

Explorando el mundo

ExactasPrograma - Datos

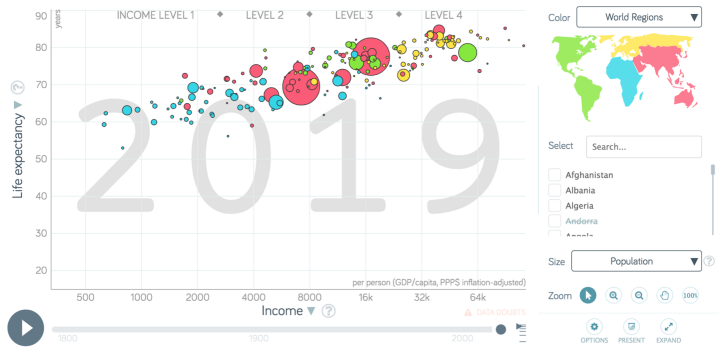
Facultad de Ciencias Exactas y Naturales, UBA

Invierno 2020

- Gapminder.org nos ofrece una recopilación de indicadores de los países del mundo de los últimos 200 años



Exploremos el mundo



- Vamos a explorar algunos sets de datos de Gapminder, y graficar algunos indicadores vs otros en diferentes años

Mas allá del NaN...

- Gapminder.org nos ofrece una recopilación de indicadores de los países del mundo de los últimos 200 años



- Teniendo datos ya curados, ya no nos tenemos que preocupar...¿no?

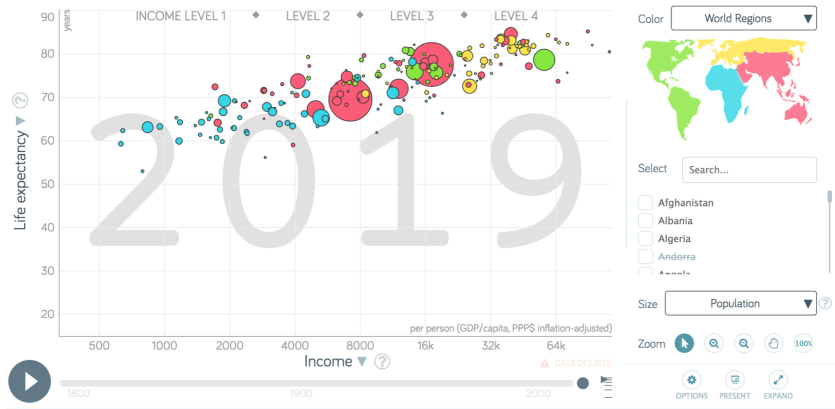
Mas allá del NaN...

- Gapminder.org nos ofrece una recopilación de indicadores de los países del mundo de los últimos 200 años

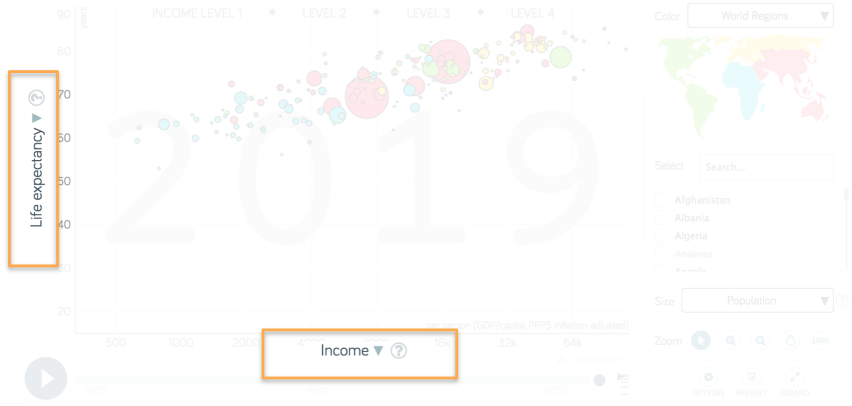


- Teniendo datos ya curados, ya no nos tenemos que preocupar...¿no?
- Diferentes fuentes, definiciones de indicadores, unidades, tipos de pre- o post-procesamiento, sesgos en la obtención de los datos...

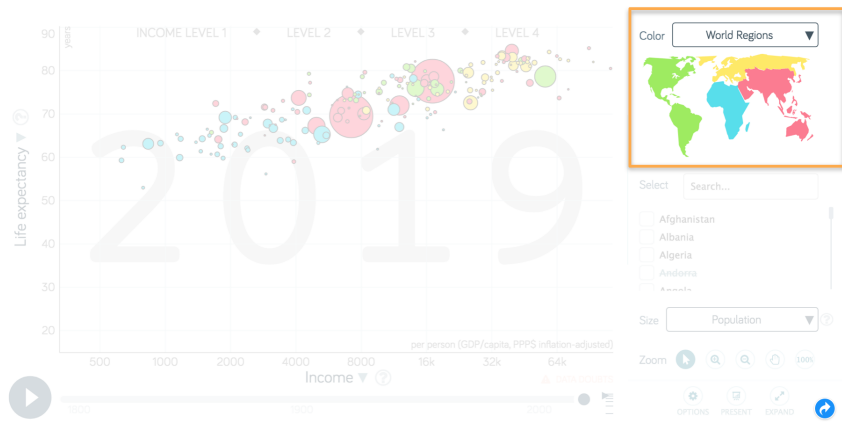
Qué necesitamos para reproducir este gráfico?



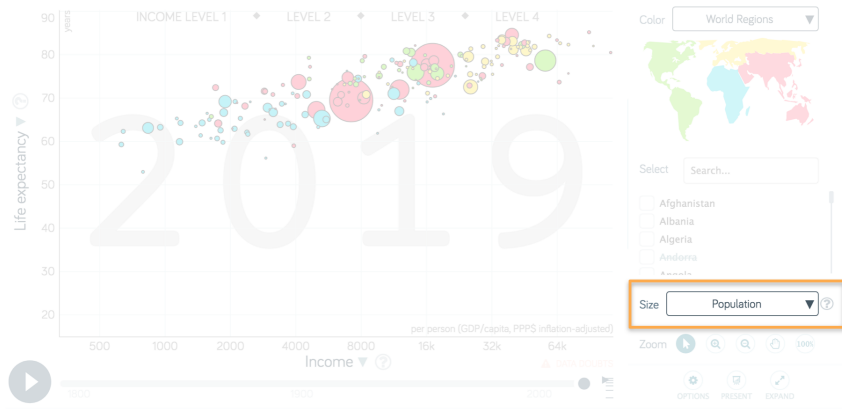
Qué necesitamos para reproducir este gráfico?



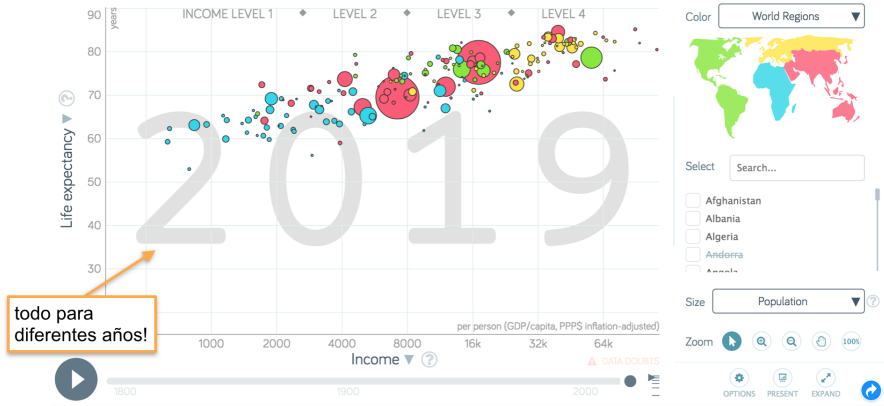
Qué necesitamos para reproducir este gráfico?



Qué necesitamos para reproducir este gráfico?



Qué necesitamos para reproducir este gráfico?



Información de Gapminder

- Cada **indicador** tiene una tabla (.csv) con datos para todos los países y los últimos 200 años aproximadamente

Información de Gapminder

- Cada **indicador** tiene una tabla (.csv) con datos para todos los países y los últimos 200 años aproximadamente

Vamos a trabajar con varias tablas:

- PBI per cápita
- Emisiones de CO₂
- Esperanza de vida al nacer
- Tamaño de las poblaciones
- Información geográfica

Vamos a combinarlas para construir nuevos dataframes que nos ayuden a hacer los gráficos que queremos usando pandas y seaborn.

Repasemos... ¿qué era un DataFrame?

```
pbi = pd.read_csv("income_per_person_gdppercapita_ppp_inflation_adjusted  
.csv?dl=1", index_col=0)
```

pbi =

country	1800	1801	2010
Afghanistan	603				
Albania	667				
Algeria	715				
...					
Venezuela	1210				
Vietnam	778				
Yemen	877				
Zambia	663				
Zimbabwe	869				

- `pbi.columns` → nombre de las columnas
- `pbi.index` → nombre de las filas

y ¿si extraemos una fila o una columna del DataFrame?

Nos queda un Pandas *Series*

pbi =

country	1800	1801	2010
Afghanistan	603				
Albania	667				
Algeria	715				
...					
Venezuela	1210				
Vietnam	778				
Yemen	877				
Zambia	663				
Zimbabwe	869				

`Af = pbi.loc['Afghanistan']`

`mundo_1800 = pbi['1800']`

Ambos `Af` y `mundo_1800` son *Series*

¿Cómo es eso del Series?

pbi =

country	1800	1801	...
Afghanistan	603		
Albania	667		
Algeria	715		
...			
Venezuela	1210		
Vietnam	778		
Yemen	877		
Zambia	663		
Zimbabwe	869		

mando_1800.index

```
mando_1800 = pbi['1800']  
print(mando_1800)
```

```
country  
Afghanistan    603  
Albania        667  
Algeria        715  
Andorra        1200  
Angola         618  
...  
Venezuela     1210  
Vietnam       778  
Yemen         877  
Zambia        663  
Zimbabwe      869  
Name: 1800, Length: 193, dtype: int64
```

mando_1800.values

¿Cómo es eso del Series?

pbi =

country	1800	1801	
Afghanistan	603		
Albania	667		
Algeria	715		
...			
Venezuela	1210		
Vietnam	778		
Yemen	877		
Zambia	663		
Zimbabwe	869		

Af.index

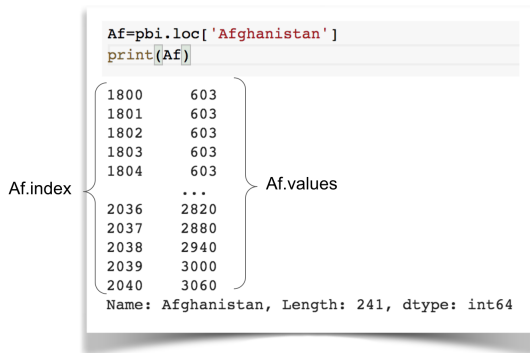
```
Af=pbi.loc['Afghanistan']  
print(Af)
```

```
1800    603  
1801    603  
1802    603  
1803    603  
1804    603  
...  
2036   2820  
2037   2880  
2038   2940  
2039   3000  
2040   3060
```

Af.values

```
Name: Afghanistan, Length: 241, dtype: int64
```


Nos facilita graficar...



Con `Af.plot()` se grafica `Af.index` (eje x) vs `Af.values` (eje y)

Para agendar

- El jueves 20 de agosto a las 16hs
- Un invitado economista!

¡A trabajar!