

INTRODUCTION

WORKSHOP FORMAT



Interactive geospatial visualization dashboards with R and Shiny.

Objectives:

The objective of this workshop is to familiarize with the development of web apps using R and Shiny. Students will learn how to clean and process geospatial data with R to generate maps and other visualizations, and how to translate those into a interactive web dashboard that can be distributed and shared. By completing this workshop you will be able to:

- Identify the basic components for a shiny app
- Generate static and interactive maps with R
- Strategies to make your app more efficient
- Upload your application to a server so it can be shared and distributed.

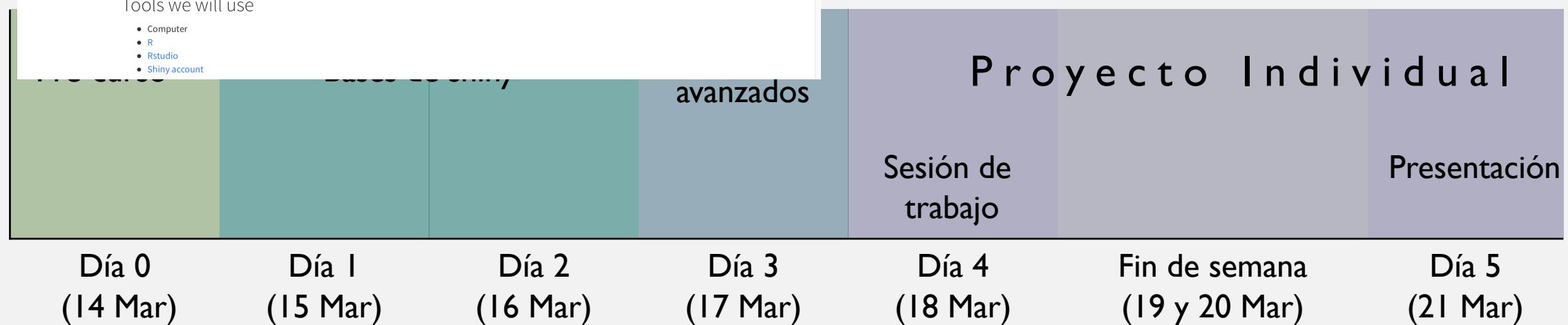
Previous knowledge with geospatial data and R helpful is desired, but not mandatory. Pre workshop materials covering the basics of R and geospatial data will be provided a few days before the workshop.

Who is this course for?

Participants must have a basic understanding with R and spatial data. Pre-workshop materials and office hours will be offered to help with any questions regarding the material and installation of software. It is expected that the participants will already have installed the software since day 01 of the workshop.

Tools we will use

- Computer
- R
- Rstudio
- Shiny account



FORMATO DEL WORKSHOP: LABS

CADMS Home Pre Workshop ▾ Contact ▾

1 Introducción

2 Interfaz (UI)

3 Servidor (server)

4 Corriendo la aplicación

Lab Ib

1 Introducción

En este laboratorio vamos a modificar la plantilla inicial de shiny para poder cargar nuestros datos y hacer figuras reactivas.

Iniciaremos cargando las librerías que vamos a utilizar

```
# Cargar las librerías
library(shiny)
library(dplyr) # Para manipulacion de datos
library(ggplot2) # Para las figuras
```

Ahora cargaremos los datos que usaremos. Asegurate de poner la dirección del archivo correctamente. En el caso de mi proyecto, yo tengo mis datos en una carpeta que se llama *Datos* y quiero cargar el archivo *Vacunacion.csv*.

```
# Cargar los datos
df <- read.csv('../Datos/Processed/Vacunacion.csv')
```

La librería `shiny` nos proporciona varias funciones para crear nuestra aplicación. La principal es la función `shinyApp`, la cual nos pide dos argumentos `ui` y `server`. En este laboratorio vamos a crear dos objetos que contengan todo lo necesario para ser usados como argumentos en esta función.

2 Interfaz (UI)

La interfaz (UI), es donde definimos el tipo de input y outputs que tendrá nuestra aplicación, así como cualquier otra información relevante que nos ayude a explicar la aplicación. La información de nuestra aplicación estará definida en diferentes niveles anidados con la siguiente jerarquía:

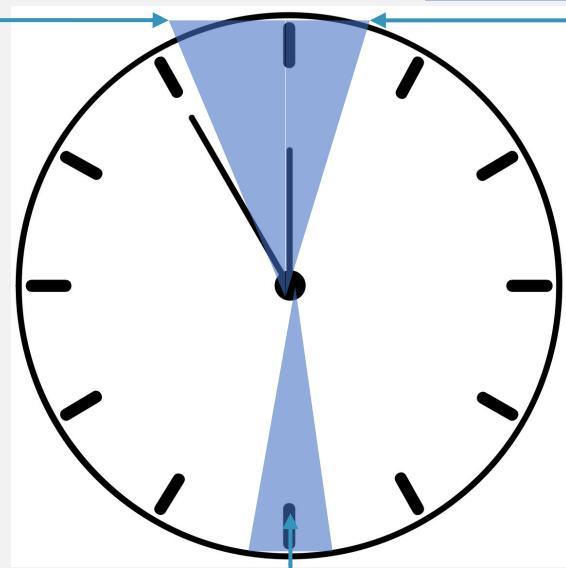
- Página (Page)
 - Panel
 - Inputs
 - Outputs

Hay diferentes especificaciones para el formato de la página (en este ejercicio usamos `fluidPage()` por ejemplo), y los paneles se pueden organizar en diferentes layouts, para esta aplicación estamos usando el layout `sidebarLayout()`. Para ver qué otras opciones de layout tenemos, podemos consultar las funciones llamando la librería escribiendo `shiny::` (sin cargarla) y usando `tab` para que RStudio nos muestre la lista de funciones disponibles. Al ir escribiendo la palabra `layout`, la opción de autocompletar de RStudio irá reduciendo las opciones a las funciones que contengan la palabra `layout` en su nombre, por ejemplo:



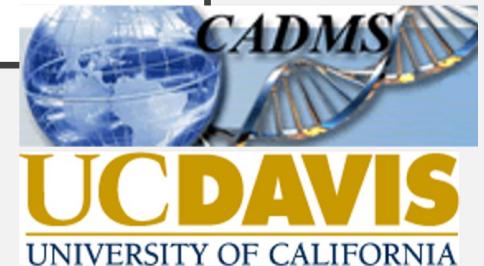
Conclusion

Introducción



Avances

WHO AM I?



Nombre: Jose Pablo Gomez



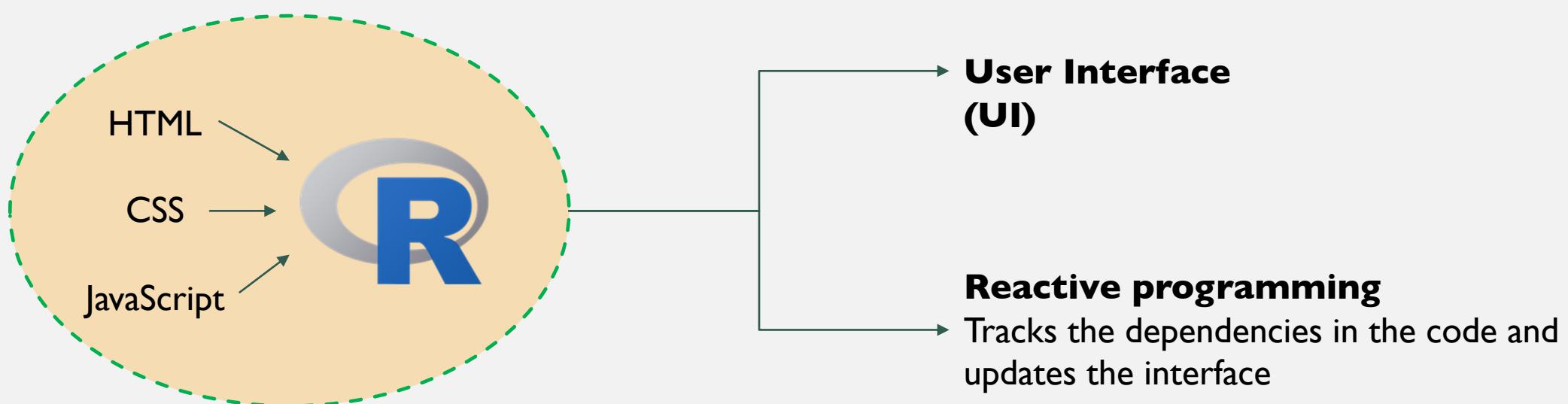
Estudios académicos:

- Médico Veterinario Zootecnista. (UANL)
- Master in Preventive Veterinary Medicine (UC Davis)
- Doctorado en Epidemiología (UC Davis)

<http://spablo-temporal.network/>

WHAT IS A SHINY APP?

Shiny is an R library that allows to interface with HTML and JavaScript to create interactive applications that can be run locally or on a web server



SOME EXAMPLES



Tracking Home

Data Visualizations ▾

Global Map

U.S. Map

Data in Motion

Tracking FAQ

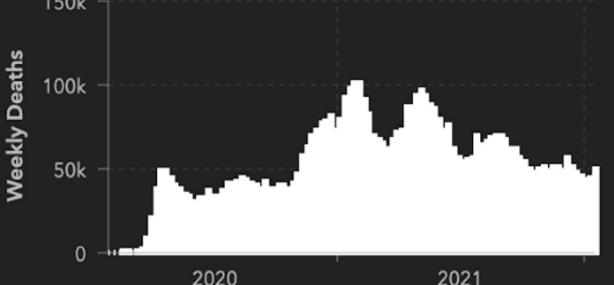
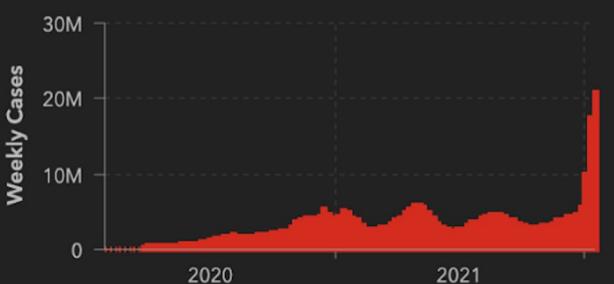


COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)

<https://coronavirus.jhu.edu/map.html>Last Updated at (M/D/YYYY)
21/1/2022 12:21Total Cases
344.002.803Total Deaths
5.578.967Total Vaccine Doses Administered
9.759.350.036Cases | Deaths by
Country/Region/Sovereignty**US**
28-Day: **17.479.329** |
42.845
Totals: **69.591.919** | **861.392****France**
28-Day: **6.721.223** | **5.825**
Totals: **15.715.702** | **129.105****Italy**
28-Day: **3.901.202** | **6.345**
Totals: **9.603.856** | **142.963****United Kingdom**
28-Day: **3.881.284** | **5.523**
Totals: **15.814.135** | **153.998****India**
28-Day: **3.793.401** | **9.263**
Totals: **38.566.027** | **488.396****Spain**
28-Day: **3.116.356** | **2.580**
Totals: **8.834.363** | **91.599**28-Day Cases
64.312.80828-Day Deaths
186.10928-Day Vaccine Doses Administered
809.080.419

Esri, FAO, NOAA

Powered by Esri



Admin0

Admin0

28-Day

28-Day

Weekly



DOCS HELP LOGIN <

Genomic epidemiology of novel coronavirus - Global subsampling



Built with [nextstrain/ncov](#). Maintained by the [Nextstrain team](#). Enabled by data from [GISAID](#).

Showing 3347 of 3347 genomes sampled between Dec 2019 and Jan 2022.

<https://nextstrain.org/ncov/gisaid/global>

Dataset

ncov
gisaid
global

Date Range

2019-11-23 2022-01-19

▶ PLAY

RESET

Color By

Clade

Filter Data

Type filter query here...

Tree Options

Layout

RECTANGULAR

RADIAL

UNROOTED

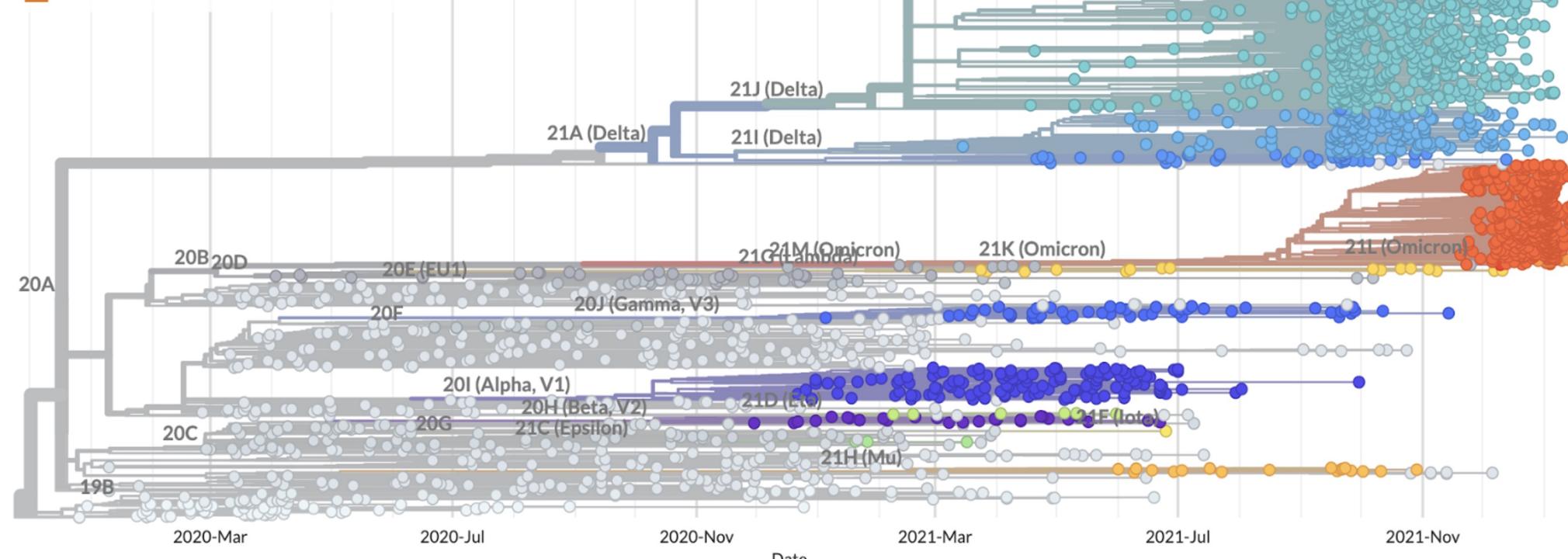
CLOCK

SCATTER

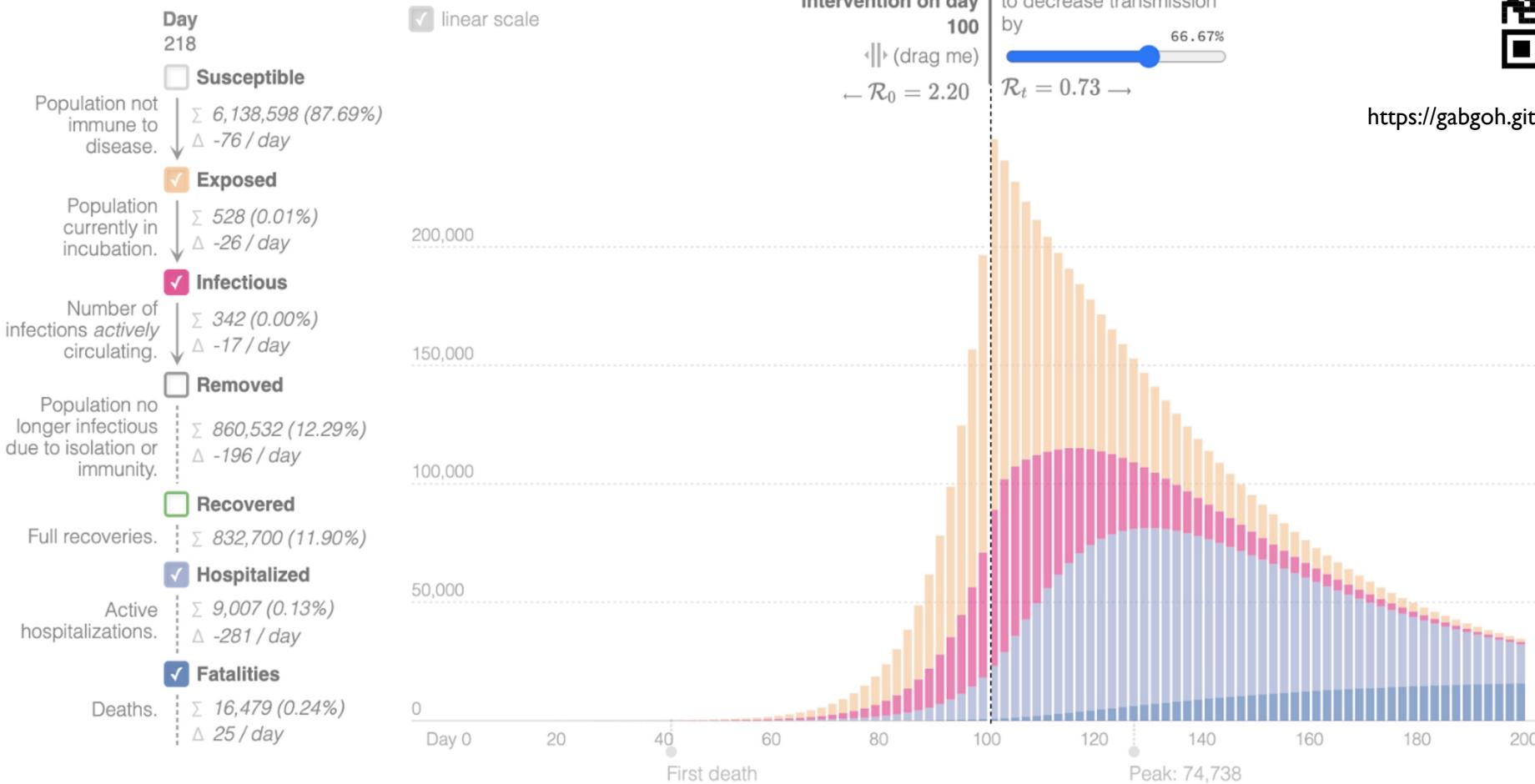
Phylogeny

Clade ▲

- 20H (Beta, V2)
- 20I (Alpha, V1)
- 20J (Gamma, V3)
- 21A (Delta)
- 21I (Delta)
- 21J (Delta)
- 21C (Epsilon)
- 21D (Eta)
- 21F (Iota)
- 21G (Lambda)
- 21H (Mu)
- 21L (Omicron)
- 21K (Omicron)
- 21M (Omicron)
- 19A
- 19B
- 20A
- 20C
- 20G
- 20F
- 20E (EU1)
- 20B
- 20D



Epidemic Calculator



<https://gabgoh.github.io/COVID/index.html>

Transmission Dynamics

Population Inputs

Size of population.

7,000,000

Basic Reproduction Number \mathcal{R}_0

Measure of contagiousness: the number of secondary infections each infected individual produces.

Transmission Times

Length of incubation period, T_{inc} .

5.20 days

Duration patient is infectious, T_{inf} .

Clinical Dynamics

Mortality Statistics

Case fatality rate.

2.00 %

Recovery Times

Length of hospital stay

28.6 Days

Care statistics

Hospitalization rate.

20.00 %

Time from end of incubation to death.

Recovery time for mild cases

Time to hospitalization.

1-way Analysis Of Variance (ANOVA)



<https://saskiacotto.de/shiny/>

Mean of population 1, μ_1 :

10 13 40

Mean of population 2, μ_2 :

10 23 40

Mean of population 3, μ_3 :

10 26 40

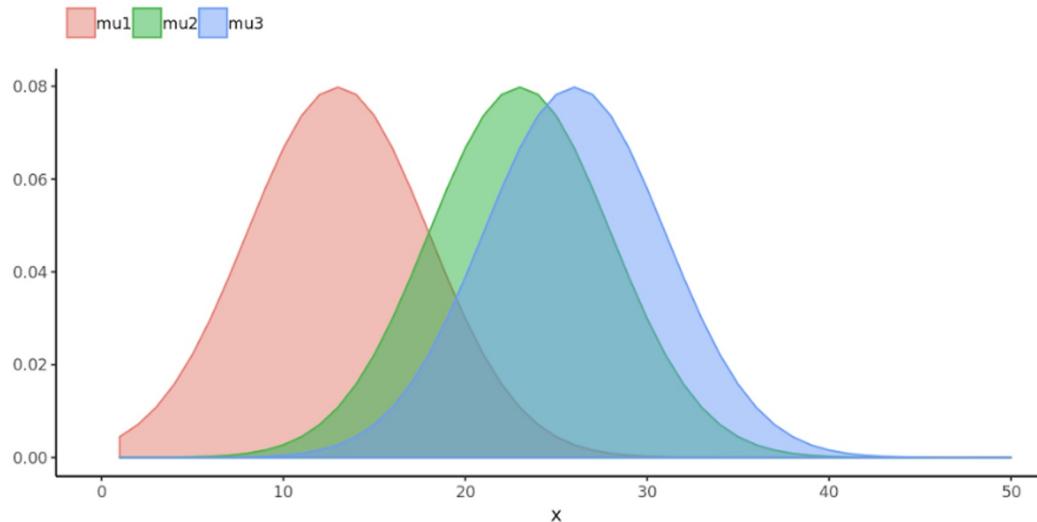
Population standard deviation, σ :

1 5 20

Sample size

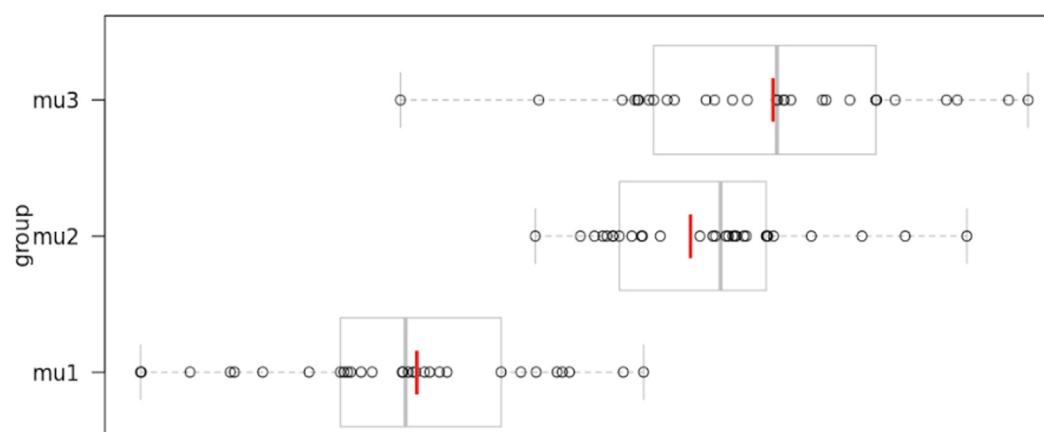
2 30 100

Population distributions



Observed sample data

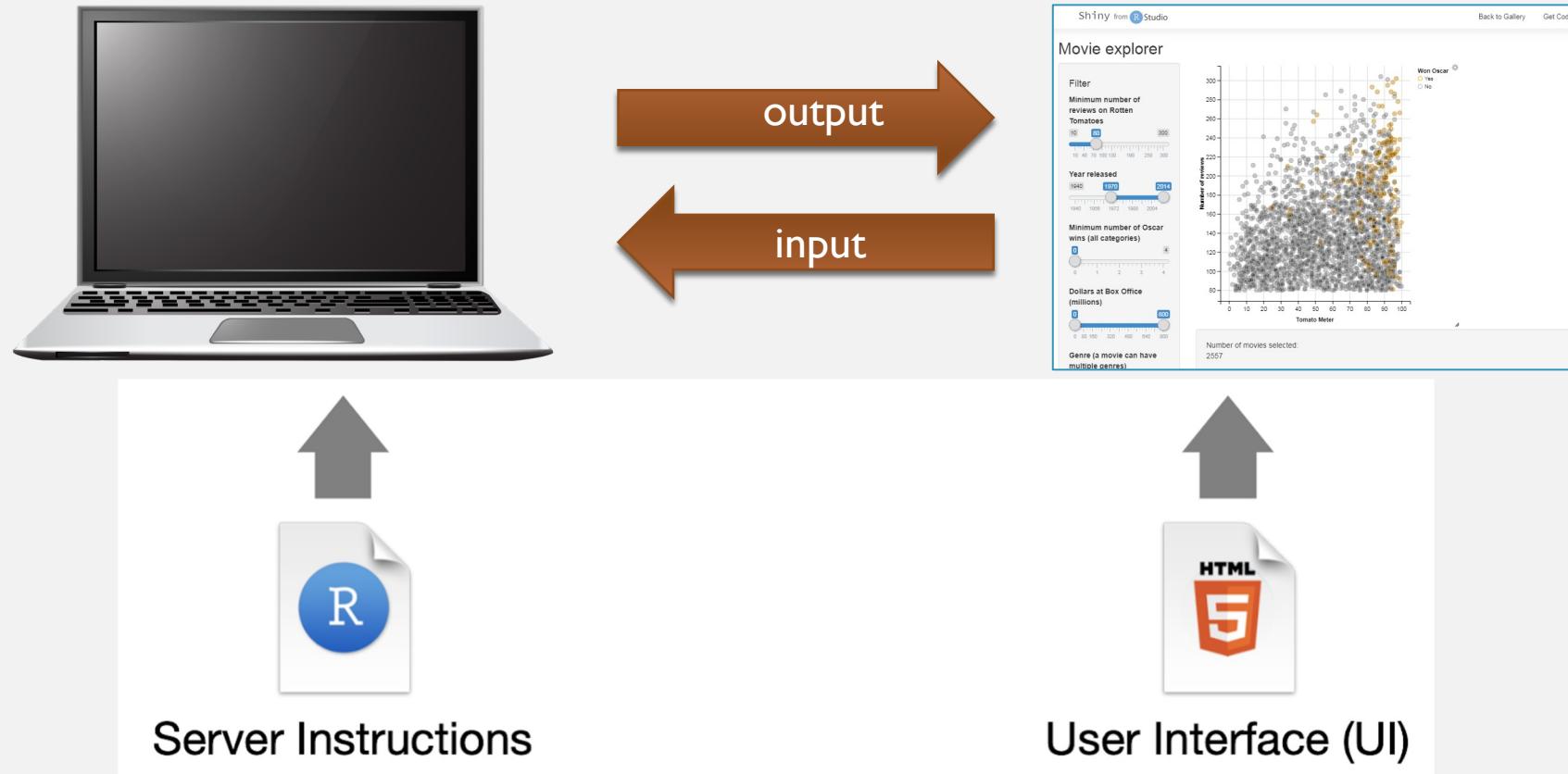
The red bars denote group means.



HOW TO BUILD AN APP

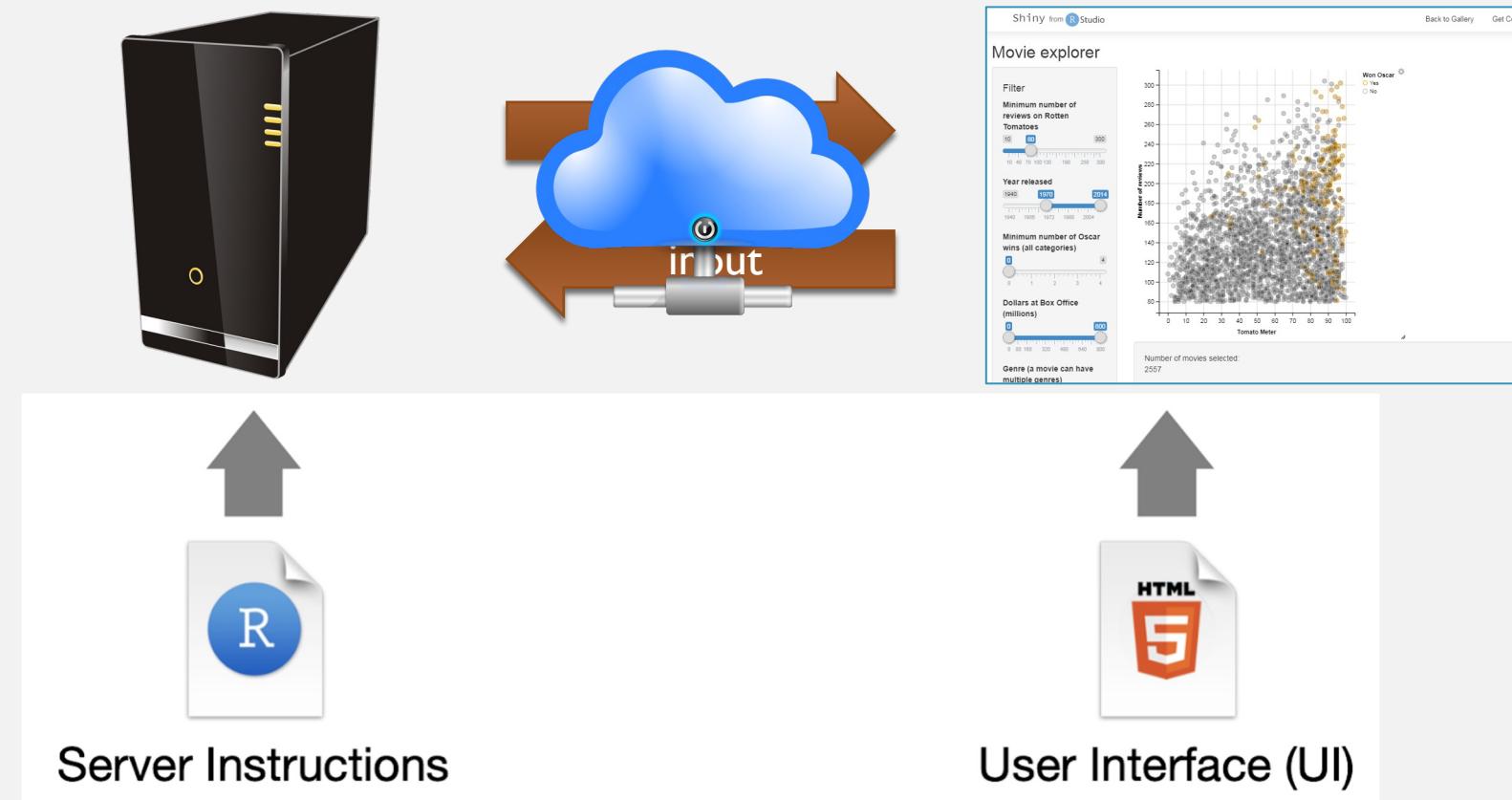
STRUCTURE OF A SHINY APP

A shiny app us a website connected to a computer with an active R session



ARQUITECTURA DE SHINY APP

Una Shiny app es una página web conectada a un servidor con sesión abierta de R



BASIC COMPONENTS

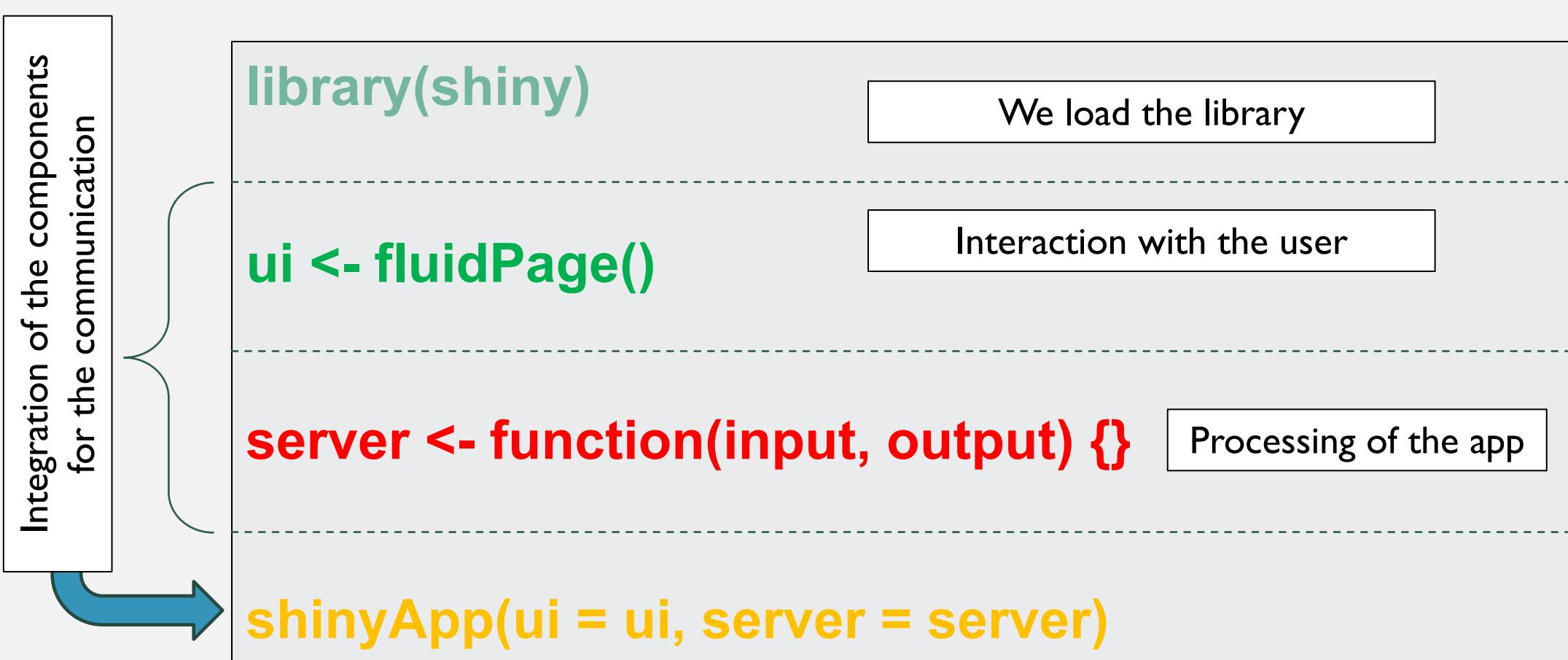
```
library(shiny)

ui <- fluidPage()

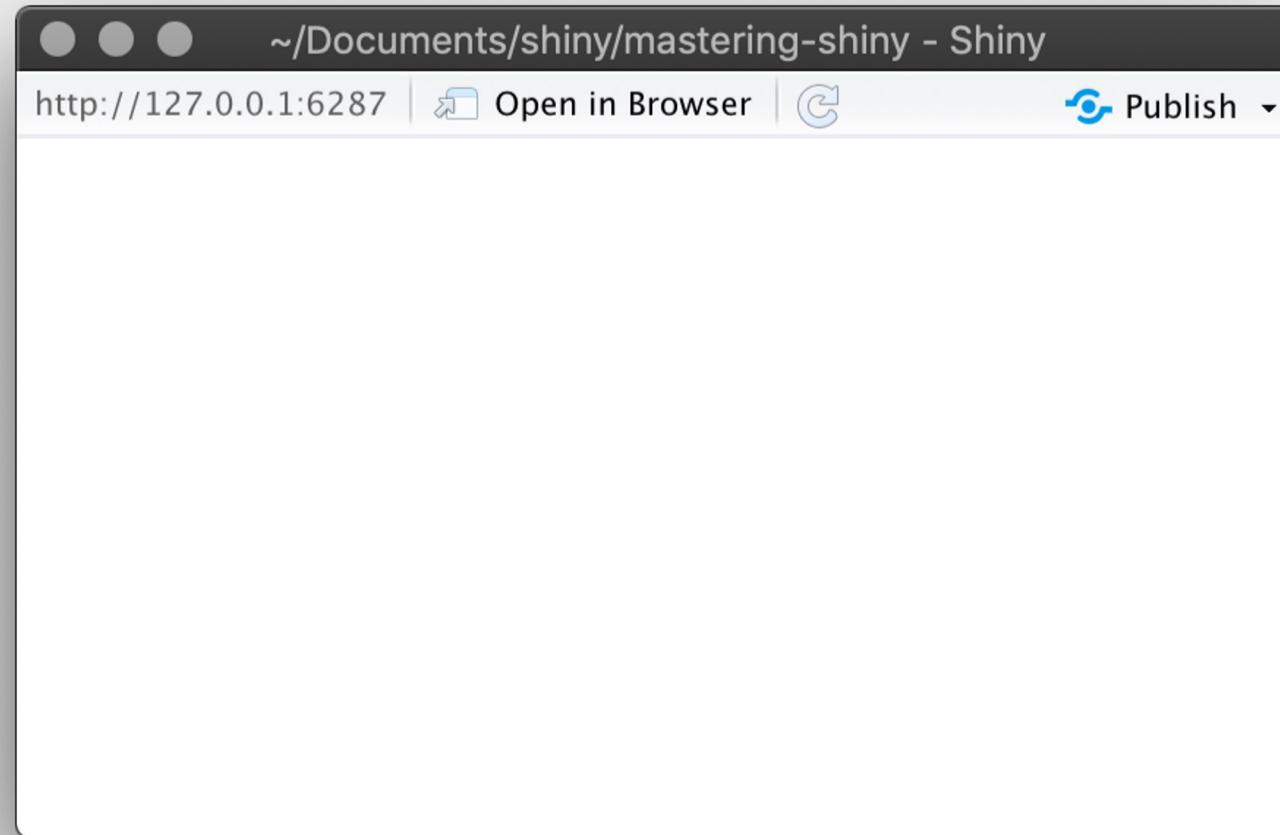
server <- function(input, output) {}

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

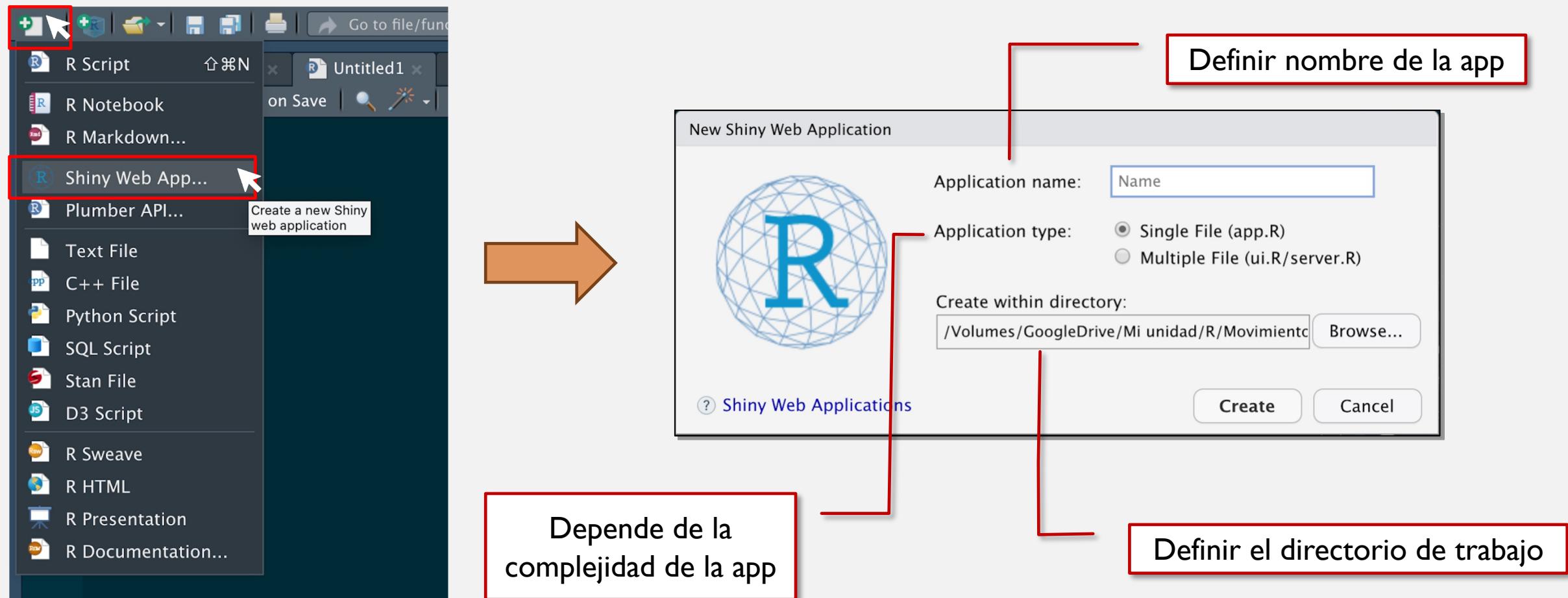
BASIC COMPONENTS



PLANTILLA DE LA APP

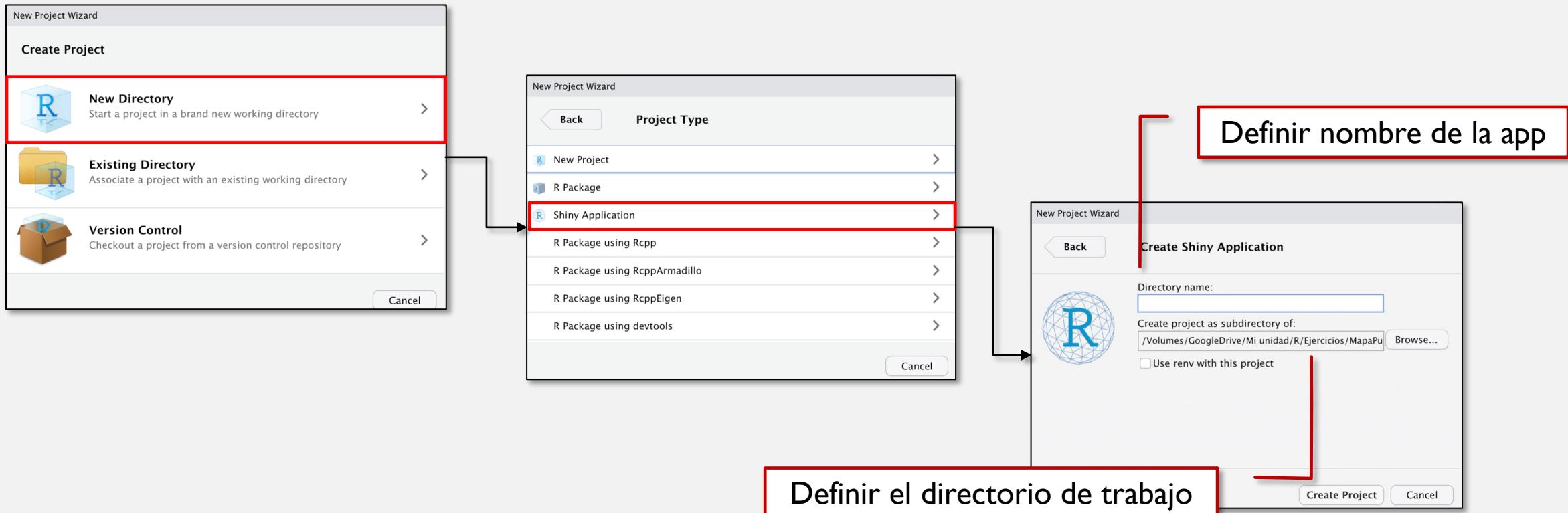


EXERCISE I: CREATING YOUR FIRST APP IN R STUDIO



EJERCICIO I: CREAR UN PROYECTO DE SHINY EN RSTUDIO

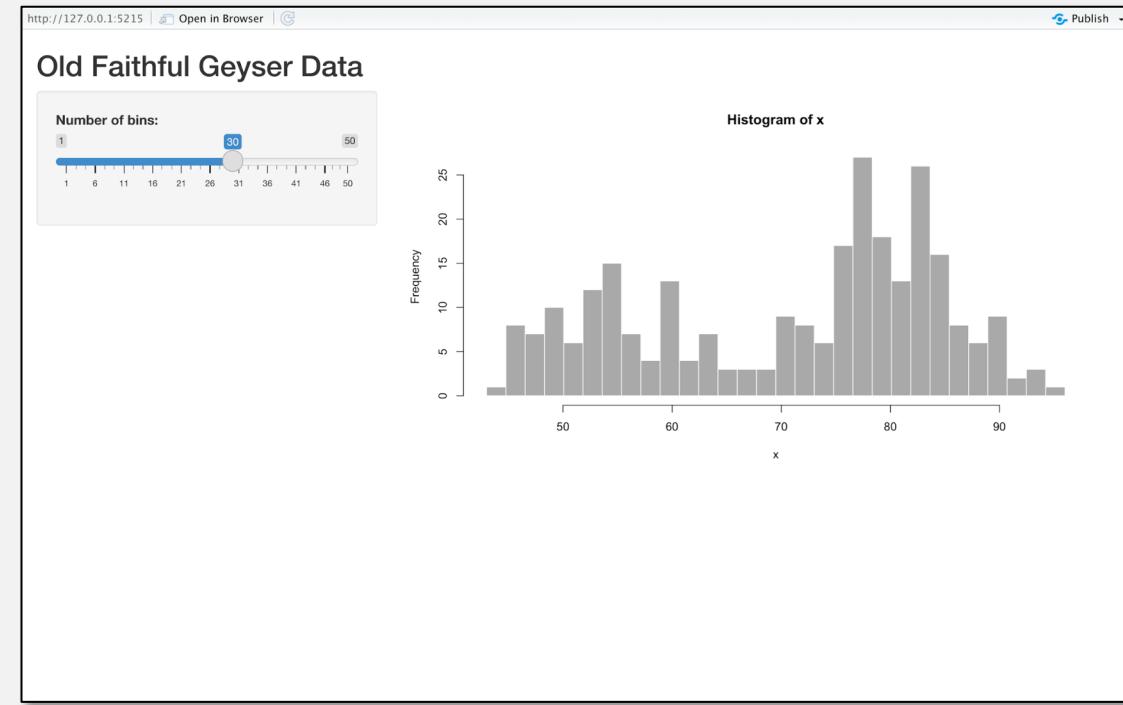
File > New Project > New Directory > Shiny Application > Create Shiny Application



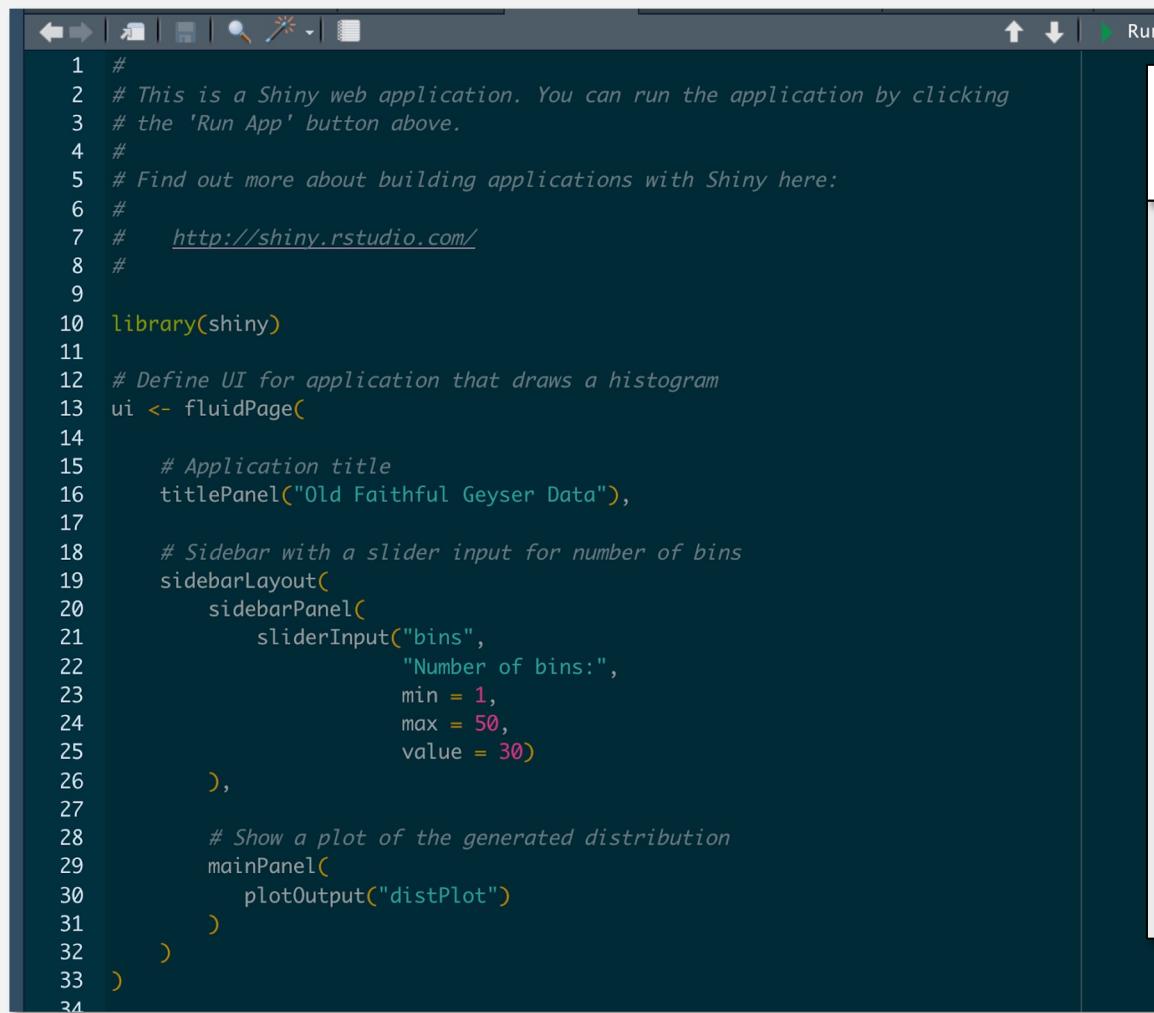
EJERCICIO I: CREAR UN PROYECTO DE SHINY EN RSTUDIO

Abre la app almacenada en un directorio a directory with runApp().

```
1 #  
2 # This is a Shiny web application. You can run the application by clicking  
3 # the 'Run App' button above.  
4 #  
5 # Find out more about building applications with Shiny here:  
6 #  
7 #     http://shiny.rstudio.com/  
8 #  
9  
10 library(shiny)  
11  
12 # Define UI for application that draws a histogram  
13 ui <- fluidPage(  
14  
15     # Application title  
16     titlePanel("Old Faithful Geyser Data"),  
17  
18     # Sidebar with a slider input for number of bins  
19     sidebarLayout(  
20         sidebarPanel(  
21             sliderInput("bins",  
22                 "Number of bins:",  
23                 min = 1,  
24                 max = 50,  
25                 value = 30)  
26         ),  
27  
28         # Show a plot of the generated distribution  
29         mainPanel(  
30             plotOutput("distPlot")  
31         )  
32     )  
33 )  
34 )
```



EJERCICIO I: CREAR UN PROYECTO DE SHINY EN RSTUDIO



```
1 #  
2 # This is a Shiny web application. You can run the application by clicking  
3 # the 'Run App' button above.  
4 #  
5 # Find out more about building applications with Shiny here:  
6 #  
7 #   http://shiny.rstudio.com/  
8 #  
9  
10 library(shiny)  
11  
12 # Define UI for application that draws a histogram  
13 ui <- fluidPage(  
14  
15   # Application title  
16   titlePanel("Old Faithful Geyser Data"),  
17  
18   # Sidebar with a slider input for number of bins  
19   sidebarLayout(  
20     sidebarPanel(  
21       sliderInput("bins",  
22         "Number of bins:",  
23         min = 1,  
24         max = 50,  
25         value = 30)  
26     ),  
27  
28     # Show a plot of the generated distribution  
29     mainPanel(  
30       plotOutput("distPlot")  
31     )  
32   )  
33 )
```

Identificar dentro del código los elementos de la estructura de la app

librerias

library(shiny)

User interface

ui <- fluidPage()

Server

server <- function(input, output) {

Run

shinyApp(ui = ui, server = server)