



**Universidad
Autónoma de
Coahuila**



Serie de Indicadores Económicos y Sociales en Aplicación Shiny

Centro de Investigaciones Socioeconómicas

Universidad Autónoma de Coahuila

Guía de aplicación y descripción de paquetería Shiny



Introducción

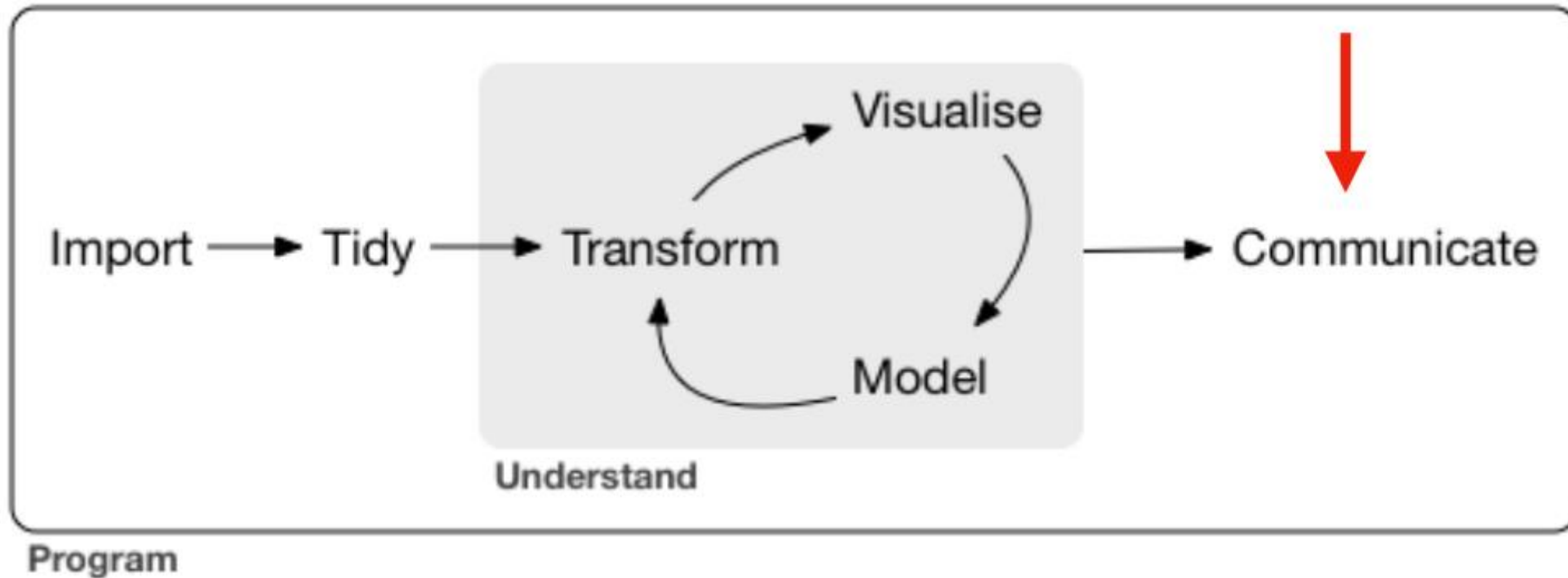
Shiny es un paquete de R que permite la creación de aplicaciones interactivas. Toma los datos de tu trabajo de investigación creado en R y lo expone mediante un navegador web de tal manera en que cualquier persona pueda revisarlo y visualizarlo. Permite la creación de aplicaciones web sin tener un conocimiento profundo de HTML, CSS y JavaScript (los lenguajes de la web). Esto lo hace ideal para personas que quieren hacer aplicaciones web de sus investigaciones pero no quieren aprender a programar.

- ❑ Sirve para crear dashboards (páginas web que llevan el registro de algo).
- ❑ Para dar a conocer resultados de investigaciones y bases de datos interesantes.
- ❑ Ayuda en la exploración rápida de bases de datos.

En palabras de **Hadley Wickham** creador de librerías como ggplot, dplyr, reshape:

“Shiny te da la habilidad de pasar algunos de tus superpoderes de R a cualquier persona que posea un navegador web.”

Aquí es donde entra Shiny



Ejemplos de lo que se puede hacer en Shiny



DRAFT

[Time series](#)

[Individual profiles](#)

[User guide](#)

Introduction

This application is designed to interactively visualize lake profile data collected by DWQ and cooperators and provide a platform for performing and reviewing aquatic life use based water quality assessments based on lake profile data.

Source code and additional information for this application are available via the [DWQ Lake Profile Dashboard GitHub repository](#) .

This application is available under an [MIT license](#) .

Data

All data used by this application are available through either [Utah DWQ AWQMS](#) or [EPA WQP](#) . The data used in this application are collected as water column profiles at lake sites throughout Utah. Lake profiles consist of measurements of dissolved oxygen, temperature, and pH at regular depth intervals through the water column. Intervals of 1 meter are typical for Utah DWQ sampling, but may be adjusted depending on the depth at any individual site. Data have been pre-processed to ensure consistency in parameter names and units and to calculate frequencies of exceedance of water quality criteria

Ejemplos de lo que se puede hacer en Shiny

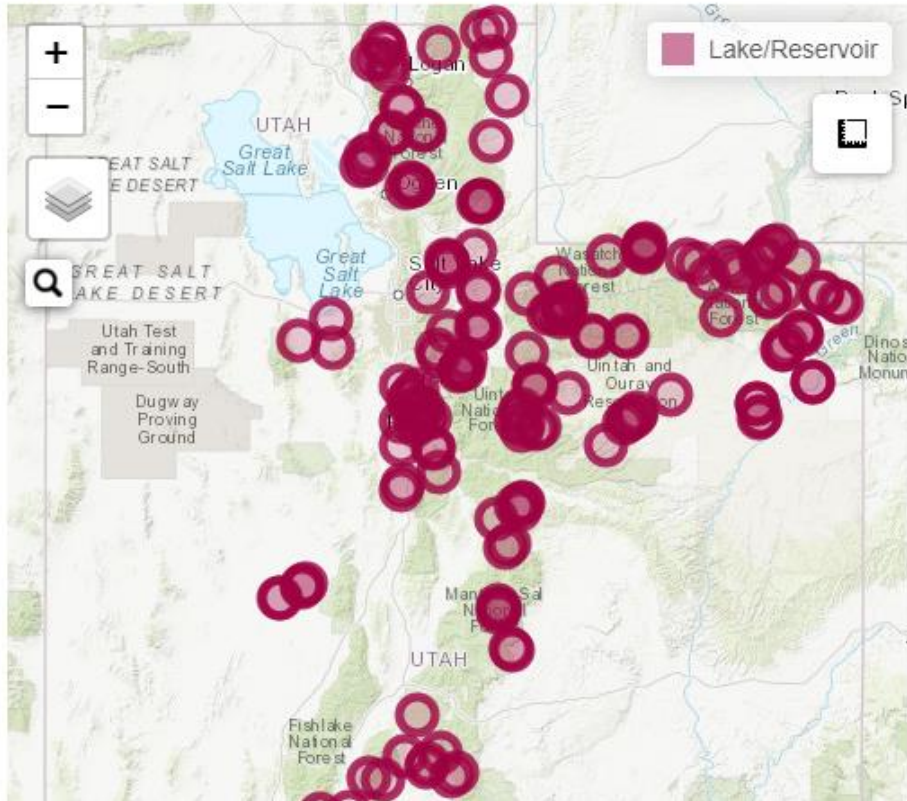


UTAH DEPARTMENT of
ENVIRONMENTAL QUALITY
WATER
QUALITY

DRAFT

Map Table

Click a site



Time series Individual profiles User guide

Plot type:

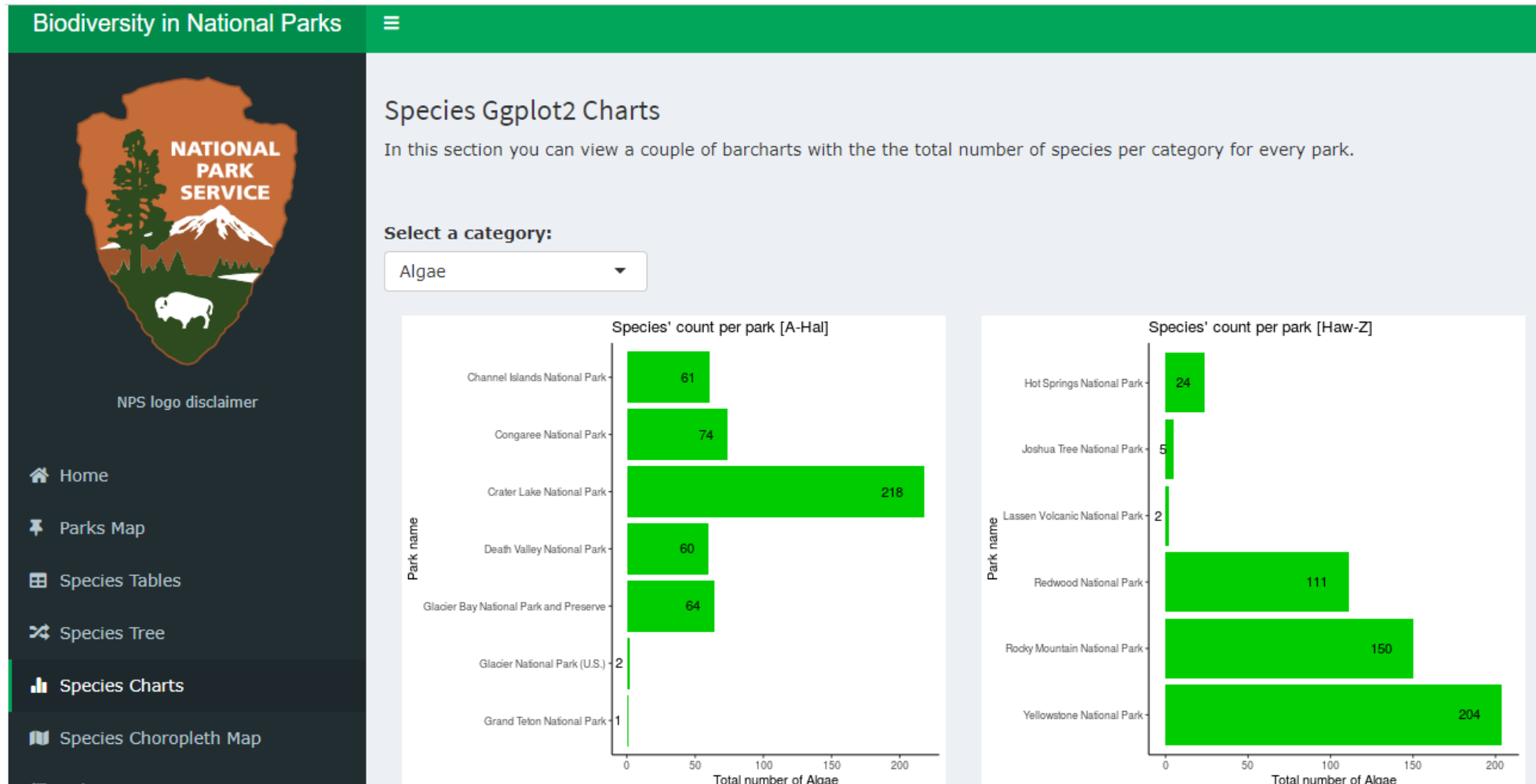
Heatmap Habitable width Water column exceedances

Heatmap parameter:

Dissolved oxygen

Show all profile dates

Ejemplos de lo que se puede hacer en Shiny



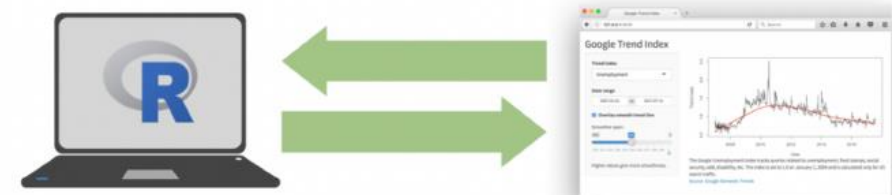
¿Cómo funciona el paquete Shiny?

Todas las aplicaciones Shiny se componen de una página que el usuario visita, y por detrás de esta página existe una computadora que está corriendo códigos de R.

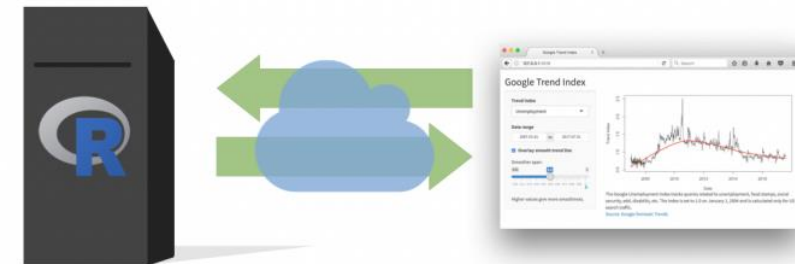


La aplicación puede correr de dos formas: la forma local (en mi computadora) y desplegada en internet.

Cuando corremos la aplicación local, la computadora que corre R es nuestra computadora.



Cuando la app está en internet, la computadora que corre el R es un servidor web.



Estructura de una aplicación Shiny

¿Qué elementos debe de contener?

```
library(shiny)
```

```
ui <- fluidPage()
```

Interfaz de usuario, controla el diseño y la apariencia de la aplicación.

```
server <- function(input, output) {}
```

La función del servidor contiene las instrucciones necesarias para crear la aplicación.

```
shinyApp(ui = ui, server = server)
```

Diseño de la interfaz (ui.R)

El diseño de la interfaz de usuario está relacionado con:

- La estructura (barras laterales, paneles, pestañas, etc.).
- El aspecto (colores, tipos de fuentes, etc.)

La estructura de la interfaz de usuario puede ser:

- fluidPage
- Shinydashboad

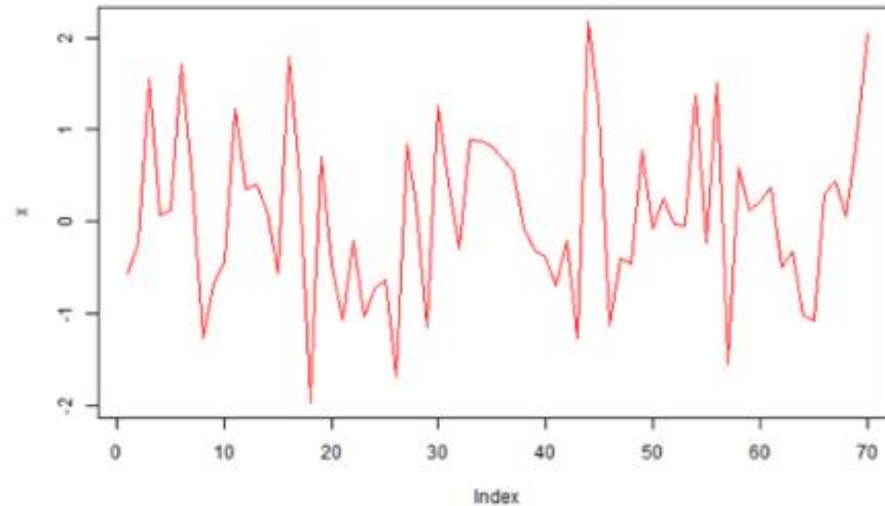
En cualquier caso:

- titlePanel: título de la aplicación
- sidebarLayout: para los contenidos

Diseño del servidor (server.R)

- Se define el server en donde estará toda la lógica de nuestra aplicación.
- Principalmente serán funciones que dependan de inputs.

Ejemplo de código para aplicación

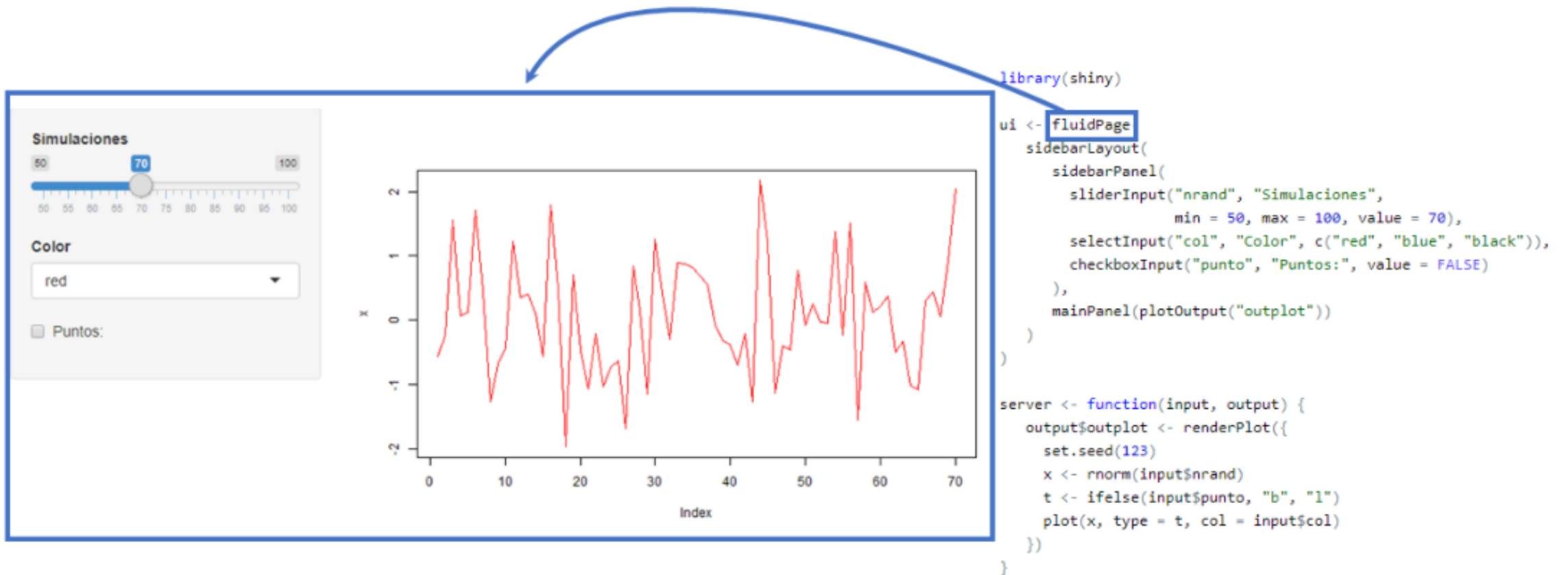


```
library(shiny)

ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      sliderInput("nrand", "Simulaciones",
                 min = 50, max = 100, value = 70),
      selectInput("col", "Color", c("red", "blue", "black")),
      checkboxInput("punto", "Puntos:", value = FALSE)
    ),
    mainPanel(plotOutput("outplot"))
  )
)

server <- function(input, output) {
  output$outplot <- renderPlot({
    set.seed(123)
    x <- rnorm(input$nrand)
    t <- ifelse(input$punto, "b", "l")
    plot(x, type = t, col = input$col)
  })
}
```

Ejemplo de código para aplicación

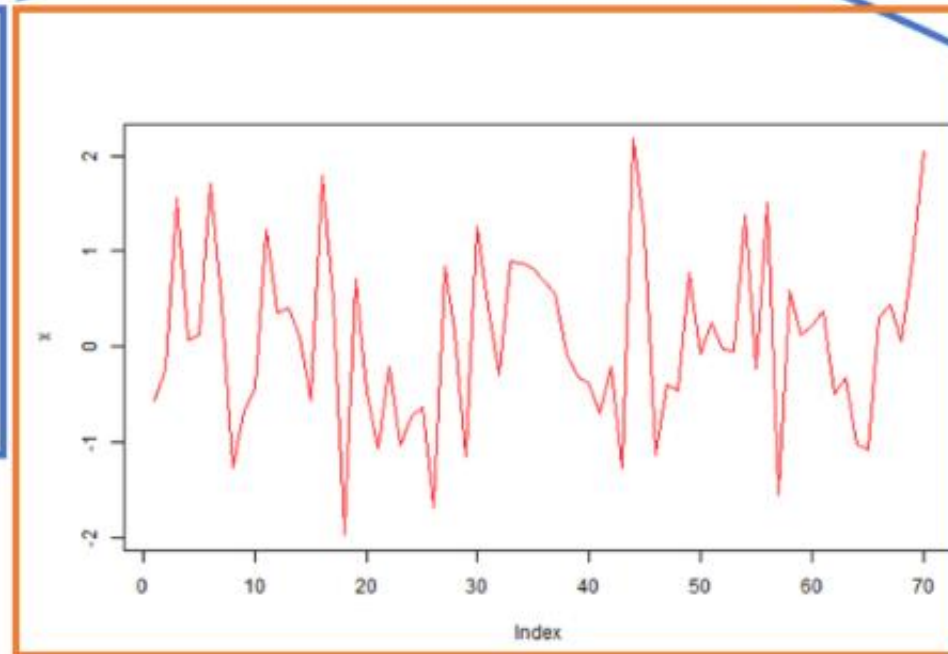


```
library(shiny)
ui <- fluidPage
  sidebarLayout(
    sidebarPanel(
      sliderInput("nrand", "Simulaciones",
                 min = 50, max = 100, value = 70),
      selectInput("col", "Color", c("red", "blue", "black")),
      checkboxInput("punto", "Puntos:", value = FALSE)
    ),
    mainPanel(plotOutput("outplot"))
  )
)

server <- function(input, output) {
  output$outplot <- renderPlot({
    set.seed(123)
    x <- rnorm(input$nrand)
    t <- ifelse(input$punto, "b", "l")
    plot(x, type = t, col = input$col)
  })
}
```

- fluidPage es una de las partes mas esenciales, es donde vamos a elaborar nuestra aplicación.

Ejemplo de código para aplicación



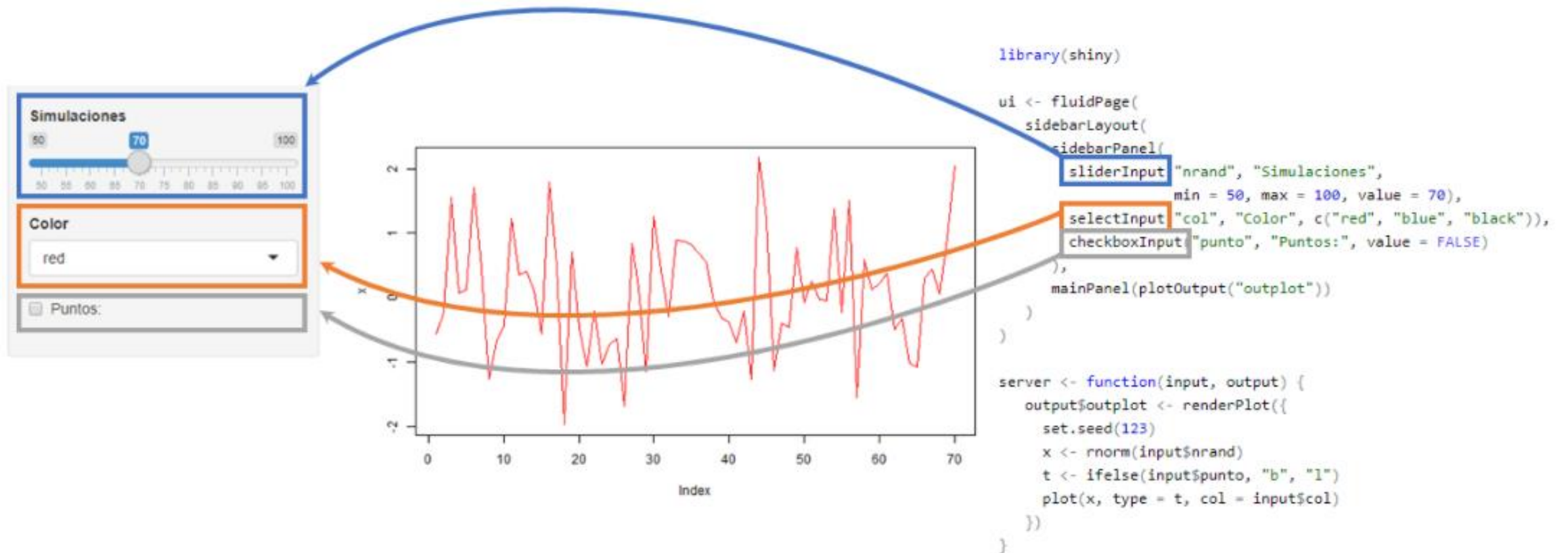
```
library(shiny)

ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      sliderInput("nrand", "Simulaciones",
                 min = 50, max = 100, value = 70),
      selectInput("col", "Color", c("red", "blue", "black")),
      checkboxInput("punto", "Puntos:", value = FALSE)
    ),
    mainPanel(plotOutput("outplot"))
  )
)

server <- function(input, output) {
  output$outplot <- renderPlot({
    set.seed(123)
    x <- rnorm(input$nrand)
    t <- ifelse(input$punto, "b", "l")
    plot(x, type = t, col = input$col)
  })
}
```

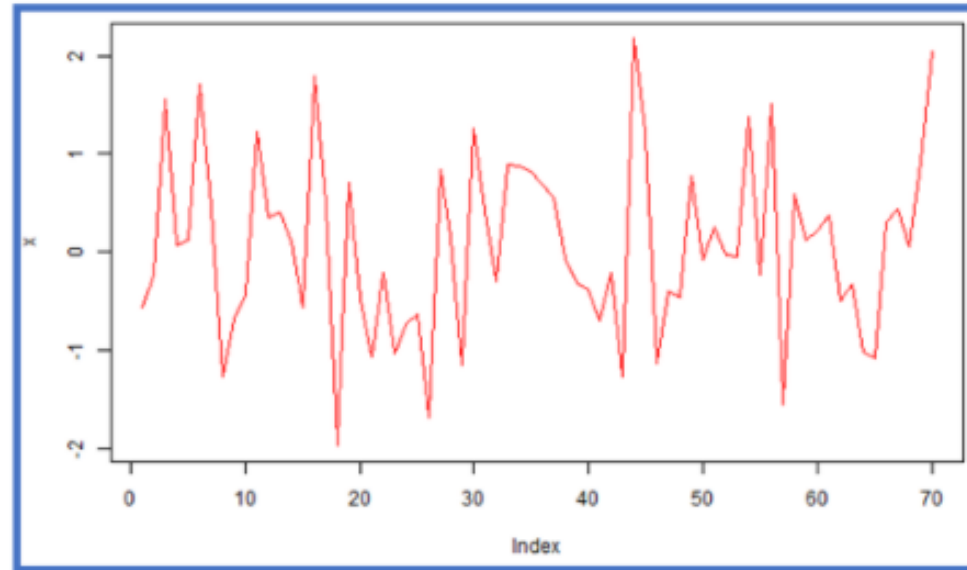
- sidebarPanel es el panel con el que va a interactuar el usuario.
- mainPanel es lo que va a ver el usuario cuando interactúa con los datos

Ejemplo de código para aplicación



- sliderInput sirve para mostrar un botón para desplegar un rango de números.
- selectInput sirve para seleccionar algo (en este caso son colores)

Ejemplo de código para aplicación

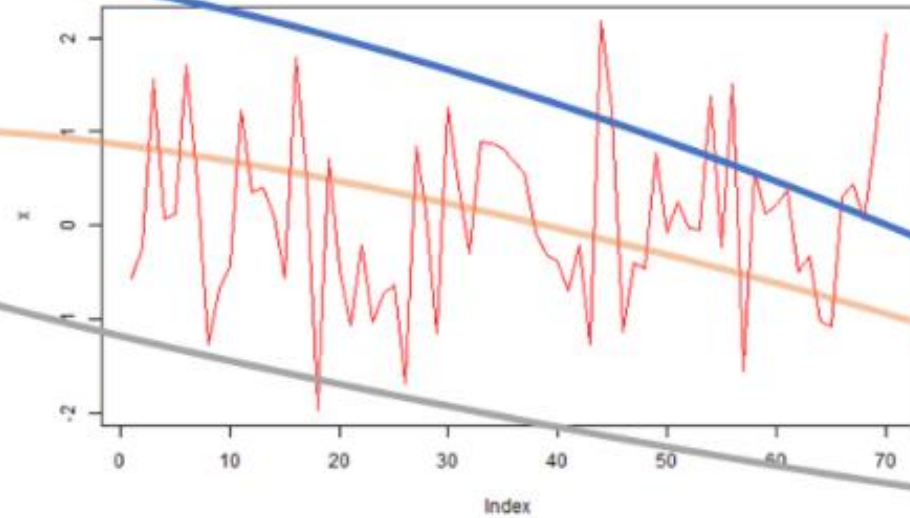


```
library(shiny)

ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      sliderInput("nrand", "Simulaciones",
                 min = 50, max = 100, value = 70),
      selectInput("col", "Color", c("red", "blue", "black")),
      checkboxInput("punto", "Puntos:", value = FALSE)
    ),
    mainPanel(plotOutput("outplot"))
  )
)

server <- function(input, output) {
  output$outplot <- renderPlot({
    set.seed(123)
    x <- rnorm(input$nrand)
    t <- ifelse(input$punto, "b", "l")
    plot(x, type = t, col = input$col)
  })
}
```

Ejemplo de código para aplicación

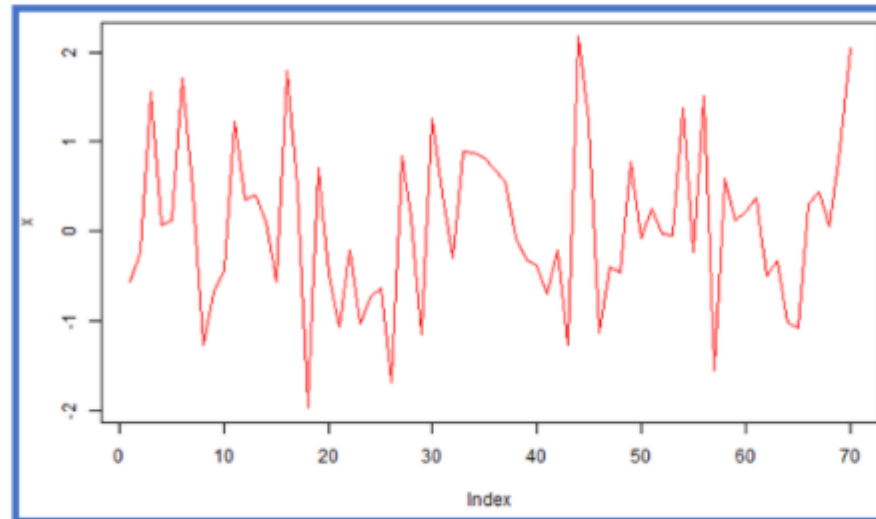


```
library(shiny)

ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      sliderInput("nrand", "Simulaciones",
                 min = 50, max = 100, value = 70),
      selectInput("col", "Color", c("red", "blue", "black")),
      checkboxInput("punto", "Puntos:", value = FALSE)
    ),
    mainPanel(plotOutput("outplot"))
  )
)

server <- function(input, output) {
  output$outplot <- renderPlot({
    set.seed(123)
    x <- rnorm(input$nrand)
    t <- ifelse(input$punto, "b", "l")
    plot(x, type = t, col = input$col)
  })
}
```

Ejemplo de código para aplicación



```
library(shiny)

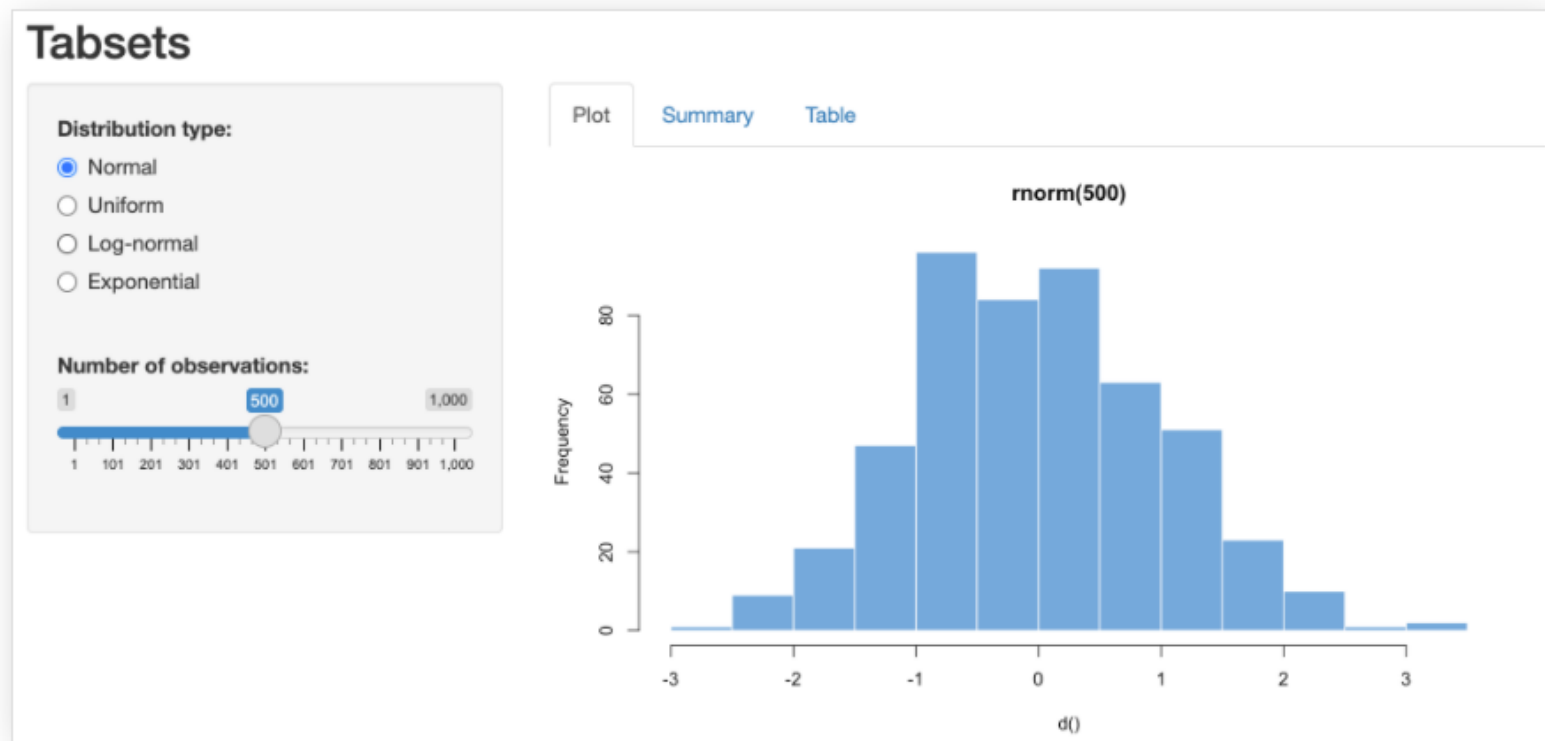
ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      sliderInput("nrand", "Simulaciones",
                 min = 50, max = 100, value = 70),
      selectInput("col", "Color", c("red", "blue", "black")),
      checkboxInput("punto", "Puntos:", value = FALSE)
    ),
    mainPanel(plotOutput("outplot"))
  )
)

server <- function(input, output) {
  output$outplot <- renderPlot({
    set.seed(123)
    x <- rnorm(input$nrand)
    t <- ifelse(input$punto, "b", "l")
    plot(x, type = t, col = input$col)
  })
}
```

fluidPage, sidebarPanel, mainPanel definen el diseño de nuestra app.

Existen muchas más formas de organizar una app:

A menudo, las aplicaciones necesitan subdividir su interfaz de usuario en secciones. Esto se puede lograr usando la función `tabsetPanel()`. Por ejemplo:



```
ui <- fluidPage(  
  titlePanel("Tabsets"),  
  sidebarLayout(  
    sidebarPanel(  
      mainPanel(  
        tabsetPanel(  
          tabPanel("Plot", plotOutput("plot")),  
          tabPanel("Summary", verbatimTextOutput("summary")),  
          tabPanel("Table", tableOutput("table"))  
        ))  
      )  
    )  
  )  
)
```

Más ejemplos y detalles <http://shiny.rstudio.com/articles/layout-guide.html>

Las funciones render y output

```
ui <- fluidPage(  
  sidebarLayout(  
    sidebarPanel(  
      sliderInput("nrand", "Simulaciones",  
                  min = 50, max = 100, value = 70),  
      selectInput("col", "Color", c("red", "blue", "black")),  
      checkboxInput("punto", "Puntos:", value = FALSE)  
    ),  
    mainPanel(plotOutput("outplot"))  
  )  
)  
  
server <- function(input, output) {  
  output$outplot <- renderPlot({  
    set.seed(123)  
    x <- rnorm(input$nrand)  
    t <- ifelse(input$punto, "b", "l")  
    plot(x, type = t, col = input$col)  
  })  
}
```

1. Las funciones `Output()` y `render()` trabajan juntas para agregar salidas de R a la interfaz de usuario.
2. En este caso `renderPlot` esta asociado con `plotOutput`
3. Hay muchas parejas como `renderText/textOutput` ó `renderTable/tableOutput`, entre otras

Indicadores Económicos y Sociales en aplicación Shiny



Visualización

Documentación

Sección de Visualización

Documentación y bienvenida.

Bienvenido: Esta es una aplicación para la visualización de indicadores de desarrollo económico y social. Esta aplicación se realizó con el fin de ilustrar el tipo de tableros que se pueden realizar con este tipo de datos. Versión 0.0.1 lista para mejoras y adiciones extraordinarias.

CISE

Indicadores Económicos y Sociales en aplicación Shiny

Visualización



Documentación

Sección de Visualización

Plataforma Indicadores Internos

La presente aplicación muestra indicadores de desarrollo a nivel estatal en tres tipos de visualización distintos:

- Visualización en gráfico de barras, para facilitar la comparativa entre entidades
- Visualización de series de tiempo, para la visualización de la evolución del desempeño del indicador para un estado en particular
- Mapa cloroplético, para visualizar si existen patrones regionales en el desempeño de los indicadores seleccionados

Para acceder a la visualización deseada, seleccione la información correspondiente con los controles de la izquierda

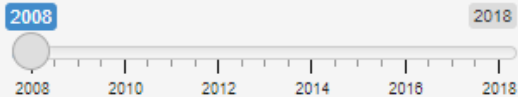
Opciones de Visualización

- Grafica de barras
- Serie de Tiempo
- Mapa

Seleccione un indicador

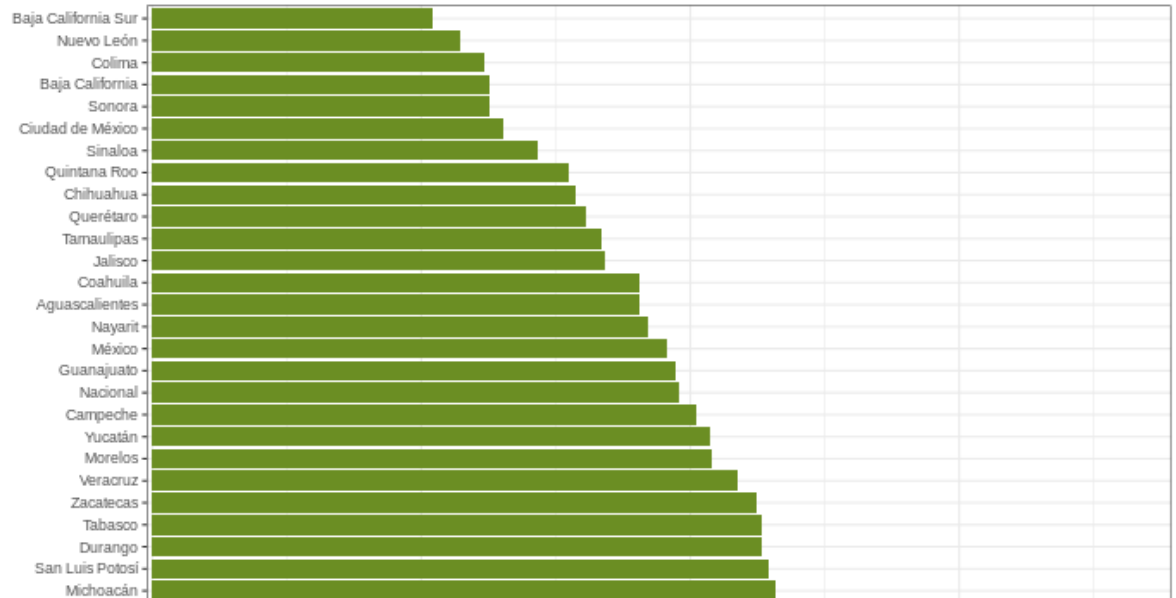
1- Porcentaje de población con ingreso inferior a la línea de bienestar

Seleccione Año



Porcentaje de población con ingreso inferior a la línea de bienestar

Año: 2008



Indicadores Económicos y Sociales en aplicación Shiny

Visualización



Documentación

Sección de Visualización

Plataforma Indicadores Internos

La presente aplicación muestra indicadores de desarrollo a nivel estatal en tres tipos de visualización distintos:

- Visualización en gráfico de barras, para facilitar la comparativa entre entidades
- Visualización de series de tiempo, para la visualización de la evolución del desempeño del indicador para un estado en particular
- Mapa cloroplético, para visualizar si existen patrones regionales en el desempeño de los indicadores seleccionados

Para acceder a la visualización deseada, seleccione la información correspondiente con los controles de la izquierda

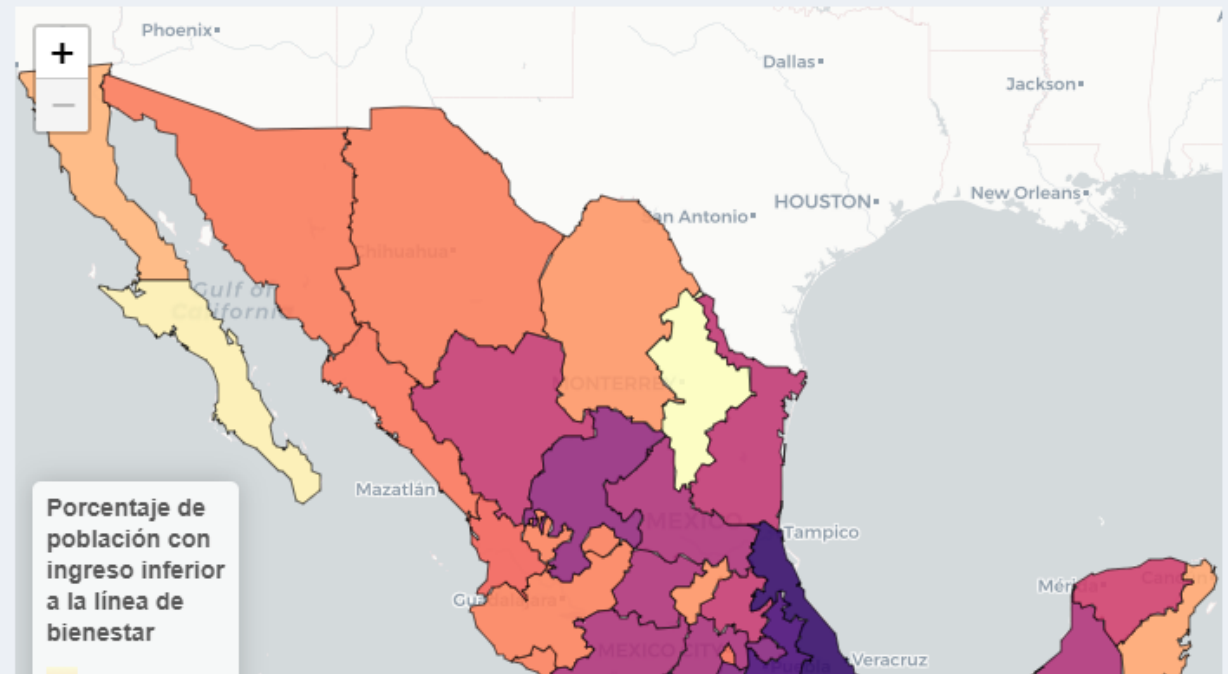
Opciones de Visualización

- Grafica de barras
- Serie de Tiempo
- Mapa

Seleccione un indicador

1- Porcentaje de población con ingreso inferior a la línea de bienestar

Seleccione Año



Indicadores Económicos y Sociales en aplicación Shiny

Visualización



Documentación

Sección de Visualización

Plataforma Indicadores Internos

La presente aplicación muestra indicadores de desarrollo a nivel estatal en tres tipos de visualización distintos:

- Visualización en gráfico de barras, para facilitar la comparativa entre entidades
- Visualización de series de tiempo, para la visualización de la evolución del desempeño del indicador para un estado en particular
- Mapa cloroplético, para visualizar si existen patrones regionales en el desempeño de los indicadores seleccionados

Para acceder a la visualización deseada, seleccione la información correspondiente con los controles de la izquierda

Opciones de Visualización

- Grafica de barras
- Serie de Tiempo
- Mapa

Seleccione un indicador

4- Porcentaje de población vulnerable por ingresos

Seleccione uno (o más) estado(s)

Aguascalientes Ciudad de México

Generame la grafica

Porcentaje de población vulnerable por ingresos

