Week 7 - GIS with leaflet & sf in R: Submission

David Ewing (82171165)

2024.09.09

Introduction

This report provides an overview of the work completed for Week 7's lab assignment involving the usage of `leaflet` and `sf` packages in R to create various maps. The objective was to have the autor of the paper demonstrate the ability to visualise population data in New Zealand through interactive and static maps. The author also took the opportunity to expand beyond what was required in the assignment and to have the code include titles for each deliverable, and improve map presentation, including the addition of titles via the `addControl()` function for `leaflet` maps. It is not expected that this report will be read by the tutors.

Methods

The tasks were completed by following the assignment's requirements, creating six distinct deliverables. Each section of the code has been provided with explanations and a brief discussion of the improvements made. This includes adding titles directly to the maps using custom HTML for `leaflet` and using `ggplot2` for choropleth maps.

Deliverable 1: Create a Basic Map Centred on New Zealand

The first task was to generate a basic map centred on New Zealand using `leaflet`. A custom title was added using the `addControl()` function to clearly indicate the map's purpose:

```
nz_leaflet_map <- leaflet() |>
  addTiles() |>
  setView(lng = 174.0, lat = -40.9, zoom = 5) |>
  addControl(
    html = "<h3>Deliverable 1: Create a basic map centered on New Zealand ...</h3>",
    position = "topright"
  )
```

This map shows the default OpenStreetMap tiles, centred on New Zealand.[1]

Deliverable 2: Add Markers for Major Cities

The next step involved adding markers for major cities in New Zealand: Auckland, Wellington, and Christchurch. The markers were enhanced by adding a title to the map using `addControl()`:

```
nz_cities_map <- leaflet(nz_cities) |>
  addTiles() |>
  setView(lng = 174.0, lat = -40.9, zoom = 5) |>
  addMarkers(~lng, ~lat, popup = ~paste(city, "<br>Population:", population)) |>
  addControl(
    html = "<h3>Deliverable 2: Add markers to the map for major cities</h3>",
    position = "topright"
  )
```

The markers allow users to interactively view population information for each city.[2]

[1] A basic map in `leaflet` displays geographic tiles but doesn't include any data-specific visualisations.

[2] A marker map in `leaflet` adds markers for geographic locations and typically includes pop-ups with information.

Deliverable 3: Generate a Heatmap for Major Cities

A heatmap was created to represent the population density of major cities using `leaflet.extras`. A custom title was added, as before, using the `addControl()` function:

```
nz_heatmap <- leaflet(nz_cities) |>
  addTiles() |>
  setView(lng = 174.0, lat = -40.9, zoom = 5) |>
  addHeatmap(lng = ~lng, lat = ~lat, intensity = ~population, radius = 15, blur = 20) |>
  addControl(
    html = "<h3>Deliverable 3: Generate a heatmap based on population ...</h3>",
    position = "topright"
  )
```

The heatmap visualises population density with more intense colouring for cities with higher populations.[3]

Deliverable 4: Create a Choropleth Map Using sf and ggplot2

To visualise population data across New Zealand's territories, a choropleth map was created using the `sf` and `ggplot2` packages. Titles were added directly in `ggplot2` using the `labs()` function:

```
choropleth_map <- ggplot(nz_data) +
  geom_sf(aes(fill = population)) +
  scale_fill_viridis_c(option = "C") +
  theme_void() +
  labs(title = "Deliverable 4: Create a choropleth map  ...", fill = "Population")
```

This map displays population density, with territories shaded based on their population figures.[4]

Deliverable 5: Custom Map Combining Techniques

For Deliverable 5, a customisation was required that combined techniques learned throughout the assignment. The custom map was enhanced by applying a logarithmic transformation to the population data, improving visual clarity across territories with widely varying population sizes. The customisation was also styled with a subtitle, colour scaling, and legend modifications:

```
enhanced_map <- ggplot(nz_data) +
  geom_sf(aes(fill = log10(population))) +
  scale_fill_viridis_c(option = "C", direction = -1) +
  theme_void() +
  labs(
    title    = "Deliverable 5: Customize a map of your own, combining all the ...",
    subtitle = "2016 Population Data from Stats NZ",
    fill     = "Log Population",
    caption  = "Source: Stats NZ"
  ) +
  theme(
    plot.title      = element_text(size = 16, face = "bold"),
    plot.subtitle   = element_text(size = 12),
    legend.position = "right",
    legend.title    = element_text(size = 10),
    legend.text     = element_text(size = 8)
  )
```

This customisation improves the readability of the map by using log-scaled population values, enhancing smaller values and visually balancing the map.[5]

[3] A heatmap visualises data density, with the intensity of colours indicating higher values. In this case, population is represented.

[4] A choropleth map uses shaded geographic regions to represent statistical variables such as population density.

[5] A customised choropleth map with logarithmic scaling highlights smaller population values, making the map more readable when large disparities exist between regions.

Deliverable 6: Extra for Experts (Optional)

For the final task, the choropleth map was converted into an interactive map using `plotly`, allowing users to zoom and explore the population data dynamically:

```
interactive_map <- ggplotly(choropleth_map)
```

This interactive map enables users to pan, zoom, and hover over territories for detailed population data.[6]

Challenges and Solutions

Several challenges were encountered during the assignment, which required careful handling. Below are the key issues and solutions:

Adding Titles to Leaflet Maps

The `leaflet` package does not natively support titles, so `addControl()` was used to add HTML-based titles. This approach provided flexibility in title placement and styling.

Handling File Paths and Quiet Mode

When reading geoJSON and CSV files, the output was cluttered with unnecessary details. This was resolved by using the `quiet = TRUE` argument in `st_read()` and suppressing column type messages in `read_csv()` with `show_col_types = FALSE`.

Logarithmic Scaling for Choropleth Map

When dealing with population data, the differences between the values were large. A logarithmic scale was applied to the population data to enhance the visual clarity of the map, making the map easier to interpret across regions with small populations.

Conclusion

This report outlines the process of creating and customising various types of maps in R using `leaflet`, `sf`, and `ggplot2`. Through the use of advanced techniques, such as heatmap generation, marker placement, and logarithmic scaling, the data was presented in a clear and interactive manner. The assignment highlighted the versatility of R for spatial data visualisation and allowed for a deeper understanding of both basic and complex mapping techniques.

[6] An interactive choropleth map allows for dynamic exploration of data through user interaction, such as zooming and panning.