Hertie School

Introduction to Data Science

Interactive Maps With Leaflet

Florian Winkler and Jiayu Yang

November 4, 2021

Outline

Introduction to Leaflet

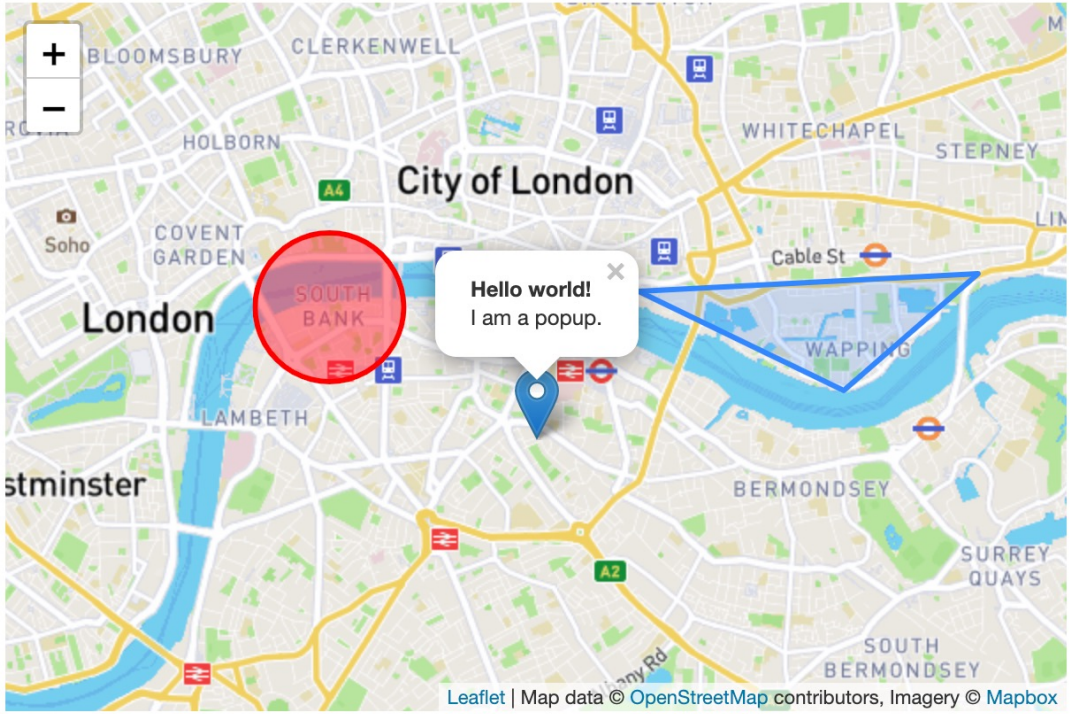
Use cases for Leaflet maps

Leaflet workflow & key features

Further references and resources

What is Leaflet?

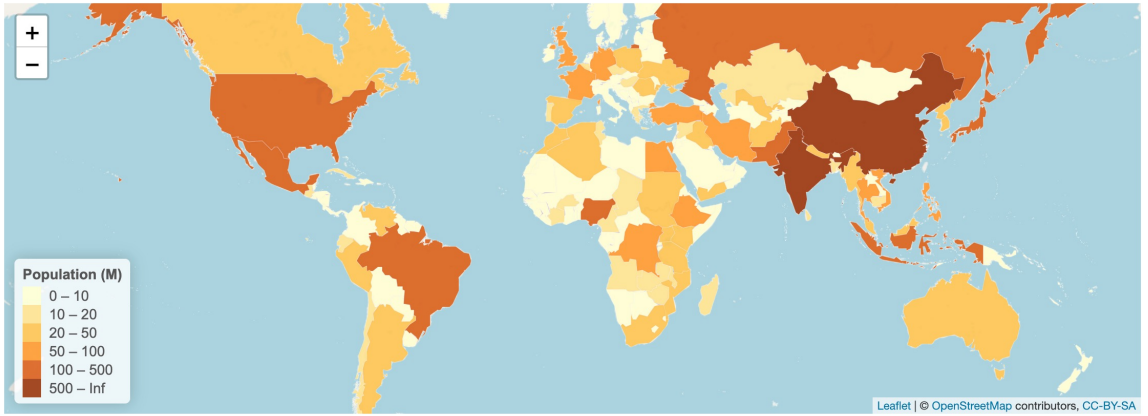
- One of the most popular open-source JavaScript libraries for interactive maps
- Creates embedding maps with tiled base layers, panning and zooming, and layer features
- Used by websites from the New York Times and Washington Post to GitHub and Flickr
- No JavaScript knowledge required: R package makes it easy to create Leaflet maps in R



Why should you use Leaflet?

Leaflet website: "*Leaflet is designed with **simplicity, performance and usability** in mind.*"

- Free and open-source
- Easy to learn: Uses familiar R framework (e.g., pipe operator %>% , similar syntax to *ggplot2*)
- Clear and comprehensive R documentation
- Flexible: Draws on powerful JavaScript library (can be extended with plugins and connected to API)
- Most importantly, Leaflet maps are interactive.

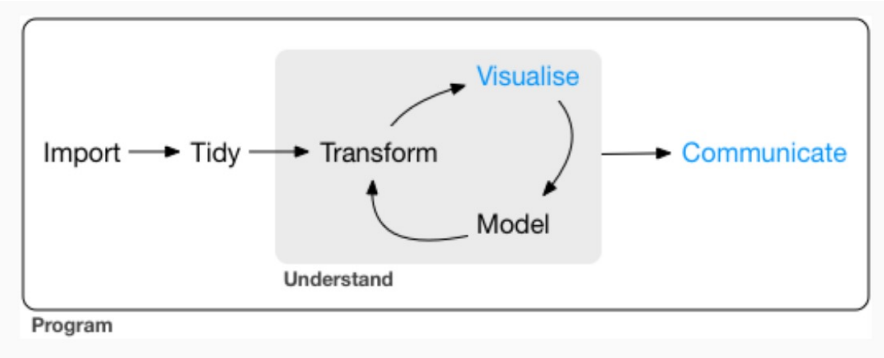


A few caveats before you start:

- Data collection: Spatial data required (e.g., latitudes and longitudes, GeoJSON data, or other formats)
- Designed for interactive environments such as web apps (not pasting images in papers & presentations)
- JavaScript knowledge is necessary for more advanced Leaflet features, API and plugins.

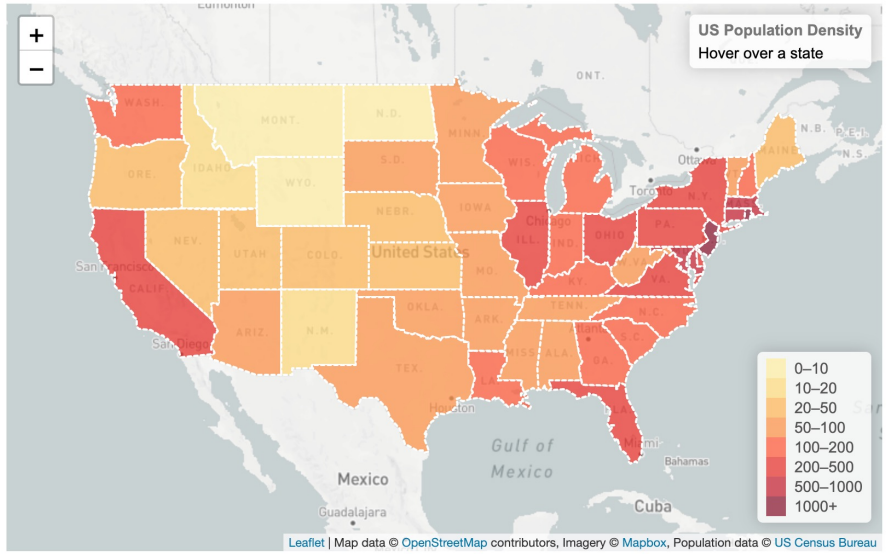
Use cases for Leaflet

Leaflet maps in the data science workflow: To **visualize and communicate** your findings.

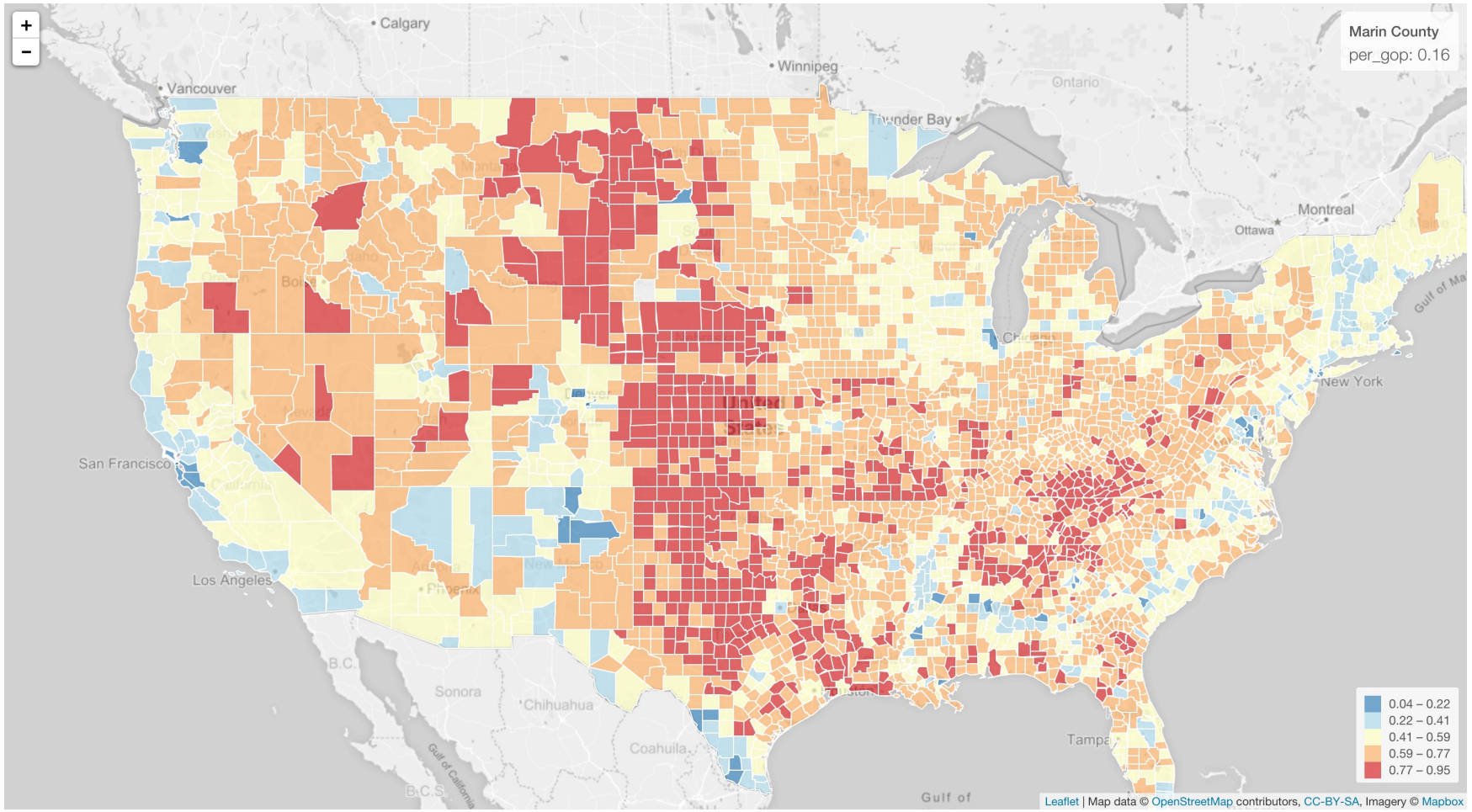


Real-world examples for inspiration:

- Electoral maps (e.g., election results)
- COVID-19 data (e.g., cases across countries)
- Socioeconomic data (e.g., development indicators)
- Climate impact data (e.g., meteorological maps)
- … the sky's the limit.



Use cases: Electoral maps



2016 US Presidential Election Results by County

(percentage of Republican voters)

Image source: *https://blog.exploratory.io/creating-geojson-out-of-shapefile-in-r-40bc0005857d*

COVID-19

Accessibility of COVID-19 vaccination centers in Germany

(motorized travel time in minutes)

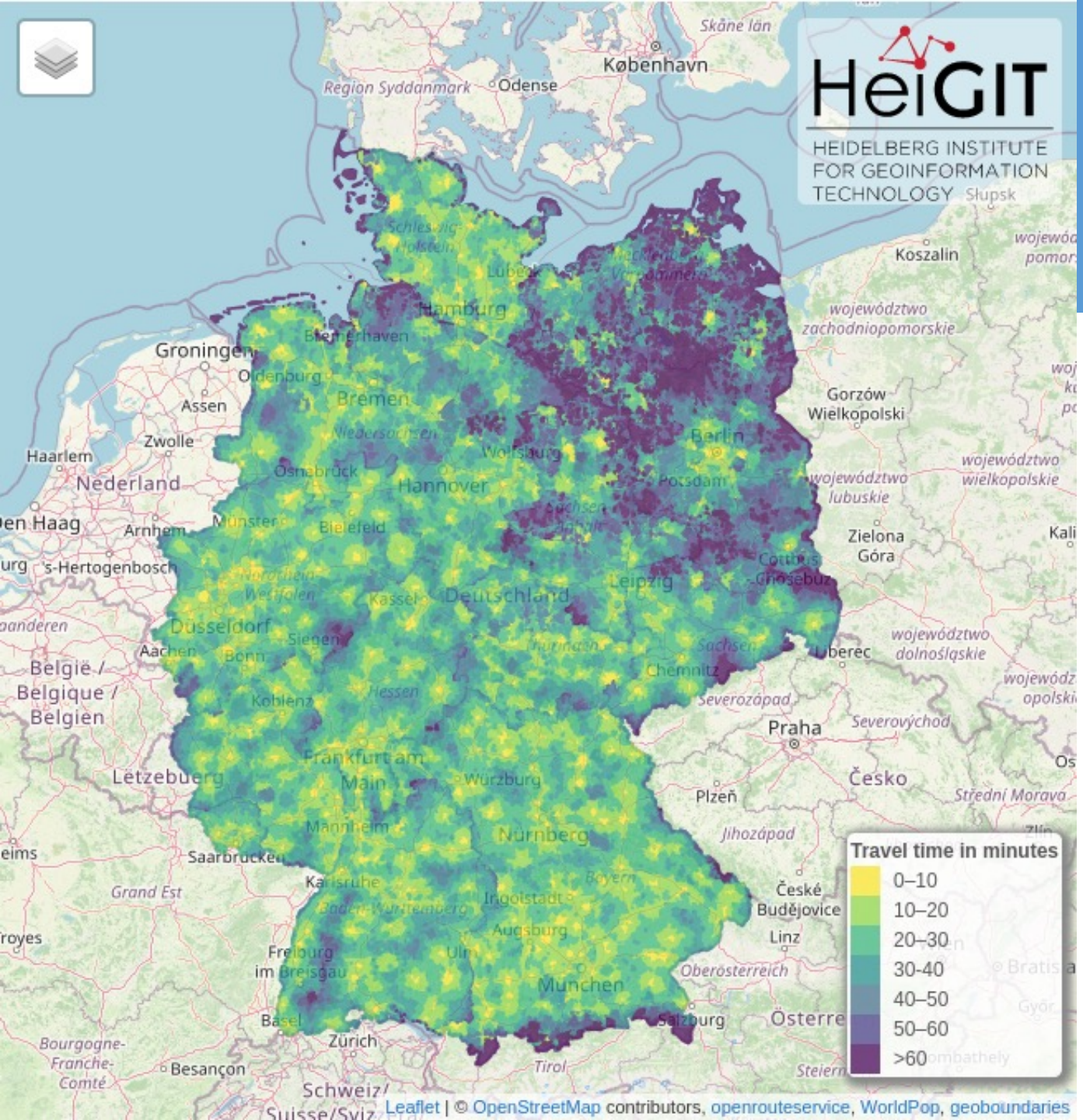
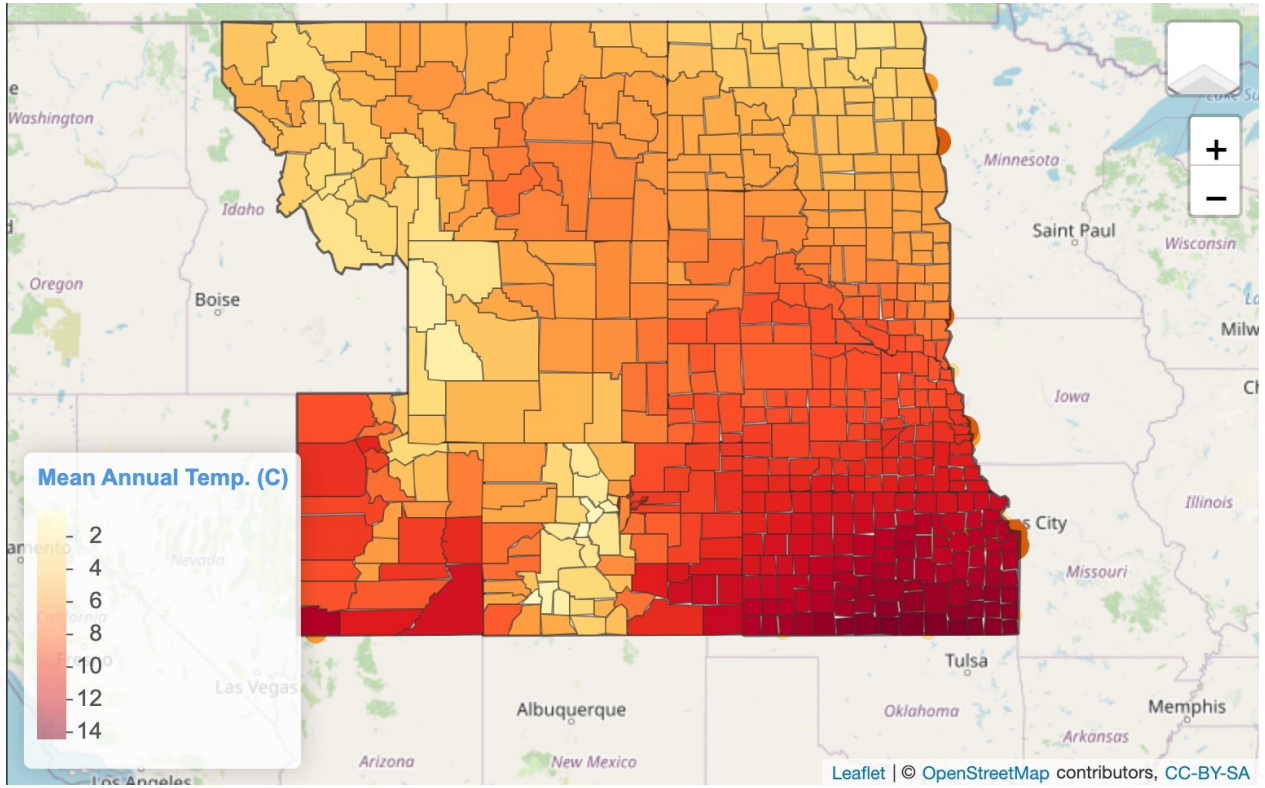


Image source: *http://k1z.blog.uni-heidelberg.de/2021/04/02/german-mass-vaccination-sites-in-open-healthcare-access-map/*

Use cases: Climate data

Mean annual temperatures and forest cover in the High Plains states in the United States



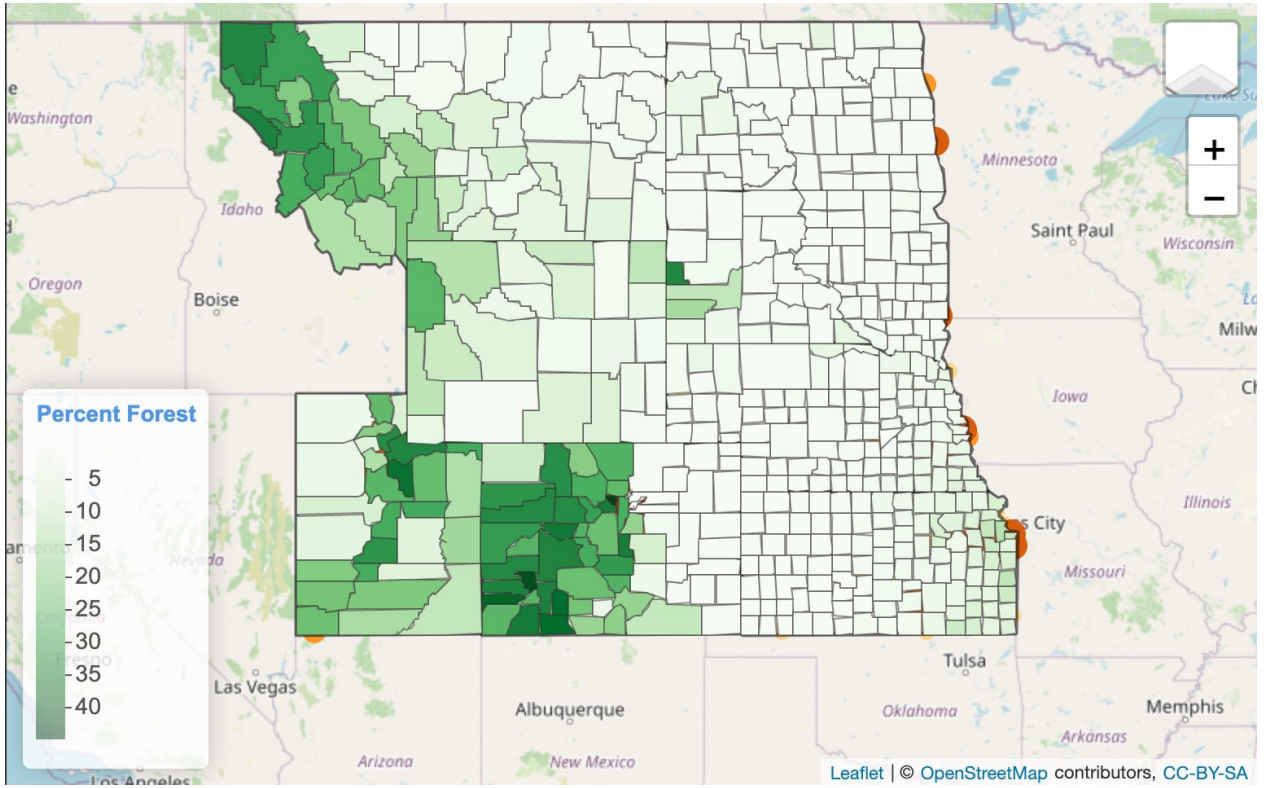
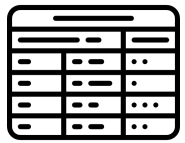


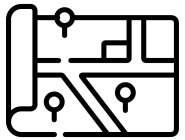
Image source: *http://www.wwview.org/spatial_analytics/leaflet/_site/index.html#content*

Basic Leaflet workflow



Collect Data

- Latitude and longitude columns
- GeoJSON
- Spatial objects from *sp* or *maps* packages



Set map widget

- Basemap (map tiles)
- Map center
- Zoom levels
- Map bounds

Add layer features

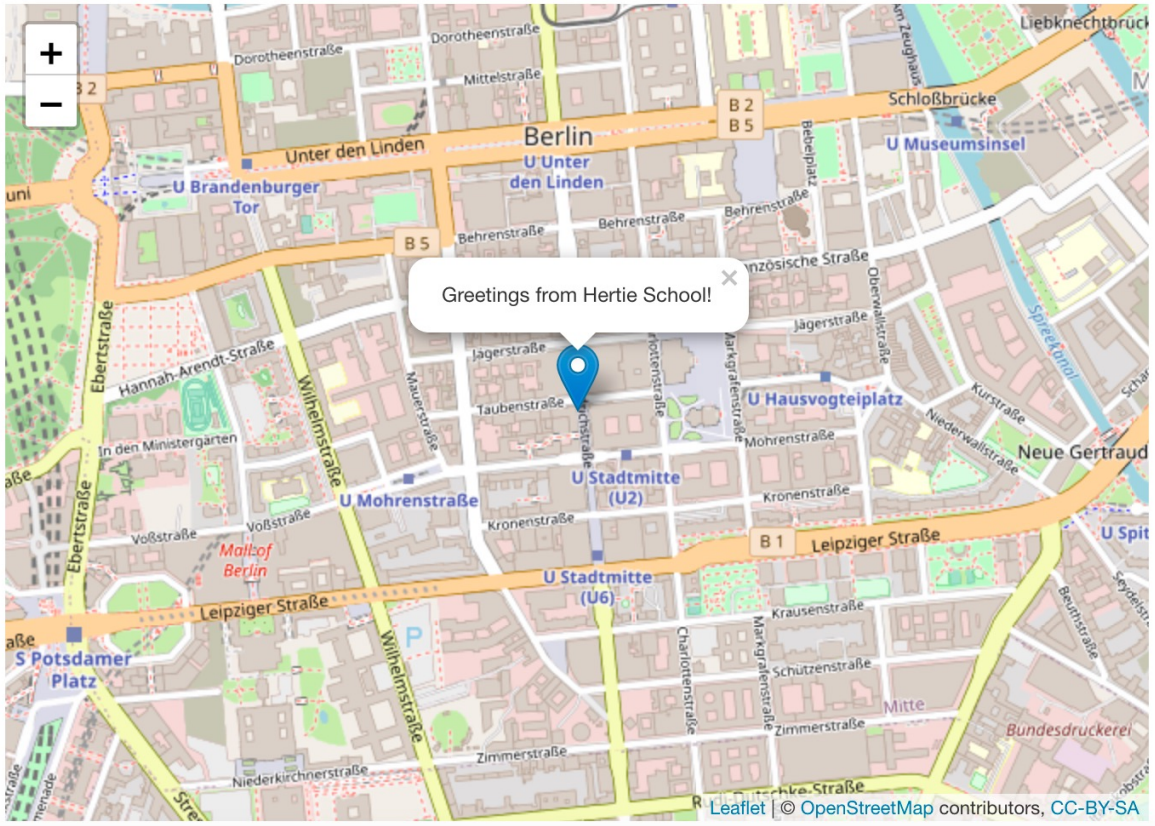
- Markers
- Labels
- Popups
- Polygons
- Lines
- Circles
- Choropleth
- Multiple layers
- … and more

Embed map

- R Markdown
- HTML webpages
- *Shiny* apps

Basic Leaflet syntax

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




```
library(leaflet)

first_map <- leaflet() %>%
  addTiles() %>%  # Add default OpenStreetMap map tiles
  addMarkers(lng=13.3893204,
             lat=52.5128055,
             popup="Greetings from Hertie School!")
first_map  # Print the map
```

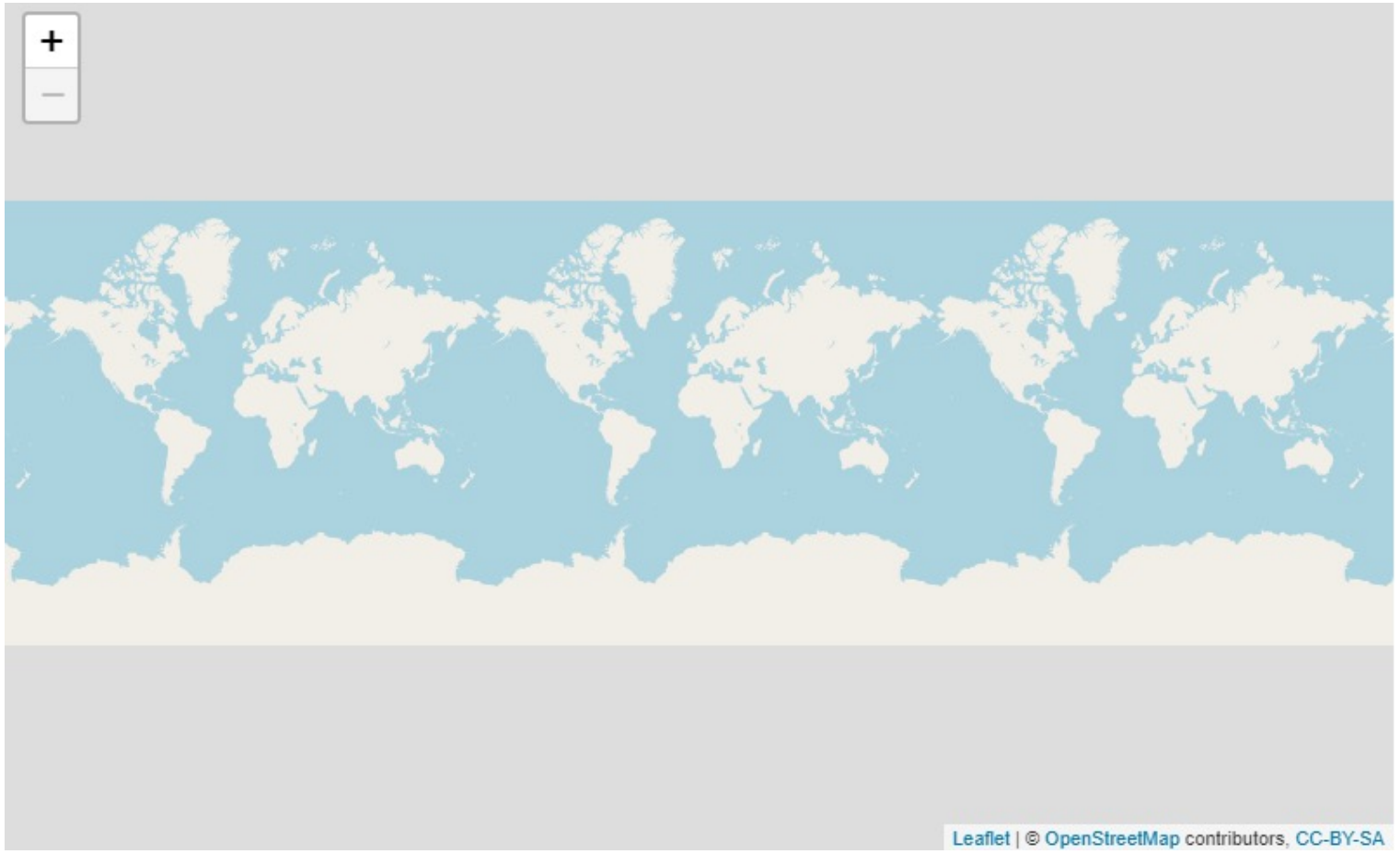
1. Create a map widget by calling *leaflet()*.
2. Add layers (i.e., features) to the map by using layer functions (e.g. *addTiles, addMarkers, addPolygons*).
3. Repeat step 2 as desired.
4. Print the map widget to display it.

Basemaps: Default map tiles

Similar to Google Maps, Leaflet supports basemaps (i.e. the underlying map) using map tiles.

We can use the default basemap, which is *OpenStreetMap*, using ***addTiles()*** with no arguments:

```
leaflet() %>%
  addTiles()
```



Basemaps: Third-party map tiles

You can also use one of many useful and interesting third-party tiles with the ***addProviderTiles()*** function:

```
leaflet() %>%
  setView(lat=52.5, lng=13.3, zoom=3) %>%
  addTiles() %>%
  addProviderTiles("NASAGIBS.ViirsEarthAtNight2012")
```

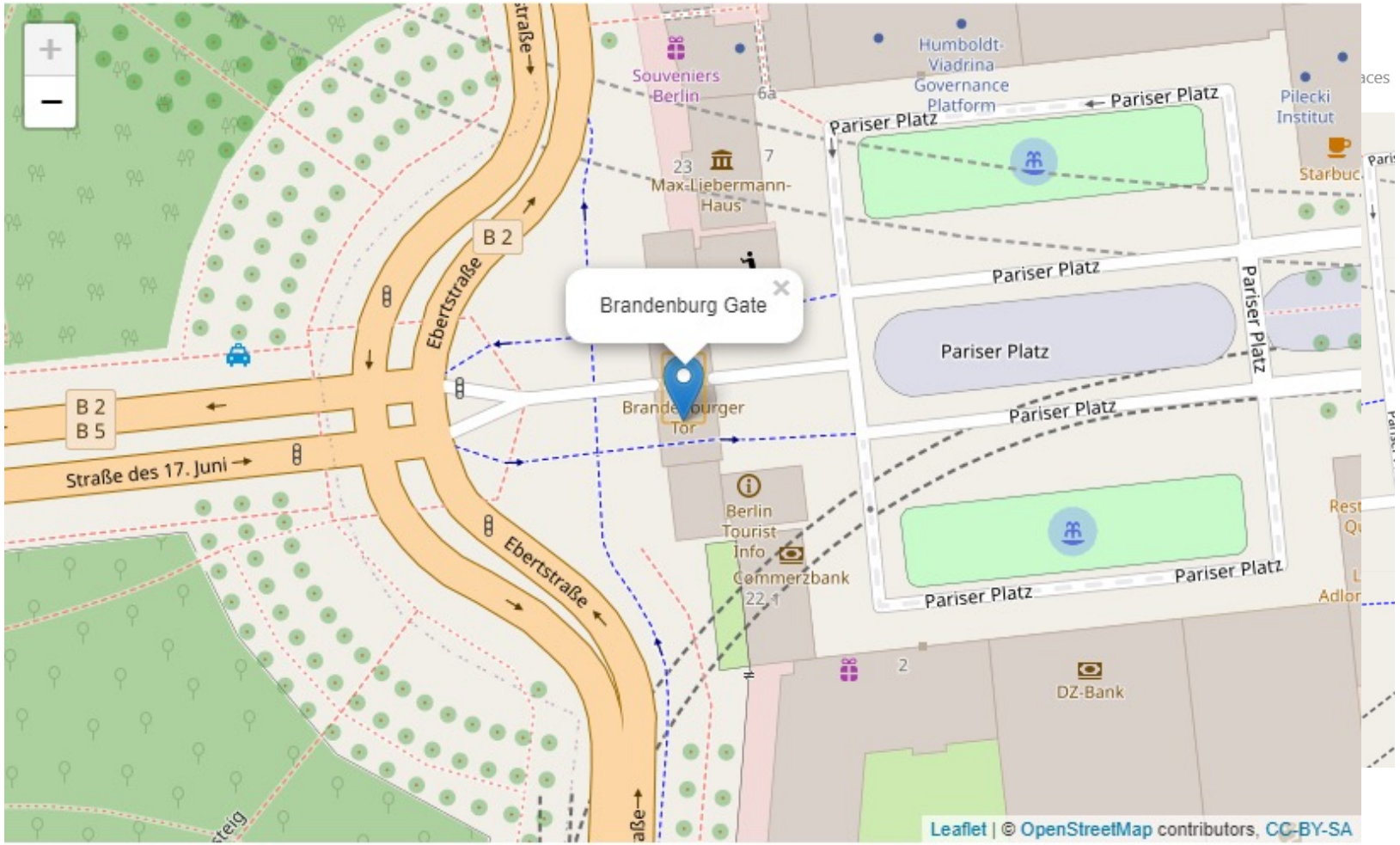
Basemaps: Third-party map tiles

```
leaflet() %>%
  setView(lat=52.5, lng=13.3, zoom=3) %>%
  addTiles() %>%
  addProviderTiles("providers$CartoDB.Positron")
```

Layer features: Markers & Popups

We can use ***addMarkers()*** to specify locations on a map with **markers** and add **popups** to show text when markers are clicked.

```
leaflet() %>%
  addTiles() %>%
  addMarkers(lng=13.377775,
             lat=52.516266,
             popup="Brandenburg Gate")
```



Layer features: Labels

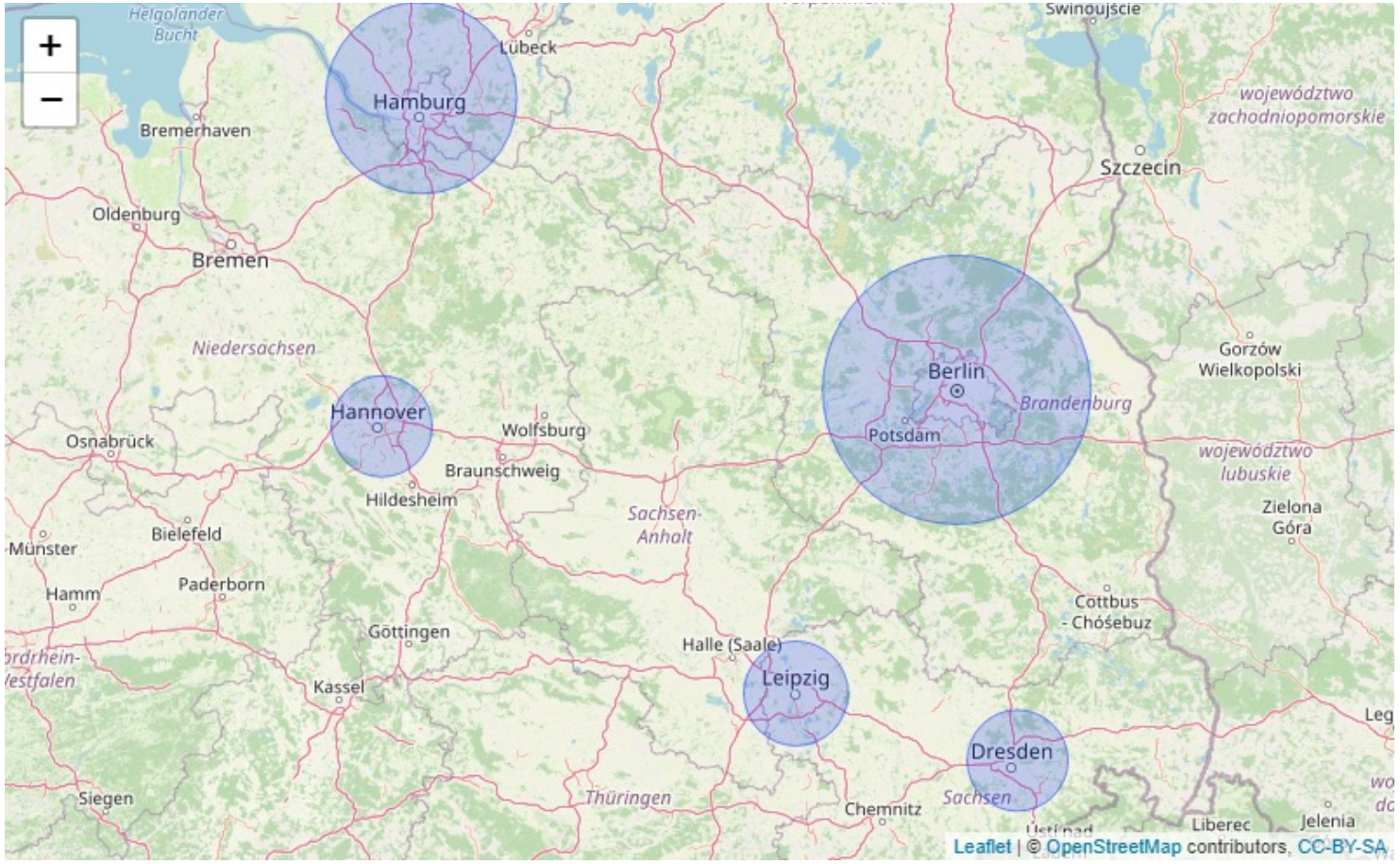
We can also use ***addMarkers()*** to add **labels.** Unlike popups, you don't need to click a marker for the label to be shown. Instead, labels are displayed on mouse over.

```
data %>%
  leaflet() %>%
  addProviderTiles(providers$CartoDB.Voyager) %>%

  addMarkers(lng = ~Longitude,
             lat = ~Latitude,
             label = ~country)
```

Layer features: Circles

Circles can be added using ***addCircles()***.

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

             popup = ~City)
```

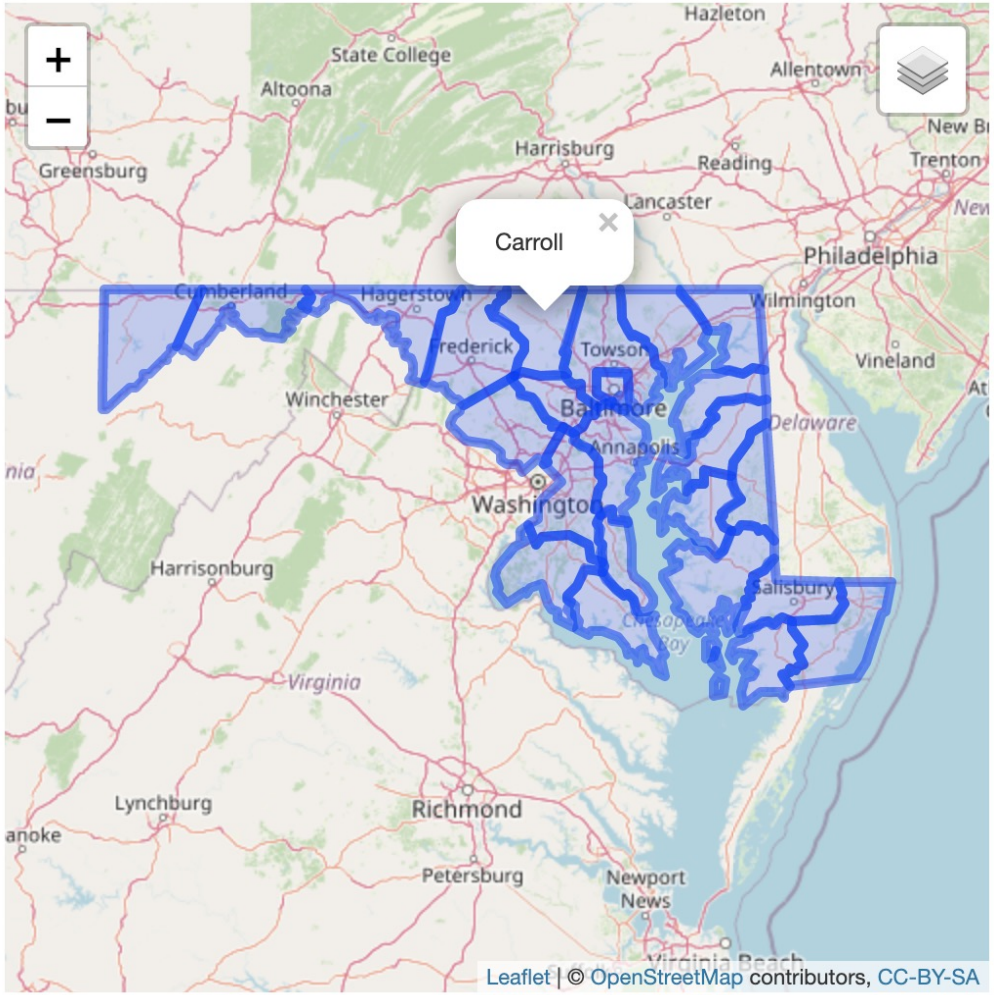


Layer features: Polygons

In mathematics, a polygon is defined as a **two-dimensional** figure with **three or more sides** (e.g., triangles or octagons).

In the GIS world, polygons often have often **irregular lines** (e.g., country borders, lakes).

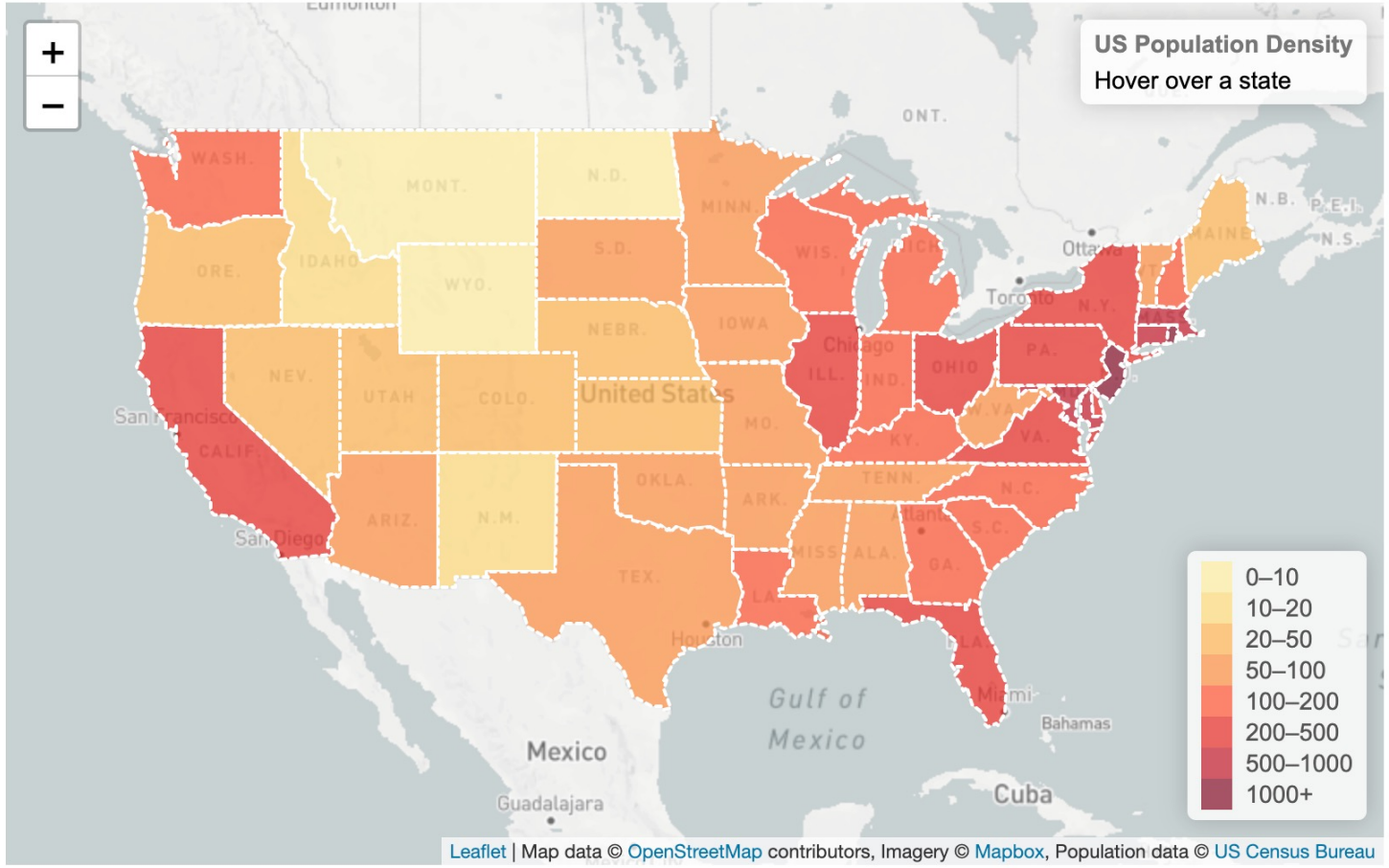
Polygons lay the groundwork for creating **choropleth maps** (see next slide).



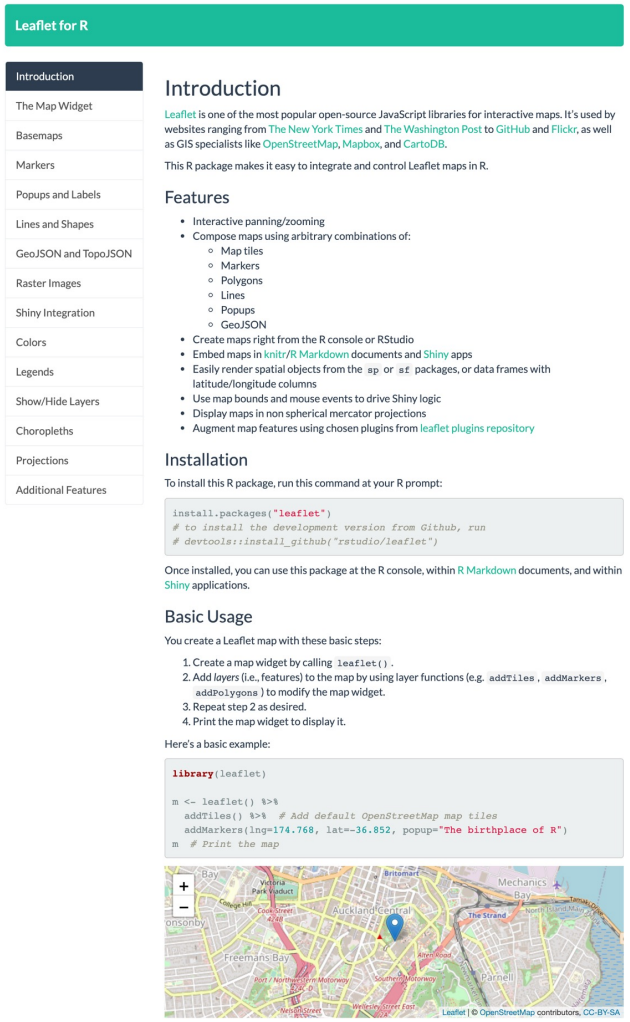
Layer features: Choropleths

Choropleth maps display geographical areas or regions that are **colored or shaded according to a variable**.

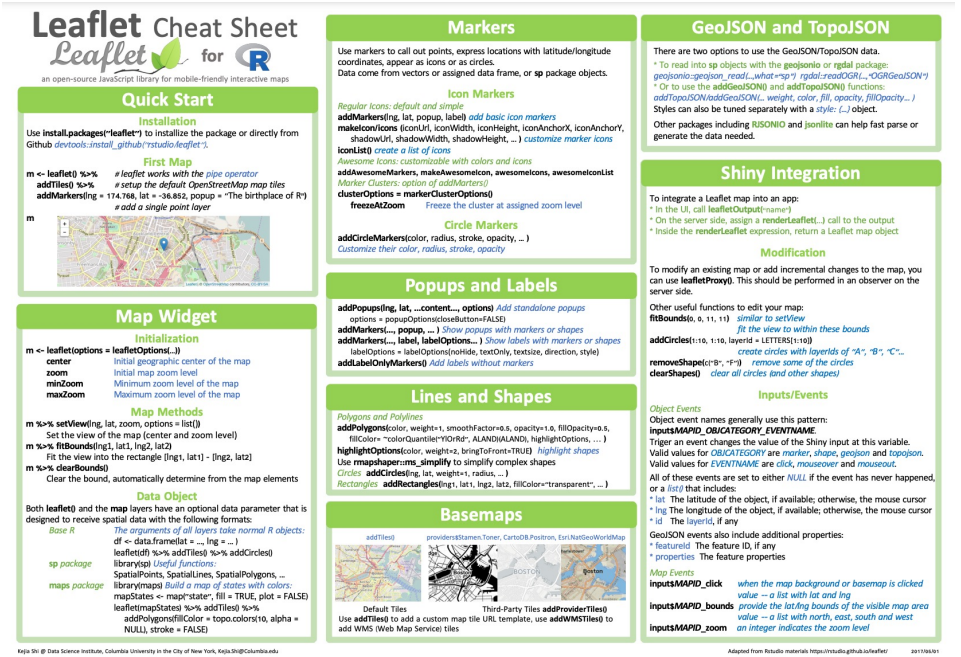
Using **color progression**, choropleth maps provide a way visualize **variation or patterns** across locations.



Further references and resources (1)

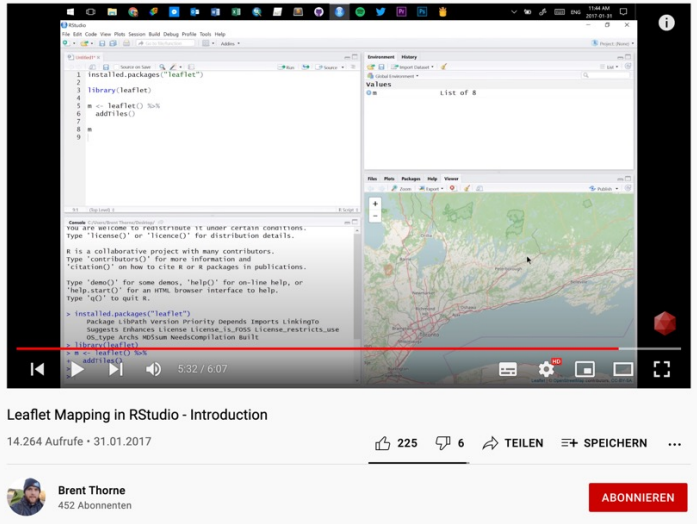


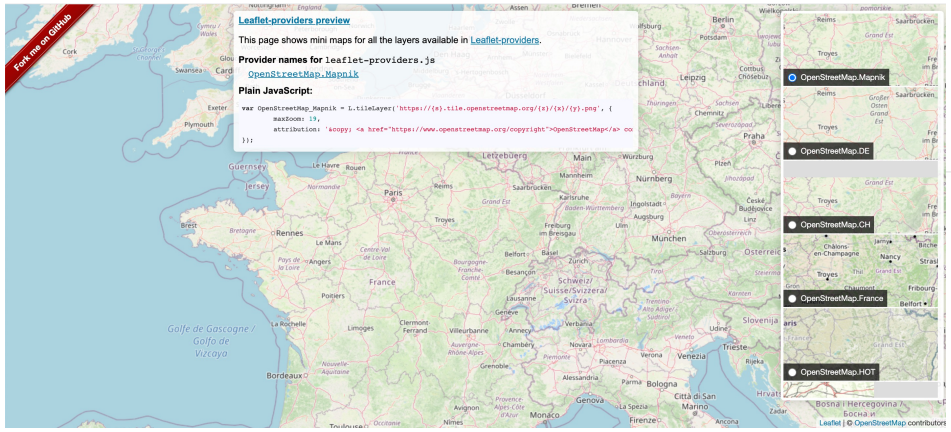
- Official documentation and tutorial for [Leaflet in R]
- Leaflet [cheat sheet]

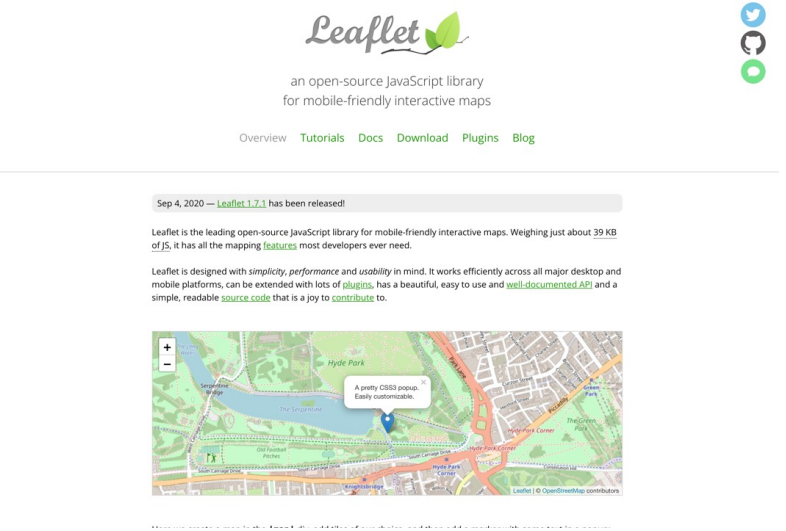


Further references and resources (2)

- Hands-on YouTube tutorial for basic Leaflet features
- Collection of free third-party basemaps (map tiles) for Leaflet
- Leaflet website for JavaScript







Coming up: Live tutorial