



Ciencia de Datos para Políticas Públicas

Módulo 2 - Clase 1: Visualización de datos

Pablo Aguirre Hörmann

15/06/2021

Información general

- 6 clases: dos partes (3 c/u)
- Office hours (*Zoom*): horario por definir
- Ayudantías: 1 y 22 de julio
- Web curso para preguntas fuera de horario de clases
- Dedicación de tiempo
- PPTs se subirán al terminar cada clase

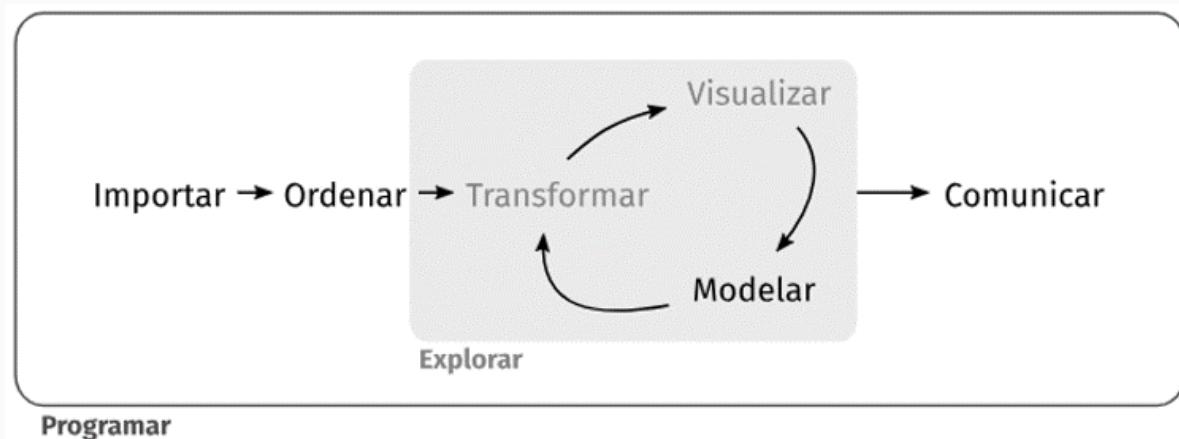
Qué, Por qué y Cómo de este curso

¿Qué?

- R
- Visualización de datos
- Manejo de datos
- Reportar resultados
- Inferencia Estadística/Econometría

Qué, Por qué y Cómo de este curso

¿Qué?



Qué, Por qué y Cómo de este curso

¿Por qué?

- Análisis/Investigación reproducible
- Control/Flexibilidad

Reproducibilidad

Objetivo: realizar un reporte basado en datos de **Datos Públicos**

Reproducibilidad

Objetivo: realizar un reporte basado en datos de Datos Públicos

SIN PROGRAMACIÓN

1. Ingresar a la web y descargar datos
2. Limpiar, ordenar, y analizar datos en *MS Excel*
3. Escribir documento en *MS Word*
4. Guardar en "algun lado" (ojalá no en "Mis Documentos" o "Descargas")

Reproducibilidad

Objetivo: realizar un reporte basado en datos de Datos Públicos

SIN PROGRAMACIÓN

1. Ingresar a la web y descargar datos
2. Limpiar, ordenar, y analizar datos en *MS Excel*
3. Escribir documento en *MS Word*
4. Guardar en "algun lado" (ojalá no en "Mis Documentos" o "Descargas")

CON PROGRAMACIÓN

1. Crear una carpeta para el proyecto/tarea
 - o datos
 - o gráficos
 - o resultados
2. Descargar datos desde R
3. Limpiar, ordenar, y analizar datos en R
4. Escribir documento en R Markdown

Reproducibilidad

Seis meses después quieres repetir la tarea (o un/a colega)

Reproducibilidad

Seis meses después quieres repetir la tarea (o un/a colega)

SIN PROGRAMACIÓN

1. Recordar que se hizo
2. Ingresar a la web y descargar datos
3. Limpiar, ordenar, y analizar datos en *MS Excel*
4. Escribir documento en *MS Word*
5. Guardar en "algún lado" (ojalá no en "Mis Documentos" o "Descargas")

Reproducibilidad

Seis meses después quieres repetir la tarea (o un/a colega)

SIN PROGRAMACIÓN

1. Recordar que se hizo
2. Ingresar a la web y descargar datos
3. Limpiar, ordenar, y analizar datos en *MS Excel*
4. Escribir documento en *MS Word*
5. Guardar en "algún lado" (ojalá no en "Mis Documentos" o "Descargas")

CON PROGRAMACIÓN

1. Ejecutar el código

¿Cómo?

- **RStudio Cloud**
 - Pero vayan practicando en la versión de escritorio
- **Aprender haciendo**
 - Demostración - Ejercicio
 - Todo código que se muestre estará disponible

Dicho eso...

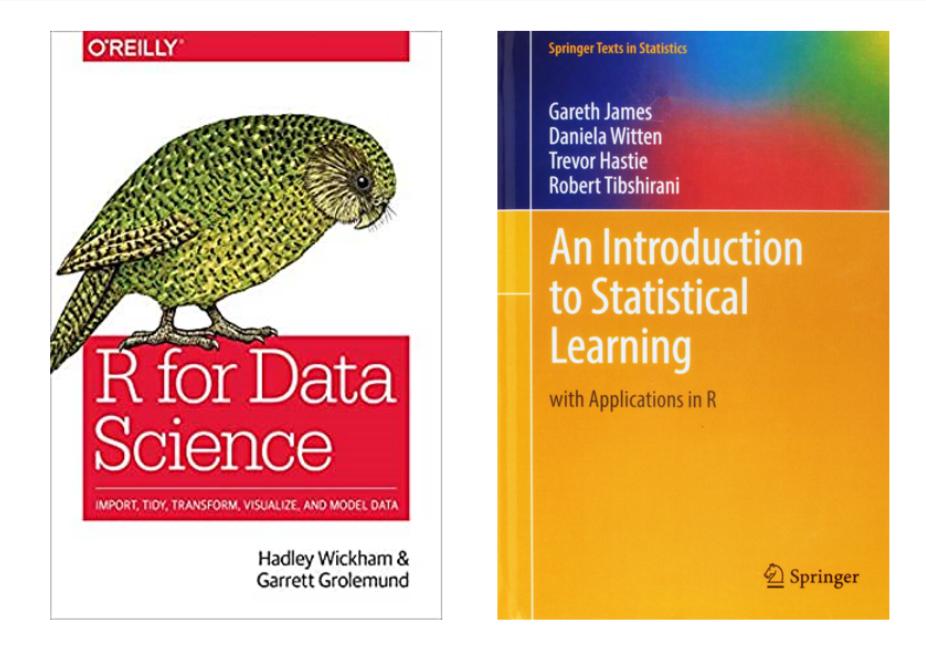
- **Este curso no es suficiente**
 - Requiere de dedicación y práctica para poder sacar el mayor provecho
 - Prueba y error
- **Contamos con horas finitas de clases**
 - Hay detalles que no podremos cubrir y que es importante que puedan revisar/practicar por su parte
 - Trataré de ir mencionando estos a medida que avancemos

Fuentes valiosas de información



#rstats

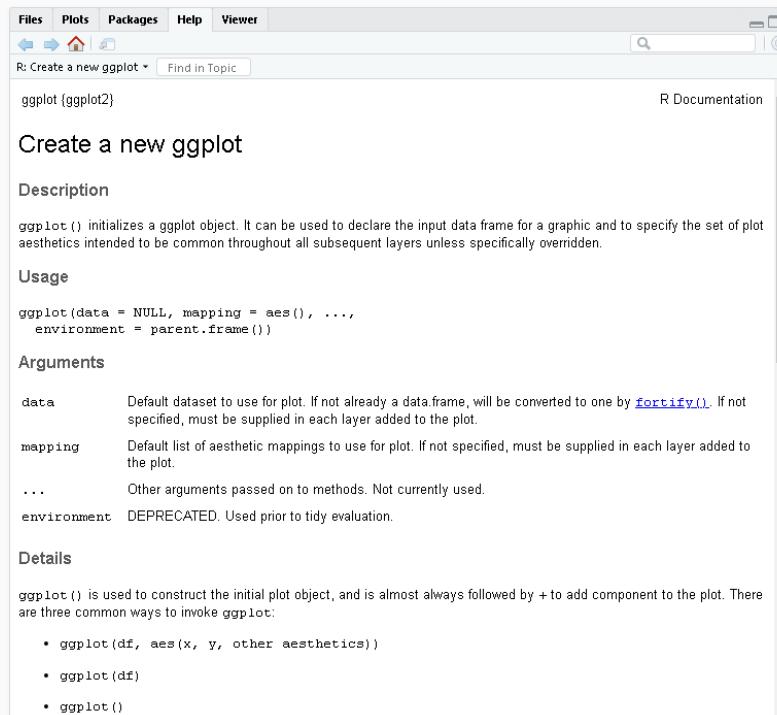
Fuentes valiosas de información



- <https://r4ds.had.co.nz/> | <https://es.r4ds.hadley.nz/>
- <http://faculty.marshall.usc.edu/gareth-james/ISL/>

Además de google...

No se aburran de usar `?nombrefunción`. Ej: `?ggplot`



¿Qué veremos hoy?

- **Visualización de datos**

- Manejo de datos
- Transformación de datos
- Reportar resultados
- Inferencia Estadística/Econometría

Pero antes...

Recordemos

- Instalar paquetes: `install.packages("nombrepaqute")`
- Cargar paquetes: `library(nombrepaqute)`
- Principales paquetes que usaremos:
 - `readr`: cargar datos
 - `tidyverse`: lograr datos *tidy* (ordenados)
 - `dplyr`: "verbos" (funciones) de manipulación de datos
 - `ggplot2`: visualización

Pero antes...

Recordemos

- Funciones, `funcion(argumento1, argumento2, ...)`

- `log(x = 100, base = 10)`
 - `log(100, 10)`
 - `filter(censo, region = "RM")`

- `%>%`: "Encadenar" código

- `censo %>% filter(region = "RM")`

- `+`: "Sumar" capas en `ggplot2`

Imprime pero no guarda

```
read_csv("../datos/datos_mundo.csv")  
  
## # A tibble: 1,704 x 6  
##   pais      continente  anio ExpVida      pob gdpPercap  
##   <chr>      <chr>    <dbl>   <dbl>    <dbl>     <dbl>  
## 1 Afghanistan Asia      1952     28.8  8425333    779.  
## 2 Afghanistan Asia      1957     30.3  9240934    821.  
## 3 Afghanistan Asia      1962     32.0  10267083   853.  
## 4 Afghanistan Asia      1967     34.0  11537966   836.  
## 5 Afghanistan Asia      1972     36.1  13079460   740.  
## 6 Afghanistan Asia      1977     38.4  14880372   786.  
## 7 Afghanistan Asia      1982     39.9  12881816   978.  
## 8 Afghanistan Asia      1987     40.8  13867957   852.  
## 9 Afghanistan Asia      1992     41.7  16317921   649.  
## 10 Afghanistan Asia     1997     41.8  22227415   635.  
## # ... with 1,694 more rows
```

Guarda pero no imprime

```
datos_mundo ← read_csv(".. /datos/datos_mundo.csv")
```

Guarda e imprime

```
datos_mundo ← read_csv("../datos/datos_mundo.csv")
datos_mundo

## # A tibble: 1,704 x 6
##   pais      continente  anio ExpVida      pob gdpPercap
##   <chr>     <chr>    <dbl>   <dbl>    <dbl>    <dbl>
## 1 Afghanistan Asia     1952    28.8  8425333    779.
## 2 Afghanistan Asia     1957    30.3  9240934    821.
## 3 Afghanistan Asia     1962    32.0  10267083   853.
## 4 Afghanistan Asia     1967    34.0  11537966   836.
## 5 Afghanistan Asia     1972    36.1  13079460   740.
## 6 Afghanistan Asia     1977    38.4  14880372   786.
## 7 Afghanistan Asia     1982    39.9  12881816   978.
## 8 Afghanistan Asia     1987    40.8  13867957   852.
## 9 Afghanistan Asia     1992    41.7  16317921   649.
## 10 Afghanistan Asia    1997    41.8  22227415   635.
## # ... with 1,694 more rows
```

Guarda e imprime

```
(datos_mundo ← read_csv(".. /datos/datos_mundo.csv"))
```

```
## # A tibble: 1,704 x 6
##   pais      continente  anio ExpVida      pob gdpPercap
##   <chr>     <chr>    <dbl>  <dbl>    <dbl>    <dbl>
## 1 Afghanistan Asia     1952    28.8  8425333    779.
## 2 Afghanistan Asia     1957    30.3  9240934    821.
## 3 Afghanistan Asia     1962    32.0  10267083   853.
## 4 Afghanistan Asia     1967    34.0  11537966   836.
## 5 Afghanistan Asia     1972    36.1  13079460   740.
## 6 Afghanistan Asia     1977    38.4  14880372   786.
## 7 Afghanistan Asia     1982    39.9  12881816   978.
## 8 Afghanistan Asia     1987    40.8  13867957   852.
## 9 Afghanistan Asia     1992    41.7  16317921   649.
## 10 Afghanistan Asia    1997    41.8  22227415   635.
## # ... with 1,694 more rows
```

Mala práctica

```
library(readr)
library(dplyr)
library(ggplot2)
datos_mundo <- read_csv("datos/datos_mundo.csv")
ggplot(filter(datos_mundo, anio = 2007), aes(x = ExpVida, y = pob)) + geom_point()
```

Buena práctica

```
# Cargar paquetes
library(readr)
library(dplyr)
library(ggplot2)

# Cargar datos
datos_mundo <- read_csv("/datos/datos_mundo.csv")

# Crear subset con datos del año 2007
datos_mundo_select <- datos_mundo %>%
  filter(anio = 2007)

# Gráfico de ExpVida vs pob
ggplot(datos_mundo_select, aes(x = ExpVida, y = pob)) +
  geom_point()
```

¿Qué veremos hoy?

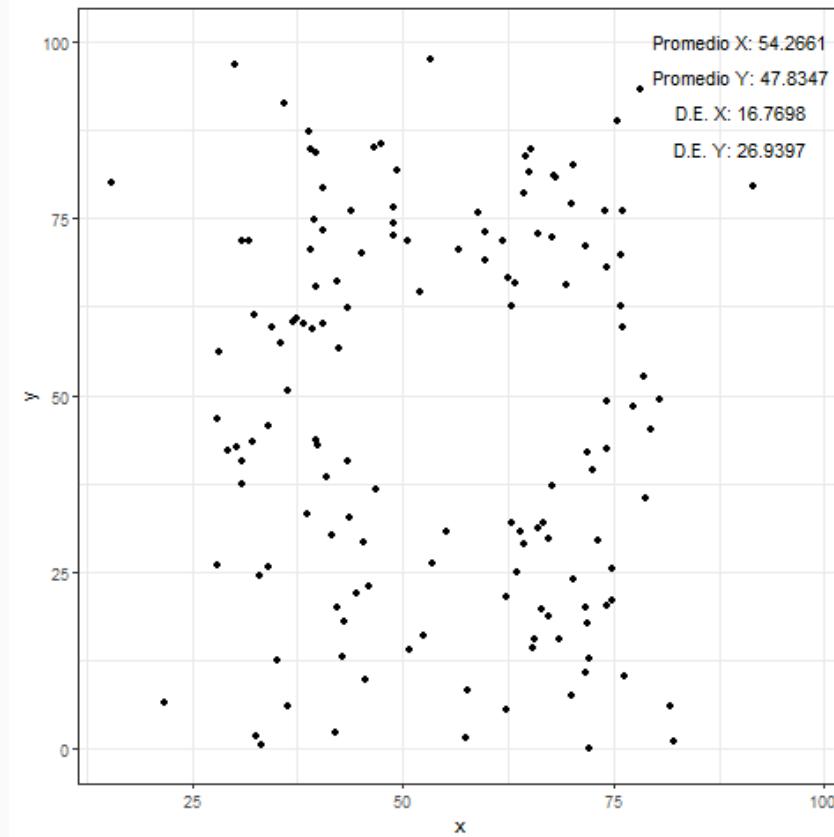
- **Visualización de datos**

- Manejo de datos
- Transformación de datos
- Reportar resultados
- Inferencia Estadística/Econometría

¿Por qué visualizar?

Dos razones principales:

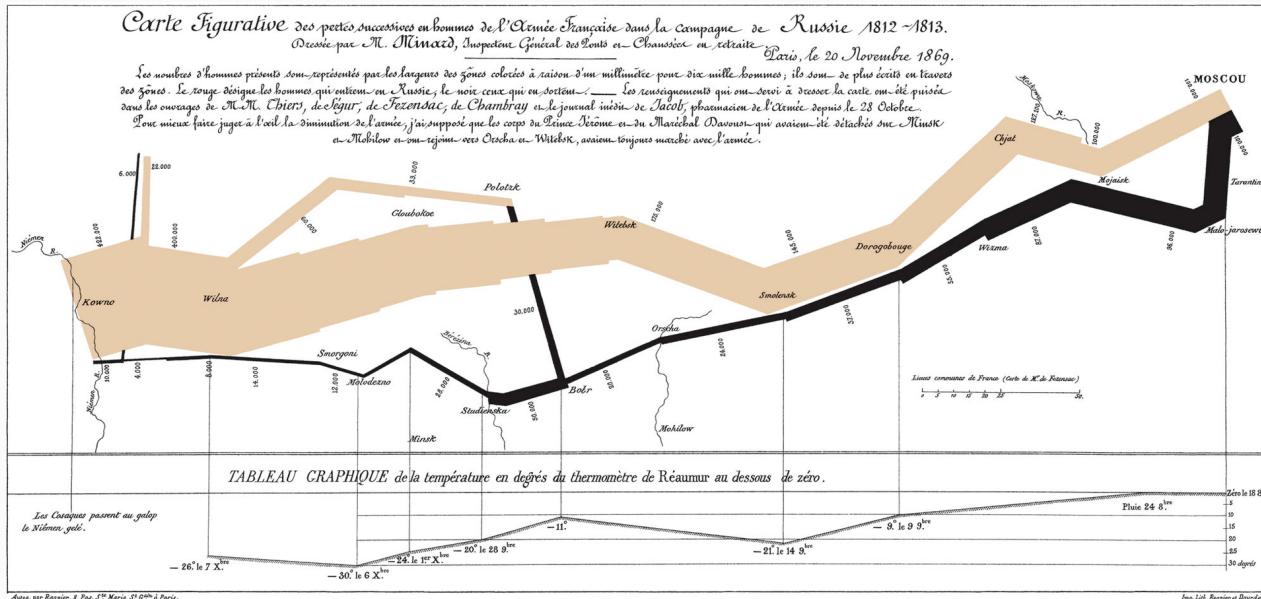
1. **Para uno:** entender los datos con el fin de guiar análisis posteriores (análisis exploratorio)



¿Por qué visualizar?

Dos razones principales:

- 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



(Charles Joseph Minard, 1869)

¿Por qué visualizar?

Dos razones principales:

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

Lo que deberíamos buscar

- Mostrar los datos y no mentir con estos
 - Contar una historia (¿una relación? ¿causalidad? ¿un patrón? ¿un quiebre?)
 - Transmitir y convencer
- Minimizar aspectos innecesarios
- Visualizaciones deben complementar el texto y tener suficiente información para "sobrevivir por sí mismas"

Una idea general

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

¿Cuántos 3 hay?

1269548523612356987458245
0124036985702069568312781
2439862012478136982173256

Una idea general

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

¿Y ahora?

126954852**3**612**3**56987458245
01240**3**6985702069568**3**12781
24**3**98620124781**3**698217**3**256

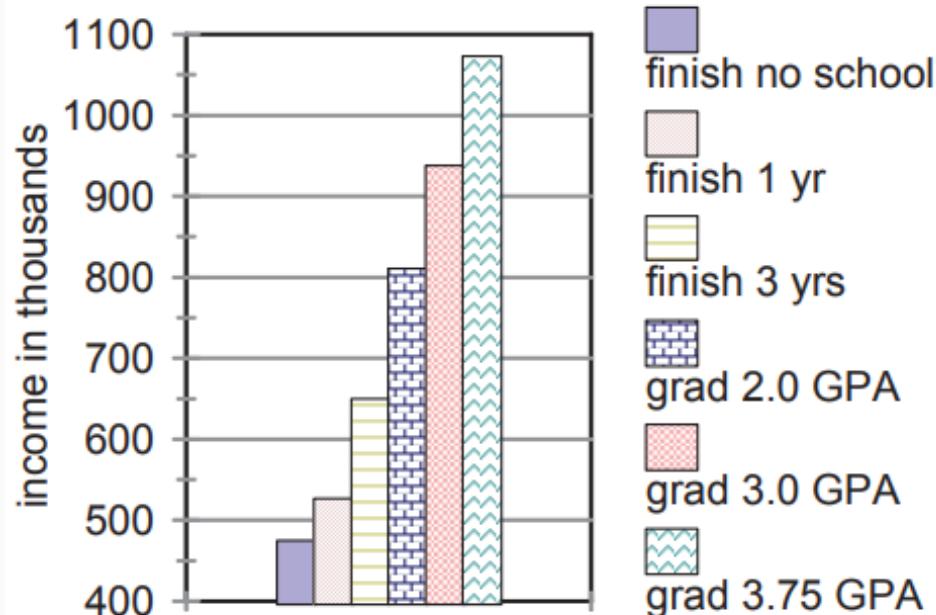
Principios de (Edward) Tufte

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

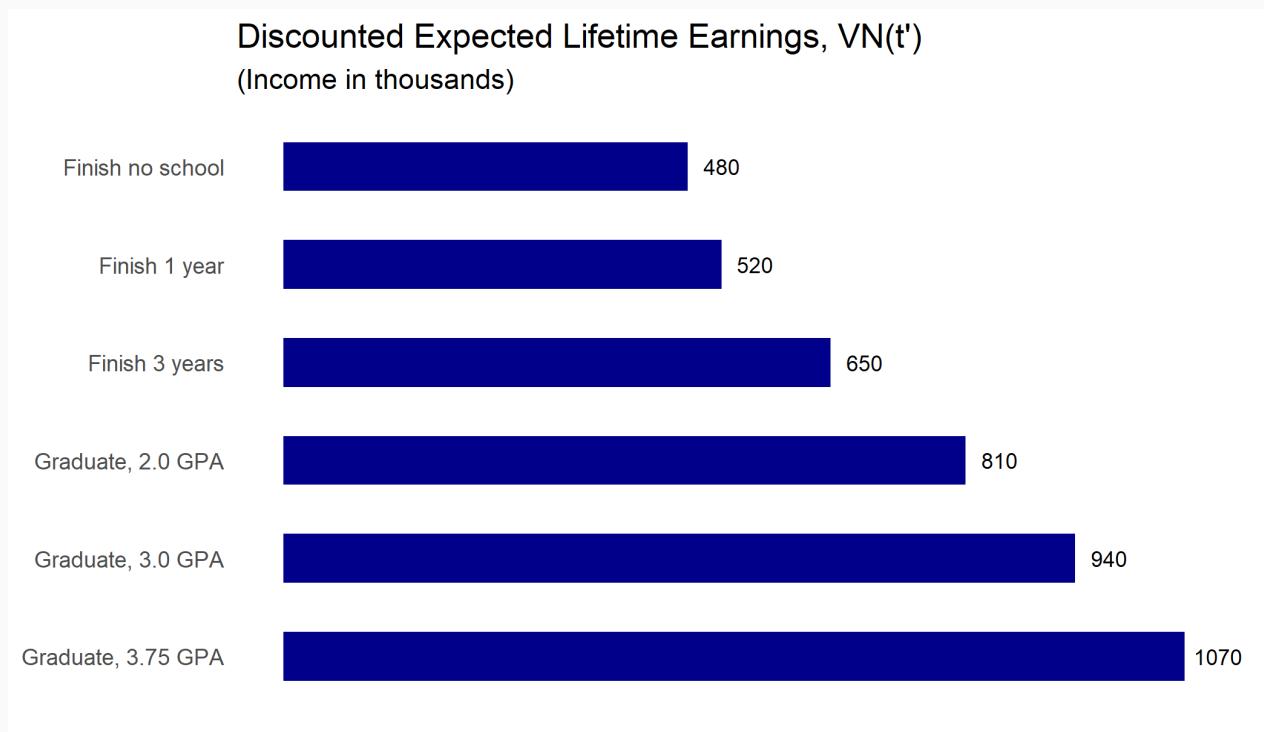
- Muestra los datos
- Maximiza el ratio entre datos y *tinta* (o texto)
- Evita *tinta* que no corresponda a datos
- No ser redundante en la información
- Evita la *basura visual*
- Prefiere disposiciones horizontales

¿Qué opinan?

Figure 2 Discounted Expected Lifetime
Earnings, $VN(t')$



Apliquemos los principios



¿Qué hay por detrás?

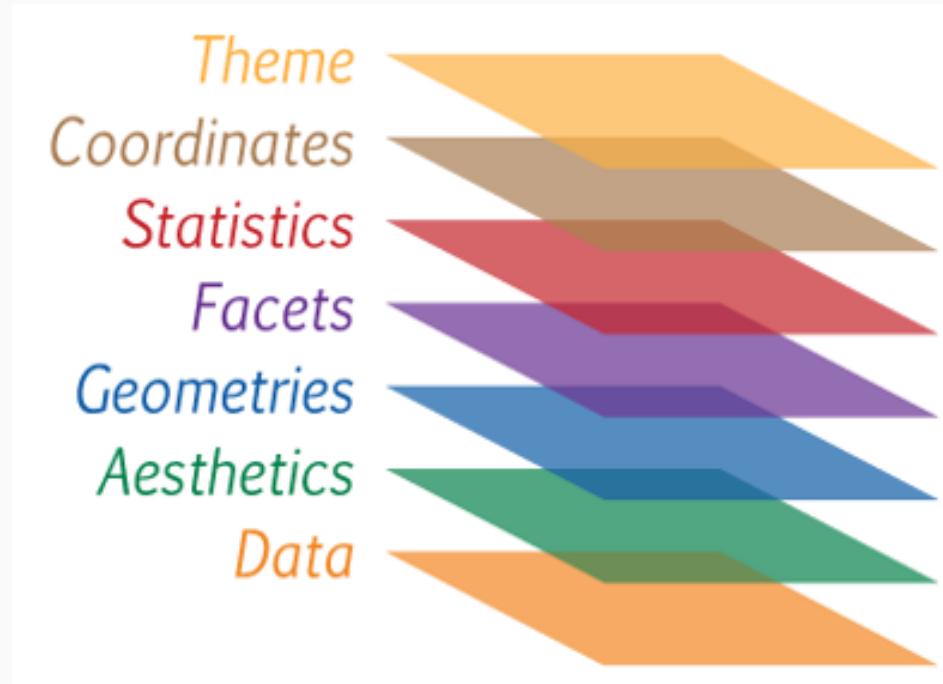
R - ggplot2

```
ggplot(datos_graf, aes(x = reorder(educ, -inc), y = inc, label = inc)) +  
  geom_col(width = 0.5, fill = "dark blue") +  
  geom_text(nudge_y = 40, size = 3) +  
  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_blank(),  
        panel.grid.minor.x = element_blank(),  
        axis.text.x = element_blank()) +  
  labs(title = "Discounted Expected Lifetime Earnings, VN(t')",  
       subtitle = "(Income in thousands)")
```

Visualización de datos 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

- CódigoViz.R

Datos

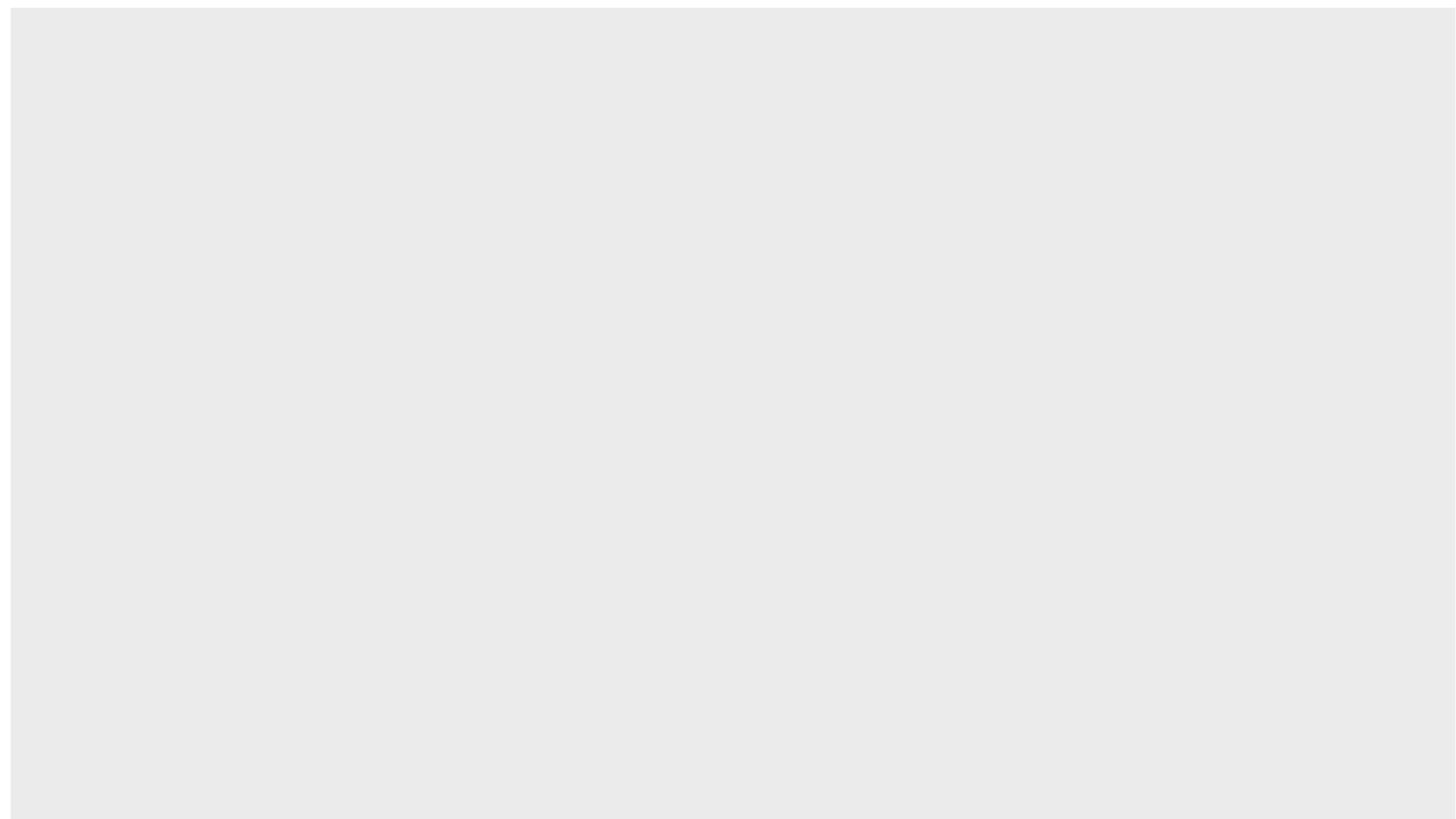
```
library(readr)
library(ggplot2)
(datos_mundo <- read_csv("../datos/datos_mundo2007.csv"))

## # A tibble: 142 x 6
##   pais     continente  anio  ExpVida      pob gdpPercap
##   <chr>    <chr>     <dbl>    <dbl>    <dbl>    <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
```

Histograma

Gráfico base (datos)

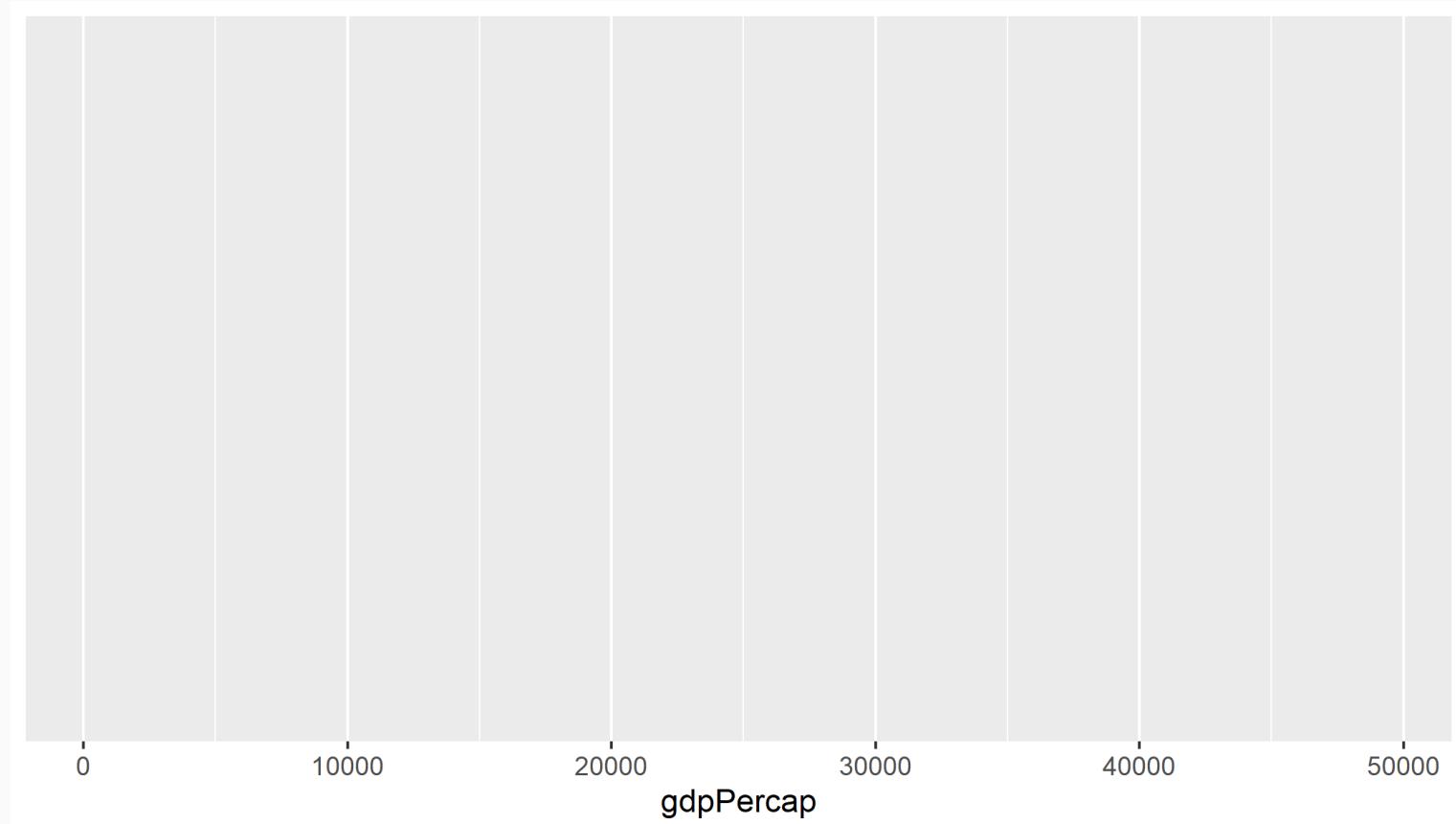
```
ggplot(datos_mundo)
```



Histograma

Agregar capa (aes)

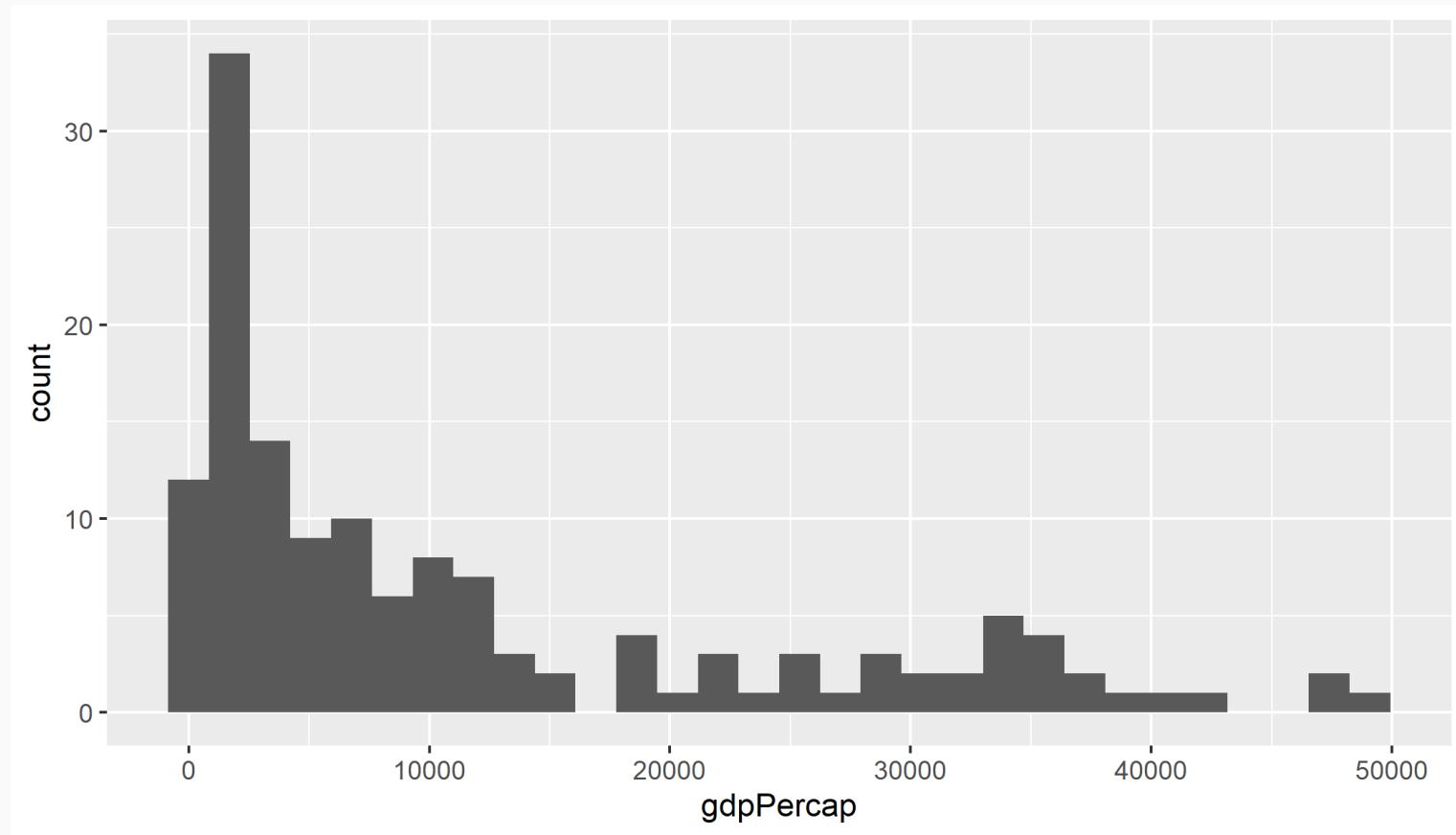
```
ggplot(datos_mundo, aes(x = gdpPercap))
```



Histograma

Agregar capa (geom)

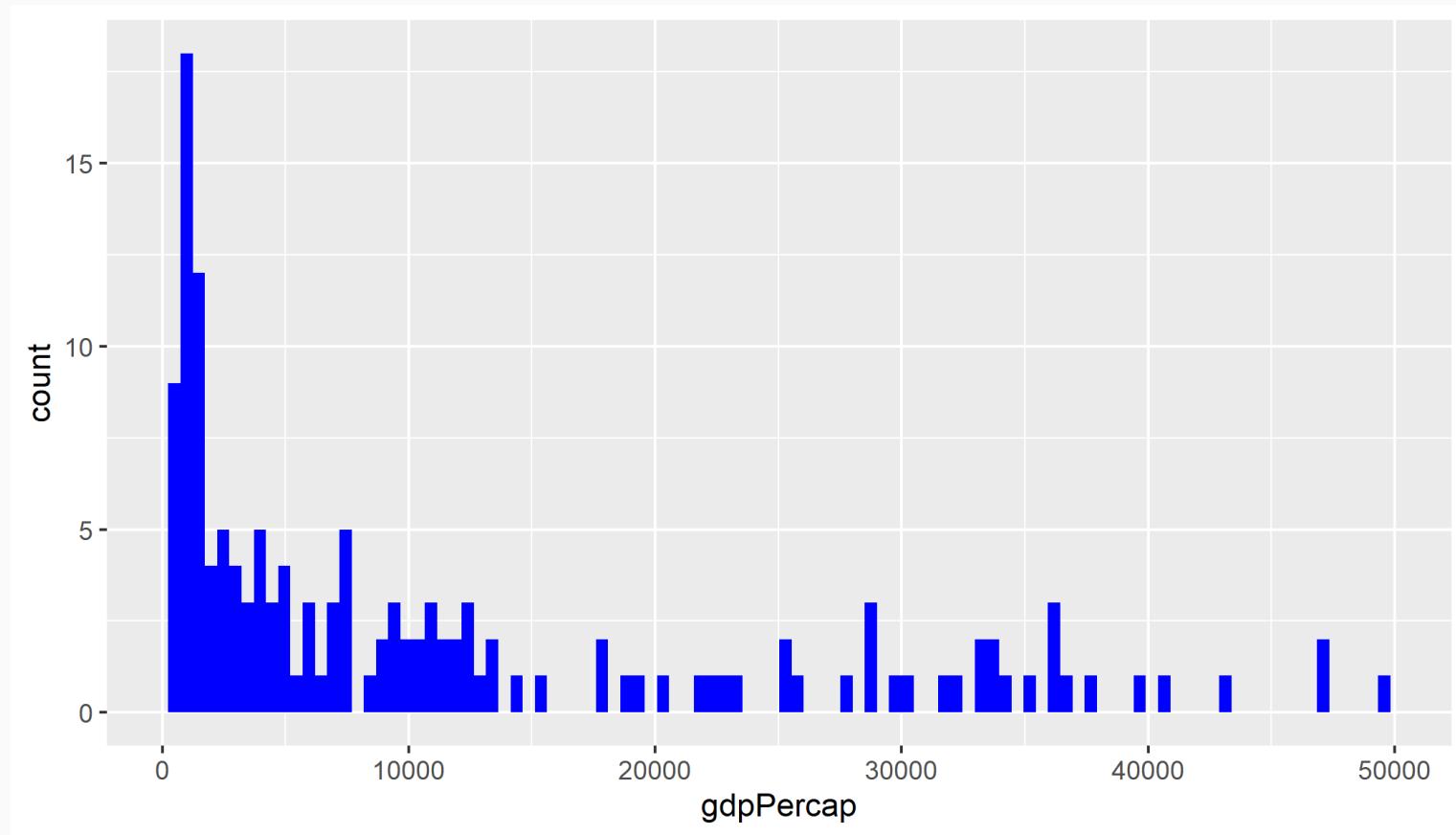
```
ggplot(datos_mundo, aes(x = gdpPerCap)) +  
  geom_histogram()
```



Histograma

Cambiar algunos argumentos

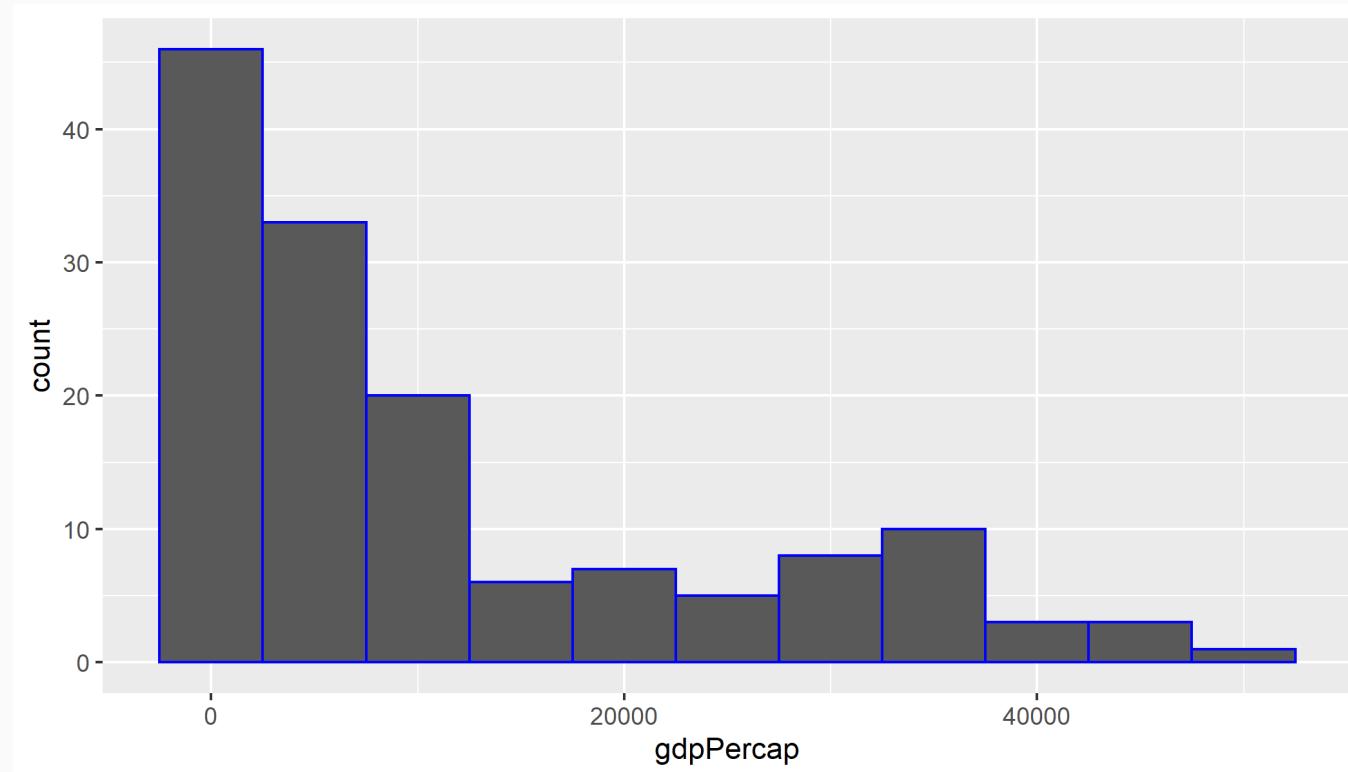
```
ggplot(datos_mundo, aes(x = gdpPercap)) +  
  geom_histogram(bins = 100, fill = "blue")
```



Histograma

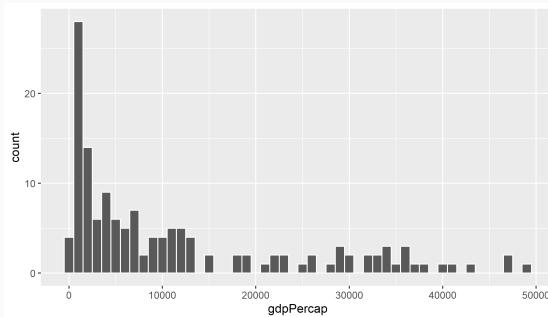
Cambiar algunos argumentos

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

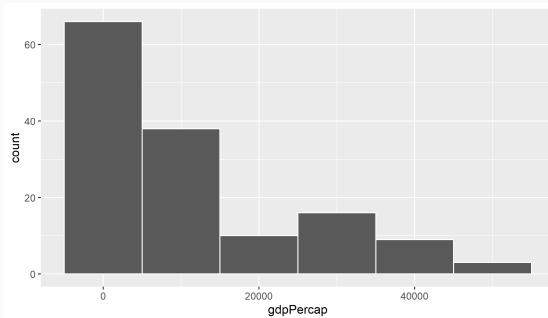


Histograma

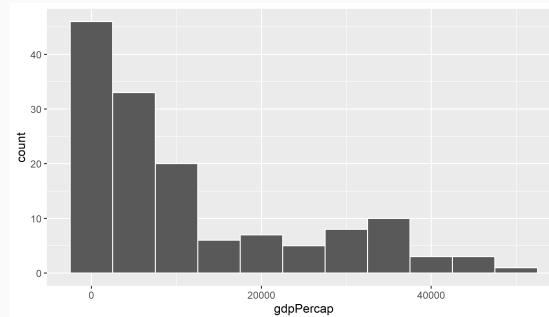
```
ggplot(datos_mundo, aes(x = gdpPercap)) +  
  geom_histogram(binwidth = 1000, col = "white")
```



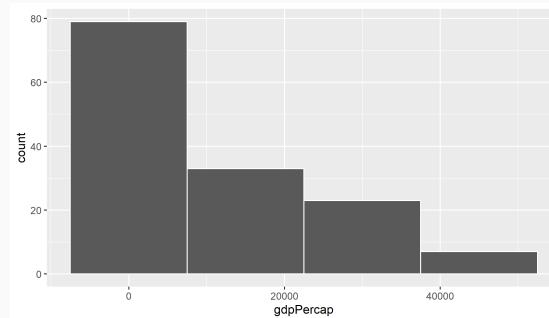
```
ggplot(datos_mundo, aes(x = gdpPercap)) +  
  geom_histogram(binwidth = 10000, col = "white")
```



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

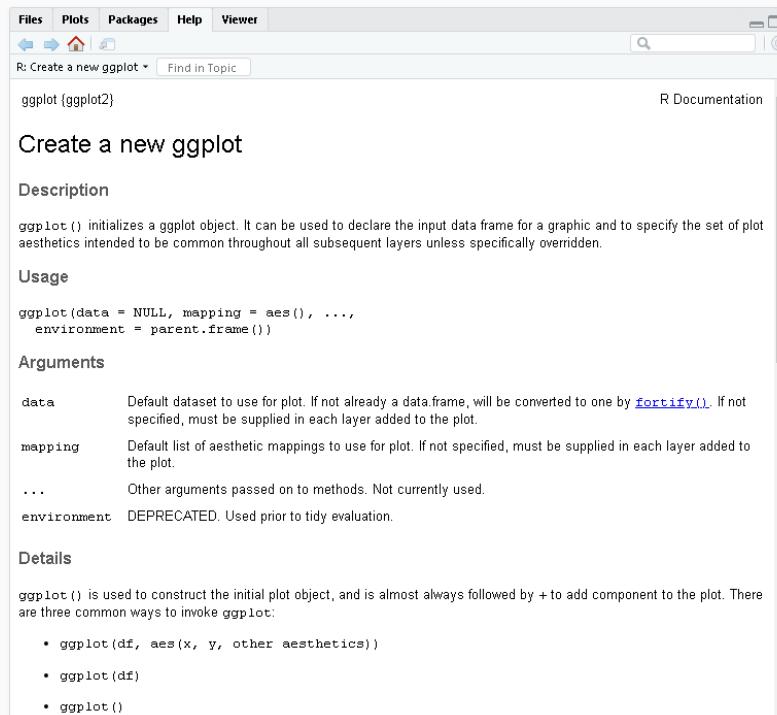


```
ggplot(datos_mundo, aes(x = gdpPercap)) +  
  geom_histogram(binwidth = 15000, col = "white")
```



Muchos argumentos

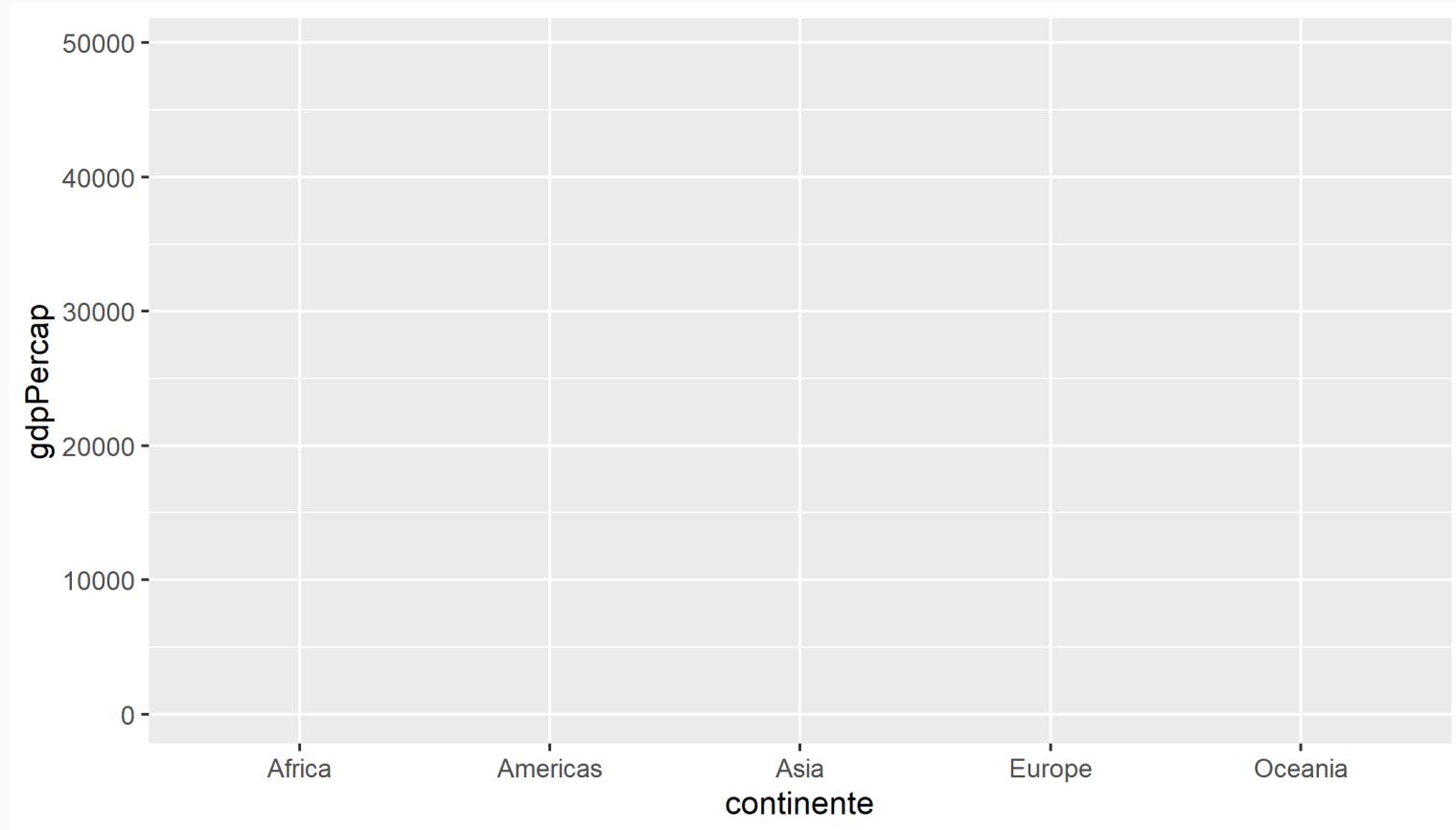
NO OLVIDAR: Siempre consulten `?nombrefunción`. Ej: `?ggplot`



The screenshot shows the R Help Viewer window. The title bar says "R Documentation". The main area displays the documentation for the `ggplot` function from the `ggplot2` package. The title is "Create a new ggplot". The "Description" section states that `ggplot()` initializes a ggplot object, used to declare input data frame and specify plot aesthetics. The "Usage" section shows the function signature: `ggplot(data = NULL, mapping = aes(), ..., environment = parent.frame())`. The "Arguments" section details four parameters: `data` (dataset), `mapping` (aesthetic mappings), `...` (other arguments), and `environment` (deprecated). The "Details" section notes that `ggplot()` constructs the initial plot object and lists three ways to invoke it: `ggplot(df, aes(x, y, other aesthetics))`, `ggplot(df)`, and `ggplot()`.

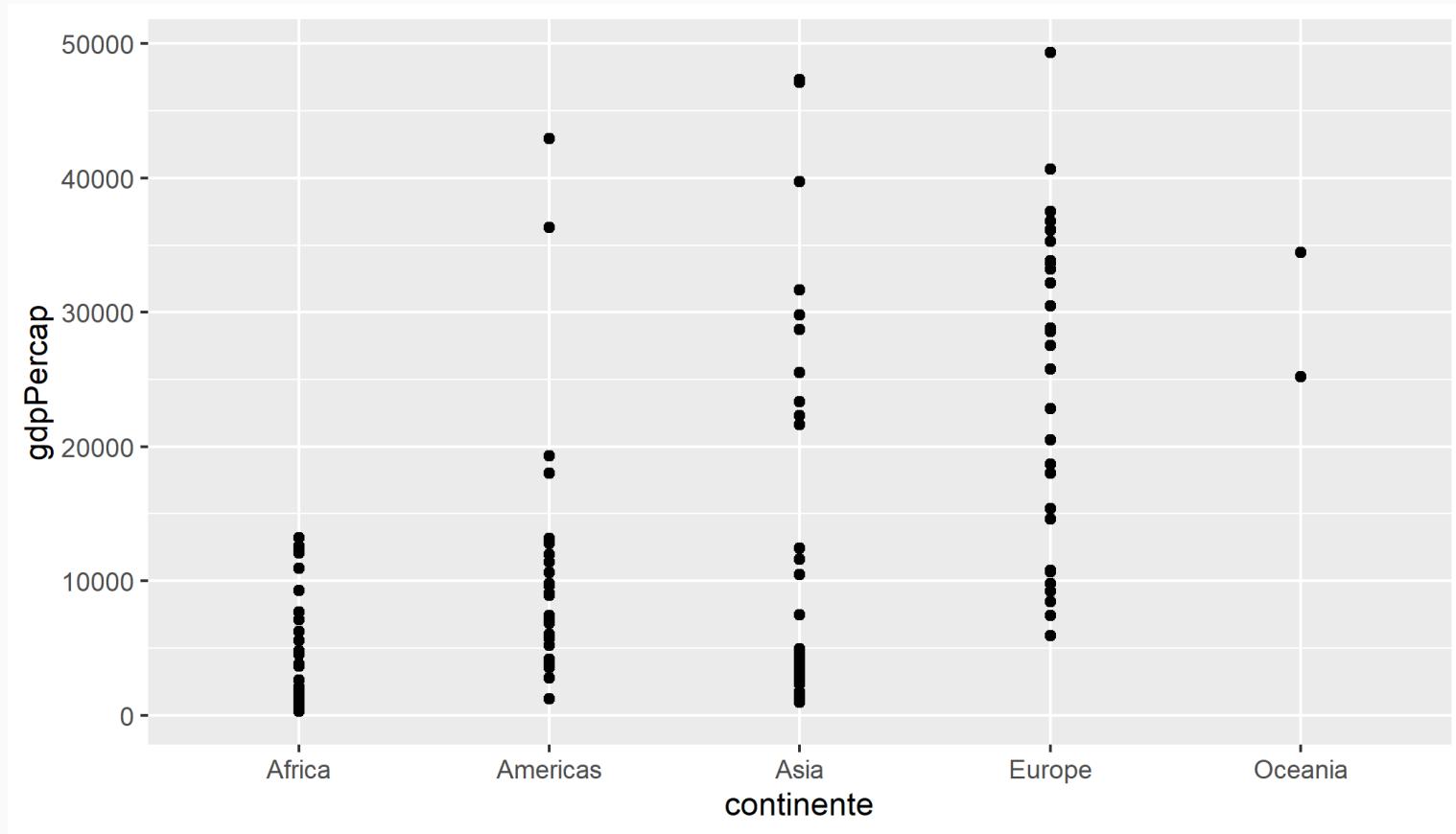
Más de una variable

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



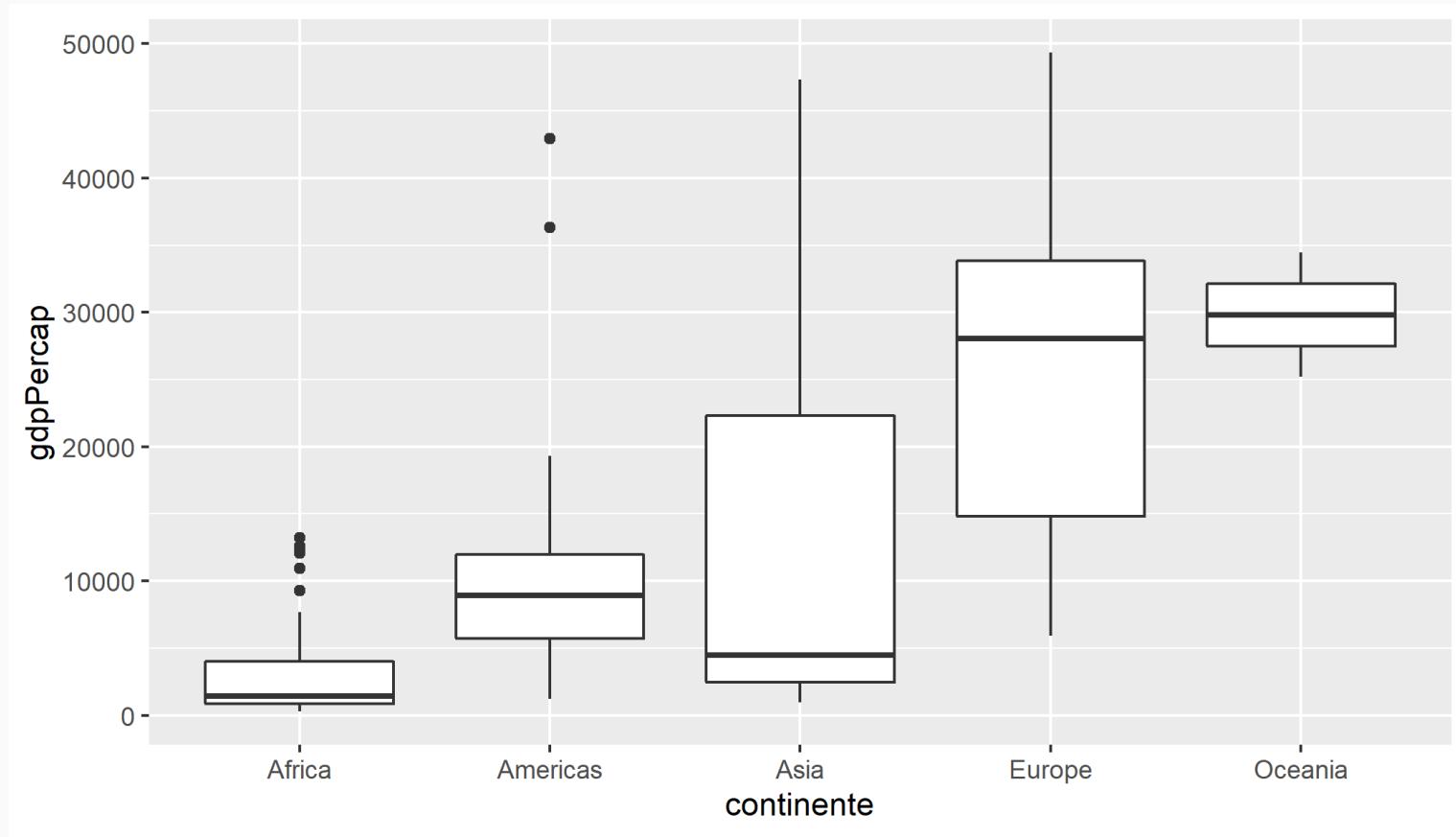
Capa de puntos (geom_point)

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

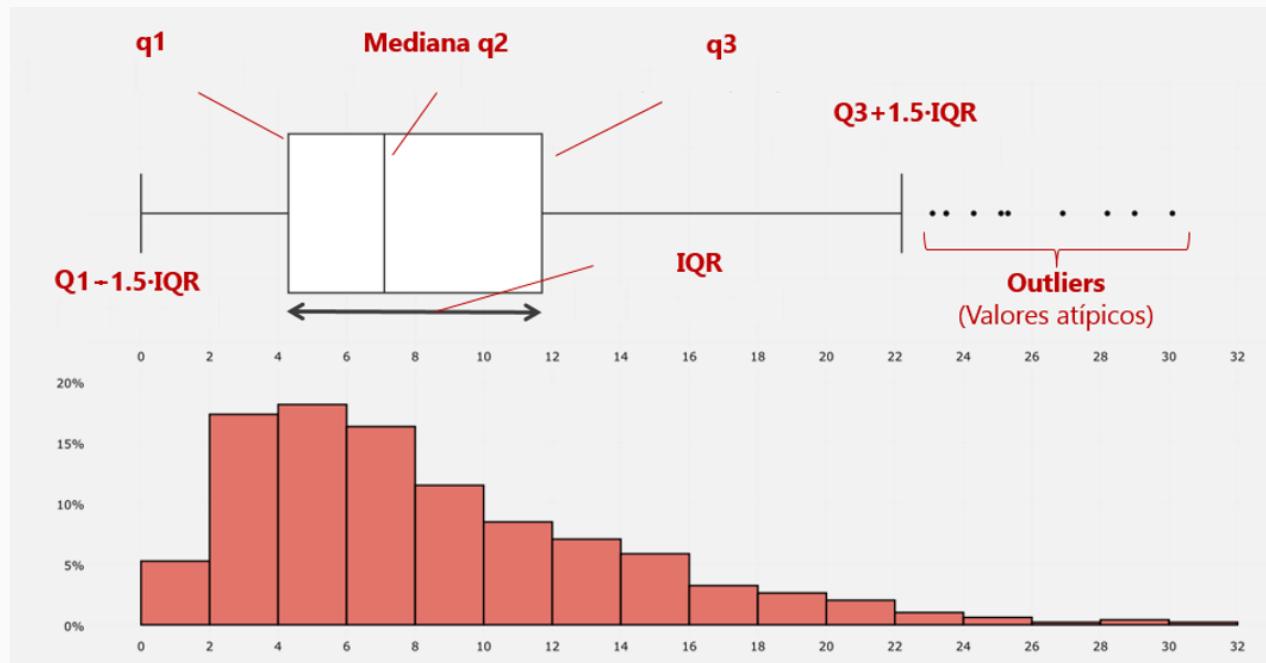


Boxplot

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

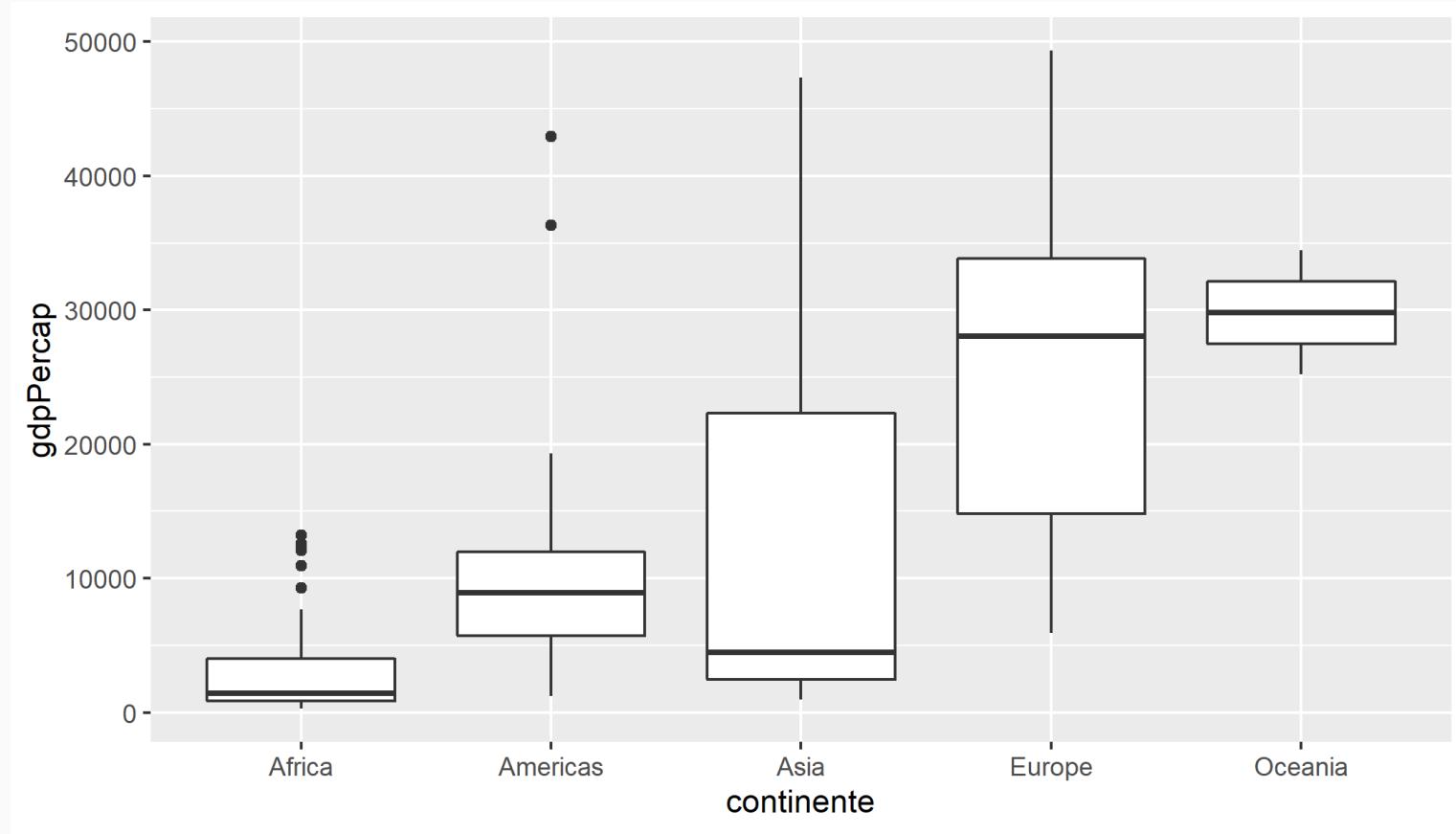


¿Qué nos muestra el boxplot?



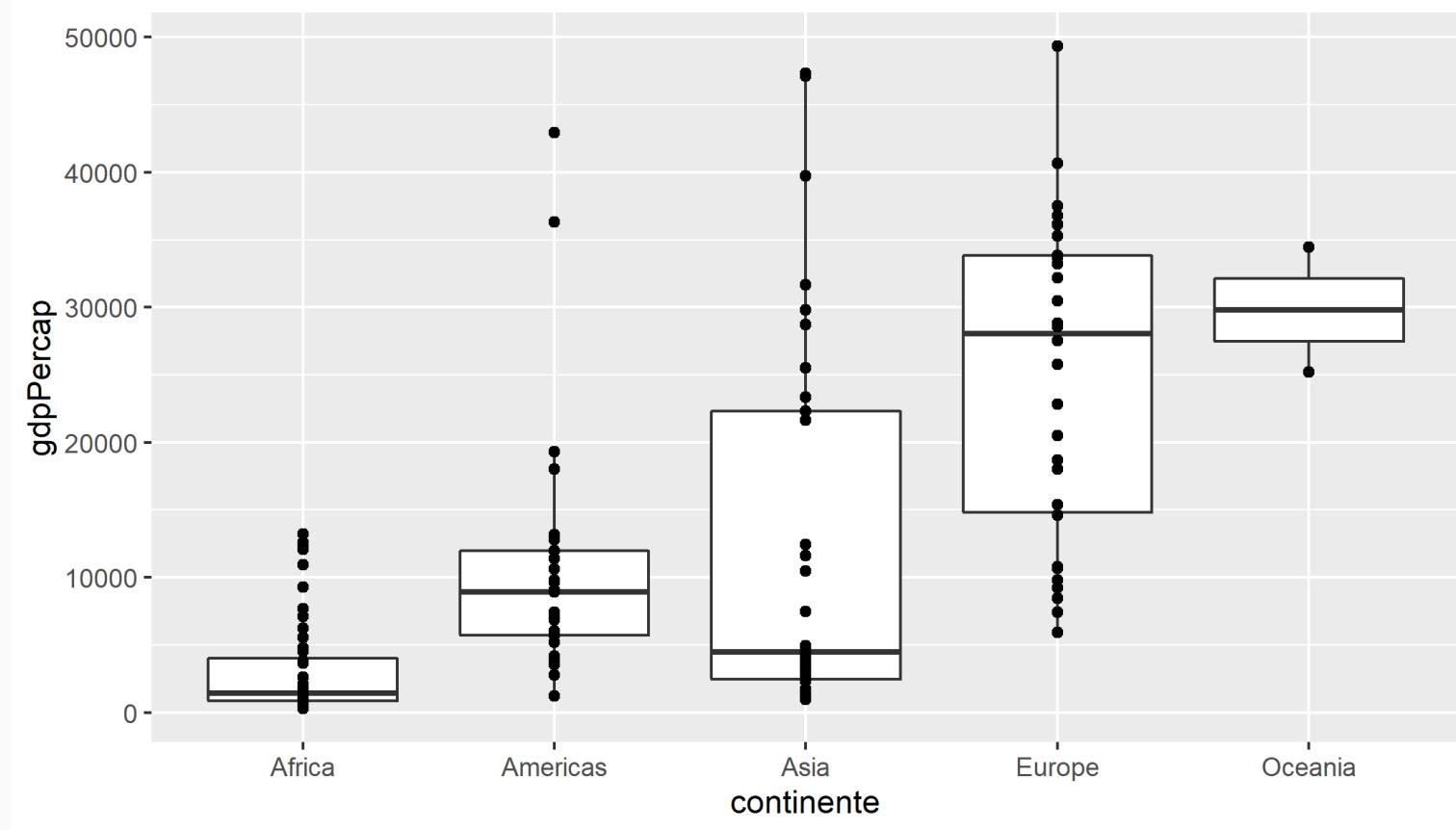
Juntar más de una capa (geom)

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



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()
```

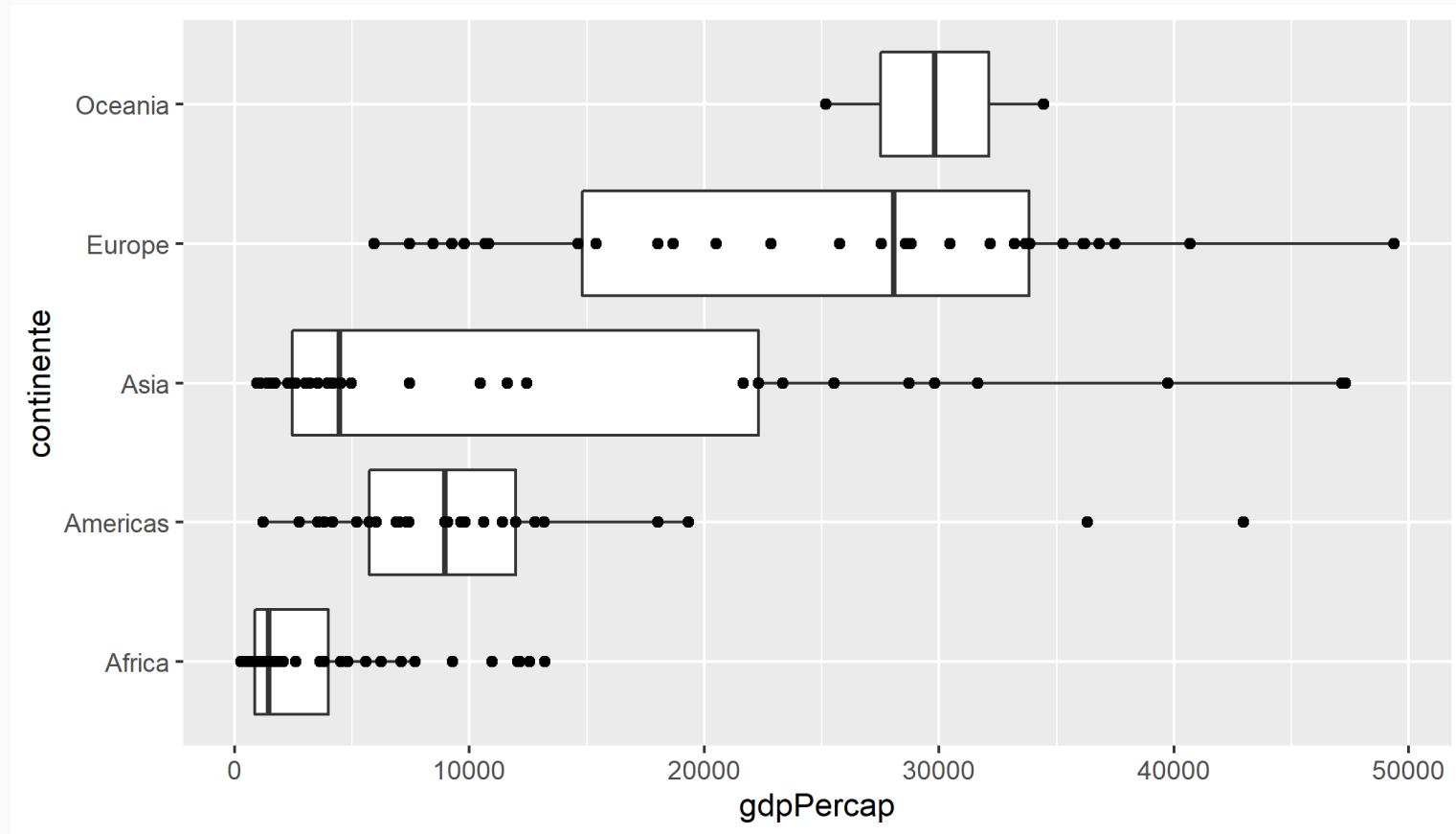
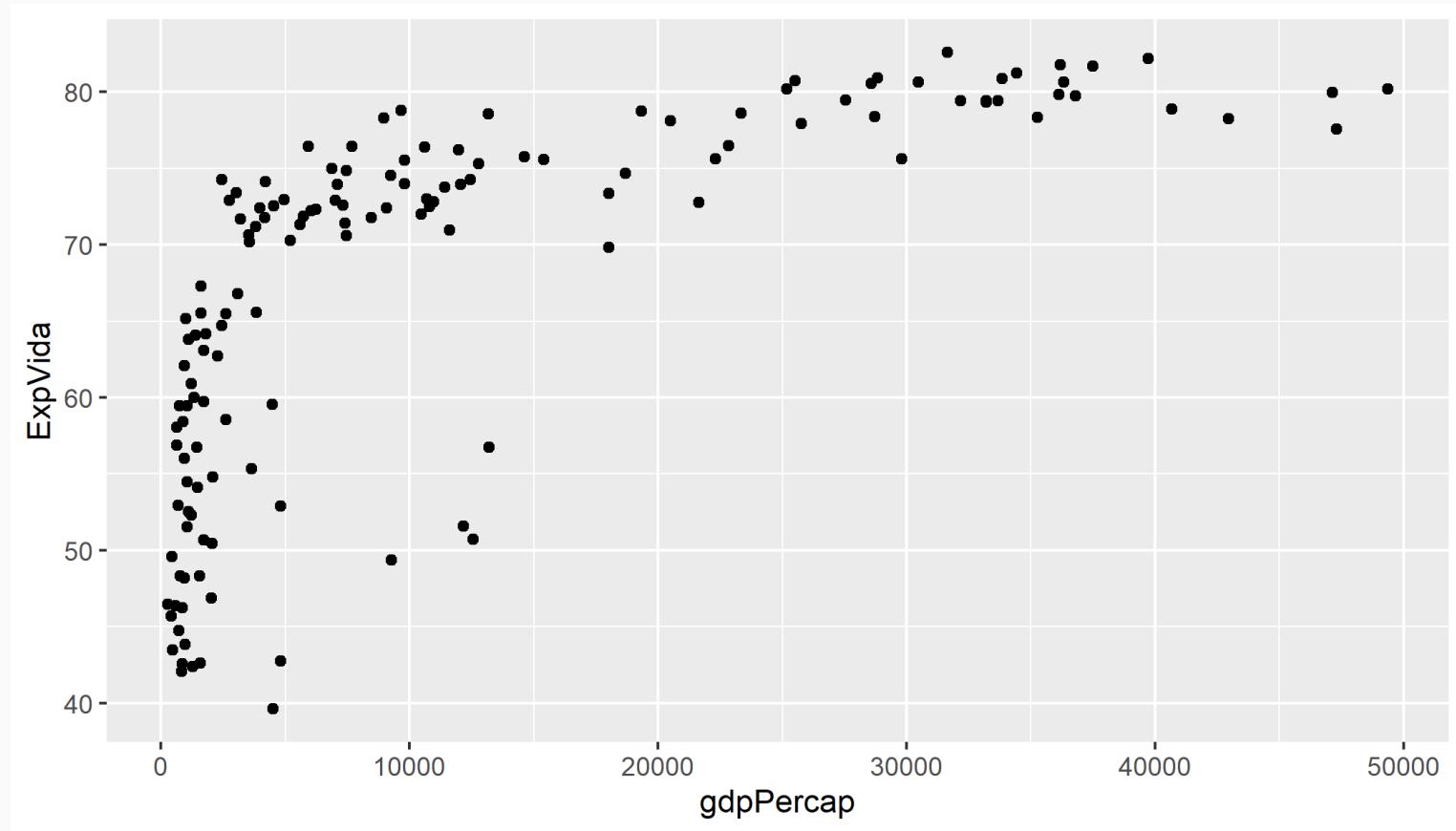


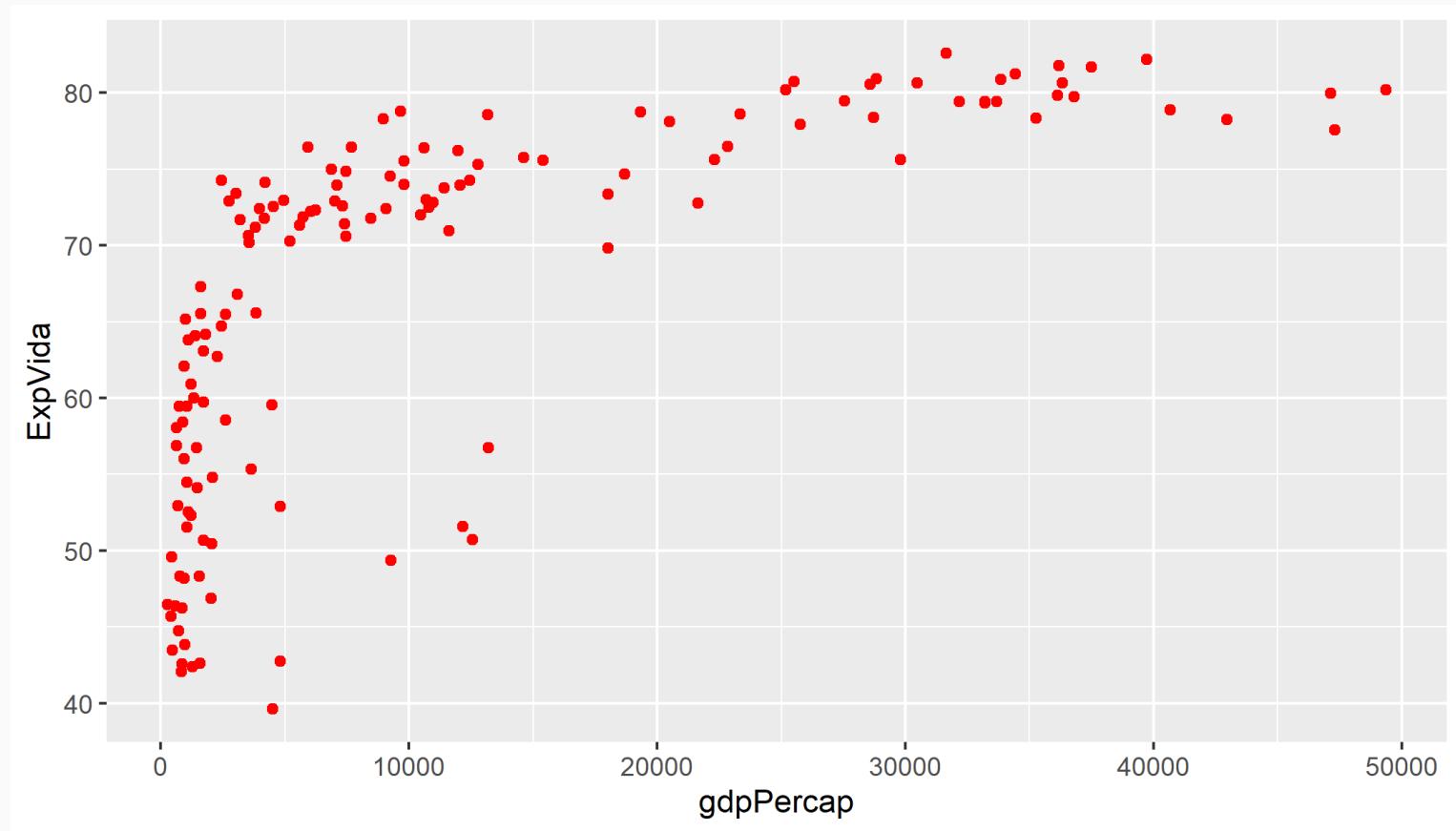
Gráfico de dispersión

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



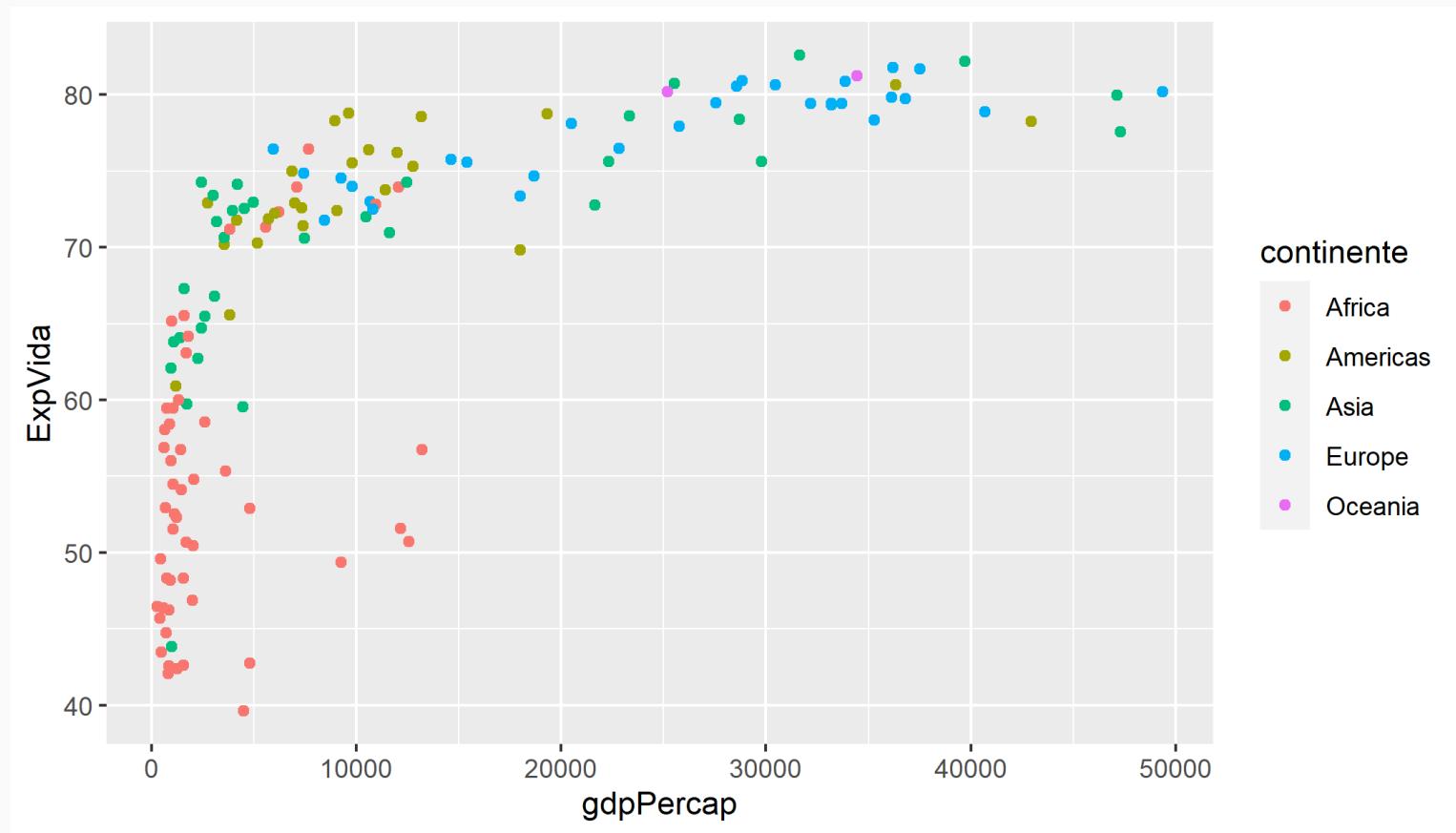
Cambiar el color de los puntos

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



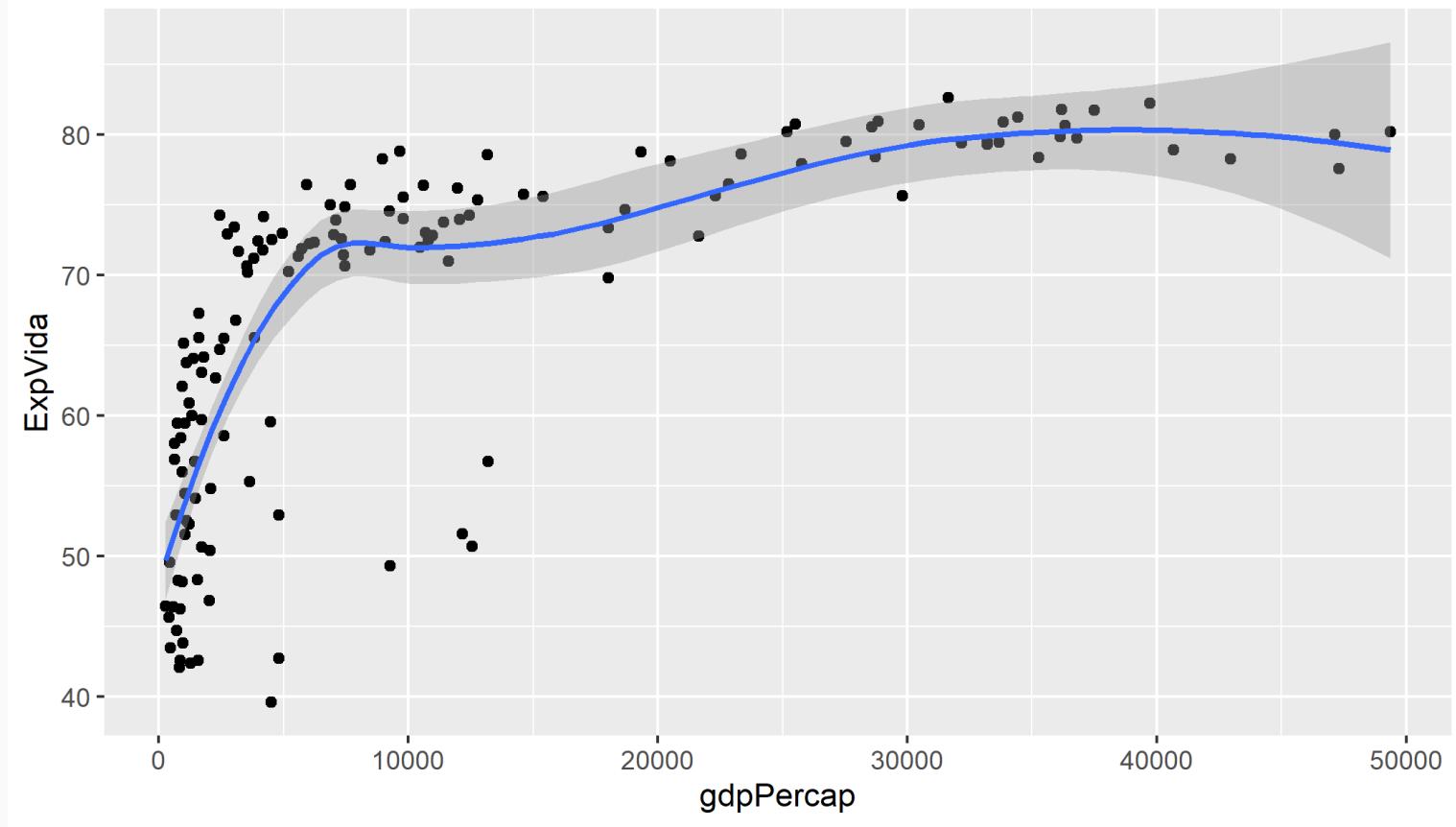
Color para agregar más información

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



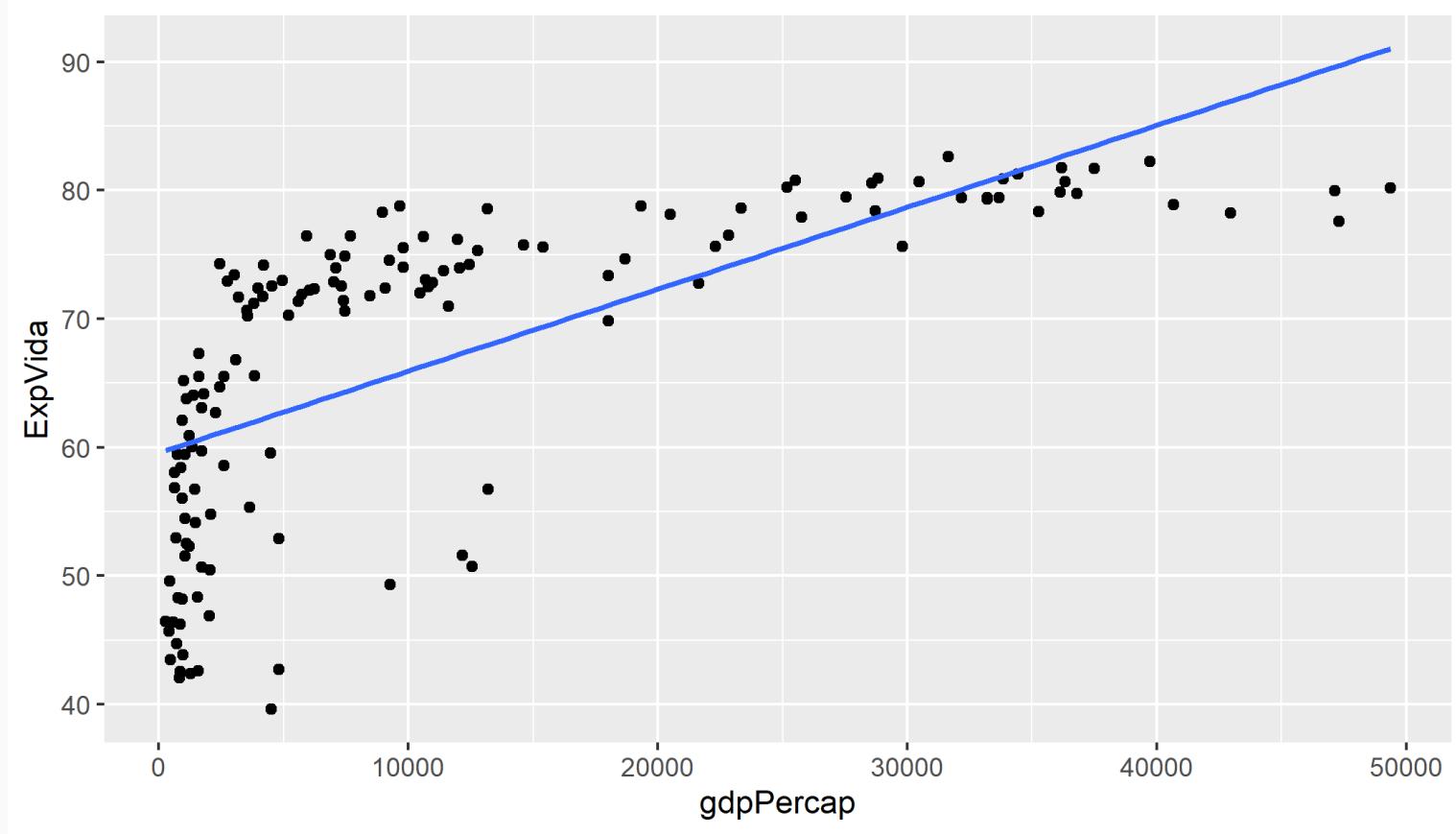
Linea de tendencia

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



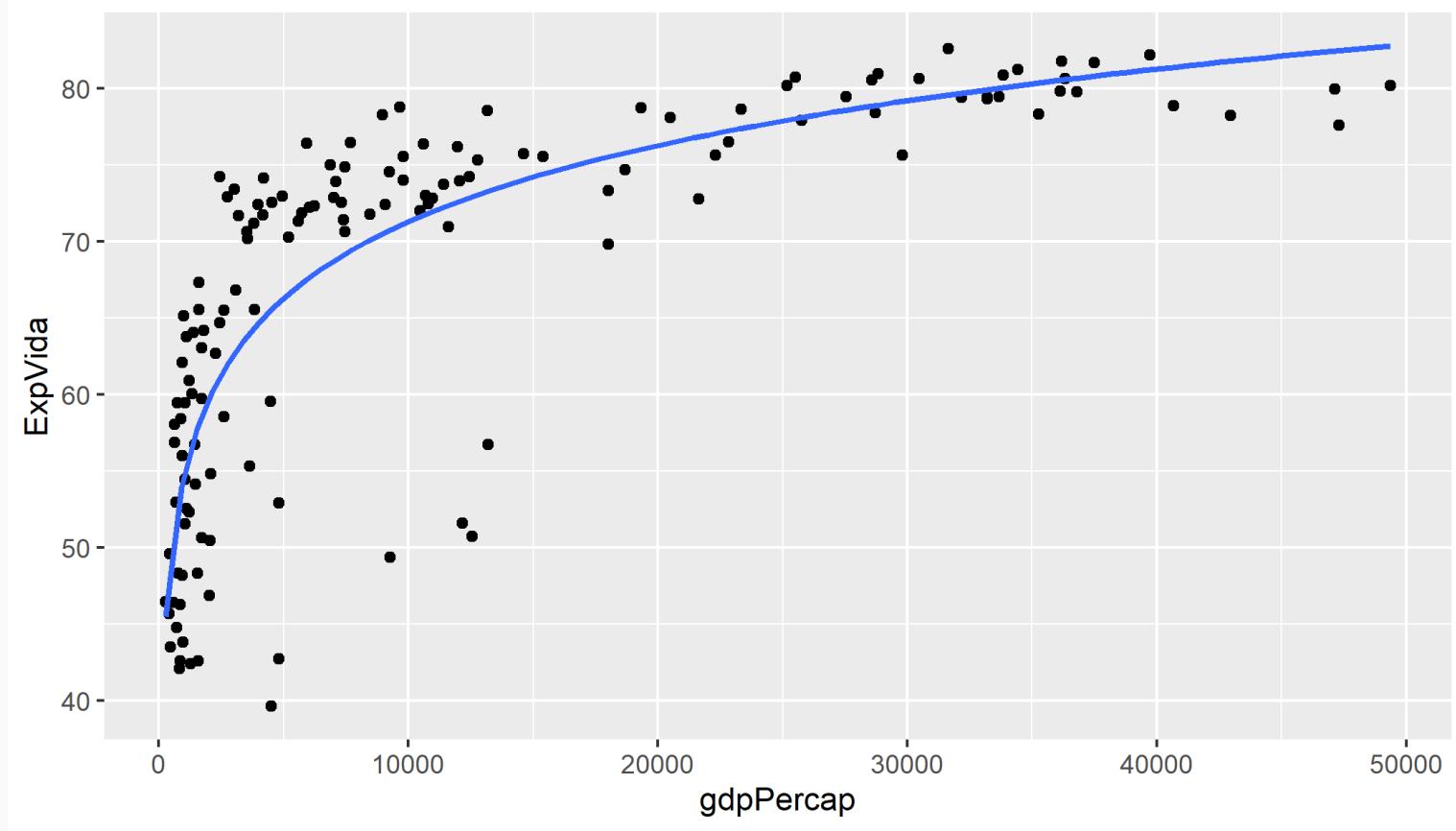
Linea de tendencia

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



Linea de tendencia

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



Ejercicio

Ejercicio - Script

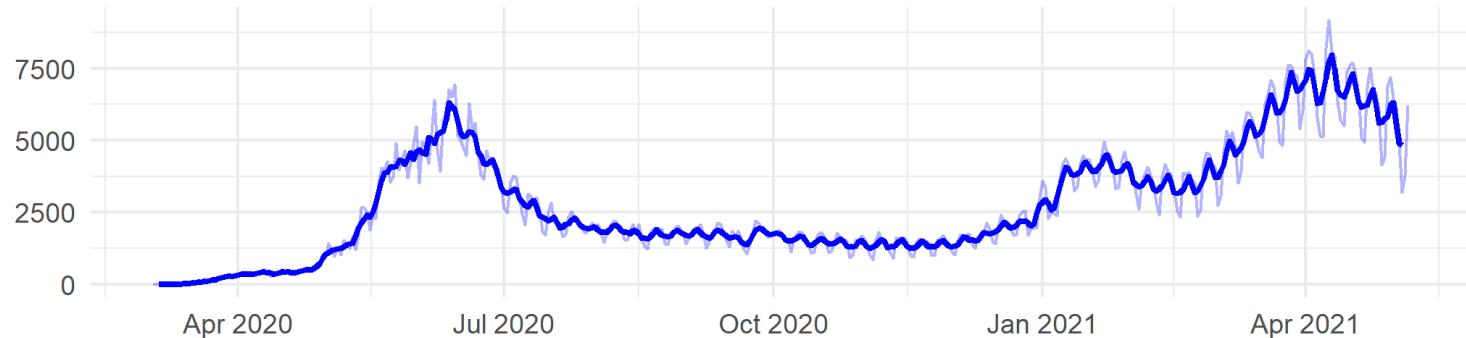
- EjercicioViz.R

Demo - Datos COVID Chile

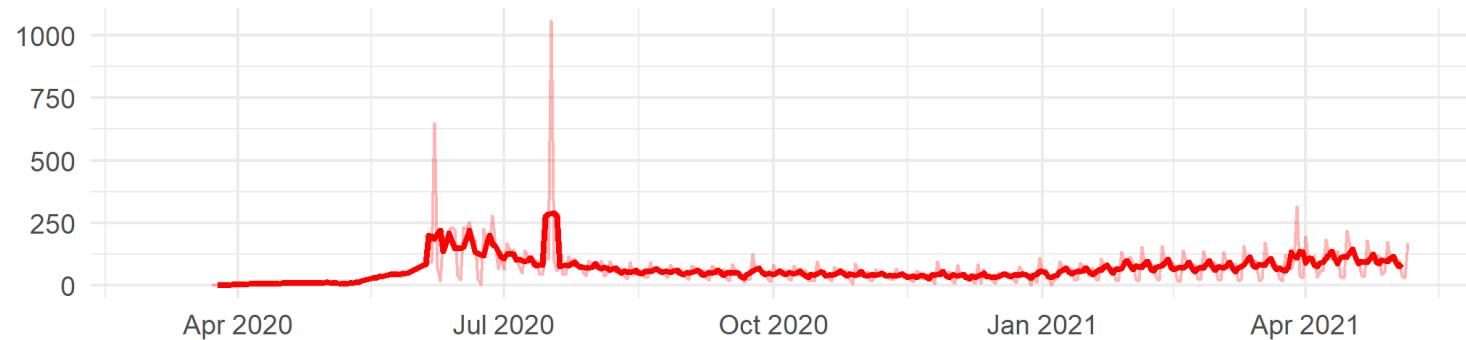
COVID en Chile

¿Segunda o primera ola?

Casos diarios COVID-19 a nivel nacional



Fallecidos diarios por COVID-19 a nivel nacional



Datos obtenidos desde repositorio del Ministerio de Ciencia y Tecnología (<https://github.com/MinCiencia/Datos-COVID19>). Los datos han sido manipulados, no citar.

Demo - Script

- CódigoCovidViz.R

Datos COVID-19

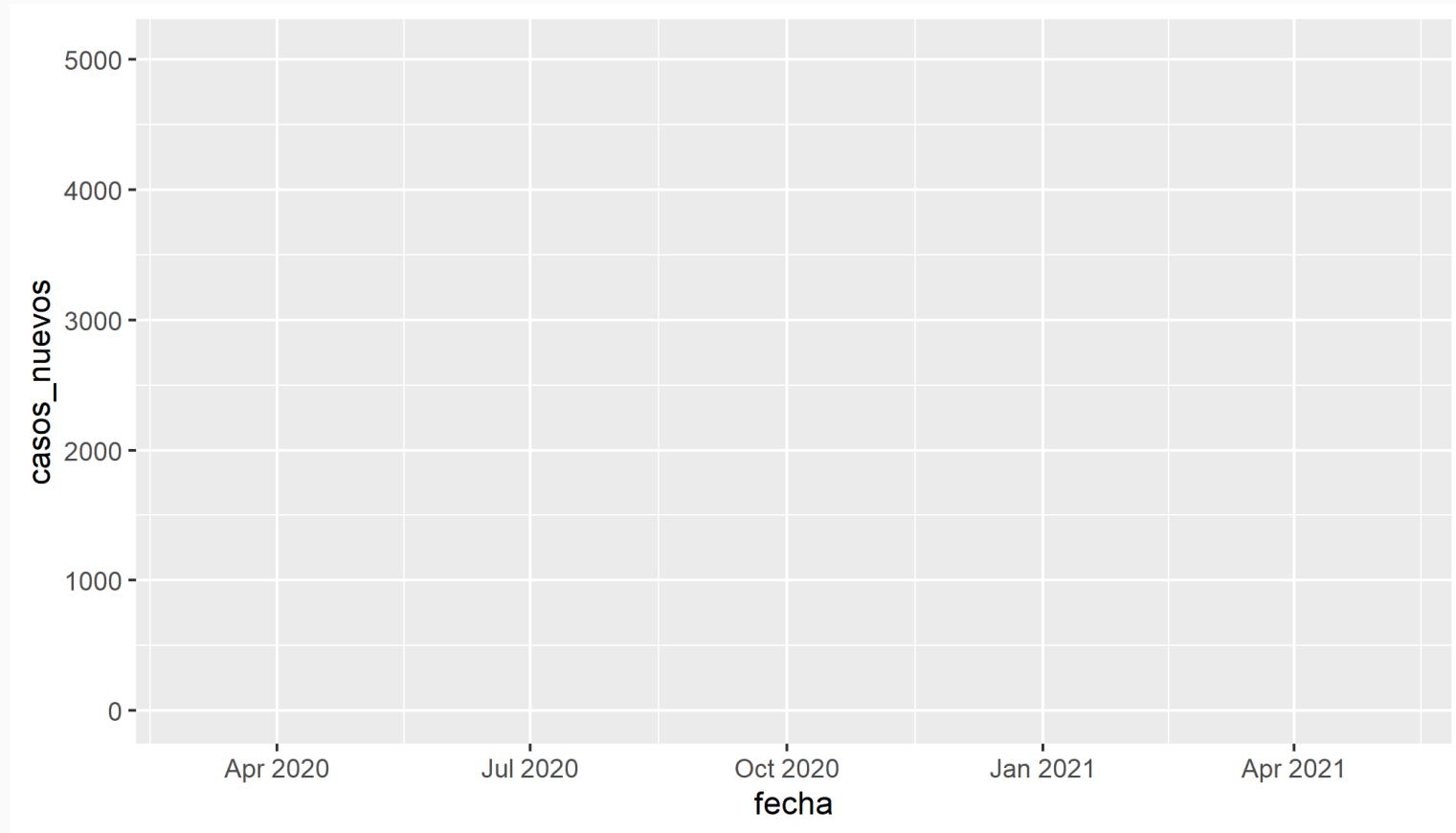
Cargar datos

```
library(readr)
(datos_covid ← read_csv("../datos/DatosCOVIDRegion.csv"))

## # A tibble: 7,310 x 6
##   region      fecha    casos_nuevos fallecidos fallecidos_d casos_nuevos_d
##   <chr>       <date>     <dbl>        <dbl>        <dbl>        <dbl>
## 1 Arica y Parin~ 2020-03-03      NA          NA          NA          0
## 2 Arica y Parin~ 2020-03-04      NA          NA          NA          0
## 3 Arica y Parin~ 2020-03-05      0           NA          NA          0
## 4 Arica y Parin~ 2020-03-06      0           NA          NA          0
## 5 Arica y Parin~ 2020-03-07      0           NA          NA          0
## 6 Arica y Parin~ 2020-03-08      0           NA          NA          0
## 7 Arica y Parin~ 2020-03-09      0           NA          NA          0
## 8 Arica y Parin~ 2020-03-10      0           NA          NA          0
## 9 Arica y Parin~ 2020-03-11      0           NA          NA          0
## 10 Arica y Parin~ 2020-03-12      0           NA          NA          0
## # ... with 7,300 more rows
```

Gráfico base

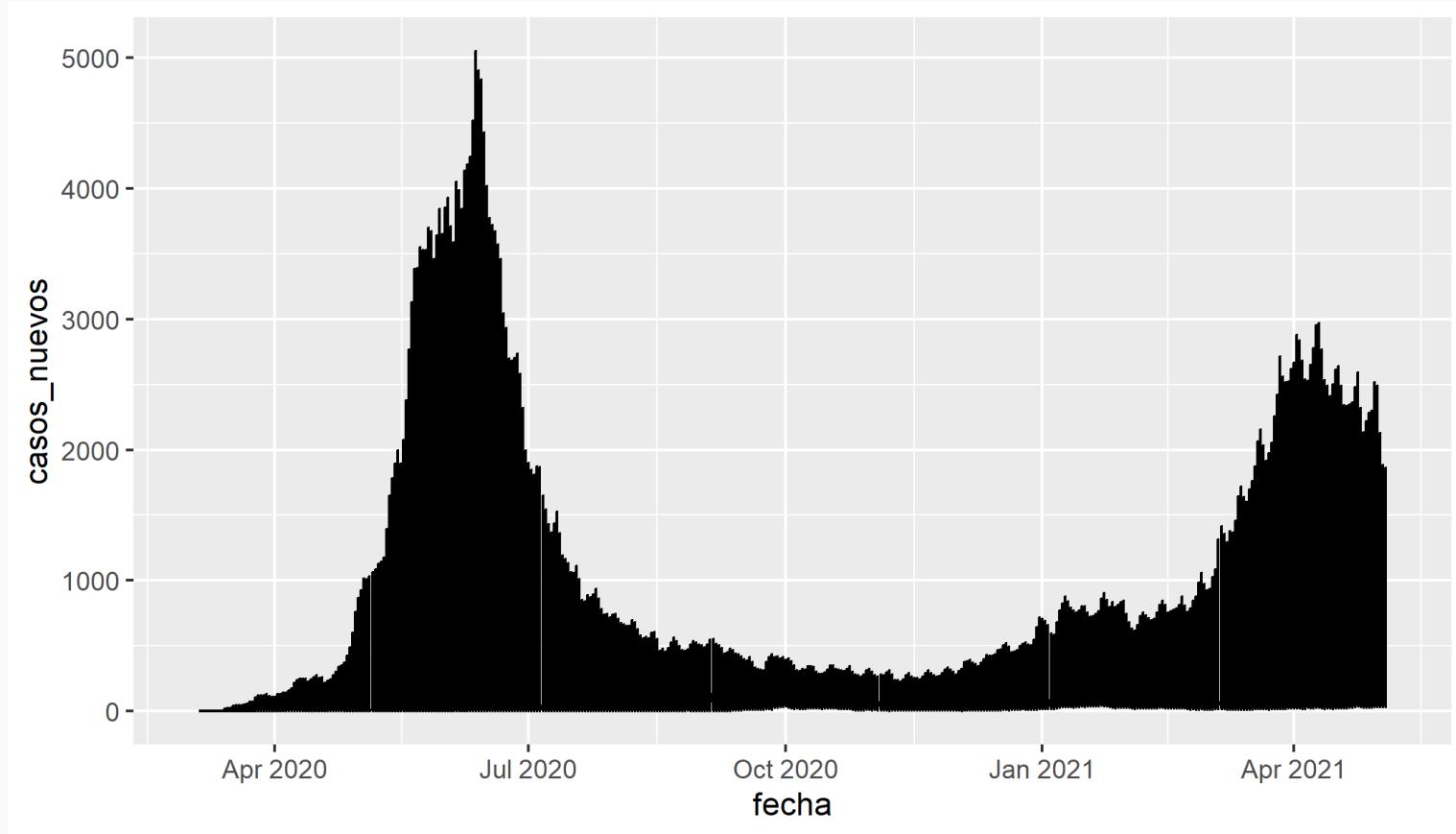
```
library(ggplot2)
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos))
```



Agregar `geom` de lineas

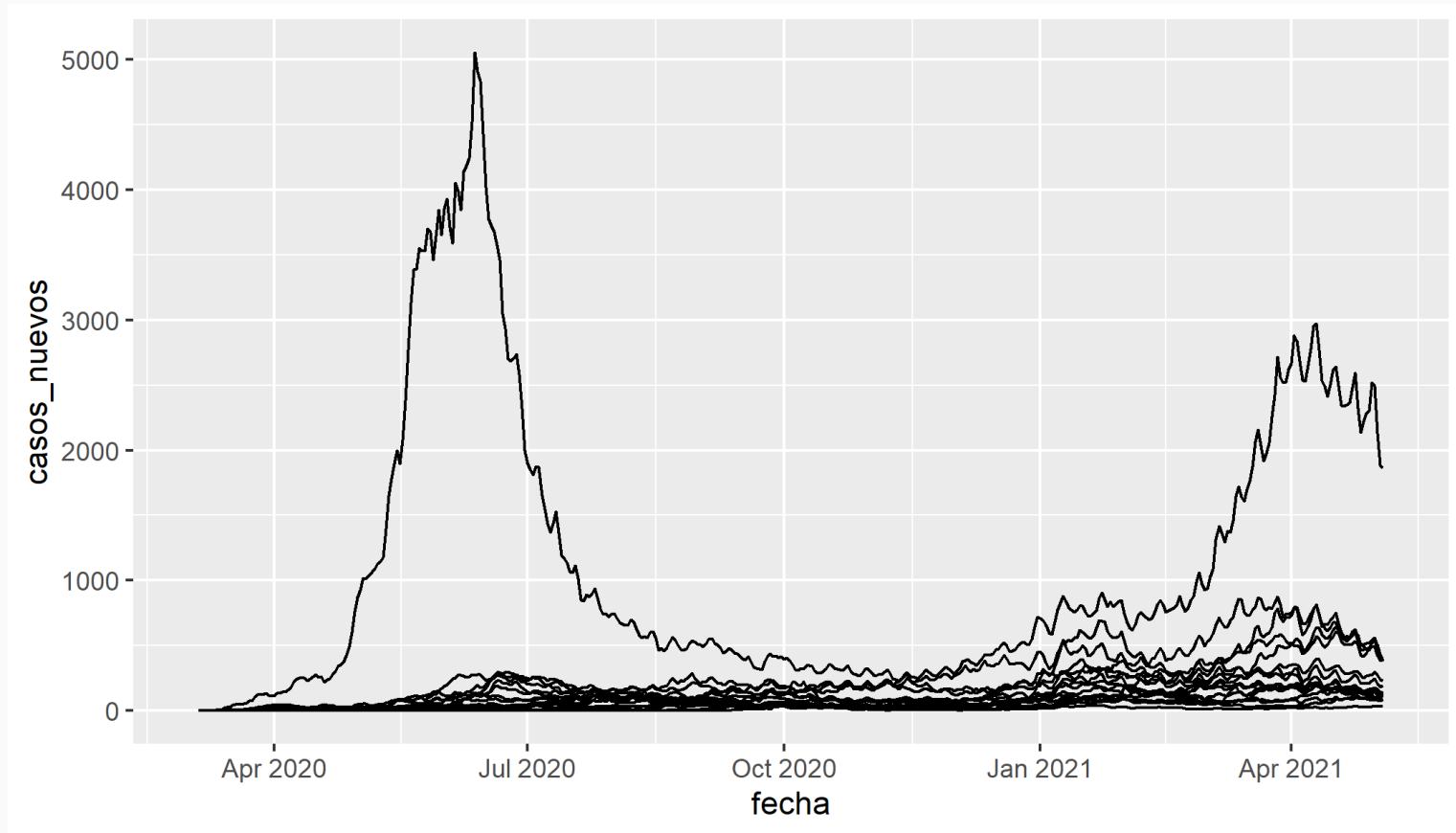
Pero algo se ve mal

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



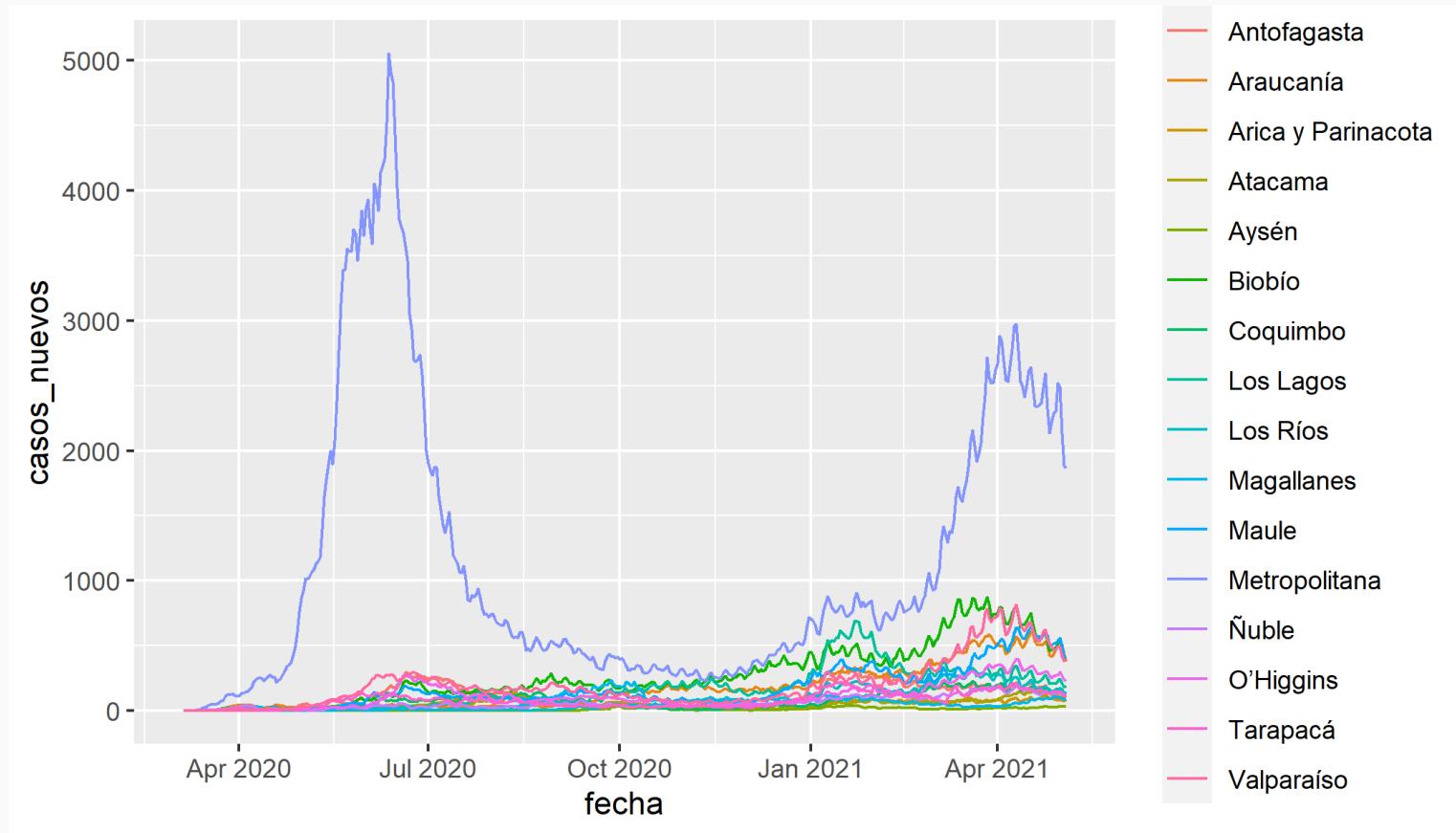
Una linea para cada región

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



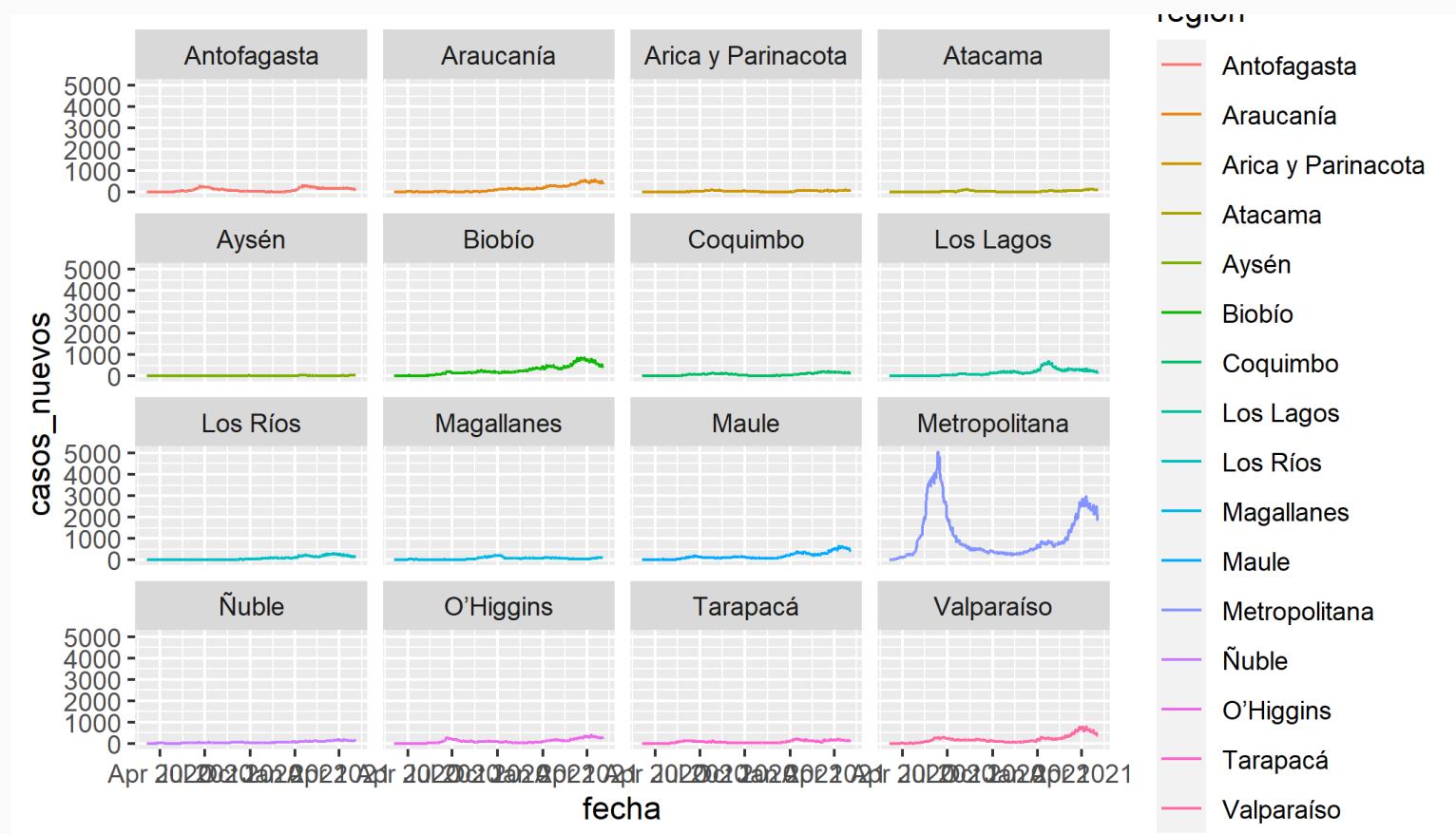
Una linea para cada región

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



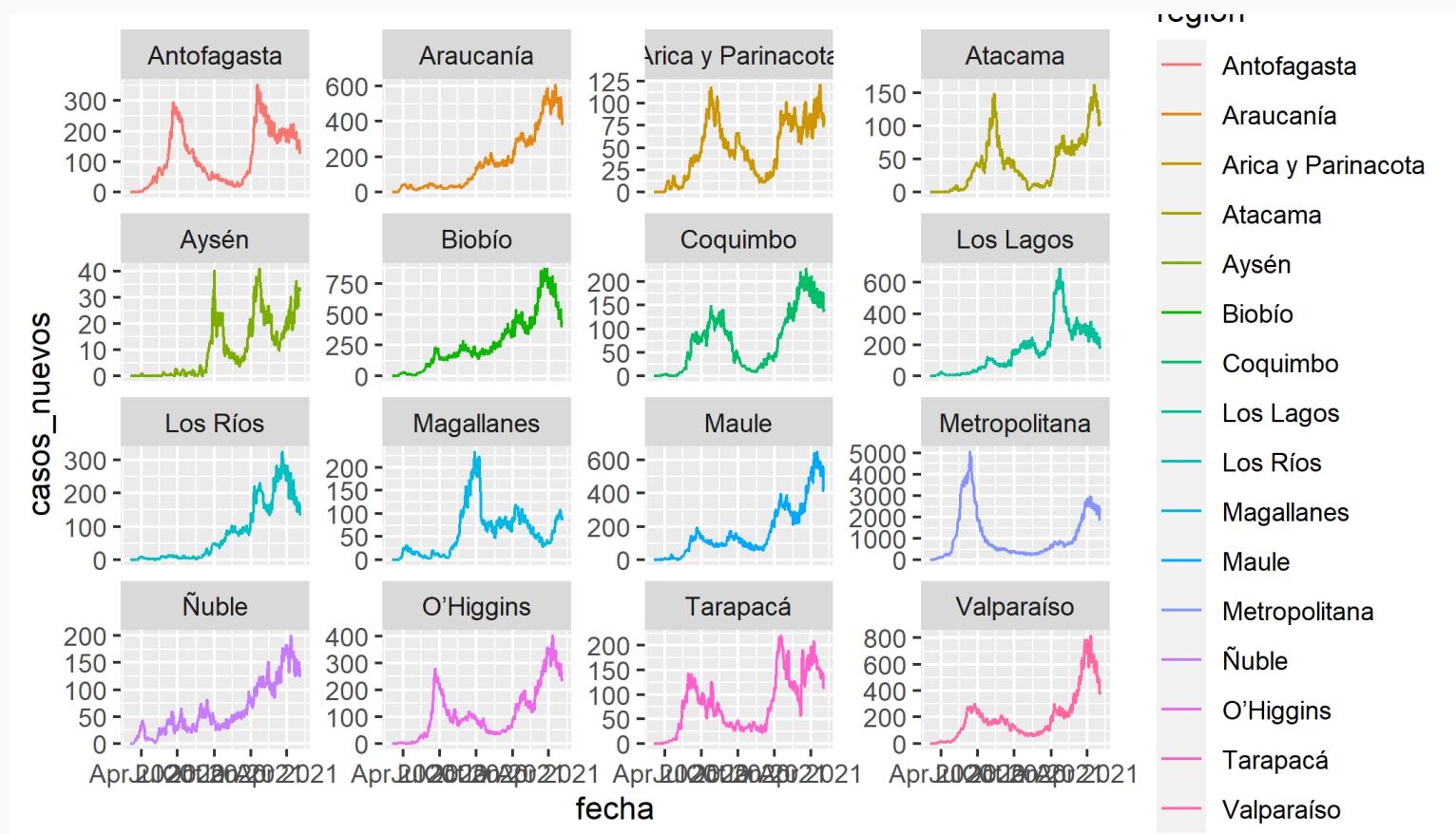
Cada linea en su propio panel

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



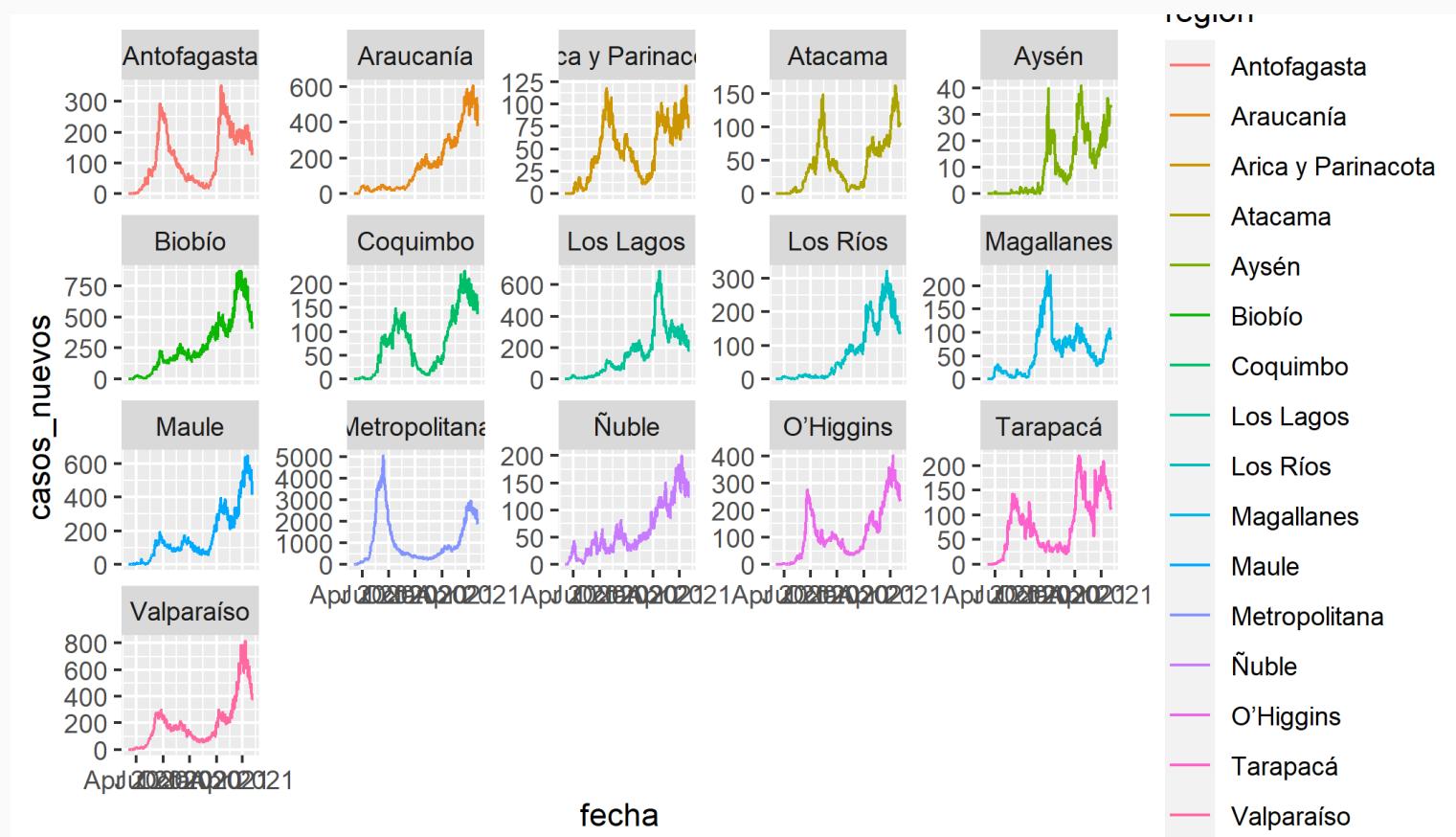
Cada linea en su propio panel

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



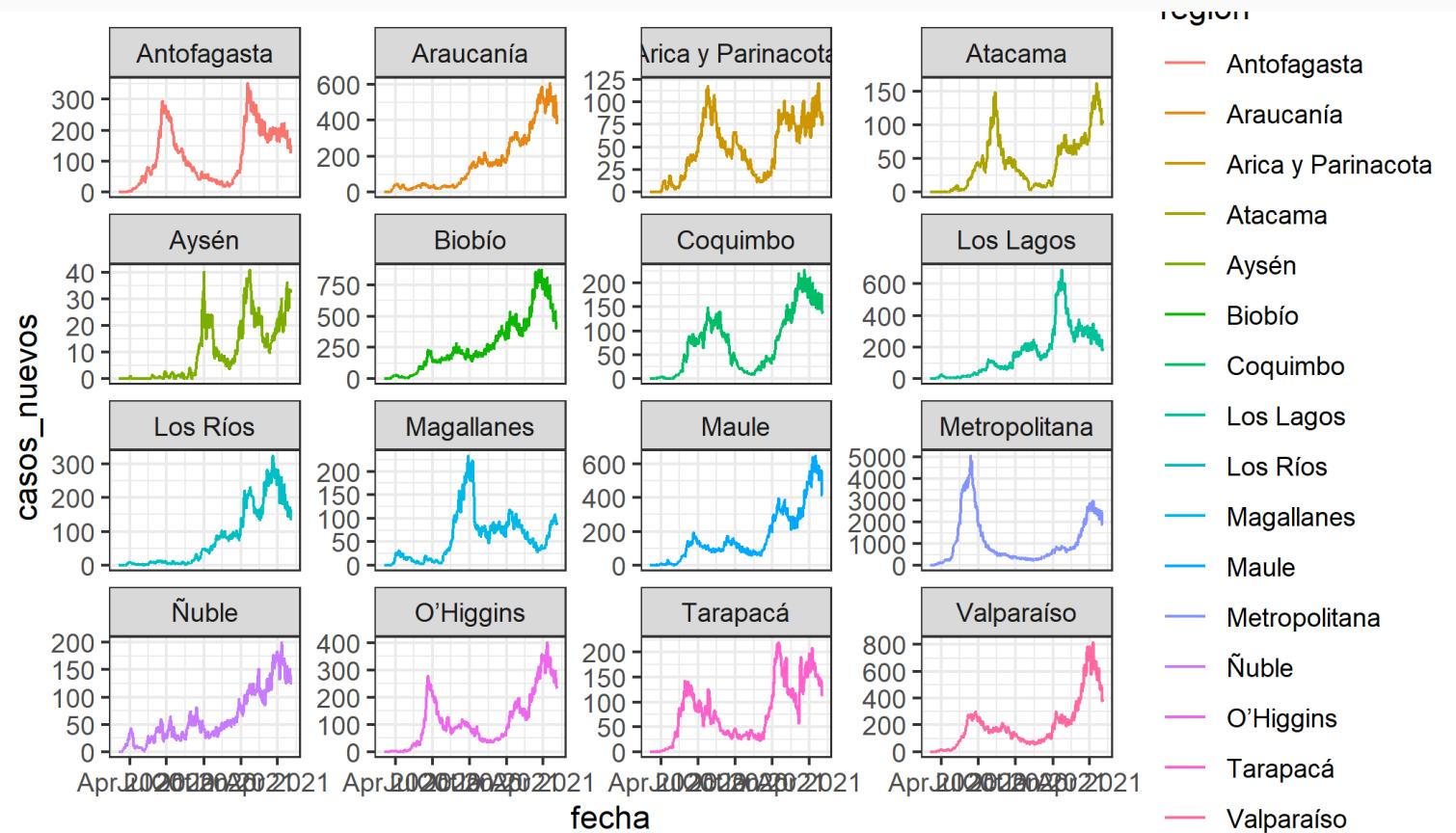
Cada linea en su propio panel

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y", ncol = 5)
```



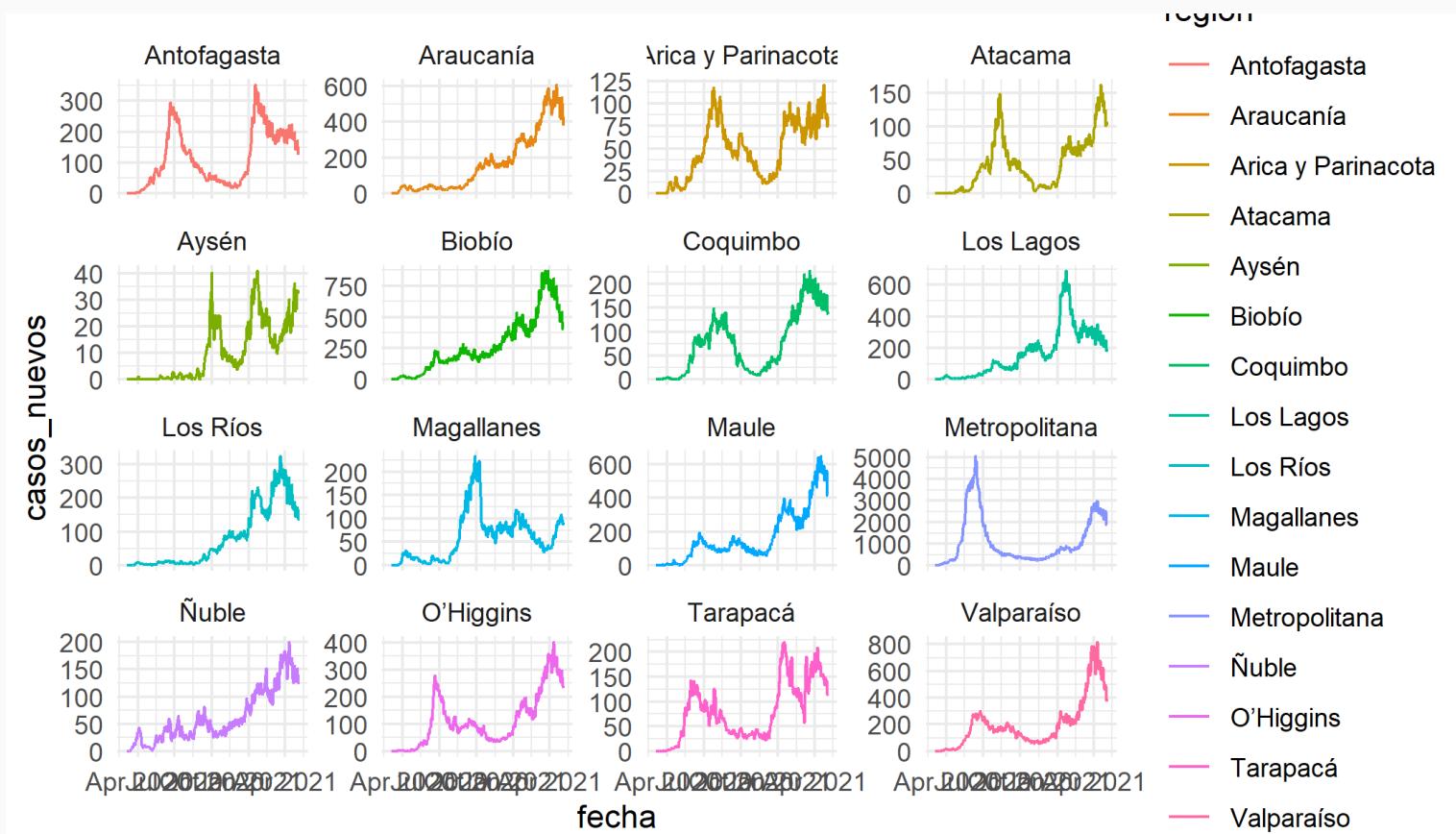
Fondo blanco parece mejor

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



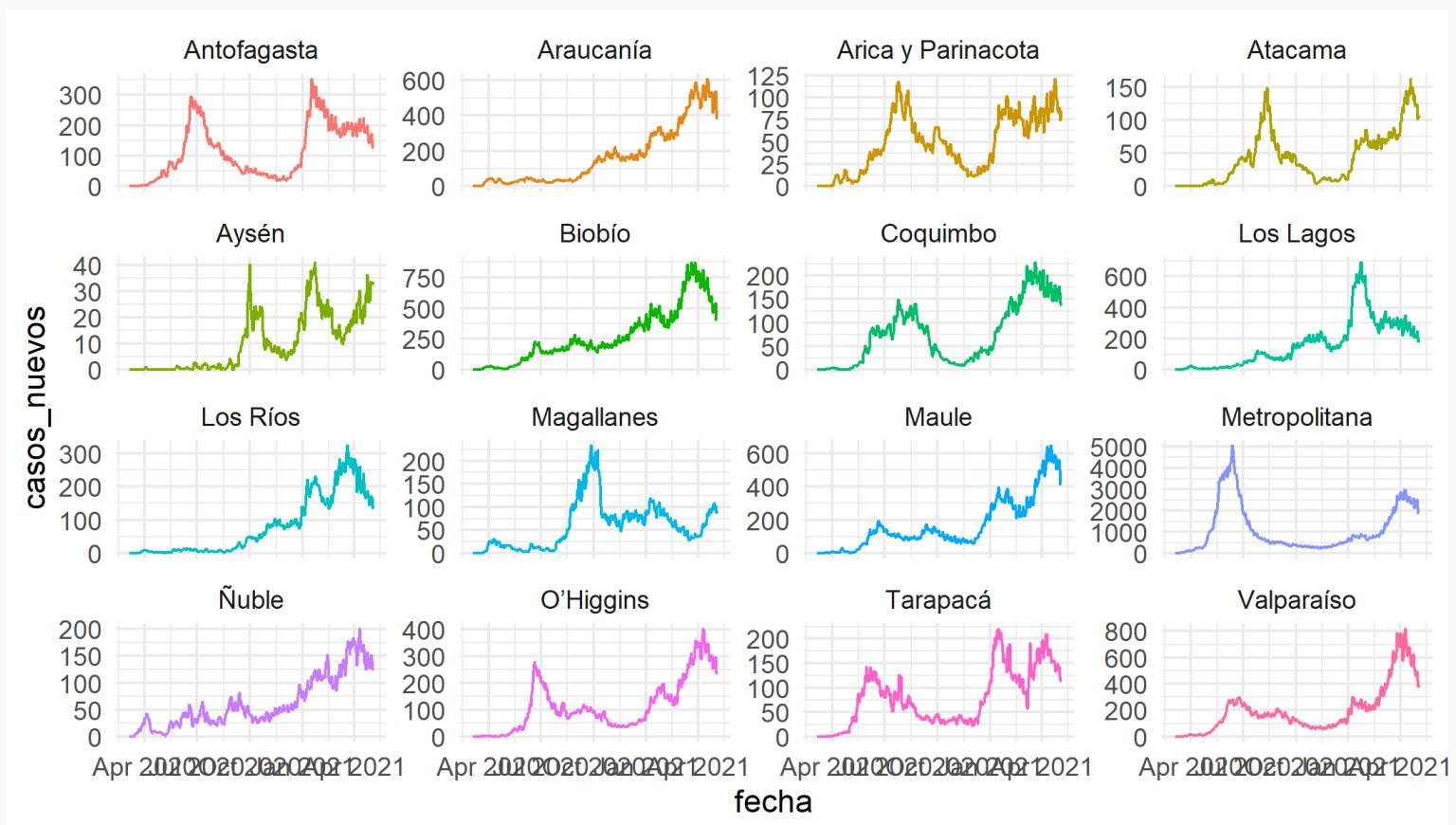
Fondo blanco parece mejor

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

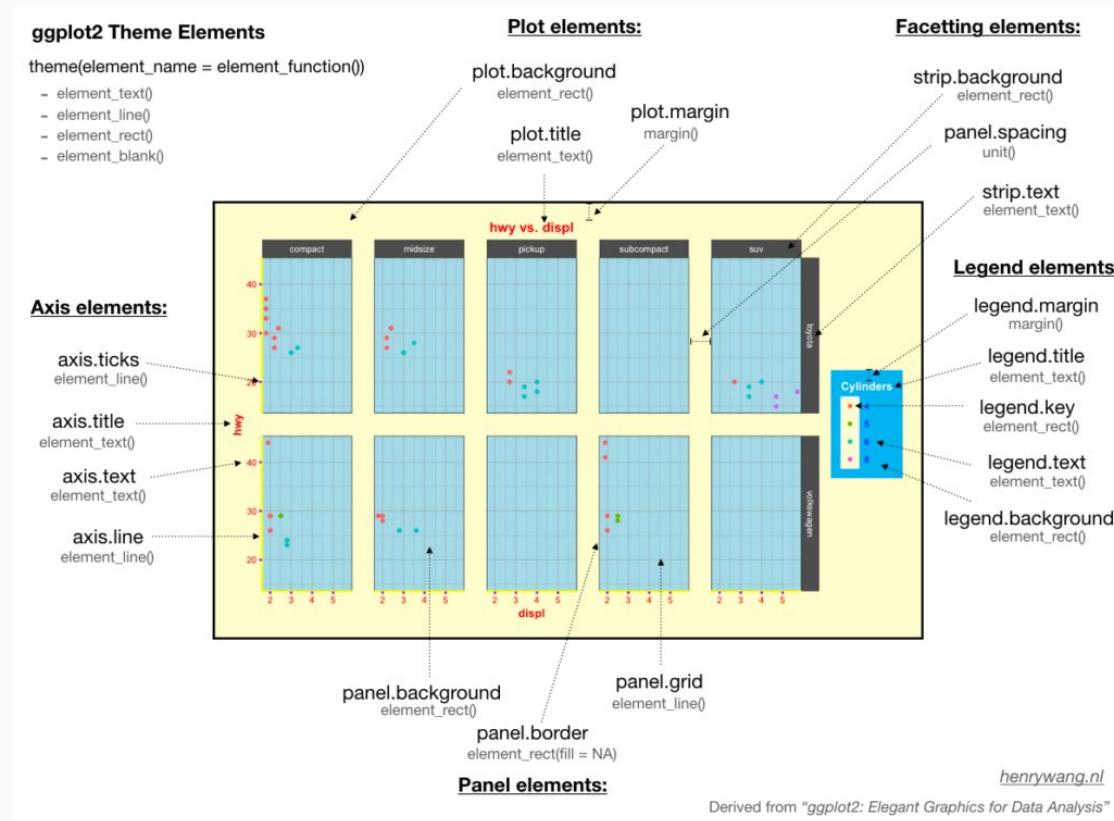


La leyenda no sirve mucho

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



Modificar theme



"Themes" prefabricados

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

Complete themes

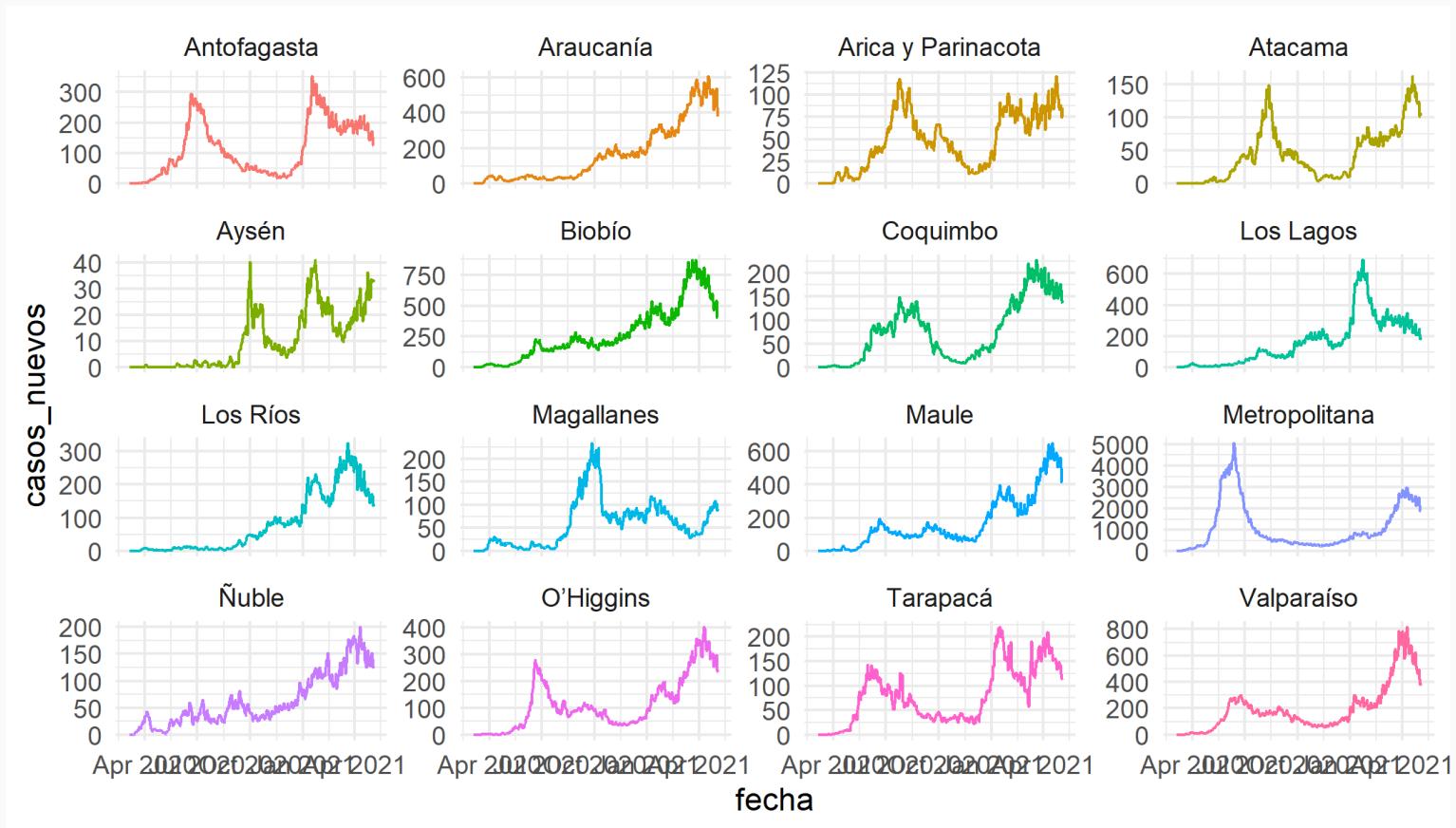
Source: R/theme-defaults.r (<https://github.com/tidyverse/ggplot2/blob/master/R/theme-defaults.r>)

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(  
  base_size = 11,
```

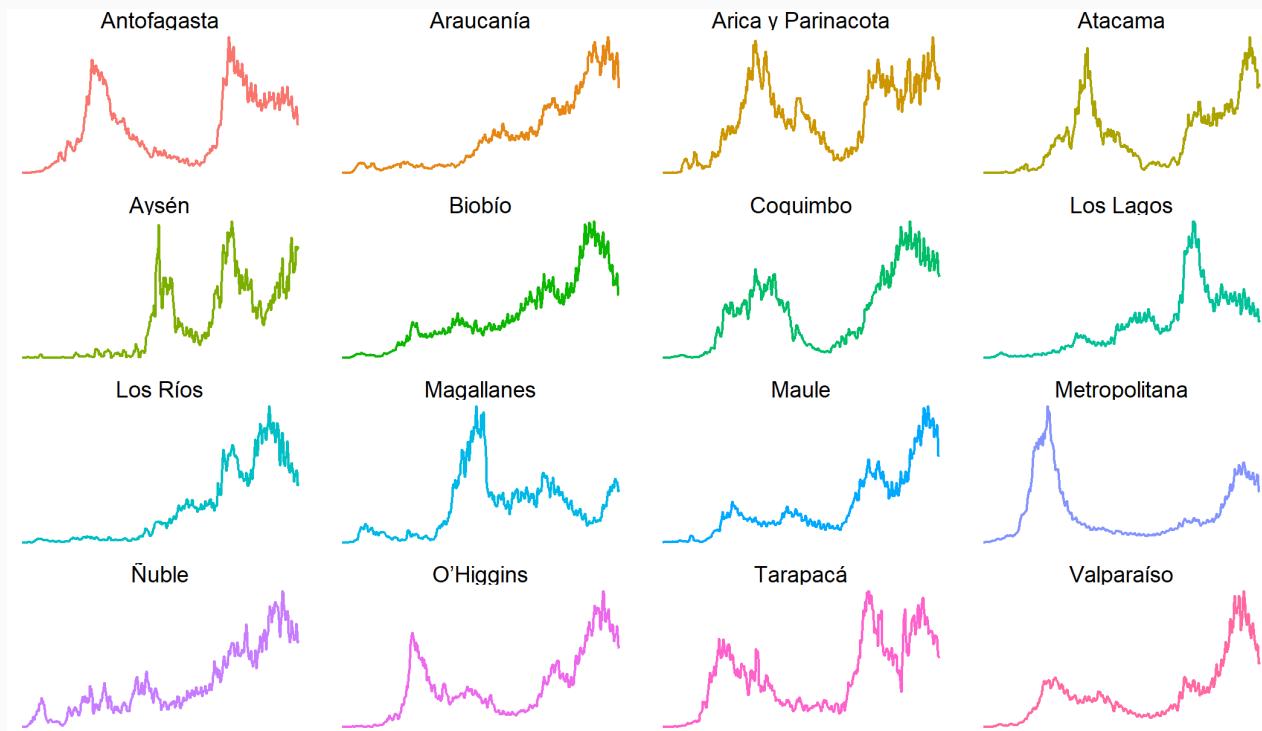
La leyenda no sirve mucho

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



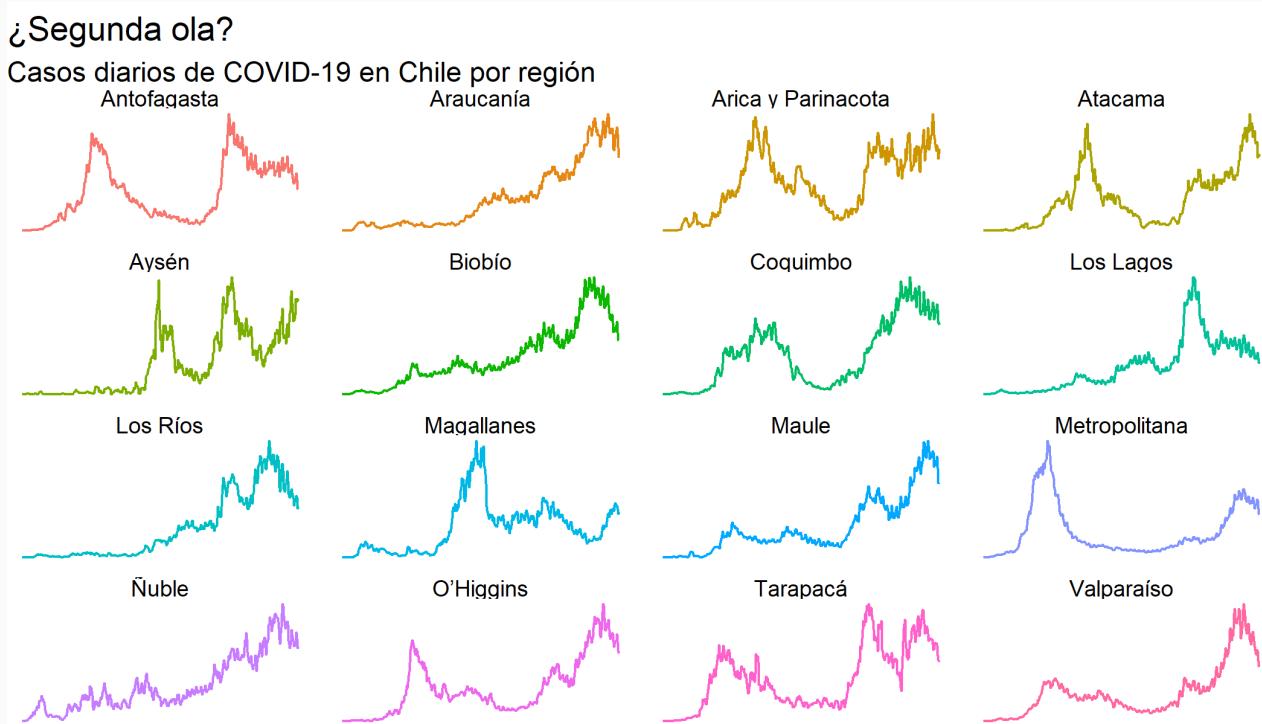
Cambiemos el theme

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y") +  
  theme_void() +  
  theme(legend.position = "none")
```



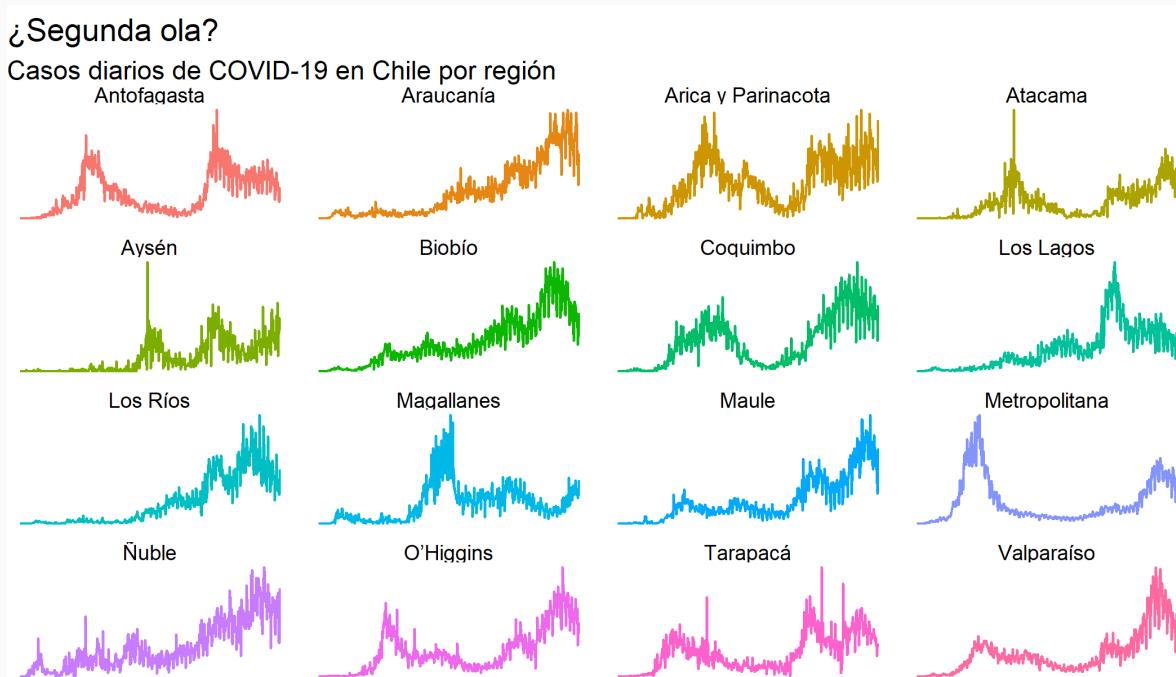
Títulos/Ejes

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y") +  
  theme_void() + theme(legend.position = "none") +  
  labs(title = "¿Segunda ola?",  
       subtitle = "Casos diarios de COVID-19 en Chile por región",  
       x = NULL, y = NULL)
```



Último detalle

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line(aes(y = casos_nuevos_d)) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y") +  
  theme_void() + theme(legend.position = "none") +  
  labs(title = "¿Segunda ola?",  
       subtitle = "Casos diarios de COVID-19 en Chile por región",  
       x = NULL, y = NULL)
```



Último detalle

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line(aes(y = casos_nuevos_d), alpha = 0.3) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y") +  
  theme_void() + theme(legend.position = "none") +  
  labs(title = "¿Segunda ola?",  
       subtitle = "Casos diarios de COVID-19 en Chile por región",  
       x = NULL, y = NULL)
```

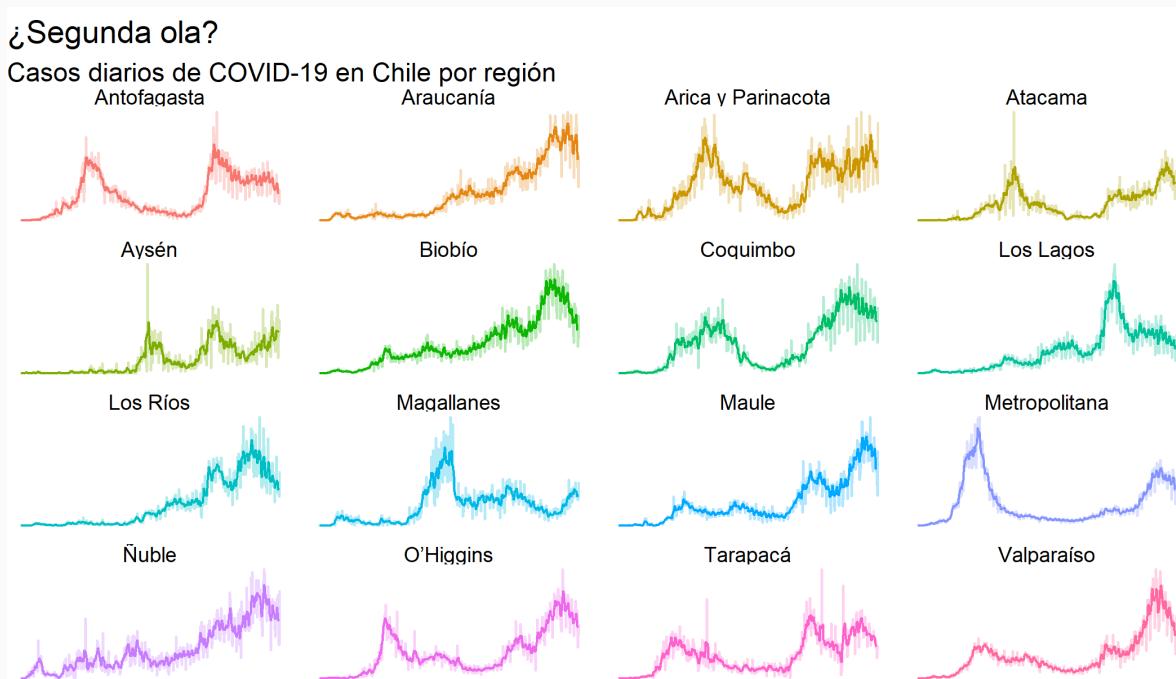
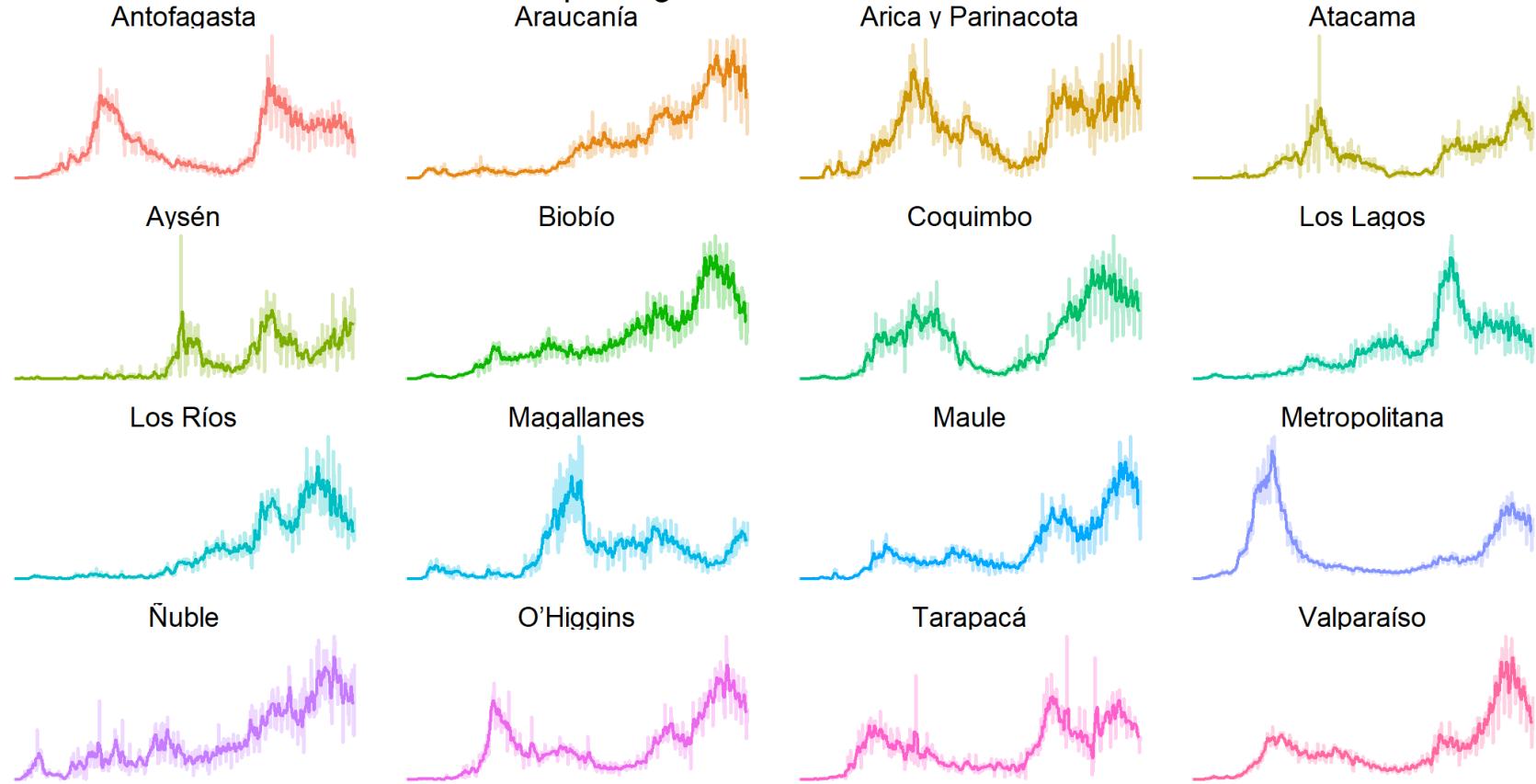


Gráfico final

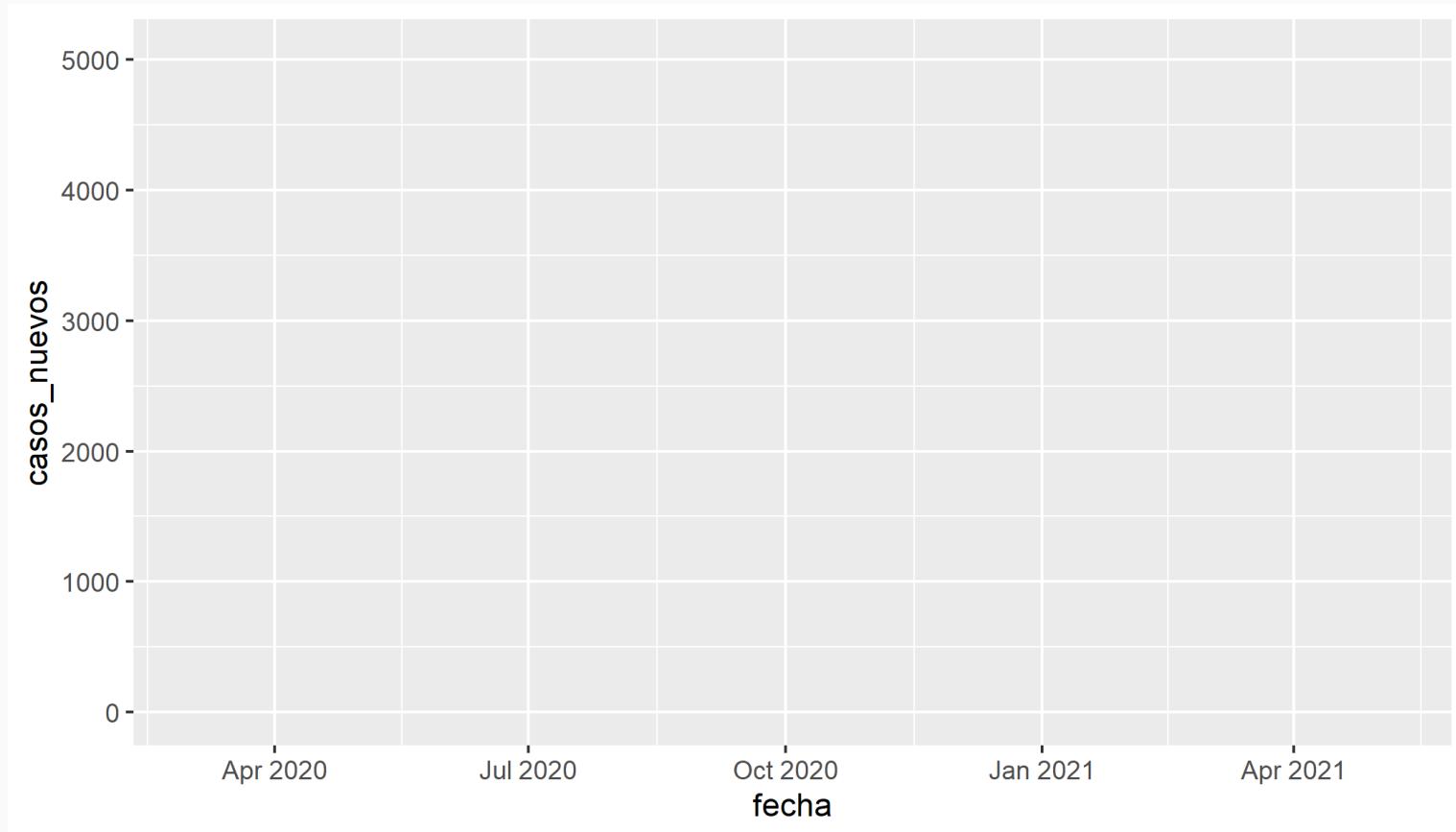
¿Segunda ola?

Casos diarios de COVID-19 en Chile por región



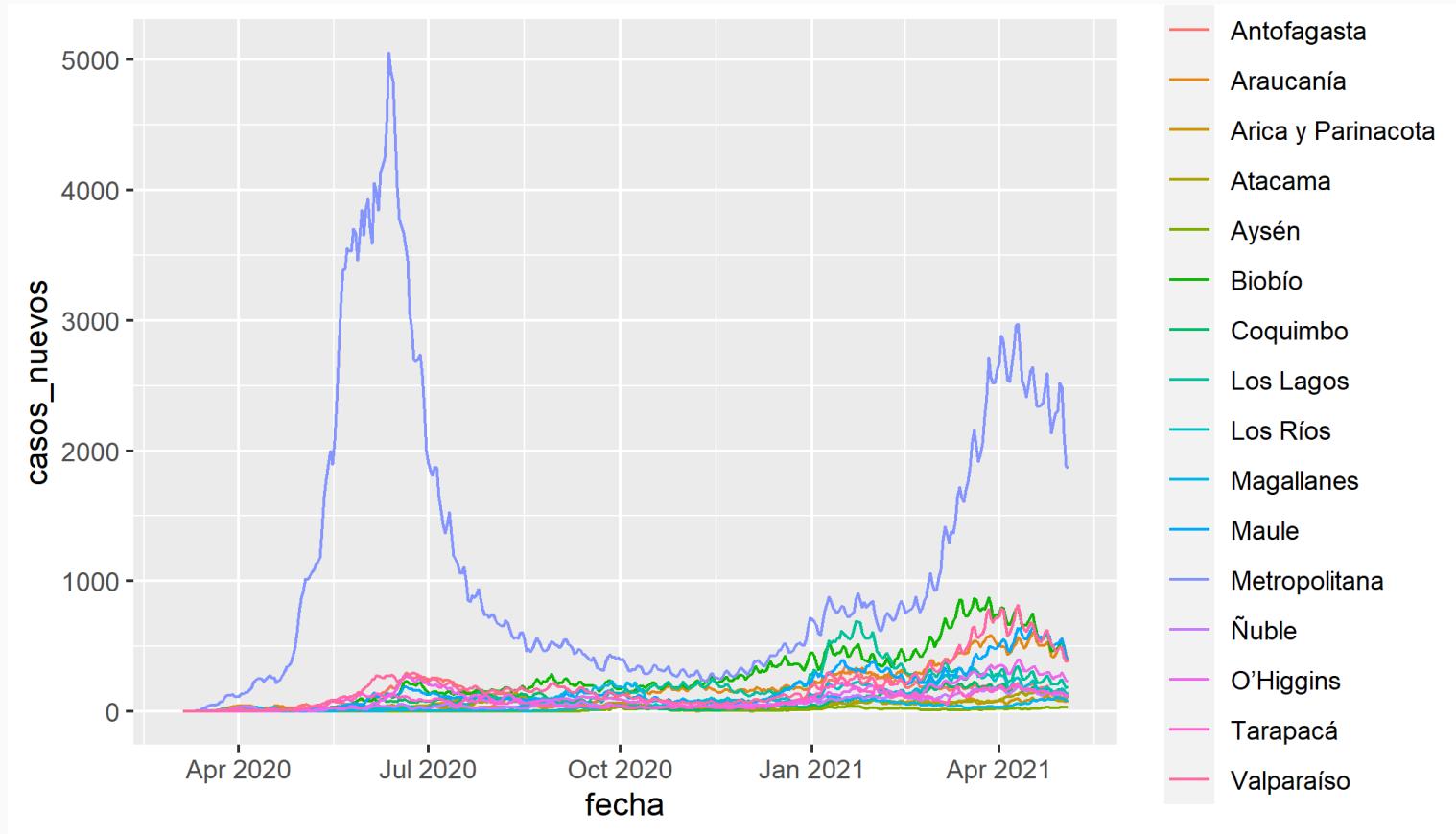
Resumen

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos))
```



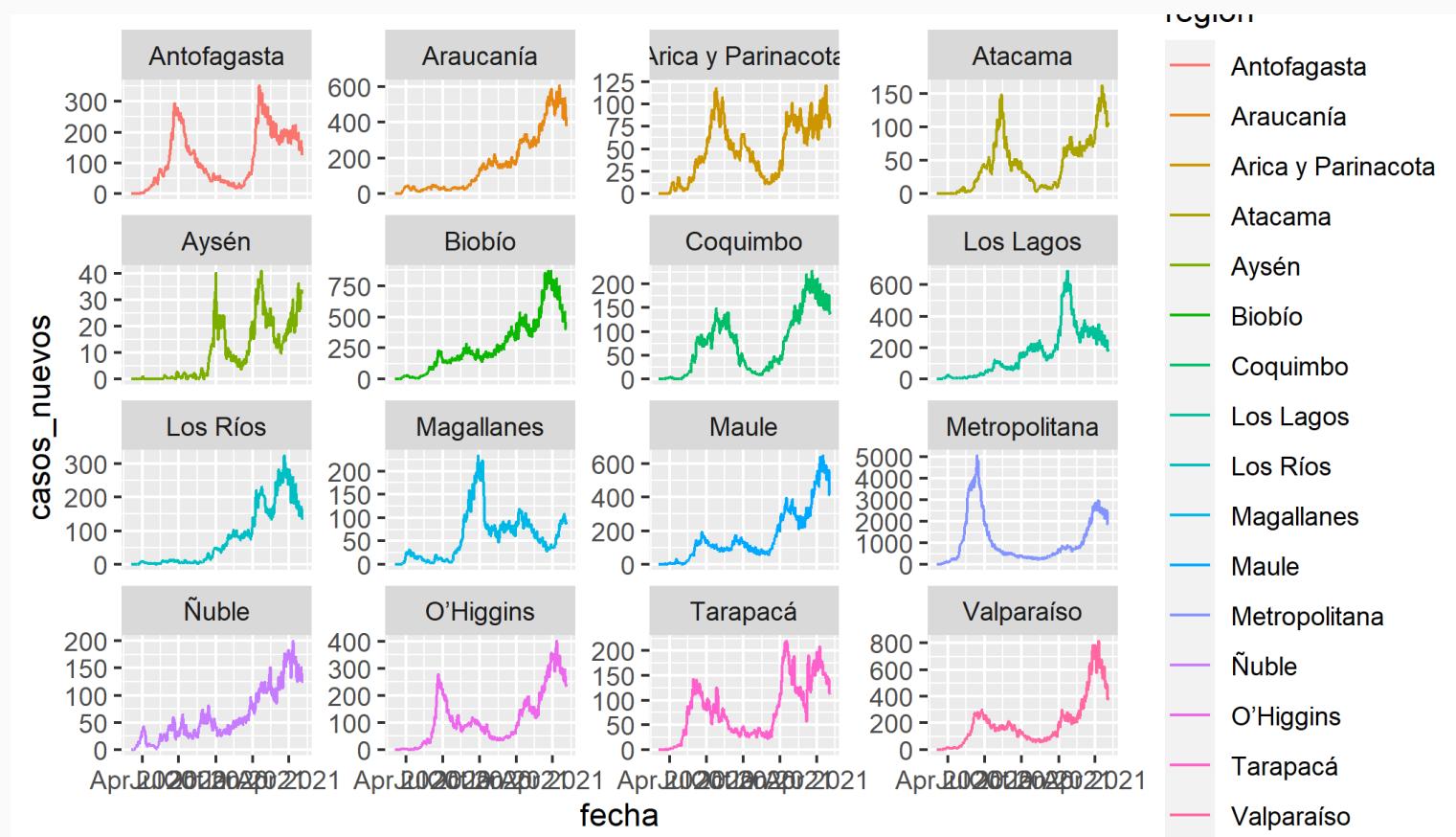
Resumen

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



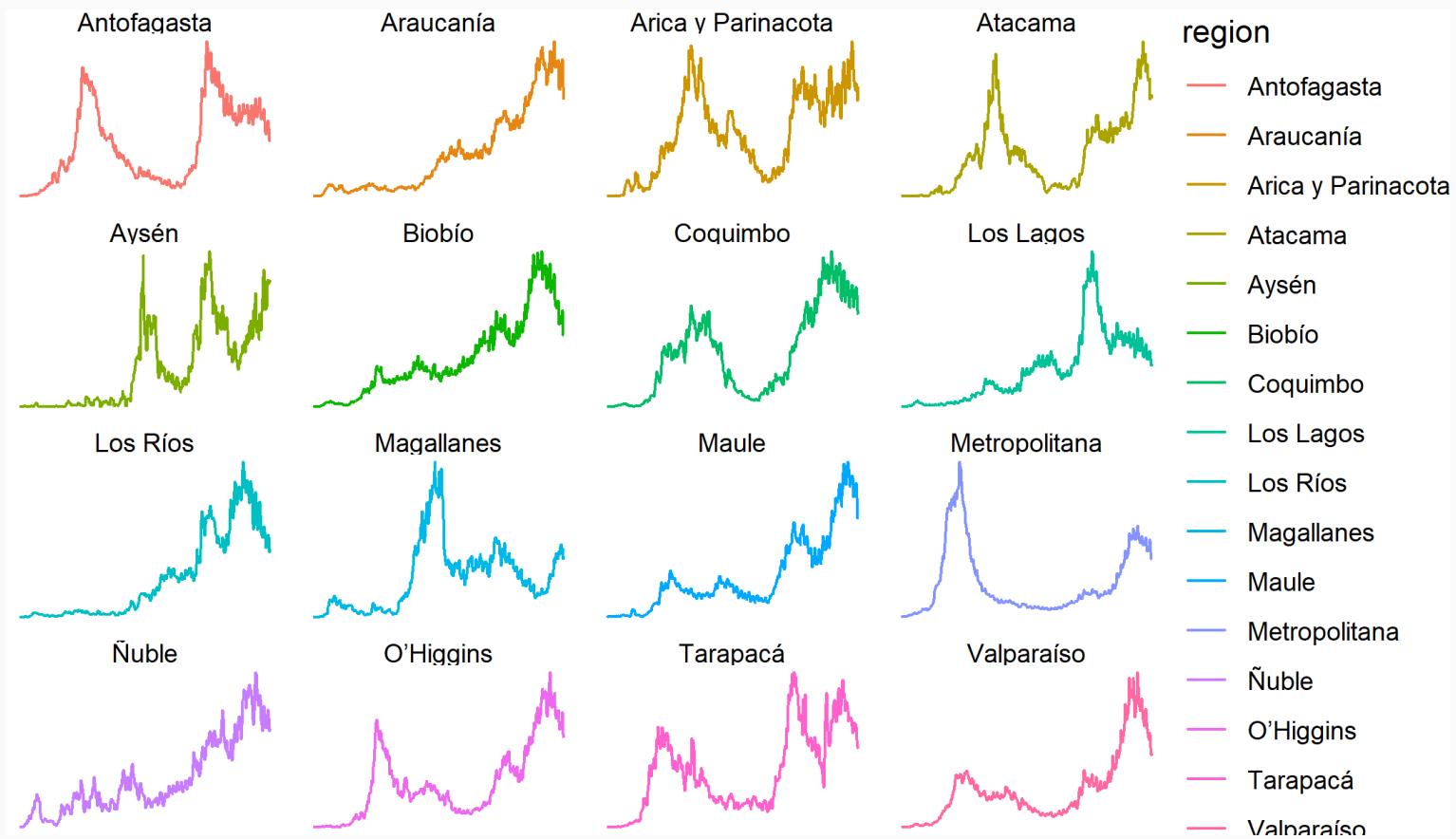
Resumen

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



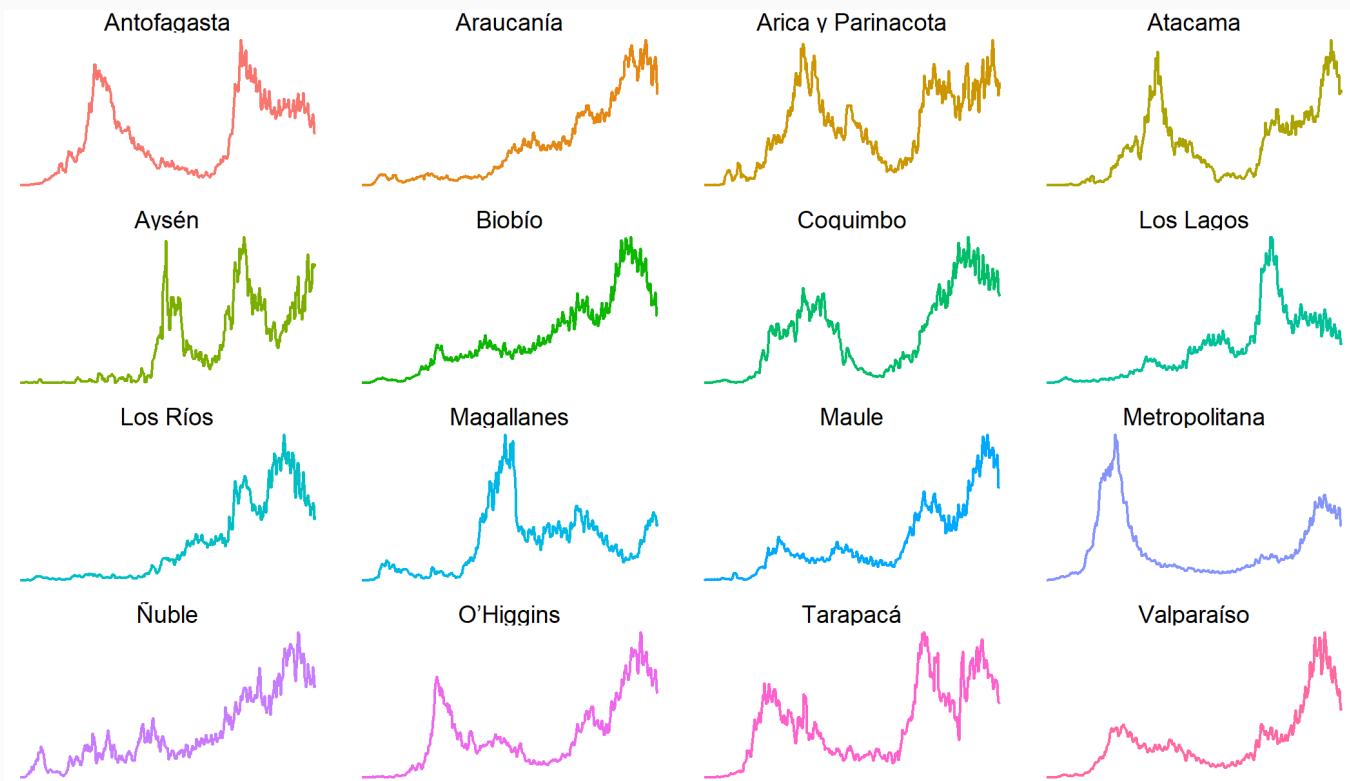
Resumen

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



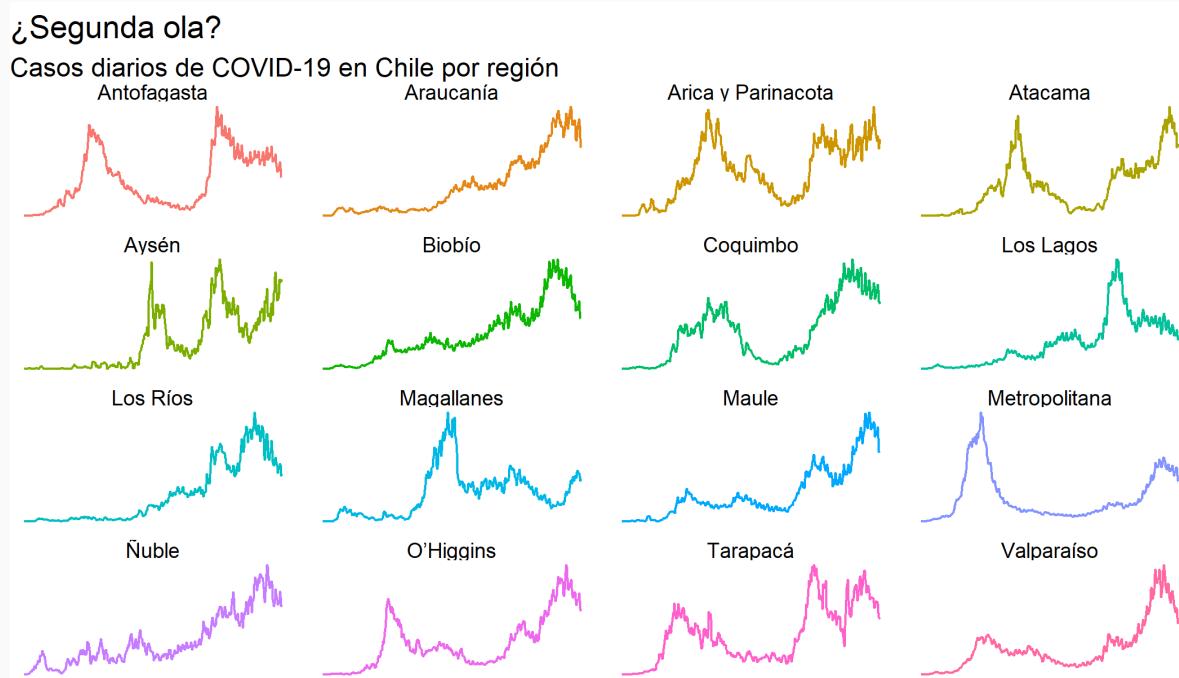
Resumen

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y") +  
  theme_void() +  
  theme(legend.position = "none")
```



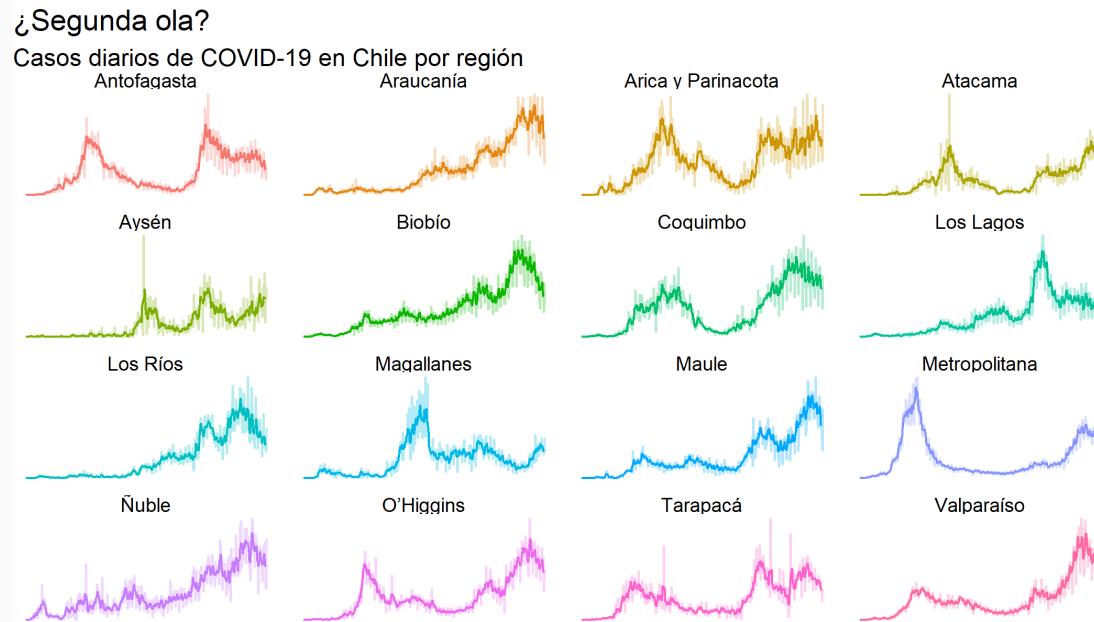
Resumen

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y") +  
  theme_void() +  
  theme(legend.position = "none") +  
  labs(title = "¿Segunda ola?",  
       subtitle = "Casos diarios de COVID-19 en Chile por región", x = NULL, y = NULL)
```



Resumen

```
ggplot(datos_covid, aes(x = fecha, y = casos_nuevos, col = region)) +  
  geom_line(aes(y = casos_nuevos_d, col = region), alpha = 0.3) +  
  geom_line() +  
  facet_wrap(vars(region), scales = "free_y") +  
  theme_void() +  
  theme(legend.position = "none") +  
  labs(title = "¿Segunda ola?",  
       subtitle = "Casos diarios de COVID-19 en Chile por región", x = NULL, y = NULL)
```



Ejercicio

Ejercicio - Script

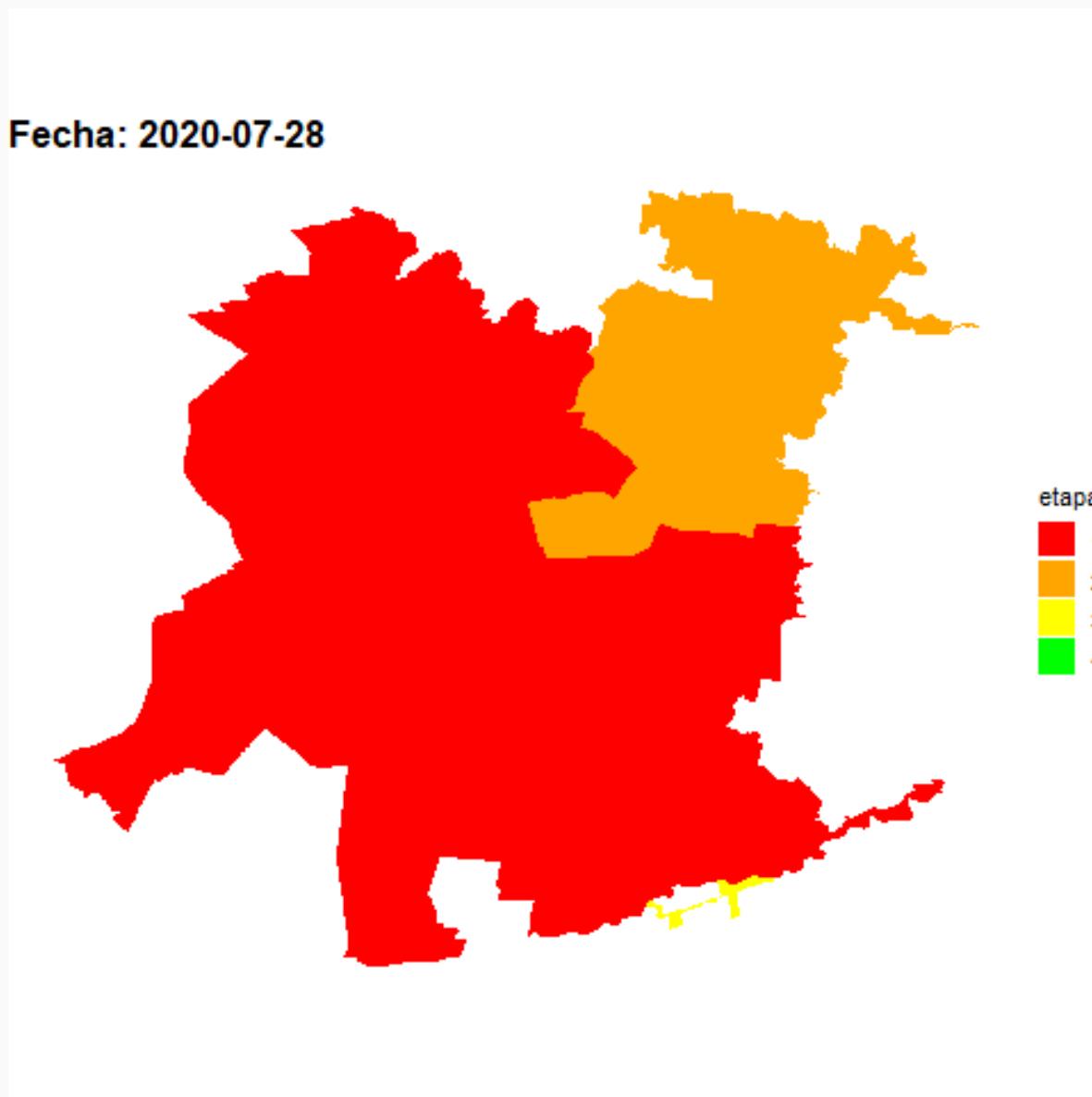
- EjercicioVizIII.R

Muchas más posibilidades

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



Muchas más posibilidades



Para que revisen

- EjemplosViz.html

Cargar paquetes

Datos a utilizar

Gráficos con `iris`

Gráfico inicial

Diferenciar puntos según especie

Líneas de tendencia

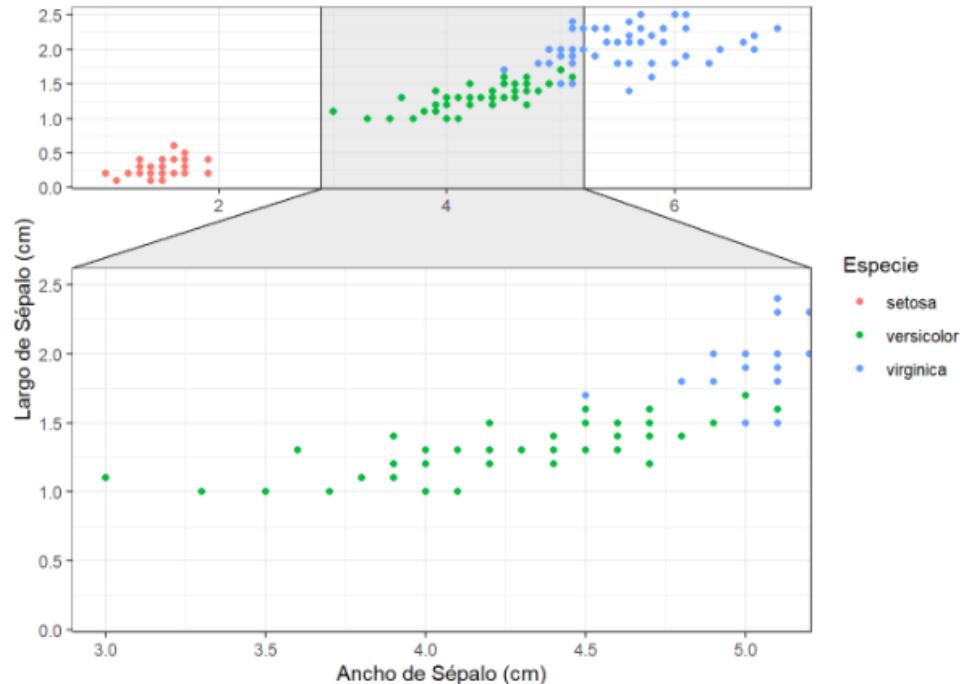
Facet

Paquetes complementarios

Gráficos con aplicaciones

```
library(ggforce)

ggplot(iris, aes(Petalo.Largo, Petalo.Ancho, colour = Especie)) +
  xlab("Ancho de Sépalo (cm)") +
  ylab("Largo de Sépalo (cm)") +
  geom_point() +
  facet_zoom(x = Especie == "versicolor") +
  theme_bw() # cambié de theme_minimal a theme_bw para destacar
```



Siguiente clase

- Manejo de datos

