Expansión Estratégica de Biogenesys con Python

Alumno: Hillebrand, Francisco Javier

Email: franhillebrand@gmail.com

Cohorte: Data Analytics Part-Time 09

Fecha de Entrega: 10/10/2025

Institución: Biogenesys

ÍNDICE

Introducción **3**
EDA e Insights **3**
Propiedades Estadísticas del Dataset 3
Matriz de Correlaciones 4
Visualización de algunos Insights 6
Promedio de casos nuevos por país 6
Comparación de casos confirmados, densidad poblacional y dosis administradas por país 6
Correlación entre nuevos casos confirmados, aumento de las muertes y aumento de la población a lo largo del tiempo 7
Comparación de la estructura sanitaria del país 8
Comparación del porcentaje de mortalidad femenina y masculina 9
Identificación de Tendencias y Patrones 9
Evolución de dosis administradas por mes de cada país 9
Casos nuevos vs. temperatura promedio 10
Infraestructura médica vs. mortalidad 11
Relación entre infraestructura médica y dosis administradas 12
Relación entre vacunación, aumento y recuperación de afectados 12
Correlación entre fallecimientos de COVID-19, fallecimientos por comorbilidad, el porcentaje población preponderante a la diabetes y el porcentaje de fumadores en la población 14
Análisis Temporal 14
Tasa de cambio de casos confirmados en general 14
Promedio Móvil de Casos Confirmados por País 15
Tasa de cambio de fallecimientos en general 17
Promedio Móvil de Muertes por País 18
Análisis Espacial 21
Total de muertes por País 21
Total de Casos Confirmados por País 21
Otras Correlaciones 22
Relación entre casos confirmados, fallecimientos y las población urbana y rural 22
Relación entre la la acumulación de fallecimientos y casos confirmados en las áreas urbanas y rurales 23
Análisis del Dashboard **23**
Desarrollo del Proyecto **24**
AVANCE N°1 24
Diccionario y Limpieza de Datos 24
Importar librerías 26
Leer Archivo 27
Compruebo la cantidad de registros y columnas 27
Filtro el Dataframe para obtener solo los registros de los países donde Biogenesys busca expandirse 27
Filtra los datos en fechas mayores a 2021-01-01 27
Trabajo con valores nulos y faltantes 27

Identificar variables claves..... 28
Guardar los datos filtrados en un archivo CSV..... 29
Calculo estadísticas descriptivas..... 29
Función que obtiene la mediana, varianza y rango de un conjunto de valores..... 30
AVANCE N°2..... 31
Importar librerías..... 31
Análisis Estadístico..... 31
Visualización de Datos con Matplotlib y Seaborn..... 32
Promedio de Casos Confirmados por país..... 32
Comparación de casos confirmados por país y su densidad poblacional..... 33
Correlación entre nuevos casos confirmados, aumento de las muertes y aumento de la población a lo largo del tiempo..... 35
Comparación de la estructura sanitaria del país..... 36
Comparación del porcentaje de mortalidad femenina y masculina..... 38
Identificación de Tendencias y Patrones..... 38
Evolución de dosis administradas por mes de cada país..... 39
Casos nuevos vs. temperatura promedio..... 40
Infraestructura médica vs. mortalidad..... 40
Relación entre infraestructura médica y dosis administradas..... 41
Relación entre vacunación, aumento y recuperación de afectados..... 42
AVANCE N°3..... 43
Importar Librerías..... 43
EDA..... 43
Análisis Temporal..... 43
Tasa de cambio de casos confirmados en general..... 44
Promedio Móvil de Casos Confirmados por País y Promedio Móvil de Muertes por País..... 44
Análisis Espacial..... 46
Total de muertes por País..... 46
Total de Casos Confirmados por País..... 46
Otras Correlaciones..... 47
Relación entre casos confirmados, fallecimientos y las población urbana y rural..... 47
Relación entre la la acumulación de fallecimientos y casos confirmados en las áreas urbanas y rurales..... 47
AVANCE N°4..... 47
Conexión de Python con Power BI..... 47
Predicción de nuevos casos..... 49
Predicción de fallecimientos..... 52
Creación de Dashboards en Power BI..... 53
Visualizaciones Estáticas vs Interactivas..... 53
Conclusiones y Recomendaciones..... 54
Reflexión Personal..... 56

Introducción

El presente informe detalla el proceso y los resultados del análisis estratégico enfocado en la incidencia de COVID-19 en los países hacia los que pretende expandirse Byogenesis. El propósito central fue transformar datos brutos en recomendaciones accionables para informar decisiones críticas relacionadas con la salud pública y, específicamente en lo que respecta a la ubicación estratégica de futuros laboratorios y centros de vacunación.
Para alcanzar este objetivo, el proyecto se estructuró de tal manera que se pudieron cumplir los siguientes logros organizacionales clave:

- Calidad y Acceso a Datos (ETL y Limpieza): Se realizaron operaciones eficientes de Extracción, Transformación y Carga (ETL), aplicando técnicas de limpieza de datos para estandarizar formatos, manejar valores faltantes y asegurar que los datos fueran de alta calidad y completamente confiables para análisis posteriores.
- Análisis Exploratorio y Descubrimiento de Insights: Se completó un análisis exploratorio de datos (EDA) exhaustivo. Mediante el uso de estadísticas, mediciones (como la correlación) y visualizaciones, se lograron identificar tendencias claras en la incidencia de COVID-19, revelando oportunidades estratégicas para la intervención.
- Desarrollo de Herramientas para la Decisión: Se desarrollaron visualizaciones eficientes y dashboards interactivos (como se demuestra en el anexo de gráficos comparativos y mapas de calor) que permiten a los tomadores de decisiones explorar los datos desde múltiples perspectivas, facilitando una toma de decisiones informada y ágil sobre la demanda de vacunas, la logística de distribución y la infraestructura sanitaria requerida en los países analizados.

En resumen, este análisis ofrece insights valiosos para una expansión estratégica y sostenible de las operaciones de Byogenesis.

EDA e Insights

Propiedades Estadísticas del Dataset

En el apartado Análisis Estadístico se hizo un análisis de medidas de tendencia central y dispersión de las variables del dataset, y podemos destacar algunos insights:

1. Si tenemos en cuenta el promedio y la mediana de la acumulación de casos confirmados y la aparición de nuevos casos por día, Argentina y Brasil son aquellos que más casos acumulados y en promedio mayor cantidad de casos nuevos por día registran:
    a. Analizando el caso de Argentina, puede que esto tenga que ver con que fue el país al que menos vacunas se le administró en promedio, por ende el contagio fue mayor.
    b. En el caso de Brasil, puede que sea por el hecho de que es, por diferencia, el país con mayor población. A más población, más contagios.
2. Hay que observar el caso de México, ya que siendo el país con mayor densidad poblacional (lo que puede facilitar el contagio del virus), está entre los primeros 3 países que, en promedio, más casos confirmados y fallecimientos tuvieron. También por eso es el país que más vacunas administró, en promedio, detrás de Brasil.

3. Teniendo en cuenta la cantidad de muertes por día, y el acumulado de muertes a través del tiempo, Brasil es el país con los números más altos en estas variables. Teniendo esto en cuenta, hay que recordar que es el país que más contagios diarios registró.
4. Anteriormente mencionamos a Brasil y Argentina, ambos con muchos casos nuevos por día, y en el caso de Brasil, es el país que más fallecidos tuvo. Sin embargo, son dos de los primeros cuatro países con un mayor PBI per cápita, por lo que podrían costear más vacunas.
5. También es importante destacar que Brasil y Chile son los dos países que más enfermeros tienen cada 1.000 habitantes, un dato importante para la administración de vacunas.
6. Si bien Argentina fue uno de los países más afectados como se dijo en el primer insight, hay que tener en cuenta que es el país con la mayor cantidad promedio de médicos cada 1.000 habitantes, un dato importante para la administración de vacunas.

Si desea ver las medidas de cada variable por país, puede hacerlo en Stats por país.

Matriz de Correlaciones

A partir de la Matriz de Correlaciones generada en el apartado Análisis Estadístico, puedo resaltar algunos insights:

1. Existe una fuerte correlación positiva entre los casos positivos acumulados de COVID-19 y las muertes acumuladas por COVID-19, lo que indica que en las regiones o períodos con mayor número de contagios también se registran más muertes.
   Esto se percibe ya que las variables cumulative_confirmed y cumulative_deceased tienen una correlación positiva de 0,90.
2. Los casos positivos acumulados y las muertes acumuladas aumentan en todas las edades a lo largo del tiempo, pero un poco más en personas de 60 años o más. Esto se percibe en la siguiente imagen.

	cumulative_confirmed	cumulative_deceased
human_development_index	0,22	0,44
population_age_00_09	0,67	0,74
population_age_10_19	0,69	0,75
population_age_20_29	0,77	0,81
population_age_30_39	0,80	0,83
population_age_40_49	0,81	0,84
population_age_50_59	0,84	0,85
population_age_60_69	0,86	0,86
population_age_70_79	0,86	0,85
population_age_80_and_older	0,86	0,85

   También se puede destacar que el número de recuperados aumenta especialmente en la población de personas mayores a 40 años.

	cumulative_recovered
human_development_index	0,20
population_age_00_09	0,68
population_age_10_19	0,70
population_age_20_29	0,77
population_age_30_39	0,79
population_age_40_49	0,81
population_age_50_59	0,83
population_age_60_69	0,84
population_age_70_79	0,84
population_age_80_and_older	0,84

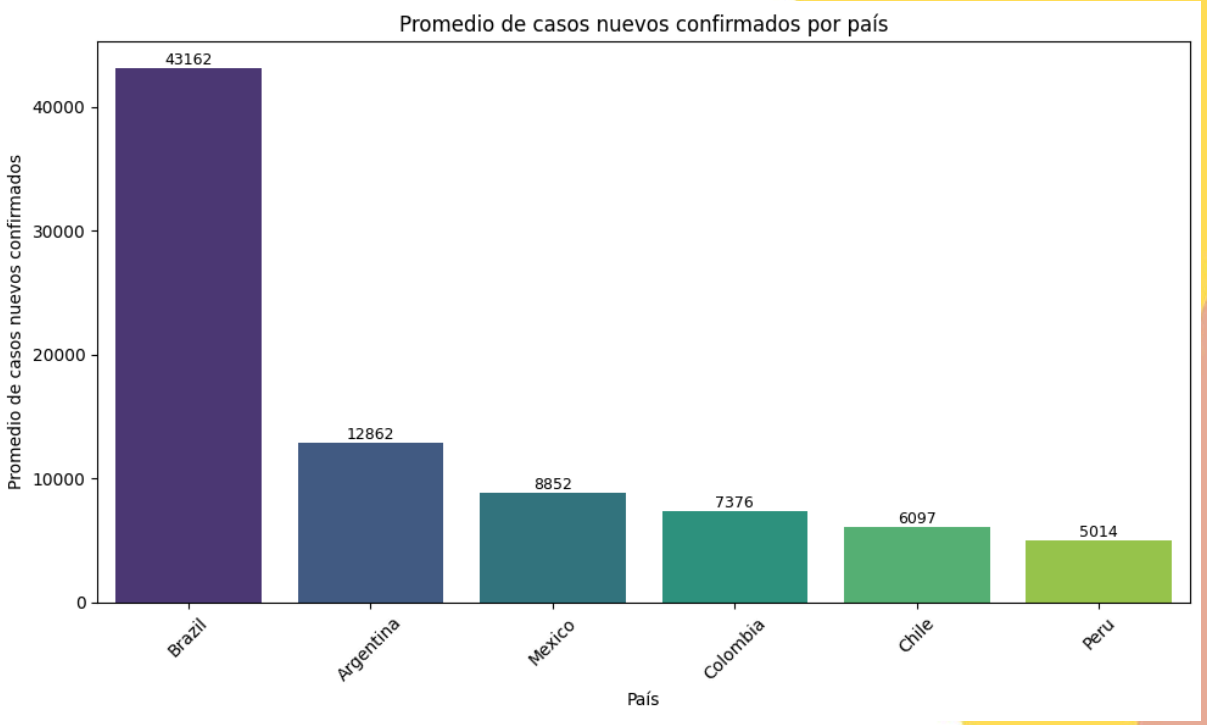
3. Existe una fuerte correlación positiva entre los casos positivos acumulados y los recuperados acumulados, lo cual refleja que en las regiones o períodos con mayor número de contagios también se registran más personas que lograron recuperarse.
   Esto se percibe ya que las variables cumulative_confirmed y cumulative_recovered tienen una correlación positiva de 0,95.
4. Existe una fuerte correlación positiva entre las muertes acumuladas y los recuperados acumulados, lo cual refleja que en las regiones o períodos con mayor cantidad de muertes también se registran más personas que lograron recuperarse.
   Esto se percibe ya que las variables cumulative_deceased y cumulative_recovered tienen una correlación positiva de 0,90.
5. A medida que aumenta la población de ambos géneros, aumenta el acumulado de casos positivos para ambos por igual.
   Esto se percibe ya que la correlación positiva entre cumulative_confirmed y las variables population_male y population_female es de 0,78 para ambos.
6. A medida que aumenta la población de ambos géneros, aumenta el acumulado de muertes para ambos por igual.
   Esto se percibe ya que la correlación positiva de cumulative_deceased y las variables population_male y population_female es de 0,82 para ambos.
7. A medida que aumenta la población de ambos géneros, aumenta el acumulado de recuperados para ambos por igual. Viendo los dos insights anteriores, vemos que el número de recuperados crece menos o igual con respecto a los casos confirmados y las muertes, siempre y cuando comparemos la correlación positiva de estas variables con la población de ambos géneros.
   Esto se percibe ya que la correlación positiva de cumulative_recovered y las variables population_male y population_female es de 0,78 para ambos.
8. Se puede destacar que la administración de vacunas aumenta al mismo tiempo que aumenta el acumulado de recuperados.
   Esto se percibe ya que la correlación positiva entre cumulative_vaccine_doses_administered y cumulative_recovered es de 0,88.

Si desea ver las correlaciones entre todas las variables puede acceder al documento correlaciones_byogenesis

Visualización de algunos Insights

Promedio de casos nuevos por país

En este caso vemos lo anteriormente mencionado, Brasil y Argentina son los países que en promedio han tenido la mayor cantidad de nuevos casos por día.

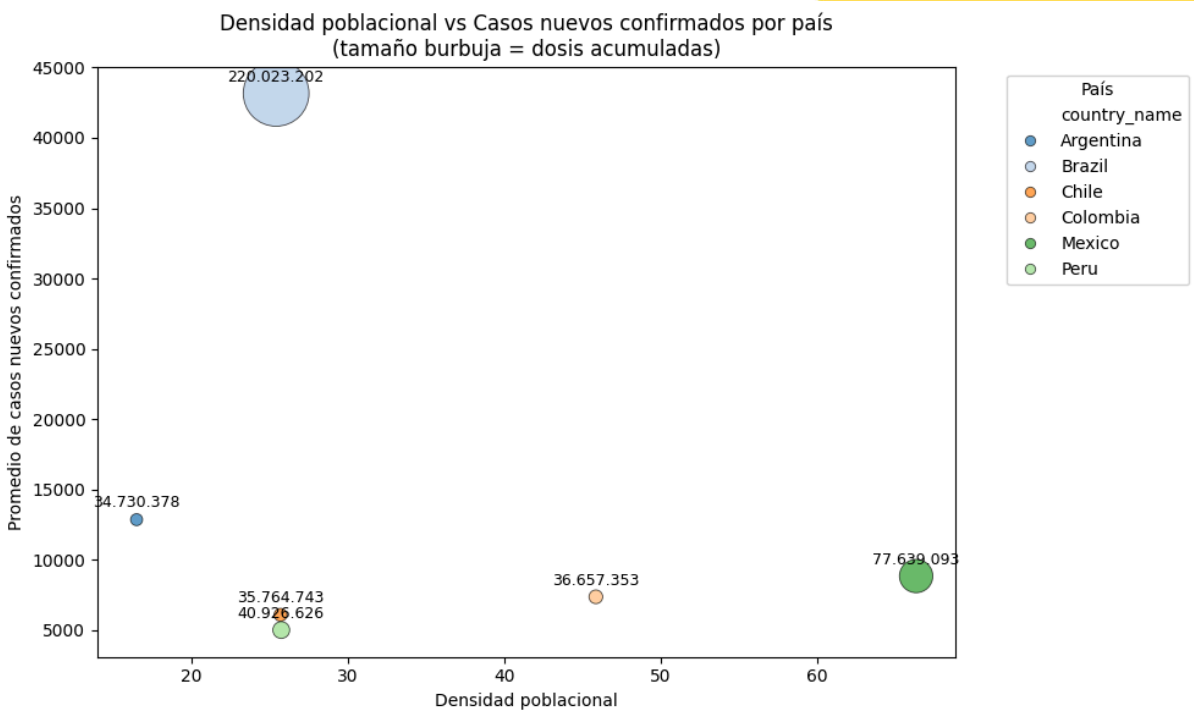


Comparación de casos confirmados, densidad poblacional y dosis administradas por país

Vemos que los países que tienen mayor densidad poblacional no son aquellos que más casos confirmados por día tienen, la densidad poblacional no parece ser determinante para el contagio del virus.

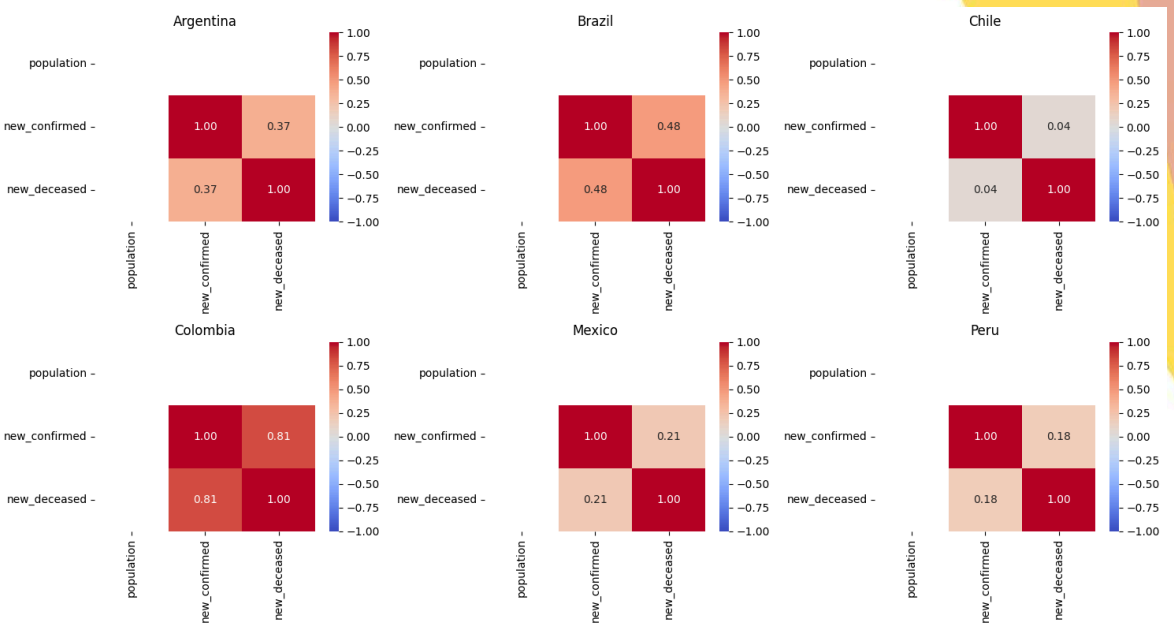
Si tenemos en cuenta también la administración de vacunas, vemos en el siguiente gráfico, que México fue el segundo país que más vacunas administró por lo que no aumentaron los casos.

Sigue siendo interesante el caso de Brasil, donde se administran más vacunas, y aún así han tenido el promedio más alto de casos confirmados por día, por lo que en este país puede haber una oportunidad para ofrecer vacunas. Otro punto a destacar, es que se le debe dar prioridad a la Argentina, es el país al que menos vacunas se le administraron, y el número promedio de casos confirmados por día es de los más altos.

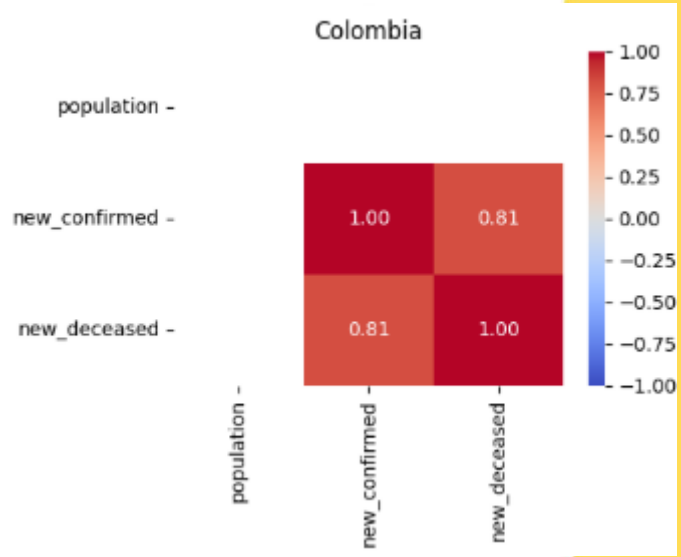


Correlación entre nuevos casos confirmados, aumento de las muertes y aumento de la población a lo largo del tiempo

En este caso hice un heatmap para evaluar la correlación entre las variables new_confirmed, new_deceased y population por país

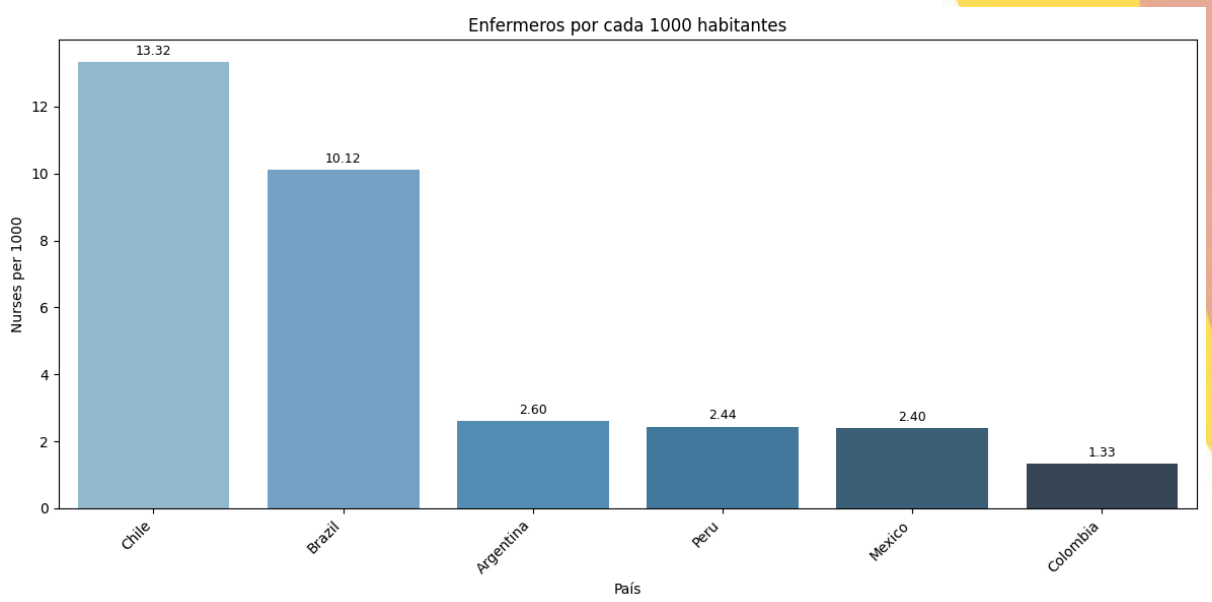


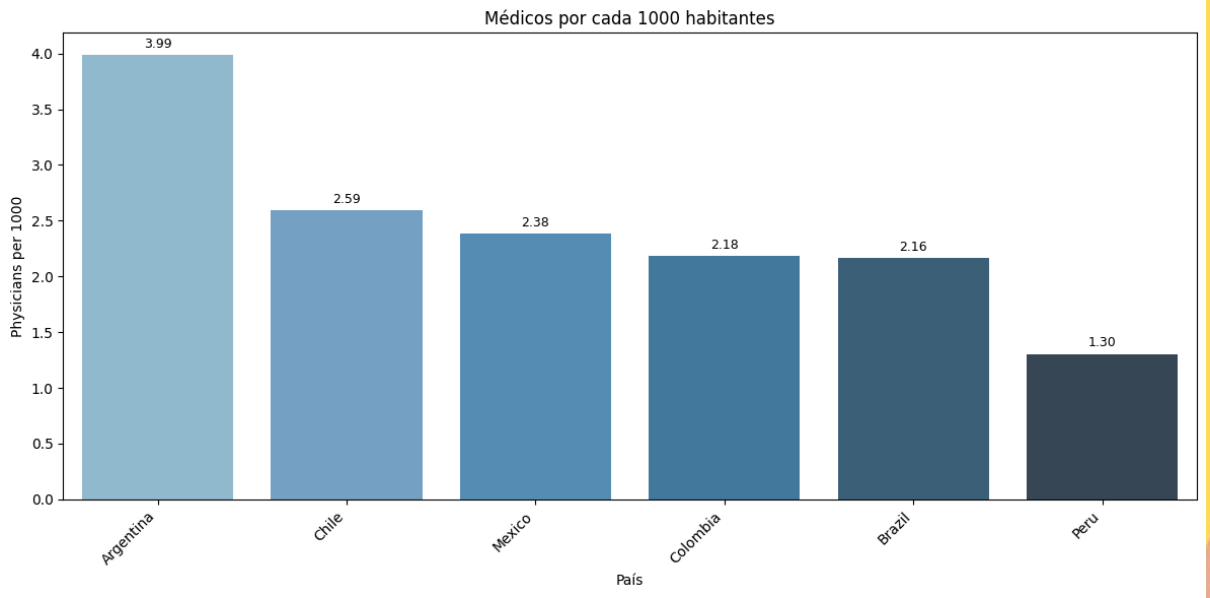
Lo más destacable aquí es que en Colombia hay una fuerte correlación positiva entre los casos confirmados y el número de muertes por día, por ende, estas dos variables aumentan a la vez, una correlación que no es tan fuerte en los demás países.



Comparación de la estructura sanitaria del país

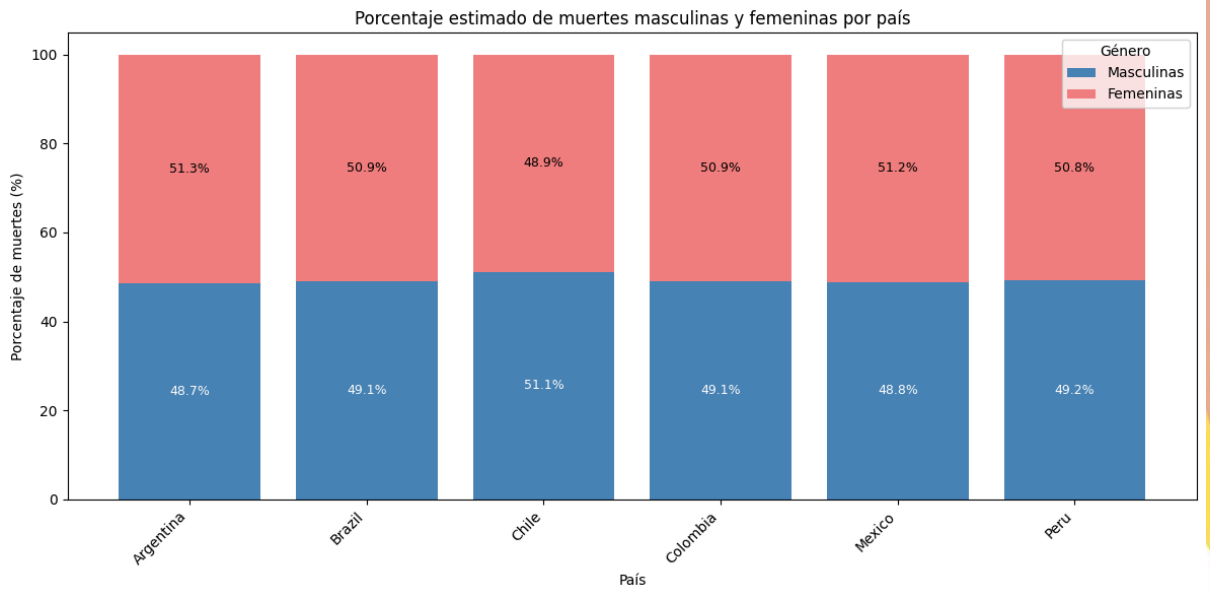
Hay que destacar la situación de Chile, es el primer país con mayor cantidad promedio de enfermeros por 1000 habitantes, y el segundo en cantidad promedio de médicos por 1000 habitantes, quizás por ello tuvo menor cantidad de casos confirmados por día. Además es interesante tener esto en cuenta para la administración de vacunas.
Brasil fue el país que más dosis administró y puede que tenga que ver con qué tiene uno de los mejores números promedio de médicos por 1000 habitantes. Y además, es el segundo país en cuanto a la cantidad de enfermeros por 1000 habitantes.





Comparación del porcentaje de mortalidad femenina y masculina

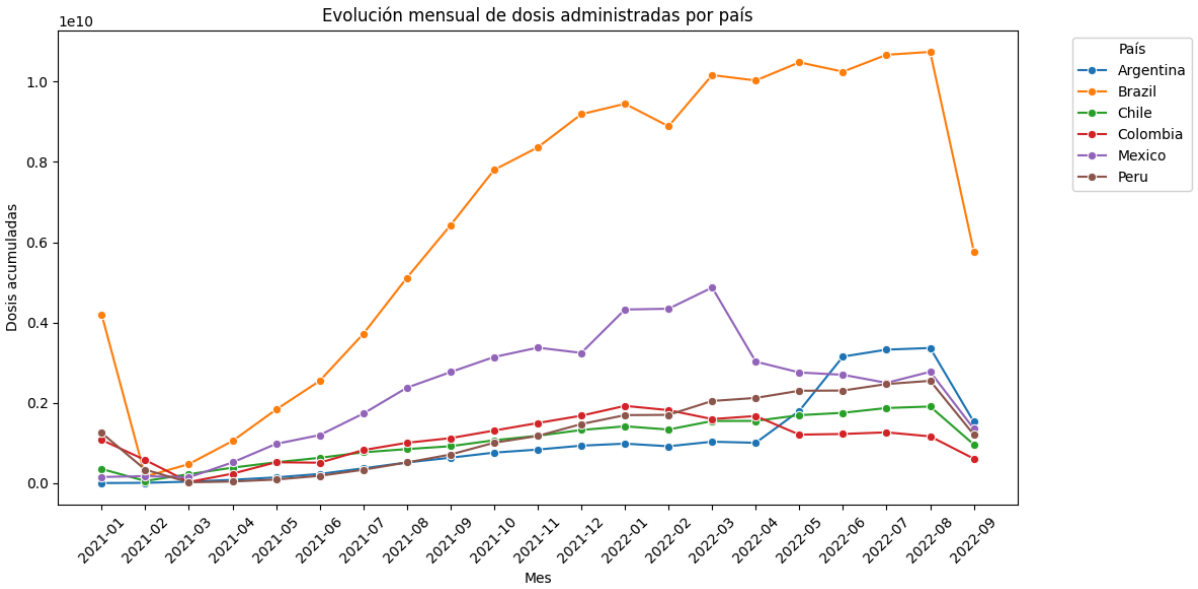
Por lo que vemos, la enfermedad afectó casi por igual a ambos géneros, entonces las muertes por género no creo que sea una variable clave para tomar la decisión hacia donde expandirse.



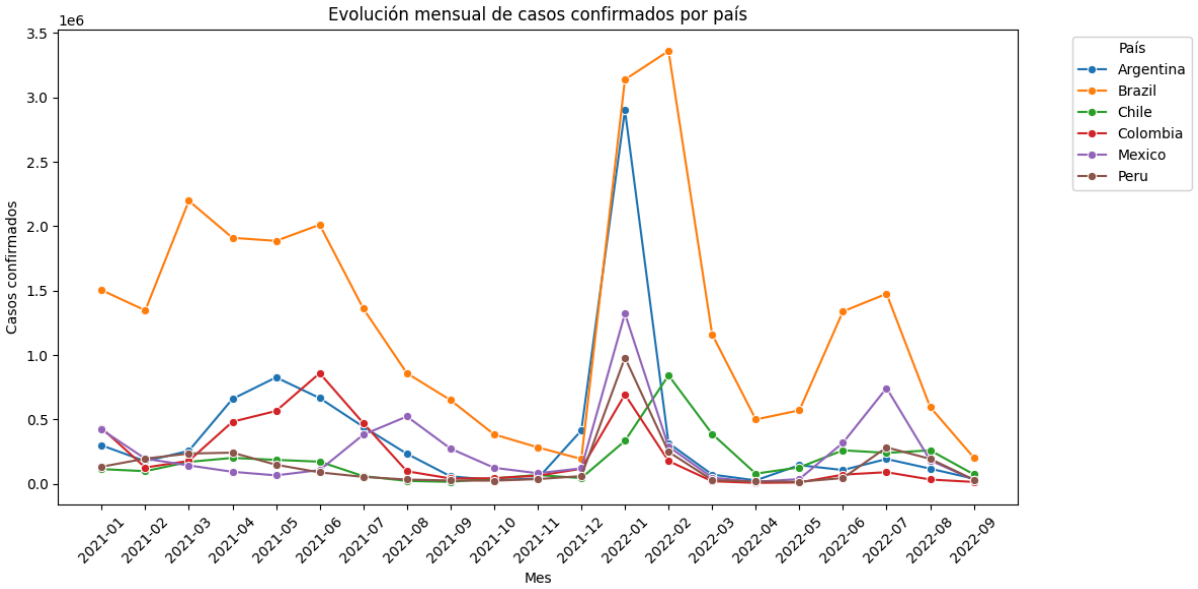
Identificación de Tendencias y Patrones

Evolución de dosis administradas por mes de cada país

En este gráfico destaco el aumento de dosis administradas en diciembre de 2021 en Argentina, mismo mes donde vemos que hubo una suba en la cantidad de casos (segundo gráfico).

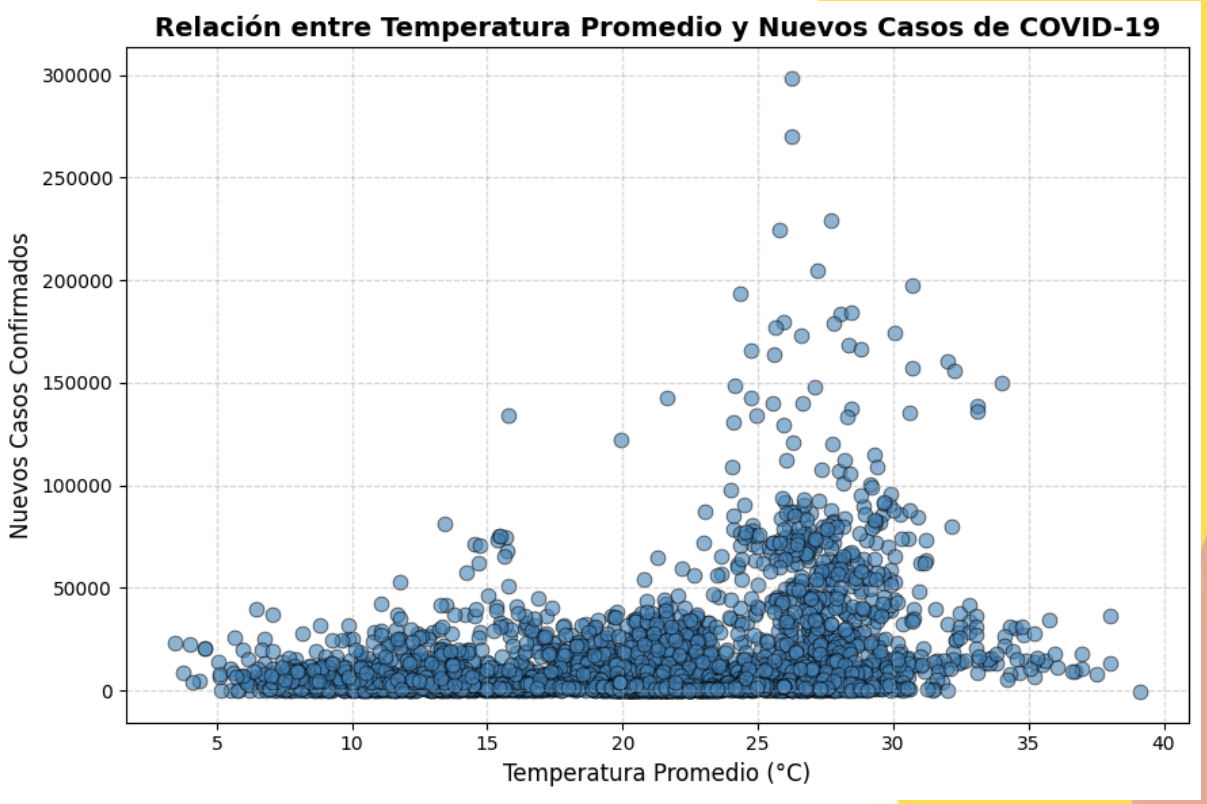


En abril de 2021, todos los países comenzaron a administrar más vacunas, mes en el que se fue estabilizando la cantidad de contagios, al menos hasta diciembre de ese año. Ya en diciembre, subieron los casos y la administración de vacunas también.



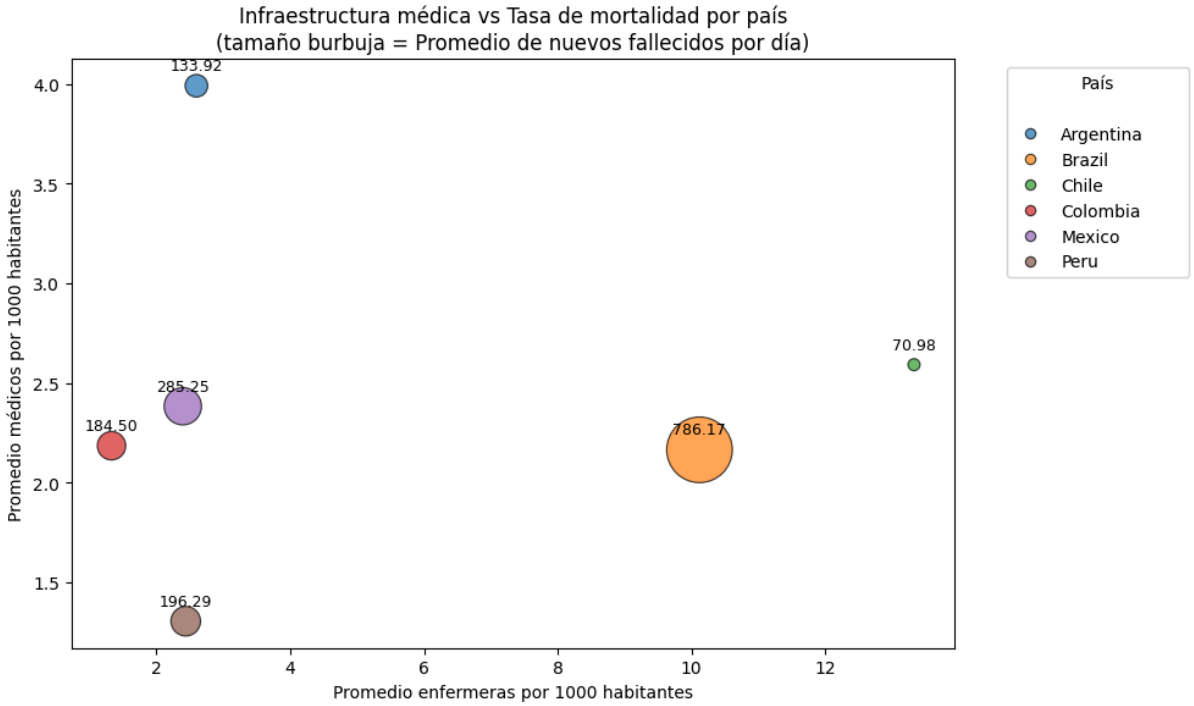
Casos nuevos vs. temperatura promedio

Rápidamente podemos observar que en las temperaturas más cálidas, el número de nuevos casos confirmados aumenta bastante, especialmente entre los 20° y los 35°. Puede que haya una relación entre el aumento de la temperatura, y el aumento de los casos.



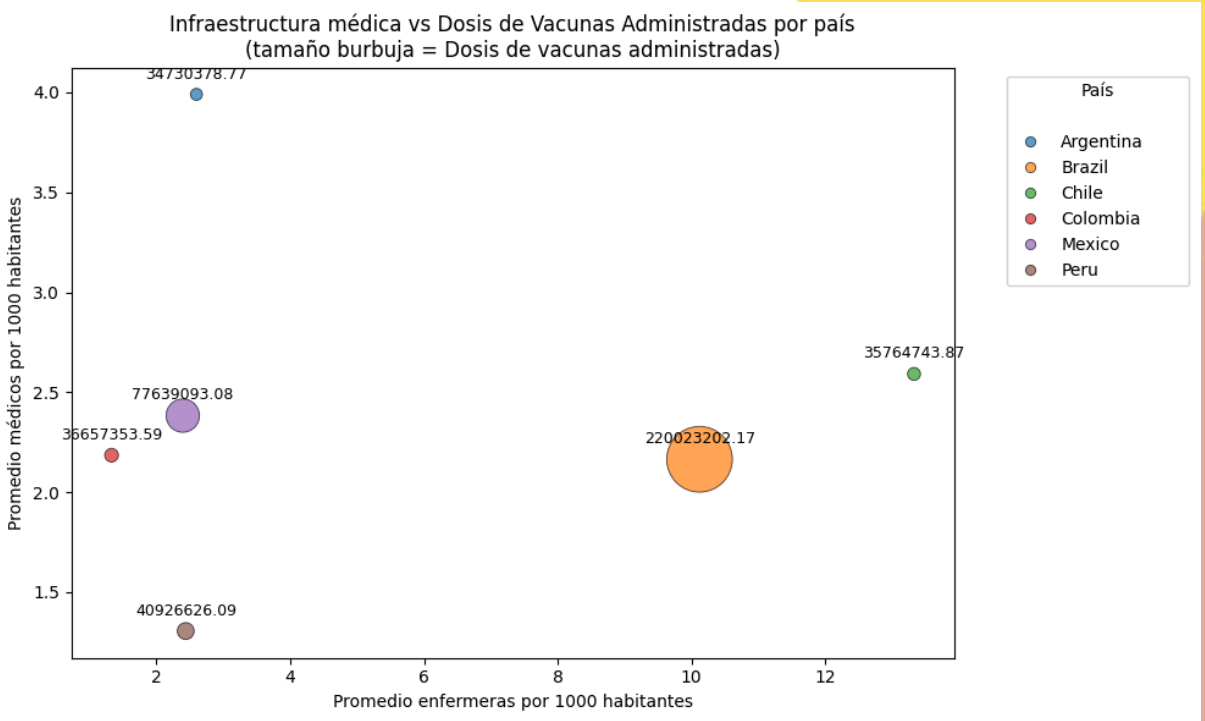
Infraestructura médica vs. mortalidad

Este gráfico vuelve a destacar la infraestructura médica chilena, que es la más preparada y por ello considero que es el país con menor cantidad promedio de muertes por día.
Viendo este gráfico puede que Brasil no puede con tantos casos nuevos por día como vimos antes, y quizás por ello es el país con el mayor promedio de muertes al día.



Relación entre infraestructura médica y dosis administradas

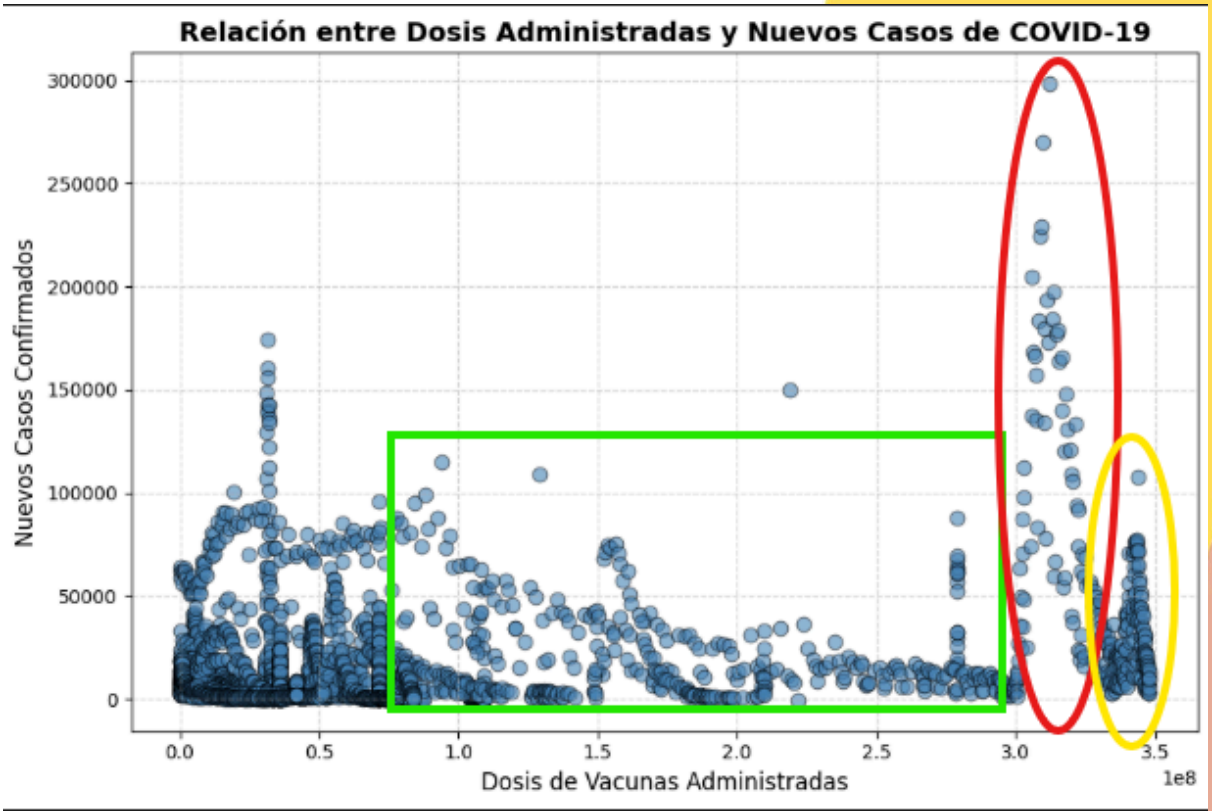
Con este gráfico, podemos observar que no hay una clara correlación entre la cantidad de médicos y enfermeros, y la cantidad de dosis administradas. Por ejemplo, en este caso puede que una mejor infraestructura médica, puede lograr una mayor administración de vacunas (es el caso de Brasil). Pero no es siempre así, ya que médicos y enfermeros no solo se dedican a administrar vacunas, sino también a atender a las personas afectadas. Por eso vemos que Argentina pese a tener el mejor promedio de médicos cada 1000 habitantes, es el país que menos vacunas administró.



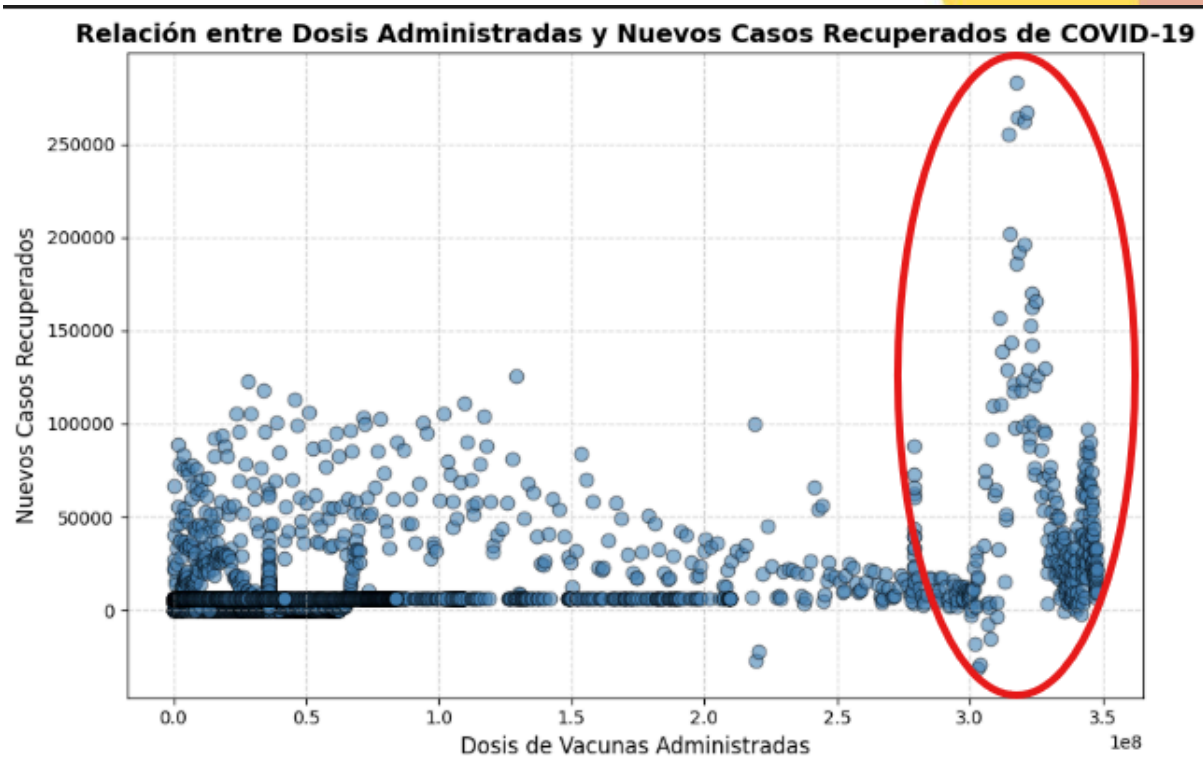
También hay que tener en cuenta que el dataset no tiene información sobre la cantidad de vacunas que compró cada país, claramente un país que compra más vacunas, puede administrar aún más. La falta de este dato (que considero importante), demuestra por qué no hay un claro patrón entre la cantidad de médicos y enfermeros, y las vacunas administradas.

Relación entre vacunación, aumento y recuperación de afectados

Como era de esperarse, a medida que suben los casos, suben las dosis administradas. Igualmente, vemos en este gráfico que hay una disminución en los nuevos casos a medida que sube la administración de vacuna (rectángulo verde).
También vemos que luego de un pico de nuevos casos confirmados (círculo rojo), aumentó aún más la administración de vacunas (círculo amarillo).

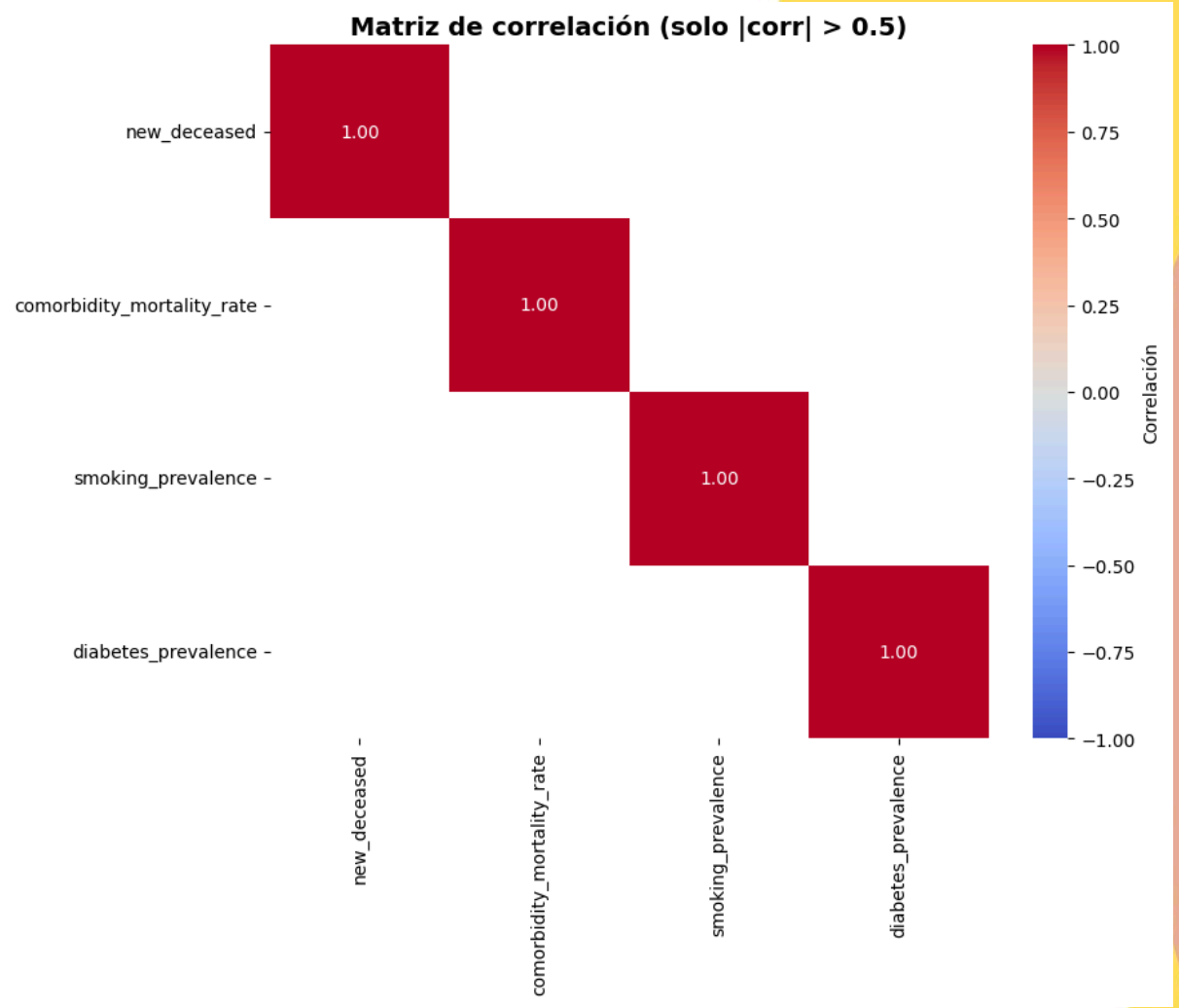


En este segundo gráfico vemos como aumentan la cantidad de recuperados a medida que aumentan las dosis de vacuna administradas.



Correlación entre fallecimientos de COVID-19, fallecimientos por comorbilidad, el porcentaje población preponderante a la diabetes y el porcentaje de fumadores en la población

En este caso vemos que no hay una fuerte correlación (ni positiva ni negativa), entre los fallecidos de COVID-19, los fallecidos por comorbilidad, los fumadores de la población y la población que tiene preponderancia a la diabetes.

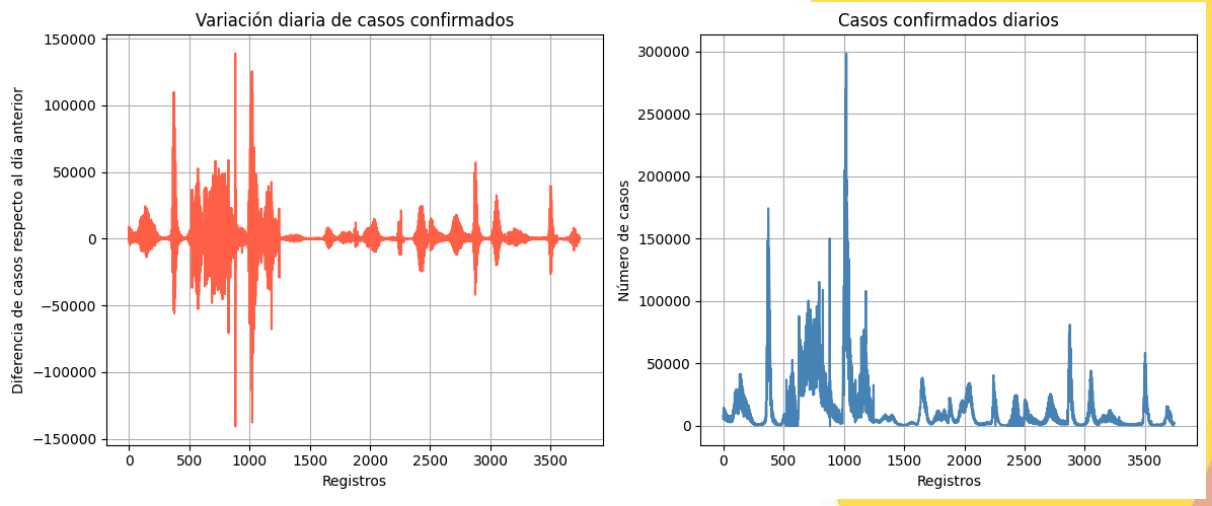


Análisis Temporal

Tasa de cambio de casos confirmados en general

Haciendo un análisis de la serie temporal de la variable new_confirmed encontré clara tendencia a la baja en la incidencia de casos a lo largo del tiempo. La serie inició con una fase de alta volatilidad, marcada por un pico de, aproximadamente, 300.000 casos nuevos, que representó el máximo histórico. Posteriormente, la serie experimentó un declive sostenido y una reducción en la volatilidad (según el gráfico de diferencia simple, cuyas líneas son de color rojo), lo que indica que la magnitud de los cambios diarios se redujo drásticamente. Hacia la etapa final del año 2022, la incidencia de 'new_confirmed' alcanzó

un estado de estabilidad, sugiriendo que la situación se controló y se mantuvo en niveles menores.

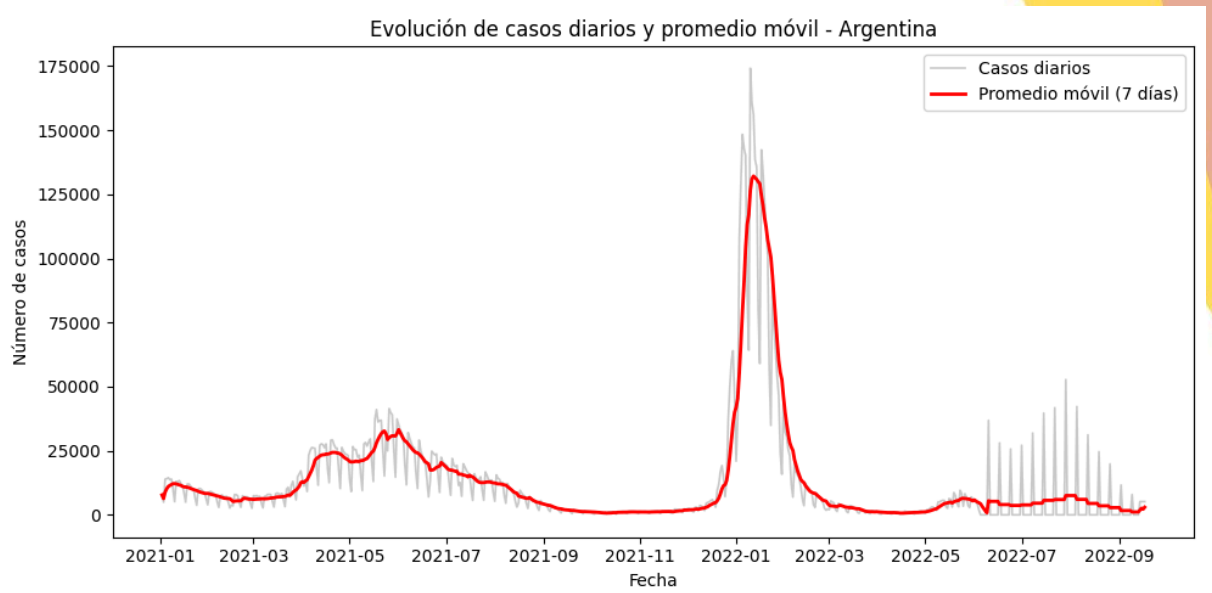


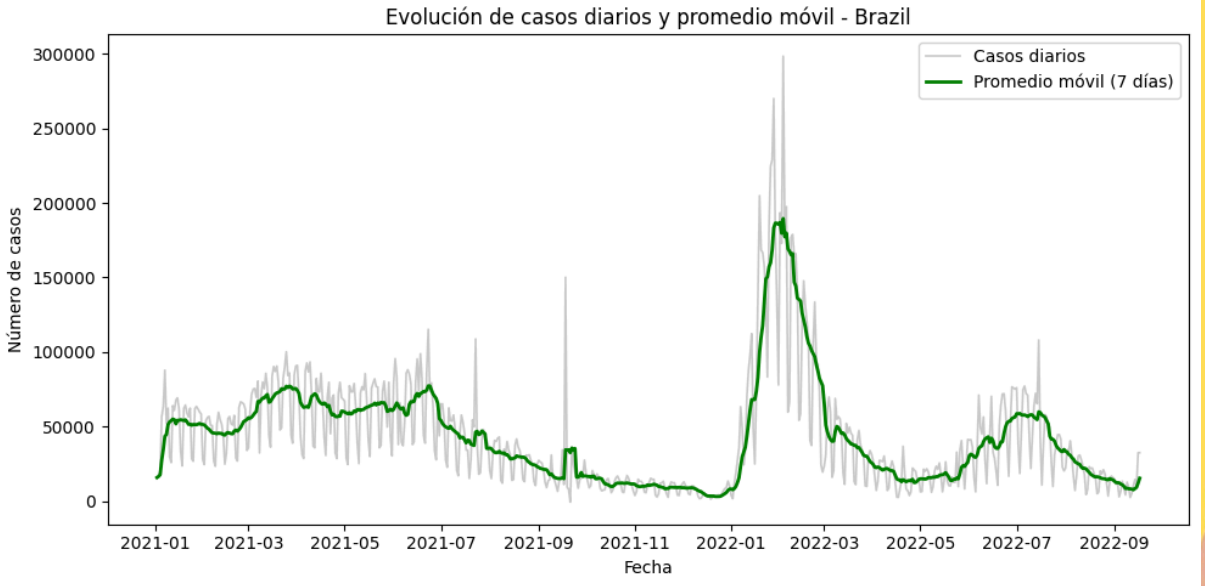
Promedio Móvil de Casos Confirmados por País

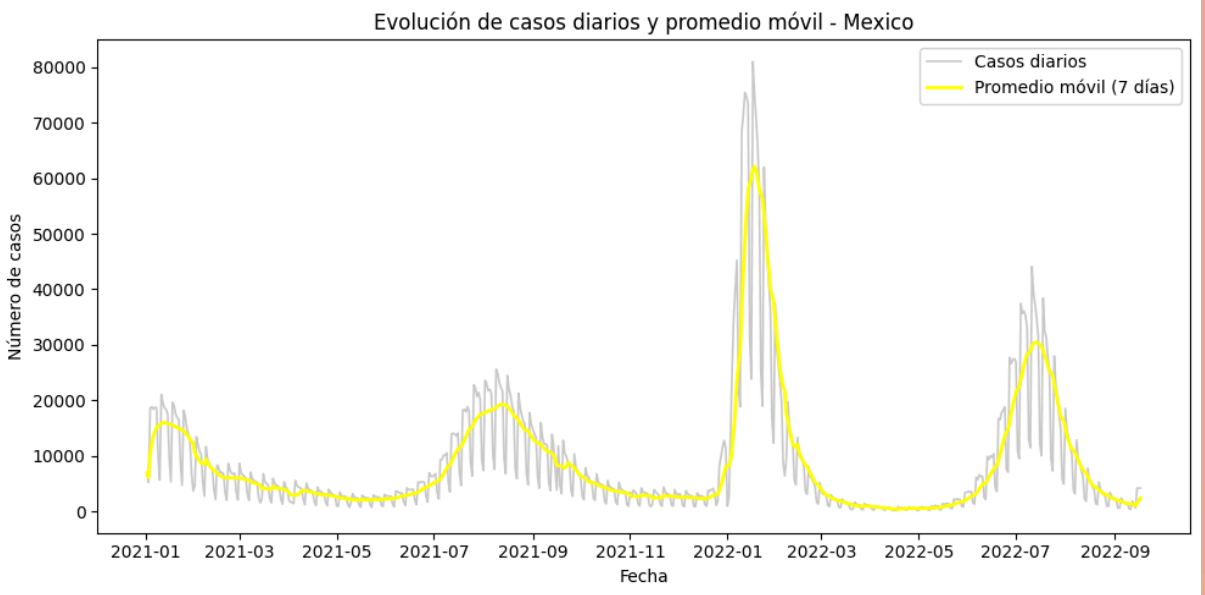
En este caso haremos un análisis más detallado, y observaremos cómo ha avanzado la enfermedad a través del tiempo en cada país.

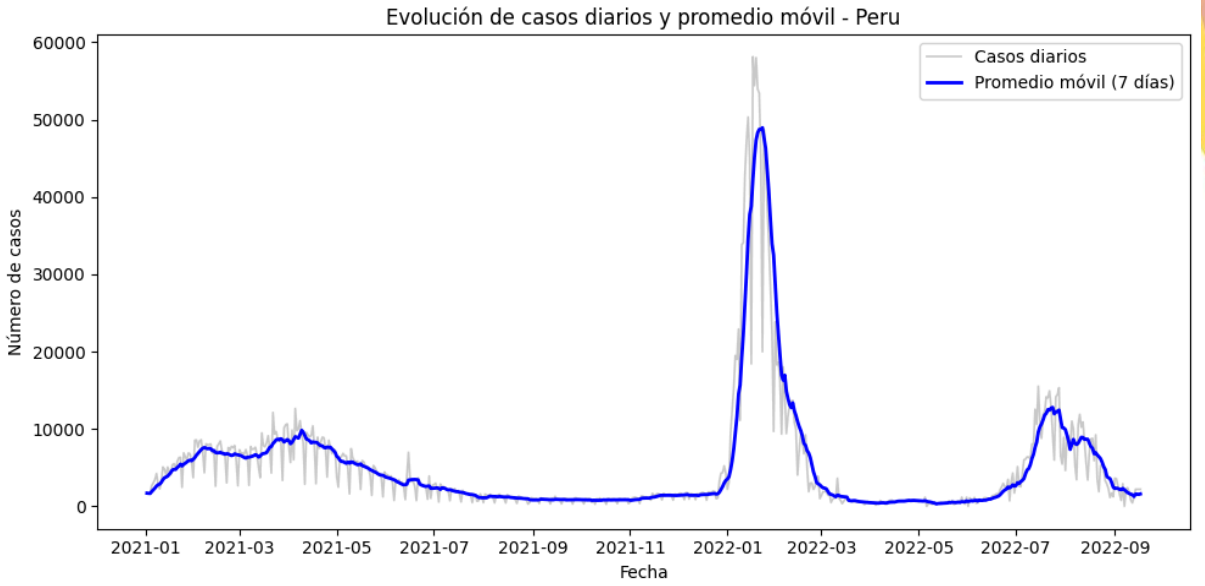
Considero que hay un comportamiento parecido en todos los países, en enero suele haber un aumento en el número de casos, estos meses suelen ser los más cálidos en todos los países menos México, esto demuestra la correlación anteriormente expuesto, donde vemos que suelen haber más casos en temperaturas más cálidas. También veo un aumento, no tan pronunciado, a partir de junio (mes cálido en México), el cual suele aumentar hasta julio aproximadamente.

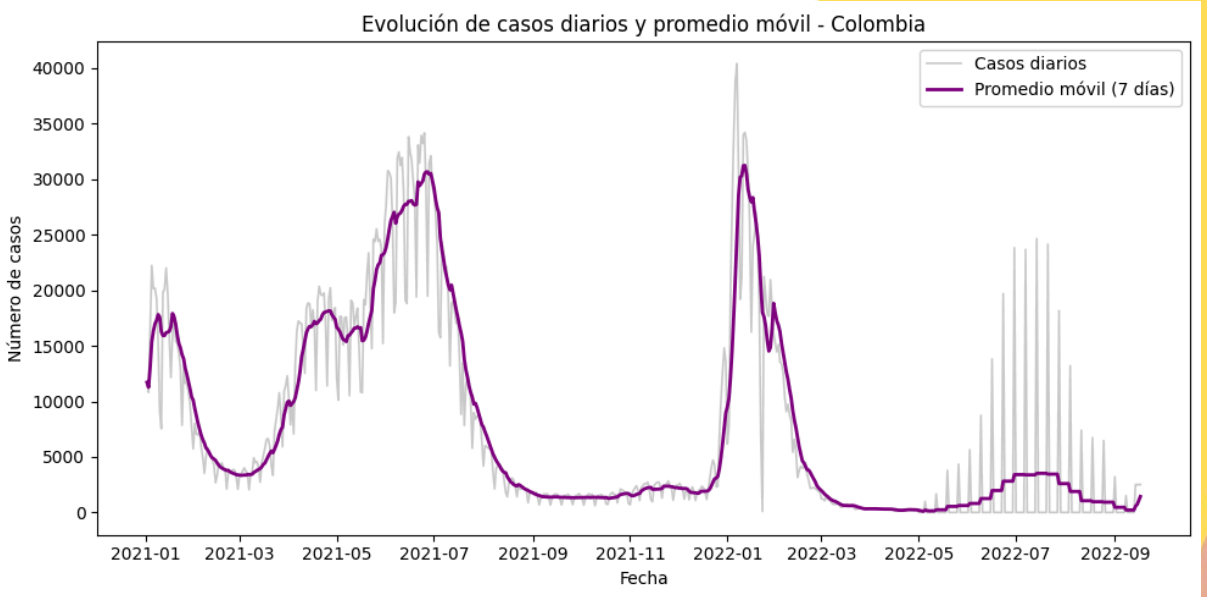
Además hay que hacer hincapié en lo siguiente: A partir de agosto y hasta diciembre, suelen disminuir los casos.

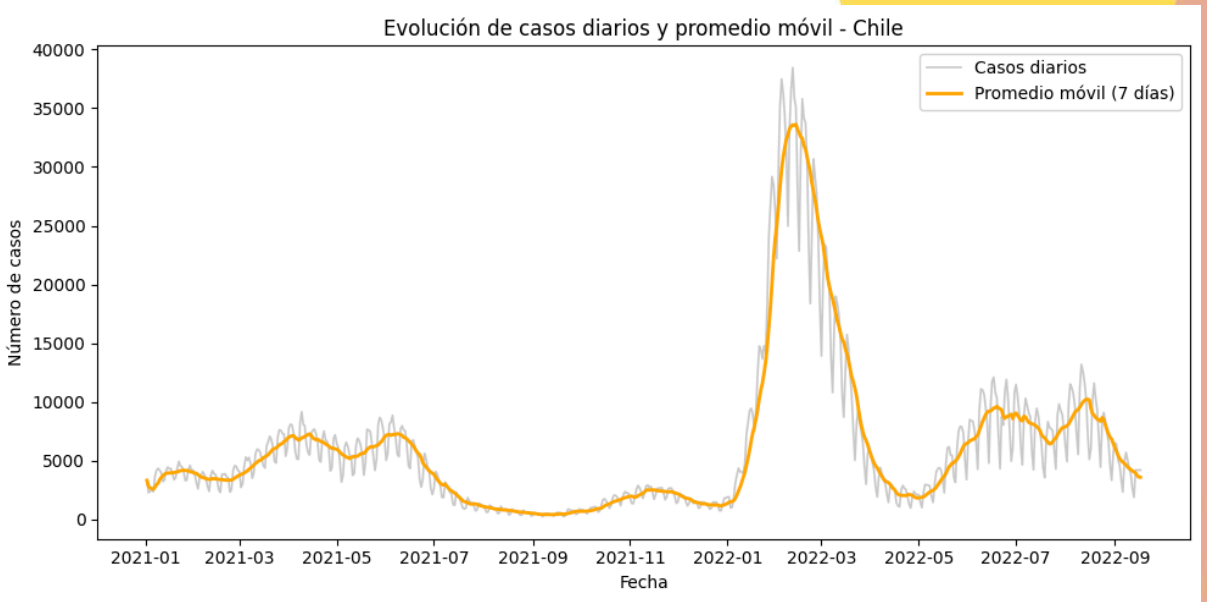



Evolución de casos diarios y promedio móvil - Brazil
Casos diarios
Promedio móvil (7 días)
Número de casos
Fecha
300000
250000
200000
150000
100000
50000
0
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09


Evolución de casos diarios y promedio móvil - Mexico
Casos diarios
Promedio móvil (7 días)
Número de casos
Fecha
80000
70000
60000
50000
40000
30000
20000
10000
0
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09

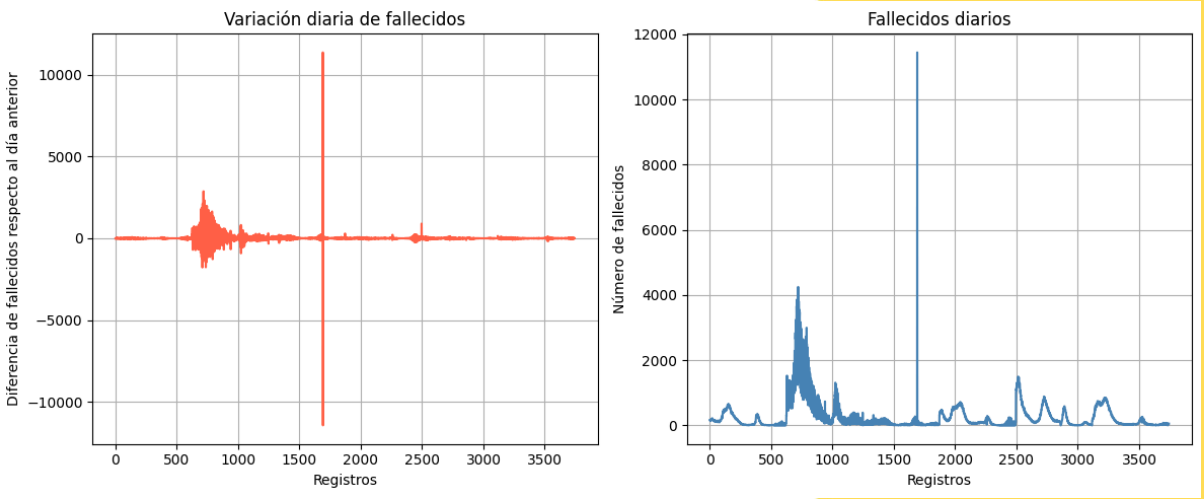

Evolución de casos diarios y promedio móvil - Peru
Casos diarios
Promedio móvil (7 días)
Número de casos
Fecha
60000
50000
40000
30000
20000
10000
0
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09





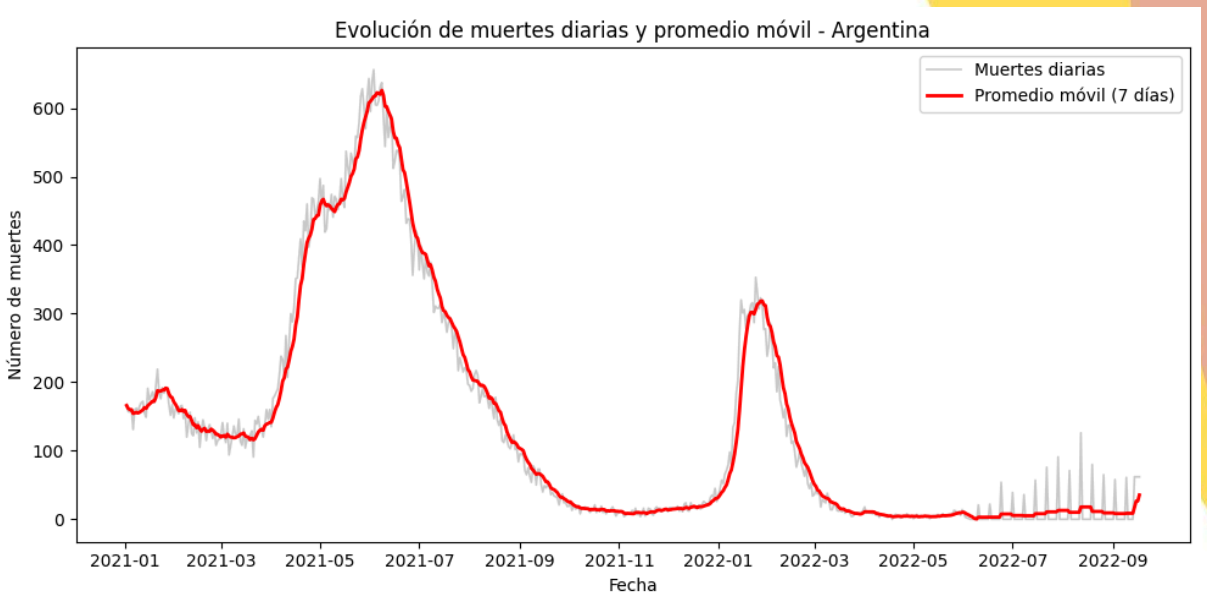
Tasa de cambio de fallecimientos en general

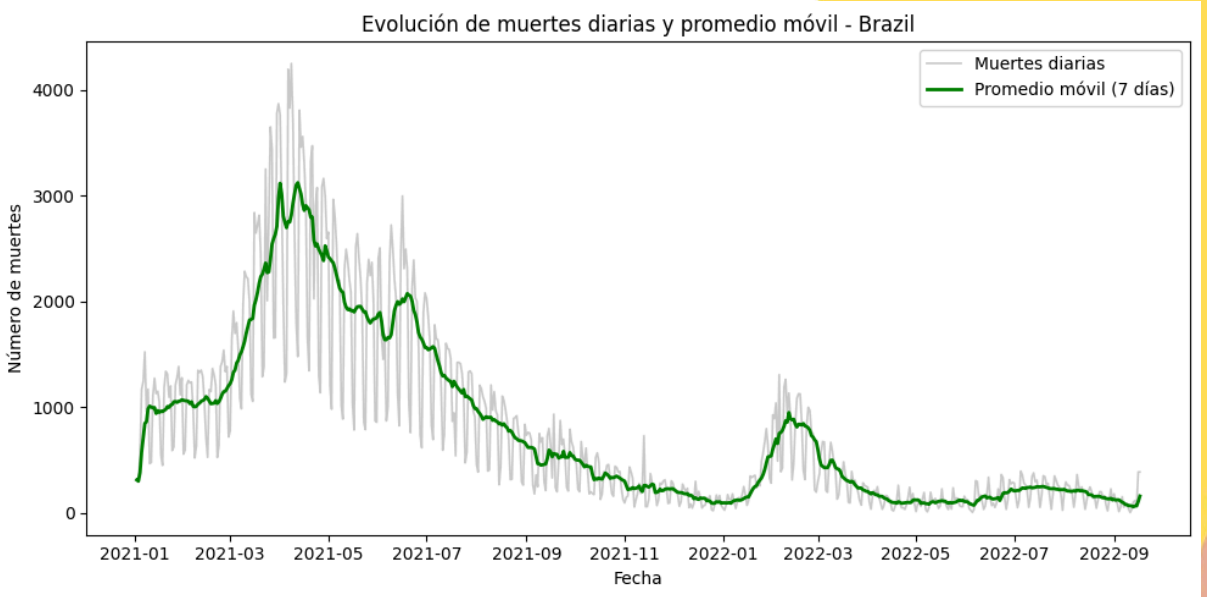
Hice un análisis parecido al análisis de los casos confirmados, pero esta vez con la cantidad de fallecidos diarios y su variación. Aquí pude observar una dinámica similar de brote seguida de una estabilización. El gráfico de "Fallecidos diarios" (derecha) muestra una serie con valores principalmente bajos, pero con varios picos (el mayor superando los 11.000 fallecidos), que representan los momentos de mayor impacto de la pandemia. Por su parte, el gráfico de "Variación diaria de fallecidos" (izquierda) exhibe grandes fluctuaciones, con variaciones positivas y negativas que alcanzan ±10.000, lo cual es atípico e indica una alta volatilidad. A pesar de estos picos y la volatilidad, la serie muestra una clara estabilización a lo largo del tiempo, donde la incidencia y la variación diarias se reducen significativamente, manteniéndose en niveles entre 1000 y 2000 durante la mayor parte del período final.

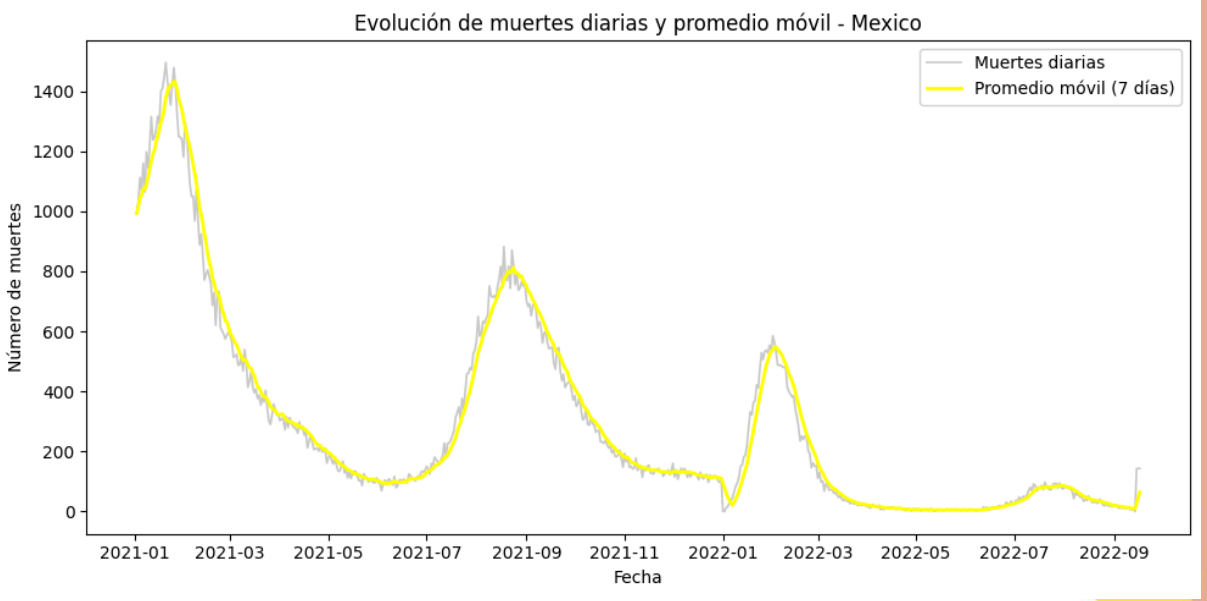


Promedio Móvil de Muertes por País

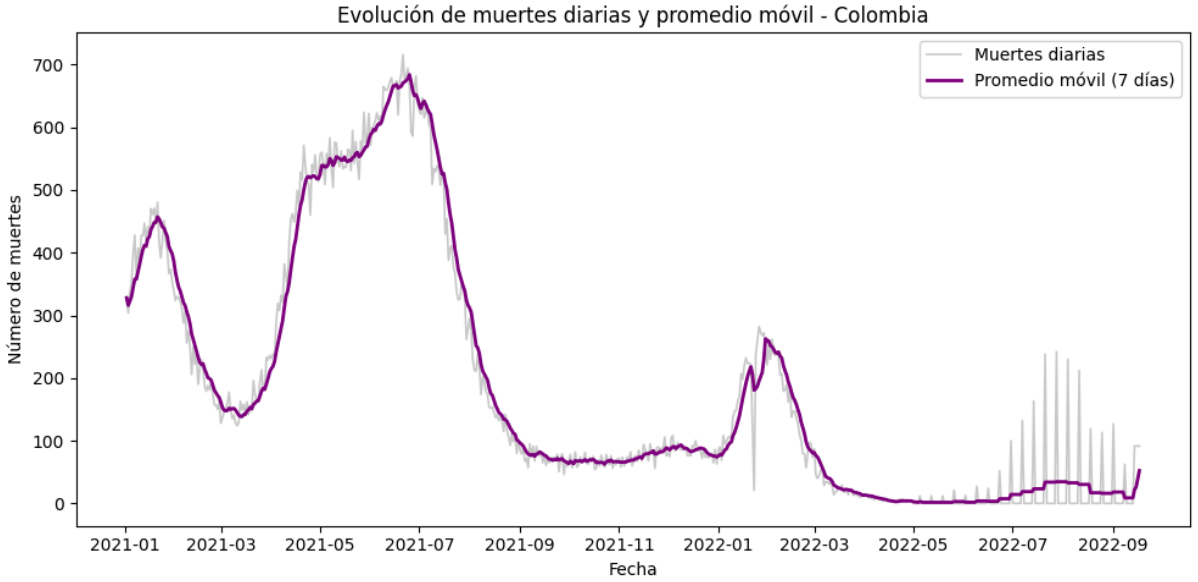
En general, podemos decir que las muertes disminuyeron en 2022, sin contar Chile que tuvo un fuerte pico en abril de ese año. El resto de los países tuvo el pico de fallecimientos en 2021, en los primeros siete meses del año.
En México feo un patrón de aumentos de fallecidos en enero y julio de ambos años. En febrero y septiembre disminuyen las muertes.
En Perú, Colombia y Brasil, en enero de ambos años, hubo un aumento de fallecimientos.




Evolución de muertes diarias y promedio móvil - Brazil
Muertes diarias
Promedio móvil (7 días)
Número de muertes
0
1000
2000
3000
4000
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09
Fecha


Evolución de muertes diarias y promedio móvil - Mexico
Muertes diarias
Promedio móvil (7 días)
Número de muertes
0
200
400
600
800
1000
1200
1400
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09
Fecha


Evolución de muertes diarias y promedio móvil - Peru
Muertes diarias
Promedio móvil (7 días)
Número de muertes
0
200
400
600
800
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09
Fecha

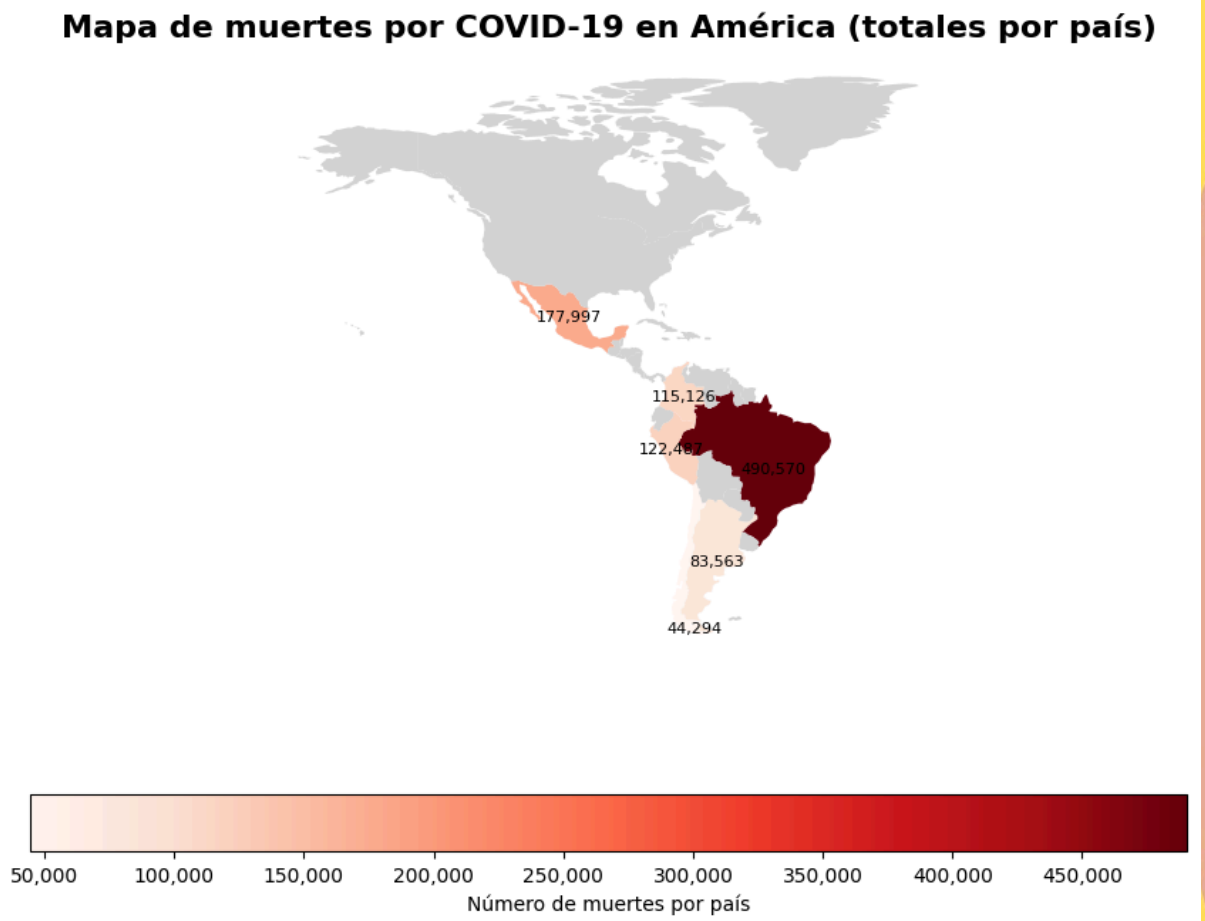

Evolución de muertes diarias y promedio móvil - Colombia
Muertes diarias
Promedio móvil (7 días)
Número de muertes
0
100
200
300
400
500
600
700
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09
Fecha


Evolución de muertes diarias y promedio móvil - Chile
Muertes diarias
Promedio móvil (7 días)
Número de muertes
0
2000
4000
6000
8000
10000
12000
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
2022-03
2022-05
2022-07
2022-09
Fecha

Análisis Espacial

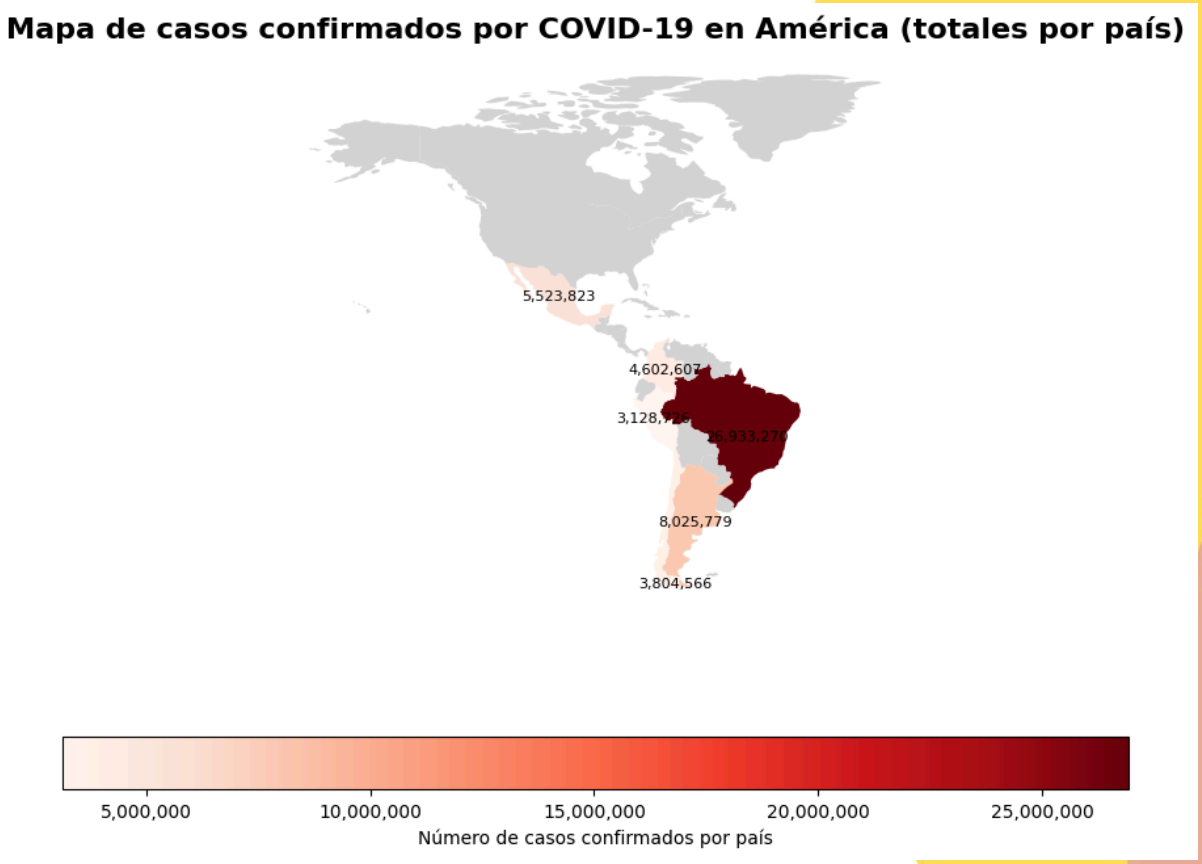
Total de muertes por País

En este caso hice un análisis por País, y en el mapa vemos claramente que Brasil fue el país al que más le afectó la enfermedad, entre este país y el segundo país con más muertes (México) hay más de 300 mil casos de diferencia.
El país menos afectado fue Chile con 44.294 muertes, el país con la mejor infraestructura médica de todos.



Total de Casos Confirmados por País

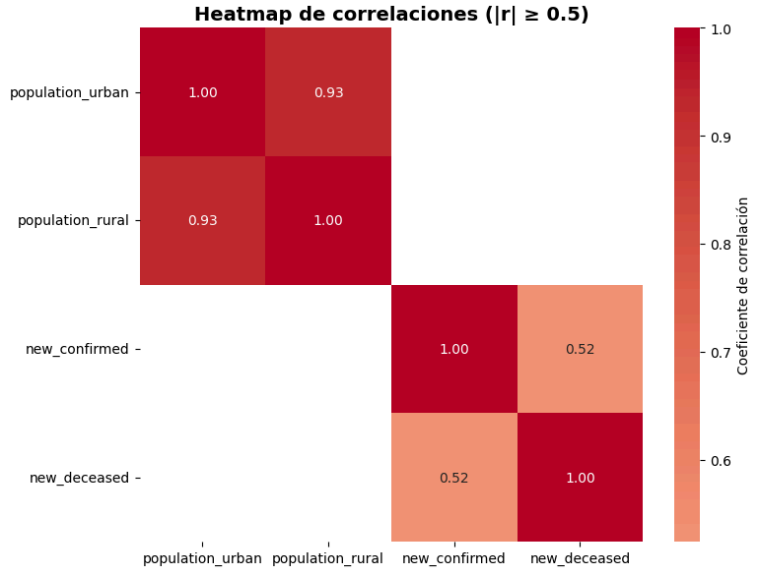
Nuevamente Brasil es el país más afectado con más de 26 millones de casos confirmados, hay una diferencia de más de 18 millones de casos en comparación al segundo país con más casos confirmados (Argentina).
Observé un dato interesante en Argentina, es el segundo país con menos muertes, pero el segundo que más casos confirmados tuvo.



Otras Correlaciones

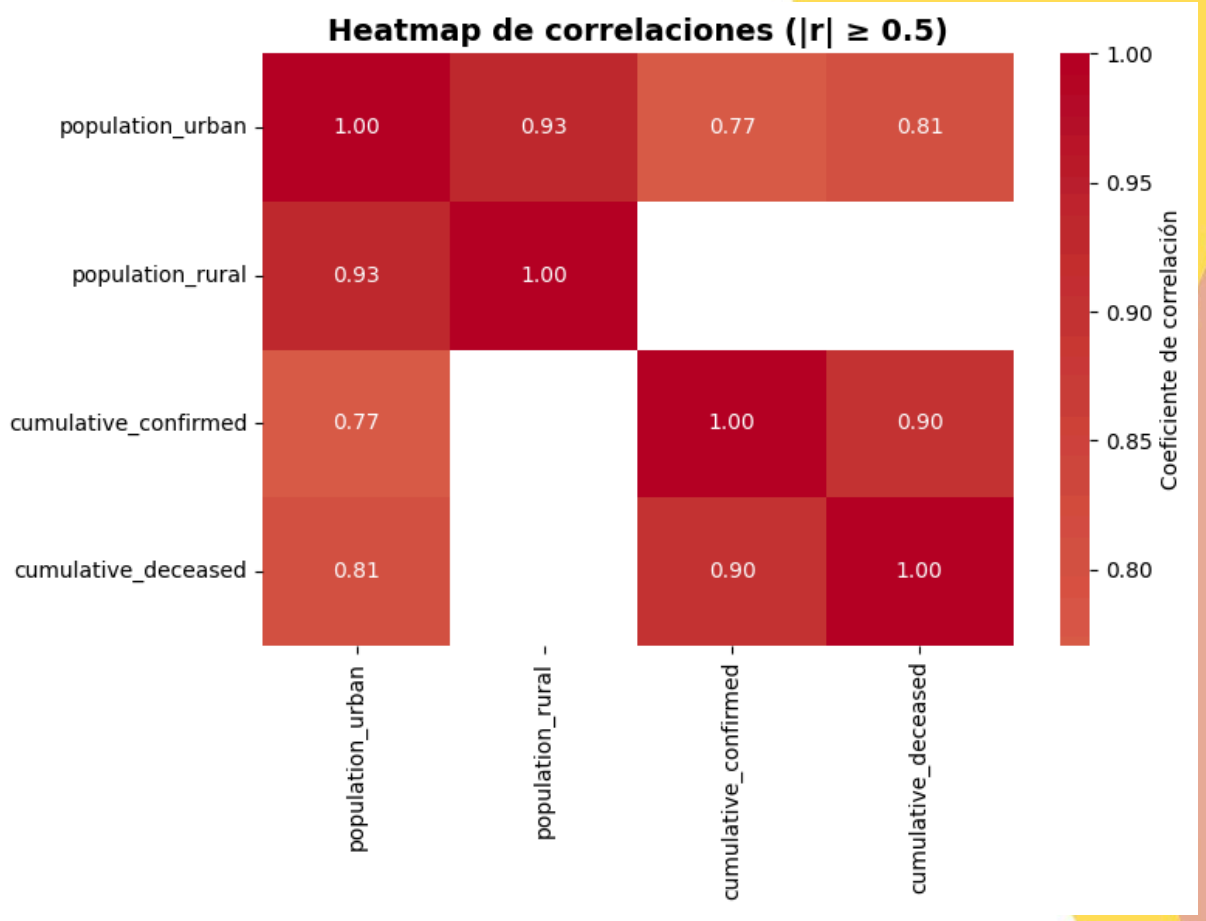
Relación entre casos confirmados, fallecimientos y las población urbana y rural

Quise investigar si hay una correlación entre las poblaciones urbanas y rurales, y los casos confirmados y los fallecimientos, pero luego de graficar el heatmap pude observar que no, o que al menos no hay una fuerte correlación positiva ni negativa.



Relación entre la la acumulación de fallecimientos y casos confirmados en las áreas urbanas y rurales

Se puede observar que en las poblaciones urbanas más grandes, hay una mayor cantidad de casos confirmados, y también una mayor cantidad de fallecidos, en comparación a las poblaciones rurales.



Análisis del Dashboard

El dashboard del proyecto lo encontrará en PI.pbix
El dashboard se divide en 3 tableros:

- Análisis General: En este tablero verá (leyendo de izquierda a derecha en cada fila, comenzando desde la esquina superior izquierda) un análisis de algunas variables determinantes para entender el impacto del COVID-19 en cada país. Algunas de estas variables son: el número de casos confirmados, el número de fallecidos, la cantidad de dosis administradas, la infraestructura sanitaria, etc.
- Análisis Poblacional: En este tablero verá el impacto de la enfermedad pero teniendo en cuenta el número total de habitantes en cada país y el PBI per cápita de cada uno.
- Análisis Predictivo: En este tablero hice una estimación de posibles casos confirmados y posibles fallecimientos en el futuro, teniendo en cuenta los datos del

dataset. Esta información será útil para decidir hacia qué país debe expandirse Byogenesis.

En los 3 tableros puede seguir el patrón Z para observar el contenido.

Desarrollo del Proyecto

AVANCE N°1

Diccionario y Limpieza de Datos

En este Proyecto, el Diccionario de Datos nos fue proporcionado en la consigna, pero si tenemos en cuenta que este informe se va a presentar al cliente, me parece importante que el diccionario esté disponible para él.

Sin embargo, hice algunos cambios en el tipo de datos de las columnas que se verán reflejados en el diccionario:

- La columna *date* es de tipo datetime.
- Cambio los tipos de datos de las columnas *country_code, location_key* y *country_name* a string.
- Las columnas que almacenan los datos acerca de la población serán enteros, ya que no son necesarios los decimales. Menos la columna *population_density* que se mantendrá como float.
- Las columnas *new_confirmed, new_deceased, cumulative_confirmed, cumulative_deceased, cumulative_vaccine_doses_administered, new_recovered y cumulative_recovered*, también serán de tipo entero.

El código utilizado para hacer estos cambios lo puede ver en PIDA_M4_Francisco_Hillebrand.ipynb .

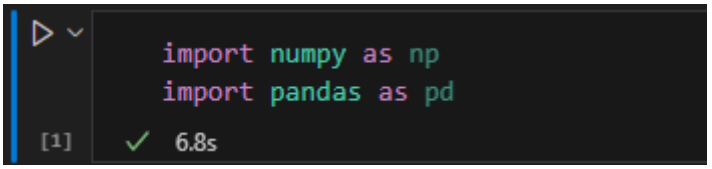
Name	Type	Description	Example
date	datetime	ISO 8601 date (YYYY-MM-DD) of the datapoint. Only has dates greater than 2021-01-01	3/30/2020
population	integer	Total count of humans	51606633
population_male	integer	Total count of males	25846211
population_female	integer	Total count of females	25760422
rural_population	integer	Population in a rural area	9568386
urban_population	integer	Population in an urban area	42038247
population_density	float	Population per squared kilometer of land area	529.3585
human_development_index	float	Composite index of life expectancy, education, and per capita income indicators	0.903
population_age_00_To_79	integer	Estimated population between the ages of ${lower} and ${upper}, both inclusive	42038247
population_age_80_and_older	integer	Estimated population over the age of 80	477081

Name	Type	Description	Example
gdp	float [USD]	Gross domestic product; monetary value of all finished goods and services	24450604878
gdp_per_capita	float [USD]	Gross domestic product divided by total population	1148
new_confirmed	integer	Count of new cases confirmed after positive test on this date	34
new_deceased	integer	Count of new deaths from a positive COVID-19 case on this date	2
cumulative_confirmed	integer	Cumulative sum of cases confirmed after positive test to date	6447
cumulative_deceased	integer	Cumulative sum of deaths from a positive COVID-19 case to date	133
new_recovered	integer	Count of new recoveries from a positive COVID-19 case on this date	13
cumulative_recovered	integer	Cumulative sum of recoveries from a positive COVID-19 case to date	133
latitude	float	Floating point representing the geographic coordinate	30.9756
longitude	float	Floating point representing the geographic coordinate	112.2707
area_sq_km	float [squared kilometers]	Area encompassing this region	3729
area_rural_sq_km	float [squared kilometers]	Area encompassing rural land in this region	3729
area_urban_sq_km	float [squared kilometers]	Area encompassing urban land this region	3729
life_expectancy	float [years]	Average years that an individual is expected to live	75.722
smoking_prevalence	float [%]	Percentage of smokers in population	16.9
diabetes_prevalence	float [%]	Percentage of persons with diabetes in population	13.3
infant_mortality_rate	float	Infant mortality rate (per 1,000 live births)	9.8
adult_male_mortality_rate	float	Mortality rate, adult, male (per 1,000 male adults)	143.719
adult_female_mortality_rate	float	Mortality rate, adult, female (per 1,000 male adults)	98.803
pollution_mortality_rate	float	Mortality rate attributed to household and ambient air pollution, age-standardized (per 100,000 population)	13.3
comorbidity_mortality_rate	float [%]	Mortality from cardiovascular disease, cancer, diabetes or cardiorespiratory disease between exact ages 30 and 70	16.6

Name	Type	Description	Example
nurses_per_1000	float	Nurses and midwives (per 1,000 people)	5.8974
physicians_per_1000	float	Physicians (per 1,000 people)	1.609
location_key	string	Unique string identifying the region	US_CA_06001
country_code	string	ISO 3166-1 alphanumeric 2-letter code of the country	US
country_name	string	American English name of the country, subject to change.	Brazil
cumulative_vaccine_doses_administered	integer	Cumulative sum of vaccine doses administered to persons	923449
average_temperature	float [celsius]	Recorded hourly average temperature	11.22
minimum_temperature	float [celsius]	Recorded hourly minimum temperature	1.74
maximum_temperature	float [celsius]	Recorded hourly maximum temperature	19.42
rainfall	float [millimeters]	Rainfall during the entire day	51
relative_humidity	float [%]	The amount of water vapor present in air expressed as a percentage of the amount needed for saturation at the same temperature	43.09
population_largest_city	integer	Population in the largest city of the country	91256325

Importar librerías

Importo las librerías numpy y pandas con el siguiente código



El código también se puede corroborar en el archivo PIDA_M4_Francisco_Hillebrand.ipynb, que encontrará en la carpeta de Drive.

Leer Archivo

```
#Esta es la ubicación del archivo CSV en mi computadora
csv_path = 'C:/Users/FranciscoJH/Downloads/data_latinoamerica.csv'
try:
    df_latinoamerica = pd.read_csv(csv_path)
    print('Archivo leído correctamente.')
    display(df_latinoamerica.head())
except FileNotFoundError:
    print('Error al leer el CSV. Verifica la ruta del archivo.')
```

✓ 1m 48.1s

Compruebo la cantidad de registros y columnas

```
try:
    df_latinoamerica  # comprobar si la variable existe
except NameError:
    print('df_latinoamerica no está definido. Asegúrate de ejecutar la celda que carga el CSV primero.')
else:
    print(df_latinoamerica.shape)
```

Filtro el Dataframe para obtener solo los registros de los países donde Biogenesys busca expandirse

```
selected_countries = ['Colombia', 'Argentina', 'Chile', 'Mexico', 'Peru', 'Brazil'] # Países por los que filtro el DataFrame

try:
    df_latinoamerica
except NameError:
    print('df_latinoamerica no está definido. Ejecuta primero la celda que carga el CSV.')
else:
    if 'country_name' not in df_latinoamerica.columns:
        print("La columna 'country_name' no existe en el DataFrame. Columnas disponibles:", df_latinoamerica.columns.tolist())
    else:
        df_expansion = df_latinoamerica[df_latinoamerica['country_name'].isin(selected_countries)].copy()
```

Filtra los datos en fechas mayores a 2021-01-01

```
# Filtrar por fecha > 2021-01-01
cutoff = pd.to_datetime('2021-01-01')

# Convertir a datetime la columna date y luego filtrar
try:
    df_expansion['date'] = pd.to_datetime(df_expansion['date'], errors='coerce')
    df_expansion = df_expansion[df_expansion['date'] > cutoff].copy()
except Exception as e:
    print('Error al filtrar por fecha:', e)
```

Trabajo con valores nulos y faltantes

Para estos valores tome las siguientes decisiones:

- Se eliminarán los registros que tengan valores nulos en todas las columnas.

```
df_expansion.dropna(how='all', inplace=True)
#El dataframe no tiene filas con estas características
# se puede comprobar con la siguiente linea print(df_expansion.shape), colocandola antes y despues de ejecutar dropna
```

✓ 5.5s

- Para los valores faltantes, decidí agrupar los registros por país, y reemplazarlos por la mediana de cada columna del dataframe. Si la mediana por país no se encuentra, entonces lo reemplazo por la mediana global de la columna.

```
country_col = 'country_name'

try:
    df_expansion
except NameError:
    print('df_expansion no está definido. Ejecuta la celda que crea df_expansion primero.')
else:
    if country_col not in df_expansion.columns:
        print(f'No se encontró la columna de país: {country_col}')
    else:
        # detectar columnas numéricas
        num_cols = df_expansion.select_dtypes(include=['number']).columns

        df_expansion[num_cols] = df_expansion[num_cols].astype(float)

        for c in num_cols:
            df_expansion[c] = df_expansion.groupby(country_col)[c].transform(lambda s: s.fillna(s.median()))
            df_expansion[c] = df_expansion[c].fillna(df_expansion[c].median())
```

Identificar variables claves

Considere que las variables claves para cumplir los objetivos propuestos para mi análisis son:

- **date:** Necesaria para hacer un análisis temporal del avance del COVID-19 y de las vacunación contra él.
- **population, population_male, population_female, urban_population, population_density y rural_population:** Importante para entender el verdadero impacto del COVID-19 y las vacunas en la población y sus diferentes clasificaciones.
- **population_age:** Estas columnas son muy importantes para descubrir el impacto de la enfermedad en las distintas edades.
- **new_confirmed:** Necesaria para conocer el avance de la enfermedad.
- **new_deceased:** Necesario para conocer el impacto negativo de la enfermedad.
- **new_recovered y cumulative_recovered:** Importante para conocer la respuesta de la población a la enfermedad.
- **nurses y physicians_per_1000:** Necesario para conocer si el sistema médico se encuentra o no preparado.
- **country_name:** Para analizar los países correctos.
- **cumulative_vaccine_doses_administered:** Es necesario saber cuantas vacunas se suministraron hasta una fecha específica.
- **cumulative_confirmed y cumulative_deceased:** Para conocer la tasa de aumento de contagios y de muertes por COVID-19.

Guardar los datos filtrados en un archivo CSV

```
output_filename = 'C:/Users/FranciscoJH/Downloads/DatosFinalesFiltrado.csv'

try:
    df_expansion
except NameError:
    print('df_expansion no está definido. Ejecuta primero las celdas previas para crear y limpiar df_expansion.')
else:
    df_expansion.to_csv(output_filename, index=False)
    print(f'df_expansion guardado en: {output_filename}')
```

Calculo estadísticas descriptivas

```python
#Leo el archivo CSV que contiene los datos filtrados y limpios
csv_path = 'C:/Users/FranciscoJH/Downloads/DatosFinalesFiltrado.csv'
df_datos_filtrados = pd.read_csv(csv_path)

#Voy a guardar las estadísticas en un archivo CSV
output_stats_csv = 'C:/Users/FranciscoJH/Downloads/EstadisticasDatosFinalesFiltrado.csv'

try:
    df_datos_filtrados
except NameError:
    print('df_datos_filtrados no está definido. Ejecuta primero las celdas previas para crear y limpiar df_datos_filtrados.')
else:
    stats = []
    # Separo las columnas numéricas y no numéricas en dos listas separadas
    num_cols = df_datos_filtrados.select_dtypes(include='number').columns.tolist()
    non_num_cols = df_datos_filtrados.select_dtypes(exclude='number').columns.tolist()

    # Usando for para columnas numéricas
    for c in num_cols:
        col = df_datos_filtrados[c]
        #Utilizo la función .describe para obtener estadísticas básicas de las columnas números
        desc = col.describe()
        #Creo un diccionario con las estadísticas, que luego guardaré en el CSV
        stats.append({
            'column': c,
            'mean': float(desc['mean']) if not pd.isna(desc['mean']) else None,
            'std': float(desc['std']) if not pd.isna(desc['std']) else None,
            'median': float(col.median()) if not pd.isna(col.median()) else None, # Mediana calculada directamente, .describe no la incluye
            'min': float(desc['min']) if not pd.isna(desc['min']) else None,
            '25%': float(desc['25%']) if not pd.isna(desc['25%']) else None,
            '50%': float(desc['50%']) if not pd.isna(desc['50%']) else None,
            '75%': float(desc['75%']) if not pd.isna(desc['75%']) else None,
            'max': float(desc['max']) if not pd.isna(desc['max']) else None,
        })
```

```python
    # Usando while para columnas no-numéricas: contar valores únicos y top/freq (si aplica)
    i = 0
    while i < len(non_num_cols):
        c = non_num_cols[i]
        col = df_datos_filtrados[c]
        #Calculo la moda (top) y su frecuencia (freq)
        top = col.mode().iloc[0]
        freq = int(col.value_counts(dropna=True).iloc[0])
        stats.append({
            'column': c,
            #Calculo la cantidad de valores únicos en la columna
            'unique': int(col.nunique(dropna=True)),
            'top': top,
            'freq': freq,
        })
        i += 1
    stats_df = pd.DataFrame(stats)
    stats_df.to_csv(output_stats_csv, index=False)
    print(f'Estadísticas guardadas en: {output_stats_csv}')
    display(stats_df.head(50))
```

Una vez calculadas las estadísticas, respondo las preguntas:

1. ¿Qué implican estas métricas y cómo pueden ayudar en el análisis de datos?
    - mean (media): valor promedio; mide tendencia central, sensible a outliers.
    - median (mediana): otra medida de tendencia central, menos sensible a outliers en comparación con la media.
    - std (desviación estándar): dispersión alrededor de la media; permite evaluar variabilidad.
    - min/max: valores extremos; detectan posibles outliers o errores de entrada.
    - percentiles (25%, 50% (mediana), 75%): resumen de la distribución; la mediana es útil cuando la distribución es asimétrica.
    - unique (para columnas no numéricas): cantidad de valores distintos; útil para decidir si tratar como variable categórica o textual.
    - top/freq (modo y frecuencia): moda y su frecuencia; útil para imputar categóricas o entender concentración.

    Estas medidas, permiten:

- Decidir transformaciones, imputaciones (mediana vs media), y detectar columnas con poca información.
- Identifican outliers y errores de captura.
- Permiten comparar dispersión entre países y priorizar limpieza o análisis.

2. ¿Se muestran todas las estadísticas en todas las columnas durante el análisis?
   No; las funciones aplicadas distinguen tipos: para columnas numéricas se calculan media, distribución estándar, media, mínimo, máximo y los percentiles; para no-numéricas se calculan moda y su frecuencia, y la cantidad de valores únicos. Por lo tanto, no todas las métricas se aplican a todas las columnas.
3. ¿Cuál es la razón de la respuesta anterior y cómo podría afectar la interpretación de los resultados obtenidos?
   Estadísticas como la media requiere datos numéricos. Aplicarlas a columnas no numéricas no tiene sentido y produce errores o resultados engañosos. Para variables categóricas se usan conteos y la moda.
   Por eso es importante decidir métricas por tipo de variables. Es necesario inspeccionar outliers antes de sacar conclusiones.

Función que obtiene la mediana, varianza y rango de un conjunto de valores

```python
def summary_stats(values, ddof=0):
    #Convierto los valores a una Serie de pandas
    s = pd.Series(values).dropna().astype(float)
    if s.empty:
        return {'median': None, 'variance': None, 'range': None}

    median = float(s.median())
    variance = float(s.var(ddof=ddof))
    value_range = float(s.max() - s.min())
    return {'median': median, 'variance': variance, 'range': value_range}

# Ejemplo de uso:
# summary_stats(df_datos_filtrados['population'])
```

1. ¿Qué representa la mediana?
   La mediana es el valor que se encuentra en el centro del conjunto de datos una vez ordenados.
   Divide la muestra en dos partes iguales: 50% de los datos son menores o iguales a la mediana y 50% son mayores o iguales; además, es más robusta frente a valores extremos que la media.
2. ¿Cómo varía la dispersión de los datos en el conjunto de datos analizado, en términos de la varianza y el rango?
   La varianza mide cuánto se alejan los datos, en promedio, respecto a la media.
   - Una varianza alta significa que los datos están muy dispersos.
   - Una varianza baja indica que los datos están concentrados alrededor de la media.

   El rango mide simplemente la diferencia entre el valor máximo y el mínimo.

- Es una medida de dispersión simple, pero puede verse afectada por valores atípicos.

3. ¿Qué nos puede indicar esto sobre la consistencia o la variabilidad de los datos en relación con la mediana?
    - Si el rango y la varianza son pequeños, los datos están consistentes, concentrados cerca de la mediana, lo que indica poca variabilidad.
    - Si el rango y la varianza son grandes, los datos son variables, lo que significa que, aunque la mediana marque un valor central, los valores individuales se alejan bastante de ese centro.

AVANCE N°2

Importar librerías

```
import matplotlib.pyplot as plt
import seaborn as sns
```

Análisis Estadístico

En este caso hice los siguientes pasos:

- Agrupe los registros por país, y para cada variable calculé las siguientes medidas:
    - Media
    - Mediana
    - Desviación estándar
    - Varianza
    - Mínimo
    - Máximo

Para observar los valores obtenidos, puedo hacerlo en Stats por país

- Cree una matriz de correlaciones entre todas las variables del dataset, la puede ver en correlaciones_byogenesis.

Los insights encontrados a partir de este análisis lo podrá ver en el apartado EDA e Insights

```python
# Usamos el DataFrame filtrado previamente: df_byogenesis
try:
    df_byogenesis
except NameError:
    print('df_byogenesis no está definido. Ejecuta primero la celda que carga el CSV.')
else:
    # Excluimos columnas no numéricas y de fecha para evitar errores en los cálculos
    exclude_cols = ['date', 'location_key', 'country_name', 'country_code']
    #Ahora selecciono solo las columnas numéricas
    num_cols = [c for c in df_byogenesis.columns if df_byogenesis[c].dtype.kind in 'fi' and c not in exclude_cols]

    # Calculo las medidas de tendencia central y dispersión por país
    stats = df_byogenesis.groupby('country_name')[num_cols].agg([
        np.mean, np.median, np.std, np.var, np.min, np.max
    ])
    print('Medidas de tendencia central y dispersión por país (solo columnas numéricas relevantes):')
    display(stats)

    # Correlaciones entre variables numéricas
    if len(num_cols) >= 2:
        corr = df_byogenesis[num_cols].corr()
        print('Matriz de correlación entre variables relevantes:')
        display(corr)
    else:
        print('No hay suficientes columnas numéricas relevantes para calcular correlaciones.')
```

Visualización de Datos con Matplotlib y Seaborn

En este apartado se verá el código utilizado para realizar los gráficos del apartado Visualización de los Insights

Promedio de Casos Confirmados por país

```python
try:
    df_byogenesis
except NameError:
    print('df_byogenesis no está definido. Ejecuta primero la celda que carga el CSV.')
else:
    if 'new_confirmed' not in df_byogenesis.columns or 'country_name' not in df_byogenesis.columns:
        print('Las columnas necesarias no existen en el DataFrame.')
    else:
        promedio_por_pais = df_byogenesis.groupby('country_name')['new_confirmed'].mean().sort_values(ascending=False)
        plt.figure(figsize=(10,6))
        ax = sns.barplot(x=promedio_por_pais.index, y=promedio_por_pais.values, palette='viridis')
        plt.title('Promedio de casos nuevos confirmados por país')
        plt.xlabel('País')
        plt.ylabel('Promedio de casos nuevos confirmados')
        plt.xticks(rotation=45)
        # Agregar etiquetas a cada barra
        for i, v in enumerate(promedio_por_pais.values):
            ax.text(i, v, f'{v:.0f}', ha='center', va='bottom', fontsize=9)
        plt.tight_layout()
        plt.show()
```

Comparación de casos confirmados por país y su densidad poblacional

```
try:
    df_byogenesis
except NameError:
    print('df_byogenesis no está definido. Ejecuta primero la celda que carga el CSV.')
else:
    required_cols = [
        'country_name',
        'population_density',
        'new_confirmed',
        'cumulative_vaccine_doses_administered'
    ]
    if not all(col in df_byogenesis.columns for col in required_cols):
        print('Faltan columnas necesarias en el DataFrame:',
              [col for col in required_cols if col not in df_byogenesis.columns])
    else:
        # Calcular promedios por país
        df_comp = df_byogenesis.groupby('country_name', as_index=False).agg({
            'new_confirmed': 'mean',
            'population_density': 'mean',
            'cumulative_vaccine_doses_administered': 'mean'
        })

        # Renombrar columnas
        df_comp.rename(columns={
            'new_confirmed': 'Promedio casos nuevos confirmados',
            'population_density': 'Densidad poblacional',
            'cumulative_vaccine_doses_administered': 'Promedio dosis acumuladas'
        }, inplace=True)

        # Escalar tamaño de burbujas
        df_comp['Tamaño burbuja'] = (
            df_comp['Promedio dosis acumuladas'] / df_comp['Promedio dosis acumuladas'].max() * 1500
        )

        plt.figure(figsize=(10, 6))
```

```python
# Gráfico de dispersión
ax = sns.scatterplot(
    x='Densidad poblacional',
    y='Promedio casos nuevos confirmados',
    size='Tamaño burbuja',
    hue='country_name',
    data=df_comp,
    palette='tab20',
    sizes=(50, 1500),
    alpha=0.7,
    edgecolor='black',
    legend='full'
)

# Quitar la leyenda del tamaño de burbujas (solo dejar la de países)
handles, labels = ax.get_legend_handles_labels()
# Encontrar el índice donde empieza la leyenda de tamaño
if "Tamaño burbuja" in labels:
    size_index = labels.index("Tamaño burbuja")
    handles = handles[:size_index]
    labels = labels[:size_index]
ax.legend(handles, labels, title='País', bbox_to_anchor=(1.05, 1), loc='upper left')

# Agregar etiquetas con las dosis acumuladas
for i, row in df_comp.iterrows():
    plt.text(
        row['Densidad poblacional'],
        row['Promedio casos nuevos confirmados'] + 0.02 * df_comp['Promedio casos nuevos confirmados'].max(),
        f"{int(row['Promedio dosis acumuladas']):,}".replace(',', '.'),
        fontsize=9,
        ha='center'
    )

plt.title('Densidad poblacional vs Casos nuevos confirmados por país\n(tamaño burbuja = dosis acumuladas)')
plt.xlabel('Densidad poblacional')
plt.ylabel('Promedio de casos nuevos confirmados')
plt.tight_layout()
plt.show()
```

Correlación entre nuevos casos confirmados, aumento de las muertes y aumento de la población a lo largo del tiempo

```
paises = df_byogenesis['country_name'].unique()
num_paises = len(paises)

# Definir número de filas y columnas para los subplots
cols = 3  # podés cambiar a 2 o 4 según convenga
rows = (num_paises + cols - 1) // cols  # ceil para que alcance para todos los países

fig, axes = plt.subplots(rows, cols, figsize=(cols*5, rows*4))
axes = axes.flatten()  # para iterar fácilmente aunque haya más de una fila

for i, pais in enumerate(paises):
    df_pais = df_byogenesis[df_byogenesis['country_name'] == pais]
    df_corr = df_pais[['population', 'new_confirmed', 'new_deceased']].corr()

    sns.heatmap(
        df_corr,
        annot=True,
        cmap='coolwarm',
        vmin=-1,
        vmax=1,
        fmt=".2f",
        ax=axes[i]  # asignar a cada subplot
    )
    axes[i].set_title(f'{pais}')

# Eliminar ejes sobrantes si hay más subplots que países
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()
```

Comparación de la estructura sanitaria del país

```
# Agrupar por país y calcular el promedio de nurses_per_1000
df_nurses = (
    df_byogenesis
    .groupby('country_name', as_index=False)['nurses_per_1000']
    .mean()
    .sort_values('nurses_per_1000', ascending=False)
)

plt.figure(figsize=(12, 6))
ax = sns.barplot(
    data=df_nurses,
    x='country_name',
    y='nurses_per_1000',
    palette='Blues_d'
)

# Agregar etiquetas encima de cada barra (formato con 2 decimales)
for p in ax.patches:
    height = p.get_height()
    ax.annotate(
        f'{height:.2f}',
        xy=(p.get_x() + p.get_width() / 2, height),
        xytext=(0, 3),                # desplazamiento en puntos
        textcoords='offset points',
        ha='center',
        va='bottom',
        fontsize=9
    )
```

```
ax.set_title('Enfermeros por cada 1000 habitantes')
ax.set_xlabel('País')
ax.set_ylabel('Nurses per 1000')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
# Agrupar por país y calcular el promedio de physicians_per_1000
df_physicians = (
    df_byogenesis
    .groupby('country_name', as_index=False)['physicians_per_1000']
    .mean()
    .sort_values('physicians_per_1000', ascending=False)
)

plt.figure(figsize=(12, 6))
ax = sns.barplot(
    data=df_physicians,
    x='country_name',
    y='physicians_per_1000',
    palette='Blues_d'
)

# Agregar etiquetas encima de cada barra (formato con 2 decimales)
for p in ax.patches:
    height = p.get_height()
    ax.annotate(
        f'{height:.2f}',
        xy=(p.get_x() + p.get_width() / 2, height),
        xytext=(0, 3),                # desplazamiento en puntos
        textcoords='offset points',
        ha='center',
        va='bottom',
        fontsize=9
    )

ax.set_title('Médicos por cada 1000 habitantes')
ax.set_xlabel('País')
ax.set_ylabel('Physicians per 1000')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

Comparación del porcentaje de mortalidad femenina y masculina

```python
# Agrupar por país
df_mortalidad = df_byogenesis.groupby('country_name', as_index=False).agg({
    'population': 'mean',
    'population_male': 'mean',
    'population_female': 'mean',
    'new_deceased': 'sum'
})

# Calcular muertes estimadas por género (proporcionales al tamaño poblacional)
df_mortalidad['muertes_masculinas'] = (
    df_mortalidad['new_deceased'] * (df_mortalidad['population_male'] / df_mortalidad['population'])
)
df_mortalidad['muertes_femeninas'] = (
    df_mortalidad['new_deceased'] * (df_mortalidad['population_female'] / df_mortalidad['population'])
)

# Calcular el total de muertes (para normalizar)
df_mortalidad['total_muertes_estimadas'] = (
    df_mortalidad['muertes_masculinas'] + df_mortalidad['muertes_femeninas']
)

# Calcular porcentajes de muertes por género
df_mortalidad['pct_masculinas'] = (
    df_mortalidad['muertes_masculinas'] / df_mortalidad['total_muertes_estimadas'] * 100
)
df_mortalidad['pct_femeninas'] = (
    df_mortalidad['muertes_femeninas'] / df_mortalidad['total_muertes_estimadas'] * 100
)
```

```python
# Crear gráfico de barras apiladas al 100 %
plt.figure(figsize=(12, 6))
bars_male = plt.bar(df_mortalidad['country_name'], df_mortalidad['pct_masculinas'],
                    label='Masculinas', color='steelblue')
bars_female = plt.bar(df_mortalidad['country_name'], df_mortalidad['pct_femeninas'],
                      bottom=df_mortalidad['pct_masculinas'], label='Femeninas', color='lightcoral')

# Agregar etiquetas de porcentaje sobre cada segmento
for bar_m, bar_f, pct_m, pct_f in zip(bars_male, bars_female,
                                      df_mortalidad['pct_masculinas'],
                                      df_mortalidad['pct_femeninas']):
    # Etiqueta hombres
    plt.text(bar_m.get_x() + bar_m.get_width()/2, bar_m.get_height()/2,
             f"{pct_m:.1f}%", ha='center', va='center', color='white', fontsize=9)
    # Etiqueta mujeres
    plt.text(bar_f.get_x() + bar_f.get_width()/2,
             bar_m.get_height() + bar_f.get_height()/2,
             f"{pct_f:.1f}%", ha='center', va='center', color='black', fontsize=9)

plt.title('Porcentaje estimado de muertes masculinas y femeninas por país')
plt.xlabel('País')
plt.ylabel('Porcentaje de muertes (%)')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Género')
plt.tight_layout()
plt.show()
```

Identificación de Tendencias y Patrones

En este apartado se verá el código utilizado para realizar los gráficos del apartado Identificación de Tendencias y Patrones

Evolución de dosis administradas por mes de cada país

```python
# Asegurarse de que la columna de fecha esté en formato datetime
df_byogenesis['date'] = pd.to_datetime(df_byogenesis['date'])

# Crear una columna con el mes (año-mes)
df_byogenesis['month'] = df_byogenesis['date'].dt.to_period('M').astype(str)

# Calcular el promedio (o suma, según el caso) de dosis administradas por mes y país
df_monthly = (
    df_byogenesis
    .groupby(['country_name', 'month'])['cumulative_vaccine_doses_administered']
    .sum()
    .reset_index()
)

# Crear el gráfico
plt.figure(figsize=(12, 6))
sns.lineplot(
    data=df_monthly,
    x='month',
    y='cumulative_vaccine_doses_administered',
    hue='country_name',
    marker='o'
)

plt.title('Evolución mensual de dosis administradas por país')
plt.xlabel('Mes')
plt.ylabel('Dosis acumuladas')
plt.xticks(rotation=45)
plt.legend(title='País', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

```python
# Asegurarse de que la columna de fecha esté en formato datetime
df_byogenesis['date'] = pd.to_datetime(df_byogenesis['date'])

# Crear una columna con el mes (año-mes)
df_byogenesis['month'] = df_byogenesis['date'].dt.to_period('M').astype(str)

# Calcular el promedio (o suma, según el caso) de dosis administradas por mes y país
df_monthly = (
    df_byogenesis
    .groupby(['country_name', 'month'])['new_confirmed']
    .sum()
    .reset_index()
)

# Crear el gráfico
plt.figure(figsize=(12, 6))
sns.lineplot(
    data=df_monthly,
    x='month',
    y='new_confirmed',
    hue='country_name',
    marker='o'
)

plt.title('Evolución mensual de casos confirmados por país')
plt.xlabel('Mes')
plt.ylabel('Casos confirmados')
plt.xticks(rotation=45)
plt.legend(title='País', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Casos nuevos vs. temperatura promedio

```
plt.figure(figsize=(9,6))  # Tamaño del gráfico

# Diagrama de dispersión simple
sns.scatterplot(
    data=df_byogenesis,
    x='average_temperature_celsius',              # Eje X: temperatura promedio
    y='new_confirmed',                  # Eje Y: nuevos casos confirmados
    color='steelblue',                  # Color uniforme
    alpha=0.6,                          # Transparencia para ver concentración de puntos
    s=60,                               # Tamaño de los puntos
    edgecolor='black'                   # Borde negro para contraste
)

# Título y etiquetas
plt.title('Relación entre Temperatura Promedio y Nuevos Casos de COVID-19', fontsize=14, weight='bold')
plt.xlabel('Temperatura Promedio (°C)', fontsize=12)
plt.ylabel('Nuevos Casos Confirmados', fontsize=12)

# Cuadrícula y formato general
plt.grid(True, linestyle='--', alpha=0.5)  # Cuadrícula suave
plt.tight_layout()                         # Ajusta los márgenes automáticamente

# Mostrar gráfico
plt.show()
```

Infraestructura médica vs. mortalidad

```
try:
    df_byogenesis
except NameError:
    print('df_byogenesis no está definido. Ejecuta primero la celda que carga el CSV.')
else:
    required_cols = [
        'country_name',
        'nurses_per_1000',
        'physicians_per_1000',
        'new_deceased'
    ]
    if not all(col in df_byogenesis.columns for col in required_cols):
        print('Faltan columnas necesarias en el DataFrame:',
              [col for col in required_cols if col not in df_byogenesis.columns])
    else:
        # Calcular promedios por país
        df_comp = df_byogenesis.groupby('country_name', as_index=False).agg({
            'nurses_per_1000': 'mean',
            'physicians_per_1000': 'mean',
            'new_deceased': 'mean'
        })

        # Renombrar columnas
        df_comp.rename(columns={
            'nurses_per_1000': 'Promedio enfermeras por 1000 habitantes',
            'physicians_per_1000': 'Promedio médicos por 1000 habitantes',
            'new_deceased': 'Promedio de nuevos fallecidos por día'
        }, inplace=True)

        # Escalar tamaño de burbujas
        df_comp['Tamaño burbuja'] = (
            df_comp['Promedio de nuevos fallecidos por día'] / df_comp['Promedio de nuevos fallecidos por día'].max() * 1500
        )
```

```python
plt.figure(figsize=(10, 6))

# Gráfico de dispersión
ax = sns.scatterplot(
    x='Promedio enfermeras por 1000 habitantes',
    y='Promedio médicos por 1000 habitantes',
    size='Tamaño burbuja',
    hue='country_name',
    data=df_comp,
    palette = {
        'Argentina': '#1f77b4',
        'Brazil': '#ff7f0e',
        'Chile': '#2ca02c',
        'Colombia': '#d62728',
        'Mexico': '#9467bd',
        'Peru': '#8c564b'
    },
    sizes=(50, 1500),
    alpha=0.7,
    edgecolor='black',
    legend='full'
)

# Quitar la leyenda del tamaño de burbujas (solo dejar la de países)
handles, labels = ax.get_legend_handles_labels()
# Encontrar el índice donde empieza la leyenda de tamaño
if "Tamaño burbuja" in labels:
    size_index = labels.index("Tamaño burbuja")
    handles = handles[:size_index]
    labels = labels[:size_index]
ax.legend(handles, labels, title='País', bbox_to_anchor=(1.05, 1), loc='upper left')
# Eliminar el texto "country_name" que aparece debajo del título de la leyenda
legend = ax.get_legend()
if legend:
    for text in legend.texts:
        if text.get_text() == "country_name":
            text.set_visible(False)


# Agregar etiquetas con las dosis acumuladas
for i, row in df_comp.iterrows():
    plt.text(
        row['Promedio enfermeras por 1000 habitantes'],
        row['Promedio médicos por 1000 habitantes'] + 0.02 * df_comp['Promedio médicos por 1000 habitantes'].max(),
        f"{row['Promedio de nuevos fallecidos por día']:.2f}",
        fontsize=9,
        ha='center'
    )


plt.title('Infraestructura médica vs Tasa de mortalidad por país\n(tamaño burbuja = Suma de nuevos fallecidos por día)')
plt.xlabel('Promedio enfermeras por 1000 habitantes')
plt.ylabel('Promedio médicos por 1000 habitantes')
plt.tight_layout()
plt.show()
```

Relación entre infraestructura médica y dosis administradas

```python
try:
    df_byogenesis
except NameError:
    print('df_byogenesis no está definido. Ejecuta primero la celda que carga el CSV.')
else:
    required_cols = [
        'country_name',
        'nurses_per_1000',
        'physicians_per_1000',
        'cumulative_vaccine_doses_administered'
    ]
    if not all(col in df_byogenesis.columns for col in required_cols):
        print('Faltan columnas necesarias en el DataFrame:',
              [col for col in required_cols if col not in df_byogenesis.columns])
    else:
        # Calcular promedios por país
        df_comp = df_byogenesis.groupby('country_name', as_index=False).agg({
            'nurses_per_1000': 'mean',
            'physicians_per_1000': 'mean',
            'cumulative_vaccine_doses_administered': 'mean'
        })

        # Renombrar columnas
        df_comp.rename(columns={
            'nurses_per_1000': 'Promedio enfermeras por 1000 habitantes',
            'physicians_per_1000': 'Promedio médicos por 1000 habitantes',
            'cumulative_vaccine_doses_administered': 'Promedio de dosis de vacunas administradas'
        }, inplace=True)

        # Escalar tamaño de burbujas
        df_comp['Tamaño burbuja'] = (
            df_comp['Promedio de dosis de vacunas administradas'] / df_comp['Promedio de dosis de vacunas administradas'].max() * 1500
        )
```

```python
plt.figure(figsize=(10, 6))

# Gráfico de dispersión
ax = sns.scatterplot(
    x='Promedio enfermeras por 1000 habitantes',
    y='Promedio médicos por 1000 habitantes',
    size='Tamaño burbuja',
    hue='country_name',
    data=df_comp,
    palette = {
        'Argentina': '#1f77b4',
        'Brazil': '#ff7f0e',
        'Chile': '#2ca02c',
        'Colombia': '#d62728',
        'Mexico': '#9467bd',
        'Peru': '#8c564b'
    },
    sizes=(50, 1500),
    alpha=0.7,
    edgecolor='black',
    legend='full'
)

# Quitar la leyenda del tamaño de burbujas (solo dejar la de países)
handles, labels = ax.get_legend_handles_labels()
# Encontrar el índice donde empieza la leyenda de tamaño
if "Tamaño burbuja" in labels:
    size_index = labels.index("Tamaño burbuja")
    handles = handles[:size_index]
    labels = labels[:size_index]
ax.legend(handles, labels, title='País', bbox_to_anchor=(1.05, 1), loc='upper left')
# Eliminar el texto "country_name" que aparece debajo del título de la leyenda
legend = ax.get_legend()
if legend:
    for text in legend.texts:
        if text.get_text() == "country_name":
            text.set_visible(False)


# Agregar etiquetas con las dosis acumuladas
for i, row in df_comp.iterrows():
    plt.text(
        row['Promedio enfermeras por 1000 habitantes'],
        row['Promedio médicos por 1000 habitantes'] + 0.02 * df_comp['Promedio médicos por 1000 habitantes'].max(),
        f"{row['Promedio de dosis de vacunas administradas']:.2f}",
        fontsize=9,
        ha='center'
    )


plt.title('Infraestructura médica vs Dosis de Vacunas Administradas por país\n(tamaño burbuja = Dosis de vacunas administradas)')
plt.xlabel('Promedio enfermeras por 1000 habitantes')
plt.ylabel('Promedio médicos por 1000 habitantes')
plt.tight_layout()
plt.show()
```

Relación entre vacunación, aumento y recuperación de afectados

```python
plt.figure(figsize=(9,6))  # Tamaño del gráfico

# Diagrama de dispersión simple
sns.scatterplot(
    data=df_byogenesis,
    x='cumulative_vaccine_doses_administered',              # Eje X: dosis acumuladas de vacuna
    y='new_confirmed',                  # Eje Y: nuevos casos confirmados
    color='steelblue',                  # Color uniforme
    alpha=0.6,                          # Transparencia para ver concentración de puntos
    s=60,                               # Tamaño de los puntos
    edgecolor='black'                   # Borde negro para contraste
)

# Título y etiquetas
plt.title('Relación entre Dosis Administradas y Nuevos Casos de COVID-19', fontsize=14, weight='bold')
plt.xlabel('Dosis de Vacunas Administradas', fontsize=12)
plt.ylabel('Nuevos Casos Confirmados', fontsize=12)

# Cuadrícula y formato general
plt.grid(True, linestyle='--', alpha=0.5)  # Cuadrícula suave
plt.tight_layout()                         # Ajusta los márgenes automáticamente

# Mostrar gráfico
plt.show()
```

```
plt.figure(figsize=(9,6))  # Tamaño del gráfico

# Diagrama de dispersión simple
sns.scatterplot(
    data=df_byogenesis,
    x='cumulative_vaccine_doses_administered',               # Eje X: dosis acumuladas de vacuna
    y='new_recovered',                 # Eje Y: nuevos casos recuperados
    color='steelblue',                 # Color uniforme
    alpha=0.6,                         # Transparencia para ver concentración de puntos
    s=60,                              # Tamaño de los puntos
    edgecolor='black'                  # Borde negro para contraste
)

# Título y etiquetas
plt.title('Relación entre Dosis Administradas y Nuevos Casos Recuperados de COVID-19', fontsize=14, weight='bold')
plt.xlabel('Dosis de Vacunas Administradas', fontsize=12)
plt.ylabel('Nuevos Casos Recuperados', fontsize=12)

# Cuadrícula y formato general
plt.grid(True, linestyle='--', alpha=0.5)  # Cuadrícula suave
plt.tight_layout()                         # Ajusta los márgenes automáticamente

# Mostrar gráfico
plt.show()
```

AVANCE N°3

Importar Librerías

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
```

EDA

Análisis Temporal

En esta sección dejo el código utilizado para obtener los gráficos que se muestran en la sección Análisis Temporal.

Tasa de cambio de casos confirmados en general

```
plt.figure(figsize=(12,5))

# --- Gráfico 1: Diferencia diaria de casos (variación) ---
plt.subplot(1, 2, 1)
df_byogenesis['new_confirmed'].diff().plot(color='tomato')
plt.title('Variación diaria de casos confirmados')
plt.xlabel('Registros')
plt.ylabel('Diferencia de casos respecto al día anterior')
plt.grid(True)

# --- Gráfico 2: Casos confirmados diarios ---
plt.subplot(1, 2, 2)
df_byogenesis['new_confirmed'].plot(color='steelblue')
plt.title('Casos confirmados diarios')
plt.xlabel('Registros')
plt.ylabel('Número de casos')
plt.grid(True)

plt.tight_layout()
plt.show()
```

Promedio Móvil de Casos Confirmados por País y Promedio Móvil de Muertes por País

```
#Primero creo una función que calcula el promedio móvil de una columna dada
def promedio_movil(df, columna, ventana=7):
    """
    Parámetros:
        df: DataFrame
        columna: str -> nombre de la columna a suavizar
        ventana: int -> tamaño de la ventana (por defecto 7 días)
    Devuelve:
        Serie con el promedio móvil
    """
    return df[columna].rolling(window=ventana, min_periods=1).mean()
```

```
# Asegurar que 'date' sea tipo fecha
df_byogenesis['date'] = pd.to_datetime(df_byogenesis['date'])

# Ordenar para el cálculo correcto del promedio móvil
df_byogenesis = df_byogenesis.sort_values(by=['country_name', 'date'])

# Aplicar función a cada país
df_byogenesis['pm_casos'] = df_byogenesis.groupby('country_name').apply(lambda x: promedio_movil(x, 'new_confirmed')).reset_index(level=0, drop=True)
df_byogenesis['pm_muertes'] = df_byogenesis.groupby('country_name').apply(lambda x: promedio_movil(x, 'new_deceased')).reset_index(level=0, drop=True)
```

```python
pais = 'Argentina'
df_pais = df_byogenesis[df_byogenesis['country_name'] == pais]

plt.figure(figsize=(10,5))

# Serie diaria original
sns.lineplot(
    data=df_pais,
    x='date',
    y='new_confirmed',
    label='Casos diarios',
    alpha=0.4,
    linewidth=1.2,
    color='gray'
)

# Promedio móvil
sns.lineplot(
    data=df_pais,
    x='date',
    y='pm_casos',
    label='Promedio móvil (7 días)',
    color='red',
    linewidth=2,
    ci=None  # Quitar banda de confianza (error bars)
)

plt.title(f'Evolución de casos diarios y promedio móvil - {pais}')
plt.xlabel('Fecha')
plt.ylabel('Número de casos')
plt.legend()
plt.tight_layout()
plt.show()
```

```python
pais = 'Argentina'
df_pais = df_byogenesis[df_byogenesis['country_name'] == pais]

plt.figure(figsize=(10,5))

# Serie diaria original
sns.lineplot(
    data=df_pais,
    x='date',
    y='new_deceased',
    label='Muertes diarias',
    alpha=0.4,
    linewidth=1.2,
    color='gray'
)

# Promedio móvil
sns.lineplot(
    data=df_pais,
    x='date',
    y='pm_muertes',
    label='Promedio móvil (7 días)',
    color='red',
    linewidth=2,
    ci=None  # Quitar banda de confianza (error bars)
)

plt.title(f'Evolución de muertes diarias y promedio móvil - {pais}')
plt.xlabel('Fecha')
plt.ylabel('Número de muertes')
plt.legend()
plt.tight_layout()
plt.show()
```

Análisis Espacial

Total de muertes por País

```python
#Importo ticker para manipular la barra de color del mapa
from matplotlib import ticker as mticker
# Agrupar tu dataset por país
df_muertes = df_byogenesis.groupby('country_name', as_index=False)['new_deceased'].sum()

# Cargar el mapa y filtrar América
world = gpd.read_file("https://naciscdn.org/naturalearth/110m/cultural/ne_110m_admin_0_countries.zip")
america = world[world['CONTINENT'].isin(['North America', 'South America'])]
merged = america.merge(df_muertes, how='left', left_on='NAME', right_on='country_name')

# Crear figura y ejes
fig, ax = plt.subplots(figsize=(15, 8))

# Graficar el mapa
merged.plot(
    column='new_deceased',
    cmap='Reds',
    legend=True,
    ax=ax,
    missing_kwds={"color": "lightgrey", "label": "Sin datos"},
    legend_kwds={'label': "Número de muertes por país", 'orientation': "horizontal", 'shrink': 0.7}
)

# Buscar el colorbar que creó geopandas
cbar = ax.get_figure().axes[-1]  # normalmente es el último eje
cbar.xaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))  # muestra enteros con separador de miles

# Agregar etiquetas con número exacto de muertes sobre cada país
for idx, row in merged.iterrows():
    if not pd.isna(row['new_deceased']):
        x, y = row['geometry'].representative_point().coords[0]
        ax.text(x, y, f"{int(row['new_deceased']):,}", ha='center', va='center', fontsize=8, color='black')

# Personalizar
ax.set_title("Mapa de muertes por COVID-19 en América (totales por país)", fontsize=16, fontweight='bold')
ax.axis('off')
plt.show()
```

Total de Casos Confirmados por País

```python
#Importo ticker para manipular la barra de color del mapa
from matplotlib import ticker as mticker
# Agrupar tu dataset por país
df_casos = df_byogenesis.groupby('country_name', as_index=False)['new_confirmed'].sum()

# Cargar el mapa y filtrar América
world = gpd.read_file("https://naciscdn.org/naturalearth/110m/cultural/ne_110m_admin_0_countries.zip")
america = world[world['CONTINENT'].isin(['North America', 'South America'])]
merged = america.merge(df_casos, how='left', left_on='NAME', right_on='country_name')

# Crear figura y ejes
fig, ax = plt.subplots(figsize=(15, 8))

# Graficar el mapa
merged.plot(
    column='new_confirmed',
    cmap='Reds',
    legend=True,
    ax=ax,
    missing_kwds={"color": "lightgrey", "label": "Sin datos"},
    legend_kwds={'label': "Número de casos confirmados por país", 'orientation': "horizontal", 'shrink': 0.7}
)

# Buscar el colorbar que creó geopandas
cbar = ax.get_figure().axes[-1]  # normalmente es el último eje
cbar.xaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))  # muestra enteros con separador de miles

# Agregar etiquetas con número exacto de casos confirmados sobre cada país
for idx, row in merged.iterrows():
    if not pd.isna(row['new_confirmed']):
        x, y = row['geometry'].representative_point().coords[0]
        ax.text(x, y, f"{int(row['new_confirmed']):,}", ha='center', va='center', fontsize=8, color='black')

# Personalizar
ax.set_title("Mapa de casos confirmados por COVID-19 en América (totales por país)", fontsize=16, fontweight='bold')
ax.axis('off')
plt.show()
```

Otras Correlaciones

Relación entre casos confirmados, fallecimientos y las población urbana y rural

```
# Seleccionar las columnas relevantes
vars_interes = ['population_urban', 'population_rural', 'new_confirmed', 'new_deceased']

# Calcular la matriz de correlación
corr = df_byogenesis[vars_interes].corr()

# Crear la máscara para ocultar correlaciones débiles
mask = (corr.abs() < 0.5)  # True donde la correlación es menor a 0.5 o mayor a -0.5

# Configurar el gráfico
plt.figure(figsize=(8, 6))
sns.heatmap(
    corr,
    annot=True,             # Muestra los valores en cada celda
    fmt=".2f",              # Dos decimales
    cmap="coolwarm",        # Colores rojo-azul
    mask=mask,              # Aplica la máscara
    center=0,               # Centra la escala en 0
    cbar_kws={'label': 'Coeficiente de correlación'}
)

plt.title("Heatmap de correlaciones (|r| ≥ 0.5)", fontsize=14, fontweight='bold')
plt.tight_layout()
plt.show()
```

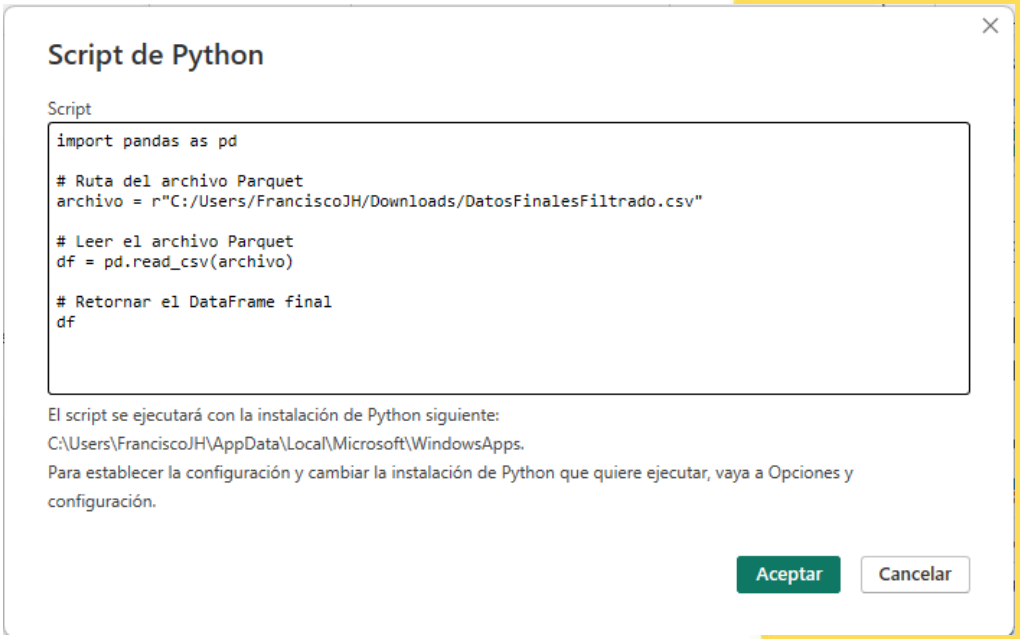
Relación entre la la acumulación de fallecimientos y casos confirmados en las áreas urbanas y rurales

```
# Seleccionar las columnas relevantes
vars_interes = ['population_urban', 'population_rural', 'cumulative_confirmed', 'cumulative_deceased']

# Calcular la matriz de correlación
corr = df_byogenesis[vars_interes].corr()

# Crear la máscara para ocultar correlaciones débiles
mask = (corr.abs() < 0.75)  # True donde la correlación es menor a 0.5 o mayor a -0.5

# Configurar el gráfico
plt.figure(figsize=(8, 6))
sns.heatmap(
    corr,
    annot=True,             # Muestra los valores en cada celda
    fmt=".2f",              # Dos decimales
    cmap="coolwarm",        # Colores rojo-azul
    mask=mask,              # Aplica la máscara
    center=0,               # Centra la escala en 0
    cbar_kws={'label': 'Coeficiente de correlación'}
)

plt.title("Heatmap de correlaciones (|r| ≥ 0.5)", fontsize=14, fontweight='bold')
plt.tight_layout()
plt.show()
```

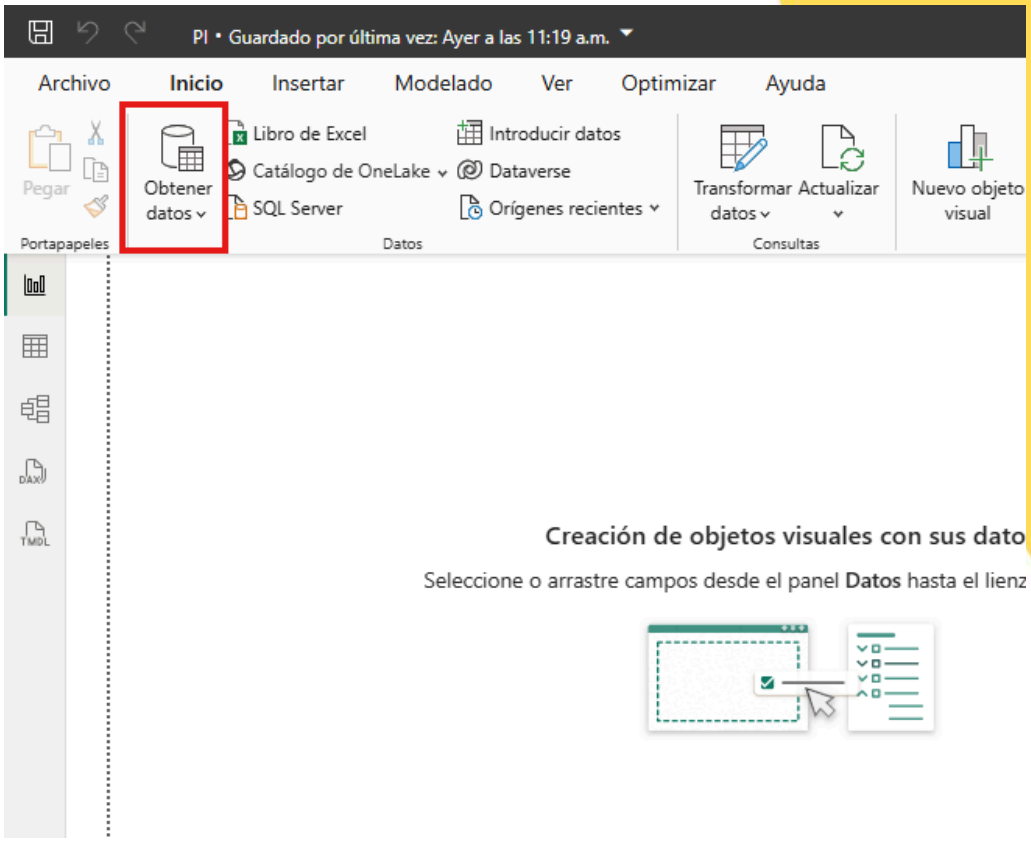
AVANCE N°4

Conexión de Python con Power BI

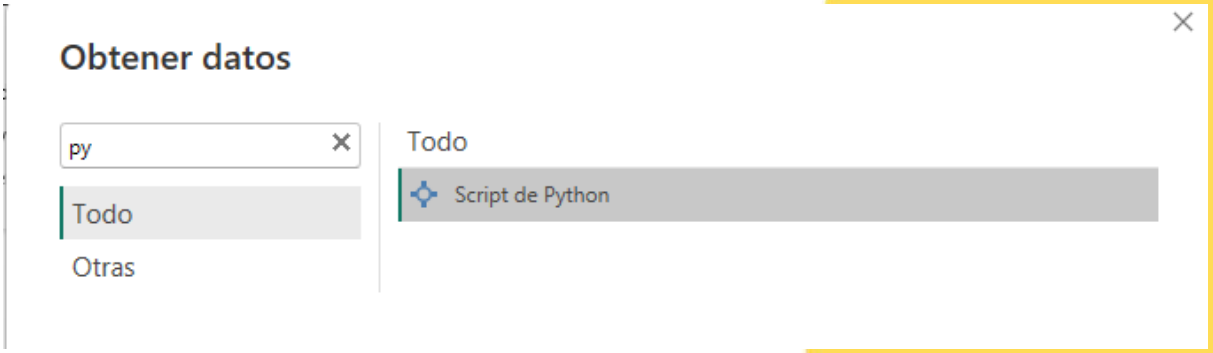
Para importar el CSV a Power BI utilice el siguiente script.



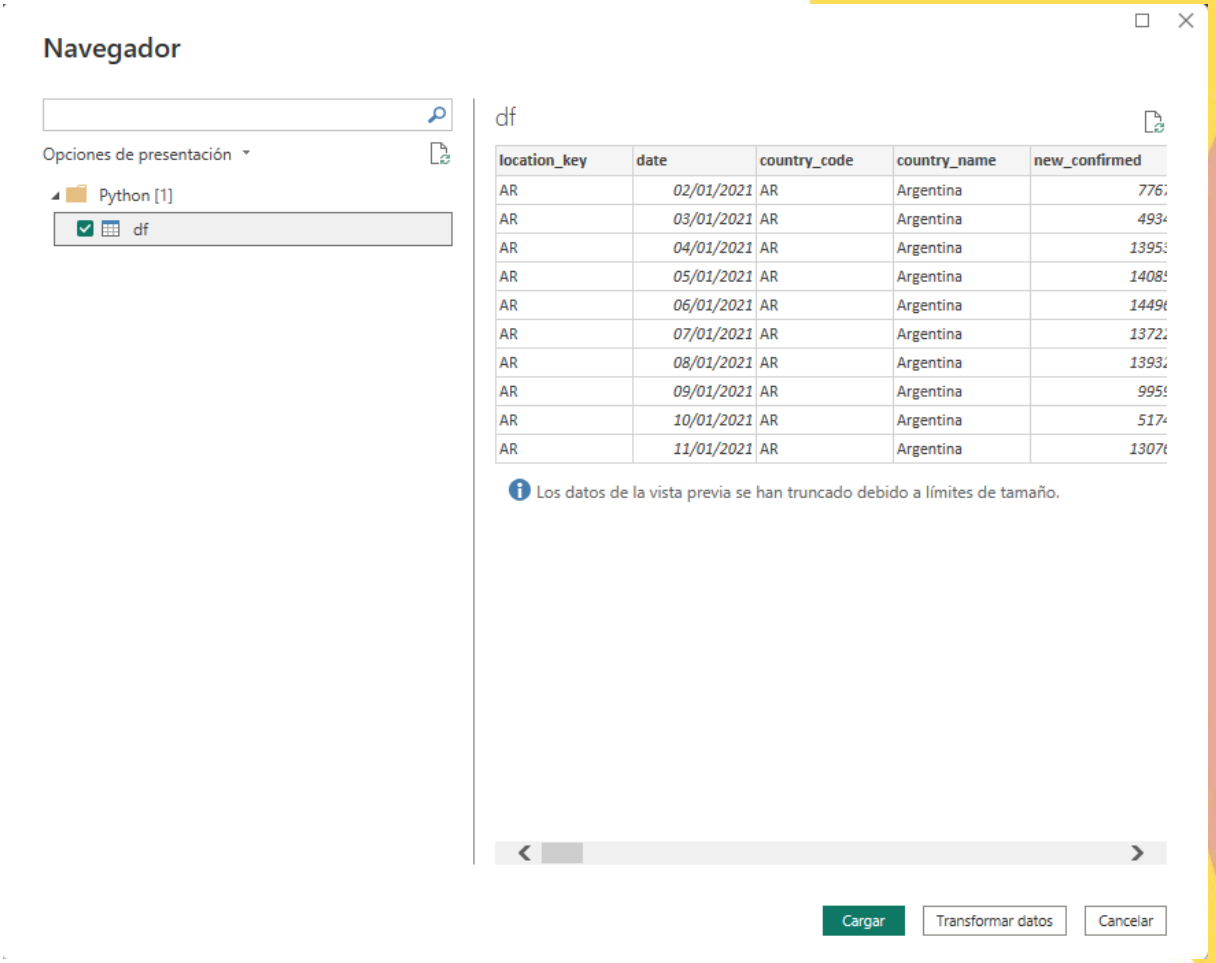
Esto lo hice desde la opción “Obtener datos”, en la pestaña “Inicio”.



Desde allí elegí la opción “Script de Python”, donde cargué el script anterior.



Una vez establecida la conexión, seleccione mi Dataframe, y cargue los datos.



Predicción de nuevos casos

Este fue el código utilizado para generar una predicción de nuevos posibles casos confirmados de COVID-19. La predicción se ve en el tablero de Power BI.

```python
import pandas as pd
from prophet import Prophet
import matplotlib.pyplot as plt
import matplotlib.cm as cm # Necesario para usar una paleta de colores
import numpy as np # Importar numpy para la paleta de colores

#Cargar y preparar los datos
df = pd.read_csv("C:/Users/FranciscoJH/Downloads/DatosFinalesFiltrado.csv")

# Renombrar columnas a formato Prophet
df = df.rename(columns={'date': 'ds', 'new_confirmed': 'y', 'country_name': 'country'})

# Convertir la columna de fecha a tipo datetime (si no lo está ya)
df['ds'] = pd.to_datetime(df['ds'])

#Predicción por País (Loop)
# Obtener la lista única de países para iterar
paises = df['country'].unique()

# DataFrame vacío para almacenar todas las predicciones
todas_predicciones = pd.DataFrame()

# Iterar sobre cada país
for pais in paises:
    print(f"Prediciendo para: {pais}")

    # Filtrar los datos para el país actual
    df_pais = df[df['country'] == pais].copy()

    # Crear y entrenar el modelo (se entrena uno nuevo para cada país)
    # Se añade la configuración para manejar potencial estacionalidad
    modelo = Prophet(yearly_seasonality=True, weekly_seasonality=True, daily_seasonality=False)
    # Usar 'df_pais' que solo tiene los datos del país actual
    modelo.fit(df_pais)

    # Crear el dataframe futuro: 15 meses adicionales
    futuro = modelo.make_future_dataframe(periods=15, freq='M')

    # Hacer la predicción
    prediccion_pais = modelo.predict(futuro)

    #Eliminar valores negativos de la predicción
    # La predicción estimada es 'yhat'. Si es menor que 0, se establece a 0.
    prediccion_pais['yhat'] = prediccion_pais['yhat'].apply(lambda x: max(0, x))
    # También se aplica a los límites inferiores de confianza (yhat_lower)
    prediccion_pais['yhat_lower'] = prediccion_pais['yhat_lower'].apply(lambda x: max(0, x))

    # Añadir la columna de país a las predicciones
    prediccion_pais['country'] = pais

    # Seleccionar las columnas relevantes y añadirlas al dataframe global
    predicciones_filtradas = prediccion_pais[['ds', 'country', 'yhat', 'yhat_lower', 'yhat_upper']]
    todas_predicciones = pd.concat([todas_predicciones, predicciones_filtradas], ignore_index=True)

# --- 4. Guardar el resultado ---
todas_predicciones.to_csv(
    "C:/Users/FranciscoJH/Downloads/prediccion_covid_por_pais.csv", index=False
)
print("\nPredicciones guardadas en 'prediccion_covid_por_pais.csv'")
```

```
# Visualizar el resultado


plt.figure(figsize=(30, 8))

# Generar una paleta de colores para todos los países
# Usamos un mapa de colores como 'viridis' o 'jet' para asegurar distinción
colores = cm.get_cmap('jet', len(paises))

# Iterar y graficar cada país en el mismo eje
for i, pais in enumerate(paises):
    df_plot = todas_predicciones[todas_predicciones['country'] == pais]
    color = colores(i)

    # Graficar solo la línea de predicción 'yhat'
    plt.plot(df_plot['ds'],
             df_plot['yhat'],
             label=f'Predicción: {pais}',
             color=color,
             linewidth=2)

    # Opcional: Graficar los datos históricos como pequeños puntos para contexto
    df_hist = df[df['country'] == pais]
    #plt.scatter(df_hist['ds'], df_hist['y'], color=color, alpha=0.4, s=5, label=f'Datos Históricos: {pais}' if i == 0 else "")


# Configuración del gráfico final
plt.title("Predicción Comparativa de Nuevos Casos de COVID-19 por País",
          fontsize=16, # <-- Título más grande
          fontweight='bold')
# 2. Etiquetas de Ejes: Aumenta el 'fontsize'
plt.xlabel("Fecha", fontsize=14) # <-- Etiqueta X más grande
plt.ylabel("Nuevos Casos Diarios (Predicción Ajustada $\geq 0$)", fontsize=14) # <-- Etiqueta Y más grande

# 3. Etiquetas de Ticks (Números/Fechas en los ejes)
# Esta función ajusta el tamaño de los números y fechas en los ejes.
plt.tick_params(axis='both', which='major', labelsize=12) # <-- Ticks más grandes

# 4. Leyenda: Aumenta el 'fontsize'
plt.legend(loc='upper left',
           bbox_to_anchor=(1.01, 1),
           fontsize=20) # <-- Texto de la leyenda más grande
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout() # Ajustar el espacio para la leyenda
plt.show()
```

Predicción de fallecimientos

```python
import pandas as pd
from prophet import Prophet
import matplotlib.pyplot as plt
import matplotlib.cm as cm # Necesario para usar una paleta de colores
import numpy as np # Importar numpy para la paleta de colores

#Cargar y preparar los datos
df = pd.read_csv("C:/Users/FranciscoJH/Downloads/DatosFinalesFiltrado.csv")

# Renombrar columnas a formato Prophet
df = df.rename(columns={'date': 'ds', 'new_deceased': 'y', 'country_name': 'country'})

# Convertir la columna de fecha a tipo datetime (si no lo está ya)
df['ds'] = pd.to_datetime(df['ds'])

#Predicción por País (Loop)
# Obtener la lista única de países para iterar
paises = df['country'].unique()

# DataFrame vacío para almacenar todas las predicciones
todas_predicciones = pd.DataFrame()

# Iterar sobre cada país
for pais in paises:
    print(f"Prediciendo para: {pais}")

    # Filtrar los datos para el país actual
    df_pais = df[df['country'] == pais].copy()

    # Crear y entrenar el modelo (se entrena uno nuevo para cada país)
    # Se añade la configuración para manejar potencial estacionalidad
    modelo = Prophet(yearly_seasonality=True, weekly_seasonality=True, daily_seasonality=False)
    # Usar 'df_pais' que solo tiene los datos del país actual
    modelo.fit(df_pais)

    # Crear el dataframe futuro: 15 meses adicionales
    futuro = modelo.make_future_dataframe(periods=15, freq='M')

    # Hacer la predicción
    prediccion_pais = modelo.predict(futuro)

    #Eliminar valores negativos de la predicción
    # La predicción estimada es 'yhat'. Si es menor que 0, se establece a 0.
    prediccion_pais['yhat'] = prediccion_pais['yhat'].apply(lambda x: max(0, x))
    # También se aplica a los límites inferiores de confianza (yhat_lower)
    prediccion_pais['yhat_lower'] = prediccion_pais['yhat_lower'].apply(lambda x: max(0, x))

    # Añadir la columna de país a las predicciones
    prediccion_pais['country'] = pais

    # Seleccionar las columnas relevantes y añadirlas al dataframe global
    predicciones_filtradas = prediccion_pais[['ds', 'country', 'yhat', 'yhat_lower', 'yhat_upper']]
    todas_predicciones = pd.concat([todas_predicciones, predicciones_filtradas], ignore_index=True)

# --- 4. Guardar el resultado ---
todas_predicciones.to_csv(
    "C:/Users/FranciscoJH/Downloads/prediccion_muertes_covid_por_pais.csv", index=False
)
print("\nPredicciones guardadas en 'prediccion_muertes_covid_por_pais.csv'")
```

```python
# Visualizar el resultado


plt.figure(figsize=(30, 8))

# Generar una paleta de colores para todos los países
# Usamos un mapa de colores como 'viridis' o 'jet' para asegurar distinción
colores = cm.get_cmap('jet', len(paises))

# Iterar y graficar cada país en el mismo eje
for i, pais in enumerate(paises):
    df_plot = todas_predicciones[todas_predicciones['country'] == pais]
    color = colores(i)

    # Graficar solo la línea de predicción 'yhat'
    plt.plot(df_plot['ds'],
             df_plot['yhat'],
             label=f'Predicción: {pais}',
             color=color,
             linewidth=2)

    # Opcional: Graficar los datos históricos como pequeños puntos para contexto
    df_hist = df[df['country'] == pais]
    #plt.scatter(df_hist['ds'], df_hist['y'], color=color, alpha=0.4, s=5, label=f'Datos Históricos: {pais}' if i == 0 else "")


# Configuración del gráfico final
plt.title("Predicción Comparativa de Fallecimientos de COVID-19 por País",
          fontsize=16, # <-- Título más grande
          fontweight='bold')
# 2. Etiquetas de Ejes: Aumenta el 'fontsize'
plt.xlabel("Fecha", fontsize=14) # <-- Etiqueta X más grande
plt.ylabel("Nuevos Fallecimientos Diarios (Predicción Ajustada $\geq 0$)", fontsize=14) # <-- Etiqueta Y más grande

# 3. Etiquetas de Ticks (Números/Fechas en los ejes)
# Esta función ajusta el tamaño de los números y fechas en los ejes.
plt.tick_params(axis='both', which='major', labelsize=12) # <-- Ticks más grandes

# 4. Leyenda: Aumenta el 'fontsize'
plt.legend(loc='upper left',
           bbox_to_anchor=(1.01, 1),
           fontsize=20) # <-- Texto de la leyenda más grande
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout() # Ajustar el espacio para la leyenda
plt.show()
```

Creación de Dashboards en Power BI

El dashboard lo encontrará en `PI.pbix`.

Visualizaciones Estáticas vs Interactivas

Visualización	Ventajas
Estática	• Ideales para reportes impresos o presentaciones fijas. • Consistencia y control total del formato. • No requieren conexión a Internet ni plataformas adicionales. • Adecuadas para auditorías o registros históricos.
	• Los usuarios pueden filtrar, seleccionar rangos de fechas o hacer drill-down en los datos. • La interacción ayuda a descubrir

Visualización	Ventajas
Interactiva	patrones, anomalías y relaciones que no se ven en gráficos fijos. ● Los dashboards pueden conectarse a fuentes de datos en tiempo real. ● Cada perfil (gerente, analista, técnico) puede visualizar los datos relevantes para su área.

Conclusiones y Recomendaciones

Luego de realizar el análisis, podemos observar los siguientes resultados:

- Brasil es el país con más casos y fallecimientos, por ende, también fue el país que más vacunas administró.
- México es el segundo país que más fallecidos registró, y por ende el segundo país que más vacunas otorgó. Hay que tener en cuenta que es el país con mayor densidad poblacional, algo muy importante teniendo en cuenta que el COVID-19 se contagia fácilmente.
- Argentina es el segundo país con más casos confirmados, pero el segundo con menos fallecidos, quizás se deba a que fue el tercer país que más vacunas administró. Además, es el país con menor densidad poblacional.
- Colombia es el segundo país con mayor densidad poblacional.
- Chile es el país con el promedio de médicos y enfermeros cada 1000 habitantes seguido por Brasil. En el caso de Chile, es el país con mejor expectativa de vida, y en el caso brasilero, su expectativa de vida es similar al resto de países (menos Chile).
- Brasil y México son los países con el PBI más alto.
- Los contagios y fallecidos suben en temperaturas de entre 20°C y 30°C.
- Los casos y fallecimientos suben en países con mayor población.
- Brasil y México son los países con mayor población de riesgo (60 años o más).
- El orden de los países en cuanto a mayor PBI per cápita son:
    - Chile
    - Argentina
    - México
    - Brasil
    - Perú
    - Colombia
- A medida que se administran más vacunas, ha subido la cantidad de recuperados.
- Se estima que Brasil tendrá la mayor cantidad de fallecidos para enero de 2023, y el segundo país con mayor cantidad de nuevos casos para la misma fecha.
- Se estima que para enero de 2023, Argentina tendrá la mayor cantidad de casos confirmados, y será el segundo país con más fallecimientos.
- Se estima que Colombia será el tercer país con más casos confirmados y el segundo en fallecimientos para enero de 2023.

- Se estima que México será el país con menos casos confirmados y menos fallecimientos a enero de 2023.
- Al igual que México, Chile será el país con menos fallecimientos a enero de 2023. Y se estima que para esa fecha será el segundo país con menos contagios.

Teniendo en cuenta estos resultados, como analista de datos, recomiendo a Byogenesis lo siguiente:

- Expandirse principalmente a Brasil: Recomiendo que la mayor cantidad de laboratorios se instalen en este país. Esto debido a que: fue el país más afectado en el pasado y además se estima que sea uno de los más afectados a futuro, por ello necesitaran muchas dosis de vacunas, las cuales Byogenesis puede otorgar. Además, Brasil es el país con el mayor PBI, por lo que es más probable que pueda costear las vacunas que nosotros ofrecemos. Otro dato importante de este país es la buena cantidad de médicos y enfermeros cada 1000 habitantes que posee, los cuales son necesarios para la administración de vacunas. También hay que tener en cuenta que es un país cálido, lo cual, como vimos, aumenta la probabilidad de contagios. Por último, es importante destacar que el futuro de Brasil es riesgoso.
- Uno de los dos países donde más laboratorios pondría detrás de Brasil, sería México: Recomiendo esto porque es el segundo país con el mejor PBI, y el tercero con el mejor PBI per cápita, por lo que es más probable que su gobierno costee un programa de vacunación. Es importante destacar que es el país de mayor densidad poblacional y además el segundo país con mayor población de riesgo (personas de 60 años o más).
- El segundo país donde más laboratorios pondría detrás de Brasil y junto a México sería Argentina: Fue el segundo país con más casos, pero fue el segundo con menos fallecidos, esto gracias a que fue el tercer país que más vacunas administró. Es decir, que Argentina conoce los beneficios de la vacuna, por lo que es probable que las solicite. Además, las estimaciones futuras de Argentina no son alentadoras.
- El cuarto país donde más invertiría en la instalación de laboratorios sería Colombia: Un país cálido cuya proyección a futuro no es buena (en cuanto a fallecimientos y nuevos casos), por lo que es probable que necesite vacunas. Además es el segundo país con mayor densidad poblacional, por lo que es probable que aumenten los contagios. También es importante destacar que es el país con el peor número de médicos y enfermeros por cada 1000 habitantes, por lo que no posee demasiado personal sanitario para administrar vacunas.
- El quinto país donde invertiría sería Perú: Su PBI es el más bajo de los 6 países, y el PBI per cápita es el segundo más bajo, por lo que considero que al gobierno peruano le costaría invertir en un programa de vacunación. Además, su proyección a futuro no es ideal, pero tampoco terrible, ya que es el cuarto país en la estimación de posibles fallecidos y casos confirmados para enero de 2023. También hay que destacar que es uno de los países con menor presencia de médicos y enfermeros por cada 1000 habitantes. Por último, es importante destacar que es el segundo país con menor población de riesgo.
- El país en el que menos invertiría sería Chile: Este es un país que tiene una buena infraestructura sanitaria, pero se estima que a futuro sea de los menos afectados. Además fue el país menos afectado, por ende el que menos vacunas administró.

En resumen, invertiría en los países siguiendo este orden:

1. Brasil
2. México y Argentina

3. Colombia
4. Perú
5. Chile

La estrategia general recomendada para Biogenesys consiste en priorizar la instalación de laboratorios en países con alta necesidad sanitaria (gran cantidad de casos y fallecimientos en el pasado y a futuro) y capacidad económica suficiente para sostener programas de vacunación, comenzando por Brasil y México. En paralelo, mantener presencia limitada en países con menor proyección de casos, como Chile.

Reflexión Personal

Este proyecto validó que la calidad de los datos es la base de toda decisión estratégica, lo que requirió una dedicación significativa en la fase de limpieza y ETL. Como Analista de Datos, consolidé habilidades clave. Adquirí conocimientos en el modelo Prophet para la predicción de series de tiempo, gestionando activamente las limitaciones de los datos al extrapolar. Demostré la capacidad de realizar análisis multidimensionales al combinar y correlacionar variables diversas como la demografía, la incidencia de la enfermedad y el clima, lo cual fue esencial para formular la recomendación final. Finalmente, desarrollé la destreza de crear visualizaciones estratégicas en Python (heatmaps, gráficos comparativos), necesarias para la toma de decisiones. Además conecté este lenguaje a Power BI, asegurando que las conclusiones se puedan visualizar de manera interactiva, y así apoyar a la toma de decisiones.

Además, si tuviese que empezar nuevamente este proyecto, me concentraría aún más en el conocimiento de las variables, para hacer el proceso de limpieza de datos más eficiente, algo que me costó al realizar este proyecto.