

# Topic 6.2: Open Data

Samuel Langton and Reka Solymosi

1 June 2020

## Introduction

- Topic definition
- Chapter structure

## Background

- What is open data?
- What types of organisations release open data?
  - Public sector
  - Private sector
- What are the strengths/limitations of open data?

## Surveying environmental features using Open Street Map

### What is Open Street Map?

### Downloading Open Street Map Data

- APIs
- Overpass queries and overpass-turbo

### Using Open Street Map data in R

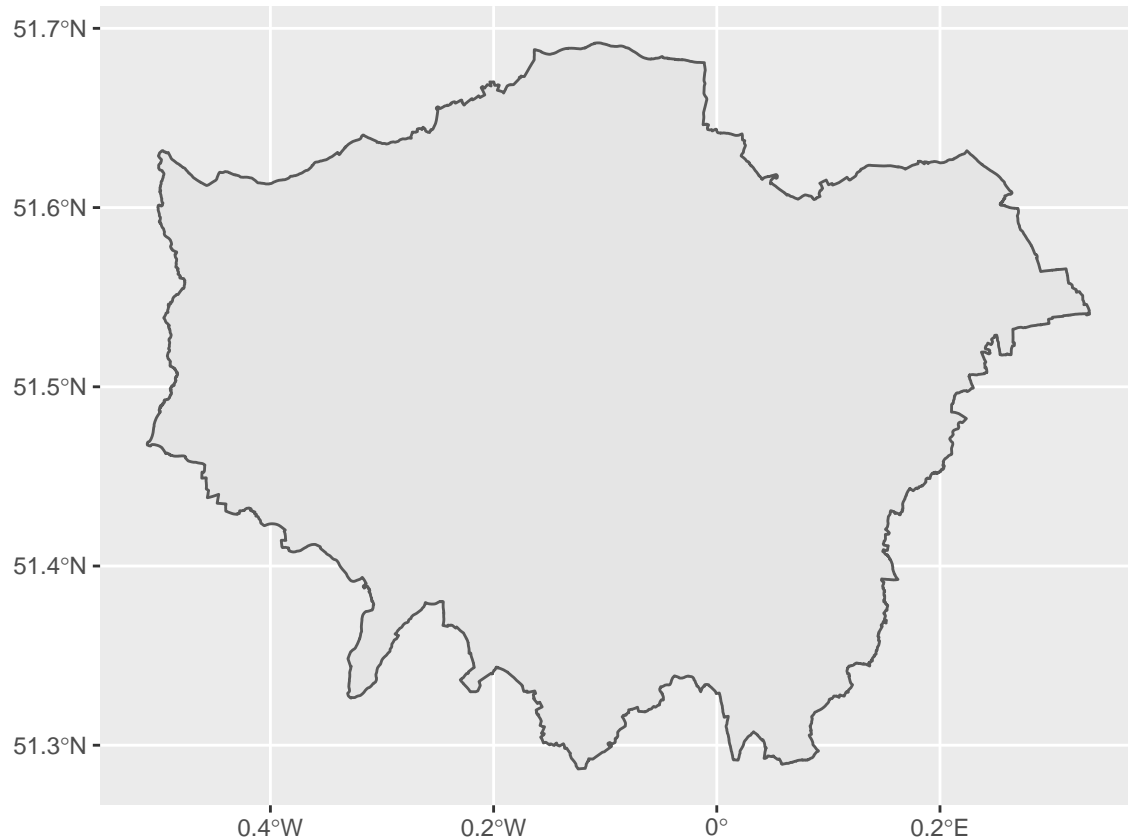
osmdata R package.

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) object.

```
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.

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



Now we have our study region, we can scrape data from the OSM API using the `opq()` function. This allows you to build an Overpass query, outlined in the previous section, from within the R environment. We specify the bounding box defined earlier, and pass this through using a pipe `%>%` to `add_osm_feature()` in which we define what we want to pull from the API. We noted earlier, features recorded on OSM with various tags: keys and values. Here, we specify that we want amenities (the key) defined as bicycle parking (the value). The next function `osmdata_sf()` ensures that the resulting object is a simple features class for easy plotting with `ggplot`. We trim the features pulled from the API based on the Greater London boundaries, otherwise we would end up with information which extends beyond the 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 bounding box
```

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
```

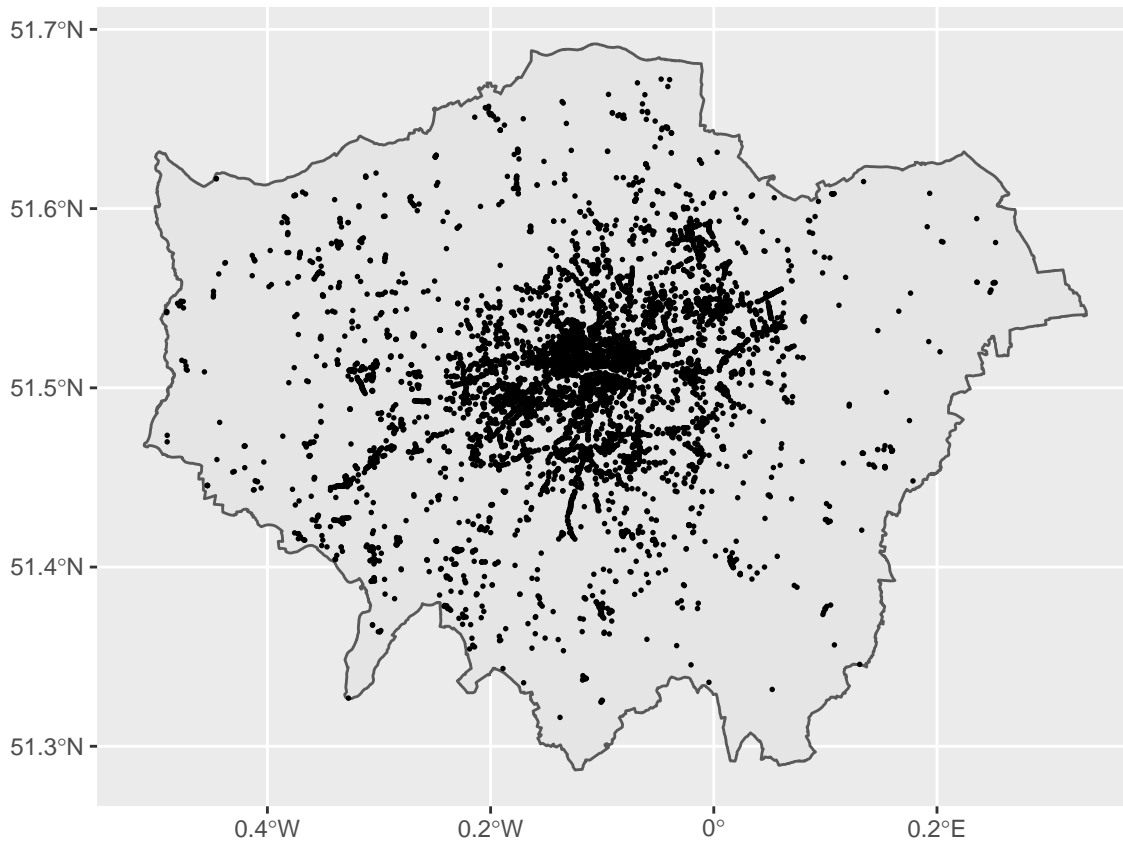
This confirms details like the bounding box, but also provides information on the simple features collected from the query. As one might expect, most information relating to bicycle parking has been recorded using points (i.e. two-dimensional vertices, coordinates) of which we have over seven thousand 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, once we have projected it as

the BNG too. 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/>.

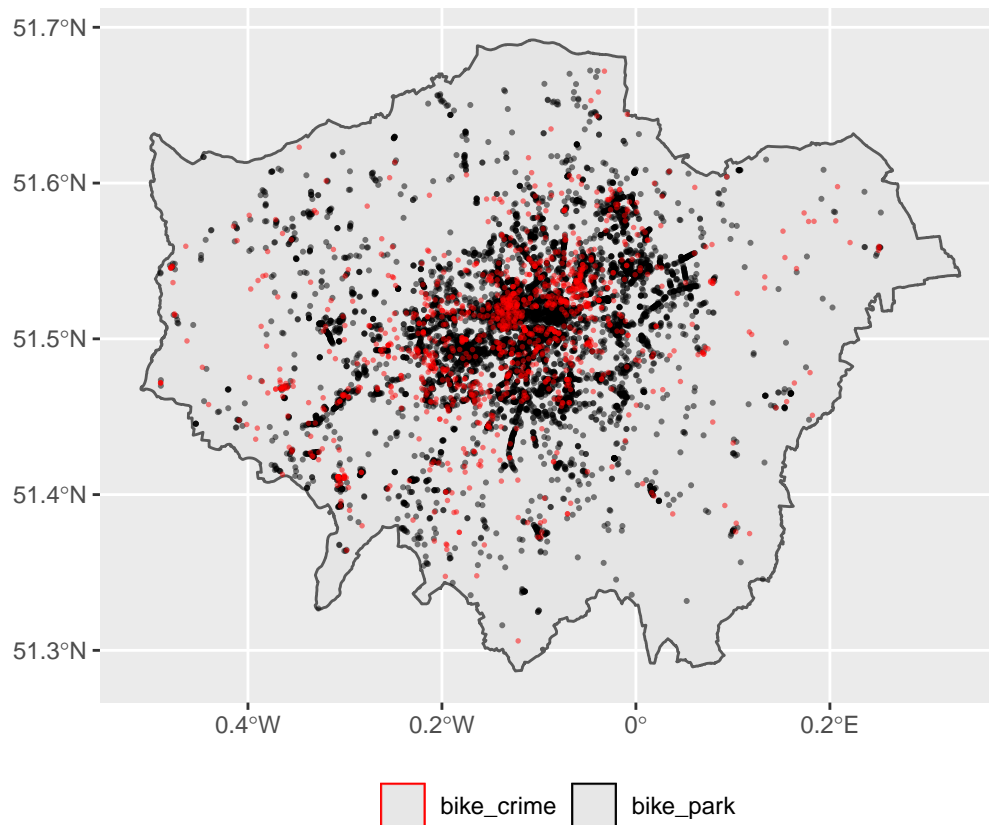
```
crime.df <- read_csv("data/2020-01-metropolitan-street.csv")
```

We then need 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, assign a CRS and then clip the points by our study region. Finally, we transform the CRS to the British National Grid so we can accurately plot these points over our bicycle parking locations.

```
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)
```

Then, finally plot everything together, with a few adjustments to the appearance.

```
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

- Open topics in open data
- Further applications to crime and place research

## Conclusion

- Re-cap
- Wrap-up