Geocoding Exercise using open source routing

for when you can't afford a trip to Europe

Associate Professor Malcolm Campbell

3/02/2023

Geocoding using Open Street Map

This short document shows us how to read in some data, geocode it, then map it. Cool, or what? First, load in the library(s) we need to get it all up and running

```
library(leaflet)
```

```
## Warning: package 'leaflet' was built under R version 4.2.3
```

```
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.2.3
```

```
## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(tidygeocoder)
```

```
## Warning: package 'tidygeocoder' was built under R version 4.2.3
```

How do we get some 'random' data and then use the Open StreetMap (OSM) geocoder to turn it into useful maps for us? First, have a quick read of the key function we need to use here https://rdrr.io/cran/tmaptools/man/geocode_OSM.html or try typing ? with the command ("?geocode_OSM") into the Console with of R. We can now use the function to geocode and map some data. Lets imagine we have data on the locations we need. See the source of the data here https://www.reddit.com/r/MapPorn/comments/105wkwy/when_you_cant_afford_a_trip_to_europe/

To make life a little easier, this data is found in the file "RoutingLocations.csv". What R allows us to do is to turn place names, for example, "London, Ontario, Canada" into a latitude and longitude. It also helps us to see why including all the information in a query is important. There is more than one London.

```
Routes <- read.csv("https://raw.githubusercontent.com/malcolmcampbell/RSpatialBasics/master/Geocoding/R
```

So after reading in the data, we want to being geocoding.

```
geocodedroutes <- tidygeocoder::geo(Routes$Locations, method = "osm") %>%
  st_as_sf(coords = c("long", "lat"), crs = 4326)
```

```
## Passing 9 addresses to the Nominatim single address geocoder
```

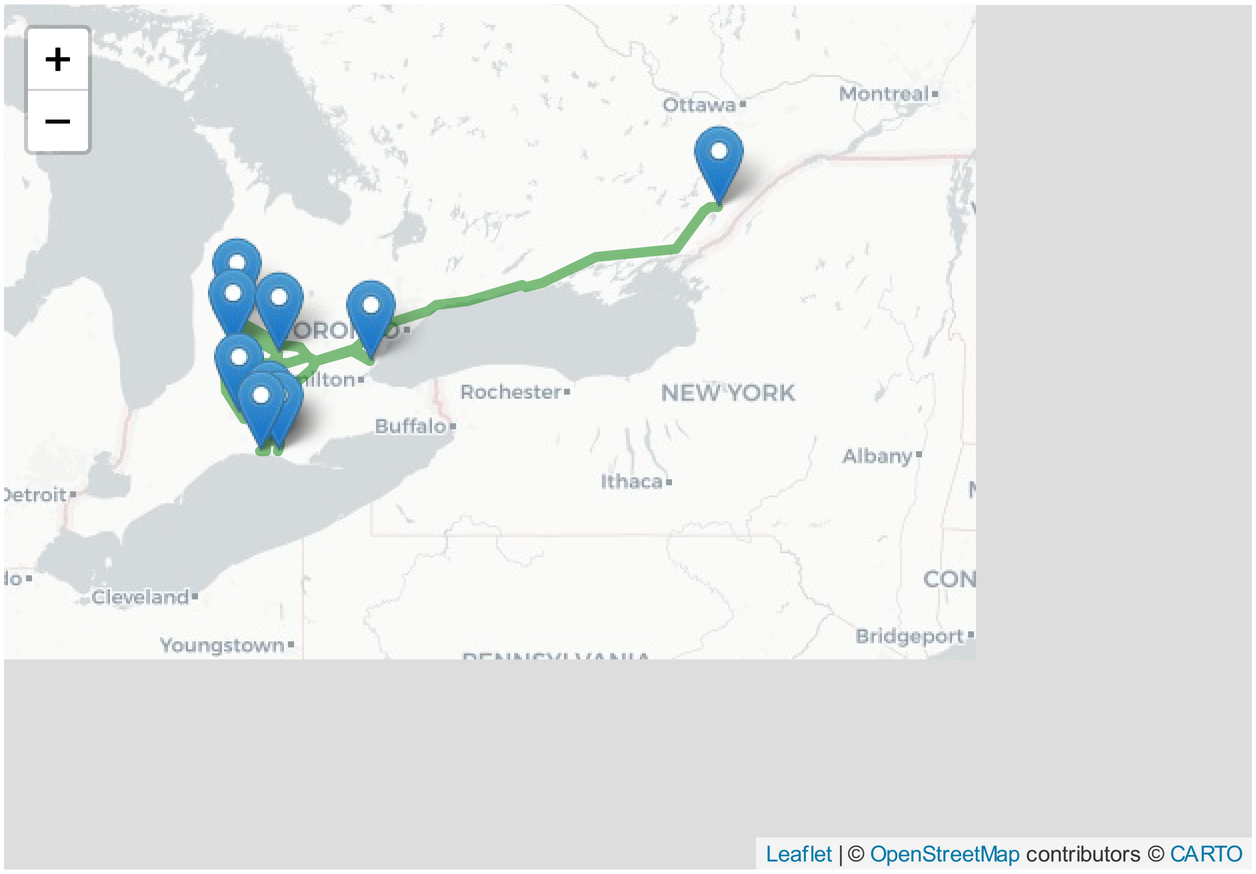
```
## Query completed in: 9.4 seconds
```

You should see a series of messages letting you know that the geocoding is working using the Open Street Map (osm) method. Next, we want to make a route to visit all of the european cities by driving, in Canada. Lets call this 'EuroCanroute' and create it using the command 'osrmRoute' which will build and the send an OSRM API query, getting travel geometry between our points.

```
EuroCanroute <- osrm::osrmRoute(loc = geocodedroutes)
```

Now for the fun bit, drawing some maps. This time, we will use leaflet to draw the maps. The route ('addPolylines') is in green so we can see it, with markers ('addMarkers') in each city to visit.

```
leaflet ( data = geocodedroutes ) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addMarkers(label = ~address) %>%
  addPolylines(data = EuroCanroute,
               label = "Europe on a budget route",
               color = "green")
```


+
−
Ottawa
Montreal
TORONTO
Hamilton
Rochester
NEW YORK
Buffalo
Albany
Ithaca
Detroit
Cleveland
Youngstown
CON
Bridgeport
PENNSYLVANIA
Leaflet | © OpenStreetMap contributors © CARTO