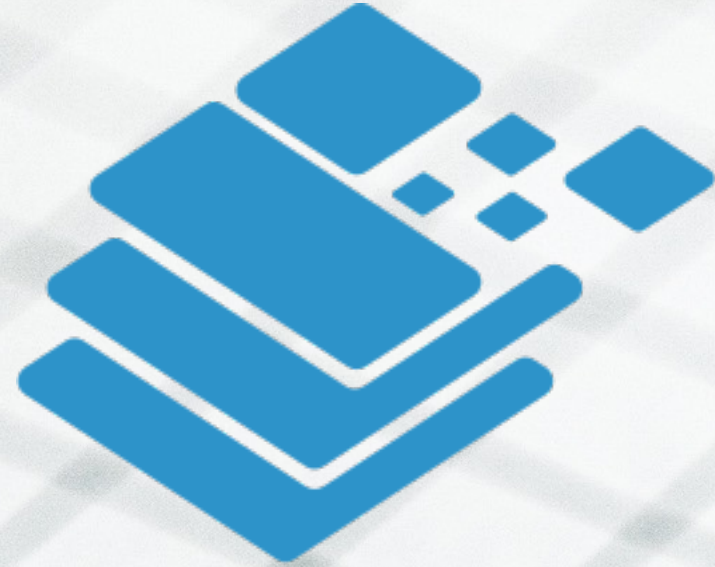




**Spatial
Lab
Analytics**

partimos pronto



**Spatial
Lab
Analytics**

ANÁLISIS DE DATOS CON R

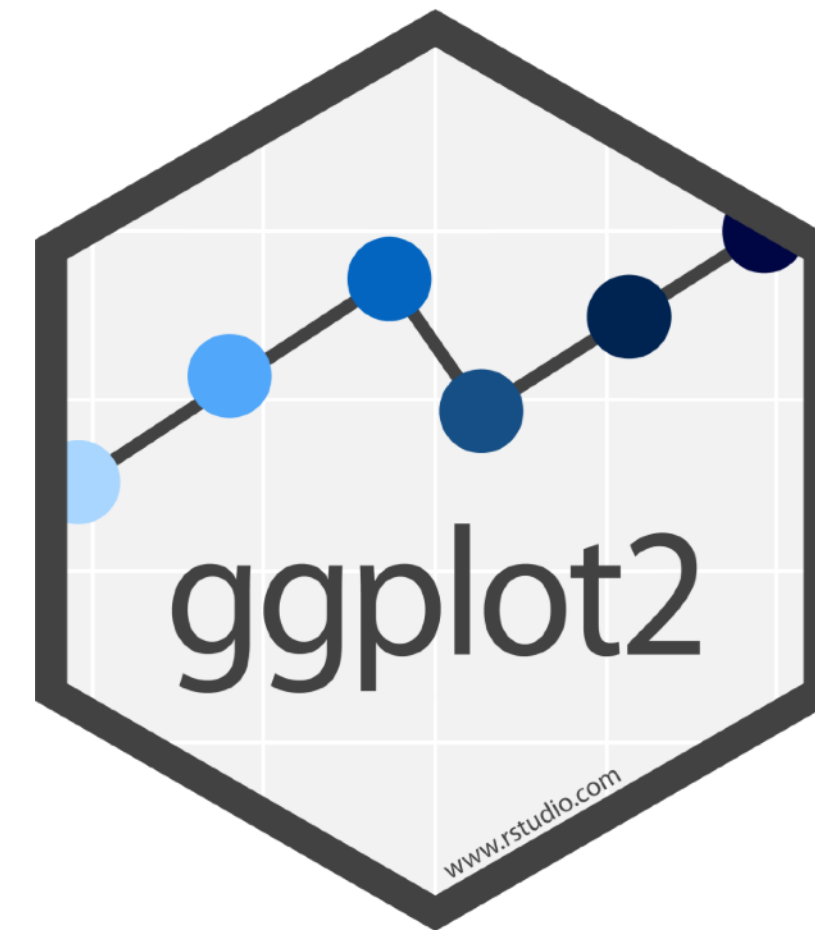
Manipulación de datos con R

Bastían Olea Herrera - baolea@uc.cl

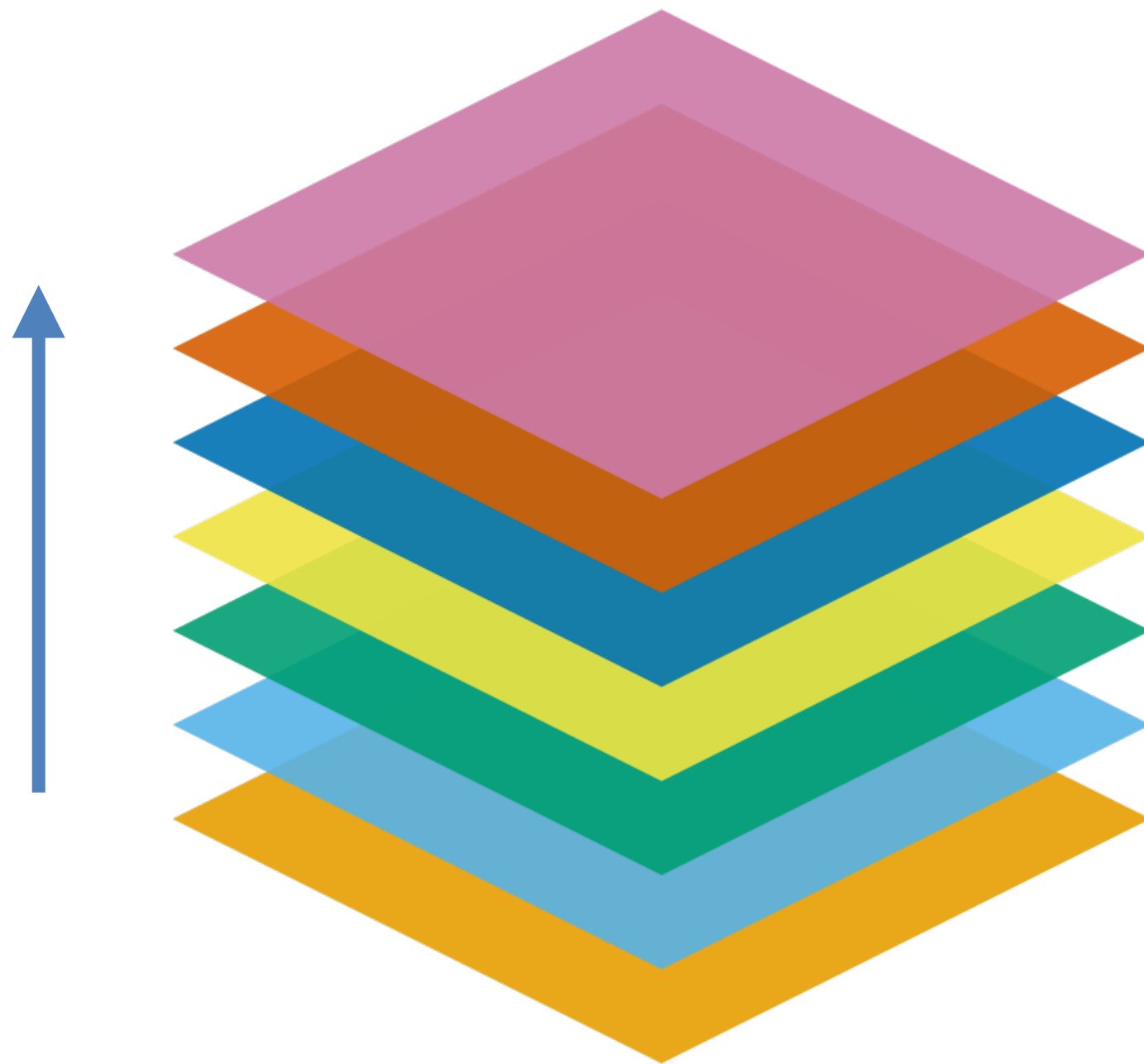
Visualización de datos usando la librería
`{ggplot2}`

ggplot2

- Librería de visualización de datos
- Opera principalmente con el paradigma de datos *tidy*
- Dibuja gráficos por medio de capas
- Produce gráficos vectoriales
- Entre sus beneficios está su flexibilidad, adaptabilidad y reusabilidad



Gramática de gráficos



Temas definición de la apariencia general y de cada elemento

Coordenadas especificación de los límites y proyección

Facetas dividir la visualización en paneles distintos

Escalas especificación de la forma en que se mapea

Capas agregar elementos en base a los datos mapeados

Mapeos conectar variables a formas de visualizarlas

Datos variables disponibles para construir el gráfico





Temas

Coordenadas

Facetas

Escalas

Capas

Mapeos

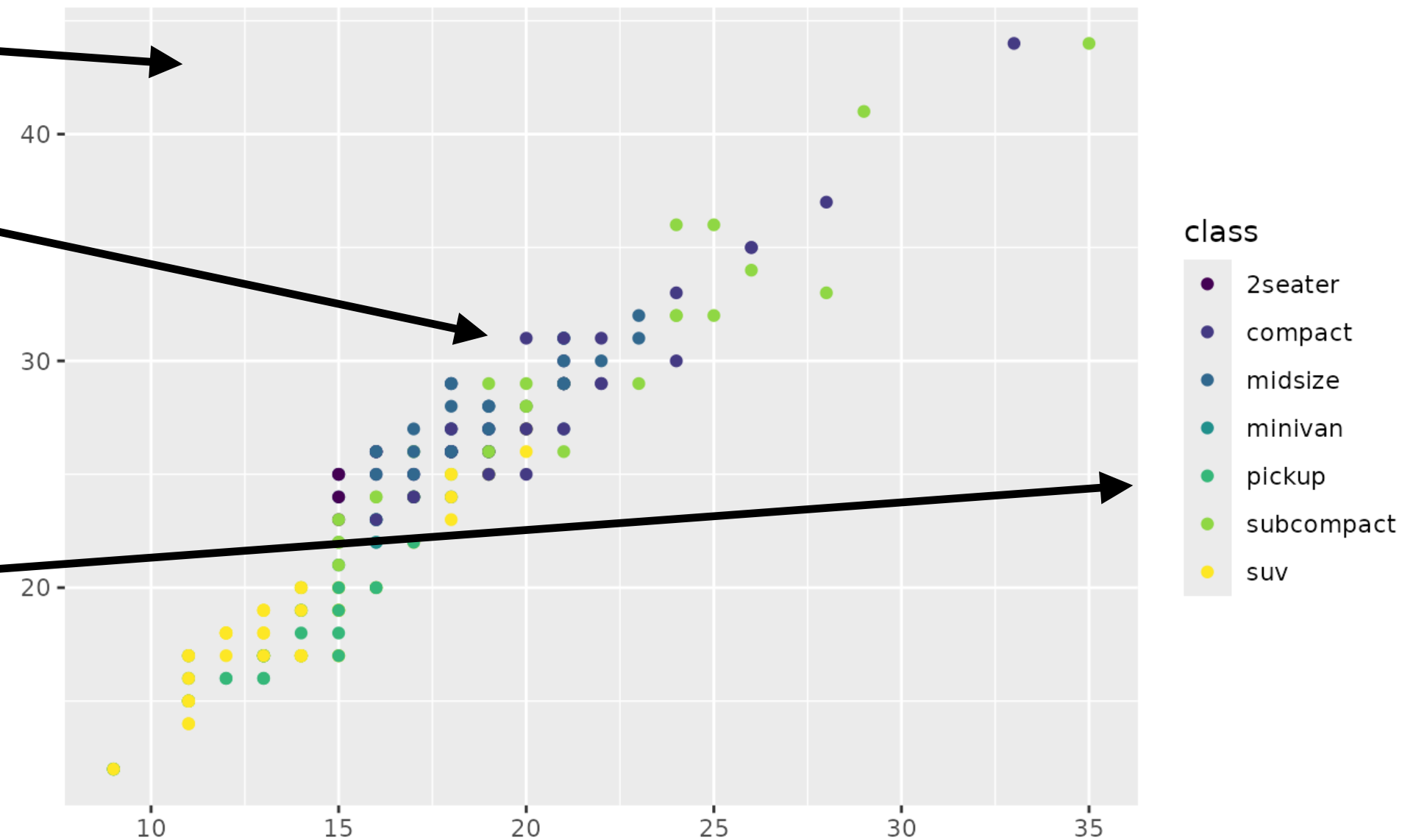
Datos

```
iris |>
  ggplot() +
  aes(Sepal.Length, Sepal.Width, color = Species) +
  geom_point() +
  scale_color_discrete() +
  coord_cartesian(xlim = c(4, 9)) +
  theme_minimal()
```



Elementos de los gráficos

- Coordenadas (coords)
- Geometrías (geoms)
 - Estéticas
 - Tamaño, color, relleno, transparencia, forma, tipo de líneas...
 - Escalas
- Guías (leyendas)
- Temas
- Facetas

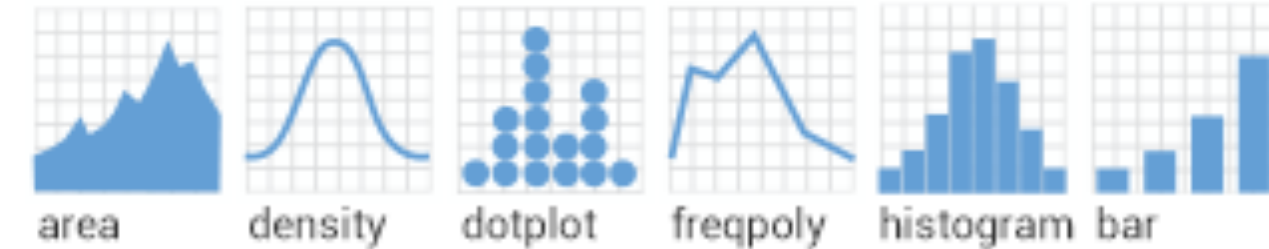


Geometrías

Basic



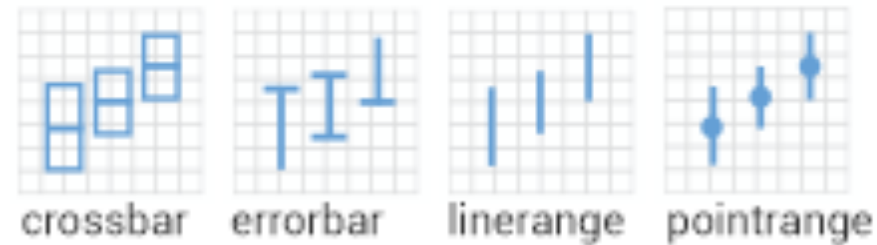
One variable



Two variables



Error



Three variables



Map



Temas

`theme_minimal()`

`theme_classic()`

`theme_void()`

Componentes

- `grid`
- `scales`
- `axis`
- `legend`

Elementos

- `element_text()`
- `element_line()`
- `element_blank()`

ggplot2 theme elements reference

Set minimal as the baseline theme:

```
theme_minimal() +  
theme(theme.element = element_type())
```

Use `element_blank()` to remove an element

Axis titles, text, ticks, and lines can be specified per axis using theme inheritance by putting `.x/.y` at the end of the theme element.

```
axis.line.y = element_line()
```

```
axis.title.y = element_text()
```

```
panel.grid.major = element_line()
```

```
panel.grid.minor = element_line()
```

```
axis.text.y
```

```
axis.text = element_text()
```

```
axis.text.x
```

```
plot.title.position = "plot"  
plot.caption.position = "plot"  
  
plot.title = element_text()  
plot.subtitle = element_text()
```

"plot" means that they will be aligned to the entire plot (instead of the panel)

```
plot.margin = margin(25, 25, 25, 25)
```

```
legend.title  
= element_text()
```

```
legend.background  
= element_rect()
```

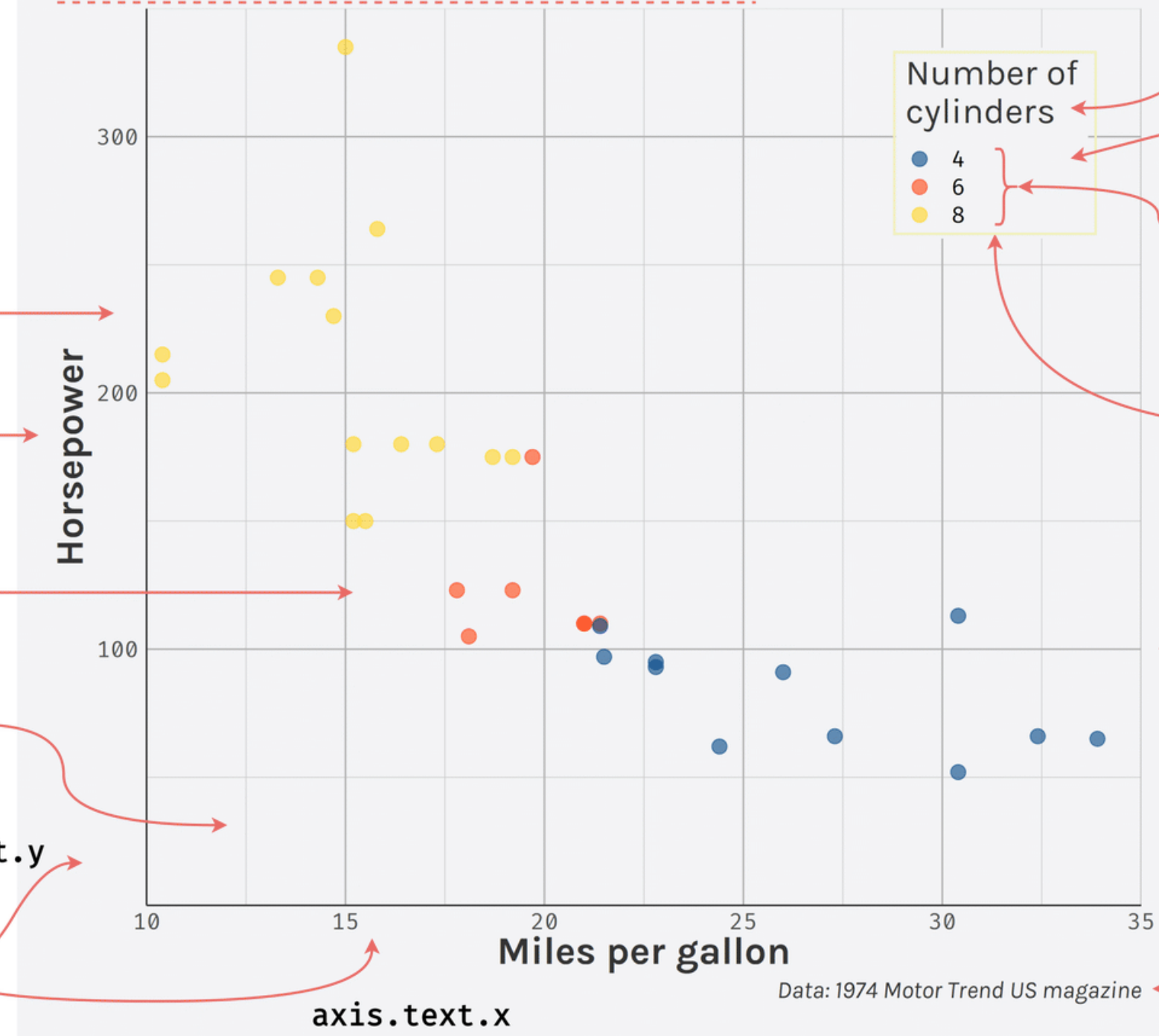
```
legend.text  
= element_text()
```

```
legend.position  
= c(.85,.85) / "none" /  
"left" / "right" /  
"bottom" / "top"
```

```
plot.background  
= element_rect()
```

```
plot.caption  
= element_text()
```

Miles per Gallon & Horsepower of 32 Automobiles(1973-74 models)



isabella-b

`text = element_text()` ← modifications will be applied to all text elements

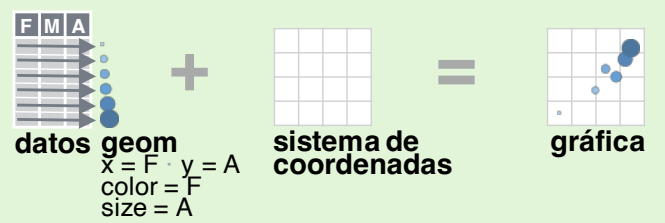
Full list of elements at ggplot2.tidyverse.org/reference/theme

Básico

ggplot2 se basa en la gramática de los gráficos, la idea de que se pueden construir todos los gráficos a partir de los mismos componentes: un conjunto de datos, un sistema de coordenadas y geoms, marcas visuales que representan puntos de datos.



Para mostrar valores, asigne variables de los datos a propiedades visuales del geom (estética) como el tamaño, el color y las ubicaciones x e y.



Complete la siguiente plantilla para crear un gráfico.

ggplot (data = **<DATOS>**) + **<FUNCIÓN_GEOM>** (mapping = aes(**<MAPEADO>**)), stat = **<STAT>**, position = **<POSICIÓN>**) + **<FUNCIÓN_COORDENADAS>** + **<FUNCIÓN_FACETADO>** + **<FUNCIÓN_ESCALA>** + **<FUNCIÓN_TEMA>**

requerido

No requerido, sensible predeterminados suministrados

ggplot(data = mpg, aes(x = cty, y = hwy)) Comienza un gráfico al que se termina añadiendo capas. Agregar una función geom por capa.

last_plot() Devuelve la última gráfica.

ggsave("plot.png", width = 5, height = 5) Guarda el último gráfico como un archivo de 5' x 5' llamado "plot.png" en el directorio de trabajo. Coincide el tipo de archivo con la extensión del archivo.

Aes

Valores estéticos comunes.

color y **fill** - texto ("red", "#RRGGBB")

linetype – entero o texto (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

size - enterp (en mm para el tamaño de los puntos y el texto)

linewidth - entero (en mm para el ancho de líneas)

shape - entero/nombre de la forma o un carácter ("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

Utilice una función geom para representar puntos de datos, utilice las propiedades estéticas del geom para representar variables. Cada función devuelve una capa.

GRÁFICAS PRIMITIVAS

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



a + geom_blank() y **a + expand_limits()**
Asegúrese de que los límites incluyan valores en todas las gráficas.



b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, 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

SEGMENTOS DE LÍNEA

Estéticas común: 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))

UNA VARIABLE continua

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

discreta

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



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

DOS VARIABLES

ambas continuas

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

una discreta, una continua

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

ambas discretas

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

TRES 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

distribución bivariada continua

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

función continua

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

visualización de 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

mapas

Dibuje el objeto geométrico apropiado en función de las características simples presentes en los argumentos de data. aes():
map_id, alpha, color, fill, linetype, linewidth.

nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"))



ggplot(nc) + geom_sf(aes(fill = AREA))



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

Extensiones

- Existen muchos paquetes que extienden las capacidades de `{ggplot2}`

`{ggforce}`: nuevas geometrías

`{patchwork}`: combinación de gráficos

`{ggrepel}`: separación de textos
sobrepuestos

`{gganimate}`: gráficos animados

`{marquee}` y `{ggtext}`: markdown y
html en textos

`{ggfx}`: shaders, filtros y efectos
especiales

`{ggiraph}` y `{plotly}`: gráficos
interactivos





**Spatial
Lab
Analytics**

Soluciones en análisis de datos

www.spatiallab.cl