Practical: Using the Leaflet library in R (Web Mapping and Analysis)

Hai Nguyen

30 November 2015

Practical: a step-by-step guide to the R leaflet package

The leaflet package

Remember from the lecture that all Leaflet applications require us to prepare Leaflet resources such as the stylesheets (CSS) and the JavaScript library (JS). In R, the [leaflet] package simplifies this step and provide us a user-friendly interface to work with Leaflet maps without even knowing JavaScript and HTML (eventhough it's very cool to know).

Installing and loading this package (and its dependencies) is very easy.

```
install.packages("leaflet")
library(leaflet)
```

For example, the following JavaScript/HTML code fragement from the lecture (loading Leaflet resource and initialising a map):

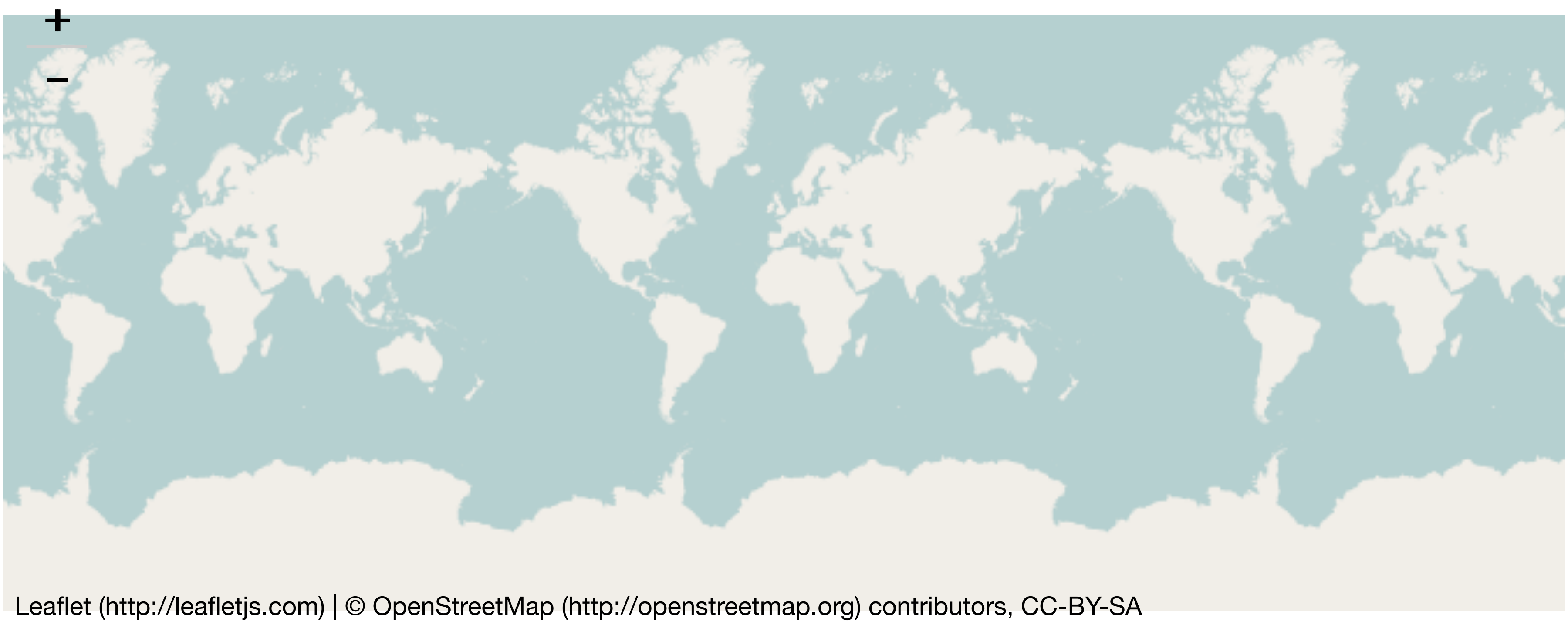
```
<head>
    <!--STEP 1: loading leaflet css and js files from the CDN-->
        <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet/v0.7.7/leafl
et.css" />
        <script src="http://cdn.leafletjs.com/leaflet/v0.7.7/leaflet.js"></script>
</head>
<body>
    <!-- your map element -->
        <div id="mymap" style="height:300px;"></div>
</body>
```

can be converted into R as follows:

```
mymap <- leaflet(height = 300)
```

Note that after running this line of code, nothing will show up as we still have not attached the tile layer (Step 2.a in the lecture). Let's do it again with a tile layer attached:

```
library(leaflet)
mymap <- leaflet(height = 300)
mymap <- addTiles(mymap)
mymap
```



Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)

Note that leaflet's functions such as addTiles modify the structure of the map object (in this case `mymap`) and return a modified map object. This modified map can be re-assigned to a variable. In the above example, for simplicity and readability we re-assign it to the old variable. The last line is to show the map object.

Looking at `addTiles` function signature (below), the default tile layer are composed by OpenStreetMap tiles. This can be changed by adding new value for `urlTemplate` parameter.

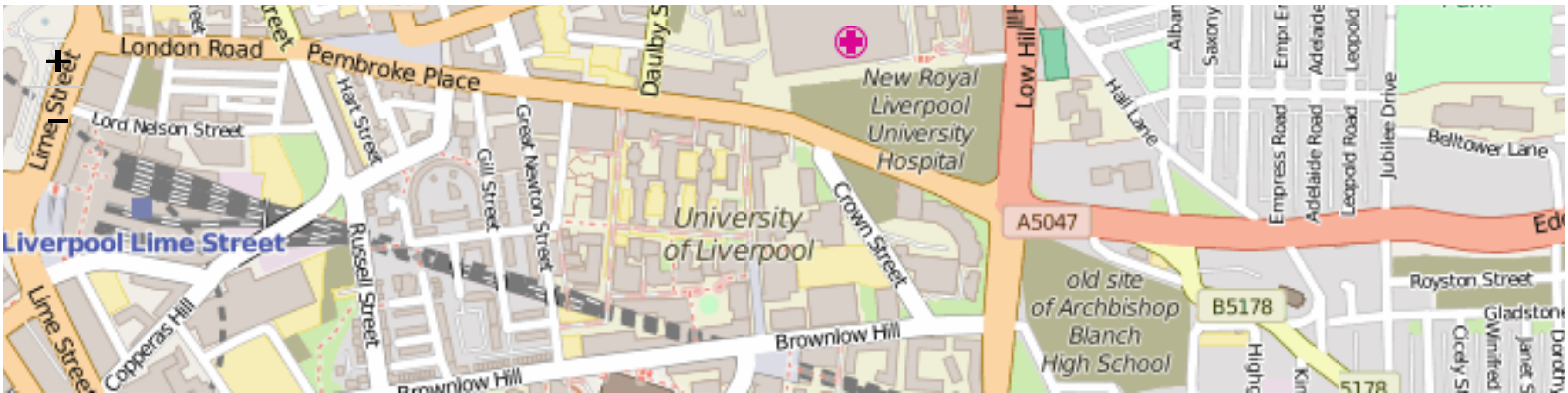
```
addTiles(map, urlTemplate = "http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png",
    attribution = NULL, layerId = NULL, group = NULL, options = tileOptions())
```

Basic Map: Markers and Popups?

In this section we will look at how to visual a map with a marker and a popup window at the map center.

After attaching the tile layer into a map object as in the previous section, we can set/change the map view by setting a centre and a zooming level with function `setView`

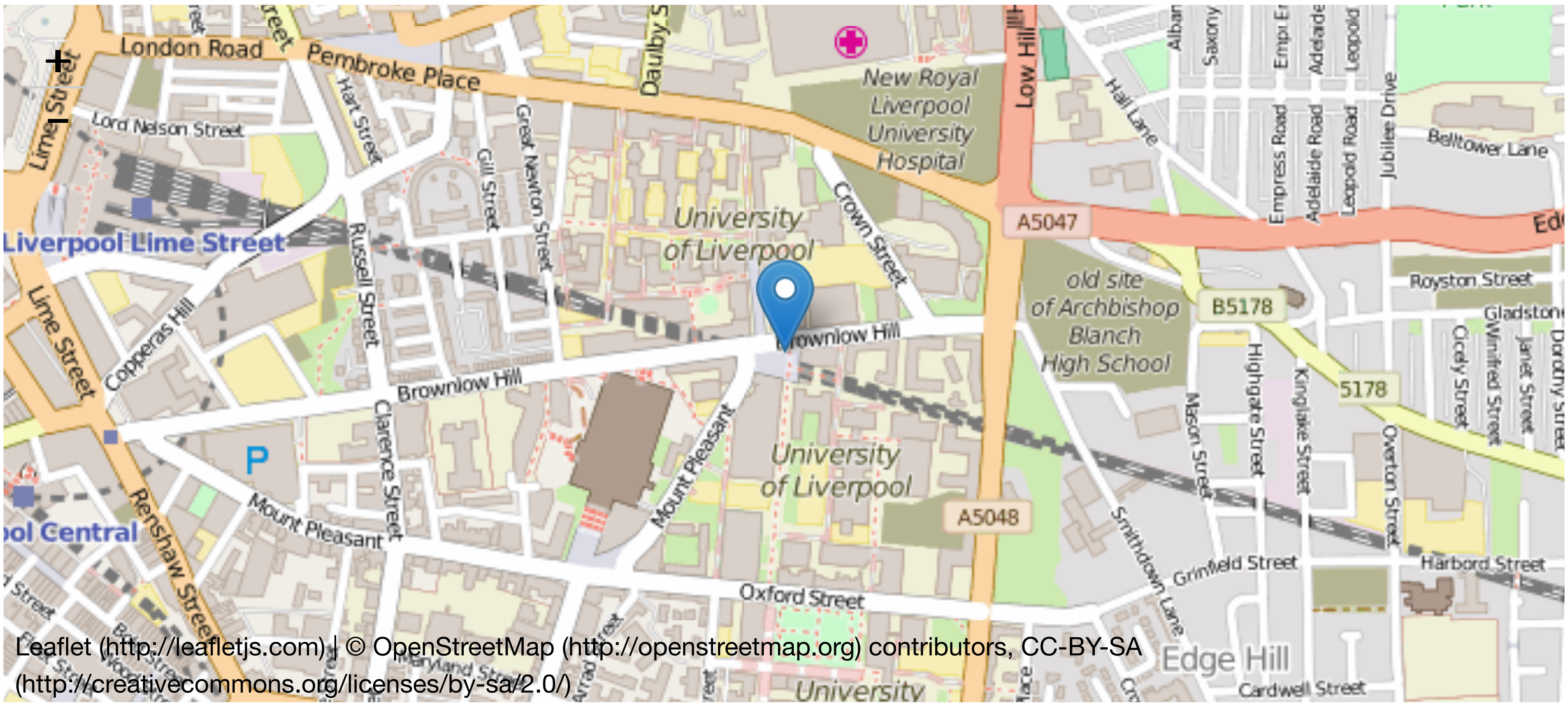
```
mymap <- setView(mymap,lng = -2.9655722, lat = 53.405936, zoom = 15)
mymap
```



Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)

The next step is to create a basic marker with popup:

```
lat <-  53.405936
lng <- -2.9655722
mymap <- addMarkers(mymap,lng,lat,popup = "<a href='http://www.liverpool.ac.uk'>Un
iversity of Liverpool</a>")
mymap
```


Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)
University of Liverpool
Brownlow Hill
Liverpool Lime Street

In the code fragment above, since we only create a simple popup for the marker, the `popup` parameter can be used for convenience (popups can also be attached to shape objects in a similar way).

Creating Leaflet map with open data

So far we have tried to create a Leaflet map with a single marker and a single popup using our hard-coded geographical information (lat,lng). In this section we will create a Leaflet map using data from from an open data-source, the airbnb (http://www.airbnb.com) dataset, available at (http://insideairbnb.com/get-the-data.html)http://insideairbnb.com/get-the-data.html (http://insideairbnb.com/get-the-data.html).

We will be creating a map of all *airbnb* listings, each of them has a marker.

Firstly, we need to download the data for London (already cleaned and available as CSV files) from here (http://data.insideairbnb.com/united-kingdom/england/london/2015-09-02/visualisations/listings.csv).

```
london_airbnb_listings <- read.csv('http://data.insideairbnb.com/united-kingdom/en
gland/london/2015-09-02/visualisations/listings.csv',sep = ',')
```

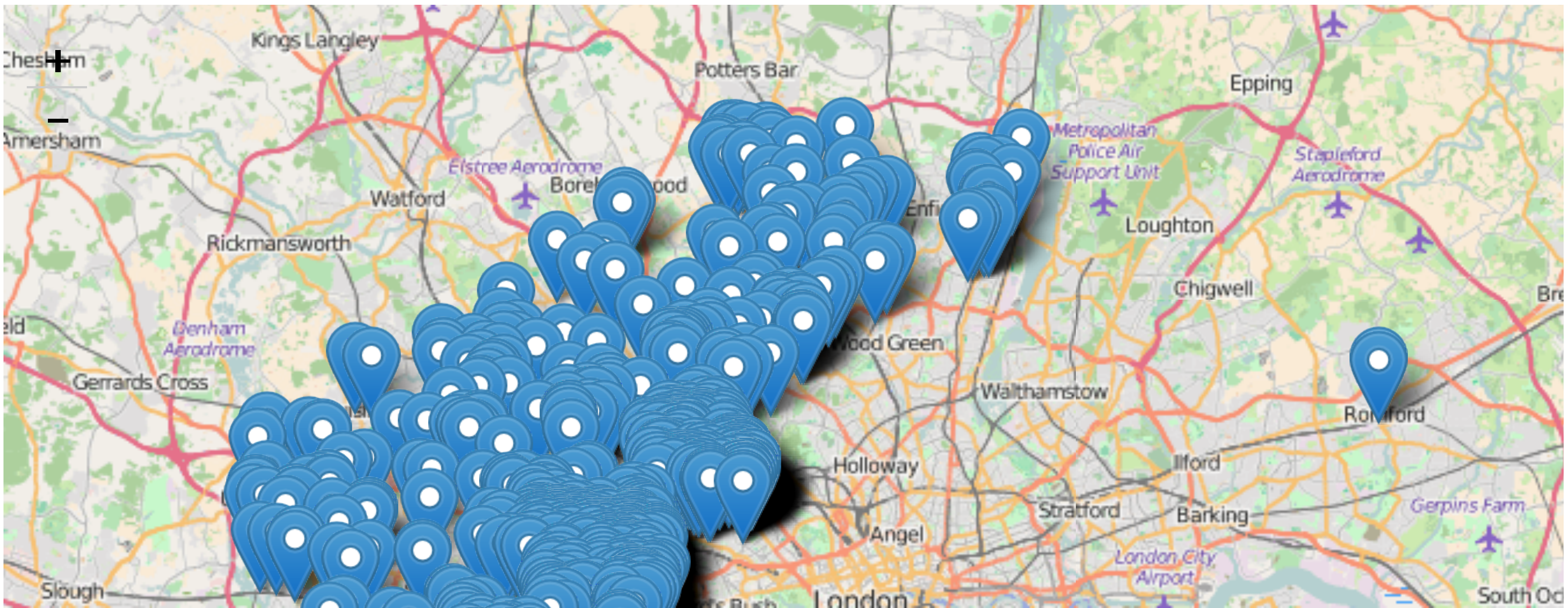
A dataframe of 25361 rows and 16 columns has been created. Let’s justhave a quick look at the dataset:

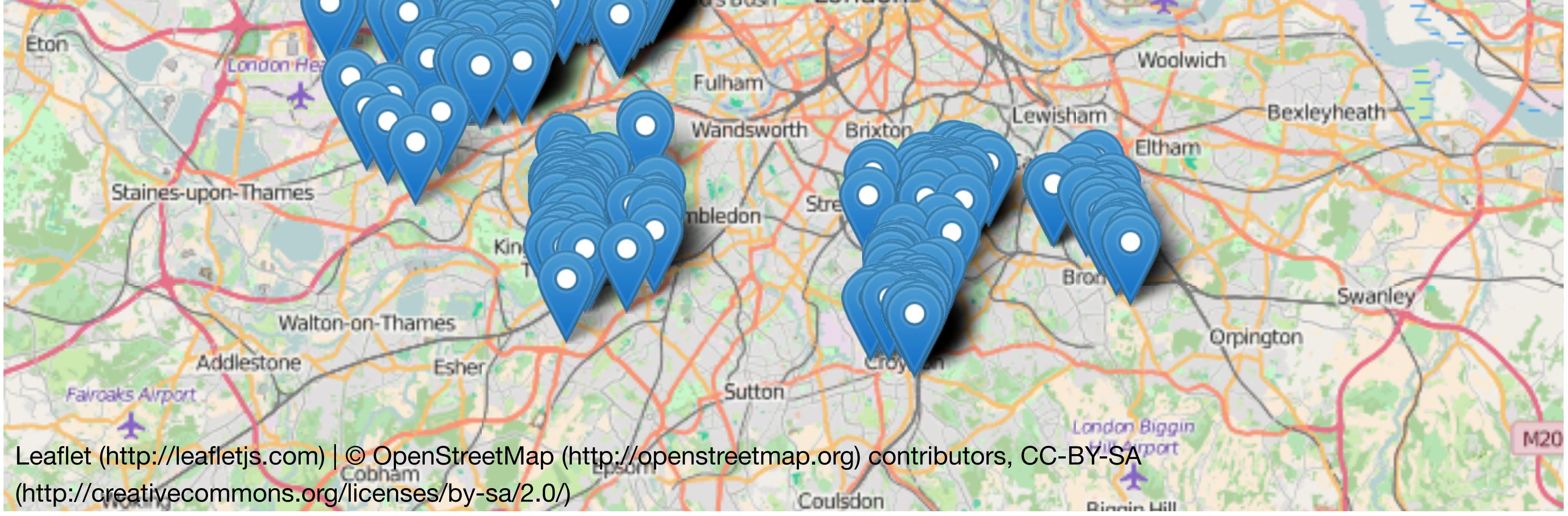
```
head(london_airbnb_listings)
```

```
##        id                             name  host_id host_name
## 1  375799 Luxury apartment : London/Wimbledon  1889832    Glynis
## 2 2284438        Bright and comfy double room 11629266     Agata
## 3 4356007       Luxury victorian family homel 22611343
## 4 7031432    Ground floor bedroom in Kingston 36862552    Andrew
## 5 7208109          A Room with Private Shower 28650375      Hugh
## 6 6003865   Double Room in Bright Period Flat 29230489   William
##   neighbourhood_group        neighbourhood latitude  longitude
## 1                  NA Kingston upon Thames 51.40296 -0.2504358
## 2                  NA Kingston upon Thames 51.39547 -0.3094916
## 3                  NA Kingston upon Thames 51.41475 -0.2895943
## 4                  NA Kingston upon Thames 51.42833 -0.3037212
## 5                  NA Kingston upon Thames 51.39616 -0.2498214
## 6                  NA Kingston upon Thames 51.40703 -0.2966220
##         room_type price minimum_nights number_of_reviews last_review
## 1 Entire home/apt    80              7                42  2015-07-11
## 2    Private room    43              2                20  2015-08-19
## 3 Entire home/apt   325              2                 6  2015-08-18
## 4    Private room    47              1                 0
## 5    Private room    25              1                 0
## 6    Private room    40              1                 0
##   reviews_per_month calculated_host_listings_count availability_365
## 1              1.05                              1              318
## 2              1.06                              1              337
## 3              1.24                              1              276
## 4                NA                              1               41
## 5                NA                              7               17
## 6                NA                              1              335
```

We can then create a map with this dataset `london_airbnb_listings` .

```
m <- leaflet(data = head(london_airbnb_listings,1000))
m <- addTiles(m)
m <- addMarkers(m,~longitude,~latitude)
m
```



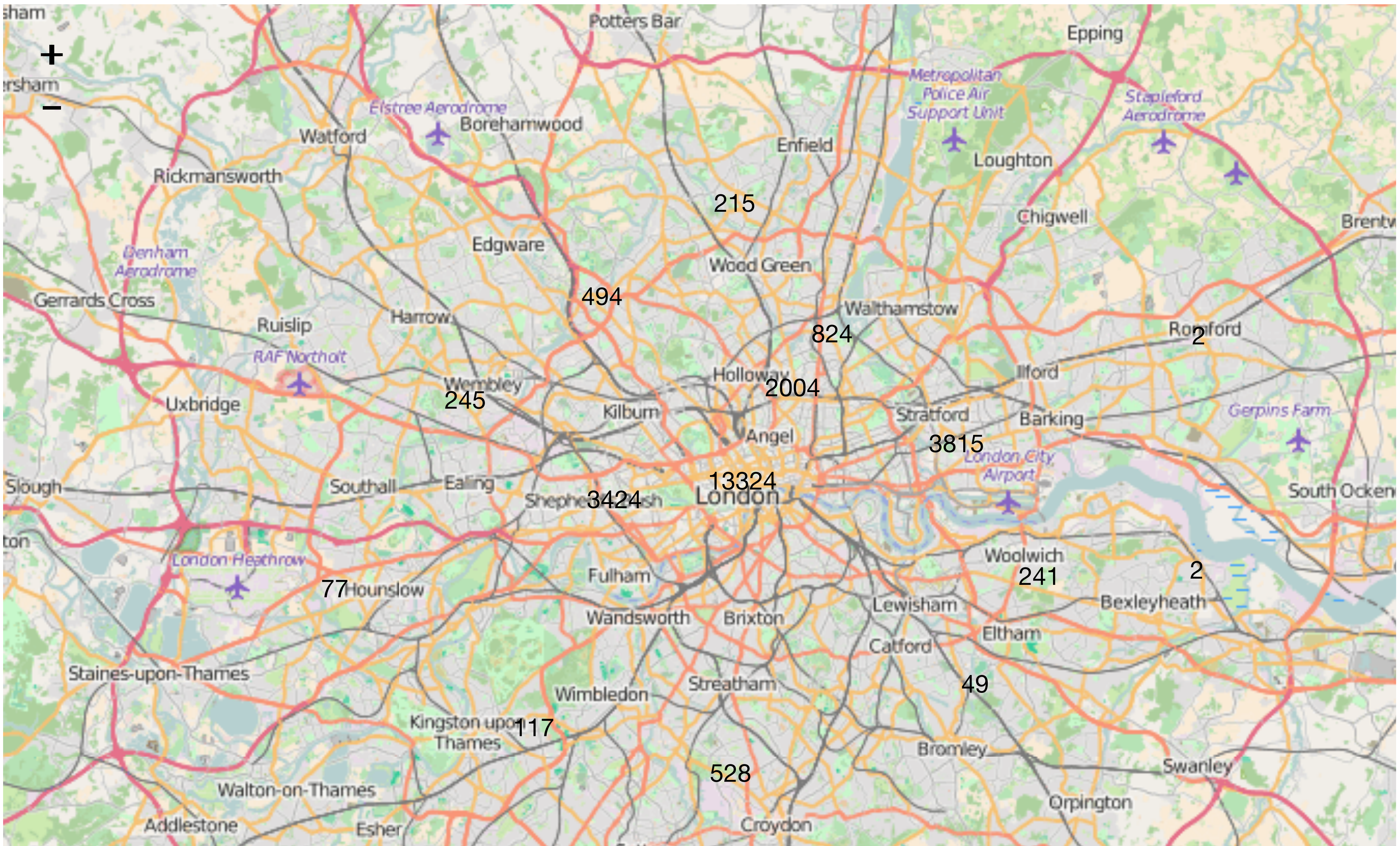


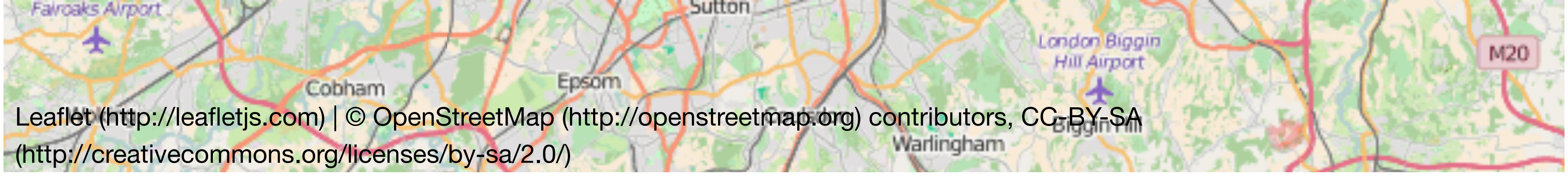
The above code-fragment is the same as in the previous section. However, instead of having a single lng/lat value for one location, we used the `longitude` and `latitude` columns in `london_airbnb_listings`. Note that ~ symbol was used to distinguish the column names from the variables (without ~, `longitude` and `latitude` are user-defined variables, as in previous section).

We can notice that the above code-fragment will take a while to render the map, due to a large number of locations (1000 - note that we only select the first 1000 instead of all 25K locations). To solve this problem, clustering methods have been used to group locations within a neighbourhood into one marker. There are many clustering methods, the one used by Leaflet is *plain greedy clustering* (more here (http://leafletjs.com/2012/08/20/guest-post-markerclusterer-0-1-released.html)).

To use this clustering method, we can re-run the code as follows:

```
clusteredMap <- leaflet(data = london_airbnb_listings)
clusteredMap <- addTiles(clusteredMap)
clusteredMap <- addMarkers(clusteredMap,~longitude,~latitude,clusterOptions = mark
erClusterOptions(),popup = ~name)
clusteredMap
```




Fairoaks Airport
Sutton
Cobham
Epsom
London Biggin Hill Airport
M20
Warlingham
Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)

Here we created a new map called `clusteredMap`. If you want to re-use the old map `m`, note that you need to re-initialise the map to remove previous markers (by calling `leaflet` again).

Lines and Shapes

So far we have been using points data only. Sometime it would be easier/better to visualise data using lines and shapes. For example, you might want to draw a line between an airbnb home to your favourite place of interest (POI) to estimate the distance, or draw a circle for each airbnb home to show how many reviews they have received.

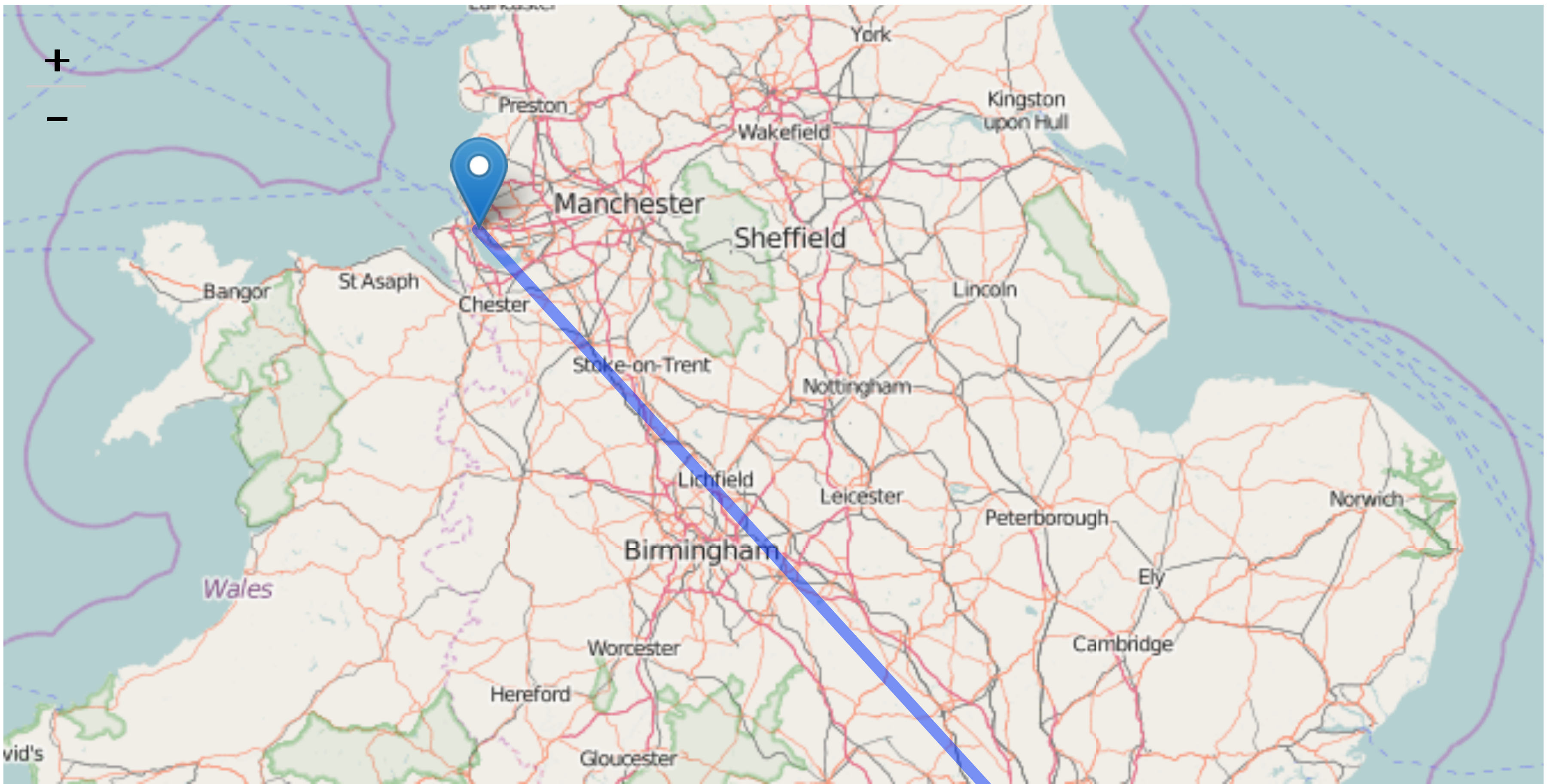
In this section, we will draw lines between 100 points from our dataset to Big Ben (https://en.wikipedia.org/wiki/Big_Ben).

Firstly, we will create a random sample of 100 points from `london_airbnb_listings`:

```
airbnb_home_sample100 <- london_airbnb_listings[sample(1:nrow(london_airbnb_listin
gs), 100),]
```

Now we will try to draw a line from a point to another.

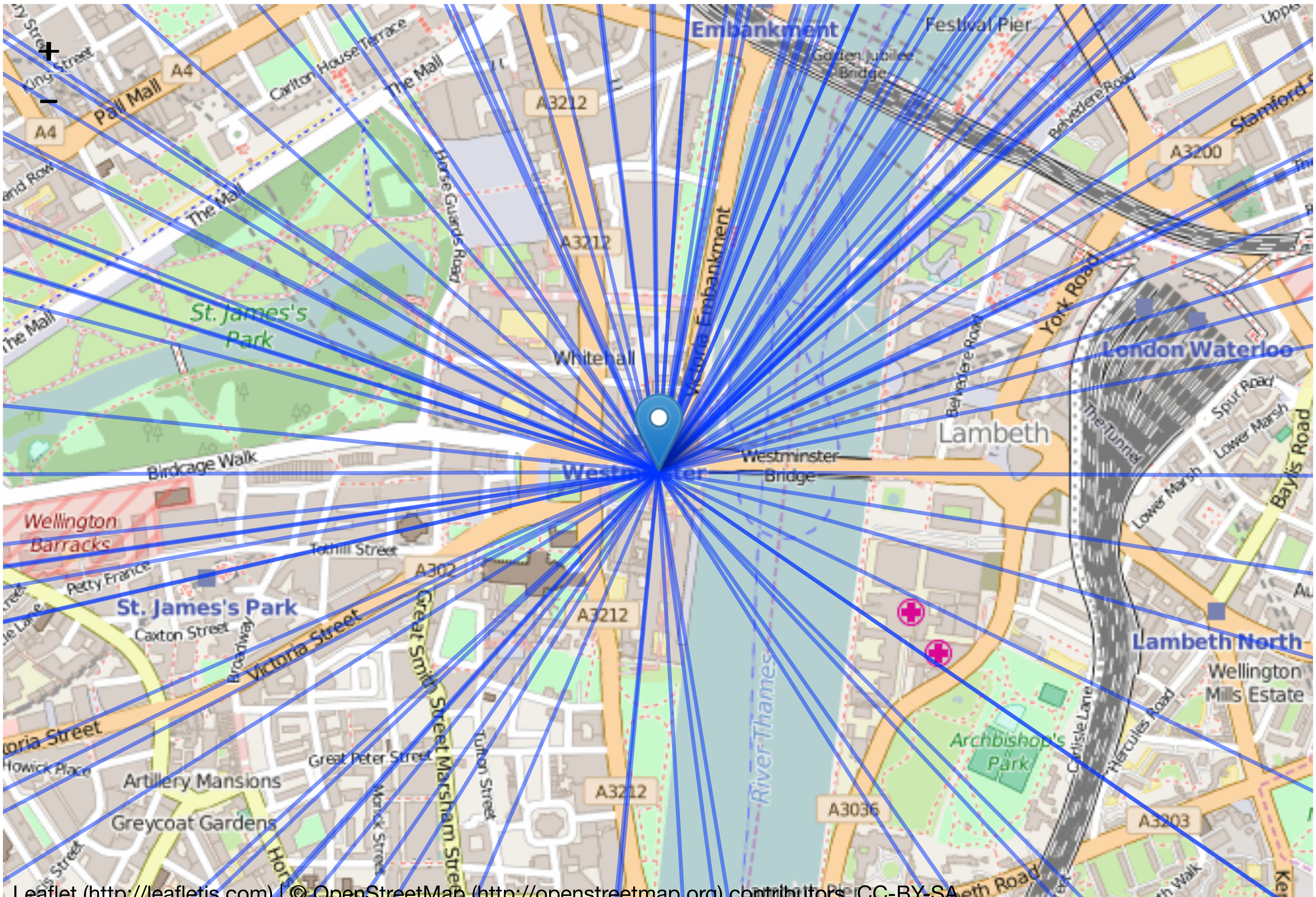
```
m <- leaflet()
m <- addTiles(m)
# add a marker at UoLiv
m <- addMarkers(m,lng = -2.9655722,lat =  53.405936)
# add a marker at Big Ben
m <- addMarkers(m,lng = -0.124661,lat =  51.500756)
# create a line between 2 points
m <- addPolylines(m,lng = c(-2.9655722,-0.124661),lat = c( 53.405936, 51.500756))
m
```

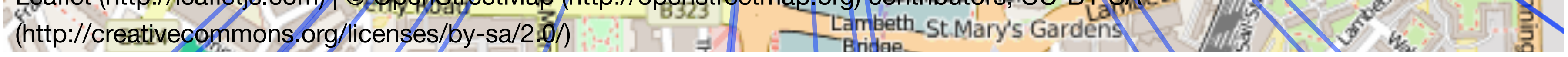


Oxford
St Albans
Chelmsford
Swansea
Cardiff
Bristol
London
Canterbury
Wells
Salisbury
Calais
Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)

Next, we will create a Leaflet map with data from the random sample, add markers for these 100 points and draw a line from each of these points to Big Ben. We will need to use a loop to draw all 100 lines.

```
m <- leaflet(data = airbnb_home_sample100)
m <- addTiles(m)
# add markers for airbnb points
m <- addMarkers(m,~longitude,~latitude,popup = ~name)
# add a marker for Big Ben
m <- addMarkers(m,lng = -0.124661,lat =  51.500756,popup='Big Ben')
# draw a line from each airbnb home to Big Ben
for (i in 1:100)
{ r <- airbnb_home_sample100[i,]
  m <- addPolylines(m,lng = c(r$longitude,-0.124661),lat = c(r$latitude, 51.500756
),weight = 2, opacity = 0.5)
}
# set view and zoom level for readability
m<-setView(m,lng = -0.124661,lat =  51.500756, zoom=15)
# show the map
m
```





(http://creativecommons.org/licenses/by-sa/2.0/)

Your Turn!

Now you have been shown how to create Leaflet maps with open data. In this section, we have some small exercise for you to try what we have learnt so far (and what we haven't covered so far). You can choose any of the below exercises (or all).

Bubble Map [Difficulty Level: +]

Create a bubble map to visual the number of review each airbnb home has. A bubble should have size in correspondence to the number of reviews. Places without any review can be ignored/filtered out in advanced. You can take a random sample of size 1000 from our London airbnb dataset.

- **Hints:** use `addCircle()` function *

Classifying Accommodation by Room Type [Difficulty Level: ++]

An *airbnb.com* accommodation has at least one of the following room type: "Entire home/apt", "Private room", and "Shared room". Create a map to visual a random sample of 1000 airbnb places and show the room type classification on the map. How you might visualise this information is up to you (data scientists always need their imagination :)).

- **Hints:** `addCirles` , `addMarkers` , colors, etc.

Price and Location [Difficulty Level: +++]

There are some open datasets about train station usages, such as ones from Office of Rail and Road (http://orr.gov.uk/statistics/published-stats/station-usage-estimates). Busy train stations with non-season tickets might locate near tourist hotspots. However, it is unclear how prices/counts of airbnb listings reflects how busy the train stations are within the same neighbourhood. As a data scientist, your task is to show this refection (if exists) on a Leaftlet map.

The original data set with column description is available at here (http://orr.gov.uk/__data/assets/excel_doc/0019/20179/Estimates-of-Station-Usage-in-2014-15.xlsx). A cleaner version for London stations only has been made ready for you here (https://www.dropbox.com/s/f1ed6vv6l3ssvem/london_trainstation_1415est.csv).

- **Hints:** note that we are only interested in non-season tickets. A basic visualisation can show bubbles with sizes and color correponding to the prices and the stations' entries/exits, respectively.

Reading and Resources:

- (https://rstudio.github.io/leaflet)https://rstudio.github.io/leaflet (https://rstudio.github.io/leaflet)
- (http://insideairbnb.com/get-the-data.html)http://insideairbnb.com/get-the-data.html (http://insideairbnb.com/get-the-data.html)