



Sesión 4

Visualización de datos en R

Campamento de invierno EGOB | UC | 04 de agosto, 2023

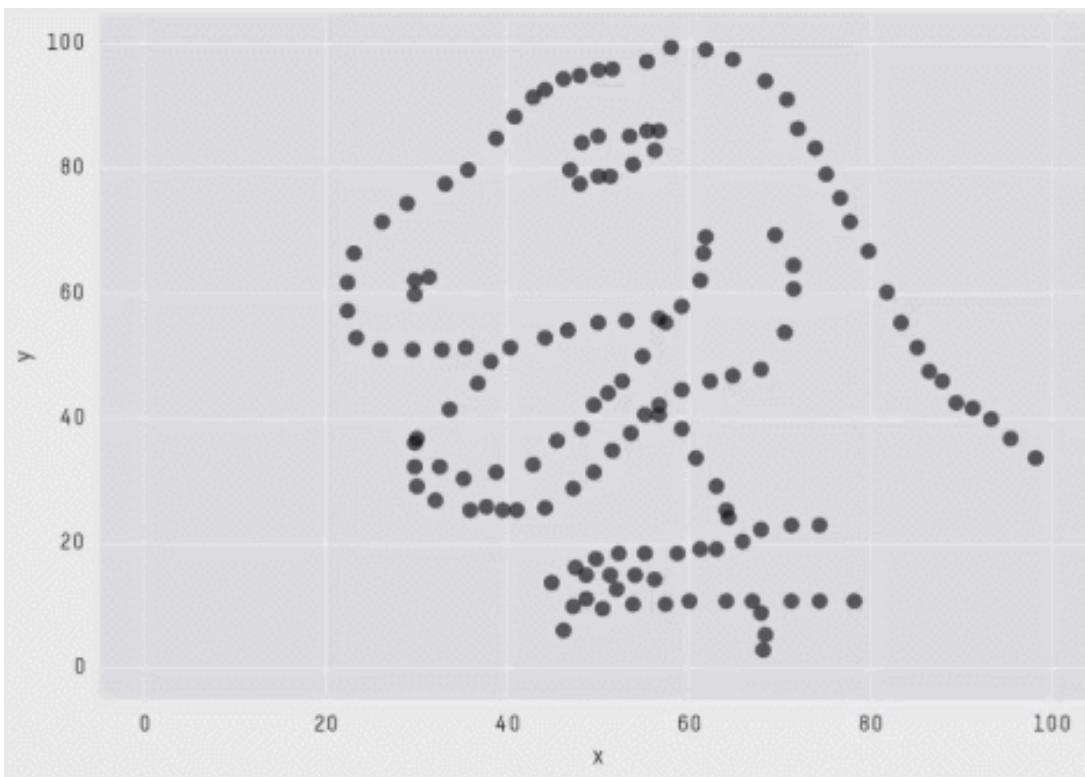
 **José D. Conejeros** |  jdconejeros@uc.cl

Guía

1. ¿Qué es un gráfico?
2. Buenos y malos gráficos
3. Algunos criterios de guía
4. Herramientas para visualizar datos
5. ggplot2: uso de capas
6. Construcción de visualizaciones básicas:
barras, líneas y puntos



Mismas estadísticas, diferentes datos

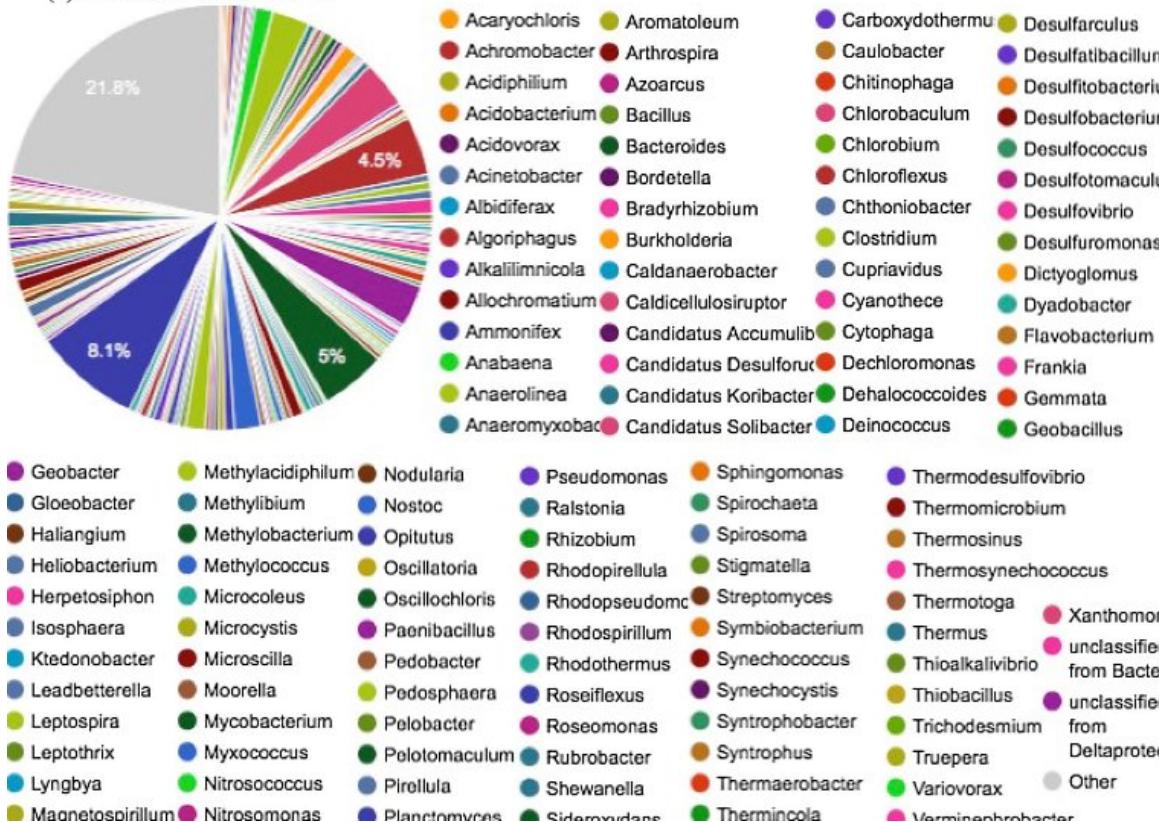


X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

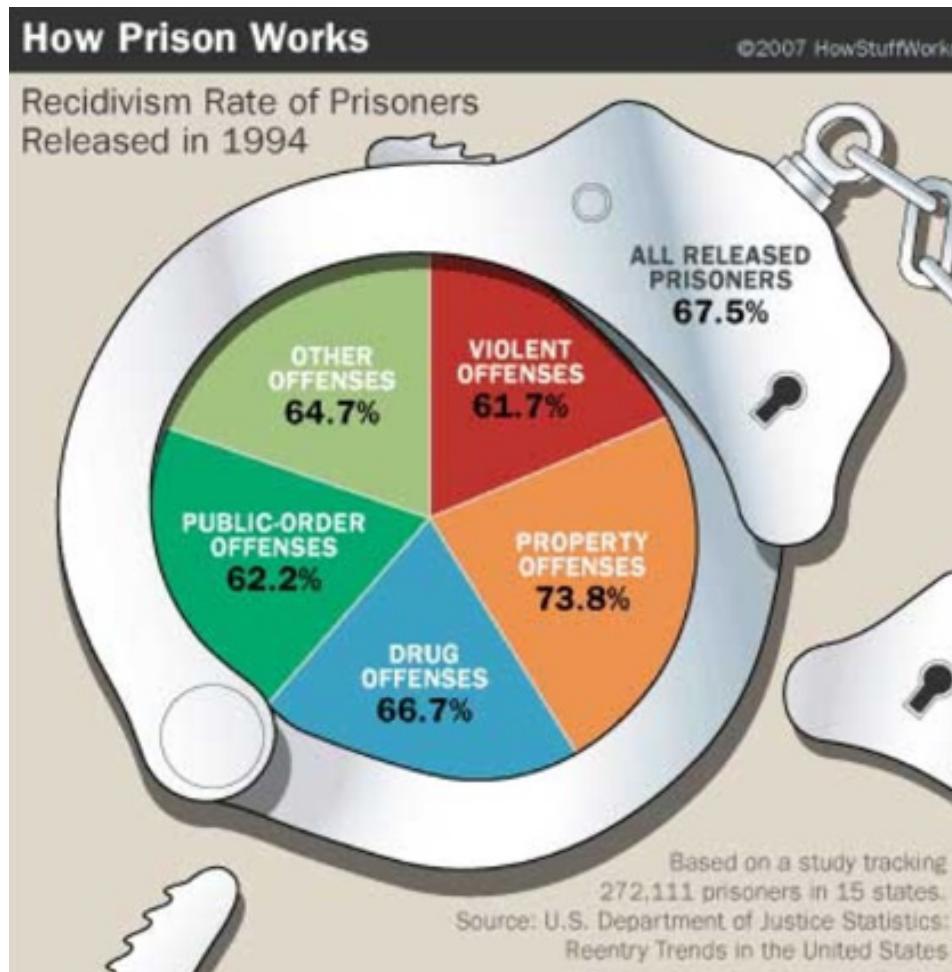
Matejka, J., & Fitzmaurice, G. (2017, May). Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In Proceedings of the 2017 CHI conference on human factors in computing systems (pp. 1290-1294).

Algunas visualizaciones de datos

(f) Distribution of Genus



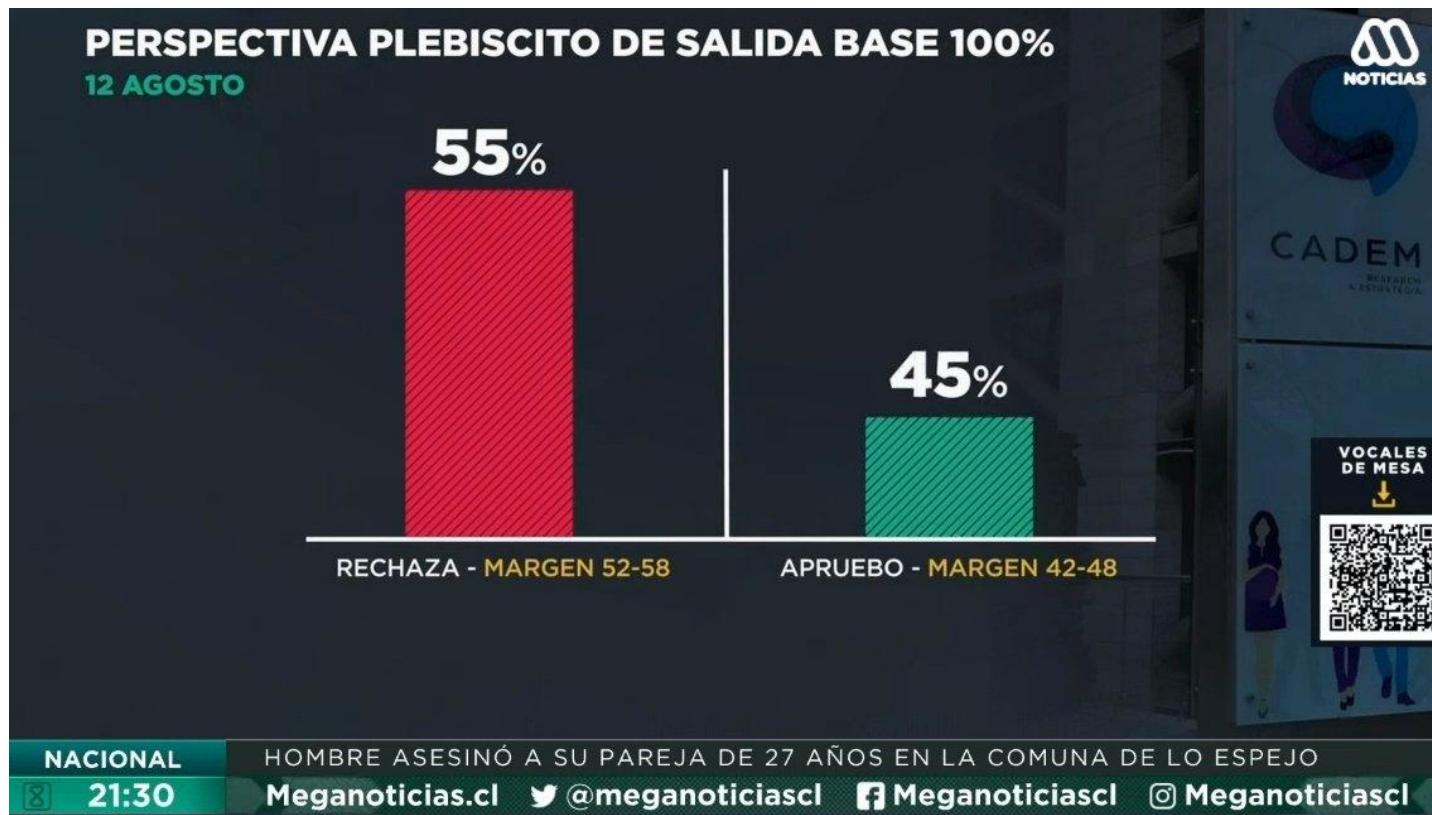
Algunas visualizaciones de datos



Algunas visualizaciones de datos



Algunas visualizaciones de datos



Algunas visualizaciones de datos



Problemas de estos gráficos

- **Estéticos:** dimensiones, colores y formas



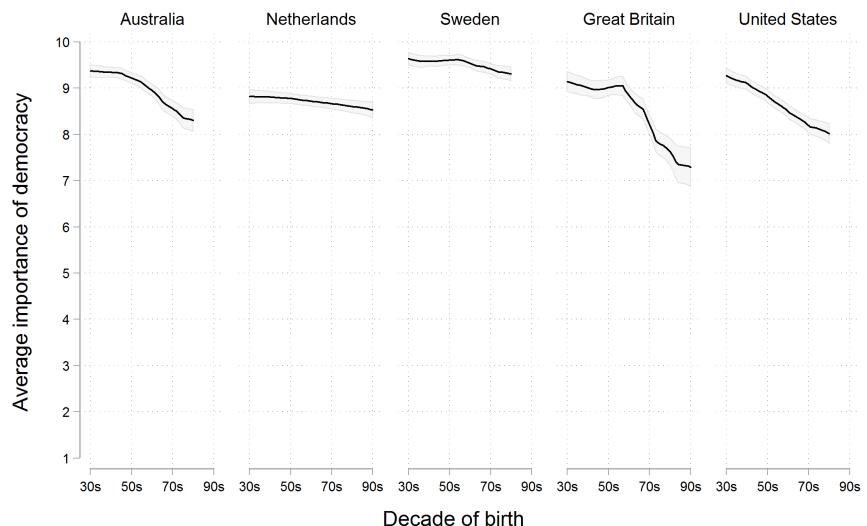
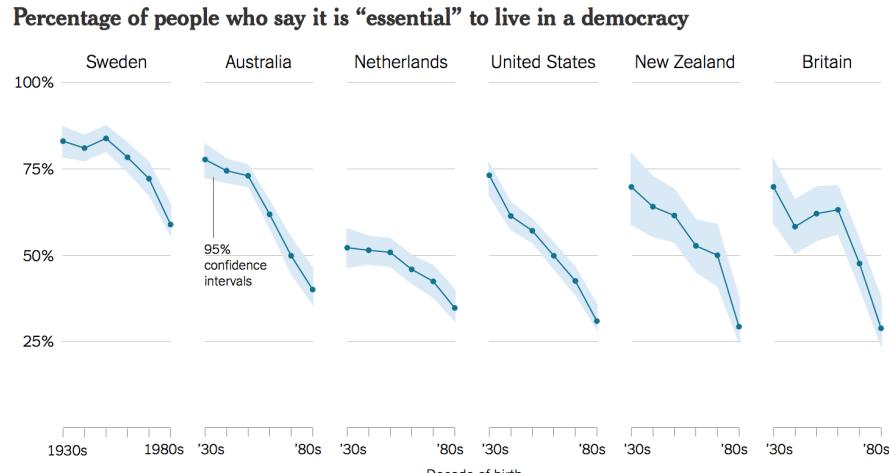
Problemas de estos gráficos

- **Estéticos:** dimensiones, colores y formas



Problemas de estos gráficos

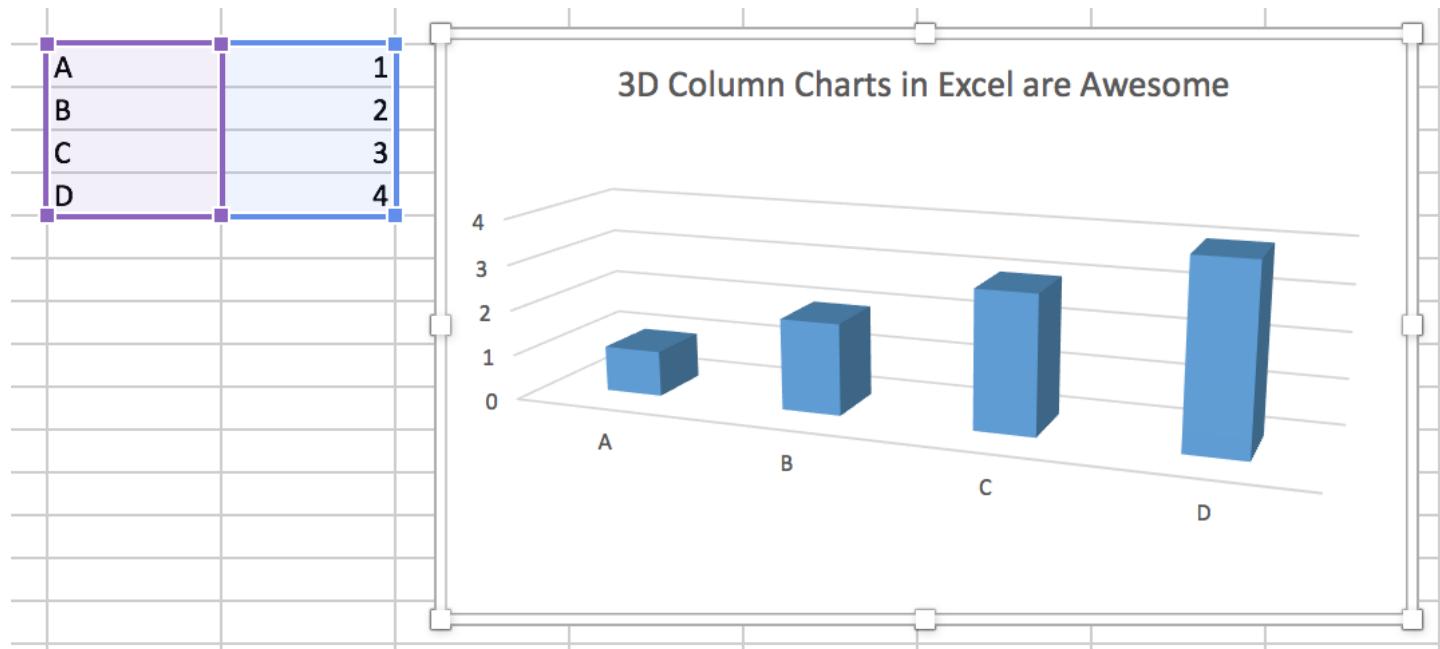
- **Sustantivos:** provenientes de la generación de datos



Taub, A. (2016). How stable are democracies? 'Warning signs are flashing red'. *New York Times*, 29(11), 2016.

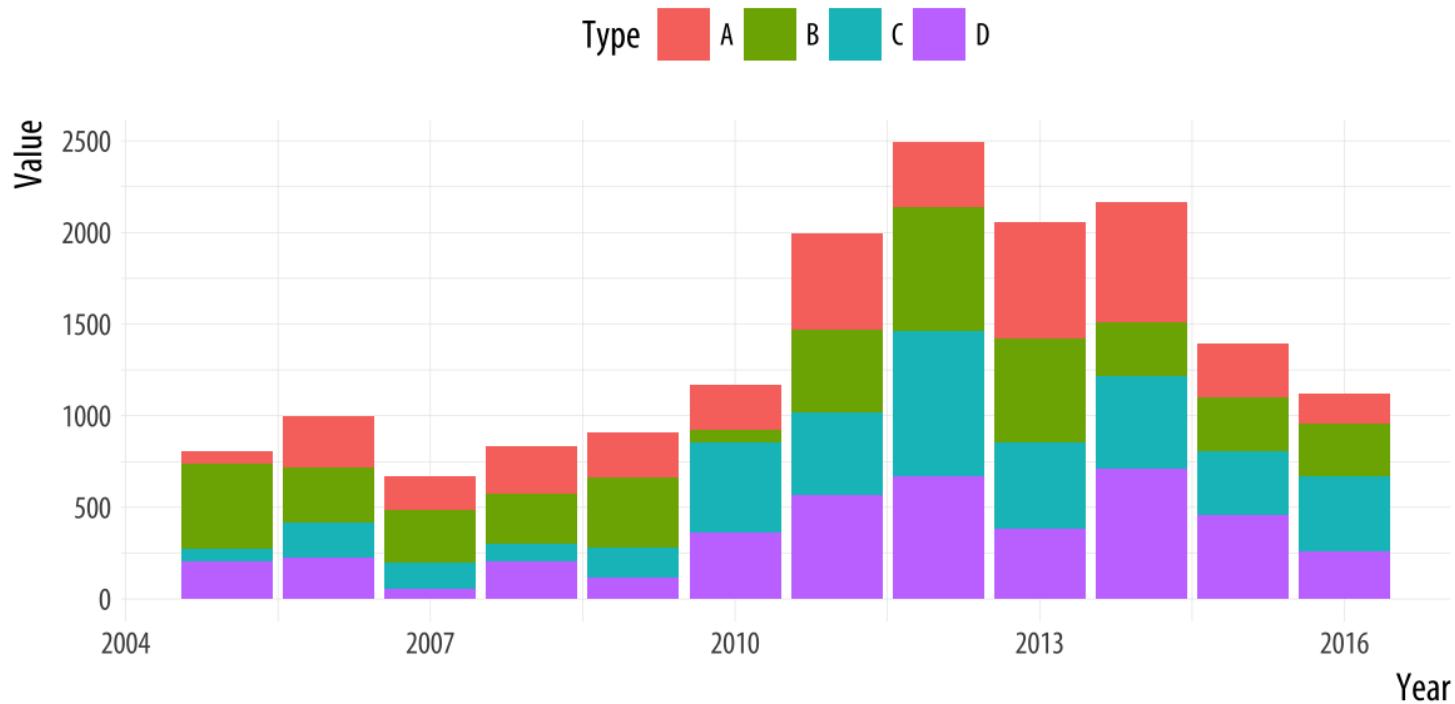
Problemas de estos gráficos

- **Percepción:** cómo las personas perciben y procesan lo que están mirando. Las visualizaciones codifican números en líneas, formas y colores. Eso significa que nuestra interpretación de estas codificaciones está parcialmente condicionada a cómo percibimos las formas y relaciones geométricas en general.



Problemas de estos gráficos

- **Percepción:** cómo las personas perciben y procesan lo que están mirando. Las visualizaciones codifican números en líneas, formas y colores.



Buenas visualizaciones

La excelencia gráfica es la presentación bien diseñada de datos interesantes: **una cuestión de sustancia, de estadísticas y de diseño**... [Consiste] en ideas complejas comunicadas con claridad, precisión y eficiencia. ... [Es] lo que le da al espectador **la mayor cantidad de ideas en el menor tiempo posible con la menor cantidad de tinta en el espacio más pequeño** ... [Es] casi siempre multivariante ... Y la excelencia gráfica requiere decir la verdad sobre los datos. (Tufte, 1983, pág. 51).

Tufte, E. R. (1985). The visual display of quantitative information. The Journal for Healthcare Quality (JHQ), 7(3), 15.



Algunas visualizaciones de datos son mejores que otras

Si bien es tentador simplemente comenzar a establecer la ley sobre lo que funciona y lo que no, el proceso de hacer un gráfico realmente bueno o realmente útil no puede reducirse a una lista de reglas simples que deben seguirse sin excepción en todas las circunstancias.

Los gráficos que haces están destinados a ser vistos por alguien. La efectividad de cualquier gráfico en particular no es solo una cuestión de cómo se ve en abstracto, sino también **una cuestión de quién lo está mirando y por qué**.

Una imagen destinada a una audiencia de expertos que lea una revista profesional puede no ser fácilmente interpretable por el público en general. Una visualización rápida de un conjunto de datos que está explorando actualmente podría no ser de mucha utilidad para sus compañeros o estudiantes.



Buenas visualizaciones

Buena estética

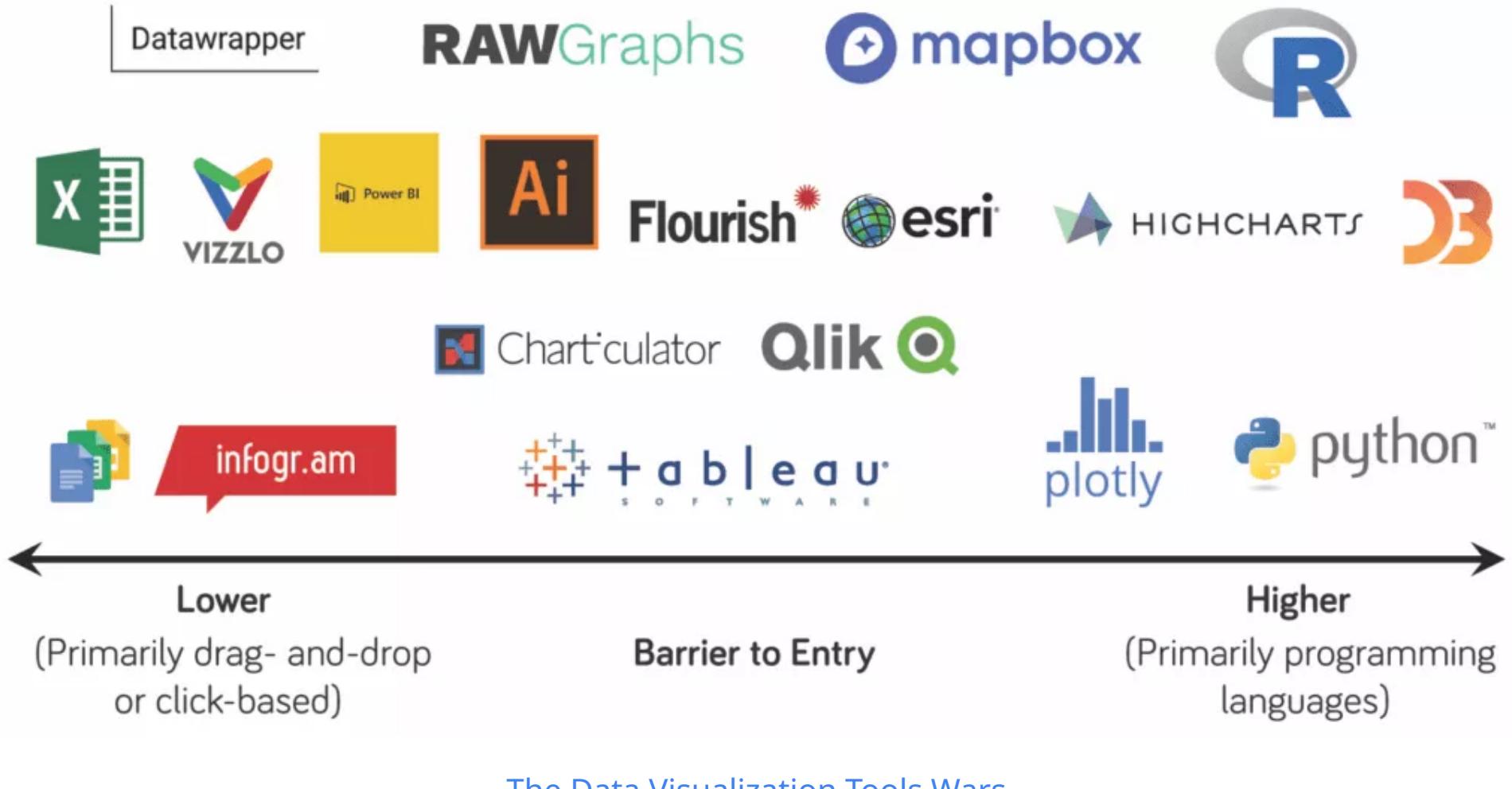
Sin problemas sustantivos

Sin problemas de percepción

Identificar un público objetivo

Honestidad y buen juicio

Tecnologías para visualizar datos



Construcción de piezas gráficas

¿Qué es un gráfico?

Un gráfico es una representación visual que resume datos estadísticos, de manera tal que podamos interpretarlos, analizarlos y entenderlos de forma más sencilla.

¿Cuándo hacemos un gráfico?

- Cuando deseamos realizar un análisis exploratorio de datos, por ejemplo, descubrir el comportamiento o distribución variable.
- Cuando necesitamos exhibir los resultados de un análisis al público, en este caso solemos prestarle más atención a la estética de los gráficos.



Ejemplos



World Health Organization

Search by Country, Territory, or Area

Covid-19 Response Fund



Donate

WHO Coronavirus (COVID-19) Dashboard

Overview

Measures

Table View

Data

More Resources

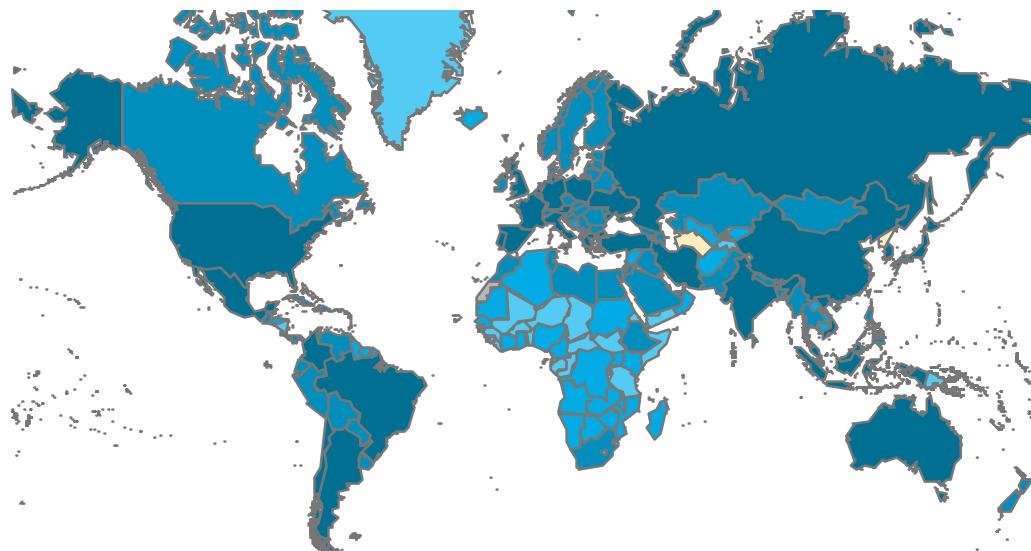
Cases

Total

352,943
new cases last 7 days

768,983,095
cumulative cases

6,953,743
cumulative deaths



Cases - Total	X
> 5,000,000	
500,001 – 5,000,000	
50,001 – 500,000	
5,001 – 50,000	

Globally, as of **1:56pm CEST, 2 August 2023**, there have been **768,983,095 confirmed** including **6,953,743 deaths**, reported to WHO. As of **30 July 2023**, a total of **13,492,099,75**

Ejemplos

Otros ejemplos

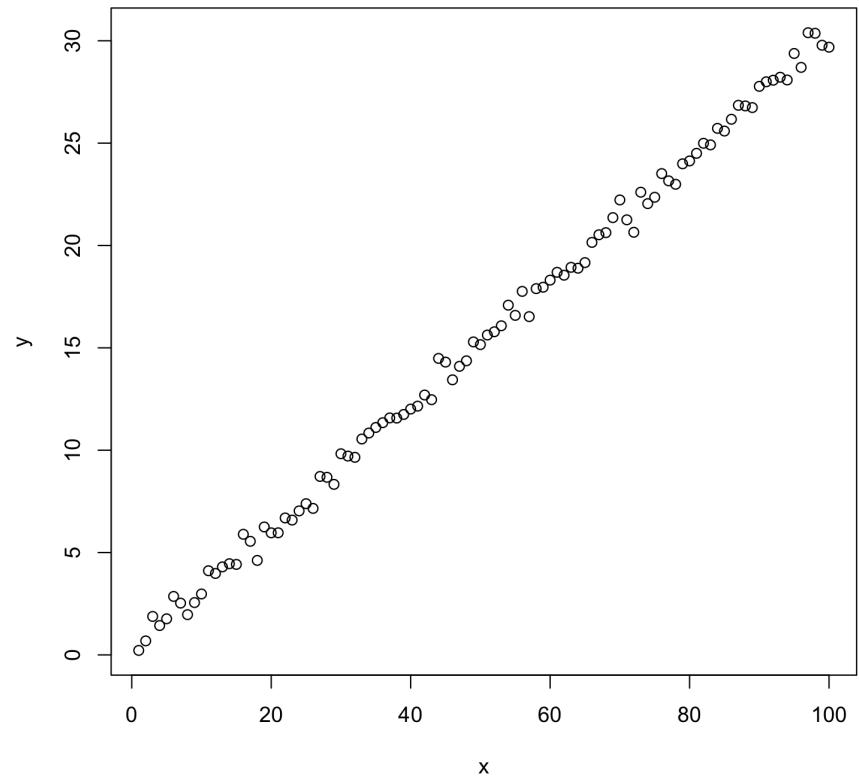
- Tutoriales:
 - [A ggplot2 Tutorial for Beautiful Plotting in R](#)
 - [Ggplot evolution](#)
- Ministerio de desarrollo social: [Data Social](#)
- The World Bank: [DataBank](#)
- Datos de miércoles:
 - [Latinoamericano](#)
 - [Anglo](#)
- Comunidad:
 - [#30díasdegráficos](#)
 - [#TidyTuesday](#)
 - [#ggplot2](#)
- Algunas aplicaciones entretenidas:
 - [Photos on spirals](#)
 - [Gerative art](#)
 - [Otros ejemplos](#)

Graficando con R Base

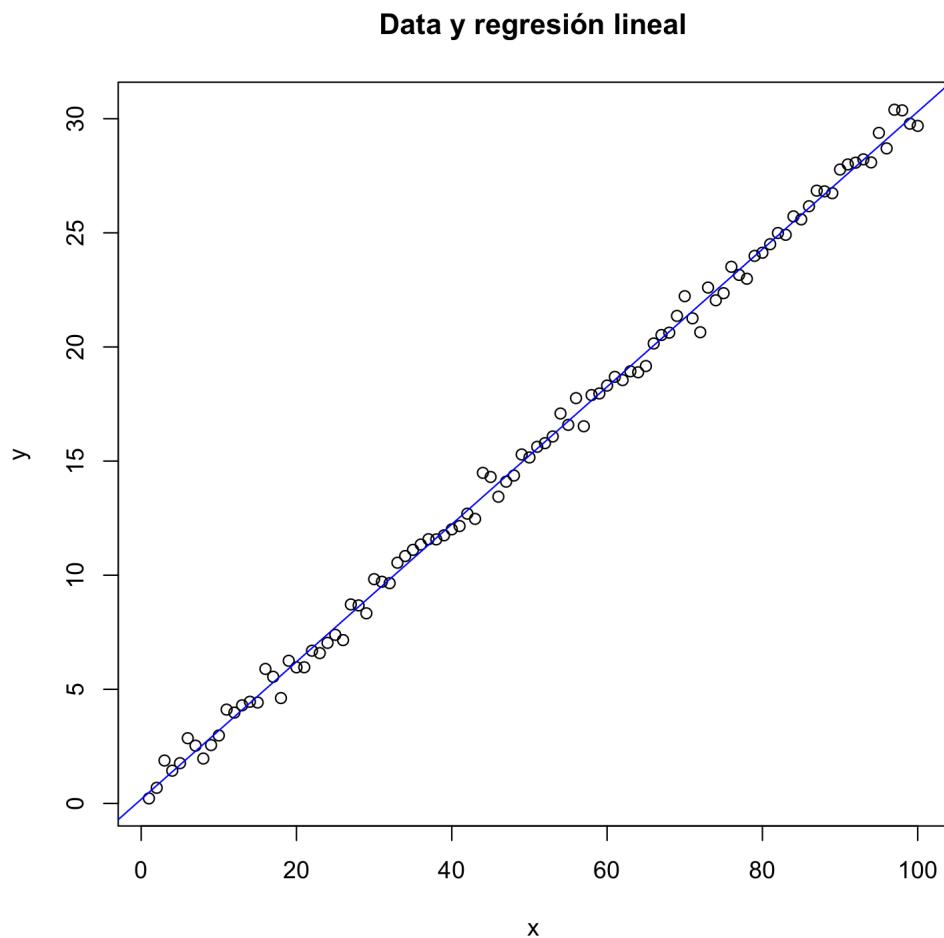
```
# Definimos Vectores
x ← 1:100
# Escalares
n ← length(x); a ← 0.2; b ← 0.3; sigma
set.seed(123) # Semilla aleatoria
y ← a+b*x+sigma*rnorm(n) # Simulamos una
head(cbind(x,y), n=15)
```

	x	y
[1,]	1	0.2197622
[2,]	2	0.6849113
[3,]	3	1.8793542
[4,]	4	1.4352542
[5,]	5	1.7646439
[6,]	6	2.8575325
[7,]	7	2.5304581
[8,]	8	1.9674694
[9,]	9	2.5565736
[10,]	10	2.9771690
[11,]	11	4.1120409
[12,]	12	3.9799069

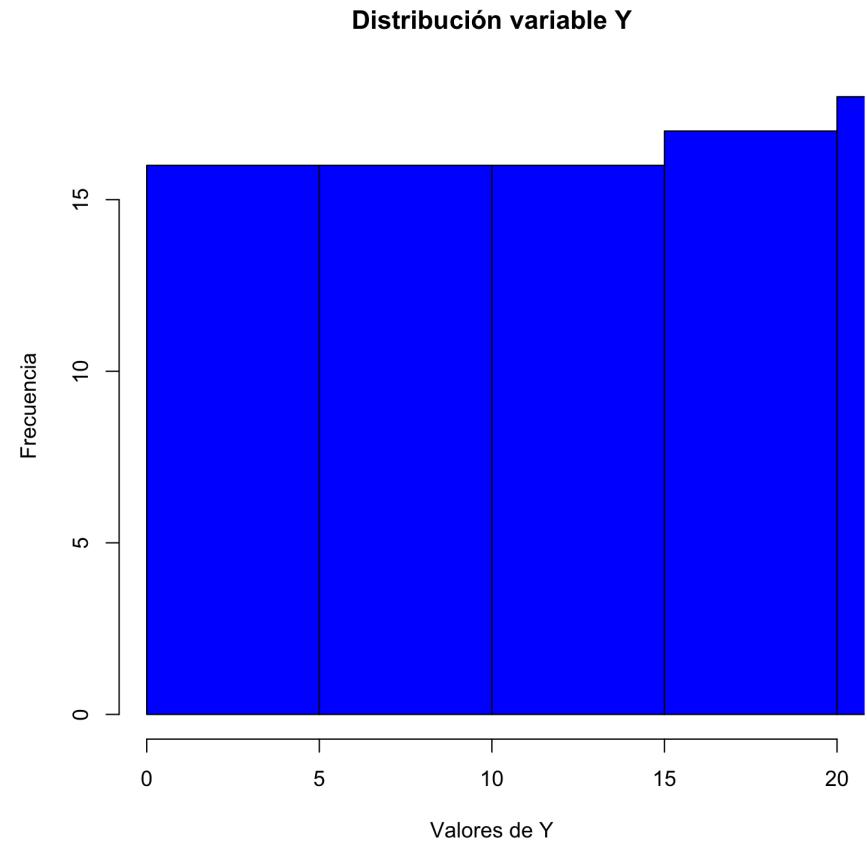
```
plot(x, y)
```



```
plot(x, y, main="Data y regresión  
abline(a_hat, b_hat, col="blue")
```



```
hist(y, main = "Distribución variable Y",
na.rm=T,
xlab="Valores de Y",
ylab="Frecuencia",
col="blue",
xlim=c(0, 20))
```



Gráficos en R

R base tiene herramientas gráficas limitadas, tanto en la cantidad de opciones que se tiene como en su personalización. Por lo que usaremos los gráficos de [ggplot2](#). Estas herramientas, permiten generar una gran cantidad de gráficos a partir de la misma base computacional.

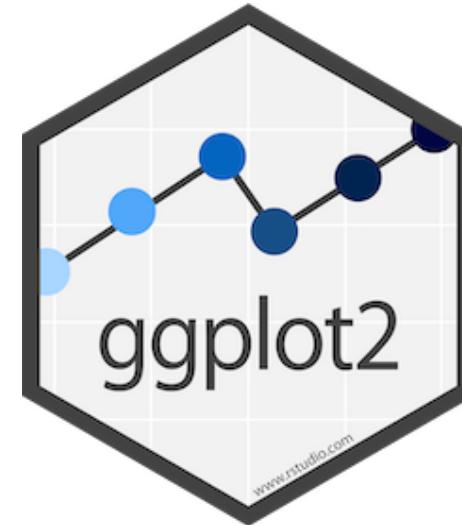
Gráfico	R base	ggplot2
Puntos	<code>plot()</code>	<code>geom_point()</code>
Lineas	<code>plot(... , type = "l")</code>	<code>geom_line()</code>
Histograma	<code>hist()</code>	<code>geom_histogram()</code>
Barras	<code>barplot()</code>	<code>geom_bar()</code>
Boxplot	<code>boxplot()</code>	<code>geom_boxplot()</code>



Ggplot2

Es un sistema para crear gráficos de forma declarativa, basado en [La Gramática de los Gráficos](#). Usted proporciona los datos, le dice a ggplot2 cómo asignar variables a la estética, qué primitivas gráficas utilizar, y él se encarga de los detalles.

Es un sistema coherente para describir y construir gráficos, combinando componentes independientes. Es una herramienta estable con 10 años en funcionamiento.



```
install.packages("ggplot2")  
library(ggplot2)
```

Gramática de los gráficos en capas

Añadimos características a los gráficos paso a paso

Describes all the non-data ink	Theme
Plotting space for the data	Coordinates
Statistical models & summaries	Statistics
Rows and columns of sub-plots	Facets
Shapes used to represent the data	Geometries
Scales onto which data is mapped	Aesthetics
The actual variables to be plotted	Data



La clave que las capas se van sumando: "+"

Componentes de un gráfico

1. Datos a utilizar
2. Paramétros estéticos con que se registrarán las variables, es decir, cómo se asignarán las variables de nuestro conjunto de datos a ciertas propiedades visuales. Esto considera ejes de gráficos, colores, etc. La función para indicar esto es `aes()` (del inglés *aesthetics*).
3. Una capa que indique la forma en que se representarán gráficamente los datos (con la función `geom_*()`).

```
ggplot(data=Datos, aes(MAPEOS)) +
```

```
  geom_TIPO_DE_GRAFICO(...) +
```

```
Otras capas
```



Tipos de gráficos

ONE VARIABLE continuous

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



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



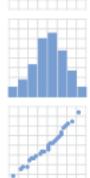
c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight



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

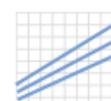
```
e <- ggplot(mpg, aes(cty, hwy))
```



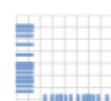
e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust



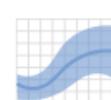
e + geom_point()
x, y, alpha, color, fill, shape, size, stroke



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



Tipos de gráficos

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

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

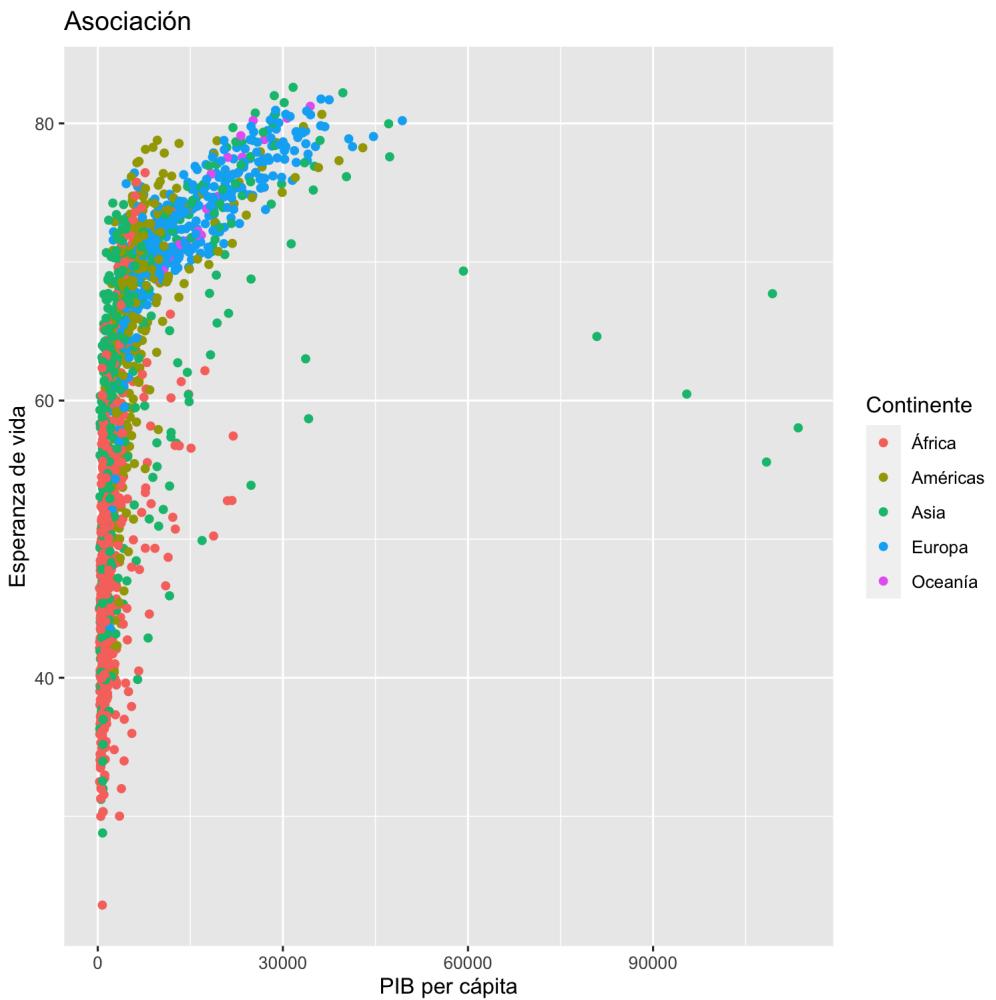


f + geom_dotplot(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill, group



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

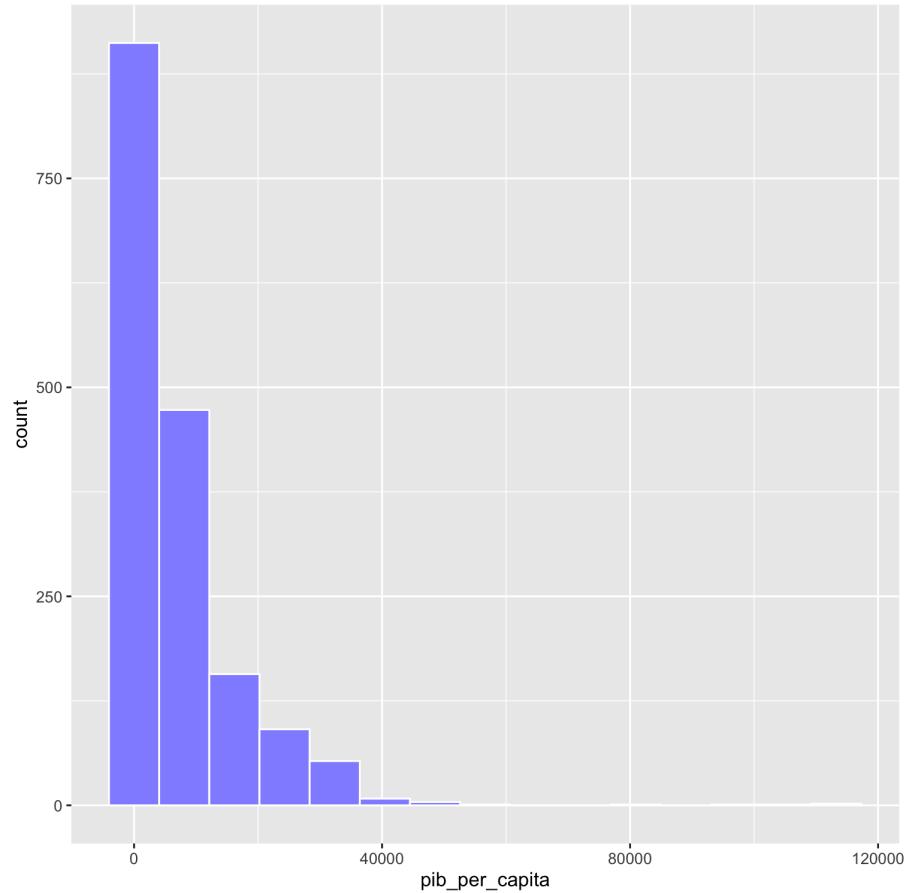
```
ggplot(paises, aes(x = pib_per_capita,
                    y = esperanza_vida,
                    colour = continente)) +
  geom_point() +
  labs(title = "Asociación",
       x = "PIB per cápita",
       y = "Esperanza de vida",
       colour='Continente')
```



Tipos de gráficos

Histogramas: Se usa para visualizar la distribución de los valores de una variable numérica.

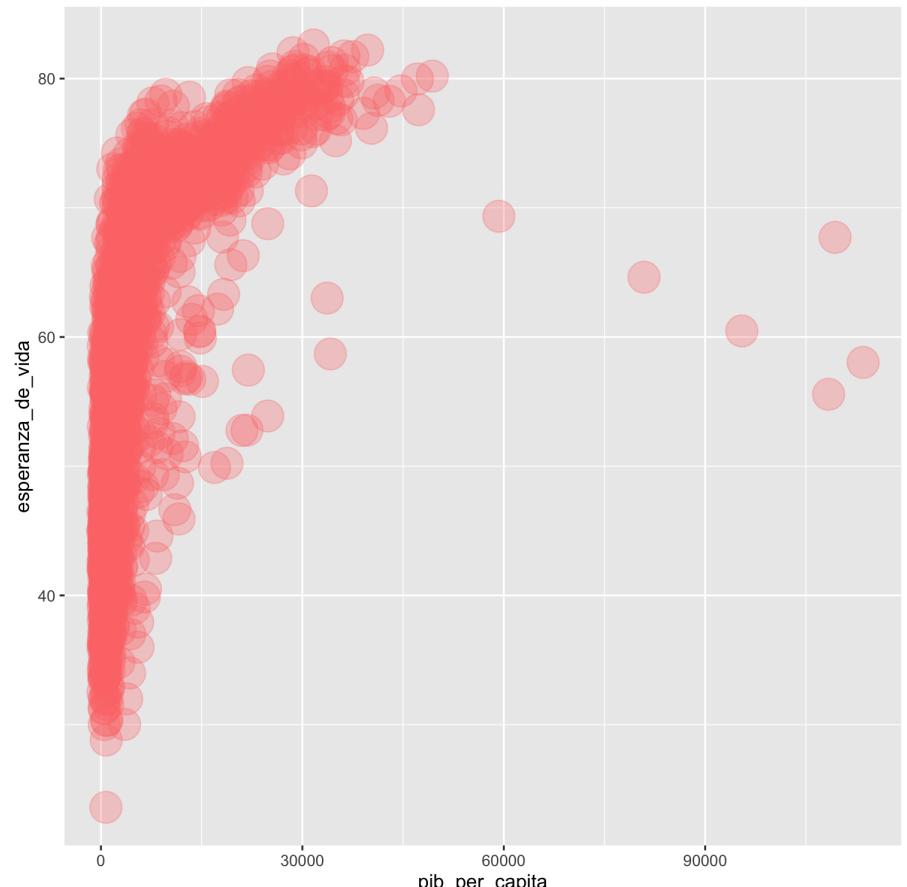
```
ggplot(data = paises,  
       mapping = aes(x = pib_per_capita))  
geom_histogram(color = "white",  
               fill = "#9292ff",  
               bins = 15)
```



Tipos de gráficos

Gráficos de puntos: Se usa para encontrar relaciones o patrones entre dos variables (al menos una debe ser numérica).

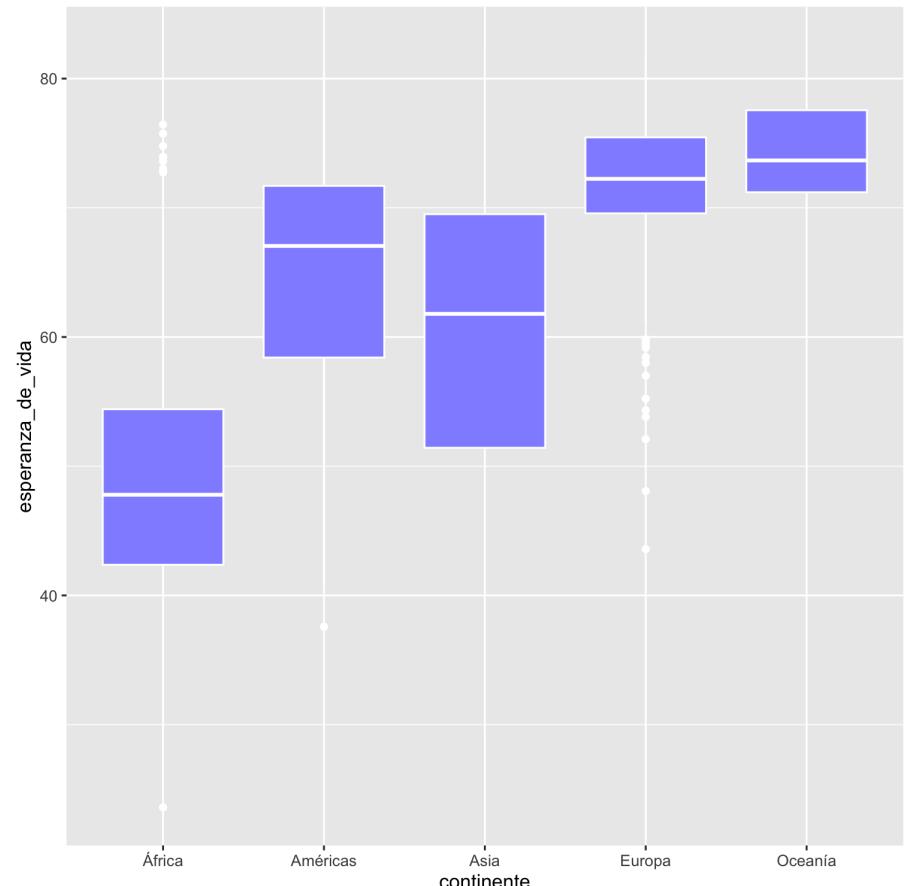
```
ggplot(data = paises, mapping = aes(x = p  
y = e  
geom_point(color = "#ff7979",  
size = 8,  
alpha = 0.3)
```



Tipos de gráficos

Boxplots: Para observar la distribución de una variable numérica. Muy útil para comparar las distribuciones por grupo.

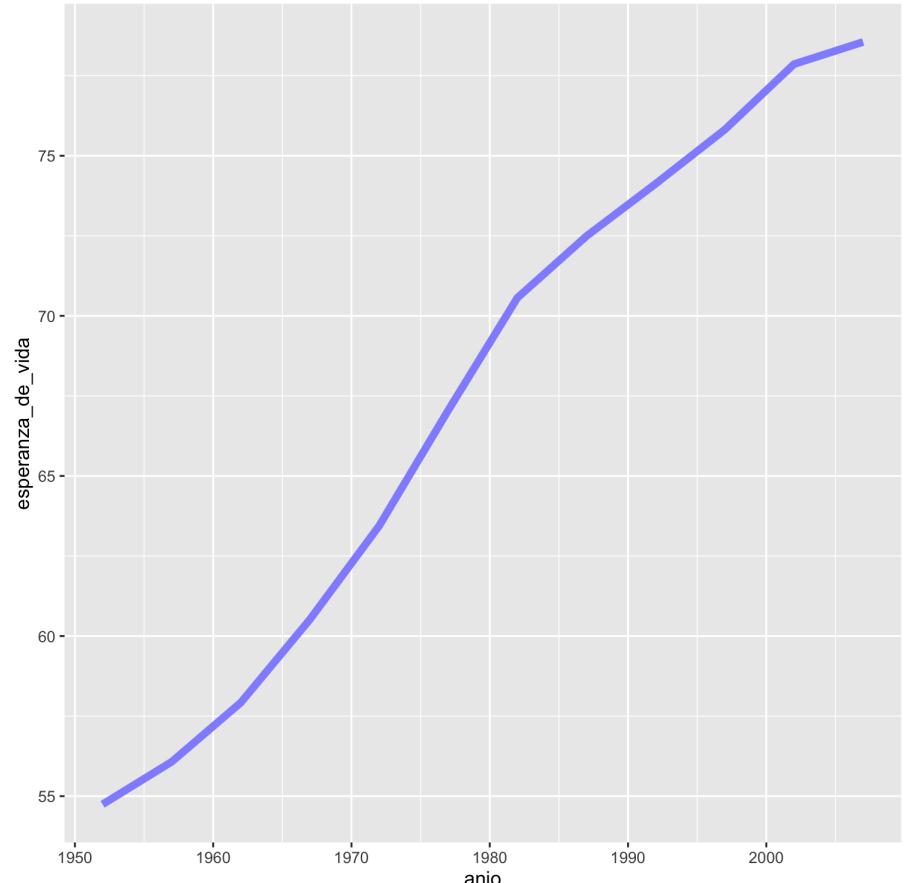
```
ggplot(data = paises, mapping = aes(x = c  
y = e  
geom_boxplot(color = "blue",  
fill = "#9292ff")
```



Tipos de gráficos

Líneas: Se usa para analizar tendencias de una variable numérica en el tiempo. En el ejemplo, se presenta la evolución de la esperanza de vida en Chile desde 1960.

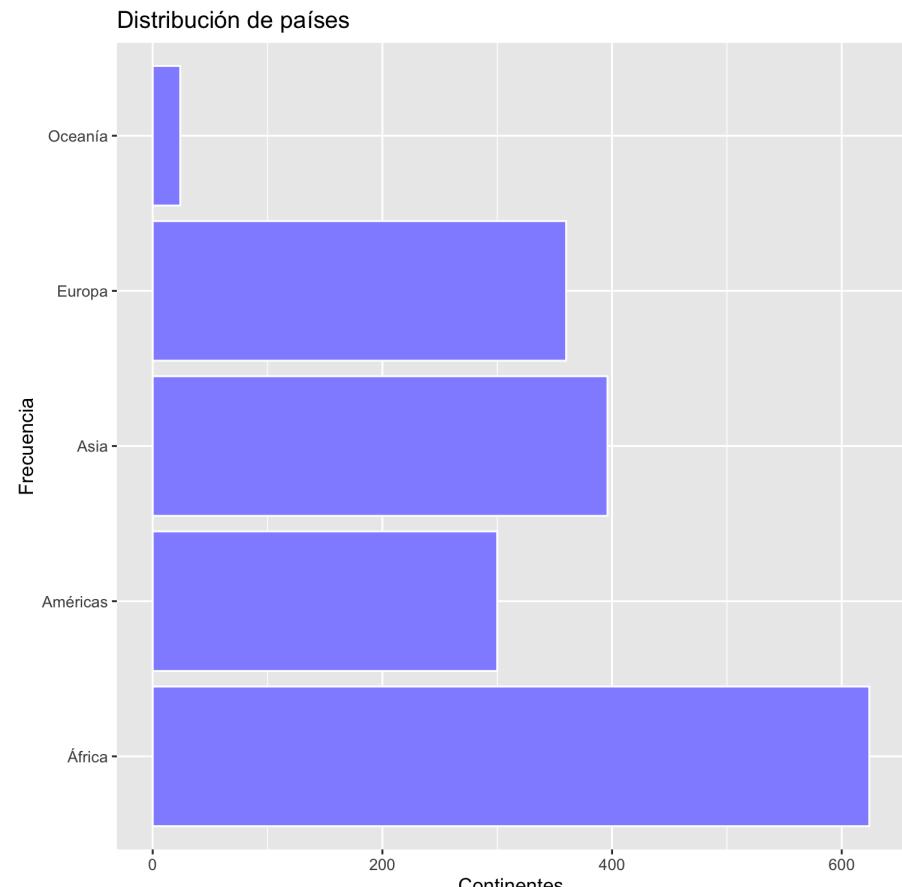
```
ggplot(data = paises[paises$pais == "Chile",]  
       aes(x = anio, y = esperanza_de_vida)) +  
  geom_line(color = "#9292ff", size = 2)
```



Tipos de gráficos

Barras: Se usa para conocer la frecuencia relativa o absoluta de las clases de una variable categórica o discreta.

```
ggplot(data = paises,  
       mapping = aes(y=continente)) +  
  geom_bar(color = "white",  
           fill = "#9292ff") +  
  labs(title = "Distribución de países",  
       x = 'Continentes',  
       y = 'Frecuencia')
```



Tipos de gráficos

The R Graph Gallery



Welcome to the R graph gallery, a collection of charts made with the [R programming language](#). Hundreds of charts are displayed in several sections, always with their reproducible code available. The gallery makes a focus on the tidyverse and [ggplot2](#). Feel free to contribute your own charts!

Contributions are welcome! If you have a chart you'd like to share, or if you'd like to help maintain the gallery, please [get in touch](#).

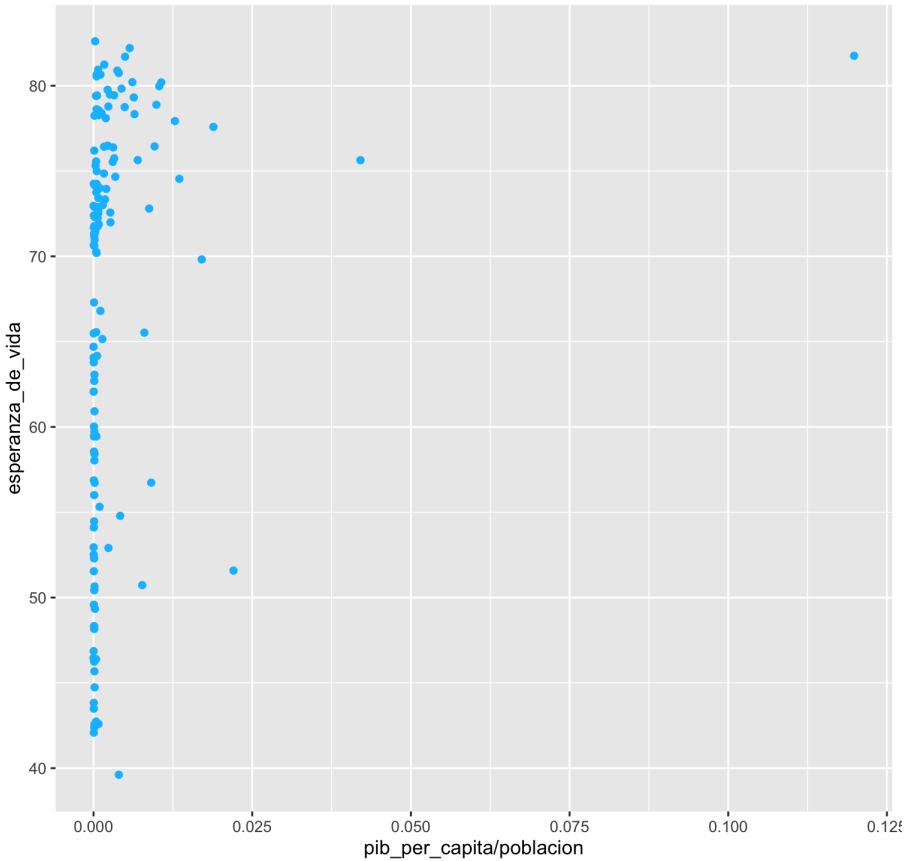


Customización de gráficos en ggplot2

Un color para todo el gráfico

```
paises %>%
  filter(anio = 2007) %>%
  ggplot(aes(x=pib_per_capita / poblacion
             y=esperanza_de_vida)) +
  geom_point(color = 'deepskyblue') +
  labs(title = 'Relación entre PIB per cá
```

Relación entre PIB per cápita y esperanza de vida

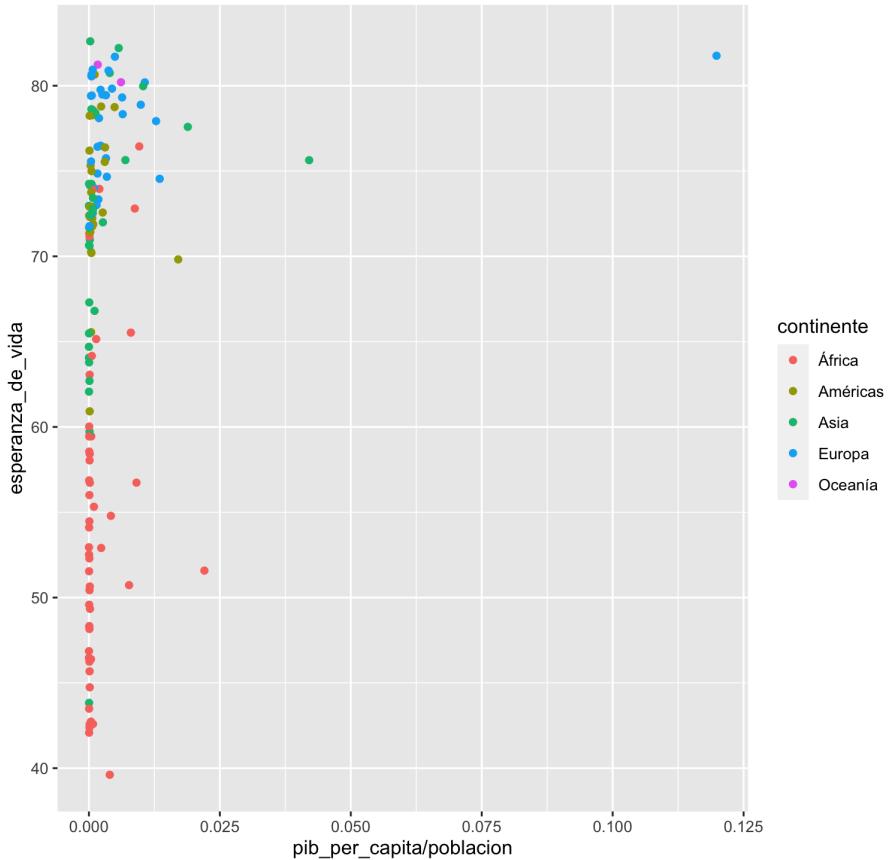


Customización de gráficos en ggplot2

Colores según una variable

```
paises %>%
  filter(anio = 2007) %>%
  ggplot(aes(x=pib_per_capita / poblacion
             y=esperanza_de_vida)) +
  geom_point(aes(color = continente)) +
  labs(title = 'Relación entre PIB per cá
```

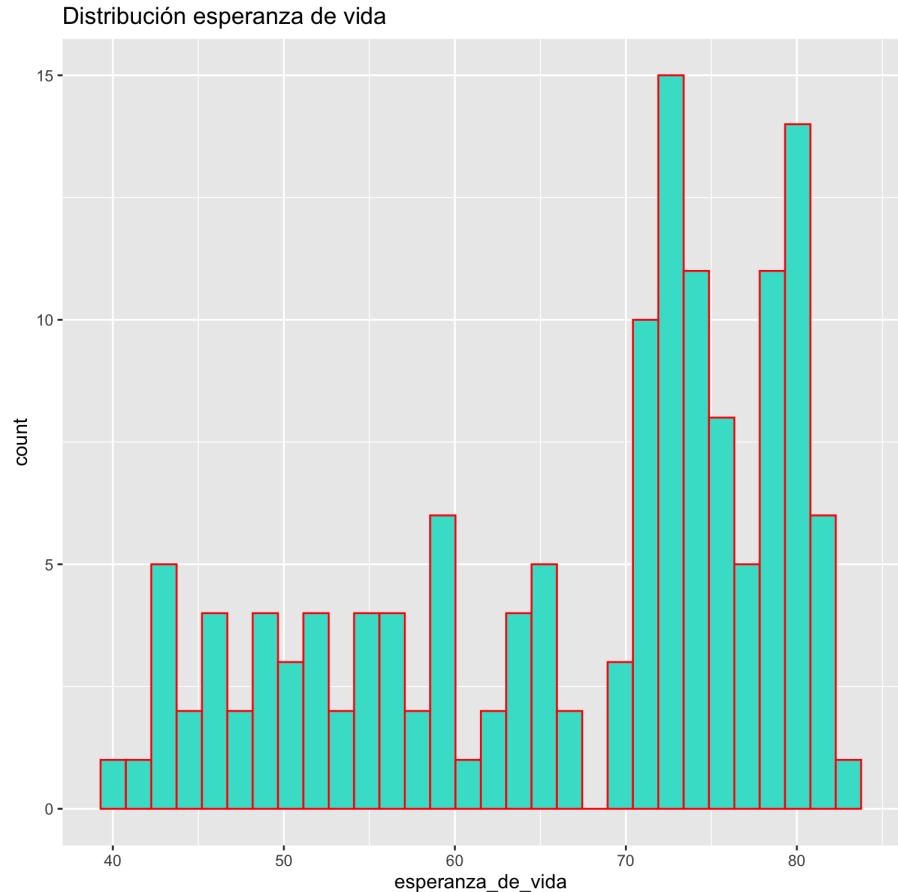
Relación entre PIB per cápita y esperanza de vida



Customización de gráficos en ggplot2

Colores en gráficos de barras e histogramas

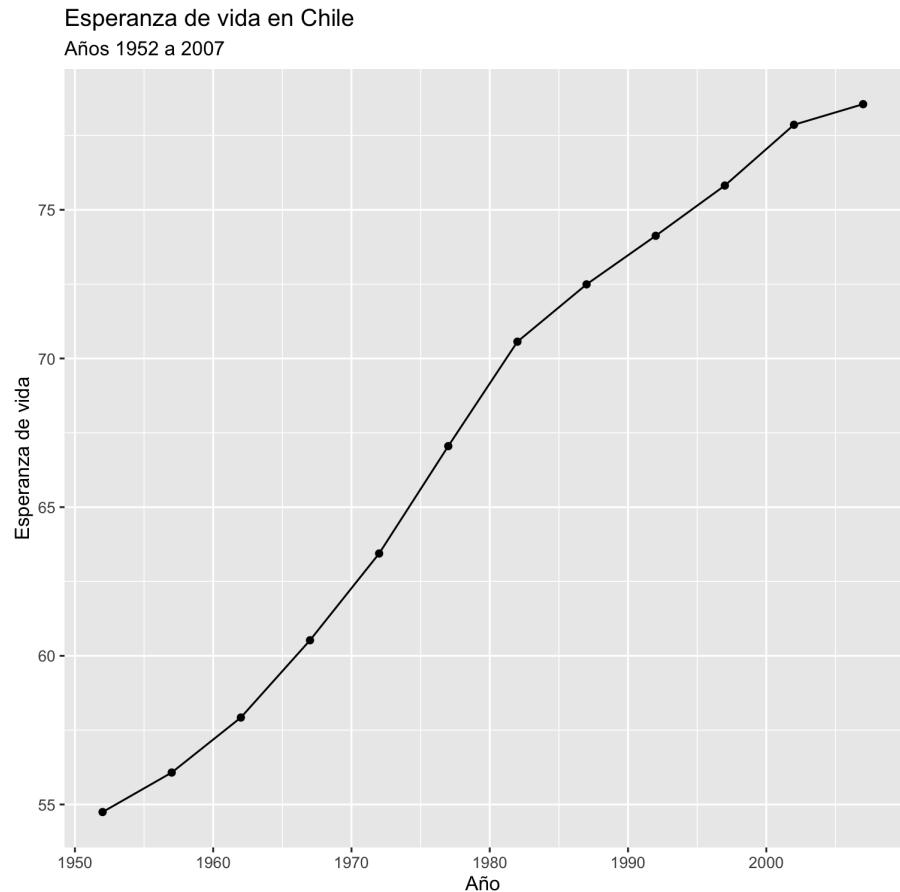
```
paises %>%
  filter(anio = 2007) %>%
  ggplot(aes(esperanza_de_vida)) +
  geom_histogram(fill = 'turquoise',
                 color = 'red') +
  labs(title = "Distribución esperanza de vida")
```



Customización de gráficos en ggplot2

Ejes y títulos

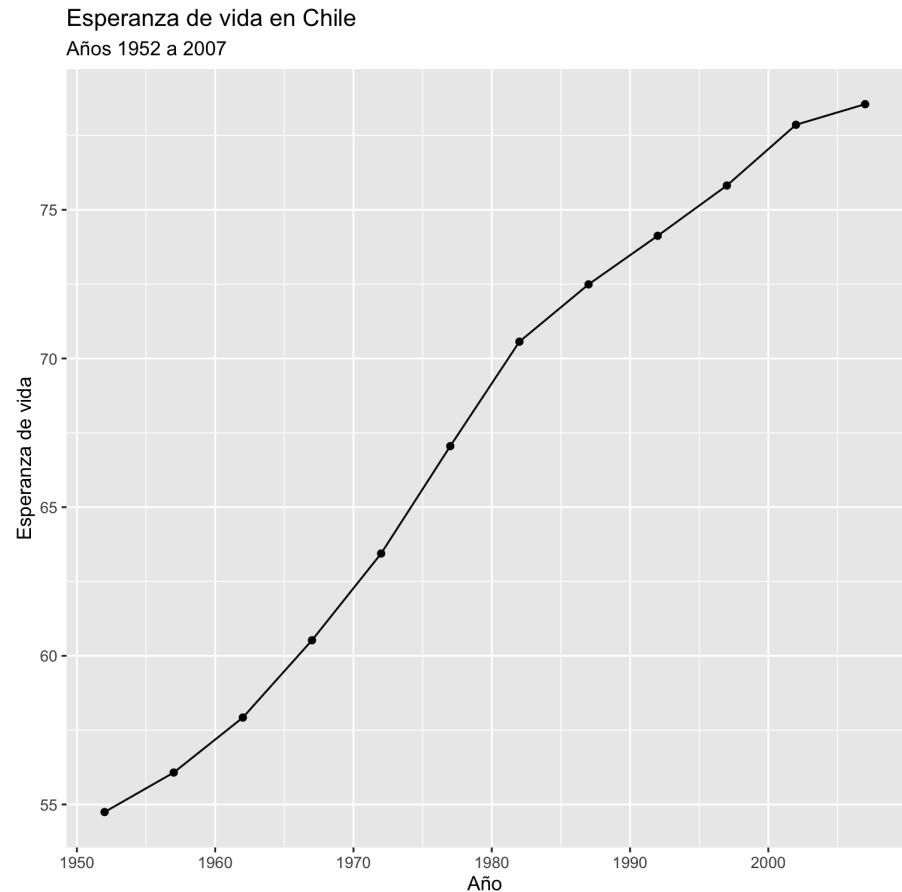
```
ggplot(data = paises[paises$pais == "Chile"],  
       aes(x = anio, y = esperanza_de_vida)) +  
  geom_line() +  
  geom_point() +  
  labs(title = "Esperanza de vida en Chile",  
       subtitle = "Años 1952 a 2007",  
       x = "Año",  
       y = "Esperanza de vida")
```



Customización de gráficos en ggplot2

Ejes y títulos (forma 2)

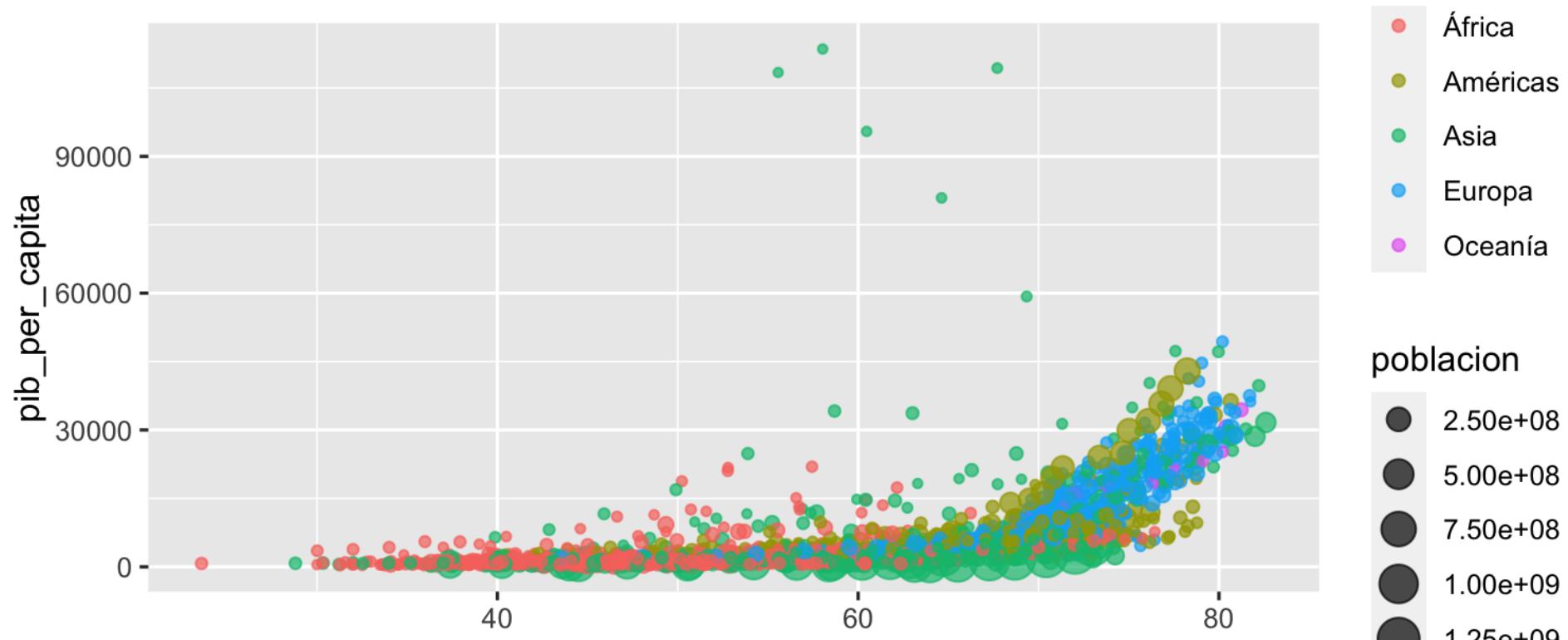
```
ggplot(data = paises[paises$pais == "Chile"],  
       aes(x = anio, y = esperanza_de_vida)) +  
  geom_line() +  
  geom_point() +  
  ggttitle(label = "Esperanza de vida en Chile",  
            subtitle = "Años 1952 a 2007") +  
  xlab(label = "Año") +  
  ylab(label = "Esperanza de vida")
```



Customización de gráficos en ggplot2

Argumentos color y size

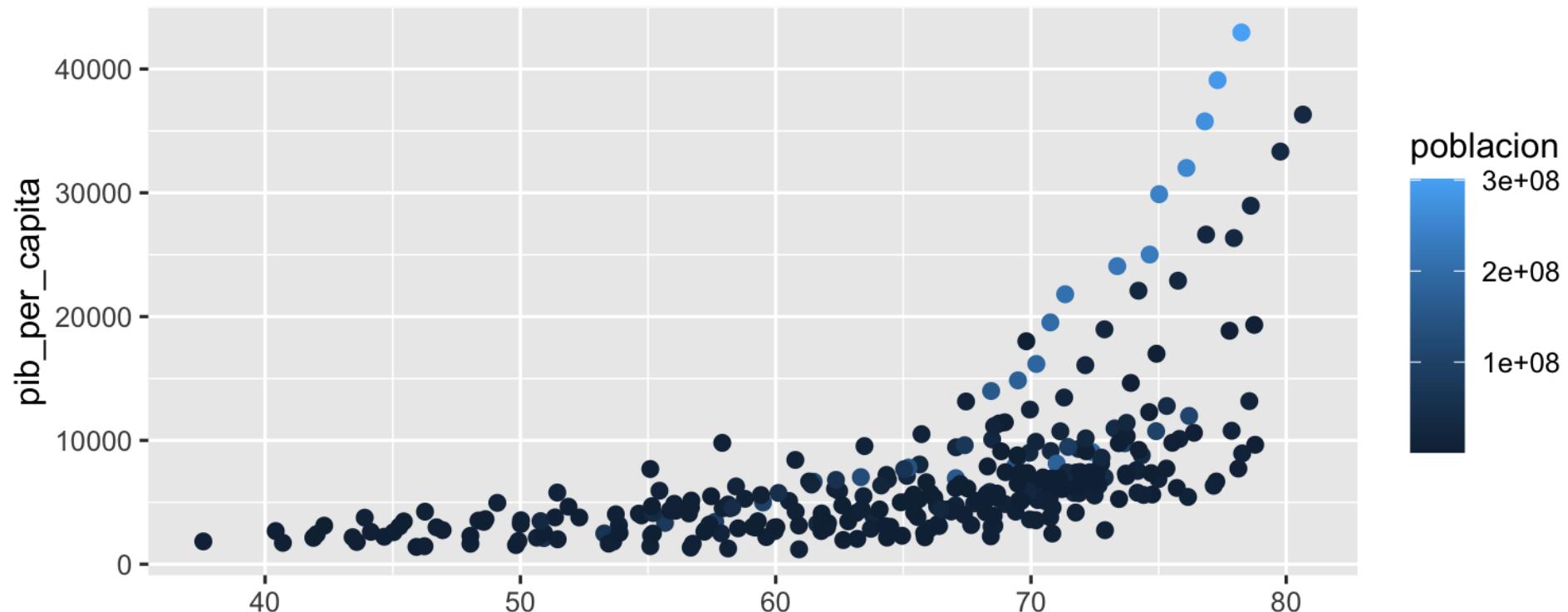
```
ggplot(data = paises,  
       aes(x = esperanza_de_vida, y = pib_per_capita)) +  
  geom_point(aes(size = poblacion, color = continente), alpha = 0.7)
```



Customización de gráficos en ggplot2

Argumentos color y size

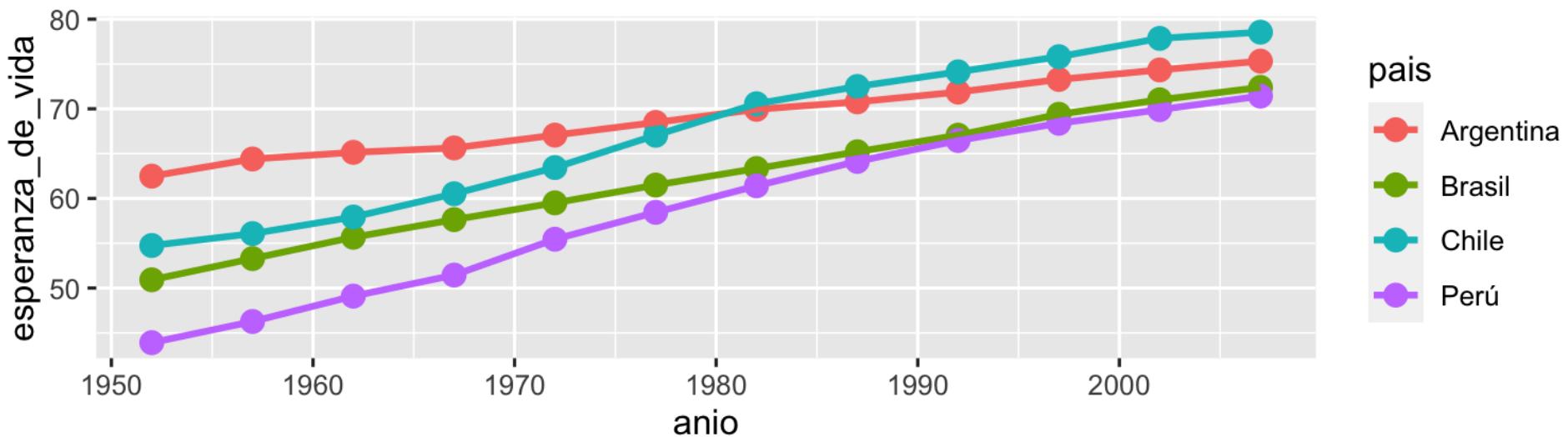
```
ggplot(data = paises[paises$continente=="Américas",],  
       aes(x = esperanza_de_vida, y = pib_per_capita)) +  
  geom_point(aes(color = poblacion), size = 2)
```



Customización de gráficos en ggplot2

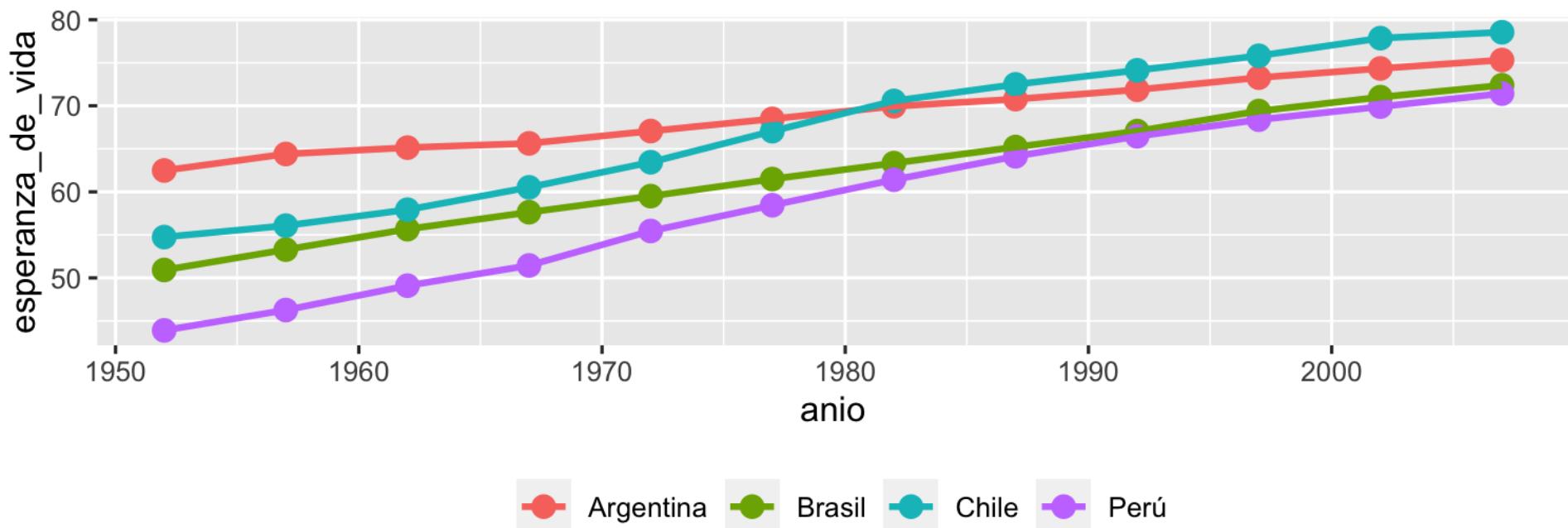
Leyendas: La leyenda puede personalizarse mediante el argumento theme(), función que permite modificar la mayoría de los elementos estéticos de un ggplot.

```
ggplot(data = paises[paises$pais %in% c("Chile", "Argentina", "Perú", "Brasil"),],  
       aes(x = anio, y = esperanza_de_vida, color = pais) ) +  
       geom_point(size = 3) +  
       geom_line(size = 1)
```



Customización de gráficos en ggplot2

```
ggplot(data = paises[paises$pais %in% c("Chile", "Argentina", "Perú", "Brasil"),],  
       aes(x = anio, y = esperanza_de_vida, color = pais) ) +  
  geom_point(size = 3) +  
  geom_line(size = 1) +  
  theme(legend.position = "bottom",  
        legend.title = element_blank(),  
        legend.background = element_rect(fill="white"))
```



Customización de gráficos en ggplot2

Temas

En la librería ggplot existe una variedad de temas precargados que permiten modificar de manera rápida cómo se ve el fondo de un gráfico. Estos son:

`theme_grey()`/`theme_gray()` - Por defecto

`theme_bw()`

`theme_linedraw()`

`theme_light()`

`theme_dark()`

`theme_minimal()`

`theme_classic()`

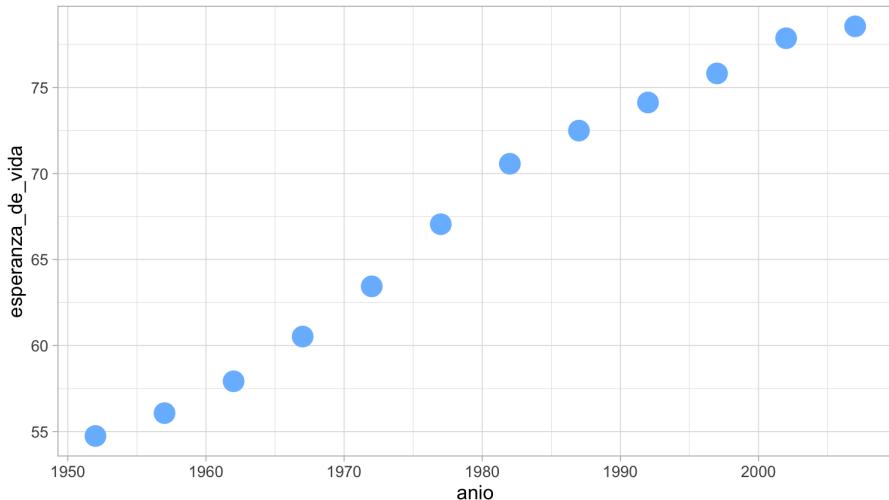
`theme_void()`

`theme_test()`

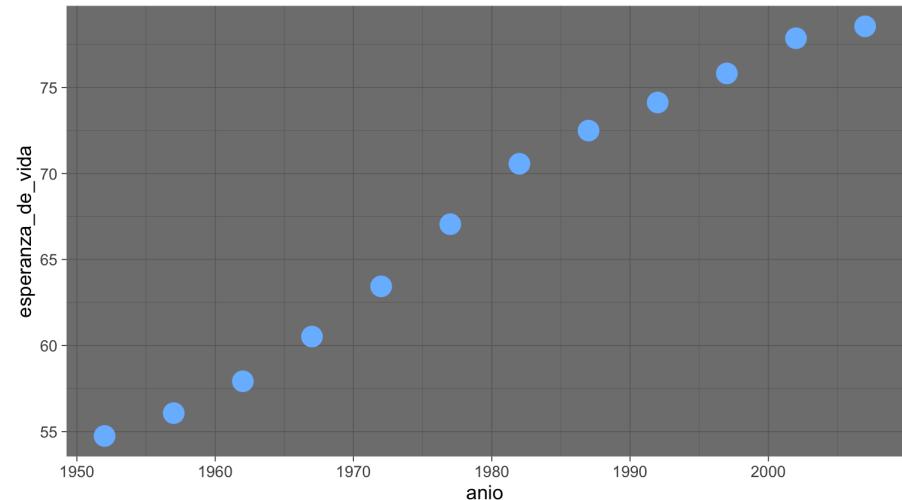
Customización de gráficos en ggplot2

Temas

```
ggplot(data = paises[paises$pais=="Chile"  
                      aes(anio, esperanza_de_vida)) +  
        geom_point(color = "#79bcff",  
                   size = 5) +  
        theme_light()
```



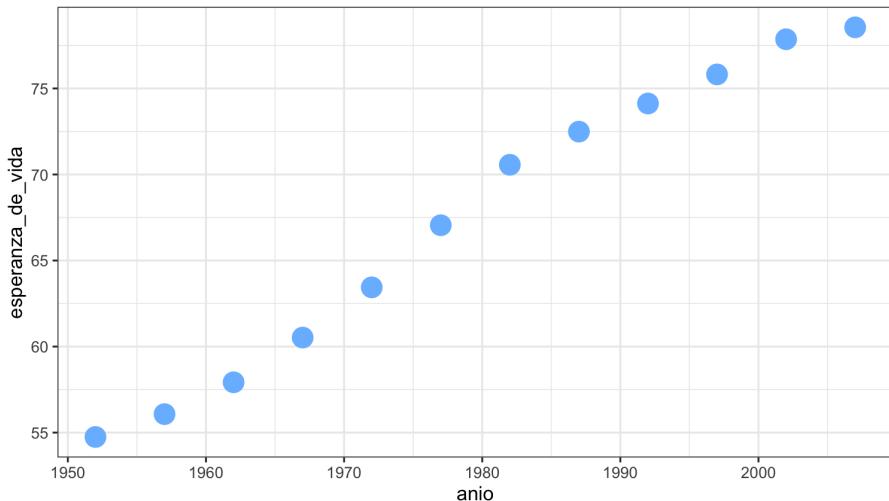
```
ggplot(data = paises[paises$pais=="Chile"  
                      aes(anio, esperanza_de_vida)) +  
        geom_point(color = "#79bcff",  
                   size = 5) +  
        theme_dark()
```



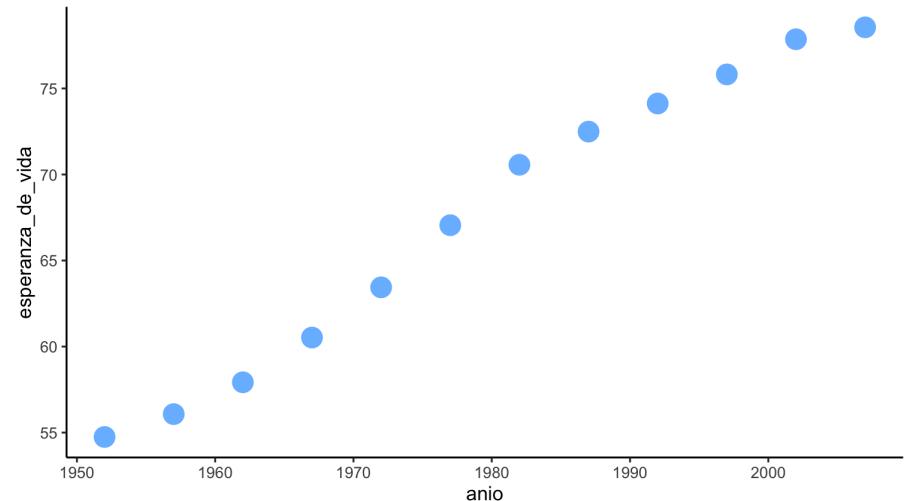
Customización de gráficos en ggplot2

Temas

```
ggplot(data = paises[paises$pais=="Chile"  
                      aes(anio, esperanza_de_vida)) +  
        geom_point(color = "#79bcff",  
                   size = 5) +  
        theme_bw()
```

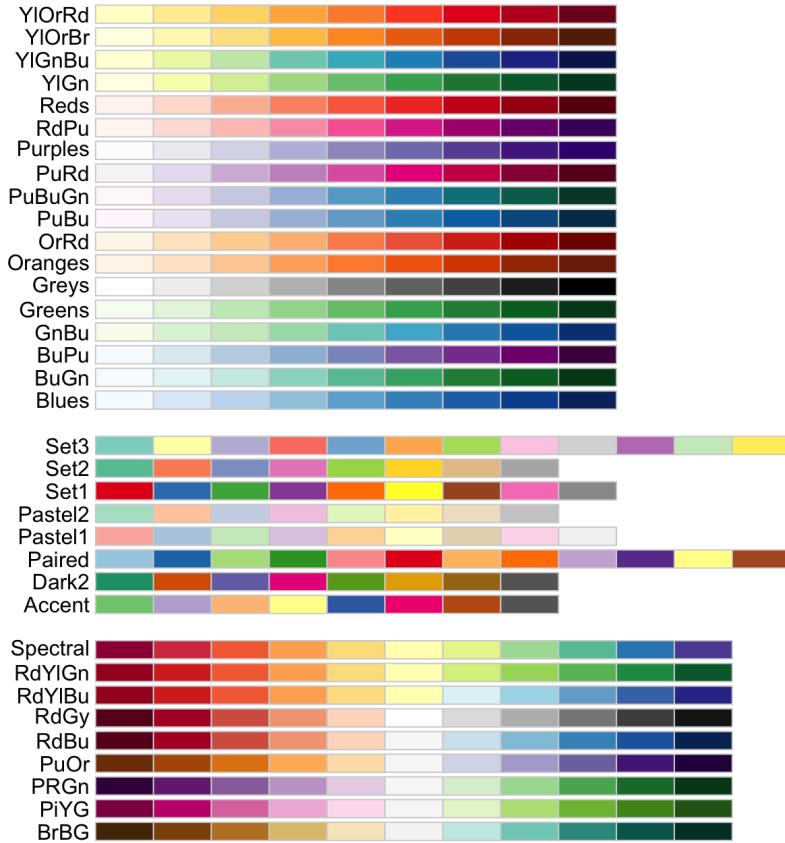


```
ggplot(data = paises[paises$pais=="Chile"  
                      aes(anio, esperanza_de_vida)) +  
        geom_point(color = "#79bcff",  
                   size = 5) +  
        theme_classic()
```



Customización de gráficos en ggplot2

Colores



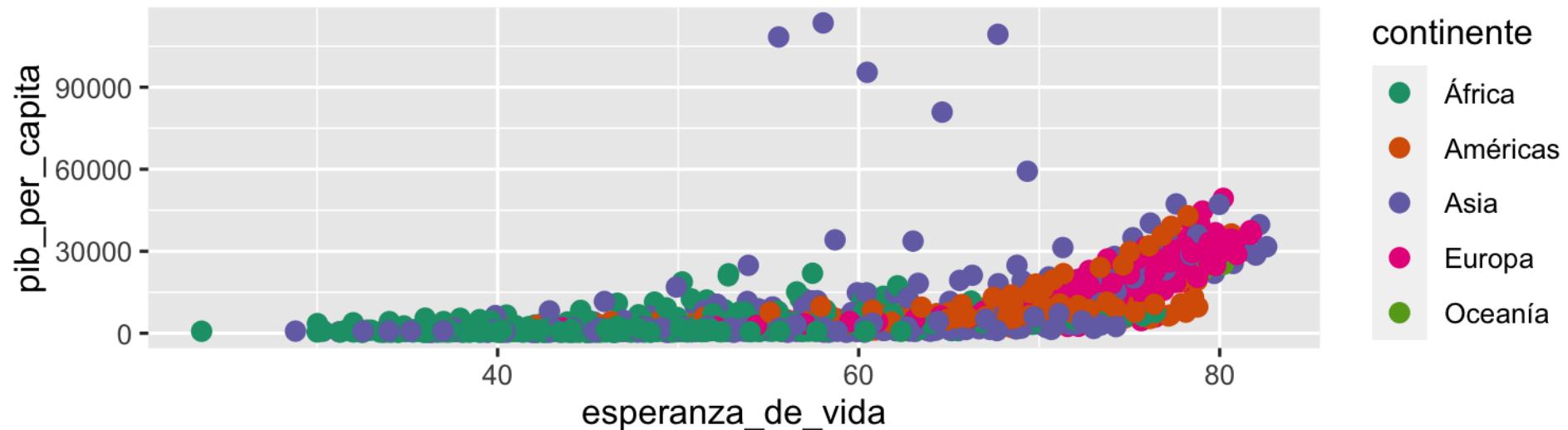
La librería [RColorBrewer](#) contiene un gran número de paletas de colores para los gráficos de R y ggplot2. Las paletas de colores se definen con el argumento `palette` dentro de los comandos `scale_color_brewer()` o `scale_fill_brewer()`.

```
install.packages("RColorBrewer")
library(RColorBrewer)
```

Customización de gráficos en ggplot2

Colores

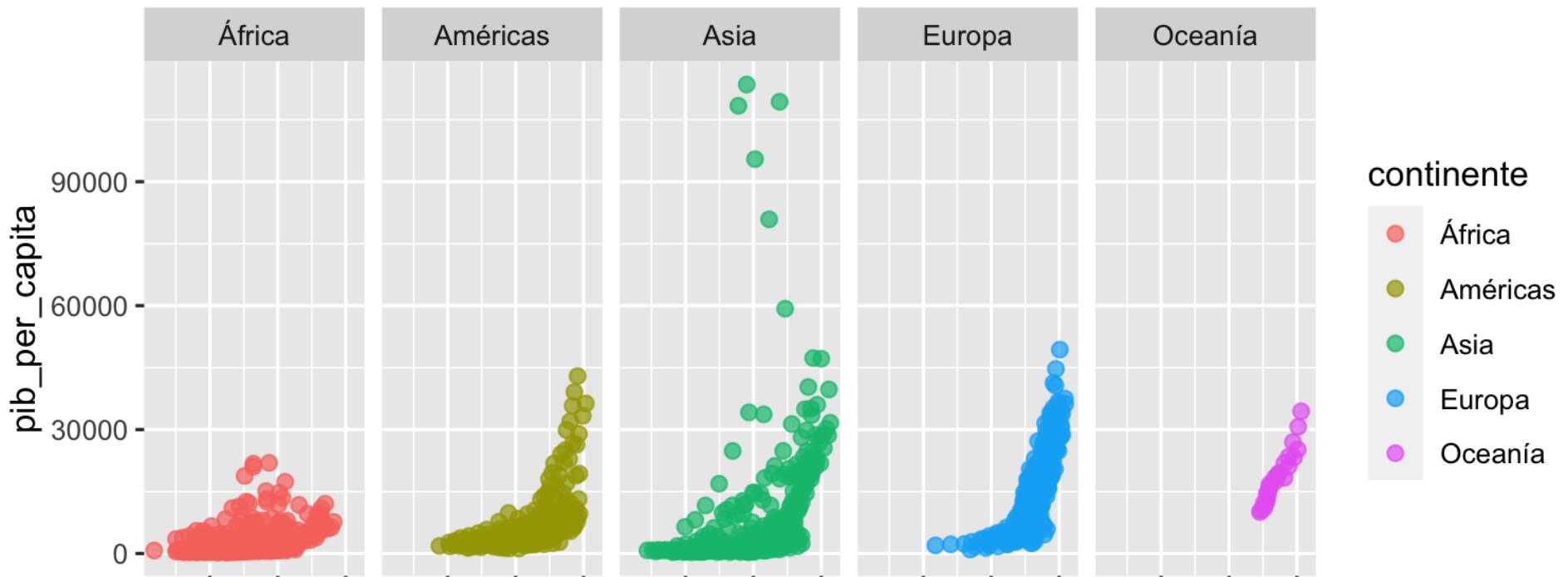
```
library(RColorBrewer)
ggplot(data=paises,
       aes(x = esperanza_de_vida,
           y = pib_per_capita,
           color = continente)) +
  geom_point(size = 2.5) +
  scale_color_brewer(palette="Dark2")
```



Customización de gráficos en ggplot2

Facetas

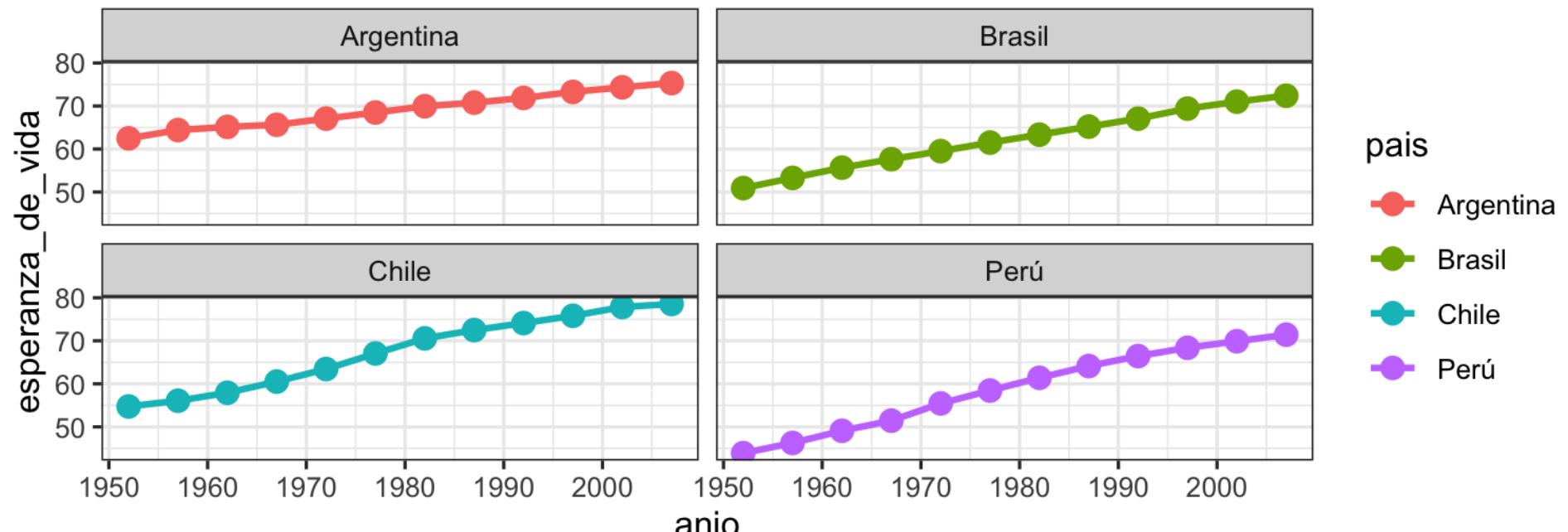
```
ggplot(data=paises, aes(x = esperanza_de_vida,  
                         y = pib_per_capita, color = continente))+  
  geom_point(size = 2, alpha = 0.7) +  
  facet_grid(~continente)
```



Customización de gráficos en ggplot2

Facetas

```
ggplot(data = paises[paises$pais %in% c("Chile", "Argentina", "Perú", "Brasil"),],  
       aes(x = anio, y = esperanza_de_vida, color = pais) ) +  
       geom_point(size = 3) + geom_line(size = 1) +  
       facet_wrap(~ pais) +  
       theme_bw()
```



Customización de gráficos en ggplot2

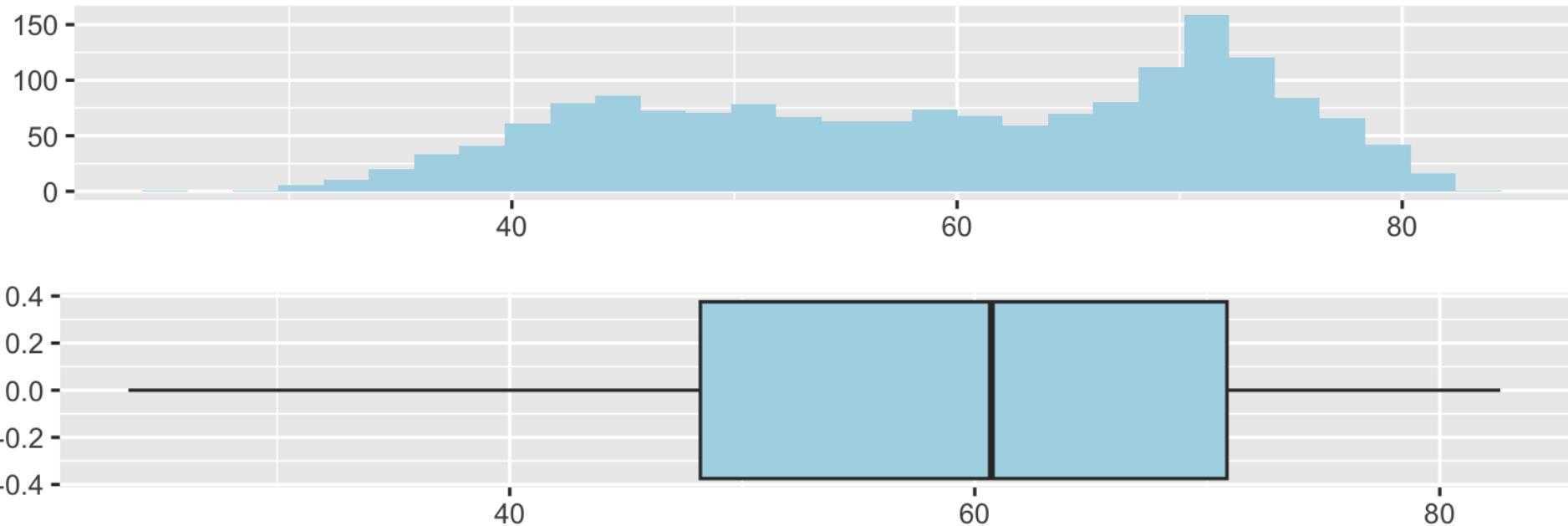
Matrices de gráficos

```
g1 = ggplot(data = paises, aes(x = esperanza_de_vida)) +  
  geom_histogram(fill = "lightblue") +  
  labs(x = element_blank(),  
       y = element_blank())  
  
g2 = ggplot(data = paises, aes(x = esperanza_de_vida)) +  
  geom_boxplot(fill = "lightblue") +  
  labs(x=element_blank())
```

Customización de gráficos en ggplot2

Matrices de gráficos

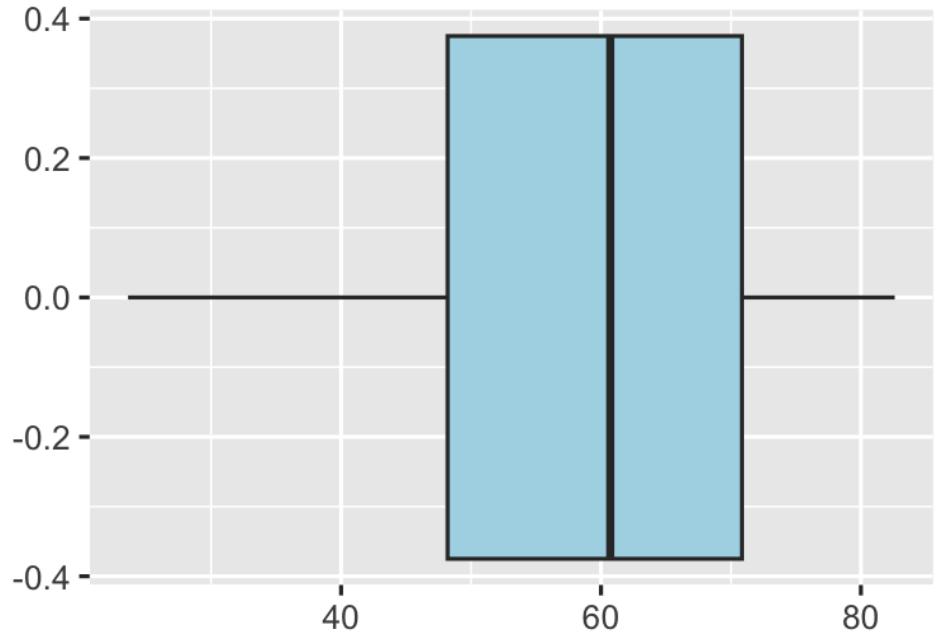
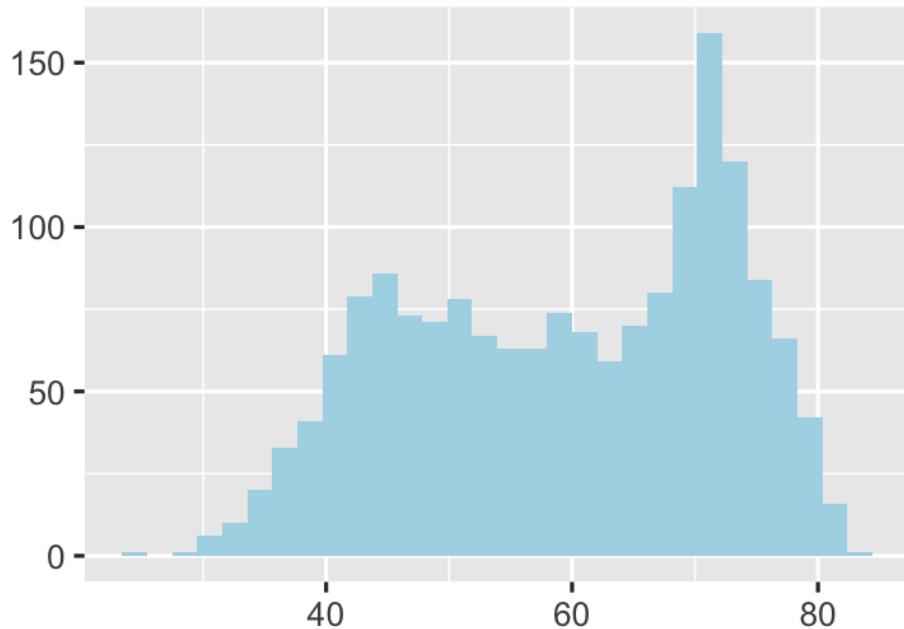
```
library(gridExtra)  
grid.arrange(g1,g2, nrow=2)
```



Customización de gráficos en ggplot2

Matrices de gráficos

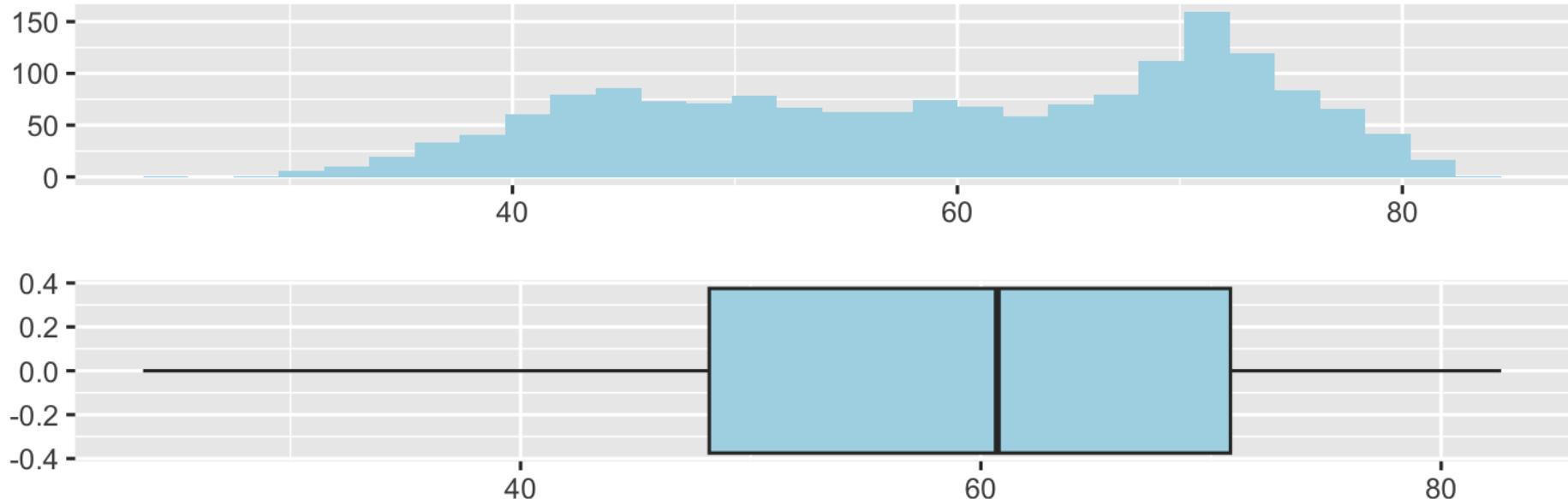
```
library(ggpubr)  
ggarrange(g1,g2, ncol=2)
```



Customización de gráficos en ggplot2

Matrices de gráficos

```
library(patchwork)  
g1 / g2
```



Customización de gráficos en ggplot2

Extensiones de ggplot2

ggplot2 extensions - gallery

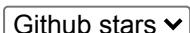
- [Add Your Extension!](#)
- [exts.ggplot2.tidyverse.org](#)
- [Navbar Link](#)



127 registered extensions available to explore



- Name
- Author
- Github stars



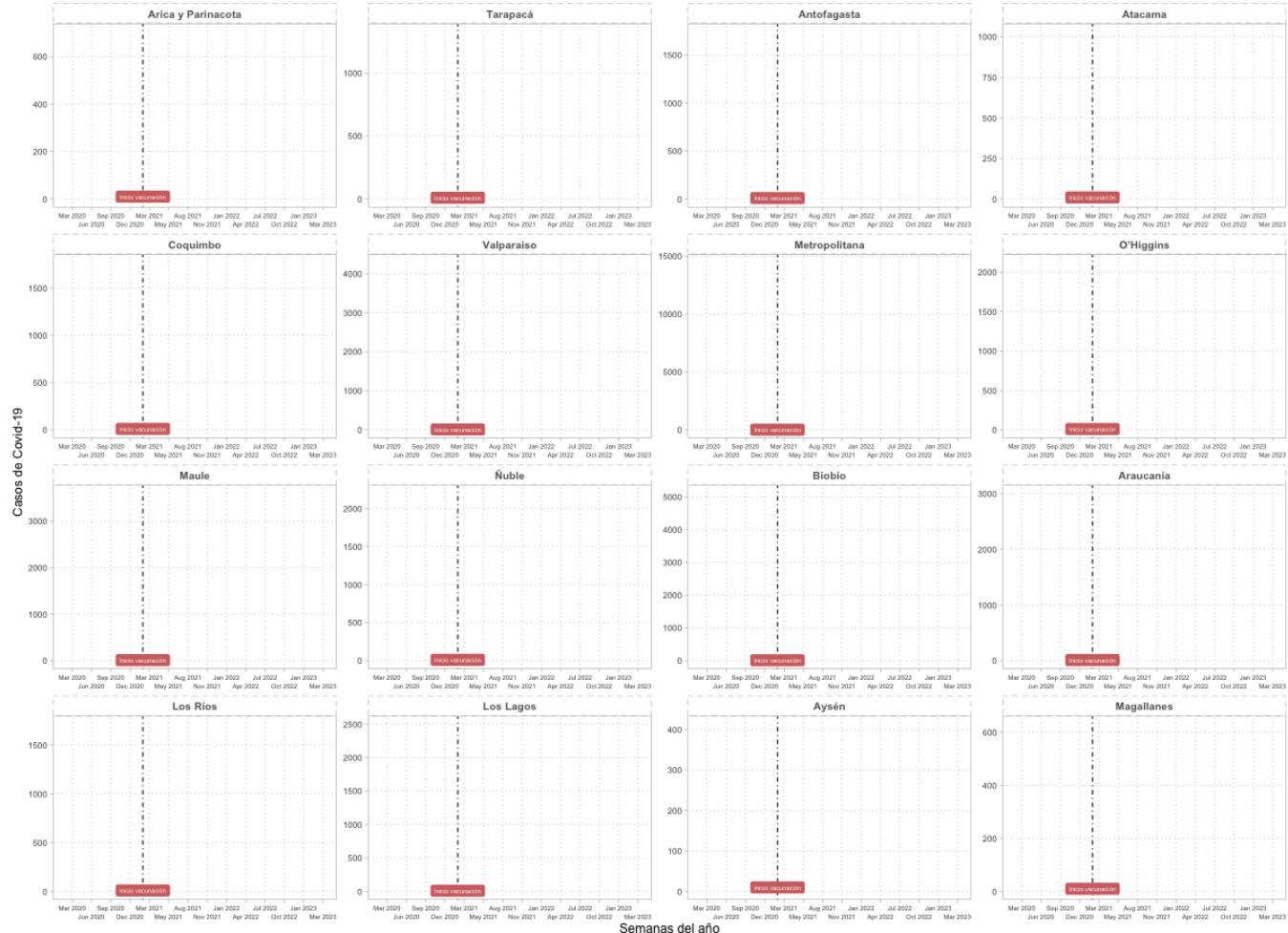
Sort



-

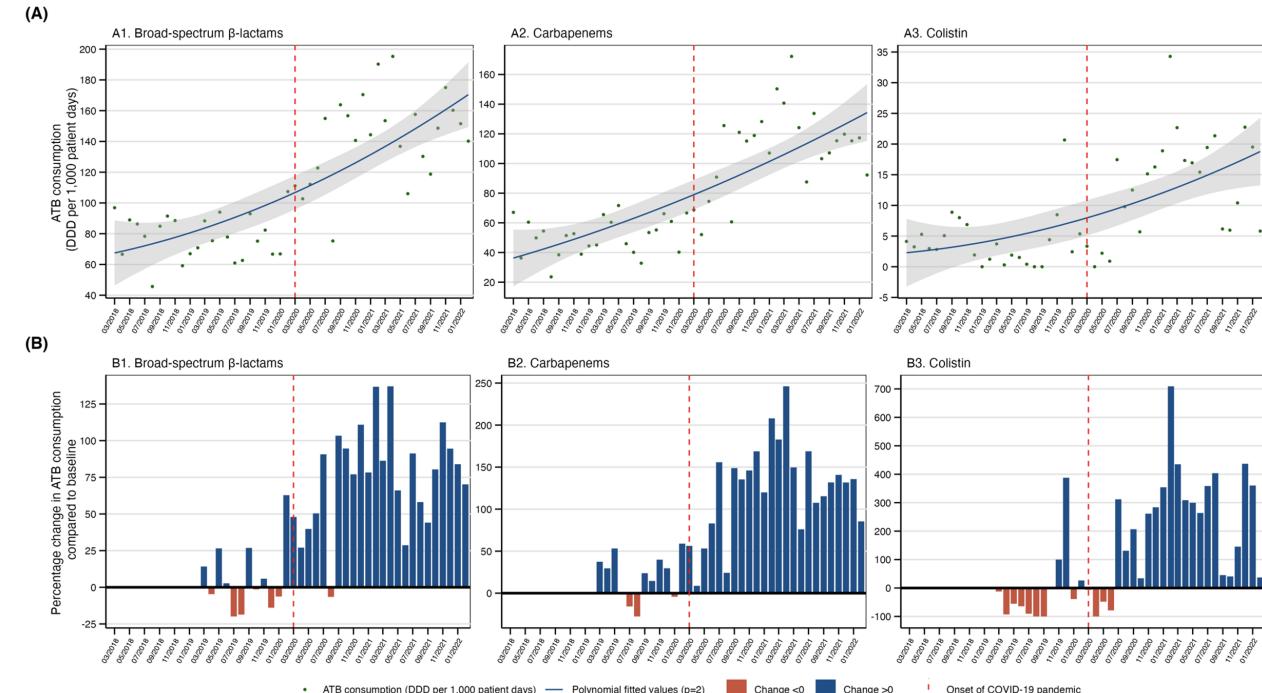
Un ejemplo

Casos de Covid-19 para el día: 2020-03-03



Un ejemplo académico

Figure 1. Hospital-wide Antibiotic consumption in DDD per 1,000 patient-days (A) and monthly percentage change over time (B), by antibiotics group, 2018-2022



ATB, Antibiotic; DDD, Defined Daily Dose; Colistin is classified as a compound active against carbapenemase-producing (CP) organisms. Broad-spectrum β -lactam ATBs include piperacillin/tazobactam, ceftazidime, meropenem, and imipenem. Carbapenems include imipenem, meropenem, and ertapenem. (B) Percentage change in antibiotic consumption over time (compared to the average antibiotic consumption between March 2018 and February 2019).

Referencias

Wickham, H., & Grolemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data. " O'Reilly Media, Inc.". Cap. 2. Recurso en línea: <https://r4ds.hadley.nz/>

Urdinez, F., & Cruz, A. (2020). R for Political Data Science: A Practical Guide. CRC Press. Cap. 3. Recurso en línea en español: <https://arcruz0.github.io/libroadp/>

Posit Cheatsheets ("hojas de trucos"): <https://posit.co/resources/cheatsheets/?type=posit-cheatsheets/>

Página oficial de ggplot2: <https://ggplot2.tidyverse.org/>





Sesión 4

Visualización de datos en R

04 de agosto, 2023

✉ José D. Conejeros | 📩 jdconejeros@uc.cl | 🌐 JDConjeros