

Ciencia de datos en R

Big Data: Marco conceptual, técnicas y aplicaciones

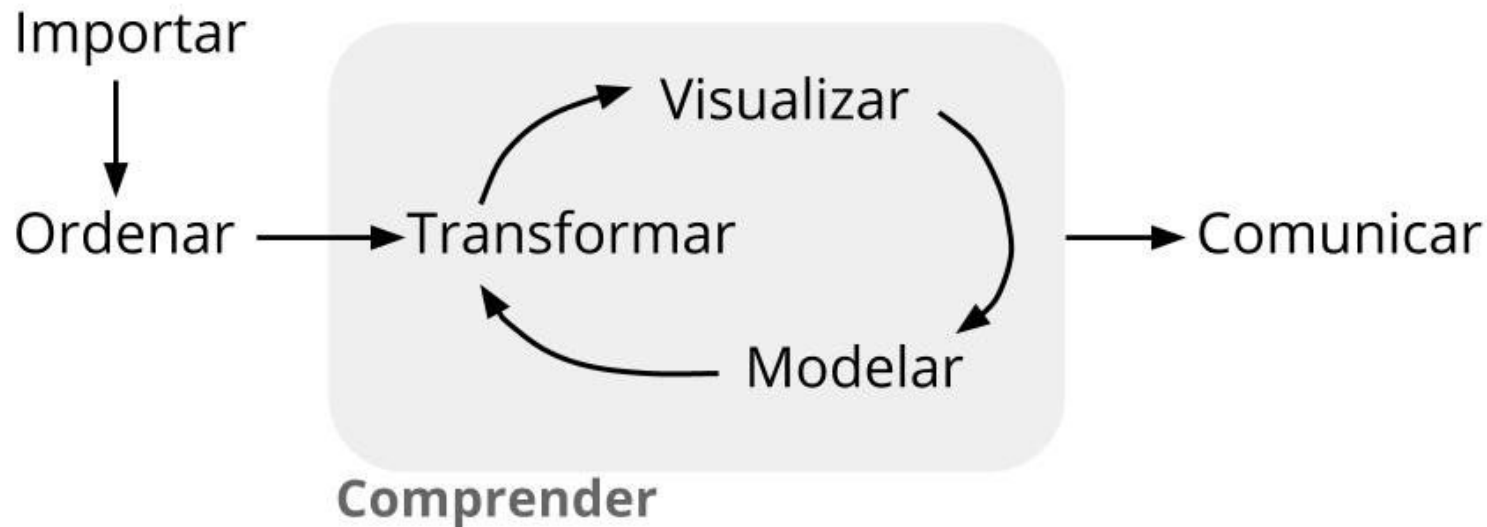
Clase 3

Temario de hoy

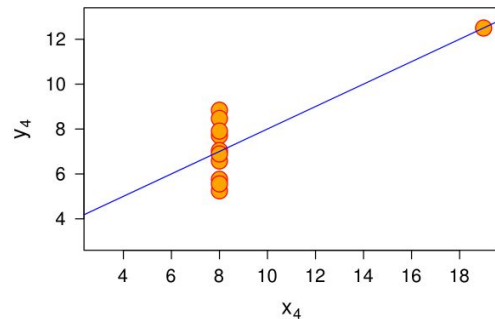
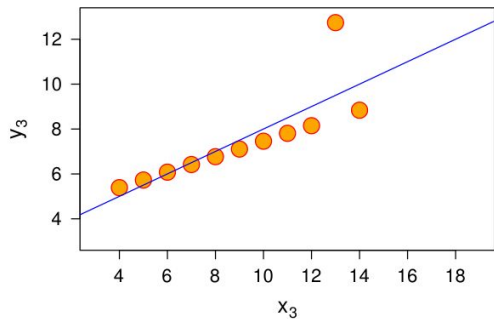
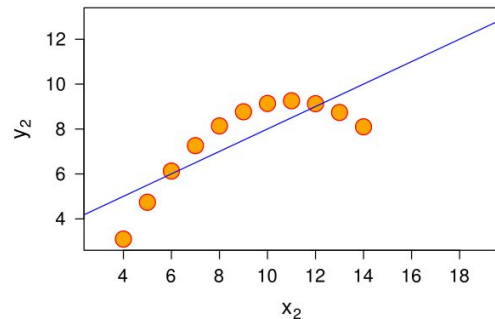
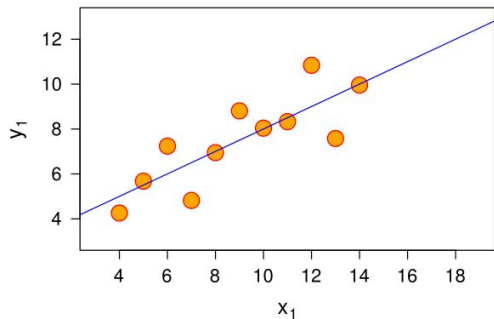
- Introducción a las visualizaciones
- ggplot2
- Otras visualizaciones
- BBC
- Recursos

¿Por dónde vamos?

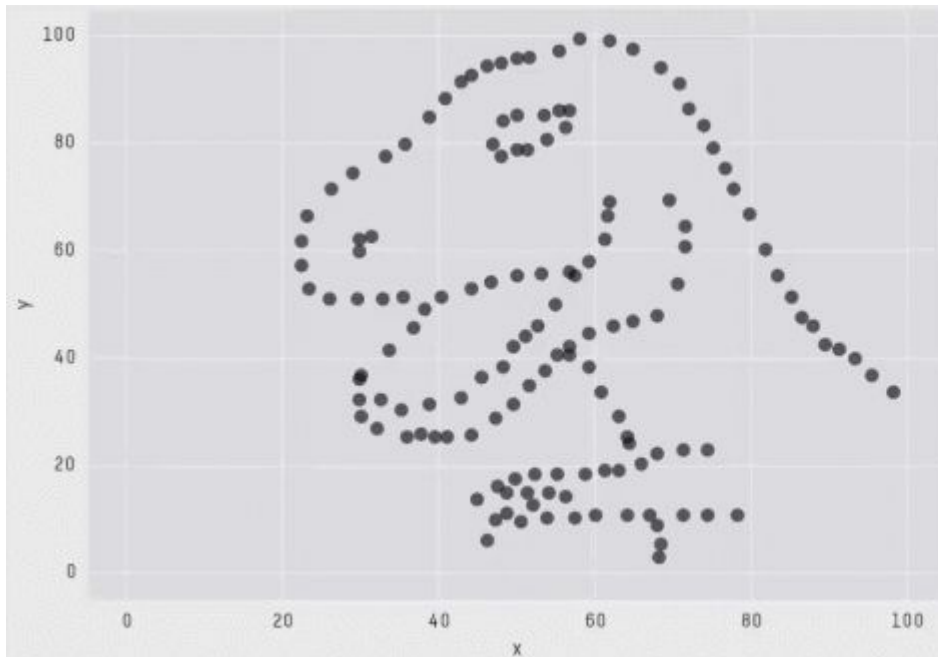
El proceso del análisis de datos



¿Por qué visualizar datos?



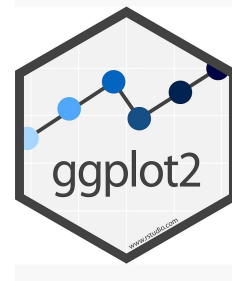
¿Por qué visualizar datos?



X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

ggplot2

- Se basa en una teoría de [The Grammar of Graphics](#)
- Todo gráfico se puede compone de:
 - ◆ Elementos esenciales:
 - Datos: la vedette de los gráficos
 - Estética: en qué eje va cada elemento y con qué atributos
 - Geometrías: barras, líneas, puntos, etc?
 - ◆ Elementos opcionales:
 - Facetado: pequeños subsets particulares
 - Estadísticas: media, cuartile, mediana, etc
 - Coordenadas: transformar los ejes, etc
 - Temas: hacerlo pituco



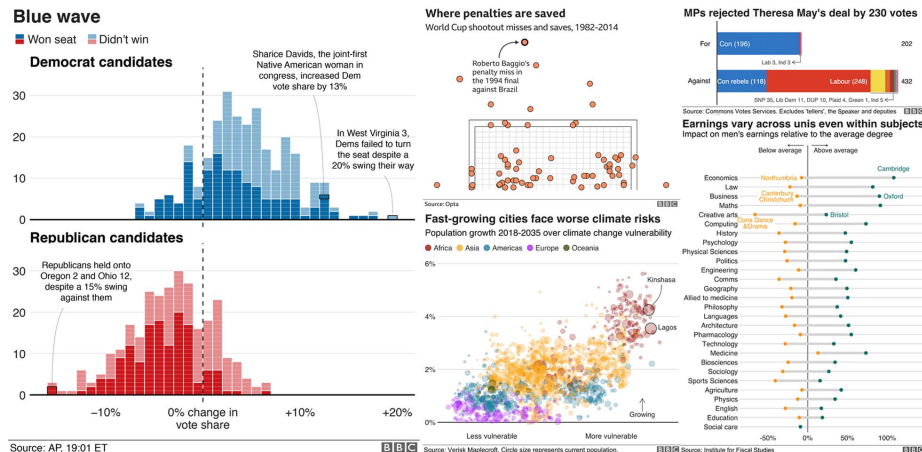
Caso de uso: BBC

→ BBC usa la base de “ggplot” para sus gráficos más rápidos.

→ Ventajas: Libertad para graficar + reproductibilidad

→ Links:

- ◆ [Recetario “bbplot”](#)
- ◆ [Nota periodística al respecto](#)
- ◆ [Presentación periodistas BBC \(video - inglés\)](#)



Un mundo de posibilidades...



Más información:
[from Data to Viz](#)

Recursos: ggplot2

- Google
- R for Data Science
 - ◆ [Cap 3: Data Visualization](#)
 - ◆ [Cap 28: Graphics for communication](#)
- Libros
 - ◆ [R Graphics Cookbook](#) (Winston Chang, 2018)
 - ◆ [ggplot2: Elegant Graphics for Data Analysis \(Use R!\)](#) (Hadley Wickham, 2016)

Recursos: Cheatsheets

- [RMarkdown](#)
- [Importar datos](#)
- [Transformación de datos](#)
- [ggplot2](#)

Visualización de Datos usando ggplot2

Guía Rápida

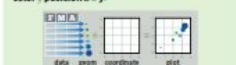


Conceptos Básicos

ggplot2 se basa en la idea que cualquier gráfica se puede construir usando estos tres componentes: **datos**, **coordenadas** y **objetos geométricos (geoms)**. Este concepto se llama **gramática de las gráficas**.



Para visualizar resultados, asigne variables a las propiedades visuales, o **estéticas**, como **tamaño**, **color**, **posición** a **x** y **y**.



Para construir una gráfica completa este patrón:



ggplot(data = <datos>) +
geom_<función> (aes(<estéticas>))
coord_<función> (aes(<coordenadas>))
geom_<función> (aes(<estéticas>))
geom_<función> (aes(<estéticas>))

ggplot(data = mpg, aes(y = hwy))
Este comando construye una gráfica, compuesta mediante agregando capas, un **geom** por capa.

qplot(x = cty, y = hwy, data = mpg, geom = "point")
Este comando es una gráfica completa, tiene datos, las estéticas están asignadas y por lo menos un **geom**.

last_plot()
Devuelve la última gráfica.

ggsave("plot.png", width = 5, height = 5)
La última gráfica es guardada como una imagen de 5 por 5 pulgadas, usa el mismo tipo de archivo que la extensión.

Geoms - Funciones geom se utilizan para visualizar resultados. Asigne variables a las propiedades estéticas del geom. Cada geom forma una capa.

Geométricas Elementales

a `geom_blank()`
(Buena para expandir límites)

b `geom_curve(aes(yend = lat + 1, wend = long - 1, curvature = 1), linejoin = "round", linemitre = 1)`
x, y, alpha, color, fill, group, linetype, size

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

a `geom_polygon(aes(group = group))`
x, y, alpha, color, fill, group, linetype, size

b `geom_rect(aes(min = long, ymin = lat, max = long + 1, ymax = lat + 1))`
x, y, alpha, color, fill, group, linetype, size

a `geom_ribbon(aes(ymin = unemployment, ymax = unemployment + 900))`
x, y, alpha, color, fill, group, linetype, size

Segmentos Lineales

propiedades básicas: *x, y, alpha, color, linetype, size*

b `geom_abline(aes(intercept = 0, slope = 1))`
x, y, alpha, color, fill, group, linetype, size

b `geom_hline(aes(intercept = lat))`
x, y, alpha, color, fill, group, linetype, size

b `geom_vline(aes(intercept = long))`
x, y, alpha, color, fill, group, linetype, size

b `geom_spoke(aes(angle = 1:155, radius = 1))`
x, y, alpha, color, fill, group, linetype, size

Una Variable

c `geom_area(aes(hwy))`
x, y, alpha, color, fill, linetype, size

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

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

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

Discreta

X Continua, Y Continua

e `geom_label(aes(label = cty, nudge_x = 1, nudge_y = 1, check_overlap = TRUE))`
x, y, alpha, color, angle, color, family, fontface, fontsize, linetype, size, weight

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

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 = "b")`
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, check_overlap = TRUE))`
x, y, alpha, color, angle, color, family, fontface, fontsize, linetype, size, weight

X Discreta, Y Continua

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(binwidth = "y", stackdir = "center")`
x, y, alpha, color, fill, group

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

X Discreta, Y Discreta

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

Tres Variables

h `geom_raster(aes(z1 = z1, z2 = z2, z3 = z3))`
x, y, alpha, color, fill, linetype, size, weight

h `geom_tile(aes(z1 = z1, z2 = z2))`
x, y, alpha, color, fill, linetype, size, weight

Dos Variables

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

h `geom_density2d()`
x, y, alpha, color, fill, linetype, size, weight

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

Distribución Bivariada Continua

i `geom_arena()`
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

Función Continua

i `geom_smooth()`
x, y, alpha, color, group, linetype, size, weight

Visualizando el Error

j `geom_crossbar(latten = 2)`
x, y, alpha, color, fill, group, linetype, size

j `geom_errorbar()`
x, y, alpha, color, fill, group, linetype, size, weight

j `geom_linerange()`
x, y, alpha, color, fill, group, linetype, size, weight

j `geom_pointrange()`
x, y, alpha, color, fill, group, linetype, shape, size, weight

Mapas

k `geom_map(aes(map = map, map = map))`
x, y, alpha, color, fill, linetype, size, weight

Argumentos

l `geom_map(aes(map = map, map = map))`
x, y, alpha, color, fill, linetype, size, weight