Geocoding

Address geocoding, or just geocoding, is the process of converting a location description (usually an address) into some form of geographic representation (usually latitude and longitude coordinates). A geocoding software or program will parse input data into standard, recognizable values, compare this information to an internal reference database of points, lines, or polygons, and then return the best match of values in the database to the input data. We use geocoding software regularly in everyday life – anytime you look up directions to a friends house, search for nearby restaurants for dinner, or hail a ride-sharing service, a geocoding tool is operating under the hood to provide the information you need.

When you need to geocode many addresses, which we often need to do when working with large datasets, this is called 'batch' geocoding. Many services offer batch geocoding, with varying degrees of speed, accuracy, and pricing. We recommend reviewing multiple services before deciding what is right for you and your team. We have provided a handout with comparisons of some geocoding services you might be interested in.

There are a variety of geocoding services available on the internet ranging from free to very expensive. The table below contains a list of some of the available geocoding services.

There are several important questions to consider when choosing a geocoding services, including:

Is it free? If not, what kind of subscription packages do they offer?

Is caching or locally/permanently storing geocoding results permitted?

Is using geocoding results with third-party basemaps permitted?

Does service have the right to store your geocoding requests and results and transmit these data to third parties?

What is their privacy and data protection policy?

Geocoder
ArcGIS World Geocoding Service
ArcGIS StreetMap Premium
Bing Maps
Census Geocoder
Geoapify
Geocode Earth
Geocodio
Google Maps
HERE
LocationIQ
Mapbox
MapQuest
OpenCage
TomTom

In order to access a geocoding service, you will have to interact with its application program interface (API). Some services, like ArcGIS, have interfaces that are internal to the software you might download for your computer. Others like Google, or Nominatum, require some programming to access. We utilize R packages to access APIs from within the R environment. We will walk you through this process using Google's Geocoding API as an example.

Getting your Data in Shape

In order for any batch geocoding process to run smoothly, we need to first get the data we want to be geocoded into the appropriate format to be processed. While each service may have different preferences for how the address data are formatted, they generally require all the address information combined into a single, string (text) variable. Some prefer commas separating different features of the string (Ellicott City, Maryland, 21042). If you are using R to clean your data, there are multiple functions in the `stringr` package that will prove useful.

When preparing your data, watch out for naming idiosyncrasies in your dataset that may confuse the geocoder and lead to mismatches or failed attempts. In the Massachusetts mortality data, for example, addresses often used shorthand for the street designation (road = RD, path = PA, CT = court). To a human reader, that is rarely a problem, but Google's Geocoding API often misidentified addresses ending in PA as being in Pennsylvania, and similarly identified CT as Connecticut. Several "drives" (DR) it struggled to identify it marked as doctor's offices (!). By reviewing and editing your data before running your geocoding program, you can avoid having to run it twice.

Setting up the Service

Depending on what geocoding service you use, you may have to take several steps gain access to the service. Google, not unlike other services, requires you to register for an API key through the Google Cloud Console. These API keys allow services to track who is using them (don't share yours!), and charge for services. Once you have the appropriate API key, you can save it to your R environment. This is recommended, as opposed to calling it in your code, as you may wish to share code without sharing your unique key.

Geocoding

Once your data are set up correctly, and you are credentialed to utilize the geocoding service, it's time to geocode your data. Here is an example of code from the package 'ggmap' which supports the Google geocoding API. Depending on the quality of your internet service, whether or not your service allows you to cache results, and how many items you have to geocode, you may want to split up the geocoding into smaller batches so that if it is interrupted and you have to start over, you aren't starting over from scratch.

Where possible, request as much output from your program as possible. In addition to latitude and longitude data, you can often receive information on the confidence of the program in the match, the precision of the match, additional geographic data, or other useful bits of information. And, importantly, when using services that cost money per request, be sure to save your results. If you find mistakes later, you can mark them, separate them from the main file, fix them, and rerun the geocoding service on the smaller sample.

Checking your Results

Once you've geocoded your results, it is important to check for accuracy. Some programs can tell you about the confidence of the match, and you can choose to review matches below a certain threshold to verify their accuracy. You can broadly verify the results by mapping the points to ensure things look appropriate. At a broad scale, like the premature mortality data, it may be hard to note small unexpected patterns in a sea of dots, but obvious issues (such as clusters of points being in a different state) can be identified using this method. You can also use some of the helpful output from the geocoding program. For example, Google offers a variable called "type" which tells you what kind of building the point represents. If you are mapping residences, it may be good to explore when some addresses come back as "electronics_store" or "dentist".

As you go through this process, be mindful of your source data. Addresses for our Massachusetts mortality data, for example, are recorded by human beings. If your geocoding service struggles to find a match, it may help to verify that the street name is spelled correctly or has the proper designation. One check we used, for example, was to verify that the address Google returned had the same zip code as the address submitted. Large clusters of data where the zip code was wrong may reveal a common misspelling.

The key to this process is to not nitpick every small mistake the geocoding service makes, but to identify broad issues that might seriously bias your results.