

Ciencia de datos en R

Big Data: Marco conceptual, técnicas y aplicaciones

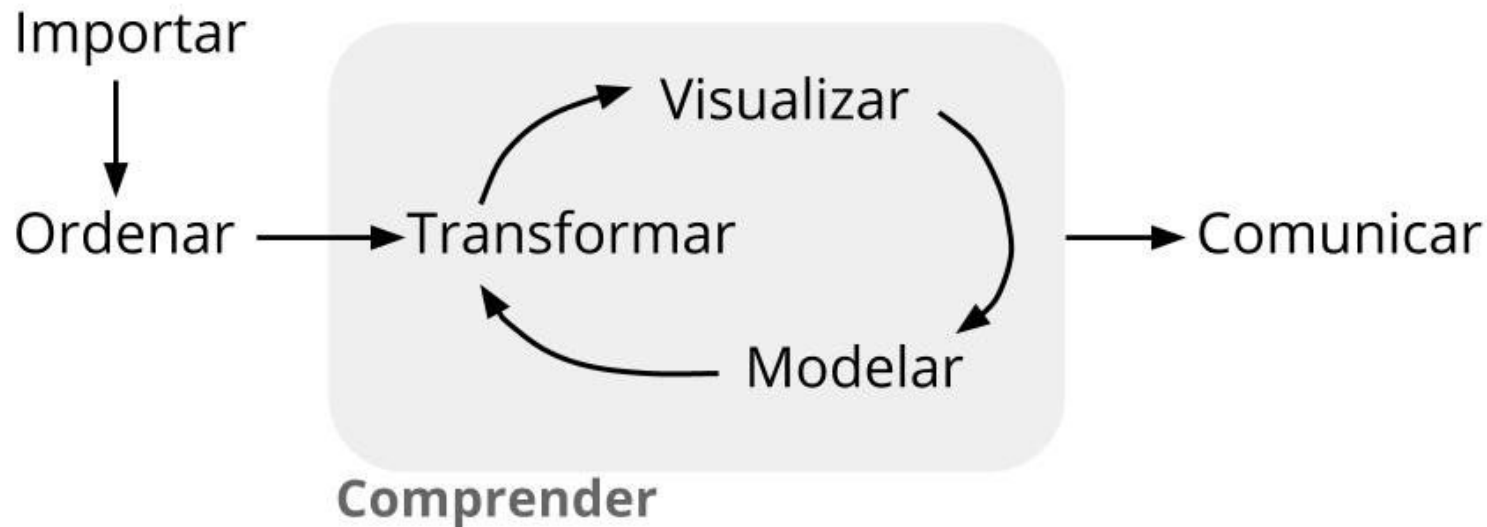
Clase 3

Temario de hoy

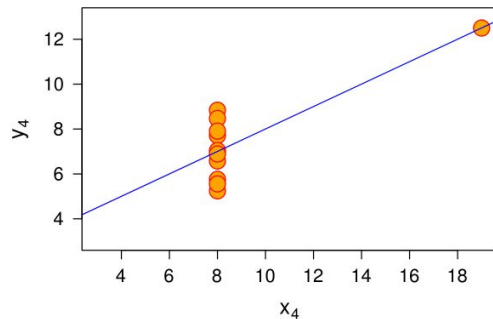
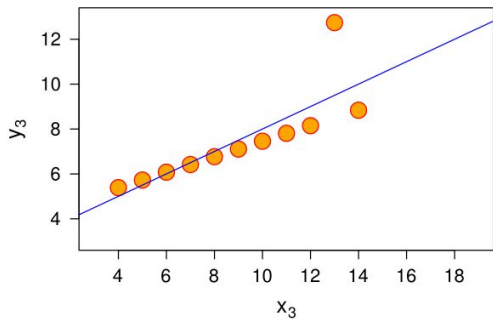
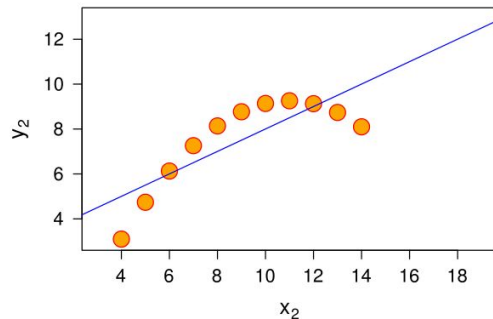
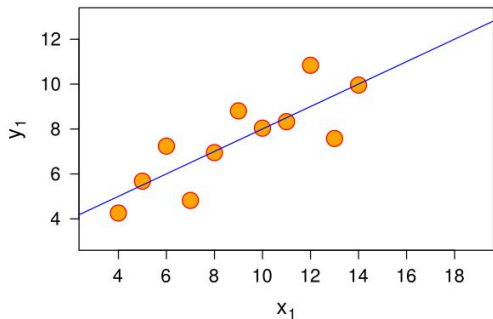
- Introducción a las visualizaciones
- ggplot2
- Otras visualizaciones
- BBC
- Recursos

¿Por dónde vamos?

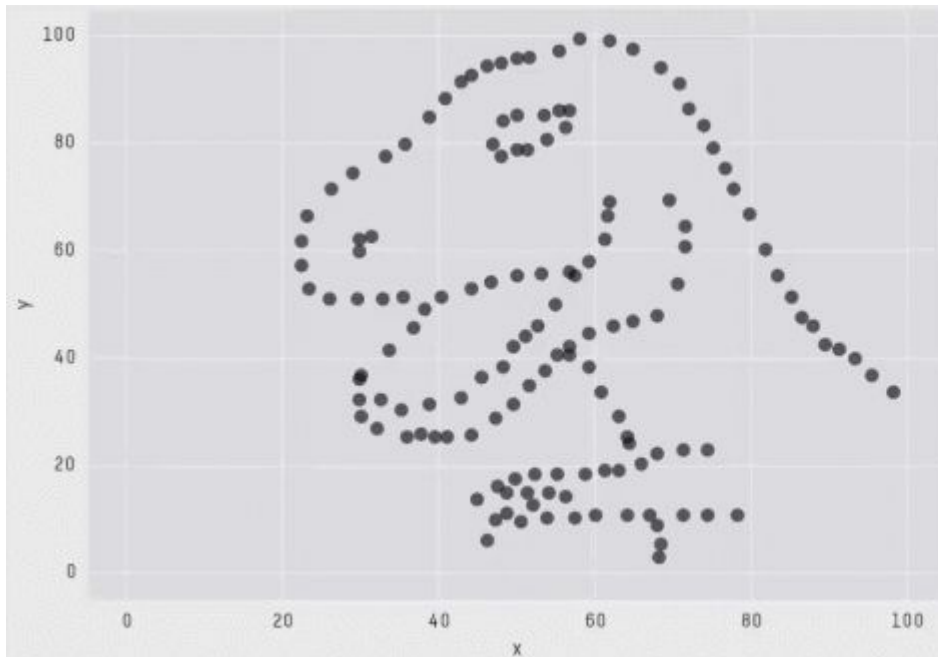
El proceso del análisis de datos



¿Por qué visualizar datos?



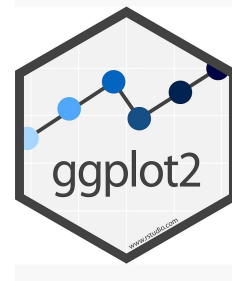
¿Por qué visualizar datos?



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



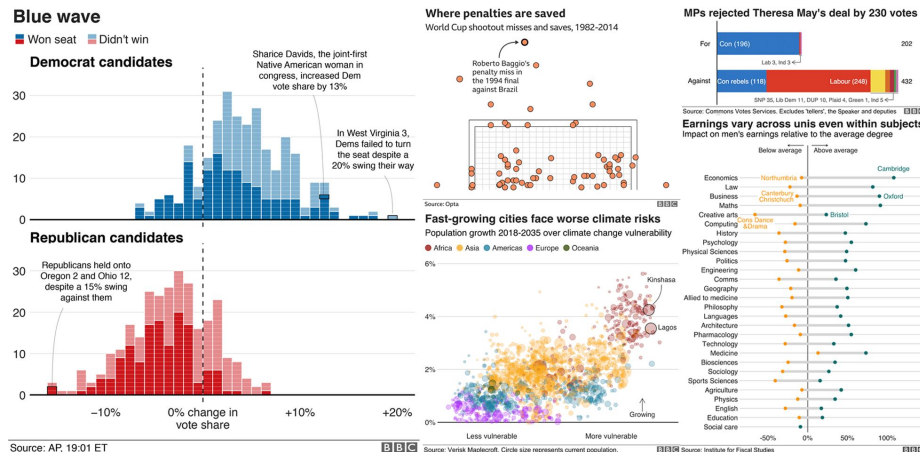
Caso de uso: BBC

→ 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](#)

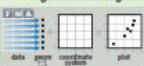
Visualización de Datos usando ggplot2

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 cada patrón



ggplot(data = mpg, aes(x = cty, y = hwy))
Este comando construye una gráfica completa, tiene datos, las estéticas están asignadas y por lo menos un geom.

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

© 2014 RStudio, Inc. RStudio es una marca registrada de RStudio, Inc. RStudio es un producto de RStudio, Inc. RStudio es un producto de RStudio, Inc.

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, xend = 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

a = 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

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

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

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

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

Una Variable

c = ggplot(mpg, aes(hwy))
x, y, alpha, color, fill, group, linetype, size

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

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

c = geom_dotplot()
x, y, alpha, color, fill, group, linetype, size

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

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

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

Discreta

d = geom_bar()
x, y, alpha, color, fill, group, linetype, size

Dos Variables

X Continua, Y Continua
e = ggplot(mpg, aes(cty, hwy))

e = geom_label(aes(label = cty, nudges_x = 1, nudges_y = 1, check_overlap = TRUE))
x, y, alpha, color, fill, group, linetype, size

e = geom_jitter(height = 2, width = 2)
x, y, alpha, color, fill, group, linetype, size

e = geom_point()
x, y, alpha, color, fill, group, linetype, size

e = geom_quantile()
x, y, alpha, color, fill, group, linetype, size

e = geom_rug(sides = "b")
x, y, alpha, color, fill, group, linetype, size

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

e = geom_text(aes(label = cty, nudges_x = 1, nudges_y = 1, check_overlap = TRUE))
x, y, alpha, color, fill, group, linetype, size

X Discreta, Y 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(binwidth = "y", stackdir = "center")
x, y, alpha, color, fill, group, linetype, size

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

X Discreta, Y Discreta
g = ggplot(count, aes(class, color))

g = geom_count()
x, y, alpha, color, fill, group, linetype, size

Tres Variables

h = geom_raster(aes(fill = z))
x, y, alpha, color, fill, group, linetype, size

h = geom_tile(aes(fill = z))
x, y, alpha, color, fill, group, linetype, size

Distribución Bivariada Continua

h = ggplot(diamonds, aes(carat, price))
x, y, alpha, color, fill, group, linetype, size

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

h = geom_density2d()
x, y, alpha, color, fill, group, linetype, size

h = geom_hex()
x, y, alpha, color, fill, group, linetype, size

Función Continua

i = ggplot(economics, aes(date, unemploy))
x, y, alpha, color, fill, group, linetype, size

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

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

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

Visualizando el Error

df = data.frame(mpg = c("a", "b"), fit = 4.5, se = 1.2)
j = ggplot(df, aes(fit, ymin = fit - se, ymax = fit + se))

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

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

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

j = geom_pointrange()
x, y, alpha, color, fill, group, linetype, size

Mapas

data = data.frame(murder = USArrests\$Murder, state = tolower(state.names[USArrests]))
map = map_data("state")
k = ggplot(data, aes(fill = murder))

k = geom_map(aes(map = map, fill = murder))
x, y, alpha, color, fill, group, linetype, size

Argumentos

label = etiqueta angular
size = tamaño
weight = peso
color = color
fill = relleno
stroke = borde
linetype = tipo de línea
group = grupo de líneas

Recursos: Repositorios de datos

- [Datos.gob.ar](https://datos.gob.ar) (repositorio datos públicos Nacion)
- [Buenos Aires data](https://datos.buenosaires.gov.ar) (repositorio datos Ciudad)
- [Datos publicos MIOPyV](https://datos.mio.pyv)