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

written by Moritz Blum (au640047)
for the course: E19 - Data Visualization [220192U001]

December 6, 2019

Abstract

Getting an insight into the financial industry is not that easy. With this project, the user is given the opportunity to examine what the current coverage of ATMs in Germany looks like. Therefore a Choropleth Map is used to show ATM counts normalized by the population, whereon the user can apply different filters. It turned out, that actually approximating the ATM utilization this way, enables users to assess the current coverage and to compare different parts of Germany.

1 Visualization Problem

Visual analysis of ATM locations is not a new field, but most of the work is done behind closed doors and not communicated to the public. First of all I give a description of the problem and describe what is out there so far.

1.1 Problem description

Nowadays, many German newspapers are claiming about closures of bank branches and ATMs in countryside areas. They are saying this is creating problems for older people. But this is hard to investigate, since there is no service which provides information across all different banks. Only maps contain this information, but these do not offer additional analysis tools. At the moment, no public anal-

ysis tool is available, for this reason this project develops a data presentation to make such data accessible and query-able through a visual interface.

There are two different views of the problem, the customer view versus the view of the financial institutions. For customers the distance to the next ATM is most important, whereas the operators mostly care about the degree of capacity utilisation. My focus is on the perspective of financial institutions and I will create an approximation of the real occupancy rate by visualizing how many people share one ATM. To do that the data about ATM locations is enriched with geospatial data and data about the population and their age.

All in all, the purpose of my visualization project is to enable the normal users the view of the operators and to follow their decisions.

1.2 About ATM locations

The visualization is about to show where many and where less many ATMS are located. Therefore geospatial locations of ATMs with unique identifier(context) with tags operator and wheelchair accessibility(content) are presented in a geospacial visualization. Considered is data across complete Germany combined with a map and population data.

1.3 Related Work

Currently there are mainly just services out there to find the next ATM like the Visa ATM Locator[1] or the Santander Branch & ATM Locator [2] which mark the locations on a map, provide some information and sometimes provide a functionality for filtering.

Other visualizations make use of hatmaps, e.g. this one, showing locations of compromised machines[3]. But the simple heatmap approach would not work for large data sets, since it will be extremely biased to the distribution of the population. Other possibilities are e.g. Choropleth Map, Star Map (Star Glyph in each region)[1] and Trellis Maps. In Chapter3 about my visualization solution, I explain which solution I took and why.

Completely different approaches are Predominance Maps [4] or Predominance Tag Maps[3]. They could show the the dominating operators, but in Germany

[1] `https://www.visa.com/atmlocator`
[2] `https://branchlocator.santander.com/`
[3] `https://bit.ly/2KQ8vMr`
[4] `https://www.arcgis.com/home/item.html?id=20ac5ee0812b49f893d4f1b769e10e29`

are special circumstances. The whole market is completely dominated by one operator who is everywhere present and only one other, who is way smaller, but still far larger than the rest. Doing such a visualization for the remaining operators would be based on too sparse and noisy data. Besides that, this would not solve the problem described in the previous section.

2 Process

After describing the underlying data set and the task, the data wrangling and the used tools are explained, before giving an overview about the components of the visualization.

2.1 Data Description

Getting a dataset from scientific source is a bit difficult, because most banks show their ATM locations for their costumers, but do not provide an API to query them. Therefore the decision was made to use OpenStreetMap[5](OSM) as data source, due to the fact that it includes the required information and is publicly available. 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 the tag "ATM" are searched. These nodes always contain information about the location in the format of a tuple of floating point values (latitude from -90 to 90, longitude from -180 to 180). In addition, there can be information about the operator as string, wheelchair accessibility as string, opening hours as string and more. 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. During the Data Wrangling the decision was made to only consider Germany, now the dataset is around 3 MB large, containing 21644ATMs and 67 different tags at the nodes.

Furthermore geospatial data and data about the population was needed. A GeoJSON file provided by Carto[6] contains geospatial information about different regions(Kreisebene: Landkreise & Kreisfreie Städte) in Germany. Geo-

[5] `https://www.openstreetmap.org/`

[6] `https://lznet.carto.com/tables/landkreise_deutschland_vereinfacht/public/map`

JSON[7] is a common format for encoding geographic data. The information I used out of this data set are the geographic shapes, IDs and the number of inhabitants of each region.

Statistics about the age of the population in each region can be queried from Destatis(Statistisches Bundesamt)[8]. This data is stored in five csv files, one for each age group (<17, 18-24, 25-44, 45-65, >65), in a table with the columns: "Schlüssel der Gebietseinheit"(ID), "Name der Gebietseinheit" (name of the region), "Bevölkerung in Prozent" (percentage of people in this age group).

2.2 Task Abstraction

The visualization should contain the following abstract tasks: Overview, Filter, Details, Comparison.

An Overview shows quantitative data about the ATM count. The user can Filter the ATMs which are shown in the overview by different features. The data shown in the Overview can be inspected in more detail or compared to another Overview with a different filter configuration.

Following the schema of the Faceted Task Characterization[4], the user task can be described as follows: The goal is to provide a visualization for Exploratory Analysis through navigation. The user should observe low-level characteristics of attribute relations and structural relations.

2.3 Data Acquisition and Wrangling

First of all, I figured out how to exactly query the OSM[9] data through Overpass Turbo[10]. It was queried in steps by sliding a window over the world to avoid server timeouts and asking for nodes with the tag "ATM". The process was done on the 12th September 2019 and took around 3 hours. Then the data of all queried windows was merged into a single JSON file.

After this I started the Data Wrangling. The data was 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

[7] https://geojson.org/

[8] https://www-genesis.destatis.de/gis/genView?GenMLURL=https://www-genesis.destatis.de/regatlas/AI002-2.xml&CONTEXT=REGATLAS01

[9] https://www.openstreetmap.de/

[10] https://overpass-turbo.eu/

similar names across all their ATMs(Sparkasse Bielefeld, sparkasse bielefeld, Kreissparkasse Bielefeld).

Before loading the data into OpenRefine[11], the merged JSON needed to be reformatted, because the structure was not read in correctly. One formatting issue was e.g. an array containing the Latitude(lon) and Latitude(lat) which needed to get split up into single entries.

The Data Wrangling task took much time, because many regular expressions and column joins were required. I recognized that the data is too large and too inconsistent to clean on my own, e.g. there were so many different bank names in different spellings and acronym, that some background knowledge about the financial sector in each country would be required. So I decided to only consider Germany, I have knowledge about. Now the dataset was around 3 MB large, containing 21644 ATMs and 67 different tags at the nodes. Then I started filtering for interesting tags. This was done manually, because there was inconsistency in the tag naming, too. Some example tags are: "wheelchair", "brand", "opening_hours", "operator", "website", "layer", "surveillance".

Then I started to establish consistent bank names for the bank for which I know that they belong together, e.g. ING or ING - Diba were named to ING. There was also inconsistency across regions, e.g. the Sparkasse in Bielefeld is called Sparkasse Bielefeld and the Sparkasse in Halle in called Kreissparkasse Halle. I tried clustering, but this was not very helpful, since each bank had way too many clusters. The most suitable solution was to search with regular expression for parts of the names and change them to a consistent naming. This needed to be done in multiple columns, since the tags were not consistent too. Then I merged everything together. The banks I differentiate are: Sparkasse, Volksbank, ING, Commerzbank, Deutsche Bank, Sparda-Bank, UBS, Postbank, DKB, Euronet, HypoVereinsbank and "other". For the other columns this was done in a similar way and the result was a table of the form: [ID] INTEGER, [OPERATOR] TEXT, [LAT] REAL, [LON] REAL, [WHEELCHAIR] TEXT.

I decided to show the ATM count on map regions and then correlate it with the population in each region. So I started searching for map data of Germany on Kreisebene that is annotated with data about the population. This data is documented by the government and publicly accessible in a GEOJSON format, but since this dataset contains every detail and further not required information, the file is way too large to handle in real time. There are ways to reduce the

[11] `https://openrefine.org/`

accuracy of GEOJSON files and therefore I found a simplified version provided by Carto[12].

To get the data about how many people are in each age group in each region, the population data from the map is multiplied with the percentages of the age group files.

It turned out that the data includes some outstanding regions, where the count of ATMs per person is unexpected high in comparison to other regions, these are not necessary errors in the data. Nonetheless, this distorts the colors of the Choropleth Map such that only a few regions have a high color saturation and the others are indistinguishable low colored. To enable the user to inspect the lower saturated areas as well, he can remove the outliers from the data. When doing so, a z-score outlier detection method is applied, to remove all regions from the data that have a z-score of greater than 3 when computing the colorscale.

2.4 Tools

A SQLite database for fast data access was set up and filled with the preprocessed data exported from OpenRefine. The map and population data is stored in two JSON files. Furthermore, I created an Python Flask[13] webserver with a interface returning a JSON response to certain query made by the visualization website. D3.js[14] is used to draw GEOJSON and Vega-Lite[15] to plot graphs.

3 Visualization Solution

The structure of my visualization follows Ben Shneidermans Visual Information Seeking Mantra, that says a useful starting point for designing interactive visualizations is based on overview first, zoom and filter, then details on demand[5]. In my visualization, a Choropleth Map gives the overview, the user can filter the ATMs according to different criteria on which the overview adjusts. Then the user can select multiple regions and get a detailed overview of the ATMs inside. Figure 1 shows how these components are put together.

[12] `https://lznet.carto.com/tables/landkreise_deutschland_vereinfacht/public/map`
[13] `https://www.palletsprojects.com/p/flask/`
[14] `https://d3js.org/`
[15] `https://vega.github.io/vega-lite/`

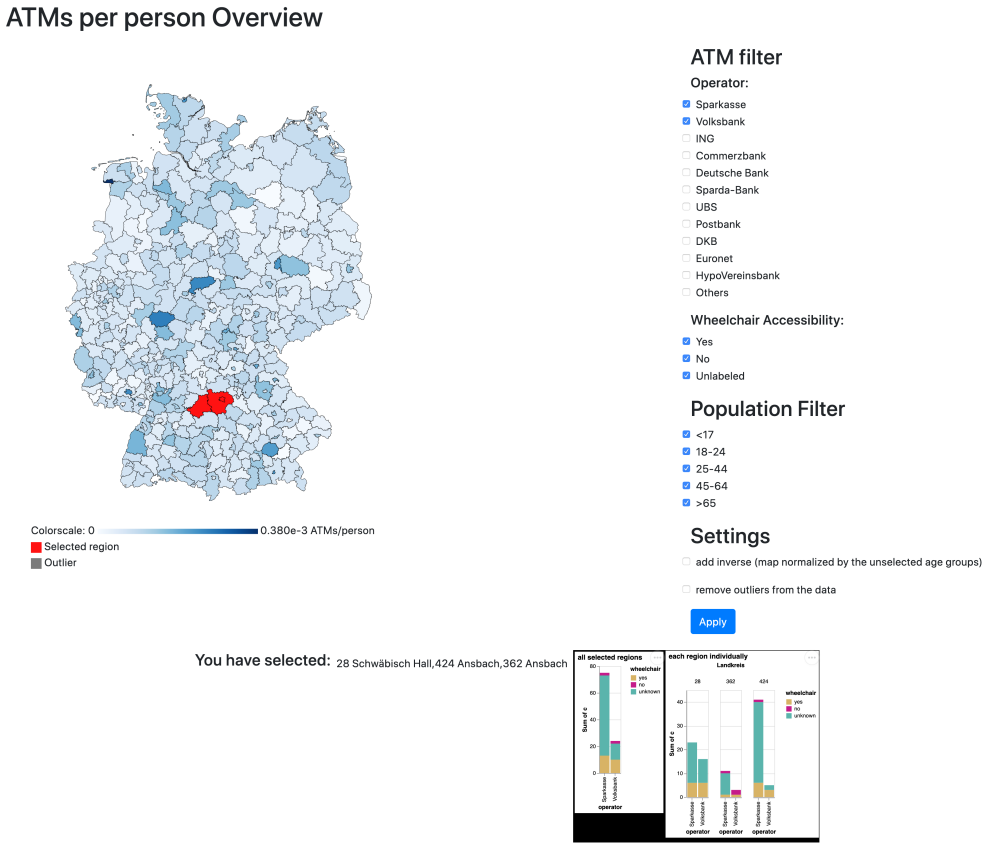


Figure 1: Visualization composition: Separation into Overview and Detail with a Filter at the right side. The user can interactively adapt the Filter and inspect regions of interest in detail.

3.1 Visualization Representation & Interactivity

I decided to use a Choropleth Map to show the number of ATMs in proportion the population. The first idea was to use a simple heatmap to visualize just the locations, but this is approach would not work for such large data sets, since it will extremely biased to the distribution of population, take a look at Figure 2a. So the idea came up to normalize by the population, but therefore the map needed to be discretised into parts, for which data about the population is available. As a result, the Choropleth Map shows Germany on Kreisebene on which the number of ATMs per person is mapped.

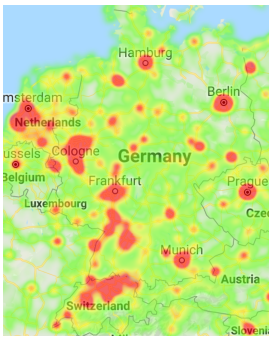
The filter is placed on the right side next to the map and gives the user two different options. The first option is the ability to filter for AMTs of certain operators or with a certain wheelchair accessibility. The second option is to restrict the population by which is normalized to certain age groups. By doing this, operators can adjust the settings to their target group to get their individual coverage, e.g. the Sparkasse has a target group that is older than 65 and required wheelchair accessibility (setting shown in the video).

To inspect the regions in detail, one can select multiple on them to get a detailed overview of the ATMs inside. In the Stacked Bar Chart the relative count is mapped onto the height of the bar and the different states of the wheelchair property are shown inside this by different colors. On the left side a summary of all selected regions is shown and on the right side a trellis plot shows each region individually.

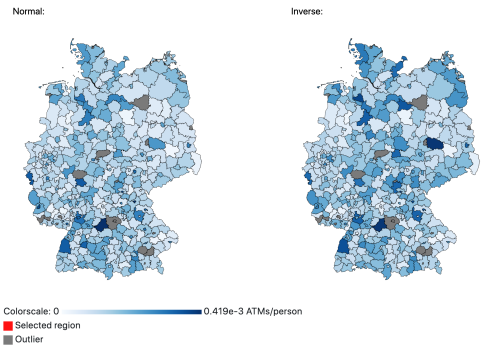
For the Choropleth Map, a continuous blue single hue color-scale is used (interpolateBlues[16]), to avoid the common used divergent green-red coloration, that often implies a valuation and required a neutral point of reference. The selected regions are highlighted in red and the removed outlier regions are grayed out. In the Stacked Bar Chart, a categorical color map from the Colorbrewer[17] is used to subdivide the bars.

In addition to the interactivity of overview first, zoom and filter, then details on demand, the user can remove outlier regions if he expects an advantage from it or he can display an inverse of the current settings, both shown in Figure 2b. In the inverse representation, the same ATM Filter is shown relative to the age groups that are not selected in the Population Filter.

[16] `https://github.com/d3/d3-scale-chromatic`
[17] `http://colorbrewer2.org`



(a) Heatmap without normalization: Looks similar to a heat map of the population and has therefore spots around every larger city.



(b) Statistical outliers are grayed out

Figure 2

3.2 Expressive and Effective

The colored Choropleth Map is expressive for my kind of spatial data, since it maintains the spacial component and can be normalized. The Stacked Bar Chart makes use of position and length which are the best features for quantitative data according to the ranking of perceptual tasks shown by Mackinlay[2].

The filtering optimally supports the task of investigating the map for regions that pop out or to find patterns. By comparing with the inverse, the user can identify differences to the not target group. E.g. if there is no difference, this could be an indicator, that there is not enough focus on the target group.

4 Discussion

The users can actually investigate and find outstanding regions and that works up to a certain point. My visualization indeed provides an analysis tool that can be made publicly available and used by everyone and different filters can adjust it to the preferences of the user.

The problem of possibly missing or wrong data falsify or distort the map colors. But one can assume that the OSM data will be updated constantly and will become more accurate over time and then the new data can be inserted in my visualization.

The limitations are that is it hard to understand what the "normalization" does, but it is necessary in some way and this seemed to be the best solution.

Furthermore, it is also a bit difficult to see differences in the inverse and to reason what follows from the these. But this tool is given to the user as an option and is not required to use.

A thing on the rendering side is that sowing the Stacked Bar Chart when hovering over regions would be great, but this either requires enormous preprocessing or an implementation in a faster programming language, but the idea came up too late for that.

5 Conclusion

To summarize everything, the visualization project makes it possible to make out regions with an above-average number of ATMs. I realized that a normalization to debias data can be difficult and make the resulting data hard to understand and interpret. Furthermore, I learned to visualize geospatial data with D3.js. It turned out, that the selection of color scales if multiple features make use of color is pretty difficult, since one has to find a configuration where each color is used only one time and is appropriate for its use case.

References

[1] Michael Friendly. A.-m. guerry's moral statistics of france: Challenges for multivariable spatial analysis. *Statistical Science - STAT SCI*, 22, 08 2007.

[2] Jock Mackinlay. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)*, 5(2):110–141, 1986.

[3] Martin Reckziegel, Muhammad Faisal Cheema, Gerik Scheuermann, and Stefan Jänicke. Predominance tag maps. *IEEE transactions on visualization and computer graphics*, 24(6):1893–1904, 2018.

[4] Hans-Jörg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann. A design space of visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2366–2375, 2013.

[5] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*, pages 336–343. IEEE, 1996.