

INTRODUCTION TO DATA SCIENCE

Fall 2022

INTERACTIVE MAPS WITH LEAFLET

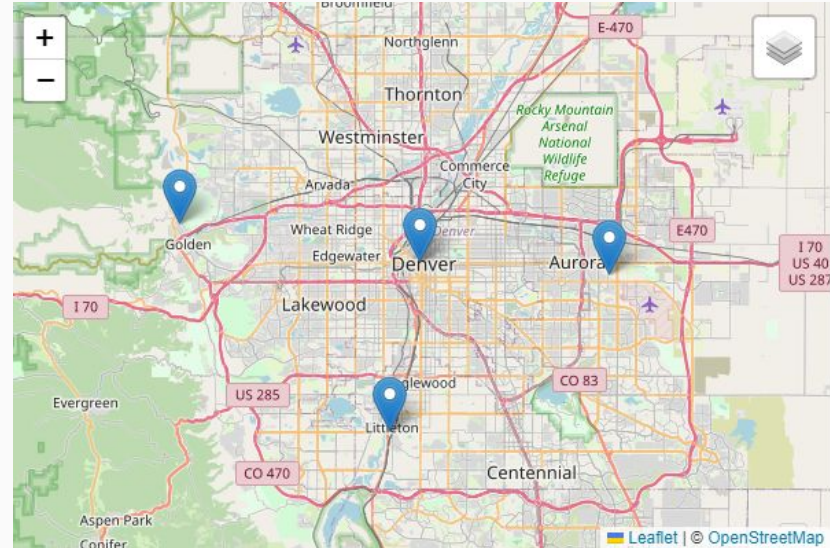
Agenda

- Introduction and Purpose
- Key Functionalities
- Additional Features
- Further Resources



What is Leaflet?

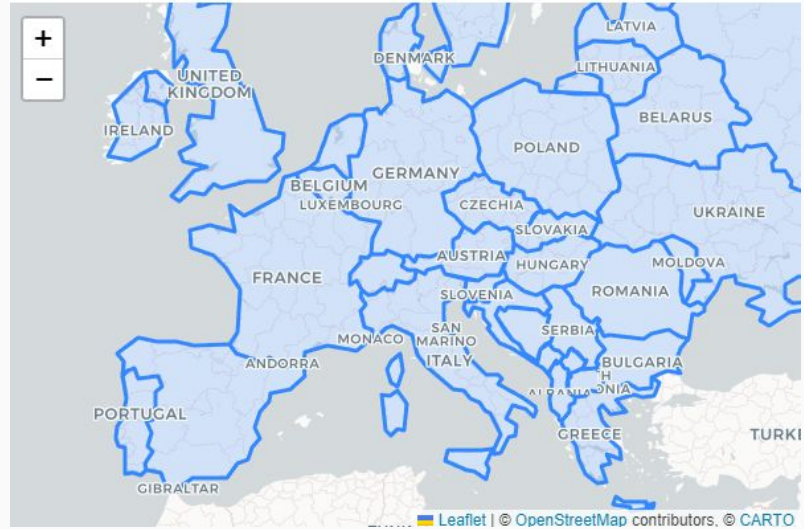
- Simple and extremely lightweight library for interactive maps
- Works across all major desktop and mobile platforms
- Used by companies such as Facebook, Washington Post, European Commission, Financial Times, etc.
- Has multiple layers, popups and marker functionalities



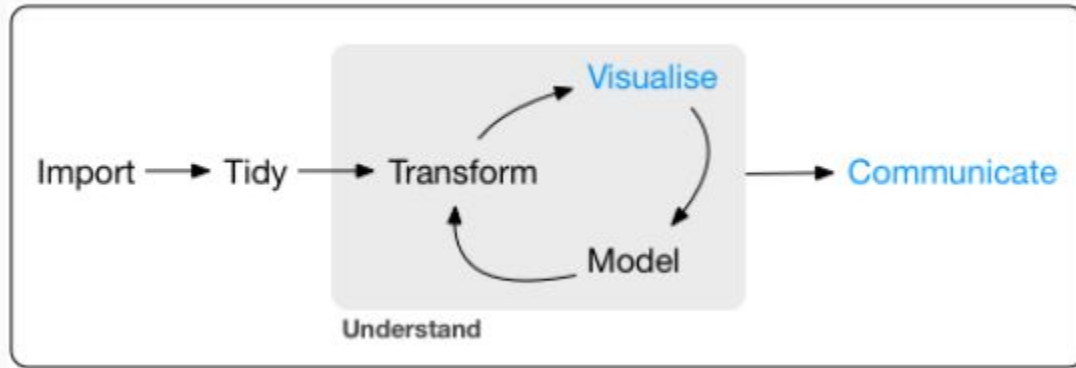
Why Leaflet?

- INTERACTIVE
- Free and Open Source
- Versatile across platforms
- Well-documented API
- Can be extended with plugins
- Easy to learn with R syntax
- High degree of customisation

REQUIRES Spatial coordinates!



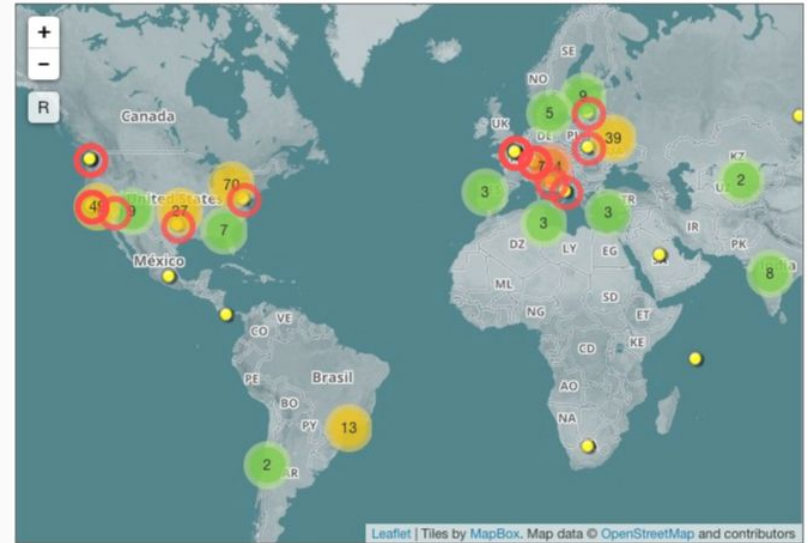
In the Data Science Workflow



The leaflet package is part of the **visualisation** and the **communication** steps of the data science workflow. Leaflet is used for **explanatory visualisation**, although it must be noted that it is designed for use in **interactive environments**.

Use Cases for Leaflet

- Comparing natural features with human activity
- Infectious diseases maps
- Meteorological maps
- Election results
- Socioeconomic Indicators
- Consumption of goods
- Levels of crime

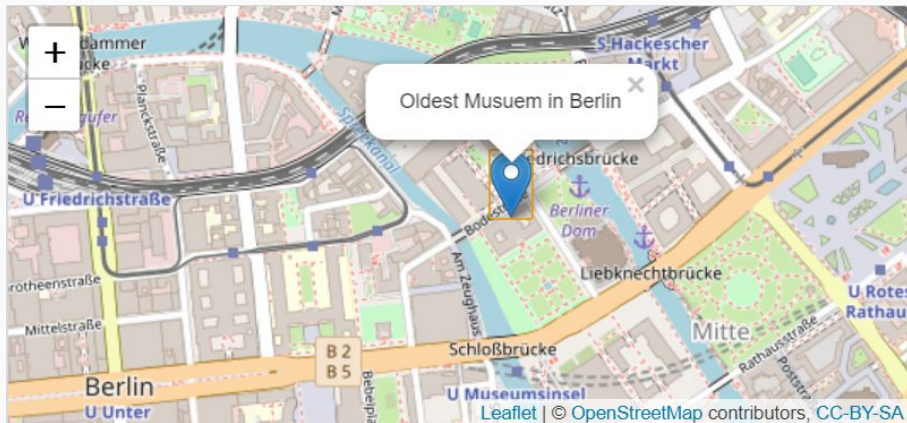


Basic Syntax

```
library(leaflet)

map <- leaflet() %>%
  addTiles() %>%
  addMarkers(lat = 52.51962731511821,
            lng = 13.398745867266065,
            popup = "Oldest Musuem in Berlin")
```

map



Typical Workflow:

1. Load the package using *library()*
2. Create a base map using *leaflet()*
3. Add markers such as the spatial coordinates, popups and other layers as needed
4. Repeat with multiple layers to provide additional information
5. Print the map

Basemaps

Leaflet uses OpenStreetMap as the default basemap, which is the underlying map which is initially parsed before adding layers.

```
10 leaflet().addTo(map)  
11   .addTiles()
```



Third-party Basemaps

Basemap providers

Ready-to-go basemaps, with little or no configuration at all.

| Plugin | Description | Demo | Maintainer |
|---|--|----------------------|--------------------------------------|
| Azure Maps Leaflet plugin | A leafletjs plugin that makes it easy to overlay all the different tile layers available from the Azure Maps . Supports using an Azure Maps subscription key or Azure Active Directory for authentication. | Demo | Ricky Brundritt |
| Bing Maps Layer | Add Bing Maps tiles to your Leaflet Map. Requires Leaflet v1.0.0.beta.2 or later. | Demo | Gregor MacLennan |
| Esri Leaflet | A set of tools for using ArcGIS services with Leaflet. Support for map services, feature layers, ArcGIS Online tiles and more. | | Patrick Arlt |
| L.GridLayer.GoogleMutant | Displays Google maps (with minimal artifacts thanks to a DOM mutation observer technique). | Demo | Iván Sánchez |
| L.MapkitMutant | Displays Apple's MapkitJS basemaps. | | Iván Sánchez |
| L.TileLayer.HERE | Displays map tiles from HERE maps. | Demo | Iván Sánchez |
| L.TileLayer.Kartverket | Provides easy setup of the tile layers from Kartverket (The Norwegian Mapping Authority) | | Kultur og naturreise |
| Leaflet.ChineseTmsProviders | Contains configurations for various Chinese tile providers — TianDiTu, MapABC, GaoDe, etc. | Demo | Tao Huang |
| | NASA EOSDIS GIBS imagery | | |

There are a list of many other third-party base map [providers](#), which can be included using the `addProviderTiles()` function.

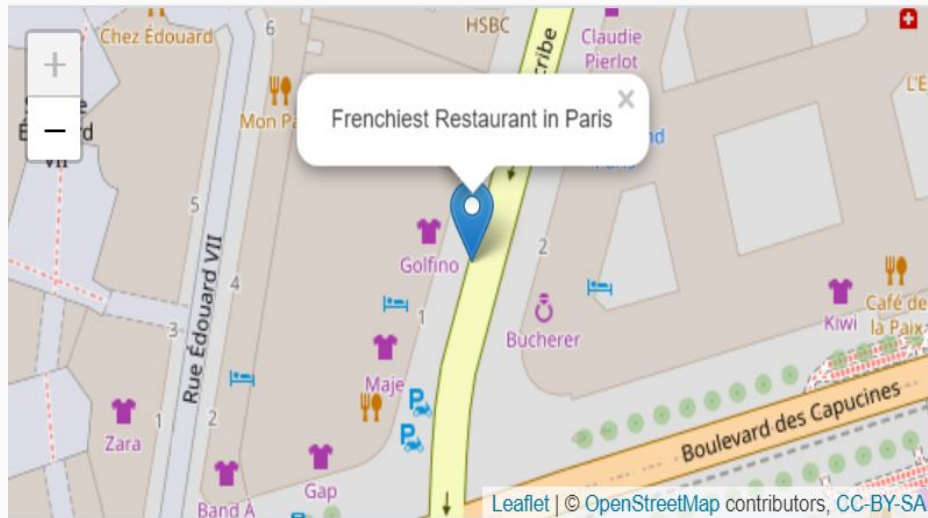


```
10 leaflet() %>%
11   addTiles() %>%
12   addProviderTiles("Leaflet.GIBS") %>%
13   addMarkers(lat = 25.34,
14             lng = 29.34)
```

Layer Features: Markers, labels and popups

The ***addMarkers()*** function can be used to pinpoint specific locations on a map, using spatial coordinates, and other tags which identify that location.

```
map <- leaflet() %>%  
  addTiles() %>%  
  addMarkers(lat = 48.8707084280529,  
             lng = 2.330018786839553,  
             popup = "Frenchiest Restaurant in Paris")  
map
```



Layer Features: Lines and Shapes

Add spatial lines and shapes to maps: ***addCircles()*** or ***addRectangles()*** functions.

Suitable to highlight a specific area or to visualize different types of quantities (e.g. population, magnitude of quakes).

```
cities_col <- read.csv(textConnection("
City,Lat,Long,Pop
Bogotá,4.60971,-74.08175,7901653
Medellín,6.25184,-75.56359,2612958
Cali, 3.43722,-76.5225,2280907
Barranquilla,10.96854,-74.78132,1312473
Cartagena,10.39972,-75.51444,1055035
Cúcuta,7.89391,-72.50782,791986
Villaviciencio,4.142,-73.62664,554173
Pasto,1.21361,-77.28111,393009
"))

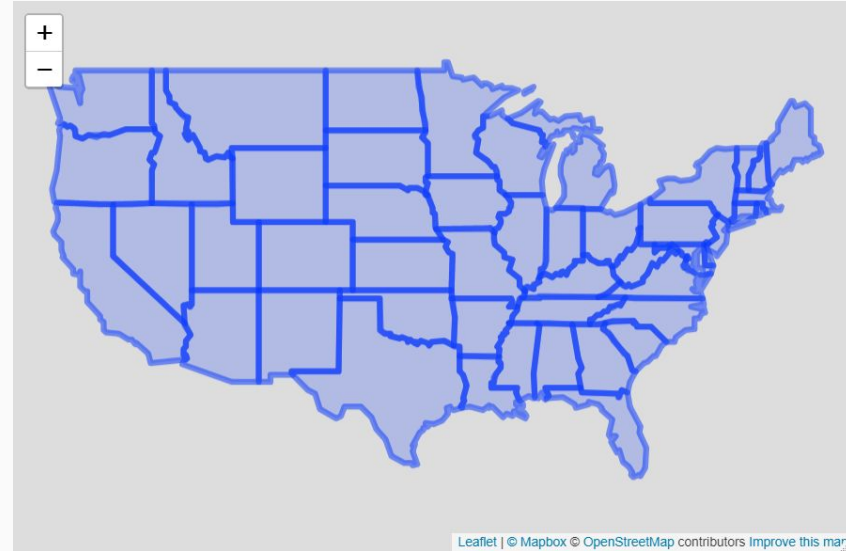
leaflet(cities_col) %>% addTiles() %>%
  addCircles(lng = ~Long, lat = ~Lat, weight = 1,
    radius = ~sqrt(Pop) * 30, popup = ~City)
```



Layer Features: Polygons

Polygons are used to delimit geographical areas in the base maps. These are the foundations for base maps and for Choropleth maps.

addPolygons() is the main function to include uniform polygons.



Layer Features: Colours and Legends

Create a palette of colours and then apply it to the `addPolygons` function.

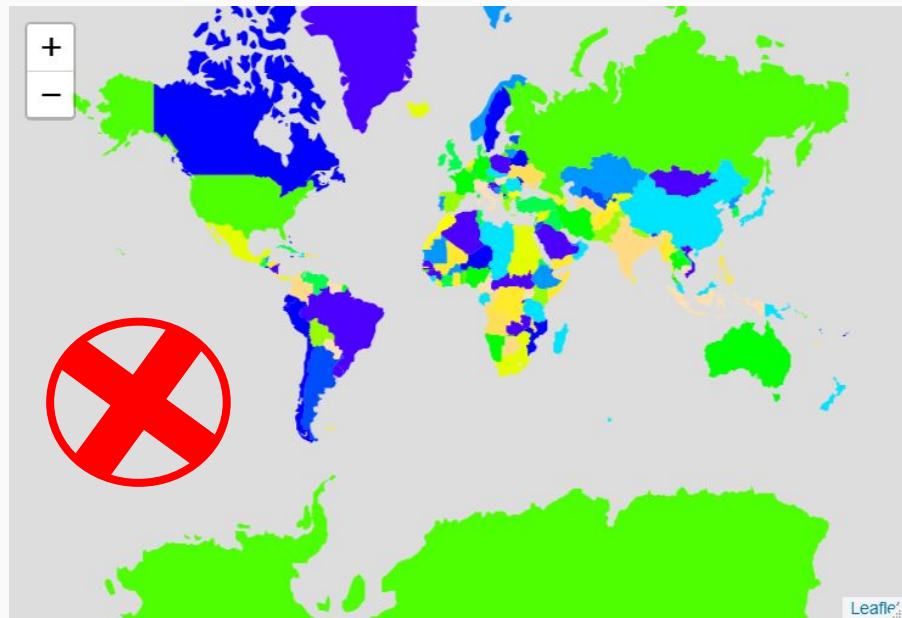
For continuous data:

- **`colorNumeric()`**(Palette, Domain): create a continuous palette.
- **`colorBin()`** for a fixed number of output colours using binning.
- **`colorQuantile()`** for a fixed number of output colours using quantiles.

For categorical data:

- **`colorFactor()`** for a fixed number of category colours.

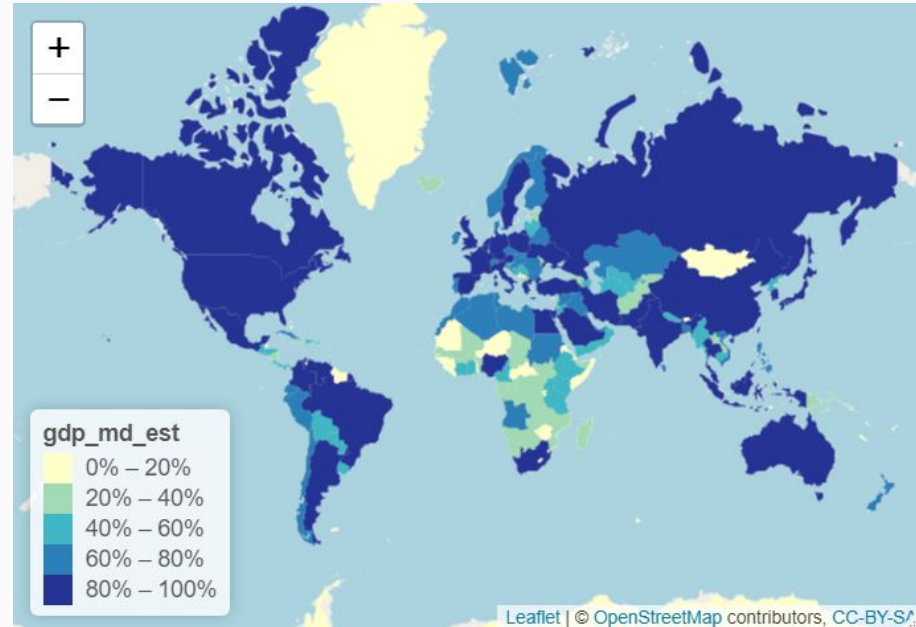
Sometimes a lot of different colours are a bad idea!



Layer Features: Colours and Legends (cont.)

Always include legends after adding colours to your map!

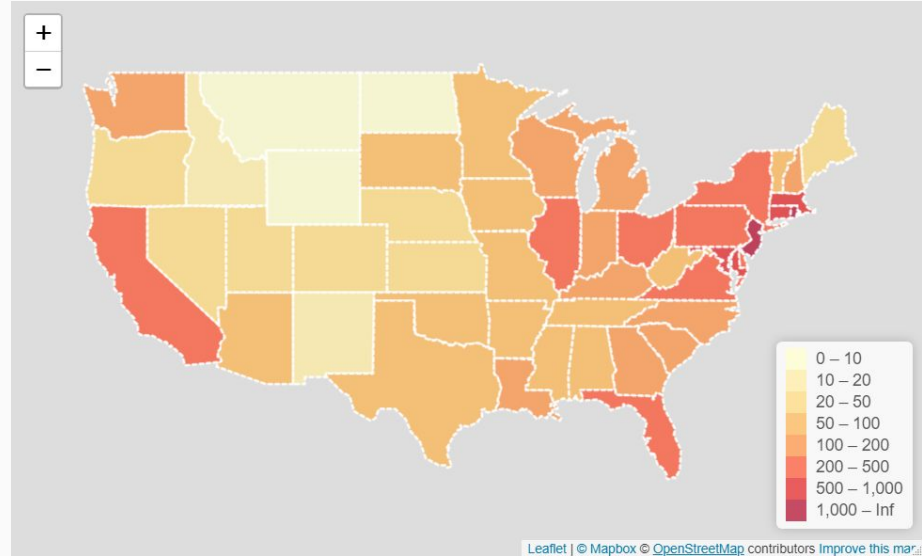
Easiest way to create legends is to provide a palette of colours and values to the **`addLegend()`** function. It will calculate the rest for you!



Layer Features: Choropleths

Choropleth maps are perfect to visualise geographical patterns of a variable or to compare two variables at the same time (e.g. the turnout margin between two candidates in an election).

We will work with Choropleth maps in the live tutorial.



Additional Resources

The complete set of leaflet features can be explored in the official documentation of the [R package](#).

A leaflet [cheat sheet](#) always will be helpful if you get stuck...

In case you want to create interactive maps from your home country, the [Database of Global Administrative Areas](#) will be useful to find a suitable base map.