Topic 6.2: Open Data

Samuel Langton (MMU) & Reka Solymosi (University of Manchester)

June 2020

Introduction

- Topic definition.
- Advantages and exciting prospects in open data.
- Chapter structure outline.
- Prerequisites (reference *Geocomputation with R*).

Background

- What is open data?
- What types of organisations release open data?
  - Public sector
  - Private sector
- Why is it needed?
- What are the strengths?
- What are the limitations?
- Substantive examples in criminology (focus on crime and place research).
- Conclude with: a key one is Open Street Map.

Open Street Map

What is Open Street Map?

- Brief history.
- Motivations for and advantages of an open web mapping platform.
- Primary features (keys, values, elements).
- Contributing to the data yourself.

Downloading Open Street Map Data

- Define and explain APIs.
- For OSM, describe overpass queries and overpass-turbo.
- Query example (brief).

Using Open Street Map data in R

- Made easy through the `osmdata` package in R.
- More straightforward than overpass-turbo.
- Compatible with both `sp` and `sf` classes of spatial data.
- We focus on `sf` as its compatible with the `tidyverse`.
- Outline the key documentation and references.

Walk-through

First, ensure that you have the relevant packages installed in R. Although the main package used in this demonstration is `osmdata`, for querying the Open Street Map API, we use a number of additional packages for data handling and visualisation. If you don't have these packages installed, use the `install.packages()` function prior to loading each one with `library()`.

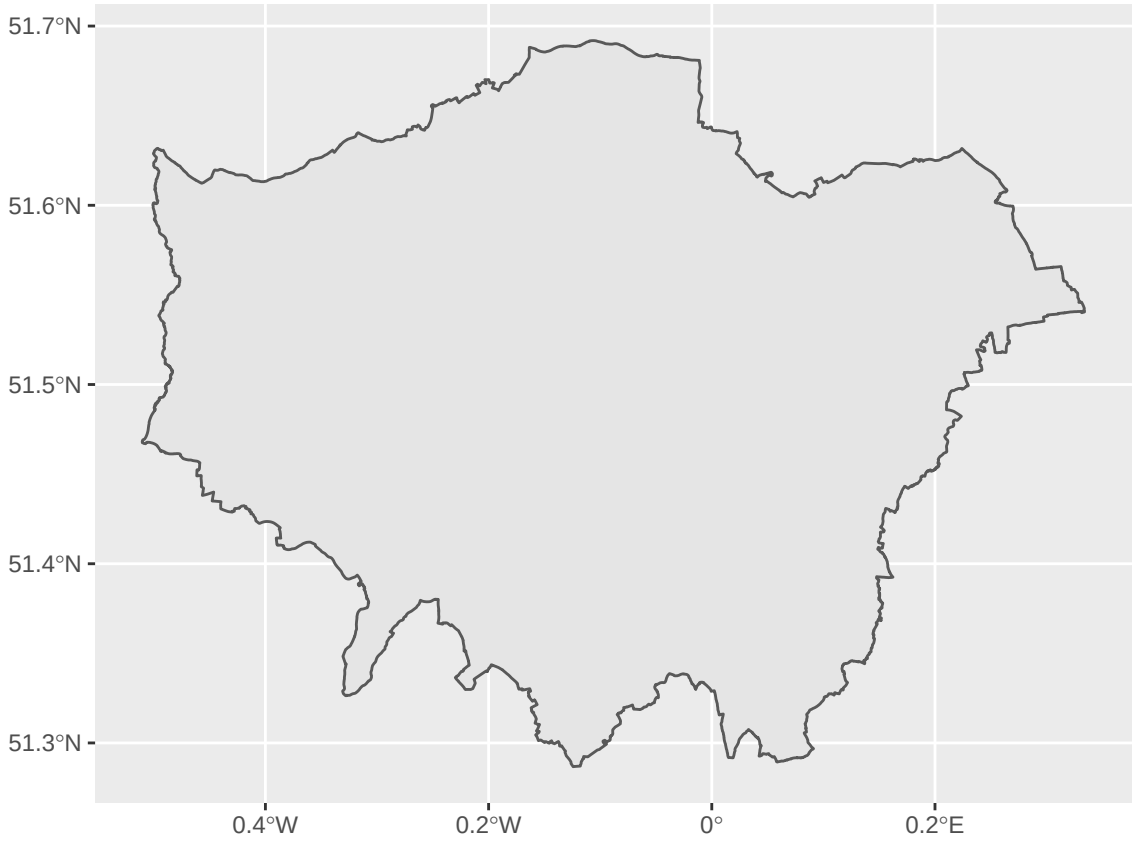
```
library(osmdata)
library(ggplot2)
library(sf)
library(readr)
library(dplyr)
library(tidyr)
```

All queries begin with a bounding box specification to define the study region. This can be obtained manually, which requires some existing knowledge about an area using the latitude and longitude coordinates, but it is generally easier to use a search term. Here, we select Greater London in the United Kingdom using the `getbb()` function, specifying that we want the content as a simple features (sf) polygon.

```
bb_sf <- getbb(place_name = "greater london united kingdom", format_out = "sf_polygon")
```

We now have our study region defined as the administrative boundaries of Greater London. This can be visualised using `ggplot2` and the *simple features* geometry `geom_sf()` available with `sf`. Note that by default, the Coordinate Reference System (CRS) is the World Geodetic System 84.

```
ggplot(data = bb_sf) +
  geom_sf()
```



Now we have our study region, we can scrape data from the OSM API using the `opq()` function, which is

short for 'Overpass query'. This allows you to build an Overpass query, outlined in the previous section, from within the R environment. We specify the bounding box object which is our study area, and pass this through using a pipe `%>%` to `add_osm_feature()` in which we define what we want to pull from the API. As we noted earlier, features in OSM have are defined through keys and values. Here, we specify that we want amenities (the key) defined as bicycle parking (the value). This query is then piped through to `osmdata_sf()` which ensures that the resulting object is a *simple features* class for easy plotting with `ggplot2`. We trim the features pulled from the API using `tim_osmdata()` to ensure that everything stays within the boundaries of our study region.

```
bikes_sf <- opq(bbox = bb_sf) %>%                                    # select bounding box
  add_osm_feature(key = 'amenity', value = 'bicycle_parking') %>% # select features
  osmdata_sf() %>%                                                   # specify class
  trim_osmdata(bb_poly = bb_sf)                                      # trim to region
```

The resulting object `bikes_sf` contains lots of information. We can view the contents of the object by simply executing the object name into the Console.

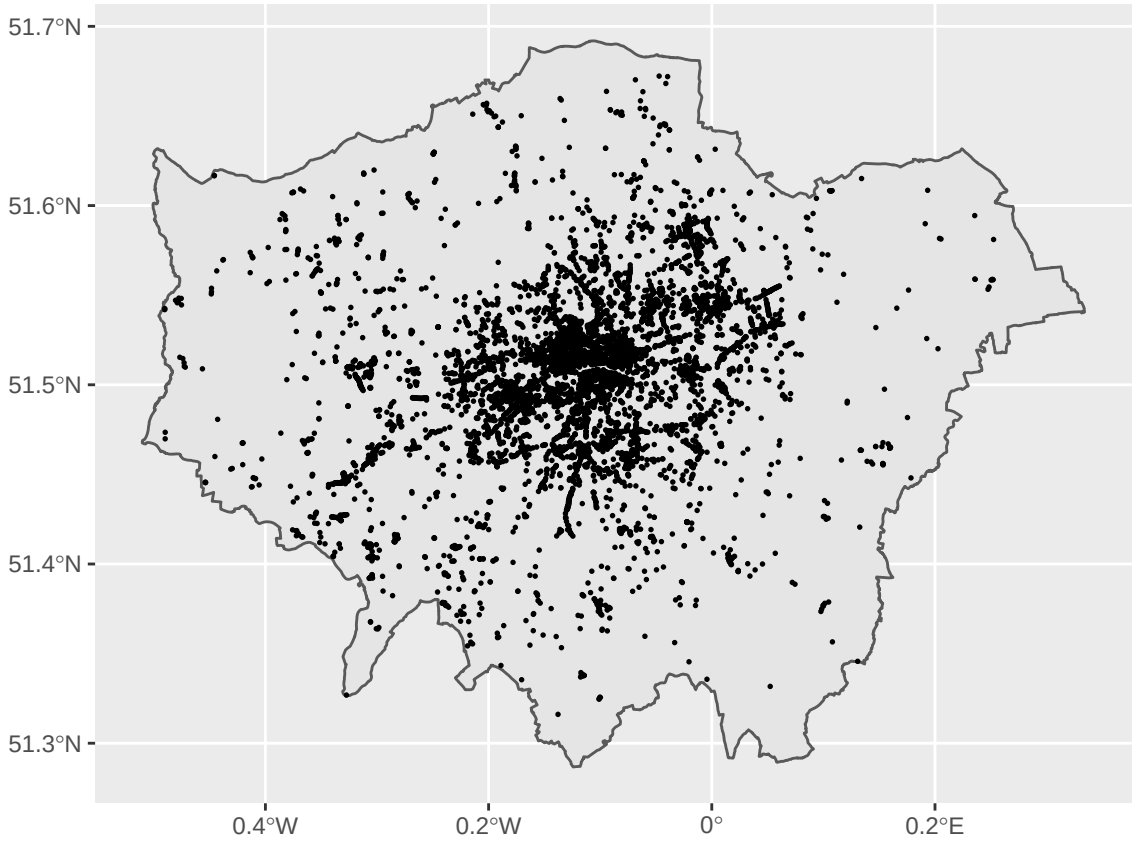
```
bikes_sf
```

This confirms details like the bounding box, but also provides information on the simple features collected from the query. As one might expect, most information relating to bicycle parking has been recorded using points (i.e. two-dimensional vertices, coordinates) of which we have over seven thousand at the time of writing. We also have around one hundred polygons. For now, let's extract the point information only and then transform the CRS to the BNG.

```
bikes_points_sf <- bikes_sf$osm_points
```

We can then plot these points over our original boundaries of Greater London. We reduce the default size of the point to ensure that we avoid too much overlap between bicycle parking locations.

```
ggplot() +
  geom_sf(data = bb_sf) +
  geom_sf(data = bikes_points_sf, size = 0.3)
```



As we can see, most bicycle parking spaces are clustered around the city centre, especially just north of the river Thames. It is also possible to make out key roads flowing in and out of the city centre, which contain bicycle parking all along the street.

Using open police recorded crime data we can then plot actual incidences of bicycle theft to explore whether there is a spatial relationship between bike theft and parking spots in Greater London. For this example, we just use crime recorded as occurring in January 2020. First, let's load in the data as it downloaded raw from https://data.police.uk/data/.

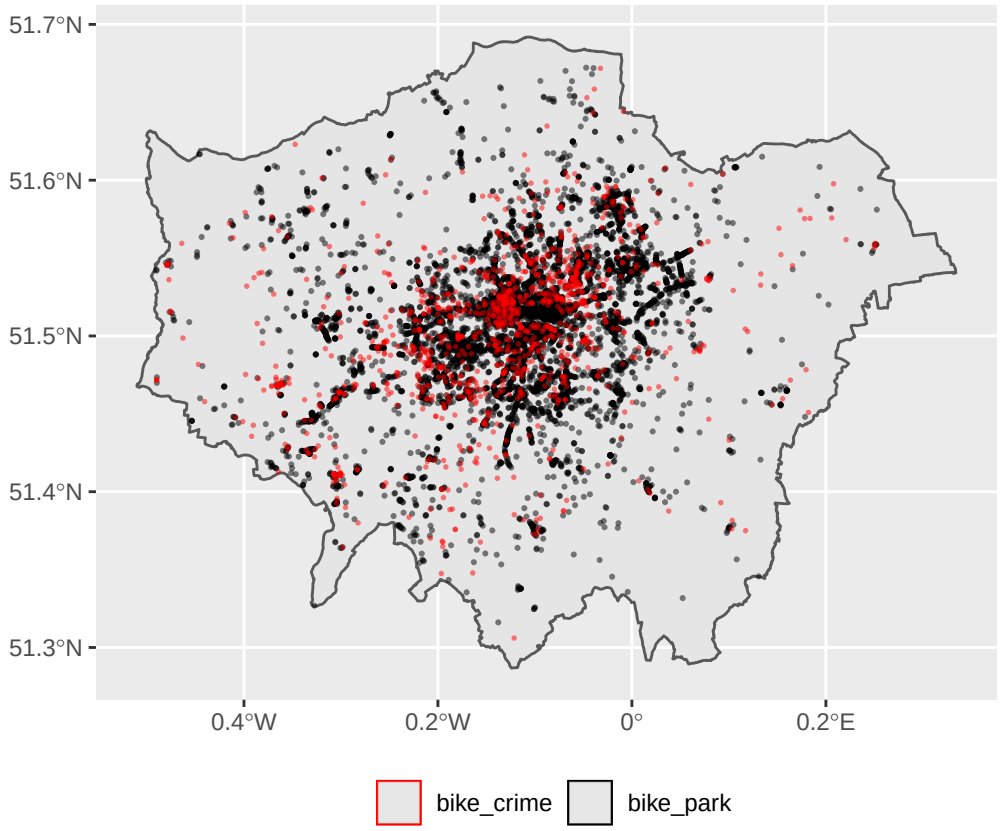
```
crime.df <- read_csv("data/2020-01-metropolitan-street.csv")
```

We then need to conduct a bit of preliminary data handling: filter crimes which were tagged as bicycle theft, convert the latitude and longitude columns to coordinates with *simple features*, state the WGS 84 CRS and then clip the points by our study region.

```
bike.crime.sf <- crime.df %>%
  filter(`Crime type` == "Bicycle theft") %>%
  drop_na(Longitude, Latitude) %>%
  st_as_sf(coords = c(x = "Longitude", y = "Latitude"), crs = 4326) %>%
  st_intersection(bb_sf)
```

To demonstrate the data in its entirety, we can plot the Greater London boundaries, overlayed with the bicycle parking space locations, and open crime data about bicycle thefts. It is worth clarifying that open police recorded crime data in England and Wales is spatially anonymised by a process of snapping points to a pre-defined grid (see Tompson et al., 2015). For that reason, many of these points overlap, and thus a degree of transparency is used for the points.

```r
ggplot() +
  geom_sf(data = bb_sf) +
  geom_sf(data = bikes_points_sf, aes(colour = "bike_park"), size = 0.4, alpha = 0.5) +
  geom_sf(data = bike.crime.sf, aes(colour = "bike_crime"), size = 0.3, alpha = 0.5) +
  scale_colour_manual(name = NULL, values = c(bike_park = "black", bike_crime = "red")) +
  theme(legend.position = "bottom")
```



Future of open data

- Threats to its sustainability (e.g. licence expiry).
- Prospects in crime of place research.
- Examples of cool projects (e.g. Colouring London).
- Suggestions for new avenues which can expand the field.

Conclusion

- Re-cap on what we've covered.
- Wrap-up the key points.

References

Tompson, L., Johnson, S., Ashby, M., Perkins, C., & Edwards, P. (2015). UK open source crime data: accuracy and possibilities for research. *Cartography and geographic information science*, 42(2), 97-111.