of the OV-bike for cultural tourism?
- What does the walking time map from Analysis 2 add to your understanding compared to the zone map from Analysis 1?
- What are the limitations of this analysis? Think about: OSM road quality, missing locations, straight-line vs. network distance.

What You Have Learned

Technical skills

- Identifying and loading the correct Geofabrik files (roads, transport, pois)
- Setting project CRS to EPSG:28992 and applying it to every export
- Loading a CartoDB Positron basemap via QuickMapServices
- Filtering a road network for walking or cycling
- Snapping station points to the network and correcting manually
- Service Area (from layer) for multiple start points simultaneously
- Convex Hull to convert lines to polygons per station
- Difference to create exclusive rings
- Count Points in Polygon per ring per station
- Merging layers and Graduated classification with Natural Breaks
- Clipping a point layer to a single municipality for focused analysis
- Shortest Path (layer to point) from a set of locations to a single destination
- Calculating walking time via Field Calculator: "cost" / 80
- Print Layout at 1:200,000 with Only show items inside linked maps

Conceptual understanding

- How a network isochrone works and why it is more accurate than a circular buffer
- Why Convex Hull is needed after Service Area
- The difference between cumulative zones and exclusive rings
- Converting metres to minutes using walking or cycling speed
- Why EPSG:28992 is essential for network analysis in the Netherlands