# SDS Bootcamp - Madrid March 2024
## Scalable spatial analytics

In this session you will learn:

- How to **automate complex spatial analysis pipelines** using an intuitive drag-and-drop interface.
- How to **make operations on geospatial data** such as selecting events happening in an area, or creating composite scores.
- How to **scale your analysis** using spatial indexes and tilesets for analyzing and visualizing large amounts of data fast and efficiently.

---

The purpose of this analysis is to identify areas with a deficit of mobile phone antennas. We would like to identify busy areas, i.e., areas with a lot of human activity to later verify if the number of antennas in these locations are enough to satisfy demand while providing a high quality service.

We'll start by analyzing cell tower data, and then continue to measure human activity using population, POIs, roads and nighttime light data because we're not only interested in residents, but also floating populations. These four variables are a good proxy of human activity and will be combined through a composite score. Once the composite score is computed, we'll be interested in understanding city dynamics.

Note that this use case is translatable to other fields such as: insurance (where accidents occur), CPG (where my products are sold), retail (where I can open a new store), etc.

---

The session is organized into 3 subsections, with the goal of analyzing telco data for network optimization, telecommunications investment, and coverage assessment:

1. **Analyzing cell tower data** to understand the spatial patterns of 4G mobile phone antennas in Madrid.
2. **Preparing external data**, including creating a Human Activity Index, that we will later use to understand antenna's patterns more in detail.
3. Discovering **why antennas are placed at specific locations** by running advanced analysis.

**Requirements**

*Platform*
You only need to have a CARTO trial account to follow this session.

*Data*
We have given public access to it for a smooth experience during the workshop:

- Cell tower data (OpenCelliD)
- Population, POIs, urbanity (CARTO Spatial Features)
- Nighttime light data (Earth Observation Group)
- Green spaces (parks and forests) and road segments (OpenStreetMaps)
- Building footprints (Overture Maps Foundation, available through Data Observatory)
- Delimitations of the different neighborhoods within the municipality of Madrid (Geoportal del Ayuntamiento de Madrid)

*Repository*
You can access files with the complete workflows used in this session here.

## 1. Analyzing cell tower data

The purpose of this analysis is to understand the spatial patterns of 4G mobile phone antennas in Madrid while learning how to analyze and manipulate spatial point data.

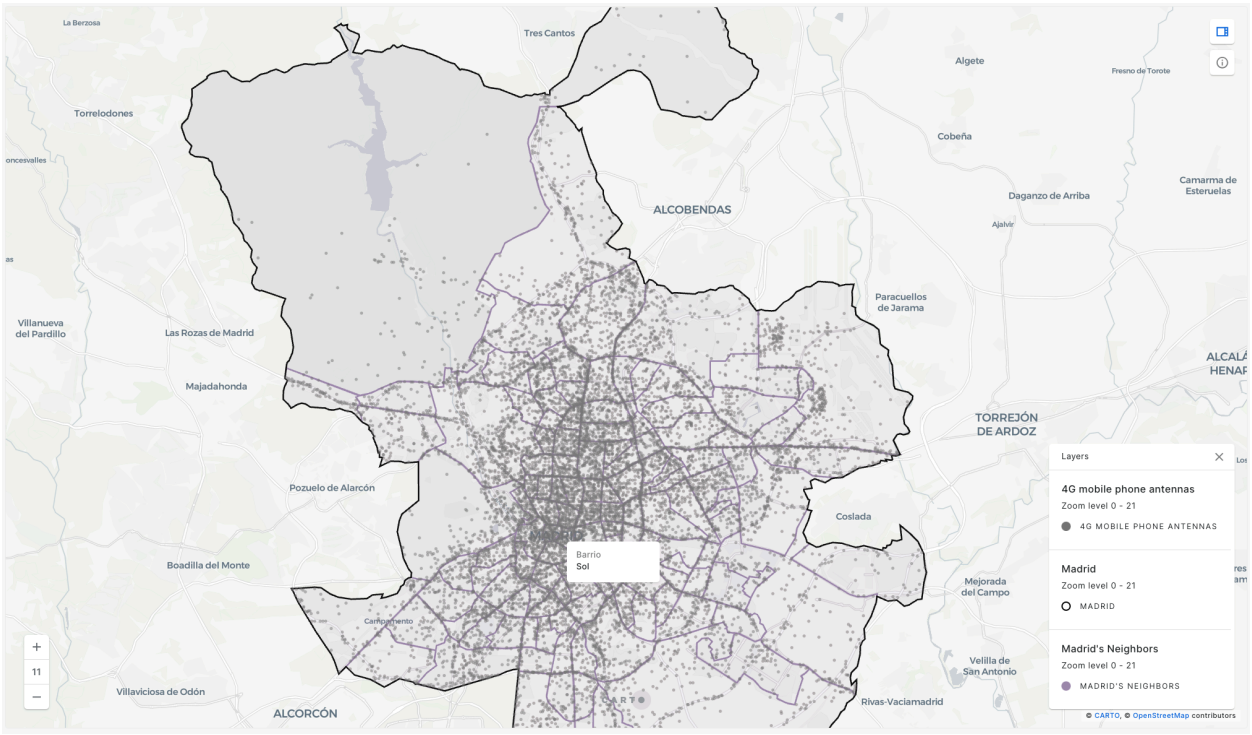
We will start by loading the cell tower and the Madrid boundaries datasets into the workflow to visualize the distribution of the antennas in our Area of Interest (AOI). To do this, use the `Get Table by Name` component and copy the table's fully qualified names (FQN):

- `cartobq.docs.cell_towers_esp`
- `cartobq.docs.barrios_madrid`

The next step is to clean and filter the antennas dataset. Keeping in mind that we are only interested in 4G antennas, we will create the following pipeline:

1. Use the `Simple Filter` component to filter out antennas with `radio = 4G`
2. Then, `Select` the `lat` and `lon` columns to later `Remove Duplicated` rows. This way, we are only keeping antennas with a unique location.
3. Use the `ST GeogPoint` component to create point geographies based on the columns containing latitude and longitude values.
4. Lastly, we will `Generate UUID` (Universally Unique Identifier) to keep track of each antenna in our data.

Now, let us select those antennas that lie within our AOI, the municipality of Madrid. Since we have a dataset with several geographies, namely each neighborhood in the municipality, we first need to `COMBINE` the `geom` column into a single geometry using the `Summarize` component. Next, we can perform a `Spatial Filter` to select only the antennas that `INTERSECT` the Madrid boundary.




Analyzing cell tower data

Once we have processed the cell tower data, we can start analyzing it using some advanced tools. We will perform three different analysis:

- *Looking for areas of high concentration* of mobile phone antennas to identify where to allocate resources or take further action.
- *Defining working areas* to allocate maintenance teams.
- *Computing the spatial autocorrelation* of the antennas to define if there is a clear relationship throughout space.

## Looking for areas of high concentration of mobile phone antennas

Now, we want to understand where we can find more antennas, and whether there are locations with higher concentrations or not. To do this, we first need to statistically prove whether antennas are clustered in space or not by computing the Nearest Neighbor Index (NNI) of the set of points in an AOI as follows:

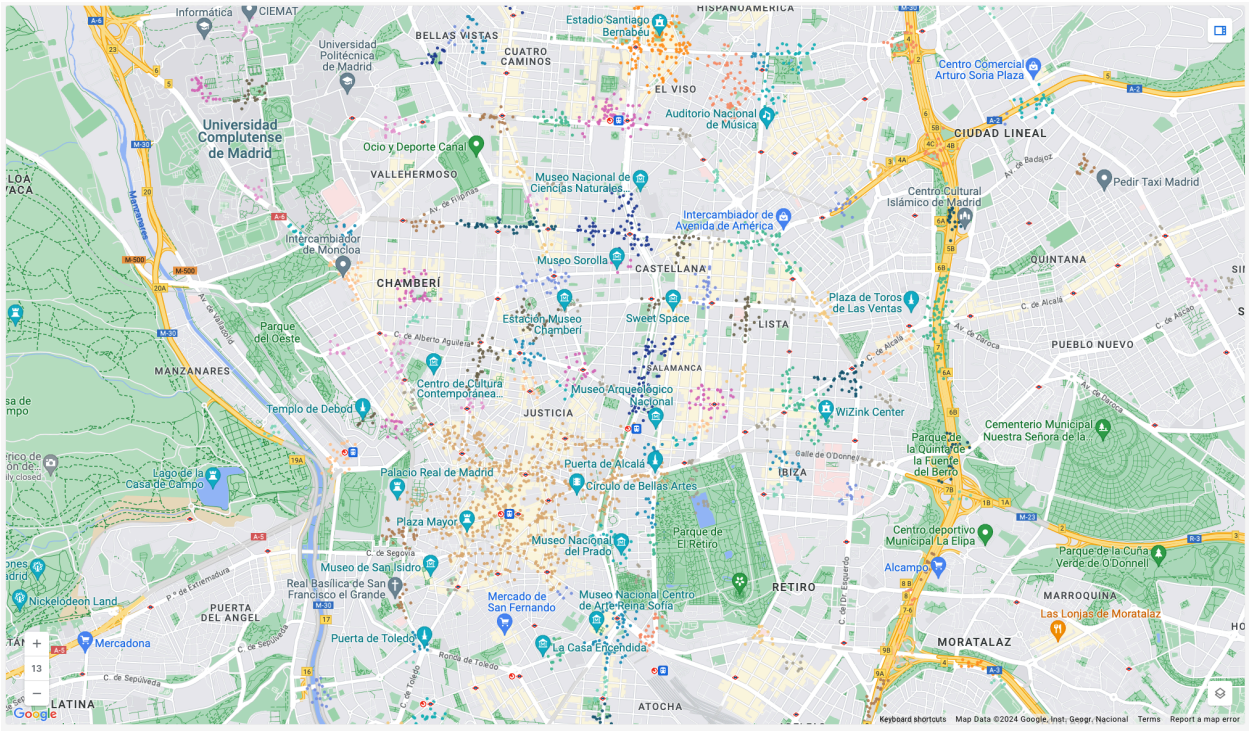
1. Run a `K-Nearest Neighbors` algorithm to get the distance of each antenna to their closest neighbor. Select `1` as the number of neighbors.
2. Get the area of the AOI in square `meters` using the `ST Area` component.

3. Compute the Nearest Neighbor Index by connecting the output of the two previous steps to a `Custom SQL Select` component, where you should type the following query:

```
SELECT
   COUNT(*) n,
   ANY_VALUE(geom_union_area) area,
   AVG(distance) d_obs,
   0.5 / SQRT(COUNT(*) / ANY_VALUE(geom_union_area)) d_exp,
   AVG(distance)*SQRT(COUNT(*) / ANY_VALUE(geom_union_area))/0.5 ratio,
   (AVG(distance)-0.5 / SQRT(COUNT(*) / ANY_VALUE(geom_union_area))) / 0.26136 *
SQRT(POW(COUNT(*),2) / ANY_VALUE(geom_union_area)) z_score
FROM `$a`, `$b`
```

A ratio lower than 1 indicates a clustered pattern, a ratio larger than one indicates a dispersed pattern, and a ratio equal to 1 indicates a random pattern.

Then, we will use the `ST_Cluster_DBSCAN` component to cluster the antennas within the AOI and find locations with high concentrations of cell towers. Select a search radius of `100` m and `10` as the minimum number of geographies to consider. To analyze the results, we will `Group by` `cluster` and `COUNT` the number of antennas (`id`) within.



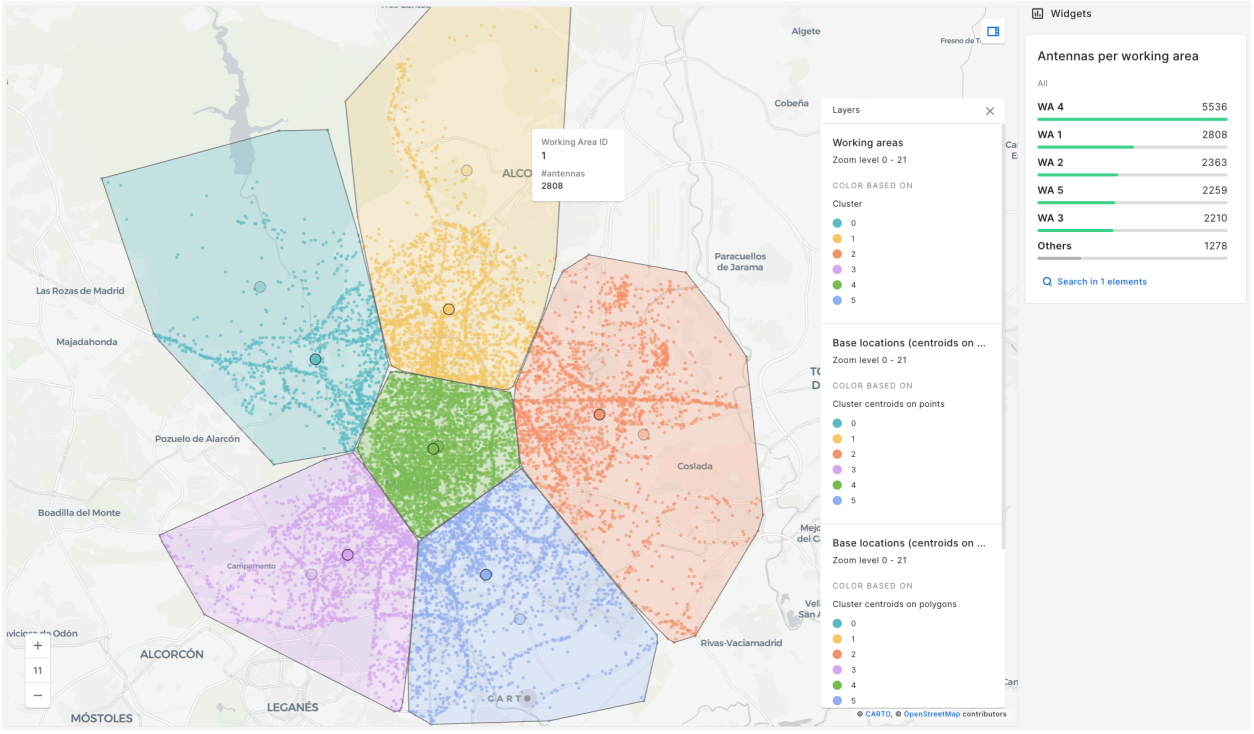

Looking for areas of high concentration of mobile phone antennas

## Defining working areas

Suppose we'd like to organize teams of technicians in charge of cell tower maintenance in six areas so that they are as close as possible to mobile phone antennas. In order to identify these areas, we run a k-means cluster analysis on the geographical coordinates using the `ST_Cluster_K-Means` component indicating that we want to get `6` clusters.

As a next step, we can calculate new base locations for each team to keep all the equipment as the centroids of all antennas falling in the same cluster. To do this, we will `Group by` `cluster`, and aggregate the `geom` column using the `CENTROID` method. We can also compute the `CONVEXHULL` of the `geom` column to get the area (or envelope) defined for each team, and the `COUNT` of the cell `id`s to see how many antennas are covered in each region.

Notice that we could also compute the centroid of each final area (the output of computing the convex hull) rather than using the geometries of cell towers, but this way, base locations won't be located as close as possible to mobile phone antennas. We could do this using the `ST_Centroid` component on the `geom_convexhull` column.




Defining working areas

## Computing the spatial autocorrelation of the antennas

To get a measure of spatial autocorrelation, we must:

1. Process the data into ***spatial index*** cells to obtain the distribution of antennas in the Madrid area. To do this, we will:
    - polyfill Madrid's geometry using the `H3 Polyfill` component (with `8` as resolution and `INTERSECT` as method), and
    - get the counts of antennas within each H3 cell by first assigning the corresponding H3 cell to each antenna using the `H3 from GeoPoint` component (with resolution `8`), and then aggregating by `h3` to `COUNT` the number of cell towers (`id`).

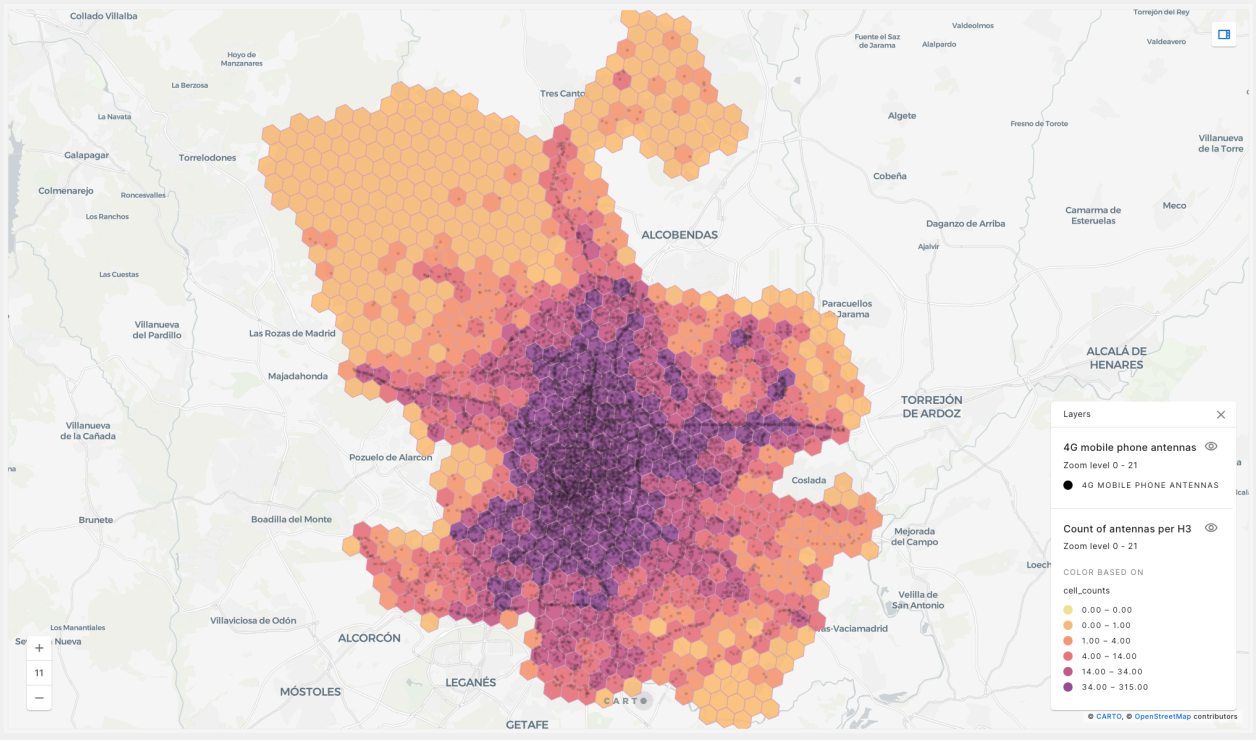
    Then, we have to `Join` both tables with a `FULL OUTER` join using the h3 index as primary key, and replace `NULL` values in the antenna's counts by `0` using a `Select` component with the following input. This way, we will have a cell tower count value for each H3 cell within the municipality of Madrid.

    ```
    h3, COALESCE(id_count_joined, 0) cell_counts
    ```

2. Compute the `Moran's I` statistic to measure the spatial dependency of data, i.e. whether characteristics at nearby locations are correlated. Use an `exponential` kernel of size `1`. Here, positive/negative values indicate positive/ negative autocorrelation (clustered/dispersed).

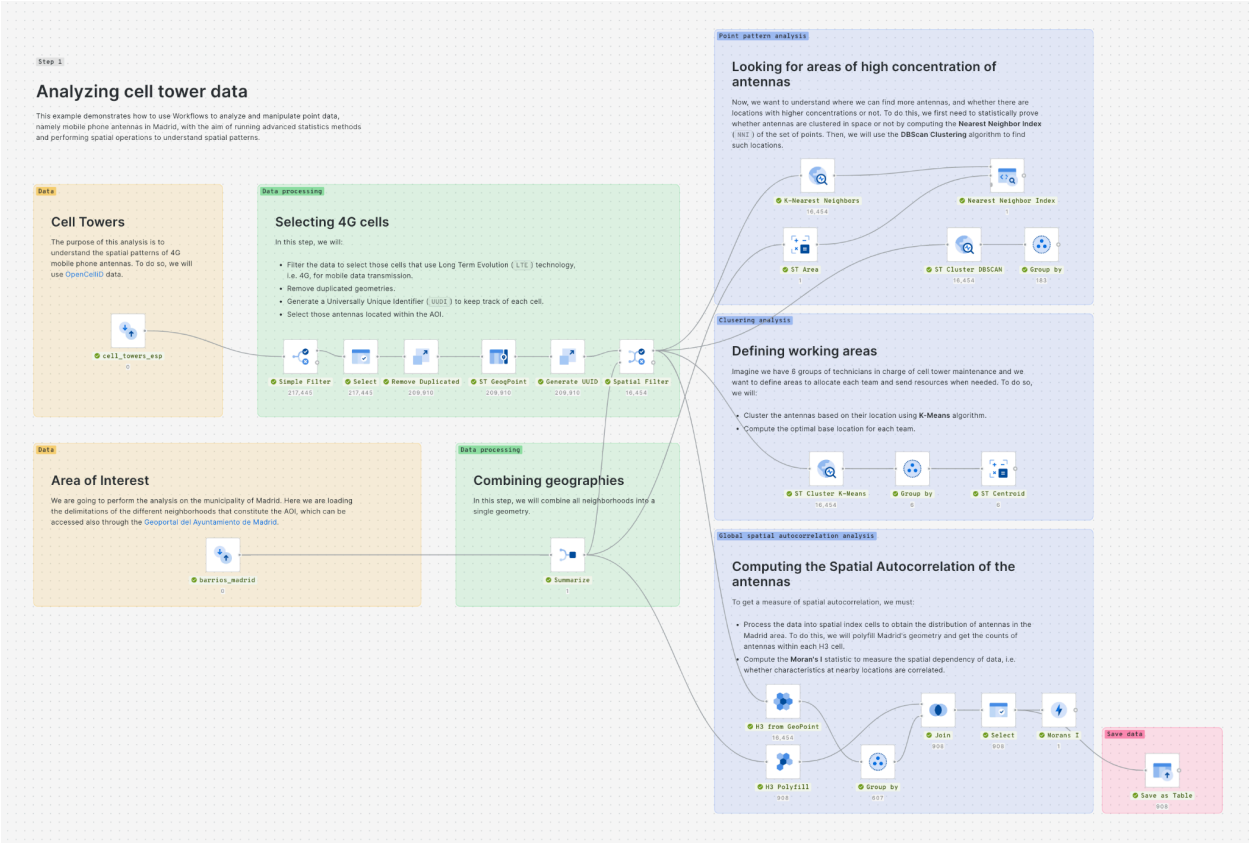
Lastly, we will `Save as Table` the processed antenna's count grid in the following destination, as we will use it later in a deeper analysis:

- `CARTO Data Warehouse > organization > private > cell_towers_madrid_h3z8`




Computing the spatial autocorrelation of the antennas

**Complete workflow:**




## 2. Preparing external data

When working with spatial data, information may come in different forms, and it is important to format it to ensure the data can be simply managed and reused. In this section, we will learn how to enrich an H3 grid with several features that we will later use to understand antenna's patterns more in detail. We have already learned how to work with point data, so here we will transform linestring and polygon data. Also, we will derive a Human Activity Index (HAI) by combining some of these variables to later understand the impacts of human mobility in mobile service coverage.

### Enriching the AOI grid with external data

As a first step, let's load the data into the canvas this time using different alternatives available in Workflows:

- *Nighttime light data*: we have processed this data from raster to H3 and uploaded it to a GCP (Google Cloud Platform) public bucket. Look for the Import file into workflow button just above the canvas and click on the canvas where you want to see your data. Select the `From URL` option and paste the following source URL and Cast the `nighttime_light` variable to `FLOAT`:

```
https://storage.googleapis.com/data_science_public/sds_bootcamps/madrid_202403/ntl_madrid_h3z8.csv
```

- *Spatial Features*: we have prepared a sample of this data and made it available for everyone in `cartobq.docs.spatial_features_h3z8_sample`. To load it, use the `Custom SQL Select` component and type in the following query, where we directly combine a few variables to create new features:

```
SELECT h3, population, retail + food_drink + education + transportation + financial + healthcare + leisure + tourism as pois, retail, food_drink, education, transportation, financial, healthcare, leisure, tourism, urbanity, female_20_to_24 + female_25_to_29 + male_20_to_24 + male_25_to_29 young_population
FROM `cartobq.docs.spatial_features_h3z8_sample`
```

- *Green Spaces & Road segments*: we have queried the OSM BigQuery public project `bigquery-public-data` to extract green spaces (`park`, `forest`, `grass`) and road segments (`highway`) geometries. To load this data, use the `Get Table by Name` component and copy the table's FQN:
  - `cartobq.docs.osm_road_segments_madrid`
  - `cartobq.docs.osm_green_spaces_madrid`

  *Note that here is a quick guide CARTO has prepared on how to extract and use the data you need*!

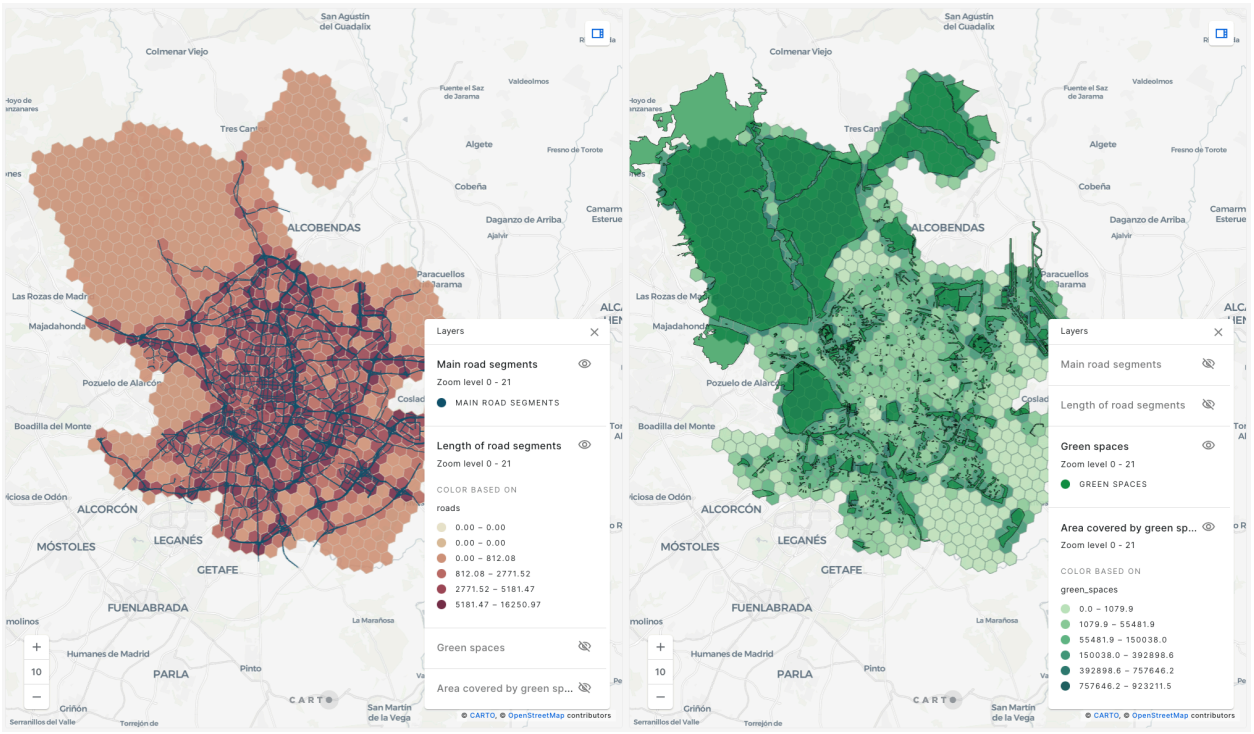
When the different sources of data are already in grid format, we just need to `Join` both tables using the spatial index h3 as primary key. That is what we are going to do with nighttime light and spatial features data.

Regarding road segments, we want to get the amount of road kilometers within each H3 cell. To process this, we will first filter the dataset using a `Where` clause to select specific road types:

```
highway_type IN ('tertiary', 'secondary', 'primary', 'tertiary_link', 'secondary_link', 'primary_link', 'trunk', 'trunk_link','motorway', 'motorway_link')
```

We can see all road types available using the `Select Distinct` component. Then, we will compute the length of each segment (`geometry_length`) using the `ST Length` component selecting `meters` as units. Then, we will use the `Enrich H3 Grid` component to enrich a target H3 table with road information: we will specify the `geometry_length` variable to be aggregated using the `SUM`.

The process for green spaces is very similar, but in this case we will compute the area of each polygon (using the [ST_Area]) in `meters` before enriching the H3 cell with the resulting `geom_area` variable using the `SUM` aggregation method.




Enriching the AOI grid with external data

Once we have all data in an H3 grid, we will clean a bit the results:

- We will drop unnecessary columns (`h3_joined`) using [Drop Columns]
- We will fill in incomplete information by replacing `NULL` values through a [Select] call:

```
h3,
COALESCE(nighttime_light,0) nighttime_light,
COALESCE(population_joined,0) population,
COALESCE(pois_joined,0) pois,
COALESCE(retail_joined, 0) retail,
COALESCE(food_drink_joined,0) food_drink,
COALESCE(education_joined,0) education,
COALESCE(transportation_joined,0) transportation,
COALESCE(financial_joined,0) financial,
COALESCE(healthcare_joined,0) healthcare,
COALESCE(leisure_joined,0) leisure,
COALESCE(urbanity_joined,'remote') urbanity,
COALESCE(young_population_joined,0) young_population,
COALESCE(geometry_length_sum,0) roads,
COALESCE(geom_area_sum,0) green_spaces
```

- We will get an ordinal version of the categorical variable `urbanity`, as it will be easy to work with in next steps. Use the `Case When` component to do this and name the new variable `urbanity_ordinal`. Add the following conditions:

```
When urbanity
= Very_High_density_urban → 5
= High_density_urban → 4
= Medium_density_urban → 3
= Low_density_urban → 2
= rural → 1
Else 0
```

**Creating a Human Activity Index**

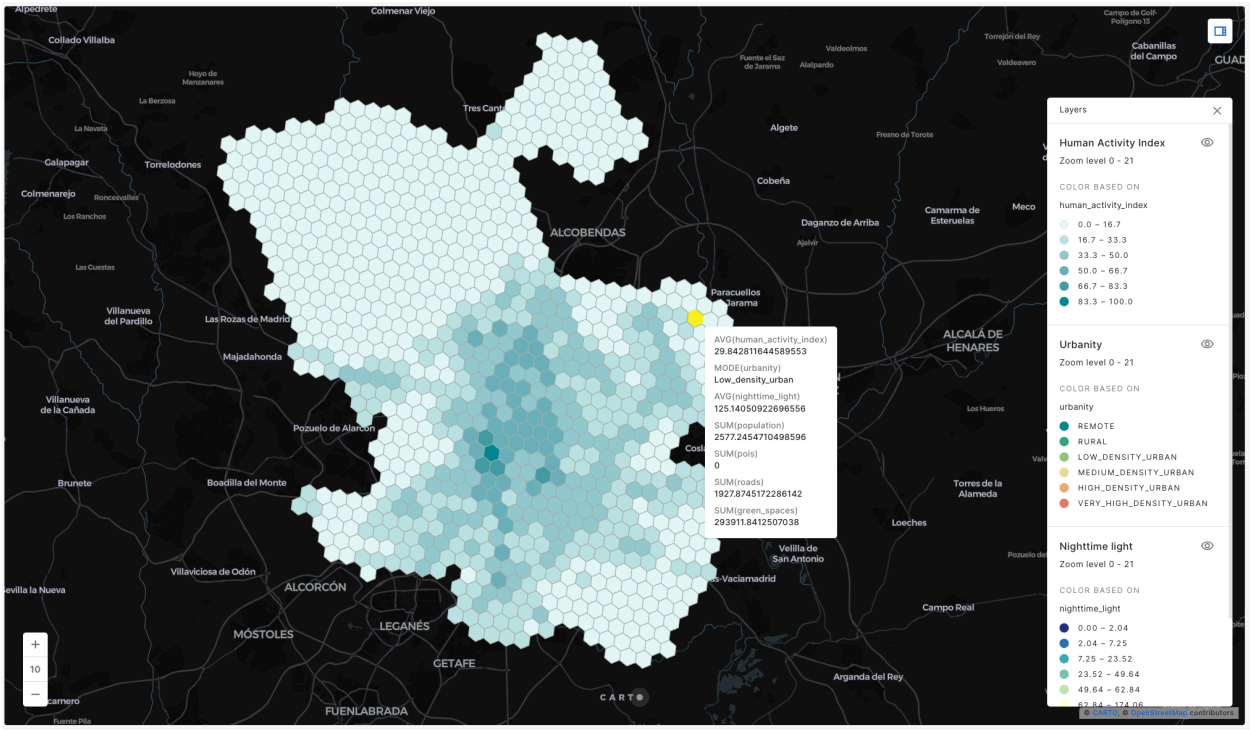
Now, it is time to create the HAI using the `Composite Score Unsupervised` component (check out our guide on *How to create a composite score with your spatial data* if you want to learn more about best practices!). The first step is to select the variables that we want to use to compute the index. We will `Select` road kilometers, total population, total number of pois and nighttime light data as a proxy for human activity, since it has been proven that they correlate well with human activity.

```
h3, roads, population, pois, nighttime_light
```

Before aggregating all the information into a single feature, our HAI, a good practice is to compute the `Cronbach Alpha Coefficient` of the input features (`roads`, `population`, `pois`, `nighttime_light`), which provides a measure of internal consistency or reliability of the data, based on the strength of correlations between individual variables. Higher alpha means higher consistency, with usually 0.65 being the minimum acceptable value of internal reliability.

To create our score, we will select the `FIRST_PC` method, that derives the spatial composite from a Principal Component Analysis as the first principal component score. The following parameters need to be specified:

- *Correlation variable:* the variable from the initial set of inputs that we want to have a positive correlation with the final score. We selected `nighttime_light`.
- *Correlation threshold:* the minimum absolute value of the correlation between each individual variable and the first principal component score. We selected `0,5`.
- *Output formatting:* how we want the final score to be processed (either bucketized, rescaled or none). We selected a `RETURN_RANGE` from `0` to `100`.

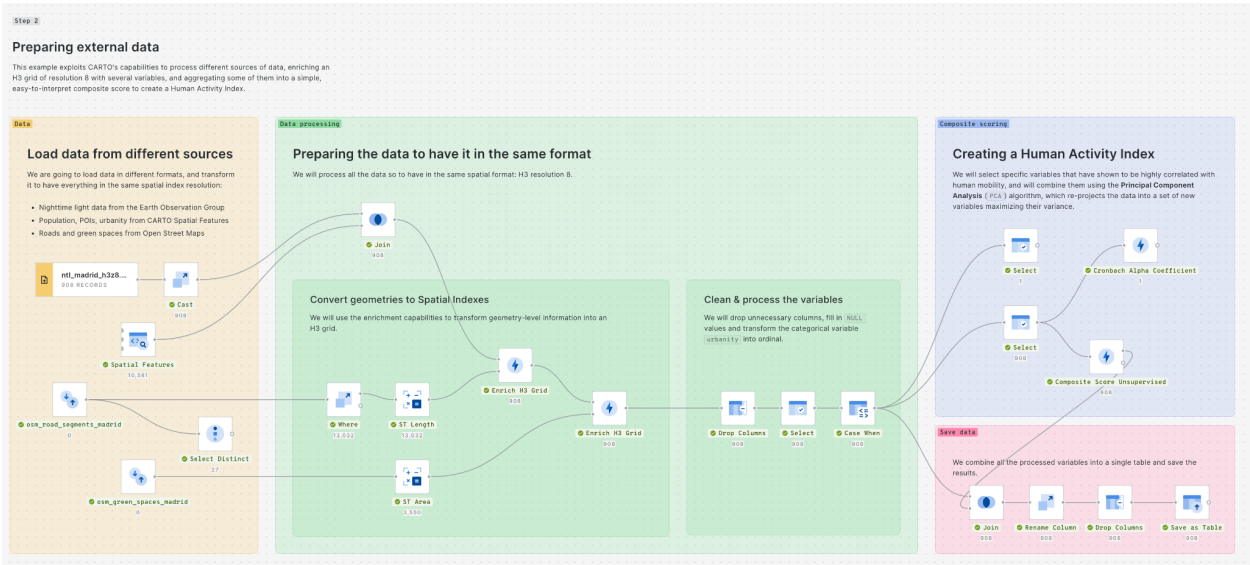



Creating a Human Activity Index

Lastly, we will Join the `spatial_score` (HAI) feature to our external features dataset, Rename Column from `spatial_score_joined` to `human_activity_index`, use Drop Columns to remove `h3_joined`, and Save as Table the results in the following destination, as we will use it the following section:

- `CARTO Data Warehouse > organization > private > features_madrid_h3z8`

## Complete workflow:



## 3. Why are antennas placed at specific locations?

In this analysis, we will run several advanced methods to explore the logic behind the placement of mobile phone antennas presence and to identify locations where demand is not satisfied. We will cover the following aspects using the data we have previously prepared:

- *Exploring local patterns* of mobile antennas to understand the spatial characteristics of such patterns
- *Understanding the effect of external data on antenna's presence*
- *Locating areas with a deficit of mobile phone antennas* to evaluate building's potential for new cell tower installations.

Drag and drop to the canvas the two previously saved tables and load Madrid's buildings dataset too, which will be used in the last step:

- `CARTO Data Warehouse > organization > private > cell_towers_madrid_h3z8`
- `CARTO Data Warehouse > organization > private > features_madrid_h3z8`
- `cartobq.docs.buildings_mad`

**Exploring local patterns of mobile phone antennas**

We will use the `Local Moran's I` statistic, a local measure of spatial autocorrelation, to identify quads:

- Local clusters of high values (HH) or low values (LL).
- Local spatial outliers in which a high value is surrounded primarily by low values (HL), and outliers in which a low value is surrounded primarily by high values (LH).

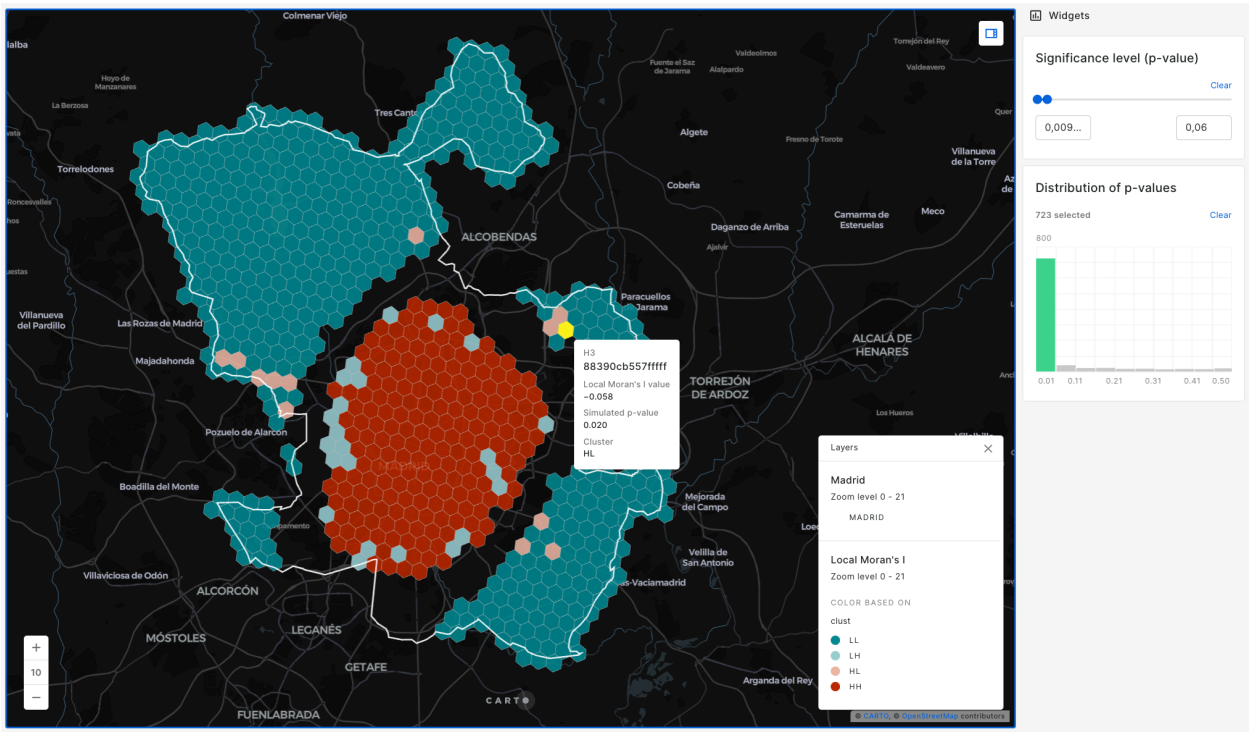
Use a `uniform` kernel of size `4`, and `100` permutations. Then, we are going to `Create Column` `clust`, that translates the output of the LMI into something easier to understand:

```
REPLACE(REPLACE(REPLACE(REPLACE(CAST(quad as string), '1', 'HH'), '2', 'LL'), '3',
'LH'), '4', 'HL')
```

Then, we will `Join` the input table (variables) with the resulting one to later obtain descriptive statistics to discover the characteristics of such hotspots, coldspots and outliers. To do this, `Group by` `clust_joined` and aggregate the following variables using the `AVG`, so as to get a measure of the average value of each feature in each cluster:

```
cell_counts_joined, nighttime_light, population, pois, roads, green_spaces,
urbanity_ordinal, human_activity_index, retail, food_drink, education,
transportation, financial, healthcare, leisure, young_population
```

You can Order by `clust_joined` to get the results sorted from HH to LL.

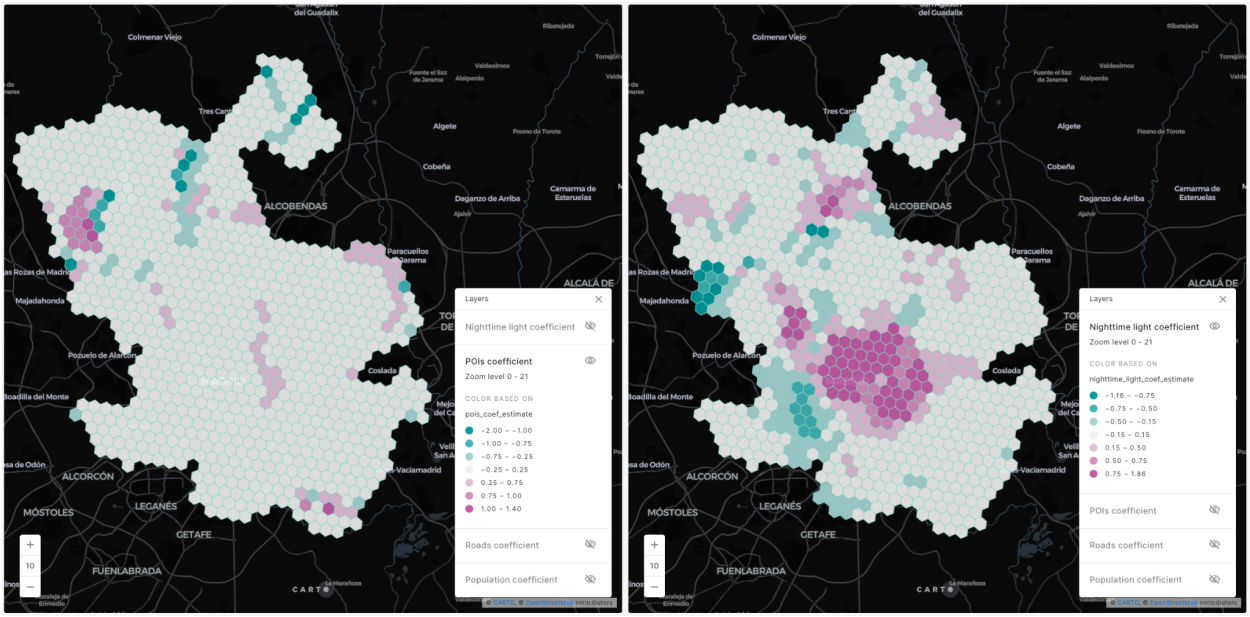



Exploring local patterns of mobile phone antennas

| clust_joined | cell_counts_joined_avg | nighttime_light_avg | population_avg | roads_avg | green_spaces_avg | urbanity_ordinal_avg | human_activity_index_avg | leisure_avg |
|---|---|---|---|---|---|---|---|---|
| HH | 58.638 | 61.615 | 9,772.255 | 5,315.243 | 119,409.772 | 4.524 | 39.7 | 3.817 |
| HL | 26.765 | 62.296 | 6,465.567 | 4,647.162 | 98,474.084 | 3.412 | 30.314 | 1.382 |
| LH | 9.632 | 40.138 | 4,400.032 | 2,408.86 | 270,734.844 | 3.838 | 18.744 | 1.721 |
| LL | 2.532 | 17.495 | 1,292.431 | 908.496 | 370,202.615 | 1.846 | 6.763 | 0.222 |

## Understanding the effect of external data on antenna's presence

The GWR component runs a Geographically Weighted Regression to understand the effect of the `nighttime_light` data, `population`, amount of `roads`' infrastructure and `pois` presence (feature variables) on mobile phone antenna's quantity (target variable, `cell_counts_joined`). Use a `gaussian` kernel of size `3` and click on `fit intercept`.




Understanding the effect of external data on antenna's presence

## Locating areas with a deficit of mobile phone antennas

In this last section, we would like to verify if the number of antennas is enough to satisfy demand in different locations while providing a high quality service. Lastly, we will identify potential buildings to install new antennas by enriching building data with the Getis Ord output. We will adjust the results by taking into account the area of the buildings as a proxy for placement availability:

- *Large buildings in hotspot areas will have higher positive values than small buildings*, as there is more space to place new antennas in demanding locations.
- *Small buildings in coldspot areas will have higher negative values than large buildings*, as these are areas with sufficient capabilities, so the smaller the building the less interested we are in installing new cell towers.

First, we will create a spatial score that combines the HAI with the antenna's count. Since we want to identify locations with relatively high human mobility and lack of antennas' information, we first need to *reverse* the cell tower data. We will use the Summarize component to get the MAX of the `cell_counts_joined`. Then we will use a Custom SQL Select to run the following query:

```
SELECT h3,
cell_counts_joined_max - cell_counts_joined AS cell_counts_inv,
human_activity_index
FROM `$a`, `$b`
```

Now, use the `Composite Score Unsupervised` procedure with the `CUSTOM_WEIGHTS` option to combine both variables using the same weights through a weighted average. Select `STANDARD_SCALER` as the scaling method and a `LINEAR` aggregation.

Then, compute the `Getis Ord` statistic on the derived `spatial_score` to identify statistically significant spatial clusters of high values (hot spots, lack of coverage) and low values (cold spots, sufficient coverage). Use a `uniform` kernel of size `1`.

To enrich the building geometries using the output of the Getis Ord drag and drop the `Enrich Polygons` component. Notice that we need to work with geometries here, so we will first get the boundaries of the Getis Ord H3 cells using the `H3 Boundary` component. Enrich the data by aggregating the `gi` value with the `AVG` and the `p_value`, that represents the significance of the statistic, with the `MAX`.

To finish with the analysis, use the `ST Area` component to get the area in square `meters` of each building. We will do the following:

- For buildings with a `gi >= 0` (hotspots), we will adjust the `gi` by multiplying it by the logarithm of the building area.
- For buildings with a `gi < 0` (coldspots), we will adjust the `gi` by multiplying it by the logarithm of the ratio between the maximum area of all coldspot buildings and the area of each building itself.

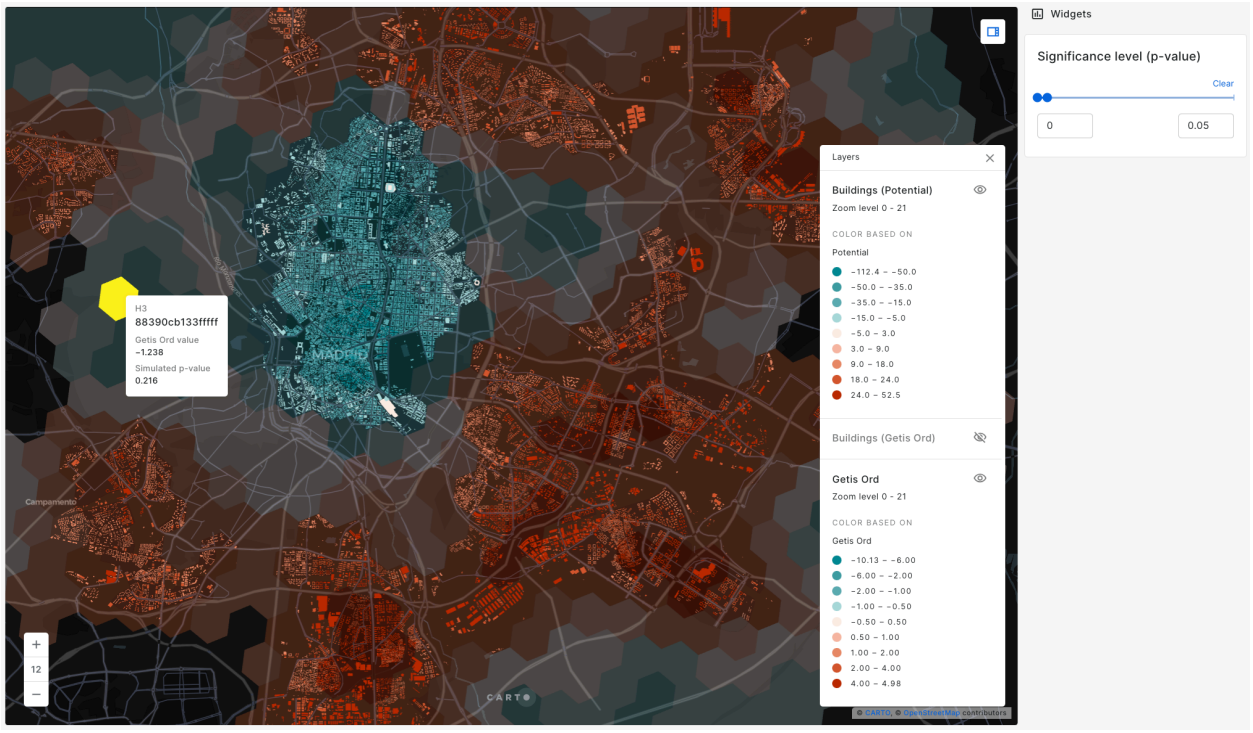
Use the `Where` filter (`gi_avg < 0`) followed by the `Summarize` component to get the `MAX geom_area` of all coldspot buildings. Then, run the following `Custom SQL Select`:

```
SELECT a.*,
   CASE WHEN gi_avg >= 0 THEN gi_avg * LOG(geom_area)
   ELSE gi_avg * LOG(geom_area_max/geom_area)
   END potential_score
FROM `$a` a , `$b`
```

To visualize the results correctly, we will use the `Create Tileset` component to create a tileset, which allows to process and visualize very large spatial datasets stored in BigQuery. Select the following parameters:

- Table details: `<project>.<dataset>.potential_buildings_madrid_tileset`
- Tileset mnemonic name: `potentialbuildings_madrid`
- Tileset description: `Aggregated tileset with all buildings in Madrid showing hot/cold spots for new mobile phone antennas placement`
- Minimum zoom level: `10`
- Maximum zoom level: `16`

Lastly, you can get notified when the workflow has finished by using the `Send by Email` component, which also saves a table to a bucket and shares it with you.




[Locating areas with a deficit of mobile phone antennas]

## Complete workflow: