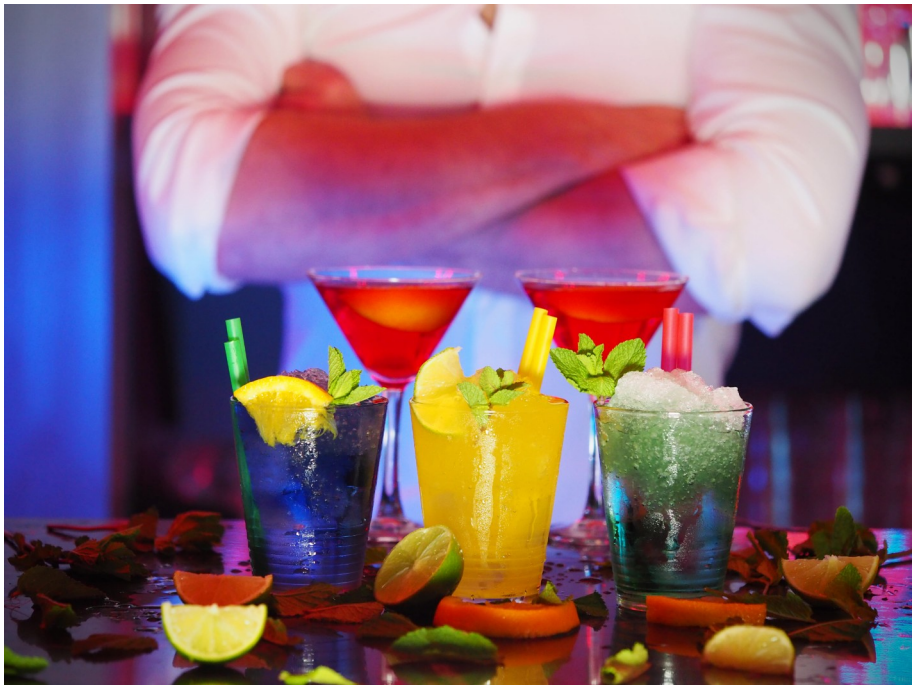


# *Battle of the neighborhoods*

Markus Gern – 04.02.2021

## Introduction

This analysis is the final project in the Course "*Applied Data Science*" by IBM. All recently learned new tools that are useful to solve the business problem will be applied here, like data acquiring, cleaning, sorting and visualization on a map.

## Business problem

Pub crawls exist in a lot of major cities by now. They are not the best way to get to know locals because mostly they are done by tourists, but they are an excellent way of getting to know the nightlife. If one wants to get to know the city by day, all tourist attraction are available, but the nightlife in a city makes up a significant part of its character, so not getting to know local venues and bars would be missing out a major part of a city's character. Pub crawls are mostly done by foot, so an important part is, that all venues are in walk-able distances. Everyone new to a city needs a map, so why not integrate the map with the best venues for a pub crawl?

The question to solve is: which city is best suited for a pub crawl, where are good venues in a close radius around city center? For this example the cities of Manhattan (New York), Toronto, Munich and Berlin are chosen for analysis, but the code shall be written universally.


## Target audience

There are multiple possible target audiences: first there are Hostels chains that can see in which city it is worth to invest. Second, activity providers like Tripadvisor can leverage this information to advertise a professional pub crawl. Lastly, tourists may pay a small amount of money for a convenient map of a city with all important venues for nightlife. Most cities give out free maps with touristy attractions, but what's missing there are the good pubs to visit in the evening.

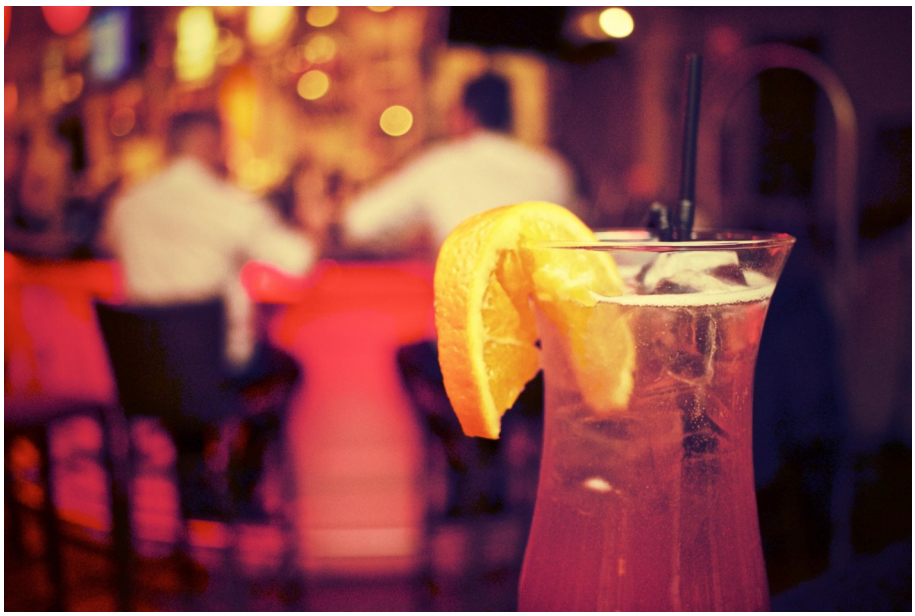
## Data acquisition

All data acquisition was done via the Foursquare API with multiple GET-calls. In a first call, all venues for a city within the desired radius had to be found. At first, the GET-request used the "search"-endpoint with the search-term "pub", but this yielded very poor results, as e.g. "public library" was also in the results. There is another far superior way of using the search by restricting the results to certain categories. All available categories can be found here [https://developer.foursquare.com/docs/build-with-foursquare/categories/]. With a search call for the CategoryID of '4bf58dd8d48988d116941735' only results withing the category of Bars (Nightlife) were given. This yielded perfectly clean results.

In a second call, the ratings of the venues had to be found. This is a premium call and limited to 400 calls a day, so as described later, a saver and loader function was written to minimize the API-calls.

## Methodology

As described earlier, the data was acquired with GET-calls to the Forusquare API. After that, the json formatted results were filtered and fitted in dataframes, with one dataframe for each city containing the ID, name, location and rating. To make the code univerally useful, a dictionary was used which had the citynames as key and the dictionaries as venues. With that dictionary, an iteration over all cities became very easy.



In development, the code was run multiple times, which was problematic, because for 50 venues for each city, one run of the program alone almost reached the daily limit of premium calls. Therefore a saver and loader function was written, which saved all keys in a txt-file and all dataframes in individual csv-files. This worked perfectly, therefore only one run of the program is necessary to save all the data locally. Only when new cities are added the API-Calls have to be run again by manually deleting the “save”-files.

With all data present, sorting the venues by rating for each city was only a matter of one line of code. To calculate the total score, the sum of the top 10 venues was calculated. For the highest ranking city, a map with all venues was plotted, with a marker for city center in addition.

| | id | venue_name | latitude | longitude | rating |
|---|---|---|---|---|---|
| 14 | 572f9097498ed768a7f4cca2 | The High | 48.133101 | 11.572939 | 9.2 |
| 3 | 4bbc6329afe1b7136d4d304b | Biergarten am Viktualienmarkt | 48.135194 | 11.576365 | 8.8 |
| 7 | 4cec06cb595cb1f7f16bdc14 | Weinhaus Neuner | 48.137694 | 11.568340 | 8.8 |
| 17 | 529cad3411d216970da64d58 | Déjà Bu? | 48.133001 | 11.576824 | 8.8 |
| 5 | 4ade0ca8f964a5205b6821e3 | Bar Centrale | 48.136821 | 11.579873 | 8.7 |
| 49 | 4bdcb658c79cc928ecc887e9 | Pusser's | 48.138500 | 11.580426 | 8.6 |
| 44 | 4b74556cf964a520ddd52de3 | Bar Tabacco | 48.139715 | 11.572672 | 8.4 |
| 35 | 568e8f9d498e8e46ab27027c | Herzog | 48.140279 | 11.569027 | 8.4 |
| 21 | 4ca1f27e46978cfaed37c07f | Madam Strip Club | 48.136821 | 11.579639 | 8.4 |
| 19 | 52913b2c11d264c5751010d2 | Frauen 26 | 48.134523 | 11.578873 | 8.4 |

*Figure 1: first ten venues of Munich, sorted by ranking*

Another advanced approach would be to use a route opimization algorithm to find the shortest route for all pubs. A great example is provided here: [https://openrouteservice.org/example-optimize-pub-crawl-with-ors/](https://openrouteservice.org/example-optimize-pub-crawl-with-ors/)

## Results

The results are that within the chosen cities of Toronto, Manhattan, Berlin and Munich, the ranking of the top 10 pubs in each city are:

(1) Munich with 86.5 points
(2) Berlin with 81.9 points
(3) Toronto with 81 points
(4) Manhattan with 34.8 points

This rating is valid for this date and may change as new venues open up and old venues close down. But still, a map for the highest ranking city (Munich) was plotted for convenient use for tourists, owner of hostels or an online provider of activities.

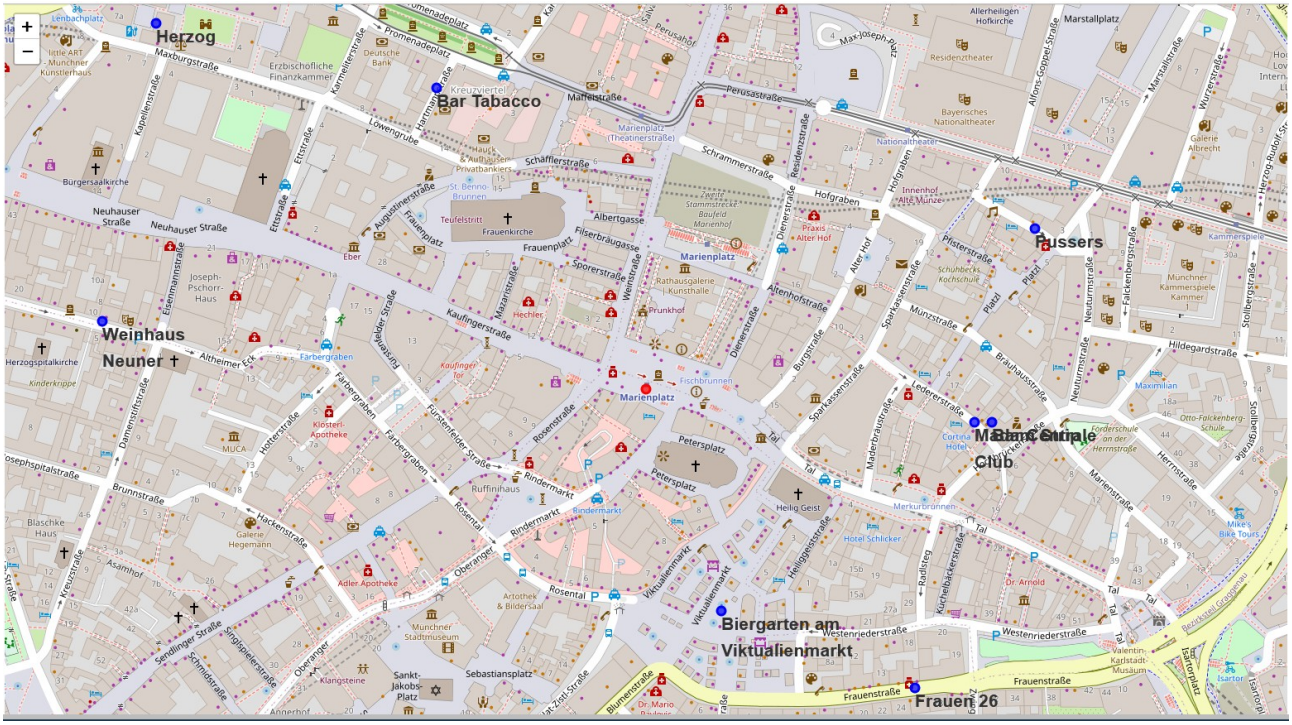



*Figure 2: the final map of Munich with its top 10 venues*

## Discussion

The quality and simplicity of the Foursquare API beyond the tutorial use was surprisingly good. Therefore, the results of this research can be trusted. In a side search request to the API it was found out, that among the top 20 of venues, one venue didn't exist anymore, so that might be the case for the here presented data, although it can be assumed that this should remain the exception.

One obvious weakness, also visible in Figure 2, is when two venues are very close horizontally to each other, because then the text overlaps. It was tried to turn the text 45° but the method of

“marker” in the folium-package and the DivIcon don’t support turning. An idea for further development would be to number the results and to give an external legend on the side of the map.

One main question is if this code is commercially applicable. The map data can be used, but one problem is, that the used Foursquare-account was a personal account. Foursquare states, that the data “is not available for commercial use, as Foursquare-data is not allowed to be commercially used unless the holder has a Start-Up or Enterprise Tier Account” [https://developer.foursquare.com/docs/usage-guidelines/]. Therefore the data and results presented in this report are free for personal use but not commercially. In the footnote, the license of this code is mentioned, which also prohibits commercial use. If there is interest in using of maps or the code, the author of this report may be contacted.

## Conclusion

In this report, the question of finding the best city for a pub crawl was answered, using techniques learned in the “*Applied Data Science*” by IBM. The results were little surprising, giving that a German city world famous for its beer festival won the rating. I hope this report provided you with some insight and may be useful for personal use.