

Animaciones en R

GIFS en R

Actualmente El formato GIF (graphics interchange format) es muy utilizado en el despliegue de imágenes animadas para páginas web, lo cual hace atractivo y llamativo al mensaje que se quiera transmitir. Existen en la web diversas alternativas para la creación de estos archivos, sin embargo en este caso nos centraremos en el uso del software estadístico R para la creación de GIF, para tal fin se usará el paquete `gganimate`. En el siguiente artículo se hablará sobre la estructura que debe tener la data para la creación de GIFs, así como de los diversos argumentos que se deben tener en cuenta al momento de usar las funciones de R para crear este tipo de archivos.

Data a usar

La data con la que se va a trabajar es “`gdp_tidy.csv`”. La misma nos muestra información sobre el producto interno bruto (GDP) para los años 1990, 2000, y del 2009 al 2017, de 217 países, esta data fue obtenida de la página del Banco Mundial. La misma cuenta con las siguientes columnas,

- **country_name**: esta columna contiene el nombre del país.
- **country_code**: esta variable contiene la abreviatura del país.
- **year**: esta columna contiene el año del GDP para cada país. Posee información para los años 1990, 2000, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016 y 2017.
- **value**: esta variable contiene el valor del GDP expresado en billones de dólares.

Para empezar el estudio se cargarán las librerías necesarias para el mismo,

- `tidyverse`: librería que contiene funciones que ayudan al procesamiento y agrupación de datos.
- `gganimate`: librería que permite la creación de GIFs.
- `av`: librería que permite la exportación de la animación creada a un video.

Un vistazo a la data se presenta a continuación,

```
#CARGO LIBRERIAS
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.2.1      v purrr   0.3.3
## v tibble  2.1.3      v dplyr   0.8.3
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
## Warning: package 'tibble' was built under R version 3.5.3

## Warning: package 'tidyr' was built under R version 3.5.3

## Warning: package 'readr' was built under R version 3.5.3

## Warning: package 'purrr' was built under R version 3.5.3

## Warning: package 'dplyr' was built under R version 3.5.3

## Warning: package 'stringr' was built under R version 3.5.3

## Warning: package 'forcats' was built under R version 3.5.3

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gganimate)
```

```
## Warning: package 'gganimate' was built under R version 3.5.3
```

```
library(av)
```

```
## Warning: package 'av' was built under R version 3.5.3
```

```
#CARGO DATA A USAR
```

```
gdp_tidy <- read_csv("./data/gdp_tidy.csv")
```

```
## Parsed with column specification:
## cols(
##   country_name = col_character(),
##   country_code = col_character(),
##   year = col_double(),
##   value = col_double()
## )
```

```
head(gdp_tidy)
```

```
## # A tibble: 6 x 4
##   country_name country_code year      value
##   <chr>         <chr>    <dbl>    <dbl>
## 1 Afghanistan  AFG      1990         NA
## 2 Albania      ALB      1990  2028553750
## 3 Algeria      DZA      1990 62045099643.
## 4 American Samoa ASM      1990         NA
## 5 Andorra      AND      1990  1029048482.
## 6 Angola       AGO      1990 11228764963.
```

Una vez cargada la data, se procede a darle formato esto con el fin de agrupar a los diez primeros países y crear un ranking por año, es importante crear esta estructura pues la misma será la base de la animación a crear. El proceso a seguir es el siguiente,

- 1) Agrupar la data por año: para esta tarea la función que nos ayudará será “group_by”, a la misma se le debe pasar como argumento la variable a considerar, en este caso “year”.
- 2) Agregar una variable “ranking”: para este paso se usará la función “mutate”, la cual permite agregar columnas a un dataframe, en total se agregarán tres columnas, las cuales serán obtenidas a partir de las columnas originales.
 - rank: representa un ranking por cada año para todos los países que estén en la base de datos.
 - value_rel: representa el porcentaje que representa el valor del GDP con relación al primer clasificado de un año en específico.
 - value_lbl: representa el valor de GDP expresado en billones.
- 3) Agrupar por país: en este paso se vuelve a usar la función “group_by” con el fin de agrupar las observaciones por países.
- 4) Filtrar primeros 10 países: finalmente se usa la función “filter” para filtrar las observaciones y quedarnos con sólo aquellas que estén en el top diez.

Un vistazo a la data se presenta a continuación,

```
#PROCESO DATA
gdp_formatted <- gdp_tidy %>%
  group_by(year) %>%
  # The * 1 makes it possible to have non-integer ranks while sliding
  mutate(rank = rank(-value),
         Value_rel = value/value[rank==1],
         Value_lbl = paste0(" ",round(value/1e9))) %>%
  group_by(country_name) %>%
  filter(rank <=10) %>%
  ungroup()

head(gdp_formatted)
```

```
## # A tibble: 6 x 7
##   country_name country_code year   value  rank Value_rel Value_lbl
##   <chr>         <chr>    <dbl>  <dbl> <dbl>    <dbl> <chr>
## 1 Brazil      BRA      1990 4.62e11    10    0.0773 " 462"
## 2 Canada      CAN      1990 5.94e11     7    0.0993 " 594"
## 3 France      FRA      1990 1.27e12     4    0.212  " 1269"
## 4 Germany     DEU      1990 1.76e12     3    0.295  " 1765"
## 5 Italy        ITA      1990 1.18e12     5    0.197  " 1177"
## 6 Japan       JPN      1990 3.13e12     2    0.524  " 3133"
```

Paquete gganimate

El paquete gganimate es una extensión del paquete ggplot2 diseñado para crear gráficos de ggplot animados. De esta forma podemos ver de forma dinámica en un mismo gráfico, por ejemplo, cómo la información ha ido evolucionando con el paso de los años.

Con el fin de crear la animación, primero se va a crear el gráfico base usando la función “ggplot”, donde usamos la data que hemos procesado.

```
#CREO ANIMACION
anim <- ggplot(gdp_formatted, aes(rank, group = country_name,
  fill = as.factor(country_name), color = as.factor(country_name))) +
  geom_tile(aes(y = value/2,
    height = value,
    width = 0.9), alpha = 0.8, color = NA) +
  geom_text(aes(y = 0, label = paste(country_name, " ")), vjust = 0.2, hjust = 1) +
  geom_text(aes(y=value,label = Value_lbl, hjust=0)) +
  coord_flip(clip = "off", expand = FALSE) +
  scale_y_continuous(labels = scales::comma) +
  scale_x_reverse() +
  guides(color = FALSE, fill = FALSE) +
  theme(axis.line=element_blank(),
    axis.text.x=element_blank(),
    axis.text.y=element_blank(),
    axis.ticks=element_blank(),
    axis.title.x=element_blank(),
    axis.title.y=element_blank(),
    legend.position="none",
    panel.background=element_blank(),
    panel.border=element_blank(),
    panel.grid.major=element_blank(),
    panel.grid.minor=element_blank(),
    panel.grid.major.x = element_line( size=.1, color="grey" ),
    panel.grid.minor.x = element_line( size=.1, color="grey" ),
    plot.title=element_text(size=25, hjust=0.5, face="bold", colour="grey", vjust=-1),
    plot.subtitle=element_text(size=18, hjust=0.5, face="italic", color="grey"),
    plot.caption =element_text(size=8, hjust=0.5, face="italic", color="grey"),
    plot.background=element_blank(),
    plot.margin = margin(2,2, 2, 4, "cm")) +
  transition_states(year, transition_length = 4, state_length = 1) +
  view_follow(fixed_x = TRUE) +
  labs(title = 'GDP per Year : {closest_state}',
    subtitle = "Top 10 Countries",
    caption = "GDP in Billions USD | Data Source: World Bank Data")
```

Función animate

Una vez creado el objeto “anim” usaremos la función “animate”, la cual nos va a permitir crear el GIF, los principales argumentos de la misma son,

- **plot**: objeto a ser animado, puede ser un objeto obtenido mediante la función “ggplot”.
- **nframes**: número de marcos a renderizar, por defecto el valor es 100. Mientras más grande sea este valor, la animación se hará más larga.
- **fps**: velocidad de fotogramas de la animación en fotogramas / seg.
- **width**: ancho de la animación resultante.
- **height**: alto de la animación resultante.
- **renderer**: función utilizada para representar los cuadros generados en una animación.

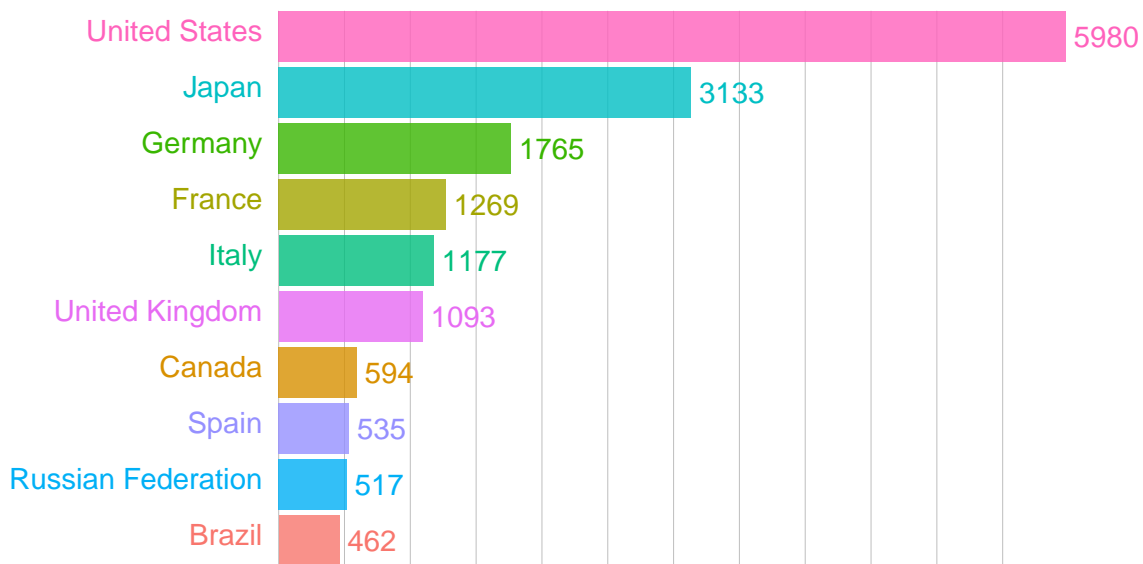
El top diez de países para el año 1990, se presenta a continuación,

```

anim1 <- ggplot(gdp_formatted[which(gdp_formatted$year=="1990"),], aes(rank, group = country_name,
                                fill = as.factor(country_name), color = as.factor(country_name))) +
  geom_tile(aes(y = value/2,
                height = value,
                width = 0.9), alpha = 0.8, color = NA) +
  geom_text(aes(y = 0, label = paste(country_name, " ")), vjust = 0.2, hjust = 1) +
  geom_text(aes(y=value,label = Value_lbl, hjust=0)) +
  coord_flip(clip = "off", expand = FALSE) +
  scale_y_continuous(labels = scales::comma) +
  scale_x_reverse() +
  guides(color = FALSE, fill = FALSE) +
  theme(axis.line=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks=element_blank(),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        legend.position="none",
        panel.background=element_blank(),
        panel.border=element_blank(),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(),
        panel.grid.major.x = element_line( size=.1, color="grey" ),
        panel.grid.minor.x = element_line( size=.1, color="grey" ),
        plot.title=element_text(size=25, hjust=0.5, face="bold", colour="grey", vjust=-1),
        plot.subtitle=element_text(size=18, hjust=0.5, face="italic", color="grey"),
        plot.caption =element_text(size=8, hjust=0.5, face="italic", color="grey"),
        plot.background=element_blank(),
        plot.margin = margin(2,2, 2, 4, "cm"))

```

anim1



Exportación a formato GIF

El siguiente comando va a permitir la creación del GIF. El mismo se puede apreciar en el siguiente [enlace](#).

```
#animate(anim, 100, fps = 20, width = 1200, height = 1000,
#         renderer = gifski_renderer("gganim.gif"))
```

Exportación a video

Con la ayuda del paquete av, el siguiente comando nos va a permitir generar un video en formato mp4. El mismo se puede descargar [aquí](#).

```
# Video output
# a <- animate(
#   anim + enter_fade() + exit_fly(y_loc = 1),
#   renderer = av_renderer()
# )
#
# #GUARDO VIDEO
# anim_save("animation1.mp4", animation = a )
```

De esta manera se ha generado un archivo GIF la cual es una manera alternativa y muy interesante de presentar información, ya que se cuenta con la característica de la animación lo cual lo hace atractivo a la vista. Para mayor comodidad los archivos necesarios para generar el GIF se encuentran en mi repositorio, para acceder haga click [aquí](#).