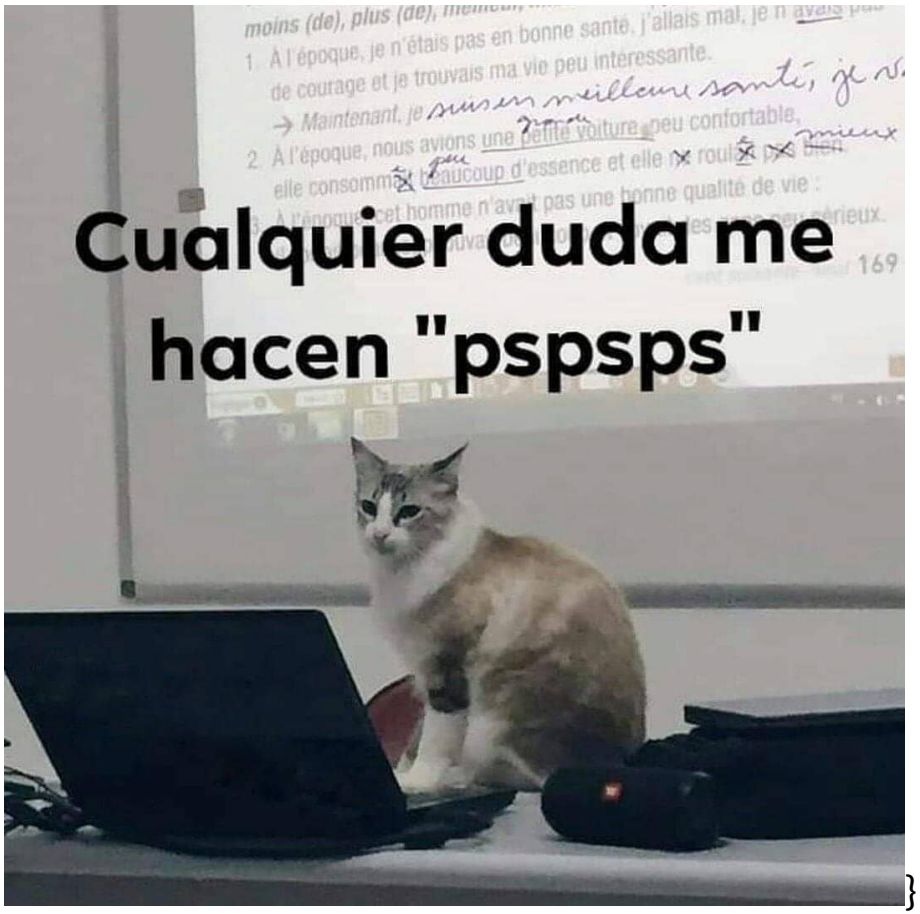
# Introducción a Shiny app de R

Mayo 2021

# Índice

# Cualquier duda y/o comentario es bienvenida :)




}

# R Shiny app


## Descripción:

- Shiny es un paquete que permite crear aplicaciones web interactivas, directamente desde Rstudio, este paquete viene instalado por defecto con R. Y tiene como finalidad la interacción gráfica con los datos.

- R Shiny App es un paquete destinado al desarrollo de aplicaciones interactivas de visualización de datos. Permite la instalación de una aplicación web corriendo en R en un servidor, y que podremos personalizar de forma sencilla con sintaxis HTML, CSS o Javascript.

## Componentes:

Una app de shiny consta (al menos) de dos archivos:

- Un script para la interfaz del usuario (ui.R), que recibe los inputs y muestra los outputs.
- Un script para los cálculos (server.R) que realiza los cálculos necesarios.

# Referencias de shiny app

En el sitio oficial de Shiny donde hay una variedad de plataformas, artículos, foros y demás elaboradas con shiny de R. Shiny app tiene una gran comunidad de entusiasta desarrolladores que comparten lo elaborado con esta paquetería a los demás usuarios.

Aquí encontrarás un manual tambien llamada hoja de referencia en la cual se presenta las funciones que integra una shiny dividida por UI y Server los cuales en su conjunto forman una shiny.

El paquete Shiny tiene once ejemplos incorporados que muestran a grandes razgos como funciona:

```
library(shiny)
#runExample ("01_hello")      # histograma
#runExample ("02_text")       # resumen y tabla
#runExample ("03_reactivity") # desplegables reactivos
#runExample ("04_mpg")        # boxplot reactivos
#runExample ("05_sliders")    # deslizadores reactivos
#runExample ("06_tabsets")    # varias pestanas
#runExample ("07_widgets")    # textos de ayuda y botones
#runExample ("08_html")       # Shiny desde HTML
#runExample ("09_upload")     # carga de archivos
#runExample ("10_download")   # descarga de archivos
#runExample ("11_timer")      # dia y hora
```

## Libros y material electrónico

- Learn ggplot2 Using Shiny App - Keon Woong Moon
- Interactive web-based data visualization with R, plotly, and shiny- Carson Sievert

## Más ejemplo de shiny con sus código al público

- Portafolio de presentación de Xiscapericas
- Plataforma interactiva de Diego López

# Crear un proyecto nuevo

Antes de crear la aplicación es necesario tener un orden de los elementos que integrarán la shiny, es por eso que una ventaja de los proyectos es que cualquier fichero que creemos o elemento que utilizaremos se guardará en la carpeta del proyecto que acabamos de crear.

- Hacer clic en el menú **Archivo**, luego en **Nuevo proyecto**.
- Hacer clic en **Nuevo directorio** .
- Hacer clic en **Nuevo proyecto**.
- Introducir el nombre del directorio para guardar tu proyecto, por ejemplo: **shiny_cucea**.

# Elementos que utilizaremos:

El material de esta sesion se encuentra en:

- Repositorio

Basado en el trabajo de:

- GitHub de Campos J

# Deployment de shiny app

Antes de construir la shiny, debemos configurar.

Para esto, nos vamos a hacer una cuenta en Shinyapps.io. Hay varios tipos de cuenta, con distintos objetivos y con sus precios correspondientes. Como siempre, la gente de RStudio ofrece una opcion gratuita con las caracteristicas suficientes para comenzar a probar.

El plan gratuito incluye 5 aplicaciones alojadas y hasta 25 horas de actividad mensuales (la app permanece idle hasta algun acceso). Esto ultimo es muy importante a la hora de subir la version final de nuestra aplicacion, ya que las horas se contabilizan por ciclo mensual y cuando se alcanza el limite en la cuenta gratuita deja de estar disponible para el acceso publico hasta el proximo ciclo.

# Vincular cuenta en RStudio

- En el menú principal clic en **Tools**.
- Clic en **Global options**.
- Clic en **Publishing**.
- Después clic en **connect**.
- Se abrirá un cuadro de diálogo de Connect Account, dar clic en **Shinyapps.io**.
- Dar clic en **your accout on Shinyapps** la cual te lleva al sitio oficial. Deberás iniciar sesión (sing up) ya sea que creas tu cuenta o que ingreses con tu email.
- Una vez inscrito en la página, ir al menú de **Account** y posterior a **Tokens**
- Clic en **Show secret** y después en **Copy to clipboard** deberás asegurarte que lo has copiado correctamente.
- Una vez copiado, lo pegas en el recuadro blanco, le das clic **Connect...**
- Dado el último paso verás que ya se intrego tu cuenta, es posible que algunas paqueterias se descarguen en automatico ya que R se debe enlazar a la web.
- A continuación clic en **Apply**, y después en **OK**.

Nuestra cuenta ya se encuentra vinculada al IDE. Esto nos va a permitir publicar la app directamente desde Rstudio con suma facilidad y con bastante control.

Procedimiento ilustrativo

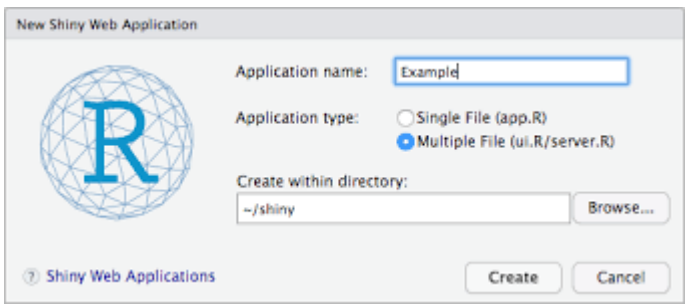
# Creación shiny app de R

# Creación de shiny

Una aplicación shiny tiene 2 funciones o archivos, uno llamado ui en donde se describe la interfaz gráfica de la aplicación, y otro llamado server en donde se escribe la lógica de la aplicación.

## Para crear una nueva aplicación shiny:

- Haga click en la pestaña **Archivo**.
- Seleccione la opción **Nuevo archivo**.
- Seleccione **Shiny Web App**.
- Ponga nombre a su aplicación **(Sugerencia: shiny_example)**.
- En la opcion Application type, seleccione **Single File (app.R)**.
- Guarde su aplicación en una carpeta de fácil acceso



# Importar bases de datos y manipulación que integrarán la shiny

Librerias necesarias:

```
library(shiny)  # permite la generación de app
library(shinythemes) # personalizar theme en shiny

library(readxl) # leer archivos xlxs
library(plotly) # genera gráficos interactivos
library(dplyr)  # manipulación de dataframes

library(readxl) # importar archivos xlxs
```

Base de datos (df) con los registro de la deuda interna por país:

```
df <- read_excel("/Users/paolamanzor/Desktop/Taller/shiny_cucea/shiny_cucea/inflac

head(df, n=7)
```

```
## # A tibble: 7 x 6
##   pais              inflacion   aÒo codigo region             continente
##   <chr>                 <dbl> <dbl> <chr>  <chr>              <chr>
## 1 Afganistan          0.0224   2019 AFG    Asia meridional     Asia  
## 2 Albania             0.00382  2019 ALB    Europa meridional   Europa &nbs
## 3 Alemania            0.0214   2019 DEU    Europa occidental   Europa &nbs
## 4 Algeria            -0.00651  2019 ALGR   África meridional   África &nbs
## 5 Andorra             0.0151   2019 AND    Europa meridional   Europa &nbs
## 6 Angola              0.359    2019 AGO    Europa meridional   Europa &nbs
## 7 Antigua y Barbuda   0.0251   2019 ATG    Caribe              América &nb
```

# Manipulación de datos, renombrar variables

```
df %>% names()
```

```
## [1] "pais"       "inflacion"  "aÒo"        "codigo"     "region"
## [6] "continente"
```

```
names(df)[names(df) == "aÒo"] <- "año"
names(df)[names(df) == "codigo"] <- "abreviatura"

#Renombrados
df %>% names()
```

```
## [1] "pais"        "inflacion"   "año"         "abreviatura" "region"
## [6] "continente"
```

```
#Creación de un conteo o ranking
df %>%
    select(pais, inflacion, abreviatura, continente, region) %>%
    filter(inflacion!=0) %>%
    arrange( -as.numeric(inflacion)) %>%
    mutate(rank = 1:n())->  ranked_by_inf
```

# Generando ranking para la visualización

```
# Primeros 10 paises con mayor inflación

#dput(ranked_by_inf_less$pais)
ranked_by_inf_top10<-ranked_by_inf %>%
    filter(rank<=10)%>%
    mutate(pais=factor(pais,
                       levels=c("Argentina",  "Angola",     "Uzbekistan", "Sudan",
                                "Surinam",    "Haiti",      "Turquia",    "Egipto"
                                "Etiopia",    "Nigeria")))

ranked_by_inf_top10 %>% head(n=2)
```

```
## # A tibble: 2 x 6
##   pais      inflacion abreviatura continente region              rank
##   <fct>         <dbl> <chr>       <chr>      <chr>              <int>
## 1 Argentina     0.515 ARG         América    América del Sur  
## 2 Angola        0.359 AGO         Europa     Europa meridional  
```

```
# Últimos 10 paises con menor inflación
ranked_by_inf_less<-ranked_by_inf %>%
    filter(rank>=165) %>%
    mutate(pais=factor(pais,
                       levels=c("Emiratos Arabes Unidos", "Burkina Faso",
                                "Brunei",                 "Oman",
                                "Liberia",                "Catar",
                                "Zimbabue",               "Kuwait",
                                "Guinea Ecuatorial",      "Guinea Bissau")))
```

# Construcción de UI

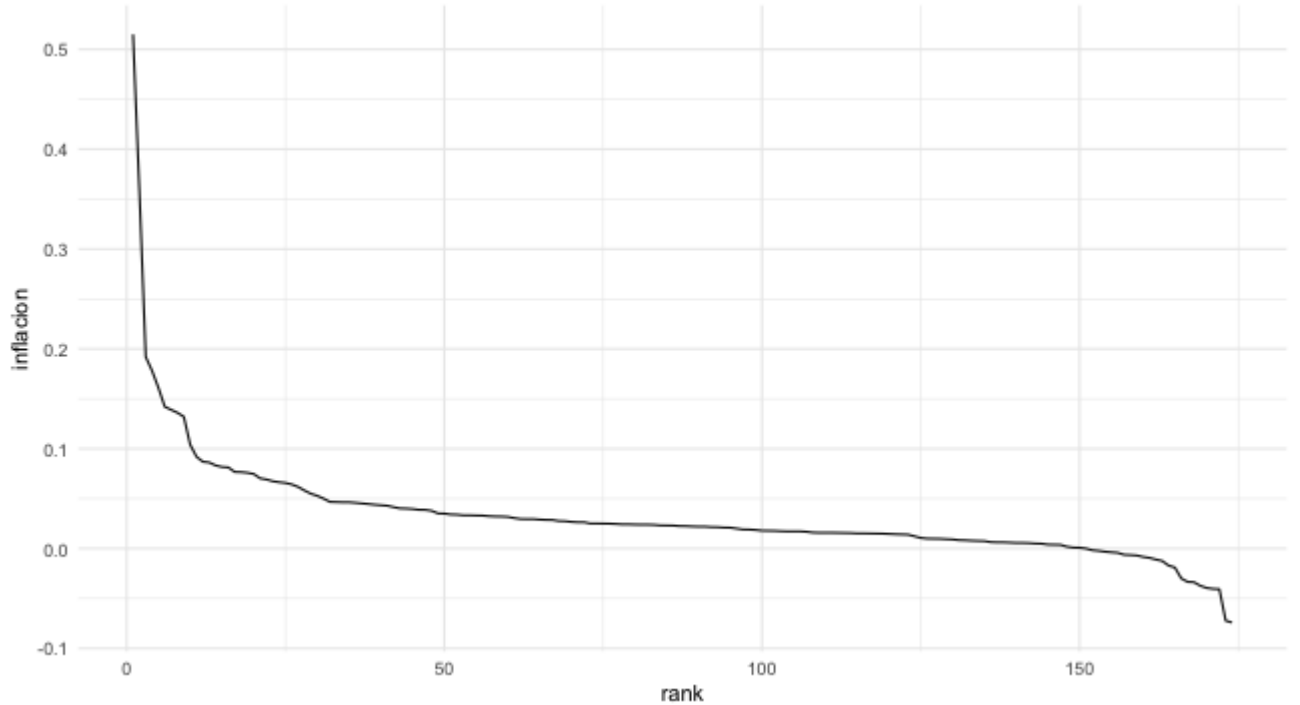
```
ui <- navbarPage("inflacion mundial",
                 theme = shinytheme("united"),
                 tabPanel("Inicio", icon = icon("home"),
                          "Inflación",
                          plotlyOutput("rank")),

                 tabPanel("Visualizaciones", icon = icon("bar-chart-o"),
                          h2(aling= "center", "Mapa mundial de inflacion"),
                          h4(align = "justify","El mapa interactivo muestra la inf
                          sliderInput(inputId = "rango",
                                      label = "Ranking",
                                      min = 1,
                                      max = 174,
                                      value = c(1,174),
                                      sep = ""),
                          submitButton(text = "Create my plot!"),
                          plotOutput(outputId = "lineal"),
                plotlyOutput("plot", width = "1200px"),
                plotlyOutput("top10"),
                plotlyOutput("less10"),

                 ),
                tabPanel("Descarga",icon = icon("fas fa-cloud-download-alt"),
                         downloadButton('downloadData', 'Download'),
                         DT::dataTableOutput("table1"))
)
```

# Construcción de Server

```
#Gráfico lineal
    ranked_by_inf %>%
        ggplot() +
        geom_line(aes(x = rank, y = inflacion)) +
        theme_minimal()
```

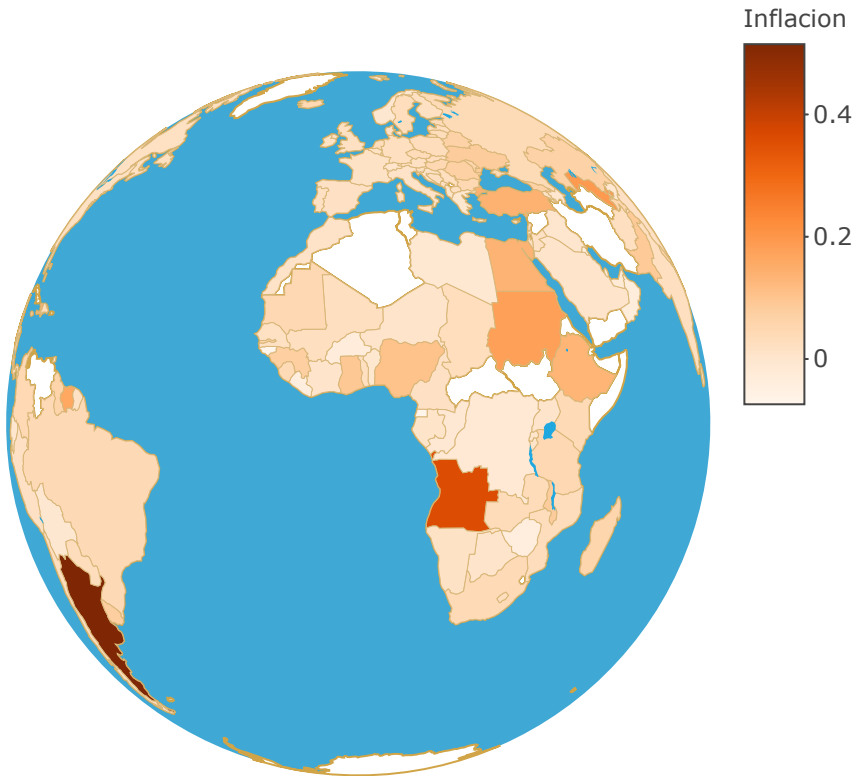


```r
#Mapa mundial
    #Set country boundaries as light grey
        l <- list(color = toRGB("#d9b779"), width = 0.5)
        #Specify map projection and options
        g <- list(
            showframe = FALSE,
            showcoastlines = FALSE,
            projection = list(type = 'orthographic'),
            resolution = '100',
            showcountries = TRUE,
            countrycolor = '#d1a547',
            showocean = TRUE,
            oceancolor = '#3fa8d4',
            showlakes = TRUE,
            lakecolor = '#1fa8e0',
            showrivers = F,
            rivercolor = '#24AAEC')

        p <- plot_geo(df) %>%
            add_trace(z = ~inflacion, color = ~inflacion, colors = 'Oranges',
                      text = ~pais, locations = ~abreviatura, marker = list(line =
            colorbar(title = 'Inflacion') %>%
            layout(title = '', geo = g)
```

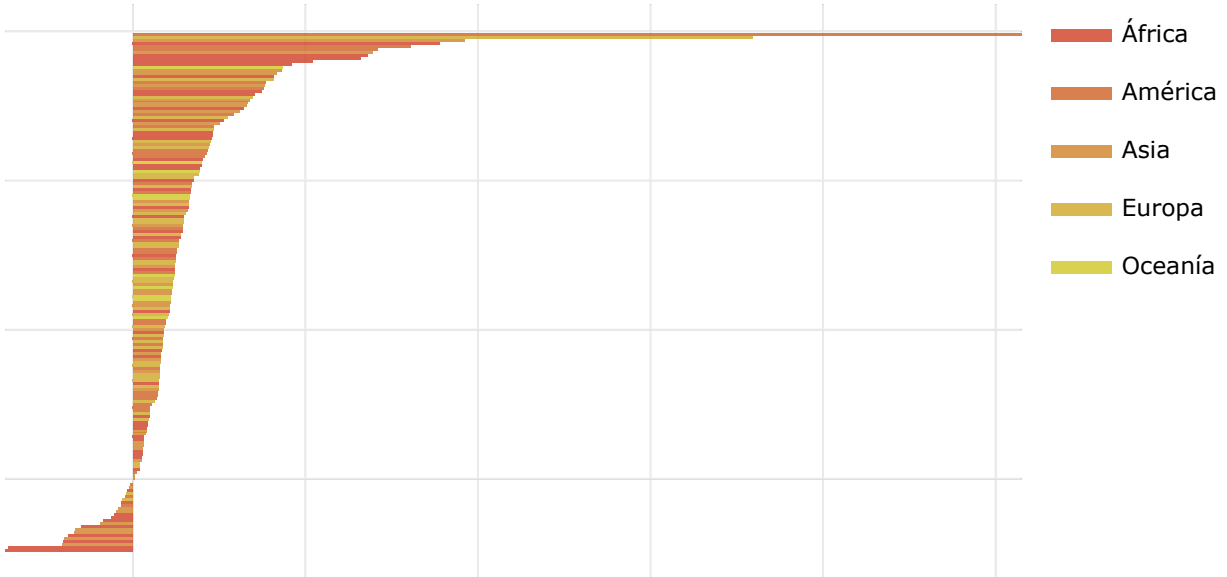
```
p <- plot_geo(df) %>%
            add_trace(z = ~inflacion, color = ~inflacion, colors = 'Oranges',
                      text = ~pais, locations = ~abreviatura, marker = list(line =
            colorbar(title = 'Inflacion') %>%
            layout(title = '', geo = g)

          p
```



```
rank<-ggplot(ranked_by_inf, aes(rank, group = continente,
                                fill = as.factor(continente), color = as.f
            geom_tile(aes(y = inflacion/2,
                          height = inflacion,
                          width = 0.9), alpha = 0.8, color = NA) +
            scale_fill_manual(values=c("#d03f26", "#d06126", "#d08326",    "#d0a52
            coord_flip(clip = "off", expand = FALSE) +
            scale_y_continuous(labels = scales::comma) +
            guides(color = FALSE, fill = FALSE) +
            labs(title='', x = "", y = " ") +
            scale_x_reverse()+
            theme_minimal()+
            theme(axis.ticks.y= element_blank(),
                  axis.text.y = element_blank())

        ggplotly(rank, dynamicTicks = TRUE)
```



# Tabla

```
downloadHandler(
        filename = 'Download.csv',
        content = function(file) {
            write.csv(Data[input[["table1_rows_all"]],], file, row.names = FALSE)}
    )
```

Download

```
ranked_by_inf
```

```
## # A tibble: 174 x 6
##    pais       inflacion abreviatura continente region                  rank
##    <chr>          <dbl> <chr>       <chr>      <chr>                  <int>
##  1 Argentina      0.515 ARG         América    América del Sur  
##  2 Angola         0.359 AGO         Europa     Europa meridional  
##  3 Uzbekistan     0.192 UZB         Asia       Asia central  
##  4 Sudan          0.178 SDN         África     África septentrional  
##  5 Surinam        0.161 SUR         América    América del Sur  
##  6 Haiti          0.142 HTI         América    Caribe  
##  7 Turquia        0.139 TUR         Asia       Asia occidental  
##  8 Egipto         0.136 EGY         África     África septentrional  
##  9 Etiopia        0.132 ETH         África     África oriental  
## 10 Nigeria        0.104 NGA         África     África occidental  
## # … with 164 more rows
```

# ¡Gracias por tu atención!

En este momento puedes realizar más preguntas que tengas.