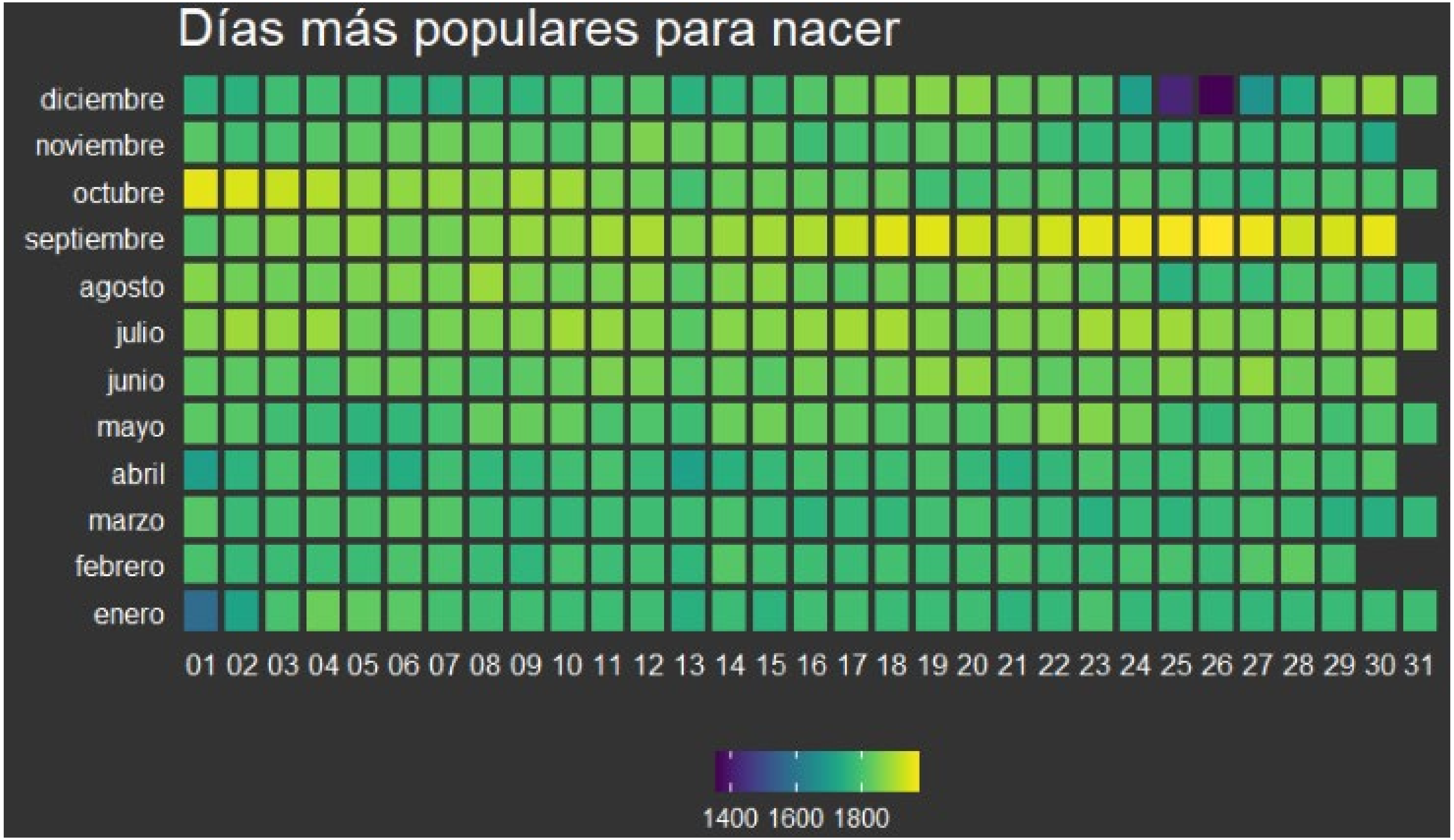
Curso R / Rstudio 2022

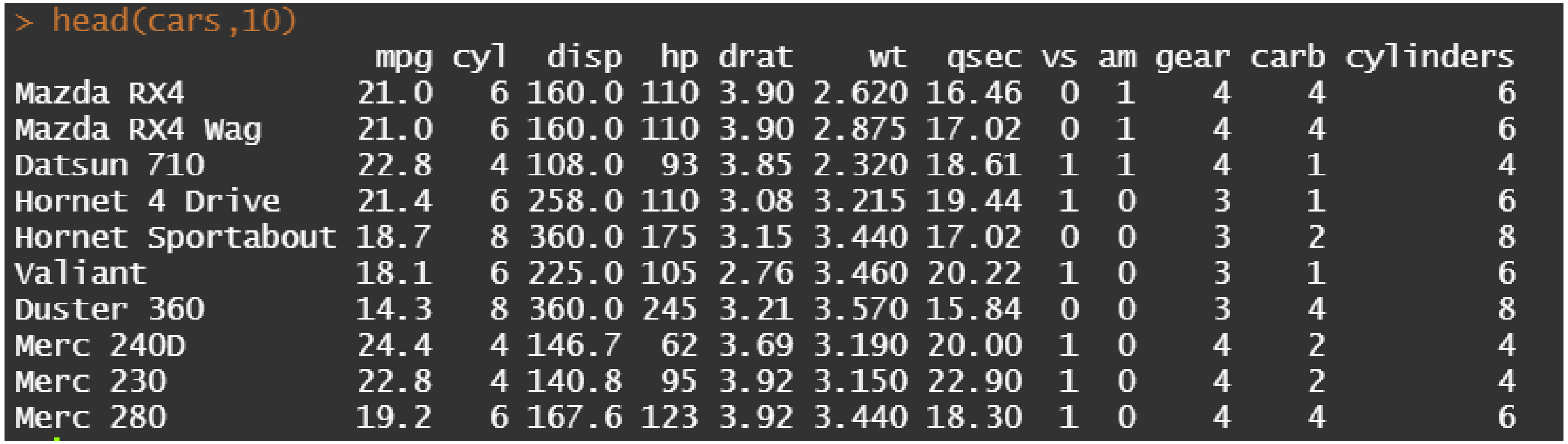
GGPLOT2


Días más populares para nacer
diciembre
noviembre
octubre
septiembre
agosto
julio
junio
mayo
abril
marzo
febrero
enero
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1400 1600 1800

Ggplot2

- One of the most attractive aspects of the R system is its capability to produce state-of-the-art statistical graphics (R by example).
- Ggplot2 (o ggplot) es un sistema organizado de visualización de datos. Forma parte del tidyverse.
- Tiene 3 components básicos:
  - Data (data frame con los datos a graficar)
  - Aesthetic mapping (relación entre las variables de la gráfica y determinados aspectos como color o la forma)
  - Geoms (capa que define el tipo de gráfica a plotear)
    - Ejemplos: geom_point, geom_line, geom_area, geom_tile

Base: mtcars



```
> head(cars,10)
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb cylinders
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4         6
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4         6
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1         4
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1         6
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2         8
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1         6
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4         8
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2         4
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2         4
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4         6
```

Plot básico

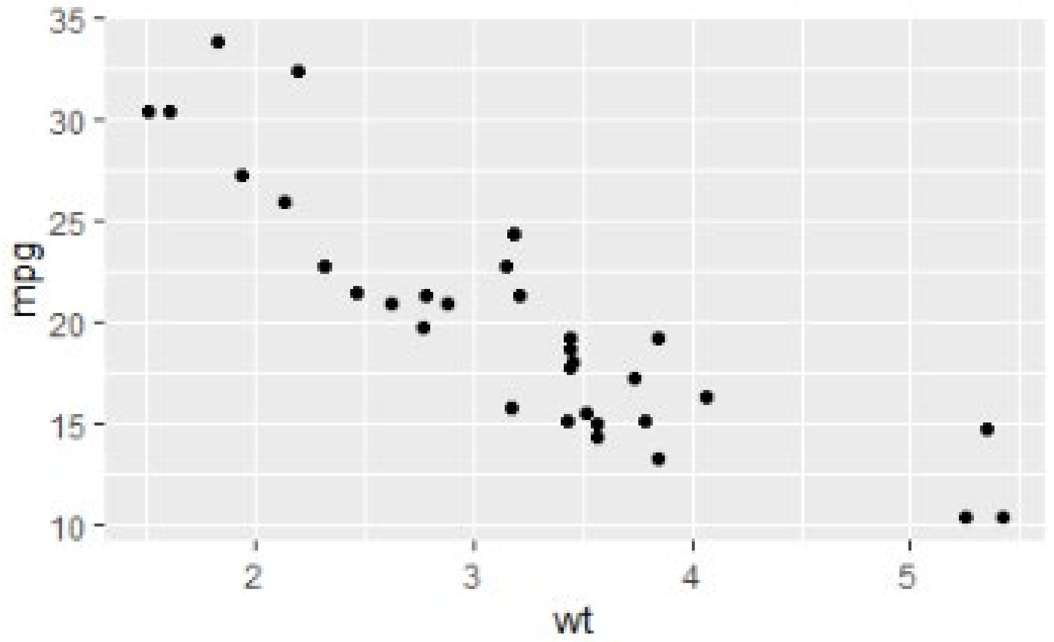
data

aesthetics

Las capas se agregan con +

```
ggplot(data = cars, aes(x = wt, y = mpg)) +
      geom_point()
```

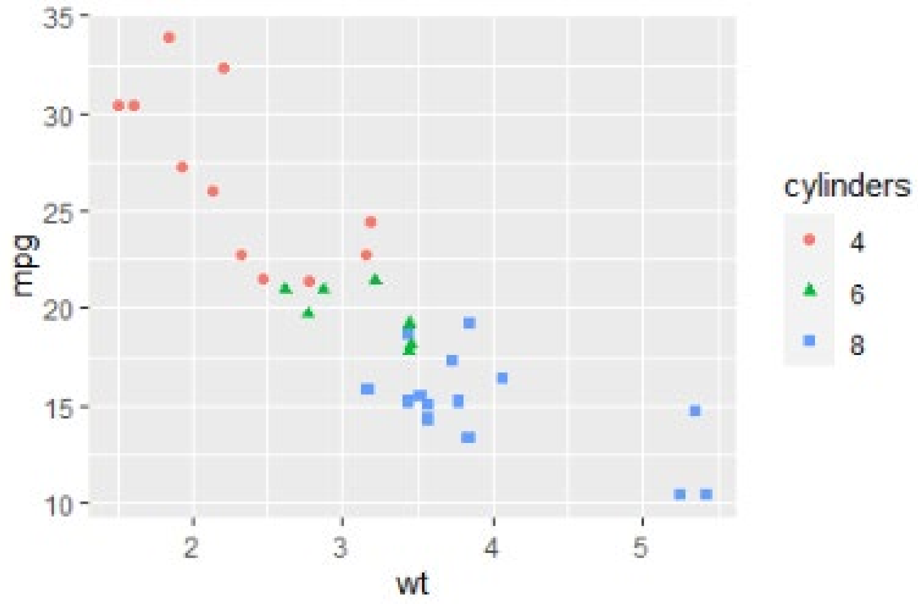
geom



Agregando aesthetics

aesthetics

```
ggplot(data = cars, aes(x = wt, y = mpg, color = cylinders, shape = cylinders)) +
    geom_point()
```

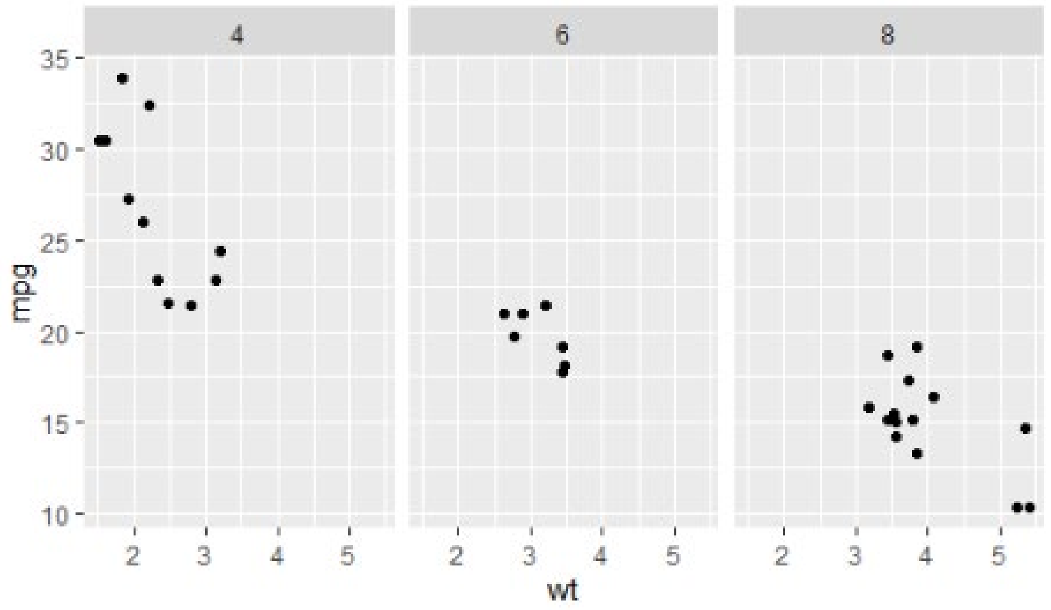


Ojo: si quieres un valor fijo para color o shape, el argumento va afuera de aesthetics.

facet_wrap

```
ggplot(data = cars, aes(x = wt, y = mpg)) +
  geom_point()+
  facet_wrap(~cylinders)
```

facet_wrap divide en varias graficas con base en una variable

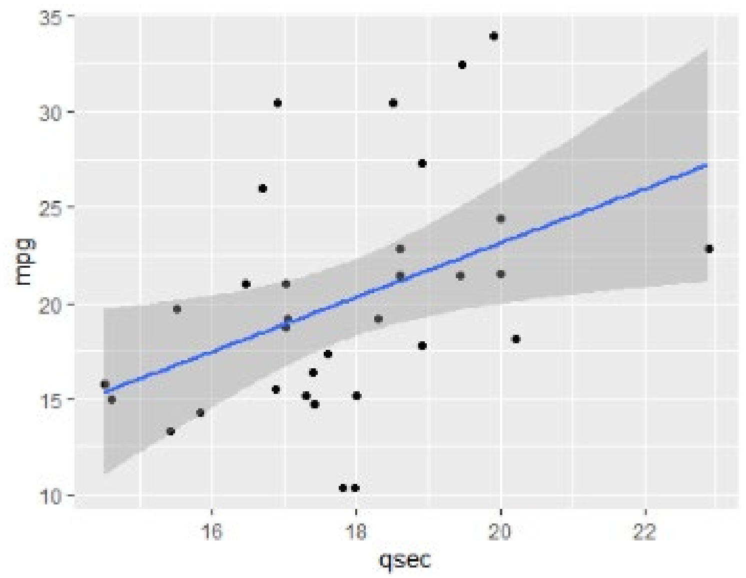


geom_smooth

```
ggplot(data = cars, aes(x = qsec, y = mpg)) +
  geom_point()+
  geom_smooth(method = "lm")
```

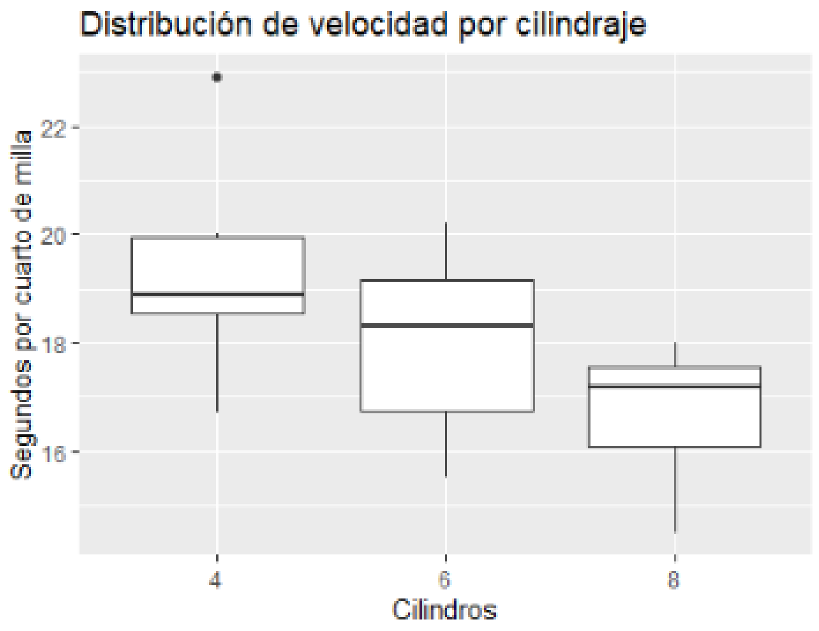
geom_smooth agrega una línea de tendencia al gráfico

Ojo: con el help puedes obtener distintos métodos de optimización. El default es “loess” (corre regresiones logales)



geom_boxplot

```
ggplot(data = cars, aes(x = cylinders, y = qsec))+
  geom_boxplot()+
  labs(title = "Distribución de velocidad por cilindraje",
       x = "Cilindros",
       y = "Segundos por cuarto de milla")
```



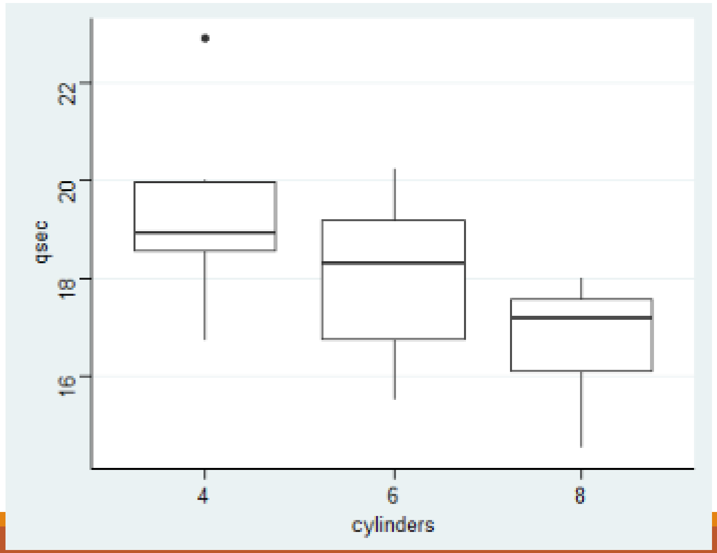
Tema

Más opciones para modificar temas en:

[Modify components of a theme — theme • ggplot2 (tidyverse.org)]

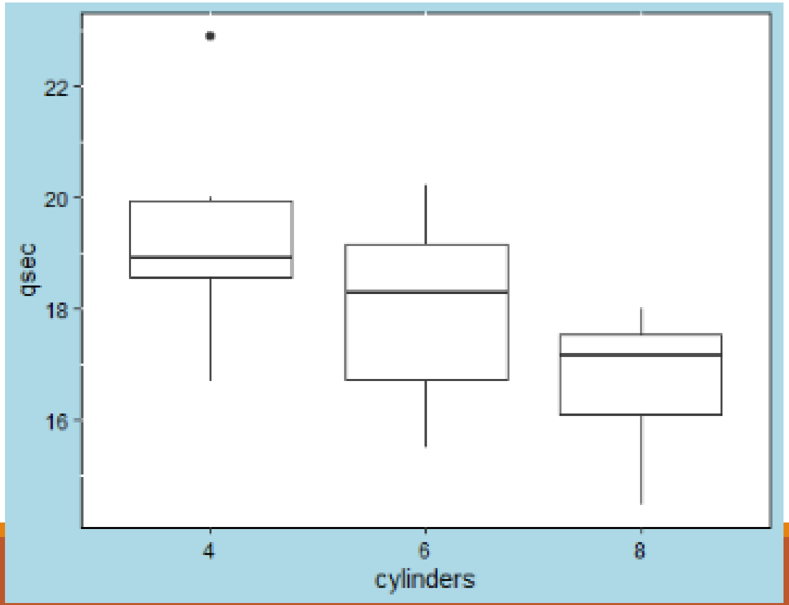
Existen temas predefinidos:

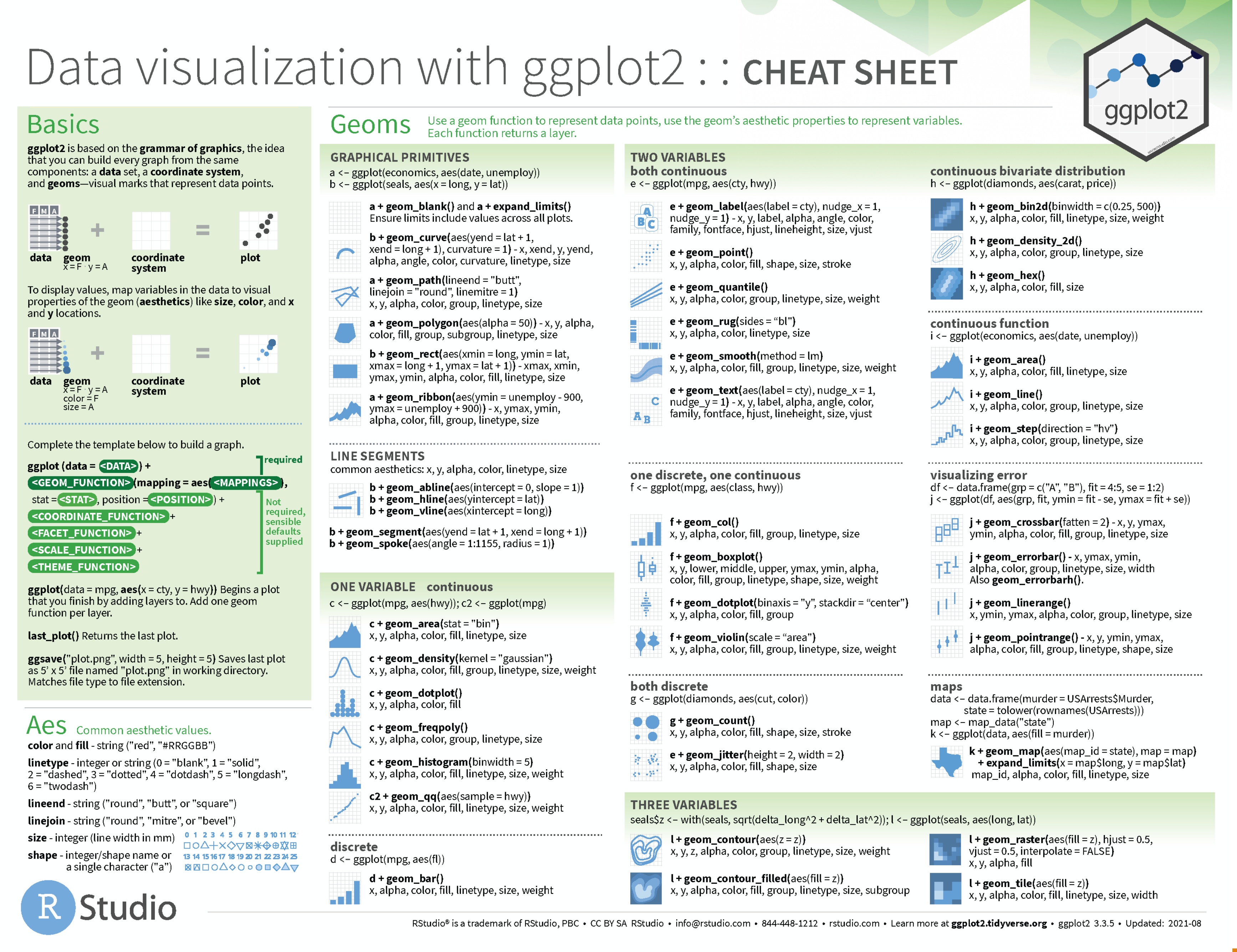
```
library(ggthemes)

ggplot(data = cars, aes(x = cylinders, y = qsec)) +
  geom_boxplot()+
  theme_stata()
```



O podemos modificar los componentes manualmente:

```
ggplot(data = cars, aes(x = cylinders, y = qsec))+
  geom_boxplot()+
  theme(panel.background = element_rect(fill = "white"),
        plot.background = element_rect(fill = "lightblue"),
        axis.text = element_text(colour = "black"))
```





Data visualization with ggplot2 : : CHEAT SHEET

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.

data + geom (x = F · y = A) + coordinate system = plot

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

data + geom (x = F · y = A, color = F, size = A) + coordinate system = plot

Complete the template below to build a graph.

```
ggplot (data = <DATA> ) +
  <GEOM_FUNCTION>(mapping = aes( <MAPPINGS> ),
  stat = <STAT> , position = <POSITION> ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

required

Not required, sensible defaults supplied

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

lineend - string ("round", "butt", or "square")

linejoin - string ("round", "mitre", or "bevel")

size - integer (line width in mm)

shape - integer/shape name or a single character ("a")

0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24 25

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank() and **a + expand_limits()**
Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1)
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)) - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly()
x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy))
x, y, alpha, color, fill, linetype, size, weight

discrete
d <- ggplot(mpg, aes(fl))

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

both continuous
e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl")
x, y, alpha, color, linetype, size

e + geom_smooth(method = lm)
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous
f <- ggplot(mpg, aes(class, hwy))

f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill, group

f + geom_violin(scale = "area")
x, y, alpha, color, fill, group, linetype, size, weight

both discrete
g <- ggplot(diamonds, aes(cut, color))

g + geom_count()
x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

continuous bivariate distribution
h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, color, fill, size

continuous function
i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size

visualizing error
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2) - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width
Also **geom_errorbarh()**.

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map)
+ expand_limits(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z))
x, y, alpha, color, fill, group, linetype, size, subgroup

l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)
x, y, alpha, fill

l + geom_tile(aes(fill = z))
x, y, alpha, color, fill, linetype, size, width

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **ggplot2.tidyverse.org** • ggplot2 3.3.5 • Updated: 2021-08

Puedes encontrar el cheatsheet solo buscando “ggplot2 cheatsheet”

Para más ejemplos de gráficas

The R Graph Gallery – Help and inspiration for R charts (r-graph-gallery.com)

- Vamos al código “ggplot.R”