Ciencia de Datos para Políticas Públicas

Clase 02 - Visualización de datos

Pablo Aguirre Hormann

12/08/2020

¿Qué veremos hoy?

- ¿Qué es y por qué visualizar datos?
- Visualización con `ggplot2`
- Demostraciones
  - `Clase02_CodigoViz.R`
  - `Clase02_CodigoCovid.R`
- Ejercicio
  - `Clase02_Ejercicio.R`
- Tarea

Antes de empezar

- ¿Alguna pregunta?

¿Qué es y por qué visualizar datos?

Dos razones principales de por qué visualizar

- **Para uno**: entender los datos con el fin de guiar análisis posteriores (análisis exploratorio)
- **Para otros**: contar una historia sobre los datos y resultados con el fin de comunicar algo

Lo que deberíamos buscar

- Mostrar los datos
- No mentir con estos
- Contar una historia (¿una relación? ¿causalidad? ¿un patrón? ¿un quiebre?)
- Reducir el ruido (o lo innecesario)
- Transmitir y convencer
- Visualizaciones deben complementar el texto y tener suficiente información para “sobrevivir por sí mismas”

¿Qué opinan?

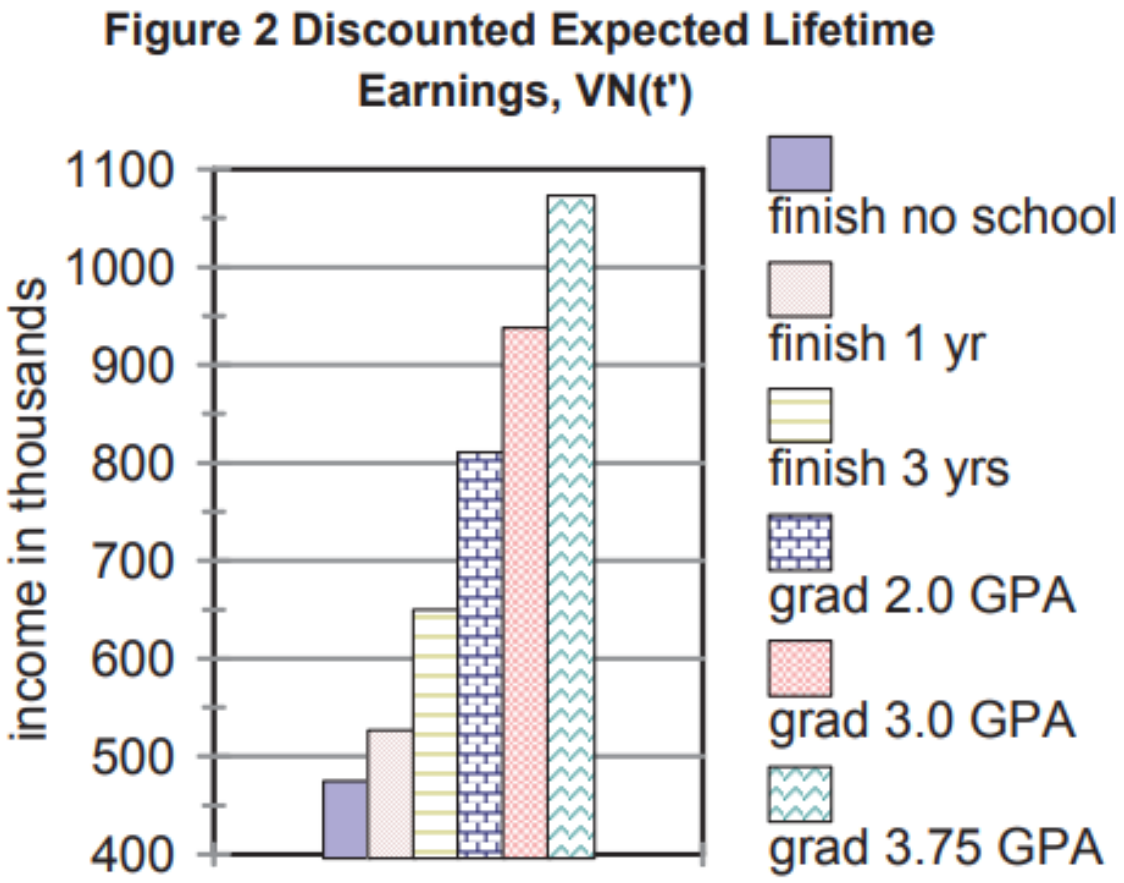


Figure 2 Discounted Expected Lifetime Earnings, VN(t')
income in thousands
1100
1000
900
800
700
600
500
400
finish no school
finish 1 yr
finish 3 yrs
grad 2.0 GPA
grad 3.0 GPA
grad 3.75 GPA

Una idea general

- El cerebro solo puede procesar un cierto número de atributos de forma instantanea (*pre-attentive attributes*)
  - Forma, posición, color, tamaño
- Queremos buscar la variación justa en estos atributos para enfocarnos en lo que importa

¿Cuántos 3 hay?

$$126954852361235698745824 5$$
$$012403698570206956831278 1$$
$$243986201247813698217325 6$$

¿Cuántos 3 hay?

1269548523612356987458245
0124036985702069568312781
2439862012478136982173256

¿Y ahora?

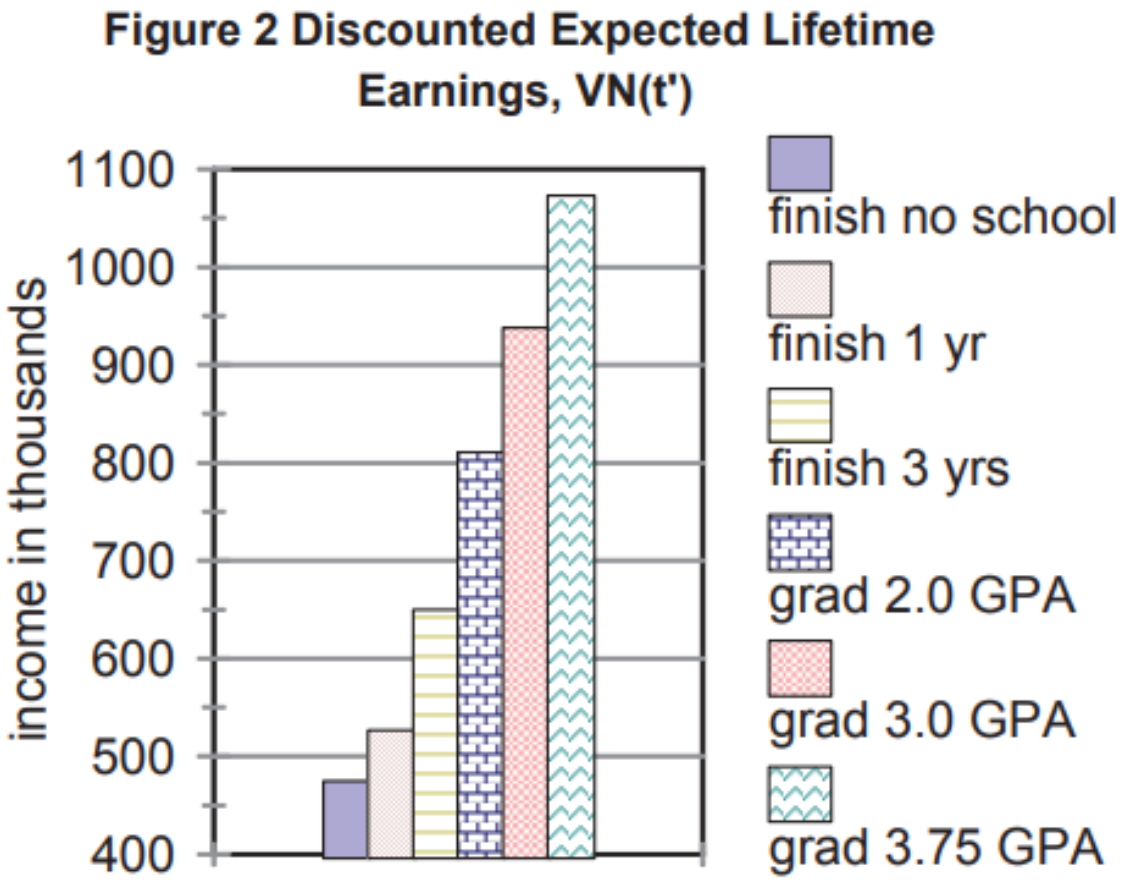
1269548523612356987458245
0124036985702069568312781
2439862012478136982173256

Principios de (Edward) Tufte

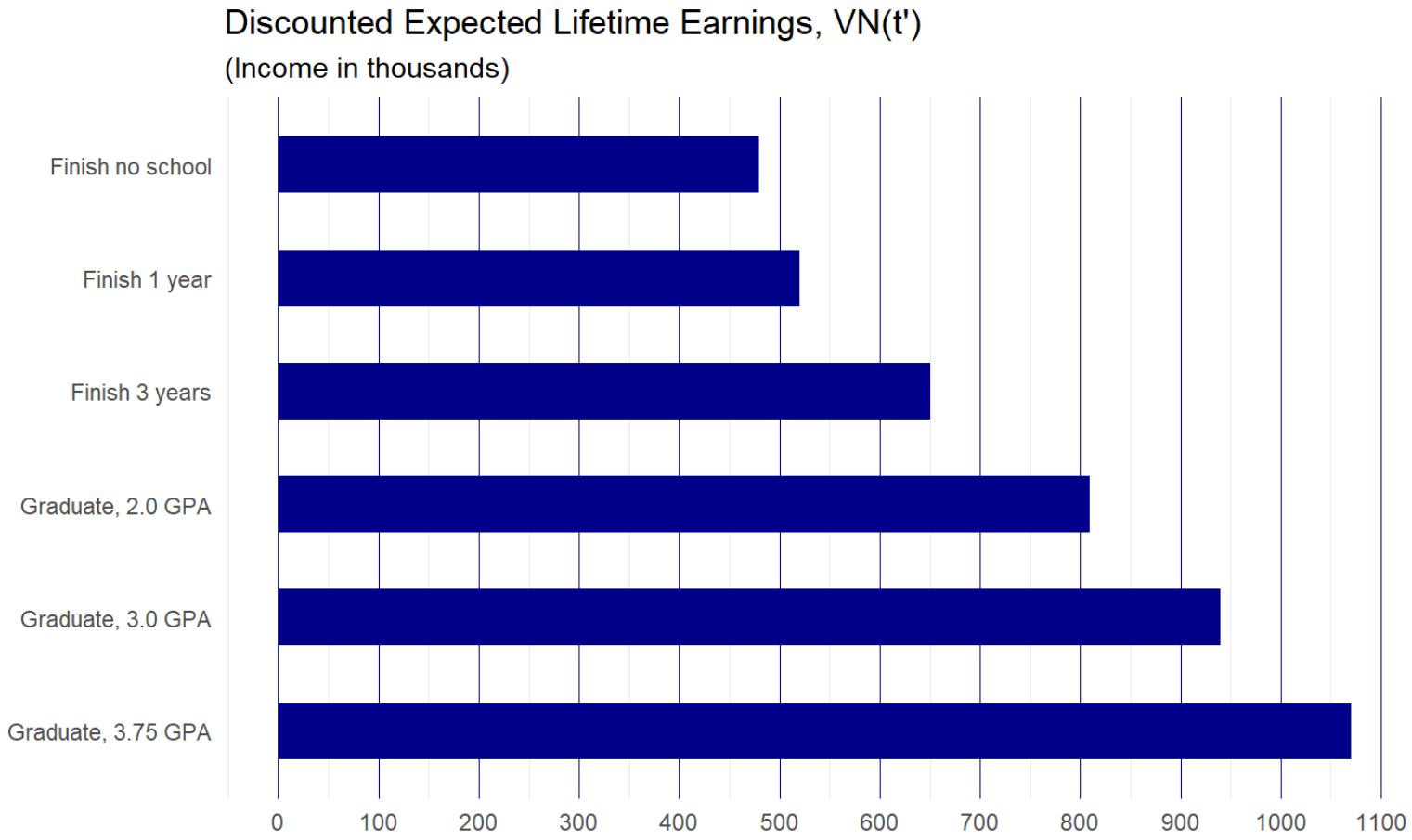
https://www.edwardtufte.com/tufte/

- Muestra los datos
- Maximiza la razón datos/*tinta*
- Borra (lo más posible) la *tinta* que no corresponda a datos
- Borra *tinta* redundante
- Evita la *basura visual*
- Gráficos deben tender a ser horizontales

Apliquemos los principios



Tratemos de hacerlo mejor



Que hay por detrás

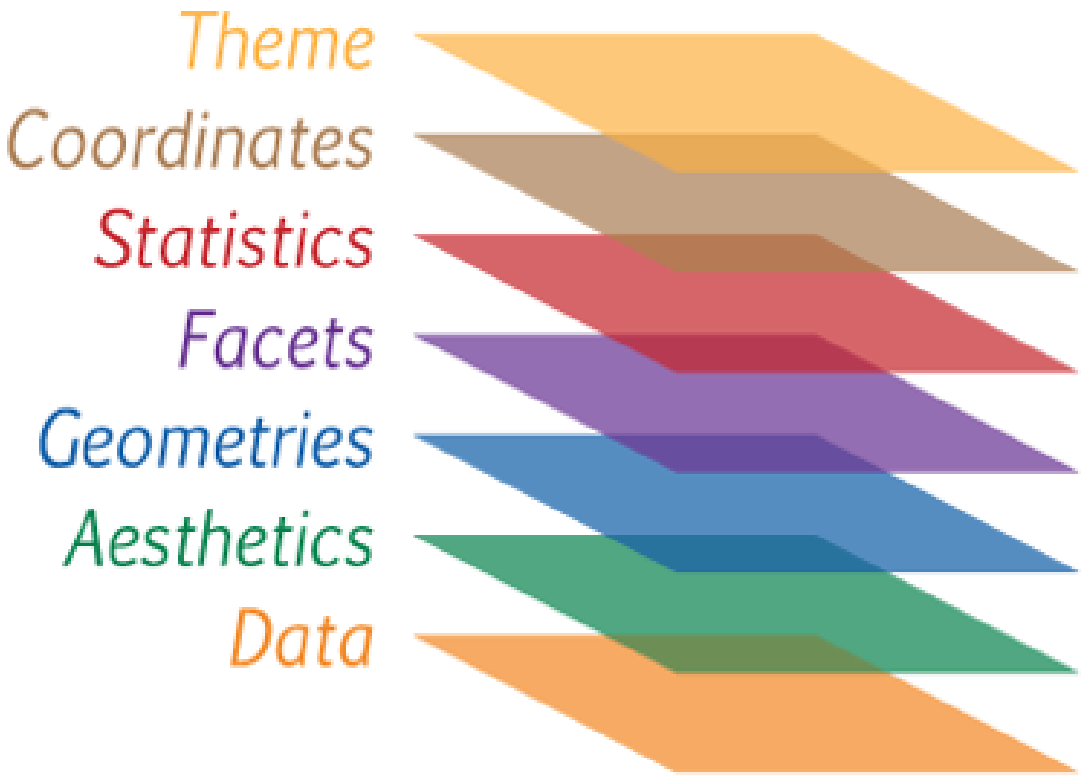
ggplot2

```
ggplot(datos_graf, aes(x = reorder(educ, -inc), y = inc)) +
  geom_col(width = 0.5, fill = "dark blue") +
  coord_flip() +
  labs(x = NULL, y = NULL) +
  scale_y_continuous(n.breaks = 9) +
  theme_minimal() +
  theme(panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_line(colour = "dark blue", size = 0.1)) +
  labs(title = "Discounted Expected Lifetime Earnings, VN(t')",
       subtitle = "(Income in thousands)")
```

Visualización de datos en R con ggplot2

Lógica de **ggplot2**

"gg" por Grammar of Graphics



Forma general de **ggplot2**

```
library(ggplot2)
ggplot(datos, aes(x = var1, y = var2)) +
  geom_XXX(...) +
  otros(...)
```

- geom_point()
- geom_line()
- geom_xxxx()
- facet
- theme

Demo - Visualizar Datos de indicadores de países

Demo

Script

- `Clase02_CodigoViz.R`

Datos

```
library(readr)
(datos_mundo <- read.csv("datos/datos_mundo2007.csv"))
```

```
##                       pais continente anio ExpVida       pob  gdpPercap
## 1              Afghanistan       Asia 2007  43.828  31889923   974.5803
## 2                  Albania     Europe 2007  76.423   3600523  5937.0295
## 3                  Algeria     Africa 2007  72.301  33333216  6223.3675
## 4                   Angola     Africa 2007  42.731  12420476  4797.2313
## 5                Argentina   Americas 2007  75.320  40301927 12779.3796
## 6                Australia    Oceania 2007  81.235  20434176 34435.3674
## 7                  Austria     Europe 2007  79.829   8199783 36126.4927
## 8                  Bahrain       Asia 2007  75.635    708573 29796.0483
## 9               Bangladesh       Asia 2007  64.062 150448339  1391.2538
## 10                 Belgium     Europe 2007  79.441  10392226 33692.6051
## 11                   Benin     Africa 2007  56.728   8078314  1441.2849
## 12                 Bolivia   Americas 2007  65.554   9119152  3822.1371
## 13  Bosnia and Herzegovina     Europe 2007  74.852   4552198  7446.2988
## 14                Botswana     Africa 2007  50.728   1639131 12569.8518
## 15                  Brazil   Americas 2007  72.390 190010647  9065.8008
## 16                Bulgaria     Europe 2007  73.005   7322858 10680.7928
## 17            Burkina Faso     Africa 2007  52.295  14326203  1217.0330
## 18                 Burundi     Africa 2007  49.580   8390505   430.0707
## 19                Cambodia       Asia 2007  59.723  14131858  1713.7787
## 20                Cameroon     Africa 2007  50.430  17696293  2042.0952
```

Gráficos con una variable

Histograma

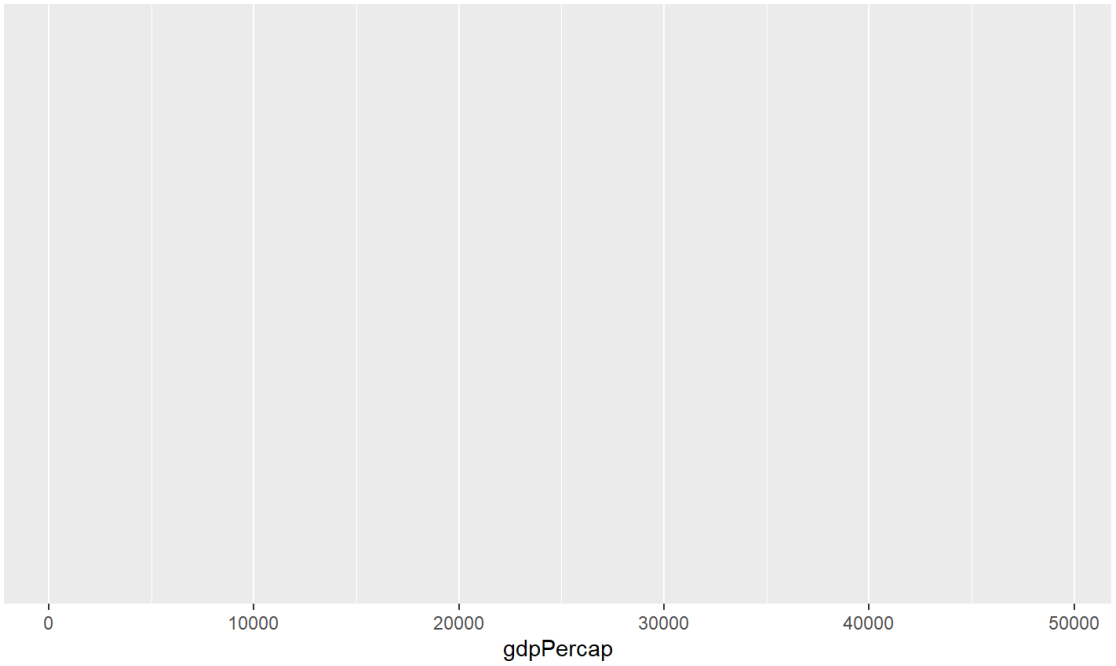
Gráfico base (datos)

```
ggplot(datos_mundo)
```

Histograma

Agregar capa (aes)

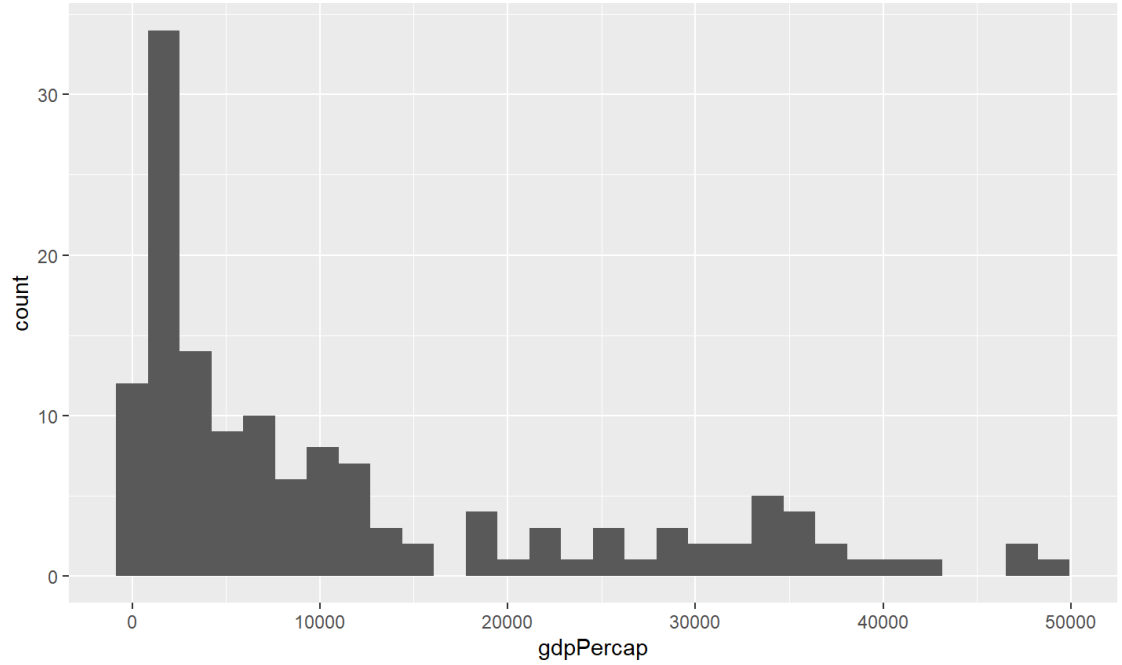
```
ggplot(datos_mundo, aes(x = gdpPercap))
```



Histograma

Agregar capa (geom)

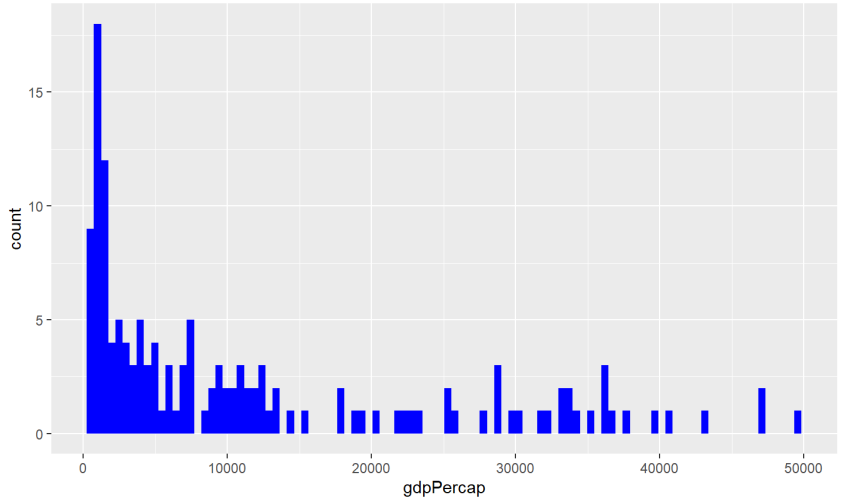
```
ggplot(datos_mundo, aes(x = gdpPercap)) +
  geom_histogram()
```



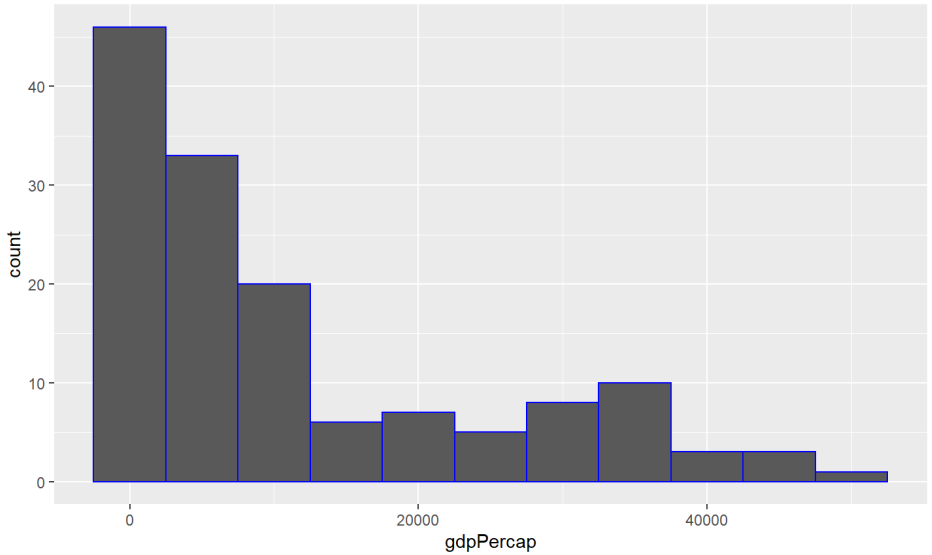
Histograma

cambiar algunos argumentos

```
ggplot(datos_mundo, aes(x = gdpPercap)) +
  geom_histogram(bins = 100, fill = "blue")
```



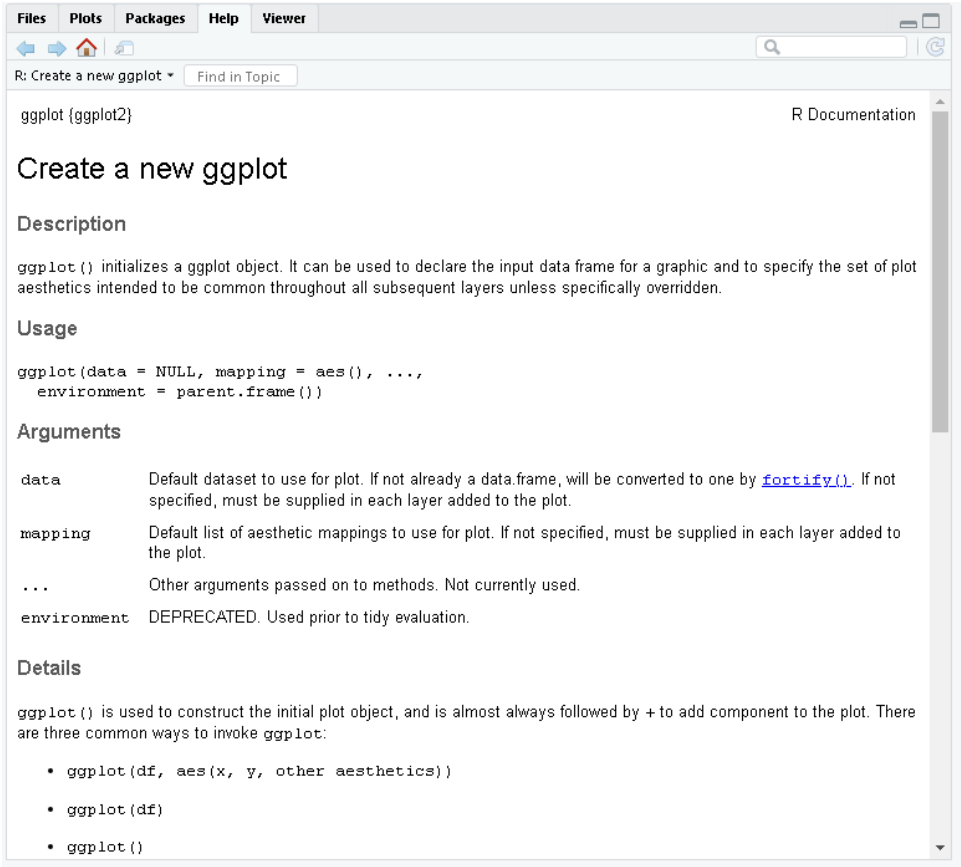

```
ggplot(datos_mundo, aes(x = gdpPercap)) +
  geom_histogram(binwidth = 5000, col = "blue")
```



Muchos argumentos

No olviden…

Siempre consulten `?nombrefunción`. Ej: `?ggplot`

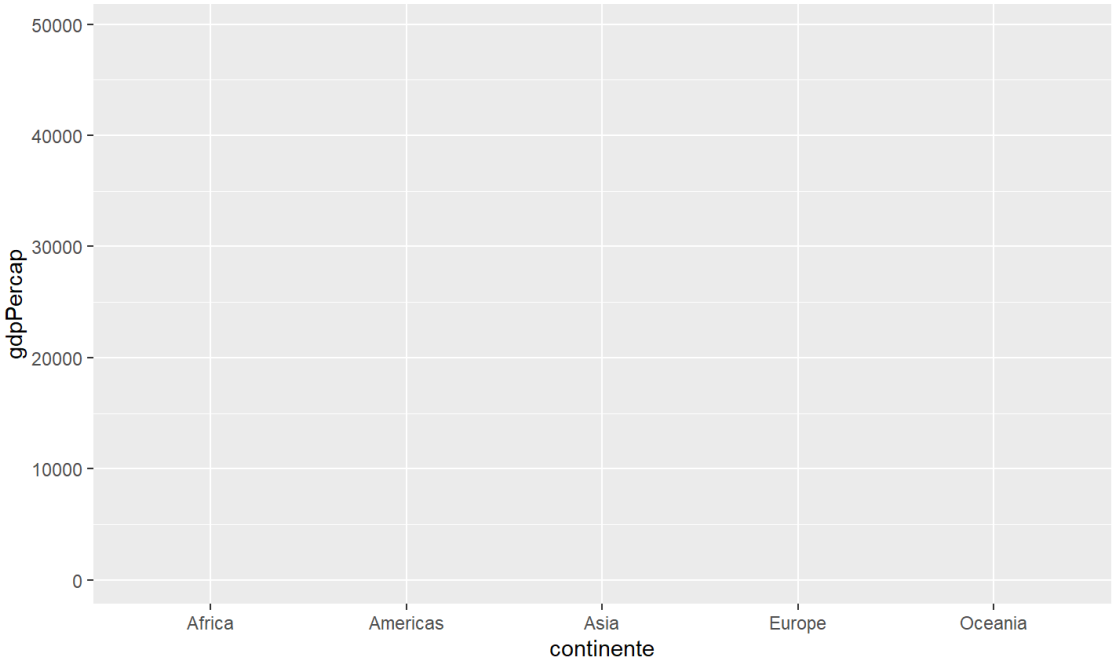


Gráficos con dos variables

Una variable categórica y una numérica

Gráfico base

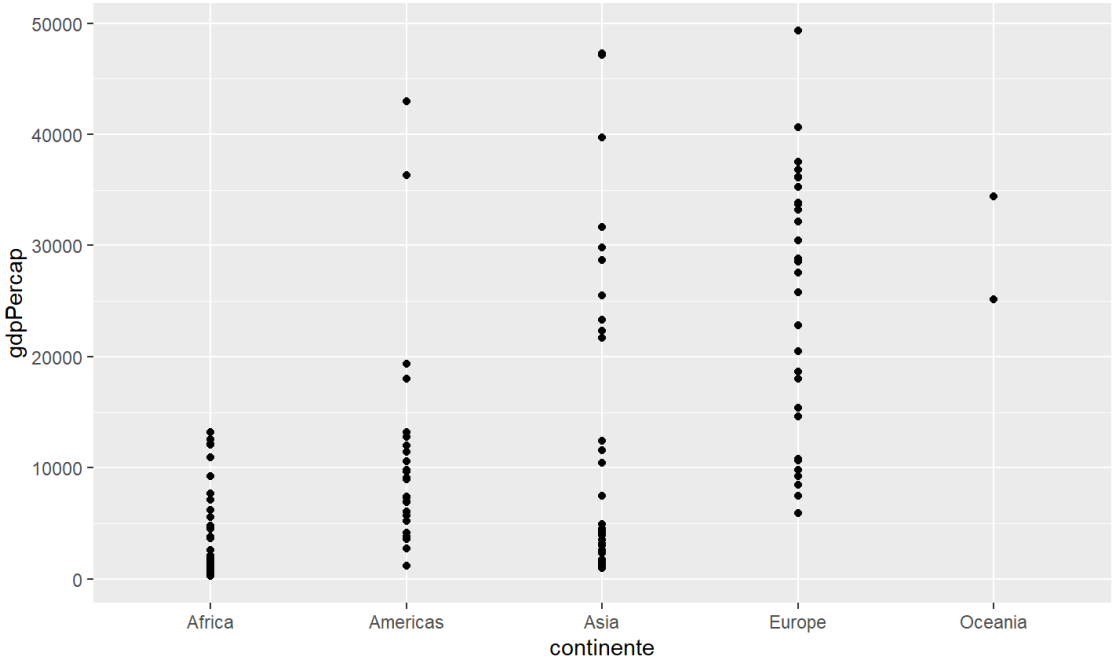
```
ggplot(datos_mundo, aes(x = continente, y = gdpPercap))
```



Una variable categórica y una numérica

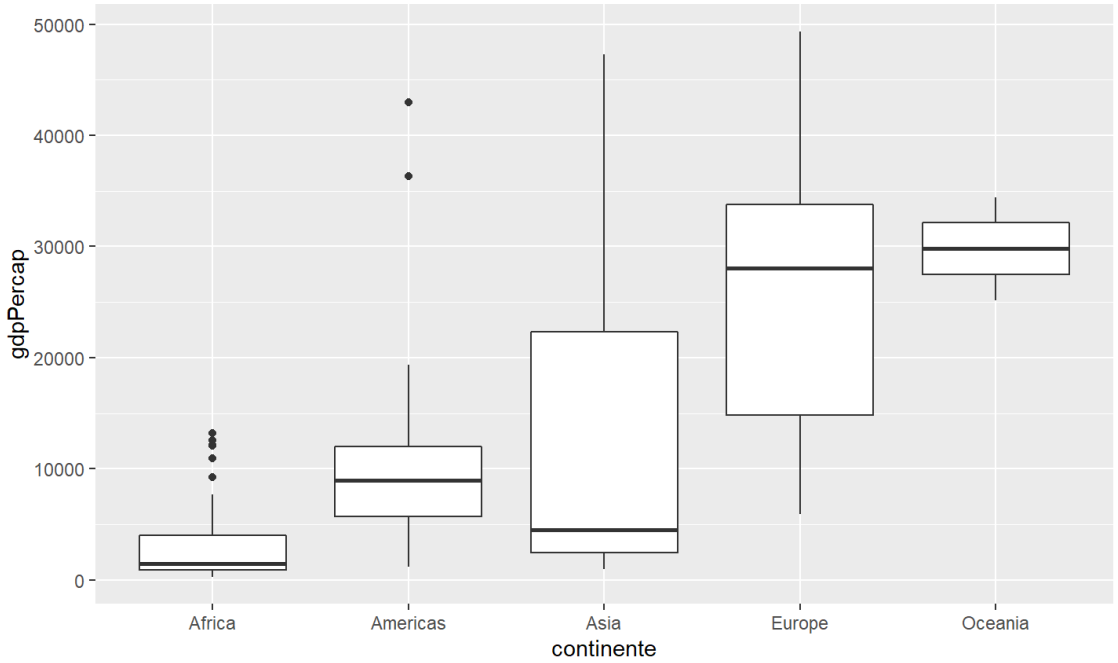
capa de puntos (`geom_point`)

```
ggplot(datos_mundo, aes(x = continente, y = gdpPercap)) +
  geom_point()
```

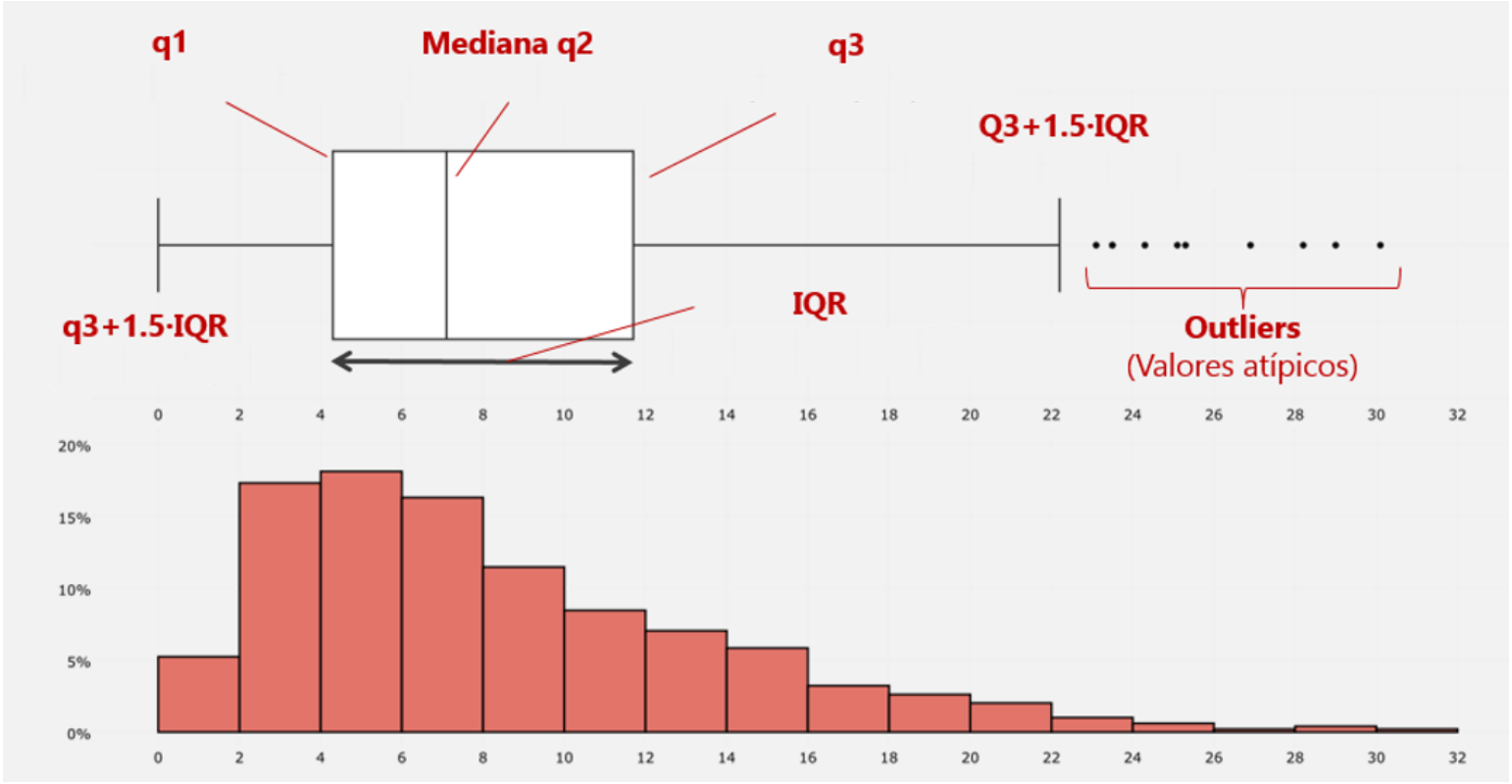


Boxplot

```
ggplot(datos_mundo, aes(x = continente, y = gdpPercap)) +
  geom_boxplot()
```

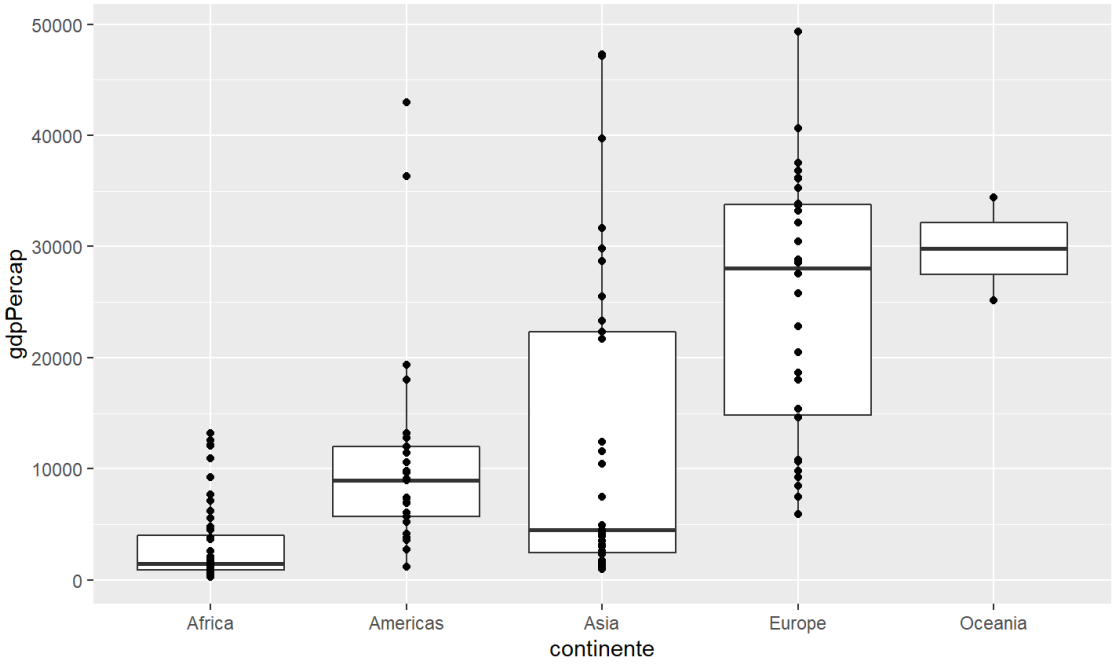


¿Qué nos muestra el boxplot?



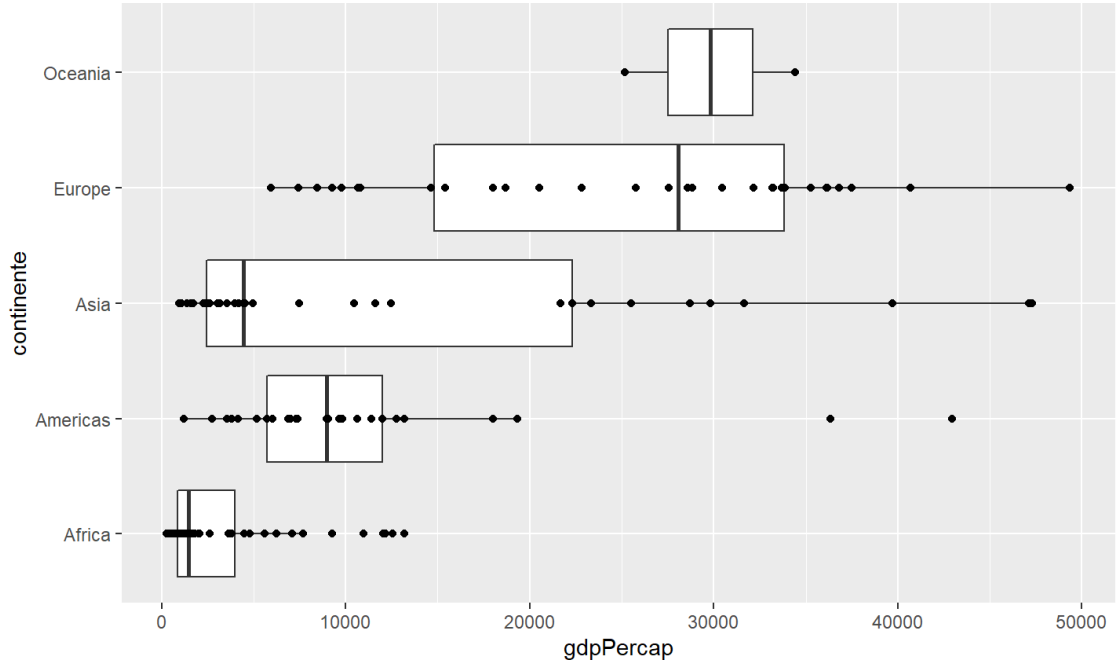
Juntar más de una capa (geom)

```
ggplot(datos_mundo, aes(x = continente, y = gdpPercap)) +
  geom_boxplot() +
  geom_point()
```



Invertir los ejes

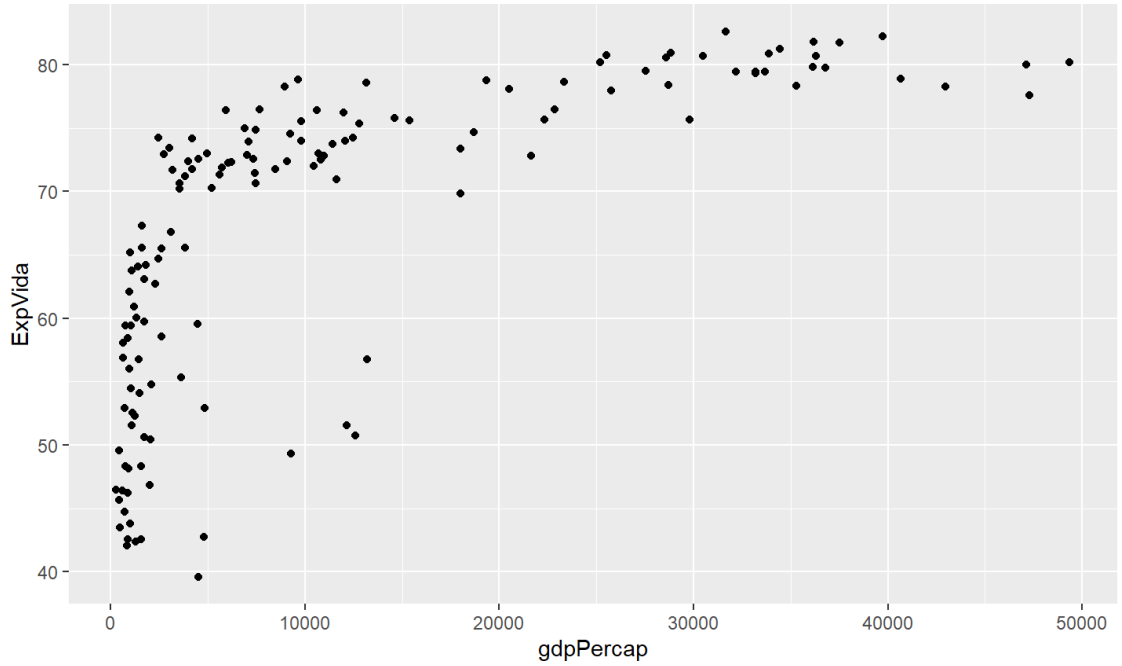
```
ggplot(datos_mundo, aes(x = continente, y = gdpPercap)) +
  geom_boxplot() +
  geom_point() +
  coord_flip()
```



Dos variables numéricas

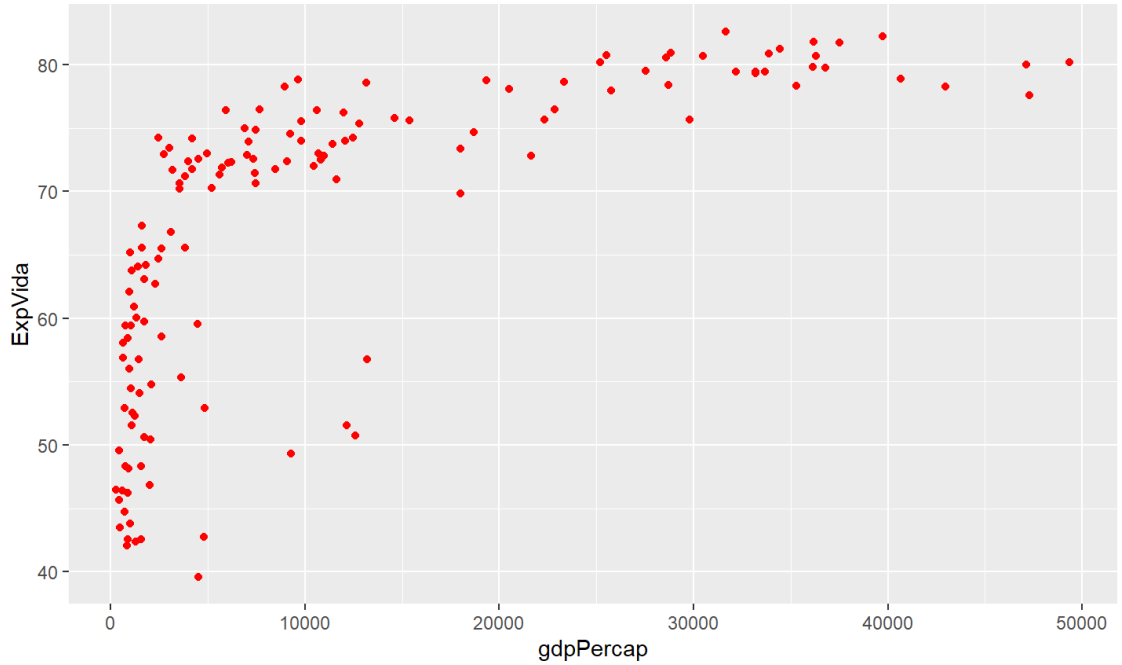
Gráfico de dispersión

```
ggplot(datos_mundo, aes(x = gdpPercap, y = ExpVida)) +
  geom_point()
```



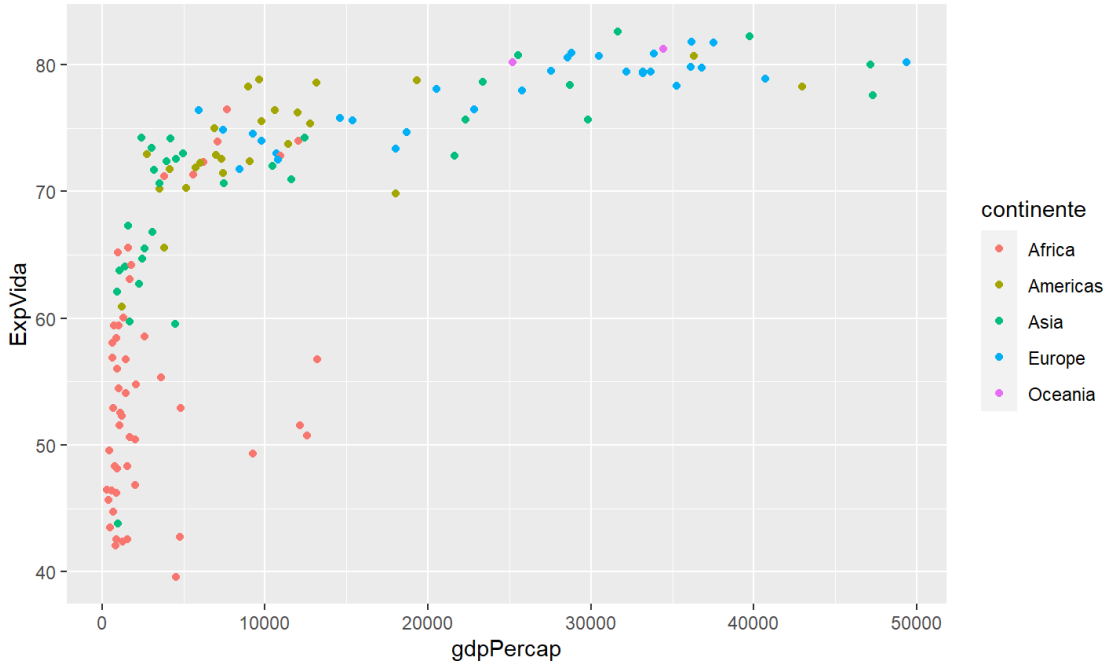
También se puede cambiar el color de los puntos

```
ggplot(datos_mundo, aes(x = gdpPercap, y = ExpVida)) +
  geom_point(col = "red")
```



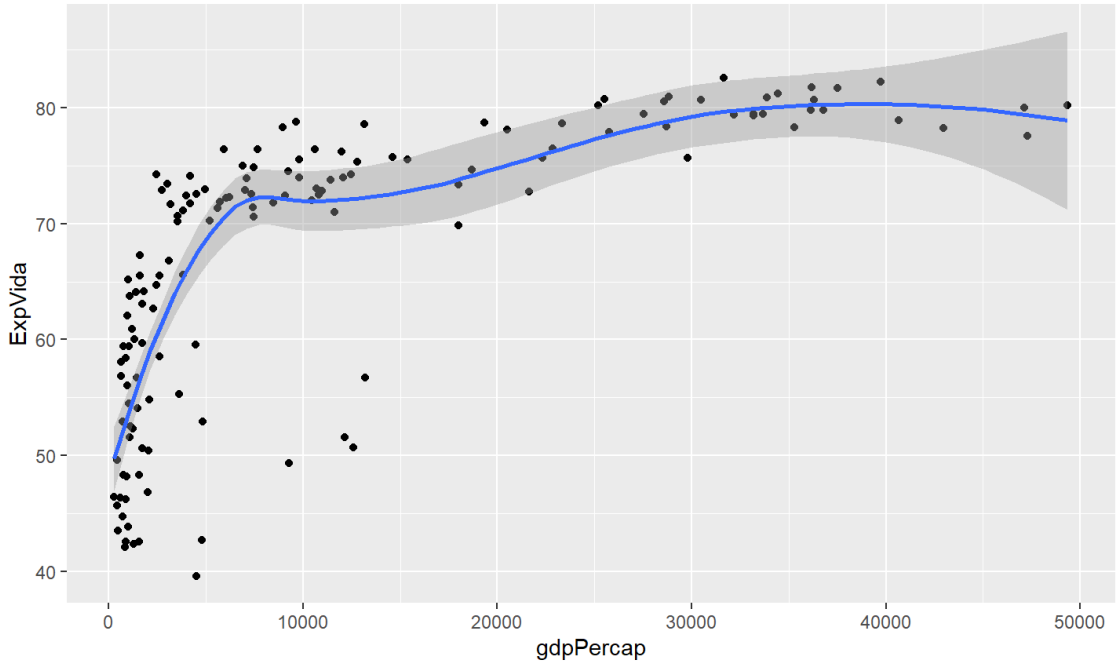
O usar el color para agregar más información

```
ggplot(datos_mundo, aes(x = gdpPercap, y = ExpVida)) +
  geom_point(aes(col = continente))
```



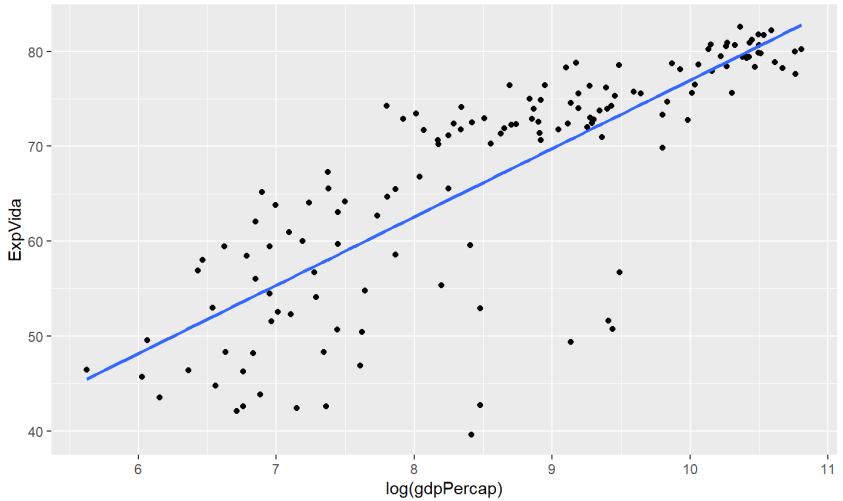
Agregar linea de tendencia (i)

```
ggplot(datos_mundo, aes(x = gdpPercap, y = ExpVida)) +
  geom_point() +
  geom_smooth()
```

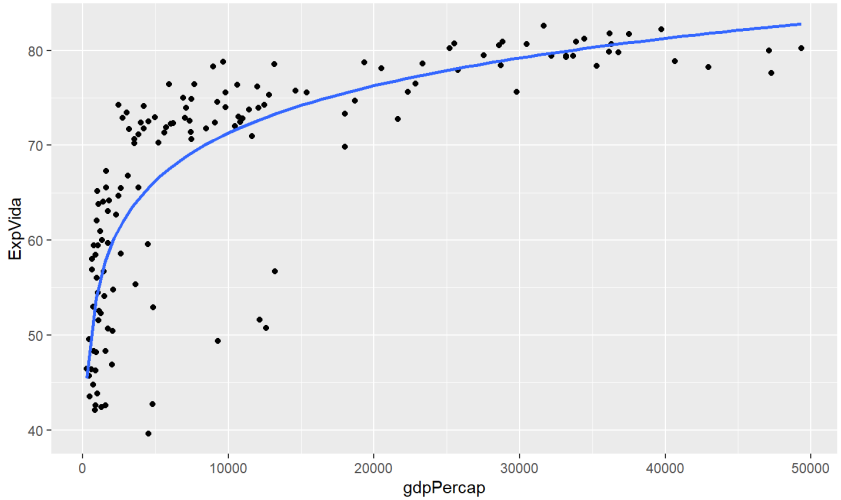


Agregar linea de tendencia (ii)

```
ggplot(datos_mundo,
       aes(x = log(gdpPercap), y = ExpVida)) +
  geom_point() +
  geom_smooth(se = FALSE, method = "lm")
```




```
ggplot(datos_mundo,
       aes(x = gdpPercap, y = ExpVida)) +
  geom_point() +
  geom_smooth(se = FALSE, method = "lm",
              formula = y ~ log(x))
```



Ejercicio

Ejercicio

Script

- `Clase02_Ejercicio.R`

Respuestas

```r
# Hacer un histograma de "ExpVida" utilizando 15 divisiones/barras (bins)
ggplot(datos_ejercicio, aes(x = ExpVida)) +
  geom_histogram(bins = 15)

# Hacer un gráfico de puntos de "anio" (x) vs "ExpVida" (y)
ggplot(datos_ejercicio, aes(x = anio, y = ExpVida)) +
  geom_point()

# Repetir el gráfico anterior y diferenciar los puntos con un color distinto según su continente
ggplot(datos_ejercicio, aes(x = anio, y = ExpVida, col = continente)) +
  geom_point()

# Agregar al gráfico anterior un geom de lineas (geom_line)
ggplot(datos_ejercicio, aes(x = anio, y = ExpVida, col = continente)) +
  geom_point() +
  geom_line()

# Repetir el gráfico anterior para las otras dos variables presentes en "datos_ejercicio"
ggplot(datos_ejercicio, aes(x = anio, y = pob, col = continente)) +
  geom_point() +
  geom_line()

ggplot(datos_ejercicio, aes(x = anio, y = gdpPercap, col = continente)) +
  geom_point() +
  geom_line()
```

Demo - Datos COVID

Demo

Script

- `Clase02_CodigoCovid.R`

Datos COVID-19

Cargar datos

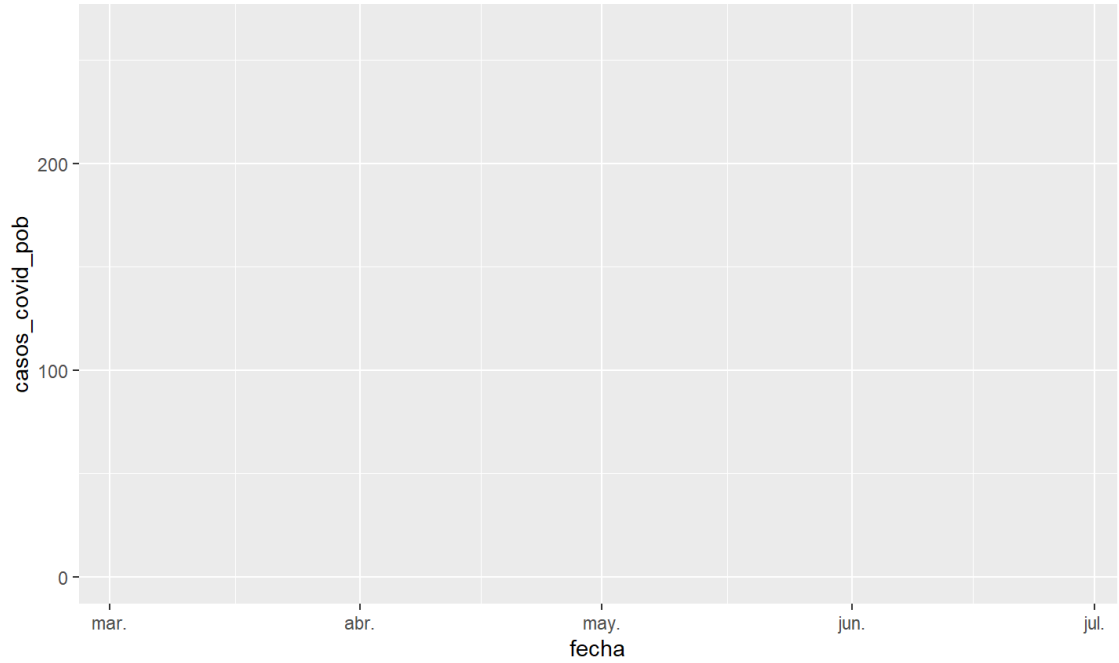
```
library(readr)
(datos_covid <- read_csv("datos/covid_datos_region.csv"))
```

```
## # A tibble: 1,888 x 11
##    region macroregion fecha      casos_covid codigo_region poblacion n_pcr_dia
##    <chr>  <chr>       <date>           <dbl>         <dbl>     <dbl>     <dbl>
##  1 Arica~ Norte Gran~ 2020-03-03           0            15    252110         0
##  2 Arica~ Norte Gran~ 2020-03-04           0            15    252110         0
##  3 Arica~ Norte Gran~ 2020-03-05           0            15    252110         0
##  4 Arica~ Norte Gran~ 2020-03-06           0            15    252110         0
##  5 Arica~ Norte Gran~ 2020-03-07           0            15    252110         0
##  6 Arica~ Norte Gran~ 2020-03-08           0            15    252110         0
##  7 Arica~ Norte Gran~ 2020-03-09           0            15    252110         0
##  8 Arica~ Norte Gran~ 2020-03-10           0            15    252110         0
##  9 Arica~ Norte Gran~ 2020-03-11           0            15    252110         0
## 10 Arica~ Norte Gran~ 2020-03-12           0            15    252110         0
## # ... with 1,878 more rows, and 4 more variables: muertes_covid <dbl>,
## #   pcr_acum <dbl>, pcr_acum_pob <dbl>, casos_covid_pob <dbl>
```

Datos creados a partir de API generada por *@pachamaltese* (https://github.com/pachamaltese) para DATA UC (https://coronavirus.mat.uc.cl/)

Gráfico base

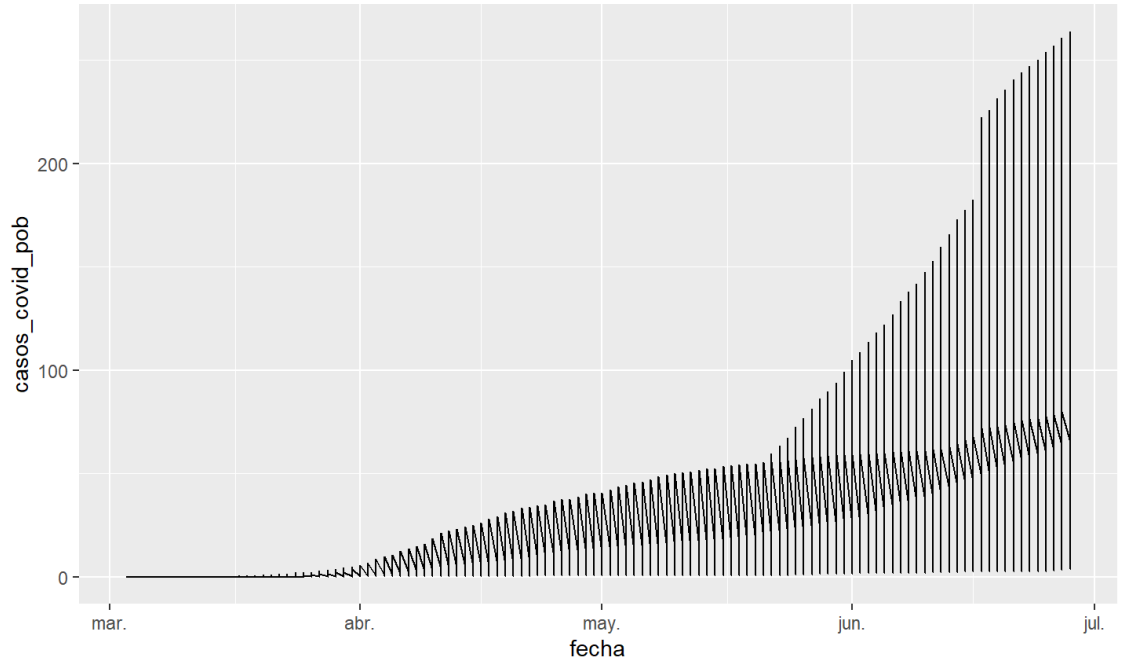
```
library(ggplot2)
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob))
```



Agregar **geom** de lineas

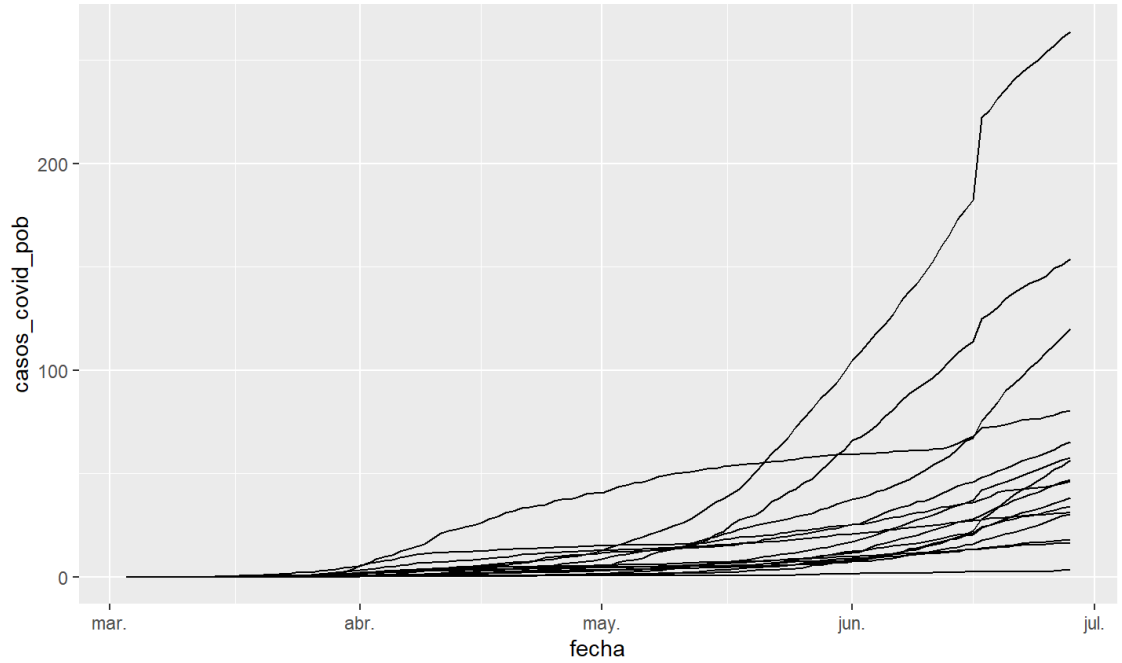
Pero algo se ve mal

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line()
```



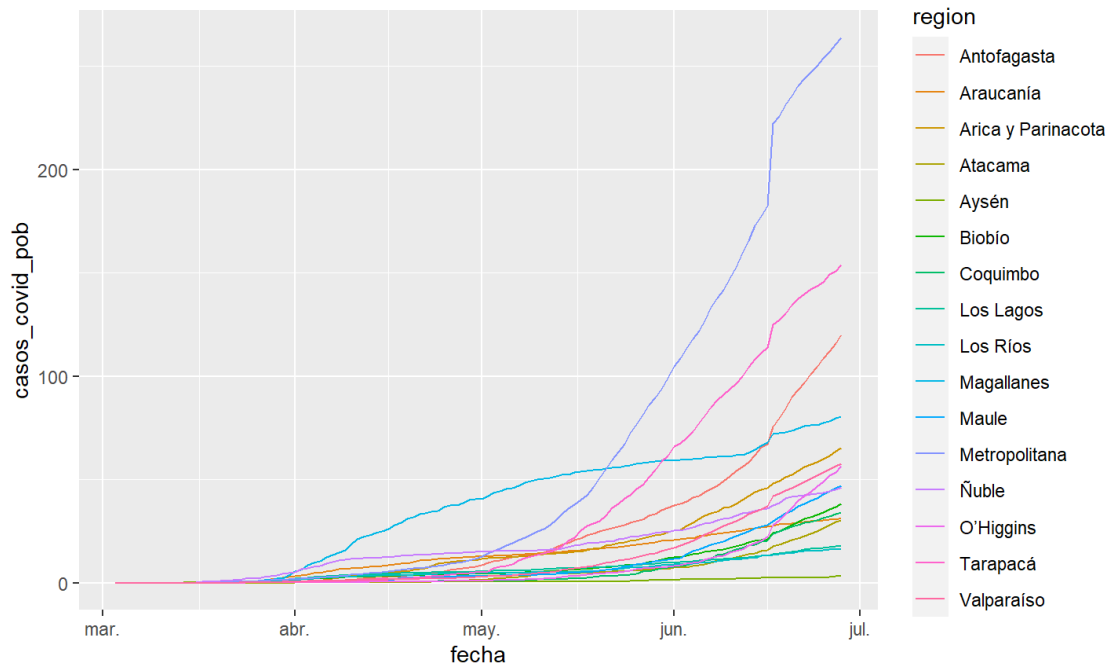
Cada linea representa una región (i)

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob, group = region)) +
  geom_line()
```



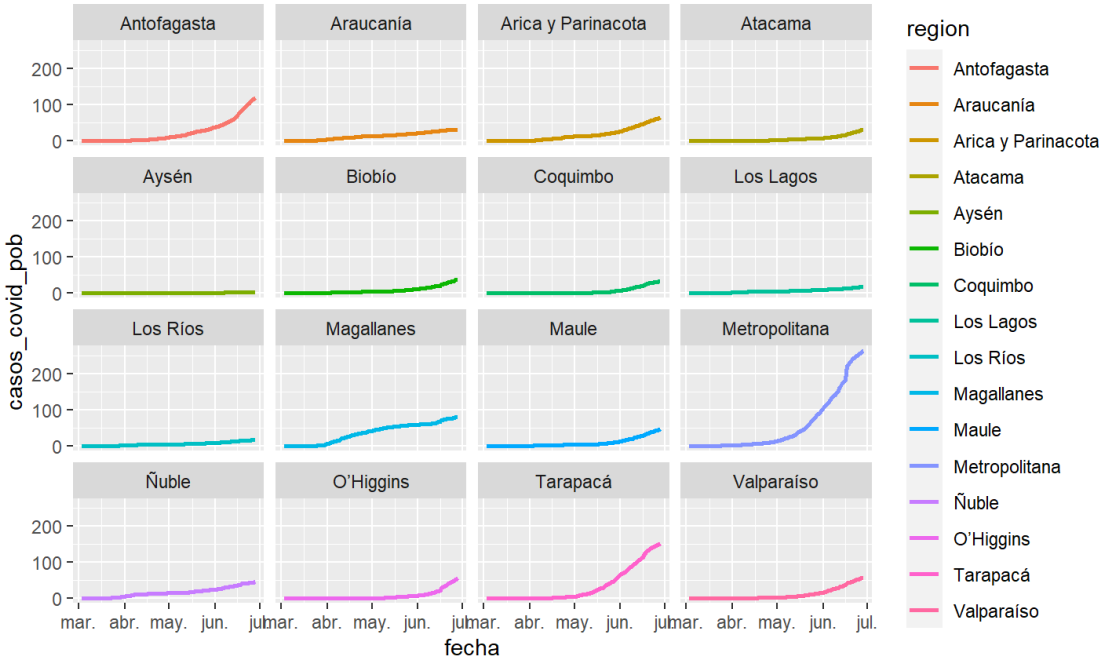
Cada linea representa una región (ii)

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob, col = region)) +
  geom_line()
```



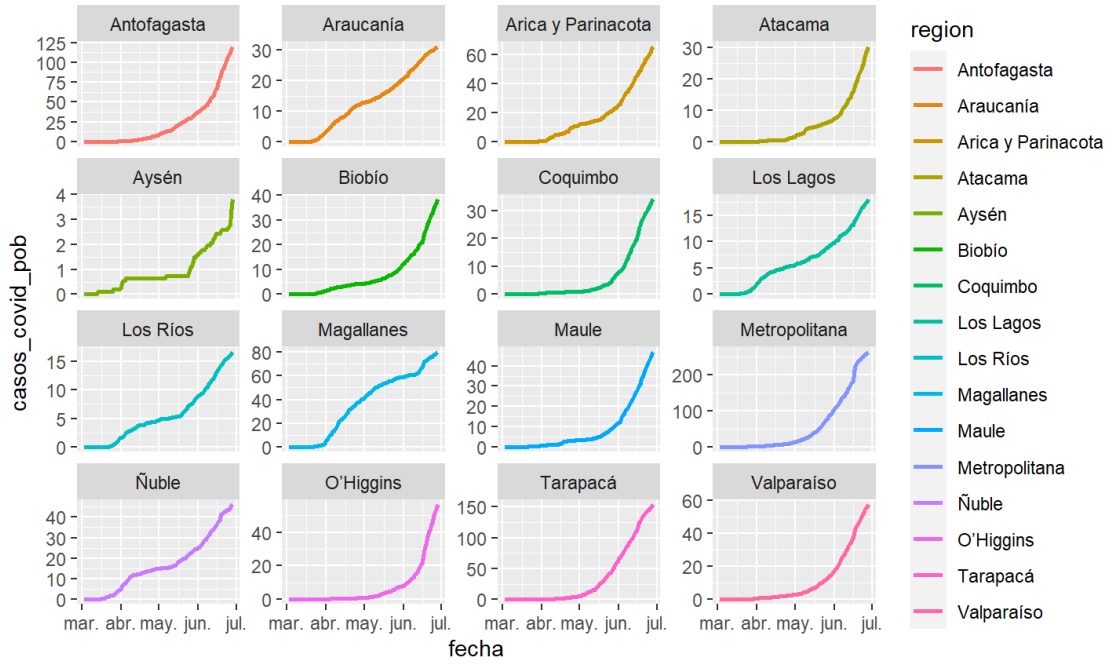
Separemos cada linea en su propio panel (i)

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line(aes(col = region), size = 1) +
  facet_wrap(vars(region ))
```



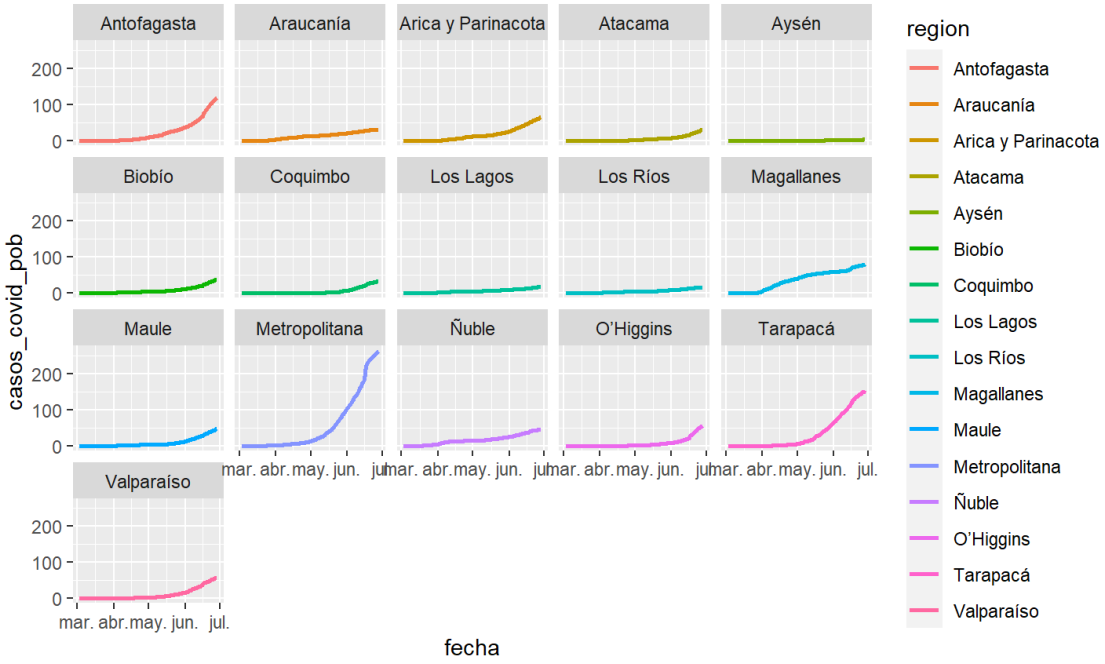
Separemos cada linea en su propio panel (ii)

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line(aes(col = region), size = 1) +
  facet_wrap(vars(region), scales = "free_y")
```



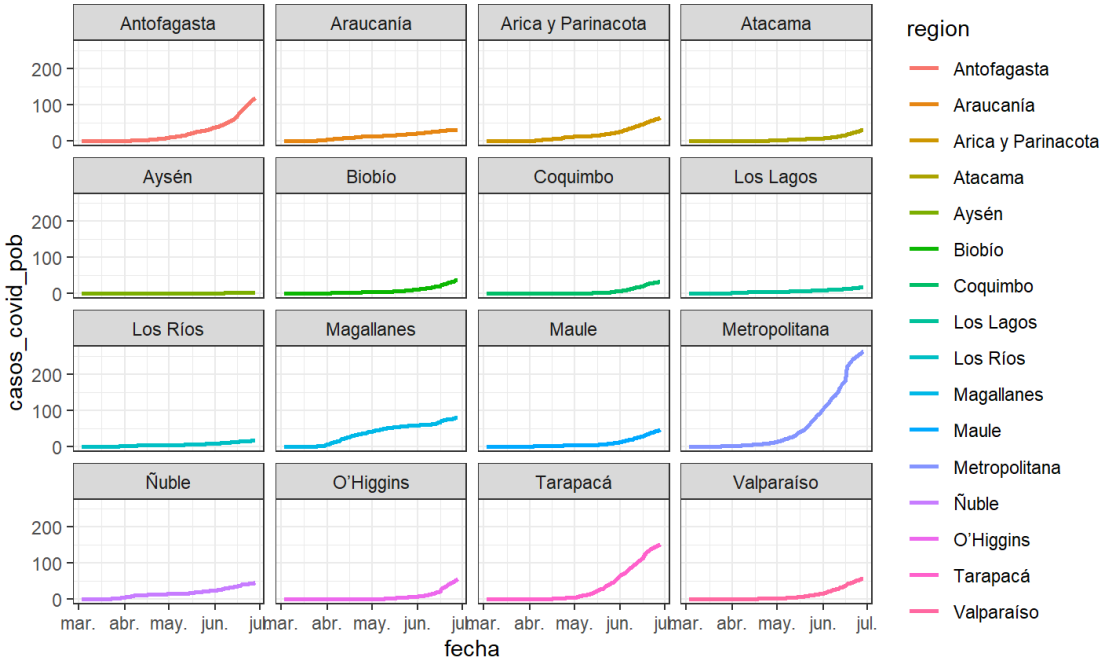
Separemos cada linea en su propio panel (iii)

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line(aes(col = region), size = 1) +
  facet_wrap(vars(region), ncol = 5)
```



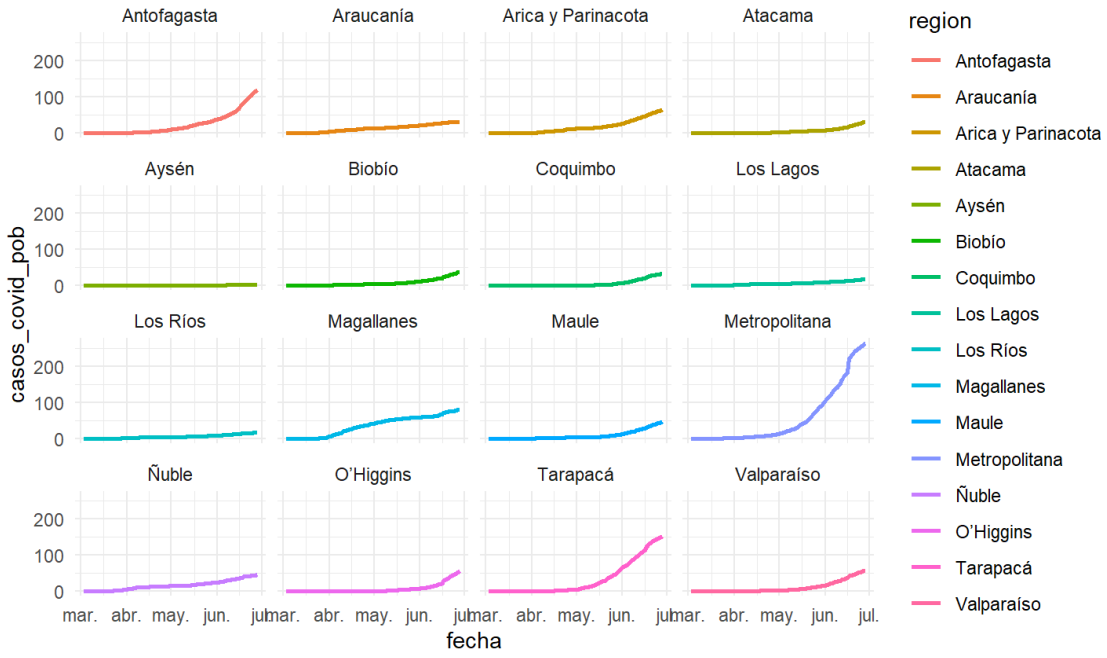
Fondo blanco pareciera quedar mejor (i)

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line(aes(col = region), size = 1) +
  facet_wrap(vars(region)) +
  theme_bw()
```



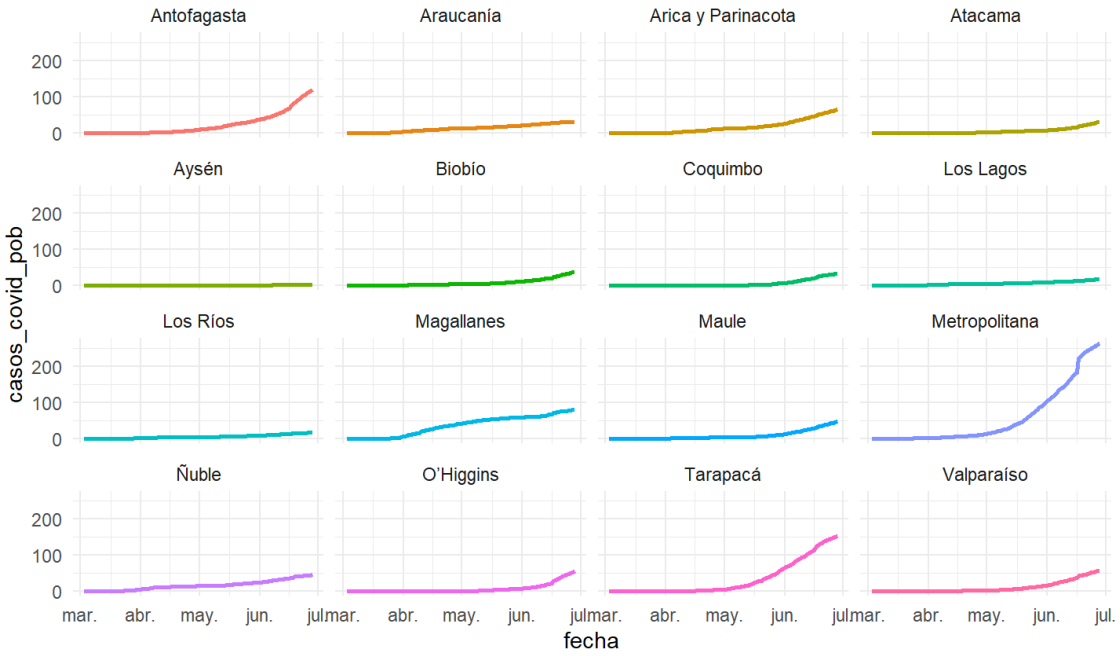
Fondo blanco pareciera quedar mejor (ii)

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line(aes(col = region), size = 1) +
  facet_wrap(vars(region)) +
  theme_minimal()
```



La leyenda pareciera no ser de mucha ayuda

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line(aes(col = region), size = 1) +
  facet_wrap(vars(region)) +
  theme_minimal() +
  theme(legend.position = "none")
```



“Themes” prefabricados

https://ggplot2.tidyverse.org/reference/ggtheme.html

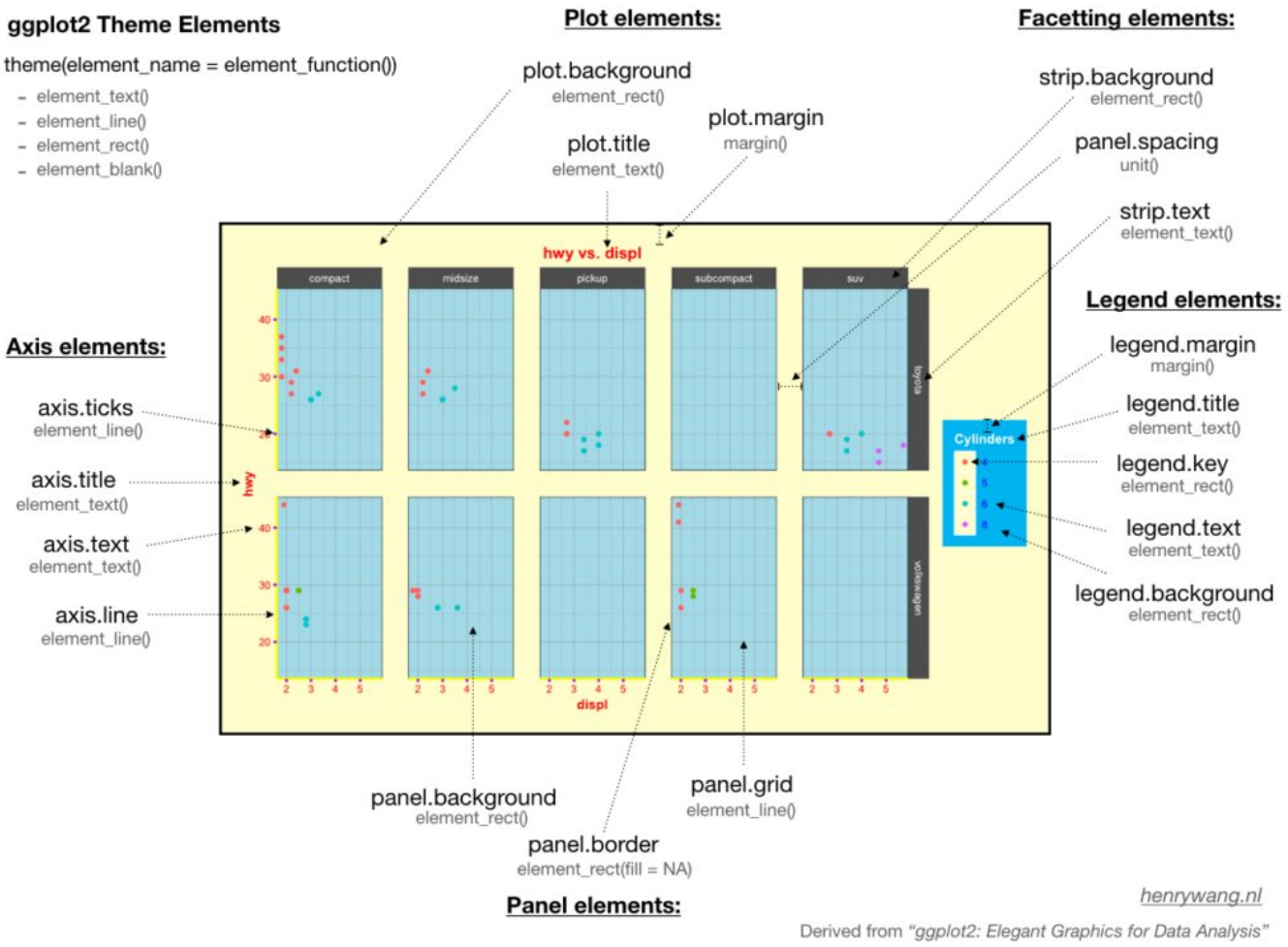
Complete themes

Source: `R/theme-defaults.r` (https://github.com/tidyverse/ggplot2/blob/master/R/theme-defaults.r)

These are complete themes which control all non-data display. Use `theme()` if you just need to tweak the display of an existing theme.

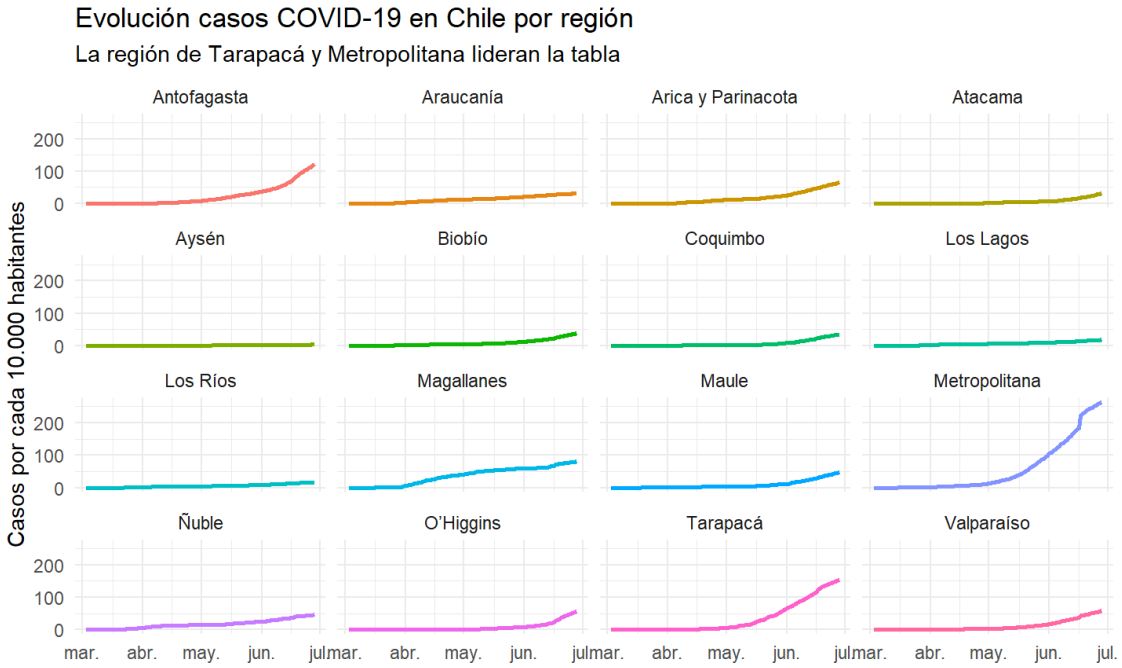
```
theme_grey (https://ggplot2.tidyverse.org/reference/ggtheme.html)(
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

Cómo modificar detalles de nuestos gráficos



Títulos/Ejes como detalles finales

```
ggplot(datos_covid, aes(x = fecha, y = casos_covid_pob)) +
  geom_line(aes(col = region), size = 1) +
  facet_wrap(vars(region)) + theme_minimal() +
  theme(legend.position = "none") +
  labs(title = "Evolución casos COVID-19 en Chile por región",
       subtitle = "La región de Tarapacá y Metropolitana lideran la tabla",
       x = NULL, y = "Casos por cada 10.000 habitantes")
```



Muchas más posibilidades

https://www.data-to-viz.com/

from Data to Viz

Tarea 1

- Criticar gráfico y proponer uno mejor
- Buscar un gráfico, criticarlo, y proponer uno mejor

Idea de trabajo

- Jueves 20 de agosto
- Instrucciones en CANVAS

Siguiente clase

- Manejo de datos
- `install.packages("dplyr")`
- `install.packages("tidyr")`