

```
library(flexdashboard)
```

```
rladies_global %>%  
  filter(city %in% Madrid) %>%  
  by(Marta Vicente)
```



Flexdashboard: Easy | interactive dashboards for R

La importancia de la visualización...



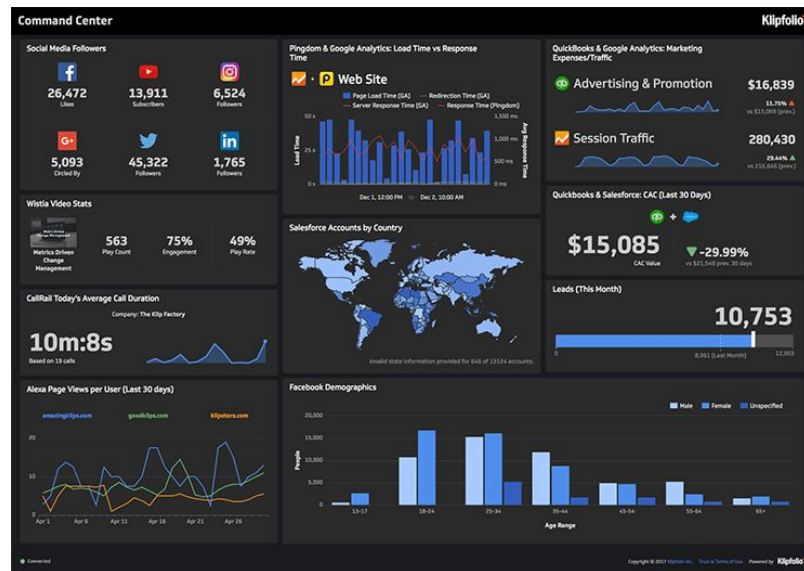
Uno de los mayores problemas con los científicos de datos es el gran muro al que se enfrentan cuando se trata de una visualización.

En la mayoría de las organizaciones, el rol de los científicos de datos no solo involucra la construcción de modelos estadísticos sofisticados, sino más bien la extracción de información valiosa de datos, cuyo resultado final es una visualización.

Los científicos de datos deben realizar un esfuerzo comunicativo: de nada sirven los descubrimientos si no se trasladan correctamente.



¿Qué es un dashboard y para qué sirve?



- ❖ El cuadro de mando o dashboard es una herramienta empleada por las empresas para analizar los datos generados, detectar problemas, patrones de comportamiento, tendencias, seguimiento modelos estadísticos, etc.
- ❖ El uso de los Dashboard se ha hecho cada vez más popular en las organizaciones y actualmente incluye todos los ámbitos de las empresas y organizaciones.
- ❖ Un dashboard debe transformar los datos en información y ésta en conocimiento para el negocio:
 - Transmitir información de manera eficiente
 - Proporcionar una interfaz de usuario intuitiva
 - Look atractivo
 - Exploración

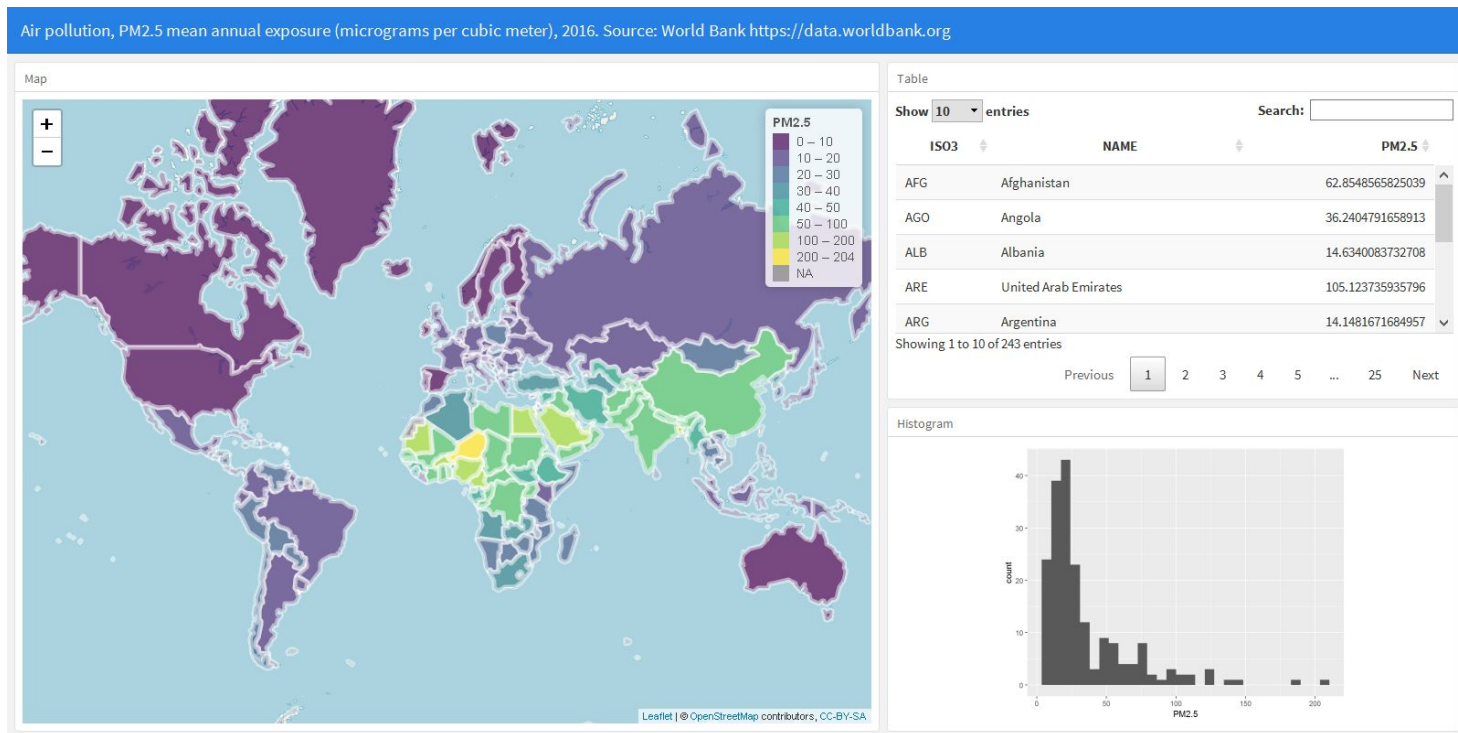
¿Qué es Flexdashboard?

<https://rmarkdown.rstudio.com/flexdashboard/index.html>



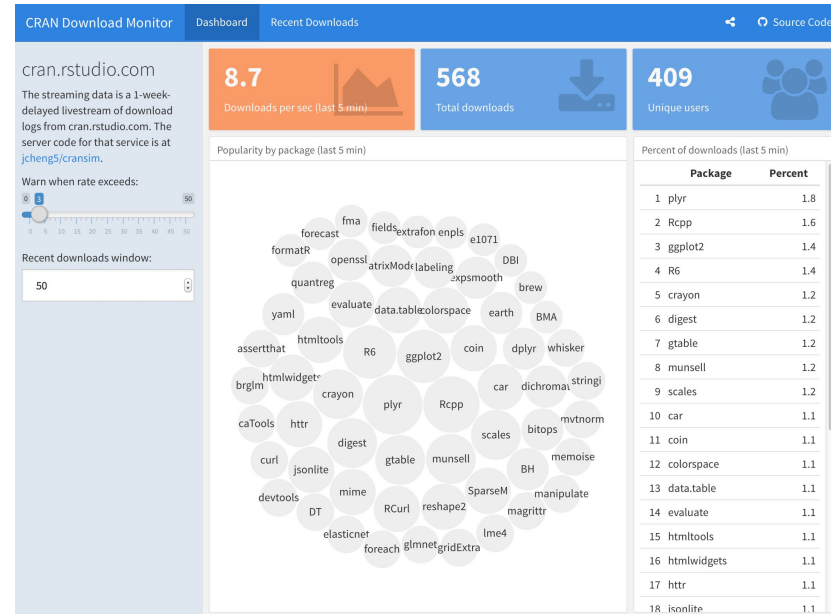
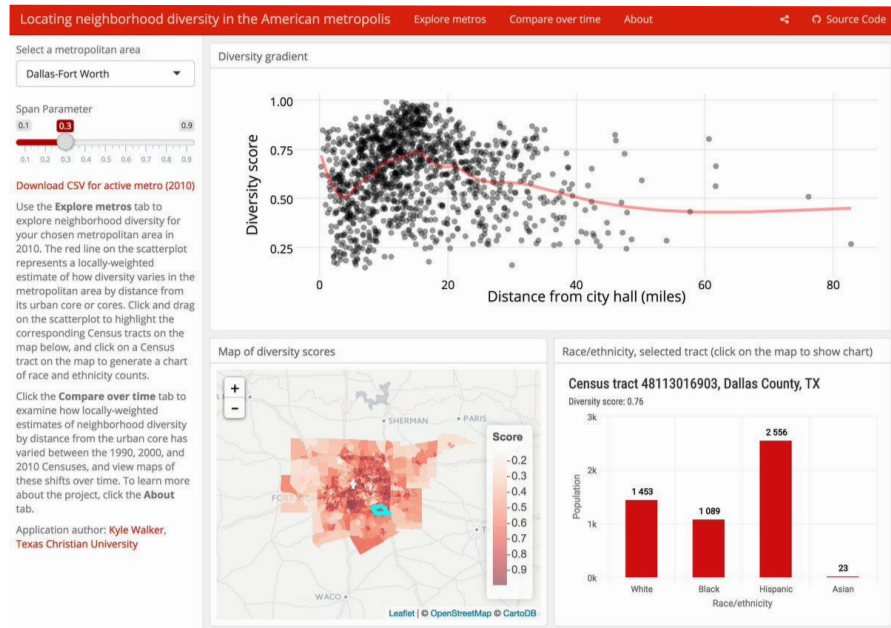
- ❖ Se trata de un **paquete o librería de R**, disponible desde mediados de 2016, que permite la generación de **dashboards** en documentos de RMarkdown, en el que se puede además añadir componentes en Shiny para que sean más interactivos.
- ❖ Permite crear cuadros de mando simples, flexibles y elegantes fácilmente.
- ❖ Puede ser **estático o dinámico** (tener componentes interactivo de [Shiny](#)).
- ❖ Los **diseños** de los cuadros de mandos están basados en **filas y columnas**. Los componentes se redimensionan de manera inteligente para llenar el navegador y se adaptan para mostrarlos en dispositivos móviles.
- ❖ Soporte de todos los **componentes gráficos existentes** en R: ggplot2, knitr, kable, html widgets(plotly, leaflet), data.table....
- ❖ Dispone de un amplio soporte para anotaciones de texto, para añadir narrativa de contexto o análisis en los paneles de control.
- ❖ Permite diseños de **storyboards** para presentar secuencias de visualizaciones y comentarios relacionados.

Ejemplos de Flexdashboard



<https://rmarkdown.rstudio.com/flexdashboard/examples.html>

Ejemplos de Flexdashboard



<https://rmarkdown.rstudio.com/flexdashboard/examples.html>



Pasos a tener en cuenta para hacer un Flexdashboard



Metodología...



1. ***Pensar lo que quiero pintar***, la estructura, qué quiero contar, qué información quiero averiguar... Aunque luego sufra modificaciones, un primer borrador del que partir es imprescindible.
2. Una vez definida la información que necesito, extraemos los datos y generamos los ***datasets***.

Metodología...



3. Comenzamos a definir la **estructura del dashbaord en R**: manipulación y edición de todos los componentes que formaran parte de nuestro cuadro de mando apoyándonos en la documentación.

4. Después de solventar bugs, errores, y una vez que hemos dejado de tirarnos de los pelos porque te ha dejado de funcionar algo que hace un rato lo hacía 🤔 ... llega el momento en que ejecutas correctamente y de manera individual cada uno de los chunks (partes del código) de tal forma que ves la luz. Ahí es cuando se ejecuta la totalidad del Markdown de tal forma que es posible ***visualizar el dashboard en su totalidad***.



¿Cómo empezar
a construir
nuestro primer
Flexdashboard?

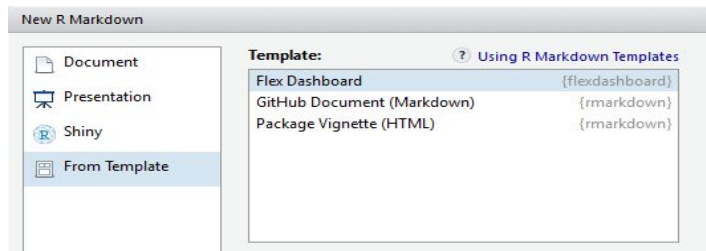
Pasos para nuestro primer Flexdashboard...



1. Install packages

```
install.packages("flexdashboard", type = "source")
```

2. New R Markdown -> From Template



3. YAML Header (just like R Markdown)

4. Specify layouts or orientations

- rows or columns
- tabset
- fill or scroll

```
---  
title: "My First Flex Dashboard"  
output:  
  flexdashboard::flex_dashboard:  
    orientation: columns  
    vertical_layout: fill  
---
```

Pasos para nuestro primer Flexdashboard...



5. Specify content using markdown headers

- ===== for pages
- ----- for columns
- #### for boxes
- *** for storyboards
- > for text notes

6. Code and Content

- Text after **###** is title of the box
- Code inside R chunk produces output in that box
- Text outside code chunks appears as plain text

```
column {data-width=650}
```

```
### Chart A
```

```
```{r}  
Insert code for plot
```
```

```
column {data-width=350}
```

```
### Chart B
```

```
```{r}  
Insert code for plot
```
```

```
### Chart C
```

```
```{r}  
Insert code for plot|
```
```



¿Qué cosas
suelo modificar
en un
flexdashboard?



Layout: Filas, Columnas and Tabsets



- Especifica si deseas un diseño de columna o un diseño de fila utilizando la sección de encabezado YAML y la etiqueta de orientación
- Especifica si desea rellenar o desplazar el diseño.
 - Todas las visualizaciones encajan en la pantalla con relleno (preferible para menos visualizaciones).
 - Todas las visualizaciones ocupan alturas naturales y tienen navegadores cuando se necesita espacio.

```
1 |---
2 title: "Single Column (Fill)"
3 output:
4   flexdashboard::flex_dashboard:
5     vertical_layout: fill
6 |---
7
8 ### Chart 1
9
10 ```{r}
11 ```
12
13
14 ### Chart 2
15
16 ```{r}
17 ```
18
19
20
21
22
23
24
25
26
```

```
1 |---
2 title: "Multiple Columns"
3 output: flexdashboard::flex_dashboard
4 |---
5
6 Column {data-width=600}
7 |-----
8
9 ### Chart 1
10
11 ```{r}
12 ```
13
14
15 Column {data-width=400}
16 |-----
17
18 ### Chart 2
19
20 ```{r}
21 ```
22
23
24 ### Chart 3
25
26 ```{r}
27 ```
28
29
```

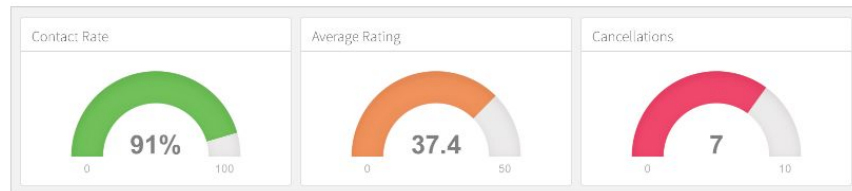
```
1 |---
2 title: "Tabset Column"
3 output: flexdashboard::flex_dashboard
4 |---
5
6 Column
7 |-----
8
9 ### Chart 1
10
11 ```{r}
12 ```
13
14 Column {.tabset}
15 |-----
16
17 ### Chart 2
18
19 ```{r}
20 ```
21
22 ### Chart 3
23
24 ```{r}
25 ```
26
```

Gráficos R y componentes



- Se puede utilizar cualquier gráfico creado con gráficos R: base, celosía, ggplot2, etc.
- La salida tabular también se puede mostrar usando `knitr :: kable ()`
- Al crear documentos interactivos (usando Shiny), los gráficos pueden cambiar dinámicamente usando `renderPlot ()` y las tablas pueden cambiar dinámicamente usando `renderTable ()`
- Los cuadros de valor se pueden usar para mostrar valores simples que pueden transmitir datos altamente resumidos.
- Las cajas de medidores se pueden usar para mostrar valores en un medidor cuando se representan rangos.
- Los cuadros de valores y cuadros de indicadores pueden contener enlaces `href` a otras secciones o páginas de paneles de control.

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|-------------------|------|-----|------|-----|------|-------|-------|----|----|------|------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |



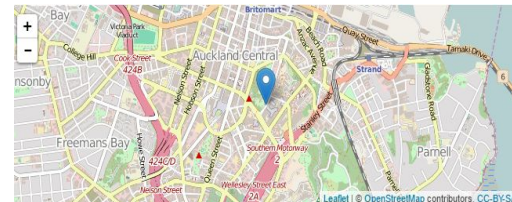
Gráficos R y componentes



HTML Widgets: El marco htmlwidgets proporciona enlaces R de alto nivel para bibliotecas de visualización de datos JavaScript. Los gráficos basados en htmlwidgets son ideales para usar con flexdashboard porque pueden redimensionarse dinámicamente, por lo que casi siempre encajarán perfectamente dentro de los límites de sus contenedores de flexdashboard.

❖ Algunos widgets disponibles:

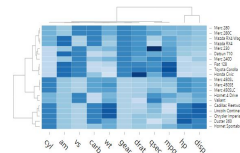
- ❖ Leaflet – <http://rstudio.github.io/leaflet/>
- ❖ Dygraphs – <http://rstudio.github.io/dygraphs>
- ❖ Plotly – <https://plot.ly/r/>
- ❖ DataTables <http://rstudio.github.io/DT/>



Show: 10 entries Search:

| | mpg | cyl | displ | hp | drat | wt | qsec | vs | am | gear | carb |
|-------------------|------|-----|-------|-----|------|-------|-------|----|----|------|------|
| Mazda RX4 | 21 | 6 | 160 | 110 | 3.9 | 2.62 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21 | 6 | 160 | 110 | 3.9 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.32 | 18.61 | 1 | 1 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.44 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225 | 105 | 2.76 | 3.46 | 20.22 | 1 | 0 | 3 | 1 |

Showing 1 to 6 of 6 entries Previous 1 Next



Dimensionamiento componentes

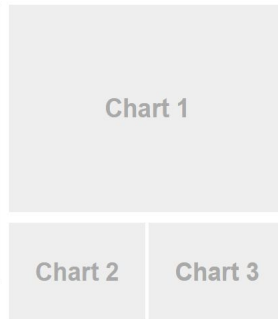


- ❖ Para crear un diseño ideal para su panel de control, es importante comprender cómo se determina el tamaño de los gráficos individuales:
- ❖ El ancho de los gráficos en flexdashboard está determinado en última instancia por el ancho del navegador.
- ❖ La altura de los gráficos se determina de una o dos formas, dependiendo de si su tablero de instrumentos está configurado para ajustarse a los límites verticales de la página (`vertical_layout: fill`) o para desplazarse (`vertical_layout: scroll`):

Focal Chart (Top)

This layout fills the page completely and gives prominence to a single chart at the top (with two secondary charts included below). To achieve this layout it uses `orientation: rows` and specifies `data-height` attributes on each row to establish their relative sizes.

```
1 ---
2 title: "Focal Chart (Top)"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: rows
6 ---
7
8 Row (data-height=650)
9 .....
10
11 ## Chart 1
12
13 {r}
14
15
16 Row (data-height=350)
17 .....
18
19 ## Chart 2
20
21 {r}
22
23
24 ## Chart 3
25
26 {r}
27
28
```

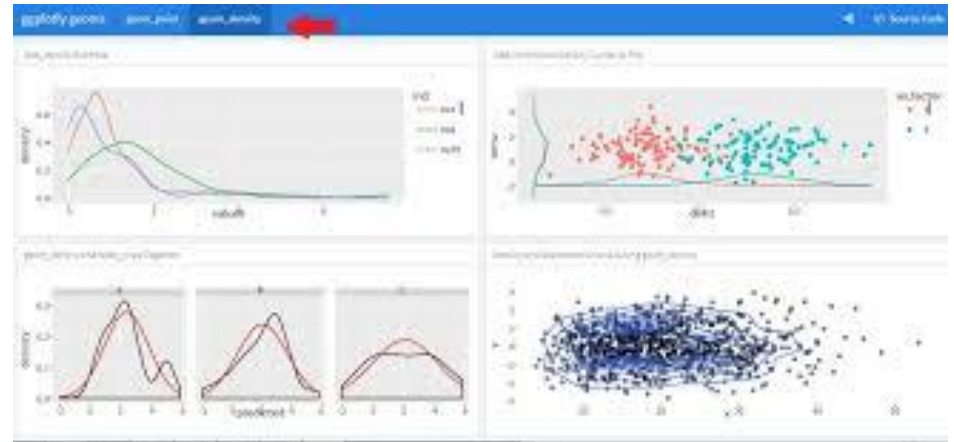


Múltiples paginas



Puedes dividir el cuadro de mando en varias páginas. Cada página que defina tendrá su propia pestaña de navegación de nivel superior.

```
1 ---
2 title: "Multiple Pages"
3 output: flexdashboard::flex_dashboard
4 ---
5
6 Page 1
7 =====
8
9 ## Chart 1
10
11 {{r}}
12
13 ## Chart 2
14
15 {{r}}
16
17
18
19 Page 2
20 =====
21
22 ## Chart 1
23
24 {{r}}
25
26
27 ## Chart 2
28
29 {{r}}
30
31
```



Storyboards



```
1 ---
2 title: "Storyboard Commentary"
3 output:
4   flexdashboard::flex_dashboard:
5     storyboard: true
6 ---
7
8 ### Frame 1
9
10 ```{r}
11 ```
12
13 ***
14
15 Some commentary about Frame 1.
16
17 ### Frame 2 {data-commentary-width=400}
18
19 ```{r}
20 ```
21
22 ***
23
24 Some commentary about Frame 2.
25
```

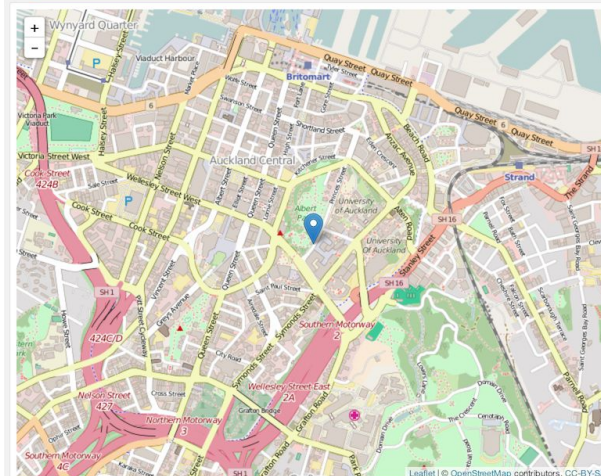
HTML Widgets Showcase

Leaflet is a JavaScript library for creating dynamic maps that support panning and zooming along with various annotations.

d3heatmap creates interactive D3 heatmaps including support for row/column highlighting and zooming.

Dygraphs provides rich facilities for charting time-series data in R and includes support for many interactive features.

Plotly provides bindings to the library and allows you to easily your ggplot2 graphics into an R web-based version.



<https://rstudio.github.io/leaflet/>

- Interactive panning/zooming
- Compose maps using arbitrary combinations of map tiles, markers, polygons, lines, popups, and GeoJSON.
- Create maps right from the R console or RStudio
- Embed maps in knitr/R Markdown documents and Shiny apps
- Easily render Spatial objects from the sp package, or data frames with latitude/longitude columns
- Use map bounds and mouse events to drive Shiny logic

Aprendiendo más...

<https://rmarkdown.rstudio.com/flexdashboard/using.html>



El sitio web de flexdashboard incluye una extensa documentación sobre cómo crear sus propios paneles de control, que incluyen:

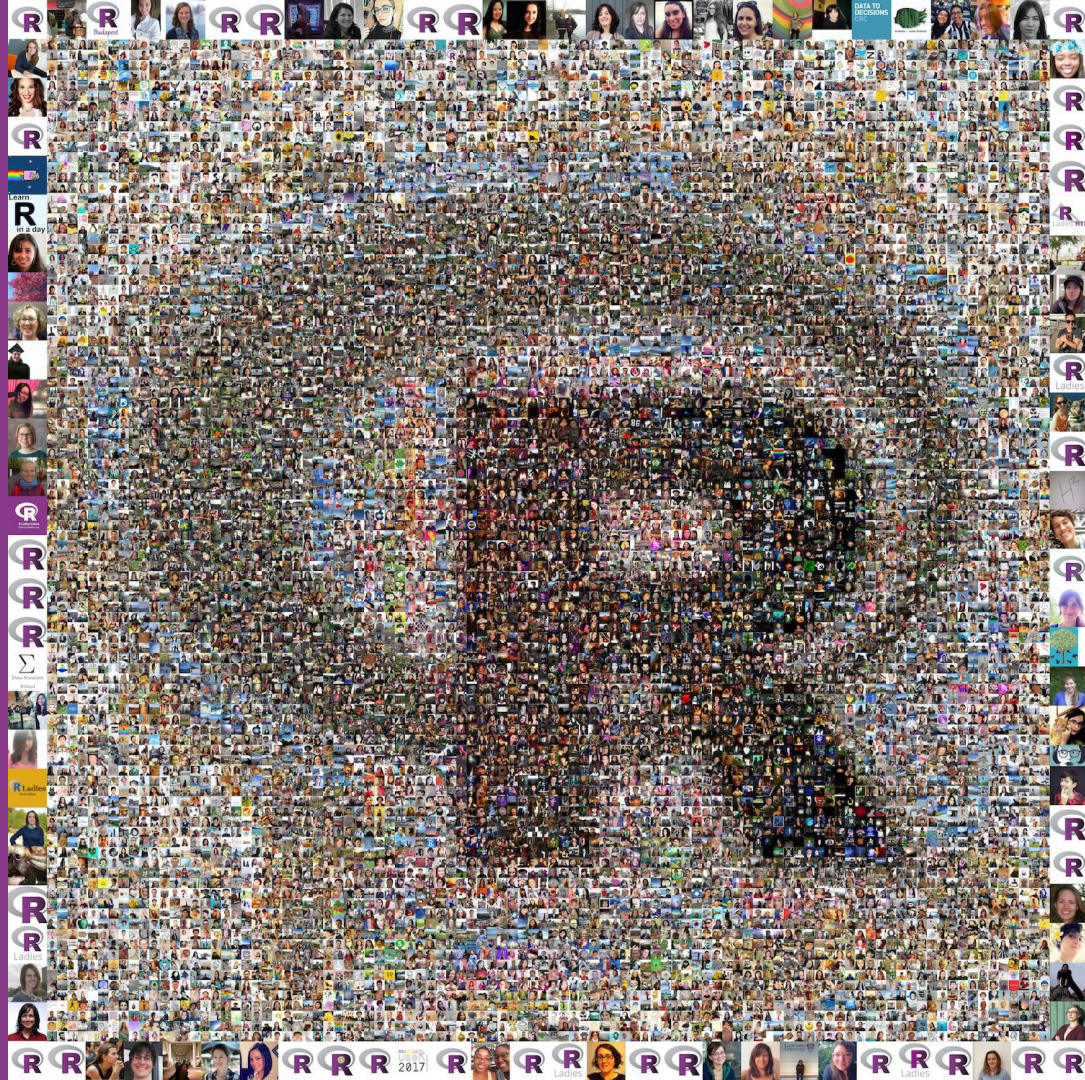
- ❖ Una guía del usuario para todas las funciones y opciones de flexdashboard, incluidas las orientaciones de diseño (basado en filas vs. columnas), el tamaño de los gráficos, los diversos componentes compatibles, la creación de temas y la creación de paneles con varias páginas.
- ❖ Los detalles sobre el uso de Shiny para crear paneles de control que permiten a los espectadores cambiar los parámetros subyacentes y ver los resultados inmediatamente, o que se actualicen de forma incremental a medida que cambian los datos subyacentes.
- ❖ Una variedad de diseños de muestra que puedes utilizar como punto de partida para tus propios paneles.
- ❖ Muchos ejemplos de flexdashboard en acción (incluidos los enlaces al código fuente si se desea profundizar en cómo se creó cada ejemplo).

Recursos...



- ❖ <https://www.r-bloggers.com/flexdashboard-easy-interactive-dashboards-for-r/>
- ❖ <https://datascienceplus.com/flexdashboard-in-r-what-why-and-how/>
- ❖ <https://rmarkdown.rstudio.com/flexdashboard/examples.html>
- ❖ CURSOS:

<https://www.datacamp.com/courses/building-dashboards-with-flexdashboard>



¡GRACIAS!



twitter.com/Martita_cdc
www.linkedin.com/in/martavicentesanchez/