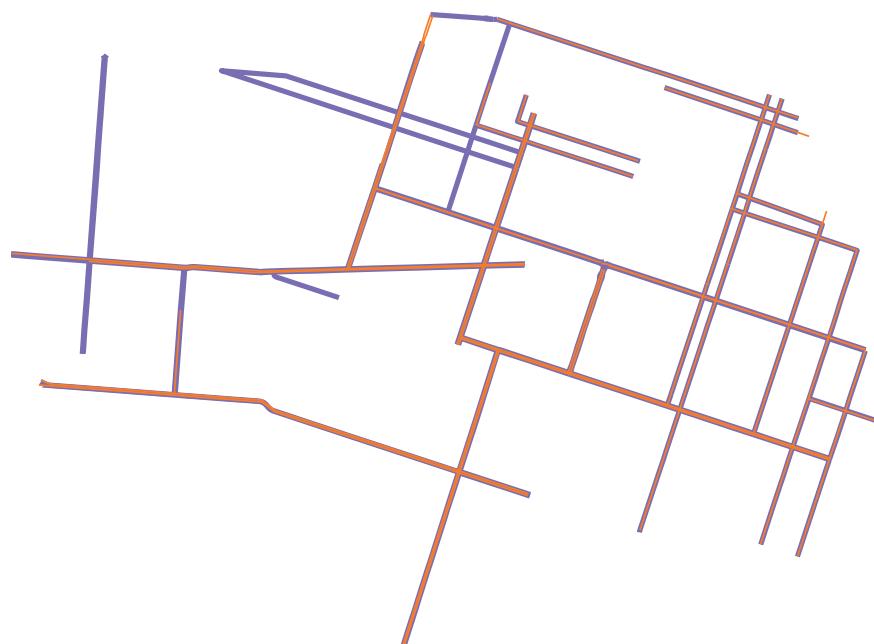


# BikeDNA Report

## Rosario



Bicycle Infrastructure  
Data & Network Assessment





# About BikeDNA

This report was automatically generated by BikeDNA (Bicycle Data & Network Assessment) by converting Jupyter notebooks into pdf. BikeDNA is a tool for assessing the quality of [OpenStreetMap \(OSM\)](#) and other bicycle infrastructure data sets in a reproducible way. It provides planners, researchers, data maintainers, cycling advocates, and others who work with bicycle networks a detailed, informed overview of data quality in a given area.

BikeDNA is maintained at <https://github.com/anerv/BikeDNA> by Ane Rahbek Vierø, Anastassia Vybornova, and Michael Szell. It is available under the [AGPL 3.0 license](#).



A fair amount of research projects on OpenStreetMap and other forms of volunteered geographic information (VGI) have already been conducted, but few focus explicitly on bicycle infrastructure. Doing so is important because paths and tracks for cyclists and pedestrians often are mapped last and are more likely to have errors ([Barron et al., 2014](#), [Neis et al. 2012](#)). Moreover, the spatial distribution of dips in data quality in crowdsourced data are often not random but correlate with population density and other characteristics of the mapped area ([Forghani and Delavar, 2014](#)), which requires a critical stance towards the data we use for our research and planning, despite the overall high quality of OSM.

*Data quality* covers a wide range of aspects. The conceptualization of data quality used here refers to *fitness-for-purpose* ([Barron et al., 2014](#)) - this means that data quality is interpreted as whether or not the data fulfils the user needs, rather than any universal definition of quality. To particularly support network-based research and planning, BikeDNA provides insights into the topological structure of the bicycle network apart from data coverage.

The purpose is not to give any final assessment of the data quality, but to highlight aspects that might be relevant for assessing whether the data for a given area is fit for use. While BikeDNA can make use of a reference dataset to compare with OSM, if one is available, BikeDNA cannot give any final assessment of the quality of a refernece data compared to OSM. However, OSM data on bicycle infrastructure is often at a comparable or higher quality than governmental datasets, and the interpretation of differences between the two requires adequate local knowledge.

# 1a. Initialize OSM data

This notebook:

- Loads the polygon defining the study area and then creates a grid overlay for the study area.
- Downloads street network data for the study area using OSMnx.
- Creates a network only with bicycle infrastructure (with queries defined in `config.yml` ).
- Creates additional attributes in the data to be used in the analysis.

## Sections

- [Load data for study area and create analysis grid](#)
- [Download and preprocess OSM data](#)

## Load data for study area and create analysis grid

This step:

- Loads settings for the analysis from the configuration file `config.yml` .
- Reads data for the study area.
- Creates a grid overlay of the study area, with grid cell size as defined in `config.yml` .

### Load data for study area

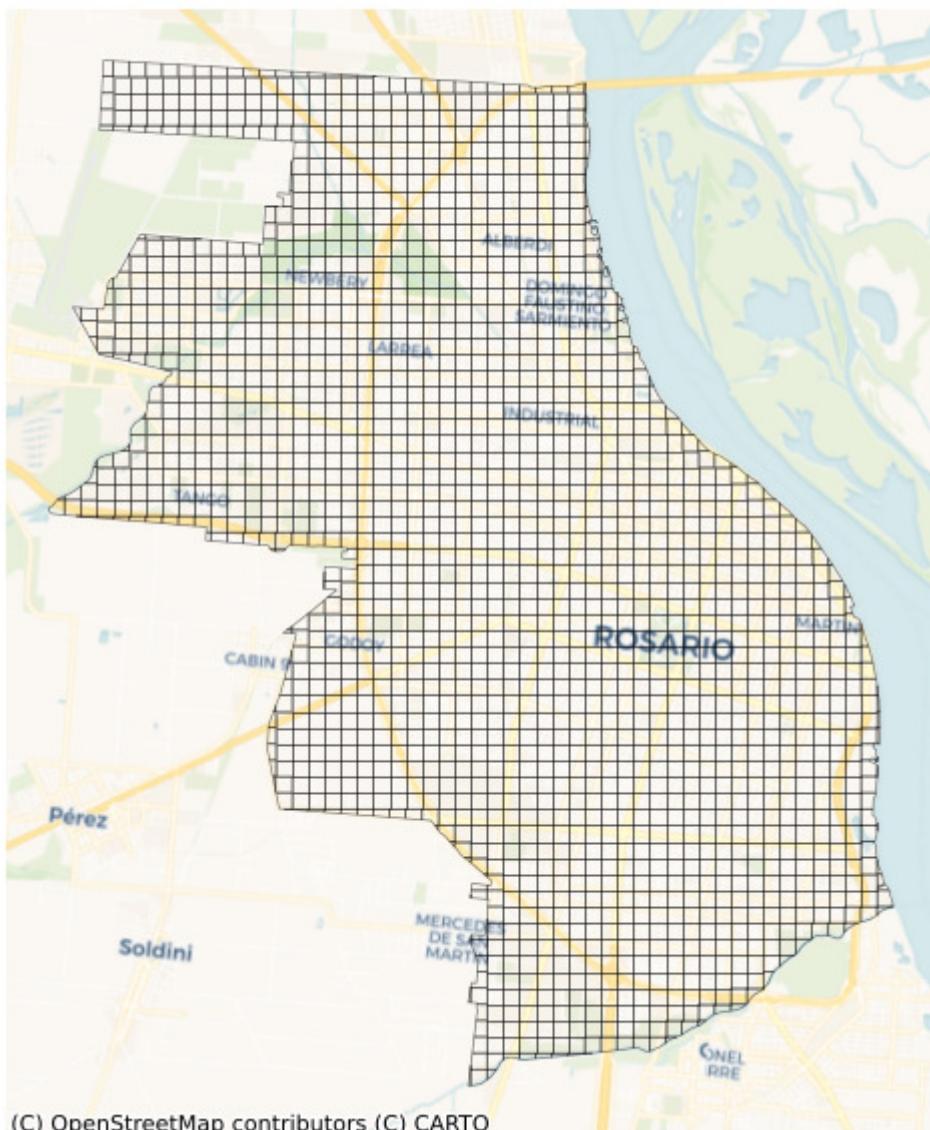
The study area is defined by the user-provided polygon. It will be used for the computation of **global** results, i.e. quality metrics based on all data in the study area.

The size of the study area is 159.16 km<sup>2</sup>.

### Create analysis grid

The grid contains 1930 square cells with a side length of 300 m and an area of 0.09 km<sup>2</sup>. This grid will be used for local (grid cell level) analysis:

## Rosario study area (1930 grid cells, side length 300m)



## Download and preprocess OSM data

This step:

- Downloads data from OpenStreetMap using OSMnx.
- Projects the data to the chosen CRS.
- Creates a subnetwork consisting only of bicycle infrastructure.
- Classifies all edges in the bicycle network based on whether they are protected or unprotected bicycle infrastructure, how they have been digitized, and whether they allow for bidirectional travel or not.
- Simplifies the network.
- Creates copies of all edge and node data sets indexed by their intersecting grid cell.

### OSM data model

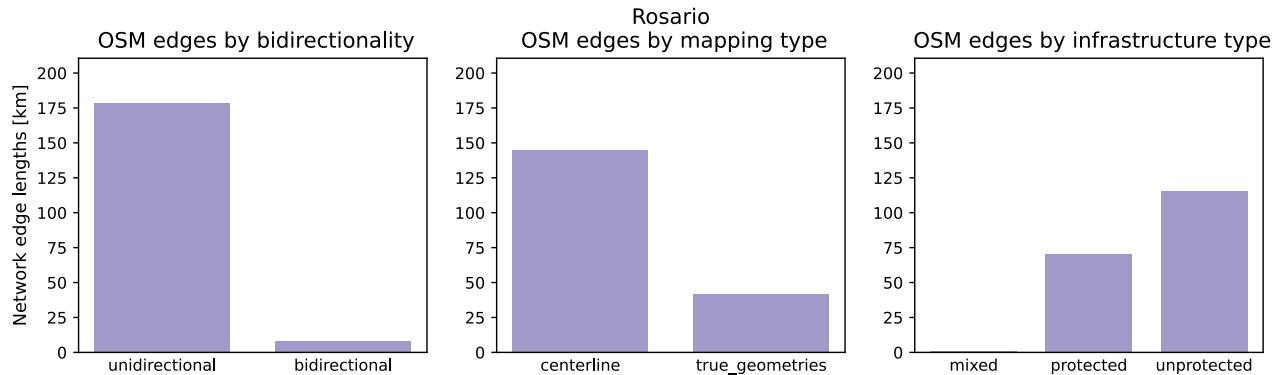
In OSM, street network data are stored using *nodes* (points) and *ways* (lines). In BikeDNA, OSM data are converted to a network structure consisting of *nodes* and *edges* (we use the terminology used in OSMnx). Edges represents the actual infrastructure, such as bike lanes and paths, while nodes represents the start and end points for the edges, as well as all intersections. For further details, read more about the [OSM data model](#) and the [network data model](#).

No update necessary. Returning original bicycle graph.

Edges where 'bicycle\_bidirectional' is False: 3310 out of 3534 (93.66%)  
 Edges where 'bicycle\_bidirectional' is True: 224 out of 3534 (6.34%)

Edges where the geometry type is 'centerline': 2425 out of 3534 (68.62%)  
 Edges where the geometry type is 'true\_geometries': 1109 out of 3534 (31.38%)

Edges where the protection level is 'unprotected': 2029 out of 3534 (57.41%)  
 Edges where the protection level is 'protected': 1425 out of 3534 (40.32%)  
 Edges where the protection level is 'mixed': 80 out of 3534 (2.26%)



The network covers an area of 122.47 km<sup>2</sup>.

The length of the OSM network with bicycle infrastructure is 175.49 km.



# 1b. Intrinsic Analysis of OSM Data

This notebook analyzes the quality of OSM bicycle infrastructure data for a given area. The quality assessment is *intrinsic*, i.e. based only on the one input data set without making use of external information. For an extrinsic quality assessment that compares the OSM data to a user-provided reference data set, see the notebooks 3a and 3b.

The analysis assesses the *fitness for purpose* (Barron et al., 2014) of OSM data for a given area. Outcomes of the analysis can be relevant for bicycle planning and research - especially for projects that include a network analysis of bicycle infrastructure, in which case the topology of the geometries is of particular importance.

Since the assessment does not make use of an external reference data set as the ground truth, no universal claims of data quality can be made. The idea is rather to enable those working with OSM-based bicycle networks to assess whether the data are good enough for their particular use case. The analysis assists in finding potential data quality issues but leaves the final interpretation of the results to the user.

The notebook makes use of quality metrics from a range of previous projects investigating OSM/VGI data quality, such as Ferster et al. (2020), Hochmair et al. (2015), Barron et al. (2014), and Neis et al. (2012).

## Familiarity required

For a correct interpretation of some of the metrics for spatial data quality, some familiarity with the area is necessary.

## Sections

- Data completeness
  - Network density
- OSM tag analysis
  - Missing tags
  - Incompatible tags
  - Tagging patterns
- Network topology
  - Simplification outcome
  - Dangling nodes
  - Under/overshoots
  - Missing intersection nodes
- Network components
  - Disconnected components
  - Components per grid cell
  - Component length distribution
  - Largest connected component
  - Missing links

- Component connectivity
- Summary

## Data completeness

### Network density

In this setting, network density refers to the length of edges or number of nodes per km<sup>2</sup>. This is the usual definition of network density in spatial (road) networks, which is distinct from the *structural* network density known more generally in network science. Without comparing to a reference data set, network density does not in itself indicate spatial data quality. For anyone familiar with the study area, network density can however indicate whether parts of the area appear to be under- or over-mapped.

### Method

The density here is not based on the geometric length of edges, but instead on the computed length of the infrastructure. For example, a 100-meter-long bidirectional path contributes with 200 meters of bicycle infrastructure. This method is used to take into account different ways of mapping bicycle infrastructure, which otherwise can introduce large deviations in network density. With `compute_network_density`, the number of elements (nodes, dangling nodes, and total infrastructure length) per unit area is calculated. The density is computed twice: first for the study area for both the entire network ('global density'), then for each of the grid cells ('local density'). Both global and local densities are computed for the entire network and for protected and unprotected infrastructure.

### Interpretation

Since the analysis conducted here is intrinsic, i.e. it makes no use of external information, it cannot be known whether a low-density value is due to incomplete mapping, or due to actual lack of infrastructure in the area. However, a comparison of the grid cell density values can provide some insights, for example:

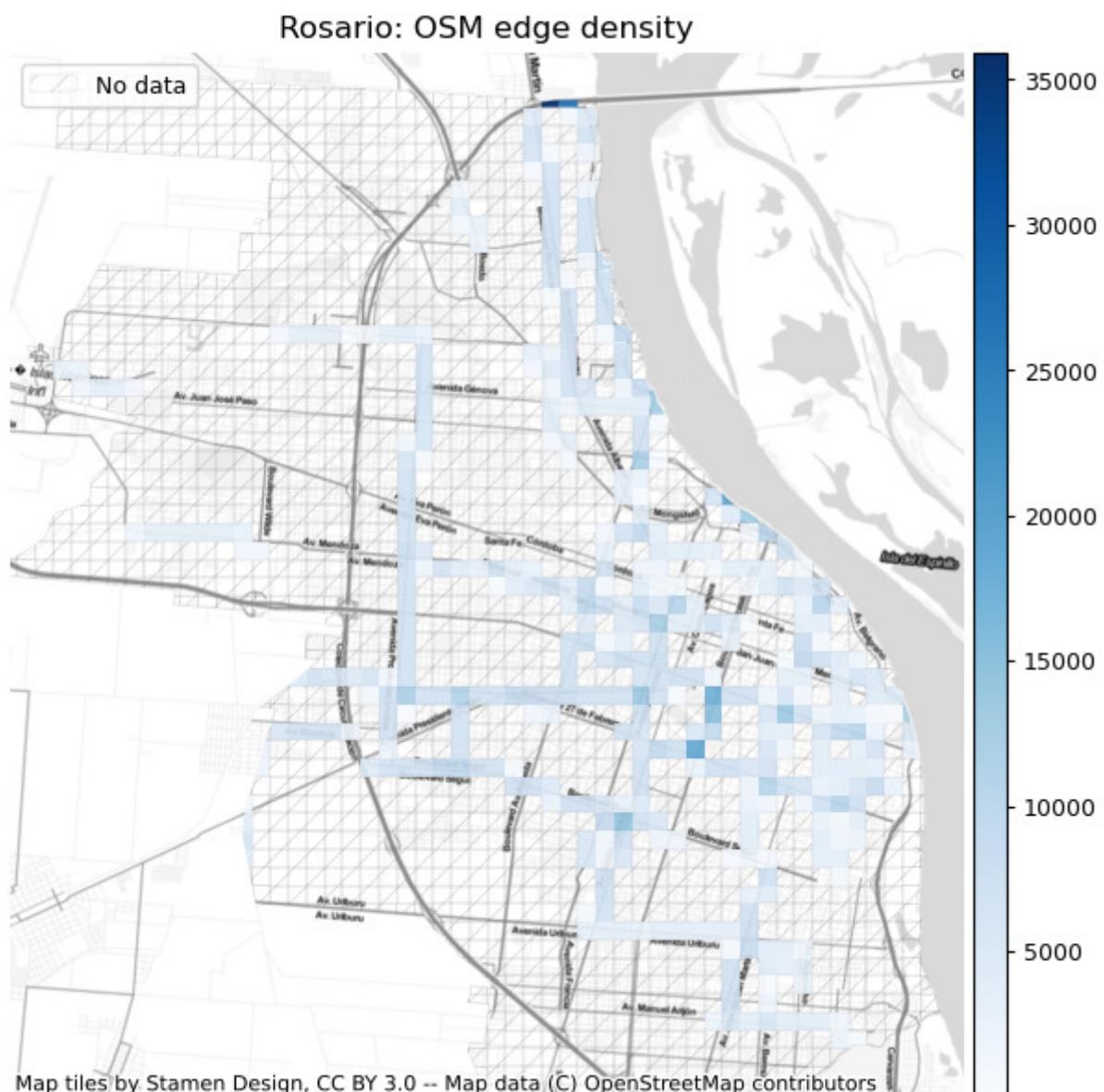
- lower-than-average infrastructure density indicates a locally sparser network
- higher-than-average node density indicates that there are relatively many intersections in a grid cell
- higher-than-average dangling node density indicates that there are relatively many dead ends in a grid cell

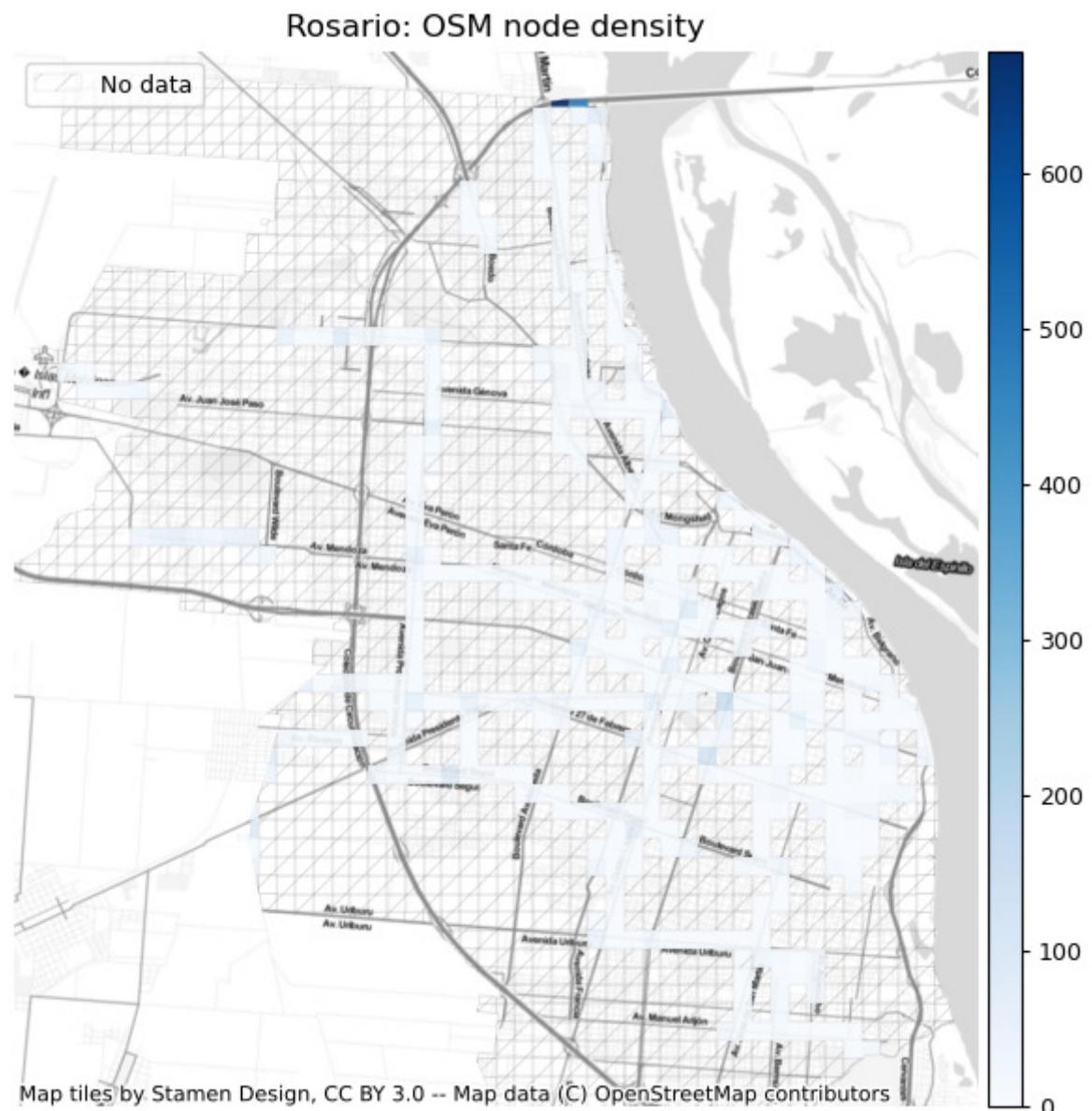
### Global network density

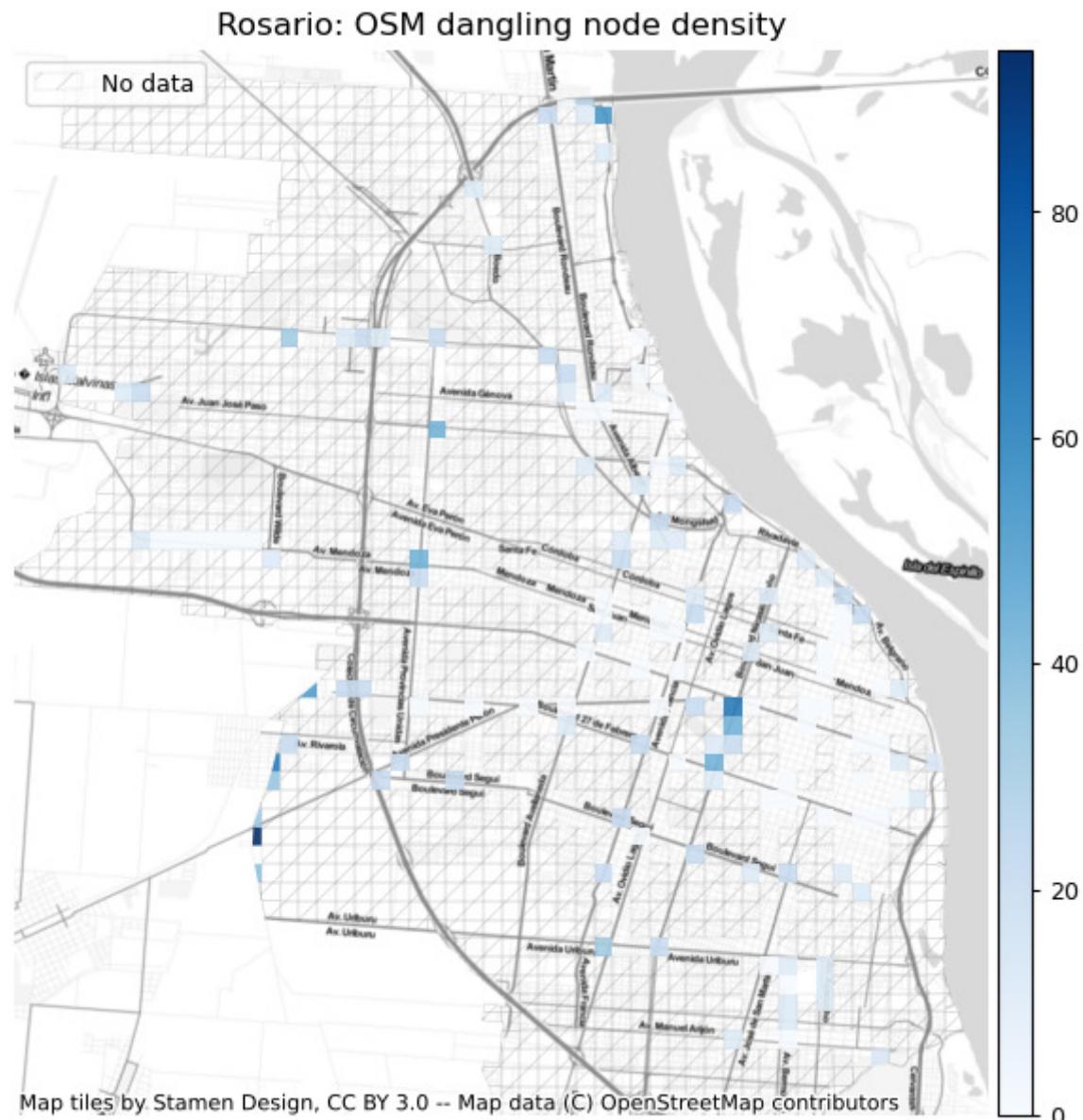
For the entire study area, there are:

- 1102.58 meters of bicycle infrastructure per km<sup>2</sup>.
- 2.15 nodes in the bicycle network per km<sup>2</sup>.
- 0.90 dangling nodes in the bicycle network per km<sup>2</sup>.
- 373.38 meters of protected bicycle infrastructure per km<sup>2</sup>.
- 726.17 meters of unprotected bicycle infrastructure per km<sup>2</sup>.
- 3.04 meters of mixed protection bicycle infrastructure per km<sup>2</sup>.

### Local network density



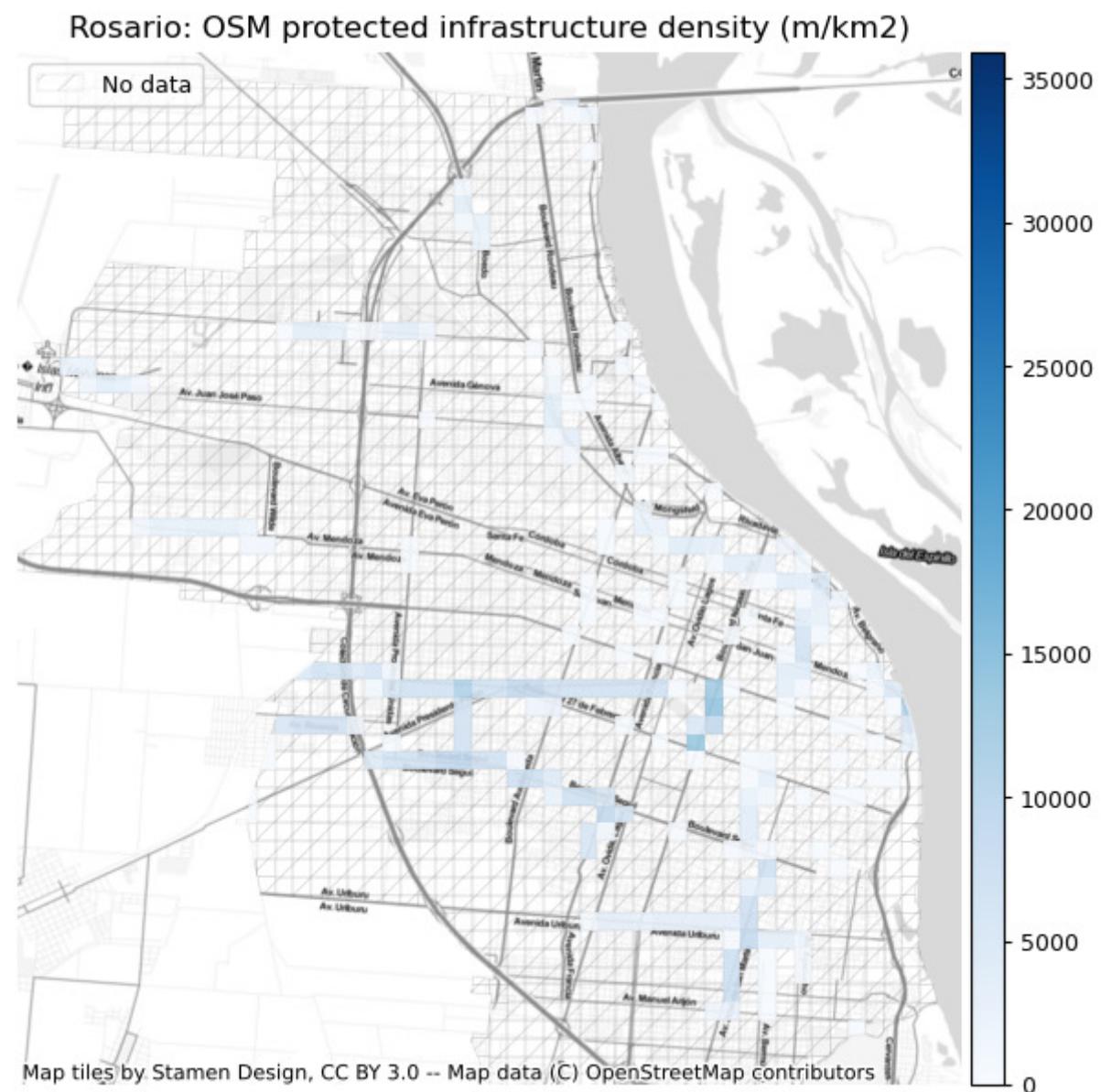


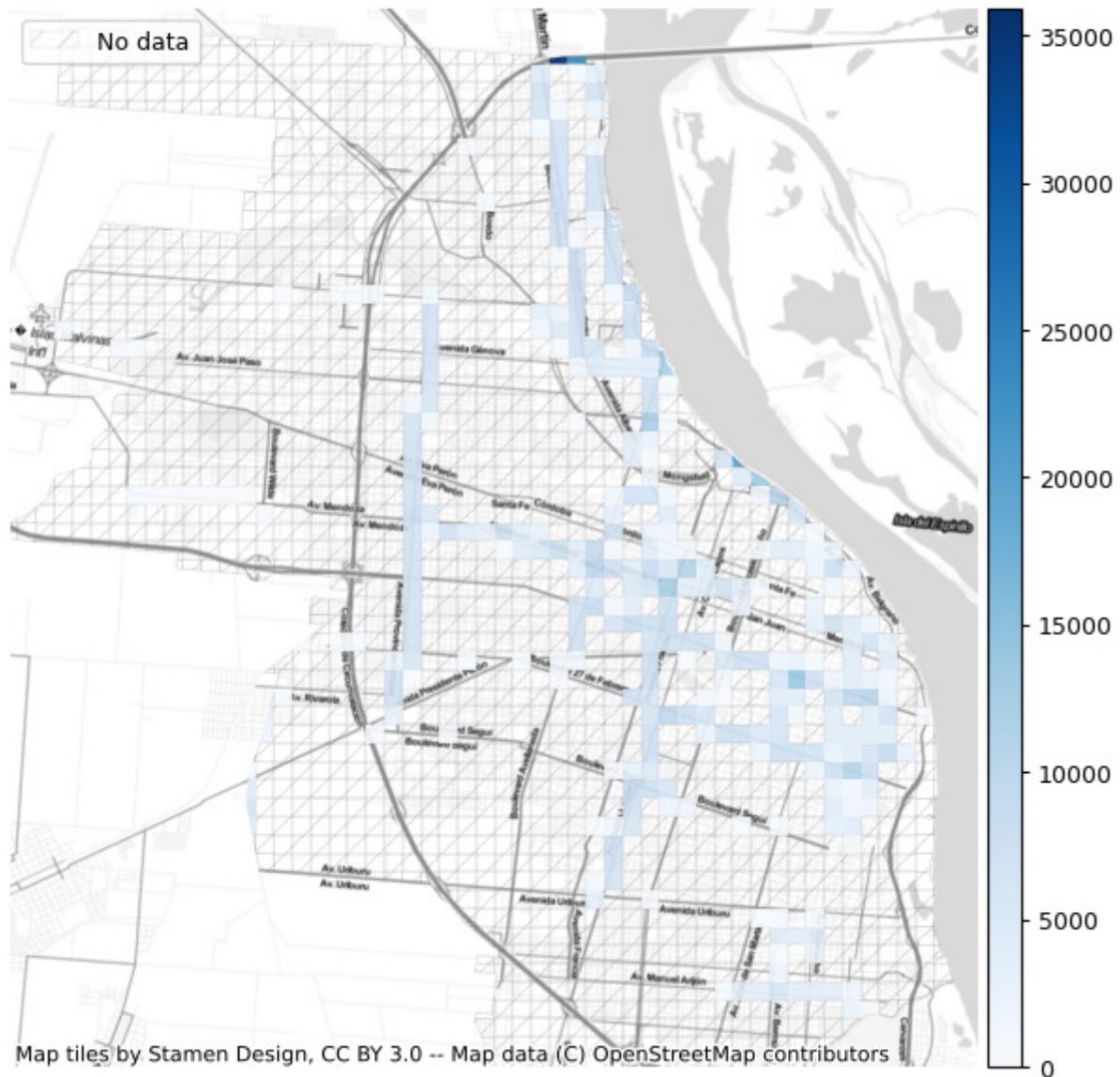


#### Densities of protected and unprotected infrastructure

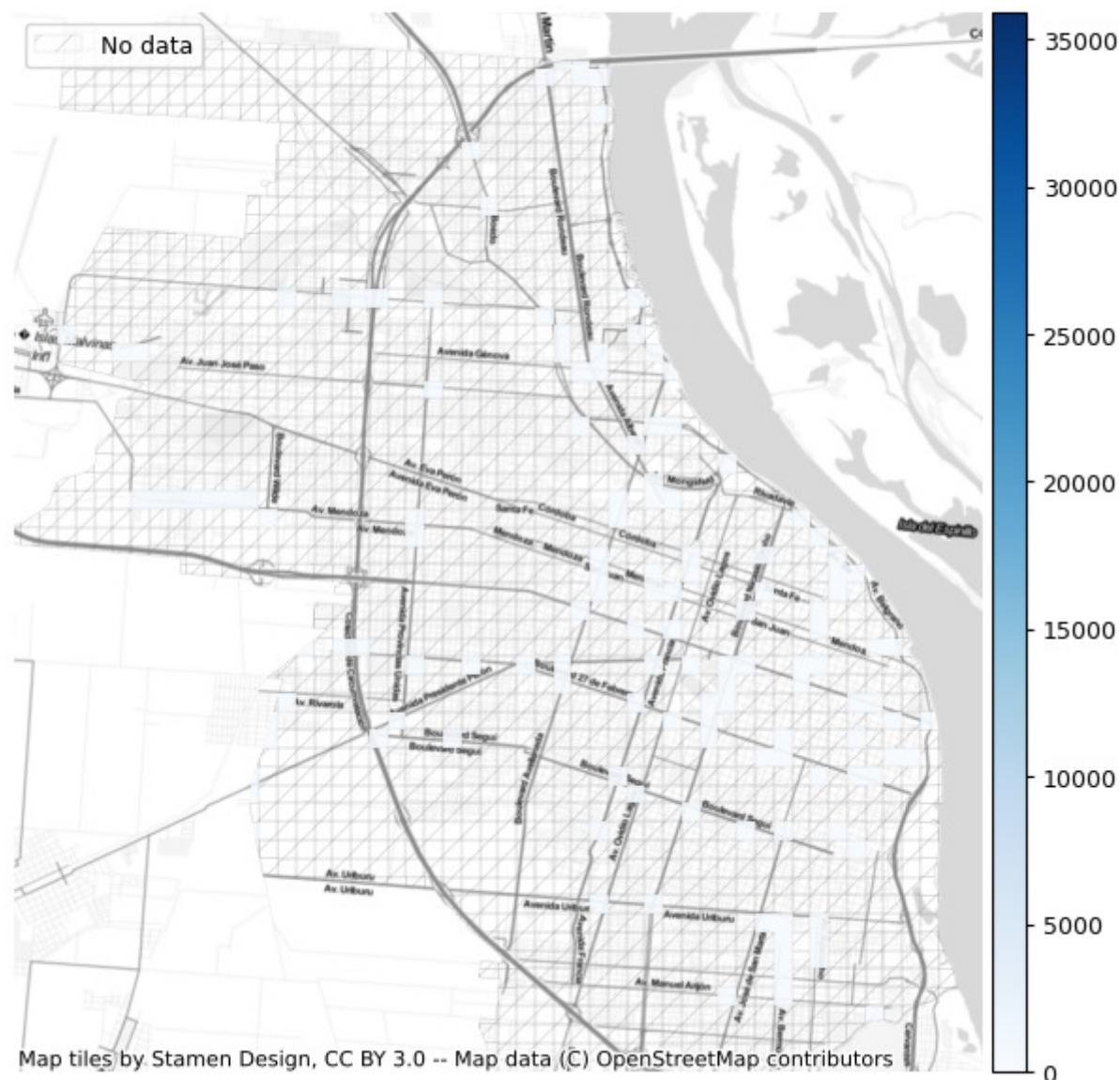
In BikeDNA, *protected infrastructure* refers to all bicycle infrastructure which is either separated from car traffic by for example an elevated curb, bollards, or other physical barriers, or for cycle tracks that are not adjacent to a street.

*Unprotected infrastructure* are all other types of lanes that are dedicated for bicyclists, but which only are separated by car traffic by e.g., a painted line on the street.



**Rosario: OSM unprotected infrastructure density for (m/km<sup>2</sup>)**

## Rosario: OSM mixed protection infrastructure density (m/km<sup>2</sup>)



## OSM tag analysis

For many practical and research purposes, more information than just the presence/absence of bicycle infrastructure is of interest. Information about e.g. the width of the infrastructure, speed limits, streetlights, etc. can be of high relevance, for example when evaluating the bike friendliness of an area or an individual network segment. The presence of these tags describing attributes of the bicycle infrastructure is however highly unevenly distributed in OSM, which poses a barrier to evaluations of bikeability and traffic stress. Likewise, the lack of restrictions on how OSM features can be tagged sometimes result in conflicting tags which can undermine the evaluation of cycling conditions.

This section includes analyzes of missing tags (edges with tags that lack information), incompatible tags (edges with tags labelled with two or more contradictory tags), and tagging patterns (the spatial variation of which tags are being used to describe bicycle infrastructure).

For the evaluation of tags, the non-simplified edges should be used to avoid issues with tags that have been aggregated in the simplification process.

## Missing tags

The information that is required or desirable to obtain from the OSM tags depends on the use case - for example, the tag `lit` for a project that studies light conditions on cycle paths. The workflow below allows to quickly analyze the percentage of network edges that have a value available for the tag of interest.

### Method

We analyze all tags of interest as defined in the `existing_tag_analysis` section of `config.yml`. For each of these tags, `analyze_existing_tags` is used to compute the total number and the percentage of edges that have a corresponding tag value.

### Interpretation

On the study area level, a higher percentage of existing tag values indicates in principle a higher quality of the data set. However, this is different from an estimation of whether the existing tag values are truthful. On the grid cell level, lower-than-average percentages for existing tag values can indicate a more poorly mapped area. However, the percentages are less informative for grid cells with a low number of edges: for example, if a cell contains one single edge that has a tag value for `lit`, the percentage of existing tag values is 100% - but given that there is only 1 data point, this is less informative than, say, a value of 80% for a cell that contains 200 edges.

### Global missing tags

```
Analysing tags describing:  
surface - width - speedlimit - lit -
```

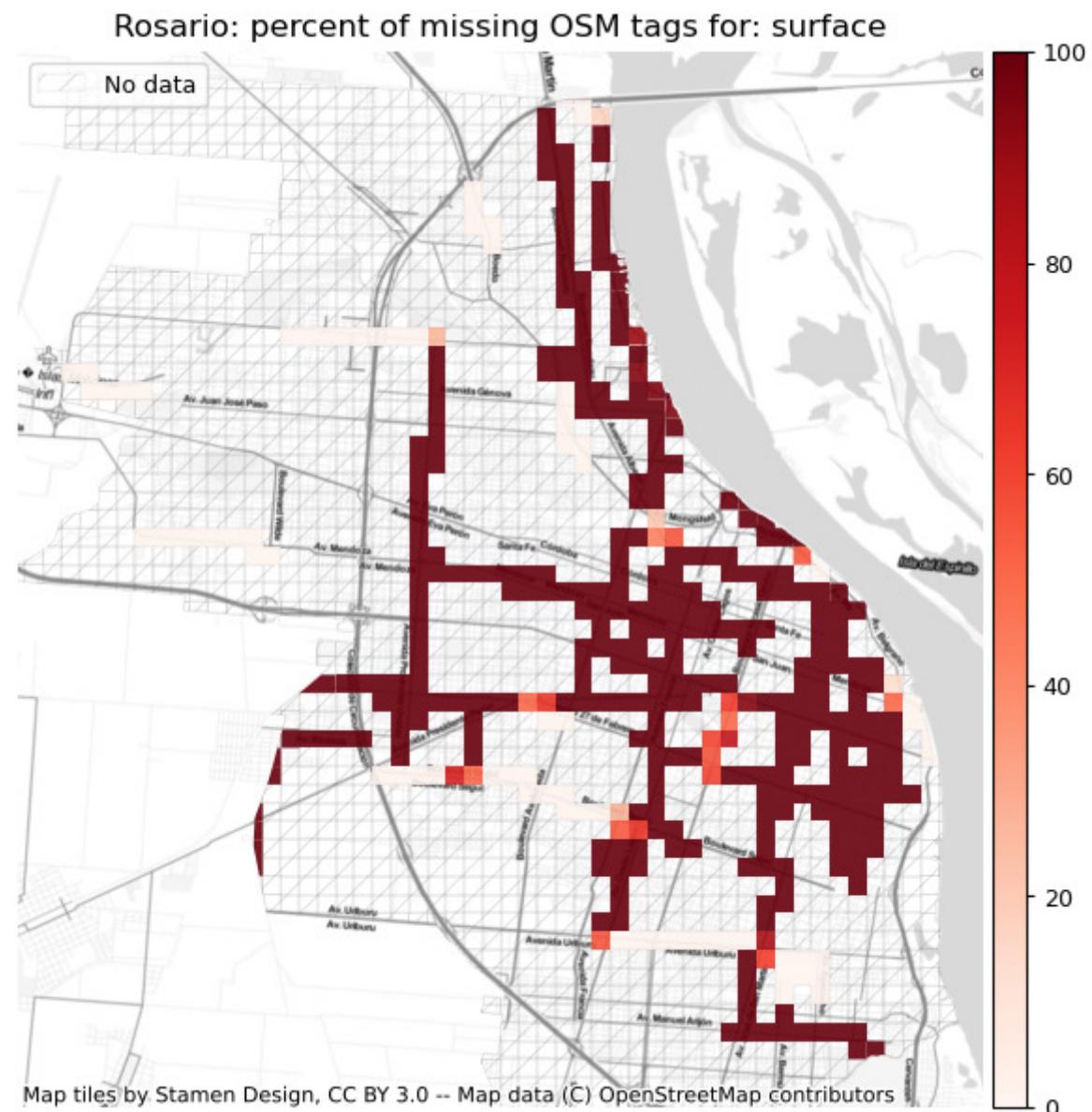
```
surface: 1109 out of 3534 edges (31.38%) have information.  
surface: 42 out of 187 km (22.47%) have information.
```

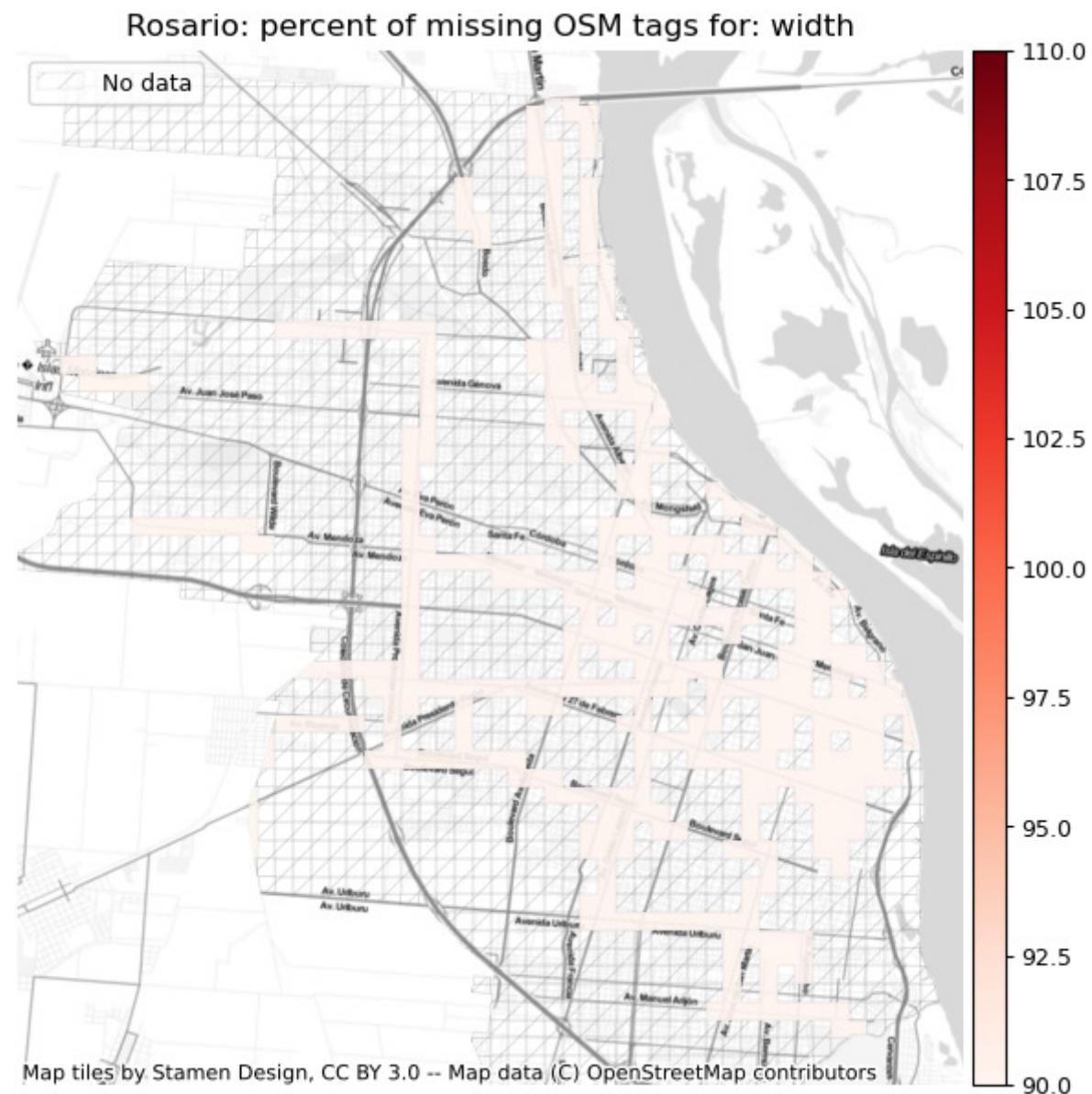
```
width: 0 out of 3534 edges (0.00%) have information.  
width: 0 out of 187 km (0.00%) have information.
```

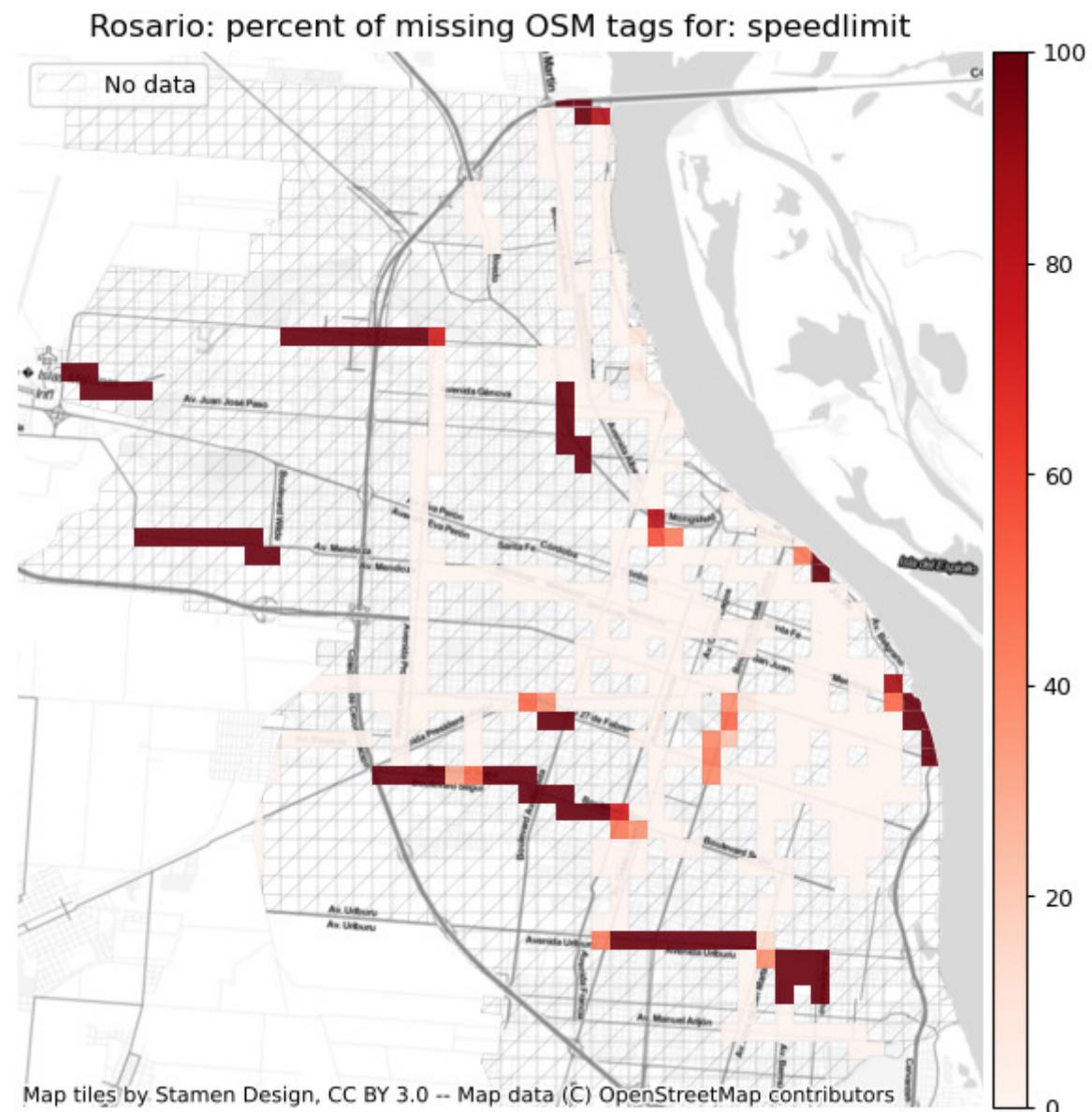
```
speedlimit: 2461 out of 3534 edges (69.64%) have information.  
speedlimit: 147 out of 187 km (78.48%) have information.
```

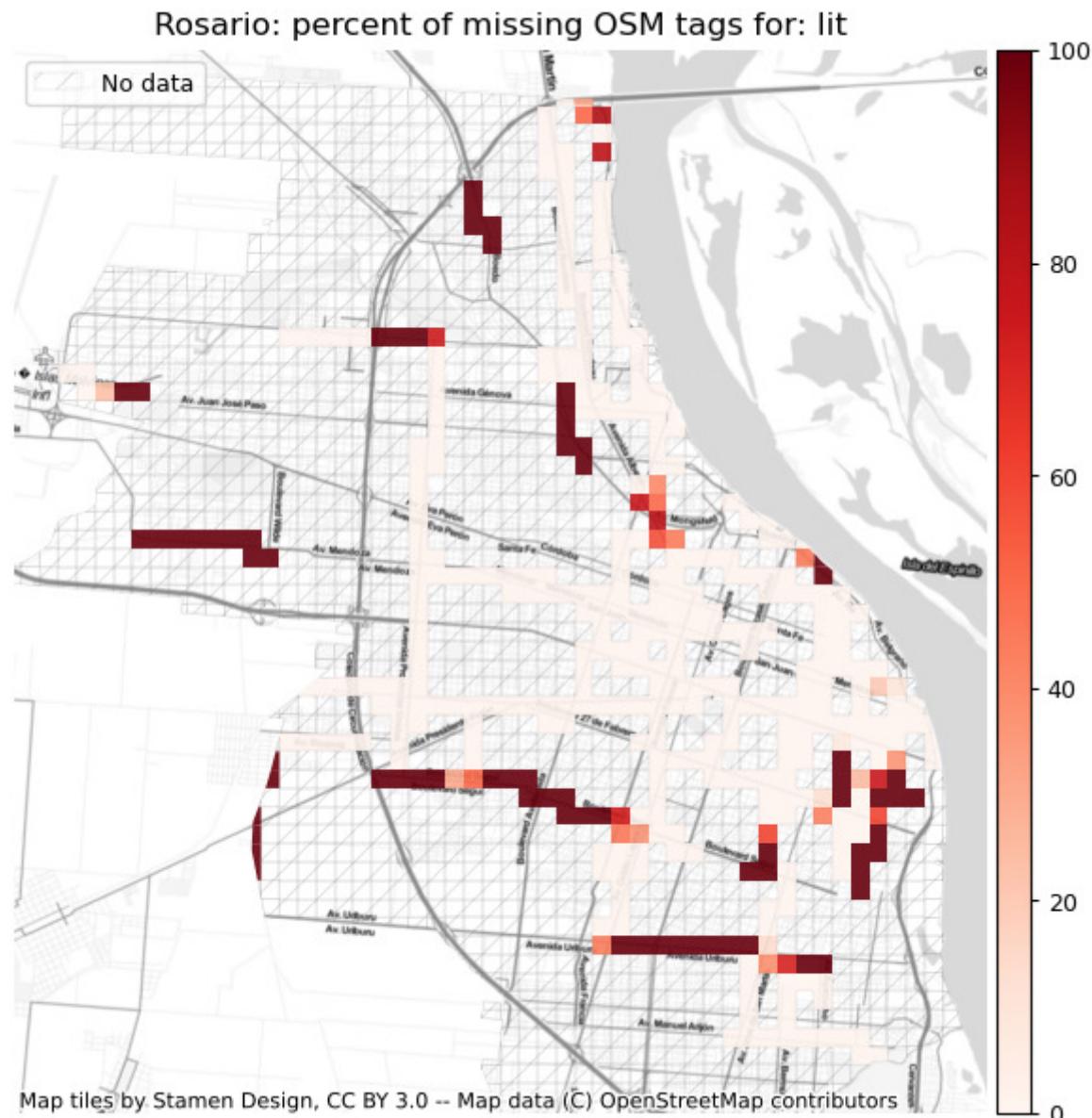
```
lit: 2862 out of 3534 edges (80.98%) have information.  
lit: 151 out of 187 km (80.92%) have information.
```

### Local missing tags









## Incompatible tags

Given that the tags in OSM data lack coherency at times and there are no restrictions in the tagging process (cf. [Barron et al., 2014](#)), incompatible tags might be present in the data set. For example, an edge might be tagged with the following two contradicting key-value pairs: `bicycle_infrastructure = yes` and `bicycle = no`.

### Method

In the `config.yml` file, a list of incompatible key-value pairs for tags in the `incompatible_tags_analysis` is defined. Since there is no limitation to which tags a data set could potentially contain, the list is, by definition, non-exhaustive, and can be adjusted by the user. In the section below, `check_incompatible_tags` is run, which identifies all incompatibility instances for a given area, first on the study area level and then on the grid cell level.

### Interpretation

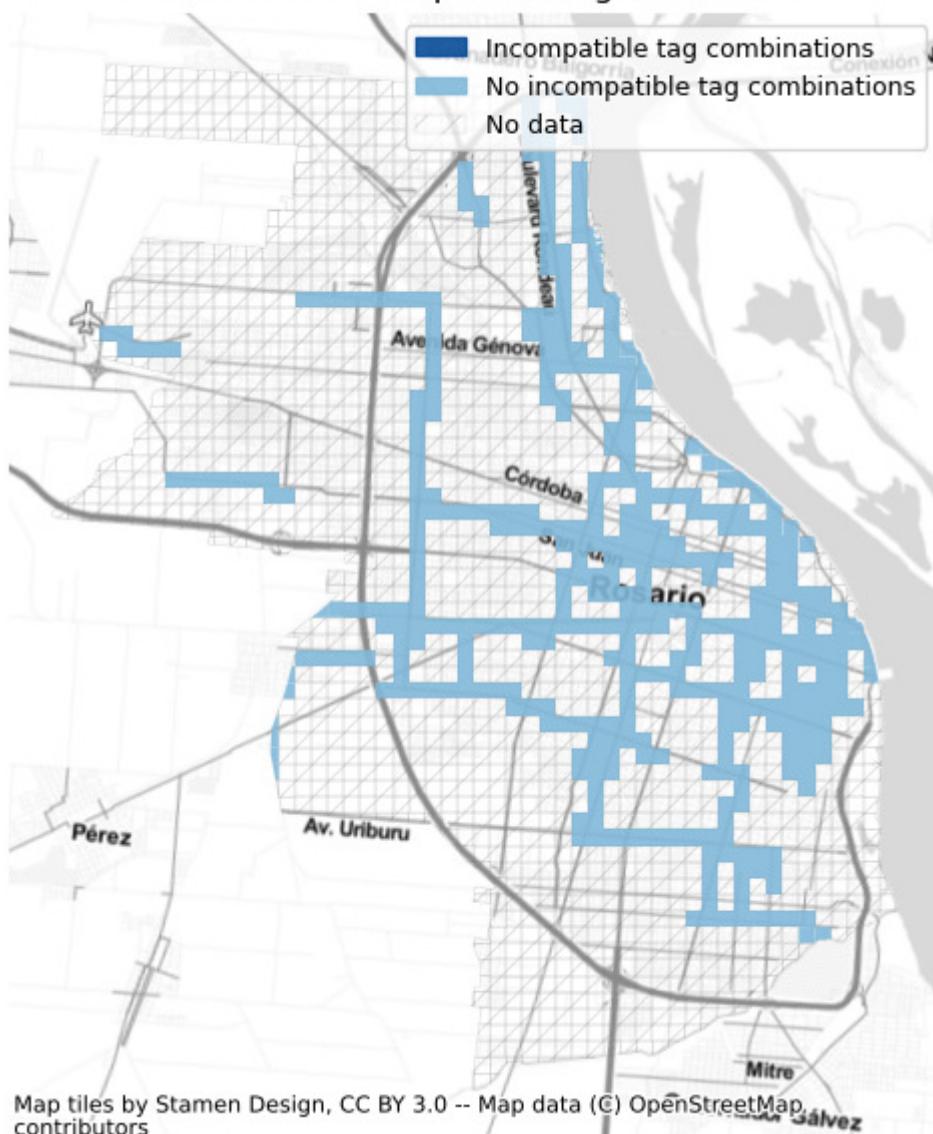
Incompatible tags are an undesired feature of the data set and render the corresponding data points invalid; there is no straightforward way to resolve the arising issues automatically, making it necessary to either correct the tag manually or to exclude the data point from the data set. A higher-than-average number of incompatible tags in a grid cell suggests local mapping issues.

#### Global incompatible tags (total number)

In the entire data set, there are 0 incompatible tag combinations (of those defined in the configuration file).

#### Local incompatible tags (per grid cell)

##### Rosario: OSM incompatible tag combinations



#### Plotting incompatible tag geometries

## Tagging patterns

Identifying bicycle infrastructure in OSM can be tricky due to the many different ways in which the presence of bicycle infrastructure can be indicated. The [OSM Wiki](#) is a great resource for recommendations for how OSM features should be tagged, but some inconsistencies and local variations can remain. The analysis of tagging patterns allows to visually explore some of the potential inconsistencies.

Regardless of how the bicycle infrastructure is defined, examining which tags contribute to which parts of the bicycle network allows to visually examine patterns in tagging methods. It also allows to estimate whether some elements of the query will lead to the inclusion of too many or too few features.

Likewise, 'double tagging' where several different tags have been used to indicate bicycle infrastructure can lead to misclassifications of the data. For this reason, identifying features that are included in more than one of the queries defining bicycle infrastructure can indicate issues with the tagging quality.

### Method

We first plot individual subsets of the OSM data set for each of the queries listed in `bicycle_infrastructure_queries`, as defined in the `config.yml` file. The subset defined by a query is the set of edges for which this query is *True*. Since several queries can be *True* for the same edge, the subsets can overlap. In the second step below, all overlaps between 2 or more queries are plotted, i.e. all edges that have been assigned several, potentially competing, tags.

### Interpretation

The plots for each tagging type allow for a quick visual overview of different tagging patterns present in the area. Based on local knowledge, the user may estimate whether the differences in tagging types are due to actual physical differences in the infrastructure or rather an artefact of the OSM data. Next, the user can access overlaps between different tags; depending on the specific tags, this may or may not be a data quality issue. For example, in case of '`'cycleway:right'`' and '`'cycleway:left'`', having data for both tags is valid, but other combinations such as '`'cycleway='track'`' and '`'cycleway:left=lane'`' gives an ambiguous picture of what type of bicycle infrastructure is present.

### Tagging types

Interactive map saved at [results/OSM/Rosario/maps\\_interactive/taggingtypes\\_osm.html](#)

### Multiple tagging

## Rosario: OSM bicycle infrastructure defined with tags: highway + cycleway



Interactive map saved at [results/OSM/Rosario/maps\\_interactive/taggingcombinations\\_osm.html](results/OSM/Rosario/maps_interactive/taggingcombinations_osm.html)

## Network topology

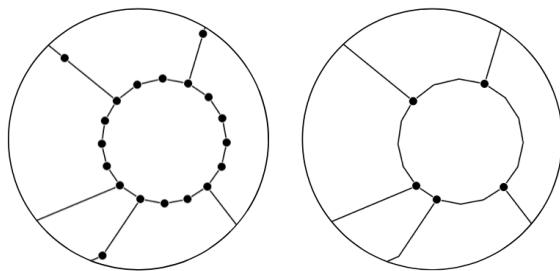
This section explores the geometric and topological features of the data. These are, for example, network density, disconnected components, and dangling (degree one) nodes. It also includes exploring whether there are nodes that are very close to each other but do not share an edge - a potential sign of edge undershoots - or if there are intersecting edges without a node at the intersection, which might indicate a digitizing error that will distort routing on the network.

Due to the fragmented nature of most bicycle networks, many metrics, such as missing links or network gaps, can simply reflect the true extent of the infrastructure ([Natera Orozco et al., 2020](#)). This is different for road networks, where e.g., disconnected components could more readily be interpreted as a data quality issue. Therefore, the analysis only takes very small network gaps into account as potential data quality issues.

## Simplification outcome

To compare the structure and true ratio between nodes and edges in the network, a simplified network representation which only includes nodes at endpoints and intersections was created in notebook 1a by removing all interstitial nodes.

Comparing the degree distribution for the networks before and after simplification is a quick sanity check for the simplification routine. Typically, the vast majority of nodes in the non-simplified network will be of degree two; in the simplified network, however, most nodes will have degrees other than two. Degree two nodes are retained in only two cases: if they represent a connection point between two different types of infrastructure; or if they are needed in order to avoid self-loops (edges whose start and end points are identical) or multiple edges between the same pair of nodes.



*Non-simplified network (left) and simplified network (right).*

### Method

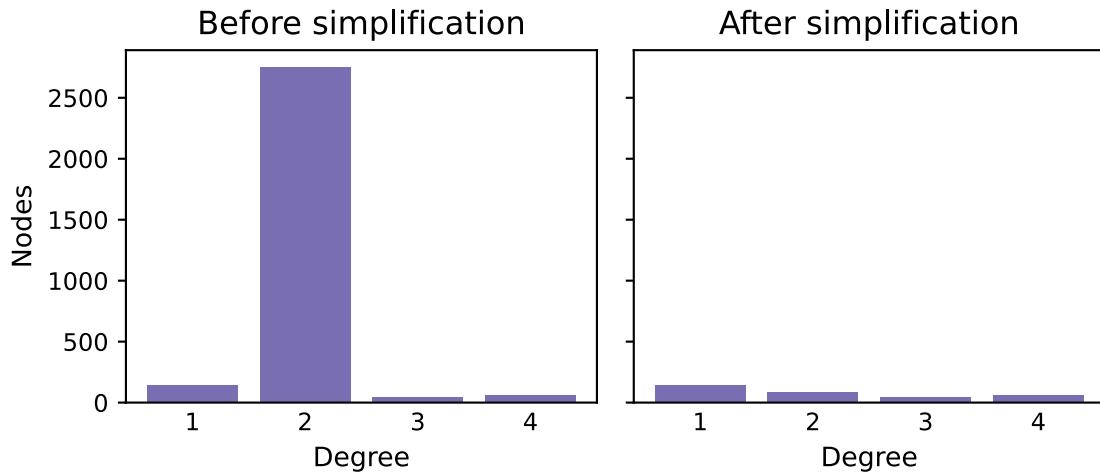
The degree distributions before and after simplification are plotted below.

### Interpretation

Typically, the degree distribution will go from high (before simplification) to low (after simplification) counts of degree two nodes, while it will not change for all other degrees (1, or 3 and higher). Further, the total number of nodes will see a strong decline. If the simplified graph still maintains a relatively high number of degree two nodes, or if the number of nodes with other degrees changes after the simplification, this might point to issues either with the graph conversion or with the simplification process.

Simplifying the network decreased the number of edges by 89.9% and the number of nodes by 88.6%.

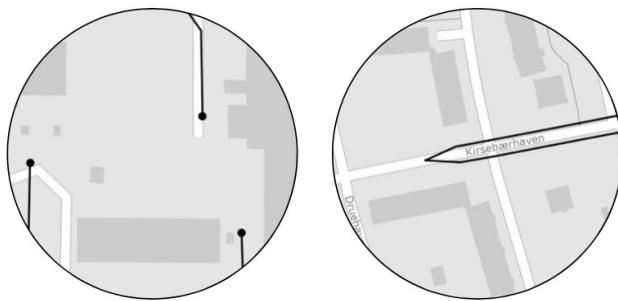
### Rosario: OSM degree distributions



### Dangling nodes

Dangling nodes are nodes of degree one, i.e. they have only one single edge attached to them. Most networks will naturally contain a number of dangling nodes. Dangling nodes can occur at actual dead-ends (representing a cul-de-sac) or at the endpoints of certain features, e.g. when a bicycle path ends in the middle of a street. However, dangling nodes can also occur as a data quality issue in case of over/undershoots (see next section). The number of dangling nodes in a network does to some extent also depend on the digitization method, as shown in the illustration below.

Therefore, the presence of dangling nodes is in itself not a sign of low data quality. However, a high number of dangling nodes in an area that is not known for containing many dead-ends can indicate digitization errors and problems with edge over/undershoots.



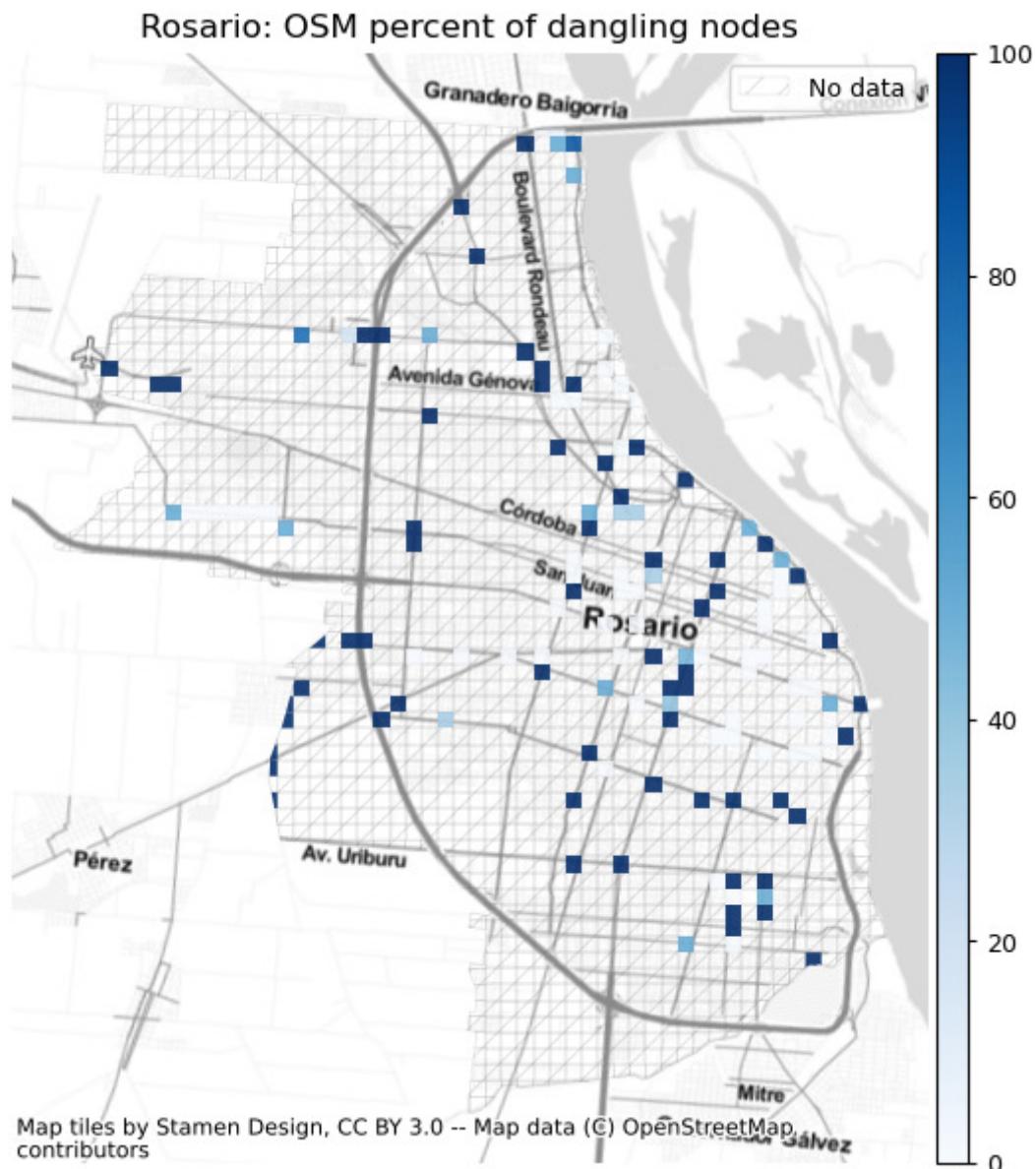
*Left: Dangling nodes occur where road features end. Right: However, when separate features are joined at the end, there will be no dangling nodes.*

### Method

Below, a list of all dangling nodes is obtained with the help of `get_dangling_nodes`. Then, the network with all its nodes is plotted. The dangling nodes are shown in color, all other nodes are shown in black.

### Interpretation

We recommend a visual analysis in order to interpret the spatial distribution of dangling nodes, with particular attention to areas of high dangling node density. It is important to understand where dangling nodes come from: are they actual dead-ends or digitization errors (e.g., over/undershoots)? A higher number of digitization errors points to lower data quality.

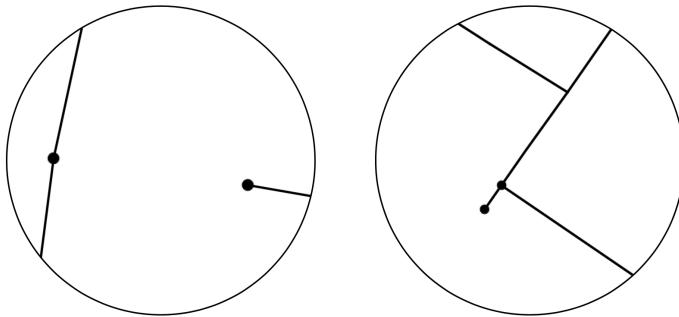


Interactive map saved at [results/OSM/Rosario/maps\\_interactive/danglingmap\\_osm.html](results/OSM/Rosario/maps_interactive/danglingmap_osm.html)

## Under/overshoots

When two nodes in a simplified network are placed within a distance of a few meters, but do not share a common edge, it is often due to an edge over/undershoot or another digitizing error. An undershoot occurs when two features are supposed to meet, but instead are just in close proximity to each other. An overshoot occurs when two features meet and one of them extends beyond the other. See the image

below for an illustration. For a more detailed explanation of over/undershoots, see the [GIS Lounge website](#).



*Left: Undershoots happen when two line features are not properly joined, for example at an intersection. Right: Overshoots refer to situations where a line feature extends too far beyond an intersecting line, rather than ending at the intersection.*

## Method

**Undershoots:** First, the `length_tolerance` (in meters) is defined in the cell below. Then, with `find_undershoots`, all pairs of dangling nodes that have a maximum of `length_tolerance` distance between them, are identified as undershoots, and the results are plotted.

**Overshoots:** First, the `length_tolerance` (in meters) is defined in the cell below. Then, with `find_overshoots`, all network edges that have a dangling node attached to them and that have a maximum length of `length_tolerance` are identified as overshoots, and the results are plotted.

The method for over/undershoot detection is inspired by [Neis et al. \(2012\)](#).

## Interpretation

Under/overshoots are not necessarily always a data quality issue – they might be instead an accurate representation of the network conditions or of the digitization strategy. For example, a cycle path might end abruptly soon after a turn, which results in an overshoot. Protected cycle paths are sometimes digitized in OSM as interrupted at intersections which results in intersection undershoots.

The interpretation of the impact of over/undershoots on data quality is context dependent. For certain applications, such as routing, overshoots do not present a particular challenge; they can, however, pose an issue for other applications such as network analysis, given that they skew the network structure. Undershoots, on the contrary, are a serious problem for routing applications, especially if only bicycle infrastructure is considered. They also pose a problem for network analysis, for example for any path-based metric, such as most centrality measures like betweenness centrality.

1 potential overshoots were identified using a length tolerance of 3 m.  
0 potential undershoots were identified using a length tolerance of 3 m.

Interactive map saved at [results/OSM/Rosario/maps\\_interactive/underovershoots\\_3\\_3.osm.html](#)

## Missing intersection nodes

When two edges intersect without having a node at the intersection - and if neither edges are tagged as a bridge or a tunnel - there is a clear indication of a topology error.

## Method

First, with the help of `check_intersection`, each edge which is not tagged as either tunnel or bridge is checked for any *crossing* with another edge of the network. If this is the case, the edge is marked as having an intersection issue. The number of intersection issues found is printed and the results are plotted for visual analysis. The method is inspired by [Neis et al. \(2012\)](#).

## Interpretation

A higher number of intersection issues points to a lower data quality. However, it is recommended with a manual visual check of all intersection issues with a certain knowledge of the area, in order to determine the origin of intersection issues and confirm/correct/reject them.

```
-----
AttributeError                                     Traceback (most recent call last)
Cell In[32], line 1
----> 1 missing_nodes_edge_ids, edges_with_missing_nodes = eval_func.find_missing_intersections(
      2     osm_edges, "edge_id"
      3 )
      5 count_intersection_issues = (
      6     len(missing_nodes_edge_ids) / 2
      7 ) # The number of issues is counted twice since both intersecting osm_edges are returned
      9 print(
      10    f"{count_intersection_issues:.0f} place(s) appear to be missing an intersection node
or a bridge/tunnel tag."
      11 )

File ~/Library/CloudStorage/OneDrive-ITU/projects/BikeDNA-usecases/src/evaluation_functions.py:4
70, in find_missing_intersections(edges, edge_id_col, return_edges)
  461 """
  462 Detects topological errors in gdf with edges from OSM data.
  463 If two edges are intersecting (i.e. no node at intersection) and neither is tagged as a
bridge or a tunnel,
  464 it is considered an error in the data.
  465 """
  466 # Don't include tunnels or bridges
  467 edges_subset = edges.loc[
  468     ~(
--> 470         edges.tunnel.isin(
  471             ["yes", "Yes", True, "passage", "building_passage", "movable"]
  472         )
  473         | edges.bridge.isin(
  474             ["yes", "Yes", True, "passage", "building_passage", "movable"]
  475         )
  476     )
  477 ].copy()
  478 edges_subset["intersection_issues"] = edges_subset.apply(
  479     lambda x: check_crossing(row=x, gdf=edges_subset), axis=1
  480 )
  481 )
  482 missing_nodes = list(
  483     edges_subset.loc[
  484         (edges_subset.intersection_issues.notna())
  485     ][edge_id_col].values
  486 )
  487 
```

```
File ~/opt/anaconda3/envs/bikedna/lib/python3.11/site-packages/pandas/core/generic.py:5902, in N
DFrame.__getattr__(self, name)
5895 if (
5896     name not in self._internal_names_set
5897     and name not in self._metadata
5898     and name not in self._accessors
5899     and self._info_axis._can_hold_identifiers_and_holds_name(name)
5900 ):
5901     return self[name]
-> 5902 return object.__getattribute__(self, name)

AttributeError: 'GeoDataFrame' object has no attribute 'tunnel'
```

## Network components

Disconnected components do not share any elements (nodes/edges). In other words, there is no network path that could lead from one disconnected component to the other. As mentioned above, most real-world networks of bicycle infrastructure do consist of many disconnected components ([Natera Orozco et al., 2020](#)). However, when two disconnected components are very close to each other, it might be a sign of a missing edge or another digitizing error.

### Method

First, with the help of `return_components`, a list of all (disconnected) components of the network is obtained. The total number of components is printed and all components are plotted in different colors for visual analysis. Next, the component size distribution (with components ordered by the network length they contain) is plotted, followed by a plot of the largest connected component.

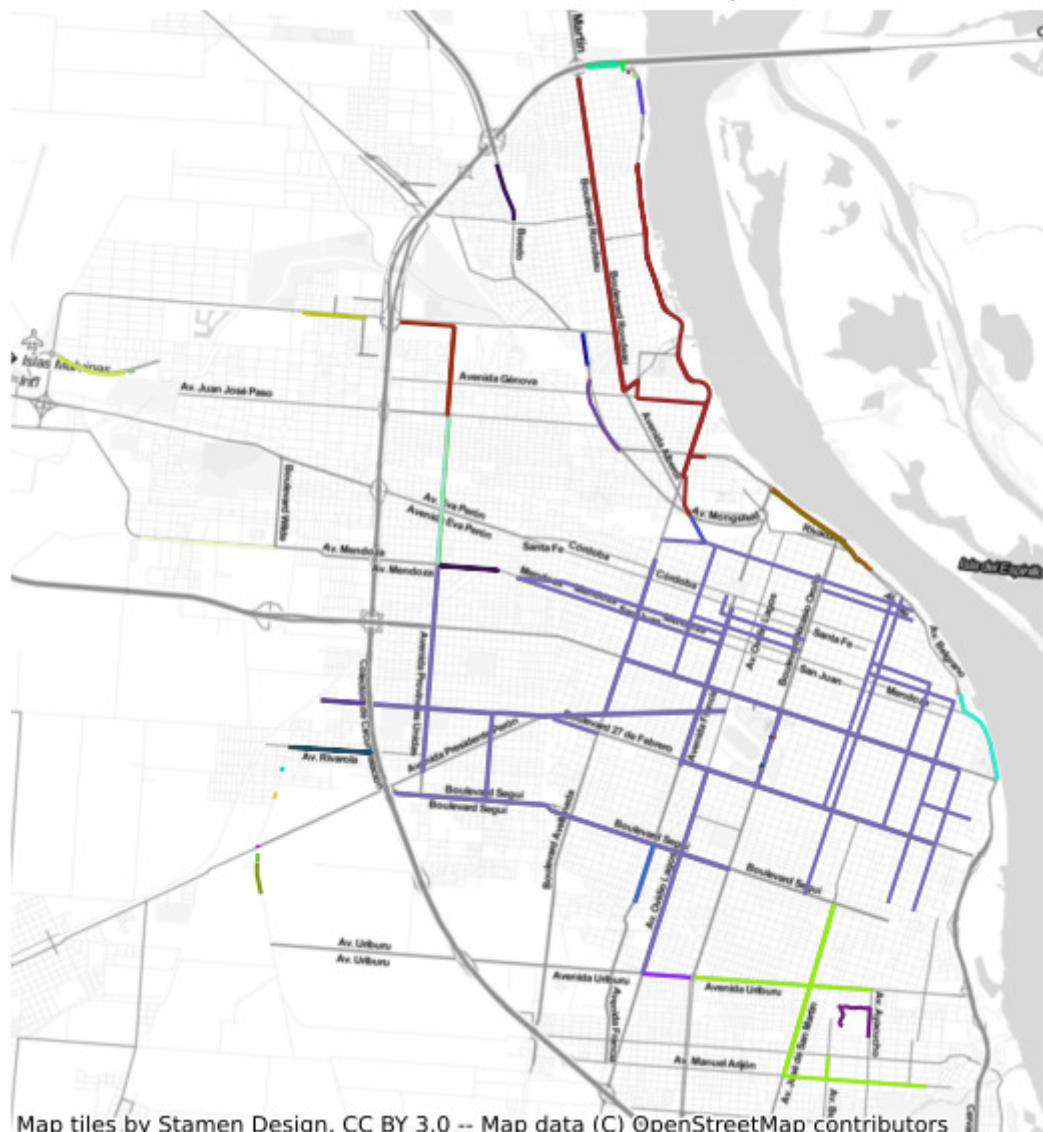
### Interpretation

As with many of the previous analysis steps, knowledge of the area is crucial for a correct interpretation of component analysis. Given that the data represents the actual infrastructure accurately, bigger components indicate coherent network parts, while smaller components indicate scattered infrastructure (e.g., one single bicycle path along a street that does not connect to any other bicycle infrastructure). A high number of disconnected components in near vicinity of each other indicates digitization errors or missing data.

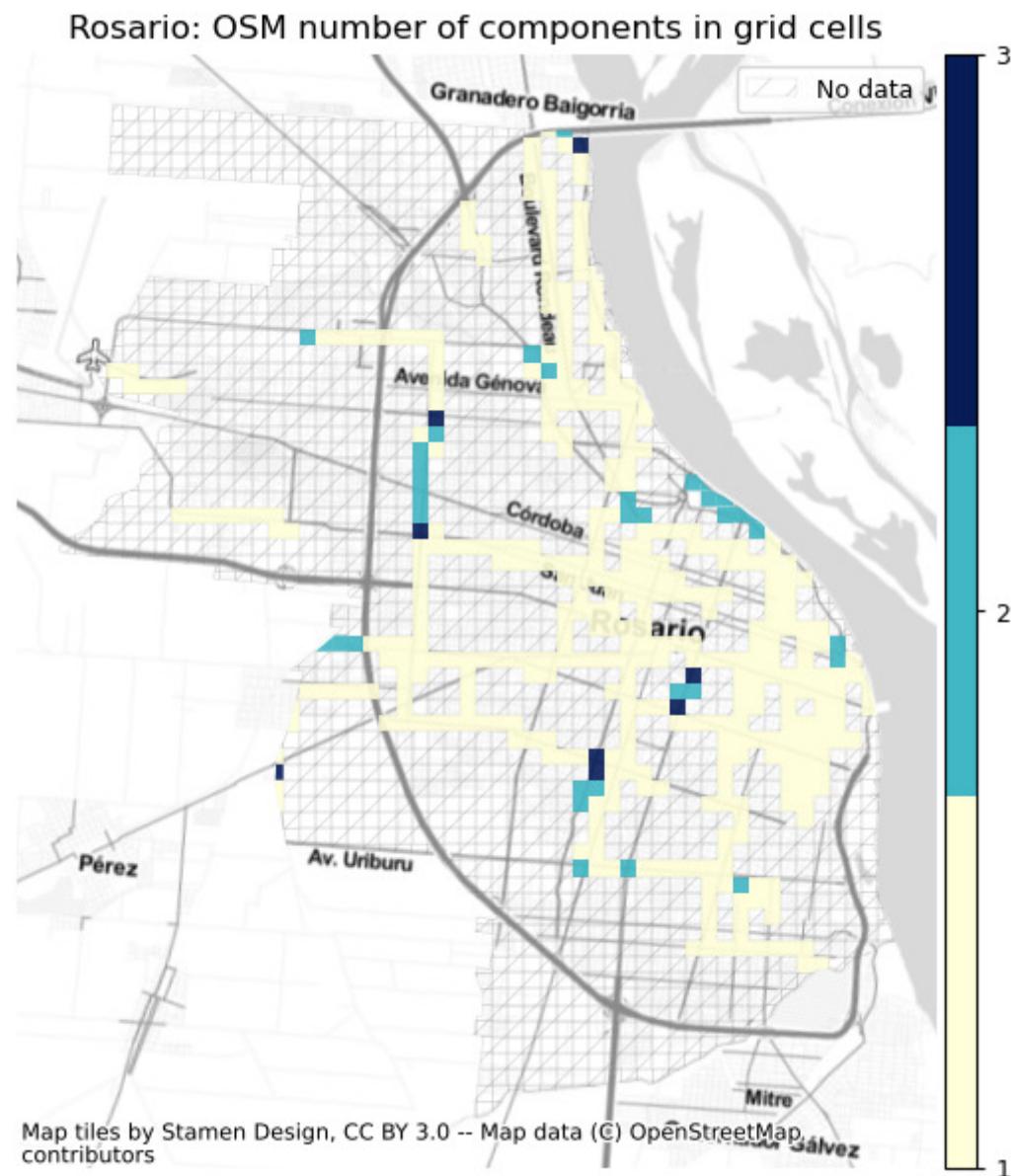
## Disconnected components

The network in the study area has 43 disconnected components.

### Rosario: OSM disconnected components



Components per grid cell



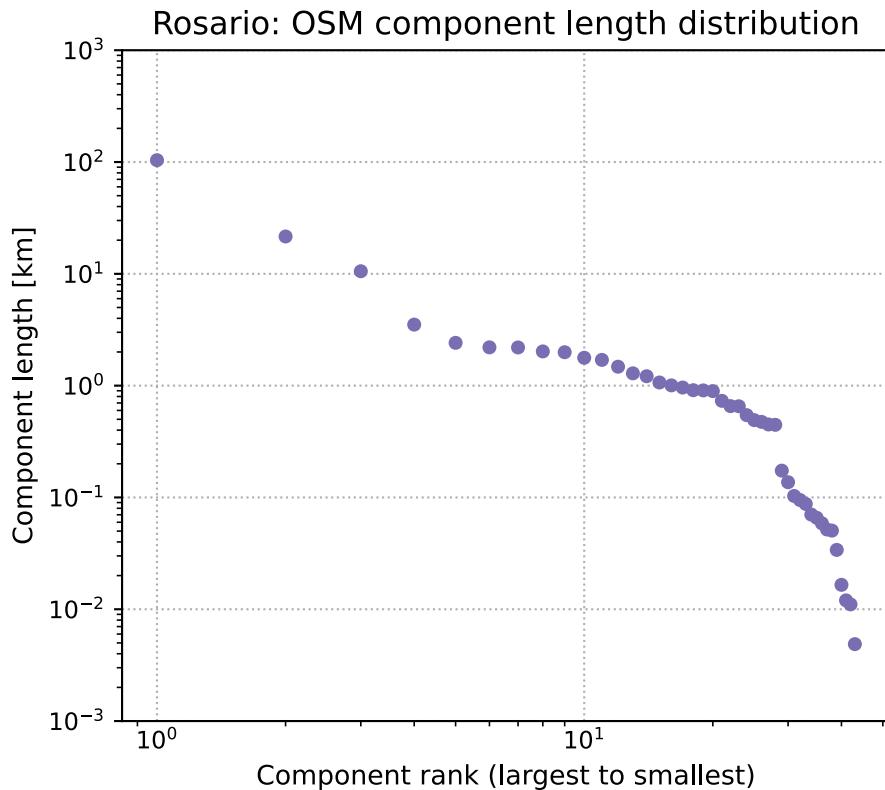
### Component length distribution

The distribution of all network component lengths can be visualized in a so-called *Zipf plot*, which orders the lengths of each component by rank, showing the largest component's length on the left, then the second largest component's length, etc., until the smallest component's length on the right. When a Zipf plot follows a straight line in [log-log scale](#), it means that there is a much higher chance to find small disconnected components than expected from traditional distributions ([Clauset et al., 2009](#)). This can mean that there has been no consolidation of the network, only piece-wise or random additions ([Szell et al., 2022](#)), or that the data itself suffers from many gaps and topology errors resulting in small disconnected components.

However, it can also happen that the largest connected component (the leftmost marker in the plot at rank  $10^0$ ) is a clear outlier, while the rest of the plot follows a different shape. This can mean that at the

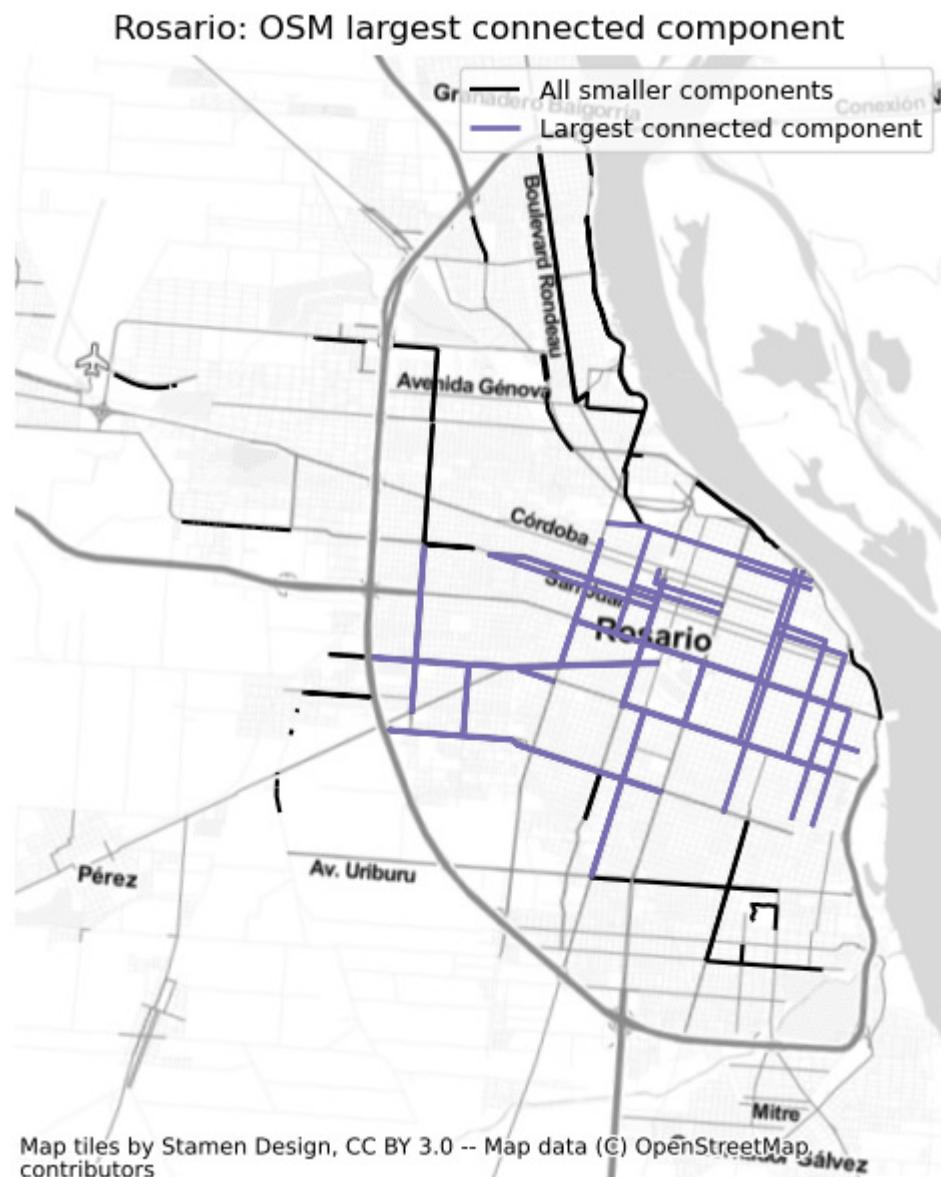
infrastructure level, most of the infrastructure has been connected to one large component, and that the data reflects this - i.e. the data is not suffering from gaps and missing links to a large extent.

Bicycle networks might also be somewhere inbetween, with several large components as outliers.



### Largest connected component

The largest connected component contains 61.48% of the network length.



## Missing links

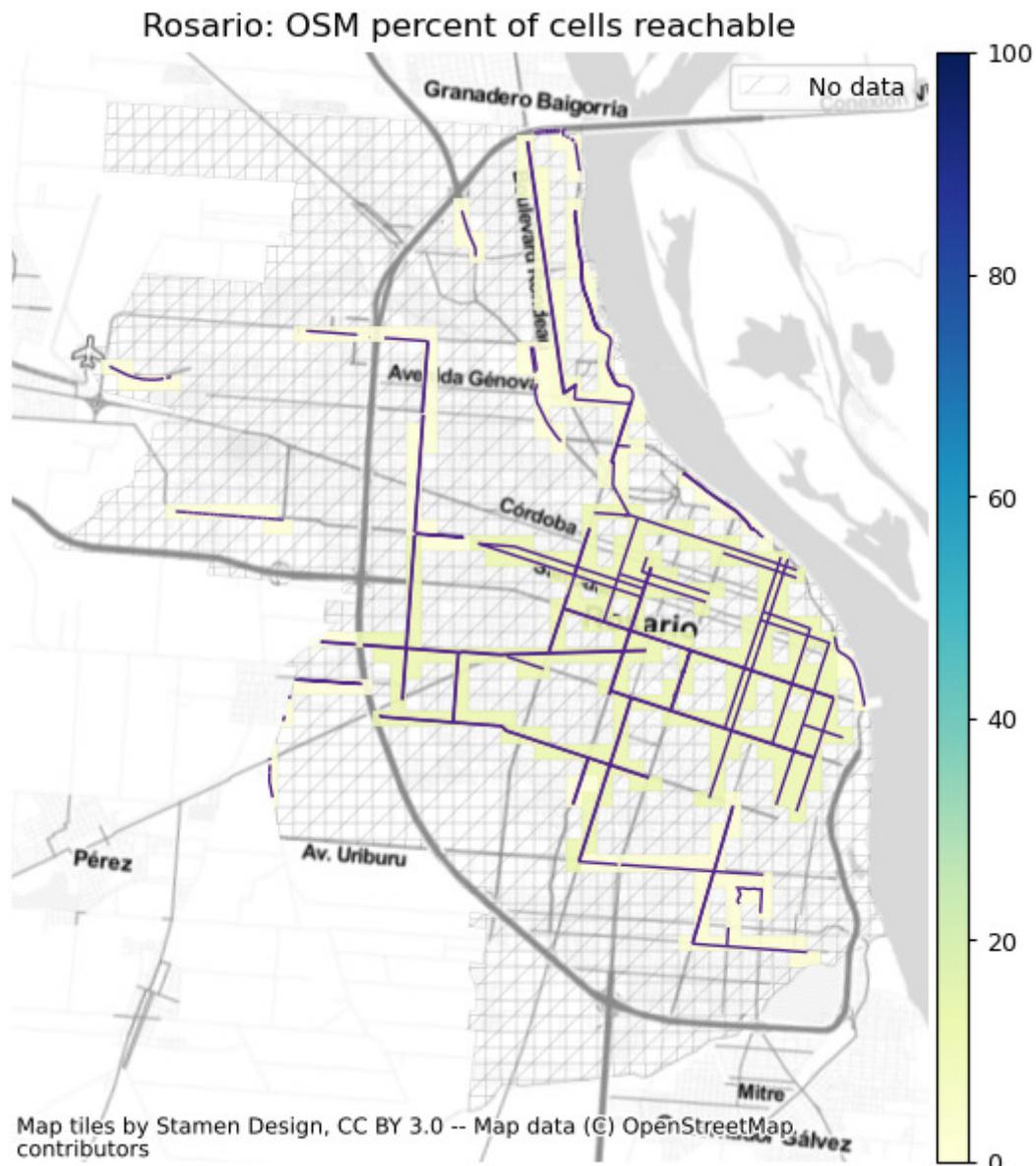
In the plot of potential missing links between components, all edges that are within the specified distance of an edge on another component are plotted. The gaps between disconnected edges are highlighted with a marker. The map thus highlights edges which, despite being in close proximity of each other, are disconnected and where it thus would not be possible to bike on cycling infrastructure between the edges.

Running analysis with component distance threshold of 10 meters.

Interactive map saved at [results/OSM/Rosario/maps\\_interactive/component\\_gaps\\_10\\_osm.html](results/OSM/Rosario/maps_interactive/component_gaps_10_osm.html)

## Component connectivity

Here we visualize differences between how many cells can be reached from each cell. This is a crude measure for network connectivity but has the benefit of being computationally cheap and thus able to quickly highlight stark differences in network connectivity.



## Summary

### Intrinsic Quality Metrics - OSM data

Total infrastructure length (km) 175

Protected bicycle infrastructure density (m/km<sup>2</sup>) 373

Unprotected bicycle infrastructure density (m/km2)	726
Mixed protection bicycle infrastructure density (m/km2)	3
Bicycle infrastructure density (m/km2)	1,103
Nodes	342
Dangling nodes	143
Nodes per km2	2
Dangling nodes per km2	1
Incompatible tag combinations	0
Overshoots	1
Undershoots	0
Components	43
Length of largest component (km)	104
Largest component's share of network length	61%
Component gaps	13

## 2a. Initialize reference data

This notebook:

- Loads the polygon defining the study area and then creates a grid overlay for the study area.
- Loads the reference data.
- Processes the reference data to create the network structure and attributes needed in the analysis.

### Sections

- [Load data for study area and create analysis grid](#)
- [Load and preprocess reference data](#)

## Load data for study area and create analysis grid

This step:

- Loads settings for the analysis from the configuration file `config.yml`.
- Reads data for the study area.
- Creates a grid overlay of the study area, with grid cell size defined in `config.yml`.

### Load data for study area

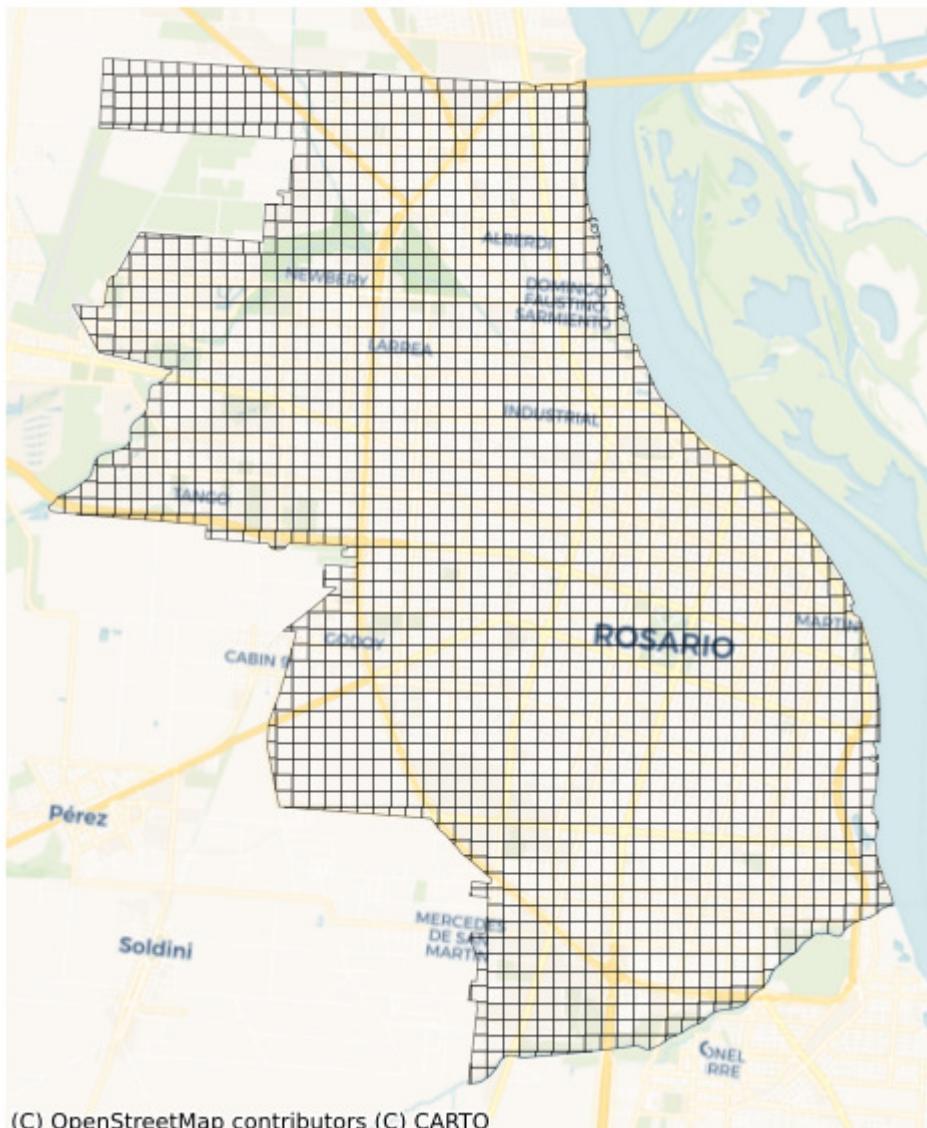
The study area is defined by the user-provided polygon. It will be used for the computation of **global** results, i.e. quality metrics based on all data in the study area.

The size of the study area is 159.16 km<sup>2</sup>.

### Create analysis grid

The grid contains 1930 square cells with a side length of 300 m and an area of 0.09 km<sup>2</sup>. This grid will be used for local (grid cell level) analysis:

## Rosario study area (1930 grid cells, side length 300m)



## Load and preprocess reference data

This step:

- Creates a network from the reference data.
- Projects it to the chosen CRS.
- Clips the data to the polygon defining the study area.
- Measures the infrastructure length of the edges based on the geometry type and whether they allow for bidirectional travel or not.
- Simplifies the network.
- Creates copies of all edge and node data sets indexed by their intersecting grid cell.

### Network data model

In BikeDNA, all input data are converted to a network structure consisting of *nodes* and *edges*. Edges represents the actual infrastructure, such as bike lanes and paths, while nodes represents the start and end points for the edges, as well as all intersections. For further details, read more about the [network data model](#).

### ► Network simplification

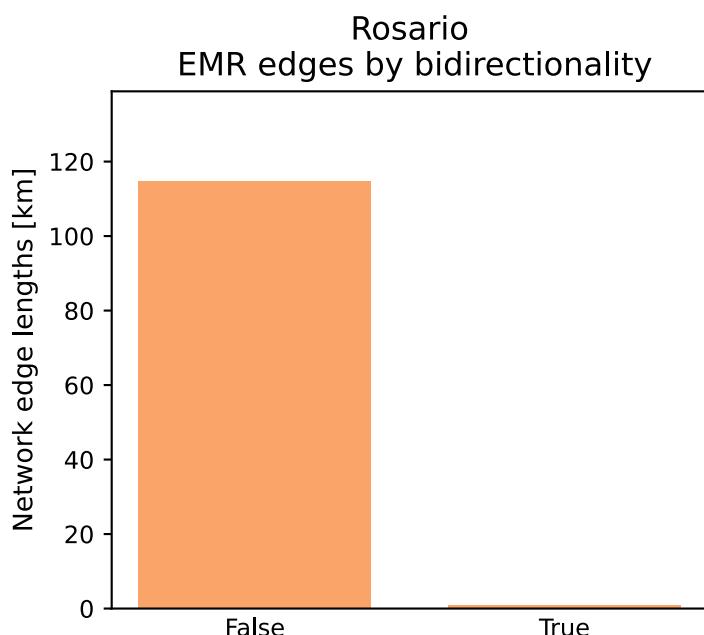
The EMR data covers an area of 96.31 km<sup>2</sup>.

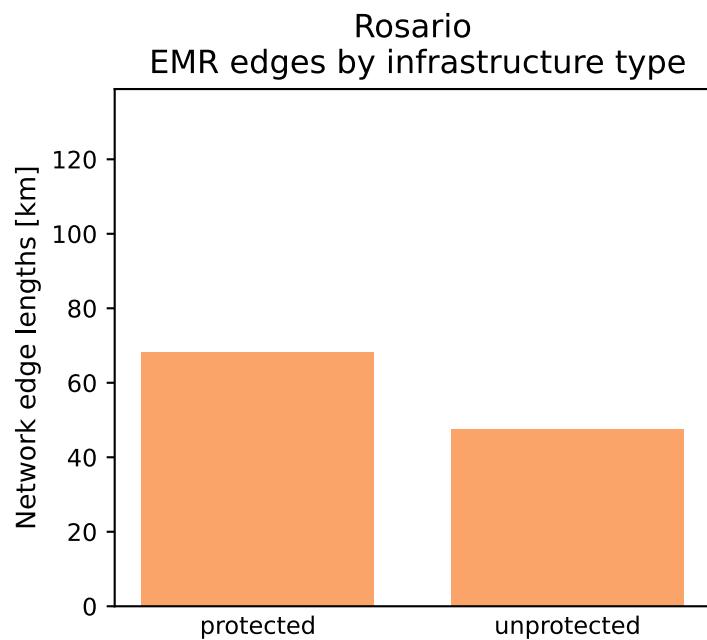
Edges where the protection level is 'protected': 672 out of 1214 (55.35%)  
Edges where the protection level is 'unprotected': 542 out of 1214 (44.65%)

Edges where 'bicycle\_bidirectional' is False: 1205 out of 1214 (99.26%)  
Edges where 'bicycle\_bidirectional' is True: 9 out of 1214 (0.74%)

Using global settings for geometry type.

The length of the EMR network is 116.71 km.







## 2b. Intrinsic Analysis of Reference Data

This notebook analyses the quality of a user-provided reference bicycle infrastructure data set for a given area. The quality assessment is *intrinsic*, i.e. based only on the one input data set, and making no use of information external to the data set. For an extrinsic quality assessment that compares the reference data set to corresponding OSM data, see the notebooks 3a and 3b.

The analysis assesses the *fitness for purpose* ([Barron et al., 2014](#)) of the reference data for a given area. Outcomes of the analysis can be relevant for bicycle planning and research - especially for projects that include a network analysis of bicycle infrastructure, in which case the topology of the geometries is of particular importance.

Since the assessment does not make use of an external reference data set as the ground truth, no universal claims of data quality can be made. The idea is rather to enable those working with bicycle networks to assess whether their data are good enough for their particular use case. The analysis assists in finding potential data quality issues but leaves the final interpretation of the results to the user.

The notebook makes use of quality metrics from a range of previous projects investigating OSM/VGI data quality, such as [Ferster et al. \(2020\)](#), [Hochmair et al. \(2015\)](#), [Barron et al. \(2014\)](#), and [Neis et al. \(2012\)](#).

### Familiarity required

For a correct interpretation of some of the metrics for spatial data quality, some familiarity with the area is necessary.

- Data completeness
  - Network density
- Network topology
  - Simplification outcome
  - Dangling nodes
  - Under/overshoots
- Network components
  - Disconnected components
  - Components per grid cell
  - Component length distribution
  - Largest connected component
  - Missing links
  - Component connectivity
- Summary

## Data completeness

### Network density

In this setting, network density refers to the length of edges or number of nodes per km<sup>2</sup>. This is the usual definition of network density in spatial (road) networks, which is distinct from the *structural* network density known more generally in network science. Without comparing to a reference data set, network density does not in itself indicate spatial data quality. For anyone familiar with the study area, network density can however indicate whether parts of the area appear to be under- or over-mapped.

## Method

The density here is not based on the geometric length of edges, but instead on the computed length of the infrastructure. For example, a 100-meter-long bidirectional path contributes with 200 meters of bicycle infrastructure. This method is used to take into account different ways of mapping bicycle infrastructure, which otherwise can introduce large deviations in network density. With `compute_network_density`, the number of elements (nodes, dangling nodes, and total infrastructure length) per unit area is calculated. The density is computed twice: first for the study area for both the entire network ('global density'), then for each of the grid cells ('local density'). Both global and local densities are computed for the entire network and for protected and unprotected infrastructure.

## Interpretation

Since the analysis conducted here is intrinsic, i.e. it makes no use of external information, it cannot be known whether a low-density value is due to incomplete mapping, or due to actual lack of infrastructure in the area. However, a comparison of the grid cell density values can provide some insights, for example:

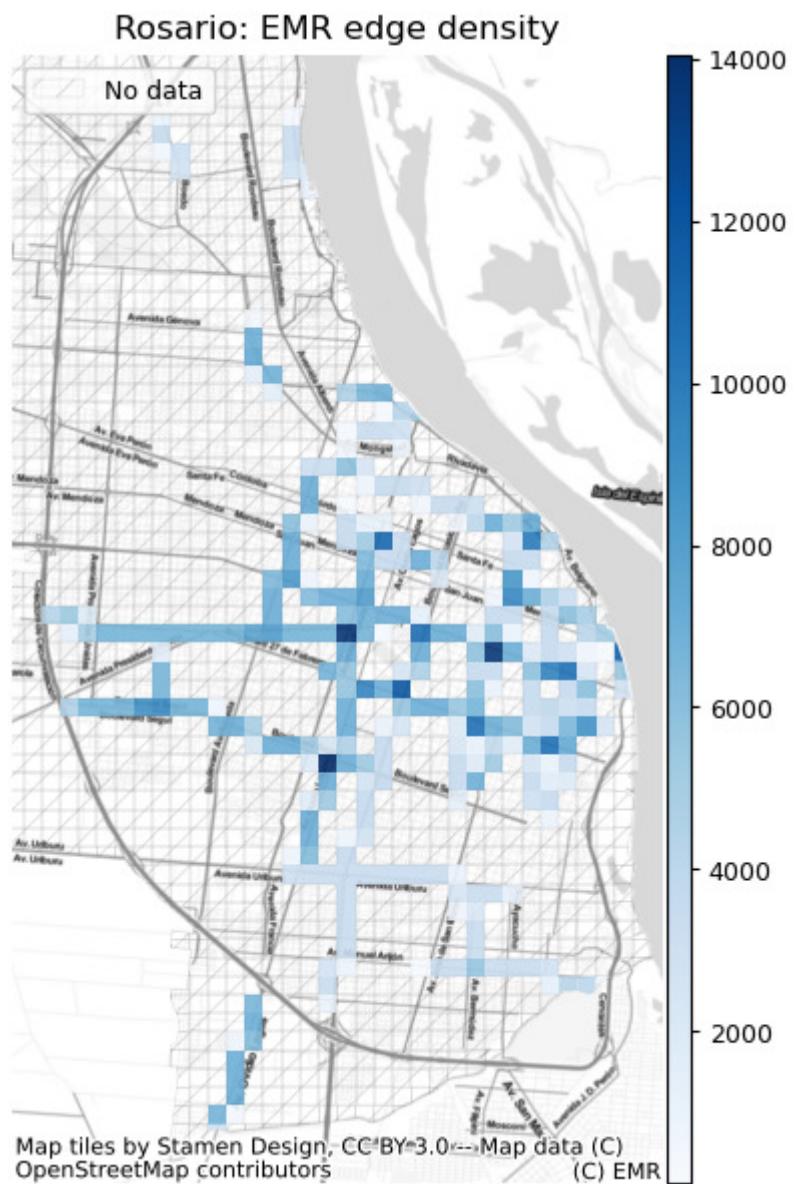
- lower-than-average infrastructure density indicates a locally sparser network
- higher-than-average node density indicates that there are relatively many intersections in a grid cell
- higher-than-average dangling node density indicates that there are relatively many dead ends in a grid cell

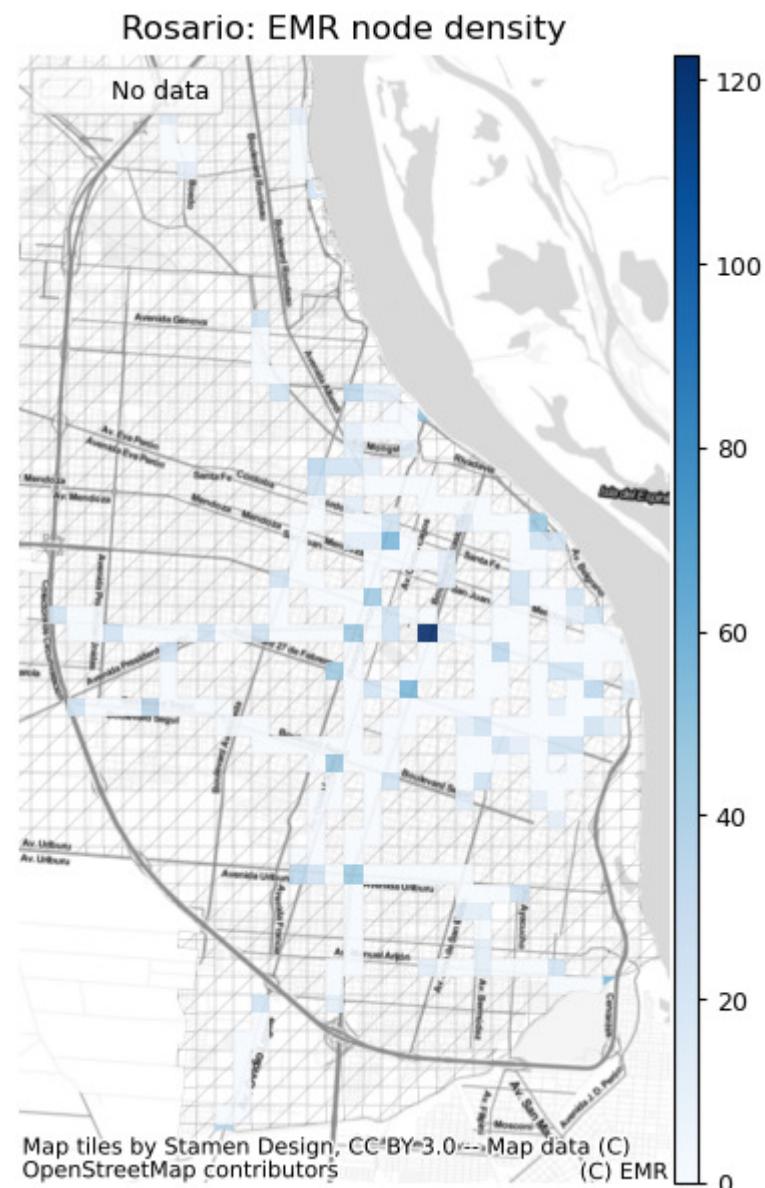
## Global network density

For the entire study area, there are:

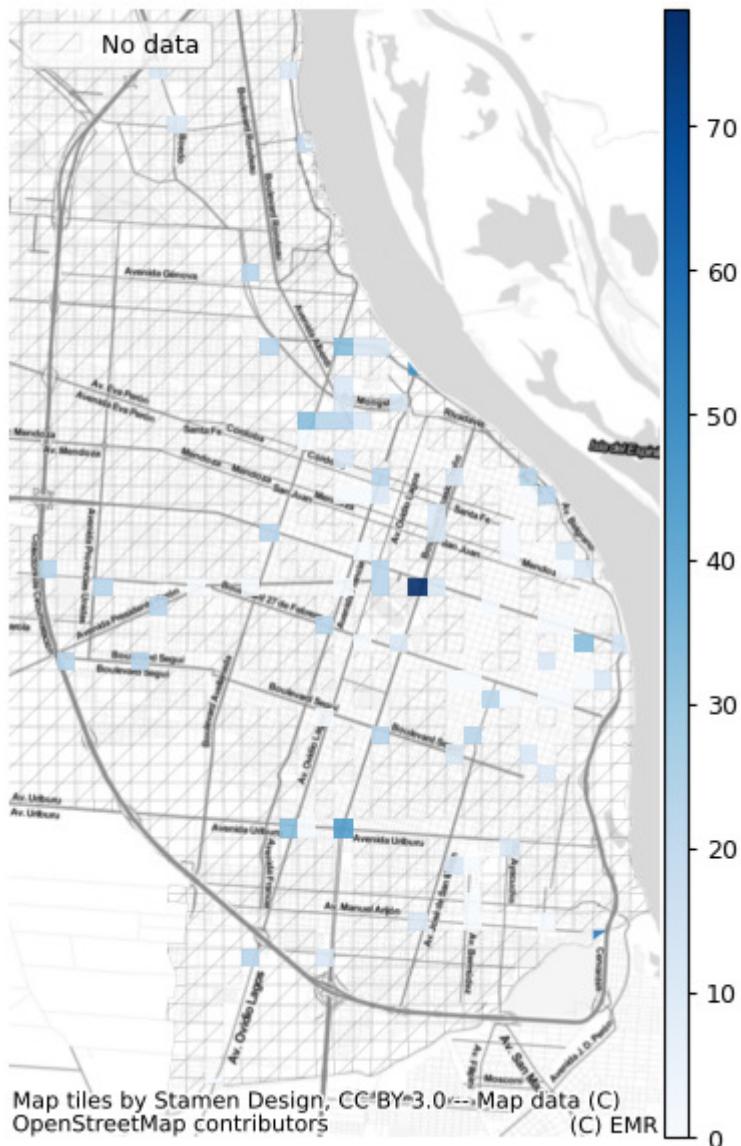
- 733.29 meters of bicycle infrastructure per km<sup>2</sup>.
- 1.06 nodes in the bicycle network per km<sup>2</sup>.
- 0.59 dangling nodes in the bicycle network per km<sup>2</sup>.
- 434.73 meters of protected bicycle infrastructure per km<sup>2</sup>.
- 298.57 meters of unprotected bicycle infrastructure per km<sup>2</sup>.
- 0.00 meters of mixed protection bicycle infrastructure per km<sup>2</sup>.

## Local network density





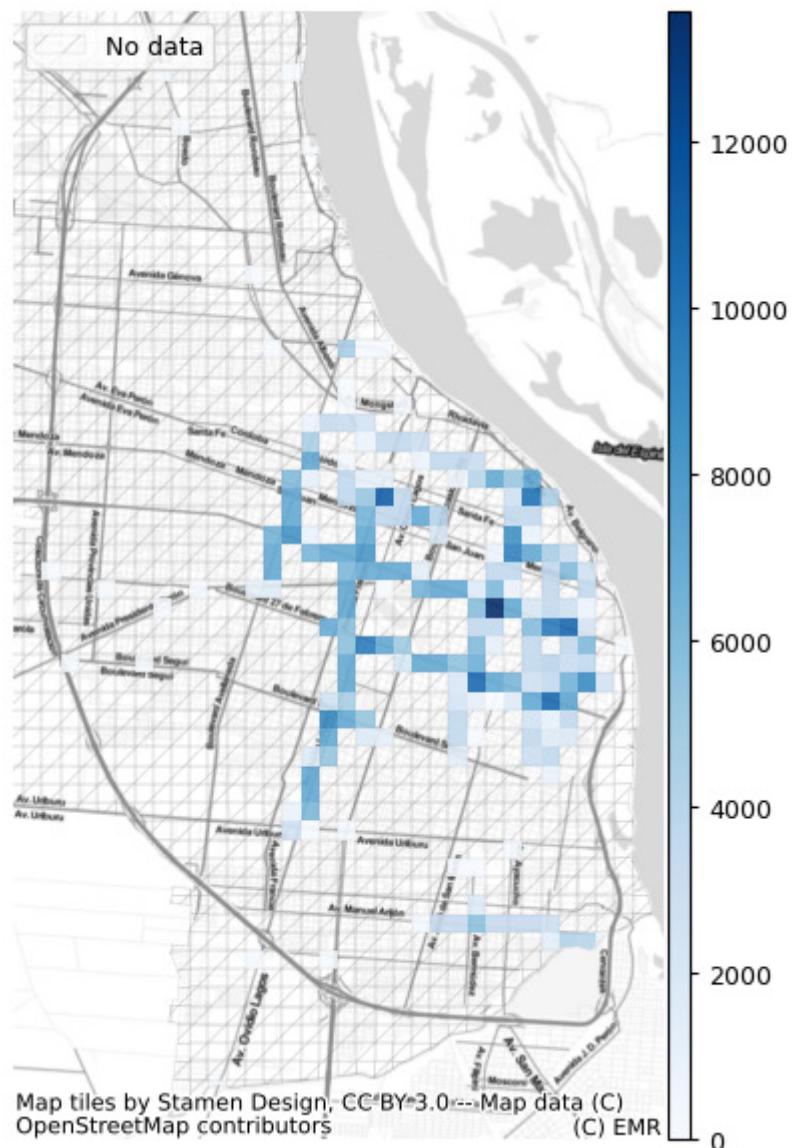
## Rosario: EMR dangling node density



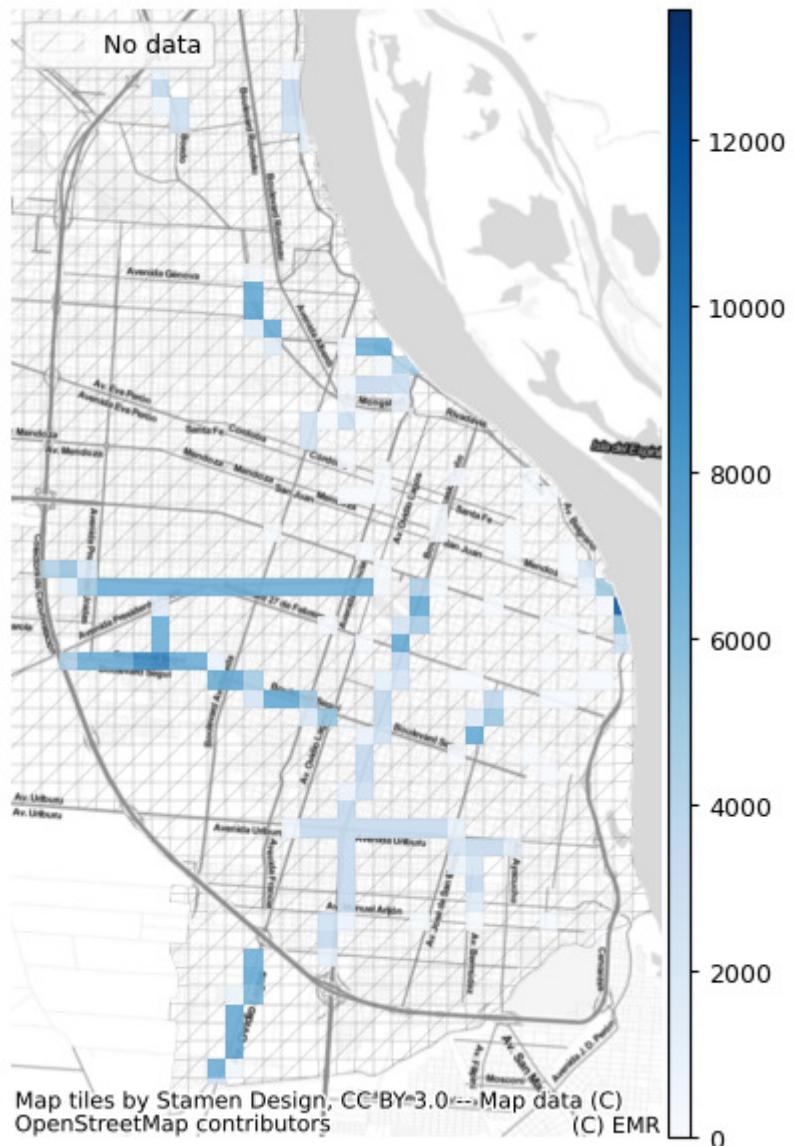
### Densities of protected and unprotected infrastructure

In BikeDNA, *protected infrastructure* refers to all bicycle infrastructure which is either separated from car traffic by for example an elevated curb, bollards, or other physical barriers, or for cycle tracks that are not adjacent to a street.

*Unprotected infrastructure* are all other types of lanes that are dedicated for bicyclists, but which only are separated by car traffic by e.g., a painted line on the street.

**Rosario: EMR protected infrastructure density (m/km<sup>2</sup>)**

## Rosario: EMR unprotected infrastructure density (m/km<sup>2</sup>)



## Network topology

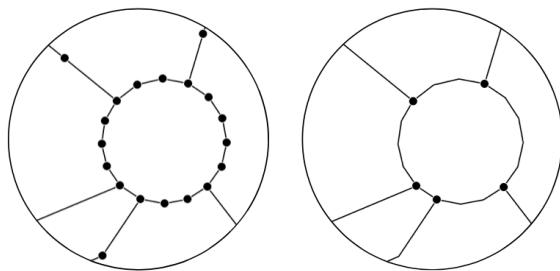
This section explores the geometric and topological features of the data. These are, for example, network density, disconnected components, and dangling (degree one) nodes. It also includes exploring whether there are nodes that are very close to each other but do not share an edge - a potential sign of edge undershoots - or if there are intersecting edges without a node at the intersection, which might indicate a digitizing error that will distort routing on the network.

Due to the fragmented nature of most bicycle networks, many metrics, such as missing links or network gaps, can simply reflect the true extent of the infrastructure ([Natera Orozco et al., 2020](#)). This is different for road networks, where e.g., disconnected components could more readily be interpreted as a data quality issue. Therefore, the analysis only takes very small network gaps into account as potential data quality issues.

## Simplification outcome

To compare the structure and true ratio between nodes and edges in the network, a simplified network representation which only includes nodes at endpoints and intersections was created in notebook 1b by removing all interstitial nodes.

Comparing the degree distribution for the networks before and after simplification is a quick sanity check for the simplification routine. Typically, the vast majority of nodes in the non-simplified network will be of degree two; in the simplified network, however, most nodes will have degrees other than two. Degree two nodes are retained in only two cases: if they represent a connection point between two different types of infrastructure; or if they are needed in order to avoid self-loops (edges whose start and end points are identical) or multiple edges between the same pair of nodes.



*Non-simplified network (left) and simplified network (right).*

### Method

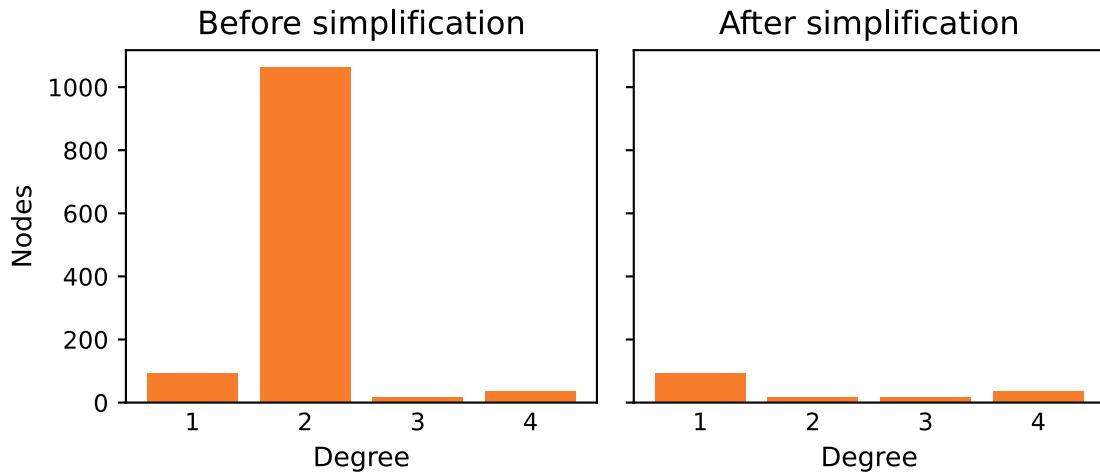
The node degree distributions before and after simplification are plotted below.

### Interpretation

Typically, the node degree distribution will go from high (before simplification) to low (after simplification) counts of degree two nodes, while it will not change for all other degrees (1, or 3 and higher). Further, the total number of nodes will see a strong decline. If the simplified graph still maintains a relatively high number of degree two nodes, or if the number of nodes with other degrees changes after the simplification, this might point to issues either with the graph conversion or with the simplification process.

Simplifying the network decreased the number of edges with 86.2% and the number of nodes with 8.6.2%.

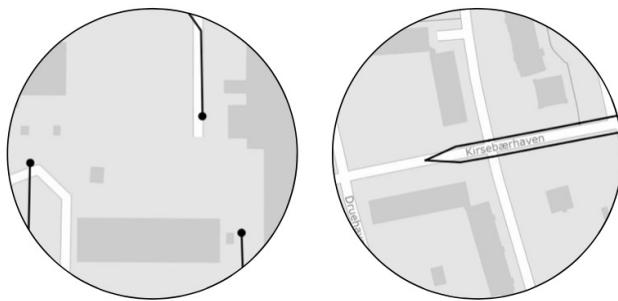
### Rosario: EMR degree distributions



### Dangling nodes

Dangling nodes are nodes of degree one, i.e. they have only one single edge attached to them. Most networks will naturally contain a number of dangling nodes. Dangling nodes can occur at actual dead-ends (representing a cul-de-sac) or at the endpoints of certain features, e.g. when a bicycle path ends in the middle of a street. However, dangling nodes can also occur as a data quality issue in case of over/undershoots (see next section). The number of dangling nodes in a network does to some extent also depend on the digitization method, as shown in the illustration below.

Therefore, the presence of dangling nodes is in itself not a sign of low data quality. However, a high number of dangling nodes in an area that is not known for containing many dead-ends can indicate digitization errors and problems with edge over/undershoots.



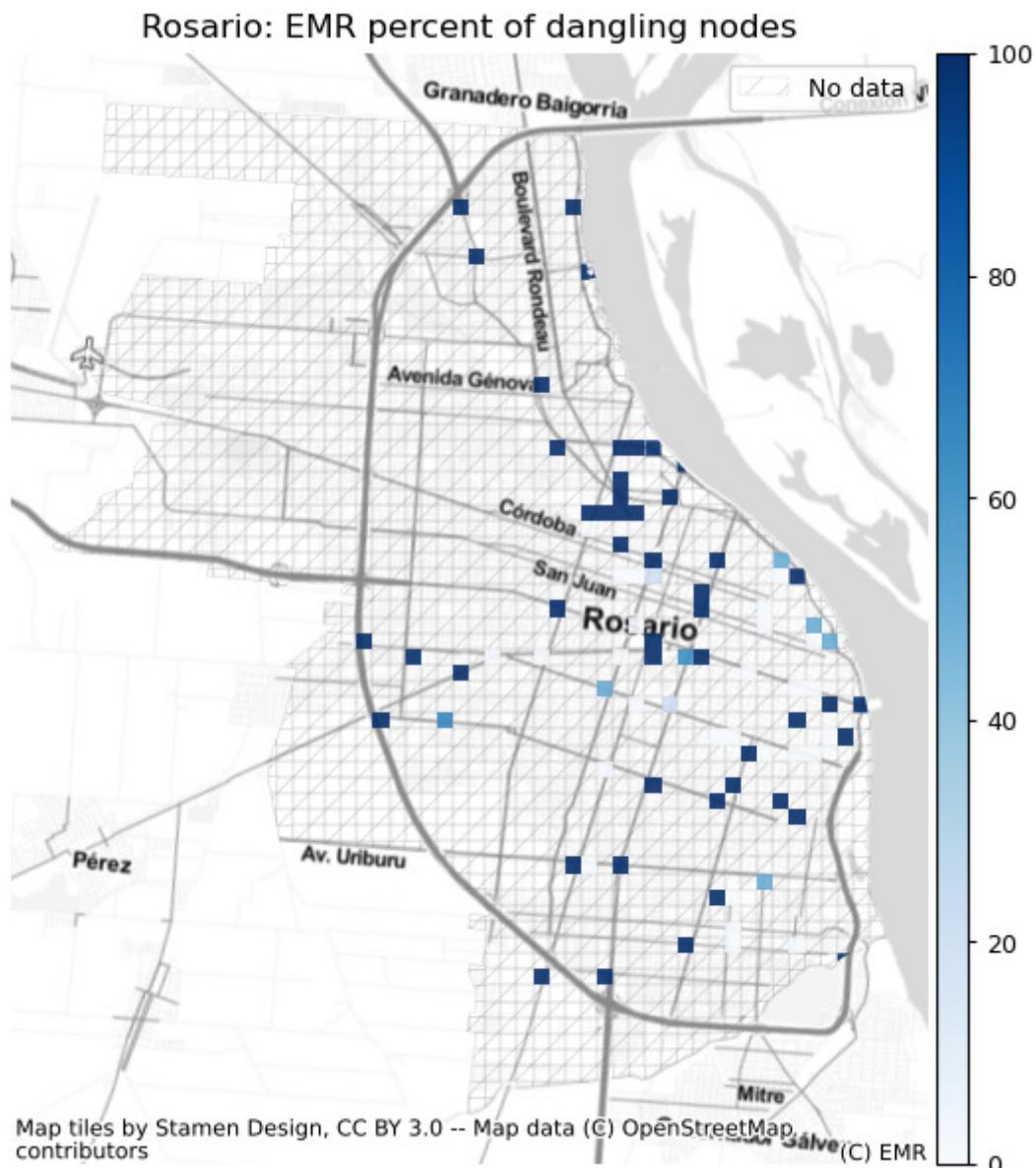
*Left: Dangling nodes occur where road features end. Right: However, when separate features are joined at the end, there will be no dangling nodes.*

### Method

Below, a list of all dangling nodes is obtained with the help of `get_dangling_nodes`. Then, the network with all its nodes is plotted. The dangling nodes are shown in color, all other nodes are shown in black.

### Interpretation

We recommend a visual analysis in order to interpret the spatial distribution of dangling nodes, with particular attention to areas of high dangling node density. It is important to understand where dangling nodes come from: are they actual dead-ends or digitization errors (e.g., over/undershoots)? A higher number of digitization errors points to lower data quality.

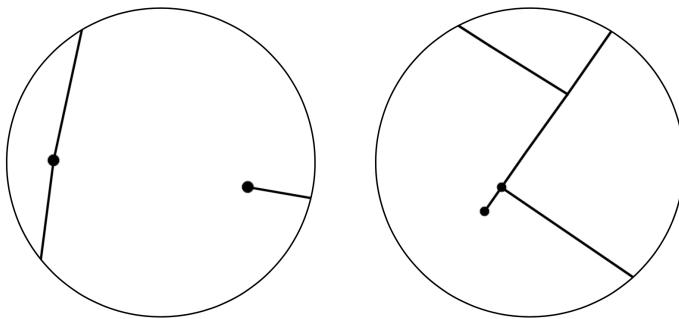


Interactive map saved at [results/REFERENCE/Rosario/maps\\_interactive/folium\\_danglingmap\\_reference.html](results/REFERENCE/Rosario/maps_interactive/folium_danglingmap_reference.html)

## Under/overshoots

When two nodes in a simplified network are placed within a distance of a few meters, but do not share a common edge, it is often due to an edge over/undershoot or another digitizing error. An undershoot occurs when two features are supposed to meet, but instead are just in close proximity to each other. An

overshoot occurs when two features meet and one of them extends beyond the other. See the image below for an illustration. For a more detailed explanation of over/undershoots, see the [GIS Lounge website](#).



*Left: Undershoots happen when two line features are not properly joined, for example at an intersection. Right: Overshoots refer to situations where a line feature extends too far beyond an intersecting line, rather than ending at the intersection.*

### Method

**Undershoots:** First, the `length_tolerance` (in meters) is defined in the cell below. Then, with `find_undershoots`, all pairs of dangling nodes that have a maximum of `length_tolerance` distance between them, are identified as undershoots, and the results are plotted.

**Overshoots:** First, the `length_tolerance` (in meters) is defined in the cell below. Then, with `find_overshoots`, all network edges that have a dangling node attached to them and that have a maximum length of `length_tolerance` are identified as overshoots, and the results are plotted.

The method for over/undershoot detection is inspired by [Neis et al. \(2012\)](#).

### Interpretation

Under/overshoots are not necessarily always a data quality issue - they might be instead an accurate representation of the network conditions or of the digitization strategy. For example, a cycle path might end abruptly soon after a turn, which results in an overshoot. Protected cycle paths are sometimes digitized as interrupted at intersections which results in intersection undershoots.

The interpretation of the impact of over/undershoots on data quality is context dependent. For certain applications, such as routing, overshoots do not present a particular challenge; they can, however, pose an issue for other applications such as network analysis, given that they skew the network structure. Undershoots, on the contrary, are a serious problem for routing applications, especially if only bicycle infrastructure is considered. They also pose a problem for network analysis, for example for any path-based metric, such as most centrality measures like betweenness centrality.

1 potential overshoots were identified with a length tolerance of 3 m.  
19 potential undershoots were identified with a length tolerance of 3 m.

Interactive map saved at `results/REFERENCE/Rosario/maps_interactive/overundershoots_3_3_reference.html`

## Network components

Disconnected components do not share any elements (nodes/edges). In other words, there is no network path that could lead from one disconnected component to the other. As mentioned above, most real-world networks of bicycle infrastructure do consist of many disconnected components ([Natera Orozco et al., 2020](#)). However, when two disconnected components are very close to each other, it might be a sign of a missing edge or another digitizing error.

### Method

First, with the help of `return_components`, a list of all (disconnected) components of the network is obtained. The total number of components is printed and all components are plotted in different colors for visual analysis. Next, the component size distribution (with components ordered by the network length they contain) is plotted, followed by a plot of the largest connected component.

### Interpretation

As with many of the previous analysis steps, knowledge of the area is crucial for a correct interpretation of component analysis. Given that the data represents the actual infrastructure accurately, bigger components indicate coherent network parts, while smaller components indicate scattered infrastructure (e.g., one single bicycle path along a street that does not connect to any other bicycle infrastructure). A high number of disconnected components in near vicinity of each other indicates digitization errors or missing data.

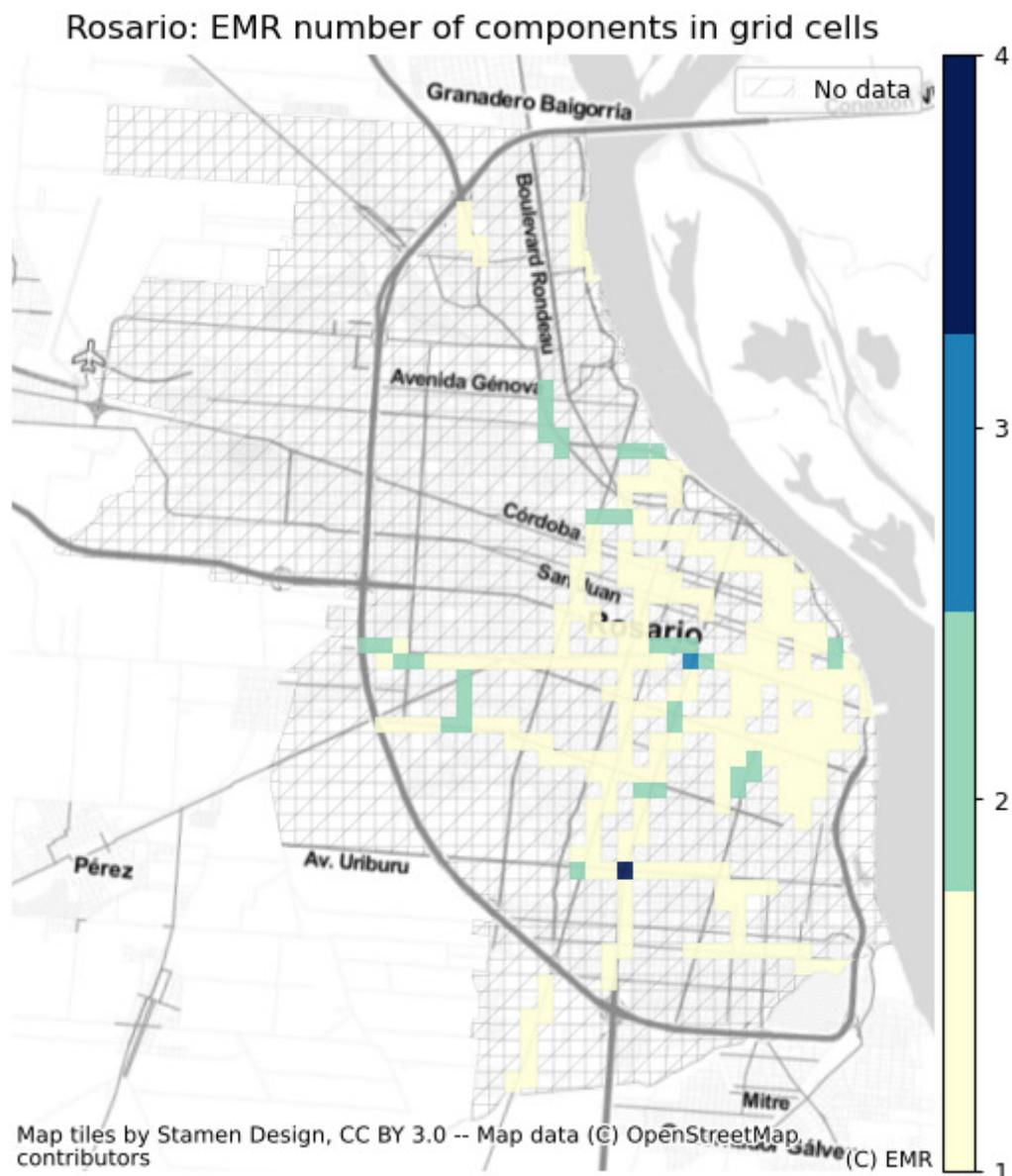
### Disconnected components

The network in the study area has 25 disconnected components.

## Rosario: EMR disconnected components



Components per grid cell



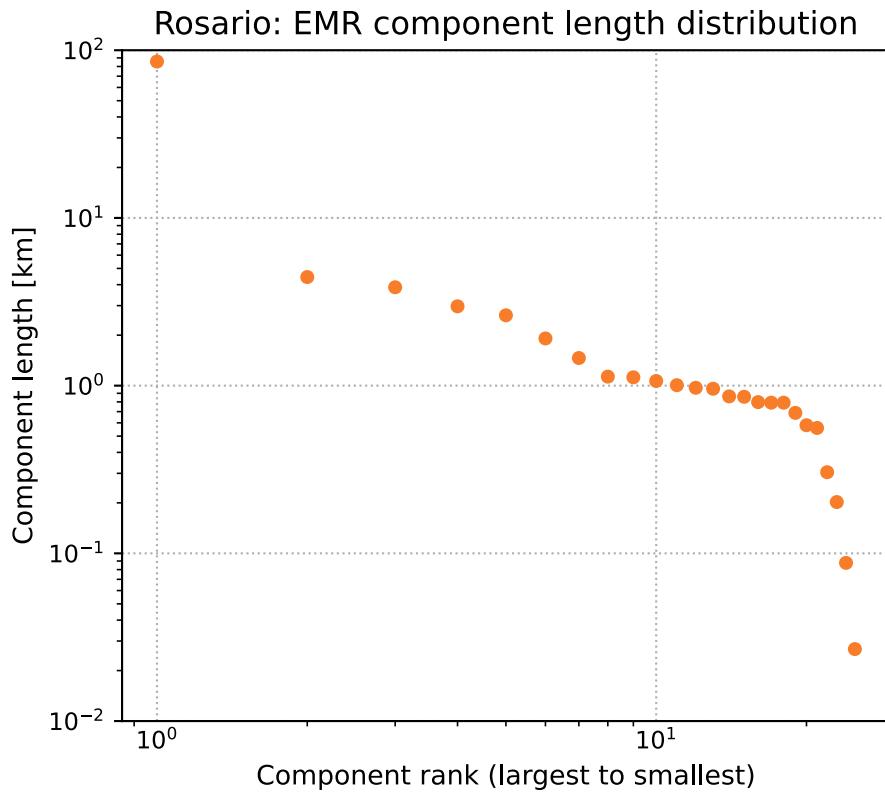
### Component length distribution

The distribution of all network component lengths can be visualized in a so-called *Zipf plot*, which orders the lengths of each component by rank, showing the largest component's length on the left, then the second largest component's length, etc., until the smallest component's length on the right. When a Zipf plot follows a straight line in [log-log scale](#), it means that there is a much higher chance to find small disconnected components than expected from traditional distributions ([Clauset et al., 2009](#)). This can mean that there has been no consolidation of the network, only piece-wise or random additions ([Szell et al., 2022](#)), or that the data itself suffers from many gaps and topology errors resulting in small disconnected components.

However, it can also happen that the largest connected component (the leftmost marker in the plot at rank  $10^0$ ) is a clear outlier, while the rest of the plot follows a different shape. This can mean that at the

infrastructure level, most of the infrastructure has been connected to one large component, and that the data reflects this - i.e. the data is not suffering from gaps and missing links to a large extent.

Bicycle networks might also be somewhere inbetween, with several large components as outliers.



The largest connected component contains 74.01% of the network length.

### Largest connected component



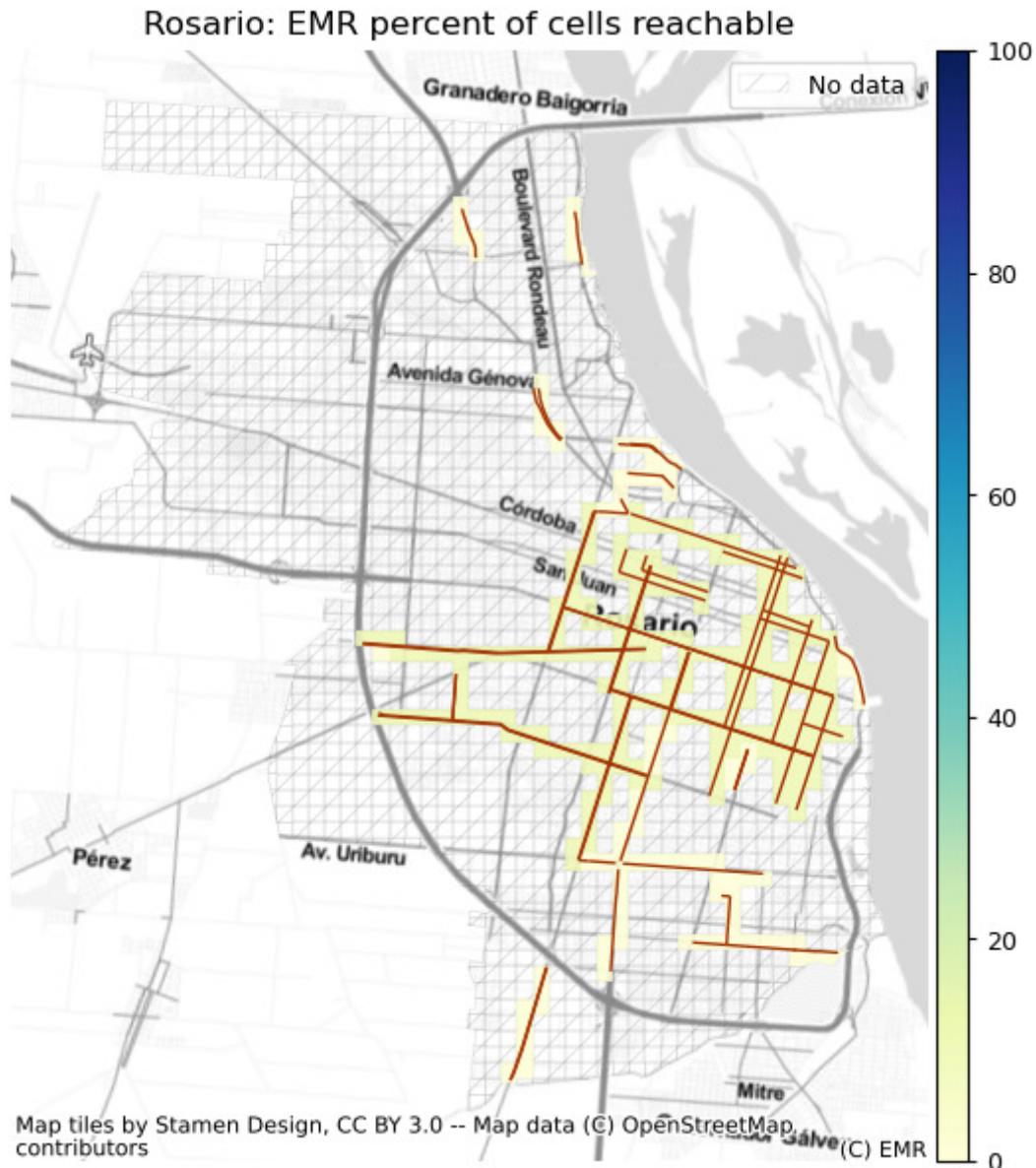
## Missing links

In the plot of potential missing links between components, all edges that are within the specified distance of an edge on another component are plotted. The gaps between disconnected edges are highlighted with a marker. The map thus highlights edges which, despite being in close proximity of each other, are disconnected and where it thus would not be possible to bike or cycle infrastructure between the edges.

Interactive map saved at results/REFERENCE/Rosario/maps\_interactive/component\_gaps\_10\_reference.html

## Component connectivity

Here we visualize differences between how many cells can be reached from each cell. This is a crude measure for network connectivity but has the benefit of being computationally cheap and thus able to quickly highlight stark differences in network connectivity.



## Summary

### Intrinsic Quality Metrics - EMR data

Total infrastructure length (km) 117

Protected bicycle infrastructure density (m/km<sup>2</sup>) 435

Unprotected bicycle infrastructure density (m/km2)	299
Mixed protection bicycle infrastructure density (m/km2)	0
Bicycle infrastructure density (m/km2)	733
Nodes	168
Dangling nodes	94
Nodes per km2	1
Dangling nodes per km2	1
Overshoots	1
Undershoots	19
Components	25
Length of largest component (km)	86
Largest component's share of network length	74%
Component gaps	18

# 3a. Extrinsic Analysis: Comparison of OSM & Reference Data

This notebook compares the provided reference bicycle infrastructure data set with OSM data in the same area in a so-called extrinsic quality assessment. To run this part of the analysis, a reference data set thus must be available for comparison.

This analysis is based on comparing the reference data set to OSM and highlighting how and where they differ, both in terms of *how much* bicycle infrastructure is mapped in the two data sets, and of *how* the infrastructure is mapped, pinpointing differences in network structure.

**All differences are computed for the reference data in relation to OSM, taking the OSM data as the base line.** For example, the difference in network density is computed by calculating reference density minus OSM density. Hence, positive difference values (over 0) indicate how much higher the reference value is; negative difference values (below 0) indicate how much lower the reference value is. Accordingly, if differences are given in percent, the OSM value is taken to be the total value (100%).

While the analysis is based on a comparison, it makes no a priori assumptions about which data set is better. The same goes for the identified differences: BikeDNA does not allow an automatic conclusion as to which data set is of better quality, but instead requires the user to interpret the meaning of the differences found, e.g., whether differing features are results of errors of omission or commission, and which data set is more correct. However, many low values can be an indication that the reference data is of lower completeness than the OSM data.

The goal is that the identified differences can be used to both assess the quality of the reference and OSM data sets, and to support the decision of which data set should be used for further analysis.

## Familiarity required

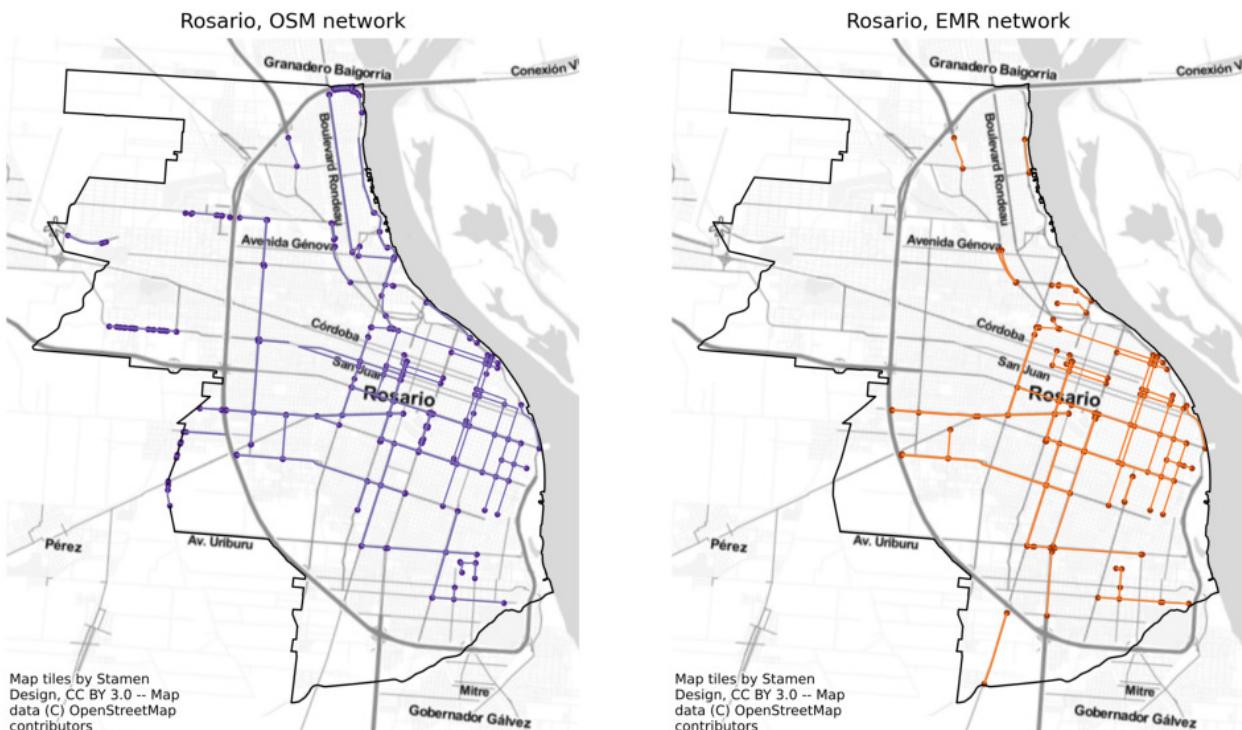
For a correct interpretation of some of the metrics for spatial data quality, some familiarity with the area is necessary.

## Sections

- Data completeness
  - Network length
  - Network density
- Network topology
  - Simplification outcomes
  - Alpha, beta, and gamma indices
  - Dangling nodes
  - Under/overshoots
- Network components
  - Disconnected components
  - Component size distribution
  - Largest connected component

- Missing links
- Components per grid cell
- Component connectivity
- Summary

### OSM versus reference network



## Data completeness

This section compares the OSM and reference data sets in terms of data completeness. The goal is to identify whether one data set has more bicycle infrastructure mapped than the other, and if so, whether those differences are concentrated in some areas.

The section starts with a comparison of the total length of the infrastructure in both data sets. Then, infrastructure, node and dangling node densities (i.e., the length of infrastructure/nodes per km<sup>2</sup>) is compared first at a global (study area) and at local (grid cell) level. Finally, density differences for protected and unprotected bicycle infrastructure are compared separately.

Computing gridded local density differences as a measure of data quality has also been applied by e.g. [Haklay \(2010\)](#).

### Method

To account for differences in how bicycle infrastructure has been mapped, the computation of network length and density is based on the infrastructure length, not the geometric length of the network edges.

For example, a 100 meter long **bidirectional** path (geometric length: 100m) contributes with 200 meters of bicycle infrastructure (infrastructure length: 200m).

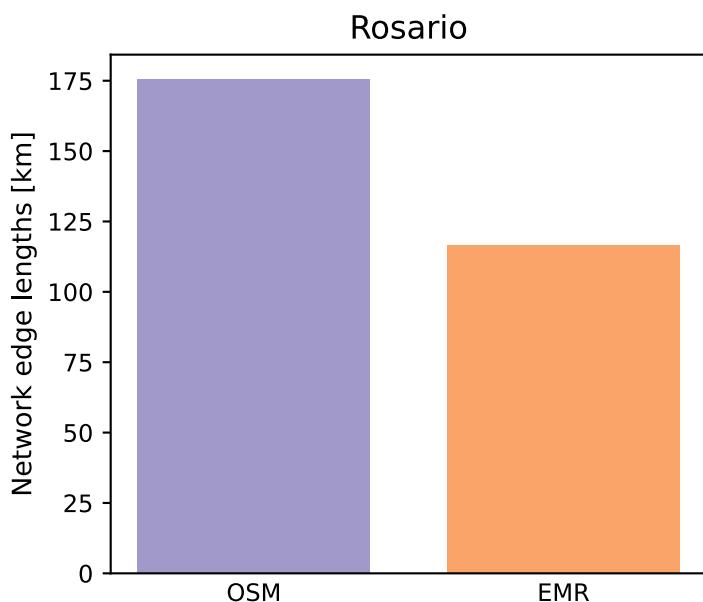
### Interpretation

Density differences can point to incomplete data. For instance, if a grid cell has a significantly higher edge density in the OSM than in the reference data set, this can indicate unmapped, missing features in the reference data set, or that a street mistakenly has been tagged as bicycle infrastructure in OSM.

### Network length

Length of the OSM data set: 175.49 km  
 Length of the reference data set: 116.71 km

The reference data set is 58.78 km shorter than the OSM data set.  
 The reference data set is 33.49% shorter than the OSM data set.



### Network Density

#### Global network densities

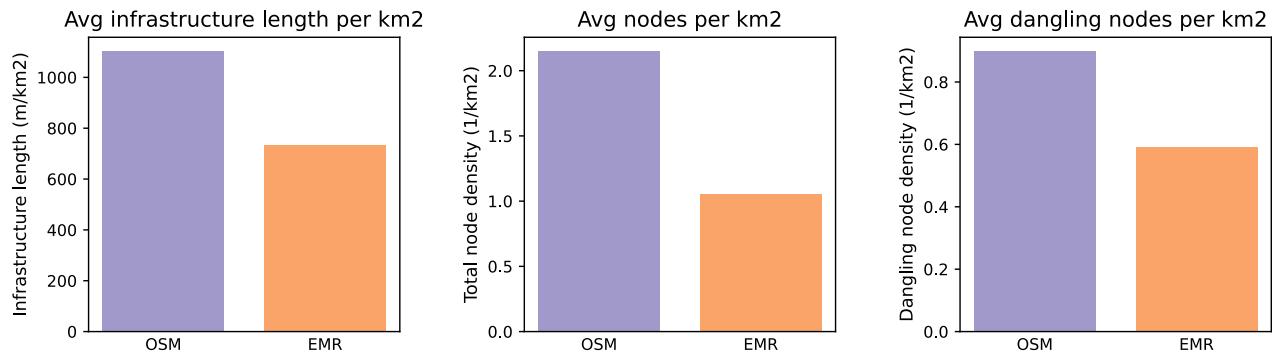
In the OSM data, there are:

- 1102.58 meters of cycling infrastructure per km<sup>2</sup>.
- 2.15 nodes in the cycling network per km<sup>2</sup>.
- 0.90 dangling nodes in the cycling network per km<sup>2</sup>.

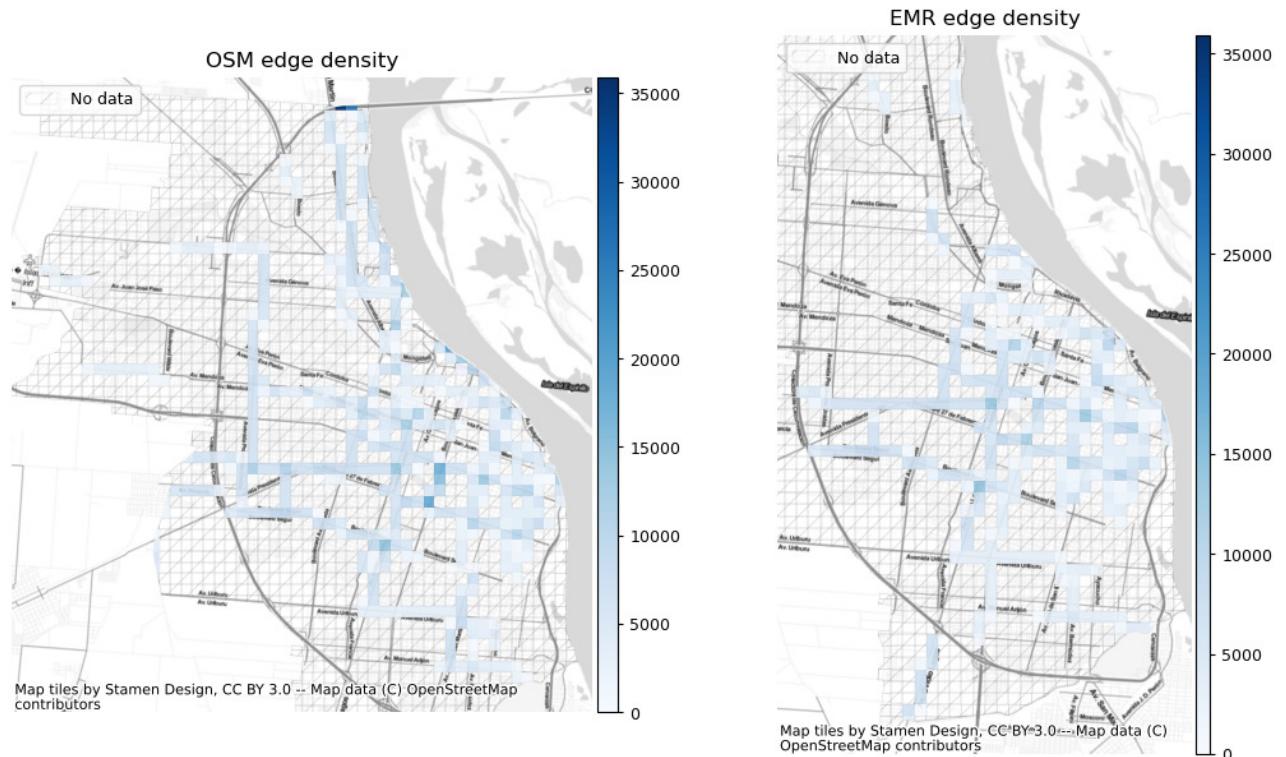
In the reference data, there are:

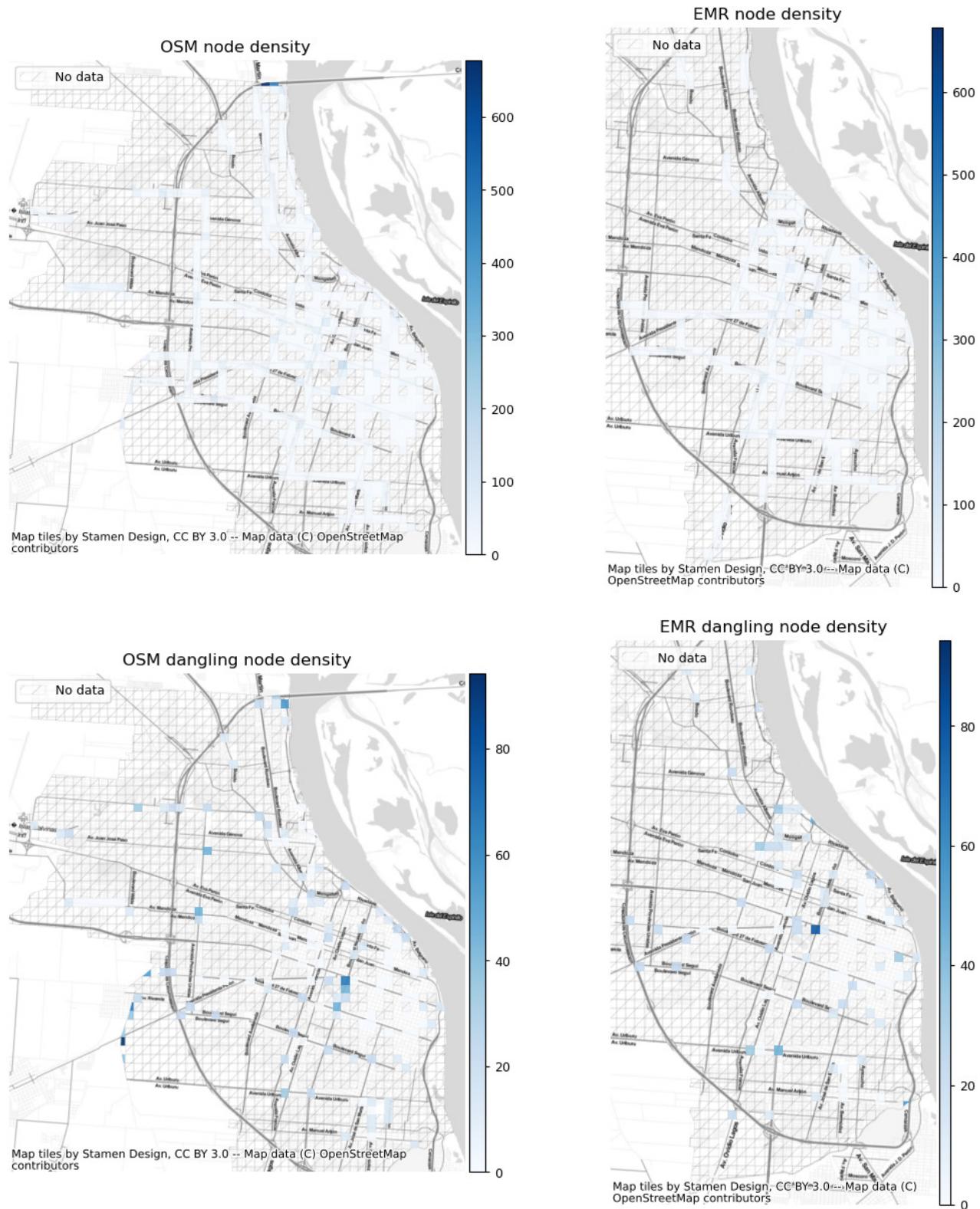
- 733.29 meters of cycling infrastructure per km<sup>2</sup>.
- 1.06 nodes in the cycling network per km<sup>2</sup>.
- 0.59 dangling nodes in the cycling network per km<sup>2</sup>.

### Global network densities (per km<sup>2</sup>)



### Local network densities



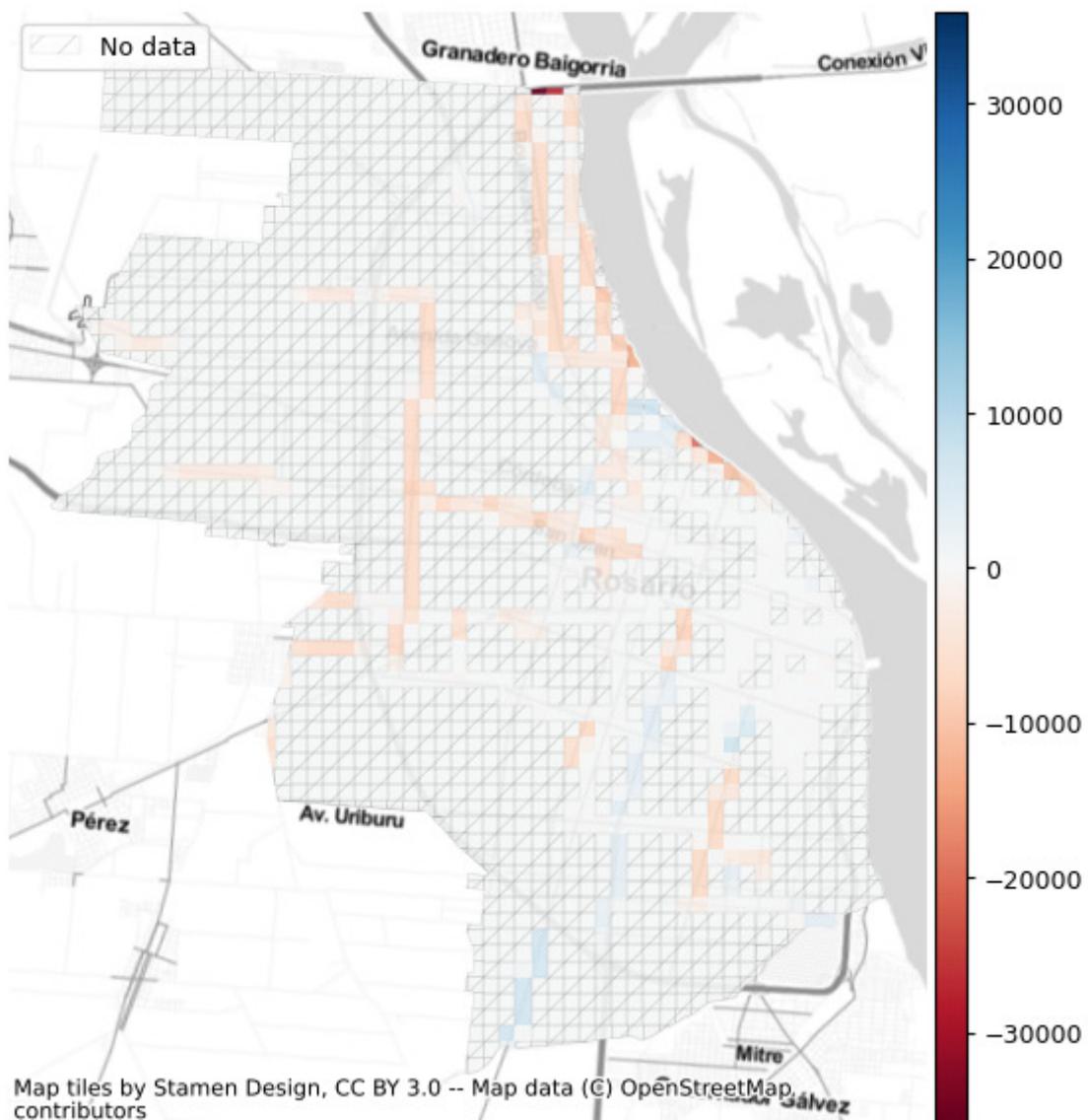


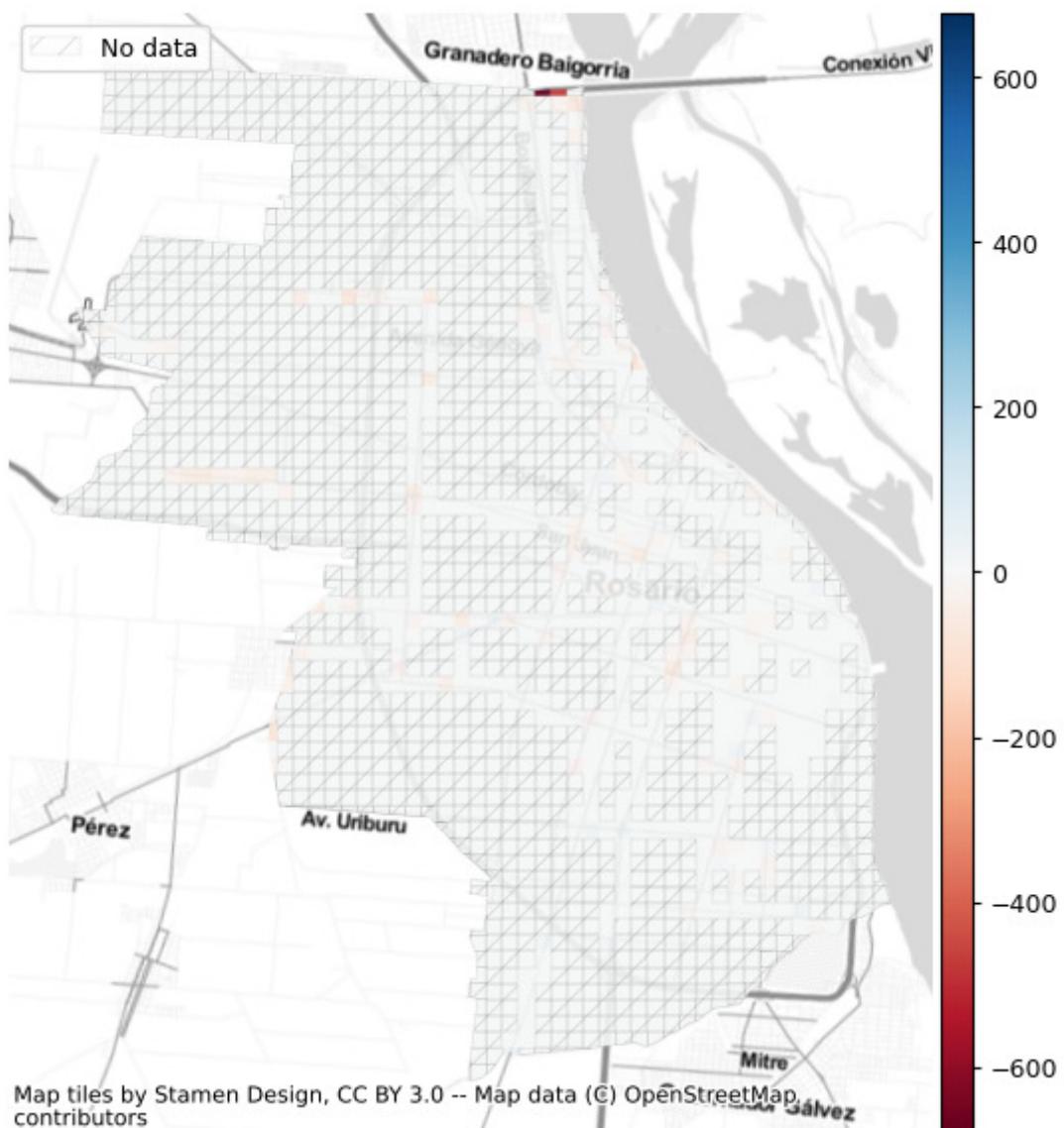
### Local differences in network densities

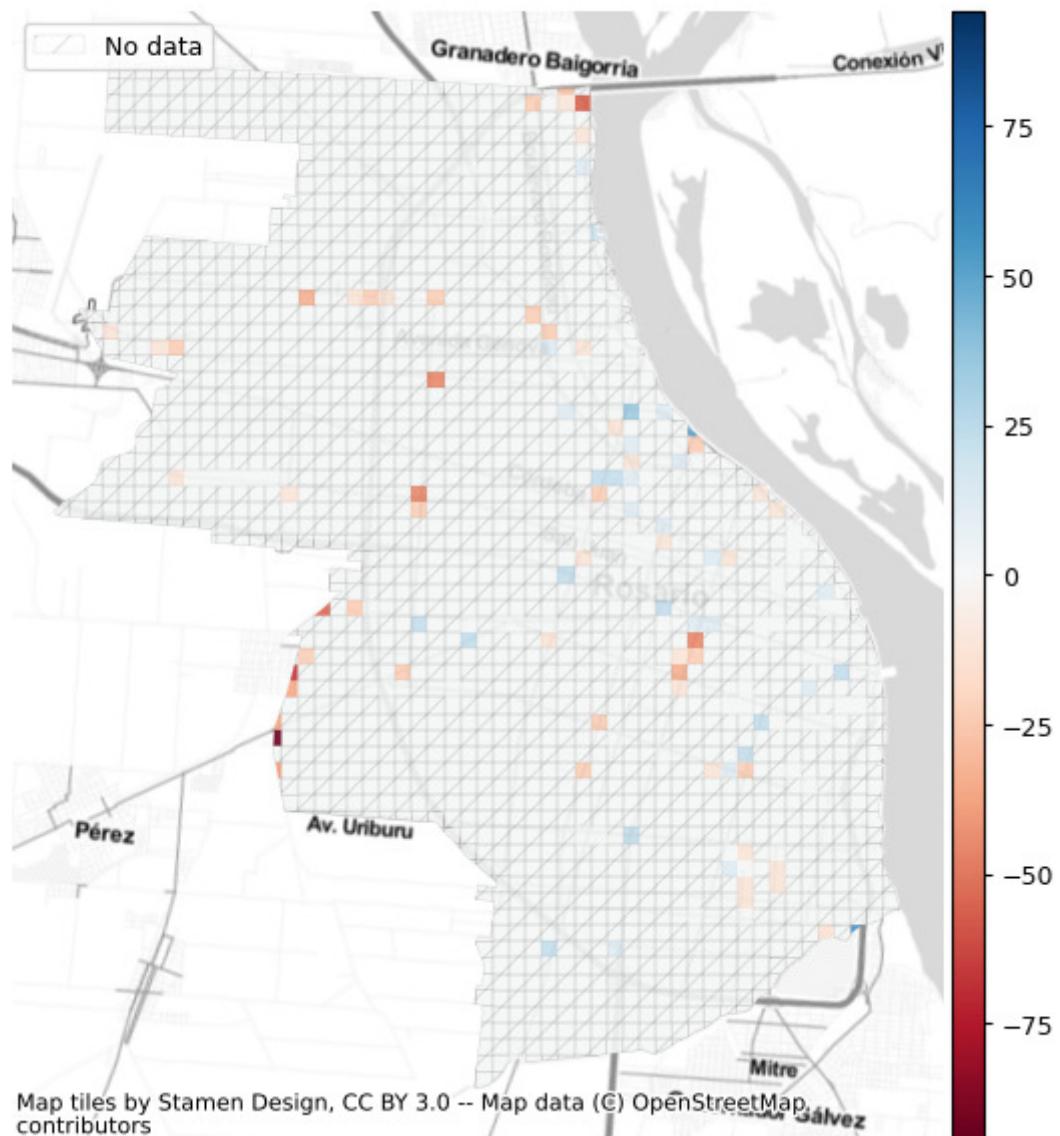
The densities in the OSM data are taken as base line for comparison, with absolute differences computed as **reference value - OSM value**. Hence, positive values indicate that the reference

density of the infrastructure type is higher than the OSM density; negative values indicate that the reference density is lower than the OSM density.

### Rosario: EMR edge density differences to OSM (m/km<sup>2</sup>)

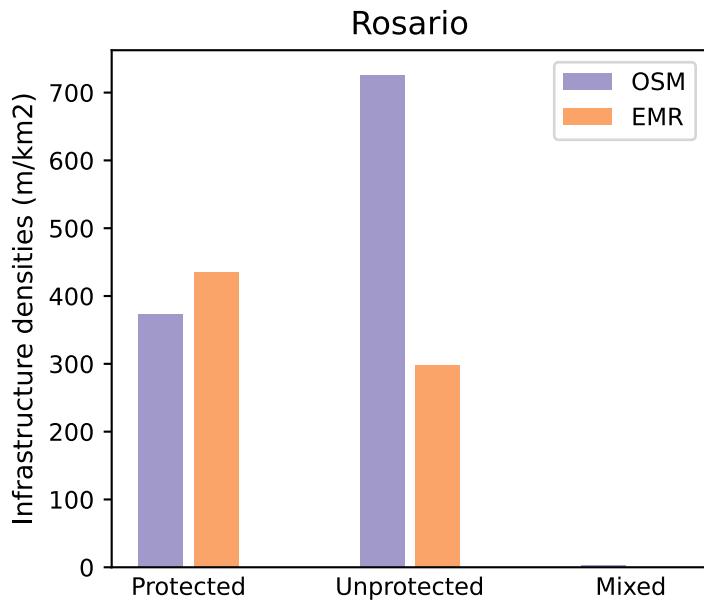


**Rosario: EMR node density differences to OSM (m/km<sup>2</sup>)**

**Rosario: EMR dangling node density differences to OSM (m/km<sup>2</sup>)**

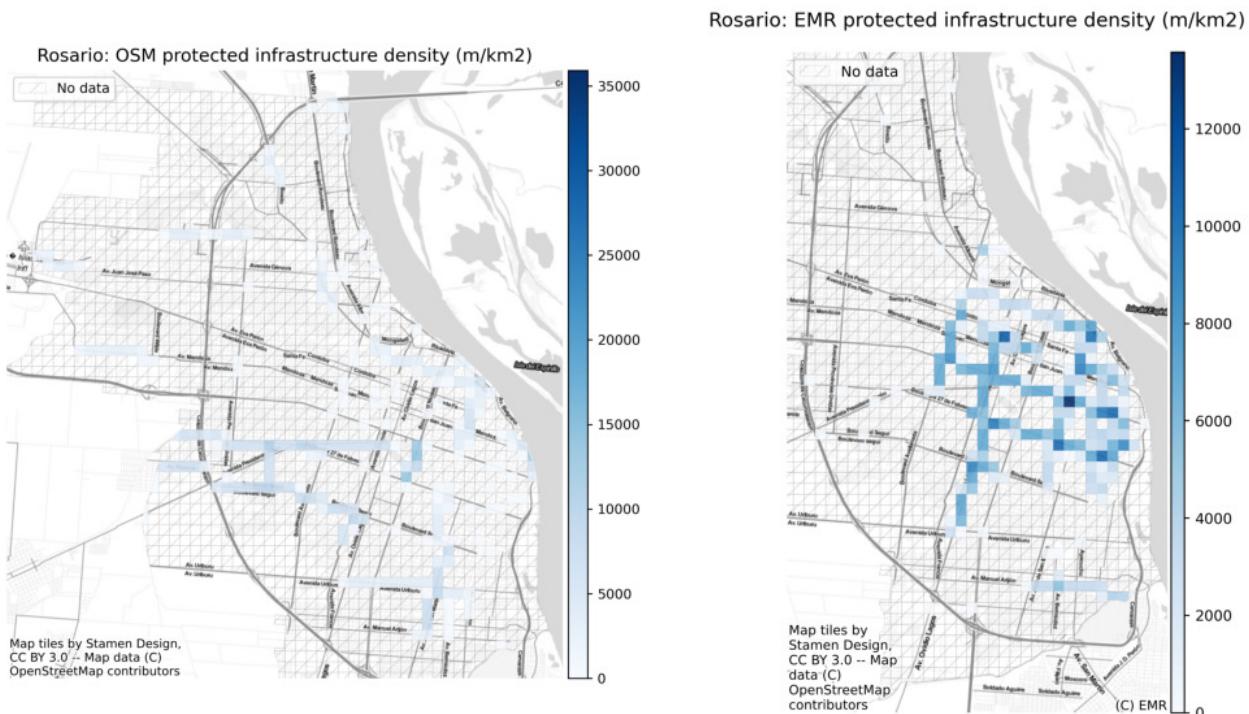
Densities of protected and unprotected bicycle infrastructure

**Global network densities for protected/unprotected infrastructure**

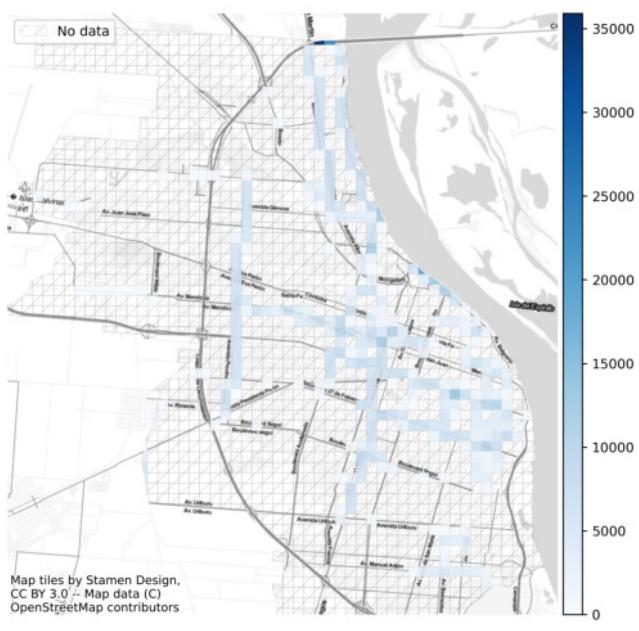


#### Local network densities for protected/unprotected infrastructure

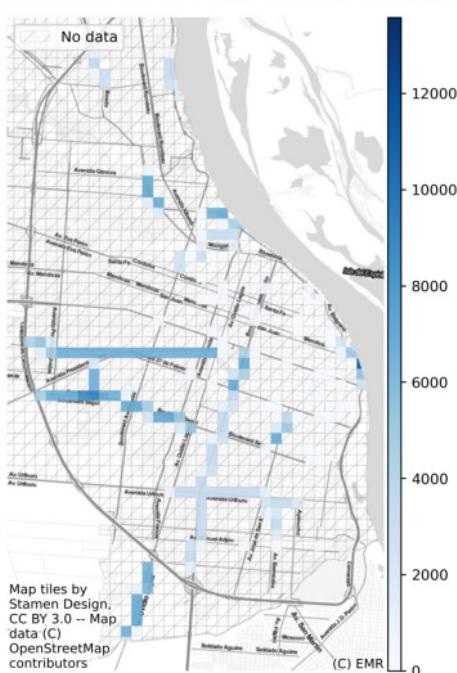
No infrastructure is mapped as mixed protected/unprotected in the EMR data.



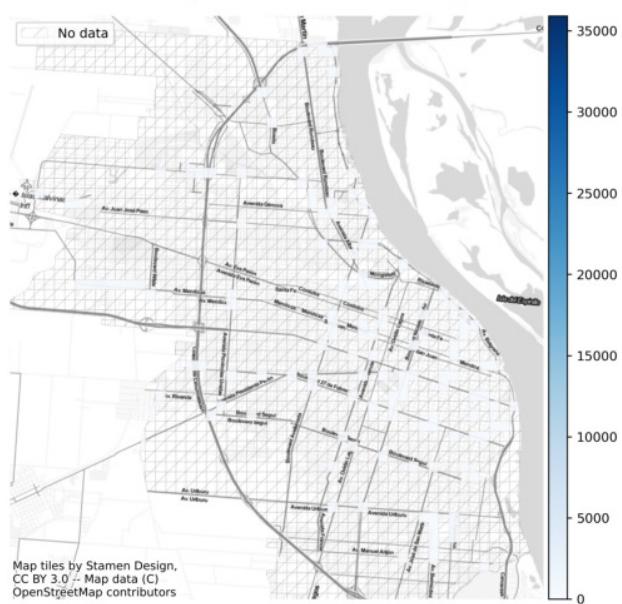
Rosario: OSM unprotected infrastructure density for (m/km<sup>2</sup>)



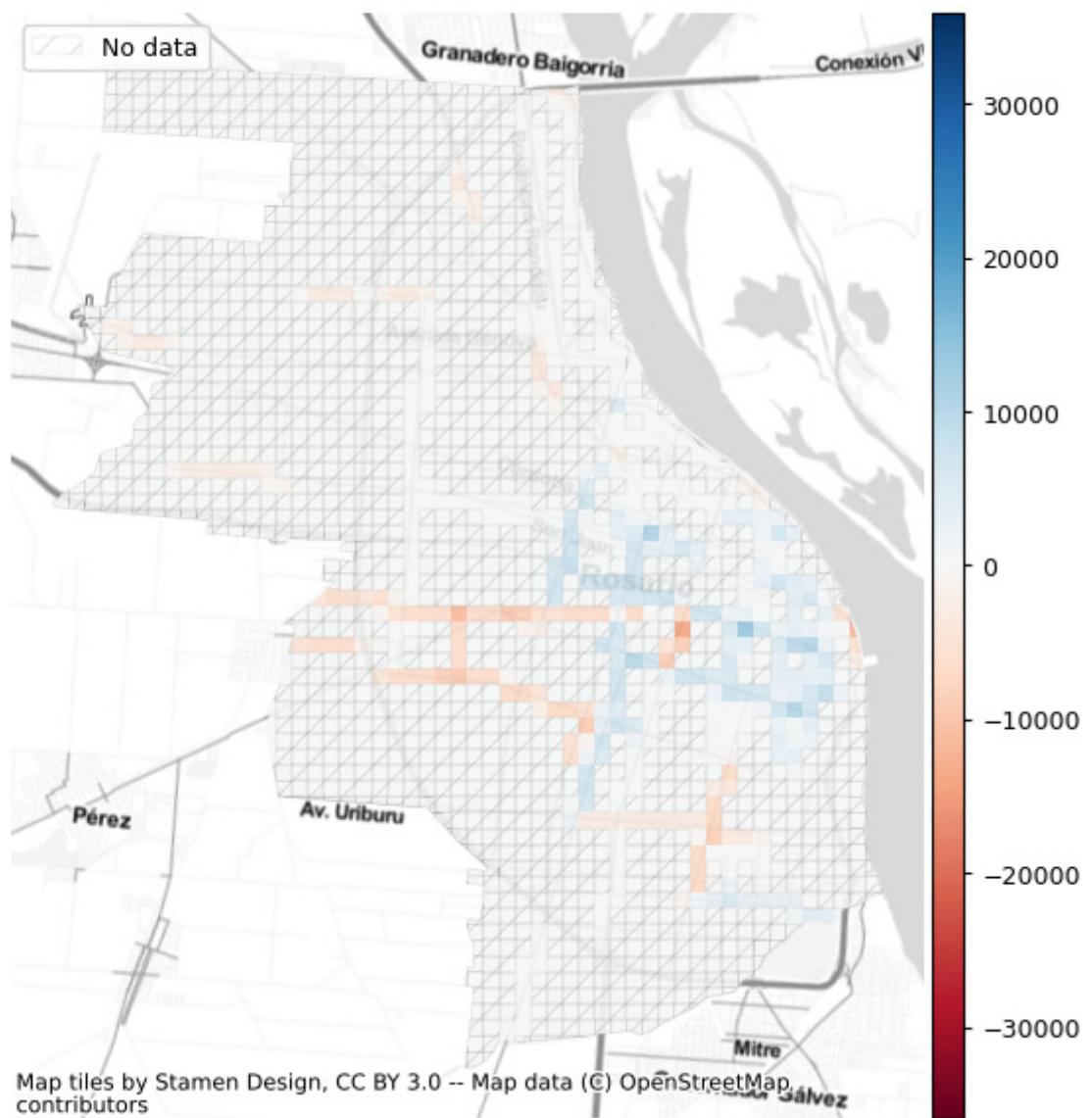
Rosario: EMR unprotected infrastructure density (m/km<sup>2</sup>)

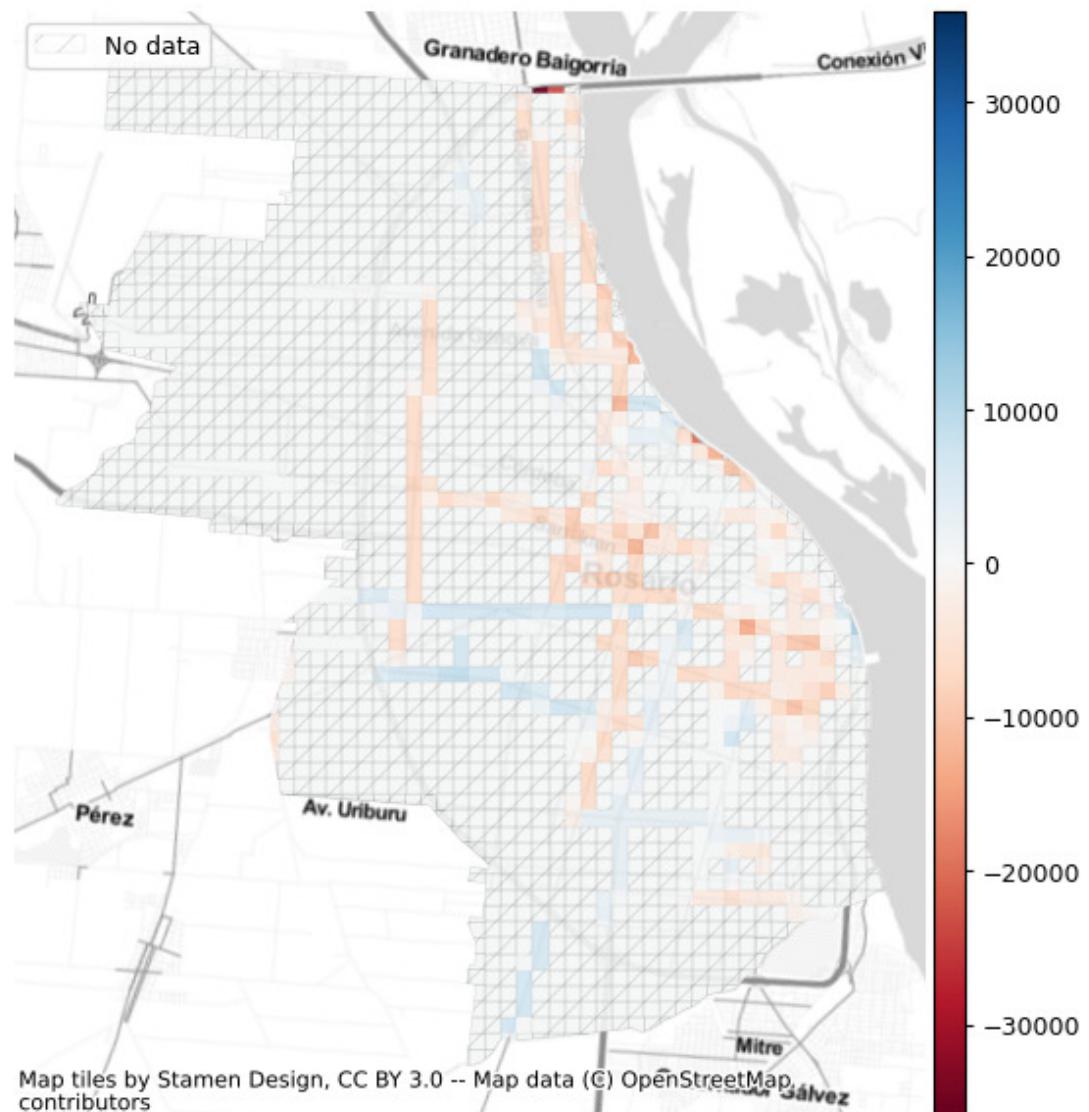


Rosario: OSM mixed protection infrastructure density (m/km<sup>2</sup>)

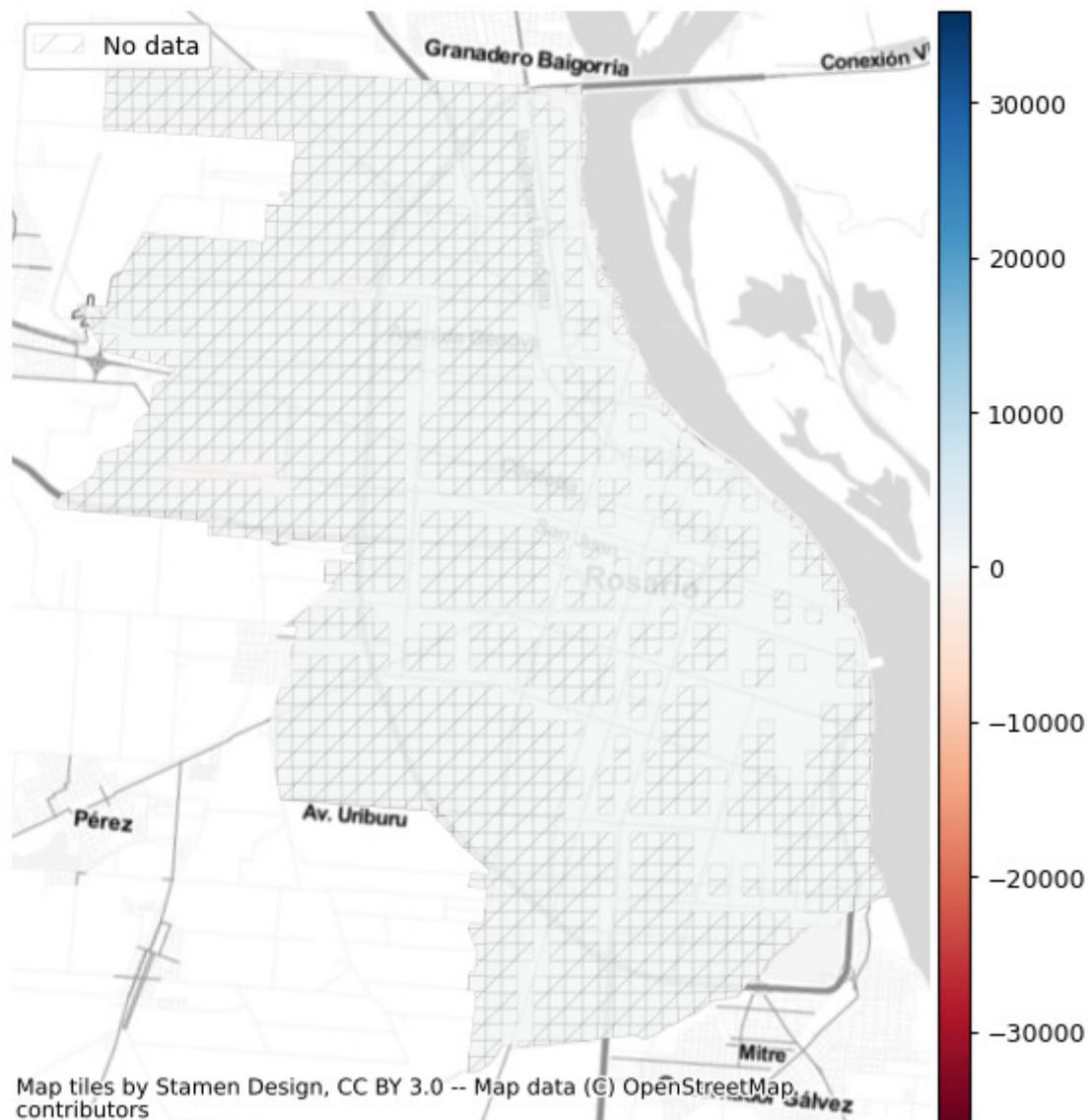


### Differences in infrastructure type density

Rosario: EMR protected infrastructure density differences to OSM (m/km<sup>2</sup>)

Rosario: EMR unprotected infrastructure density differences to OSM (1/km<sup>2</sup>)

### Rosario: EMR mixed infrastructure density differences to OSM (1/km<sup>2</sup>)



## Network topology

After having compared data completeness, i.e. *how much* infrastructure is mapped, here we focus on differences in network topology, which gives information about *how* the infrastructure is mapped in both data sets. Here we also analyze the extent to which network edges are connected to one or more other edges, or if they end in a dangling node. The extent to which edges are properly connected to adjacent edges are important for, for example, analyzes of accessibility and routing.

When working with data on bicycle networks, a data set without gaps between actually connected network elements is preferred - while of course reflecting the real conditions. Identifying the dangling nodes in a network is a quick and easy way to identify edges that end in a 'dead end'. Under- and overshoots offer a more precise picture of respectively network gaps and overextended edges, that give a misleading count of dangling nodes.

## Method

To identify potential gaps or missing links in the data, first the dangling nodes in both data sets are plotted. Then, the local percentage of dangling nodes out of all nodes in each data set is plotted separately. Finally, we show the local difference in the percent of dangling nodes.

Under and overshoots in both OSM and reference data are finally plotted together in an interactive plot for further inspection.

## Interpretation

If an edge ends in a dangling node in one data set but not the other, this indicates a problem with the data quality. There either is a missing connection in the data, or two edges have been connected erroneously. Similarly, different local rates in the share of dangling nodes indicates differences in how the bicycle networks have been mapped - although differences in data completeness of course should be considered in the interpretation.

Undershoots are clear indications of misleading gaps in network data - although they might also represent actual gaps in bicycle infrastructure. Comparing undershoots in one data set with another data set can help identify whether it is a question of data quality or the quality of the actual infrastructure. Systematic differences in the presence of undershoots or gaps across intersections might be an indication in differing digitizing strategies, since some approaches will map a bike lane crossing a street as a connected stretch, while others will introduce a gap in the width of the crossing street. While both approaches are valid, data sets created with the former method are more suited for routing-based analysis.

Overshoots will often be less consequential for analysis, but a high number of overshoots will introduce false dangling nodes and distort measures for network structure based on e.g., node degree or the ratio between nodes and edges.

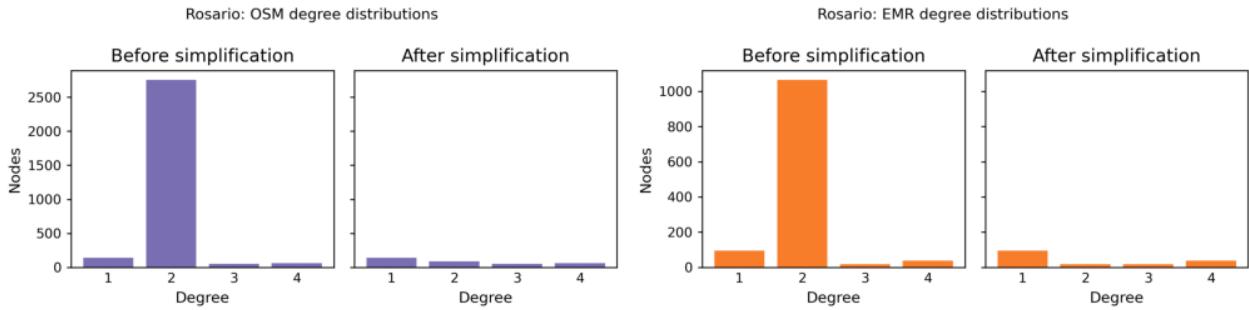
## Simplification outcomes

Simplifying the OSM network decreased the number of edges by 89.9%.  
Simplifying the OSM network decreased the number of nodes by 88.6%.

Simplifying the EMR network decreased the number of edges by 86.2%.  
Simplifying the EMR network decreased the number of nodes by 86.2%.

## Node degree distribution

Note that the two figures below have different y-axis scales.



## Alpha, beta, and gamma indices

In this subsection, we compute and contrast the three aggregated network metrics alpha, beta, and gamma. These metrics are often used to describe network structure, but as measures of data quality, they are only meaningful when compared to the values of a corresponding data set. For this reason, alpha, beta, and gamma are only part of the extrinsic analysis and not included in the intrinsic notebooks.

While no conclusion can be drawn about data quality based on any of the three metrics by itself, a comparison of the metrics for the two data sets can indicate differences in network topology, and hence differences in how the infrastructure has been mapped.

### Method

All three indices are computed with `eval_func.compute_alpha_beta_gamma`.

The **alpha** value is the ratio of actual to possible cycles in the network. A network cycle is defined as a closed loop – i.e. a path that ends on the same node that it started from. The value of alpha ranges from 0 to 1. An alpha value of 0 means that the network has no cycles at all, i.e. it is a tree. An alpha value of 1 means that the network is fully connected, which is very rarely the case.

The **beta** value is the ratio of existing edges to existing nodes in the network. The value of beta ranges from 0 to  $N-1$ , where  $N$  is the number of existing nodes. A beta value of 0 means that the network has no edges; a beta value of  $N-1$  means that the network is fully connected (see also gamma value of 1). The higher the beta value, the more different paths (on average) can be chosen between any pair of nodes.

The **gamma** value is the ratio of existing to *possible* edges in the network. Any edge that connects two of the existing network nodes is defined as "possible". Hence, the value of gamma ranges from 0 to 1. A gamma value of 0 means that the network has no edges; a gamma value of 1 means that every node of the network is connected to every other node.

For all three indices, see [Ducruet and Rodrigue, 2020](#). All three indices can be interpreted in respect to network connectivity: The higher the alpha value, the more cycles are present in the network; the higher the beta value, the higher the number of paths and thus the higher the complexity of the network; and the higher the gamma value, the fewer edges lie between any pair of nodes.

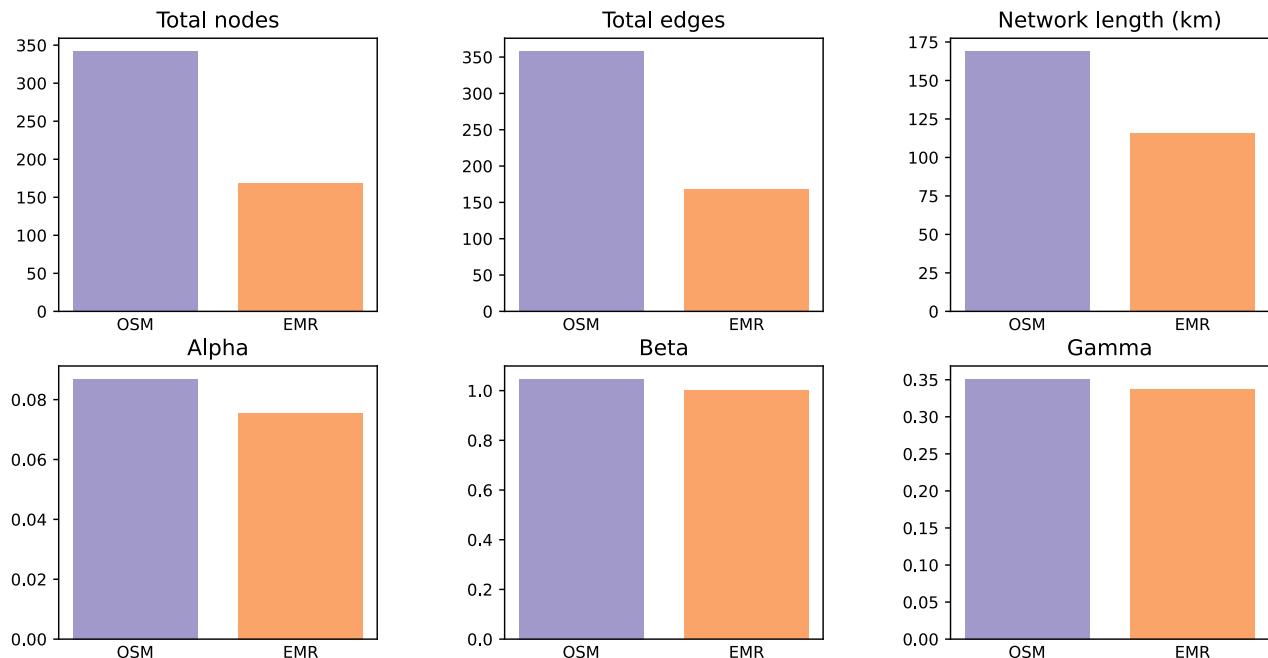
### Interpretation

These metrics do not say much about the data quality itself, nor are they useful for a topological comparison of networks of similar size. However, some conclusions can be drawn through a comparison. For example, if the indices are very similar for the two networks, despite the networks e.g. having very

different geometric lengths, this suggests that the data sets have been mapped in roughly the same way, but that one simply includes more features than the other. However, if the networks have roughly the same total geometric length, but the values from alpha, beta and gamma differ, this can be an indication that the structure and topology of the two data sets are fundamentally different.

Alpha for the simplified OSM network: 0.09  
 Beta for the simplified OSM network: 1.05  
 Gamma for the simplified OSM network: 0.35

Alpha for the simplified EMR network: 0.08  
 Beta for the simplified EMR network: 1.00  
 Gamma for the simplified EMR network: 0.34



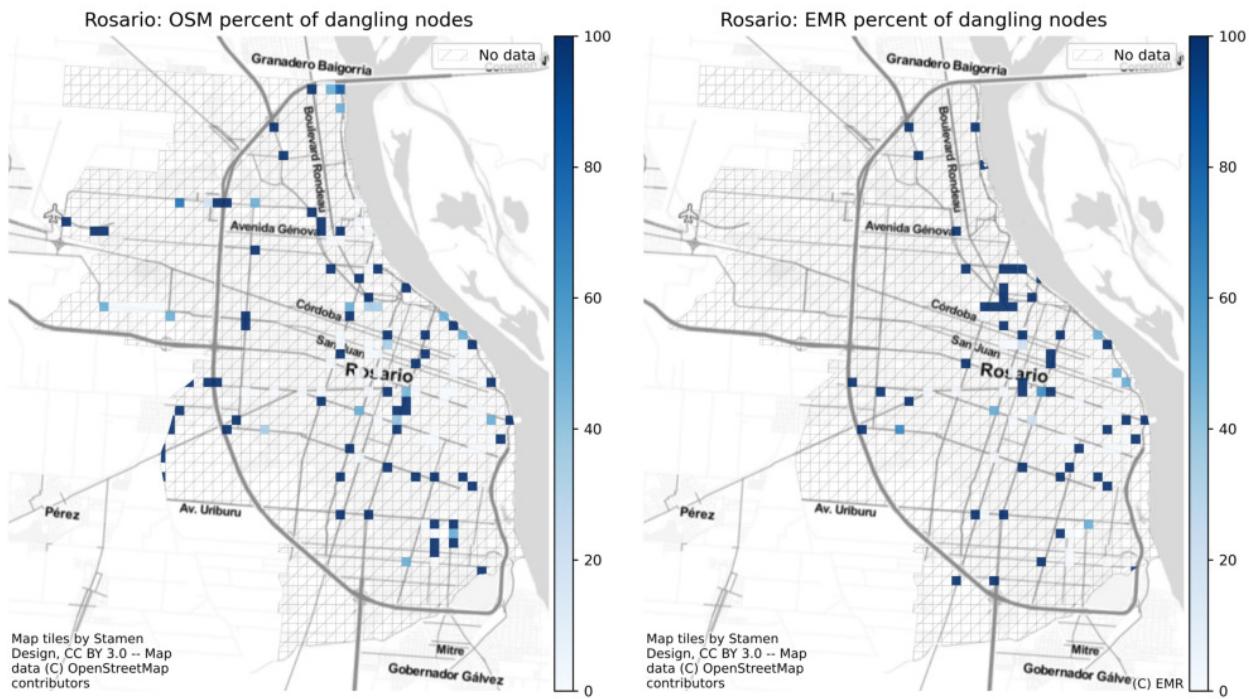
## Dangling nodes

### Dangling nodes in OSM & reference networks

Interactive map saved at results/COMPARE/Rosario/maps\_interactive/danglingmap\_compare.html

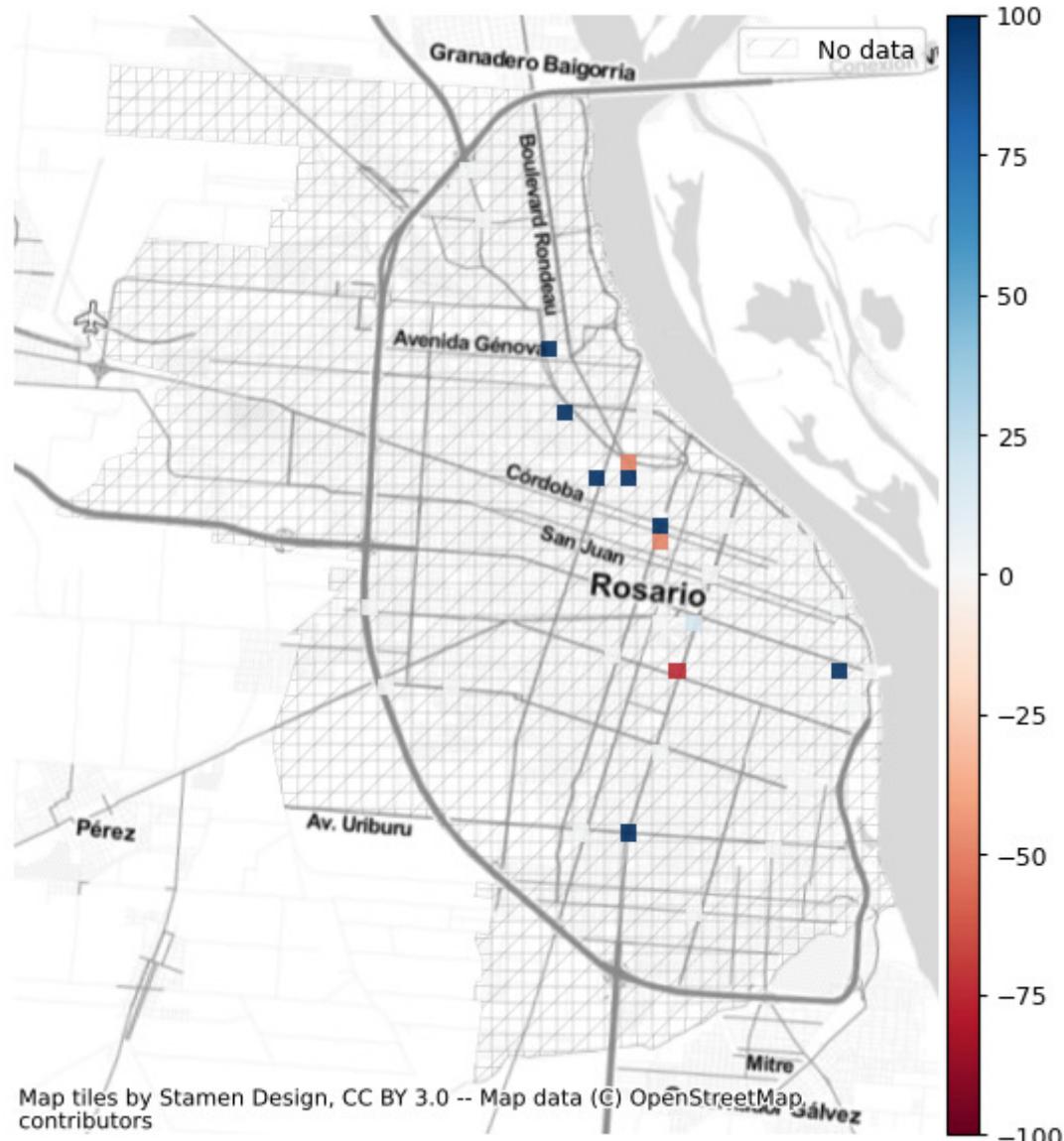
### Local values for dangling nodes

### Dangling nodes as percentage of all nodes



#### Local differences in dangling nodes percentages

### Rosario: EMR percent difference to OSM in dangling nodes



### Under/overshoots

#### Over and undershoots in OSM and reference networks

Interactive map saved at [results/COMPARE/Rosario/maps\\_interactive/overundershoots\\_3\\_3\\_compare.html](results/COMPARE/Rosario/maps_interactive/overundershoots_3_3_compare.html)

### Network components

This section takes a close look at the network component characteristics for the two data sets.

Disconnected components do not share any elements (nodes/edges). In other words, there is no network path that could lead from one disconnected component to the other. As mentioned above, most

real-world networks of bicycle infrastructure do consist of many disconnected components ([Natera Orozco et al., 2020](#)). However, when two disconnected components are very close to each other, it might be a sign of a missing edge or another digitizing error.

## Method

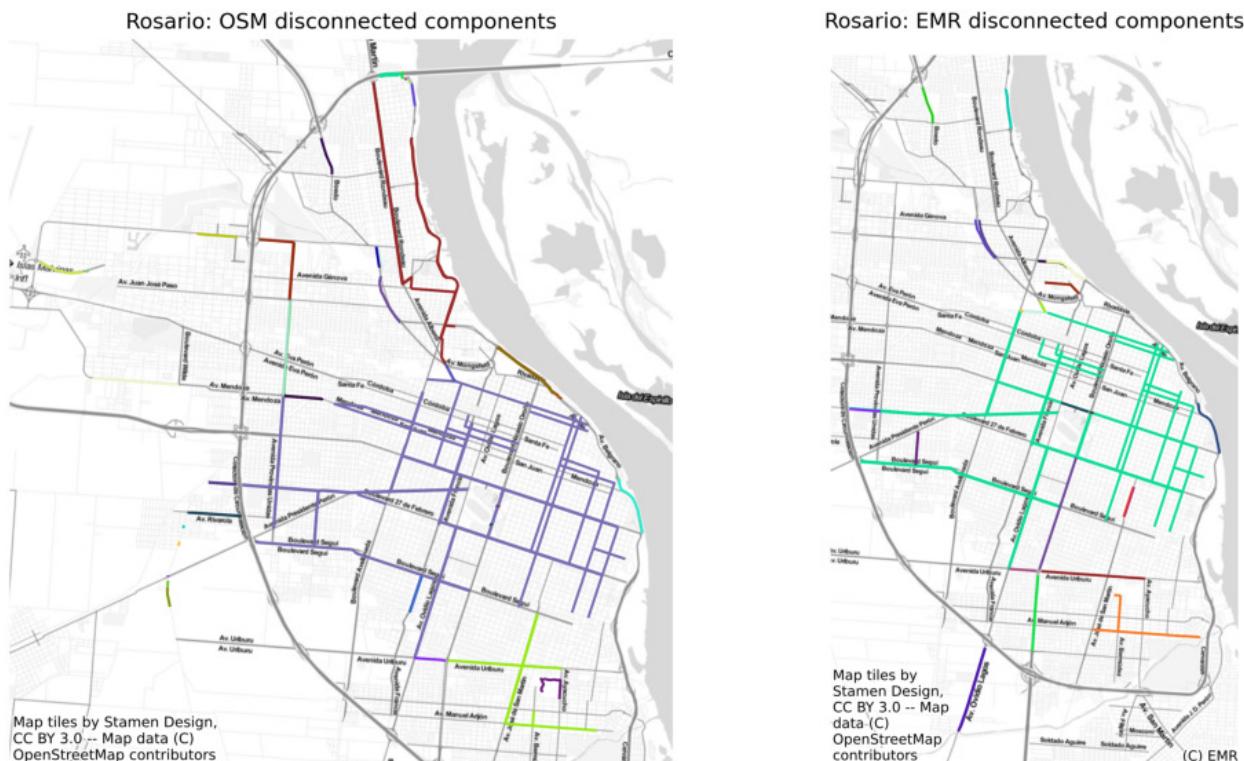
To compare the number and pattern of disconnected components in OSM and reference data, all component results from the intrinsic analyses are juxtaposed and two new plots showing respectively components gaps for OSM and reference data and the difference in component connectivity are produced.

## Interpretation

The fragmented nature of many bicycle networks make it hard to assess whether disconnected components are a question of a lack of data quality or a lack of properly connected bicycle infrastructure. Comparing disconnected components in two data sets enables a more accurate assessment of whether a disconnected component is a data or a planning issue.

## Disconnected components

The OSM network in the study area consists of 43 disconnected components. The EMR network in the study area consists of 25 disconnected components.

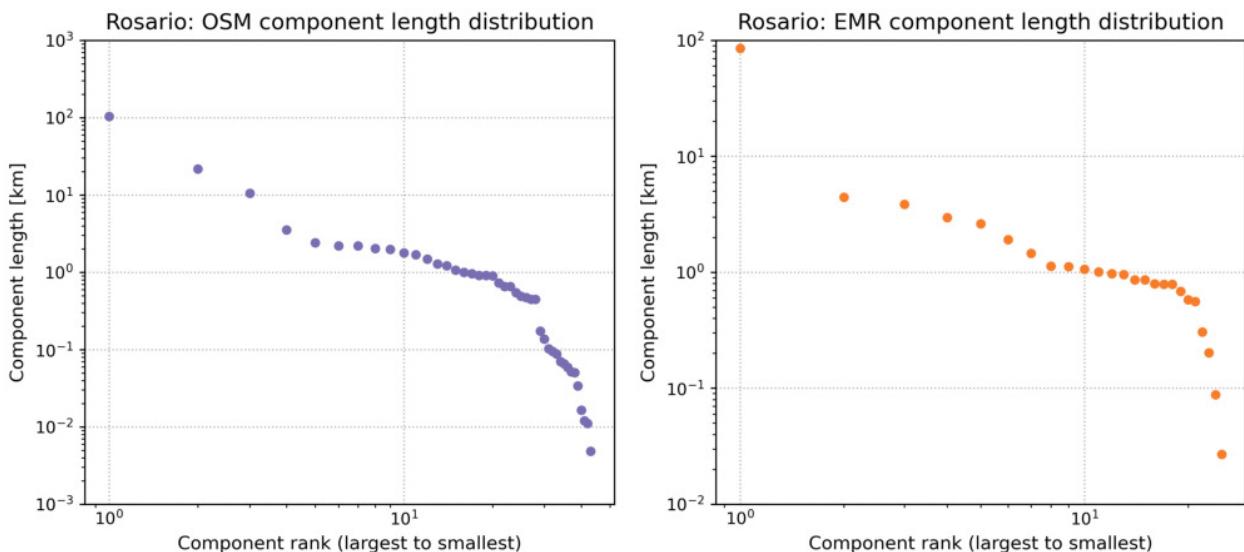


## Component length distribution

The distribution of all network component lengths can be visualized in a so-called *Zipf plot*, which orders the lengths of each component by rank, showing the largest component's length on the left, then the second largest component's length, etc., until the smallest component's length on the right. When a Zipf plot follows a straight line in [log-log scale](#), it means that there is a much higher chance to find small disconnected components than expected from traditional distributions ([Clauset et al., 2009](#)). This can mean that there has been no consolidation of the network, only piece-wise or random additions ([Szell et al., 2022](#)), or that the data itself suffers from many gaps and topology errors resulting in small disconnected components.

However, it can also happen that the largest connected component (the leftmost marker in the plot at rank  $10^0$ ) is a clear outlier, while the rest of the plot follows a different shape. This can mean that at the infrastructure level, most of the infrastructure has been connected to one large component, and that the data reflects this - i.e. the data is not suffering from gaps and missing links to a large extent. Bicycle networks might also be somewhere inbetween, with several large components as outliers.

In case of a comparison over the same region, as shown below, if one data set shows a clear outlier in its largest connected component while the other does not, and if it is also at least as large, it can in general be interpreted as being more complete.

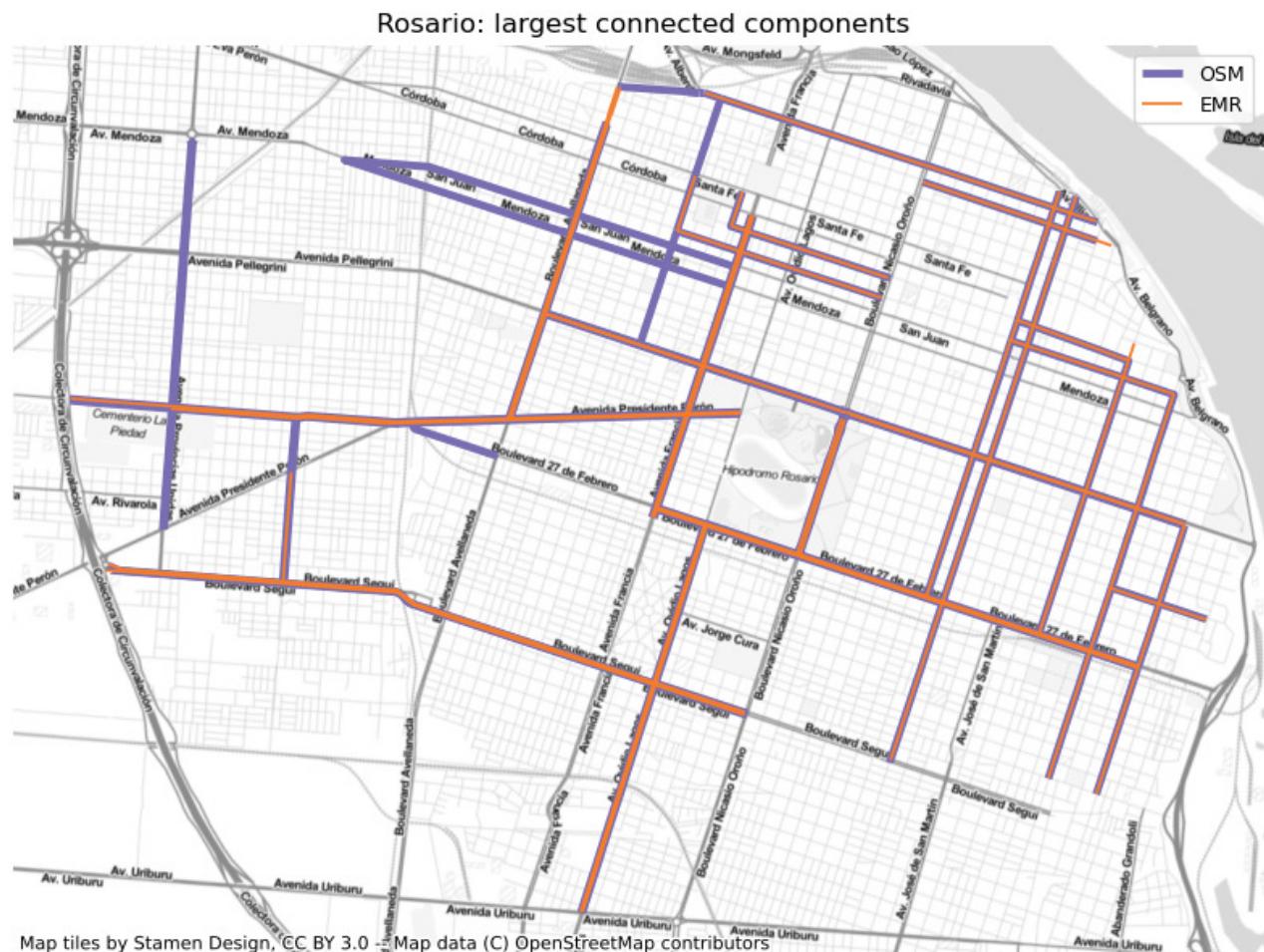


## Largest connected component

The largest connected component in the OSM network contains 61.48% of the network length. The largest connected component in the EMR network contains 74.01% of the network length.



#### Overlay of largest connected component in OSM and reference networks



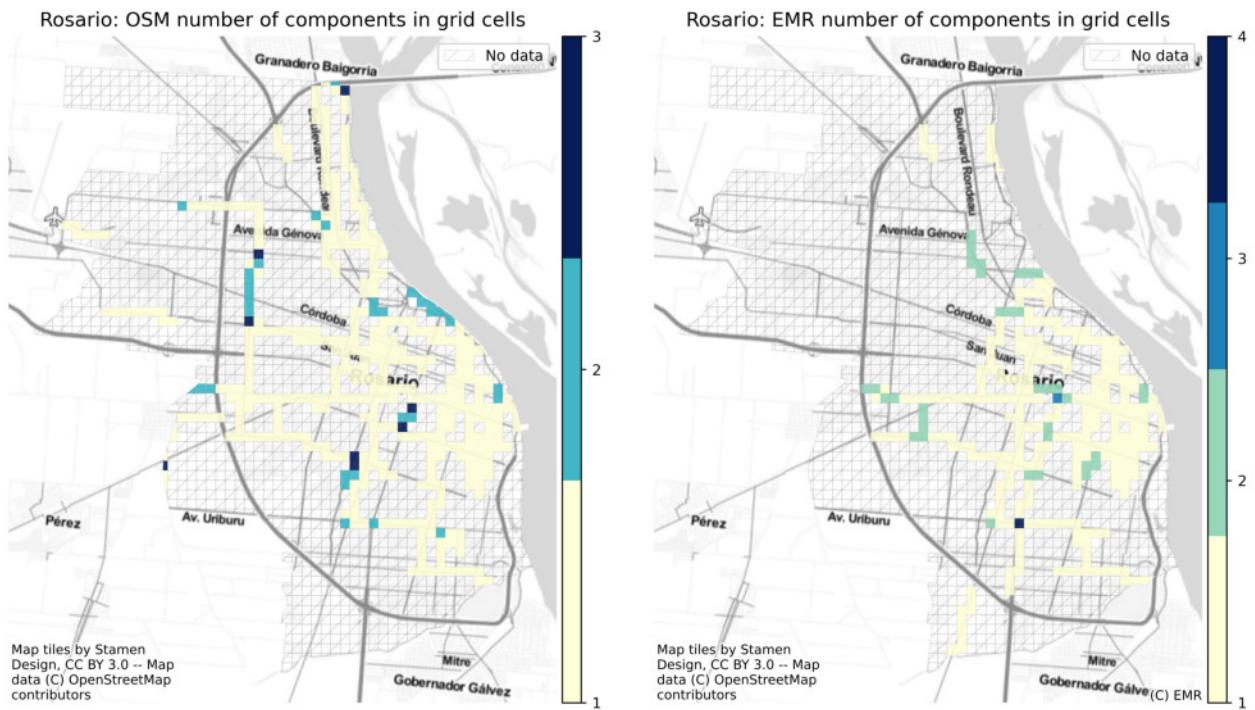
## Missing links

In the plot of potential missing links between components, all edges that are within the specified distance of an edge on another component are plotted. The gaps between disconnected edges are highlighted with a marker. The map thus highlights edges which, despite being in close proximity of each other, are disconnected and where it thus would not be possible to bike on cycling infrastructure between the edges.

Interactive map saved at [results/COMPARE/Rosario/maps\\_interactive/component\\_gaps\\_compare.html](results/COMPARE/Rosario/maps_interactive/component_gaps_compare.html)

## Components per grid cell

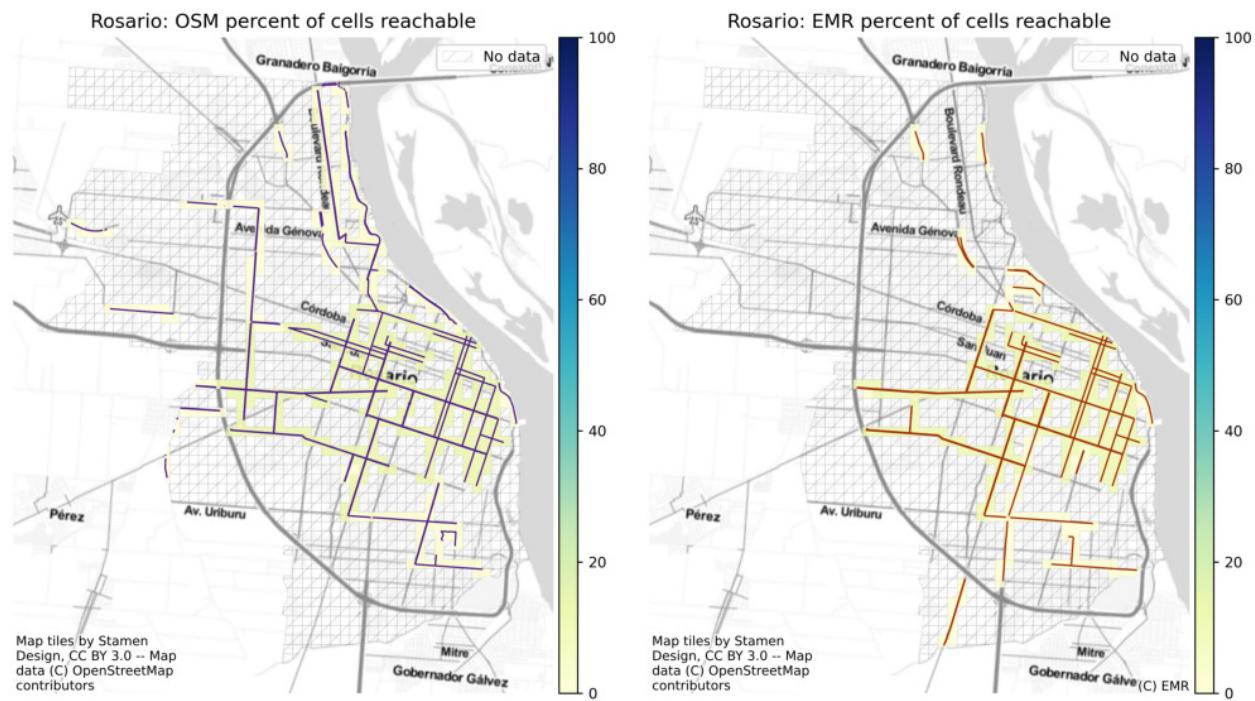
The plots below show the number of components intersecting a grid cell. A high number of components in a grid cell is generally an indication of poor network connectivity - either due to fragmented infrastructure or because of problems with the data quality.



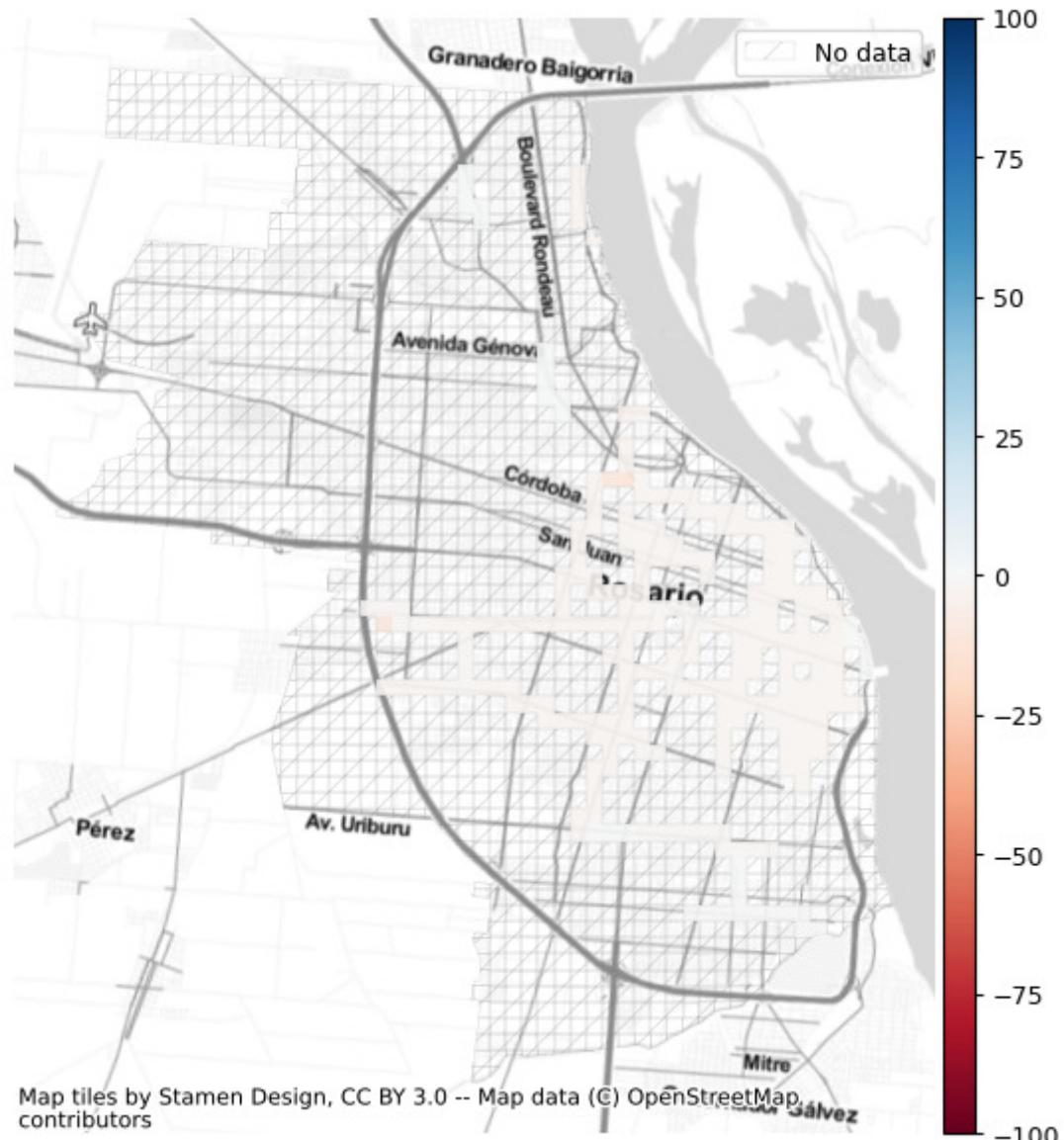
## Component connectivity

Here we visualize differences between how many cells can be reached from each cell. The metric is a crude measure for network connectivity but has the benefit of being computationally cheap and thus able to quickly highlight stark differences in network connectivity.

In the plot showing the difference in percent cells reached, positive values indicate a higher connectivity using the reference data set, while negative values means that more cells can be reached from a particular cell in the OSM data.



### Rosario: EMR difference to OSM in percent of cells reached



## Summary

### Extrinsic Quality Comparison

	OSM	EMR
<b>Total infrastructure length (km)</b>	175	117
<b>Protected bicycle infrastructure density (m/km<sup>2</sup>)</b>	373	435
<b>Unprotected bicycle infrastructure density (m/km<sup>2</sup>)</b>	726	299
<b>Mixed protection bicycle infrastructure density (m/km<sup>2</sup>)</b>	3	0

<b>Bicycle infrastructure density (m/km2)</b>	1,103	733
<b>Nodes</b>	342	168
<b>Dangling nodes</b>	143	94
<b>Nodes per km2</b>	2	1
<b>Dangling nodes per km2</b>	1	1
<b>Overshoots</b>	1	1
<b>Undershoots</b>	0	19
<b>Components</b>	43	25
<b>Length of largest component (km)</b>	104	86
<b>Largest component's share of network length</b>	61%	74%
<b>Component gaps</b>	13	18
<b>Alpha</b>	0.09	0.08
<b>Beta</b>	1.05	1.00
<b>Gamma</b>	0.35	0.34

## 3b. Feature Matching

The feature matching takes the reference data and attempts to identify corresponding features in the OSM data set. Feature matching is a necessary precondition to compare single features rather than feature characteristics on study area a grid cell level.

### Method

Matching features in two road data sets with each their way of digitizing features and a potential one-to-many relationship between edges (for example in the case where one data set only maps road center lines, while the other map the geometries of each bike lane) is not a trivial task.

The method used here converts all network edges to smaller segments of a uniform length before looking for a potential match between the reference and the OSM data. The matching is done on the basis of the buffered distance between objects, the angle, and the undirected Hausdorff distance, and is based on the works of [Koukoletsos et al. \(2012\)](#) and [Will \(2014\)](#).

Based on the matching results, the following values are computed:

- The number and length of matched and unmatched edges, in total and per grid cell
- A comparison of the attributes of the matched edges: Is their classification of cycling infrastructure as protected or unprotected the same?

### Interpretation

It is important to visually explore the feature matching results, since the success rate of the matching influences how the analysis of number of matches should be interpreted.

If the features in the two data sets have been digitized differently - e.g. if one data set has digitized bike tracks as mostly straight lines, while the other includes more winding tracks, the matching will fail. This is also the case if they are placed too far from each other. If it can be confirmed visually that the same features do exist in both data sets, a lack of matches indicates that the geometries in the two data sets are too different. If however it can be confirmed that most real corresponding features have been identified, a lack of matches in an area indicates errors of commission or omission.

### Sections

- Match features
  - [Run and plot feature matching](#)
  - [Matched and unmatched features](#)
  - [Feature matching summary](#)
- Analyze feature matching results
  - [Matched features by infrastructure type](#)
  - [Feature matching success](#)
- Summary

## Match features

Run and plot feature matching

Interactive map saved at results/COMPARE/Rosario/maps\_interactive/segment\_matches\_15\_17\_30\_compare.html

Rosario: OSM matched & unmatched features

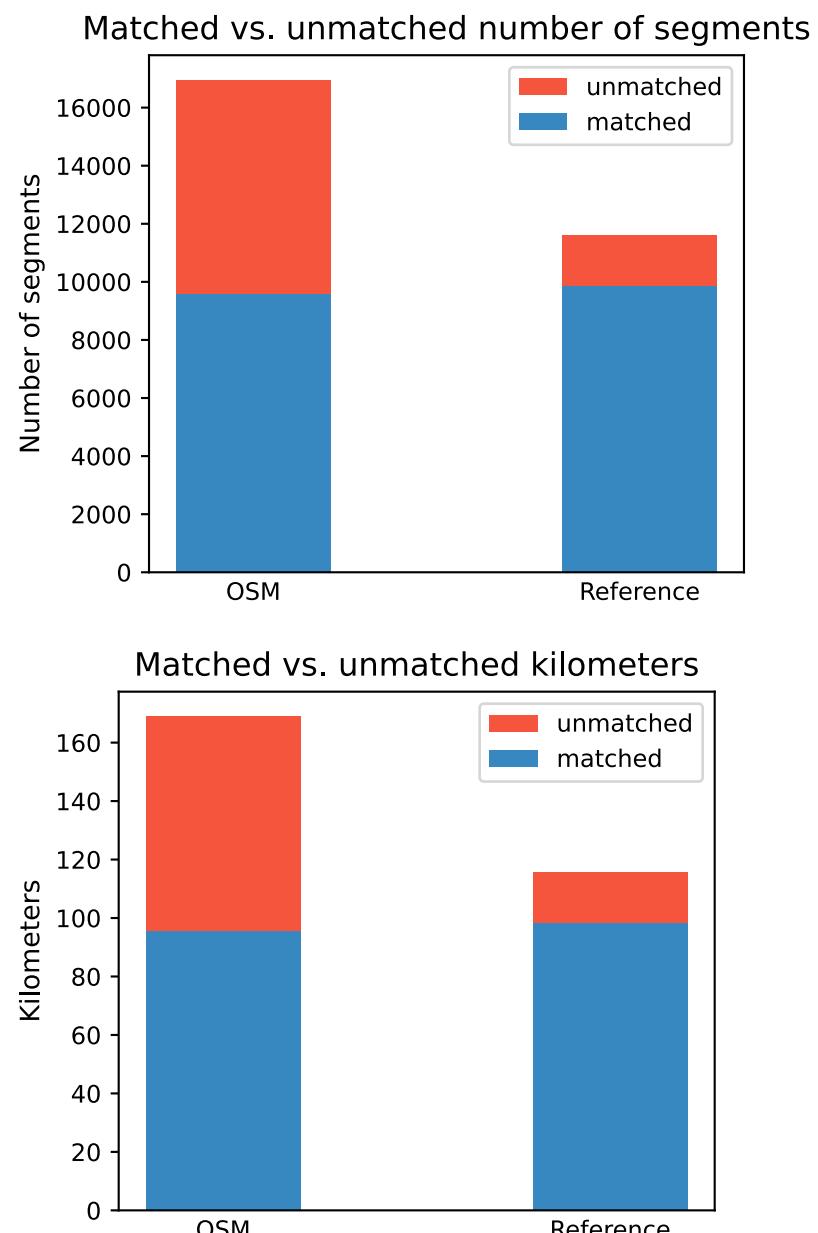


## Rosario: EMR matched & unmatched features



## Feature matching summary

Edge count: 9565 of 16956 OSM segments (56.41%) were matched with a reference segment.  
Edge count: 9859 out of 11603 EMR segments (84.97%) were matched with an OSM segment.  
Length: 95.52 km out of 168.98 km of OSM segment (56.53%) were matched with a reference segment.  
Length: 98.28 km out of 115.72 km of EMR segments (84.93%) were matched with an OSM segment.



## Analyze feature matching results

Matched features by infrastructure type



### Feature matching success

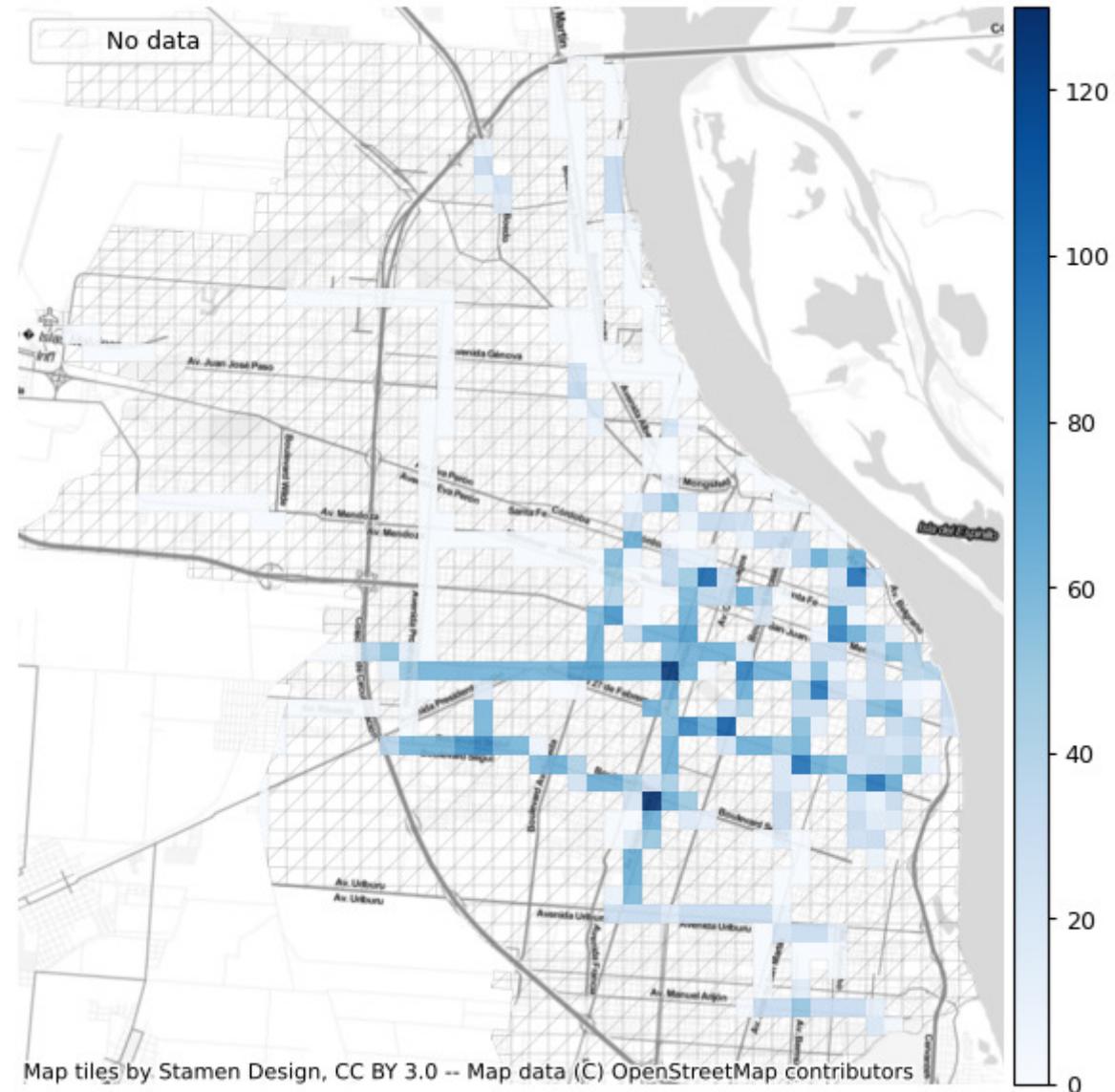
In the plots below, the count, percent, and length of matched and unmatched segments in each data set are summarized.

#### Warning

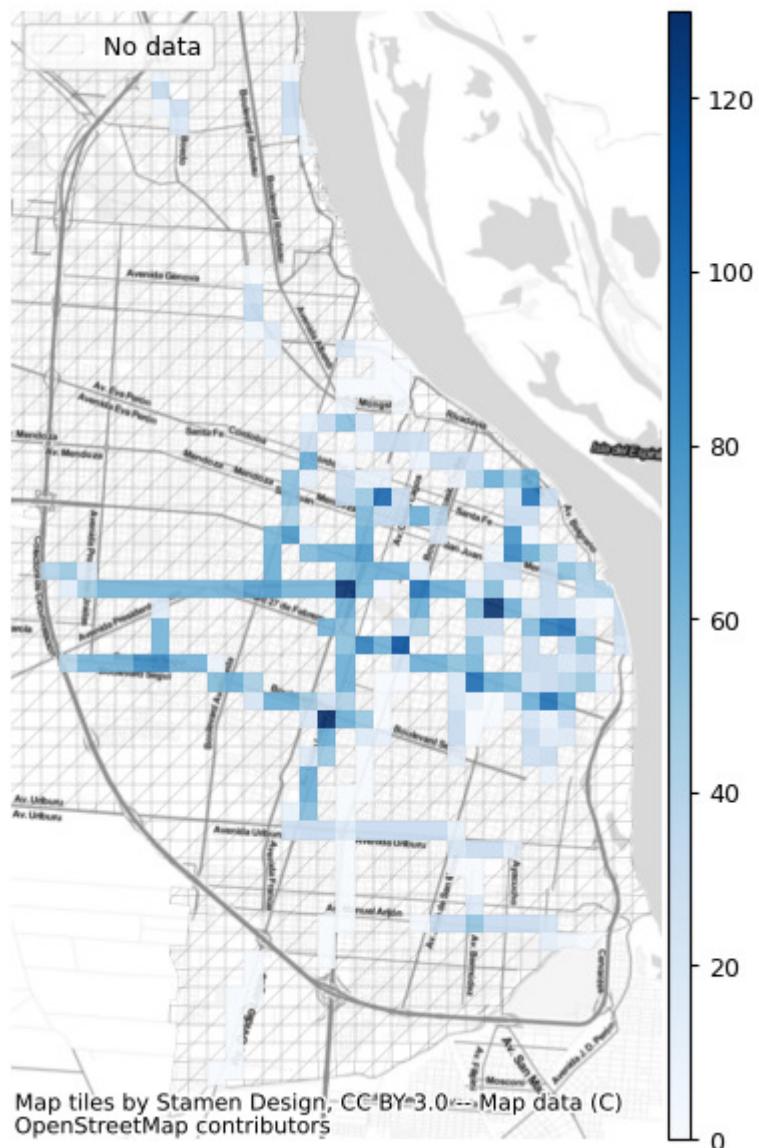
The number of matched segments in one data set in a grid cell does not necessarily reflect the number of matched segments in the other data set, since a segment can be matched to a corresponding segment in another cell. Moreover, the local count refers to segments intersected with the grid cell. For example, a segment crossing 2 cells will thus be counted as matched in 2 different cells. This does not change the relative distribution of matched/unmatched segments, but it does entail that the overall summary of matched/unmatched segments above uses a different total count of segments than the plots below.

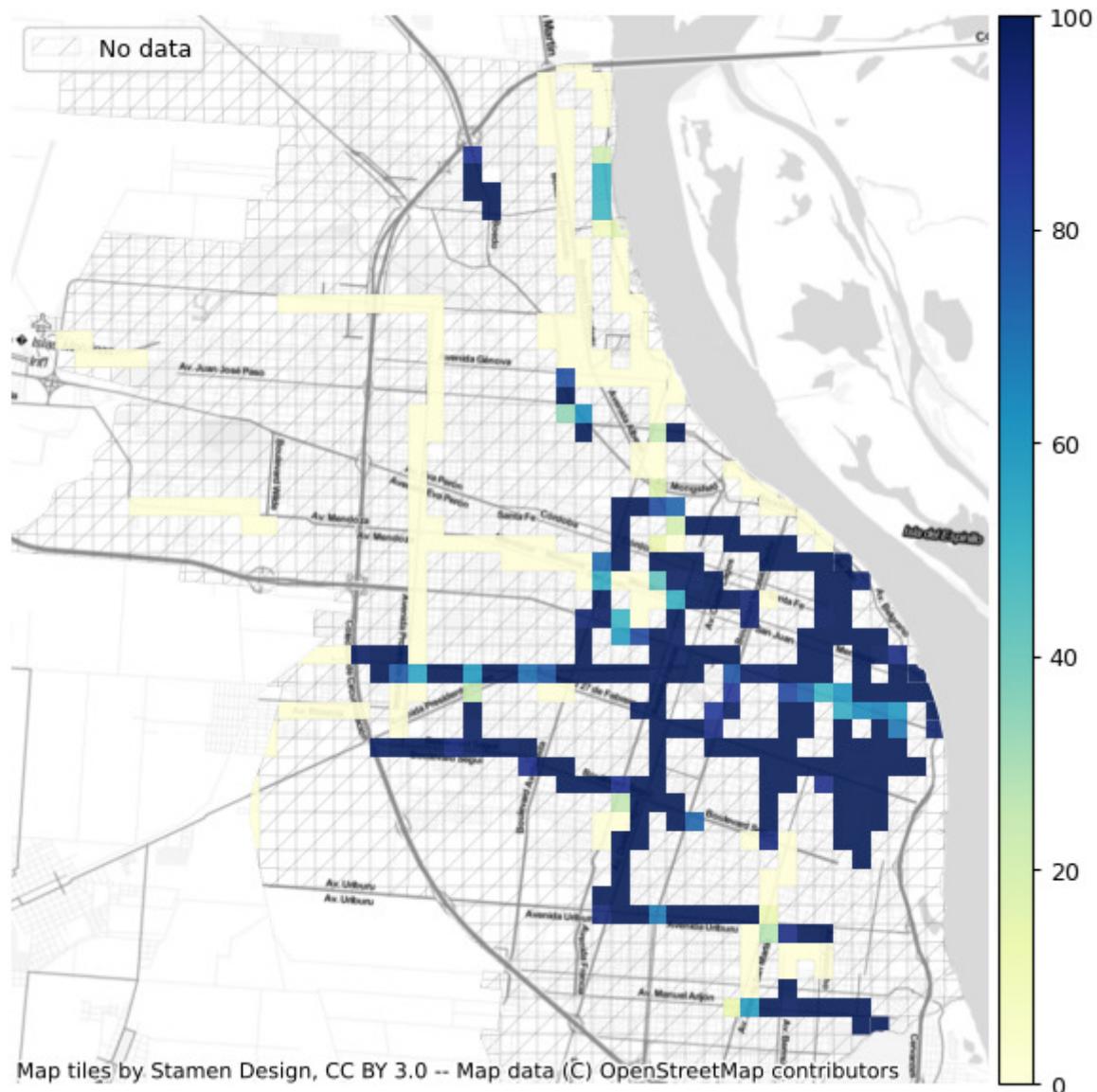
```
/var/folders/b0/lkvf88hn0673f5dlj9z0_2dr0000gn/T/ipykernel_37378/3687147659.py:4: UserWarning: `keep_geom_type=True` in overlay resulted in 17 dropped geometries of different geometry types than df1 has. Set `keep_geom_type=False` to retain all geometries
    osm_segments_joined = gpd.overlay(osm_segments[['geometry','seg_id','edge_id']], grid, how="intersection")
```

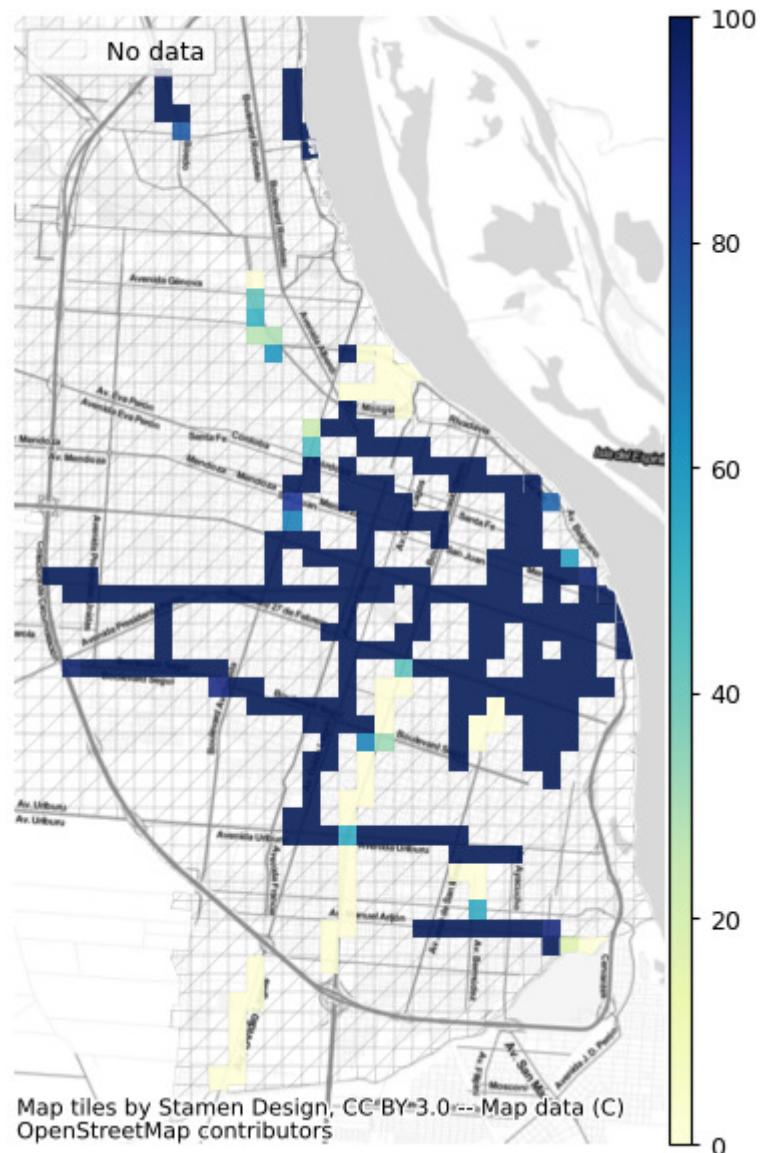
### Rosario: OSM features matched to reference data

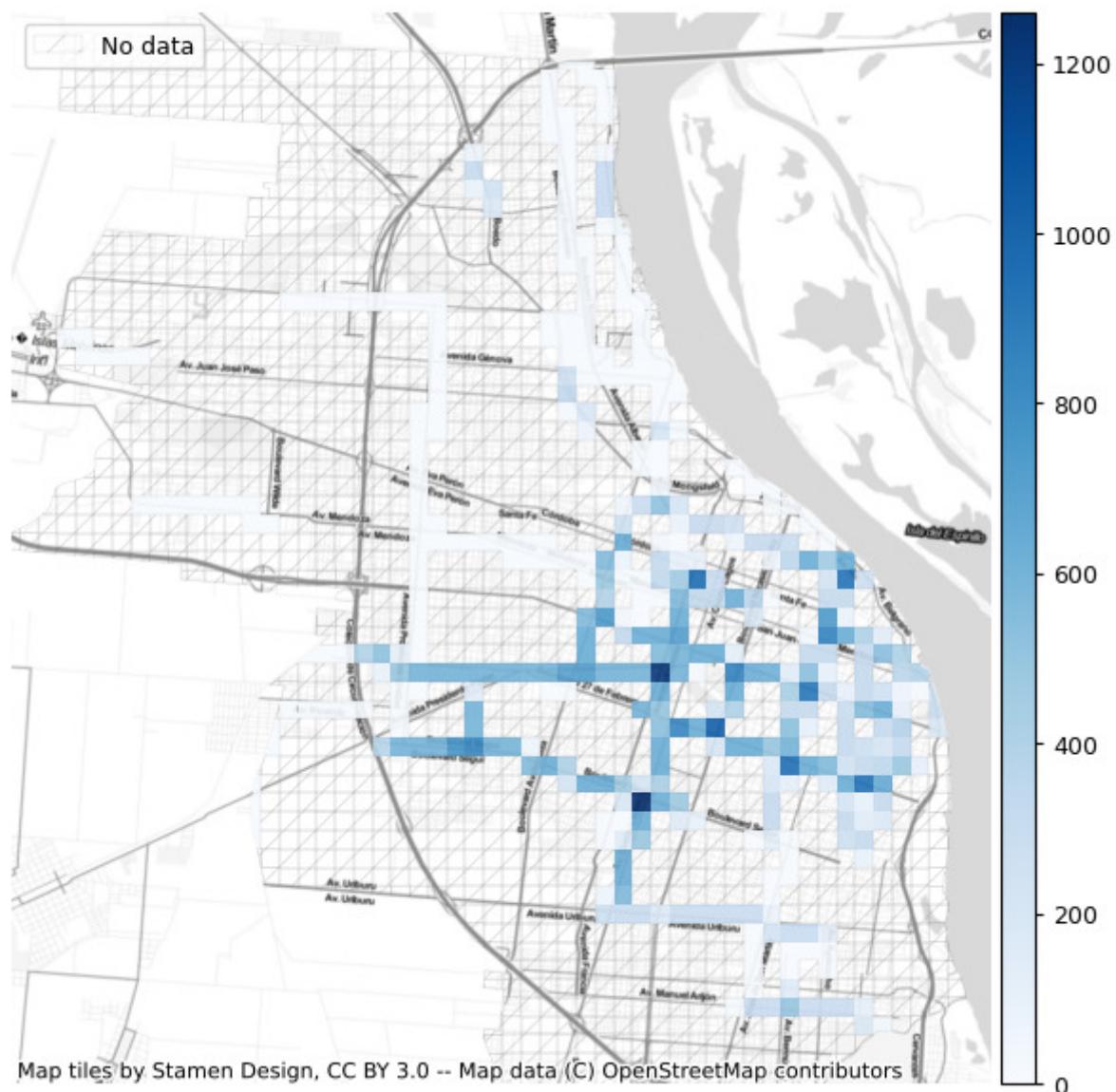


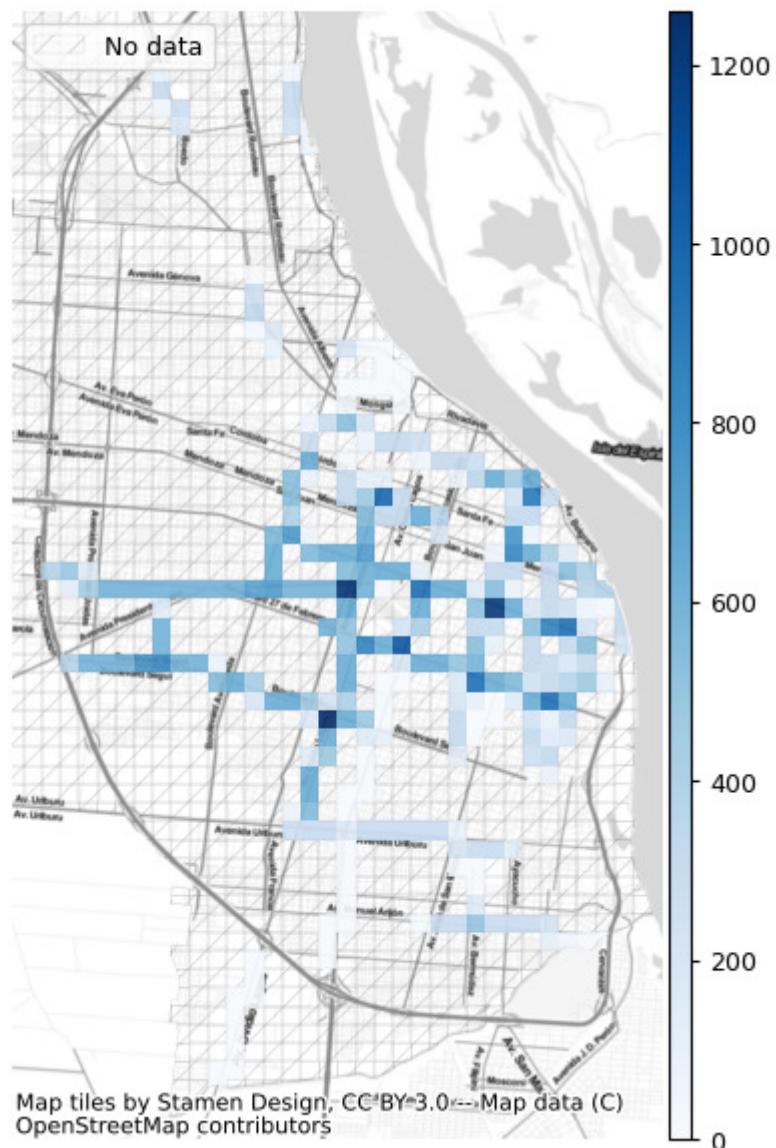
### Rosario: EMR features matched to OSM data

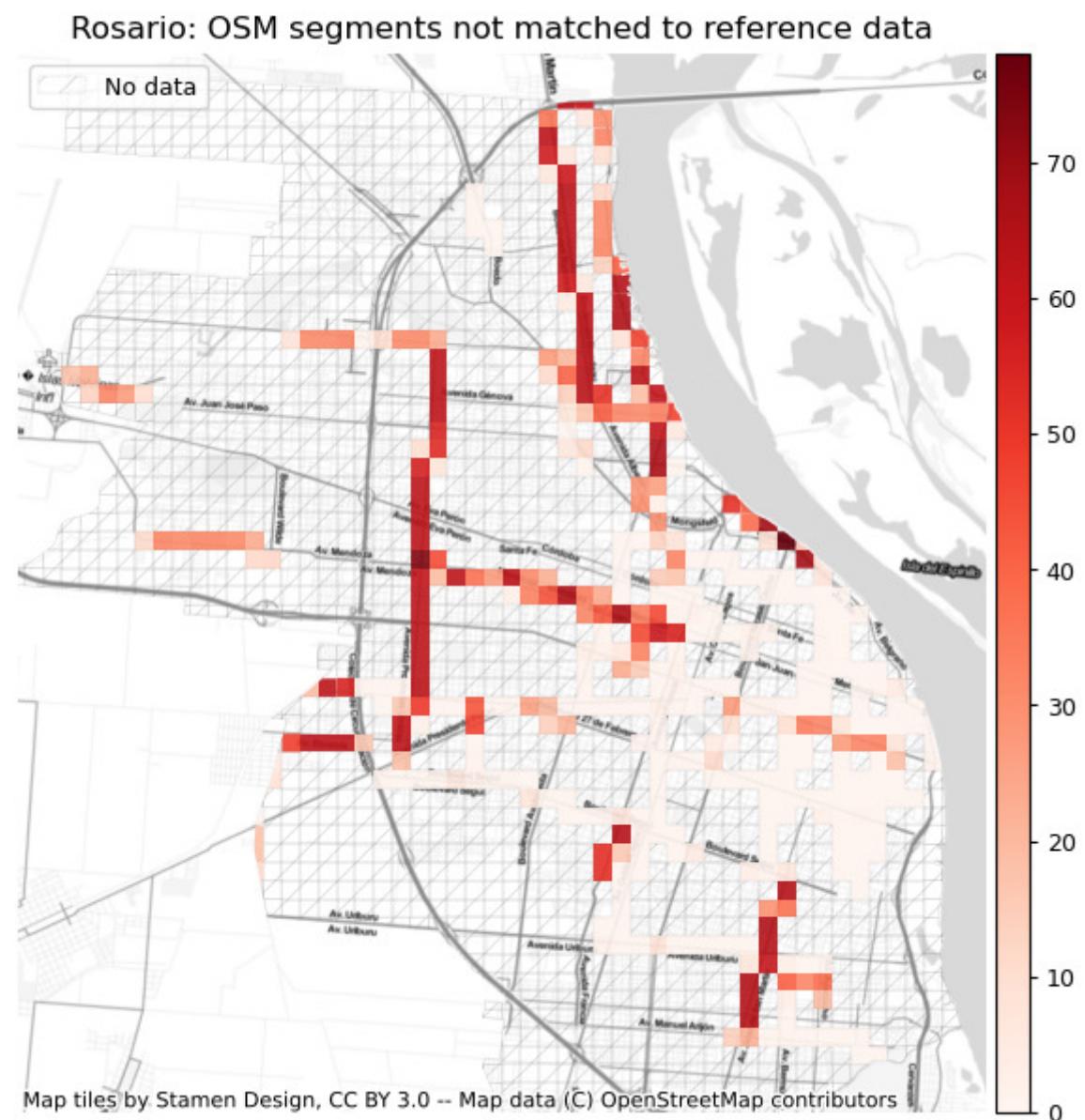


**Rosario: percent of OSM features matched to reference data2**

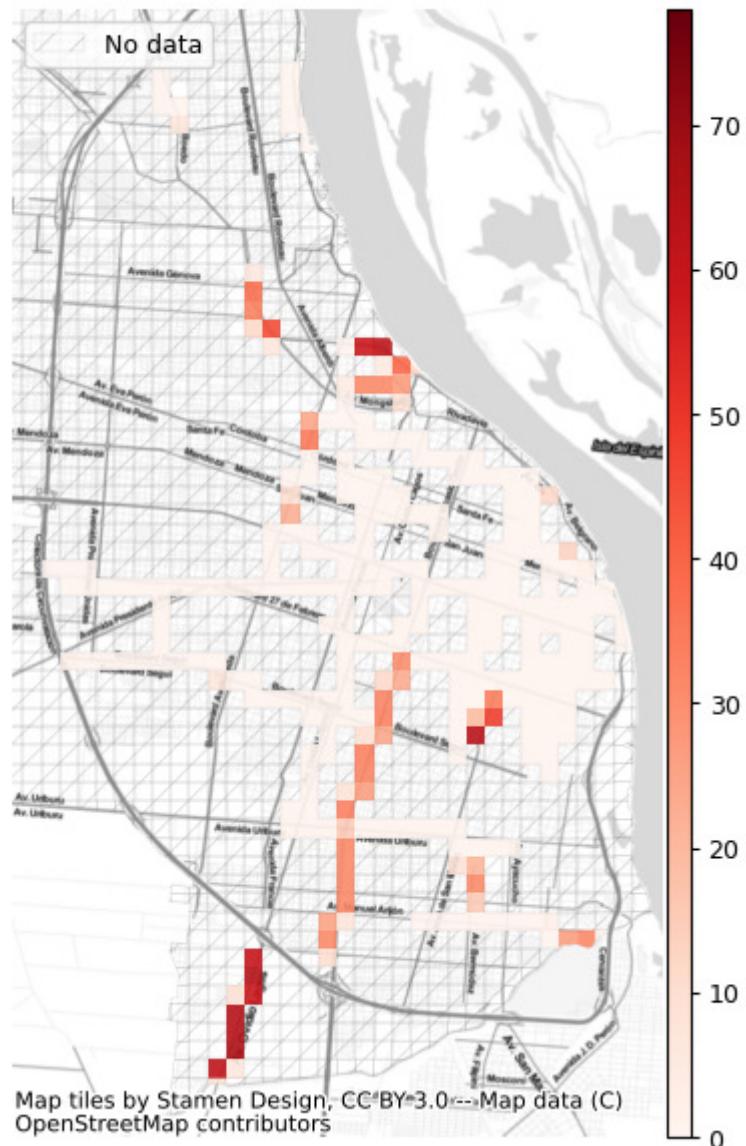
**Rosario: percent of EMR features matched to OSM data2**

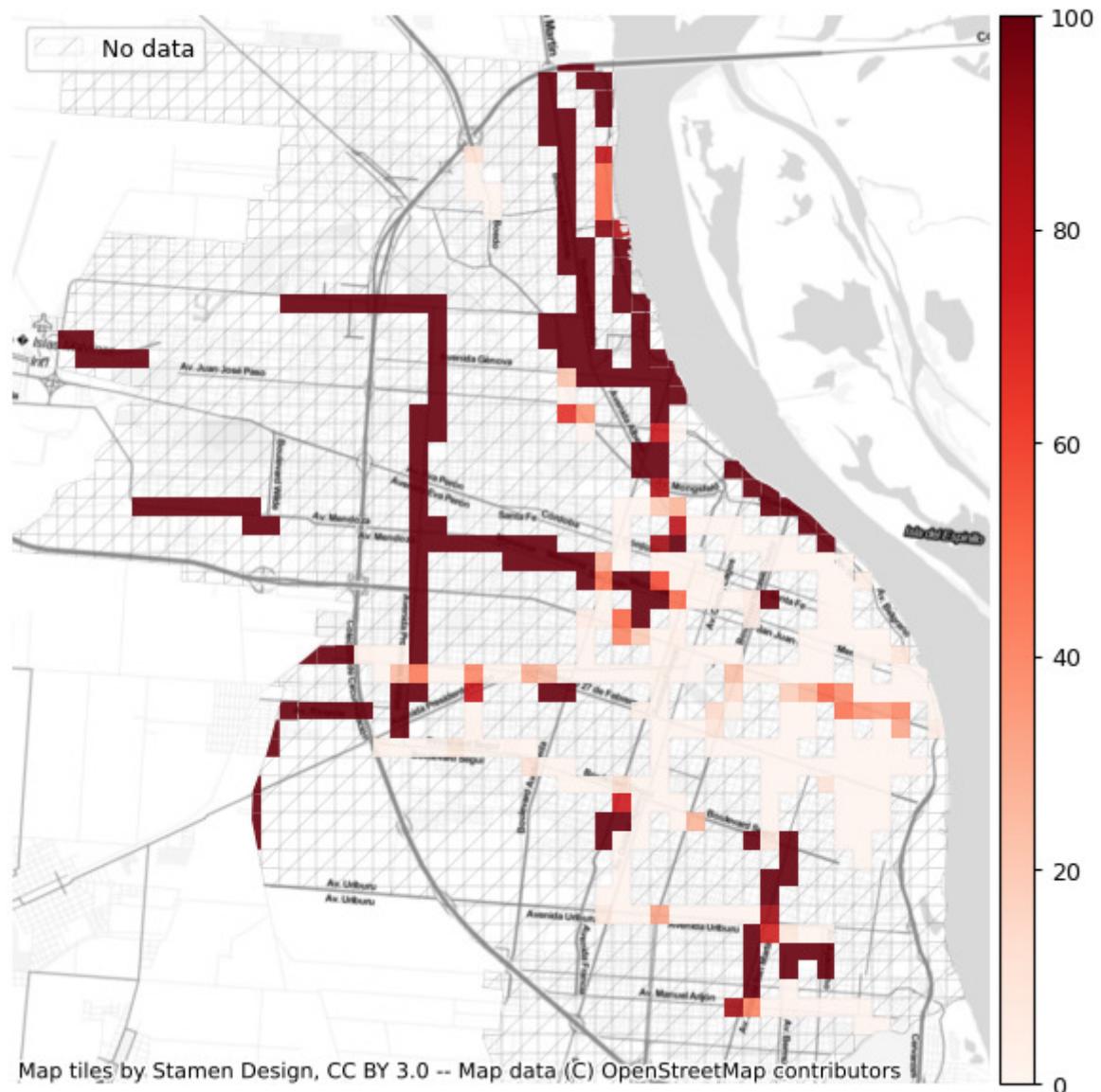
**Rosario: length of OSM features matched to reference data (m)**

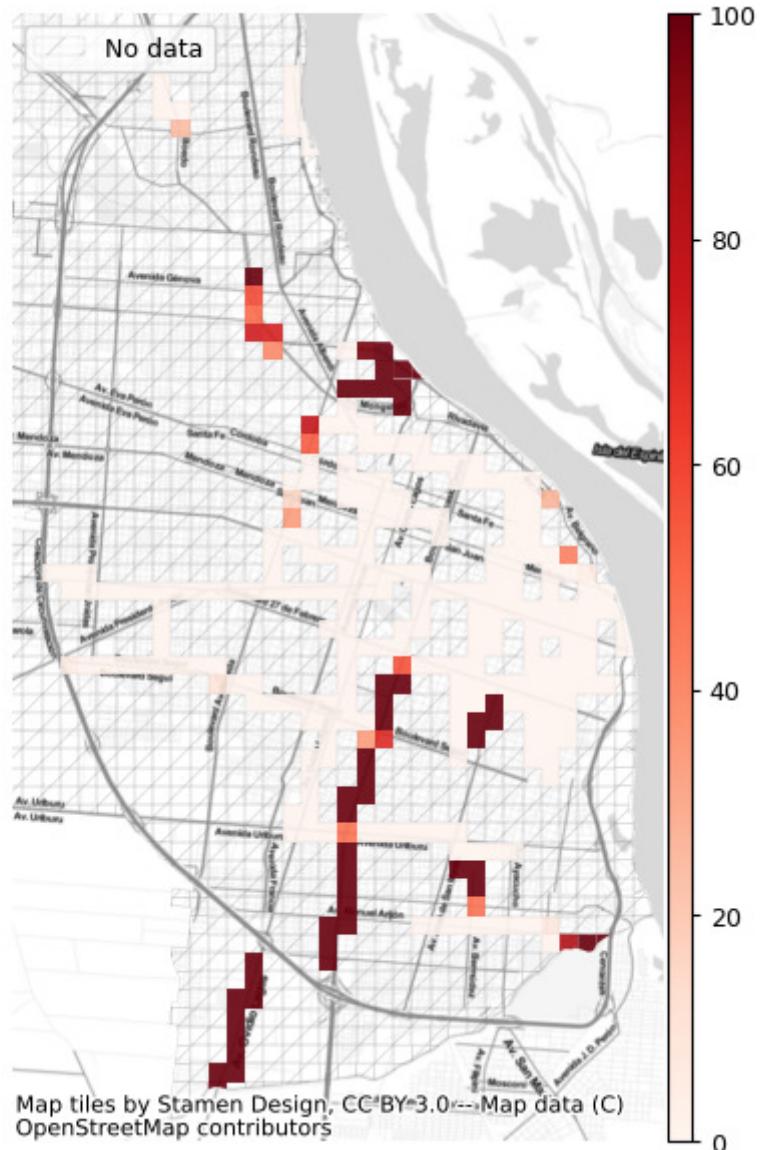
**Rosario: length of EMR features matched to OSM data (m)**

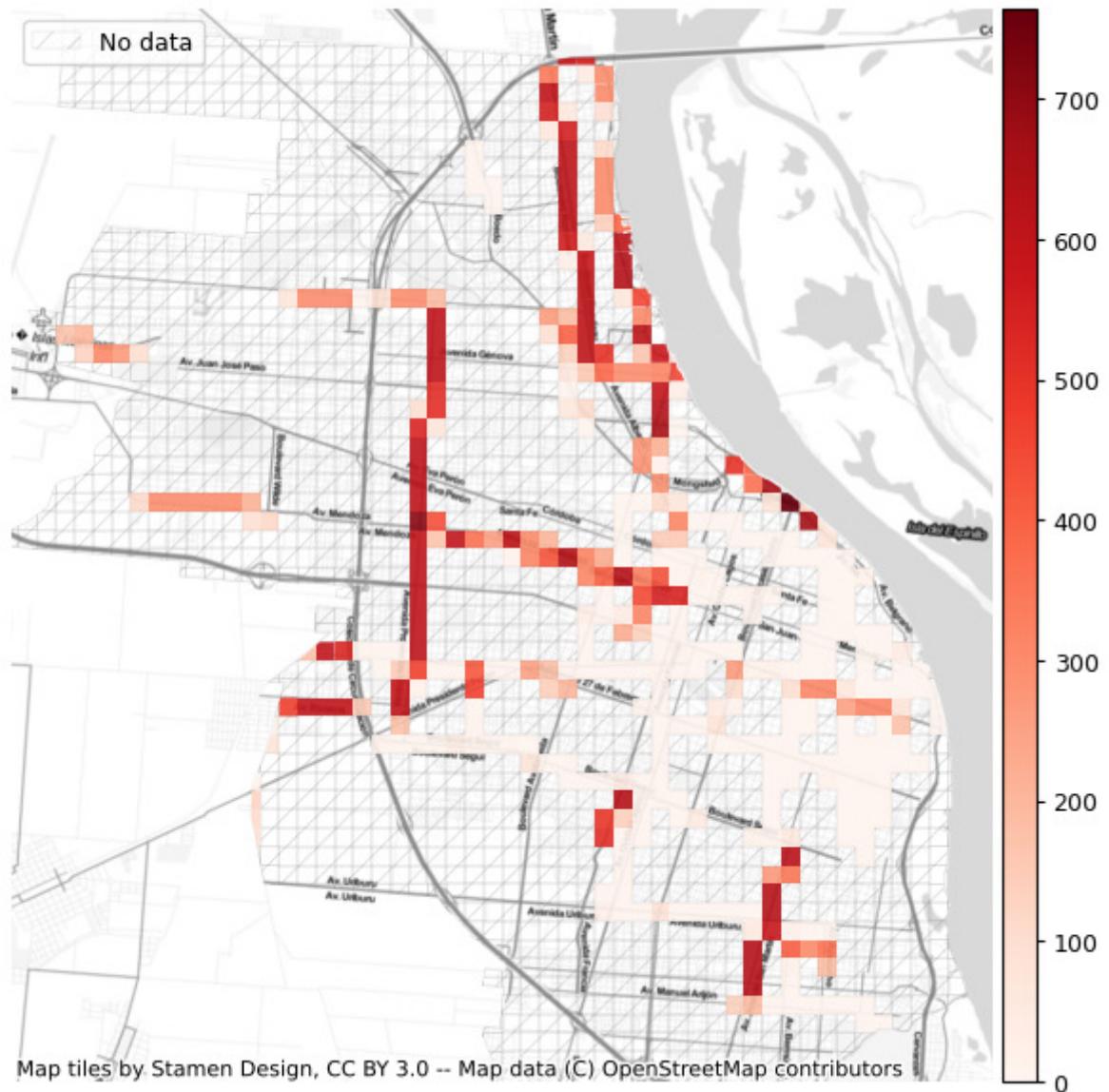


### Rosario: EMR segments not matched to OSM data

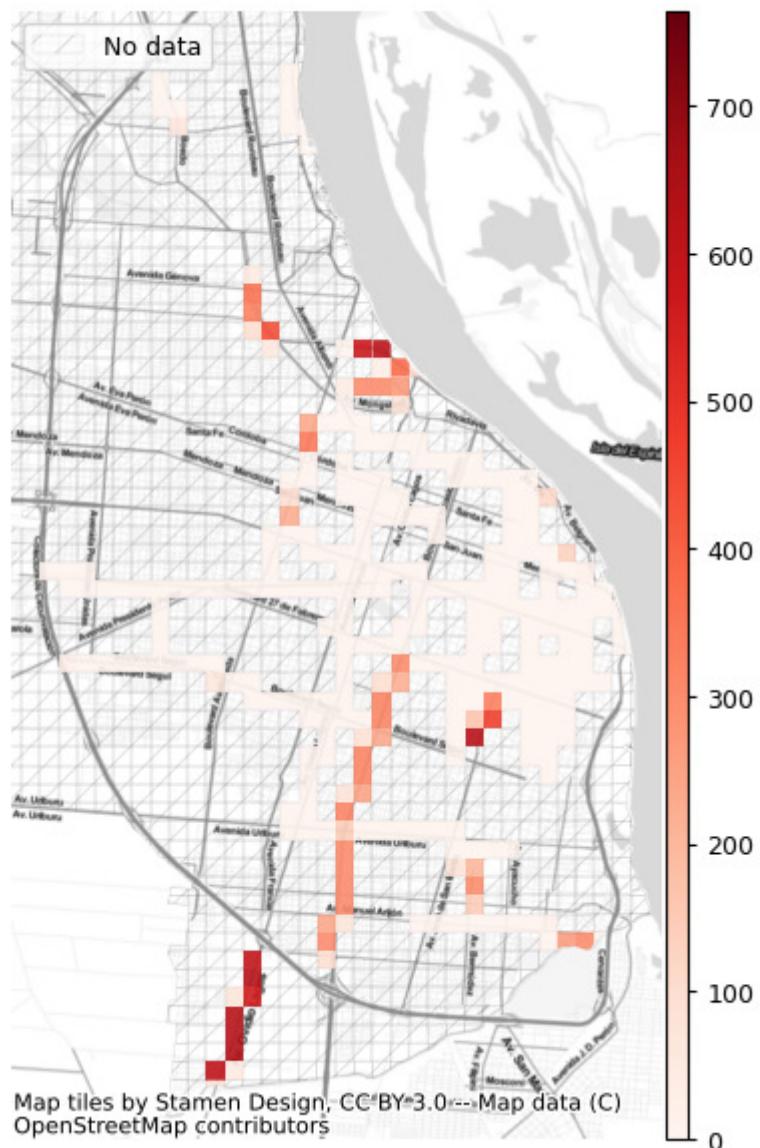


**Rosario: percent of OSM segments not matched to reference data**

**Rosario: percent of EMR segments not matched to OSM data**

**Rosario: length of OSM segments not matched to reference data**

## Rosario: length of EMR segments not matched to OSM data



## Summary

### Feature Matching Results

	OSM	EMR
<b>Count of matched segments</b>	9,565	9,859
<b>Percent matched segments</b>	56%	85%
<b>Length of matched segments (km)</b>	96	98
<b>Percent of matched network length</b>	57%	85%
<b>Local min of % matched segments</b>	7%	3%

**Local max of % matched segments** 100% 100%

**Local average of % matched segments** 90% 96%

# Appendix A: config.yml

This notebook shows the content of `config.yml`, i.e. the parameters that were used to run the analysis.

```
{
  'area_name': 'Rosario',
  'reference_name': 'EMR',
  'study_area': 'Rosario',
  'study_crs': 'EPSG:22171',
  'plot_resolution': 'low',
  'bicycle_infrastructure_queries': {'A': "highway == 'cycleway'", "B": 'cycleway in \'["lane','track','opposite_lane','opposite_track','designated','crossing']\'", "C": 'cycleway_left in \'["lane','track','opposite_lane','opposite_track','designated','crossing']\'", "D": 'cycleway_right in \'["lane','track','opposite_lane','opposite_track','designated','crossing']\'"}, "simplify_tags_queries": {"centerline_false_bidirectional_true": ["highway == ''cycleway'' & "(oneway=='no' "or "oneway_bicycle=='no')", "highway == ''track'' & "bicycle == ''designated'' & "(oneway=='no' "or "oneway_bicycle '==='no')", "highway == ''path'' & "bicycle == ''designated'' & "(oneway=='no' "or "oneway_bicycle '==='no')"], "centerline_false_bidirectional_false": ["highway '== 'cycleway' '& (oneway != 'no' or "oneway_bicycle != 'no')", "highway '== 'track' & "bicycle '== 'designated'"]}
}
```

```

'& (oneway '
"!= 'no' or "
'oneway_bicycle '
"!= 'no')",
'highway '
"== 'path' "
'& bicycle '
'== '
"'designated' "
'& (oneway '
"!= 'no' or "
'oneway_bicycle '
"!= 'no')",
'centerline_true_bidirectional_true': ['cycleway_left '
'in '
"['lane', 'track', 'opposite_lan
e', 'opposite_track', 'designated', 'crossing'] "
'and '
'(cycleway_right '
'in '
"['no', 'none', 'separate'] "
'or '
'cycleway_right.isnull() '
'or '
'cycleway_right '
'not in '
"['lane', 'track', 'opposite_lan
e', 'opposite_track', 'designated', 'crossing']) "
'and '
'oneway_bicycle '
"=='no'",
'cycleway_right '
'in '
"['lane', 'track', 'opposite_lan
e', 'opposite_track', 'designated', 'crossing']) "
'and '
'(cycleway_left '
'in '
"['no', 'none', 'separate'] "
'or '
'cycleway_left.isnull() '
'or '
'cycleway_left '
'not in '
"['lane', 'track', 'opposite_lan
e', 'opposite_track', 'designated', 'crossing']) "
'and '
'oneway_bicycle '
"=='no'",
'cycleway_in '
"['lane', 'track', 'opposite_lan
e', 'opposite_track', 'designated', 'crossing'] "
'and '
'(oneway_bicycle '
"== 'no' or "
'oneway_bicycle.isnull()', "
'cycleway_left '
'in '
"['lane', 'track', 'opposite_lan
e', 'opposite_track', 'designated', 'crossing'] "
'and '
'cycleway_right '
'in '

```

```

                    "['lane','track','opposite_lane',
'e','opposite_track','designated','crossing']]",
                    'centerline_true_bidirectional_false': ['cycleway_left ',
'in ',
"['lane','track','opposite_la
ne','opposite_track','designated','crossing']] "
                    'and '
                    '(cycleway_right '
                    'in '
                    "['no','none','separate'] "
                    'or '
                    'cycleway_right.isnull() '
                    'or '
                    'cycleway_right '
                    'not in '
                    "['lane','track','opposite_la
ne','opposite_track','designated','crossing']] "
                    'and '
                    'oneway_bicycle '
                    "!=" 'no''',
                    'cycleway_right '
                    'in '
                    "['lane','track','opposite_la
ne','opposite_track','designated','crossing']] "
                    'and '
                    '(cycleway_left '
                    'in '
                    "['no','none','separate'] "
                    'or '
                    'cycleway_left.isnull() '
                    'or '
                    'cycleway_left '
                    'not in '
                    "['lane','track','opposite_la
ne','opposite_track','designated','crossing']] "
                    'and '
                    'oneway_bicycle '
                    "!=" 'no''',
                    'cycleway '
                    'in '
                    "['lane','track','opposite_la
ne','opposite_track','designated','crossing']] "
                    'and '
                    'oneway_bicycle '
                    "==" 'yes'"]},
'osm_bicycle_infrastructure_type': {'protected': ["highway == 'cycleway'",
                    'cycleway in ',
                    "['track','opposite_track']",
                    'cycleway_left in ',
                    "['track','opposite_track']",
                    'cycleway_right in ',
                    "['track','opposite_track']"],
                    'unprotected': ['cycleway in ',
                    "['lane','opposite_lane','crossing']",
                    'cycleway_left in ',
                    "['lane','opposite_lane','crossing']",
                    'cycleway_right in ',
                    "['lane','opposite_lane','crossing']"],
                    'unknown': ["cycleway in ['designated']",
                    'cycleway_left in ',
                    "['designated']",
                    'cycleway_right in ',
                    "['designated']]"},
```

```

'osm_way_tags': ['access',
                  'barrier',
                  'bridge',
                  'bicycle',
                  'bicycle_road',
                  'crossing',
                  'cycleway',
                  'cycleway:left',
                  'cycleway:right',
                  'cycleway:both',
                  'cycleway:buffer',
                  'cycleway:left:buffer',
                  'cycleway:right:buffer',
                  'cycleway:both:buffer',
                  'cycleway:width',
                  'cycleway:left:width',
                  'cycleway:right:width',
                  'cycleway:both:width',
                  'cycleway:surface',
                  'foot',
                  'footway',
                  'highway',
                  'incline',
                  'junction',
                  'layer',
                  'lit',
                  'maxspeed',
                  'maxspeed:advisory',
                  'moped',
                  'motor_vehicle',
                  'motorcar',
                  'name',
                  'oneway',
                  'oneway:bicycle',
                  'osm_id',
                  'segregated',
                  'surface',
                  'tracktype',
                  'tunnel',
                  'width'],
'existing_tag_analysis': {'surface': {'true_geometries': ['surface',
                                                               'cycleway_surface'],
                                         'centerline': ['cycleway_surface']},
                           'width': {'true_geometries': ['width',
                                                         'cycleway_width',
                                                         'cycleway_left_width',
                                                         'cycleway_right_width'],
                                     'centerline': ['cycleway_width',
                                                   'cycleway_left_width',
                                                   'cycleway_right_width']},
                           'speedlimit': {'all': ['maxspeed']},
                           'lit': {'all': ['lit']}},
'incompatible_tags_analysis': {'bicycle_infrastructure': {'yes': [['bicycle',
                                                               'no'],
                                                               ['bicycle',
                                                               'dismount'],
                                                               ['car',
                                                               'yes']]}}},
'grid_cell_size': 300,
'reference_geometries': 'true_geometries',
'bidirectional': 'is_bidirectional',
'ref_bicycle_infrastructure_type': {'protected': ["protection == 'protected'"],
                                    'unprotected': ['protection == ']
}

```

```
        '''unprotected'''}],  
'reference_id_col': 'unique_id'}
```