



Taller R - Rifa Valentina

Visualización con ggplot2

Nicolás Ratto

2022/02/02

Contenidos

- Manipulación avanzada de data frames (funciones `pivot` y combinación de data frames)
- Manipulación de fechas: paquete `lubridate`
- Visualización de datos con `ggplot2`
- Estimar desde diseños muestrales complejos (paquete `survey` y `srvyr`)
- Hacer funciones, procesos iterativos e introducción a paquetes en R

Visualización de datos

Su relevancia, malos ejemplos y recomendaciones

La importancia de la visualización

La visualización juega un rol importante en las etapas del análisis de datos:

- Exploración
- Modelamiento
- Comunicación

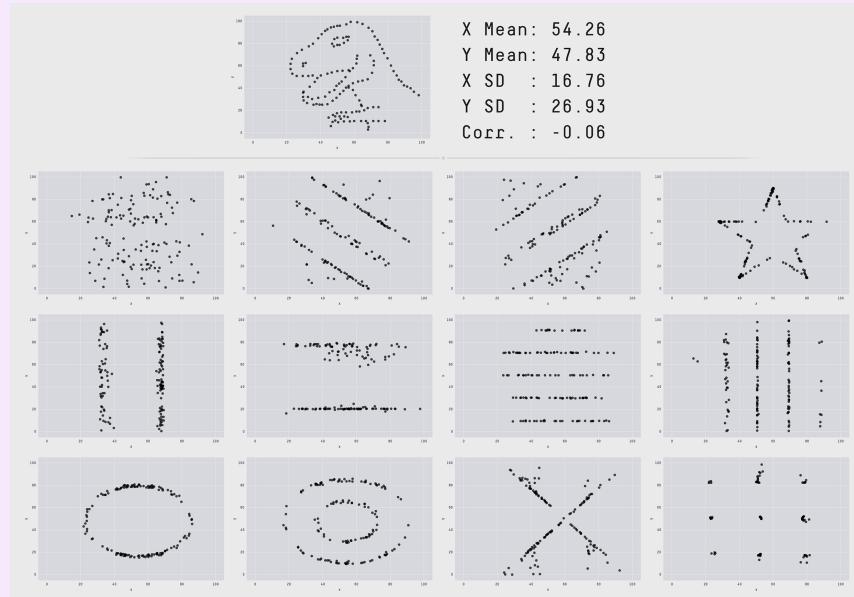
El tipo de gráfico a elaborar se desprende naturalmente de estos objetivos.

La visualización permite

- Descubrir relaciones donde no creíamos que habían (o a la inversa)
- comparar estimaciones y determinar si existen diferencias significativas
- Comunicar y atraer la atención de una audiencia.

La importancia de la visualización

Mismos estadísticos, diferentes diagramas



Matejka & Fitzmaurice, 2017

¿Qué hace a un mal gráfico?

Mal gusto

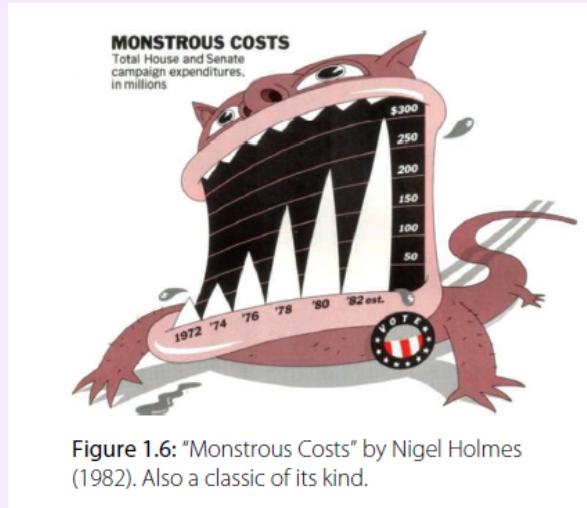
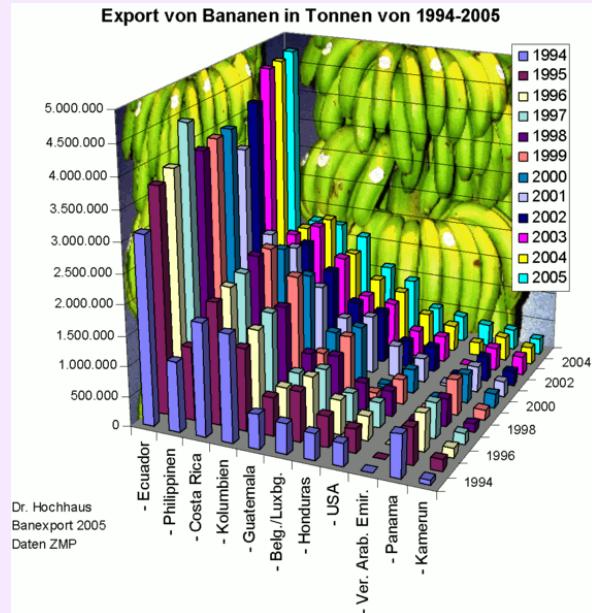
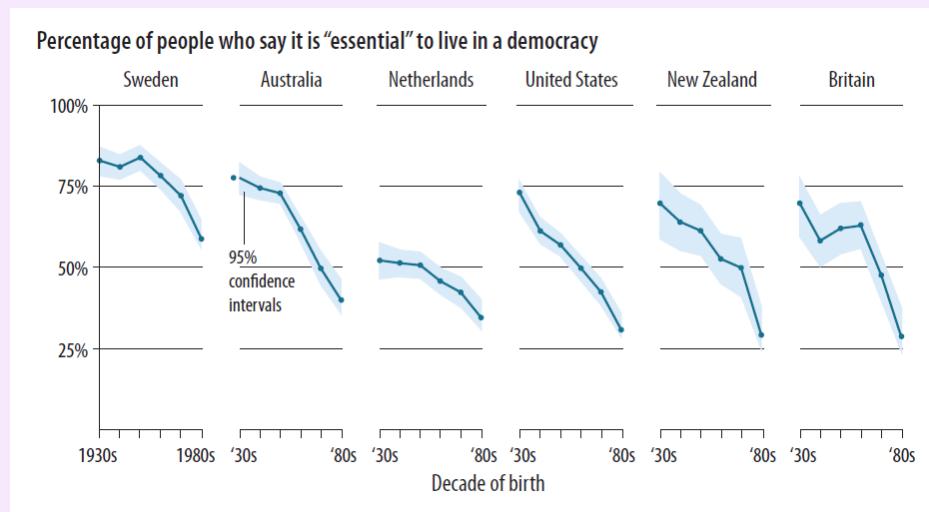


Figure 1.6: "Monstrous Costs" by Nigel Holmes (1982). Also a classic of its kind.



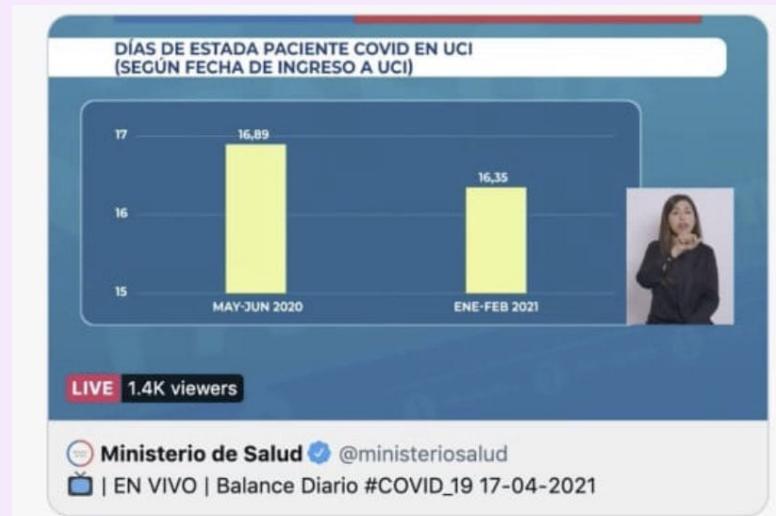
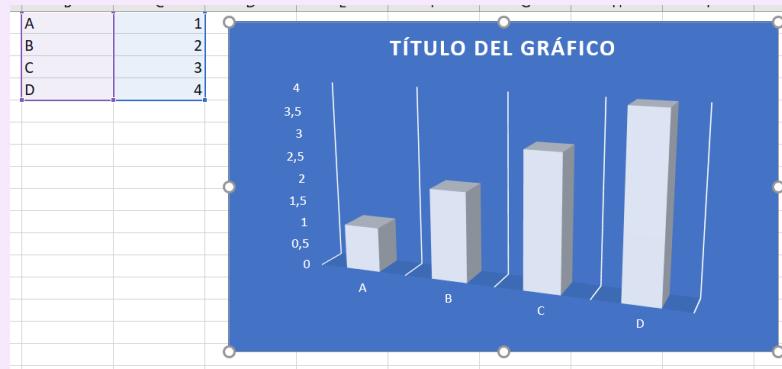
¿Qué hace a un mal gráfico?

Mal manejo de los datos



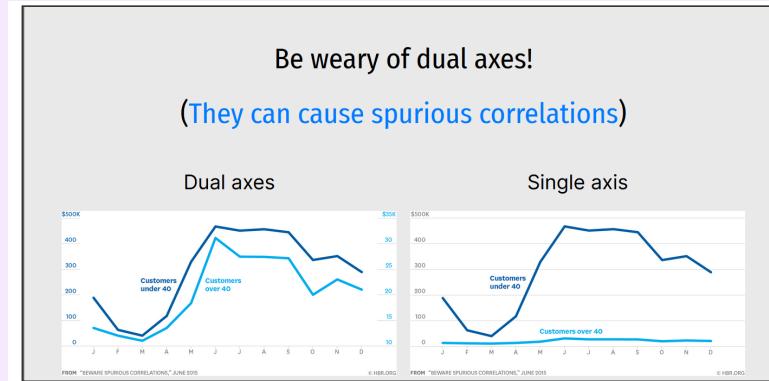
¿Qué hace a un mal gráfico?

Mala percepción



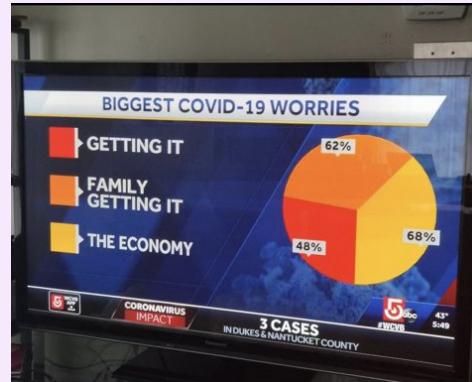
¿Qué hace a un mal gráfico?

Mala percepción

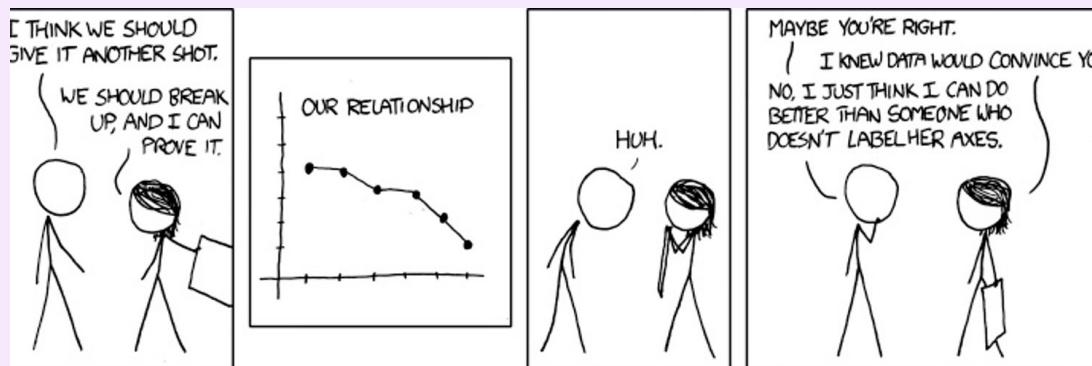


¿Qué hace a un mal gráfico?

¿El problema?

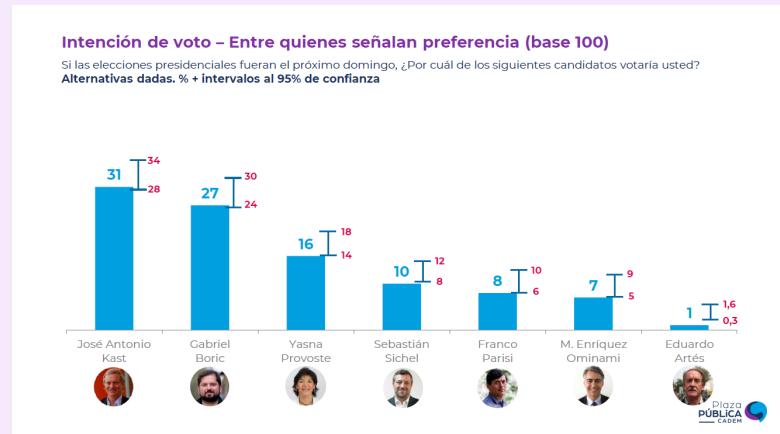


Otros problemas: no colocar etiquetas en los ejes (o no colocar leyendas)



¿Qué hace a un mal gráfico?

Ignorar el gráfico



CADEM RESEARCH & ESTRATEGIA

ESTUDIOS PLAZA PÚBLICA CHILE QUE VIENE MARCAS CIUDADANAS INSIDE CHILE CADEM ONLINE

ÚLTIMOS ESTUDIOS

PLAZA PÚBLICA | 24 OCTUBRE 2021
Kast sube 2pts a 23% y se consolida en el primer lugar, seguido por Boric que se mantiene en 20%
En tercer lugar continúa Yasna Provoste con 12%, seguida de Sebastián Sichel (7%) y Franco Parisi (6%).

PLAZA PÚBLICA | 18 OCTUBRE 2021
Kast pasa a liderar la carrera presidencial en empate estadístico con Boric que gana en la prensa

PRENSA | 20 OCTUBRE 2021
La Segunda | Boric y Kast se acercan a la segunda vuelta a un mes de las elecciones

https://github.com

Consejos sobre visualización

Las mejores visualizaciones son aquellas que requieren el uso de la "*visión instantánea*", que no requieren de un esfuerzo visual para ser comprendidas.

Los elementos del gráfico varían en la dificultad de estimar variables cuantitativas. Elementos como la posición a lo largo de una escala son más sencillos de percibir que el área de una figura o el tono del color ([Cleveland & McGuill, 1985](#)).

Todo lo visualizado nos debe servir para explicar algo pertinente (evita confundir con demasiada información)

Cuidado con incluir muchos atributos (posición, color, tamaño, textura, forma)

No usar gráficos de torta (algunos/as dicen que son [solo para comer](#))

Para comparar más de un gráfico utiliza las mismas escalas (y los mismos límites en los ejes)

Para ver activamente malos gráficos: [Graph Crimes](#)

ggplot2

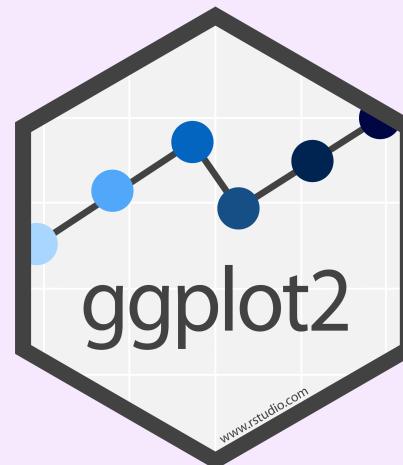
Introducción a su gramática y algunos consejos

ggplot2

Es una de las muchas formas de hacer gráficos en R.

Es parte del conjunto de paquete tidyverse.

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



La visualización implica representar datos usando líneas, formas, colores y otras cosas más.

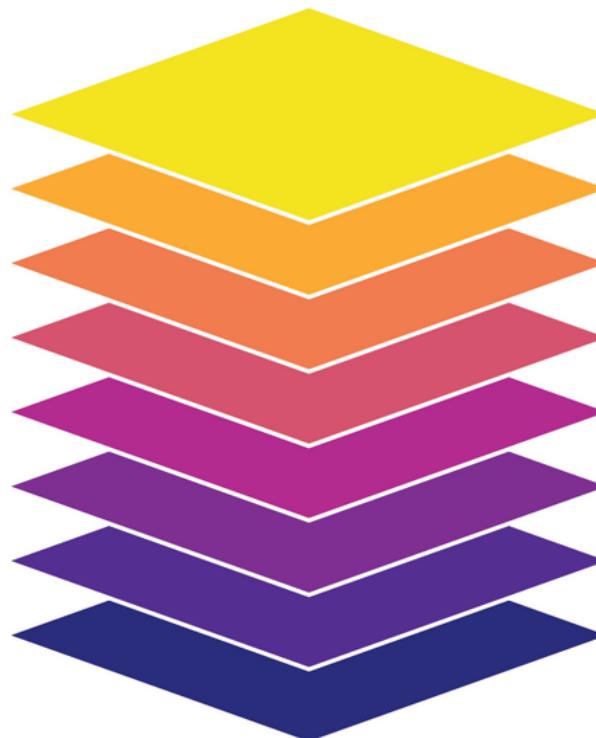
El uso de estos signos depende de lo que vamos representar (evitar sobrecargar).

La gramática de ggplot funciona por "capas".

Las capas dentro de la función ggplot se concatenan con signos +

ggplot2

Theme
Labels
Coordinates
Facets
Scales
Geometries
Aesthetics
Data



ggplot2

La gramática de ggplot

Memorizar las funciones es una mala estrategia para mejorar en ggplot.

Mejor ocupar el tiempo en comprender la lógica, hay mucho material en esta presentación y en internet para consultar.

A lo menos, debemos considerar 4 elementos para generar un gráfico.

- Lo primero y fundamental: la dataframe que queremos graficar.

```
data
```

- Lo segundo, llamar a la función `ggplot()` del paquete `ggplot2`.

```
data %>% ggplot()
```

ggplot2

- Lo tercero, definir las variables que queremos graficar.

```
data %>% ggplot(aes(x=var1, y=var2))
```

- Lo cuarto, definir si queremos visualizar líneas, barras, cajas, puntos, etc.

```
data %>% ggplot(aes(x=var1, y=var2)) + geom_point()
```

```
data %>% ggplot(aes(x=var1, y=var2)) + geom_line()
```

```
data %>% ggplot(aes(x=var1, y=var2)) + geom_bar()
```

```
data %>% ggplot(aes(x=var1, y=var2)) + geom_boxplot()
```

Aplicación simple de ggplot2

```
library(ggplot2) ; library(dplyr) ; library(tidyr)
```

```
library(gapminder)
```

```
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country   continent   year lifeExp      pop gdpPercap
##   <fct>     <fct>     <int>   <dbl>    <int>     <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.
```

```
dim(gapminder)
```

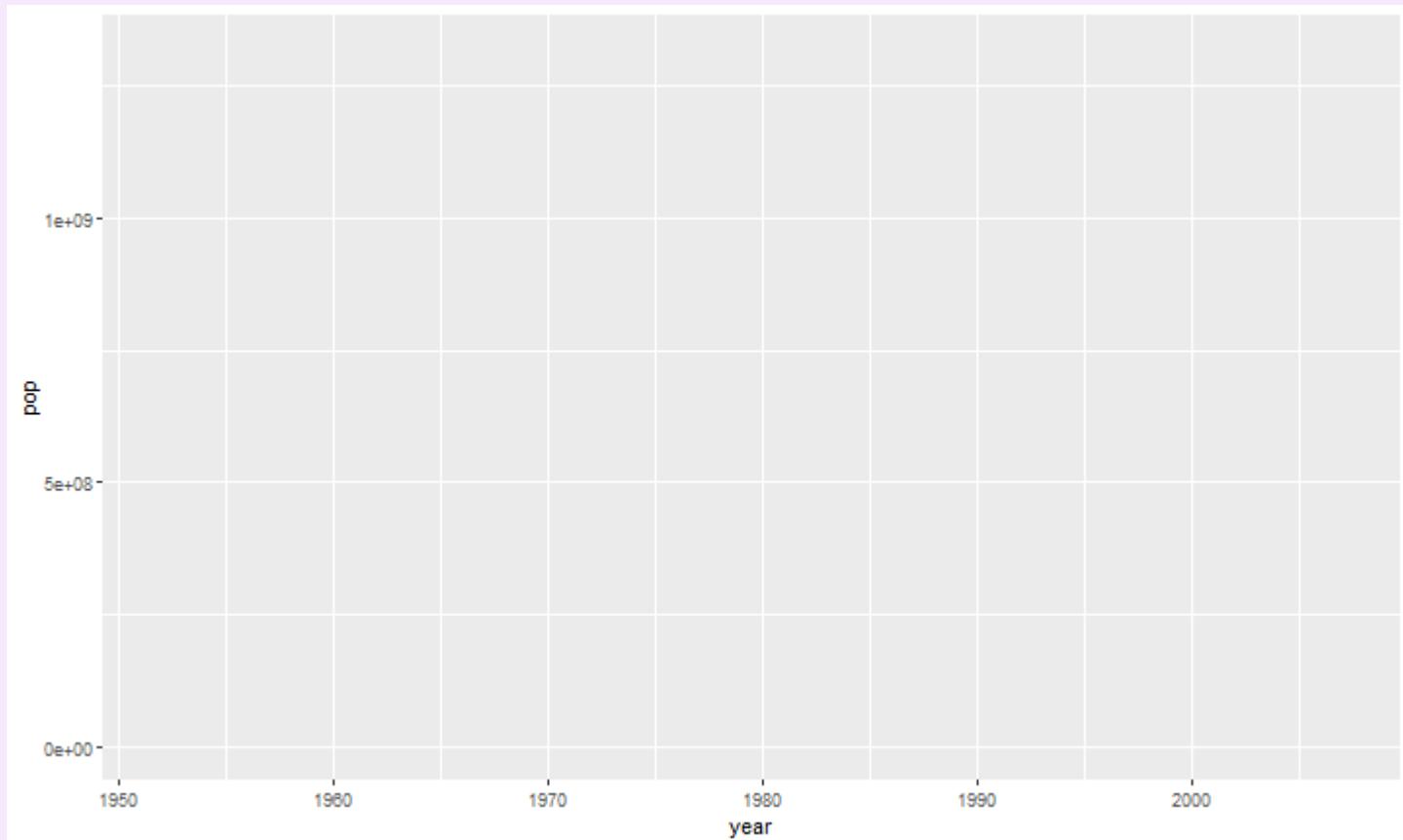
```
## [1] 1704     6
```


Aplicación simple de ggplot2

```
gapminder %>% ggplot()
```

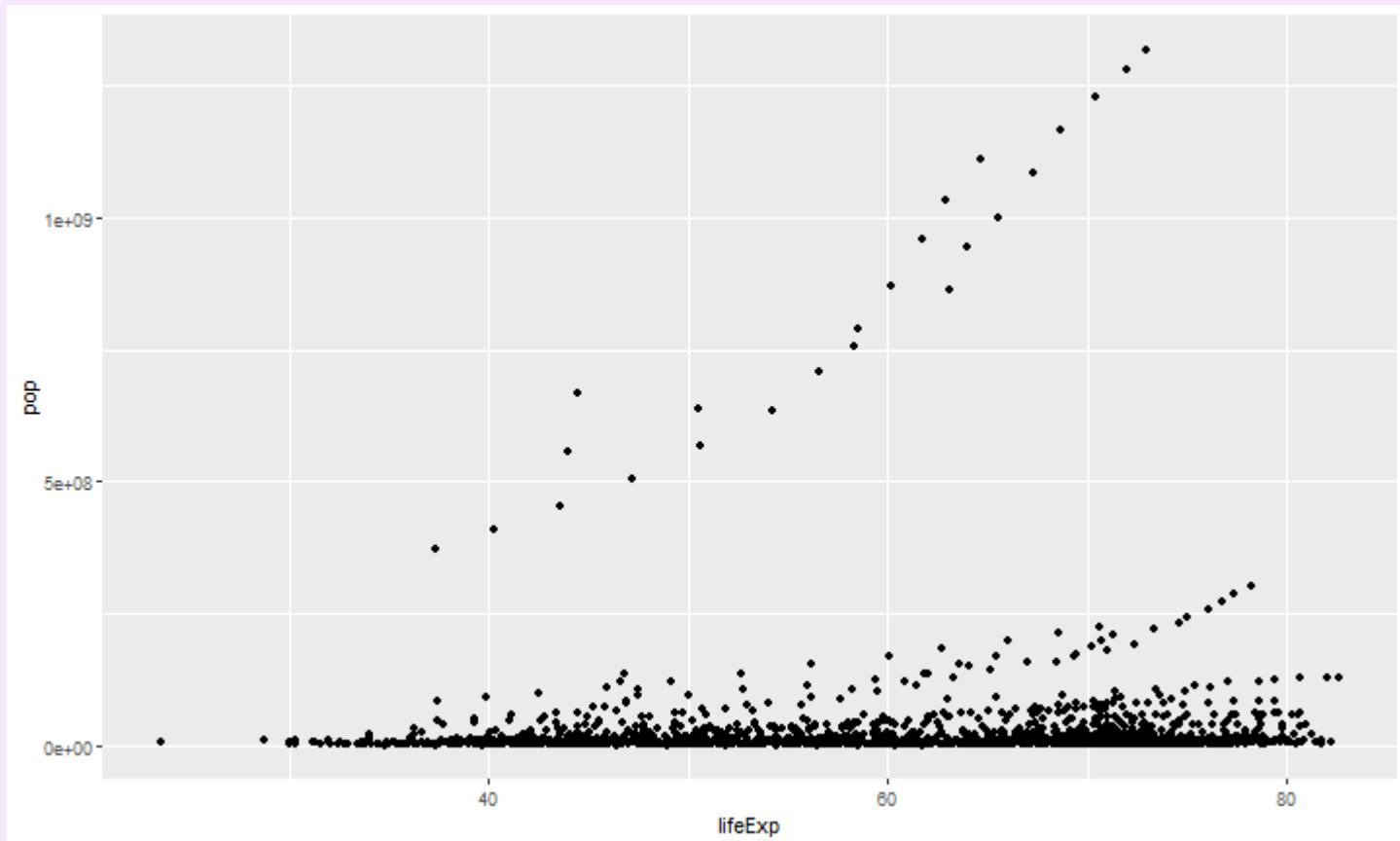
Aplicación simple de ggplot2

```
gapminder %>% ggplot(aes(x=year,y=pop))
```



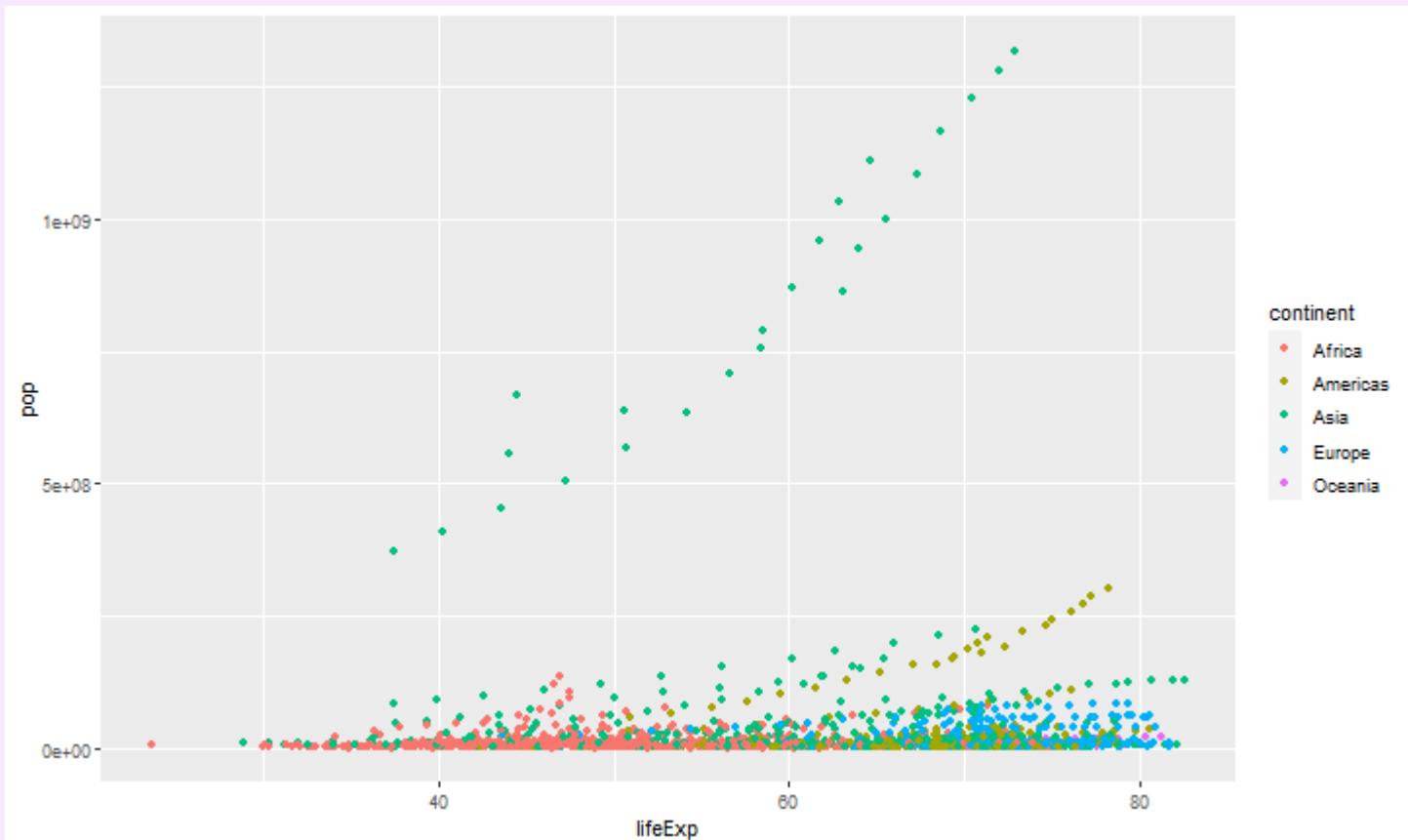
Aplicación simple de ggplot2

```
gapminder %>% ggplot(aes(x=lifeExp,y=pop)) + geom_point()
```



Agregar terceras variables

```
gapminder %>% ggplot(aes(x=lifeExp,y=pop,color=continent)) +  
  geom_point()
```



ggplot2: otras capas

Para mapear tercera, cuartas o quintas variables no ocupamos "z=".

Para llamar a estas tercera variables aplicamos alguna función o argumento que indique que cosa queremos hacer con la variable

En el caso anterior, ocupamos el argumento `color=` (colorear contornos).

También se podría ocupar el argumento `fill=` (llenar con color la figura)

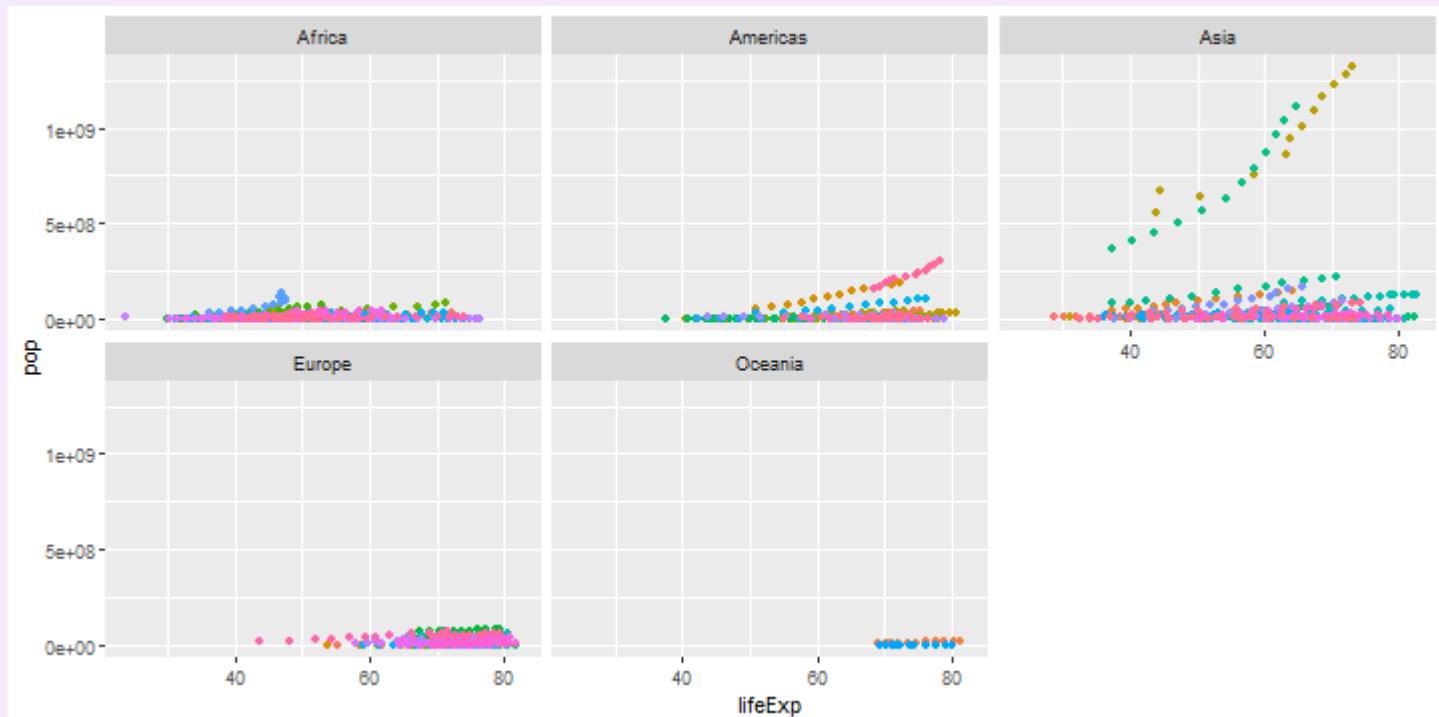
También se podría ocupar el argumento `size=` (var numéricas)

O también, la función `facet_wrap()` (dividir en paneles, var categóricas)

Hay otras muchas formas de agregar tercera variables.

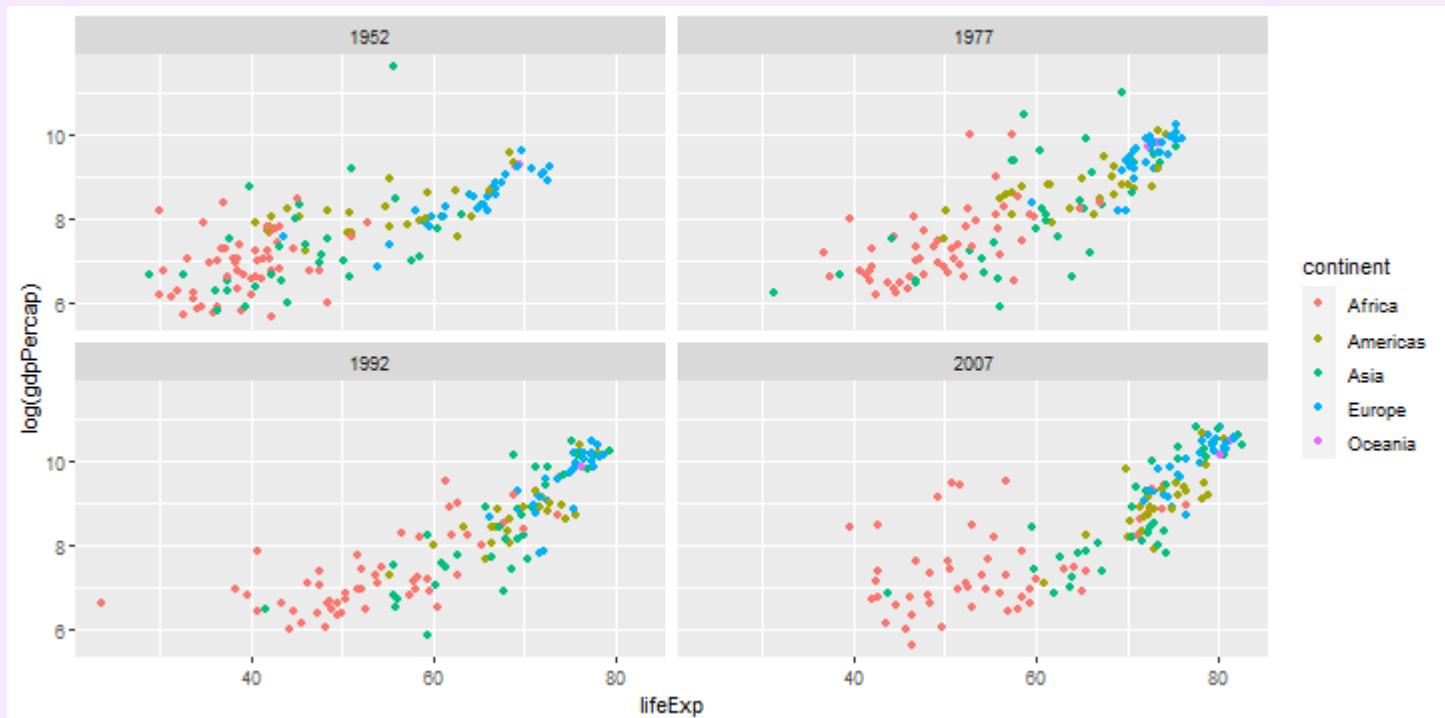
Variable para el color y paneles

```
gapminder %>% ggplot(aes(x=lifeExp,y=pop,color=country)) +  
  geom_point() +  
  facet_wrap(~continent) +  
  theme(legend.position = "none")
```



Filtrar data y luego visualizar

```
gapminder %>%  
  filter(year %in% c(1952,1977,1992,2007)) %>%  
  ggplot(aes(x=lifeExp,y=log(gdpPerCap),color=continent)) +  
  geom_point() +  
  facet_wrap(~year)
```



ggplot2. Barras

Cargar data de interés

```
data <- readRDS("data/Latinobarometro_2020_Esp_Rds_v1_0.rds")
```

¿Quién tiene más poder en Chile?

- Crear tabla

```
tabla <- data %>%
  filter(idenpa==152) %>%
  group_by(p48st_1) %>%
  summarise(n=n()) %>%
  mutate(porcentaje=n/sum(n)) %>% arrange(-n)
```

p48st_1	n	porcentaje
Las grandes empresas	537	44.75
Los partidos políticos	151	12.58
El Gobierno	131	10.92

ggplot2. Barras

- Con el dato agrupado ya podemos visualizar...

```
tabla %>%
  ggplot(aes(x=p48st_1,
             y=porcentaje)) +
  geom_bar()
```



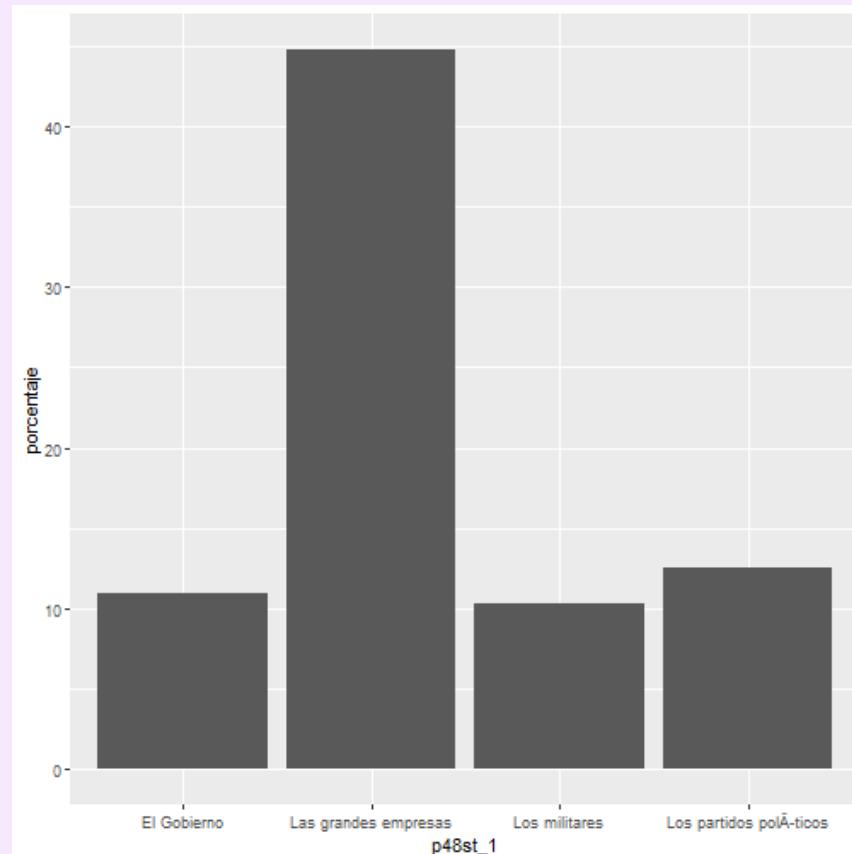
Error: stat_count() can only have an x or y aesthetic.

Por defecto `geom_bar()` visualiza una variable.

En argumento "stat" hay que señalar que queremos visualizar dos variables.

```
tabla %>%
  ggplot(aes(x=p48st_1,
             y=porcentaje)) +
  geom_bar(stat = "identity")
```

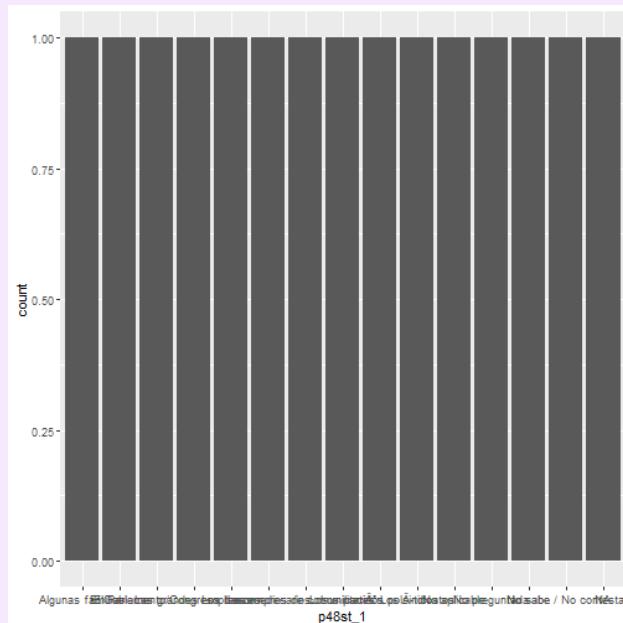
ggplot2. Barras



ggplot2. Barras

Si el argumento `stat` hay que colocar una variable y el conteo lo hace `ggplot`

```
tabla %>%  
  ggplot(aes(p48st_1)) +  
  geom_bar()
```



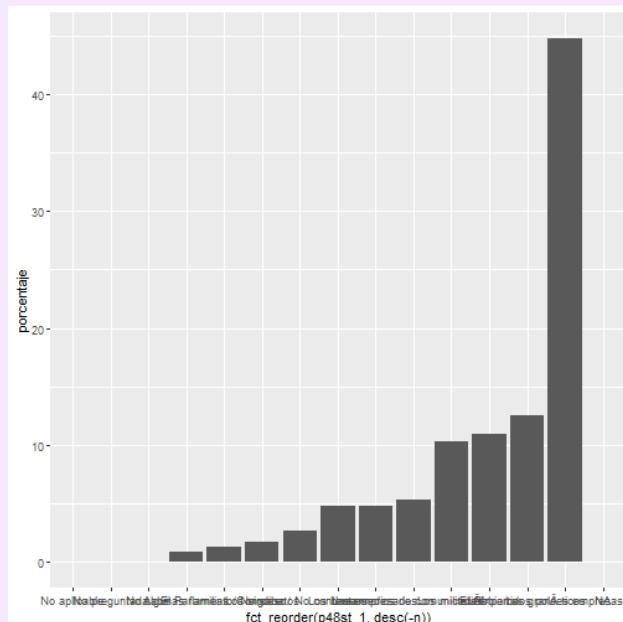
ggplot2. Barras ordendas

Volviendo a nuestro gráfico

```
tabla %>%
  ggplot(aes(x=p48st_1, y=porcentaje)) +
  geom_bar(stat = "identity")
```

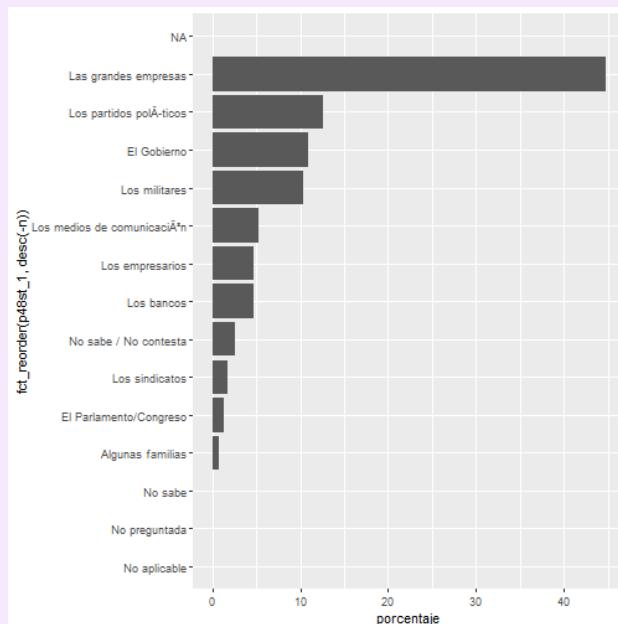
ggplot2. Barras ordendas

```
library(forcats)
tabla %>%
  ggplot(aes(x=fct_reorder(p48st_1, desc(-n)),
              y=porcentaje)) +
  geom_bar(stat = "identity")
```



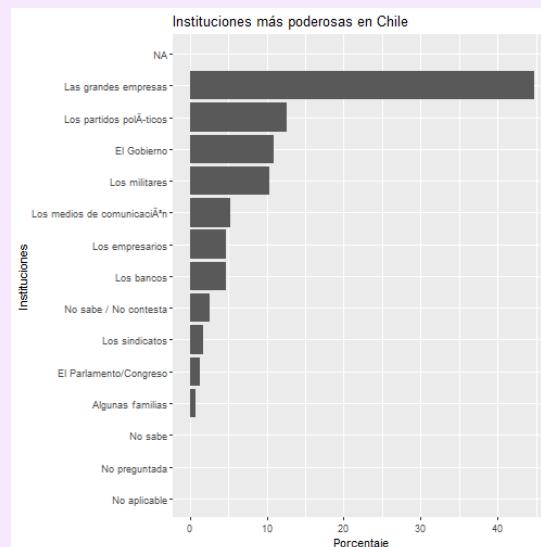
ggplot2. Invertir el gráfico

```
tabla %>%
  ggplot(aes(x=fct_reorder(p48st_1, desc(-n)),
              y=porcentaje)) +
  geom_bar(stat = "identity") +
  coord_flip()
```



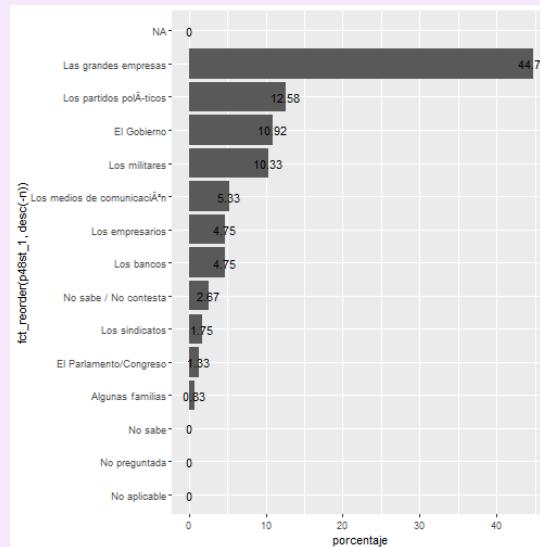
ggplot2. Agregar etiquetas

```
tabla %>%
  ggplot(aes(x=fct_reorder(p48st_1, desc(-n)),
              y=porcentaje)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(x="Instituciones",y="Porcentaje",
       title = "Instituciones más poderosas en Chile")
```



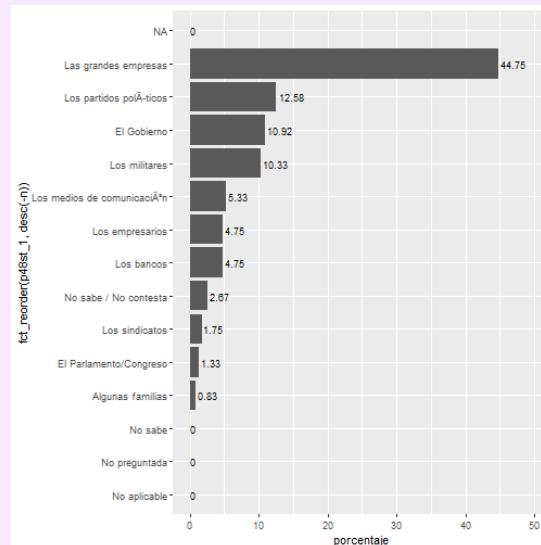
ggplot2. Agregar valores

```
tabla %>%
  ggplot(aes(x=fct_reorder(p48st_1, desc(-n)),
              y=porcentaje)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  geom_text(aes(label=porcentaje))
```



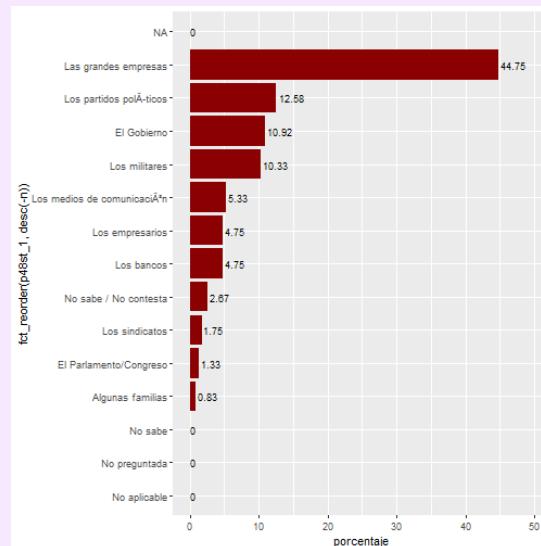
ggplot2. Definir ejes

```
tabla %>%
  ggplot(aes(x=fct_reorder(p48st_1, desc(-n)),
              y=porcentaje)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  geom_text(aes(label=porcentaje), hjust=-0.1, size=3) +
  scale_y_continuous(limits = c(0,50))
```



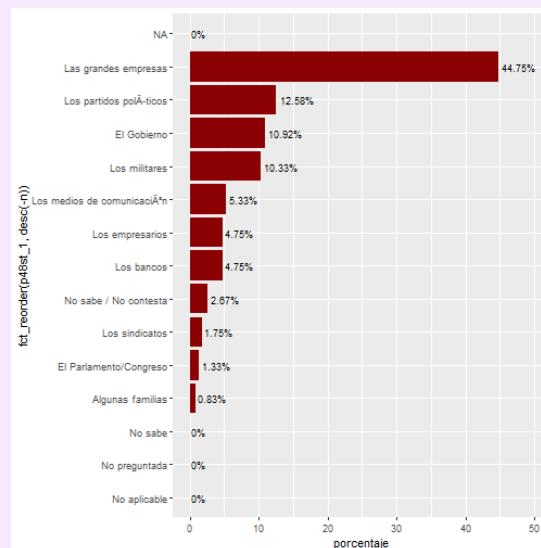
ggplot2. Tema y color

```
tabla %>%
  ggplot(aes(x=fct_reorder(p48st_1, desc(-n)),
             y=porcentaje)) +
  geom_bar(stat = "identity", fill="darkred") +
  coord_flip() +
  geom_text(aes(label=porcentaje), hjust=-0.1, size=3) +
  scale_y_continuous(limits = c(0,50))
```



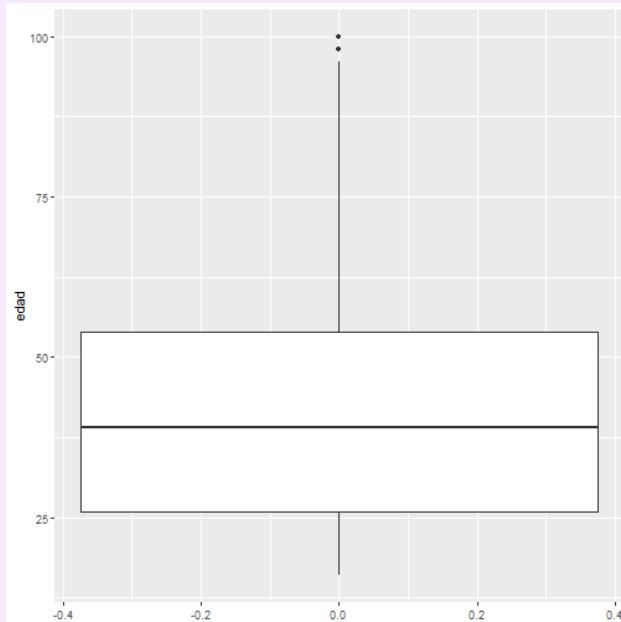
ggplot2. Signo porcentaje

```
tabla %>%
  ggplot(aes(x=fct_reorder(p48st_1, desc(-n)),
              y=porcentaje)) +
  geom_bar(stat = "identity", fill="darkred") +
  coord_flip() +
  geom_text(aes(label=paste0(porcentaje,"%")), hjust=-0.1, size=3) +
  scale_y_continuous(limits = c(0,50))
```



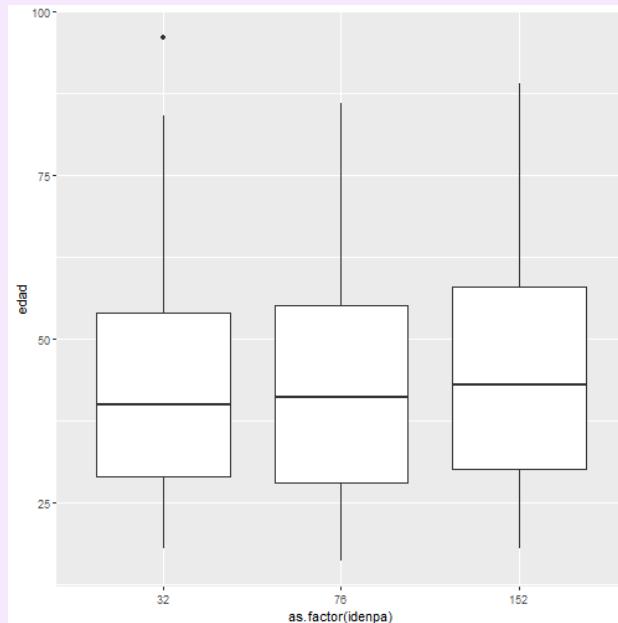
ggplot2. Graf. de cajas

```
data %>%  
  ggplot(aes(y=edad)) + geom_boxplot()
```



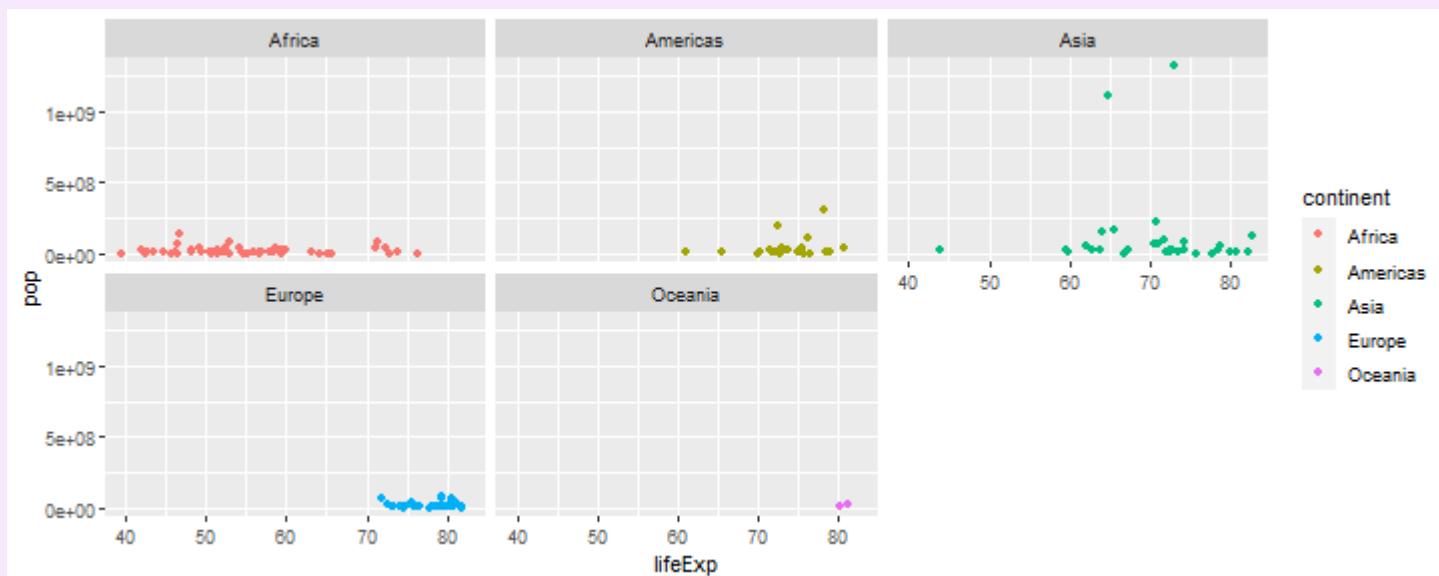
ggplot2. Graf. de cajas

```
data %>%
  filter(idenpa %in% c(152,76,32)) %>% # Chile, Brasil, Argentina
  ggplot(aes(y=edad,x=as.factor(idenpa))) +
  geom_boxplot()
```



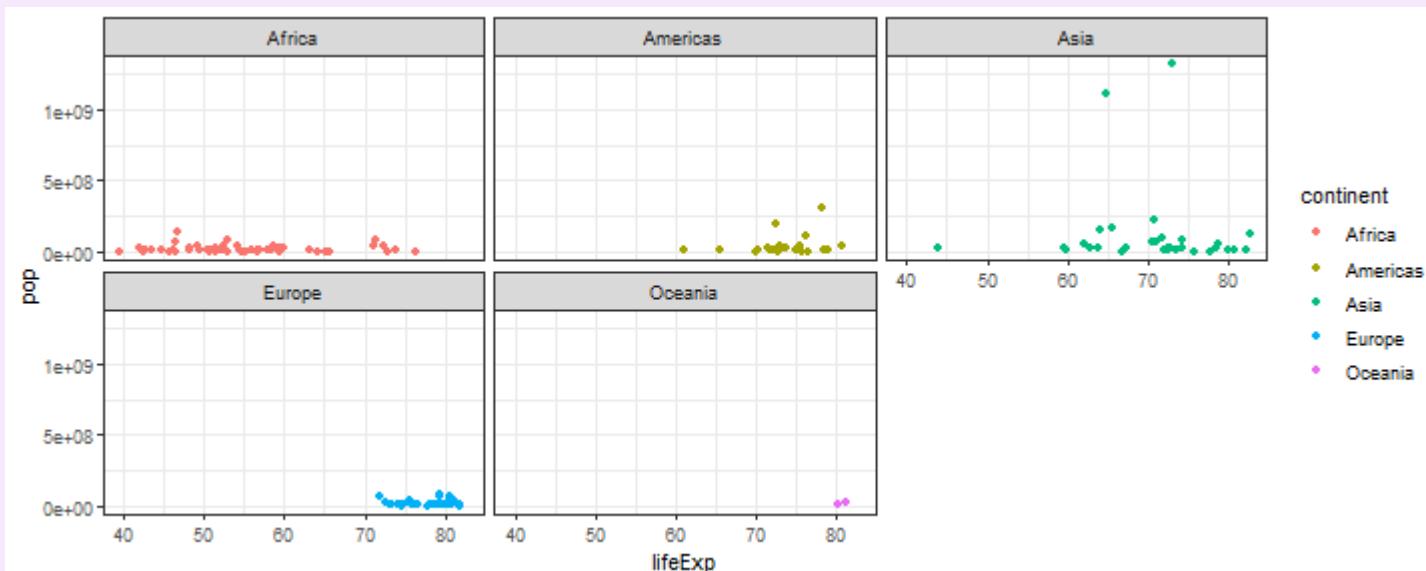
ggplot2. Otras capas

```
gapminder %>% filter(year==2007) %>%
  ggplot(aes(x=lifeExp,y=pop,color=continent)) +
  geom_point() +
  facet_wrap(~continent)
```



ggplot2. Otras capas

```
gapminder %>% filter(year==2007) %>%
  ggplot(aes(x=lifeExp,y=pop,color=continent)) +
  geom_point() +
  facet_wrap(~continent) +
  theme_bw()
```



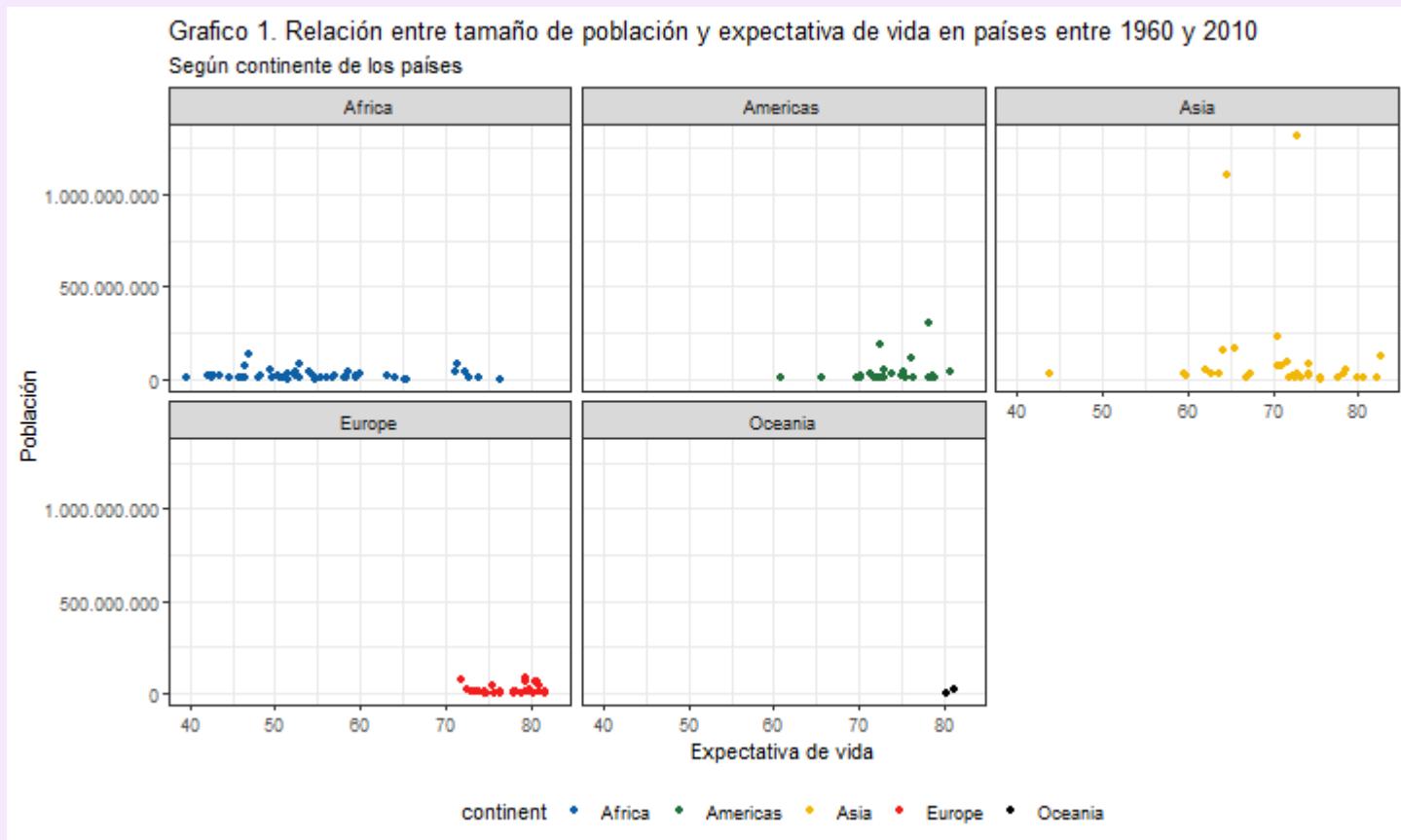
Acá [listado](#) y visualización de distintos temas.

ggplot2. Otras capas

Titulos, leyendas hacia abajo, separador de miles y colores para cada gusto

```
gapminder %>% filter(year==2007) %>%
  ggplot(aes(x=lifeExp,y=pop,color=continent)) +
  geom_point() +
  scale_color_manual(values=c( "#0D5FA6" , "#24733F" , "#F2B705" , "#F21D1D" , "black"))
  facet_wrap(~continent) +
  scale_y_continuous(labels=function(x) format(x, big.mark = ".", scientific = FALSE))
  theme_bw() +
  labs(title = "Grafico 1. Relación entre tamaño de población y expectativa de vida",
       subtitle = "Según continente de los países",
       x="Expectativa de vida",y="Población") +
  theme(legend.position = "bottom")
```

ggplot2. Otras capas



ggplot2 y transf. de datos

El ejemplo fue sencillo. La data frame tenía la estructura exacta para el gráfico que queríamos hacer.

¿Y si queremos comparar la trayectoria de la expectativa de vida?

```
gapminder %>% ggplot(aes(x=year,y=lifeExp)) +  
  geom_line()
```



ggplot2 y transf. de datos

Muchas soluciones. Depende de nuestros objetivos. Para algunas el dato esta perfecto. Para otras requiere manipulación.

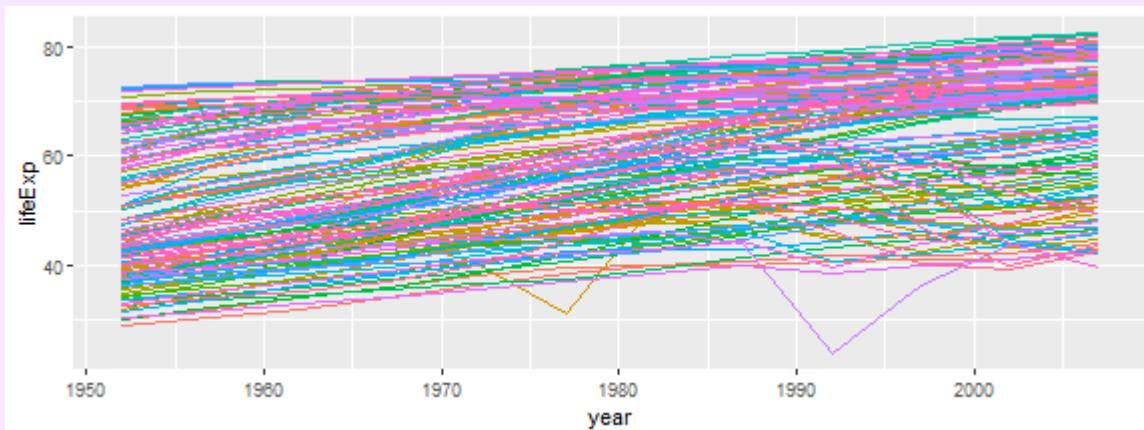
- I. Quedarnos con un solo país.

```
gapminder %>% filter(country=="Chile") %>%  
  ggplot(aes(x=year,y=lifeExp)) +  
  geom_line()
```

ggplot2 y transf. de datos

- II. Intentar visualizar todos los países (A)

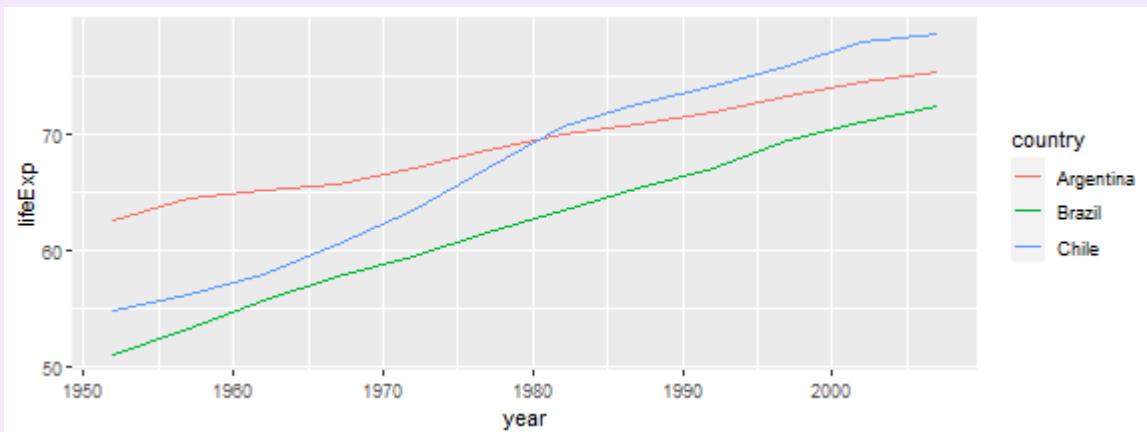
```
gapminder %>%  
  ggplot(aes(x=year,y=lifeExp,color=country)) +  
  geom_line() + theme(legend.position = "none")
```



ggplot2 y transf. de datos

- II. Intentar visualizar algunos países (B)

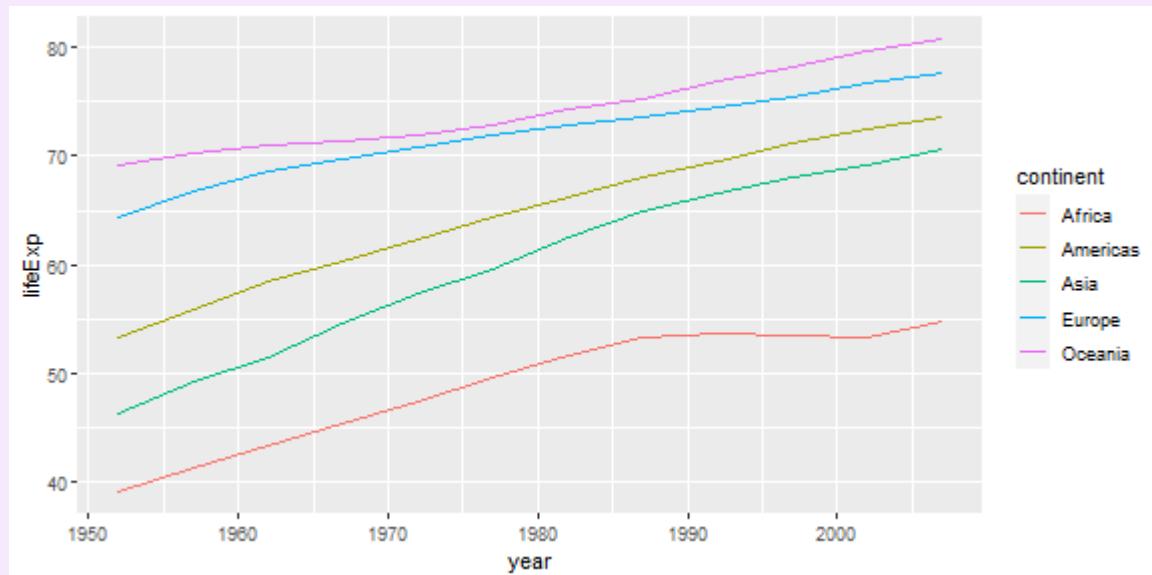
```
gapminder %>%  
  filter(country=="Chile" |  
         country=="Brazil" |  
         country=="Argentina") %>%  
  ggplot(aes(x=year, y=lifeExp, color=country)) +  
  geom_line()
```



ggplot2 y transf. de datos

- III. Agrupar la base a nivel de continentes-año y visualizar por continentes

```
gapminder %>% group_by(continent,year) %>%
  summarise(lifeExp=mean(lifeExp)) %>%
  ggplot(aes(x=year,y=lifeExp,color=continent)) +
  geom_line()
```



ggplot2 y transf. de datos

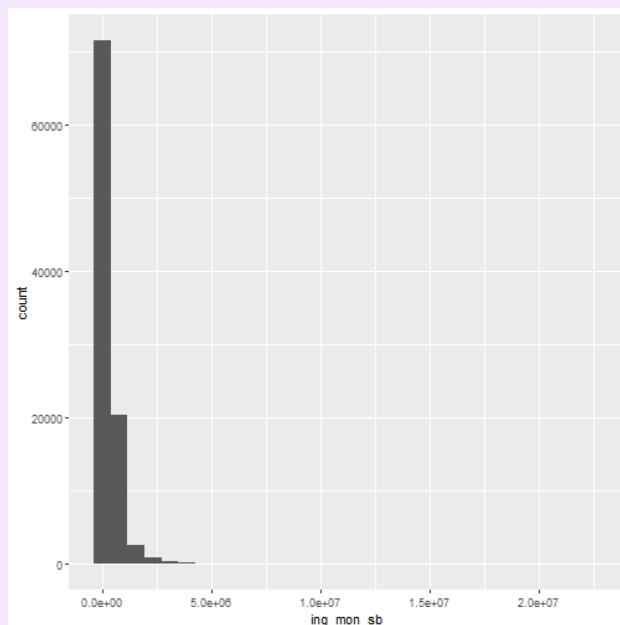
También podemos centrarnos en un solo año y algunos países.

```
gapminder %>% filter(year==2007 & country %in% c("Chile", "Brazil", "Argentina"))  
  ggplot(aes(x=country, y=lifeExp)) +  
    geom_bar(stat = "identity")
```

ggplot2. Histograma

```
esi <- haven::read_sav("data/esi-2019---personas_s.sav")
```

```
esi %>% ggplot(aes(ing_mon_sb)) + geom_histogram()
```



ggplot2. Histograma

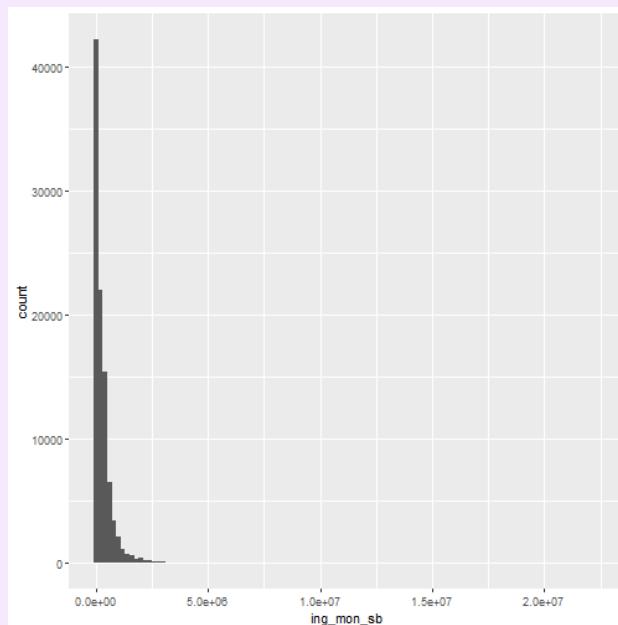
Indicar número de barras

```
esi %>% ggplot(aes(ing_mon_sb)) +  
  geom_histogram(bins = 100)
```

ggplot2. Histograma

Indicar anchura de las barras

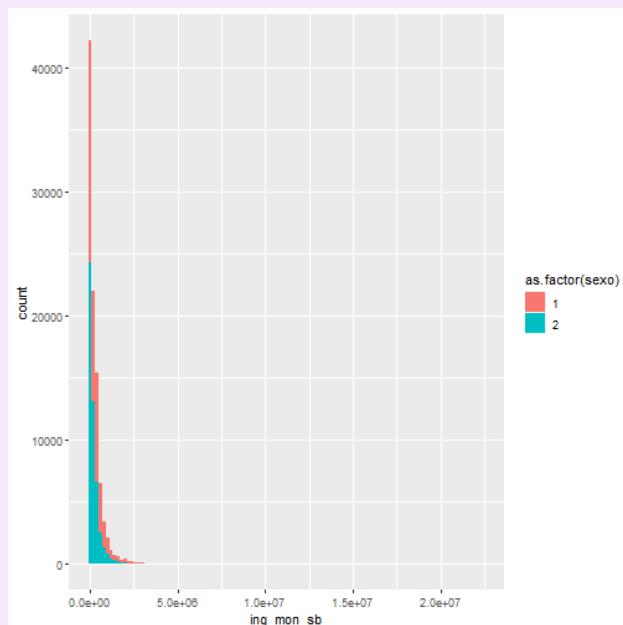
```
esi %>% ggplot(aes(ing_mon_sb)) +  
  geom_histogram(binwidth = 200000) # Rangos de $200.000
```



ggplot2. Histograma

Distinguir por tercera variable

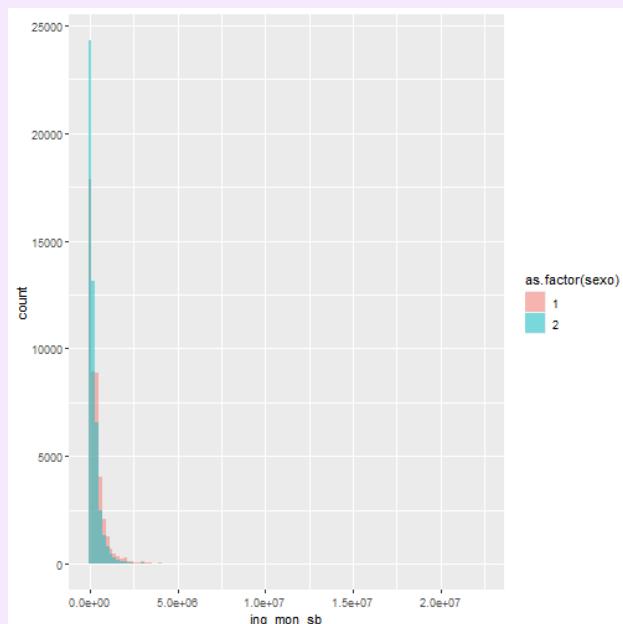
```
esi %>% ggplot(aes(ing_mon_sb,  
                     fill=as.factor(sexo))) +  
  geom_histogram(binwidth = 200000)
```



ggplot2. Histograma

Distinguir por tercera variable y detalle

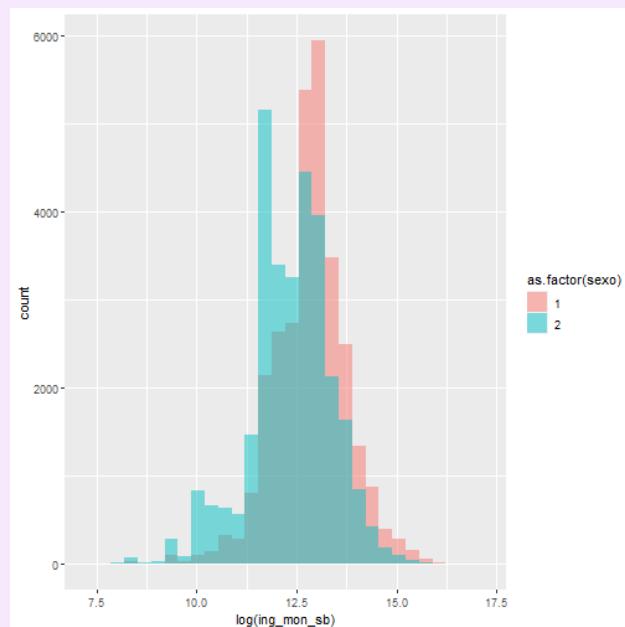
```
esi %>% ggplot(aes(ing_mon_sb,  
                     fill=as.factor(sexo))) +  
  geom_histogram(binwidth = 200000, alpha=0.5, position="identity")
```



ggplot2. Histograma

Logaritmo de ingresos para normalizar

```
esi %>% ggplot(aes(log(ing_mon_sb),  
                     fill=as.factor(sexo))) +  
  geom_histogram(alpha=0.5, position="identity")
```



Exportar gráficos

Función para exportar el último gráfico ejecutado en R.

Sirve para pegarlos en informes en word o presentaciones,

```
ggsave(  
  plot = last_plot(),  
  filename = "graphs/grafico1.png",  
  device = "png",  
  dpi = "retina",  
  units = "cm",  
  width = 25,  
  height = 15  
)
```

En RMarkdown es recomendable exportarlos y luego cargarlos como imagen (control dimensiones)

```
knitr:::include_graphics('graphs/grafico1.png') # lo mismo que 
```

Últimos consejos

Muchas veces nos vamos a encontrar con la necesidad de hacer un gráfico, pero no sabremos:

- Cuál de todos es el más adecuado.
- Cómo realizarlo.

Para ello, es esencial saber cómo buscar en Internet.

- Si no sabemos qué hacer, hacemos un barrido de las gráficas más comunes de ggplot2.
Ver [The R Graph Gallery](#).

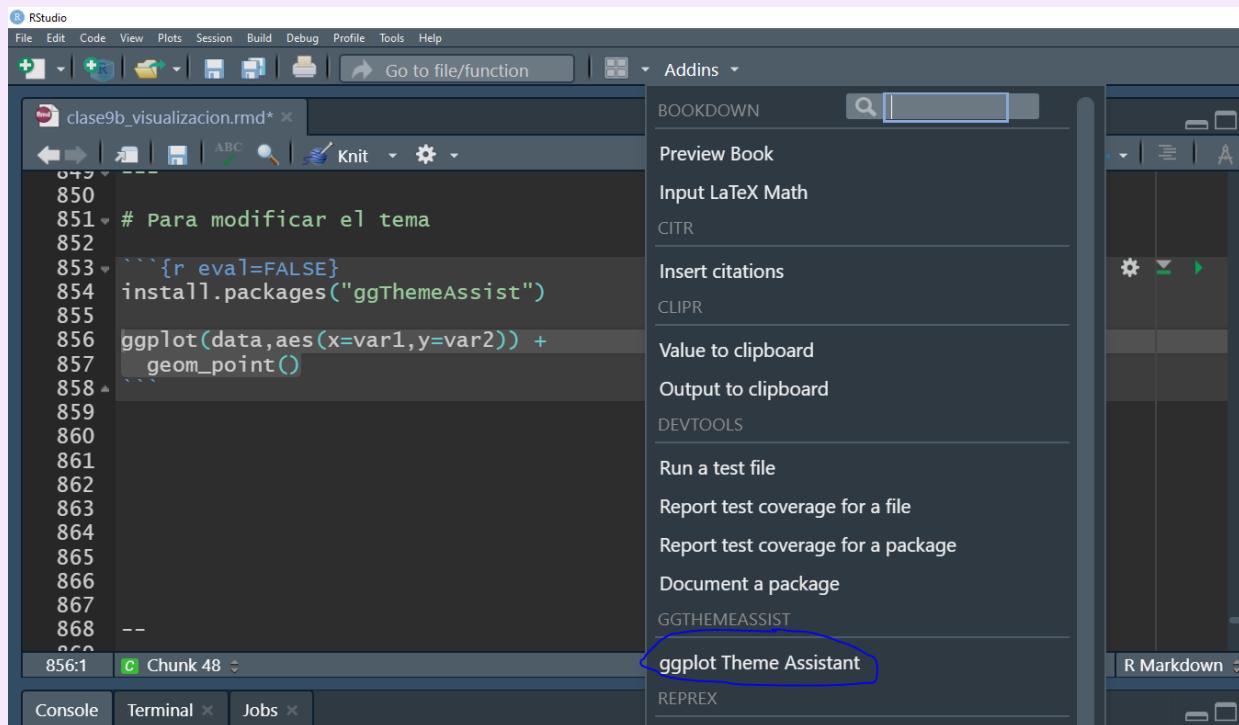
Definimos bien lo que queremos: gráfico apilado de barras de porcentajes (e.g.).

Buscamos cómo se llama el gráfico en inglés, dado que hay mucha más documentación.

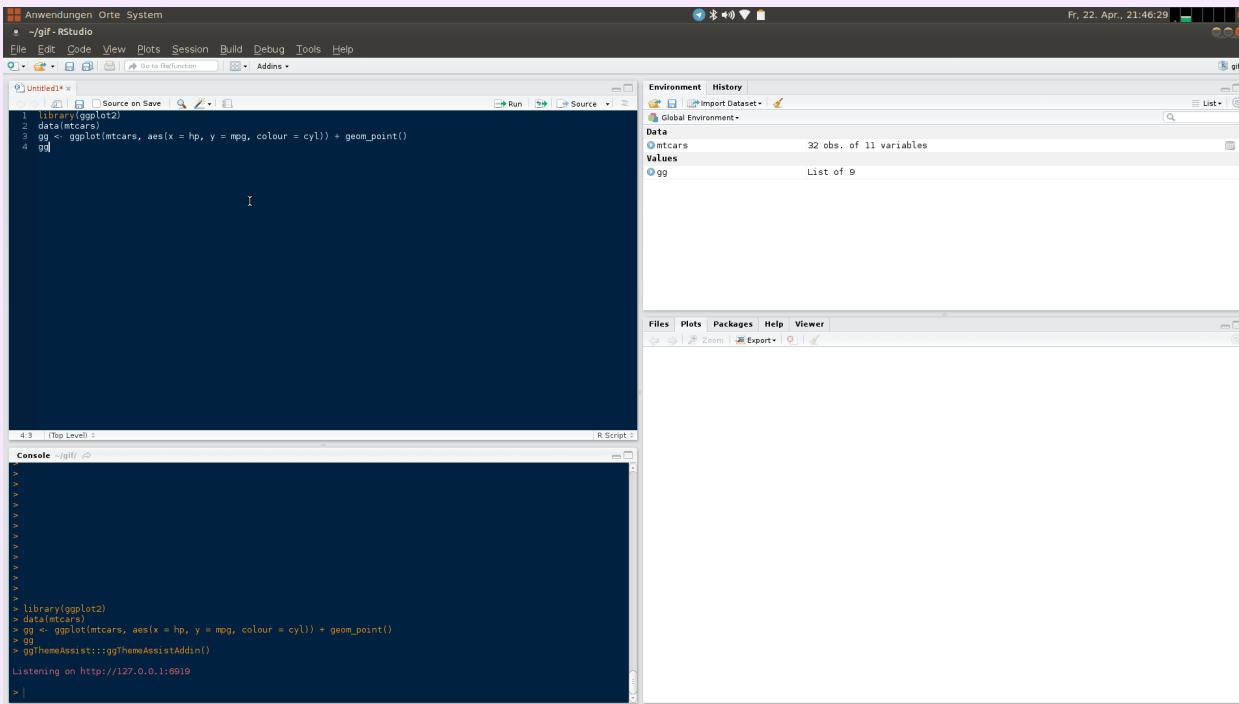
Probamos con una sintaxis previa y la adaptamos a nuestros datos, fijándonos en la estructura de éstos.

Para tema/apariencia

```
install.packages("ggThemeAssist")
```



Para modificar el tema



ggplot2:

Build a data
MASTERPIECE



Material de consulta

Healy, K. Make a plot. Capítulo en *Data visualization. A practical introduction.*

Heiss, A. Course *Data Visualization. Use R, ggplot2, and the principles of graphic design to create beautiful and truthful visualizations of data.*

Rosling, H. Video "The joy of Stats" (Con subtítulos)*.

Como "animar" gráfico de ggplot2