

COMPUTACIÓN ESTADÍSTICA CON R

CLASE 2

RUBÉN Soza

MANEJO DE BASES DE DATOS UTILIZANDO FUNCIONES DE DPLYR

¿QUÉ VIMOS LA CLASE ANTERIOR?

- » **filter()**: Seleccionar filas utilizando un criterio.
- » **select()**: Seleccionar Columnas.
- » **arrange()**: Ordenar filas.
- » **mutate()**: Crear/Modificar Columnas.
- » **summarise()**: Resumir Columnas.

CONTINUEMOS CON MÁS
FUNCIONES!

GROUP_BY + SUMMARISE: RESUMIR COLUMNAS POR GRUPOS

Group_by divide la base de datos en grupos, lo cual permite obtener medidas de resumen por grupos utilizando summarise.



GROUP_BY + SUMMARISE: *CÓDIGO*

```
pollution <- read.csv("Datasets/pollution.csv")
kable(pollution)
```

| city | size | amount |
|----------|-------|--------|
| New York | large | 23 |
| New York | small | 14 |
| London | large | 22 |
| London | small | 16 |
| Beijing | large | 121 |
| Beijing | small | 56 |

```
pollution %>%
  group_by(city) %>%
  summarise(promedio = median(amount),
            suma = sum(amount),
            n = n(),
            max = max(amount)) -> resumen
kable(resumen)
```

| city | promedio | suma | n | max |
|----------|----------|------|---|-----|
| Beijing | 88.5 | 177 | 2 | 121 |
| London | 19.0 | 38 | 2 | 22 |
| New York | 18.5 | 37 | 2 | 23 |

MUTATE + CASE_WHEN: *CREACIÓN DE INDICADORES*

`case_when()` es una función que permite, en conjunto con `mutate()`, generar variables indicadoras.

```
pollution %>%
  mutate(mayor_30 = case_when(
    amount > 30 ~ 1,
    TRUE ~ 0)) -> pollution2
kable(pollution2)
```

| city | size | amount | mayor_30 |
|----------|-------|--------|----------|
| New York | large | 23 | 0 |
| New York | small | 14 | 0 |
| London | large | 22 | 0 |
| London | small | 16 | 0 |
| Beijing | large | 121 | 1 |

| city | size | amount | mayor_30 |
|------|------|--------|----------|
|------|------|--------|----------|

| | | | |
|---------|-------|----|---|
| Beijing | small | 56 | 1 |
|---------|-------|----|---|

Se pueden crear variables con más de 2 valores:

```
pollution2 %>%  
  mutate(categorias = case_when(  
    amount >= 14 & amount < 24 ~ '[14,23]',  
    amount >= 24 & amount < 57 ~ '[24,56]',  
    TRUE ~ '>56'  
  )) -> pollution2  
kable(pollution2)
```

| city | size | amount | mayor_30 | categorias |
|----------|-------|--------|----------|------------|
| New York | large | 23 | 0 | [14,23] |
| New York | small | 14 | 0 | [14,23] |
| London | large | 22 | 0 | [14,23] |
| London | small | 16 | 0 | [14,23] |
| Beijing | large | 121 | 1 | >56 |
| Beijing | small | 56 | 1 | [24,56] |

EJEMPLO EN RSTUDIO: NUMEROS.XLSX

Encuentre el máximo y mínimo por región de la variable valor.

ACTIVIDAD 1

Teniendo en consideración la BD encuesta.xlsx:

- » Seleccione Región, Sexo, Edad, cuánto gastó en seguridad y Score Socioeconómico.
- » Seleccione hombres de Valparaíso.
- » Ordene de menor a mayor la edad. Genere un indicador para la variable edad.
- » Añada una nueva variable denominada PRSC, calculada como el Score del individuo dividido por el máximo del score observado.
- » Obtenga el promedio del PRSC para cada grupo de gasto en seguridad.

MANIPULACIÓN DE BD PARTE 2

IMPORTAR DATOS

```
songs <- read_csv("Datasets/songs.csv")  
kable(songs)
```

| song | name |
|---------------------|-------|
| Across the Universe | John |
| Come Together | John |
| Hello, Goodbye | Paul |
| Peggy Sue | Buddy |

```
artists <- read_csv("Datasets/artists.csv")  
kable(artists)
```

| name | plays |
|--------|--------|
| George | sitar |
| John | guitar |
| Paul | bass |
| Ringo | drums |

RENAME: *RENOMBRAR COLUMNAS*

Permite cambiar los nombres de las columnas seleccionadas

```
artists %>%
  rename(Nombres = name) -> artist2
kable(artist2)
```

| Nombres | plays |
|----------------|--------------|
| George | sitar |
| John | guitar |
| Paul | bass |
| Ringo | drums |

JOIN: JUNTAR BASES DE DATOS

Existen 7 funciones, en dplyr, que permiten juntar/contrastar dos bases de datos en una sola, utilizando una columna como link. Algunos ejemplos son:

- » `inner_join()`
- » `left_join()`
- » `right_join()`

LEFT_JOIN: EJEMPLO

| songs | | artists | | | | |
|---------------------|-------|---------|--------|---------------------|-------|--------|
| song | name | name | plays | song | name | plays |
| Across the Universe | John | George | sitar | Across the Universe | John | guitar |
| Come Together | John | John | guitar | Come Together | John | guitar |
| Hello, Goodbye | Paul | Paul | bass | Hello, Goodbye | Paul | bass |
| Peggy Sue | Buddy | Ringo | drums | Peggy Sue | Buddy | <NA> |

```
left_join(songs, artists, by = "name")
```

ALGUNOS OTROS EJEMPLOS:

```
artist3 <- inner_join(songs, artists, by = "name")
kable(artist3)
```

| song | name | plays |
|---------------------|------|--------|
| Across the Universe | John | guitar |
| Come Together | John | guitar |
| Hello, Goodbye | Paul | bass |

```
artist4 <- right_join(songs, artists, by = "name")
kable(artist4)
```

| song | name | plays |
|---------------------|--------|--------|
| NA | George | sitar |
| Across the Universe | John | guitar |
| Come Together | John | guitar |
| Hello, Goodbye | Paul | bass |
| NA | Ringo | drums |

¿Qué pasa si las columnas links poseen 2 nombres diferentes?

```
left_join(songs, artist2, by = c("name" = "Nombres"))
```

```
## # A tibble: 4 x 3
##   song                 name  plays
##   <chr>                <chr> <chr>
## 1 Across the Universe John  guitar
## 2 Come Together        John  guitar
## 3 Hello, Goodbye       Paul  bass 
## 4 Peggy Sue            Buddy <NA>
```

VEAMOS UN EJEMPLO EN R

DATOS DE TUBERCULOSIS

```
library(tidyverse)
data("table4a")
kable(table4a)
```

| country | 1999 | 2000 |
|-------------|--------|--------|
| Afghanistan | 745 | 2666 |
| Brazil | 37737 | 80488 |
| China | 212258 | 213766 |

```
data("table2")
kable(table2)
```

| country | year | type | count |
|-------------|------|------------|------------|
| Afghanistan | 1999 | cases | 745 |
| Afghanistan | 1999 | population | 19987071 |
| Afghanistan | 2000 | cases | 2666 |
| Afghanistan | 2000 | population | 20595360 |
| Brazil | 1999 | cases | 37737 |
| Brazil | 1999 | population | 172006362 |
| Brazil | 2000 | cases | 80488 |
| Brazil | 2000 | population | 174504898 |
| China | 1999 | cases | 212258 |
| China | 1999 | population | 1272915272 |
| China | 2000 | cases | 213766 |
| China | 2000 | population | 1280428583 |

```
cases <- read_csv("Datasets/cases.csv")  
kable(cases)
```

| country | 2011 | 2012 | 2013 |
|---------|-------|-------|-------|
| FR | 7000 | 6900 | 7000 |
| DE | 5800 | 6000 | 6200 |
| US | 15000 | 14000 | 13000 |

¿Como obtendríamos el promedio por país?

¿COMO OBTENDRÍAMOS EL PROMEDIO POR PAÍS?

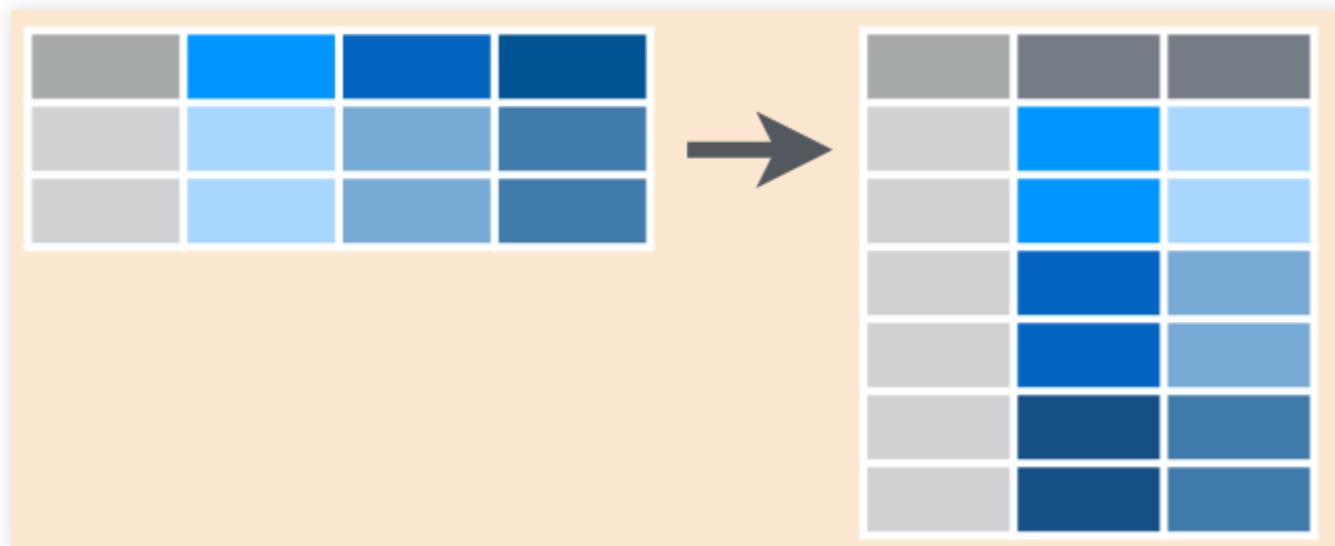
```
cases %>%
  mutate(promedio = (`2011` + `2012` + `2013`)/3) -> Promedios
kable(Promedios)
```

| country | 2011 | 2012 | 2013 | promedio |
|---------|-------|-------|-------|-----------|
| FR | 7000 | 6900 | 7000 | 6966.667 |
| DE | 5800 | 6000 | 6200 | 6000.000 |
| US | 15000 | 14000 | 13000 | 14000.000 |

Ahora pensemos en una tabla más grande, con más años..

GATHER: *RECOLECTAR*

Coloca nombres de columnas en una variable (columna) **key**, recolectando los valores (**value**) de las columnas en un **sola** columna



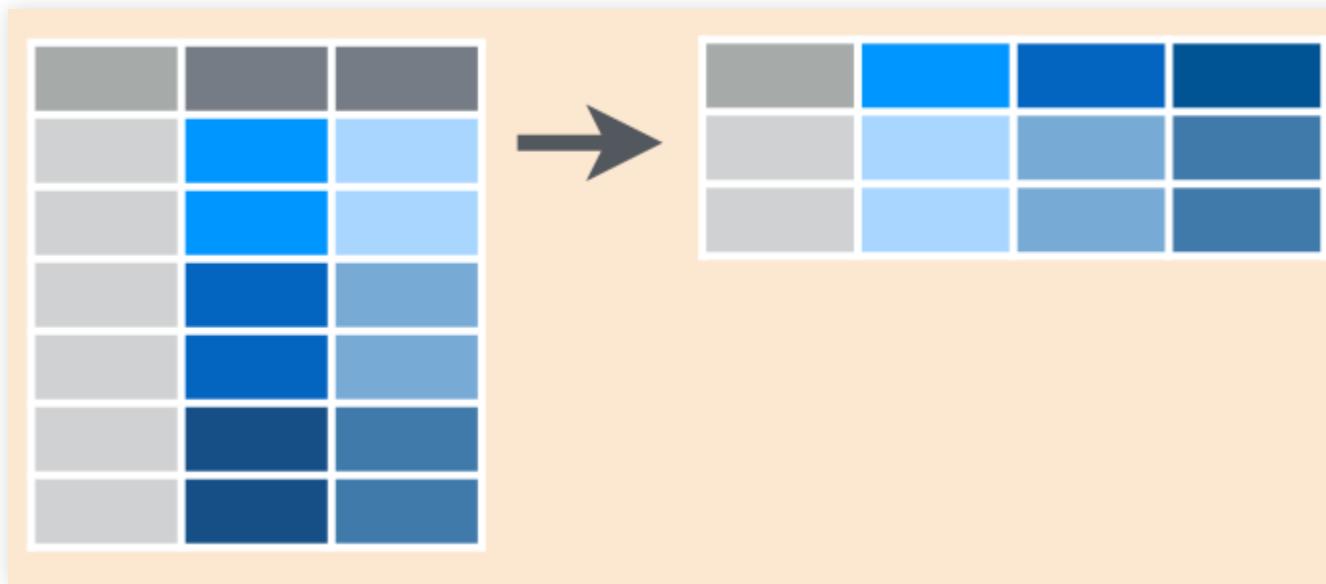
GATHER: *CÓDIGO*

```
table4ag <- gather(table4a, `1999`, `2000`, key = "year", value = "cases")  
kable(table4ag)
```

| country | year | cases |
|-------------|------|--------|
| Afghanistan | 1999 | 745 |
| Brazil | 1999 | 37737 |
| China | 1999 | 212258 |
| Afghanistan | 2000 | 2666 |
| Brazil | 2000 | 80488 |
| China | 2000 | 213766 |

SPREAD: *ESPARCIR*

Esparse un par de columnas (2, key-value) en multiples columnas



SPREAD: *CÓDIGO*

```
table2s <- spread(table2, type, count)  
kable(table2s)
```

| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 745 | 19987071 |
| Afghanistan | 2000 | 2666 | 20595360 |
| Brazil | 1999 | 37737 | 172006362 |
| Brazil | 2000 | 80488 | 174504898 |
| China | 1999 | 212258 | 1272915272 |
| China | 2000 | 213766 | 1280428583 |

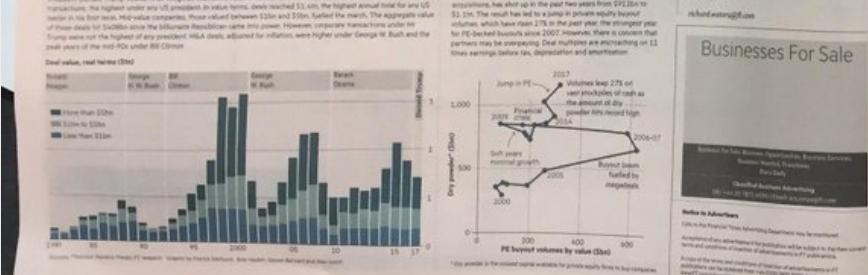
VEAMOS UN EJEMPLO EN R

ANÁLISIS EXPLORATORIO Y DESCRIPTIVO EN R

ESTRUCTURA

- » ¿Por que?
- » Definiciones
- » Recomendaciones Técnicas & Visuales
- » Ejercicios y concursos

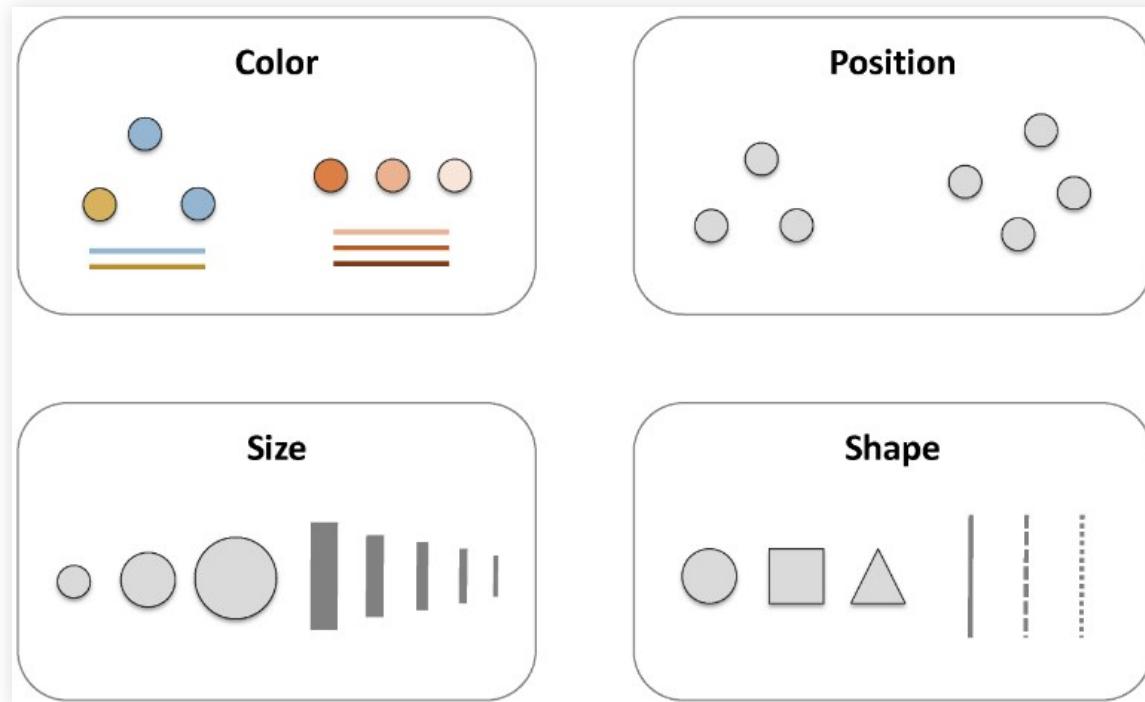
¿POR QUE VISUALIZACIÓN?



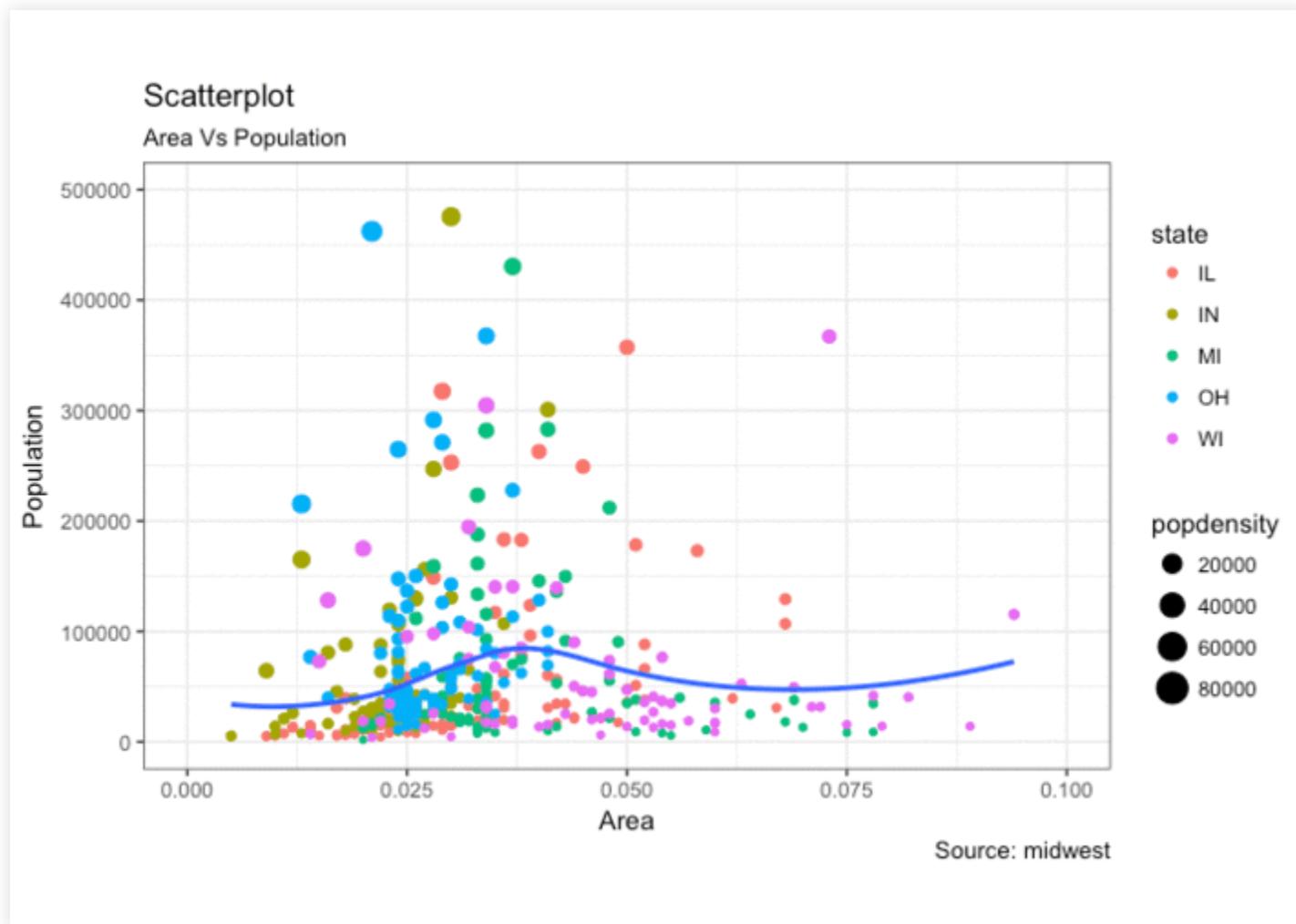


DEFINICIÓN TÉCNICA (POSIBLE)

Representación gráfica de datos **codificando la información** como:
posición, tamaño, formas, colores



Muchos sabores:

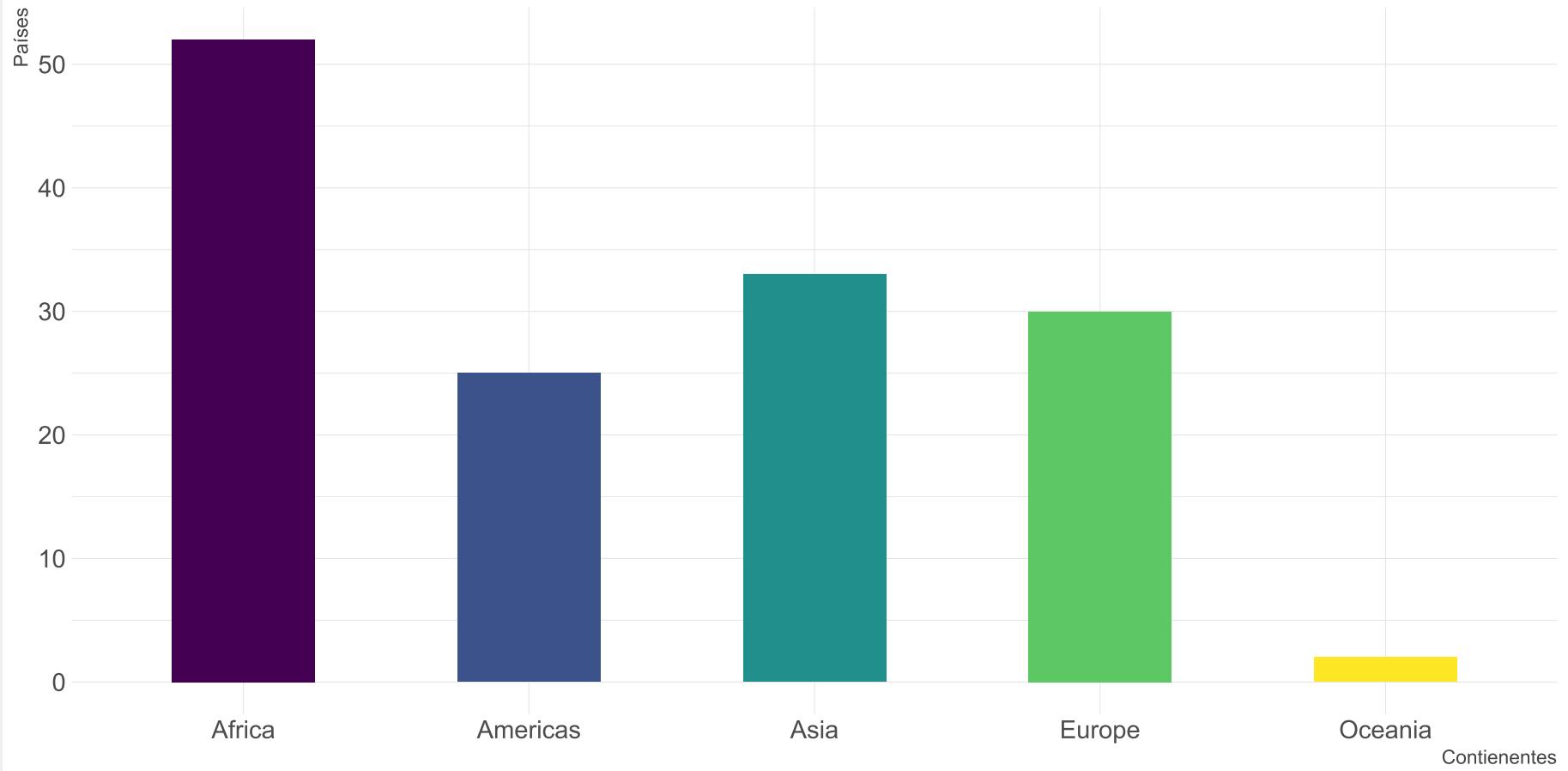


Cosas importantes sobre visualización de datos:

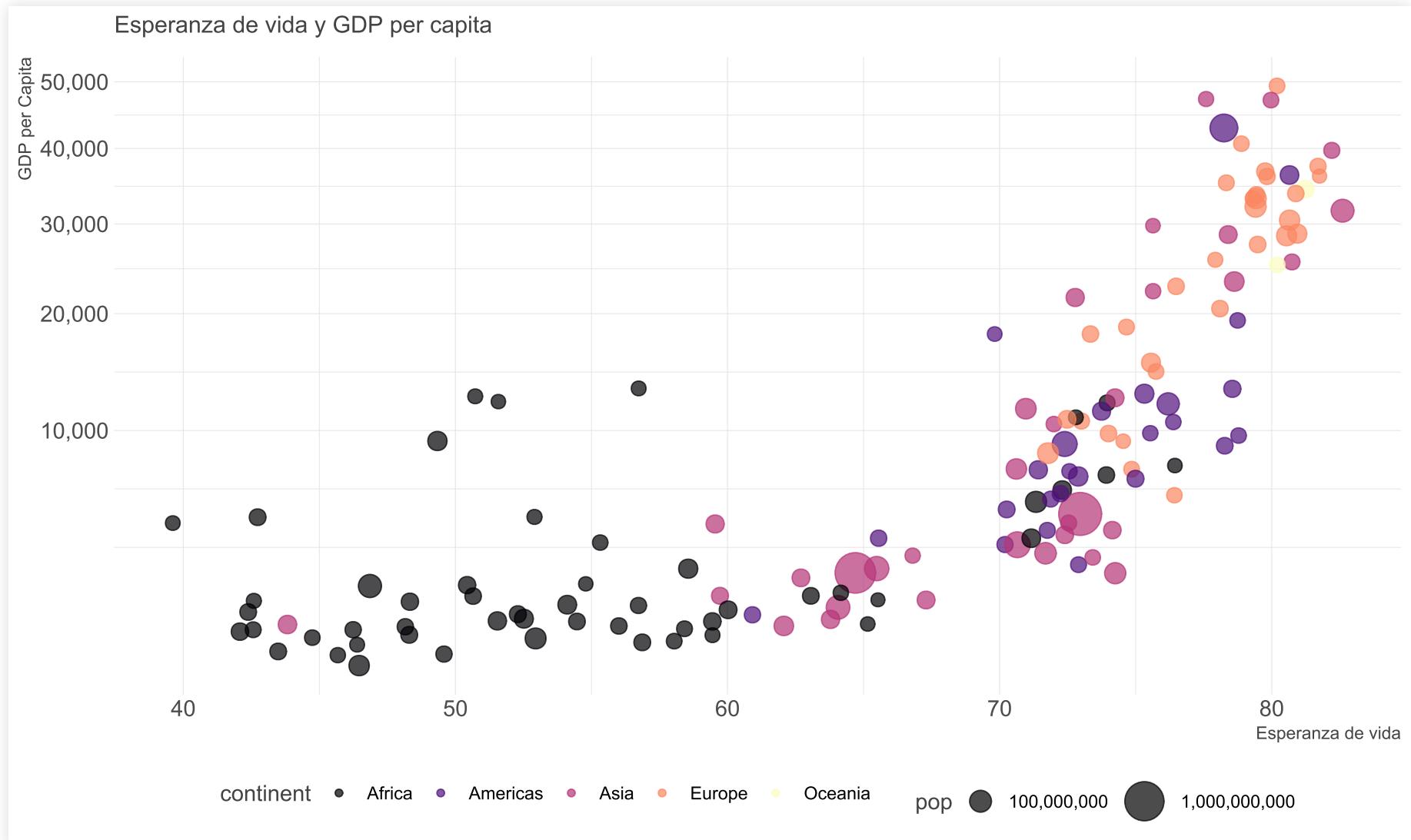
- » Lleva información, es un mensaje
- » Ejercicio mental para interpretar o *decodificar*(!!) información
- » No siempre es el fin
- » Herramienta exploratoria
- » Distintas visualizaciones en mismos datos / Distintas historias

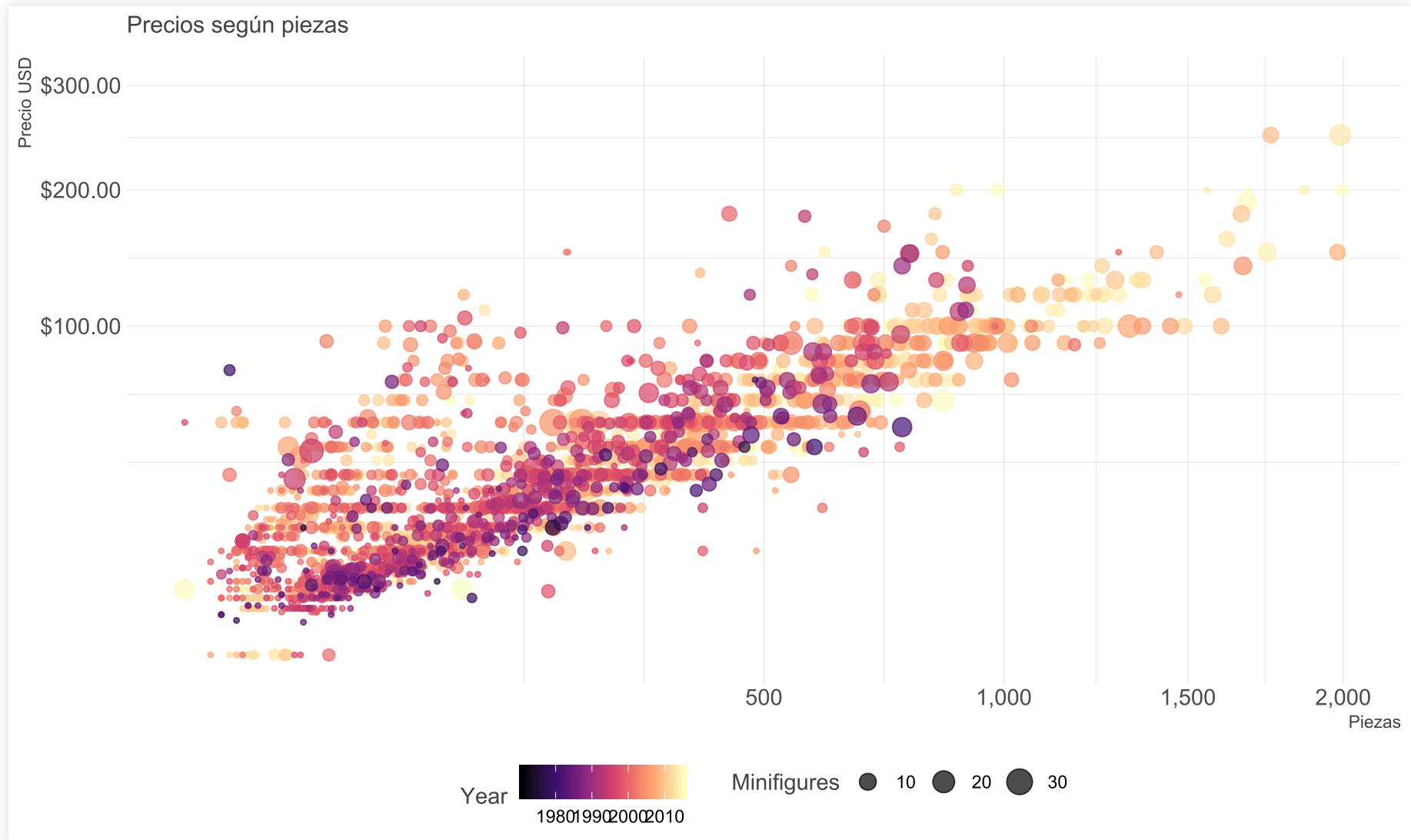
Africa tiene más países que el resto de continentes

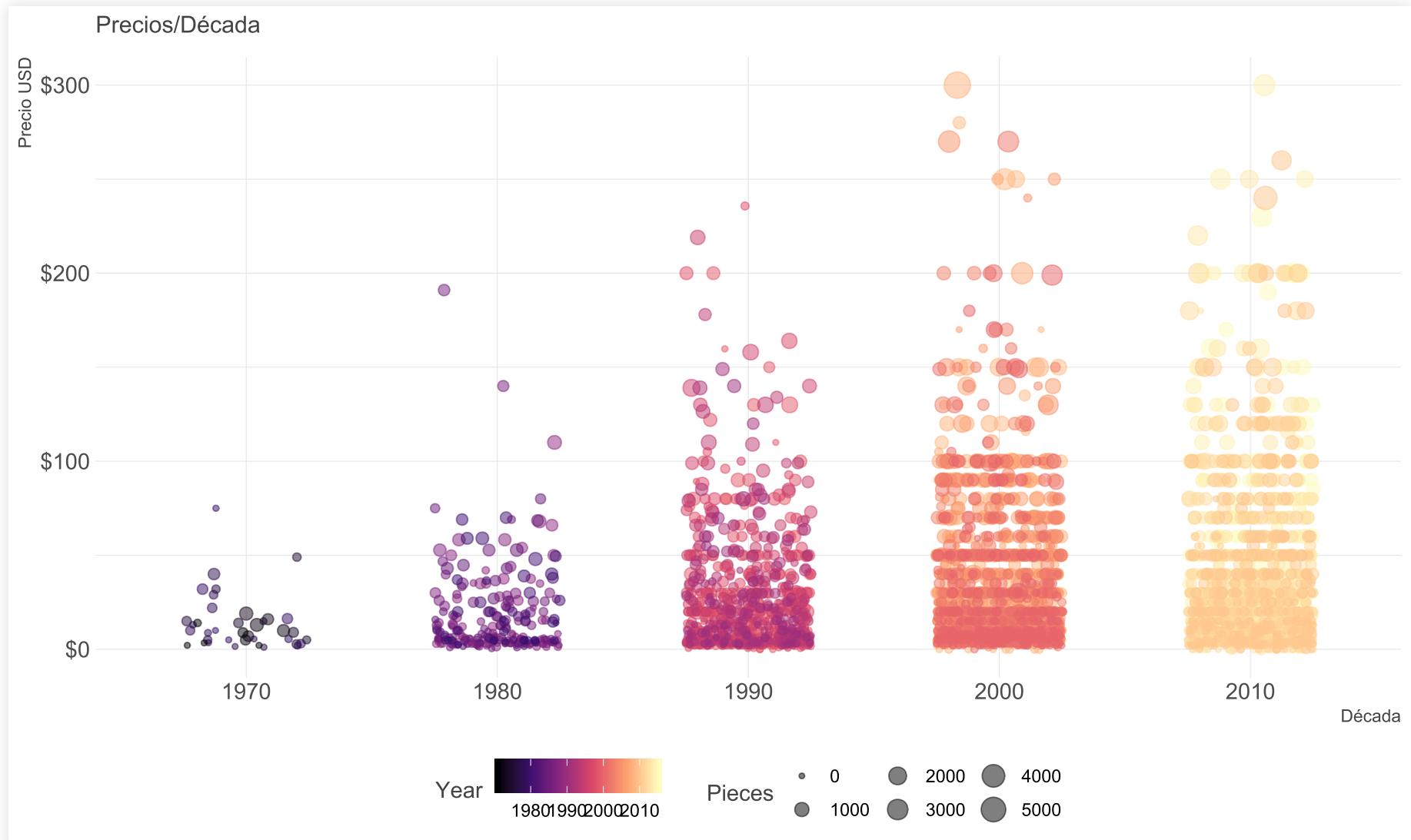
Un interesante subtítulo para contexto y dar detalles quizás puede ser más largo pero quien soy yo para decir que se debe y lo que no



Importante mencionar la fuente, en caso contrario no me creen







EJERCICIO

¿Existe el mejor gráfico? ¿Cuál de las siguientes formas funciona mejor con el título?

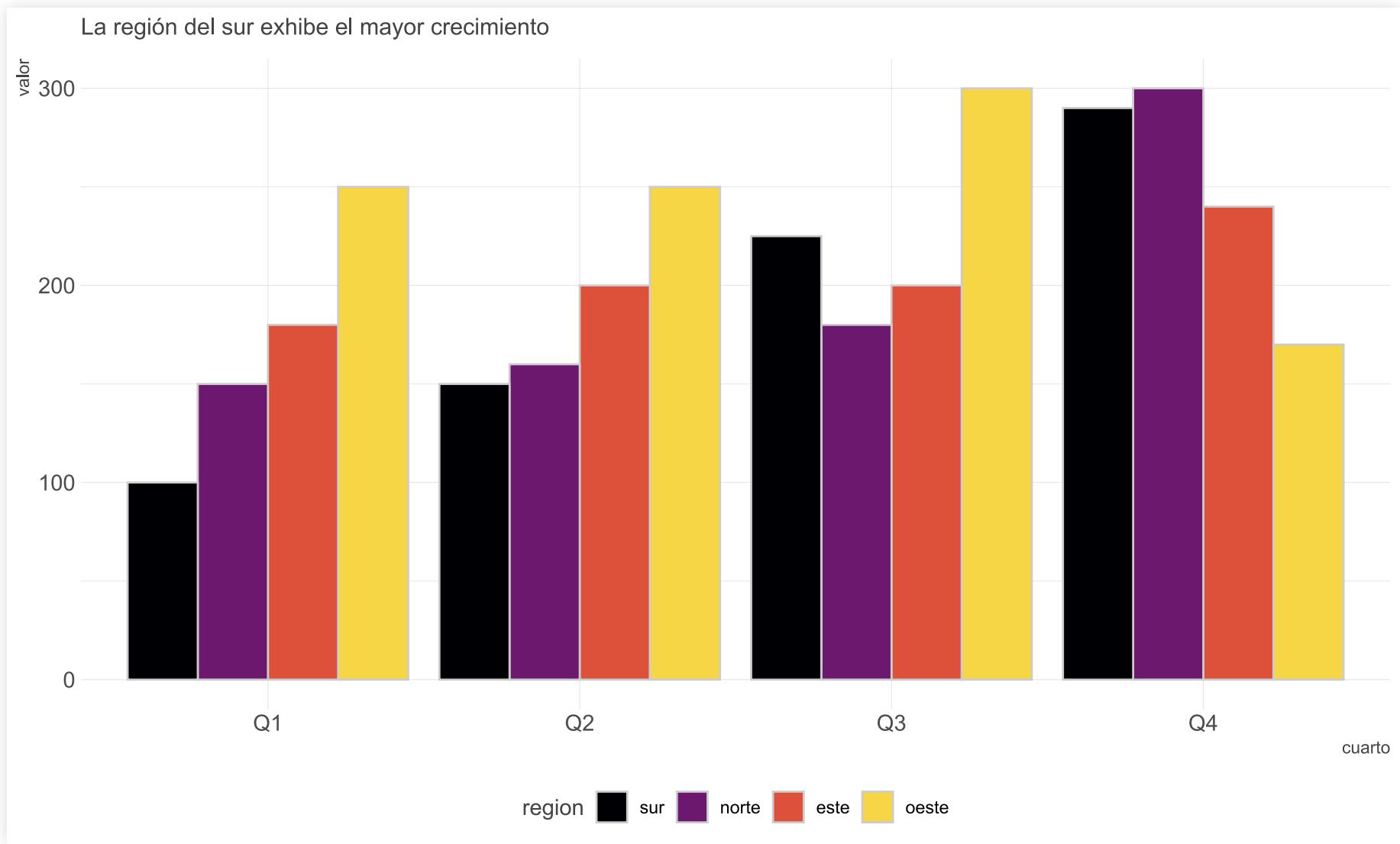
La región del sur exhibe el mayor crecimiento

Adapatado del tweet de [Lisa Charlotte Rost](#) que a su vez está viene del ejemplo del libro "Show me the numbers" de Stephen Few

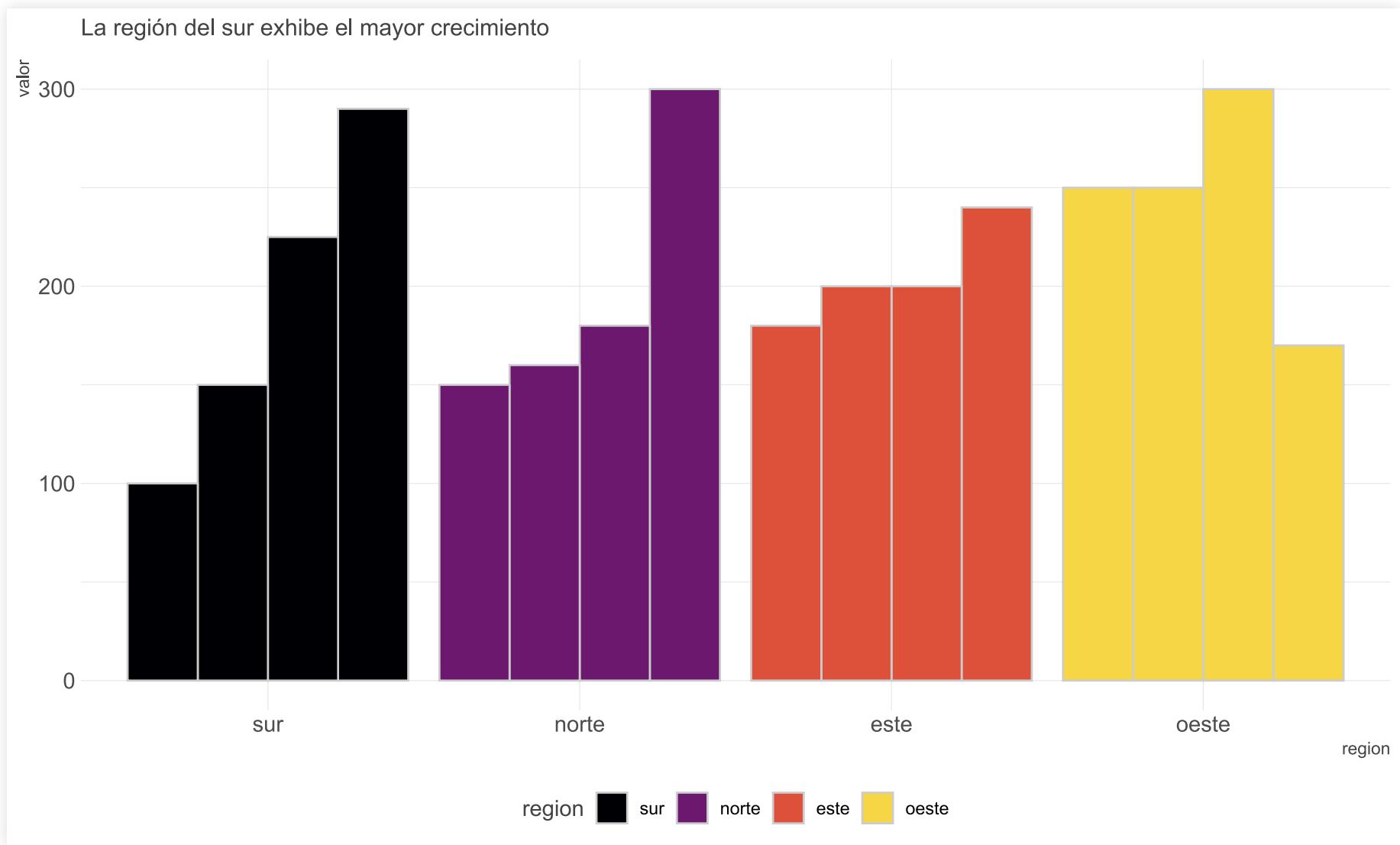
Los datos:

| region | Q1 | Q2 | Q3 | Q4 |
|--------|-----|-----|-----|-----|
| sur | 100 | 150 | 225 | 290 |
| norte | 150 | 160 | 180 | 300 |
| este | 180 | 200 | 200 | 240 |
| oeste | 250 | 250 | 300 | 170 |

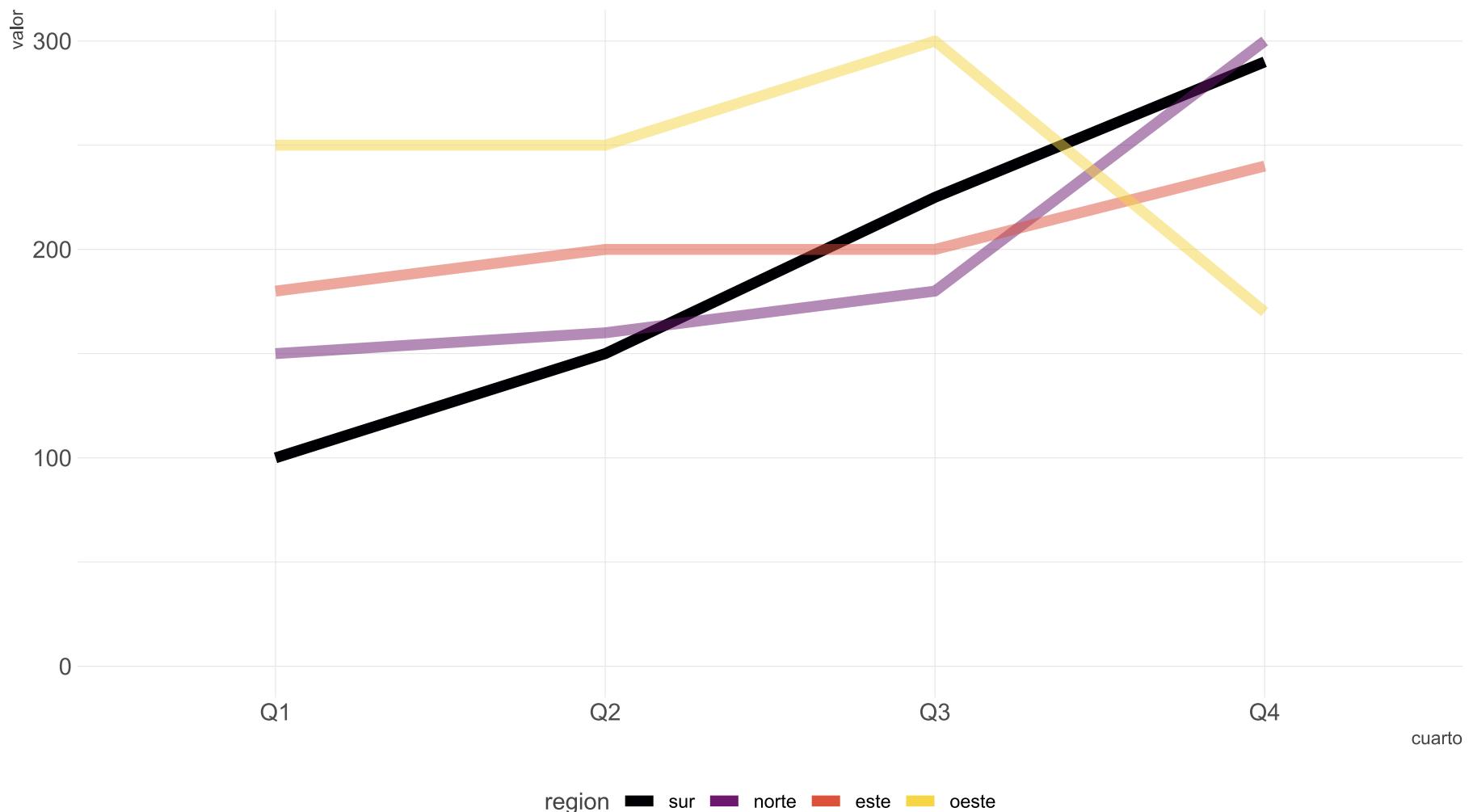
Opción #1

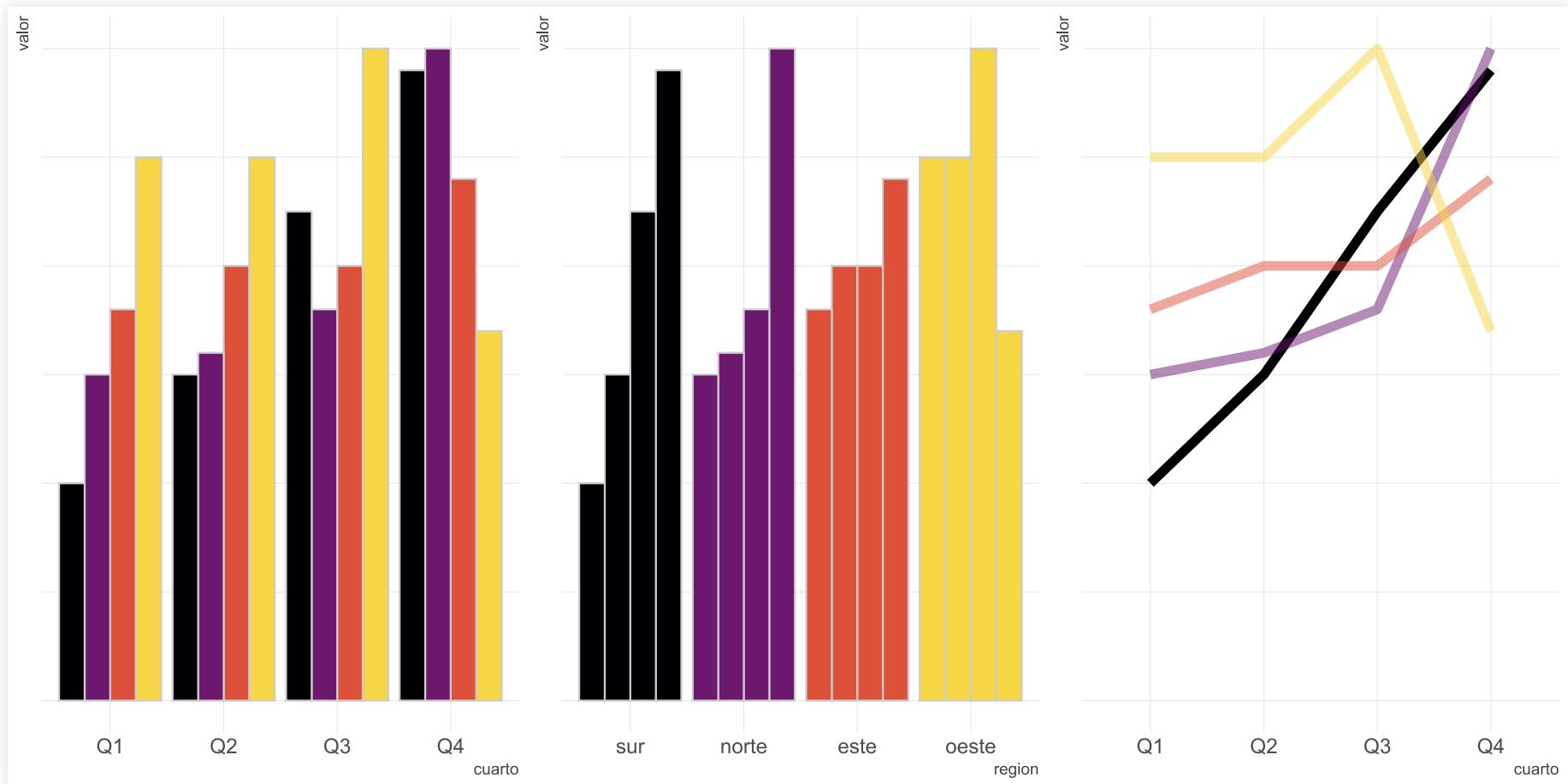


Opción #2



La región del sur exhibe el mayor crecimiento





(Posible) Respuesta

No. Dependerá de tu mensaje, de tu historia.

RECOMENDACIONES TÉCNICAS Y ALGUNAS MISCELÁNEAS

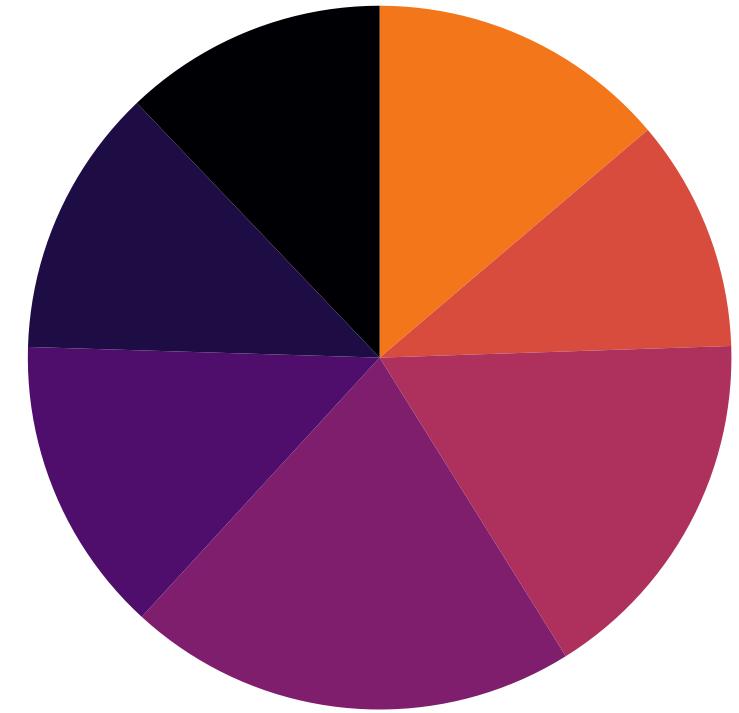
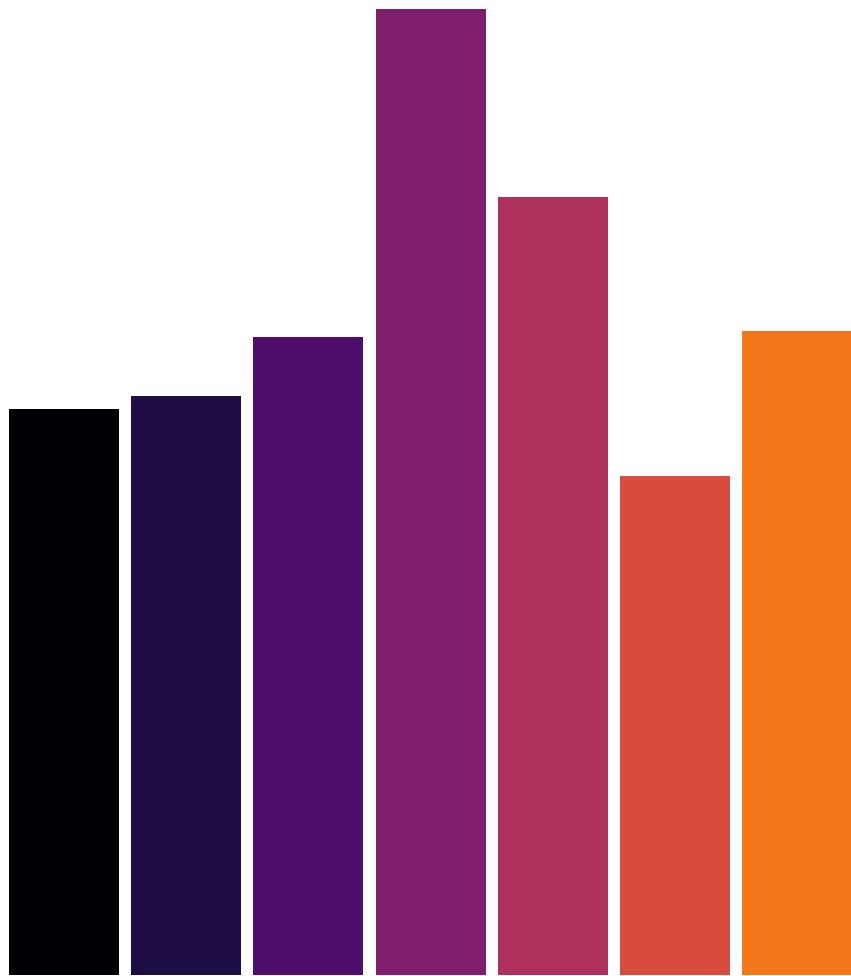
PIE CHARTS (O TORTAS)



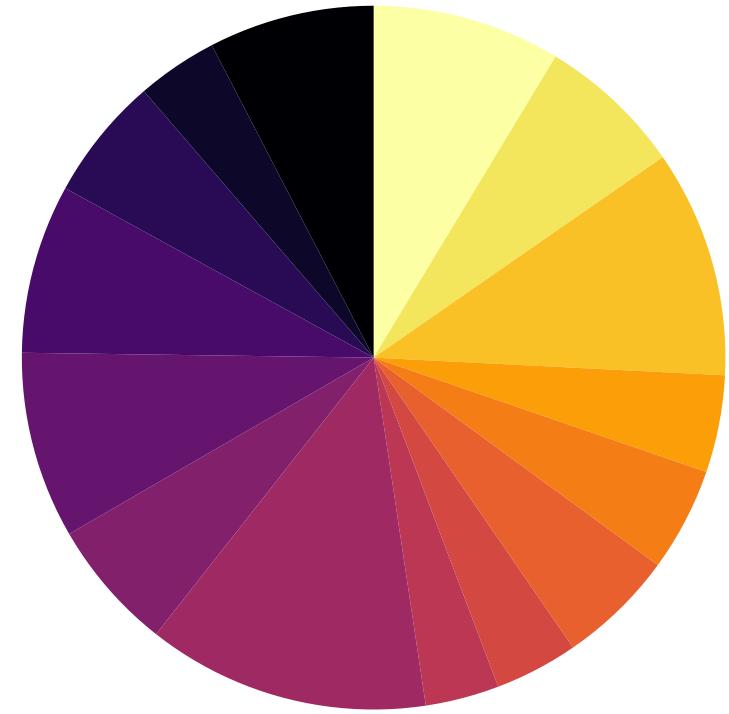
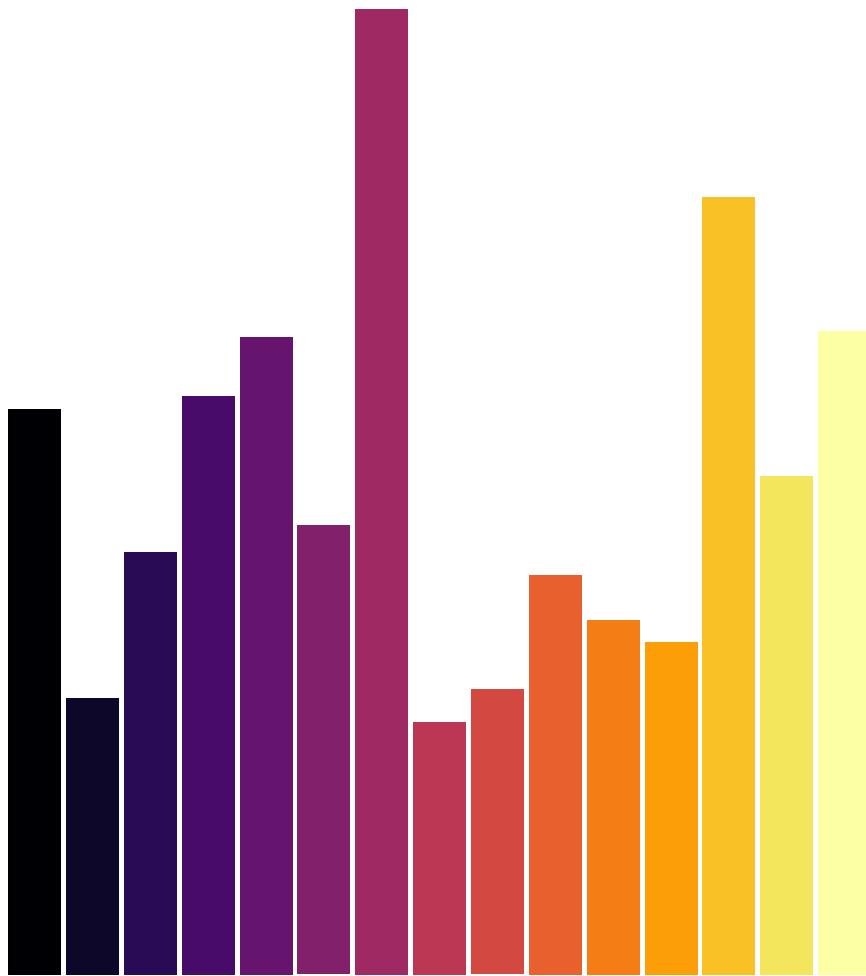
Usuales dificultades

- » Compara áreas
- » Ejercicio mental de rotar para comparar categorías

Es muy usado es el **pie chart**



Es muy (**ab**)usado es el **pie chart**



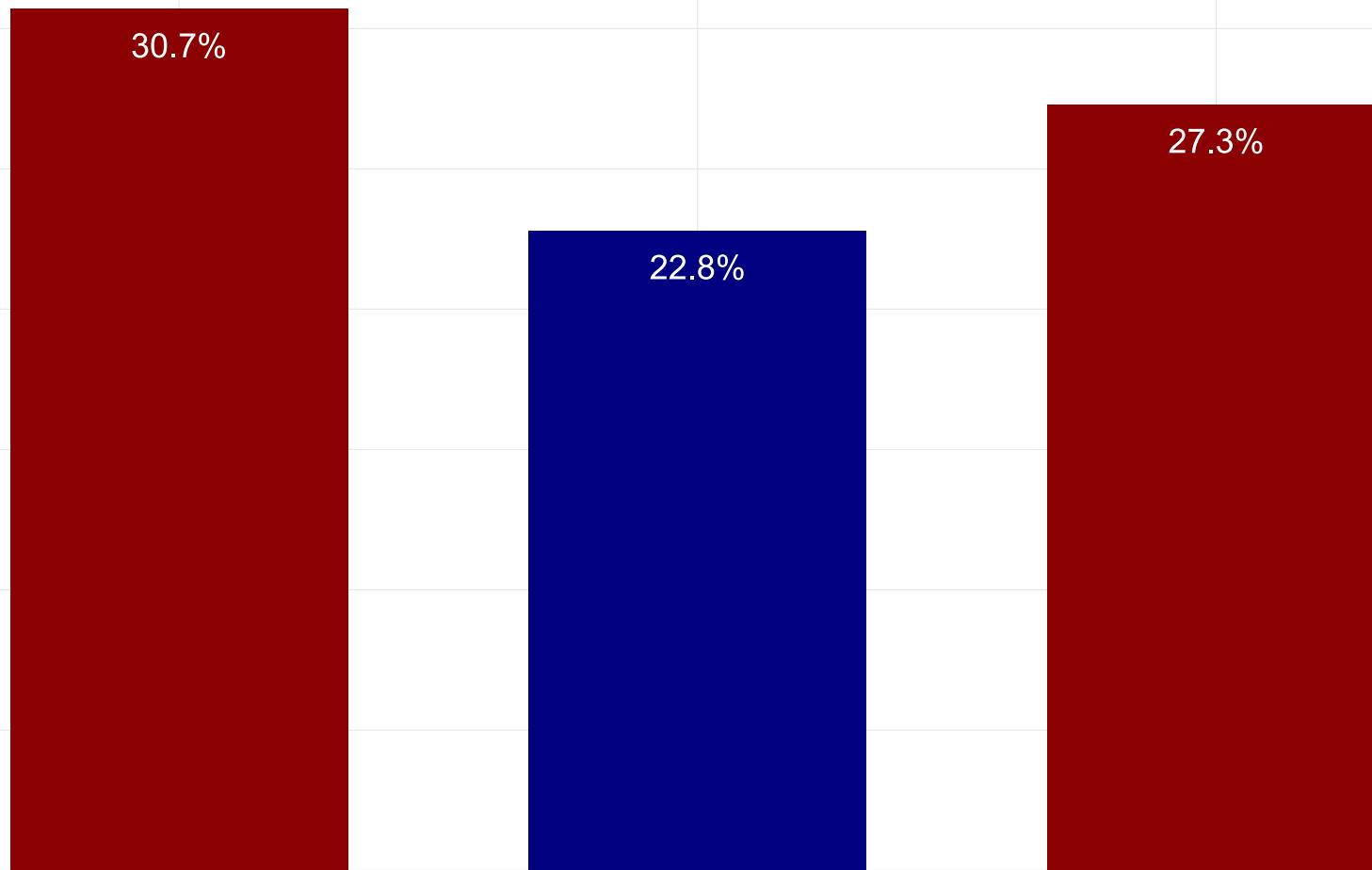
Remove
to improve
the **pie chart** edition

PUNTOS DE REFERENCIA





Índice



2010

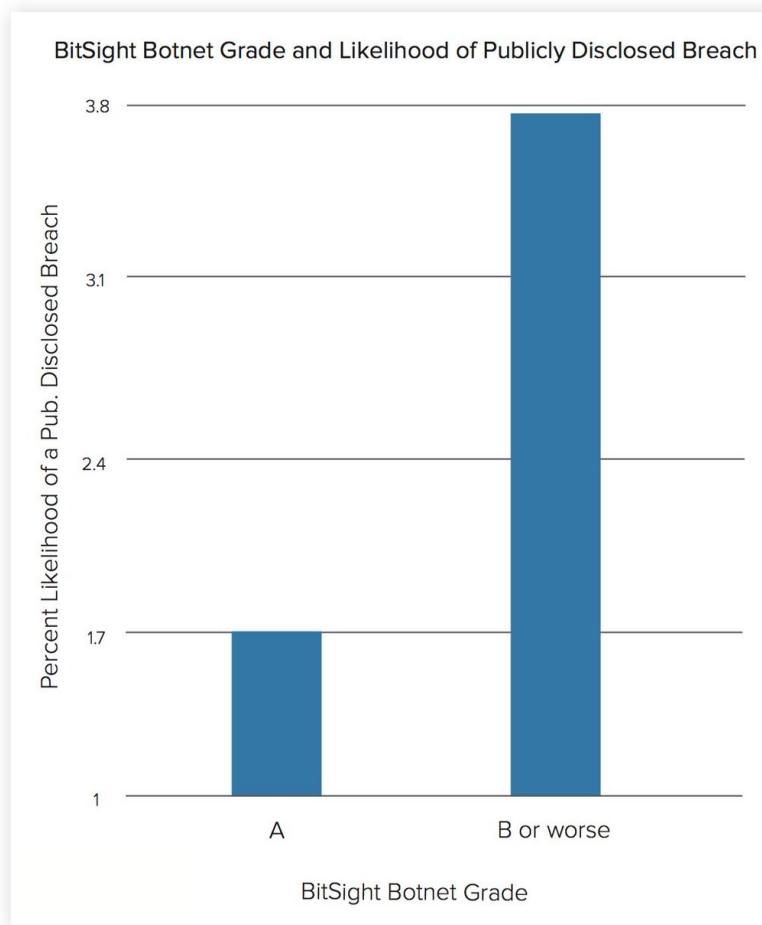
2013

2016

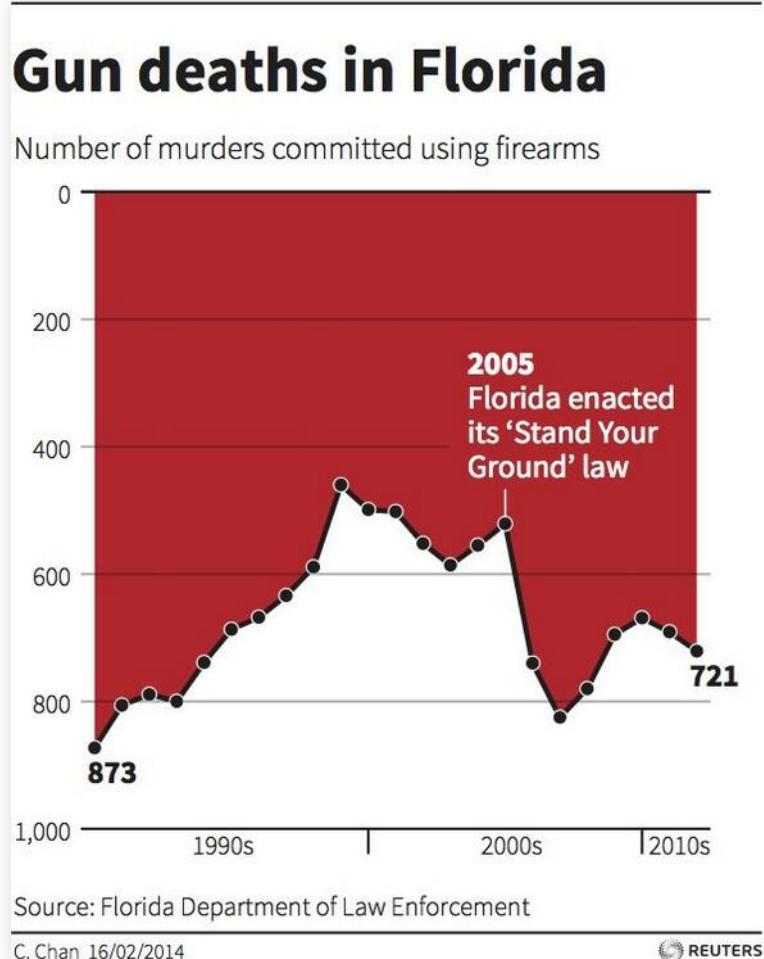
Año

CONVENCIONES

Ejercicio 2. Encuentre las *anomalías*



Rápidamente ¿Dónde se produce el máximo?



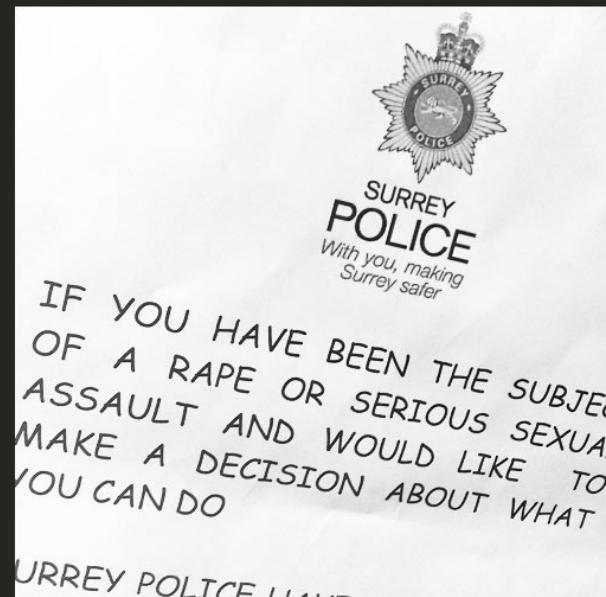
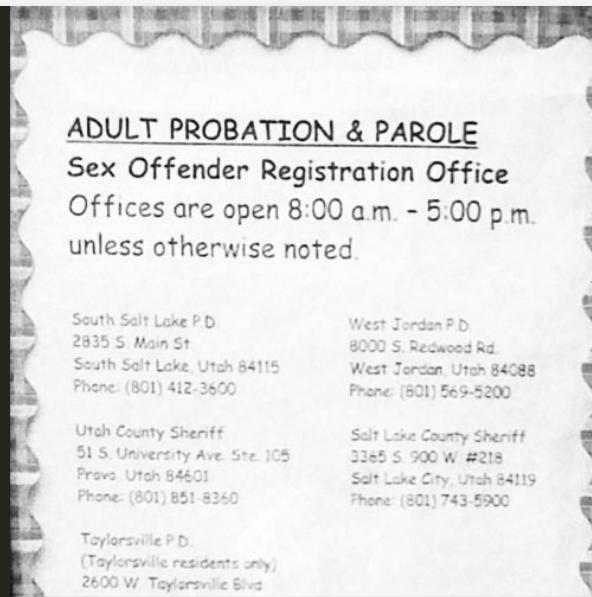
TIPOGRAFÍA

¿Conocen a ...?

Arial

Times New Roman

Comic Sans



Según **comic sans criminal**:

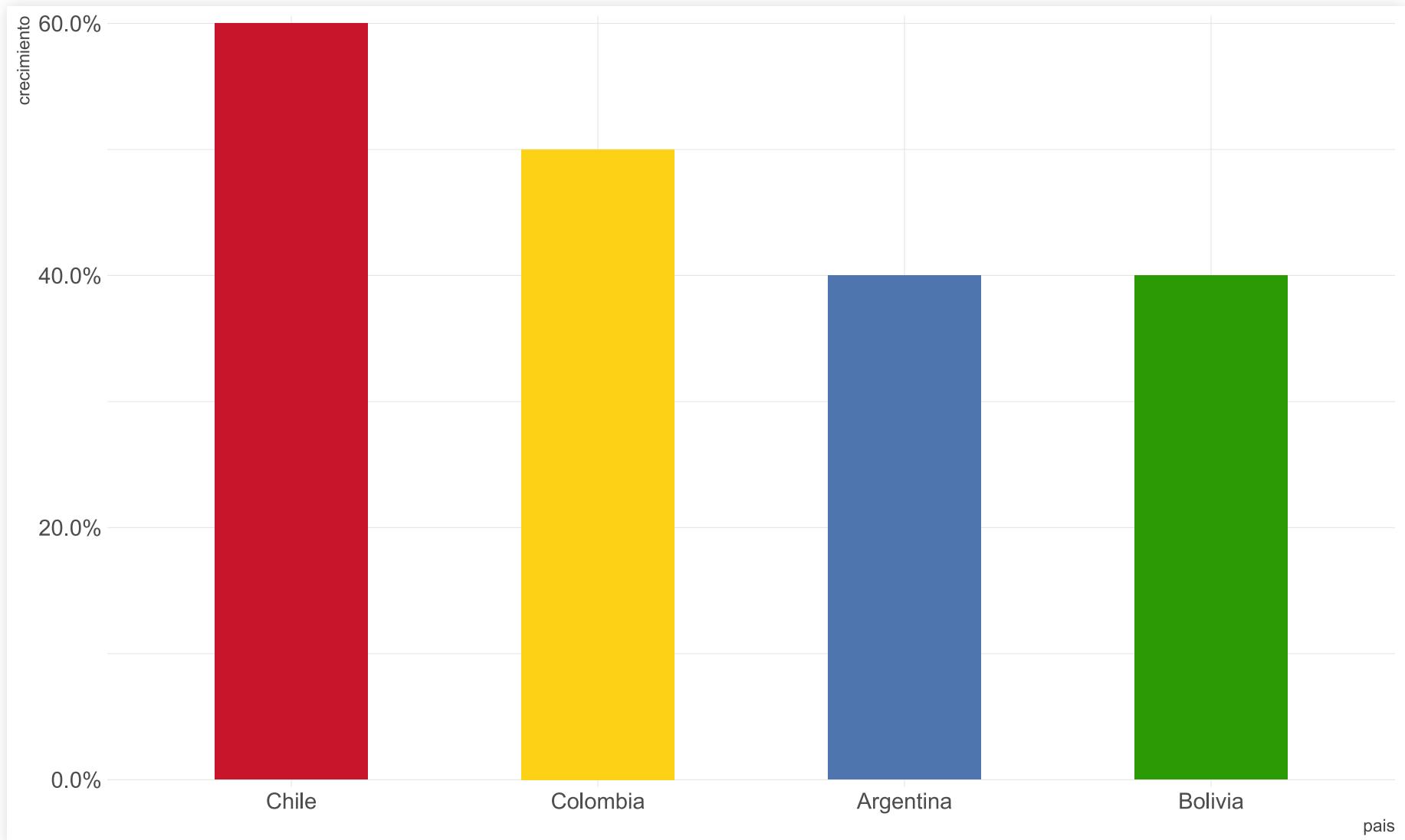
- » Fuentes tienen personalidad
- » Tiene un propósito
- » Armonía en el universo

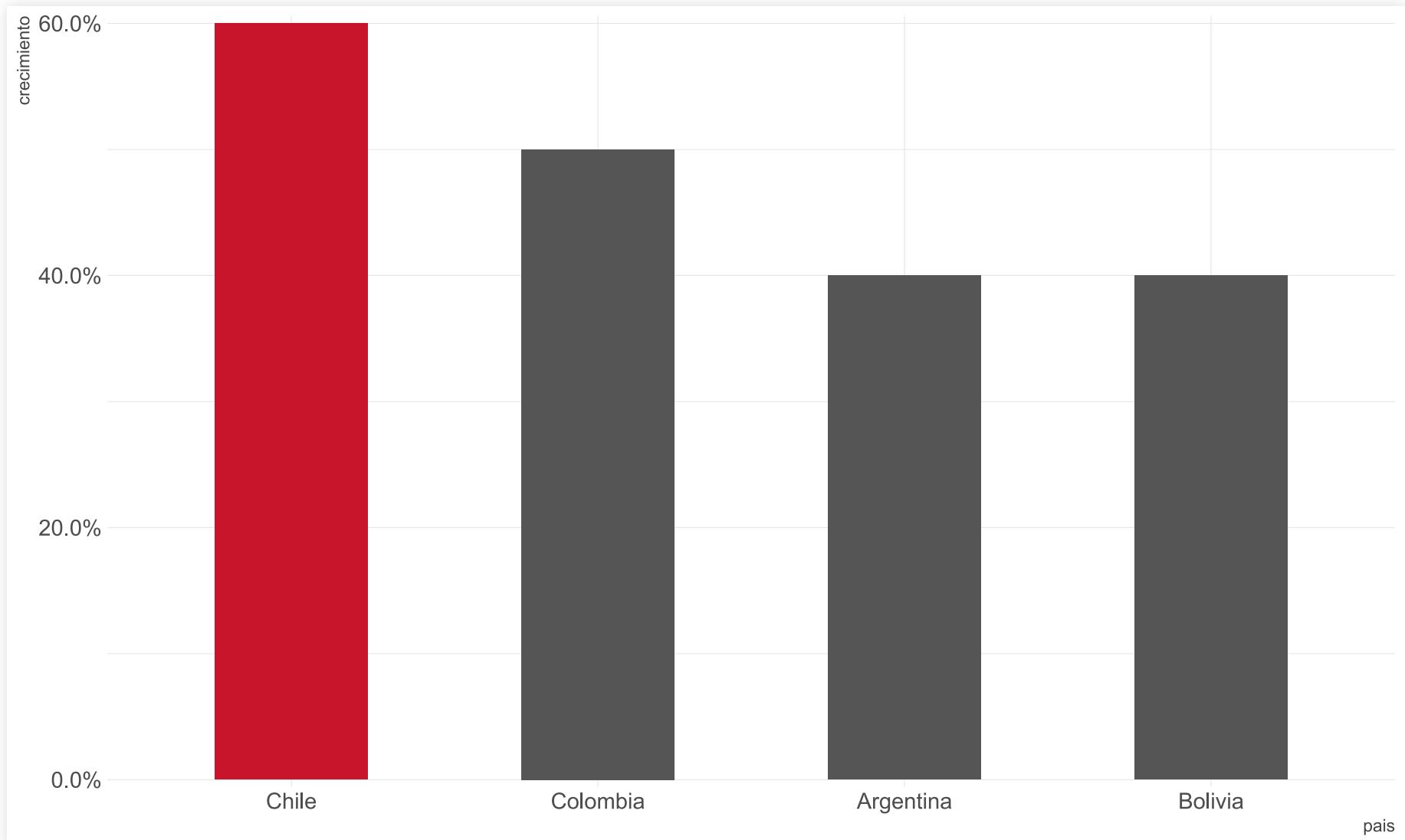
COLORES

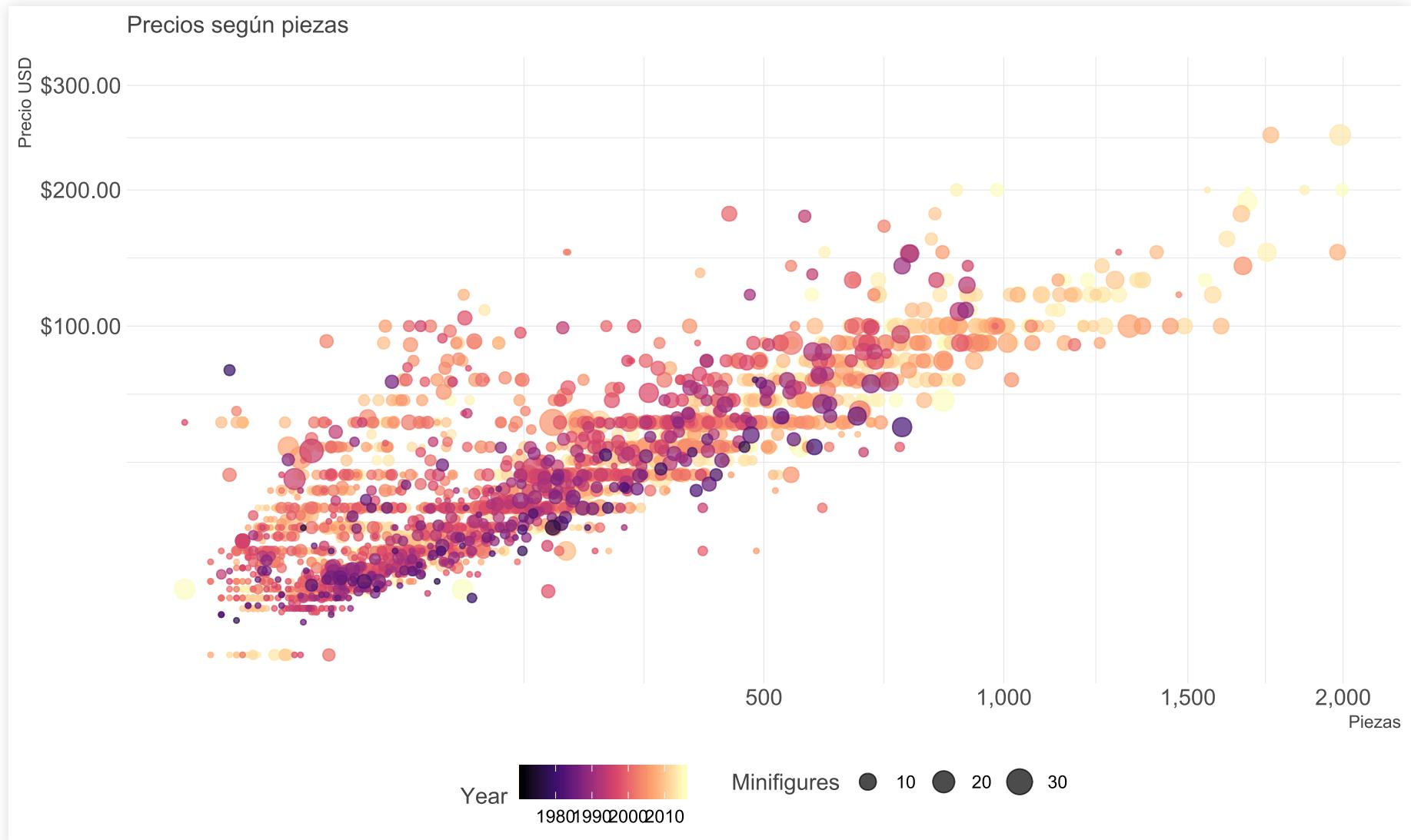
La misma idea de las convenciones!

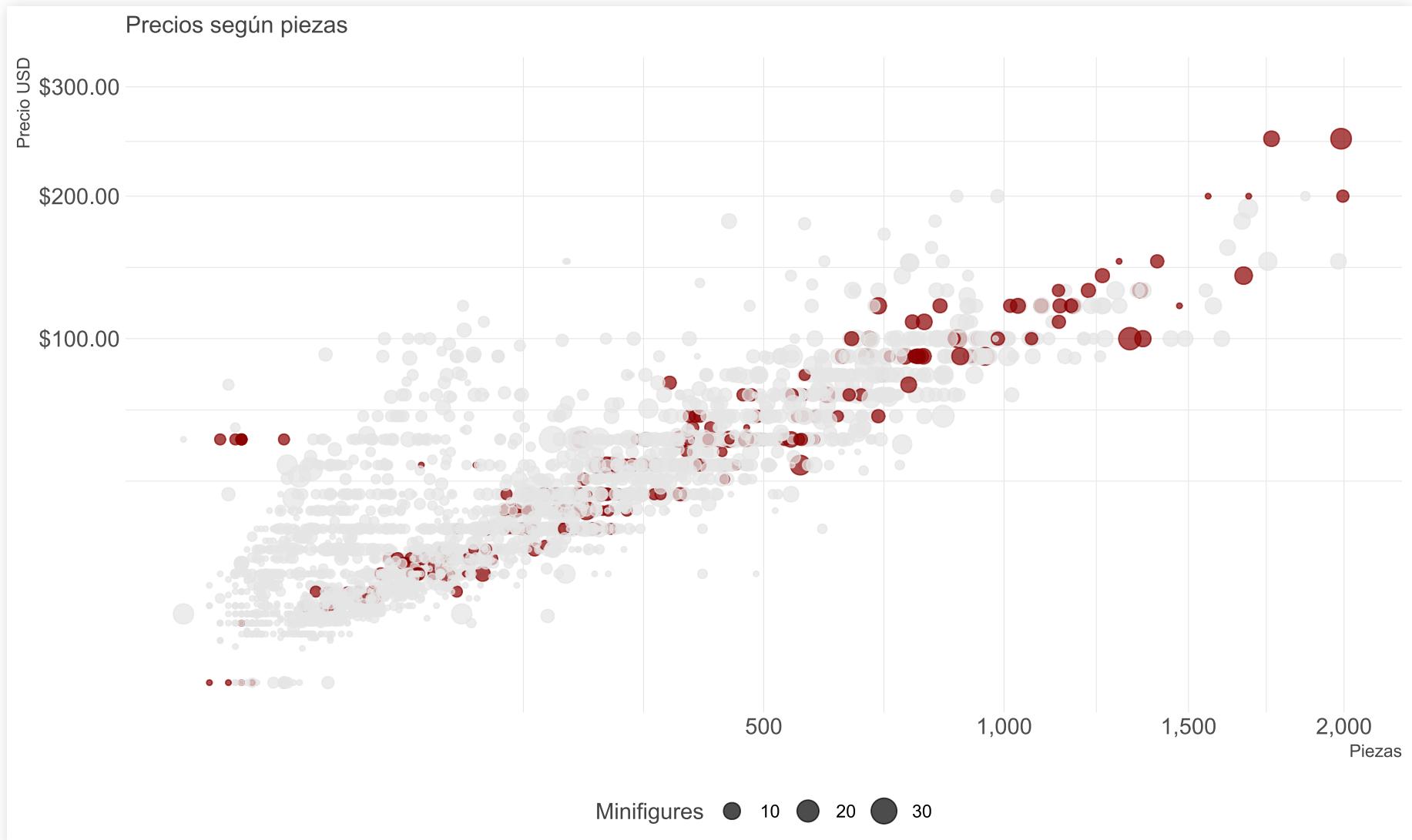
1 2 3 4 5 6

- » Dar foco y llamar la atención
- » Deben ser coherentes con el dato

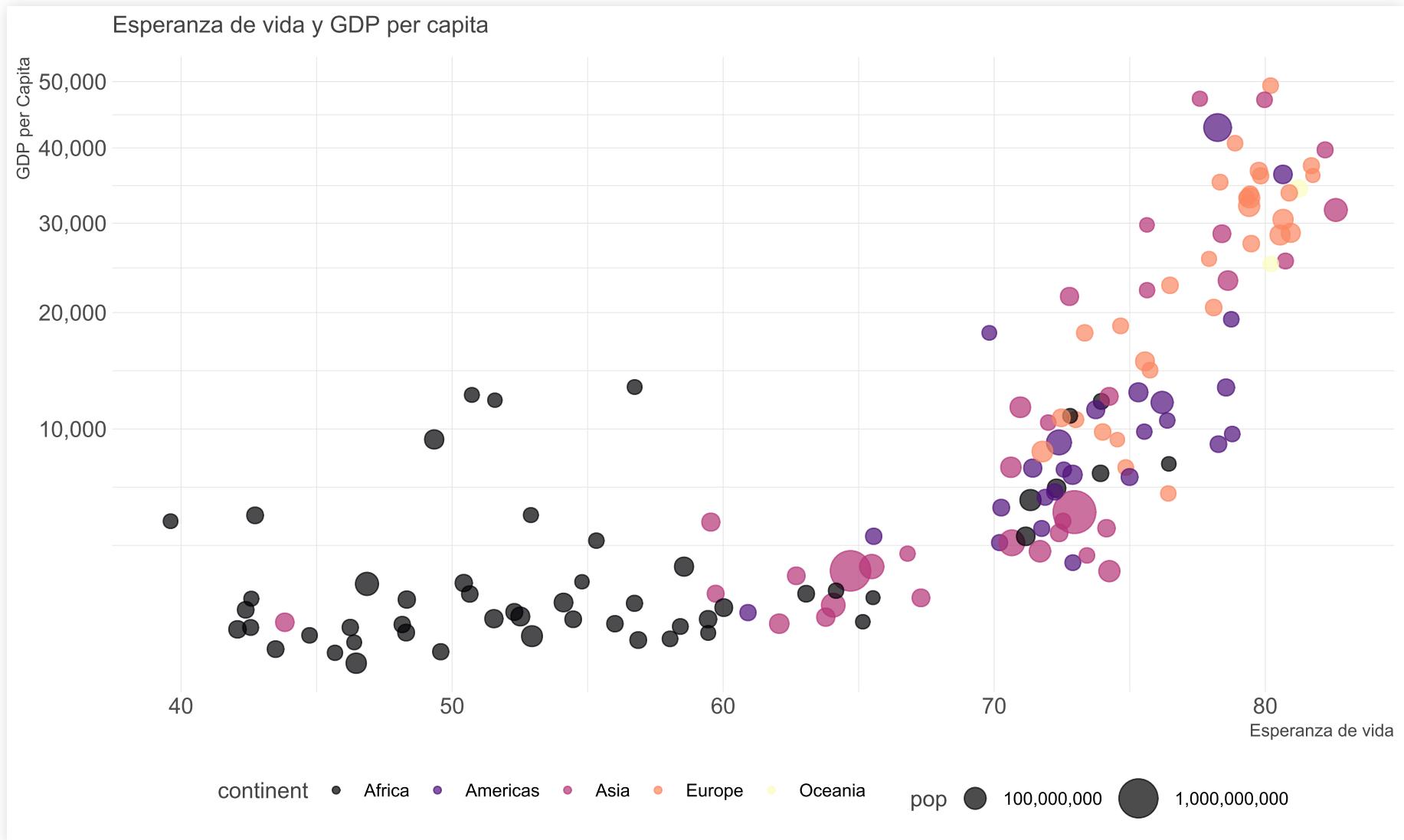




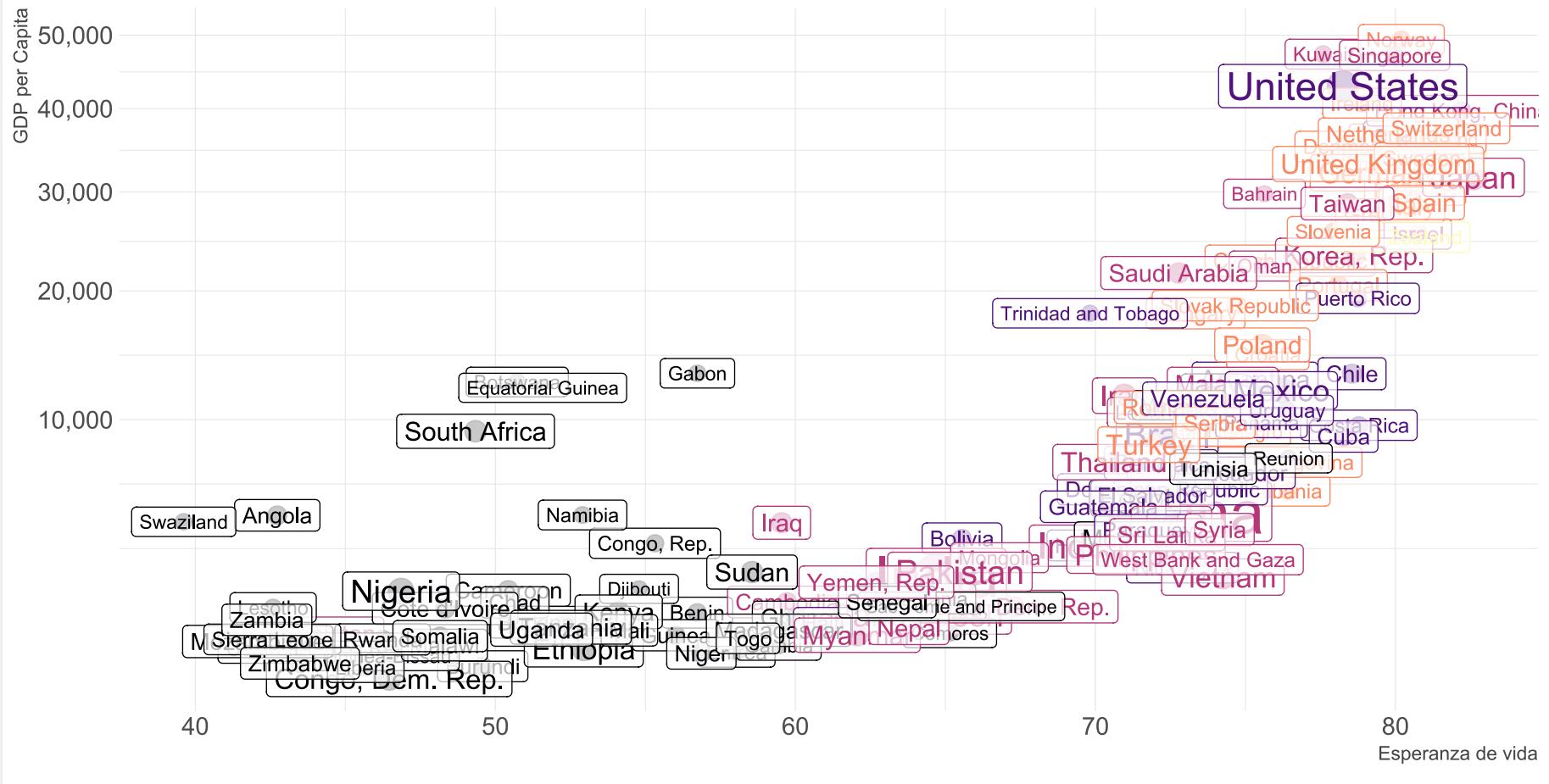




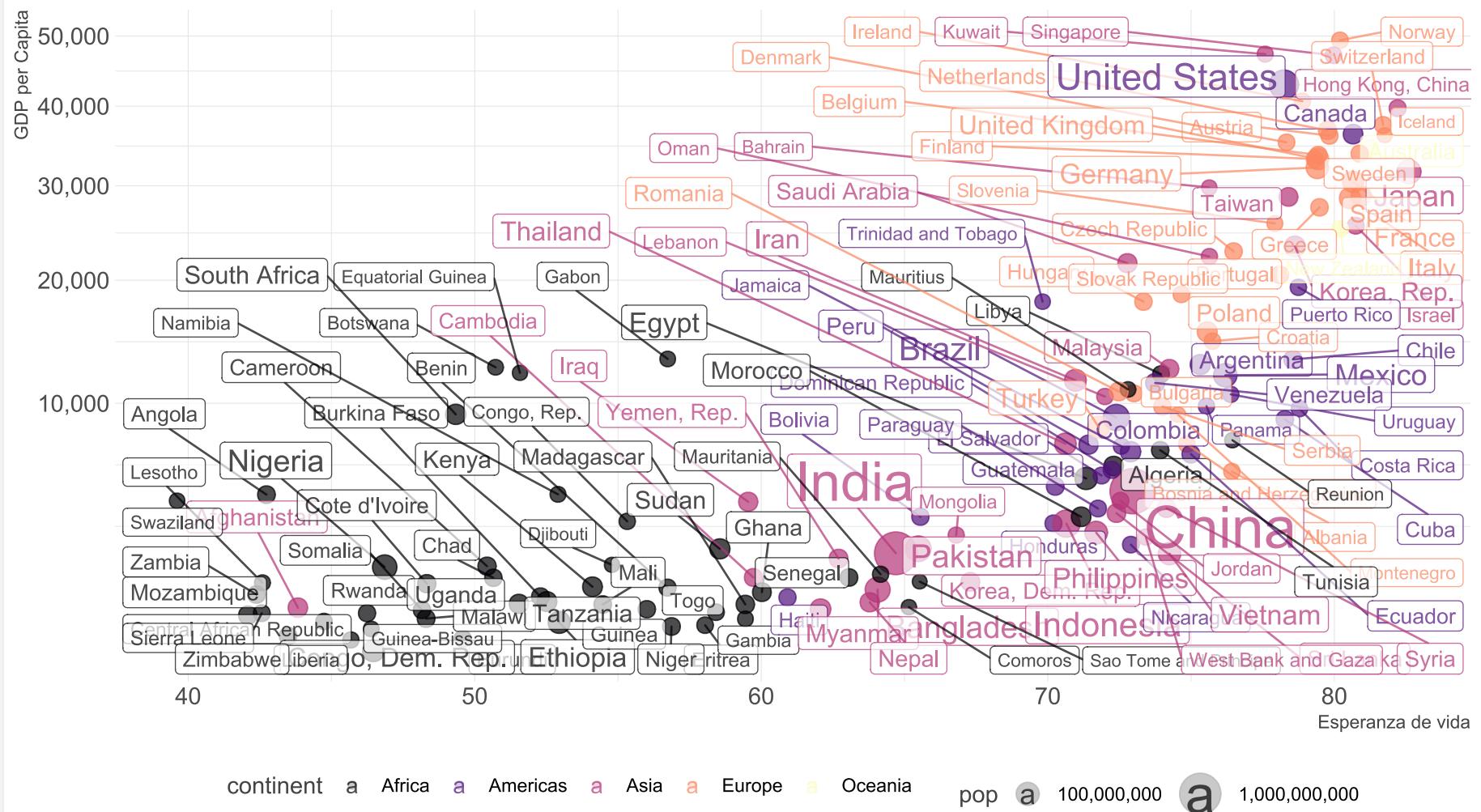
ETIQUETAS

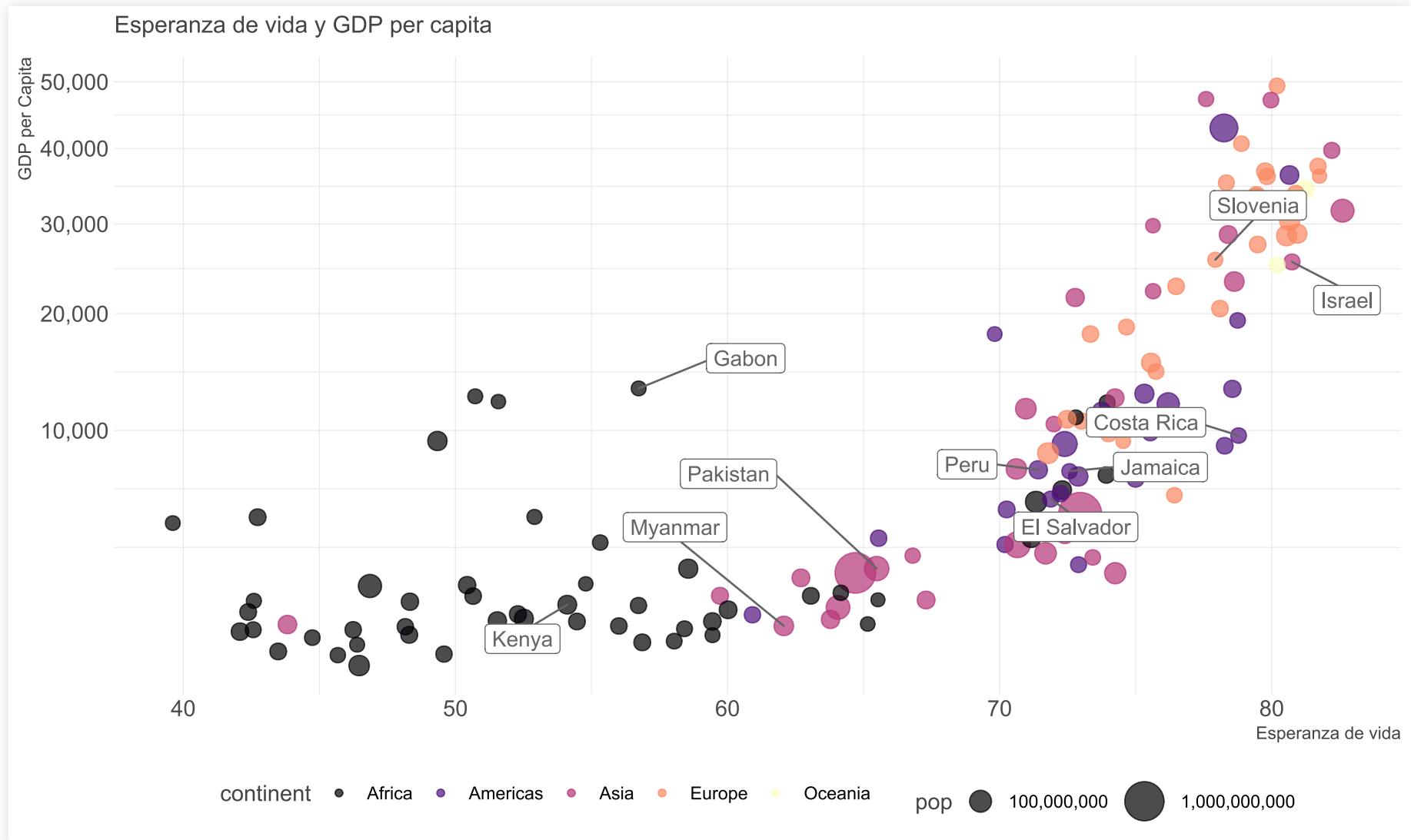


Esperanza de vida y GDP per capita



Esperanza de vida y GDP per capita





VISUALIZANDO CON GG PLOT2

GGPLOT2

Características:

- » Paquete para visualizar datos mediante capas
- » Es muy poderoso y flexible
- » Se carga junto al `tidyverse`
- » No es la única opción en R para graficar

DATOS

```
theme_set(theme_gray())
```

```
library(gapminder)
data(gapminder)
paises <- gapminder %>%
  filter(year == max(year))
paises
```

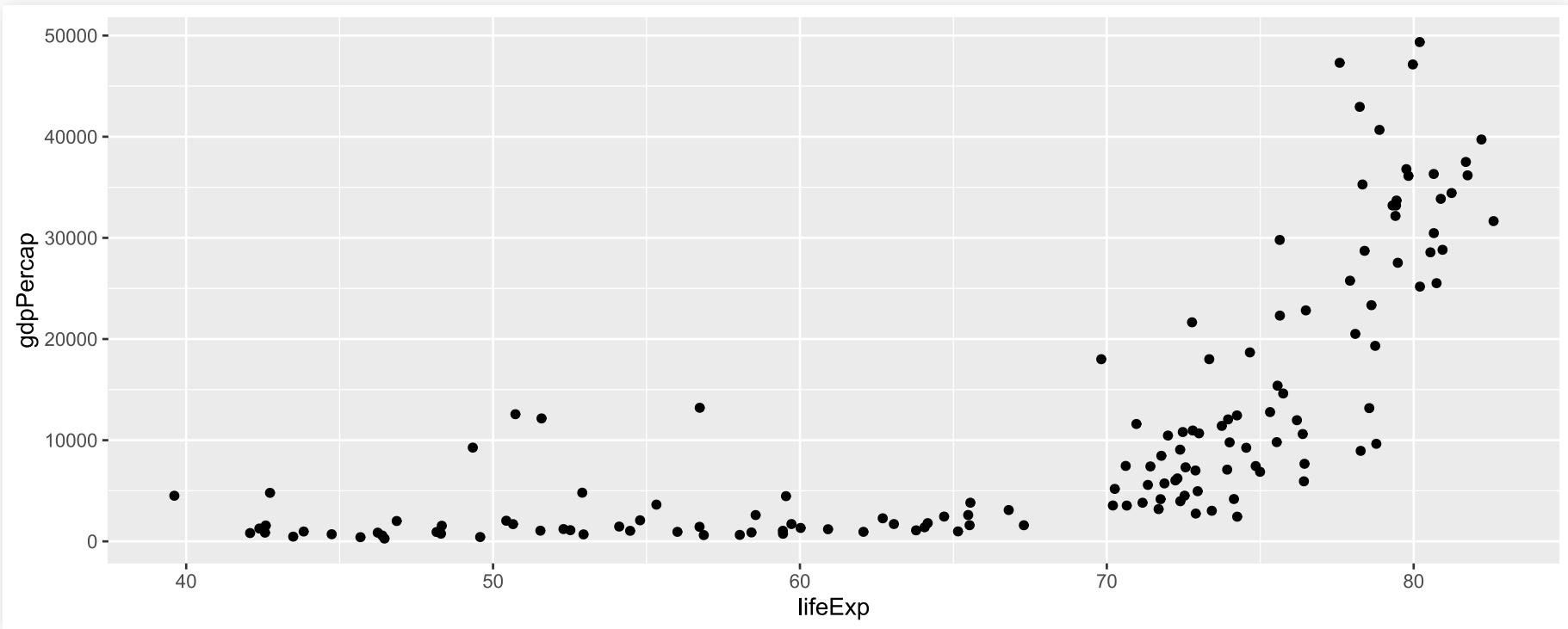
```
## # A tibble: 142 x 6
##   country     continent year lifeExp      pop gdpPercap
##   <fct>       <fct>    <int>   <dbl>     <int>     <dbl>
## 1 Afghanistan Asia     2007     43.8  31889923     975.
## 2 Albania      Europe   2007     76.4  3600523      5937.
## 3 Algeria      Africa   2007     72.3  33333216     6223.
## 4 Angola       Africa   2007     42.7  12420476     4797.
## 5 Argentina    Americas 2007     75.3  40301927    12779.
## 6 Australia    Oceania  2007     81.2  20434176    34435.
## 7 Austria      Europe   2007     79.8  8199783     36126.
## 8 Bahrain      Asia     2007     75.6  708573      29796.
## 9 Bangladesh   Asia     2007     64.1  150448339    1391.
## 10 Belgium     Europe   2007     79.4  10392226    33693.
## # ... with 132 more rows
```

CREANDO UN GRÁFICO :)

```
ggplot(data = paises)
```

MEJORANDO UN GRÁFICO

```
ggplot(data = paises) +  
  geom_point(mapping = aes(x = lifeExp, y = gdpPercap))
```



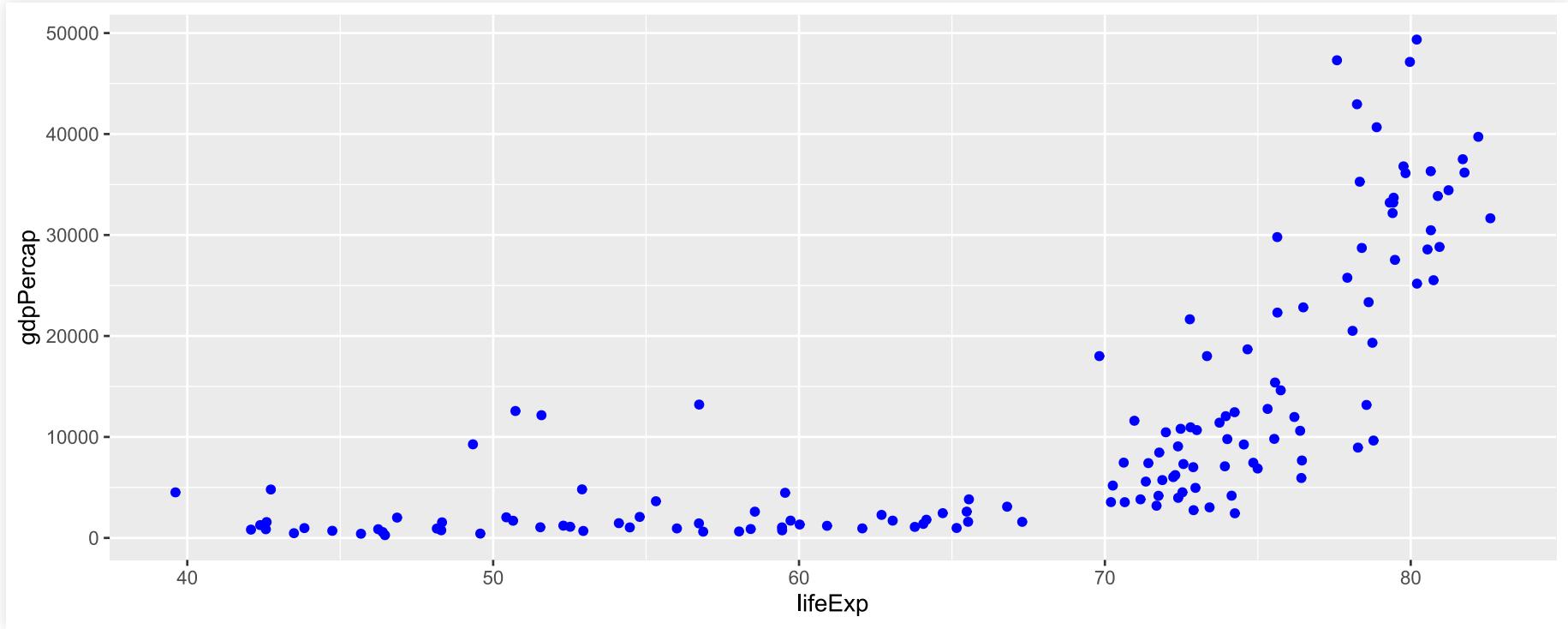
QUE SUCEDIÓ?

```
ggplot(data = paises) +  
  geom_point(mapping = aes(x = lifeExp, y = gdpPercap))
```

- » `ggplot()` crea un sistema de coordenadas al cual se pueden agregar capas
- » `ggplot(data = paises)` da un grafico vacío pues no agregamos capas
- » `geom_point()` agrega una capa de puntos al gráfico usando las filas de `paises`
- » Cada función `geom_algo` tiene un argumento de mapping que define cómo se asignan o se “mapean” las variables del conjunto de datos a propiedades visuales del `geom_algo`
- » El argumento de mapping siempre aparece emparejado con `aes()`, y los argumentos `x` e `y` especifican qué variables asignar a los ejes `x` e `y`

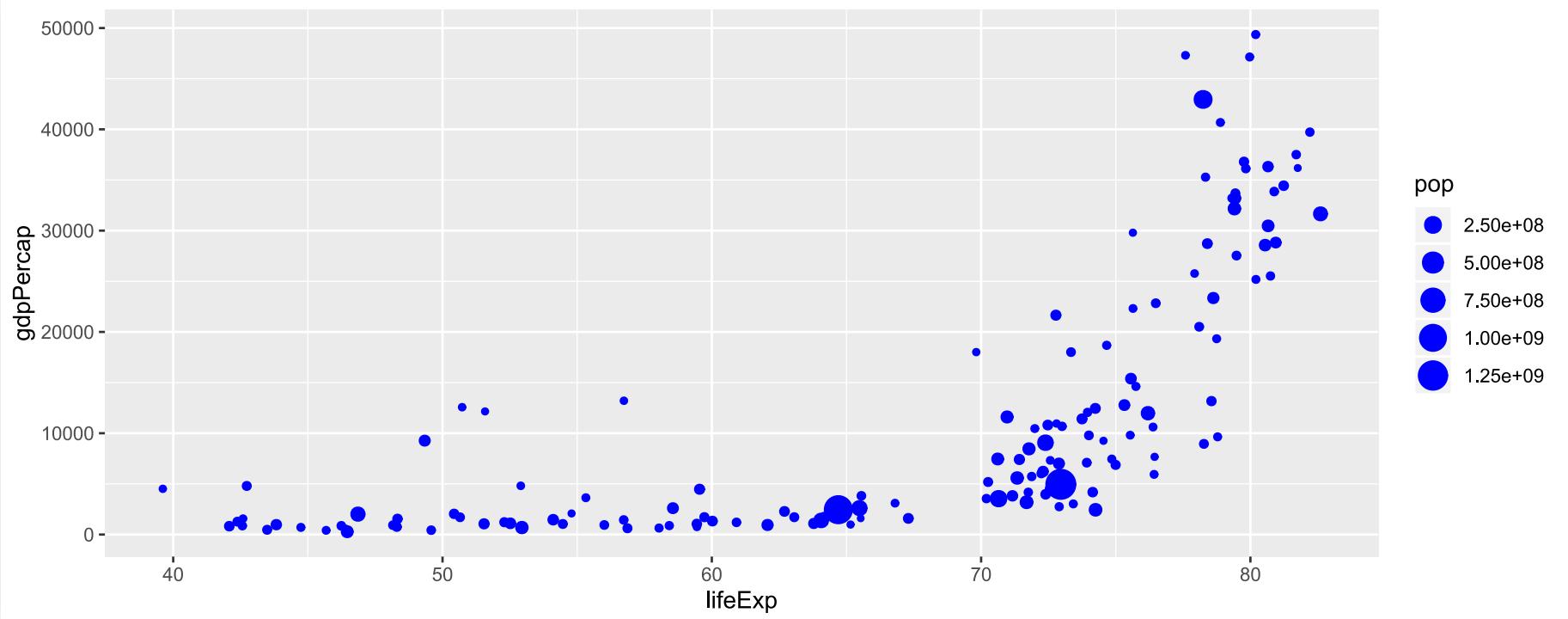
Podemos setear las propiedades estéticas de tu geom manualmente:

```
ggplot(data = paises) +  
  geom_point(mapping = aes(x = lifeExp, y = gdpPercap), color = "blue")
```



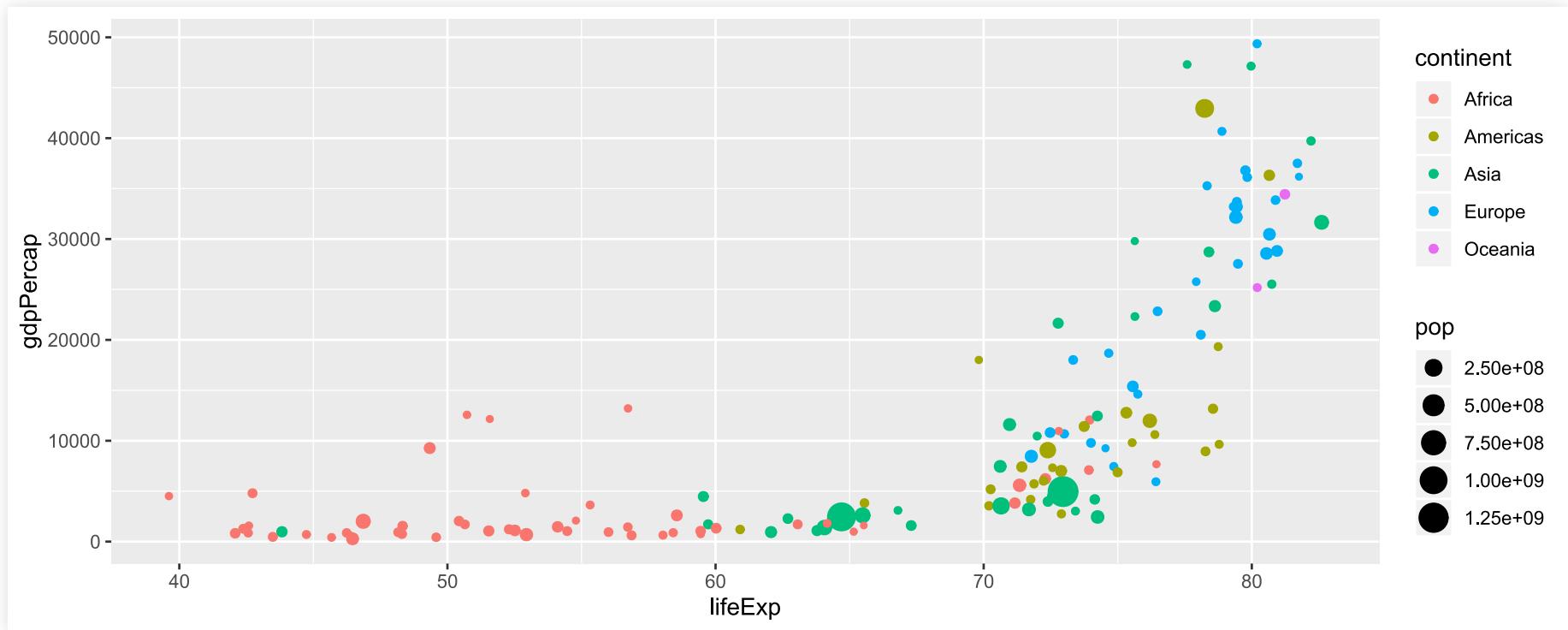
En este caso un punto no solo puede poseer x e y, puede tener tamaño dado por una variable

```
ggplot(data = paises) +  
  geom_point(mapping = aes(x = lifeExp, y = gdpPercap, size = pop), color = "blue")
```



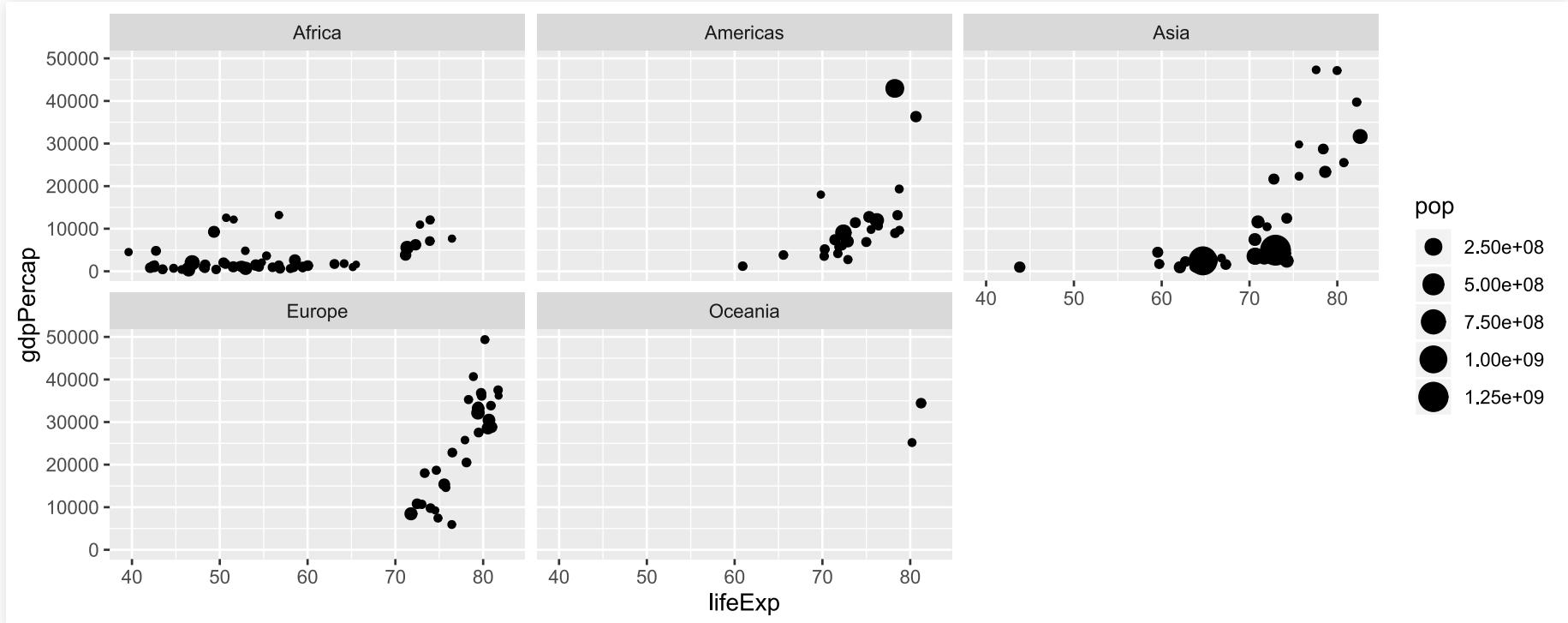
Quizás en lugar de setear color fijo, podemos asignarlo segun una variable

```
ggplot(data = paises) +  
  geom_point(mapping = aes(x = lifeExp, y = gdpPerCap, size = pop, color = continent))
```



O realizar *facets/paneles*

```
ggplot(data = paises) +  
  geom_point(mapping = aes(x = lifeExp, y = gdpPercap, size = pop)) +  
  facet_wrap(vars(continent))
```



EXISTEN MUCHOS TIPOS DE GRÁFICOS
DISPONIBLES EN `ggplot2`. VÉAMOS
ALGUNOS EJEMPLOS EN R!