Week 7 - GIS with leaflet & sf in R

DATA201/422

2024-09-09

Overview

In this lab, we'll introduce GIS (Geographical Information Systems) concepts using the leaflet and sf packages in R. GIS is crucial in data science for visualizing and analyzing spatial data. Today, we'll walk through various examples using population-level data from New Zealand, including basic maps, heatmaps, and choropleth maps. You'll get the chance to apply these techniques to create your own customized map.

Deliverables

Your task is to create different types of maps using the leaflet and sf packages. By the end of this lab, you should be able to:

- Create a basic map centered on New Zealand in leaflet.
- Add markers to the map for major cities.
- Generate a heatmap based on population data for the major cities.
- Create a choropleth map using New Zealand Statistics data using sf and ggplot.
- Customize a map of your own, combining all the techniques you've learned.

Use the resources below as a source of information to help yourself!

Resources

Here are some resources to help you along the way:

leaflet for R: https://www.geeksforgeeks.org/leaflet-package-in-r/

leaflet.extra provider tiles: http://leaflet-extras.github.io/leaflet-providers/preview/index.html

sf for R: https://r-spatial.org/book/07-Introsf.html

Understanding geojson data in R: https://rstudio.github.io/leaflet/articles/json.html

(All of the spatial data processing has been completed for you - count yourselves lucky!)

StatsNZ: https://datafinder.stats.govt.nz

Basic Map

With just longitude and latitude, it is possible to create a map in R using leaflet. Using the code below, you can produce a tiled map of Ireland. Remember it's possible to zoom in and out on the generated map.

```r
ire_map <- leaflet() %>%
  addTiles() %>%
  setView(lng = -8.240473, lat = 53.419751, zoom = 6)
ire_map
```

Task 1: Using the template above and Google to find the necessary latitude and longitude coordinates, create a map of New Zealand. You can adjust `zoom` to ensure the whole country is displayed.

Adding Markers Next, we'll add markers to our map to represent major cities in Ireland with their populations. First, we need to create a dataset of the city latitude and longitude coordinates, and the population count per city. We then layer the data on top of the map.

```r
# Create Ireland map dataset
ire_pop <- data.frame(
  city = c("Dublin", "Cork", "Galway"),
  lat = c(53.350140, 51.903614, 53.270962),
  lng = c(-6.266155, -8.468399, -9.062691),
  population = c(592713, 124391, 85910)
)

# Create Ireland map with makers
ire_map_pop <- leaflet(ire_pop) %>%
  addTiles() %>%
  setView(lng = -8.240473, lat = 53.419751, zoom = 6) %>%
  addMarkers(~lng, ~lat, popup = ~paste(city, "<br>Population:", population))
ire_map_pop
```

Task 2: Using the template above add markers to your New Zealand map for Auckland, Wellington and Christchurch.

You need to work out (Google!) the latitude and longitude coordinates, as well as the population count for each city.

Create a dataset with this information.

Furthermore, using `addProviderTiles` from leaflet.extras add a (sensible) provider tile to your NZ map - the map needs to be clear and readable.

Heatmap Task 3: After that, using the `addHeatmap` function, produce a heatmap that displays the population density in each city.

Choropleth Map

New Zealand's government makes it pretty easy to find and download their wide range of survey data like population, agriculture, business, residents surveys, etc. at StatsNZ. Also, they provide a series of the administrative or census boundaries data at different levels in ESRI Shapefile, ESRI geodatabase, and MapInfo TAB formats as complimentary so that we can visualize their survey data geographically. This is super awesome! You can download them from various locations online.

Two inputs are needed to build a choropleth map:

1. A geospatial object providing regional boundaries (territories in New Zealand in this example). The geospatial data is provided for you: ??? You'll practice how to read and represent geoJSON format with R.

2. A numeric dataset that you will use to color each geographical unit. Here you will use the population count for each NZ regional territory. This data has been provided for you: ???

After you have read in both inputs to R, it is **INCREDIBLY IMPORTANT** that you merge the two inputs before generating your choropleth map!

Read a .geoJSON file Task 4: The sf package allows you to read a geoJSON file into R. You will need to set the path to the geoJSON that has been provided for you, and then use the `st_read` function from the sf package to read the geoJSON file into an sf object.

Create a black and white plot with your regions Task 5: For plotting with ggplot2, the `geom_sf()` function allows you to represent an sf object in a map. Use the template below to generate a black and white plot of the NZ territories.

```
ggplot() +
  geom_sf() +
  theme_void()
```

Load your numeric data The population count of each territory has been found on the internet and a clean version has been provided for you (again, you're welcome). It is thus easy to read it with `read.csv()`. Before doing a choropleth map, it is a good practice to check the distribution of your variable.

Here, we have a long tail distribution: one territory in particular has a huge population count compared to others. Thus it's good practice to apply a log scale to the distribution and the color palette in the final map. It will avoid that all the variation is absorbed by these high values.

Task 6: Load the population data. Clean the data by:

- Column names
- Remove unnecessary data point(s)
- Check the distribution of population across territories

Merge geospatial and numeric data Task 7: This is a key step in creating choropleth maps. The two data sources (the geospatial data and the numeric population data) must be linked. Use a `left_join` to do this.

Make basic choropleth Task 8: Generate a first choropleth map with non-transformed population count. Building on your basic above by add `fill = your_population_count` in the aesthetic of the polygons.

```
ggplot() +
  geom_sf(aes(fill = )) +
  theme_void()
```

This map is extremely basic - jazz it up!

Task 9: We need to change the color palette, improve the legend, use a log scale transformation for the color scale, change background and add titles and explanation. Here is a template for the code to do all that. Some of the include functions will help you - you don't need to use them all - there included as suggestions.

Make sure to save your jazzy plot as a png file.

```r
nz_plot <- ggplot() +
  geom_sf(aes(fill = )) +
  theme_void() +
  scale_fill_viridis_c() +
  labs() +
  theme(
    text = ,
    plot.background = ,
    panel.background = ,
    legend.background = ,
    plot.title = ,
    plot.subtitle = ,
    plot.caption = ,
    legend.position =
  )

# View plot
nz_plot

# Save plot as png file
ggsave("nz_plot.png")
```

Extra for experts

Go further: Create an interactive choropleth map using the plotly library. In this map, you need to jazz up the colour pallette also. Please style your text so it fits the scope of the graph. Once you have successfully created a map, call over your tutor for marking. Please see *plotly* and *RColorBrewer* to assist you.

If you want you can use other data from StatsNZ (you'll need to clean it first) or use the population change data already provided in the numeric dataset above. Feel free to do this if you'd like