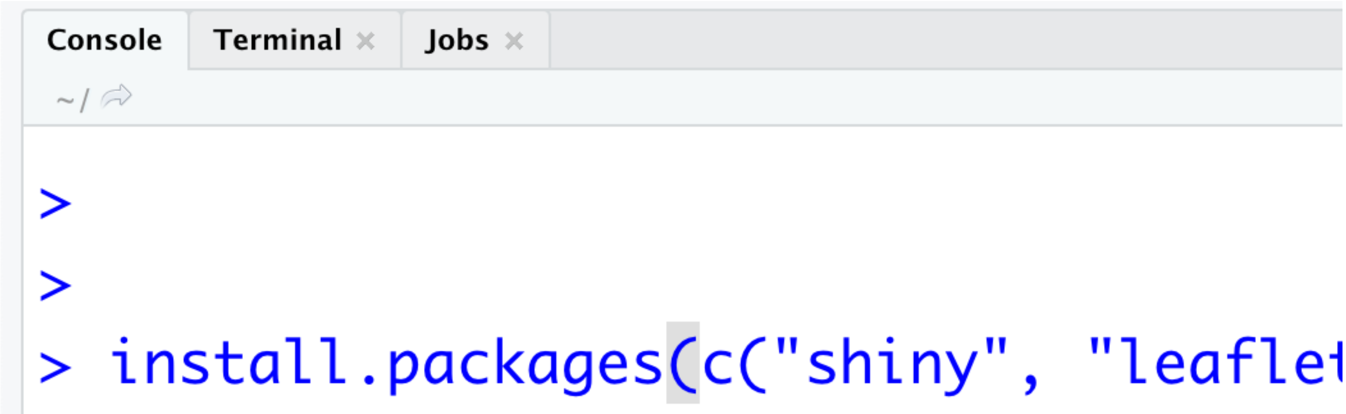
Lab: Creating an Interactive Map in R

coursera.org/learn/open-source-tools-for-data-science/supplement/6js90/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")
```



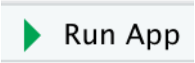
```
>
>
> install.packages(c("shiny", "leaflet
```

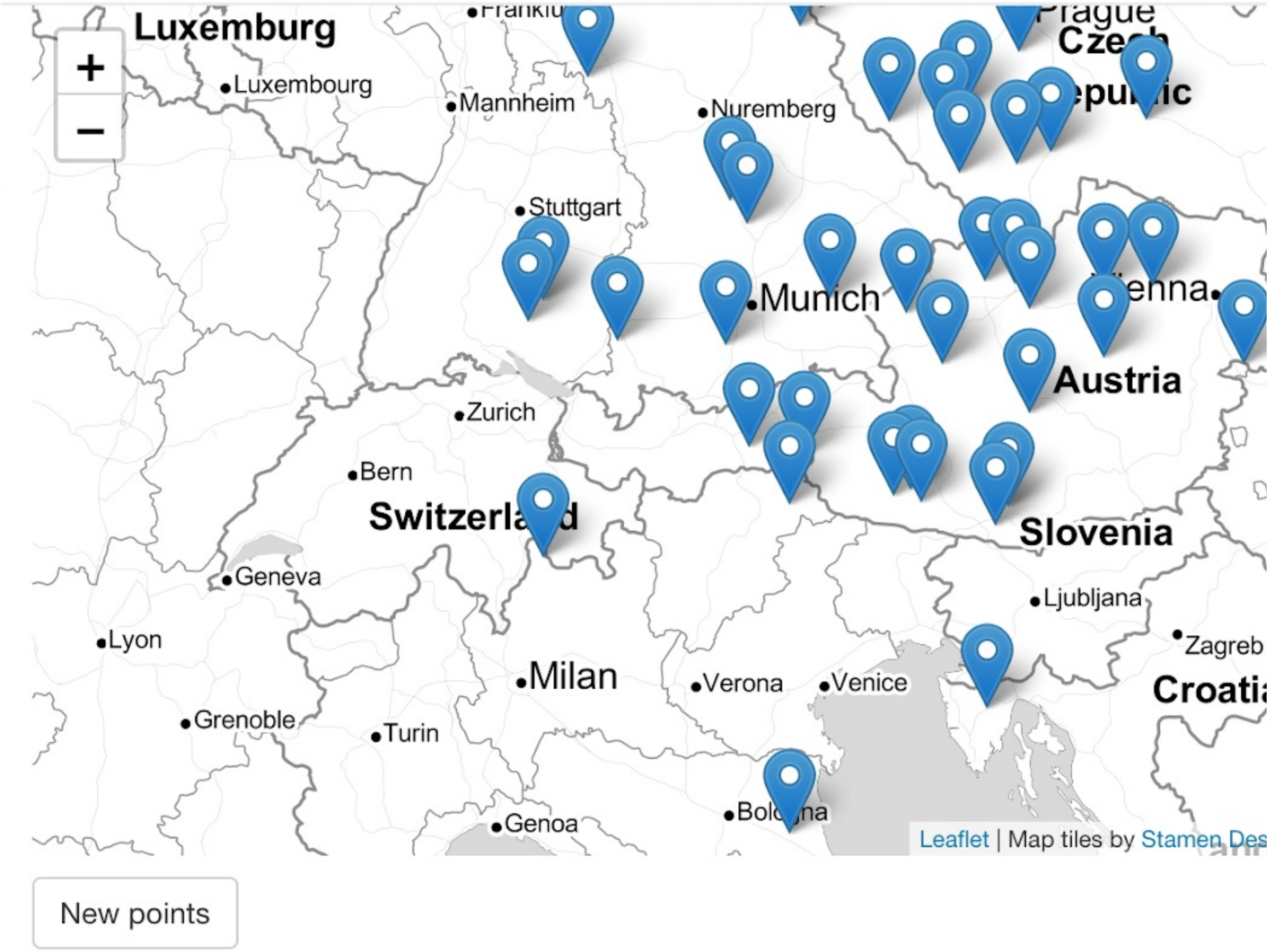
2. Paste the following code into a new R edit window in RStudio:

```
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")

)
```

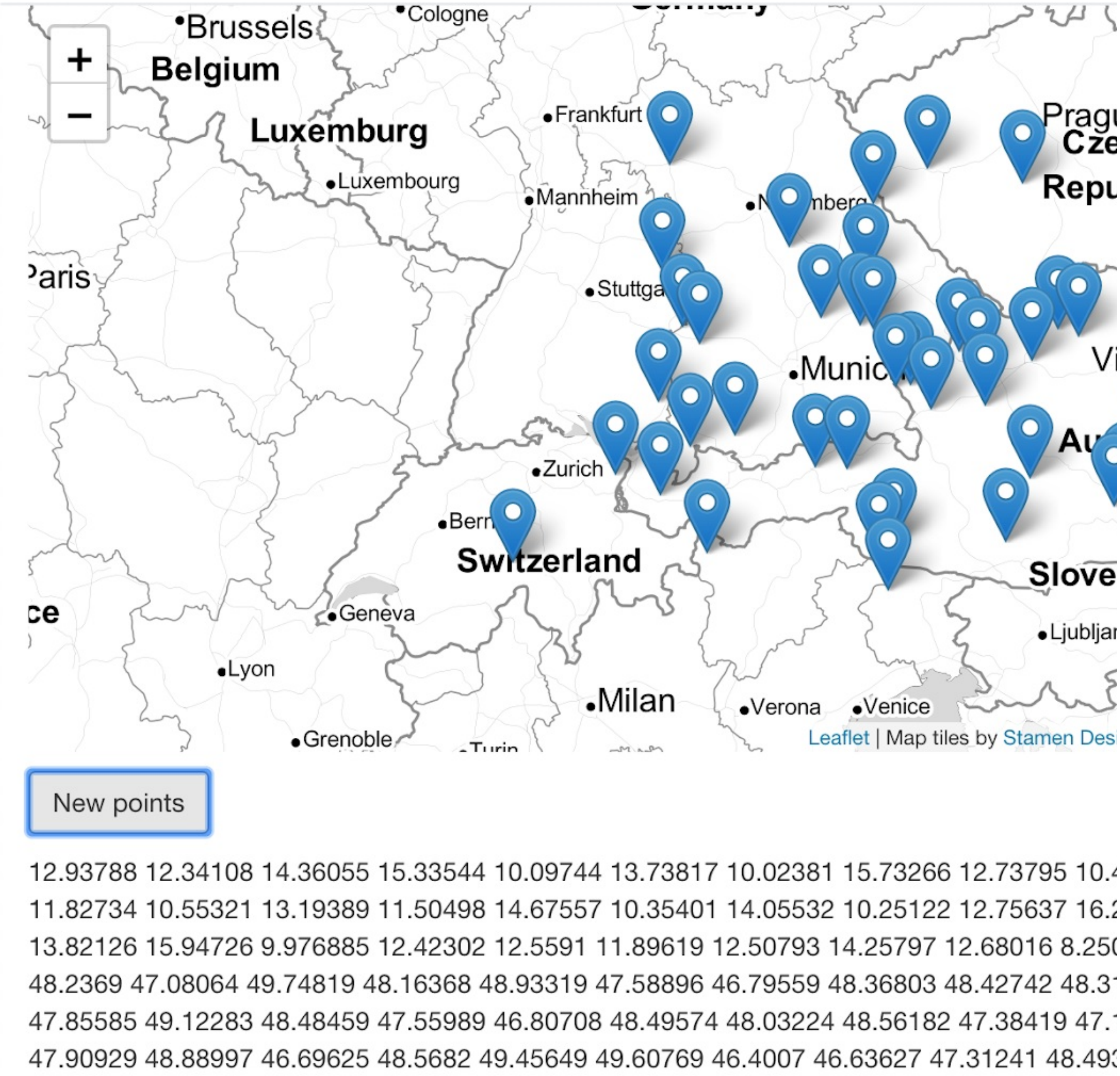
6. Then, it is time to also change the behavior of this application by changing the “server” object. Therefore, please add the following three lines to the so-called “eventhandler” responsible to execute code if the button is clicked:

```
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



```
12.93788 12.34108 14.36055 15.33544 10.09744 13.73817 10.02381 15.73266 12.73795 10.4
11.82734 10.55321 13.19389 11.50498 14.67557 10.35401 14.05532 10.25122 12.75637 16.2
13.82126 15.94726 9.976885 12.42302 12.5591 11.89619 12.50793 14.25797 12.68016 8.25
48.2369 47.08064 49.74819 48.16368 48.93319 47.58896 46.79559 48.36803 48.42742 48.3
47.85585 49.12283 48.48459 47.55989 46.80708 48.49574 48.03224 48.56182 47.38419 47.
47.90929 48.88997 46.69625 48.5682 49.45649 49.60769 46.4007 46.63627 47.31241 48.49
```

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/