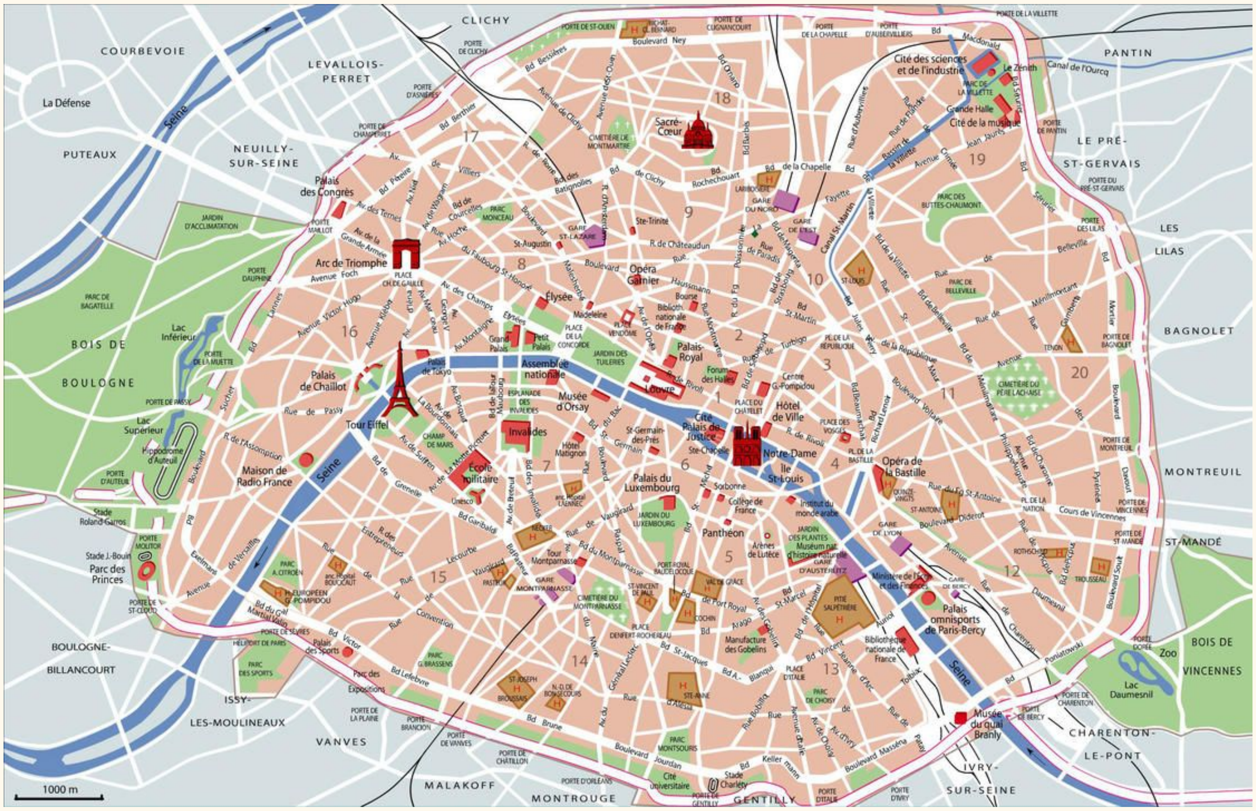
APPLIED DATA SCIENCE CAPSTONE PROJECT
FEBRUARY 2019

DATA SCIENCE CASE STUDY
RECOMMENDING NEW LOCATIONS FOR VELIB' STATIONS IN PARIS

By Lara Ramirez



INTRODUCTION

Vélib' is a large-scale bicycle sharing system in Paris.

Launched in July 2007, it was wildly successful and set a great example for the rest of the world to follow. It is now the largest bike share outside of China, with around 20,000 bikes spread over 1,200 stations, and a daily ridership of about 86,000 people.

The name *Vélib’* is a portmanteau of the French words *vélo* (“bicycle”) and *liberté* (“freedom”). The service aims to further the development of new forms of travel across the region that operate alongside existing transport options, especially polluting ones.

Beyond operating a good service, the most important aspect of the operation is choosing where to place the stations where people can hire and return the bikes, in order to maximise the chances that they will choose this type of transport.

The Paris Mayor has received a budget for 10 new stations to place in the city in 2019. His aim is to reduce areas that are commercially dense but don’t have a nearby station. He has asked for a recommendation of areas to target.

DATA DESCRIPTION

1. Paris boroughs

A geojson file of shape (20, 12), published in 2013.
Downloaded from the official Paris database.

Each row contains a borough’s following attributes:

1. **N_SQ_AR** Sequential ID
2. **C_AR** Number
3. **C_ARINSEE** INSEE Number
4. **L_AR** Name
5. **L_AROFF** Official Name
6. **N_SQ_CO** Sequential Code
7. **SURFACE** Surface
8. **PERIMETRE** Perimeter
9. **Geometry X Y** Coordinates
10. **Geometry** Complex coordinates
11. **OBJECTID** Object ID
12. **LONGUEUR** Length

2. Vélib’ stations

A geojson file of shape (1221, 6), published in 2019.
Downloaded from the official Paris database.

Each row contains a station’s following attributes:

1. **Identifiant de la station** ID
2. **Nom de la station** Name
3. **Capacité de la station** Capacity
4. **Lattitude** Latitude
5. **Longitude** Longitude
6. **Coordonnées géo[...]** Coordinates

3. Google Geocoding API

Used to get an address from coordinates (reverse geocoding).
Full documentation available here.

Required parameters for request:

1. **latlng** Latitude and longitude
2. **key** Application API key

JSON response:

1. **types[]** Type of result
2. **formatted_address** Postal address
3. **address_components[]** Separate components
    3.1. **types[]** Type of component
    3.2. **long_name** Full text description
    3.3. **short_name** Abbreviated textual name
4. **geometry** Latitude and longitude
5. **place_id** Unique ID

4. Foursquare Places API

Used to get nearby venues from coordinates. Full documentation available here.

Required parameters for request:

1. **ll** Latitude and longitude OR **near** String of a place in the world

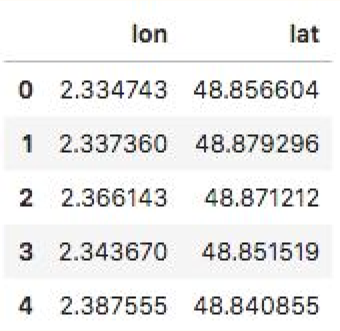
JSON response:

1. **id** Unique ID
2. **name** Best known name
3. **location** Separate components
    3.1. **address** Address
    3.2. **city** City
    3.3. **state** State
    3.4. **postalCode** Postcode
    3.5. **country** Country
    3.6. **lat** Latitude
    3.7. **lng** Longitude
    3.8. **distance** Distance
4. **categories** Categories

METHODOLOGY

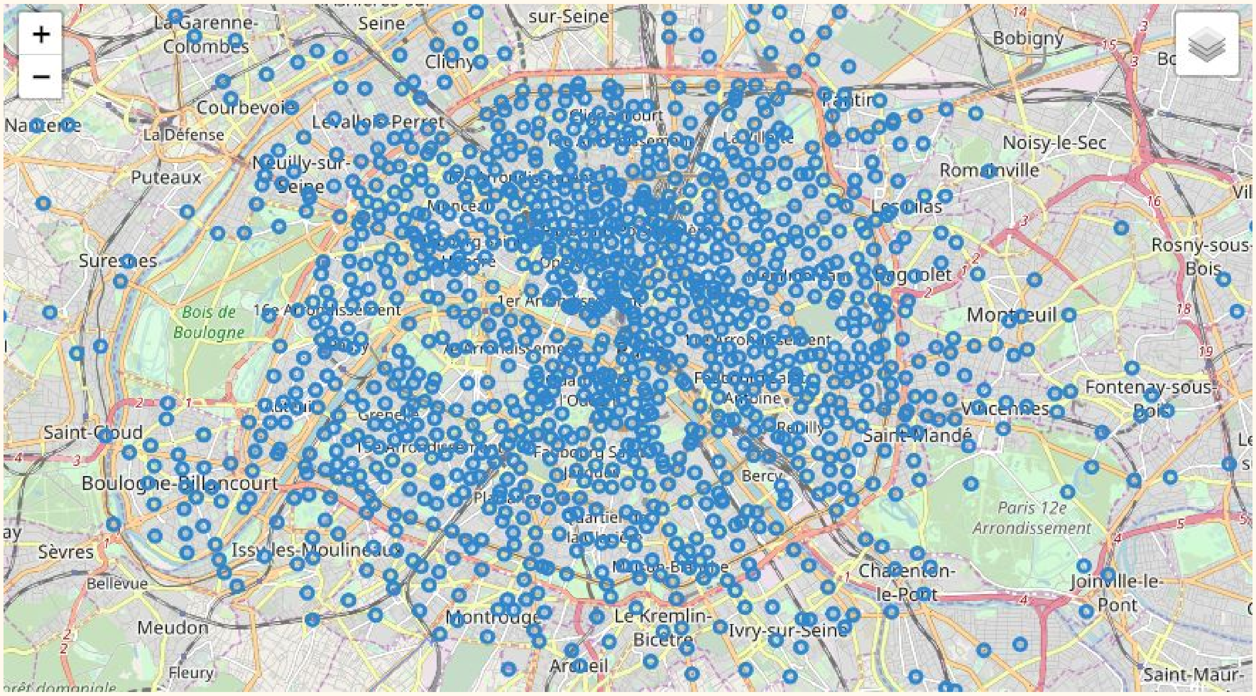
1. Mapping existing stations

I cleaned up the **Vélib' stations** database to keep only the coordinates of each station:

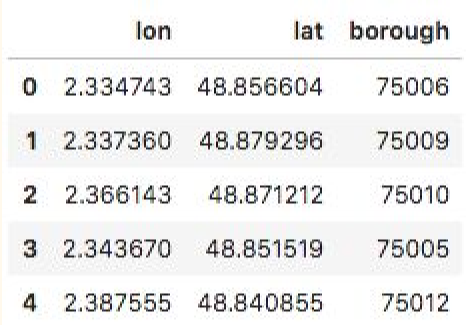


	lon	lat
0	2.334743	48.856604
1	2.337360	48.879296
2	2.366143	48.871212
3	2.343670	48.851519
4	2.387555	48.840855

Using this and the **Folium** library, I placed a 100-meter radius circle around each existing station, to visualize areas that we can consider as sufficiently covered:



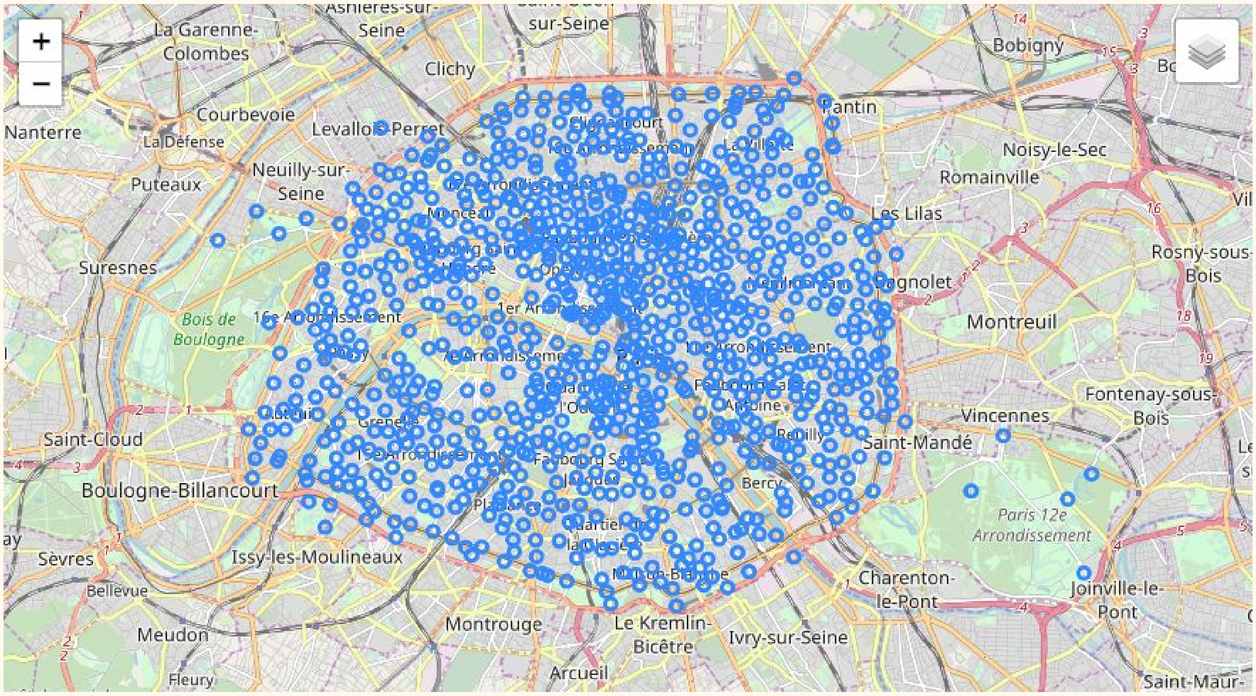
Seeing that the data also included stations outside of Paris walls, I added the borough by postcode to the stations data using the **Google Geocoding API**:



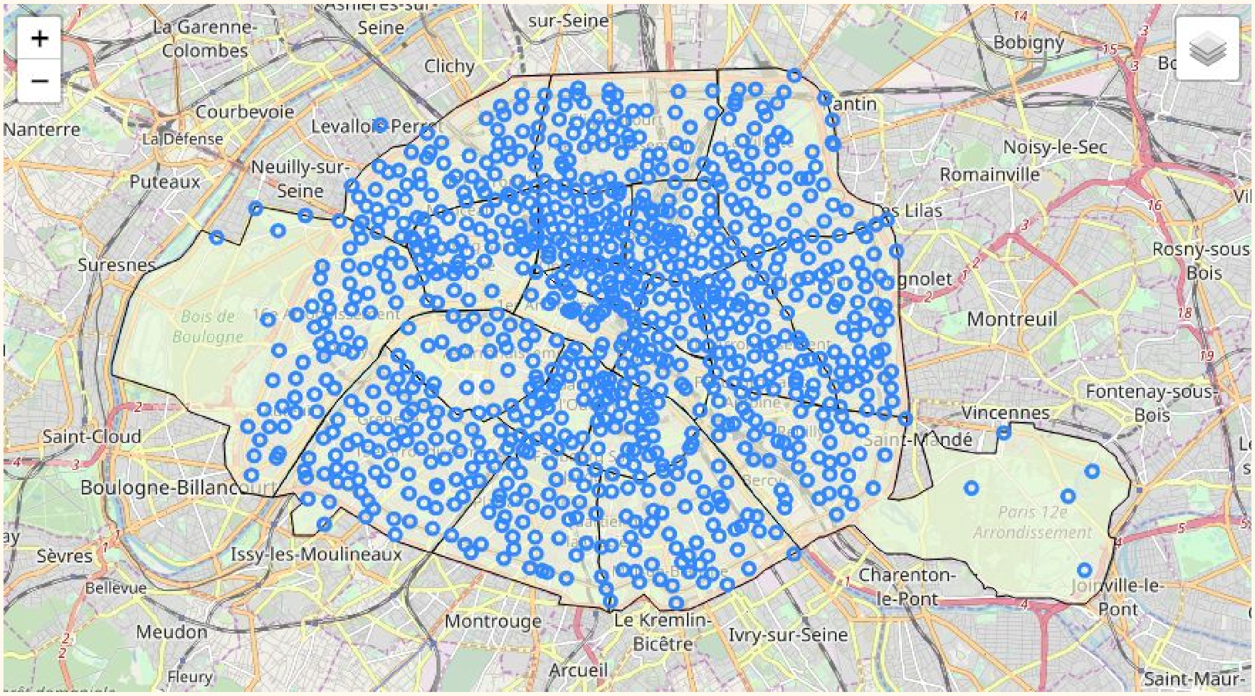
	lon	lat	borough
0	2.334743	48.856604	75006
1	2.337360	48.879296	75009
2	2.366143	48.871212	75010
3	2.343670	48.851519	75005
4	2.387555	48.840855	75012

Then used the borough as a filter to keep only Parisian locations (postcode starting with 75).

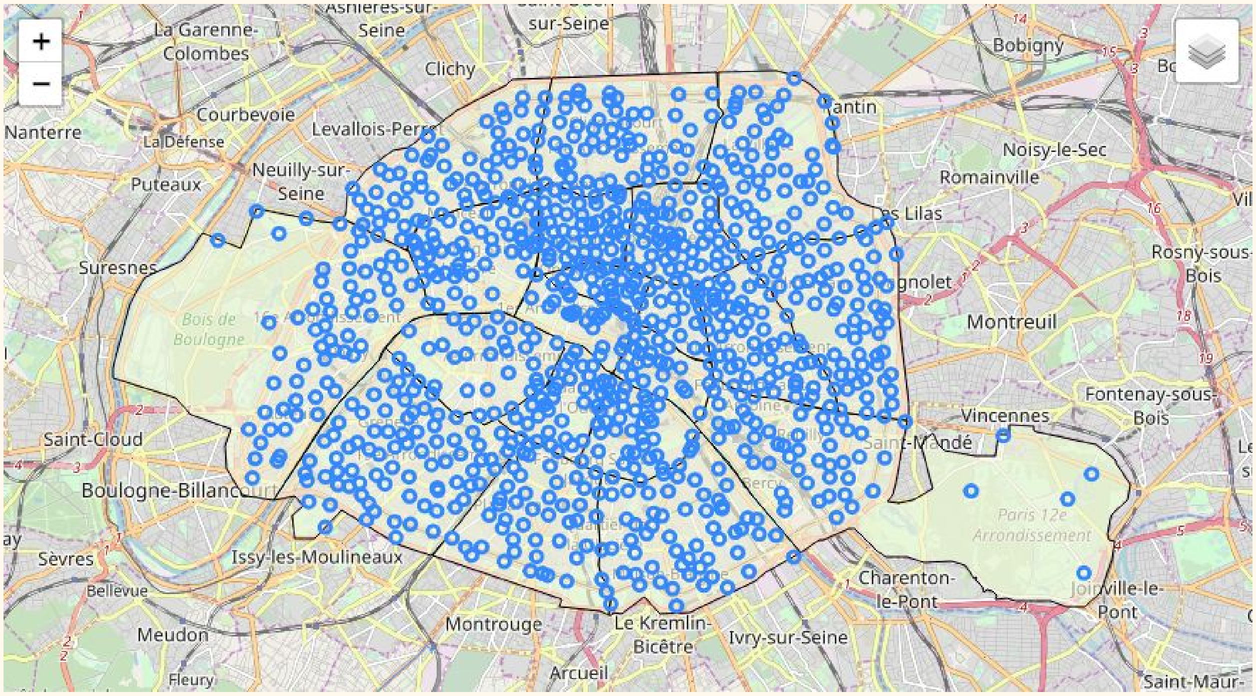
This allowed me to get a new visualisation of Parisian stations only:



Using the **Paris boroughs** dabase I added a neutral choropleth layer to better visualize the limits of the boroughs and of Paris itself, to double-check I didn't have any corrupt data:

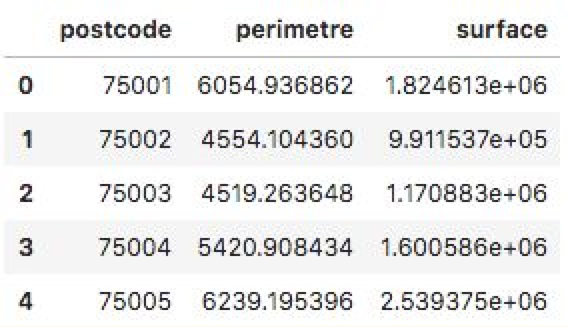


I individually removed the north-western outlier, to give me a clean final map:



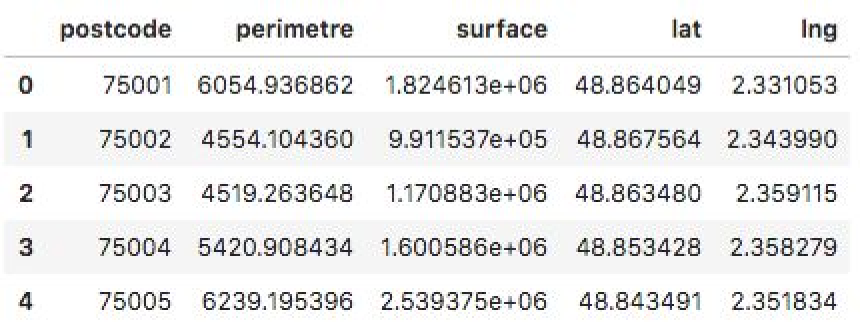
2. Mapping venue concentration

I cleaned up the **Paris boroughs** database to keep only the postcode, perimeter and surface:



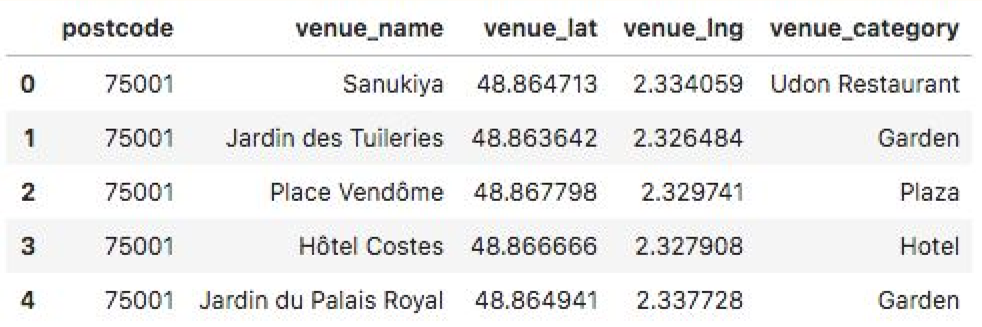
	postcode	perimetre	surface
0	75001	6054.936862	1.824613e+06
1	75002	4554.104360	9.911537e+05
2	75003	4519.263648	1.170883e+06
3	75004	5420.908434	1.600586e+06
4	75005	6239.195396	2.539375e+06

Using to the **Google Geocoding API**, I added coordinates corresponding to each postcode:



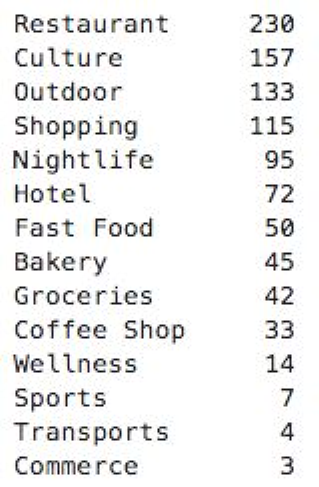
	postcode	perimetre	surface	lat	lng
0	75001	6054.936862	1.824613e+06	48.864049	2.331053
1	75002	4554.104360	9.911537e+05	48.867564	2.343990
2	75003	4519.263648	1.170883e+06	48.863480	2.359115
3	75004	5420.908434	1.600586e+06	48.853428	2.358279
4	75005	6239.195396	2.539375e+06	48.843491	2.351834

Using this and the **Foursquare Places API**, I built a separate database of nearby venues from the centre of each borough. I used the perimeter as radius and gathered the maximum of results which is 50 per request, giving me 1000 results total. For each result, I recorded the venue name, its coordinates and the category it belongs to:



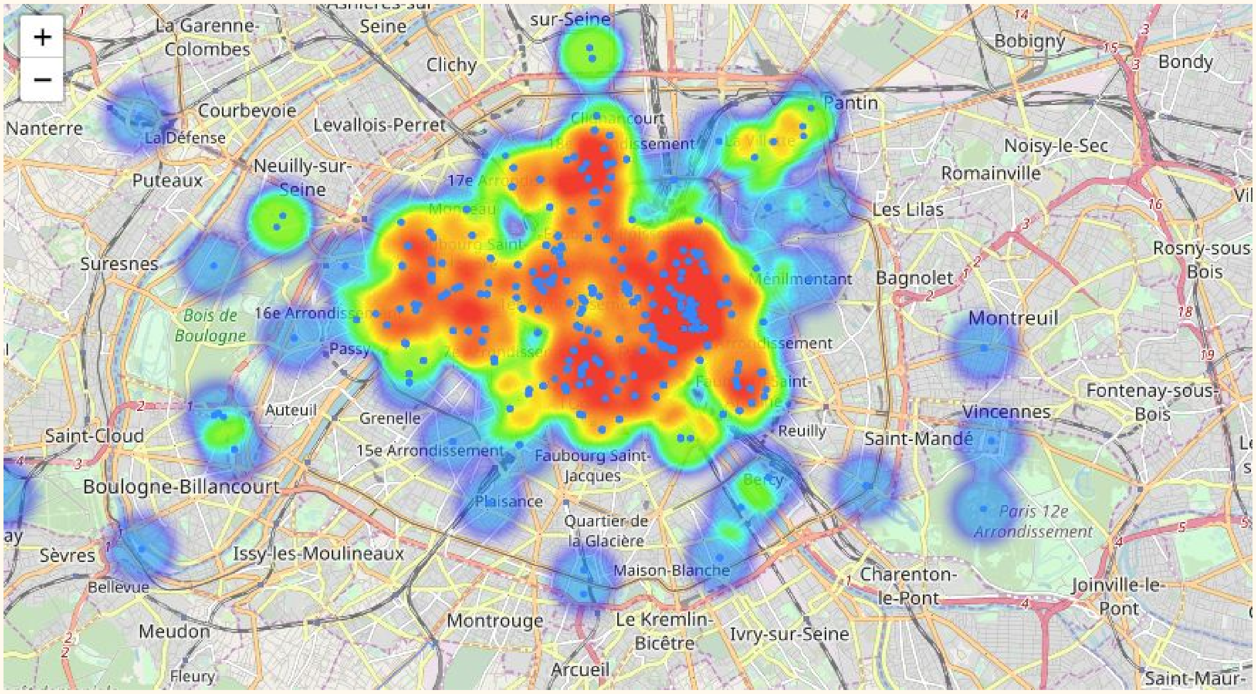
	postcode	venue_name	venue_lat	venue_lng	venue_category
0	75001	Sanukiya	48.864713	2.334059	Udon Restaurant
1	75001	Jardin des Tuileries	48.863642	2.326484	Garden
2	75001	Place Vendôme	48.867798	2.329741	Plaza
3	75001	Hôtel Costes	48.866666	2.327908	Hotel
4	75001	Jardin du Palais Royal	48.864941	2.337728	Garden

I had 89 unique categories in my resulting database, some of which could be regrouped. With some manual feature engineering I was able to reduce these to 14 main categories:

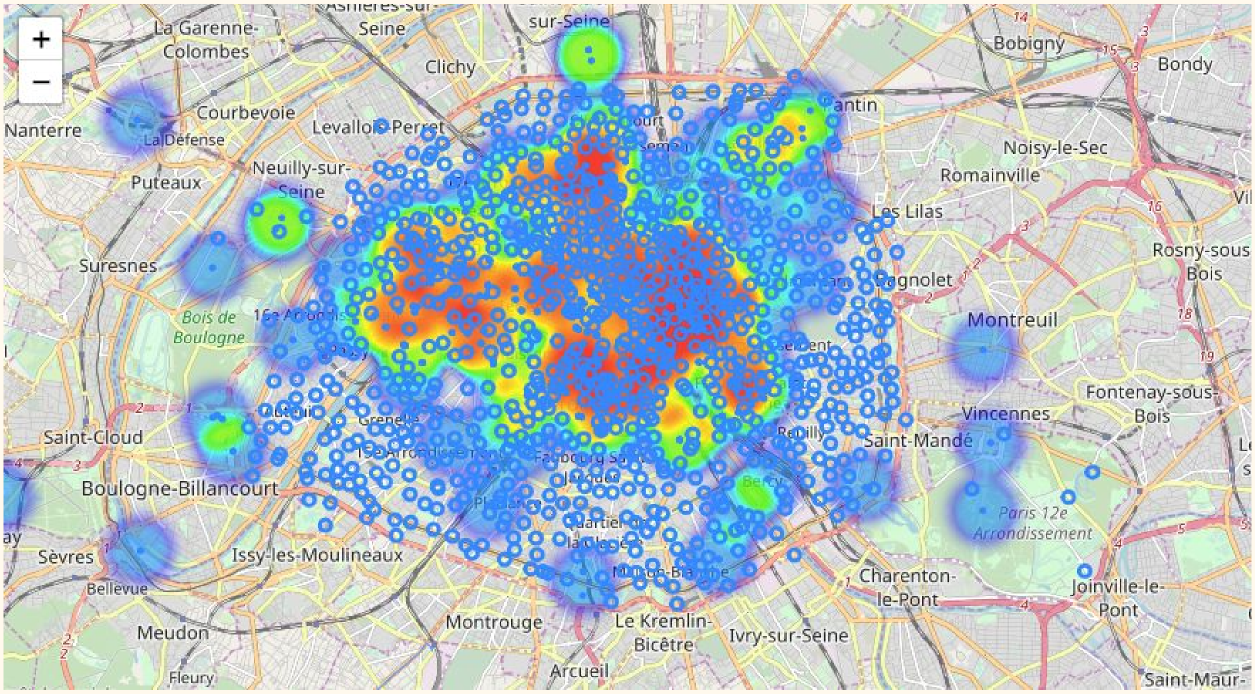


```
Restaurant     230
Culture        157
Outdoor        133
Shopping       115
Nightlife       95
Hotel           72
Fast Food       50
Bakery          45
Groceries       42
Coffee Shop     33
Wellness        14
Sports           7
Transports       4
Commerce         3
```

Mapping all venues as dots, I then created a heatmap layer to visualize concentration areas. Each dot is clickable to reveal the venue category for exploration.

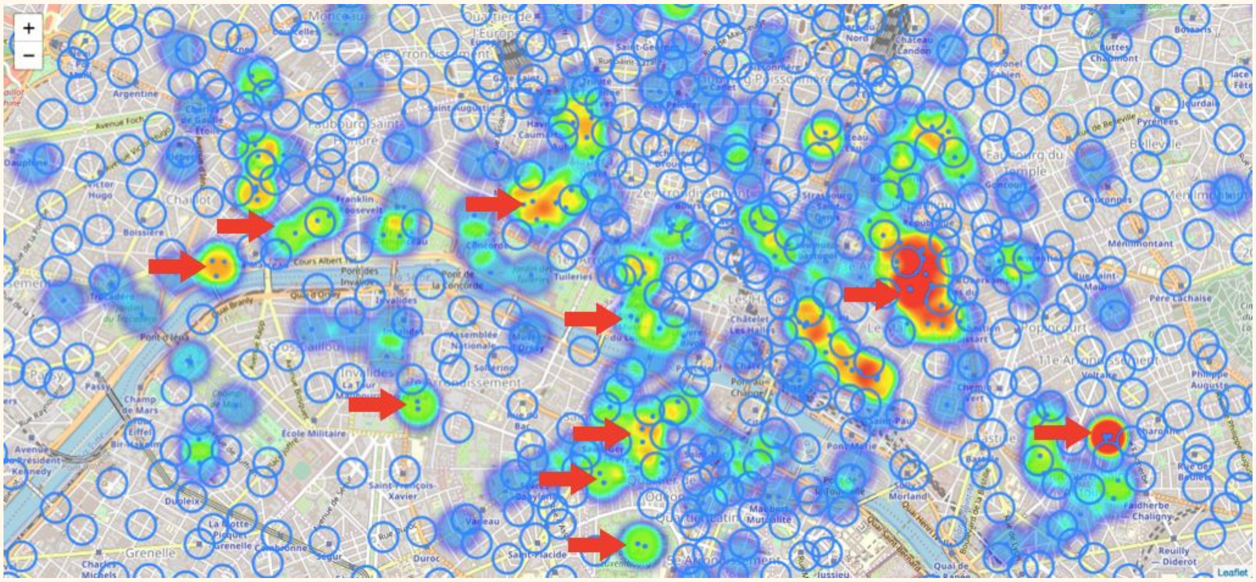


Adding our previous station areas onto this, I had a great visual tool to spot potential gaps between areas covered and commercially dense zones:



RESULTS

With a zoom on the reddest areas of the final map, it was possible to spot the 10 most commercially dense zones that were not sufficiently covered by Vélib’ stations:



DISCUSSION

Manual analysis of the data represented revealed an issue with the quality of the Foursquare data. Some important venues were omitted, and the limit of 50 results per requests - compared to the number of venues in Paris - has greatly skewed this analysis. The fact that the Foursquare data is crowd-sourced also questions its reliability. The use of a different API such as Google's Places would surely offer a much better analysis.

CONCLUSION

This visual analysis offers a valuable overview of which zones are covered by a 100-meter radius around each Vélib' station, but the quality and limits of the Foursquare data greatly impact the accuracy of the heatmap for analysis of commercial density. As this is also an important factor for the Mayor's decision, further analysis with a different data source is recommended.