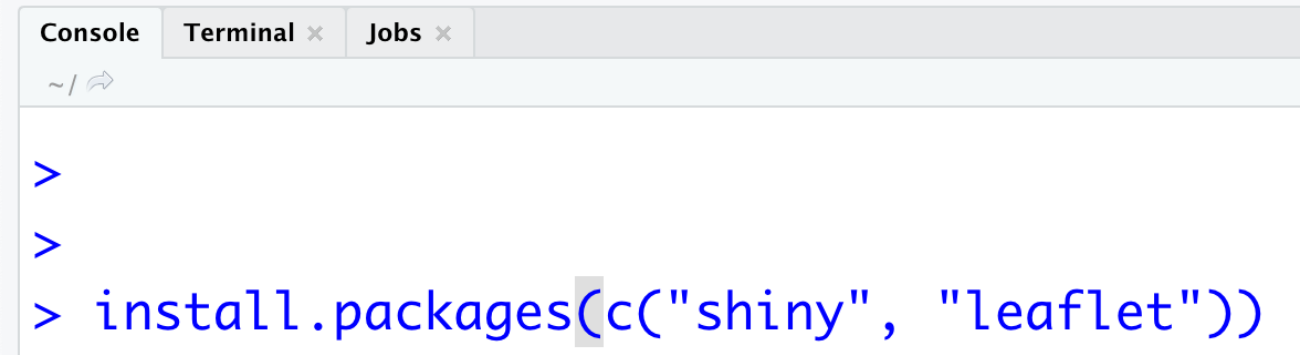
Lab: Creating an Interactive Map in R

Welcome to the 2nd RStudio lab in which we create an interactive map using two libraries. *Shiny*, which is an amazing library for prototyping interactive user interfaces in R and *Leaflet* which allows us to use OpenStreetMap data to create an interactive map. Shiny will use Leaflet as one of it's UI components (like any other UI component like button, slider or text field).

1. Install the following packages. Simply paste and execute the following command into the bottom left Console window

install.packages("shiny")

install.packages("leaflet")



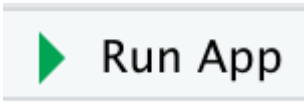
2. Paste the following code into a new R edit window in RStudio:

```
library(shiny)

library(leaflet)

r_colors <- rgb(t(col2rgb(colors()) / 255))

names(r_colors) <- colors()

 ui <- fluidPage(

  leafletOutput("mymap"),

  p(),

  actionButton("recalc", "New points"),
```

```
  p(),
  textOutput("coordinates")
)
server <- function(input, output, session) {
   points <- eventReactive(input$recalc, {
          points = cbind(rnorm(40) * 2 + 13, rnorm(40) + 48)
            output$coordinates <- renderText({
              points
            })
          return(points)
        }, ignoreNULL = FALSE)
  observeEvent(input$Map_shape_click, { # update the location selectInput on map clicks
    output$coordinates <- renderText({
      "You have selected this"
    })
  })
  output$mymap <- renderLeaflet({
    leaflet() %>%
      addProviderTiles(providers$Stamen.TonerLite,
                       options = providerTileOptions(noWrap = TRUE)
      ) %>%
```

```
    addMarkers(data = points())

  })

}

shinyApp(ui, server)
```

3. Click on the **Run App** button,



you'll see the following application appear:

4. If you click **New points**, a new set of random points will be generated and displayed. Please find the code below in the script in order to understand the creation of those points (please note that "rnorm" is a random number generator and "cbind" attaches two lists vertically to form a table structure):

```
points = cbind(rnorm(40) * 2 + 13, rnorm(40) + 48)
```

5. Now we want to change the code to display the list of points on the user interface. First of all we need to add this component on the user interface itself which means that we need to add element to the "ui" object. Please do so by adding the following two lines to the "ui":

```
p(),

textOutput("coordinates")
```

such that the complete code section looks like this:

```
ui <- fluidPage(

  leafletOutput("mymap"),

  p(),

  actionButton("recalc", "New points"),

  p(),

  textOutput("coordinates")

)
```

6. Then, it is time to also change the behavior of this application by changing the “server” object. Therefore, please add the following three lines to the so-called “eventhandler” responsible to execute code if the button is clicked:

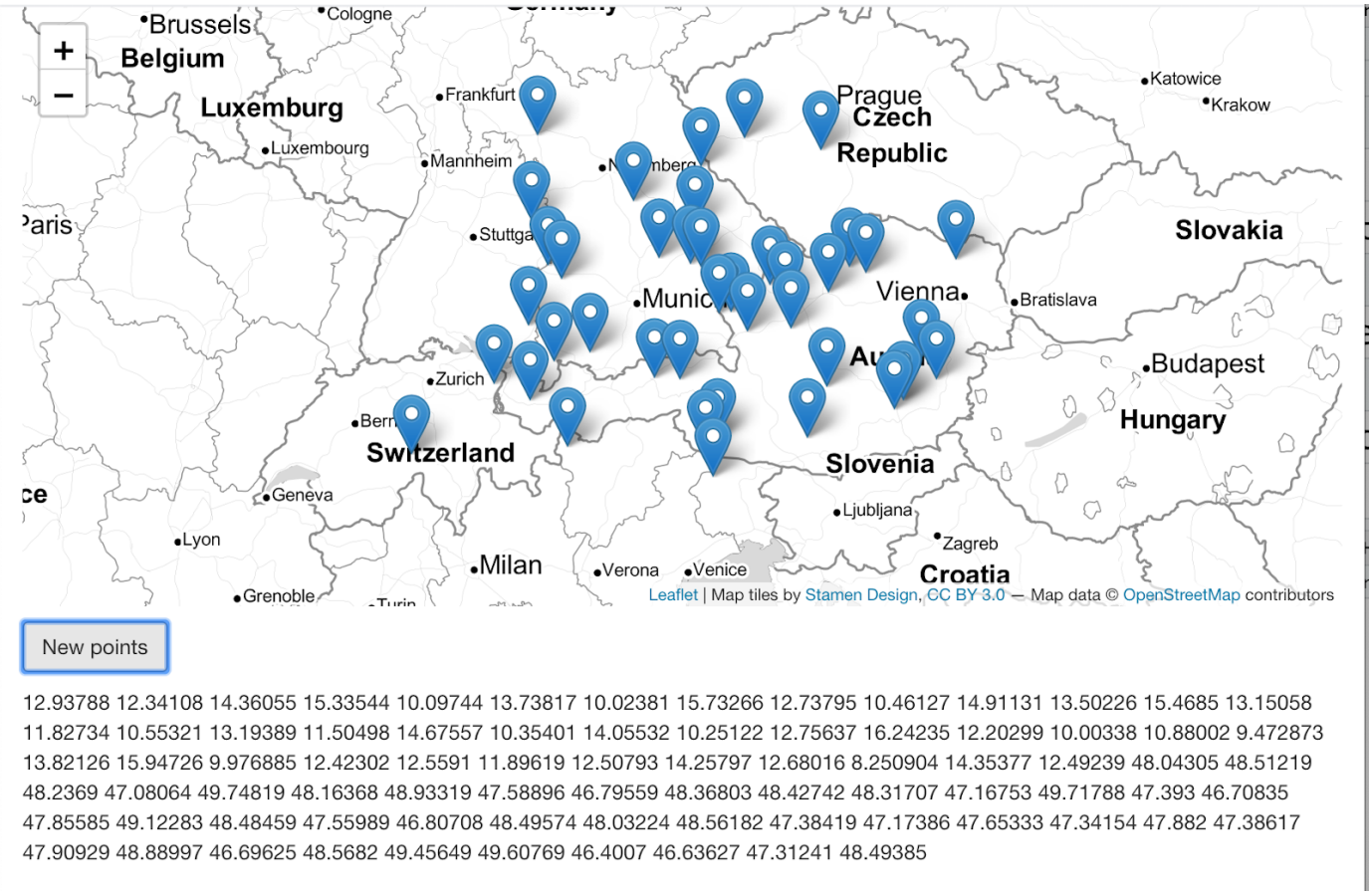
```
output$coordinates <- renderText({

      points

    })
```

such that the complete code section looks like this:

```
points <- eventReactive(input$recalc, {

          points = cbind(rnorm(40) * 2 + 13, rnorm(40) + 48)

            output$coordinates <- renderText({

              points

            })
```

```
    return(points)

  }, ignoreNULL = FALSE)
```

7. If you now re-run this application, you'll notice a list of coordinate pairs which updates every time you click the **New points** button



This concludes this lab, we hope that you had fun! If you want to know more about Shiny and Leaflet, please visit the following links:

https://shiny.rstudio.com/

https://rstudio.github.io/leaflet/