Are we running out of cash?
Visualization Project: Analysis of ATM locations provided by Open Street Map

written by Moritz Blum
E-Mail: blum-moritz@t-online.de
for the course: E19 - Data Visualization [220192U001]

September 10, 2019

1 Relevance for a data Presentation and Analysis

Nowadays, many German newspapers are claiming about closures of bank branches and ATMs in countryside areas. They are saying it is creating problems for older people. But what is the situation, are there already many cases of this problem across the country or the worls? That is hard to investigate, since there is no service exist which provides information across all different banks, because every bank analyses their data on their own. Only maps contain this information, but do not offer additional analysis tools. At the moment, no public analysis tool is available, for this reason I will develop a data presentation to make it accessible and query-able through a visual interface.

I have chosen this dataset because I was working student at one of the largest financial technology companies for 3 years and on my last days I was working on a project to collect information about the existing ATMs in the world. The project was started because the company did not collect any data about where the sold ATMs are installed. One source of information was Open Street Map (OSM), I will use it in this project too, due to the fact that it is publicly available and I know how to query the data from the servers.

My goal is to develop a visualization that can be used to find dead spots and analyze their surrounding areas. It will be useful for critics of bank closures and banks to analyse the structure of the ATM network.

2 The dataset

The OSM map is simply a very large graph with a list of tags on its nodes and edges. It can be queried through Overpass to find all nodes matching a certain query, in this case all nodes containing a tag with the value "ATM" are searched.

These nodes always contain information about the location in the format of a tuple of floating point values (latitude, longitude). The values are in the range of -90 to 90 for the latitude and between -180 and 180 for the longitude.

In addition, there can be information about the operator as string, wheelchair accessibility as boolean or string, opening hours as string and more. But this information is definitely not clean because it is collected by an open source community and everyone is more or less free to set his own tags and values. E.g. bank names can be stored under different tags(Bank, bank, Operator) and one bank can have multiple similar names across all their ATMs(Sparkasse Bielefeld, sparkasse bielefeld, Kreissparkasse Bielefeld). The dataset is an approximately 35MB large .json file containing 160 thousand ATMs across the world. Of course, not all regions of the world are equally well integrated and it is Europe, where you find the most and accurate information.

3 How to visualize?

The visualization will contain a heatmap of ATM locations, where one can click on a certain location on the map and a histogram is created that shows the distances to the next ATMs. On the one hand, one can investigate spots which are rare of ATMs and on the other hand in which range they are reachable. One would probably find different shapes of the bar histogram depending on if the clicked location is located in a large city or in rural area.

One will be able to customize the heatmap by the color gradient, the radius of data points, and the intensity of each data point. An adjustable search radius is required, otherwise the number of examples would explode and the histogram would become unable to read. Furthermore, only ATMs in this certain area are interesting, and this varies depending on the location and must be due to this adjustable by hand. Another interaction element will be that the number of bins of the histogram are adjustable and that the distances can be scaled by a certain factor. A square root for example would work against the circle area formula and lets one prove different hypothesis.

Filtering the data by certain tags could give new insights, e.g. how the histogram change if only ATMs accessible to wheelchair users are considered.

An automatic outlier detection would also be interesting to find ATMs which are in areas, where not many ATMs are located. When selecting one element of this list, the associated histogram is shown. The great thing is that the visualization will be able to work with other spatial data as well.

As regards the group project, a combination with household income or restaurant data would be interesting.

4 Concerning the implementation

The visualization will be a web application provided through a python web server implementation using Flask. It will use Google Maps to render the heat map. I decided to use this client wise heat map rendering solution because it is a fast, simple and easy to extend. For the histogram I will also use a Google service, it's called google charts. A MySQL Database or SQLite Database is going to provide fast access to the data.

5 Steps until first presentation

1. Organizational: Query data, get Google Maps API Key, set up Flask Web Server with initial template

2. Data cleaning: e.g. inconsistency in the naming of tags and their values (only the location has always the same format, all other information vary depending on the author who added the data to OSM) with probably OpenRefine

3. Write the data to a database for quick access and simple management.

4. Get comfortable with the Google Maps API and implement the heatmap.

6 Risk assessment

This list contains things that can go wrong and how to handle these:

- One problem can be, that the dataset is too large to load or get handled in a web browser. This would mean that the rendering of the heatmap is too slow for a great usability. The solution could be a random sampling of data points on which the heatmap is based and doing the calculation of the histogram on the server side. Another solution would be to think about restricting the region to e.g. Europe or Germany. This should be no problem, because it is not necessary to look at all data at the same time.

- A different problem could be that the API of Google is too complex for such a small project and it would be too much work to get into it. In this case I would change to a simpler JavaScript charting library like e.g. Chart.js. Or I will use OpenLayers instead of Google Maps, because I have some experience there.

- If the histogram is not helpful or sufficient to investigate the data, I can change the graph. This should be simple since the charting libraries usually provide many different charts and the libraries should be to a certain level modular that charts can simply be exchanged