**Instalación**

Conseguir una API de Datawrapper en la siguiente página:

Instalar la librería en R:

install.packages("devtools")

devtools::install\_github("munichrocker/DatawRappr")

**Pasos previos**

library(DatawRappr)

.

datawrapper\_auth(api\_key = clave) #Con ponerlo una vez vale, R se acuerda de cuál es tu api\_key

clave<-“1234567”#La clave hay que añadirla en la mayoría de comandos, así que lo más cómodo es guardarla en un objeto con un nombre fácil de utilizar frecuentemente

#Otra forma de escribir solo una vez la llave y que R la recuerde es con el siguiente comando: datawrapper\_auth(api\_key = Sys.getenv("DATAWRAPPER\_TOKEN"), overwrite = TRUE)

*Para probar que la llave funciona:*

dw\_test\_key()

**Para ver que llave estás usando**

dw\_get\_api\_key()

**Usos de la librería**

**Usar una base de datos de R para hacer un gráfico en Datawrapper sin tener que descargarlo**

dw\_data\_to\_chart()

**Crear un gráfico**

my\_chart <- dw\_create\_chart(type=”*tipo\_de\_grafico*”) #Mejor hacerlo así que utilizando directamente el comando, porque si no para editarla se necesita poner la ID del gráfico concreto.

Los códigos de los diferentes tipos de gráficos están apuntados 🡪 [AQUÍ](#Tipos_de_gráficos)

**Copiar un gráfico ya existente**

dw\_copy\_chart()

**Añadir datos a la gráfica**

dw\_data\_to\_chart(my\_dataframe\_in\_R, my\_chart)

#Lo mejor es crear una nueva base de datos solo con las variables que quieres usar, ya que el comando es como subir los datos a la pagina, y si hay variables de más salen cosas raras.

**Modificar la gáfica**

dw\_edit\_chart(“12345”,

title = "El título del gráfico",

intro = "El subtítulo",

annotate = “Texto debajo del gráfico”,

source\_name = "Nombre de la fuente de los datos",

source\_url = enlace a la fuente de los datos,

visualize=list(

“color-category”=conjunto\_colores,

"highlight-range" = conjunto\_rangos,

"text-annotations”=texto,

lines=lines,

…

))

A la hora de editar un gráfico, lo mejor es fijarse en el comando de un gráfico editado, donde se cambia lo mismo que queremos cambiar, o en los metadatos de un gráfico hecho a mano. De este modo se puede saber el formato requerido para la edición, ya sea una lista, una lista de listas o un simple texto.

**Conocer los metadatos de un gráfico**

dw\_retrieve\_chart\_metadata()

**Publicar o actualizar la gráfica**

dw\_publish\_chart(chart\_id=”12345”) #otra opción del comando es return\_urls, que si es =TRUE en la consola mostrará el enlace en el que se ha publicado el gráfico, y si es =FALSE publica el gráfico y ya.

**Guardar el gráfico en formato png, pdf o svg (los dos últimos solo para premium)**

dw\_export\_chart()

**Borrar un gráfico**

dw\_delete\_chart()

**Sacar una lista de todos los gráficos creados**

dw\_list\_charts()

**Sacar una lista de todas las carpetas**

dw\_list\_folders()

**Ejemplos**

Gráfico

Descripción generada automáticamente

**Tipo de gráfico**: líneas 🡪 d3-lines

**Ajustes visuales:**

dw\_edit\_chart(

chart\_id = "sanA3",

visualize = list(

lines = list(

'Max' = list(color = '#09bb9f',

width = 3),

'Min' = list(color = '#1d81a2',

width = 3))))

#Es el gráfico del tutorial:

<https://developer.datawrapper.de/docs/a-beginners-guide-with-r>

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**Tipo de gráfico**: líneas 🡪 d3-lines

**Ajustes visuales:**

dw\_edit\_chart("034u5",

title = headline,

intro = "snow height in centimeters",

annotate = text\_date\_update,

source\_name = "German Meteorological Service",

source\_url = "https://opendata.dwd.de/climate\_environment/CDC/observations\_germany/climate/daily/kl/recent/")

#Es el gráfico del siguiente ejemplo: <https://github.com/munichrocker/snow_on_zugspitze>

**Información sobre la librería:**

Página de la librería:

<https://github.com/munichrocker/DatawRappr>

Tutorial:

<https://developer.datawrapper.de/docs/a-beginners-guide-with-r>

Ejemplo de un gráfico que se actualiza “solo”:

<https://github.com/munichrocker/snow_on_zugspitze>

Ejemplo de cómo hacer un beeswarm en DataWrapper:

<https://github.com/d-qn/Datawrapper-beeswarm-from-R>

(Como en DW no existe un gráfico así, sino que se solaparían los puntos y se vería peor, se hace el gráfico en R y se crea una nueva base de datos con las coordenadas de cada punto, que después se guardará en la base de datos que se usará en DW)

Mapa

Descripción generada automáticamente

Charla Kiko Llaneras:

<https://www.youtube.com/watch?v=NqY6RgzSkj0>

**[Tipos de gráficos](#Enlace_a_Tipos_de_gráficos)**

* d3-bars : Bar Chart
* d3-bars-split : Split Bars
* d3-bars-stacked : Stacked Bars
* d3-bars-bullet : Bullet Bars
* d3-dot-plot : Dot Plot
* d3-range-plot : Range Plot
* d3-arrow-plot : Arrow Plot
* column-chart : Column Chart
* grouped-column-chart : Grouped Column Chart
* stacked-column-chart : Stacked Column Chart
* d3-area : Area Chart
* d3-lines : Line Chart
* d3-pies : Pie Chart
* d3-donuts : Donut Chart
* d3-multiple-pies : Multiple Pies
* d3-multiple-donuts : Multiple Donuts
* d3-scatter-plot : Scatter Plot
* election-donut-chart : Election Donut
* tables : Table
* d3-maps-choropleth : Choropleth Map
* d3-maps-symbols : Symbol Map
* locator-map : Locator Map