A New Charging Station for Düsseldorf

Sven Spickermann

May 22, 2020

1 Introduction

While the idea of electrical cars as an environmentally advantageous alternative to cars driven by internal combustion engines is old, the appearance of Tesla on the car market has arguably accelerated this development.

For electrical cars to be a viable alternative to conventional cars, there is a need for a widespread infrastructure in the form of charging stations as well. The aim of this project is to take a closer look at the city of Düsseldorf in Germany and to determine -for a hypothetical provider of charging stations- where would be the best district to place one. In order to achieve this, I will first analyze the existing network of charging stations in the city and then determine gaps in this network along with a possible demand for charging stations.

2 Data

2.1 Sources

The data used for the project will mostly rely on Foursquare (`www.foursquare.com`), specifically on the "EV charging station" venue category.

The data necessary to attach GPS coordinates to the different districts of Düsseldorf will be sourced from freely available online resources such as the website `www.laengengrad-breitengrad.de` and OpenStreetMap (`www.openstreetmap.org`). Information on population density in Düsseldorf sorted by district was gathered from Wikipedia[1].

2.2 Data Cleaning

The data on electric vehicle charging stations were obtained via the Foursquare API by requesting all venues within a radius of 8.5 km around the city centre in order to cover the entire city area. This yielded five stations as a result, one of them outside of Düsseldorf, which was subsequently removed for the further processing steps. In order to check the accuracy and completeness of the data, OpenStreetMap was consulted.

[1] `https://de.wikipedia.org/wiki/Liste_der_Stadtteile_von_D%C3%BCsseldorf`

Additional charging stations were discovered with Nominatim on OpenStreetMap (`https://nominatim.openstreetmap.org`), by looking for the properties amenity=charging station and motorcar=yes. Therefore, the data taken from Foursquare was found to be incomplete and was amended.

It is possible that there are even more charging stations which are for electrical vehicles, but not expressly marked as such. Without further information/investigation, it is difficult to determine whether these charging stations are for bicycles or for electrical cars. If this was a project that an actual decision of a company depended on, one would naturally have to do more research at this point.

In the following, we shall limit ourselves to the stations which are clearly marked for electrical vehicles (i.e. cars) and manually enter the data for the additional stations marked as such. After including these stations, the resulting 15 entries were stored in a dataframe containing information on name, geometrical coordinates (latitude and longitude) as well as an as yet empty column for the district each station belonged to (see figure 1).

The district borders were obtained via download of OpenStreetMap relations from `nominatim.openstreetmap.org` after editing with the 'JOSM' editor in order to limit the information to the bare geographical district limits in the form of polygons. These polygons were then saved as geojson files in a dedicated directory. Düsseldorf has 50 districts, so there were 50 files as well. In terms of data cleaning, in the case of two of the resultant files, a superfluous point had to be removed from the geojson files in order to have a clean polygon.

3 Methodology

3.1 Exploratory Data Analysis

The geojson data were imported as shapely polygons into a dedicated dataframe. Next, the population density for each district was added to the dataframe. In order to get an idea of the population distribution within the city, a choropleth map was created (see figure 2).

As can be seen, more than 40 out of the 50 city districts of Düsseldorf have a population density of 7000 inhabitants per square kilometer or less and one district has by far the highest population density of more than double that number. That district is the central district of Friedrichstadt.

After these steps, it was concluded that there was enough data to determine the optimal placement for a new charging station. The following section will discuss the algorithm used to this end and its implementation.

	Name	Latitude	Longitude	City District
0	Stromtanke	51.2372	6.72549	NaN
1	Ladestation E-mobil NRW	51.2647	6.73449	NaN
2	Ladesäule Stadtwerke Düsseldorf	51.2226	6.81245	NaN
3	Parkhaus Kunsthalle	51.2273	6.77595	NaN
4	Stadtwerke Düsseldorf	51.2766	6.79034	NaN
5	Stadtwerke Düsseldorf	51.218	6.78237	NaN
6	EON	51.2354	6.77466	NaN
7	Mennekes	51.2211	6.82248	NaN
8	DB Energie GmbH	51.2211	6.79354	NaN
9	Stadtwerke Düsseldorf	51.239	6.78032	NaN
10	Parsevalstr	51.2788	6.78575	NaN
11	Stadtwerke Düsseldorf	51.2454	6.76943	NaN
12	Aldi Süd	51.2627	6.78089	NaN
13	Stadtwerke Düsseldorf	51.2139	6.77703	NaN
14	Stadtwerke Düsseldorf	51.1938	6.81292	NaN

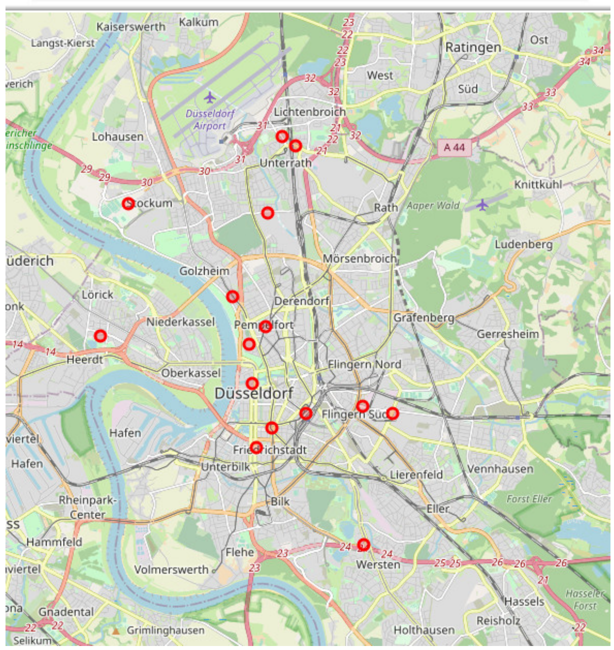

Figure 1: above: Dataframe with location data of stations. below: Map with station positions in Düsseldorf marked with red circles.

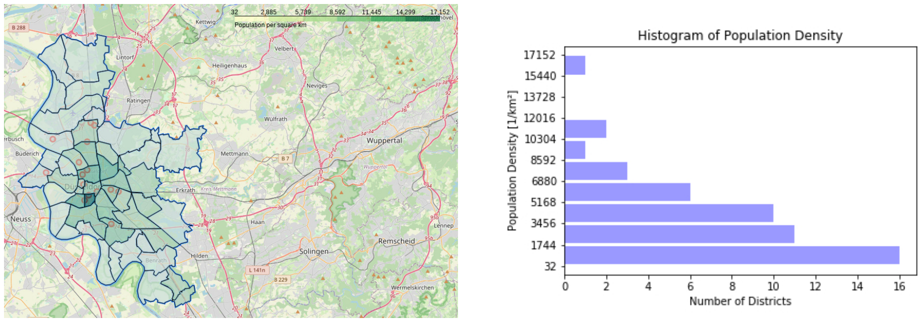

Figure 2: *Left:* Choropleth map of the city districts of Düsseldorf. Color intensity denotes population density. Charging stations marked red. *Right:* Histogram of population density in the districts.

3.2 Algorithm to Find Best Place for New Charging Station

3.2.1 Introduction and Basic Principle behind Algorithm

With the station coordinates and the population density for a each district available, we now needed an algorithm to determine the optimal place for a new charging station. The basic aim was to have an 'available population' number as large as possible within a 'radius of influence' around the new placement position. 'Available population' will be defined as population density divided by the number of charging stations, whose radius covers the position in question. The 'radius of influence' mentioned above shall be defined as half the average distance between charging stations which directly neighbor one another.

This model, as do all models, has some limitations based on the underlying assumptions. For example the number of electric vehicles per population may differ from district to district, depending for example on the income of the population. One might use the average rent in each district as a measure of this - if, on the other hand, the rent is higher, the maintenance costs for the hypothetical new charging stations would increase as well. For this reason it is not clear how this variable, even if we had a ready measure of it, should be implemented. However, these finer points are outside the scope of the model presented here and of the algorithm based on it.

3.2.2 Necessary Ingredients for the Algorithm

1. determining the radius of the 'circle of influence'
2. a function that gives the 'available population' within the circle of influence from a given point, a_p(pos, radius)
3. a grid of points over the entire city for which to apply the avai_pop function (dense enough for meaningful results, but not so dense that the algorithm

takes forever to compute)

4. determining the grid point(s) with the maximum number of available population, i.e. max(avai_pop) over all grid points from 4.

5. visualisation of the results

3.2.3 Determining the Available Population inside the Circle of Influence Without Area Calculations

A relatively simple approach to take is to first put a grid onto the circle area and for each point of this grid determine the population density of the respective district. Then in the end one can determine the average population density over all of these grid points to arrive at an average population density. This is similar to a Monte Carlo approach and likely much simpler than determining accurately the area of the resulting intersecting geometrical shapes of circles and polygons.

Overlap between different circles of influence can be taken into account by taking a circle around each grid point and counting the number of stations within its radius. This allows us to move from population density to 'available population density'.

3.2.4 Concrete Description of Algorithm

We need to define the function avai_pop(pos, radius), where the radius is constant and 'pos' is a set of coordinates. Within that function we need to use a station_counting_function depending on the radius, which loops over all the coordinates of the existing stations to determine whether they are inside the respective circles. It also maintains a station counter, which will come in handy later. We also need to determine for each grid point within the cricle around 'pos' the district it is located in and then its population density from a suitable dataframe, geo_df. The ratio of that population density and the counter result is then returned to the origin of the function call and gives the available population density for that grid point. In the end we determine the average over all the grid points in the circle around 'pos'. This is the return value of avai_pop(pos, radius).

3.3 Determining the Radius

An important quantity is the radius assigned to each station - in other words, how big is the 'area of influence' of one station and how big is the area of influence of the new station going to be?

Assuming that each person with an electrical vehicle attends to the nearest station available is certainly a logical approach, but this would lead to irregularly shaped areas of influence. It would also mean a change in areas for each new point being tested. This would likely drive up the calculation time by quite a bit. It is therefore not a practical assumption.

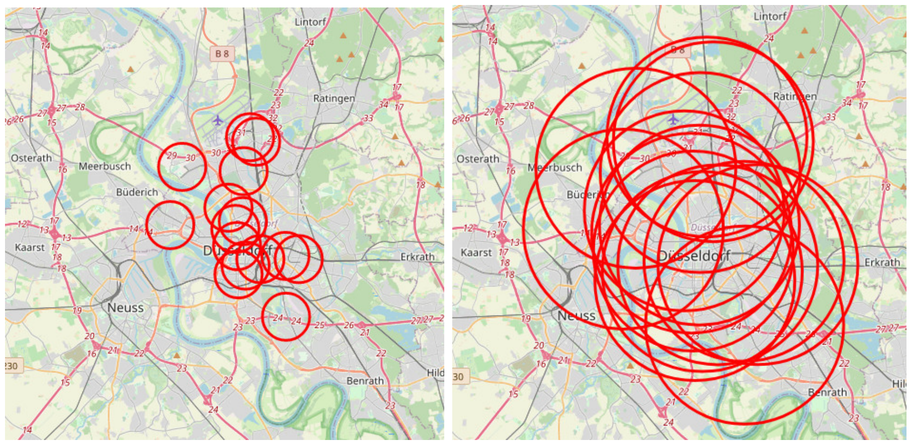

Figure 3: Circles of influence for two different radii. Left: The radius that was determined as the average distance of a station to its nearest neighbor. Right: A radius of 5 km

Instead, it is a more straightforward approach to determine an average radius by averaging over the minimal distances to the nearest neighbor. This radius will then be assigned to the new station as well and will be used as a parameter in our functions as a uniform value for all stations.

The average distance to the nearest neighbor was determined as ≈ 1.2 km. The two pictures in figure 3 illustrate the area of influence with this radius around each station. For comparison, the equivalent image for a case of a 5km radius is added as well. This goes to show the importance of the proper choice of radius for the outcome (which was incidentally also confirmed when the algorithm was run for a radius of 5 km, which returned a very different result, see next section).

One can see that in the latter case most of the points in the city would be covered by several stations whereas in the case with the average minimal distance between neighboring stations, this situation mostly occurs for the densely populated city centre.

3.4 Implementation of the Algorithm

The algorithm was implemented via a function (avai_pop) that determined for each point (pos) the 'available population density'. Available population density was defined as population density (taken from the district information dataframe) divided by the number of charging stations within radial distance to the grid point in question. The available population densities of all the grid points in the surroundings of 'pos' were then returned as a result by avai_pop.

In a final step, a loop was run over all the grid points within a given district, comparing the return of avai_pop for all of them in order to find the optimal

	City District	Geometry	Type	Population Density [1/km^2]	Polygon	Optimal Coordinates (lon,lat)	Available Population Density [1/km²]
0	Hafen	{'type': 'FeatureCollection', 'generator': 'JO...	<class 'geojson.feature.FeatureCollection'>	32	POLYGON ((6.730208 51.208949, 6.729931 51.2094...	[6.762843, 51.217879727272724]	2983.603604
1	Ludenberg	{'type': 'FeatureCollection', 'generator': 'JO...	<class 'geojson.feature.FeatureCollection'>	1077	POLYGON ((6.879068 51.261155, 6.878906 51.2604...	[6.832740421052632, 51.24189557142857]	4292.504505
2	Carlstadt	{'type': 'FeatureCollection', 'generator': 'JO...	<class 'geojson.feature.FeatureCollection'>	4804	POLYGON ((6.775664 51.225005, 6.775456 51.2249...	[6.76653125, 51.221308]	2725.098198
3	Holthausen	{'type': 'FeatureCollection', 'generator': 'JO...	<class 'geojson.feature.FeatureCollection'>	2276	POLYGON ((6.853443 51.17138, 6.853407 51.17129...	[6.8463386, 51.18835926315789]	4186.639640
4	Itter	{'type': 'FeatureCollection', 'generator': 'JO...	<class 'geojson.feature.FeatureCollection'>	749	POLYGON ((6.826055 51.157212, 6.823194 51.1611...	[6.811434714285714, 51.1737073125]	2074.171429

Figure 4: The first five rows of the dataframe containing the results for each city district. It contains the optimal position within this city district, which is the point with the maximum available population. It is noted in the rightmost column.

position for a new charging station in that district by choosing the point with the maximum value.

The results were entered into the district dataframe by it along with information on the average 'available population density' at the optimal point (see figure 4).

4 Results

The distribution of the available population density as defined earlier among the districts of Düsseldorf is shown in the choropleth map in figure 5. Comparing it to the choropleth map of the population density (fig 2), it becomes clear, that the inner city district of Friedrichstadt is no longer the most attractive location in terms of possible customer base, now that we have taken into account the placement of the existing stations.

From the aforementioned dataframe the maximum value of the available population density was determined along with its district. As a result, we found that the best point was in the district of Düsseltal at 6.81247° longitude and 51.241458° latitude, which is near its northern border with the district of Mörsenbroich. The available population density associated with this point as determined by the algorithm was 7265 people per square kilometer.

The map in figure 6 indicates the position of the hypothetical new station, marked with a yellow circle to differentiate it from the existing stations, which are marked as red circles. The underlying choropleth map illustrates the population density in each district. One can see that while the new station is not in a district with a low population density, it is not situated in the most populous districts either.

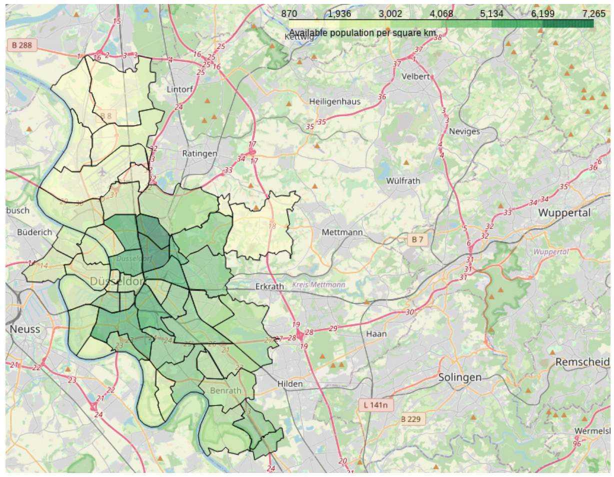

Figure 5: Choropleth map of the maximum available population density for each district of Düsseldorf.

5 Discussion

While there is no way to check the predicted best point against the objectively best point and to evaluate the algorithm in that way, one can still judge the outcome of the algorithm as at least plausible: It proposes a geometrical position which is at some distance from the other stations, yet still in a reasonably densely populated area of town. To put it differently, this result makes sense.

It is apparent that the choice of radius is an important quantity in the method presented here and has a big influence on the outcome of the algorithm. The geometrical distance between the optimal points for a radius of influence of 5 km and 1.2 km respectively was enormous - the best point was in a completely different part of town.

Within the confines of the approach of 'circles of influence' the method to determine the radius described here is probably the most realistic one. If you make the circles smaller, this results in a larger part of town not being covered by any station, which is unrealistic as customers there also have to recharge their vehicles somewhere. Conversely, if you make the area of influence bigger, you get more oberlap and the more overlap you get and the further away you are from the station under consideration, the less probable it becomes that somebody would charge their electric vehicle at this station. This would also make the approach unrealistic. The approach taken is therefore a good compromise.

Nevertheless, in future, one could consider using a more sophisticated approach where each point is assigned to the closest station and one would operate with more realistic areas of influence. On the other hand, for a more realistic

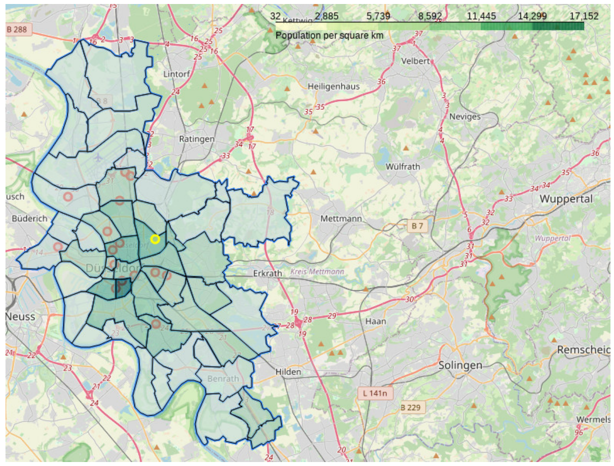

Figure 6: Choropleth map with information on the population density for each district along with the locations of the existing charging stations. The suggested new station position is marked with a yellow circle.

approach, other things should be factored in, such as the economic status of people in each district (as one measure of the likelihood of them owning or wanting to own an electric vehicle) as well as a more detailed investigation into the different types of charging stations (in terms of compatibility) and a more thorough investigation to determine an ultimate list of charging stations in the city area. As discussed in section 2.2, the Foursquare data in this regard unfortunately was wholly inadequate whereas the OpenStreetMap Data was more complete, but also far from unambiguous in this matter.

So while there is room for improvement in terms of data cleaning as well as modelling, this project provides valuable insights into a possible placement for a new charging station in the city of Duesseldorf. The approach could also easily be applied to a different city. All that is required is to import the district border polygons for the city in question along with information on the city's population density distribution by district and the placement of the existing charging stations.

6 Conclusion

A method was presented to determine the optimal placement of a new charging station for electrical vehicles in a city. It requires information on population density in the city districts as well as the positions of the existing charging stations. It then optimizes the placement of the new stations by maximizing the population contained in a certain circle of influence, whose radius depends on the density of the existing charging stations.

This method, applied to the city of Düsseldorf, was successful in determining an optimal placement for a new station and returned a plausible result. The method strives to achieve a good compromise between acceptable runtime on one hand (≈ 50 minutes in our example) and quality of output on the other.

In future, there are several possible ways the method could be improved upon, e.g. by refining some of the underlying assumptions.