Semana 02 - code sessions (intro)

Visualizacion y limpieza de datos

Carlos Daboín

May, 2022

La vez pasada

- R y Rstudio.
- Principios de programacion en R: Objetos, funciones, asignaciones (<-).
- Mil maneras de tener datos desordenados, una de tenerlos ordenados (tidy).

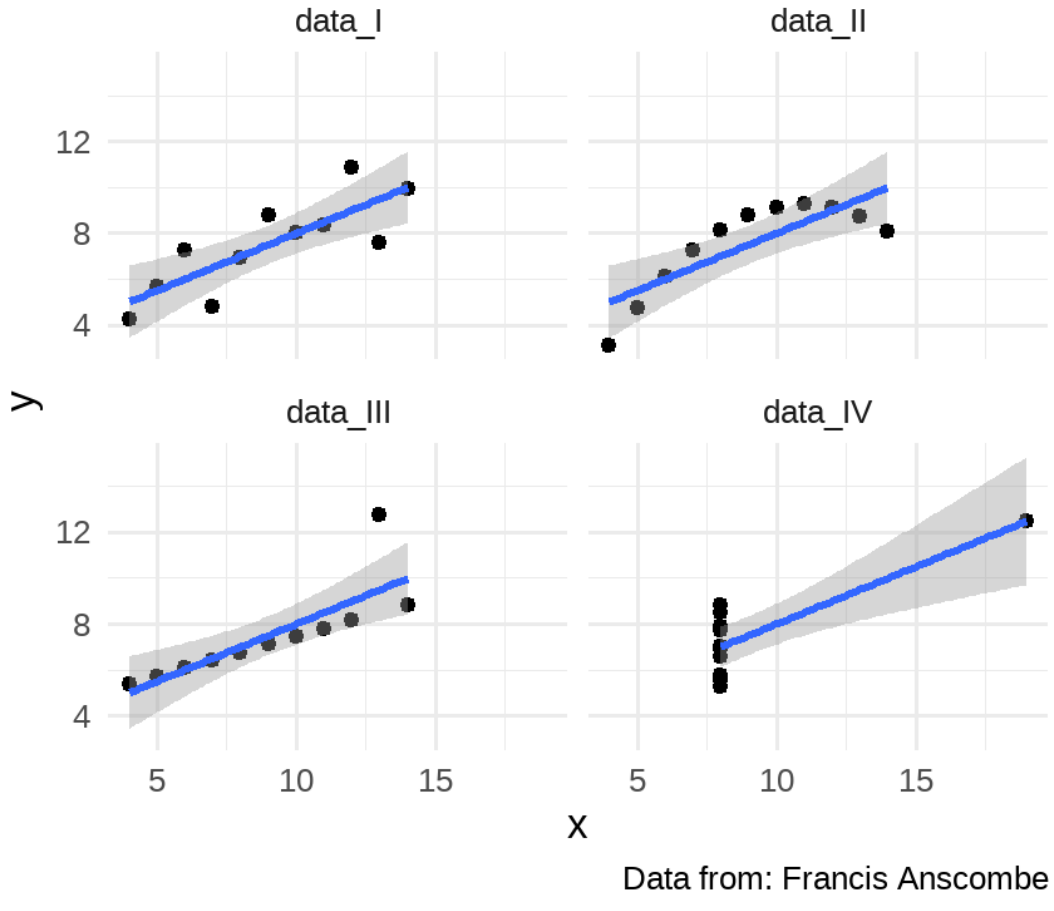
- Dplry: el abrelatas del analisis de datos (%>%, filter(), mutate(), group_by(), summarise()) .

- Quedamos en que iba a instalar R y Rstudio. Lo vamos a necesitar.

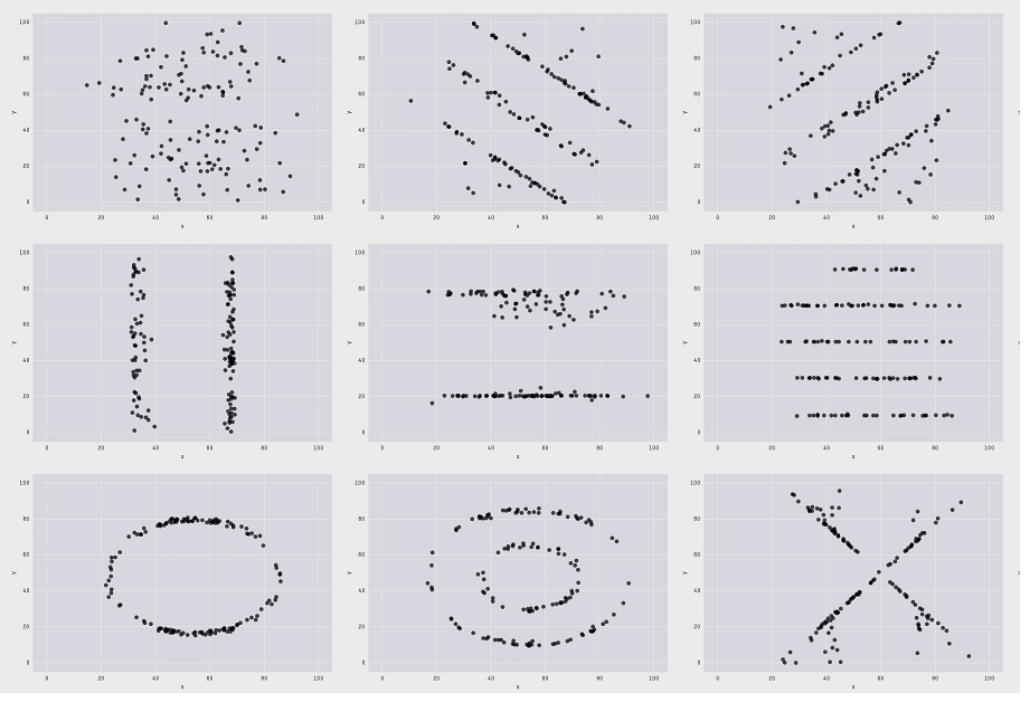
Hoy

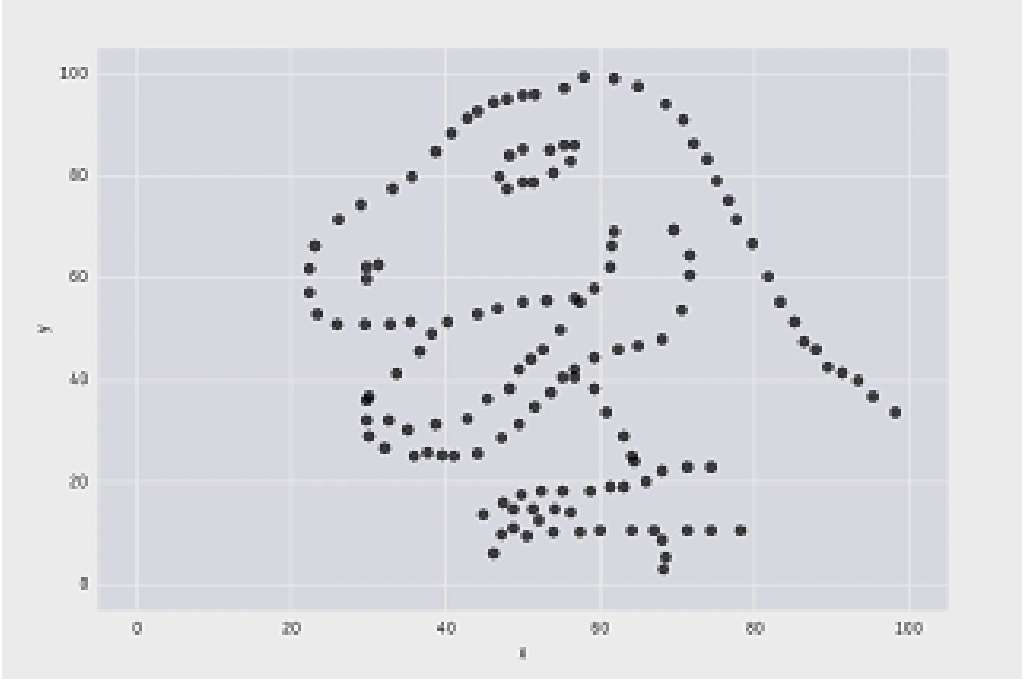
- Nos entretendremos con ggplot2 y sus virtudes
- Pondremos manos a la obra

Una imagen vale mas que mil palabras

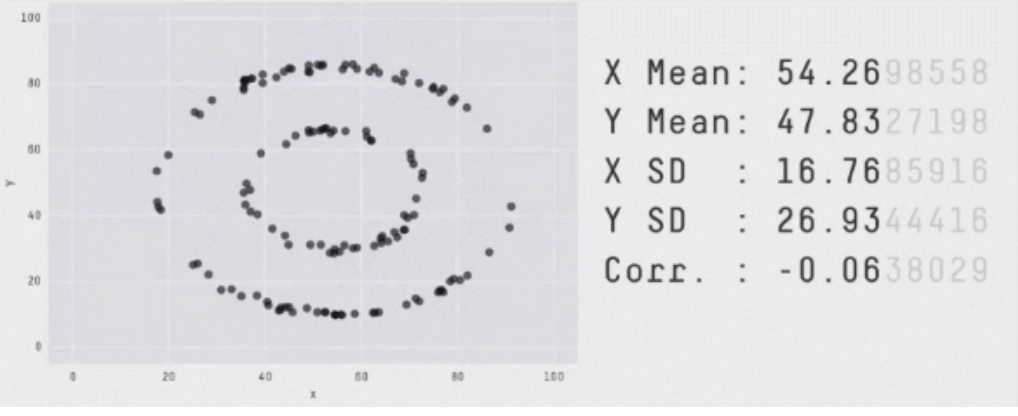


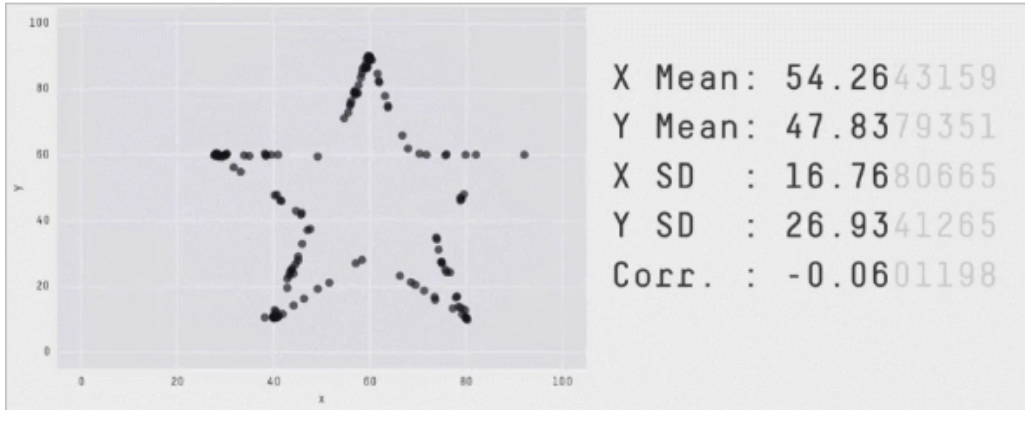
¿Que tienen en común estas figuras?





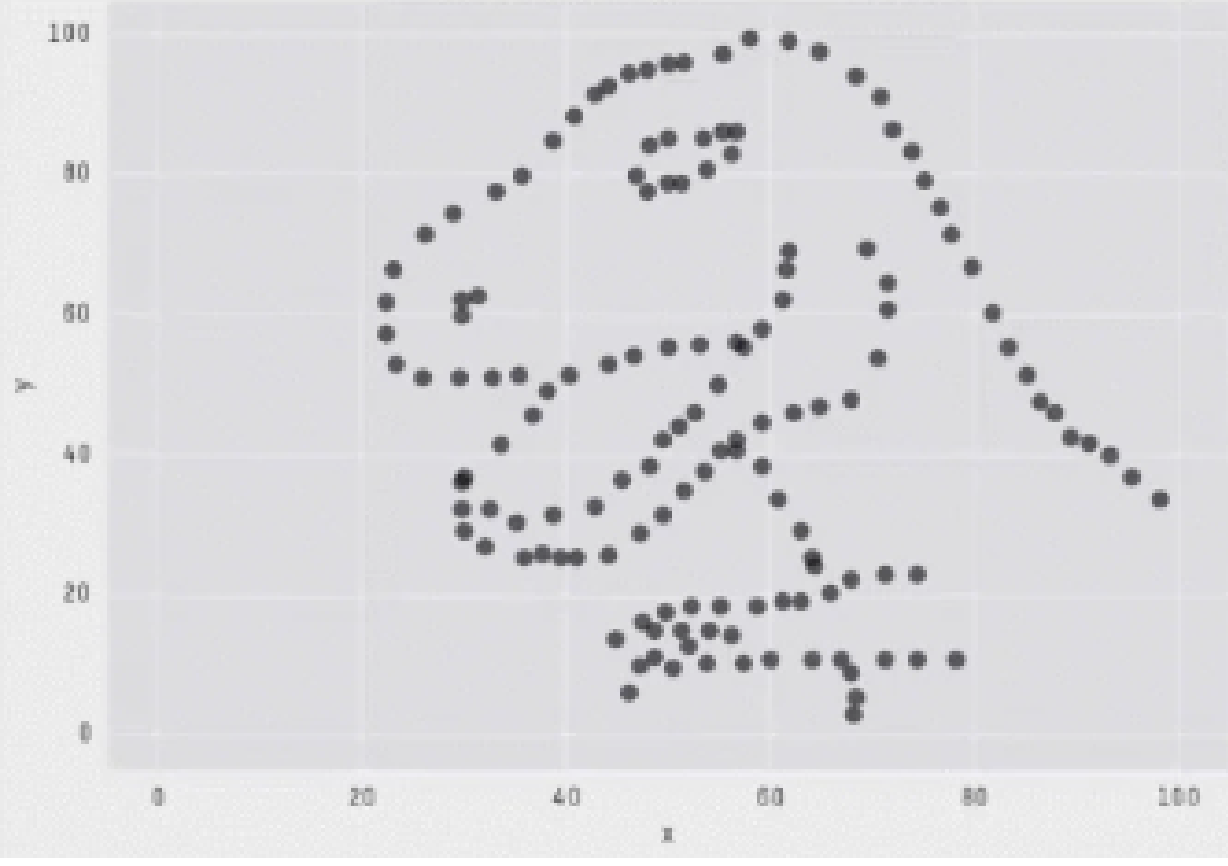
¿Que tienen en común estas figuras?





¿Que tienen en común estas figuras?

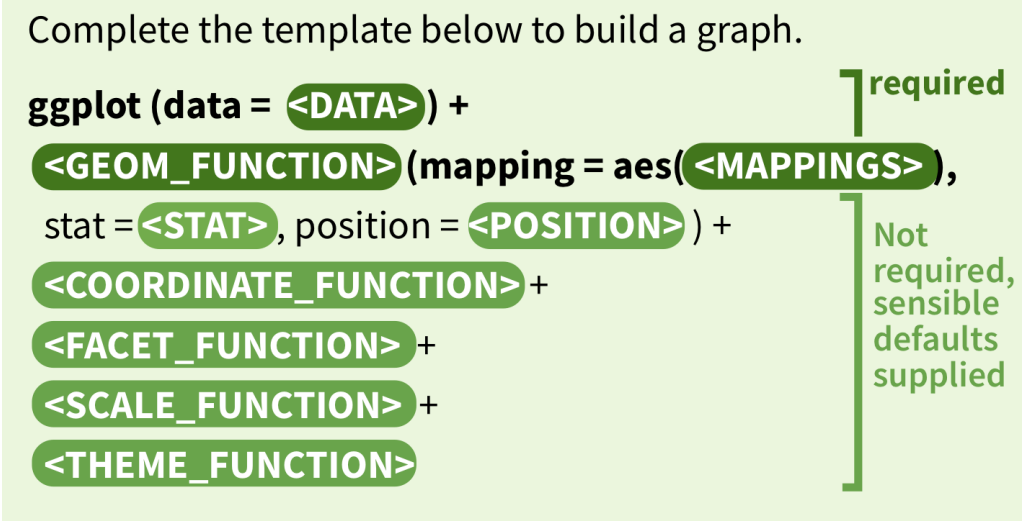
Sorpresa!



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

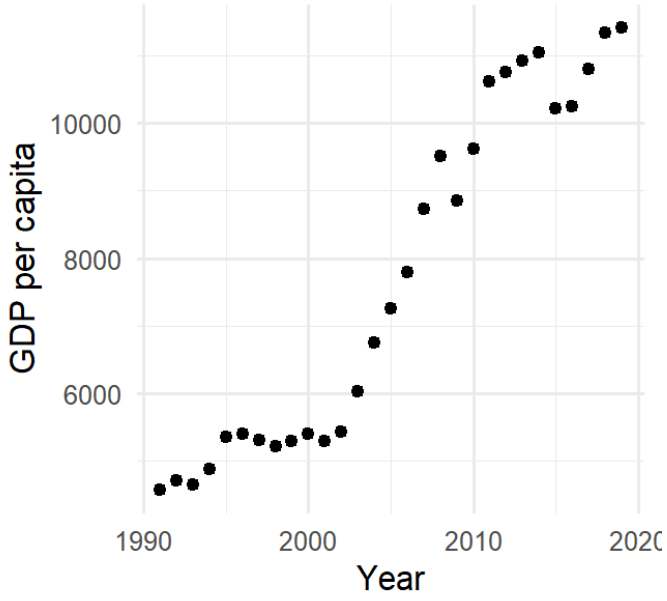
gplot2() and the grammar of graphics

1. Tu gráfica esta vinculada a los datos mediante coordenadas (**aesthetic mappings**)
2. Una vez que esaas coordenadas estan definidas puedes presentar tus graficos en distintas formas (**geoms**), tales como puntos, lineas, barras, etc
3. Puedes agregar tantas capas como gustes a una grafica

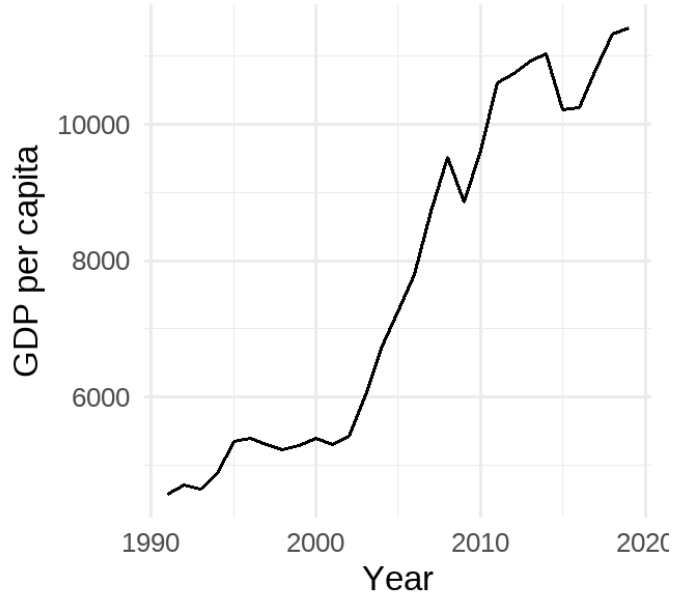


mapeo de coordenadas vs. y aplicacion de geoms

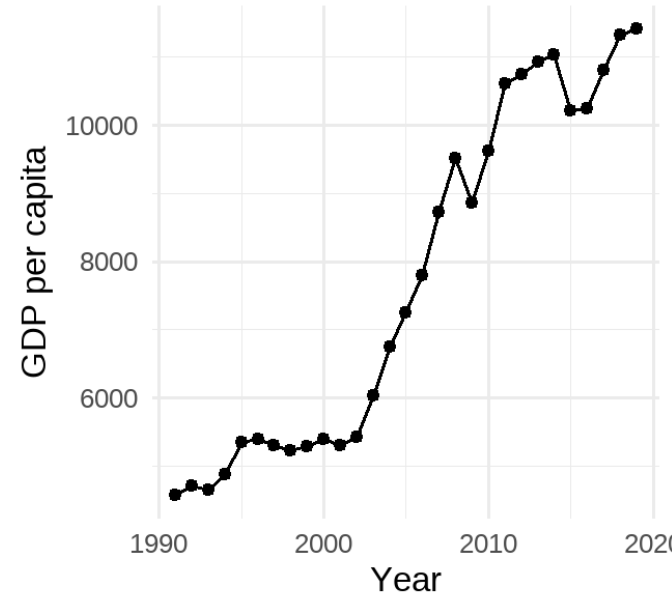
```
ggplot(data = gdp_pc_by_year,
       aes(x = year, y = gdp_p
  geom_point() +
  labs(x = "Year",
       y = "GDP per capita")
```




```
ggplot(data = gdp_pc_by_year,
       aes(x = year, y = gdp_p
  geom_line() +
  labs(x = "Year",
       y = "GDP per capita")
```

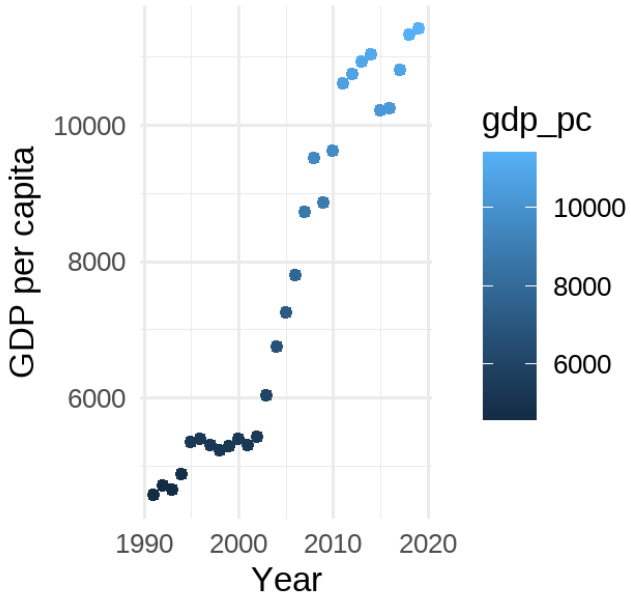



```
ggplot(data = gdp_pc_by_year,
       aes(x = year, y = gdp_p
  geom_point()+
  geom_line() +
  labs(x = "Year",
       y = "GDP per capita")
```

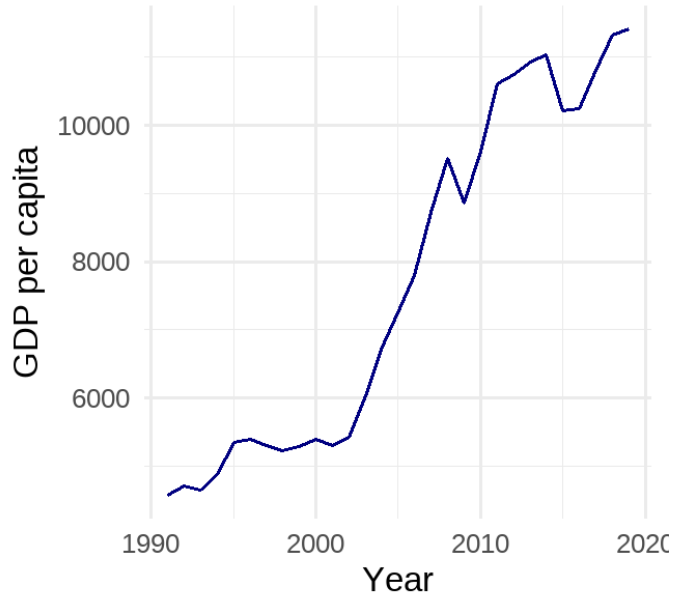


mapeo de atributos vs. hard-coded values

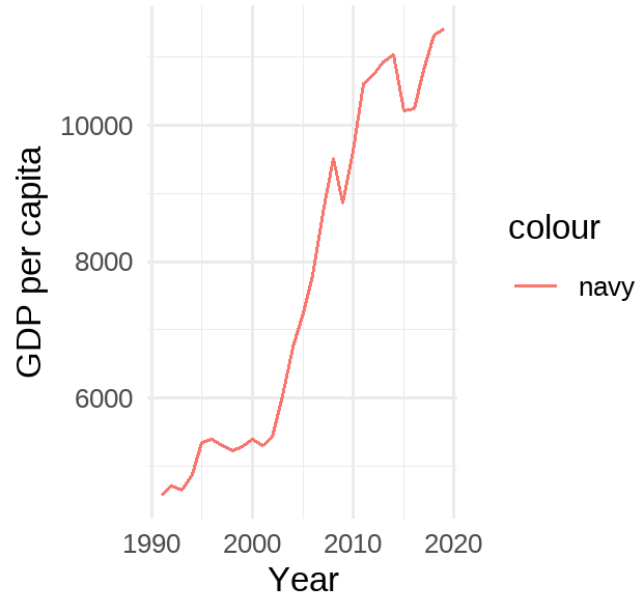
```
ggplot(data = gdp_pc_by_year,
       aes(x = year, y = gdp_p
geom_point(aes(color=gdp_pc))+
  labs(x = "Year",
       y = "GDP per capita")
```




```
ggplot(data = gdp_pc_by_year,
       aes(x = year, y = gdp_p
geom_line(color="navy")+
  labs(x = "Year",
       y = "GDP per capita")
```




```
ggplot(data = gdp_pc_by_year,
       aes(x = year, y = gdp_p
geom_line(aes(color="navy"))+
  labs(x = "Year",
       y = "GDP per capita")
```



Hablemos de la evolucion del ingreso por habitante y la esperanza de vida

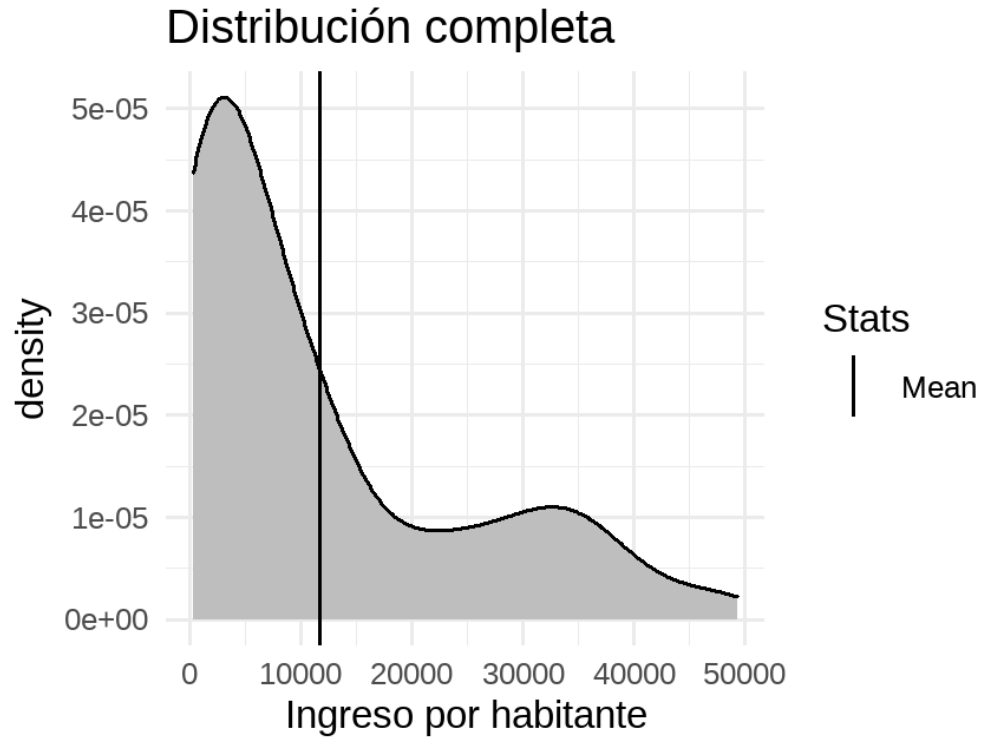
World development indicators (World Bank)

country_name	country_code	year	gdp_pc	gdp	life_exp	population	continent_name
Afghanistan	AFG	1960	59.77323	537777811	32.446	8996967	Asia
Afghanistan	AFG	1961	59.86090	548888896	32.962	9169406	Asia
Afghanistan	AFG	1962	58.45801	546666678	33.471	9351442	Asia
Afghanistan	AFG	1963	78.70643	751111191	33.971	9543200	Asia
Afghanistan	AFG	1964	82.09531	800000044	34.463	9744772	Asia
Afghanistan	AFG	1965	101.10833	1006666638	34.948	9956318	Asia

Gráficos de distribución

Distribución del ingreso por habitante

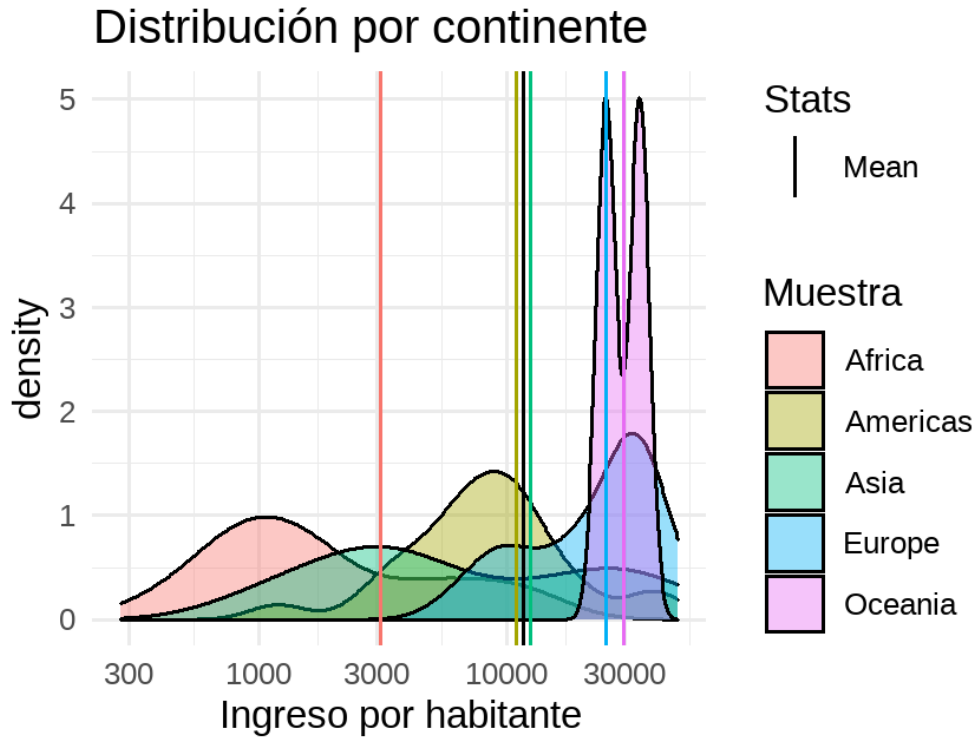
```
# Total
 ggplot(data=gapminder_07, aes(x=gdpPercap))+
  # Geom de distribuciones de densidad. Ponemo
  geom_density(fill="gray")+
  # Geom de lineas verticales. Requiren el val
  # Usamos la palabra "Mean" en punto de corte
  # linea muestre esa palabra (not correct, bu
  geom_vline(aes(xintercept = mean(gdpPercap),
  labs(title="Distribución completa",
       linetype="Stats",
       x="Ingreso por habitante")
```



Gráficos de distribución

Distribución del ingreso por habitante, por continente

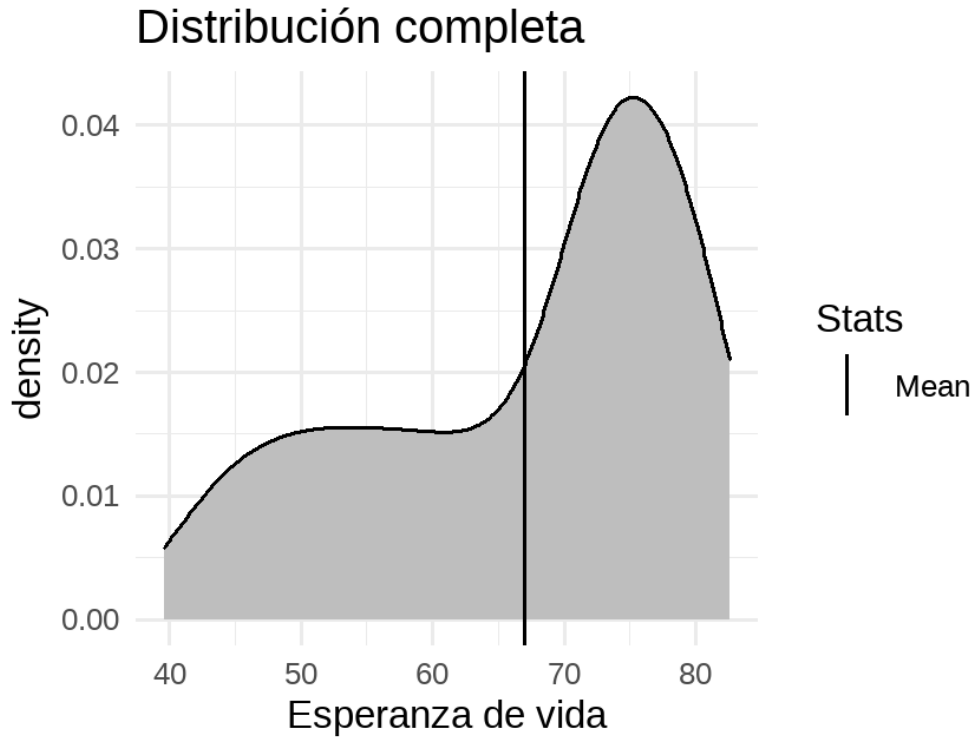
```
# Por continente
  ggplot(data= gapminder_07, aes(x=gdpPercap))+
  # Geom de distribucion de densidades, especi
  geom_density(aes(fill=continent), alpha=0.4)
  # Geom de lineas verticales
  geom_vline(aes(xintercept = mean(gdpPercap),
  # Geom de lineas verticales por contienente.
  geom_vline(data= group_by(gapminder_07,conti
                 summarise(gdpPercap=mean(gdpPer
             aes(xintercept = gdpPercap,color=
             show.legend = F)+
  scale_x_log10()+
  labs(title="Distribución por continente",
       fill="Muestra",
       linetype="Stats",
       x="Ingreso por habitante")
```



Gráficos de distribución

Distribución de la esperanza de vida en el mundo

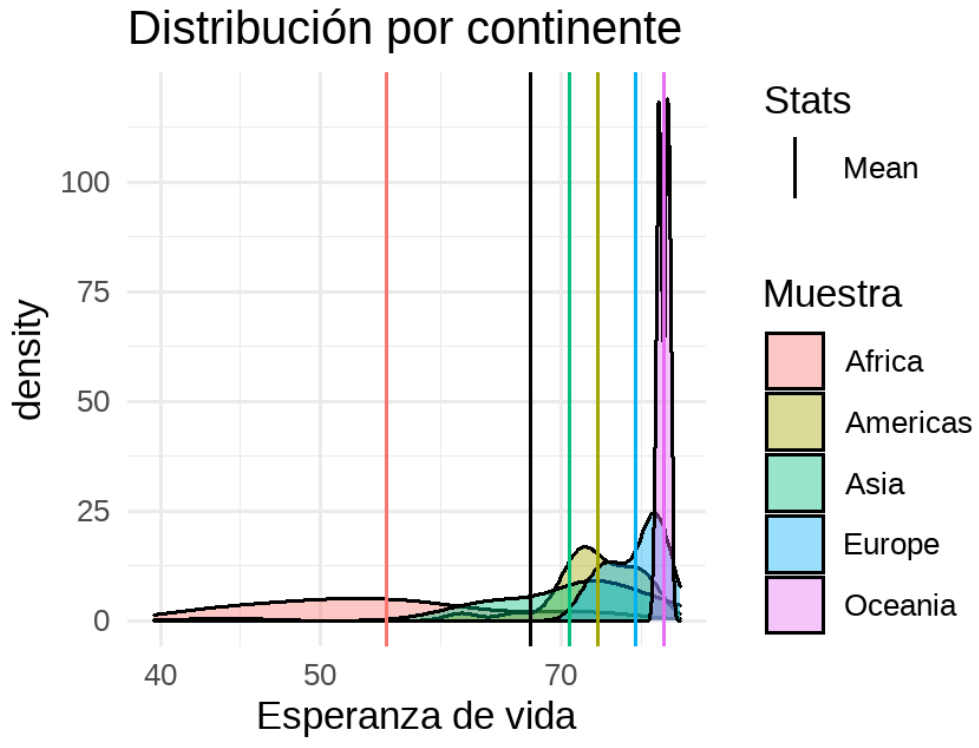
```
gapminder_07 %>%
  ggplot(aes(x=lifeExp))+
  # Geom de distribuciones de densidad. Ponemo
  geom_density(fill="gray")+
  # Geom de lineas verticales. Requiren el val
  # Usamos la palabra "Mean" en punto de corte
  # linea muestre esa palabra (not correct, bu
  geom_vline(aes(xintercept = mean(lifeExp),li
  labs(title="Distribución completa",
       linetype="Stats",
       x="Esperanza de vida")
```



Gráficos de distribución

Distribución de la esperanza de vida, por continente

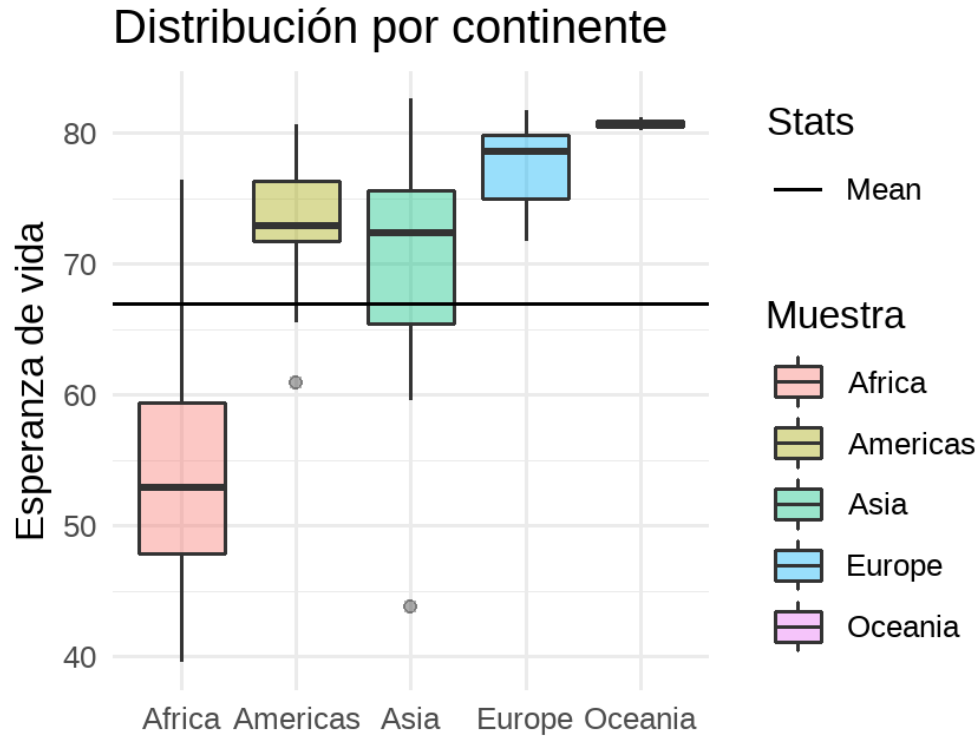
```
gapminder_07 %>%
  ggplot(aes(x=lifeExp))+
  # Geom de distribucion de densidades, especi
  geom_density(aes(fill=continent), alpha=0.4)
  # Geom de lineas verticales
  geom_vline(aes(xintercept = mean(lifeExp),li
  # Geom de lineas verticales por contienente
  # Insertamos datos agregados a nivel contine
  # lineas
  geom_vline(data= gapminder_07 %>%
                group_by(continent) %>%
                summarise(lifeExp=mean(lifeExp)
             aes(xintercept = lifeExp,color=co
             show.legend = F)+
  scale_x_log10()+
  labs(title="Distribución por continente",
       fill="Muestra",
       linetype="Stats",
       x="Esperanza de vida")
```



Gráficos de distribución

Hay una manera más practica de ver distribuciones a nivel de grupos

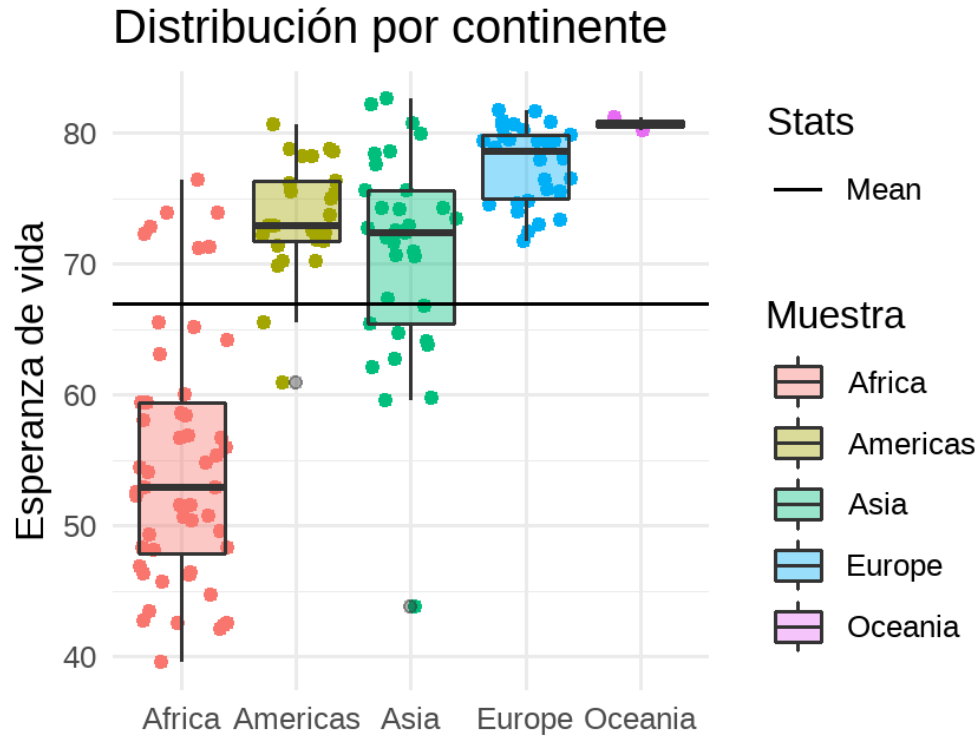
```
gapminder_07 %>%
  ggplot(aes(y=lifeExp,x=continent))+
  # Geom de distribucion por cuartiles (boxplo
  geom_boxplot(aes(fill=continent), alpha=0.4)
  # Geom de lineas verticales
  geom_hline(aes(yintercept = mean(lifeExp),li
  labs(title="Distribución por continente",
       fill="Muestra",
       linetype="Stats",
       x=NULL,
       y="Esperanza de vida")
```



Gráficos de distribución

Pongámosle una capa de pimienta

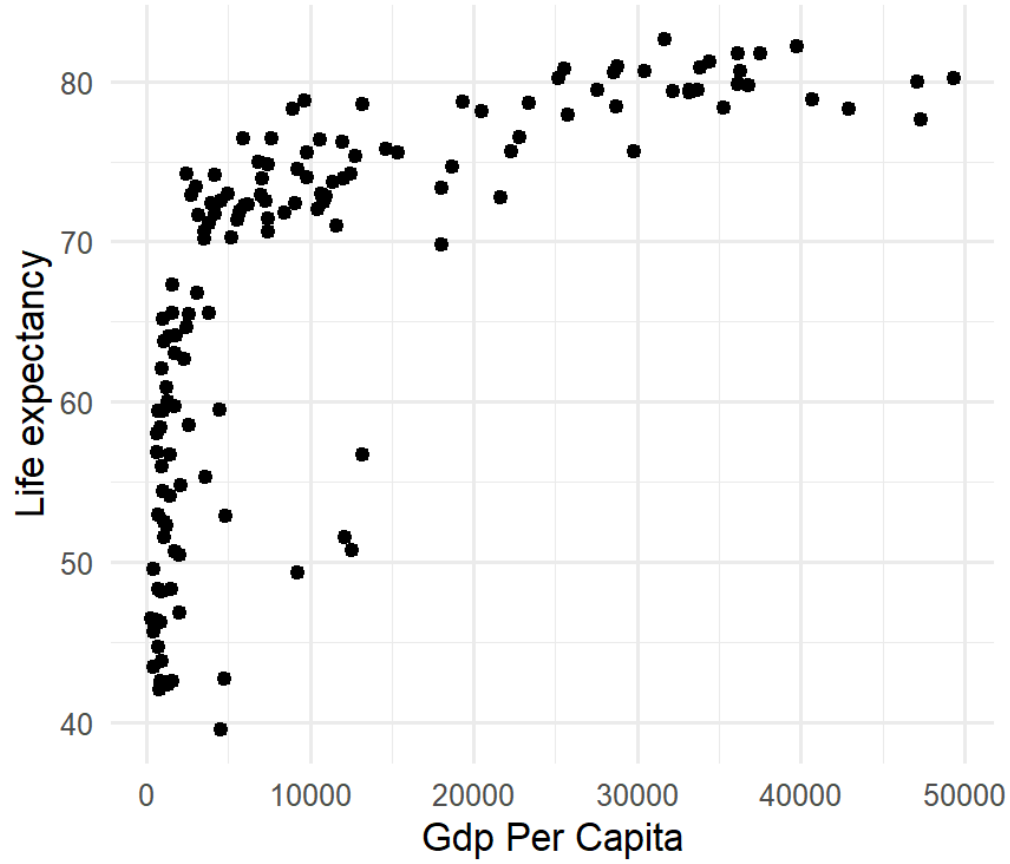
```
gapminder_07 %>%
  ggplot(aes(y=lifeExp,x=continent))+
  # Anademos un poco de pimienta con geom_jitt
  geom_jitter(aes(color=continent),show.legend
  # Geom de distribucion por cuartiles (boxplo
  geom_boxplot(aes(fill=continent), alpha=0.4)
  # Geom de lineas verticales
  geom_hline(aes(yintercept = mean(lifeExp),li
  labs(title="Distribución por continente",
       fill="Muestra",
       linetype="Stats",
       x=NULL,
       y="Esperanza de vida")
```



Relación entre dos variables continuas

Define la data, las coordenadas, y la forma

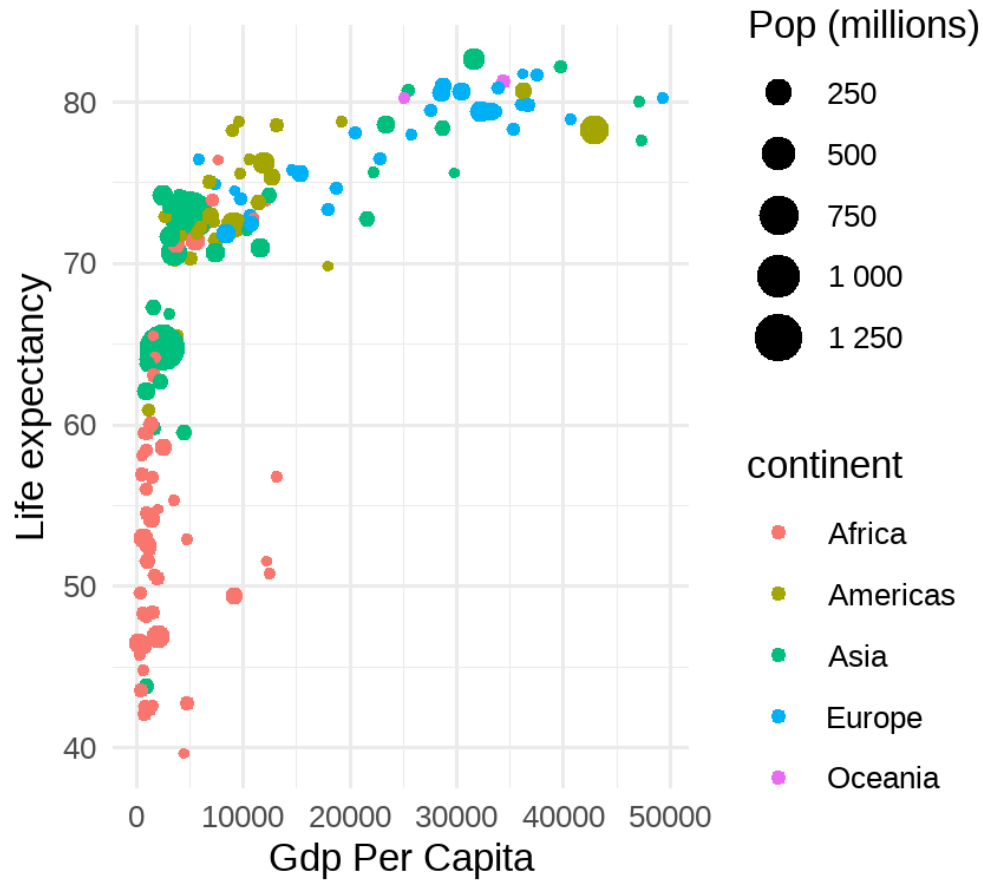
```
# Datos y coordenadas
ggplot(data=gapminder_07,
       mapping = aes(y=lifeExp,x=gdpPercap))+
  # Formas o geometrias
  geom_point()+
  labs(x="Gdp Per Capita",
       y="Life expectancy")
```



Relación entre dos variables continuas

Añade otras formas y haz cambios en el formato

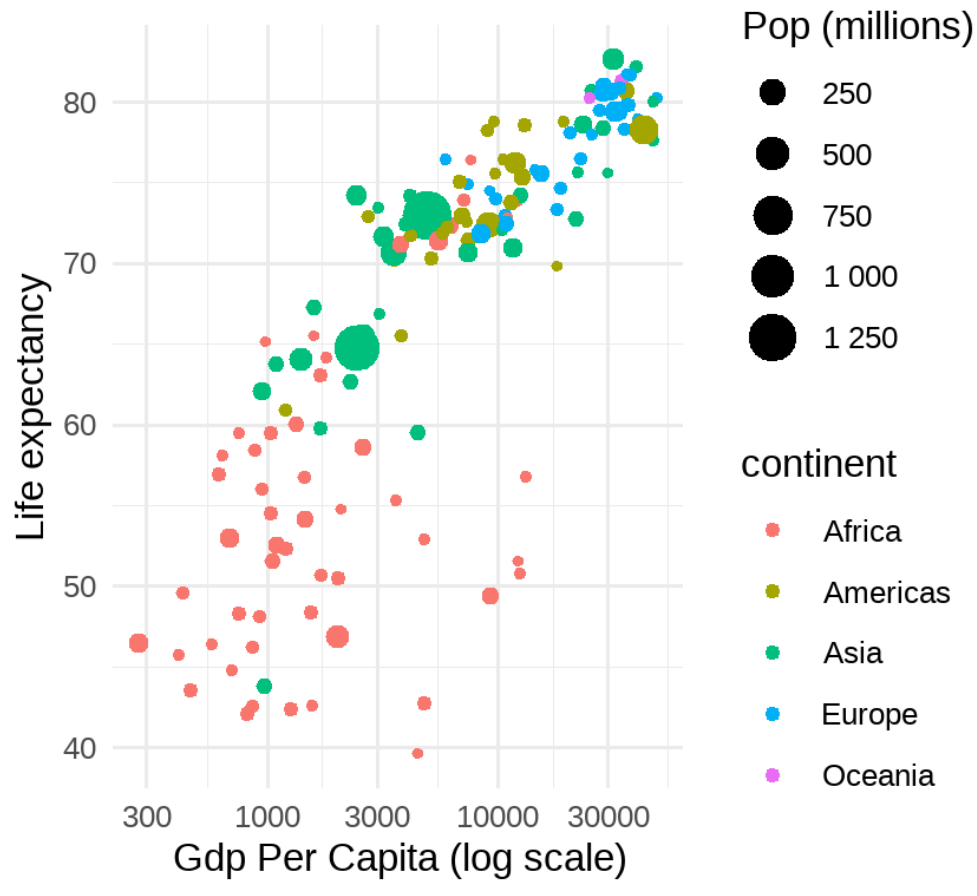
```
ggplot(data=gapminder_07,
       mapping = aes(y=lifeExp,x=gdpPercap))+
  # coordenadas para una geometria especifica
  geom_point(aes(size=pop/1000000, color=conti
  scale_size_continuous(labels=scales::number_
  labs(x="Gdp Per Capita",
       y="Life expectancy",,
       size="Pop (millions)")
```



Relación entre dos variables continuas

Cambiemos la escala de gpd per capita ¿Qué ganámos con logs?

```
ggplot(data=gapminder_07,
       mapping = aes(y=lifeExp,x=gdpPercap))+
  # coordenadas para una geometria especifica
  geom_point(aes(size=pop/1000000, color=conti
  scale_size_continuous(labels=scales::number_
  scale_x_log10()+
  labs(x="Gdp Per Capita (log scale)",
       y="Life expectancy",
       size="Pop (millions)")
```



Manos a la obra: Repliquemos estos graficos en Rstudio

Usaremos los datos del World Bank para replicar las figuras que acaban de ver

Manos a la obra: Repliquemos estos graficos en Rstudio

Pero tendremos que limpar los datos originales

Country Name	Series Name	Series Code	1960 [YR1960]
Afghanistan	GDP per capita (current US$)	NY.GDP.PCAP.CD	59.77323370321
Afghanistan	GDP (current US$)	NY.GDP.MKTP.CD	537777811.1111
Afghanistan	Life expectancy at birth, total (years)	SP.DYN.LE00.IN	32.446
Afghanistan	Population, total	SP.POP.TOTL	8996967

Manos a la obra: Repliquemos estos graficos en Rstudio

Pero tendremos que limpar los datos originales

Abran R studio

Abran el archivo "...."

Abran las librerias dplry, ggplot2.

Pero tambien abran la librerias tidyr, janitor y stringr

- En el proceso vamos a aprender sobre **pivots**, **joins** y un par de cosas mas.

tidyr::pivot_longer()

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

tidyr::pivot_wider()

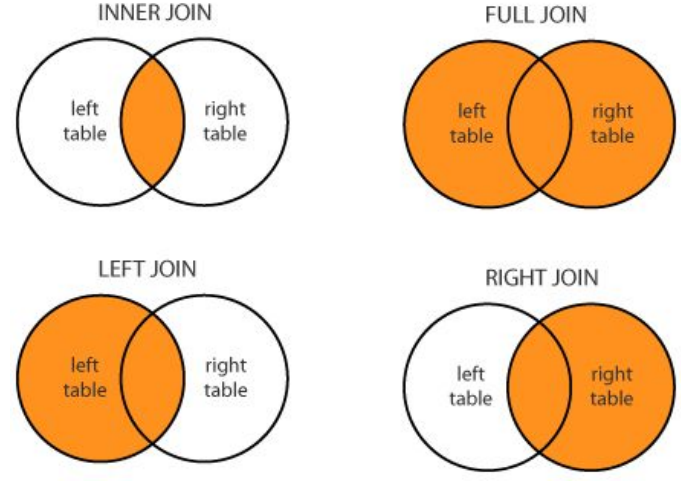
table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

tidyr::left_join()



Para finalizar: Materiales complementarios

Fuente recomendadas para seguir aprendiendo (new):

- Rstudio education: Ofrecen las mejores guias para aprender R a cualquier nivel.
- Rstudio tutorials: Tutoriales interactivos gratuitos.
- Rstudio cheatsheets: Necesitas despejar algunas dudas rápido? Revisa estas chuletas, son lo mejor.

Fuente recomendadas para seguir aprendiendo (old):

- R for Economists video series (by Nick Hungtington-Klein)
- R for Data Sicence (Wickham & Grolemund, 2017)
- Statistical Inference via Data Science (Ismay & Kim, 2022)
- Top 50 ggplot2 Visualizations - The Master List (by Selva Prabhakaran).
- Video: The best stats you'll ever see (by Hans Rosling)
- Video: Statistics without the agonizing pain (by John Rauser)

Próxima semana

A partir de aquí nos sentaremos a replicar resultados de estudios sin pararnos tanto en el proceso de limpieza y transformación de datos.

Practiquen lo siguiente (noten que ahí les deje links a los tutoriales):

- Limpieza y transformación de datos (**dplry**, **tidyr**, **janitor**, **stringr**,**readr**).
- Visualización de datos (**ggplot2**).
- Tipos de objetos en R (vectores, matrices, data.frames, tiblles, lists).
- Iteraciones (for loops, lapply functions, the purr package).

Fin de primer taller

Gracias