

## La enseñanza con Shiny

Ibar-Alonso, Raquel<sup>a</sup>, [raquel.ibar@urjc.es](mailto:raquel.ibar@urjc.es)  
Coscolluela-Martínez, Carolina<sup>a</sup>, [carolina.coscolluela@urjc.es](mailto:carolina.coscolluela@urjc.es)  
Sánchez-Muñoz, Estefanía<sup>a</sup>, [estefania.sanchez@urjc.es](mailto:estefania.sanchez@urjc.es)  
Franco Rodríguez-Lázaro<sup>b</sup>, Antonio, [fralaz@ceu.es](mailto:fralaz@ceu.es)  
Inchausti Tabuenca<sup>b</sup>, Elena, [inctab@ceu.es](mailto:inctab@ceu.es)  
Aguirre Arrabal<sup>b</sup>, Cristina, [aguiarr@ceu.es](mailto:aguiarr@ceu.es)  
Gutiérrez Jiménez<sup>b</sup>, Marta, [gutjim@ceu.es](mailto:gutjim@ceu.es)

*(a)Departamento de Economía Aplicada I, Historia de las Instituciones  
Económicas y Filosofía Moral  
Universidad Rey Juan Carlos*

*(b)Departamento de Matemática y Estadística  
Universidad CEU San Pablo*

### RESUMEN

La enseñanza tradicional de la estadística se ha mostrado en continua evolución para satisfacer el requerimiento del mercado. Hasta ahora, se han propuesto fórmulas tales como transparencias, diapositivas, programas de ordenador etc. Hoy en día, se propone una nueva herramienta, Shiny, que ha sido desarrollada en el entorno de R-Studio y que permite diseñar y crear aplicaciones web interactivas en lenguaje R. La incorporación de Shiny en clase permite realizar simulaciones de los problemas que se plantean y el alumno interactúa con la estadística obteniendo el resultado de los problemas de forma intuitiva. La experiencia dentro del aula ha sido positiva, los alumnos han sido propensos a contrastar los resultados de los problemas tradicionales con las apps creadas con Shiny y a comprobar si erraban o no en los resultados. El uso de software libre ha sido clave para la buena acogida de esta herramienta por parte de los estudiantes de Statistics I y Statistics II.

## **ABSTRACT**

There has been a continuous evolution of the methodology of teaching Statistics in order to satisfy the market requirements. Until now, the tools used have been slides, computer programs, transparencies, etc. Nowadays, it has been proposed another tool, Shiny, that has been developed under R-Studio framework. It allows to program interactive applications (apps) in R language. Shiny allows simulations of the exercises that are proposed in class, where the Students can interact with the app introducing different data and obtaining the solution to different problems. The experience teaching with shiny was positive. The Students wanted to test whether the results obtained were correct or not. The use of free software was determinant for the acceptance of Shiny between Students of Statistics I and Statistics II.

***Palabras claves:***

Shiny; enseñanza; estadística; metodología; innovación docente

***Área temática:*** Docencia

## 1. INTRODUCCIÓN

A mediados del siglo XX, la estadística se enseñaba utilizando libros con mucho contenido matemático, de dónde se estimaban los modelos que luego se usaban para resolver los problemas planteados. Más tarde, siguiendo la metodología anglosajona, empezaron a publicarse libros de estadística con contenido gráfico y con ejemplos prácticos de cada concepto, antes de pasar a exponer los ejercicios del tema en cuestión. Ya a finales del siglo XX, en los años 90 comenzaron a utilizarse como material docente las transparencias que permitía al profesor ofrecer a sus alumnos el contenido de la materia que explicaba de forma resumida. Posteriormente, estas transparencias se materializaron en diapositivas usando principalmente el programa comercial de office PowerPoint, o alguno de los programas Open Source similares.

En los tiempos actuales, donde el ciclo de vida natural de los productos es cada vez más corto y el flujo de información es constante y muy elevado, el profesor debe actualizar continuamente su material docente y las herramientas que propone para el aprendizaje de la Estadística. La aparición del Big Data y el manejo de grandes volúmenes de datos mediante programas como R (R-Studio) o Python, han propiciado que las clases teóricas sean complementadas con ejemplos dinámico que incorporan los avances tecnológicos que se tienen al alcance en cada momento. Actualmente, las aplicaciones de Shiny nos permiten dar un paso más hacia la enseñanza dinámica de la estadística.

La idea que subyace en esta forma de enseñar es fomentar la participación del alumno en el desarrollo de las clases. Inicialmente se sigue la metodología tradicional, el alumno estudia los conceptos teóricos y realiza ejercicios clásicos con un número reducido de datos, la novedad consiste en introducir en el aula la aplicación Shiny para reforzar los conocimientos adquiridos a través de la práctica.

El alumno puede utilizar la galería proporcionada por Shiny, los módulos desarrollados por el profesor o desarrollara él mismo sus aplicaciones. También se le

permite combinar código de la galería adaptándolo a las necesidades específicas en cada caso planteado. El objetivo que se persigue es que el alumno comprenda el por qué y para qué de aquello que está estudiando, que sea capaz de manejar un gran volumen de datos e interactuar con la aplicación para obtener diferentes resultados que respondan a las preguntas planteadas en el caso práctico propuesto.

En este estudio se ha observado si el alumno es propenso a la utilización de esta herramienta de enseñanza en el aula y si el tiempo dedicado a la creación de aplicaciones con Shiny le parece útil para sus estudios. La metodología se ha llevado a cabo durante el curso 2018-2019 en grupos de alumnos de primer curso del grado de Economía bilingüe de la Universidad Rey Juan Carlos de Madrid, en las asignaturas de *Statistics I* y *Statistics II*.

El resto del artículo se estructura de la siguiente manera: 2) Metodología, en la que se explica qué es Shiny, cómo se ha implantado y las aplicaciones que se han usado, 3) Resultados obtenidos y percepción del alumnado, y, 4) Conclusiones.

## **2. METODOLOGÍA**

Las primeras clases se utilizaron para explicar los conceptos de estadística correspondientes al temario propuesto en la asignatura y resolver ejercicios en pizarra con apoyo del ordenador (excel). Una vez finalizada la primera fase, se realizó la introducción a Shiny, explicando qué es y para qué sirve. La página web de Shiny<sup>1</sup> fue utilizada como fuente de información desde donde se les leyó literalmente lo que allí ponía como descripción de Shiny.

“Shiny is an R package that makes it easy to build interactive web apps straight from R. You can host standalone apps on a webpage or embed them in

---

<sup>1</sup> Página Web de Shiny: <https://shiny.rstudio.com/>

[R Markdown](#) documents or build [dashboards](#). You can also extend your Shiny apps with [CSS themes](#), [htmlwidgets](#), and JavaScript [actions](#).

Shiny combines the computational power of R with the interactivity of the modern web.”<sup>2</sup>

Se les transmitió a los alumnos que ahora su rol cambiaba y que ellos serían los responsables programadores que debía proponer soluciones informáticas para resolver eficientemente los diferentes problemas que se les fueran presentando. Esto supuso una motivación extra por parte de los alumnos ya que pasaban de una posición pasiva dentro de la clase a ser un sujeto activo.

El paquete de Shiny incluye 11 ejemplos que facilitan al programador que se familiarice con el paquete de R y de esta manera se sienta seguro para convertirse en creador de aplicaciones Shiny de forma amigable. No se requieren grandes conocimientos como programador y cualquier alumno se ve capaz de llegar a crear por sí solo.

Cualquier aplicación Shiny consta de tres bloques de código: ui; Server; y la línea de comando que ejecuta la aplicación. El lugar de ubicación del código depende de la complejidad de la aplicación que se crea; si es sencilla, se registra todo el código de la aplicación en una sola “pestaña” y si es compleja, se utilizan dos pestañas, una para cada uno de los dos primeros componentes (ui y Server), no siendo necesario programar el comando de ejecución.

Tanto el primero como el segundo de los ejemplos que propone Shiny fueron presentados a los alumnos en el momento en el que se les introduce qué es Shiny. El

---

<sup>2</sup> Shiny es un paquete de R que facilita la programación de páginas web interactivas desde R-Studio. Se pueden hospedar aplicaciones en una página web o tener un documento R Markdown. También se pueden ampliar las aplicaciones de Shiny con temas de CSS, widgets de HTML o acciones de Java para el propósito que se quiera perseguir. Por lo tanto, Shiny combina el poder de la programación de R con la modernidad de la interactividad de la Web.

primero de ellos propone la creación de un histograma a partir de unos datos ya introducidos en el paquete del ejemplo y permite que el alumno interactúe con número de intervalos en los que se agrupan los datos iniciales, colores, títulos, etc.

En el material que se preparó para los alumnos, se modificó el código de la aplicación que ofrece Shiny para que los alumnos tuvieran claras las dos partes fundamentales de toda app de Shiny, así como la línea de comando de ejecución del mismo. El código final con el que trabajaron los alumnos inicialmente se muestra en la Figura 1.

Figura 1. Código modificado.

```
library(shiny)

#####UI#####
# Define UI
ui <- fluidPage(

  # Application title
  titlePanel("Example 1. Histogram"),

  # Sidebar
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 10,
                  value = 6), submitButton("Update View")
    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("distPlot")
    )
  )
)

#####Server#####

# Define server
server <- function(input, output) {

  output$distPlot <- renderPlot({

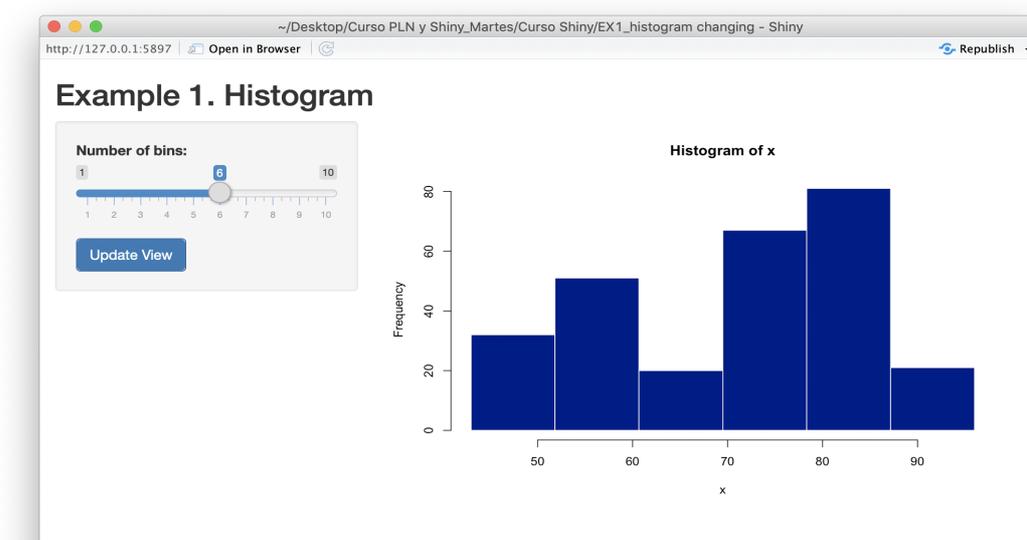
    # generate bins based on input$bins from ui.R
    x <- faithful[, 2]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(x, breaks = bins, col = 'darkblue', border = 'white')
  })
}

#####Run#####
# Run the application
shinyApp(ui = ui, server = server)
```

Cómo se puede ver en el mismo las partes están diferenciadas, precedidas y sucedidas de 6 símbolos de almohadilla. El resultado fue que los alumnos no solo vieron el diseño del histograma tal y como aparece en la Figura 2 **Error! Reference source not found.**, sino que además de interactuar con ella y cambiar el número de intervalos en la barra de la parte de la izquierda pudieron ver como incorporando o no el botón de actualizar el histograma quedaba claro que la persona que interactuaba debía apretar el botón para actualizar los resultados o bien que salía automáticamente el resultado cuando movía la barra para determinar el número de intervalos en los que quedaban representados los datos en el histograma. Además, a su vez, cambiaron algunas otras cuestiones en el código tales como el título, el color etc.

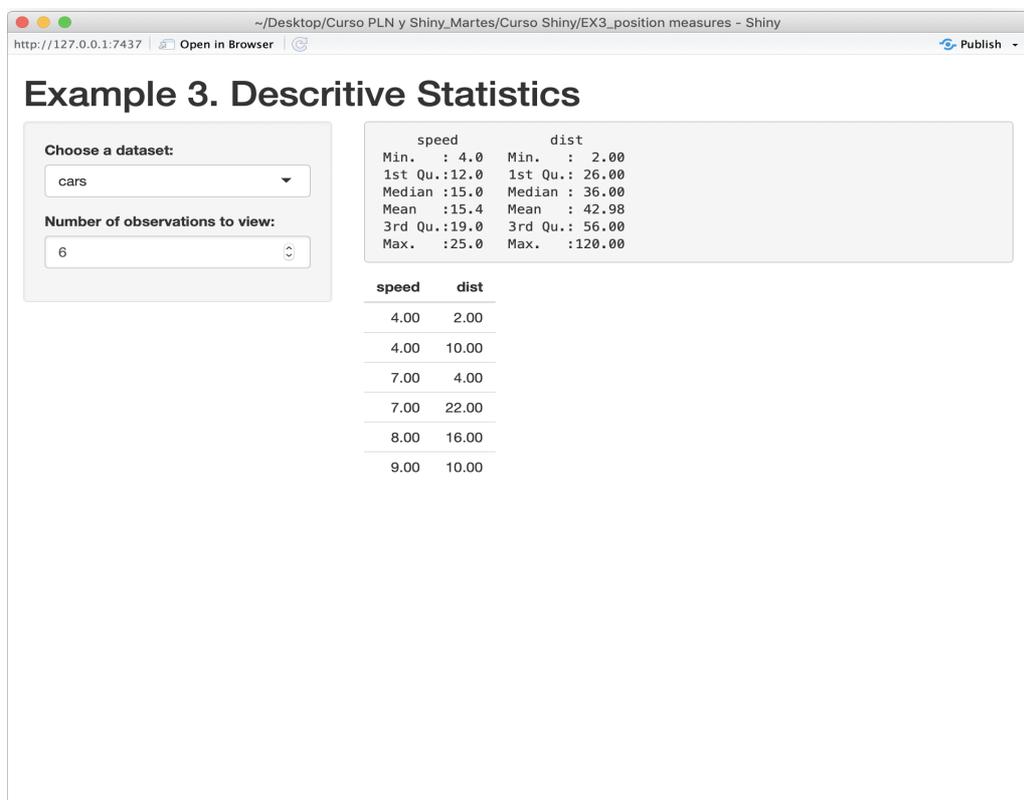
*Figura 2. Aplicación de ejemplo integrada en el paquete Shiny*



Una vez introducida la primera aplicación, se continuó con la introducción de una segunda aplicación que consiste en ver la manera de que el alumno puede introducir distintas bases de datos en una aplicación que le mostrase las medidas de posición los cuartiles la media y la mediana etc. (Figura 3).

En esta aplicación pudieron calcular rápidamente el resultado de los ejemplos que tenían en papel. Las medidas de posición de las distintas bases de datos que tenían los alumnos por equipos. (Figura3)

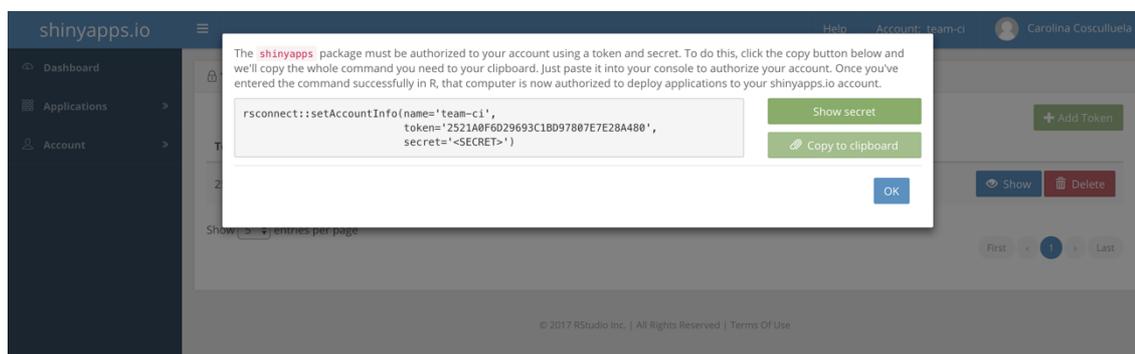
Figura 3. Estadísticos Descriptivos



Toda vez que ya sabían retocar código y habían interactuado con las distintas aplicaciones se procede a enseñarles cómo subir sus propias aplicaciones al repositorio que tiene Shiny a una página web específica para que la pudiesen compartir.

Para ello se les enseña a abrir una cuenta en el repositorio [shiny.io](http://www.shinyapps.io) de forma gratuita en el siguiente enlace que se les presenta <http://www.shinyapps.io>.

A los alumnos se les señala que cuando acceden a la cuenta deben copiar el contenido de la pestaña **tokens** mostrando previamente la información allí que aparece oculta presionando sobre el botón **Show secret**. Posteriormente se sustituye la línea de código con la información que se nos muestra, esto es, el código que permite subir nuestra app al nuestro propio repositorio tal y como aparece en la Ilustración 1 y en el código posterior.



*Ilustración 1 Información para el Deployment*

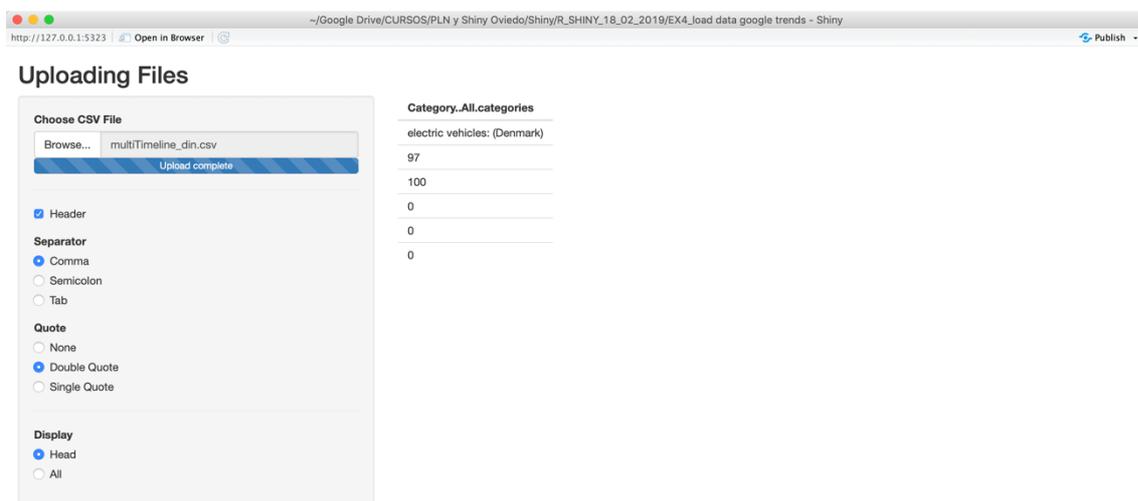
```
##Packages##
``{r package}
#install.packages('rsconnect')
...

## Show secret KEY ##
``{r take the your personal key}
library(rsconnect)
rsconnect::setAccountInfo(name='team-ci',
                           token='xxxxxxxxxxxxx',
                           secret='xxxxxxxxxxx')
...

##deploy##
``{r deploy}
#rsconnect::deployApp(appDir = getwd(), 'app1', appName = "app1")
...

```

La siguiente aplicación que se les mostró resultó ser la más interesante en su opinión. Primeramente se les indicó que buscaran la información que quisieran en la dirección de Google Trends a la que accedían a través del buscador de Google pero cuyo enlace se les facilitó también: <https://trends.google.com/trends/explore?date=all&geo=US&q=Electric%20vehicles>. Tras realizar la búsqueda de aquellos datos que estaban interesados en analizar, procedieron a realizar una app que pudiera leerlos datos usando una que estaba previamente creada e incorporada en el paquete de Shiny, tal y como aparece en la siguiente pantalla.



*Ilustración 2 Visualización de los datos extraídos de Google Trends*

El código que han utilizado para ello es el siguiente

```
library(shiny)
#####UI#####
# Define UI for data upload app ----
ui <- fluidPage(
  # App title ----
  titlePanel("Uploading Files"),
  # Sidebar layout with input and output definitions ----
  sidebarLayout(
    # Sidebar panel for inputs ----
    sidebarPanel(
      # Input: Select a file ----
      fileInput("file1", "Choose CSV File",
        multiple = FALSE,
        accept = c("text/csv",
          "text/comma-separated-values,text/plain",
          ".csv")),
      # Horizontal line ----
      tags$hr(),

      # Input: Checkbox if file has header ----
      checkboxInput("header", "Header", TRUE),
      # Input: Select separator ----
      radioButtons("sep", "Separator",
        choices = c(Comma = ",",
          Semicolon = ";",
          Tab = "\t"),
        selected = ","),
      # Input: Select quotes ----
      radioButtons("quote", "Quote",
        choices = c(None = "",
          "Double Quote" = "\"",
          "Single Quote" = "'"),
        selected = ""),
      # Horizontal line ----
      tags$hr(),
      # Input: Select number of rows to display ----
      radioButtons("disp", "Display",
        choices = c(Head = "head",
          All = "all"),
        selected = "head" ),
      # Main panel for displaying outputs ----
      mainPanel(

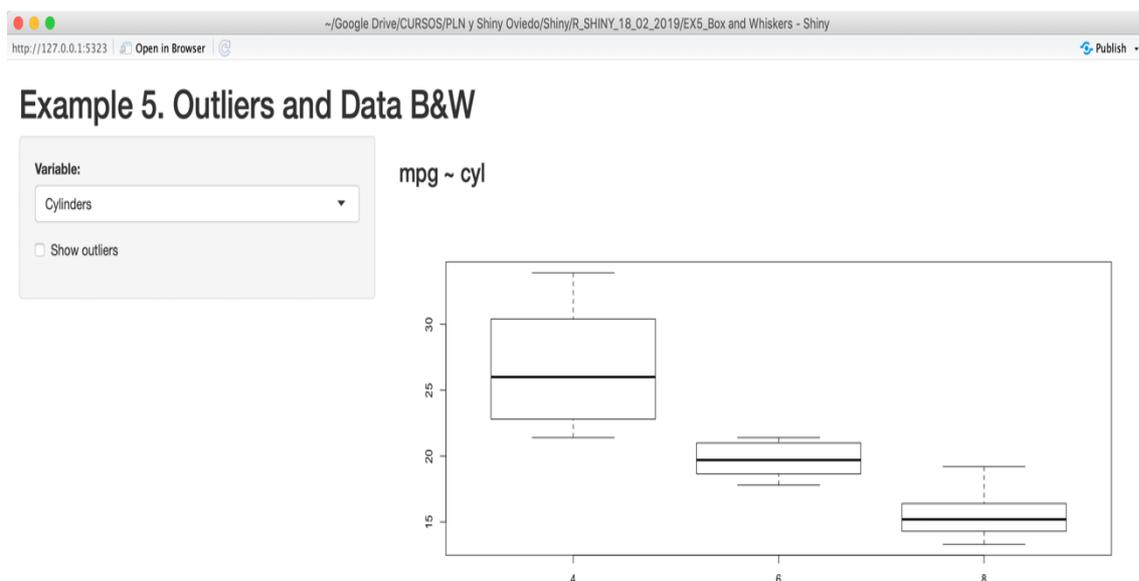
        # Output: Data file ----
        tableOutput("contents"))))
# Define server logic to read selected file ----
server <- function(input, output) {

  output$contents <- renderTable({
    # input$file1 will be NULL initially. After the user selects
    # and uploads a file, head of that data file by default,
    # or all rows if selected, will be shown.
    req(input$file1)
    # when reading semicolon separated files,
    # having a comma separator causes `read.csv` to error
    tryCatch(
      {
        df <- read.csv(input$file1$datapath,
          header = input$header,
          sep = input$sep,
          quote = input$quote)
      },
      error = function(e) {
        # return a safeError if a parsing error occurs
        stop(safeError(e)) } )

    if(input$disp == "head") {
      return(head(df))
    }
    else {
      return(df) }})

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

La siguiente aplicación que se les presentó tenía como objetivo realizar diagramas de cajas y bigotes y con los distintos datos podían entender como cambiaban los grafos. El código se muestra en el anexo y forma parte de los ejemplos del propio paquete<sup>3</sup>.



*Ilustración 3 App para explicar los gráficos de Cajas y Bigotes*

A continuación, se les mostró la aplicación para proyectar en un mapa los datos en los que estaban interesados. Un grupo se bajó datos de los condados de EEUU para proyectar dinámicamente las variables que seleccionaron.

---

<sup>3</sup> Algunos códigos han sido parcialmente modificados

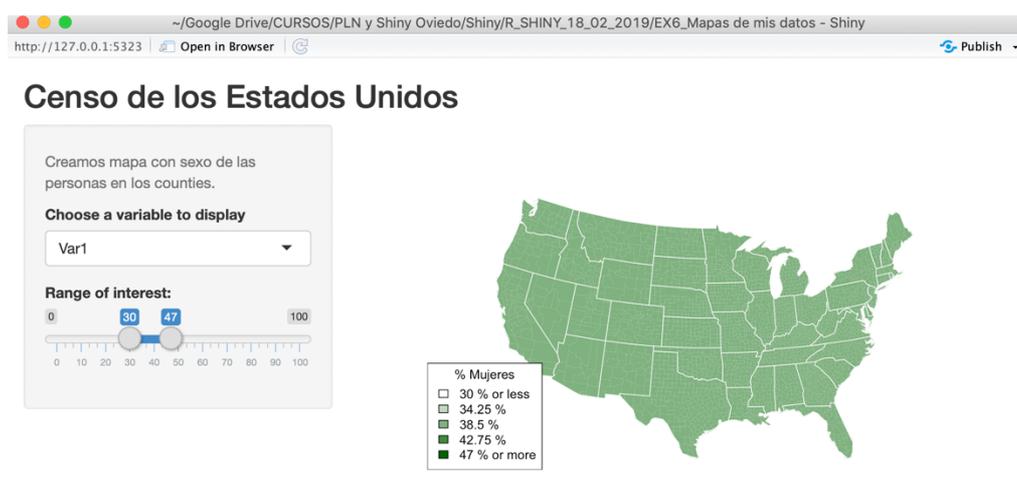


Ilustración 4 Mapeo de Datos

Por último, en este primer curso se les mostró la App de cluster K medias aunque no entraba dentro de la materia a impartir para explicar para qué sirve y que interactuaran con el número de clusters y con las variables a clasificar.

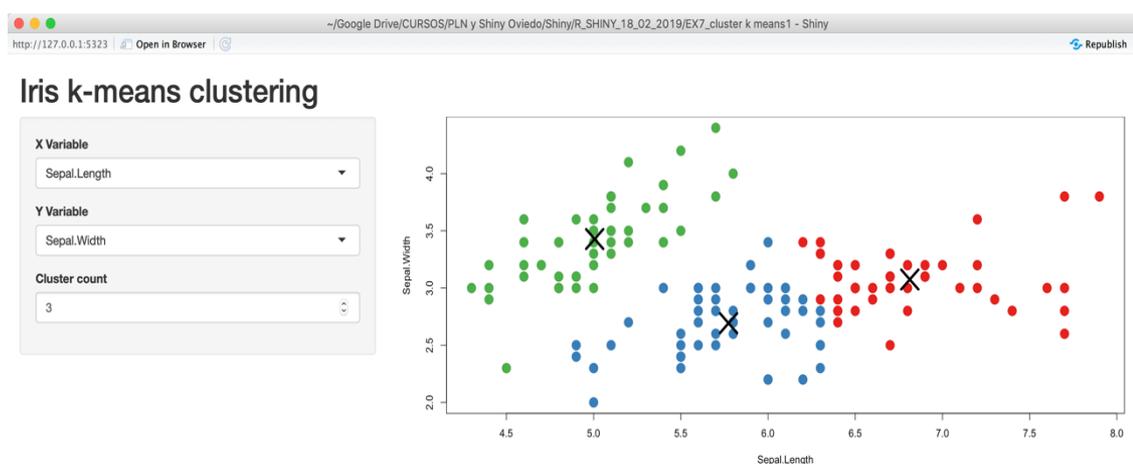


Ilustración 5 Proyección de Cluster K-means

Posteriormente se les mostró la inmensa cantidad de aplicaciones existente en el repositorio de Shiny.

El propósito de introducir primero algunos ejemplos e interactuar el código de la aplicación fue que los propios alumnos identificasen sus necesidades, lo pudieran

encontrar por sí mismos y que supieran interactuar con la aplicación que les facilitase el aprendizaje en cada momento, modificando código si fuese necesario para que pudieran practicar con gran cantidad de ejemplos los tipos de ejercicios que pudieran ser materia de examen.

Galería de Aplicaciones: <https://shiny.rstudio.com/gallery/>

Galería de Aplicaciones para la enseñanza y muchas más:  
<http://shiny.rstudio.com/gallery/see-more.html>

Tipos:

1. Todo tipo de análisis estadísticos: <https://afazali.shinyapps.io/afazali5/>
2. Introducción a la estadística: <http://www.intro-stats.com/>
3. Diferencia entre tipos de diagramas de cajas  
<https://gallery.shinyapps.io/boxplot/>
4. Comportamiento del muestreo:  
[https://gallery.shinyapps.io/sampling\\_and\\_stderr/](https://gallery.shinyapps.io/sampling_and_stderr/)
5. Juego de la correlación desde la representación gráfica:  
[https://gallery.shinyapps.io/correlation\\_game/](https://gallery.shinyapps.io/correlation_game/)
6. Regresión lineal simple: [https://gallery.shinyapps.io/simple\\_regression/](https://gallery.shinyapps.io/simple_regression/)
7. Regresión lineal Bootstrap: [https://gallery.shinyapps.io/regression\\_bootstrap/](https://gallery.shinyapps.io/regression_bootstrap/)
8. Regresión lineal múltiple: [https://gallery.shinyapps.io/multi\\_regression/](https://gallery.shinyapps.io/multi_regression/)

Otro tipo de aplicaciones más avanzadas:

1. ANOVA: [https://gallery.shinyapps.io/anova\\_shiny\\_rstudio/](https://gallery.shinyapps.io/anova_shiny_rstudio/)
2. [-catalog/#fig04-17\\_visitors-at-st-louis-science-center](https://gallery.shinyapps.io/catalog/#fig04-17_visitors-at-st-louis-science-center)
3. Teorema central del límite para la media:  
[https://gallery.shinyapps.io/CLT\\_mean/](https://gallery.shinyapps.io/CLT_mean/)
4. Teorema central del límite para la proporción:  
[https://gallery.shinyapps.io/CLT\\_prop/](https://gallery.shinyapps.io/CLT_prop/)
5. Cálculo de probabilidades en función de la distribución seguida  
[https://gallery.shinyapps.io/dist\\_calc/](https://gallery.shinyapps.io/dist_calc/)
6. Link entre la distribución de los z-values y p-values:  
<http://shinyapps.org/showapp.php?app=https://telmi.psy.lmu.de/felix/z-to-p/&by=Sven%20Hilbert&title=From%20z%20to%20p:%20A%20visualization%20of%20the%20link%20between%20the%20two%20distributions&shorttitle=From%20z%20to%20p>

7. Simulaciones de cluster *k*-means: <https://gallery.shinyapps.io/kcompshiny/>

Y algunas otras de tipo herramienta para gestionar datos:

1. Obtención de cualquier gráfico en R con tus datos: <http://shinyapps.stat.ubc.ca/r-graph>
2. Representación de árboles pero es el código: <https://gallery.shinyapps.io/>
3. Conversor de archivos a distintos formatos <http://shiny.rstudio.com/gallery/see-more.html> (buscar file converter)
4. Demostración de que el *p*-valor no es el mejor: <https://www.shinyapps.org/apps/vs-mpr/>

### 3. RESULTADOS

Los alumnos de *Statistics I* reciben la asignatura en primero de carrera y están acostumbrados a utilizar material tradicional, papel, ordenador a nivel usuario, transparencias o incluso tablet, pero la introducción de Shiny es una novedad para ellos por lo que fue determinante el empezar explicando al modo tradicional para luego introducir las Apps. En un primer momento, hubo resistencia por parte del alumnado que, a pesar de haber sido advertido, no había traído su propio ordenador con lo cual estaba viendo todas las interacciones en el ordenador del profesor o en el del compañero y no tenía la oportunidad de realizar su propia interacción.

Cuando se introdujo Shiny y se explicó en qué consiste, muchos de los alumnos miraron con cara de escépticos. Después de mostrar el funcionamiento de la primera aplicación, sin haber cambiado nada, interactuaron enseguida con el número de intervalos que debía contener el histograma.

El uso de una única aplicación tan sencilla como ésta, fue captando poco a poco la atención de todos los alumnos. La idea de que con una sola aplicación iban a poder poner en práctica todos los conocimientos aprendidos en la asignatura fue determinante para que tuviera éxito el uso de Shiny en el aula.

La aptitud fue cambiando rápidamente y después de haber visto las medidas de posición, los alumnos mostraban inquietud en clase por resolver los ejercicios a mano y comprobar el resultado con la aplicación correspondiente en Shiny. Estuvieron satisfechos de lo que habían aprendido.

El siguiente paso, la apertura de una cuenta propia y su registro en la web para poder colgar sus propias aplicaciones, supuso un reto importante ya que hubo algún problema como por ejemplo que no a todo el mundo le funcionó el *deployment*. Este punto fue subsanado ofreciendo la alternativa de compartir la aplicación usando un pendrive o mediante correo electrónico.

Finalmente, cuando se incorporaron el resto de las aplicaciones, el sentimiento generalizado fue diverso. Una minoría de los alumnos dejó de venir a clase, aunque no se puede vincular este hecho con el uso de Shiny, ya que el porcentaje de absentismo coincide con el sucedido años anteriores en las mismas fechas. Sin embargo, se identificó un grupo de alumnos que comenzaron a asistir a clase debido al interés por aprender a utilizar la aplicación. Se percibió motivación por el aprendizaje con ordenador y un grupo de alumnos tuvieron la iniciativa de apuntarse a un curso de programación en lenguaje R fuera de la universidad.

#### 4. CONCLUSIONES

Los alumnos en general son reacios a la introducción de nuevas tecnologías dentro del aula para aprender en asignaturas cuantitativas. Con esta experiencia hemos comprobado que Shiny fue una excepción y tuvo gran acogida.

La primera conclusión que podemos extraer es que en general en casi todo el alumnado matriculado en la asignatura de *Statistics I*, la implantación de Shiny tuvo una aceptación cuasi inmediata. La parte más positiva que conviene soslayar es que entre el

alumnado no muy atraído por la ciencia matemática, y mucho menos por la estadística, tuvo mayor aceptación que cualquier otra metodología implantada en el aula.

## 5. REFERENCIAS BIBLIOGRÁFICAS

1. Página principal de Shiny: <https://shiny.rstudio.com/>
2. Galería de Aplicaciones: <https://shiny.rstudio.com/gallery/>
3. Galería de Aplicaciones para la enseñanza y muchas más: <http://shiny.rstudio.com/gallery/see-more.html>
4. Galería de análisis estadísticos: <https://afazali.shinyapps.io/afazali5/>
5. Aplicaciones para la Introducción a la estadística: <http://www.intro-stats.com/>
6. Aplicación para la Diferencia entre tipos de diagramas de cajas <https://gallery.shinyapps.io/boxplot/>
7. Aplicación para el Comportamiento del muestreo: [https://gallery.shinyapps.io/sampling\\_and\\_stderr/](https://gallery.shinyapps.io/sampling_and_stderr/)
8. Juego de la correlación desde la representación gráfica: [https://gallery.shinyapps.io/correlation\\_game/](https://gallery.shinyapps.io/correlation_game/)
9. Regresión lineal simple: [https://gallery.shinyapps.io/simple\\_regression/](https://gallery.shinyapps.io/simple_regression/)
10. Regresión lineal Bootstrap: [https://gallery.shinyapps.io/regression\\_bootstrap/](https://gallery.shinyapps.io/regression_bootstrap/)
11. Regresión lineal múltiple: [https://gallery.shinyapps.io/multi\\_regression/](https://gallery.shinyapps.io/multi_regression/)
12. ANOVA: [https://gallery.shinyapps.io/anova\\_shiny\\_rstudio/-catalog/#fig04-17\\_visitors-at-st-louis-science-center](https://gallery.shinyapps.io/anova_shiny_rstudio/-catalog/#fig04-17_visitors-at-st-louis-science-center)
13. Teorema central del límite para la media: [https://gallery.shinyapps.io/CLT\\_mean/](https://gallery.shinyapps.io/CLT_mean/)
14. Teorema central del límite para la proporción: [https://gallery.shinyapps.io/CLT\\_prop/](https://gallery.shinyapps.io/CLT_prop/)
15. Cálculo de probabilidades en función de la distribución seguida [https://gallery.shinyapps.io/dist\\_calc/](https://gallery.shinyapps.io/dist_calc/)
16. Link entre la distribución de los z-values y p-values: <http://shinyapps.org/showapp.php?app=https://telli.psy.lmu.de/felix/z-to-p/&by=Sven%20Hilbert&title=From%20z%20to%20p.%20A%20visualization%20of%20the%20link%20between%20the%20two%20distributions&shorttitle=From%20z%20to%20p>
17. Simulaciones de cluster k-means: <https://gallery.shinyapps.io/kcompshiny/>
18. Obtención de cualquier gráfico en R con tus datos: <http://shinyapps.stat.ubc.ca/r-graph>
19. Representación de árboles pero es el código: <https://gallery.shinyapps.io/>
20. Conversor de archivos a distintos formatos <http://shiny.rstudio.com/gallery/see-more.html>
21. Demostración de que el p-valor no es el mejor: <https://www.shinyapps.org/apps/vs-mpr/>

## 6 ANEXOS

### 6.1 Código para realizar la App de cajas y bigotes

```
# Load packages ----
install.packages("maps")
install.packages("mapproj")
library(shiny)
library(maps)
library(mapproj)

# Load data ----
counties <- readRDS("counties.rds")

#####UI#####
# Define UI
ui<-shinyUI(pageWithSidebar(

# Application title
headerPanel("Example 5. Outliers and Data B&W"),

# Sidebar
sidebarPanel(
  selectInput("variable", "Variable:",
    list("Cylinders" = "cyl",
         "Transmission" = "am",
         "Gears" = "gear")),

  checkboxInput("outliers", "Show outliers", FALSE)
),

# Main Panel
mainPanel(
  h3(textOutput("caption")),

  plotOutput("mpgPlot")
)
))

#####Server#####

library(shiny)
library(datasets)
mpgData <- mtcars
```

```
mpgData$am <- factor(mpgData$am, labels = c("Automatic", "Manual"))

# Define server
server<-shinyServer(function(input, output) {

  # Agrupo por mpg o por lo que quiera
  formulaText <- reactive({
    paste("mpg ~", input$variable)
  })

  output$caption <- renderText({
    formulaText()
  })

  # Generate a summary of the dataset
  output$summary <- renderPrint({
    dataset <- datasetInput()
    summary(dataset)
  })

  output$mpgPlot <- renderPlot({
    boxplot(as.formula(formulaText()),
            data = mpgData,
            outline = input$outliers)
  })
})

#####Run#####

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

## **6.2 Código para la aplicación de visualizar datos en un mapa dinámico**

```
# Load packages ----
library(shiny)
library(maps)
library(mapproj)

# Load data ----
library(readr)
counties <- read_delim("datamio.csv", ";",
                      escape_double = FALSE, skip = 1)

# Source helper functions ----
```

```
source("helpers.R")

## UI
ui <- fluidPage(
  titlePanel("Censo de los Estados Unidos"),

  sidebarLayout(
    sidebarPanel(
      helpText("Creamos mapa con sexo de las personas en los counties."),

      selectInput("var",
        label = "Choose a variable to display",
        choices = c("Var1", "Var2"),
        selected = "Var1"),

      sliderInput("range",
        label = "Range of interest:",
        min = 0, max = 100, value = c(0, 100))
    ),

    mainPanel(plotOutput("map"))
  )
)

# Server logic ----
server <- function(input, output) {
  output$map <- renderPlot({
    data <- switch(input$var,
      "Var1" = counties$Var1,
      "Var2" = counties$Var2)

    color <- switch(input$var,
      "Var1" = "darkgreen",
      "Var2" = "black")

    legend <- switch(input$var,
      "Var1" = "% Mujeres",
      "Var2" = "% Hombres")

    percent_map(data, color, legend, input$range[1], input$range[2])
  })
}

#####Run#####

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

### 6.3 Código para la aplicación de cluster

```
library(shiny)

# ui
ui<-pageWithSidebar(
  headerPanel('Iris k-means clustering'),
  sidebarPanel(
    selectInput('xcol', 'X Variable', names(iris)),
    selectInput('ycol', 'Y Variable', names(iris)),
    selected=names(iris)[[2]],
    numericInput('clusters', 'Cluster count', 3,
      min = 1, max = 9)
  ),
  mainPanel(
    plotOutput('plot1')
  )
)
```

```
)  
  
#server  
server<-function(input, output, session) {  
  
  # Combine the selected variables into a new data frame  
  selectedData <- reactive({  
    iris[, c(input$xcoll, input$ycoll)]  
  })  
  
  clusters <- reactive({  
    kmeans(selectedData(), input$clusters)  
  })  
  
  output$plot1 <- renderPlot({  
    palette(c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3",  
             "#FF7F00", "#FFFF33", "#A65628", "#F781BF", "#999999"))  
  
    par(mar = c(5.1, 4.1, 0, 1))  
    plot(selectedData(),  
          col = clusters()$cluster,  
          pch = 20, cex = 3)  
    points(clusters()$centers, pch = 4, cex = 4, lwd = 4)  
  })  
  
}  
  
# Run  
shinyApp(ui = ui, server = server)
```