

Práctica 2 — Proceso KDD aplicado al dataset COVID-19 (Our World in Data)

Carrera: Ciencia de Datos

Asignatura: Minería de Datos

Nombre del Alumno: Luis Enrique Villalon Pineda

Carrera de origen: Actuaría

Objetivos:

1. Entender las fases del proceso KDD
2. Explorar cómo varió la tasa de vacunación para un país en particular (diferente a Italia)
3. Explorar cómo varió la tasa de vacunación de covid en América Latina durante los años de la pandemia
4. Identificar a los países con mayor tasa de vacunación (en América Latina)
5. Identificar las características de los países con mayor tasa de vacunación (en América Latina)

Indicaciones generales

- Ejecuta **celda por celda**, leyendo primero los comentarios
- En las secciones marcadas como **(Experimenta)**, **(Modifica)** o **(Reflexiona)**, realiza lo que se pide y **deja tus conclusiones en texto**. Eres libre de modificar las celdas sin perder de vista el objetivo (puedes cambiar el estilo de visualizaciones o utilizar, por ejemplo seaborn o plotly).
- Deberás documentar todo el proceso, no olvides incluir los metadatos de los atributos que utilices
- Descarga el dataset `owid-covid-data.csv` desde el repositorio [Our World in Data GitHub] (<https://github.com/owid/covid-19-data/tree/master/public/data>)

```
# ===== Librerías =====
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.cluster import KMeans, DBSCAN
from sklearn.metrics import silhouette_score
from sklearn.linear_model import LinearRegression
from pathlib import Path
```

Paso 0 — Exploración del banco de datos y Entendimiento del dominio del negocio:

<https://docs.owid.io/projects/covid/en/latest/dataset.html#the-data-you-find-here-and-our-data-sources>

Instrucciones:

1. Carga el banco de datos
2. Explora el banco de datos completo: dimensión, cabecera, etc.
3. Despliega la estadística básica
4. Presenta la(s) visualización que consideres pertinente para comprender el problema y los datos
5. Busca el archivo con los metadatos del banco de datos
6. Verifica que los metadatos coinciden con el tipo de dato que muestra el archivo
7. Muestra los campos que NO coinciden

Recuerda documentar todas las decisiones que tomaste en el proceso

```
# === Parámetros (Modifica) ===
data_path = Path(r'.\owid-covid-data.csv') # coloca el CSV en el mismo directorio

assert data_path.exists(), f'No se encontró el archivo: {data_path.resolve()}'
df = pd.read_csv(data_path, low_memory=False)
```

```
#2 Despliega la dimensión del data set completo: usa shape y head
df.shape
```

```
(429435, 67)
```

```
#2
df.head(10)
```

	iso_code	continent	location	date	total_cases	new_cases
\						
0	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0
1	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0
2	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0
3	AFG	Asia	Afghanistan	2020-01-08	0.0	0.0
4	AFG	Asia	Afghanistan	2020-01-09	0.0	0.0
5	AFG	Asia	Afghanistan	2020-01-10	0.0	0.0
6	AFG	Asia	Afghanistan	2020-01-11	0.0	0.0

7	AFG	Asia	Afghanistan	2020-01-12	0.0	0.0
8	AFG	Asia	Afghanistan	2020-01-13	0.0	0.0
9	AFG	Asia	Afghanistan	2020-01-14	0.0	0.0

	new_cases_smoothed	total_deaths	new_deaths
new_deaths_smoothed	...	\	

0		NaN	0.0	0.0
NaN	...			
1		NaN	0.0	0.0
NaN	...			
2		NaN	0.0	0.0
NaN	...			
3		NaN	0.0	0.0
NaN	...			
4		NaN	0.0	0.0
NaN	...			
5		0.0	0.0	0.0
0.0	...			
6		0.0	0.0	0.0
0.0	...			
7		0.0	0.0	0.0
0.0	...			
8		0.0	0.0	0.0
0.0	...			
9		0.0	0.0	0.0
0.0	...			

	male_smokers	handwashing_facilities	hospital_beds_per_thousand	\
0	NaN	37.75	0.5	
1	NaN	37.75	0.5	
2	NaN	37.75	0.5	
3	NaN	37.75	0.5	
4	NaN	37.75	0.5	
5	NaN	37.75	0.5	
6	NaN	37.75	0.5	
7	NaN	37.75	0.5	
8	NaN	37.75	0.5	
9	NaN	37.75	0.5	

	life_expectancy	human_development_index	population	\
0	64.83	0.51	41128772	
1	64.83	0.51	41128772	
2	64.83	0.51	41128772	
3	64.83	0.51	41128772	
4	64.83	0.51	41128772	
5	64.83	0.51	41128772	

6	64.83	0.51	41128772
7	64.83	0.51	41128772
8	64.83	0.51	41128772
9	64.83	0.51	41128772

excess_mortality_cumulative_absolute			
excess_mortality_cumulative \			
0		NaN	NaN
1		NaN	NaN
2		NaN	NaN
3		NaN	NaN
4		NaN	NaN
5		NaN	NaN
6		NaN	NaN
7		NaN	NaN
8		NaN	NaN
9		NaN	NaN

excess_mortality		excess_mortality_cumulative_per_million
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN
5	NaN	NaN
6	NaN	NaN
7	NaN	NaN
8	NaN	NaN
9	NaN	NaN

[10 rows x 67 columns]

3- Estadísticas básicas
df.describe()

	total_cases	new_cases	new_cases_smoothed	total_deaths \
count	4.118040e+05	4.101590e+05	4.089290e+05	4.118040e+05
mean	7.365292e+06	8.017360e+03	8.041026e+03	8.125957e+04
std	4.477582e+07	2.296649e+05	8.661611e+04	4.411901e+05
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	6.280750e+03	0.000000e+00	0.000000e+00	4.300000e+01

50%	6.365300e+04	0.000000e+00	1.200000e+01	7.990000e+02
75%	7.582720e+05	0.000000e+00	3.132900e+02	9.574000e+03
max	7.758668e+08	4.423623e+07	6.319461e+06	7.057132e+06

	new_deaths	new_deaths_smoothed	total_cases_per_million \
count	410608.000000	409378.000000	411804.000000
mean	71.852139	72.060828	112096.199420
std	1368.322990	513.636565	162240.412405
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	1916.100000
50%	0.000000	0.000000	29145.480000
75%	0.000000	3.140000	156770.190000
max	103719.000000	14817.000000	763598.600000

	new_cases_per_million	new_cases_smoothed_per_million \
count	410159.000000	408929.000000
mean	122.357073	122.713852
std	1508.778585	559.701663
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	2.790000
75%	0.000000	56.250000
max	241758.230000	34536.890000

	total_deaths_per_million	...	male_smokers
handwashing_facilities \			
count	411804.000000	...	243817.000000
161741.000000			
mean	835.514337	...	33.097758
50.649390			
std	1134.932641	...	13.853952
31.905236			
min	0.000000	...	7.700000
1.190000			
25%	24.570000	...	22.600000
20.860000			
50%	295.090000	...	33.100000
49.540000			
75%	1283.820000	...	41.500000
82.500000			
max	6601.110000	...	78.100000
100.000000			

	hospital_beds_per_thousand	life_expectancy
human_development_index \		
count	290689.000000	390299.000000
319127.000000		
mean	3.106895	73.702098
0.722178		
std	2.549168	7.387914

0.149237		
min	0.100000	53.280000
0.390000		
25%	1.300000	69.500000
0.600000		
50%	2.500000	75.050000
0.740000		
75%	4.210000	79.460000
0.830000		
max	13.800000	86.750000
0.960000		

	population	excess_mortality_cumulative_absolute	\
count	4.294350e+05		1.341100e+04
mean	1.520336e+08		5.604765e+04
std	6.975408e+08		1.568691e+05
min	4.700000e+01		-3.772610e+04
25%	5.237980e+05		1.765000e+02
50%	6.336393e+06		6.815200e+03
75%	3.296952e+07		3.912804e+04
max	7.975105e+09		1.349776e+06

	excess_mortality_cumulative	excess_mortality	\
count	13411.000000		13411.000000
mean	9.766431		10.925353
std	12.040658		24.560706
min	-44.230000		-95.920000
25%	2.060000		-1.500000
50%	8.130000		5.660000
75%	15.160000		15.575000
max	78.080000		378.220000

	excess_mortality_cumulative_per_million
count	13411.000000
mean	1772.666404
std	1991.892770
min	-2936.450000
25%	116.875000
50%	1270.800000
75%	2883.025000
max	10293.520000

[8 rows x 62 columns]

```
# !pip install skimpy
```

```
# 4 - Visualizacion
```

```
from skimpy import skim
```

```
skim(df)
```

skimpy summary

Data Summary

Data Types

Dataframe	Values	Column Type	Count
Number of rows	429435	float64	61
Number of columns	67	string	5
		int64	1

number








column p50	NA p75	NA % p100	mean hist	sd	p0	p25
total_c 63650 ases	17631 758300	4.10562 7759000 7161270 00 041	7365000 ■	44780000	0	6281
new_cas 0 es	19276 0	4.48868 4424000 8625752 0 4425	8017 ■	229700	0	0
new_cas 12 es_smo	20506 313.3	4.77511 6319000 1483693	8041 ■	86620	0	0
thed		69				
total_d 799 eaths	17631 9574	4.10562 7057000 7161270	81260 ■	441200	0	43

			041				
new_dea	18827	4.38413	71.85	1368	0	0	
0	0	103700	<div><div></div></div>				
ths		2639398					
		279					
new_dea	20057	4.67055	72.06	513.6	0	0	
0	3.14	14820	<div><div></div></div>				
ths_smo		5497339					
othed		527					
total_c	17631	4.10562	112100	162200	0	1916	
29150	156800	763600	<div><div></div></div>				
ases_pe		7161270					
r_milli		041					
on							
new_cas	19276	4.48868	122.4	1509	0	0	
0	0	241800	<div><div></div></div>				
es_per		8625752					
million		4425					
new_cas	20506	4.77511	122.7	559.7	0	0	
2.79	56.25	34540	<div><div></div></div>				
es_smoo		1483693					
thed_pe		69					
r_milli							
on							
total_d	17631	4.10562	835.5	1135	0	24.57	
295.1	1284	6601	<div><div></div></div>				
eaths_p		7161270					
er_mill		041					
ion							
new_dea	18827	4.38413	0.7623	6.983	0	0	
0	0	893.7	<div><div></div></div>				
ths_per		2639398					










_million		279					
n							
new_deaths	200570.36	4.67055127.75497339	0.7645	2.547	0	0	
othed_p		527					
er_million							
ion							
reproduction_ratio	2446181.14	56.96275.875338526	0.9115	0.3999	-0.07	0.72	
ate		203					
icu_patients	390319413	90.8912288908738924	661	2140	0	21	
		401					
icu_patients_per_million	3903196.4318.78	90.8912180.78738924	15.66	22.79	0	2.33	
er_million		401					
ion							
hosp_patients	3887797763051	90.53261545007665653	3912	9846	0	186	
		708					
hosp_patients_per_million	38877974.23159.8	90.532615277665653	126	151.2	0	31	
per_million		708					
lion							
weekly_92	418442353	97.44014838	317.9	514.4	0	17	

icu_adm		2481516					
issions		411					
weekly_4.64	41844212.65	97.4401225	9.672	13.57	0	1.55	
icu_adm		2481516					
issions		411					
_per_mi							
llion							
weekly_864	4049383893	94.2955154000	4292	10920	0	223	
hosp_ad		2784472					
mission		621					
s							
weekly_56.28	404938110	94.2955717.1	82.62	88.4	0	23.73	
hosp_ad		2784472					
mission		621					
s_per_m							
illion							
total_t	350048	81.5136	2110000	84100000	0	364700	
2067000	10250000	9214000					
ests		1672895000	0				
		781					
new_tes	354032	82.4413	67290	247700	1	2244	
8783	37230	3586000					
ts		47351750					
		288					
total_t	350048	81.5136	924.3	2195	0	43.59	
234.1	894.4	32930					
ests_pe		1672895					
r_thous		781					





and							
new_tes	354032	82.4413	3.272	9.034	0	0.29	
0.97	2.91	531.1	■				
ts_per		4735175					
thousan		288					
d							
new_tes	325470	75.7902	142200	1138000	0	1486	
6570	32200	1477000	■				
ts_smo		8258060					
thed		0					
		009					
new_tes	325470	75.7902	2.826	7.308	0	0.2	
0.85	2.58	147.6	■				
ts_smo		8258060					
thed_pe		009					
r_thous							
and							
positiv	333508	77.6620	0.09808	0.1161	0	0.02	
0.06	0.14	1	■				
e_rate		4431404					
		054					
tests_p	335087	78.0297	2404	33440	1	7.1	
17.5	54.6	1024000	■				
er_case		3674712					
		122					
total_v	344018	80.1094	5617000	18420000	0	1971000	
1439000	11620000	1358000	■				
accinat		4613270	00	00			
0	0	0000					
ions		925					
people	348303	81.1072	2487000	80060000	0	1050000	
6901000	50930000	5631000	■				
vaccina		6885326	00	0			
		000					
ted		068					

people	351374	81.8223	2287000	74040000	1	964400
6191000	47730000	5178000				
fully_v		9454166	00	0		
		000				
accinat		52				
ed						
total_b	375835	87.5184	1506000	43610000	1	602300
5765000	40190000	2817000				
oosters		8358890	00	0		
		000				
		17				
new_vac	358464	83.4734	739900	3183000	0	2010
20530	173600	4967000				
cinatio		0109678				
		0				
ns		997				
new_vac	234406	54.5847	283900	1922000	0	279
3871	31800	4369000				
cinatio		4507201				
		0				
ns_smoo		323				
thed						
total_v	344018	80.1094	124.3	85.1	0	44.77
130.6	195	410.2				
accinat		4613270				
ions_pe		925				
r_hundr						
ed						
people	348303	81.1072	53.5	29.38	0	27.88
64.3	77.78	129.1				
vaccina		6885326				
ted_per		068				
_hundre						
d						
people	351374	81.8223	48.68	29.04	0	21.22
57.92	73.61	126.9				

fully_v		9454166					
accinat		52					
ed_per							
hundred							
total_b	375835	87.5184	36.3	30.22	0	5.92	
35.91	57.62	150.5					
oosters		8358890					
_per_hu		17					
ndred							
new_vac	234406	54.5847	1851	3118	0	106	
605	2402	117100					
cinatio		4507201					
ns_smoo		323					
thed_pe							
r_milli							
on							
new_peo	237258	55.2488	106100	786700	0	43	
771	9307	2107000					
ple_vac		7351985					
0							
cinated		749					
_smooth							
ed							
new_peo	237258	55.2488	0.07468	0.1764	0	0	
0.01	0.07	11.71					
ple_vac		7351985					
cinated		749					
_smooth							
ed_per							
hundred							

stringency_index	233245 42.85 62.04	54.3143 100 8983781	42.88 	24.87	0	22.22
ex		015				
population_density	68943 88.12 222.9	16.0543 20550 5048377	394.1 	1785	0.14	37.73
sity		5192				
median_age	94772 29.7 38.7	22.0689 48.2 9763642	30.46 	9.094	15.1	22.2
		9263				
aged_65_and_over	106165 6.29 13.93	24.7220 27.05 1846612	8.684 	6.093	1.14	3.53
		4094				
aged_70_and_over	98120 3.87 8.64	22.8486 18.49 2668389	5.486 	4.136	0.53	2.06
		8612				
gdp_per_capita	101143 12290 27220	23.5525 116900 7489492	18900 	19830	661.2	4228
		007				
extreme_poverty	217439 2.5 21.4	50.6337 77.6 3968120	13.92 	20.07	0.1	0.6
y		903				
cardiovascular_deaths_rate	100570 245.5 333.4	23.4191 724.4 4375865	264.6 	120.8	79.37	175.7
th_rate		9633				
diabetes	83524 7.2 10.79	19.4497 30.53	8.556 	4.935	0.99	5.35

s_prevalence		4210299					
female_6.3 smokers	182270 19.3	42.4441 44	10.77	10.76	0.1	1.9	
male_smokers	185618 41.5	43.2237 78.1	33.1	13.85	7.7	22.6	
handwashing_facilities	267694 82.5	62.3363 100	50.65	31.91	1.19	20.86	
hospitals_per_1000	138746 4.21	32.3089 13.8	3.107	2.549	0.1	1.3	
life_expectancy	39136 75.05	9.11336 86.75	73.7	7.388	53.28	69.5	
human_development_index	110308 0.74	25.6867 0.96	0.7222	0.1492	0.39	0.6	
population	0 6336000	0 7975000	1520000	69750000	47	523800	
		000	00	0			

excess	416024	96.8770	56050	156900	-37730	176.5
6815	39130	1350000				
mortality		5939199				
ty_cumu		18				
lative						
absolut						
e						
excess	416024	96.8770	9.766	12.04	-44.23	2.06
8.13	15.16	78.08				
mortality		5939199				
ty_cumu		18				
lative						
excess	416024	96.8770	10.93	24.56	-95.92	-1.5
5.66	15.57	378.2				
mortality		5939199				
ty		18				
excess	416024	96.8770	1773	1992	-2936	116.9
1271	2883	10290				
mortality		5939199				
ty_cumu		18				
lative						
per_mil						
lion						
string						
chars	words per	total	shortest	longest	min	
column	NA	NA %	words			
max	per row	row				

[illegible]

1	continent	Our World in Data
2	location	Our World in Data
3	date	Our World in Data
4	total_cases	COVID-19 Dashboard by the WHO
5	new_cases	COVID-19 Dashboard by the WHO
6	new_cases_smoothed	COVID-19 Dashboard by the WHO
7	total_deaths	COVID-19 Dashboard by the WHO
8	new_deaths	COVID-19 Dashboard by the WHO
9	new_deaths_smoothed	COVID-19 Dashboard by the WHO

	category	description
0	Others	ISO 3166-1 alpha-3 – three-letter country code...
1	Others	Continent of the geographical location
2	Others	Geographical location
3	Others	Date of observation
4	Confirmed cases	Total confirmed cases of COVID-19. Counts can ...
5	Confirmed cases	New confirmed cases of COVID-19. Counts can in...
6	Confirmed cases	New confirmed cases of COVID-19 (7-day smoothe...
7	Confirmed deaths	Total deaths attributed to COVID-19. Counts ca...
8	Confirmed deaths	New deaths attributed to COVID-19. Counts can ...
9	Confirmed deaths	New deaths attributed to COVID-19 (7-day smoot...

6 y 7 - Verificar Metadatos(columnas)

```
missing_in_df = df_meta[~df_meta['column'].isin(df.columns)]
print("Columnas de los metadatos que NO están en nuestros datos:")
display(missing_in_df)
```

Columnas de los metadatos que NO están en nuestros datos:

```
Empty DataFrame
Columns: [column, source, category, description]
Index: []
```

Fase 1 — Identificación de los datos relevantes

Objetivo: Definir el alcance del análisis y elegir subconjuntos/variables relevantes.

Instrucciones:

1. Selecciona un **país** objetivo para el análisis (**target_country**). Toma en cuenta que no todos los países tienen todas las variables; documenta supuestos y decisiones.

He seleccionado Estados Unidos ya que, siendo una de las principales potencias mundiales, su respuesta a la pandemia resultó particularmente controvertida. El análisis se basará en examinar la evolución de la pandemia y la efectividad de las medidas implementadas. Objetivos del análisis:

1. Evolución temporal de la pandemia: Analizar la progresión de casos confirmados, muertes y hospitalizaciones a lo largo del tiempo para identificar las diferentes olas de contagio.
2. Impacto del sistema sanitario: Examinar la ocupación hospitalaria y de UCI para evaluar el colapso del sistema de salud mencionado inicialmente.
3. Efectividad de las políticas públicas: Correlacionar el índice de rigurosidad gubernamental con la evolución de casos y muertes para evaluar la efectividad de las medidas implementadas.
4. Análisis de vacunación: Estudiar el proceso de vacunación y su impacto en la reducción de casos y muertes.
5. Mortalidad excesiva: Evaluar el verdadero impacto de la pandemia mediante el análisis de la mortalidad excesiva comparada con años anteriores.

```
target_country = 'United States'
#compare_countries = ['Germany', 'France', 'Spain'] # (Opcional) para comparación
```

1. Define un **rango de fechas**

En base a: https://grok.com/share/c2hhcmQtMg%3D%3D_97b68925-08c1-4e0a-853a-6f25e12566d7 tomaremos el año de la pandemia del 20 de Enero del 2020 al 5 de Mayo del 2023

```
date_start = '2020-01-20'
date_end = '2023-05-05'
```

1. Elige variables de interés (mínimo: **casos** y **vacunación**; opcional: **UCI**, **pruebas**, **población**).

```
base_columns = [
    'date', 'location', 'total_cases', 'new_cases_smoothed',
    'total_deaths', 'new_deaths_smoothed', 'total_cases_per_million',
```

```

    'total_deaths_per_million', 'hosp_patients_per_million',
    'weekly_icu_admissions_per_million',
    'stringency_index', 'people_vaccinated_per_hundred',
    'people_fully_vaccinated_per_hundred',
    'total_boosters_per_hundred', 'excess_mortality_cumulative',
    'excess_mortality_cumulative_per_million',
    'population', 'gdp_per_capita', 'hospital_beds_per_thousand'
]

```

1. Genera una tabla con los metadatos de las variables de interés

```

meta_base = df_meta[df_meta['column'].isin(base_columns)]
display(meta_base)

```

	column \		source
2	location		
3	date		
4	total_cases		
6	new_cases_smoothed		
7	total_deaths		
9	new_deaths_smoothed		
10	total_cases_per_million		
13	total_deaths_per_million		
20	hosp_patients_per_million		
22	weekly_icu_admissions_per_million		
41	people_vaccinated_per_hundred		
42	people_fully_vaccinated_per_hundred		
43	total_boosters_per_hundred		
47	stringency_index		
48	population		
53	gdp_per_capita		
60	hospital_beds_per_thousand		
64	excess_mortality_cumulative		
66	excess_mortality_cumulative_per_million		
	category \		
2		Our World in Data	
Others			
3		Our World in Data	
Others			
4	COVID-19 Dashboard by the WHO	Confirmed	
cases			
6	COVID-19 Dashboard by the WHO	Confirmed	
cases			
7	COVID-19 Dashboard by the WHO	Confirmed	
deaths			
9	COVID-19 Dashboard by the WHO	Confirmed	
deaths			
10	COVID-19 Dashboard by the WHO	Confirmed	

cases		
13	COVID-19 Dashboard by the WHO	Confirmed
deaths		
20	National government reports and European CDC	Hospital &
ICU		
22	National government reports and European CDC	Hospital &
ICU		
41	National government reports	
Vaccinations		
42	National government reports	
Vaccinations		
43	National government reports	
Vaccinations		
47	Oxford COVID-19 Government Response Tracker, B...	Policy
responses		
48	United Nations, Department of Economic and Soc...	
Others		
53	World Bank World Development Indicators, sourc...	
Others		
60	OECD, Eurostat, World Bank, national governmen...	
Others		
64	Human Mortality Database (2021), World Mortali...	Excess
mortality		
66	Human Mortality Database (2021), World Mortali...	Excess
mortality		
		description
2	Geographical location	
3	Date of observation	
4	Total confirmed cases of COVID-19. Counts can ...	
6	New confirmed cases of COVID-19 (7-day smoothe...	
7	Total deaths attributed to COVID-19. Counts ca...	
9	New deaths attributed to COVID-19 (7-day smoot...	
10	Total confirmed cases of COVID-19 per 1,000,00...	
13	Total deaths attributed to COVID-19 per 1,000,...	
20	Number of COVID-19 patients in hospital on a g...	
22	Number of COVID-19 patients newly admitted to ...	
41	Total number of people who received at least o...	
42	Total number of people who received all doses ...	
43	Total number of COVID-19 vaccination booster d...	
47	Government Response Stringency Index: composit...	
48	Population (latest available values). See http...	
53	Gross domestic product at purchasing power par...	
60	Hospital beds per 1,000 people, most recent ye...	
64	Percentage difference between the cumulative n...	
66	Cumulative difference between the reported num...	

1. Inserta en un nuevo chunk el código que necesites para completar las tareas

```
# Filtrado por fechas y columnas
df = df.loc[(df['date'] >= date_start) & (df['date'] <= date_end),
base_columns].copy()
print('Dimensión tras selección por fechas y columnas:', df.shape)

# Dataset país objetivo
df_country = df.loc[df['location'] == target_country].copy()
print(f'Registros de {target_country}:', df_country.shape[0])
display(df_country.head())

plt.figure(figsize=(10,4))
plt.plot(df_country['date'],
df_country['people_fully_vaccinated_per_hundred'])
plt.title(f'{target_country}: Personas completamente vacunadas')
plt.xlabel('Fecha'); plt.ylabel('Personas por cada 100 habitantes')
plt.show()
```

Dimensión tras selección por fechas y columnas: (309652, 19)
Registros de United States: 1202

	date	location	total_cases	new_cases_smoothed	\
403466	2020-01-20	United States	0.0	0.0	
403467	2020-01-21	United States	0.0	0.0	
403468	2020-01-22	United States	0.0	0.0	
403469	2020-01-23	United States	0.0	0.0	
403470	2020-01-24	United States	0.0	0.0	

	total_deaths	new_deaths_smoothed	total_cases_per_million	\
403466	0.0	0.0	0.0	
403467	0.0	0.0	0.0	
403468	0.0	0.0	0.0	
403469	0.0	0.0	0.0	
403470	0.0	0.0	0.0	

	total_deaths_per_million	hosp_patients_per_million	\
403466	0.0	NaN	
403467	0.0	NaN	
403468	0.0	NaN	
403469	0.0	NaN	
403470	0.0	NaN	

	weekly_icu_admissions_per_million	stringency_index	\
403466	NaN	0.0	
403467	NaN	0.0	
403468	NaN	0.0	
403469	NaN	0.0	
403470	NaN	0.0	

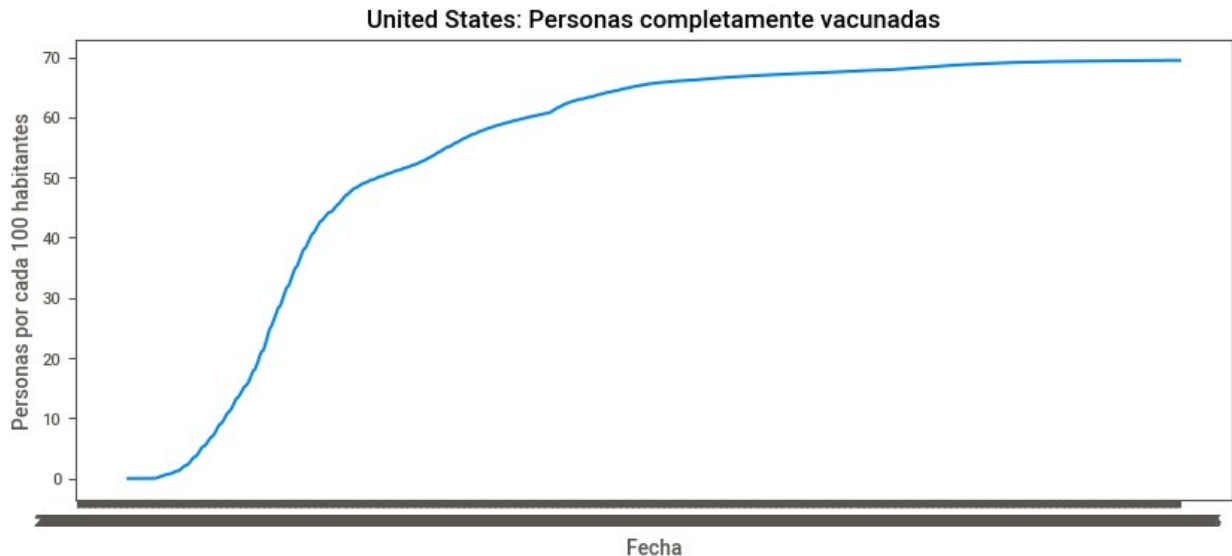
	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	\
--	-------------------------------	-------------------------------------	---

403466	NaN
NaN	
403467	NaN
NaN	
403468	NaN
NaN	
403469	NaN
NaN	
403470	NaN
NaN	

	total_boosters_per_hundred	excess_mortality_cumulative	\
403466	NaN		NaN
403467	NaN		NaN
403468	NaN		NaN
403469	NaN		NaN
403470	NaN		NaN

	excess_mortality_cumulative_per_million	population	gdp_per_capita	\
403466		NaN	338289856	
54225.45				
403467		NaN	338289856	
54225.45				
403468		NaN	338289856	
54225.45				
403469		NaN	338289856	
54225.45				
403470		NaN	338289856	
54225.45				

	hospital_beds_per_thousand
403466	2.77
403467	2.77
403468	2.77
403469	2.77
403470	2.77



Fase 2 — Limpieza de los datos

Objetivo: Detectar y tratar valores faltantes/inconsistencias.

Checklist:

- ☐ ¿Existen columnas con alta proporción de NaN?, ¿Es necesario eliminarlas?
- ☐ ¿Existen variables con ceros estructurales (p. ej., antes del inicio de vacunación)?
- ☐ ¿Es razonable hacer una Interpolación/`ffill` para la variable de interés?
- ☐ Es necesario realizar algún otro tipo de preprocesamiento? Sí/no/por qué
- ☐ ¿Qué podemos hacer con los casos diarios?
- ☐ En caso de ser necesario, realiza cualquier otro tipo de preprocesamiento que consideres pertinente
- ☐ Documenta en una celda de **Markdown** tus decisiones y por qué.

```
# Exploración de valores faltantes

# Ejemplo: forward-fill para variables acumulativas/lentas
for col in
['people_vaccinated', 'people_fully_vaccinated', 'icu_patients']:
    if col in df_country.columns:
        df_country[col] = df_country[col].fillna(method='ffill')

# Ejemplo: reemplazo de NaN en casos suavizados con 0 (justifica esta
decisión)
if 'new_cases_smoothed' in df_country.columns:
    df_country['new_cases_smoothed'] =
df_country['new_cases_smoothed'].fillna(0)

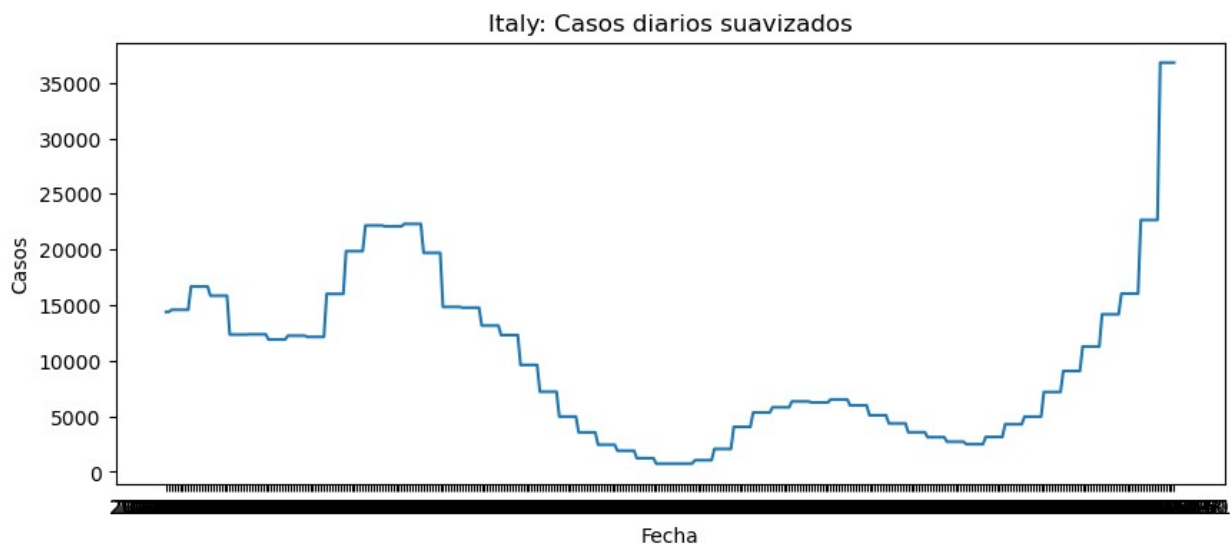
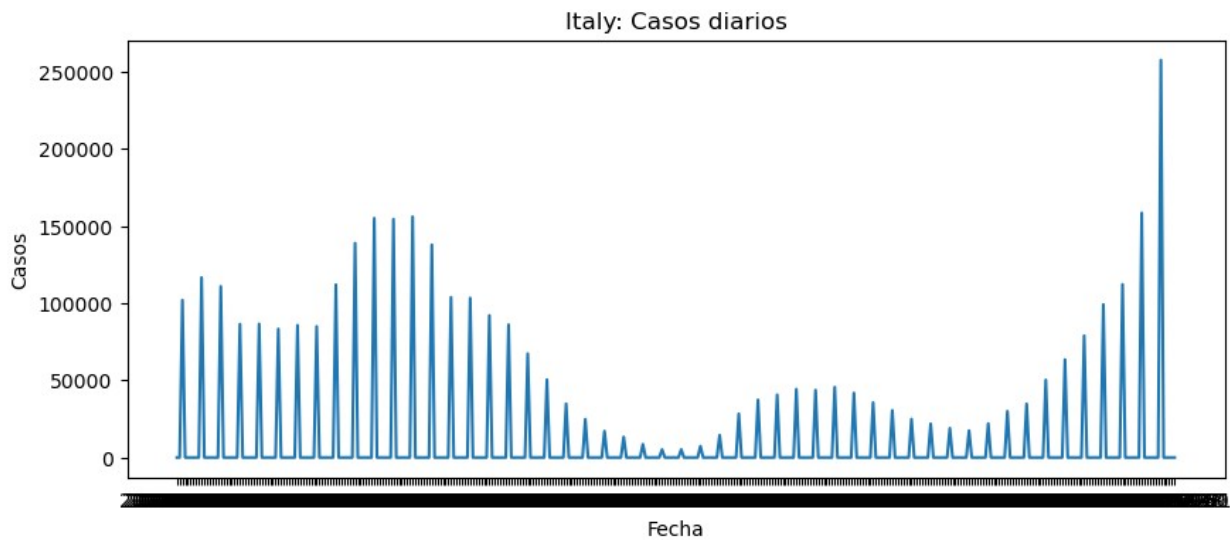
# Casos diarios
plt.figure(figsize=(10,4))
plt.plot(df_country['date'], df_country['new_cases'])
```



```
plt.title(f'{target_country}: Casos diarios')
plt.xlabel('Fecha'); plt.ylabel('Casos')
plt.show()
```

```
# Casos diarios suavizados
plt.figure(figsize=(10,4))
plt.plot(df_country['date'], df_country['new_cases_smoothed'])
plt.title(f'{target_country}: Casos diarios suavizados')
plt.xlabel('Fecha'); plt.ylabel('Casos')
plt.show()
```

```
/tmp/ipykernel_12399/818383705.py:6: FutureWarning: Series.fillna with
'method' is deprecated and will raise in a future version. Use
obj.ffill() or obj.bfill() instead.
  df_country[col] = df_country[col].fillna(method='ffill')
```



Diario de decisiones (Preprocesamiento):

- Describe qué columnas imputaste, con qué técnica y por qué.
- Señala riesgos de introducir sesgos (p. ej., rellenar con 0 vs interpolar).
- Indica qué filas/columnas eliminaste (si aplica) y el impacto esperado.

Fase 3 — Transformación y reducción

Objetivo: Crear variables derivadas, transformar escalas para análisis posterior y reducir la dimensión del banco de datos

Sugerencias de variables derivadas:

- `cases_per_million_proxy = new_cases_smoothed / (population/1e6)`

Checklist:

- ☐ ¿Es necesario transformar algún tipo de dato? Sí/no/Por qué
- ☐ Definir qué variables necesito transformar y por qué
- ☐ Definir una forma de calcular la tasa de vacunación anual
- ☐ ¿Cuándo puedo utilizar `MinMaxScaler` o `StandardScaler`? Define el caso y justifica el por qué

```
# Variables derivadas (Modifica/Extiende)
for col in
['people_fully_vaccinated', 'population', 'new_cases_smoothed']:
    assert col in df_country.columns, f'Columna faltante: {col}'

df_country['cases_per_million_proxy'] =
df_country['new_cases_smoothed'] / (df_country['population'] /
1_000_000)
```

```
display(df_country.head())
```

	iso_code	continent	location	date	new_cases
new_cases_smoothed \					
185637	ITA	Europe	Italy	2021-01-01	0.0
14385.71					
185638	ITA	Europe	Italy	2021-01-02	0.0
14385.71					
185639	ITA	Europe	Italy	2021-01-03	102019.0
14574.14					
185640	ITA	Europe	Italy	2021-01-04	0.0
14574.14					
185641	ITA	Europe	Italy	2021-01-05	0.0
14574.14					

	people_vaccinated	people_fully_vaccinated	icu_patients
population \			
185637	51939.0	NaN	2553.0
59037472			

185638	91012.0	NaN	2569.0
59037472			
185639	126889.0	NaN	2583.0
59037472			
185640	196562.0	9.0	2579.0
59037472			
185641	276867.0	11.0	2569.0
59037472			
cases_per_million_proxy			
185637	243.670833		
185638	243.670833		
185639	246.862535		
185640	246.862535		
185641	246.862535		

Fase 4 — Minería de datos

Objetivo: Aplicar al menos una técnica de minería y **explicar** resultados.

A) — Series de tiempo (correlaciones y picos)

- Correlación entre `new_cases_smoothed` y `people_fully_vaccinated`
- Identificación de **picos** en casos (usa un umbral manual sencillo si no tienes librerías extra).

B) — Clustering entre países (patrones de 2021)

- Construye una matriz país × fecha con una métrica comparable (p. ej., `delta_vaccinated_norm`).
- Aplica **K-Means** (parámetro `k` modificable) y calcula **Silhouette Score**.
- (Opcional) Prueba **DBSCAN** y comenta diferencias.

A) correlaciones y picos (Experimenta)

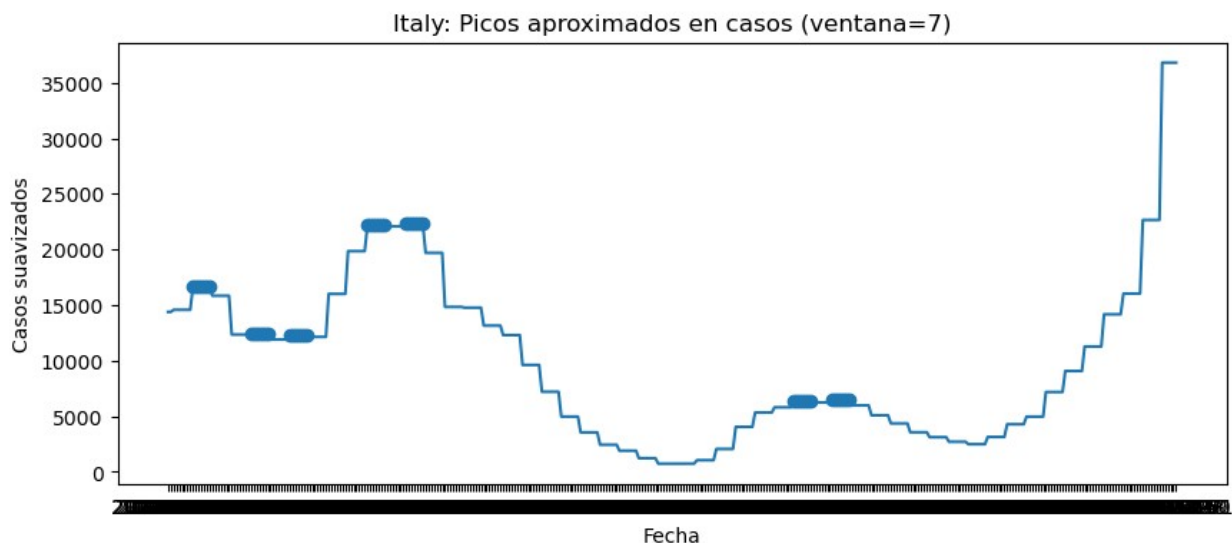
```
corr =
df_country[['new_cases_smoothed', 'people_fully_vaccinated']].corr()
print('Correlaciones (país objetivo):')
display(corr)

# Detección simple de picos en casos: valores locales mayores que
# vecinos
window = 7 # (Modifica)
series = df_country['new_cases_smoothed'].values
peaks_idx = []
for i in range(window, len(series)-window):
    if series[i] == max(series[i-window:i+window+1]) and series[i] >
0:
        peaks_idx.append(i)
```

```
plt.figure(figsize=(10,4))
plt.plot(df_country['date'], series)
plt.scatter(df_country['date'].iloc[peaks_idx], series[peaks_idx])
plt.title(f'{target_country}: Picos aproximados en casos
(ventana={window})')
plt.xlabel('Fecha'); plt.ylabel('Casos suavizados')
plt.show()
```

Correlaciones (país objetivo):

	new_cases_smoothed	people_fully_vaccinated
new_cases_smoothed	1.000000	-0.342738
people_fully_vaccinated	-0.342738	1.000000



B) Clustering entre países para identificar las características similares entre los que tienen mayor tasa de vacunación (Desarrolla)

Fase 5 — Evaluación e interpretación

Objetivo: Evaluar resultados con métricas y discutir **limitaciones**.

- Para clustering, reporta **Silhouette Score** y comenta si los grupos tienen **sentido**.
- Para series de tiempo, discute el **desfase temporal** entre vacunación y cambios en casos/uci.
- Señala **sesgos**: definición de caso, cambios de prueba, retrasos de reporte, diferencias demográficas.

- Presenta una tabla con la comparación en la tasa de vacunación anual para todos los años de la pandemia
- Genera una gráfica que muestre cómo fue cambiando, a lo largo de la pandemia, el número de personas totalmente vacunadas

Reflexión (responde)

1. **Preprocesamiento:** ¿Qué estrategia de imputación funcionó mejor y por qué?
2. **Transformación:** ¿Qué variables derivadas aportaron mayor valor analítico?
3. **Minería:** ¿Cómo cambia el resultado al variar **k** (K-Means) o el **window** para picos?
4. **Evaluación:** ¿Qué valor arrojó Silhouette Score y cómo lo interpretas en este contexto?
5. **Limitaciones:** Enumera al menos **3** limitaciones del dataset o del enfoque utilizado.

Reto

Con el banco de datos original, intenta obtener conocimiento extra sobre la relación vacunación/muertes o algún otro tema que consideres relevante (con las herramientas que conoces hasta ahora)