

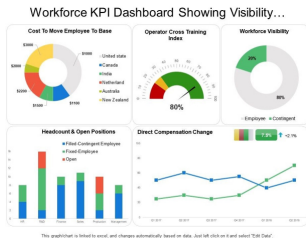
# Análisis Cluster y Aplicaciones personalizables en Shiny

Karen Calva  
Sociedad Ecuatoriana de Estadística  
R Weekend

24/10/2020

# Dashboard o cuadro de mando, ¿Qué es?

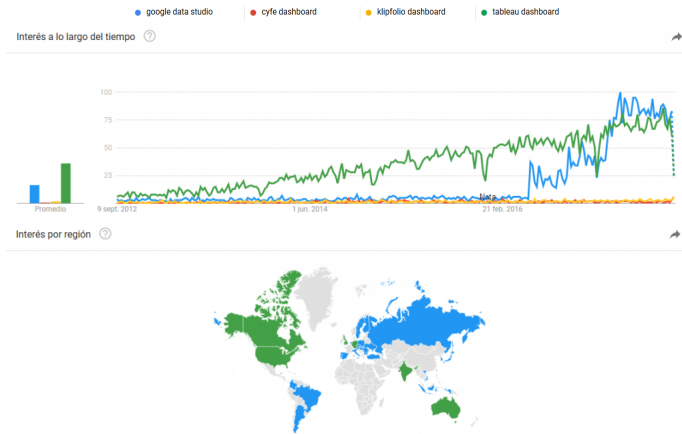
Click Aquí.



El objetivo principal de un dashboard es ofrecer visibilidad sobre los indicadores que deben alcanzar determinado valor para alcanzar los objetivos de la empresa. Además permite conocer datos de valor, analizar tendencias y fijar líneas de actuación.

# ¿Con qué herramientas se hacen dashboards?

Excel, Qlik, Tableau, Google Data Studio, Powe BI, son algunas de las herramientas más conocidas para hacer dashboards.



Introducción

Motivación

Shiny

Componentes  
Shiny

# Aplicaciones Web Interactivas, ¿Qué son?

Análisis Cluster y  
Aplicaciones  
personalizables  
en Shiny

Karen Calva  
Sociedad  
Ecuatoriana de  
Estadística  
R Weekend

Introducción

Motivación

Shiny

Componentes  
Shiny

Una aplicación web es programa informático interactivo, construido con tecnologías web (HTML, CSS, JS), que almacena (Base de datos, Archivos) y manipula datos (CRUD).

## Web Development



# Data here, data there, data everywhere

Análisis Cluster y  
Aplicaciones  
personalizables  
en Shiny

Karen Calva  
Sociedad  
Ecuatoriana de  
Estadística  
R Weekend

Introducción

Motivación

Shiny

Componentes  
Shiny



# Procesamiento de los datos con R

Análisis Cluster y  
Aplicaciones  
personalizables  
en Shiny

Karen Calva  
Sociedad  
Ecuatoriana de  
Estadística  
R Weekend

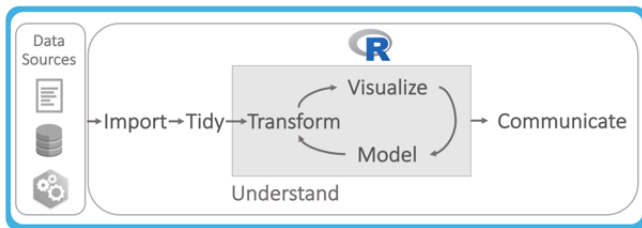
Introducción

Motivación

Shiny

Componentes  
Shiny

## R for Data Science



[R for Data Science, Wickham & Grolemund](#)

# Procesamiento de los datos con R

Análisis Cluster y  
Aplicaciones  
personalizables  
en Shiny

Karen Calva  
Sociedad  
Ecuatoriana de  
Estadística  
R Weekend

Introducción

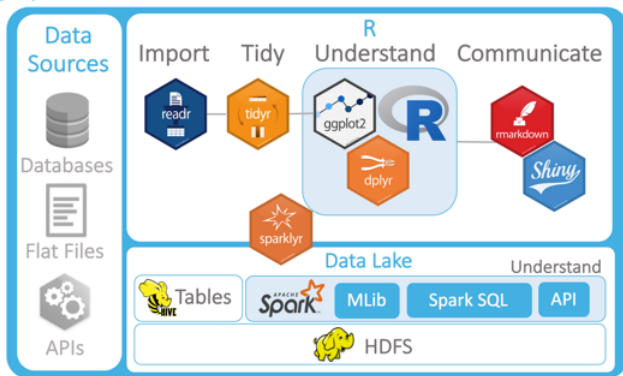
Motivación

Shiny

Componentes  
Shiny



## Using Spark & R for Data Science



Shiny es una herramienta para crear fácilmente aplicaciones web interactivas (apps) que permiten a los usuarios interactuar con sus datos sin tener que manipular el código.

## Características principales:

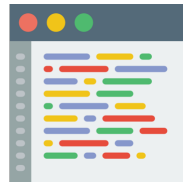
- Desarrollado por RStudio
- Fácil de aprender : Pensado para los usuarios de R.
- No es necesario conocer HTML, CSS, JavaScript.
- Versión gratis open source y Versión Pro de pago.
- Algunas aplicaciones de ecuatorianos:
  - Pronóstico e impacto de niveles extremadamente bajos de caudal en centrales hidroeléctricas, autor: David Pachacama.
  - Calculadora actuarial, autor: David Pachacama.
  - Georreferenciación de clústers, autor: Karen Calva.
  - Predicción del rendimiento académico, autor: Karen Calva.





## Características principales:

- Desarrollado por RStudio
- Fácil de aprender : Pensado para los usuarios de R.
- No es necesario conocer HTML, CSS, JavaScript.
- Versión gratis open source y Versión Pro de pago.
- El propio package de Shiny permite utilizar la versión gratis open source de la IDE RStudio como servidor no dedicado y levantar las aplicaciones en local.

```
ui =  
dashboardPage(  
  dashboardHeader(),  
  dashboardSidebar(),  
  dashboardBody()  
)
```



Nombre	Fecha de modificac...	Tipo	Tamaño
 server.R	04/06/2014 17:30	Archivo R	1 KB
 ui.R	04/06/2014 17:40	Archivo R	1 KB

## Implementación:

- **server.R** Es la parte servidor de la aplicación
- **ui.R** Es la parte cliente de la aplicación, el interfaz gráfico.

## Levantar la aplicación/servidor:

- En local con RStudio en cualquier sistema operativo.
- En un servidor linux dedicado con el Server Shiny de RStudio.
- En algun servicio de hosting especializado para Shiny  
**ShinyApps.io**

## ui.R

```
library(shiny)
shinyUI(fluidPage(
  titlePanel("Ejemplo 01"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
        "Number of bins:",
        min = 1,
        max = 100,
        value = 50)
    ),
    mainPanel(
      plotOutput("distPlot")
    )
  )
))
```

## server.R

```
library(shiny)
x <- rchisq(100000, df = 4)
shinyServer(function(input, output) {
  output$distPlot <- renderPlot({
    hist(x,
      breaks = seq(min(x), max(x),
        length.out = input$bins + 1),
      col = 'darkgray', border = 'white')
  })
})
```

## Tipos de componentes:

- Componentes de Inputs Outputs
- Componentes HTML (tags)

## Ejemplo de aplicación

## Inputs

- numericInput
- sliderInput
- textInput
- selectInput
- radioButtons
- checkboxGroupInput
- checkboxInput
- dateInput
- dateRangeInput
- actionButton
- submitButton
- fileInput

## Outputs

- server.R  $\longleftrightarrow$  ui.R
- renderPlot  $\longleftrightarrow$  plotOutput
- renderImage  $\longleftrightarrow$  imageOutput
- renderText  $\longleftrightarrow$  textOutput
- renderPrint  $\longleftrightarrow$  verbatimTextOutput, htmlOutput
- renderTable  $\longleftrightarrow$  tableOutput
- renderDataTable  $\longleftrightarrow$  dataTableOutput
- renderUI  $\longleftrightarrow$  uiOutput
- downloadHandler  $\longleftrightarrow$  downloadButton, downloadLink

## Componente

## Subcomponente

- navbarPage()  $\iff$  tabPanel(), navbarMenu()
- navbarMenu()  $\iff$  tabPanel()
- navlistPanel()  $\iff$  tabPanel()
- titlePanel()
- sidebarLayout()  $\iff$  sidebarPanel() mainPanel() (obligatorio)
- sidebarPanel()
- mainPanel()
- tabsetPanel()
- tabPanel()

- HTML(text, ...)
- h1(...), h2(...), h3(...), h4(...), h5(...), h6(...)
- p(...), pre(...), code(...)
- div(...), span(...), strong(...), em(...)
- a(...), img(...), br(...), hr(...)

## Dynamic Inputs o Dynamic UI ui.r

```
...
uiOutput("IDDynamicInput")
...
```

## server.r

```
...
output$IDDynamicInput = renderUI({
  if (input$tabpanels!=2) return()
  sliderInput(inputId="maxGraphMovies", label="Numero de nodos", value=maxGraphMovies, min=2, max=50, step=1)
})
...
if (!is.null(input$maxGraphMovies)) maxGraphMovies <- input$maxGraphMovies
...
```

# Entorno de las variables

Análisis Cluster y  
Aplicaciones  
personalizables  
en Shiny

Karen Calva  
Sociedad  
Ecuatoriana de  
Estadística  
R Weekend

Introducción

Motivación

Shiny

Componentes  
Shiny

- **global.R** : Visibles en server.R y ui.R
- **server.R**: antes de shinyServer : Compartidas para todas las sesiones
- **server.R**: dentro de shinyServer : Propias de cada sesion

Compartir Aplicaciones:

- runUrl
- runGithub
- runGist
- Servidor Shiny

Recursos adicionales:

ggplot para gráficos  
kableExtra para las tablas  
Researching  
Academy  
shiny.rstudio  
shiny dashboard  
shiny material

