# Data Collection

In Sweden, every address has a postal code. The postal code is unique to a neighbourhood and mainly used for mailmen and other logistics actors to deliver mail and packages. The postal code is also accompanied by a city name. Large cities and municipalities will have many postal codes and city names, in that case the city name can refer to a part of a city. Stockholm municipality has 27 such cities, with a total of 1605 postal codes. The area of a postal code is in general rather small. This enables a more detailed analysis of potential living areas in Stockholm. For examples, my current postal code, 417 57 Göteborg, only encompasses an area of 164 square metres, housing about 700 people. A map of this postal code can be seen in Figure 1, illustrating the area that each postal code represent.



Figure 1 A map showing postal code 417 57 in Gothenburg, Sweden.

The data needed for this project are the relevant postal codes, their coordinates and the nearby locations from the foursquare data base. The sources of data will be presented in the following section, along with some examples of what features the data contain.

## Data Sources

In this project, the necessary information was gathered from three main data sources. Two different methods were also used of collecting the data. Scraping was used in one to collect the data from one of the sources, while the other two sources were accessed through API calls. The three data sources are presented below.

### Postnummerservice (Postal Code Service)

The website postnummerservice, or Postal Code Service in English, was scraped using the Beautiful Soup package. Postnummerservice is a data base that houses postal codes, addresses etc. Sadly, they did not have a free to use API, which is why the web page was scraped for the cities within Stockholm municipality. Since very little information could be accessed for free on this web-page, only the names of the 27 cities in Stockholm municipality were acquired. Some examples of these city names are Bromma and Årsta. These city names were then used to acquire postal codes from the following data base. The scraped web-page can be accessed from the following link. https://www.postnummerservice.se/information/svenska-postnummer-och-postorter/stockholm/stockholm-kommun/

### PAP/Lite

I found a data base with a free personal API that contained the information needed. The API could be called using either a postal code or a city. For the free version, a very limited amount of 500 calls could be made every day. Since Stockholm municipality contains 1605 postal codes but only 27 cities, the latter version of the API call was used. The response contained up to 100 postal codes, which meant that for some cities, not all postal codes were returned. This was a limitation in the API that resulted in only 1111 out of the 1605 postal codes being collected. Four cities were affected by this: Stockholm, Bromma, Sundbyberg, and Solna. Most of the missing postal codes were located in central Stockholm. This was a shame, but since I would not be able to afford living in central Stockholm anyways, It does not affect the outcome of the study with regards to informing my relocation decision.

The response from the database came in CSV format, and included the following features: Postal code, city, latitude, longitude, county code, county, state code, state, notes, and date updated. However, only the following features were kept: Postal code, city, latitude, and longitude. The API can be found on: https://papilite.se/

### Foursquare

The foursquare API was used to identify nearby venues. The coordinates for each postal code were used as inputs to the API call. The limit was set to 20 venues and the radius was set to 500 meters, as this was deemed to be convenient walking distance. The response from foursquare contains a large number of features, but the following ones were kept: name of venue, venue latitude, venue longitude, and category of venue. Since calling foursquare with all of the 1111 postal codes took a long time, some lines of codes were also written so that the API was only called if a response had not been cached. This reduced the runtime drastically.