Research and Innovation

Martí Municoy, Alba Gordó, Jan-Hendrik Niemann
and Andreas Radke

November 16, 2017

Abstract

TODO: modify the abstract s.t. it fits to the content The aim of this project is to find and visualize the events posted on Meetup.com in different cities in order to analyze the density of planned activities and to make a social study regarding typical meeting times and most usual activities and interests. One aspect may concern different meeting times in Spain and Germany or group size. We will focus our research on cultural activities like languages tandems and sports. The data shall be obtained via the Meetup-API and OpenStreetMap. The most important data source for this project is the Meetup-API where we have to request the events with different Python packages like "requests" or "meetup-api". Our second task is to visualize the locations of the events on a given city map obtained by data resources like OpenStreetMap. To plot the locations we could use "matplotlib" or similar Python plotting libraries. One further aspect could be to look for interest of twitter users to make recommends to activities nearby. The geotag of the tweets provides the central information for the search.

Contents

1 Summary and Goals

The human being is social. Every day there are millions of events. Some thousand of them are posted at the webpage `Meetup.com`. It is a website where people can seek and find other people with the same interest. It helps to organize events and makes them public. For this reason we can use `Meetup.com` to analyze the behavior and the preferences of cities and its inhabitants all over the world since almost all events posted on `Meetup.com` have a place which can be expressed as latitude and longitude. By using the request-API we extract a dataset "latitude-longitude" from `Meetup.com` and plot it on a map given by Google Maps. In this way we create a density map, a so-called heatmap, where on can see all the acitivities in the selected city. Furthermore, we also consider different types of events like "food & drinks" of "sports". Combining these information with the geographical and/or political structure of a city one may can identify trending districts and neighborhoods in a city. To do so we make use of GeoJSON-files which can contain the polygonal shape of a district and its name. There are several websites where one can get geojson data or create his own data.
A third data source is `Wikipedia.org` where we performe a web scrapping to extract the population number of a city and, if available, the polulation of a district. Having this information we create a factor to evaluate the degree of acitivity. This factor is given as

$$\text{degree of activity} = \frac{\#\text{events}}{\#\text{person}}.$$

Our data is available offline and online.

This work is structured as follows: In section 2 we discuss the data life-cycle of our data. In the section 3 we present our used methods and have a closer look at Gmaps-API, Python and Jupyther, GeoJSON and the webscrapping. Afterwards, in section 4 our results are presented. In section 5, we discuss legal and ethical issues. Finally, in section 6 we discuss the limitations of our work and make proposals for future work.

2 Data Lifecycle

In the following section we have a closer look at the terms "data lifecycle" and "data lifecycle management".
"Data lifecycle" is a term which tries to describe the life of data and information. Since there is no unique definition of a data lifecycle, we define the folloing six stages: creation, storage, use, sharing, archiving and destruction [7], [9].

1. **Creation**: Data is created. It can be created as a structured or unstructured set of data, can be obtained by collection, measurements, generated or gathered. We create our data by using the request-API. During pur process we gather the needed data from the `Meetup.com` website. By performing a web scraping at `Wikipedia.org` we gather the number of population for different cities and its districts. In this way we create our data which is needed for this project.

2. **Storage**: Once the data is created it has to be stored. Depending on the purpose there are different kinds of storage solutions. However, this is not subject of this report. In our case both the data gathered from `Meetup.com` and `Wikipedia.org` is saved a csv-files on the hard drive.

3. **Use**: Data can be viewed, modified analyzed, corrected, interpreted, visualized, joined,... We use our data the visualize the behaviour of cities. We join our data from different sources and visualize it on a given map which can be seen as data as well. After visualizing the data we interpret the newly created data to obtain information about the cities.

4. **Sharing**: Data can be shared. There are several ways of sharing like sharing with persons, applications or in different operating environments. Since our data is not shared during the the process, we are not explaining this stage of the data lifecycle in detail.

5. **Archiving**: When the data leaves the active use, one should archive it in a suitable way. Since archiving is very similar to storaging, one can use the same methods. One difference may be that saved data needs to be compressed to save storage or protected to prevent unwanted changed. Therefore, the access is slightly more difficult than for data in active use. For this project we save our data on the one hand as GeoJSON-files and on the other hand as csv-files. Whereas the data saved in the GeoJSON-files are not subject to changes, since

the geopolitical structure of cities does not change during this project, the csv-files underlie changes. For this reason there is a last stage in the data lifecycle.

6. **Destruction**: There are several reasons why one should destroy data. On the one hand the volume of archived data increases and one needs to save storage to store new data. On the other hand the data gets old and is not needed anymore. It is not up to date and can be replaced by newer versions. The second case is applicable for our work. During every single run of the program new csv-files with event data and new csv-files containing populations data are created. Especially the event data underlie a rapid change.

The term "data lifecycle management" describes the process that helps to organize the flow of data throughout its lifecycle.

3 Tools and Data

3.1 Tools

The main work for our project is done in the Python programming language in version 3.6 [8]. The most notable used Python libraries are requests [3], gmaps [6], Jupyter [4] and matplotlib [2]. The requests package is used to call and request data from the Meetup API [5].

With Jupyter Notebook and gmaps we are able to plot the obtained data from the Meetup API, Wikipedia and other resources in a browser. In detail Jupyter provides the general browser session functionality whereas gmaps extends Jupyter with an interactive map provided by Google Maps [1]. One can visualize specific points or areas in various colors and densities to create meaningful maps. Matplotlib is used to transform the data from population density into specific color schemes.

3.2 Data

As already noted we used the following data sources:

- Meetup API: To get the events in the examined cities.
- Wikipedia: For obtaining data about the population density of the cities and their districts.

- Various resources for GeoJSON files: To visualize districts onto Google Maps. TODO
- Google Maps: As a general back-end for our visualization.

4 Results

5 Legal and Ethical Issues

6 Limitations and Future Work

References

[1] Google Inc. Google Maps. `https://maps.google.com`, 2017. Interactive online Map Service.

[2] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[3] Kenneth Reitz. requests - Python HTTP for Humans, version 2.14.2. `https://pypi.python.org/pypi/requests`, 2017. Python Library for sending HTTP requests.

[4] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter Development Team. Jupyter notebooks – a publishing format for reproducible computational workflows. pages 87 – 90, 2016.

[5] Meetup Inc. Meetup API. `https://www.meetup.com/meetup_api/`, 2017. [Online; accessed 11-November-2017].

[6] Pascal Buignon. gmaps, version 0.70. `https://github.com/pbugnion/gmaps`, 2017. Plugin for including interactive Google maps in the Jupyter Notebook.

[7] Philipps Universität Marburg. Was versteht man unter dem Data Life Cycle / Daten-Lebenszyklus? `https://www.uni-marburg.de/projekte/forschungsdaten/faq/datalifecycle`, 2014. [Online; accessed 15-November-2017].

[8] Python Software Foundation. Python Language reference, version 3.6. https://python.org, 2017.

[9] Spirion LCC. What is Data Lifecycle Management? https://www.spirion.com/data-lifecycle-management/, 2017. [Online; accessed 15-November-2017].