

DE DATOS A HISTORIAS: CREA GRÁFICOS IMPACTANTES EN R

María José Ginzo Villamayor

Universidade de Santiago de Compostela
Centro de Investigación e Tecnoloxía Matemática de Galicia

17 de abril de 2026

Contenidos

- » Introducción
- » ¿Por qué empezar por los gráficos básicos?
- » Gráficos básicos
- » ¿Cuándo se nos queda corto base R??
- » Salto a ggplot2
- » Buenas prácticas
- » Conclusiones

Idea principal

Mensaje de la charla

Con R podemos hacer gráficos muy rápido, pero la diferencia entre **mostrar datos** y **contar una historia** está en cómo elegimos el gráfico, qué destacamos y qué simplificamos.

Idea principal

Mensaje de la charla

Con R podemos hacer gráficos muy rápido, pero la diferencia entre **mostrar datos** y **contar una historia** está en cómo elegimos el gráfico, qué destacamos y qué simplificamos.

- Empezar con gráficos básicos de base R
- Mejorarlos con decisiones sencillas de diseño
- Dar el salto a ggplot2
- Ver ejemplos con mapas

Un buen gráfico no empieza en el código

- ¿Qué quiero contar?
- ¿Comparar?
- ¿Mostrar evolución?
- ¿Ver distribución?
- ¿Explorar relación entre variables?
- ¿Situar la información en el territorio?

Regla práctica

Primero la **pregunta**, después el **gráfico**, y solo al final el **código**.

¿Por qué empezar por los gráficos básicos?

- Porque se entienden en un minuto.
- Porque permiten explicar la lógica visual: ejes, categorías, distribución, relación.
- Porque ayudan a ver que R **base ya hace muchas cosas**.
- Y porque después se aprecia mejor lo que aporta `ggplot2`.

Idea fuerza

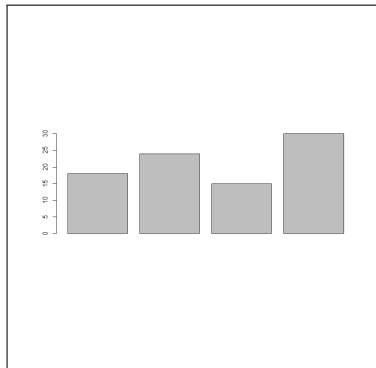
Base R y `ggplot2` no compiten: **se complementan**.

Base R es inmediatez; `ggplot2` es gramática, capas y refinamiento.

Empezar simple: gráficos básicos en R

```
ventas <- c(18, 24, 15, 30)  
barplot(ventas)
```

- Muy poca sintaxis
- Ideal para enseñar y prototipar
- Permite entender rápido la lógica gráfica



El mismo gráfico, mejor contado

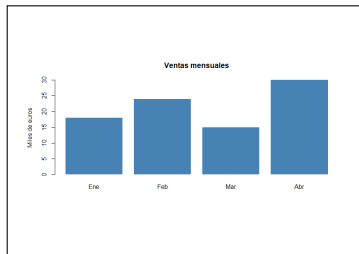
```

ventas <- c(18, 24, 15, 30)
meses <- c("Ene", "Feb", "Mar", "Abr")

barplot(
  ventas,
  names.arg = meses,
  col = "steelblue",
  border = NA,
  main = "Ventas mensuales",
  ylab = "Miles de euros"
)

```

- Título informativo
- Etiquetas claras
- Menos ruido visual
- Color con intención



Elegir bien el tipo de gráfico

Pregunta	Gráfico adecuado
Comparar categorías	Barras
Ver evolución	Líneas
Analizar distribución	Histograma / boxplot
Relacionar dos variables	Dispersión
Situar valores en el territorio	Mapa

Evitar

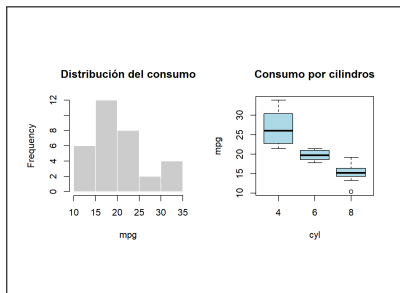
- gráfico circular por defecto
- exceso de colores
- leyendas innecesarias
- títulos genéricos como “Gráfico 1”

Distribuciones: histograma y boxplot

```
hist(mtcars$mpg,
     col = "grey80",
     border = "white",
     main = "Distribución del
consumo",
     xlab = "mpg")

boxplot(mpg ~ cyl, data = mtcars,
        col = "lightblue",
        main = "Consumo por
cilindros")
```

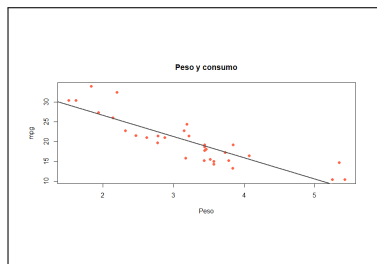
- El histograma muestra forma
- El boxplot resume y compara
- Muy útiles para explorar datos
- Y si queremos más detalle visual, podemos usar violin plots



Relaciones entre variables: dispersión

```
plot(mtcars$wt, mtcars$mpg,
     pch = 19,
     col = "tomato",
     xlab = "Peso",
     ylab = "mpg",
     main = "Peso y consumo")
abline(lm(mpg ~ wt, data = mtcars),
       col = "grey30", lwd = 2)
```

- Muy potente para descubrir patrones
- Permite ver tendencia
- Ideal para introducir modelización



Distribución de una variable

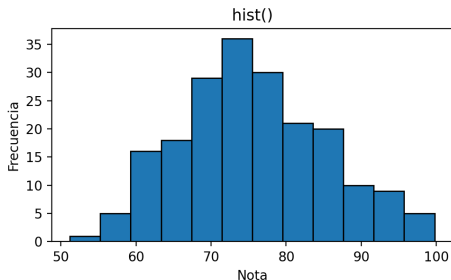
Código

```

notas <- rnorm(200, mean = 75,
              sd = 10)

hist(
  notas,
  breaks = 12,
  main = "Distribución de notas",
  xlab = "Nota"
)

```



- Un histograma no compara grupos: muestra forma.
- Sirve para hablar de dispersión, sesgo y valores extremos.

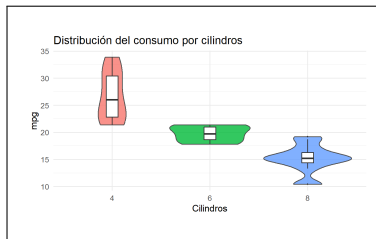
Más allá del boxplot: violin plots

```
library(ggplot2)

ggplot(mtcars,
       aes(x = factor(cyl),
           y = mpg,
           fill = factor(cyl))) +
  geom_violin(alpha = 0.8,
              color = "grey30") +
  geom_boxplot(width = 0.12,
               fill = "white",
               outlier.size = 0.7,
               show.legend = FALSE) +

  labs(
    title = "Distribución del consumo por cilindros",
    x = "Cilindros",
    y = "mpg"
  ) +
  theme_minimal(base_size = 13) +
  theme(legend.position = "none")
```

- El boxplot resume
- El violin plot añade la forma
- Es el mismo dato, pero contado con más riqueza
- Útil cuando queremos comparar distribuciones con más detalle



La visualización también se aprende

- La visualización no es solo estética.
 - Elegir un gráfico también es una decisión analítica.
 - No siempre basta con usar el gráfico más conocido.
 - Explorar alternativas puede ayudarnos a comunicar mejor.
- boxplot + puntos
 - violin plot
 - heatmap
 - facetas
 - mapas

Idea clave

Saber hacer un gráfico no es lo mismo que saber elegir **el gráfico adecuado**.

Mensaje

R no solo hace gráficos correctos: también puede hacer gráficos muy llamativos e incluso inspiradores.

¿Cuándo se nos queda corto base R?

- Cuando queremos componer gráficos complejos
- Cuando necesitamos una estética más cuidada
- Cuando queremos separar datos, geometrías y escalas
- Cuando hacemos figuras para artículos, informes o divulgación

Ahí entra `ggplot2`

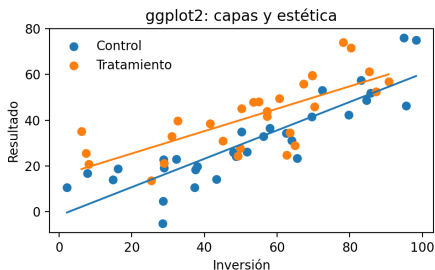
Una gramática de gráficos: construimos la figura por capas.

Salto a ggplot2: pensar en capas

Código

```
library(ggplot2)

ggplot(datos, aes(x = inversion,
                  y = resultado,
                  color = grupo))
  ) +
  geom_point(size = 2) +
  geom_smooth(method = "lm",
             se = FALSE) +
  labs(
    title = "Relación por grupo",
    ,
    x = "Inversión",
    y = "Resultado"
  )
```



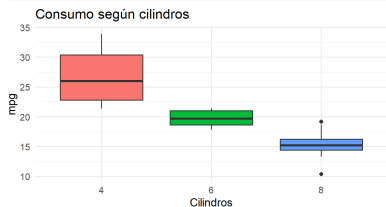
- Datos + estética + geometrías.
- Cada capa añade información.

Pequeños cambios, gran diferencia

```
ggplot(mtcars, aes(factor(cyl),
  mpg,
  fill = factor(
    cyl))) +
  geom_boxplot(show.legend =
    FALSE) +
  labs(
    title = "Consumo según
    cilindros",
    x = "Cilindros",
    y = "mpg"
  ) +
  theme_minimal(base_size = 14)
```

Mejoras sencillas

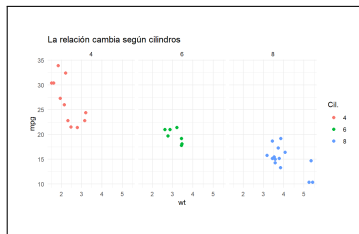
- tipografía legible
- menos elementos decorativos
- buen contraste
- etiquetas útiles
- color con significado



Contar más con facetas

```
ggplot(mtcars,
       aes(x = wt, y = mpg, color =
           factor(cyl))) +
  geom_point(size = 2) +
  facet_wrap(~ cyl) +
  theme_minimal() +
  labs(
    title = "La relación cambia según
n cilindros",
    color = "Cil."
  )
)
```

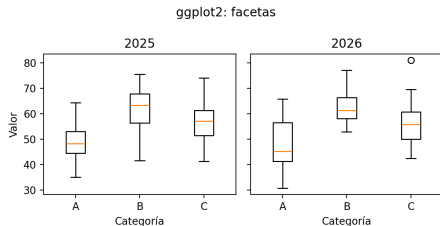
- Las facetas comparan grupos
- Reducen la sobrecarga visual
- Muy útiles en exploración



ggplot2 permite comparar mejor

Código

```
ggplot(datos,
       aes(x = categoria,
           y = valor)) +
  geom_boxplot() +
  facet_wrap(~ ano) +
  labs(
    title = "Comparación por ano",
    x = "Categoria",
    y = "Valor"
  )
```



- Las facetas ayudan a comparar sin mezclar.
- Muy útil cuando hay varios grupos o periodos.

Con R también hacemos mapas

Código

```
library(sf)
library(ggplot2)

mapa <- st_read("datos.shp")

ggplot(mapa) +
  geom_sf(aes(fill = valor)) +
  labs(title = "Mapa temático")
  +
  theme_void()
```

Mapas con sf + geom_sf()



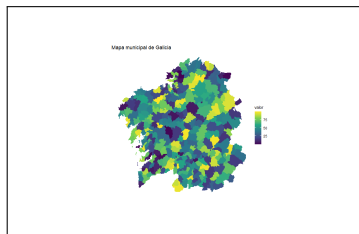
- El dato espacial también se analiza en R.
- Un mapa no es decoración: aporta contexto territorial.

Con R también hacemos mapas

```
library(mapSpain)
library(ggplot2)
# Municipios de Galicia
gal_muni <- esp_get_munic() %>%
  dplyr::filter(cpro %in% c("15", "
    27", "32", "36"))
# Variable ficticia
set.seed(123)
gal_muni$valor <- runif(nrow(gal_
  muni), 0, 100)

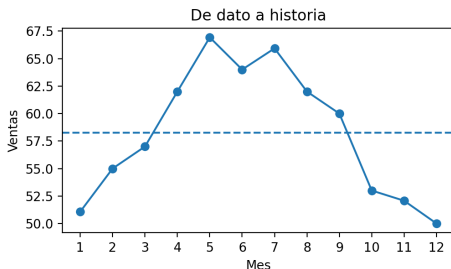
ggplot(gal_muni) +
  geom_sf(aes(fill = valor), color =
    NA) +
  scale_fill_viridis_c() +
  theme_void() +
  labs(title = "Mapa municipal de
    Galicia")
```

- R no solo hace barras y líneas
- Permite unir datos y territorio
- Muy útil en salud, medio ambiente, movilidad o administración pública



Del gráfico correcto a la historia correcta

- **No** se trata de poner muchos adornos.
- Se trata de responder una pregunta.
- Un buen título ya orienta la lectura.
- Color, etiquetas y escala deben ayudar, no distraer.



Ejemplo de enfoque

No decir sólo “ventas mensuales”, sino **“el pico de ventas llega en verano”**.

Buenas prácticas

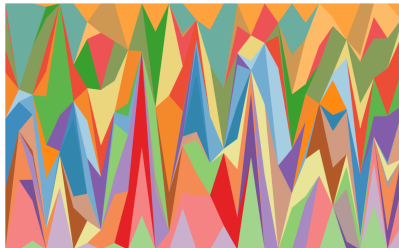
- 1 Elegir el gráfico según la pregunta
- 2 Quitar lo que no aporta
- 3 Usar títulos que cuenten algo
- 4 Destacar solo lo importante
- 5 Pensar siempre en la audiencia

En resumen

Con R podemos empezar simple, crecer rápido y llegar a visualizaciones muy potentes sin salir del mismo ecosistema.

De la visualización al data art

- R no solo permite hacer gráficos correctos.
- También puede producir visualizaciones muy llamativas.
- A veces, la frontera entre análisis y creación visual se difumina.



Yan Holtz, The R Graph Gallery

Idea

No siempre buscamos hacer arte, pero sí podemos aspirar a gráficos claros, memorables y que inviten a mirar.

Conclusiones

- 1 **Empieza simple:** si un barplot o un scatterplot sirven, úsalos.
- 2 **Refina con ggplot2:** cuando necesites capas, facetas, temas o mejor control visual.
- 3 **Piensa en el mensaje:** el gráfico debe ayudar a entender, no sólo a mirar.
- 4 Con R podemos pasar de un gráfico básico a una visualización potente sin salir del mismo ecosistema.

Idea final

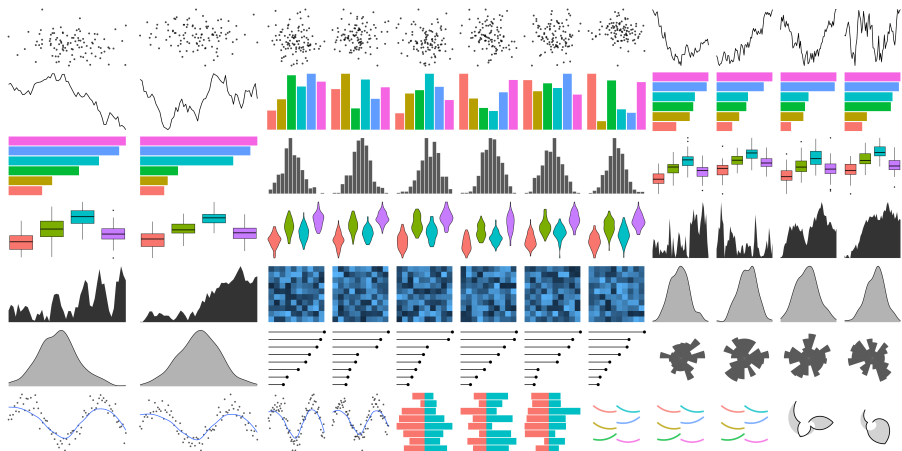
Los gráficos no son el final del análisis: son el puente entre los datos y las decisiones.

Recursos para seguir explorando

- R Graph Gallery
- DataViz Catalogue
- Documentación de ggplot2
- Ejemplos reproducibles en R

Idea final

Visualizar no es decorar: es ayudar a pensar, entender y comunicar.



Idea final

Un buen gráfico no solo se entiende: también se recuerda.

DE DATOS A HISTORIAS: CREA GRÁFICOS IMPACTANTES EN R

María José Ginzo Villamayor

Universidade de Santiago de Compostela
Centro de Investigación e Tecnoloxía Matemática de Galicia

17 de abril de 2026