shiny app where users can change values of variables, those scores are summed and ranked, and a map fill symbology updates accordingly based on the aggregated values

To build a Shiny app where users adjust variable scores, see real-time summed and ranked results, and view a dynamically updated map fill symbology, follow this modular approach using reactive logic and `leafletProxy` for the map. This process ensures users interactively change inputs, see rankings update, and visualize on the map without reloading everything.[1] [2] [3]

App Structure

- UI: Use `numericInput`, `sliderInput`, or other widgets for users to set scores per variable/region.[4]
- Server: Aggregate the scores reactively, rank them, and update both the summary table and the leaflet map accordingly.[5] [6] [2]
- Map Updating: Use `leafletProxy` to dynamically adjust polygon fills according to the aggregated score per spatial unit.[2] [3] [7]

Key R Shiny Code Steps

1. **UI Inputs**
   - Define UI widgets for user inputs (e.g., for each region or variable):[6] [4]

```
numericInput("score_region1", "Region 1 Score", value = 0)
numericInput("score_region2", "Region 2 Score", value = 0)
# Repeat for all regions/variables
```

2. **Reactive Aggregation and Ranking**
   - Calculate sum and ranks reactively:[6]

```
region_scores <- reactive({
  data.frame(region = c("Region 1", "Region 2"),
             score = c(input$score_region1, input$score_region2))
})

ranked_scores <- reactive({
  region_scores() %>% arrange(desc(score))
})
```

3. **Leaflet Map Rendering**
    - Initially render the map with base tiles and polygons:[3]

      ```
      output$map <- renderLeaflet({
        leaflet() %>%
          addProviderTiles("CartoDB.Positron") %>%
          addPolygons(data = spatial_data,
                      fillColor = "grey",
                      layerId = ~region)
      })
      ```

4. **Dynamic Fill Updates**
    - When scores change, color polygons by score/rank using `leafletProxy`:[7] [2]

      ```
      observe({
        pal <- colorNumeric("YlOrRd", domain = ranked_scores()$score)
        leafletProxy("map") %>%
          clearShapes() %>%
          addPolygons(data = spatial_data,
                      fillColor = ~pal(ranked_scores()$score[match(spatial_data$region,
                      layerId = ~region)
      })
      ```

5. **UI Ranking Table**
    - Display current rankings below or beside the map with `DT::renderDataTable` or `tableOutput`:[6]

      ```
      output$ranking <- renderTable({
        ranked_scores()
      })
      ```

Additional References

- Dynamic polygons and user-driven color schemes are key to making the map interactive and responsive.[1]
- For large data and quick updates, minimize polygon redraw or optimize by updating styles (not entire map objects).[8] [7]
- Dashboard design can be further enhanced by user filters, map mouseover tooltips, and popups as needed.[5] [2]

This modular approach is proven in advanced Shiny dashboards and is extensible to more variables, regions, and complex aggregation logic.

⁂

Implement reactive aggregation and ranking in Shiny with dplyr

To implement reactive aggregation and ranking in Shiny using `dplyr`, define your calculations inside reactive expressions, then use `dplyr` verbs for aggregation, sorting, and ranking. This approach leverages Shiny's reactivity for interactive, real-time summaries and arrangements.[22] [23] [24]

Example: Aggregation and Ranking with dplyr in Shiny

Suppose your dataset is named `df` with columns `region`, `var1`, `var2`, etc., and the user adjusts `input$var1_score`, `input$var2_score`, etc.

```
library(shiny)
library(dplyr)

ui <- fluidPage(
  # Inputs for scoring
  numericInput("var1_score", "Var 1 score", value = 1),
  numericInput("var2_score", "Var 2 score", value = 1),
  tableOutput("summary")
)

server <- function(input, output, session) {
  # Example static data frame for illustration
  df <- data.frame(
    region = c("A", "B", "C"),
    var1 = c(5, 3, 1),
    var2 = c(2, 7, 4)
  )

  # Reactive aggregation with dplyr
  scored <- reactive({
    df %>%
      mutate(
        score = var1 * input$var1_score + var2 * input$var2_score
      ) %>%
      arrange(desc(score)) %>%
      mutate(rank = row_number())
  })

  output$summary <- renderTable({
    scored()
  })
}

shinyApp(ui, server)
```

Explanations and Tips

- The `scored()` reactive calculates weighted scores and assigns ranks whenever inputs change.[23] [22]
- `mutate` creates a new score, `arrange` sorts by score, and `row_number()` assigns a rank.[24]
- You can extend this by grouping and summarizing with `group_by` and `summarise` within the `reactive` if your use case requires aggregation across groups.

This pattern is at the core of interactive, ranked summary tables or map-based visualizations in Shiny dashboards.[22] [23] [24]

⁂

1. https://stackoverflow.com/questions/37336414/dynamic-colored-polygons
2. https://www.r-bloggers.com/2024/06/interactive-map-filter-in-shiny/
3. https://rstudio.github.io/leaflet/articles/shiny.html
4. https://shiny.posit.co/r/getstarted/build-an-app/reactive-flow/ui-inputs.html
5. https://forum.posit.co/t/shiny-app-with-reactive-leaflet-assigning-colors-r-crashing/149322
6. https://unc-libraries-data.github.io/R-Open-Labs/Extras/shiny/shiny.html
7. https://github.com/rstudio/leaflet/issues/496
8. https://forum.posit.co/t/make-shiny-leaflet-map-less-cumbersome-faster/210
9. https://stackoverflow.com/questions/74768942/shiny-app-aggregate-data-set-by-selection-filter-and-create-new-variables
10. https://nirzaree.wordpress.com/2020/10/11/building-a-webapp-for-data-collection-visualization-using-r-shiny/
11. https://shiny.posit.co/r/getstarted/shiny-basics/lesson5/
12. https://www.paulamoraga.com/book-geospatial/sec-shinyexample.html
13. https://mdl.library.utoronto.ca/technology/tutorials/introduction-r-shiny
14. https://paulcbauer.github.io/shiny_workshop/08-build-your-own.html
15. https://rviews.rstudio.com/2019/10/09/building-interactive-world-maps-in-shiny/
16. https://yakdata.com/ultimate-guide-r-shiny-app-dashboards-2021/index.html
17. https://www.appsilon.com/post/r-highcharts-maps
18. https://towardsdatascience.com/create-interactive-map-applications-in-r-and-r-shiny-for-exploring-geospatial-data-96b0f9692f0f/
19. https://programminghistorian.org/en/lessons/shiny-leaflet-newspaper-map-tutorial
20. https://www.youtube.com/watch?v=aUriimb5Pbw
21. interests.programming.shiny_r
22. https://mastering-shiny.org/action-tidy.html
23. https://debruine.github.io/shinyintro/reactives.html
24. https://dplyr.tidyverse.org/reference/ranking.html
25. https://www.youtube.com/watch?v=PVOD_c4HISE

26. https://stackoverflow.com/questions/57381640/rshiny-reactive-data-frame-manipulation-in-dplyr-using-selected-input
27. https://shiny.posit.co/r/getstarted/build-an-app/reactivity-essentials/reactive-elements.html
28. https://stackoverflow.com/questions/72476299/how-to-add-an-accumulating-rank-order-number-to-a-reactive-data-frame-or-list
29. https://shiny.posit.co/r/articles/build/pool-dplyr/
30. https://yakdata.com/ultimate-guide-r-shiny-app-dashboards-2021/index.html
31. https://forum.posit.co/t/issue-making-dplyr-reactive-variables-work-in-shiny-module/3694
32. https://discindo.org/posts/2025-02-06-aggregate-snippet/
33. https://rpubs.com/Brennan01/1149922
34. https://faculty.washington.edu/otoomet/info201-book/dplyr.html
35. https://dplyr.tidyverse.org/articles/programming.html
36. https://stackoverflow.com/questions/45816510/use-dplyr-conditional-filter-in-reactive-function-in-shiny
37. https://forum.posit.co/t/aggregate-reactive-data/44891
38. https://paulcbauer.github.io/shiny_workshop/08-build-your-own.html
39. https://www.youtube.com/watch?v=UHYpaEVjwRE
40. https://www.appsilon.com/post/r-shiny-reactivity
41. https://github.com/tidyverse/dplyr/issues/4385