INTRODUCCIÓN A LA GEOCOMPUTACIÓN CON R
EJERCICIOS NO. 9 (SHINY)

INDER TECUAPETLA-GÓMEZ

El objetivo de los siguientes ejercicios es desarrollar una aplicación para visualizar una serie de imágenes satelitales en un formato similar a un "storymap". Por supuesto, la idea es utilizar las capacidades de `shiny` y el código desarrollado durante la clase para lograr este objetivo.

1. De los 23 directorios de la carpeta `/data/NDVI_MTY_cell_165` selecciona uno. A partir de aquí, los 12 archivos del directorio que selecciones será el dataset a utilizar en este ejercicio.

2. Asegúrate de agregar un nuevo `menuItem` en el `sidebarMenu` del `dashboardSidebar`.

3. Al agregar un nuevo `tabItem` al `dashboardBody`, incluye dos `fluidRow`, el primero controlará un `mapviewOutput` y el segundo un `sliderInput`. **Hint1:**

```
fluidRow(
  column(12,
         mapviewOutput("plotArea", width = "100%", height = 700)
  )
),

fluidRow(
  column(5,
         sliderInput("n", "Month to show", min = 1,
                     max = 12, value = 1, step = 1)
  )
)
```

Hint2: Revisa la documentacion de `sliderInput`. Incluso puedes revisar ejemplos en esta página y en esta otra.

4. Ahora que has concluido con la definición de elementos en el `ui`, deberás definir la función `output$plotArea` en el `server` (si has usado otro nombre para definir el objeto resultante del `mapviewOutput`, deberas usar ese nombre en lugar de `plotArea`). Esta función debe tomar en cuenta el valor seleccionado al utilizar el `sliderInput` en la aplicación.

Posteriormente, tu función debe leer el archivo `.tif` que corresponda al valor seleccionado al utilizar el `sliderInput`; recuerda que el valor 1 en el `slideInput` corresponde al archivo con terminacion `NDVI_max.tif`, el valor 2 corresponde al archivo `NDVI_max_2.tif`, etc.

Finalmente, puedes usar la función `mapview()` para producir un mapa dinámico del archivo `.tif` seleccionado. No olvides que en `shiny` para poder visualizar un `mapview` el manejador apropiado se genera con la funcion `renderLeaflet`. **Hint:**

```
output$plotArea <- renderLeaflet({
  imagenToPlot <- input$n
  r <- raster(ListaArrchivos[imagenToPlot]) # esto es pseudo código
```

Noviembre 9, 2023

```
  mp <- mapview(r)
  mp@map
})
```