Interactive maps with leaflet

Rex Parsons

2022-05-22

Contents

Chapter 1

Prerequisites

This book is intended as a non-comprehensive guide to developing interactive maps with leaflet and shiny. This is by no means comprehensive as it is based on methods that were used in developing theiTRAQI shiny app. However, since this book does focus on the applied problem of developing the iTRAQI shiny app, it includes specific help and methods for these are described here that may be otherwise difficult to find.

For a more comprehensive introduction to leaflet, see the leaflet documentation.

For a more comprehensive introduction to shiny, see the Mastering Shiny book

Chapter 2

Introduction

This book focuses on using `leaflet` and `shiny` together to make interactive maps.

Here's a simple leaflet map.

```
library(leaflet)

leaflet() %>%
  addTiles() %>%  # Add default OpenStreetMap map tiles
  addMarkers(lng=174.768, lat=-36.852, popup="The birthplace of R")
```

Before we begin adding to this map, we need to create the layers that we want to add.

In the iTRAQI app, we used markers, rasters and polygons to show the key locations and interpolations.

See the iTRAQI shiny app here and read more about it in the information tab of the app.

Chapter 3 will focus on these first steps, before making any maps or interactivity. If you're already well-versed in making these layers and the `sf` R package, you can skip to the latter chapters.

2.1 leaflet layers

- To display statistical area level 1 and 2 (SA1 and SA2) regions on the map, we will be using `sf` objects with MULTIPOLYGON geometries. These are multipolygons because some of these areas include distinct areas, such as a set of islands, that aren't contained within a single polygon.

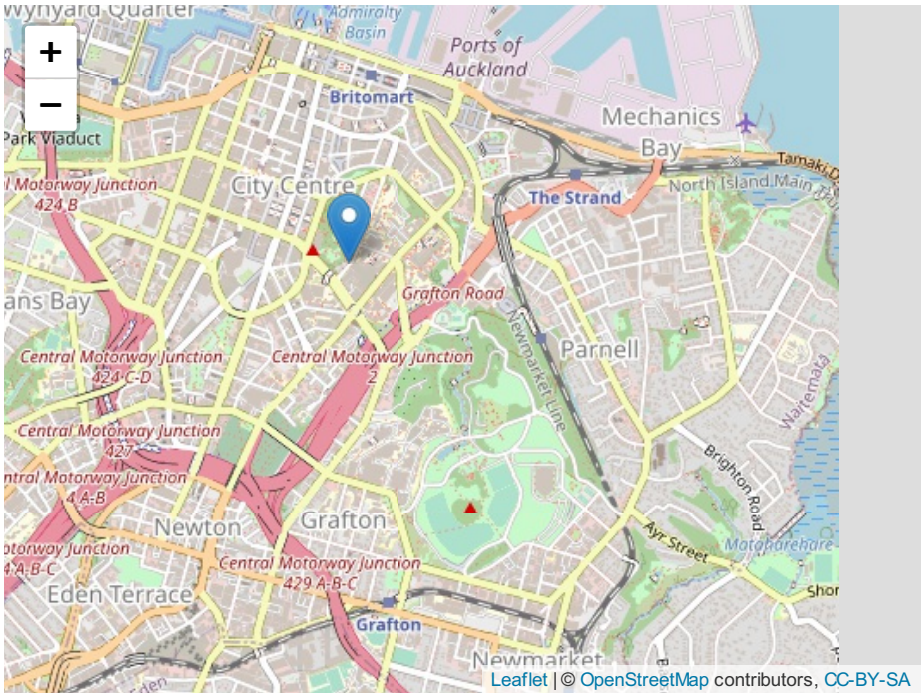

Figure 2.1: Simple leaflet map

- To display the location of acute and rehab centers and town locations with travel times that we used for interpolations, we used (spatial) data.frames that had longitudes and latitudes for their location.
- To display the continuous interpolations, we used `RasterLayer` objects.

Using a polygon and raster layer that's used in the iTRAQI map and some markers in a data.frame, we can make see the basic approach that we use to display these on a leaflet map.

First, lets make a data.frame with the coordinates for the Princess Alexandra and Townsville University Hospitals, and download a raster and polygon layer from the iTRAQI app GitHub repository.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(sf)
```

```
## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
download_layer <- function(layer_name, save_dir="input") {
  githubURL <- glue::glue("https://raw.githubusercontent.com/RWParsons/iTRAQI_app/main/input/laye
  download.file(githubURL, file.path(save_dir, layer_name), method="curl")
  readRDS(file.path(save_dir, layer_name))
}

raster_layer <- download_layer("rehab_raster.rds") %>%
  raster::raster(., layer=1)

polygons_layer <- download_layer("stacked_SA1_and_SA2_polygons_year2016_simplified.rds")
polygons_layer <- polygons_layer[polygons_layer$SA_level==2, ] # show SA2 regions for example

marker_locations <- data.frame(
```

```
  centre_name=c("Princess Alexandra Hospital (PAH)", "Townsville University Hospital")
  x=c(153.033519, 146.762041),
  y=c(-27.497374, -19.320502)
)
```

Here, in figure 2.2, we make a leaflet map with the three object types. We will use these three functions, `addPolygons()`, `addRasterImage()`, and `addMarkers()` to add almost all of the content to our leaflet maps.

```
leaflet() %>%
  addProviderTiles("CartoDB.VoyagerNoLabels") %>% # add a simple base map
  addPolygons(
    data=polygons_layer,
    fillColor="Orange",
    color="black",
    weight=1,
    group="Polygons"
  ) %>%
  addRasterImage(
    x=raster_layer,
    colors="YlOrRd",
    group="Raster"
  ) %>%
  addMarkers(
    lng=marker_locations$x,
    lat=marker_locations$y,
    label=marker_locations$centre_name,
    group="Points"
  ) %>%
  addLayersControl(
    position="topright",
    baseGroups=c("Polygons", "Raster", "Points"),
    options=layersControlOptions(collapsed = FALSE)
  )
```

Almost all of these objects were made before being used in the shiny app. Chapter 3 will introduce the methods used to make them. Chapter 4 will introduce the basics of a shiny app. Chapter 5 will introduce the more specific methods that were used to construct the iTRAQI app itself.

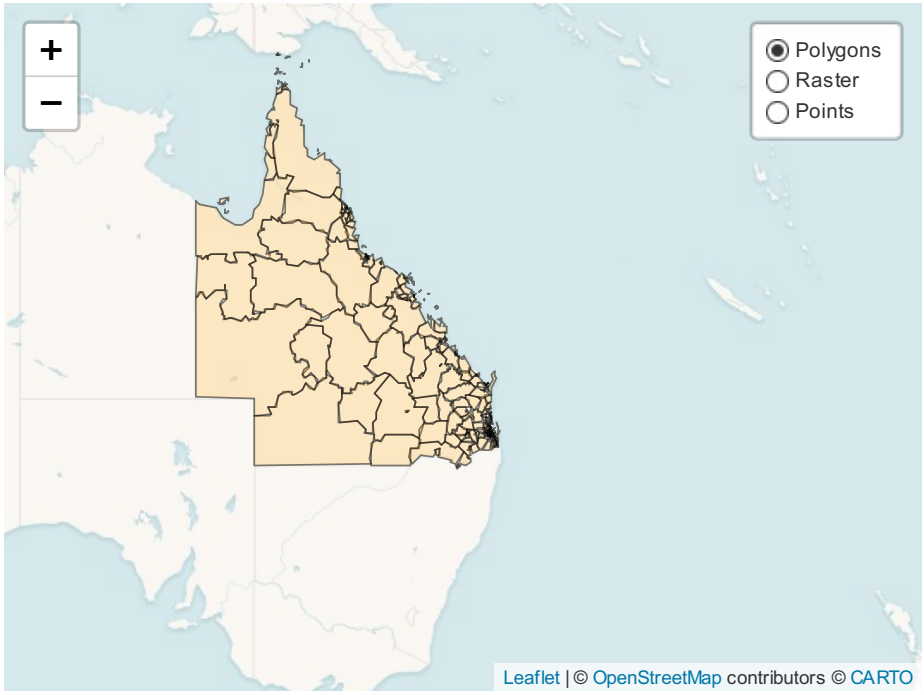

Figure 2.2: leaflet map with polygons, rasters and markers

Chapter 3

Creating the layers

This chapter will cover the necessary steps to make layers which will be visualised in the app:

- kriging
- spatial joins
- aggregating interpolations within polygons

Chapter 4

An app with a map

This chapter will have a brief intro to shiny with a map:

- ui and server
- reactivity
- leaflet
- leafletproxy

Chapter 5

An app with a map

This chapter will have a brief intro to shiny with a map:

- ui and server
- reactivity
- leaflet
- leafletproxy