

ggplot2 review

Repaso ggplot2

ggplot2: figuras combinando capas

ggplot2 asocia distintas funciones a *capas* que representan distintas partes de un gráfico. En su forma más sencilla tenemos dos funciones:

```
ggplot(data = DATA) + GEOM_FUNCTION(mapping = aes(MAPPINGS))
```

En donde `DATA` es el dataframe que contiene las columnas a graficar, `GEOM_FUNCTION` representa una función para asignar una geometría (por ejemplo, `geom_point` o `geom_line`) y `MAPPINGS` representa las variables que quiero dibujar (x e y, o tal vez solo x).

En este ejemplo general y sencillo tengo cuatro elementos: el lienzo (vacío), los datos, las variables y la geometría.

Ejemplito

May the source be with you

```
library(dplyr)
data(starwars)
glimpse(starwars) # glimpse es una versión más simple de str()

# un primer plot:
ggplot(data = starwars) + geom_point(mapping = aes(height, mass))

# sin nombrar los argumentos explícitamente:
ggplot(starwars) + geom_point(aes(height, mass))

# vale quebrarlo en dos líneas, si dejo el '+' al final de la línea
ggplot(starwars) +
  geom_point(aes(height, mass))
```

Mappings

El llamado *mapping* es uno de los argumentos de las funciones de geometría y el objeto que hay que pasarle se genera con `aes()`. El mapping conecta elementos de los datos con información visual en el gráfico.

Por ejemplo, para asignar variables a ejes, en `aes()` asignaremos una columna de nuestro dataframe a los argumentos `x` y (opcionalmente) `y`: `aes(x = height, y = mass)`. Así conectamos posición en el gráfico con columnas del dataframe.

Podemos manipular otros aspectos estéticos del gráfico con `aes()`:

```
ggplot(starwars) + geom_point(aes(height, mass))
# o, mapeando elementos estéticos a variables:
ggplot(starwars) + geom_point(aes(height, mass, color=gender))
# podemos bypassar el mapping y definir un color único para la capa entera.
ggplot(starwars) + geom_point(aes(height, mass), color="blue")

ggplot(starwars) + geom_point(aes(height, mass, size=gender))
ggplot(starwars) + geom_point(aes(height, mass, shape=gender))
ggplot(starwars) + geom_line(aes(height, mass, linetype=gender))
ggplot(starwars) + geom_point(aes(height, mass, color=gender, shape=eye_color))
```

Jerarquías

Las funciones de geometría pueden llevar su propio mapping, o pueden heredar un mapping de la función `ggplot`.

```
# las siguientes dos expresiones funcionan igual
ggplot(starwars, aes(height, mass)) + geom_point()
ggplot(starwars) + geom_point(aes(height, mass))
```

Esto nos da flexibilidad, porque podemos usar distintas variables en distintas capas:

```
ggplot(starwars) +
  geom_point(aes(height, mass)) +
  geom_smooth(aes(height, birth_year))
```

Podemos inclusive combinar data.frames distintos (no olvidar poner `'data ='`):

```
ggplot() +
  geom_point(data = starwars, aes(height, mass), color = "red") +
  geom_point(data = morley, aes(Run, Speed), color = "blue")
```

Texto

Podemos poner texto de diversas maneras, por ejemplo:

```
ggplot(starwars) +
  geom_point(aes(height, mass)) +
  labs(colour = "gender", title = "Guerra de las Galaxias", subtitle = "masa en función de")

ggplot(starwars) +
  geom_point(aes(height, mass)) +
  xlab("altura") +
  ylab("masa")
```

Ejes

Podemos manipular los ejes de diversas maneras, por ejemplo:

```
ggplot(starwars) +
  geom_point(aes(height, mass)) +
  scale_x_log10() +
  scale_y_log10()

ggplot(starwars) +
  geom_point(aes(height, mass)) +
  xlim(c(0, 200)) +
  ylim(c(0, 150))
```

Podemos usar expresiones lógicas

Podemos mapear expresiones hechas con operadores lógicos a elementos estéticos. Los operadores lógicos generan una variable auxiliar categórica.

```
# color de acuerdo a si son más o menos cuarentones
ggplot(starwars) +
  geom_point(aes(height, mass, color=birth_year > 40))
# forma de acuerdo a si no nacieron en Tatooine
ggplot(starwars) +
  geom_point(aes(height, mass, shape=homeworld != "Tatooine"))
# tamaño de acuerdo al año de nacimiento
ggplot(starwars) +
  geom_point(aes(height, mass, size=birth_year))
```

Facets

Otra manera es dividir un gráfico en más de un gráfico de acuerdo a una variable. Debemos usar una **formula**, pero que por ahora podemos pensar como un tilde antes de una variable.

```
ggplot(starwars) +
  geom_point(aes(height, mass)) + facet_grid(~gender)

# pasados de peso para los distintos géneros
ggplot(starwars) +
  geom_point(aes(height, mass, color = mass/(height/100)^2 > 25)) + facet_grid(~gender) +
```

Otros tipos de gráficos

```
# histogramas
ggplot(starwars, aes(height)) + geom_histogram()
ggplot(starwars, aes(height)) + geom_histogram(bins = 50)
# densidades
ggplot(starwars, aes(height)) + geom_density()
ggplot(starwars, aes(height)) + geom_density(bw = 0.7)
# boxplots (para cada valor que quedó en gender, sobre la variable mass)
ggplot(starwars %>% filter(gender == "male" | gender == "female")) + geom_boxplot(aes(gender,
```