

EXPANSIÓN ESTRATÉGICA DE LABORATORIOS PARA INVESTIGACIÓN Y VACUNACIÓN DE COVID-19

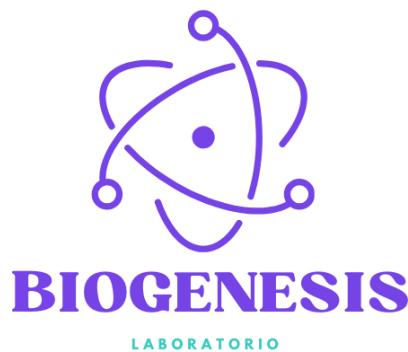
Nombre del autor: Arturo Ponce de León

Email: arturop1m@gmail.com

Cohorte: DA-FT02

Fecha de entrega: 03/06/2024

Institución:



El proyecto de **Expansión Estratégica de Laboratorios para Investigación y Vacunación de COVID-19** es una iniciativa impulsada por Biogenesisys, una empresa farmacéutica líder en el desarrollo y distribución de vacunas a nivel mundial. Esta empresa de origen Europeo, fue fundada en Barcelona – España en el año 2005.

Introducción

El objetivo principal del proyecto es identificar las ubicaciones óptimas para la expansión de laboratorios en América Latina, específicamente entre los países de Argentina, Chile, México, Perú, Colombia y Brasil. Estos laboratorios se enfocarán en la producción de vacunas, con el propósito de mejorar la respuesta a una pandemia y postpandemia, y asegurar un acceso más equitativo a las vacunas en la región.

Módulo 4

La pandemia de COVID-19 ha resaltado la necesidad crítica de mejorar las capacidades de producción y distribución de vacunas a nivel global. América Latina, una región significativamente afectada por la pandemia, presenta disparidades en la cobertura de vacunación y en la infraestructura médica. Este proyecto se enmarca en el contexto de una estrategia a largo plazo para fortalecer la resiliencia sanitaria de la región y estar mejor preparados ante futuras crisis sanitarias.

Objetivos del Proyecto:

- **Identificar Ubicaciones Estratégicas:** Utilizar análisis de datos para determinar las ubicaciones más adecuadas para nuevos laboratorios, basándose en la incidencia de COVID-19, tasas de vacunación, y disponibilidad de infraestructuras sanitarias.
- **Optimizar la Respuesta Pandémica:** Aumentar la capacidad de producción y distribución de vacunas en áreas con alta demanda y baja cobertura de vacunación.
- **Mejorar el Acceso a las Vacunas:** Asegurar un acceso más equitativo a las vacunas para la población de América Latina, mitigando los efectos de la pandemia y protegiendo a las comunidades más vulnerables.

Metodología:

- **Análisis Exploratorio de Datos:** Recolectar y analizar datos sobre la incidencia de COVID-19, tasas de vacunación, y disponibilidad de infraestructuras sanitarias en los países de interés.
- **Técnicas de Limpieza de Datos:** Asegurar la calidad y consistencia de los datos para facilitar un análisis preciso y decisiones estratégicas confiables.
- **Operaciones de ETL:** Eficientar la extracción, transformación y carga de datos para mejorar la eficacia del análisis y la toma de decisiones.
- **Desarrollo de Dashboards Interactivos:** Crear dashboards en Power BI para visualizar datos desde múltiples perspectivas, permitiendo una toma de decisiones informada y estratégica.

Módulo 4

Resultados Esperados:

- **Insights Valiosos:** Generar insights sobre posibles ubicaciones estratégicas para futuros laboratorios y centros de vacunación.
- **Toma de Decisiones Informada:** Facilitar la toma de decisiones de la directiva de Biogenesys para invertir en infraestructura que optimice la respuesta pandémica.
- **Mejora en la Cobertura de Vacunación:** Aumentar la cobertura de vacunación en regiones con baja inmunización y alta incidencia de COVID-19.
- **Fortalecimiento de la Resiliencia Sanitaria:** Contribuir al fortalecimiento de la infraestructura sanitaria en América Latina, preparando mejor a la región para futuras crisis de salud.

Desarrollo del proyecto

El desarrollo de este proyecto se dio a partir de una data recopilada por la compañía en un archivo en formato .CSV con mas de 12 millones de registros. El cual se trabajó mediante el lenguaje Python con la ayuda de la herramienta de Visual Studio Code.

A continuación, detallare las transformaciones realizadas y limpieza de datos.

1. Se creo un archivo en Python con todos los pasos necesarios para la limpieza y transformación de los datos.
2. Se declararon las librerías necesarias para desarrollar el proyecto.
3. Se procedió a cargar el archivo “data_latinoamerica.csv” mediante el comando de pandas .read_csv
4. Verificamos la carga correcta de los datos mediante el comando .info(), con el cual verificamos que se tienen mas de 12 millones de registros

Módulo 4

```
#importamos las librerias necesarias para desarrollar el proyecto
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import folium

1] ✓ 134s

- ✓ #comandos para instalar correctamente matplotlib y seaborn ...

#Ruta del archivo .CSV que se usará para el análisis
ruta_archivo = "data_latinoamerica.csv"

#Creación del DataFrame con los datos del archivo CSV
df_covid=pd.read_csv(ruta_archivo, parse_dates=['date'])

2] ✓ 3m 17.2s
```

Comprobamos que el dataset cargado tiene la cantidad de registros y columnas especificadas

```
#Con el comando .info() obtenemos la cantidad de entradas y columnas del DataFrame cargado
df_covid.info()

3] ✓ 0.7s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12216057 entries, 0 to 12216056
Data columns (total 50 columns):
 #   Name          Dtype  
 0   date          datetime64[ns]
 1   country_name  object  
 2   country_code  object  
 3   continent     object  
 4   lat           float64 
 5   lon           float64 
 6   population    int64  
 7   total_cases   int64  
 8   new_cases     int64  
 9   total_deaths  int64  
 10  new_deaths    int64  
 11  total_recovered int64 
 12  new_recovered int64 
 13  active_cases int64 
 14  critical_cases int64 
 15  total_tests   int64 
 16  new_tests     int64 
 17  tests_per_case float64 
 18  tests_per_fctn float64 
 19  tests_per_mill float64 
 20  iso3           object  
 21  iso2           object  
 22  iso             object  
 23  name           object  
 24  code            object  
 25  continent_code object  
 26  continent_name object  
 27  lat_min        float64 
 28  lat_max        float64 
 29  lon_min        float64 
 30  lon_max        float64 
 31  pop_min        float64 
 32  pop_max        float64 
 33  cases_min     float64 
 34  cases_max     float64 
 35  deaths_min    float64 
 36  deaths_max    float64 
 37  recovered_min float64 
 38  recovered_max float64 
 39  tests_min     float64 
 40  tests_max     float64 
 41  tests_mill    float64 
 42  tests_fctn    float64 
 43  tests_mill_mil float64
```

5. A continuación filtramos los datos para solo tener de los países requeridos: Argentina, Perú, Colombia, Chile, México y Brasil.
6. Observamos la cantidad de datos nulos en el dataframe cargado llamado dt_latinoamerica. Con lo que observamos que existen miles de datos nulos en gran cantidad de columnas.

Seleccionamos los países con los que se realizará el análisis: Colombia, Argentina, Chile, México, Perú y Brasil.

```
paises_selec=['Argentina','Peru','Chile','Colombia','Brazil','Mexico']
df_latinoamerica=df_covid[df_covid['country_name'].isin(paises_selec)]

5] ✓ 43.0s

#validamos la cantidad de registros y columnas del DataFrame
df_latinoamerica.shape

6] ✓ 0.1s
... (11970289, 50)
```

Módulo 4

```
[7] #Observamos la cantidad de datos nulos que tiene cada una de las columnas del DataFrame
df_latinoamerica.isnull().sum()
[7] ✓ 26.3s
... location_key 0
date 0
country_code 0
country_name 0
new_confirmed 5620283
new_deceased 5661788
cumulative_confirmed 5619937
cumulative_deceased 5661426
cumulative_vaccine_doses_administered 11706131
population 521266
population_male 1122803
population_female 1122803
population_rural 11964343
population_urban 11964343
```

7. Después de revisar los datos, finalmente filtramos los países mediante el campo location_key, quedando solo 5946 registros.

```
[8] df_latinoamerica.location_key.value_counts()
[8] ✓ 1.0s
... location_key
AR 991
MX_GRO_12040 991
MX_GRO_12031 991
MX_GRO_12032 991
MX_GRO_12033 991
...
BR_PR_411880 991
BR_PR_411885 991
BR_PR_411890 991
BR_PR_411900 991
PE_UCA_0401 991
Name: count, Length: 12079, dtype: int64

[9] #filtramos el Dataframe por el Location_Key que representa a los países del estudio
codigos_paises=['AR','PE','MX','CL','CO','BR']
df_paises=df_covid[df_covid['location_key'].isin(codigos_paises)]
df_paises.shape
[9] ✓ 1.3s
.. (5946, 50)
```

8. El cliente requería que analizáramos datos desde el año 2021, por lo que se realizó un filtro adicional para tener valores desde el 01/01/2021
9. Finalmente quedamos un dataframe con 3750 registros y 50 columnas.

Módulo 4

```
> #Filtramos los datos de los registros que sean mayores o igual al 01/01/2021
> df_paises_fecha=df_paises[df_paises['date']>='2021-01-01']

> #vemos la cantidad de registros resultantes despues del filtrado
> df_paises_fecha.shape

10] ✓ 0.0s
.. (3750, 50)
```

10. Procedemos a verificar la cantidad de datos nulos en los registros mediante el comando .isnull() del dataframe.

```
#Obtenemos el numero de valores nulos que quedan en cada una de las columnas del DataFrame
df_valores_nulos_fecha=df_paises_fecha.isnull().sum()[df_paises_fecha.isnull().sum(>0)]
print(df_valores_nulos_fecha)
#print(np.shape(df_paises_fecha.shape))

11] ✓ 0.0s
.. new_confirmed                21
new_deceased                   21
cumulative_confirmed           21
cumulative_deceased            21
cumulative_vaccine_doses_administered 589
average_temperature_celsius    42
minimum_temperature_celsius   41
maximum_temperature_celsius   41
rainfall_mm                     90
relative_humidity               42
new_recovered                  2119
cumulative_recovered           2740
dtype: int64
```

11. Mediante ciclos anidados FOR procedemos a rellenar con el valor promedio las siguientes columnas 'new_confirmed', 'new_deceased', 'average_temperature_celsius','minimum_temperature_celsius', 'maximum_temperature_celsius', 'rainfall_mm', 'relative_humidity'

```
#Actualizamos por cada pais los valores nulos de las columnas indicadas con el valor promedio del pais en esa medida
for codigo_pais in codigos_paises:
    for columna in df_valores_nulos_fecha.index:
        if columna in ('new_confirmed', 'new_deceased', 'average_temperature_celsius', 'minimum_temperature_celsius',
                       'maximum_temperature_celsius', 'rainfall_mm', 'relative_humidity'):
            df_paises_fecha.loc[df_paises_fecha['country_code']==codigo_pais,columna]=df_paises_fecha[columna].fillna(df_paises_fecha.loc[df_paises_fecha['country_code']==codigo_pais,columna].mean())

✓ 0.1s

#comprobamos que valores nulos quedan en el DataFrame
df_valores_nulos_fecha=df_paises_fecha.isnull().sum()[df_paises_fecha.isnull().sum(>0)]
print(df_valores_nulos_fecha)

✓ 0.0s
cumulative_confirmed           21
cumulative_deceased            21
cumulative_vaccine_doses_administered 589
new_recovered                  2119
cumulative_recovered           2740
dtype: int64
```

Módulo 4

12. Ahora rellenamos con 0 otros 3 campos que no se tiene de donde obtener el dato.

13. Verificamos que ahora nos quedan solo dos columnas con son datos acumulados que se pueden calcular para llenarlos.

```
#Mediante un ciclo For recorremos cada uno de los paises para actualizar en 0
for codigo_pais in codigos_paises:
    df_paises_fecha.loc[df_paises_fecha['country_code']==codigo_pais,'new_recovered']=df_paises_fecha['new_recovered'].fillna(0)
    df_paises_fecha.loc[df_paises_fecha['country_code']==codigo_pais,'cumulative_recovered']=df_paises_fecha['cumulative_recovered'].fillna(0)
    df_paises_fecha.loc[df_paises_fecha['country_code']==codigo_pais,'cumulative_vaccine_doses_administered']=df_paises_fecha['cumulative_vaccine_doses_administered'].fillna(0)

#comprobamos que valores nulos quedan en el DataFrame
df_valores_nulos_fecha=df_paises_fecha.isnull().sum()[df_paises_fecha.isnull().sum()>0]
print(df_valores_nulos_fecha)
```

	cumulative_confirmed	cumulative_deceased
	21	21
	dtype: int64	

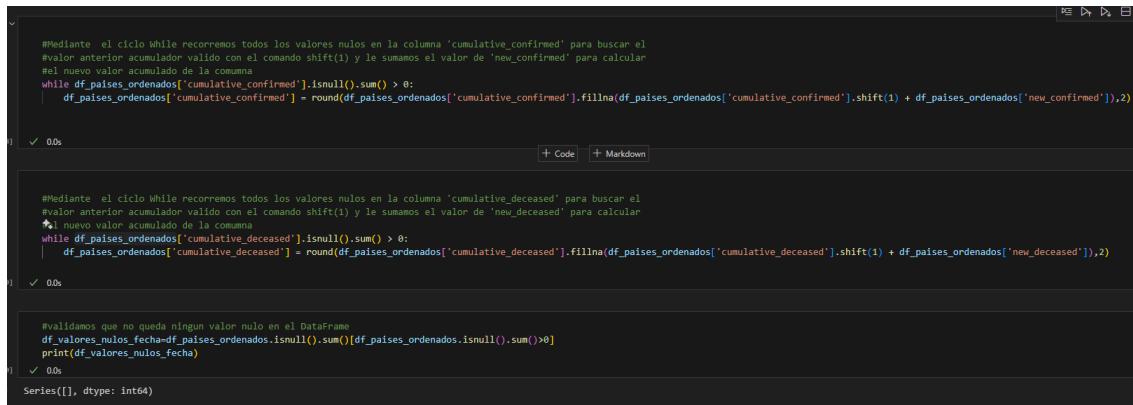
14. Para calcular esos valores primero tenemos que ordenar de manera ascendente por las columnas 'location_key' y 'date'

15. Verificamos que están ordenados los datos.

```
#ordenamos por 'loation_key' y 'date' en orden ascendente para preparar los datos para el siguiente paso
#y poder calcular los datos faltantes en las columnas de 'cumulative_cofirmed' y 'cumulative_deceased'
df_paises_ordenados=df_paises_fecha.sort_values(by=['location_key','date'], ascending=[True,True])
```

	location_key	date	country_code	country_name	new_confirmed	new_deceased	cumulative_confirmed	cumulative_deceased
366	AR	2021-01-01	AR	Argentina	2685.000000	140.000000	1662867.0	1662867.0
367	AR	2021-01-02	AR	Argentina	7767.000000	166.000000	1670634.0	1670634.0
368	AR	2021-01-03	AR	Argentina	4934.000000	157.000000	1675568.0	1675568.0
369	AR	2021-01-04	AR	Argentina	13953.000000	157.000000	1689521.0	1689521.0

16. Procedemos a llenar los datos usando un ciclo While para recorrer los datos e ir actualizando el valor acumulado con el valor acumulado del registro anterior y sumándole el nuevo valor del registro actual hasta que se terminen los valores nulos en cada una de las columnas.

Módulo 4


```

#Mediante el ciclo While recorremos todos los valores nulos en la columna 'cumulative_confirmed' para buscar el
#valor anterior acumulado valido con el comando shift(1) y le sumamos el valor de 'new_confirmed' para calcular
#el nuevo valor acumulado de la columna
while df_paises_ordenados['cumulative_confirmed'].isnull().sum() > 0:
    df_paises_ordenados['cumulative_confirmed'] = round(df_paises_ordenados['cumulative_confirmed'].fillna(df_paises_ordenados['cumulative_confirmed'].shift(1) + df_paises_ordenados['new_confirmed']),2)

#Mediante el ciclo While recorremos todos los valores nulos en la columna 'cumulative_deceased' para buscar el
#valor anterior acumulado valido con el comando shift(1) y le sumamos el valor de 'new_deceased' para calcular
#el nuevo valor acumulado de la columna
while df_paises_ordenados['cumulative_deceased'].isnull().sum() > 0:
    df_paises_ordenados['cumulative_deceased'] = round(df_paises_ordenados['cumulative_deceased'].fillna(df_paises_ordenados['cumulative_deceased'].shift(1) + df_paises_ordenados['new_deceased']),2)

#Validamos que no queda ningun valor nulo en el DataFrame
df_valores_nulos_fecha=df_paises_ordenados.isnull().sum()
print(df_valores_nulos_fecha)

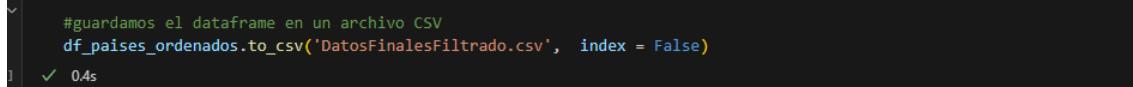
0.0s
Series([], dtype: int64)

```

17. Finalmente guardamos el dataframe en el archivo

‘DatosFinalesFiltrado.csv’ para poder ser usado más adelante.

Guardamos los datos filtrados en un archivo con el nombre DatosFinalesFiltrado.csv filtrado y limpieza.



```

#guardamos el dataframe en un archivo CSV
df_paises_ordenados.to_csv('DatosFinalesFiltrado.csv', index = False)

0.4s

```

Después de esto también se realizaron otros pasos adicionales para corregir datos puntuales de la columna ‘new_deceased’ que afectaban al país de Chile. Eso está detallado en el archivo de Python adjunto a este archivo.

Adicionalmente se obtuvieron valores de estadística descriptiva para ver la calidad de los datos, encontrando que en la mayoría de variables existe una alta variabilidad de datos, dando una media significativamente mayor que la mediana y un amplio rango de datos que sugieren una gran dispersión en los datos, con outliers que afectan significativamente las medidas de tendencia central y dispersión.

Módulo 4

Mediante el comando `describe()` se obtienen valores de estadística descriptiva

```
#Mediante el comando describe() se obtienen valores de estadística descriptiva para cada una de las
#columnas del dataframe df_paises_covid
for i in df_paises_covid.columns:
    print('*****')
    print(i)
    print('-----')
    print(df_paises_covid[i].describe())
    print('-----')

1 ✓ 0.2s
```

```
*****
country_code
-----
count      3750
unique       6
top        AR
freq      625
Name: country_code, dtype: object
*****
country_name
-----
count      3750
unique       6
```

Continuando con el análisis de los datos también observamos valores negativos en un dato que no puede ser negativo, por lo que se procedió a poner en 0 o se podía poner el valor absoluto del dato.

Observamos que hay valores negativos en unas columnas, procedemos a actualizar esos datos

```
#1ro ordenamos en forma ascendente por el campo 'new_confirmed' para ver si hay mas valores negativos
df_paises_covid.sort_values(by=['new_confirmed'], ascending=[True])
26 ✓ 0.0s
```

		country_code	country_name	new_confirmed	new_deceased	cumulative_confirmed	cumulative_deceased
location_key	date						
BR	2021-09-21	BR	Brazil	-573.0	485.0	21247094.0	21247094.0
CO	2022-08-23	CO	Colombia	0.0	0.0	6293130.0	6293130.0
	2022-06-18	CO	Colombia	0.0	0.0	6131657.0	6131657.0
	2022-06-17	CO	Colombia	0.0	0.0	6131657.0	6131657.0

También se creó una función `obtenerMedianaVarianzaRango(data)` para obtener los datos de la Mediana, Varianza y el rango para cada uno de las columnas del dataframe que se le pase como parámetro.

Módulo 4

```

def obtenerMedianaVarianzaRango(data):
    for i in data.columns:
        if data[i].dtype=='float64' or data[i].dtype=='int64':
            print('*****')
            print(i)
            print('-----')
            print('Mediana: ', round(data[i].median(),2))
            print('Varianza: ', round(data[i].var(),2))
            print('Rango: ', round(data[i].max() - data[i].min(),2))
            print('-----')

    #se llama por cada pais

    obtenerMedianaVarianzaRango(df_paises_covid)
    ✓ 0.0s

*****
new_confirmed
-----
Mediana: 5199.5
Varianza: 585823191.79
Rango: 298408.0
-----
*****

```

Se crearon diferentes tipos de gráficos usando las librerías de matplotlib y seaborn, para descubrir patrones e insights en los datos

1. Gráficos de barras de países contra el resto de variables

1. Gráficos de Barras de países contra el resto de variables

```

# Este código genera gráficos de barras para cada columna en df_paises_covid que no está en la lista de columnas excluidas.
# Cada gráfico muestra los valores de la columna para cada país.

# Se define la lista de columnas a excluir
excluded_columns = ['location_key','date','country_code','country_name','latitude',
                     'longitude','cumulative_recovered','new_recovered']

# Se itera sobre cada columna en df_paises_covid
for column in df_paises_covid.columns:
    # Se verifica si la columna actual no está en la lista de columnas excluidas
    if column not in excluded_columns:
        # Se crea una nueva figura y un conjunto de subtramas con un tamaño específico
        fig, ax = plt.subplots(figsize=(15, 8))

        # Se ordena el DataFrame por la columna actual en orden descendente
        sorted_df = df_paises_covid.sort_values(by=column, ascending=False)

        # Se crea un gráfico de barras con 'country_name' en el eje x y la columna actual en el eje y
        sns.barplot(x='country_name', y=column, data=sorted_df, ax=ax, hue='country_name')

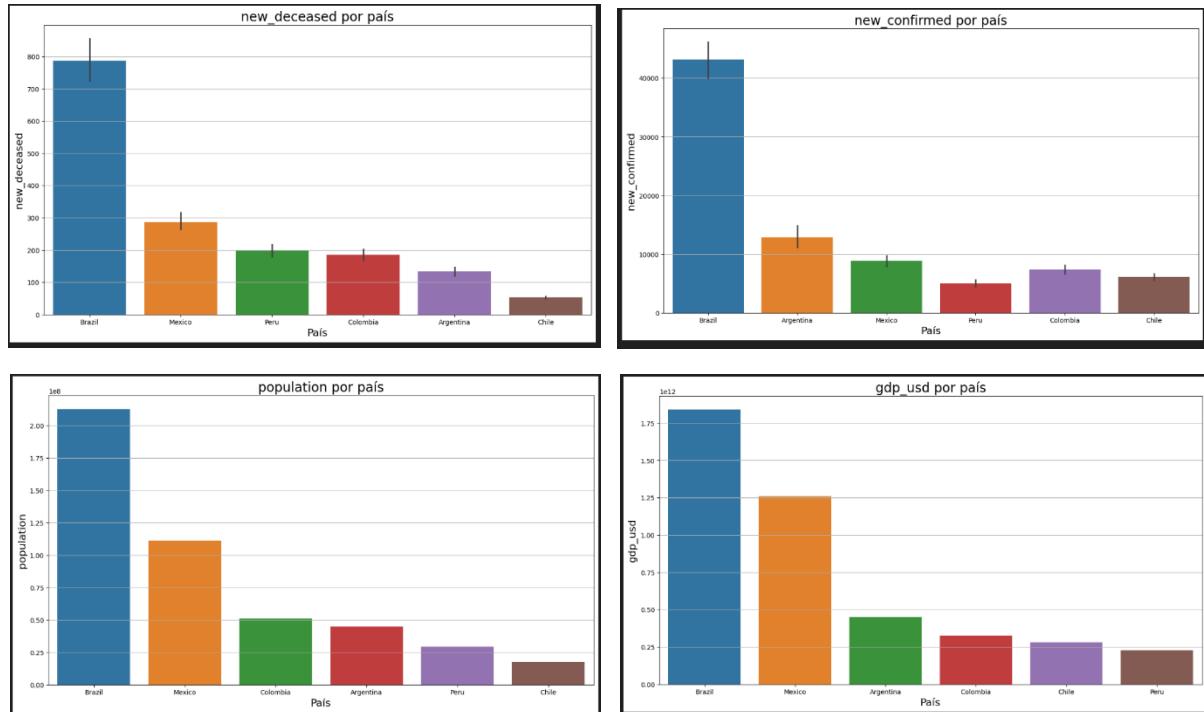
        # Se establece el título del gráfico como el nombre de la columna actual seguido de "por país"
        plt.title(f'{column} por país', fontsize=20)

        # Se ajusta el tamaño de la fuente de las etiquetas de los ejes
        plt.xlabel('País', fontsize=16)
        plt.ylabel(column, fontsize=16)

        # Se añade una cuadrícula en el eje Y para facilitar la lectura de los datos
        ax.yaxis.grid(True)

        # Se muestra el gráfico
        plt.show()

```

Módulo 4

Con este primer gráfico de barras por países nos damos una idea global de las características principales tanto a nivel de población, ingresos, infraestructura médica (enfermeras y doctores), condiciones médicas, clima entre otros.

2. Matriz de Correlación (trata de crear una submáscara para solo tomar una parte del gráfico y filtrar por los valores mayores a 0.5)

2. Matriz de Correlación, trata de crear una submáscara para solo tomar una parte del gráfico y filtrar por los valores mayores a 0.5

```
# Este código genera una gráfica de matriz de correlación para las variables numéricas en df_paises_covid.

# Se seleccionan las columnas numéricas en df_paises_covid
columnas_numericas = df_paises_covid.select_dtypes(include=['float64', 'int64']).columns

# Se calcula la matriz de correlación de las columnas numéricas
matriz_correlacion = df_paises_covid[columnas_numericas].corr()

# Se crea una máscara para el triángulo superior de la matriz de correlación
mask = np.triu(np.ones_like(matriz_correlacion, dtype=bool))

# Se filtran los valores de correlación que son absolutamente mayores que 0.5
matriz_correlacion = matriz_correlacion[abs(matriz_correlacion) > abs(0.5)]

# Se crea una nueva figura con un tamaño específico
plt.figure(figsize=(30, 15))

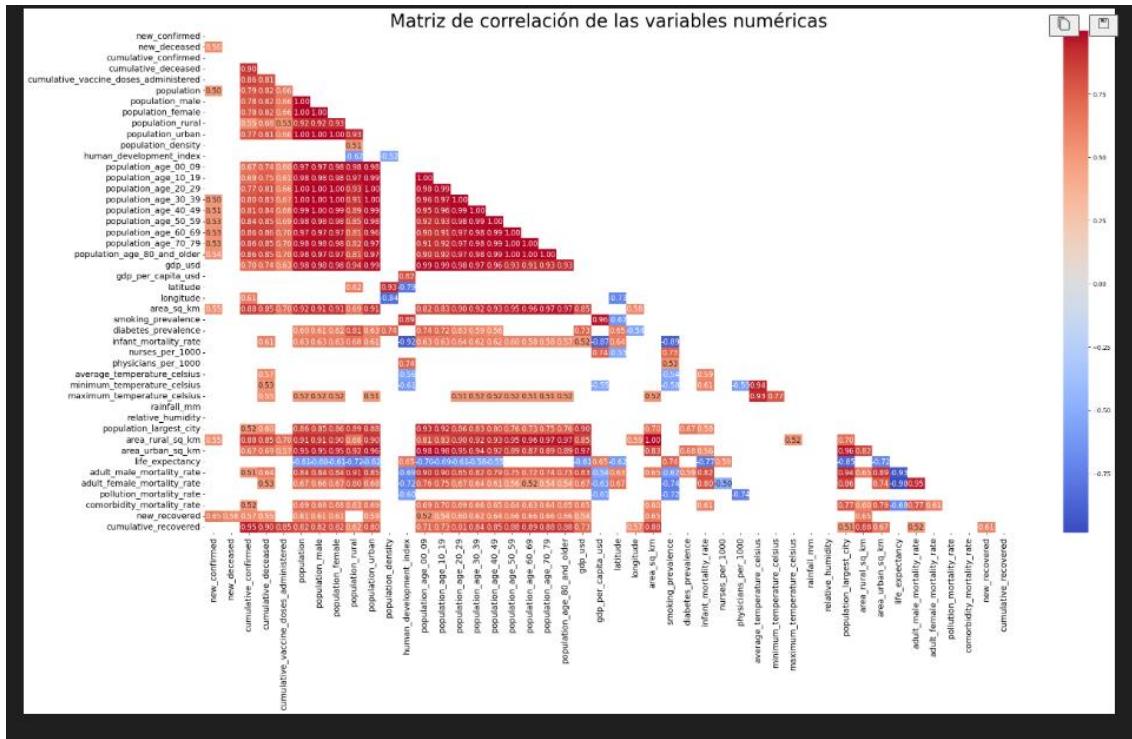
# Se crea una gráfica de mapa de calor usando seaborn
heatmap = sns.heatmap(
    matriz_correlacion,
    mask=mask,
    annot=True,
    cmap='coolwarm',
    fmt=".2f",
    annot_kws={"size": 12} # Se ajusta el tamaño de la fuente de las anotaciones
)

# Se ajusta el tamaño de la fuente de las etiquetas de los ejes
heatmap.set_xticklabels(heatmap.get_xmajorticklabels(), fontsize = 14)
heatmap.set_ymajorticklabels(heatmap.get_ymajorticklabels(), fontsize = 14)

# Se establece el título de la gráfica
plt.title('Matriz de correlación de las variables numéricas', fontsize=30)

# Se muestra la gráfica
plt.show()
```

Python



Relaciones Fuertes:

Las celdas con valores de correlación absoluta mayores a 0.5 indican relaciones significativas entre las variables. Por ejemplo, si new_confirmed y new_deceased tienen una alta correlación positiva, esto sugiere que a medida que aumentan los casos nuevos confirmados, también aumentan las nuevas muertes.

Identificación de Patrones:

El mapa de calor ayuda a identificar patrones importantes, como la relación entre la tasa de vacunación (cumulative_vaccine_doses_administered) y la disminución de casos nuevos o muertes. Si hay una correlación negativa significativa entre la tasa de vacunación y los casos nuevos confirmados, esto indicaría que la vacunación es efectiva para reducir la propagación del virus.

Factores Climáticos:

También se pueden observar correlaciones entre factores climáticos (average_temperature_celsius, minimum_temperature_celsius, maximum_temperature_celsius, relative_humidity) y las variables relacionadas con

Módulo 4

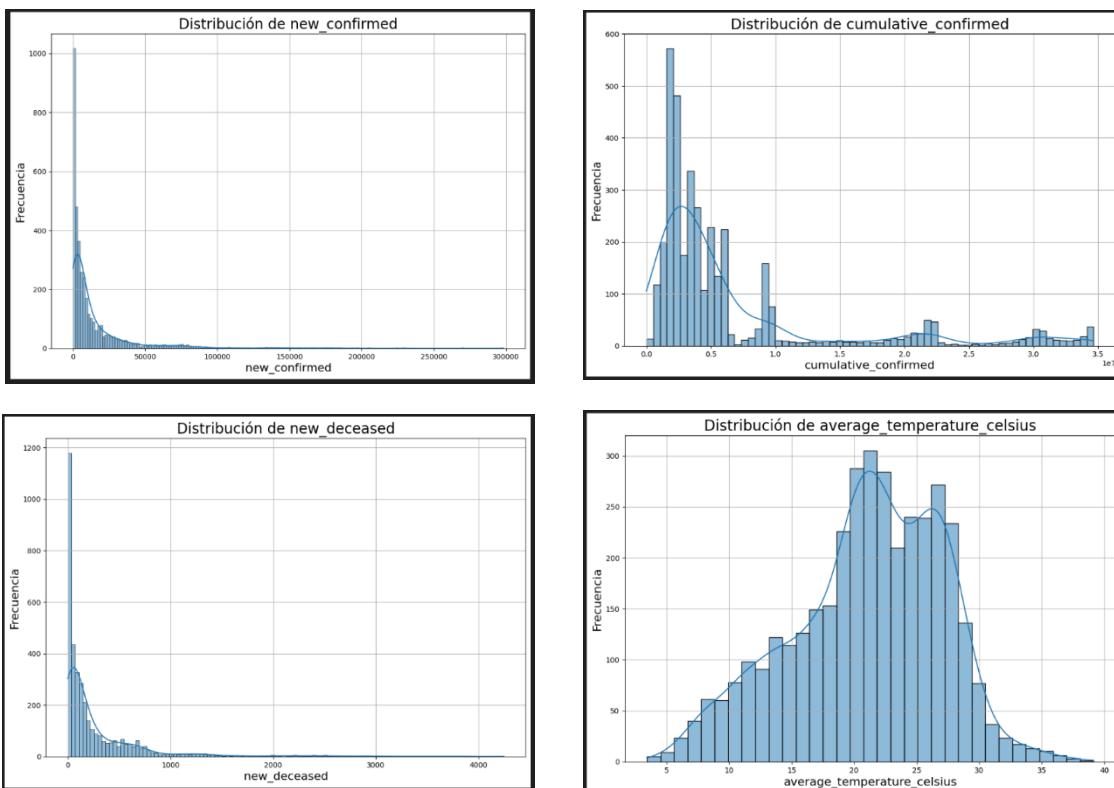
COVID-19. Estas correlaciones pueden ofrecer insights sobre cómo las condiciones climáticas afectan la incidencia del virus y la efectividad de la vacunación.

Estas correlaciones pueden proporcionar insights valiosos sobre cómo la infraestructura médica, las tasas de vacunación y los factores climáticos interactúan y afectan la incidencia del virus. Esta información es crucial para tomar decisiones informadas sobre la ubicación de nuevos laboratorios y centros de vacunación en América Latina.

3. Histogramas de las variables que "Si" tenga cambios de valores

```
# Se seleccionan las columnas para el estudio
lista_columnas=['new_confirmed','new_deceased','cumulative_confirmed','cumulative_deceased',
                'cumulative_vaccine_doses_administrated','average_temperature_celsius','minimum_temperature_celsius',
                'maximum_temperature_celsius','relative_humidity']

# Se itera sobre cada columna en df_paises_covid
for i in df_paises_covid:
    # Se verifica si la columna actual está en la lista de columnas seleccionadas
    if i in lista_columnas:
        # Se crea una nueva figura y un conjunto de subplots con un tamaño más grande
        fig, ax = plt.subplots(figsize=(12,8))
        # Se crea un histograma con la columna actual
        sns.histplot(df_paises_covid[i], kde=True)
        # Se añade una cuadrícula
        plt.grid(True)
        # Se establece el título del gráfico como el nombre de la columna actual con un tamaño de fuente mayor
        plt.title('Distribución de ' + i, fontsize=20)
        # Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
        plt.xlabel(i, fontsize=16)
        plt.ylabel('Frecuencia', fontsize=16)
        # Se muestra el gráfico
        plt.show()
```



Módulo 4

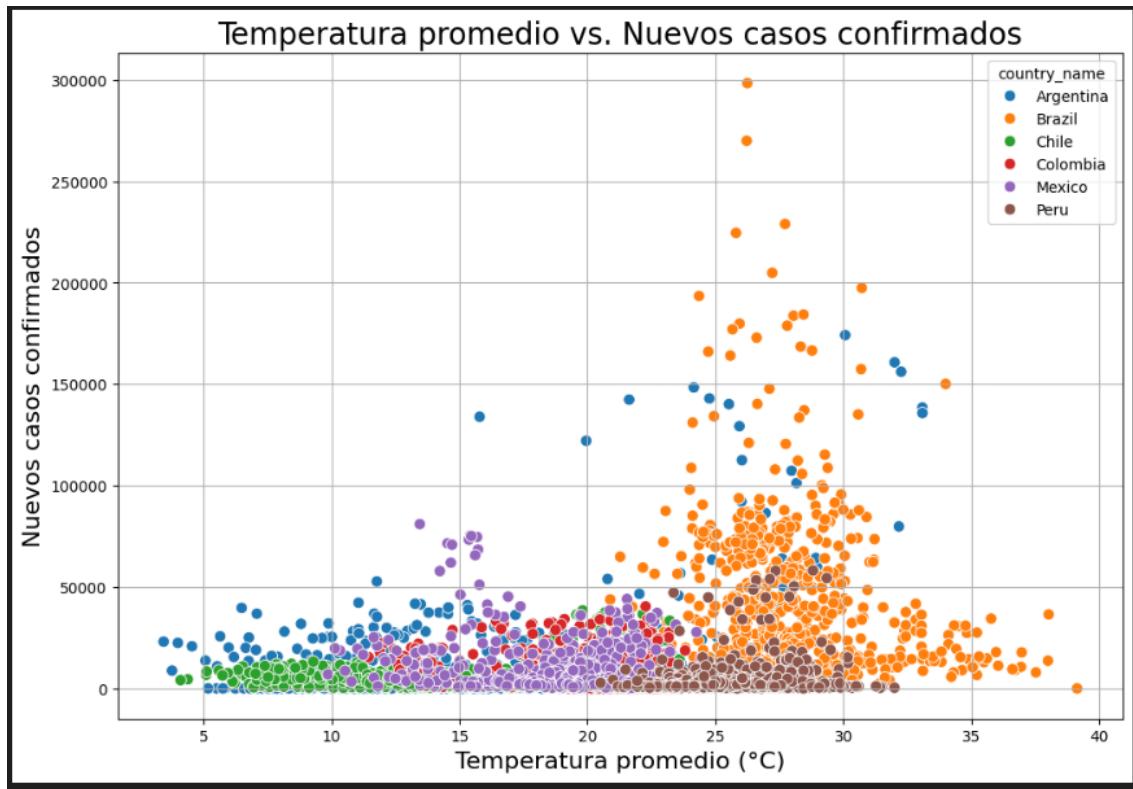
Los histogramas generados para cada columna proporcionan una visión detallada de la distribución de los datos clave relacionados con la pandemia de COVID-19 y factores climáticos. Estos gráficos son esenciales para:

- **Identificar Patrones y Tendencias:** Comprender cómo se distribuyen los casos de COVID-19, las muertes, y la vacunación en diferentes países y períodos.
- **Planificación de Recursos:** Ayudar a planificar la ubicación de nuevos laboratorios basándose en la incidencia de la enfermedad y las necesidades de vacunación.
- **Factores Climáticos:** Tener en cuenta las condiciones climáticas que pueden influir en la efectividad de las vacunas y la logística de distribución.

4. Diagrama de dispersión de la temperatura media contra los casos confirmados

```
# Este código genera un gráfico de dispersión de 'average_temperature_celsius' vs 'new_confirmed' en el DataFrame df_paises_covid.  
# Se crea una nueva figura y un conjunto de subtramas con un tamaño más grande  
fig, ax = plt.subplots(figsize=(12,8))  
  
# Se crea un gráfico de dispersión con 'average_temperature_celsius' en el eje x y 'new_confirmed' en el eje y  
# Se utiliza 'country_name' como leyenda (hue) y se ajusta el tamaño de los puntos (s)  
sns.scatterplot(x='average_temperature_celsius', y='new_confirmed', data=df_paises_covid, hue='country_name', s=60)  
  
# Se añade una cuadrícula  
plt.grid(True)  
  
# Se establece el título del gráfico como 'Temperatura promedio vs. Nuevos casos confirmados' con un tamaño de fuente mayor  
plt.title('Temperatura promedio vs. Nuevos casos confirmados', fontsize=20)  
  
# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor  
plt.xlabel('Temperatura promedio (°C)', fontsize=16)  
plt.ylabel('Nuevos casos confirmados', fontsize=16)  
  
# Se muestra el gráfico  
plt.show()
```

Módulo 4



El gráfico de dispersión proporciona una visión clara de cómo la temperatura promedio puede influir en la incidencia de casos nuevos confirmados de COVID-19 en diferentes países. Estos insights son fundamentales para tomar decisiones informadas sobre la ubicación de nuevos laboratorios y la implementación de estrategias de vacunación, contribuyendo a una respuesta pandémica más eficiente en América Latina.

5. Diagrama de dispersión de la temperatura media contra las muertes confirmadas

```
# Este código genera un gráfico de dispersión de 'average_temperature_celsius' vs 'new_deceased' en el DataFrame df_paises_covid.

# Se crea una nueva figura y un conjunto de subtramas con un tamaño más grande
fig, ax = plt.subplots(figsize=(12,8))

# Se crea un gráfico de dispersión con 'average_temperature_celsius' en el eje x y 'new_deceased' en el eje y
# Se utiliza 'country_name' como leyenda (hue) y se ajusta el tamaño de los puntos (s)
sns.scatterplot(y='average_temperature_celsius', x='new_deceased', data=df_paises_covid, hue='country_name', s=60)

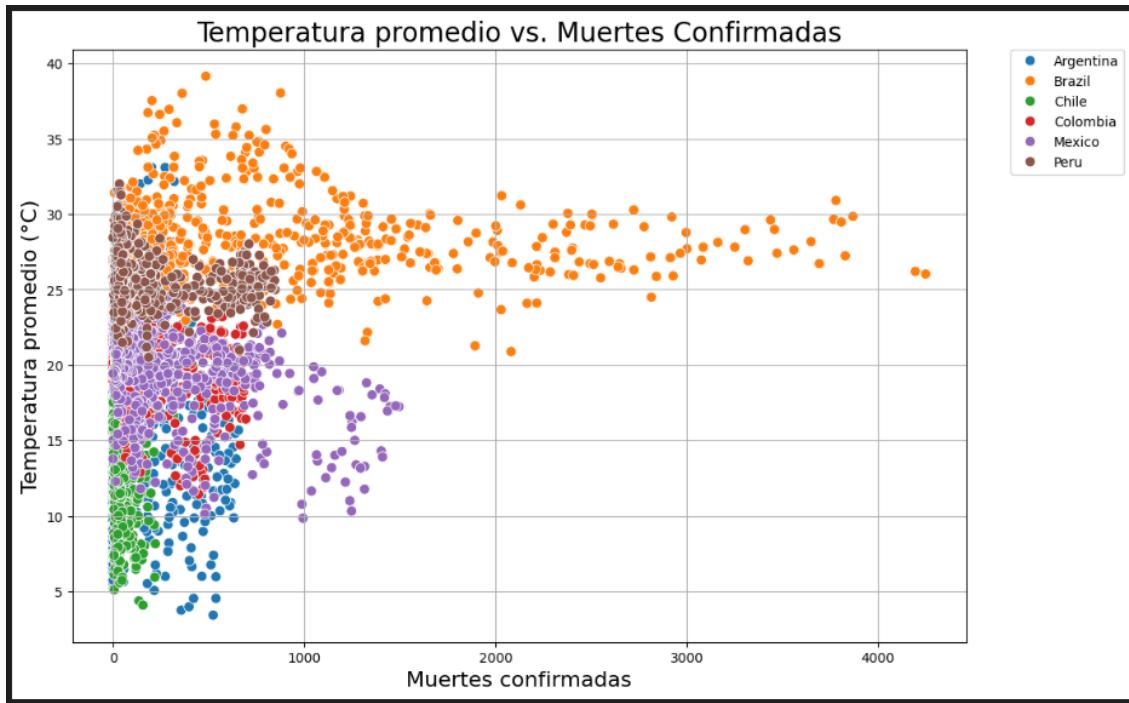
# Se añade una cuadrícula
plt.grid(True)

# Se establece el título del gráfico como 'Temperatura promedio vs. Nuevos fallecimientos' con un tamaño de fuente mayor
plt.title('Temperatura promedio vs. Muertes Confirmadas', fontsize=20)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Muertes confirmadas', fontsize=16)
plt.ylabel('Temperatura promedio (°C)', fontsize=16)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

# Se muestra el gráfico
plt.show()
```



Este gráfico de dispersión muestra la relación entre la temperatura promedio (average_temperature_celsius) y el número de nuevas muertes confirmadas (new_deceased) por COVID-19 en diferentes países.

El gráfico puede revelar si existe una relación inversa entre la temperatura promedio y las nuevas muertes confirmadas. Por ejemplo, una mayor temperatura promedio podría correlacionarse con un menor número de nuevas muertes, o viceversa.

entre la temperatura y las muertes confirmadas entre los diferentes países. Esto ayuda a identificar si ciertos países presentan patrones únicos debido a sus condiciones climáticas específicas o a sus respuestas a la pandemia.

Módulo 4**6. Comportamiento de las dosis administradas de todos los países (valor medio)**

```
# Este código genera un gráfico de líneas que muestra los valores medios mensuales de 'cumulative_vaccine_doses_administered' para cada país.

# Se crea una nueva figura y un conjunto de subtramas con un tamaño más grande
fig, ax = plt.subplots(figsize=(12,8))

# Se agrupan los datos por 'country_name'
grouped = df_paises_covid.groupby('country_name')

# Se itera sobre cada grupo
for name, group in grouped:
    # Se calcula el valor medio mensual de 'cumulative_vaccine_doses_administered' y se genera un gráfico de línea
    # Se cambia el color y el estilo de la línea para que sea más visible
    group.resample('ME').mean(numeric_only=True)['cumulative_vaccine_doses_administered'].plot(label=name)

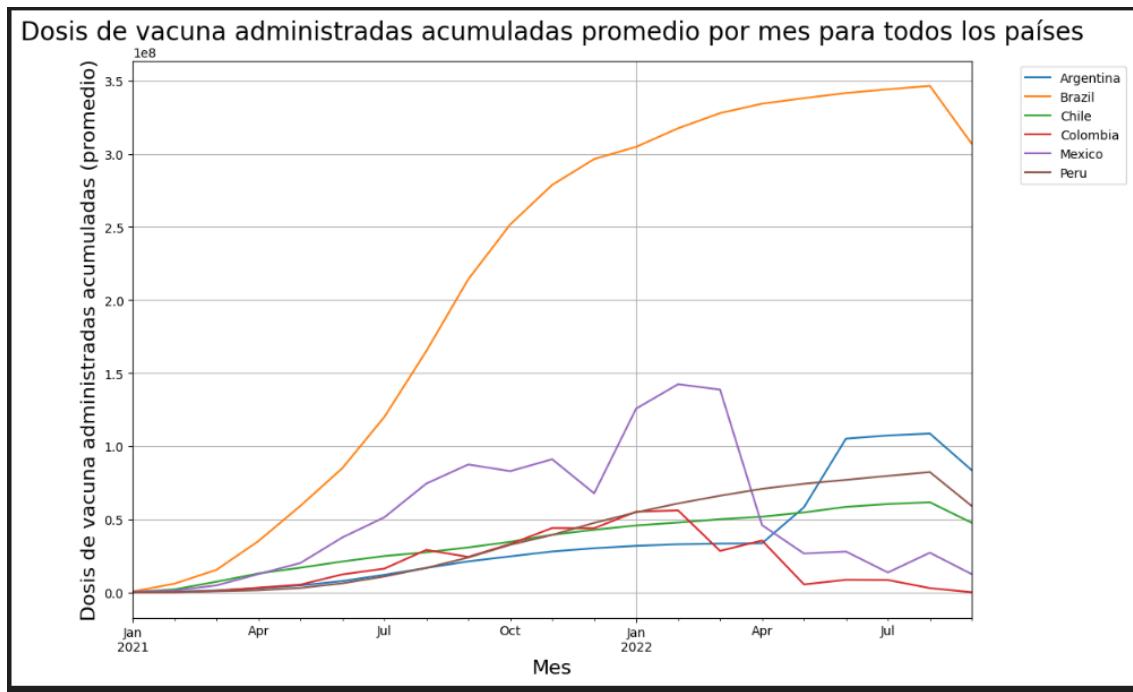
# Se añade una cuadricula
plt.grid(True)

# Se establece el título del gráfico como 'Dosis de vacuna administradas acumuladas promedio por mes para todos los países'
plt.title('Dosis de vacuna administradas acumuladas promedio por mes para todos los países', fontsize=20)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Mes', fontsize=16)
plt.ylabel('Dosis de vacuna administradas acumuladas (promedio)', fontsize=16)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()
```

**Tendencia de Vacunación:**

- El gráfico permite observar la tendencia general de vacunación en cada país. Un aumento constante en la línea indica un esfuerzo continuo y efectivo en la administración de vacunas.

Módulo 4**Comparación Entre Países:**

- Comparar las líneas de diferentes países ayuda a identificar cuáles están logrando administrar vacunas de manera más rápida y efectiva. Las diferencias en las pendientes de las líneas pueden indicar la velocidad y consistencia del programa de vacunación en cada país.

Períodos Críticos:

- Se pueden identificar períodos en los que la administración de vacunas aumentó significativamente, posiblemente debido a campañas intensivas de vacunación o la disponibilidad de nuevas dosis de vacunas.

7. Evolución de dosis administradas por mes de cada país

```
# Este código genera un gráfico de línea que muestra los valores medios mensuales de 'cumulative_vaccine_doses_administered' para todos los países.

# Se crea una nueva figura y un conjunto de subtramas con un tamaño más grande
fig, ax = plt.subplots(figsize=(12,8))

# Se calcula el valor medio mensual de 'cumulative_vaccine_doses_administered' y se genera un gráfico de línea
# Se cambia el color y el estilo de la línea para que sea más visible
df_paises_covid.resample('M').mean(numeric_only=True)[['cumulative_vaccine_doses_administered']].plot(color='blue', linestyle='--')

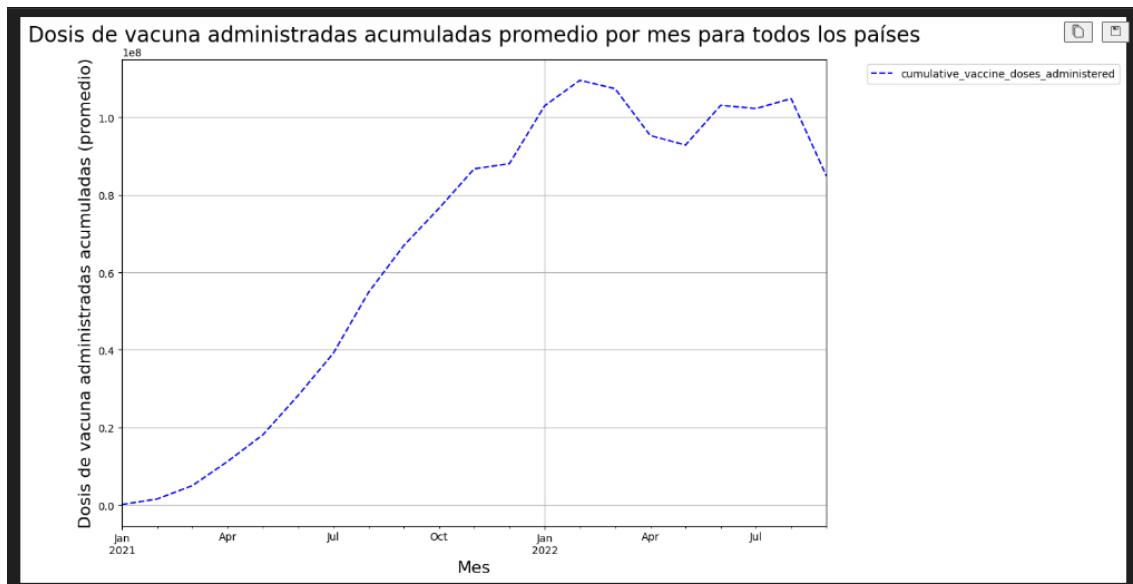
# Se añade una cuadricula
plt.grid(True)

# Se establece el título del gráfico como 'Dosis de vacuna administradas acumuladas promedio por mes para todos los países' con un tamaño de fuente mayor
plt.title('Dosis de vacuna administradas acumuladas promedio por mes para todos los países', fontsize=20)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Mes', fontsize=16)
plt.ylabel('Dosis de vacuna administradas acumuladas (promedio)', fontsize=16)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()
```



Módulo 4

El mismo gráfico se ejecutó pero para que se vean todos los países mediante un ciclo FOR

```
# Este código genera un gráfico de línea que muestra los valores promedios mensuales de 'cumulative_vaccine_doses_administered' para cada país seleccionado.

# Se crea una nueva figura y un conjunto de subtramas con un tamaño más grande
fig, ax = plt.subplots(figsize=(15,10))

# Se itera sobre cada país seleccionado:
for pais in paises_seleccionados:
    # Se seleccionan los datos para el país actual
    data_pais = df_paises_covid[df_paises_covid['country_name']==pais]
    # Se calcula el valor promedio de 'cumulative_vaccine_doses_administered' de cada mes y se genera un gráfico de líneas
    data_pais.resample('ME').mean(numeric_only=True)[['cumulative_vaccine_doses_administered']].plot(label=pais)

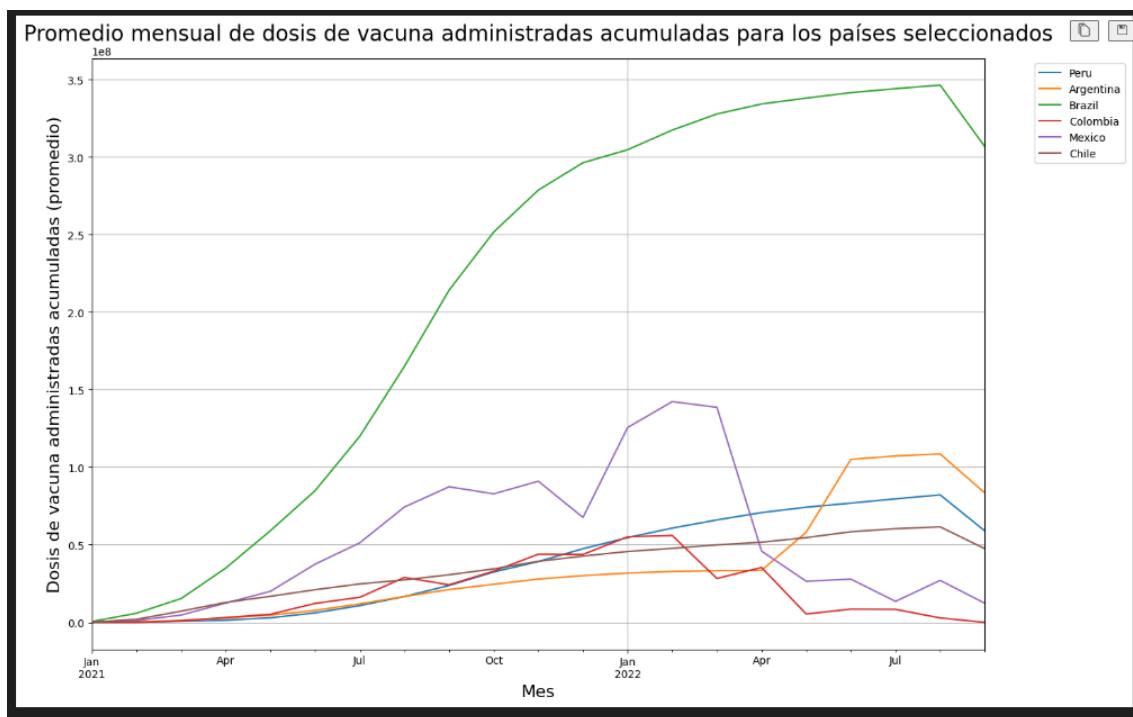
# Se añade una cuadricula
plt.grid(True)

# Se establece el título del gráfico como 'Promedio mensual de dosis de vacuna administradas acumuladas para los países seleccionados'
plt.title('Promedio mensual de dosis de vacuna administradas acumuladas para los países seleccionados', fontsize=20)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Mes', fontsize=16)
plt.ylabel('Dosis de vacuna administradas (promedio)', fontsize=16)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()
```

**Planificación de Laboratorios:**

- Los países con una menor acumulación de dosis administradas pueden beneficiarse más de la apertura de nuevos laboratorios para mejorar la producción y distribución de vacunas.
- Identificar los picos y caídas en la administración de vacunas puede ayudar a planificar la logística y asegurar un suministro constante de dosis.

Módulo 4**Estrategias de Vacunación:**

- Los insights obtenidos del gráfico pueden ayudar a diseñar estrategias de vacunación específicas para cada país seleccionado, basándose en su historial de administración de vacunas.
- Los países con líneas planas o en declive pueden necesitar apoyo adicional para aumentar sus tasas de vacunación.

8. Muertes por mes de cada país

```
# Este código genera un gráfico de líneas que muestra los valores promedios mensuales de 'cumulative_deceased' para cada país seleccionado.

# Se crea una nueva figura y un conjunto de subplots con un tamaño más grande
fig, ax = plt.subplots(figsize=(15,10))

# Se itera sobre cada país seleccionado
for pais in paises_seleccionados:
    # Se seleccionan los datos para el país actual
    data_pais = df_paises_covid[df_paises_covid['country_name']==pais]
    # Se calcula el valor promedio de 'cumulative_deceased' de cada mes y se genera un gráfico de líneas
    data_pais.resample('ME').mean(numeric_only=True)['cumulative_deceased'].plot(label=pais)

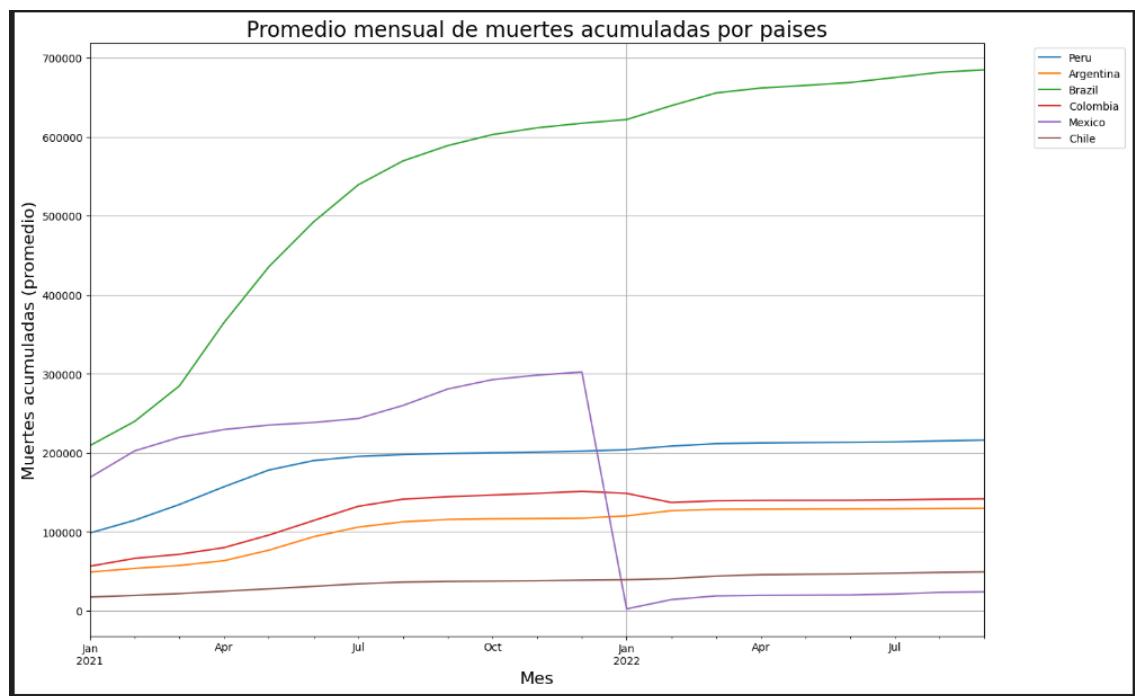
# Se añade una cuadrícula
plt.grid(True)

# Se establece el título del gráfico como 'Promedio mensual de muertes acumuladas para los países seleccionados' con un tamaño de fuente mayor
plt.title('Promedio mensual de muertes acumuladas por países', fontsize=20)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Mes', fontsize=16)
plt.ylabel('Muertes acumuladas (promedio)', fontsize=16)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()
```



Módulo 4

Aquí se observa una clara perdida de datos en las muertes acumuladas del país México

Planificación de Laboratorios:

- Los países con un aumento significativo en la mortalidad acumulada pueden beneficiarse más de la apertura de nuevos laboratorios para mejorar la producción y distribución de vacunas y tratamientos.
- Identificar los picos en la mortalidad puede ayudar a planificar recursos y estrategias de mitigación más efectivas.

Estrategias de Salud Pública:

- Los insights obtenidos del gráfico pueden ayudar a diseñar estrategias de salud pública específicas para cada país seleccionado, basándose en su historial de mortalidad.
- Los países con líneas que muestran una estabilización o disminución en la mortalidad acumulada pueden servir como ejemplos de buenas prácticas en la gestión de la pandemia.

9. Casos confirmados por mes de cada país

```
# Este código genera un gráfico de línea que muestra los valores promedios mensuales de 'cumulative_confirmed' para cada país.

# Se crea una nueva figura y un conjunto de subtramas
fig, ax = plt.subplots(figsize=(15,10))

# Se itera sobre cada país seleccionado
for pais in paises_seleccionados:
    # Se seleccionan los datos para el país actual
    data_pais = df_paises_covid[df_paises_covid['country_name']==pais]
    # Se calcula el valor promedio de 'cumulative_confirmed' de cada mes y se genera un gráfico de línea
    data_pais.resample('ME').mean(numeric_only=True)[['cumulative_confirmed']].plot(label=pais)

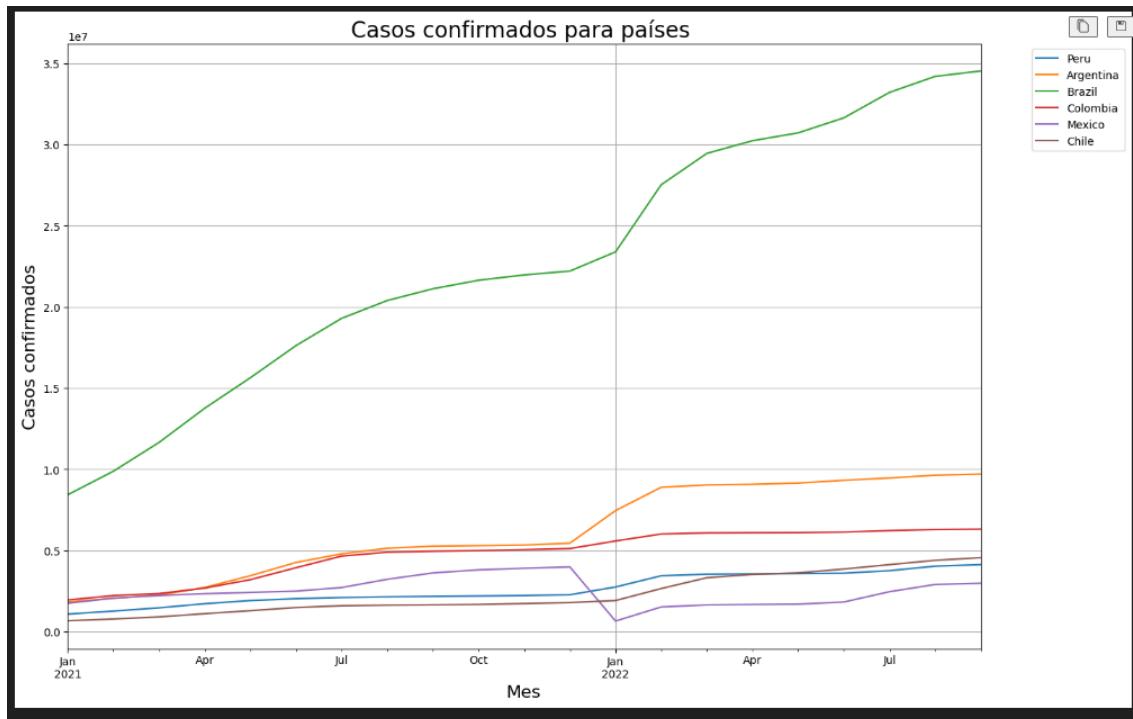
# Se añade una cuadrícula
plt.grid(True)

# Se establece el título del gráfico
plt.title('Casos confirmados para países ', fontsize=20)

# Se añaden etiquetas a los ejes x e y
plt.xlabel('Mes', fontsize=16)
plt.ylabel('Casos confirmados', fontsize=16)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()
```

Módulo 4**Planificación de Laboratorios:**

- Los países con un aumento significativo en los casos confirmados acumulados pueden beneficiarse más de la apertura de nuevos laboratorios para mejorar la producción y distribución de vacunas y tratamientos.
- Identificar los picos en los casos confirmados puede ayudar a planificar recursos y estrategias de mitigación más efectivas.

Estrategias de Salud Pública:

- Los insights obtenidos del gráfico pueden ayudar a diseñar estrategias de salud pública específicas para cada país seleccionado, basándose en su historial de casos confirmados.
- Los países con líneas que muestran una estabilización o disminución en los casos confirmados acumulados pueden servir como ejemplos de buenas prácticas en la gestión de la pandemia.

10. Recuperaciones de casos por mes de cada país

```
# Este código genera un gráfico de línea que muestra los valores promedios mensuales de 'cumulative_recovered' para cada país

# Se crea una nueva figura y un conjunto de subtramas
fig, ax = plt.subplots(figsize=(15,10))

# Se iterá sobre cada país seleccionado
for pais in paises_seleccionados:
    # Se seleccionan los datos para el país actual
    data_pais = df_paises_covid[df_paises_covid['country_name']==pais]
    # Se calcula el valor promedio de 'cumulative_recovered' de cada mes y se genera un gráfico de línea
    data_pais.resample('ME').mean(numeric_only=True)[['cumulative_recovered']].plot(label=pais)

# Se añade una cuadrícula
plt.grid(True)

# Se establece el título del gráfico
plt.title('Recuperaciones de casos por mes de cada país', fontsize=20)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Mes', fontsize=16)
plt.ylabel('Casos recuperados ', fontsize=16)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()
```



Aquí se ve que no existen datos para los países de Perú, México, Chile y Argentina. Colombia tiene una data parcial y solo Brasil tiene los datos completos de los casos recuperados en todo el periodo analizado.

11. Comparación de número de casos nuevos entre países

```
# Este código genera un gráfico de líneas que muestra la media mensual de los casos nuevos confirmados de COVID-19 para cada país seleccionado.

# Se crea una nueva figura y un conjunto de subtramas con un tamaño específico
fig, ax = plt.subplots(figsize=(15, 10))

# Se itera sobre cada país en la lista de países seleccionados
for pais in paises_seleccionados:
    # Se seleccionan los datos para el país actual, se resamplian a una frecuencia mensual y se calcula la media de los casos nuevos confirmados
    # Luego, se genera un gráfico de líneas para estos datos
    df_paises_covid[df_paises_covid['country_name']==pais].resample('ME').mean(numeric_only=True)['new_confirmed'].plot(ax=ax)

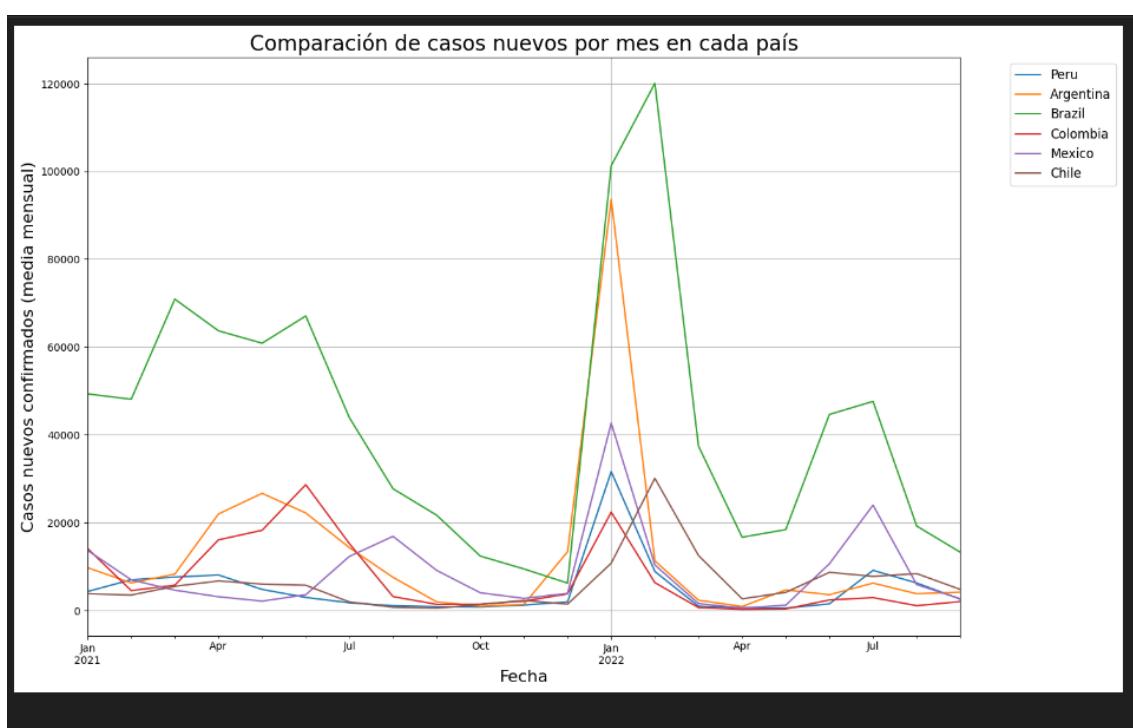
# Se establece el título del gráfico
plt.title('Comparación de casos nuevos por mes en cada país', fontsize=20)

# Se añaden etiquetas a los ejes x e y
plt.xlabel('Fecha', fontsize=16)
plt.ylabel('Casos nuevos confirmados (media mensual)', fontsize=16)

# Se añade una cuadrícula
plt.grid(True)

# Se ajusta la posición de la leyenda y se añaden los nombres de los países
plt.legend(paises_seleccionados, bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)

# Se muestra el gráfico
plt.show()
```



- Los países con un aumento significativo en los casos nuevos confirmados pueden beneficiarse más de la apertura de nuevos laboratorios para mejorar la producción y distribución de vacunas y tratamientos.
- Identificar los picos en los casos nuevos confirmados puede ayudar a planificar recursos y estrategias de mitigación más efectivas.

12. Dosis acumuladas por país

```
# Este código genera un gráfico de densidad que muestra la distribución de las dosis acumuladas de la vacuna COVID-19 para cada país.

# Se crea una nueva figura y un conjunto de subtramas con un tamaño específico
plt.figure(figsize=(20, 15))

# Se genera un gráfico de densidad con 'cumulative_vaccine_doses_administered' en el eje x y 'country_name' como la variable de agrupación
sns.displot(df_paises_covid, x="cumulative_vaccine_doses_administered", hue="country_name", kind="kde", fill=True)

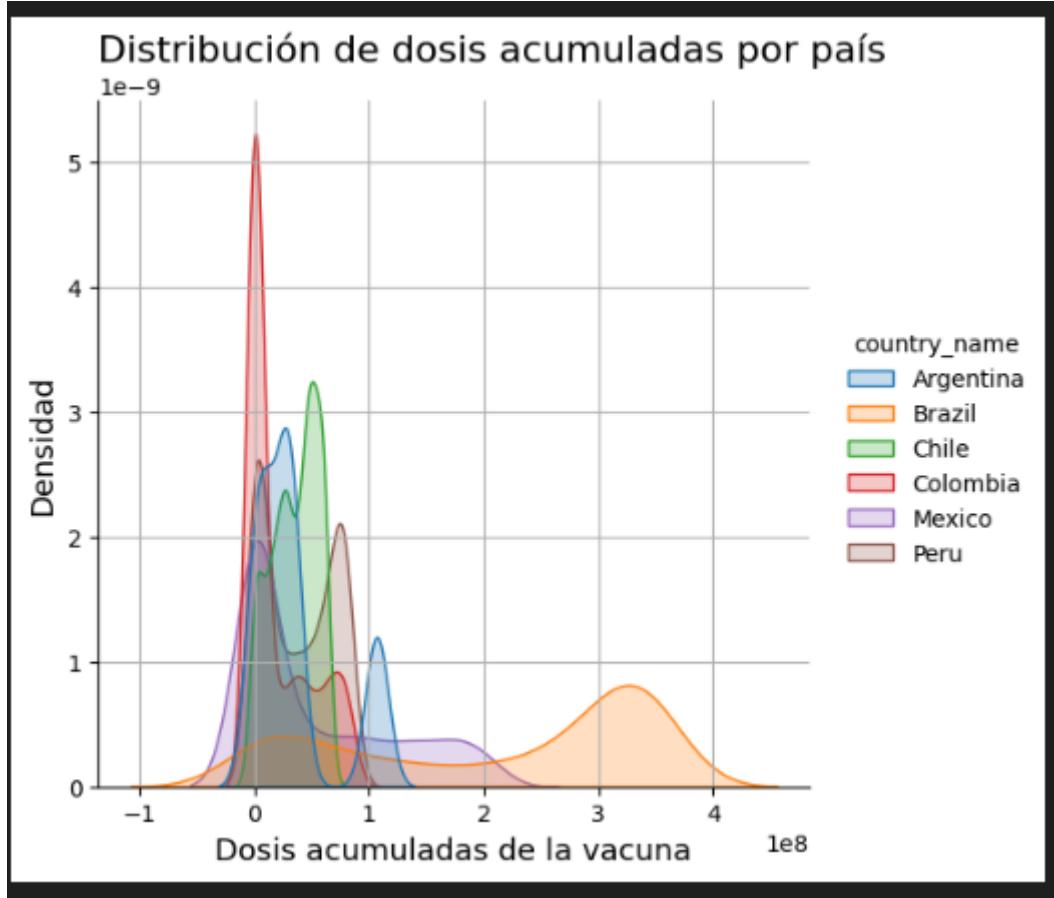
# Se establece el título del gráfico
plt.title('Distribución de dosis acumuladas por país', fontsize=16, loc='left')

# Se añaden etiquetas a los ejes x e y
plt.xlabel('Dosis acumuladas de la vacuna', fontsize=13)
plt.ylabel('Densidad', fontsize=13)

# Se añade una cuadrícula
plt.grid(True)

# Se ajusta la posición de la leyenda y se añaden los nombres de los países
# plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)

# Se muestra el gráfico
plt.show()
```



Planificación de Laboratorios:

- Los países con menor densidad de dosis acumuladas pueden beneficiarse más de la apertura de nuevos laboratorios para aumentar la producción y disponibilidad de vacunas.
- Identificar la distribución de dosis puede ayudar a planificar estrategias de distribución más efectivas y equitativas.

Estrategias de Vacunación:

- Los insights obtenidos del gráfico pueden ayudar a diseñar estrategias de vacunación específicas para cada país, enfocándose en aquellos con menor densidad de dosis administradas para mejorar la cobertura de vacunación.
- Los países con curvas de densidad más bajas necesitan intensificar sus esfuerzos de vacunación para alcanzar una mayor inmunidad poblacional.

13. Bloxplot de temperatura media de cada país

```
# Este código genera un boxplot que muestra la distribución de la temperatura media en Celsius para cada país.

# Se crea una nueva figura y un conjunto de subtramas con un tamaño específico
fig, ax = plt.subplots(figsize=(15, 10))

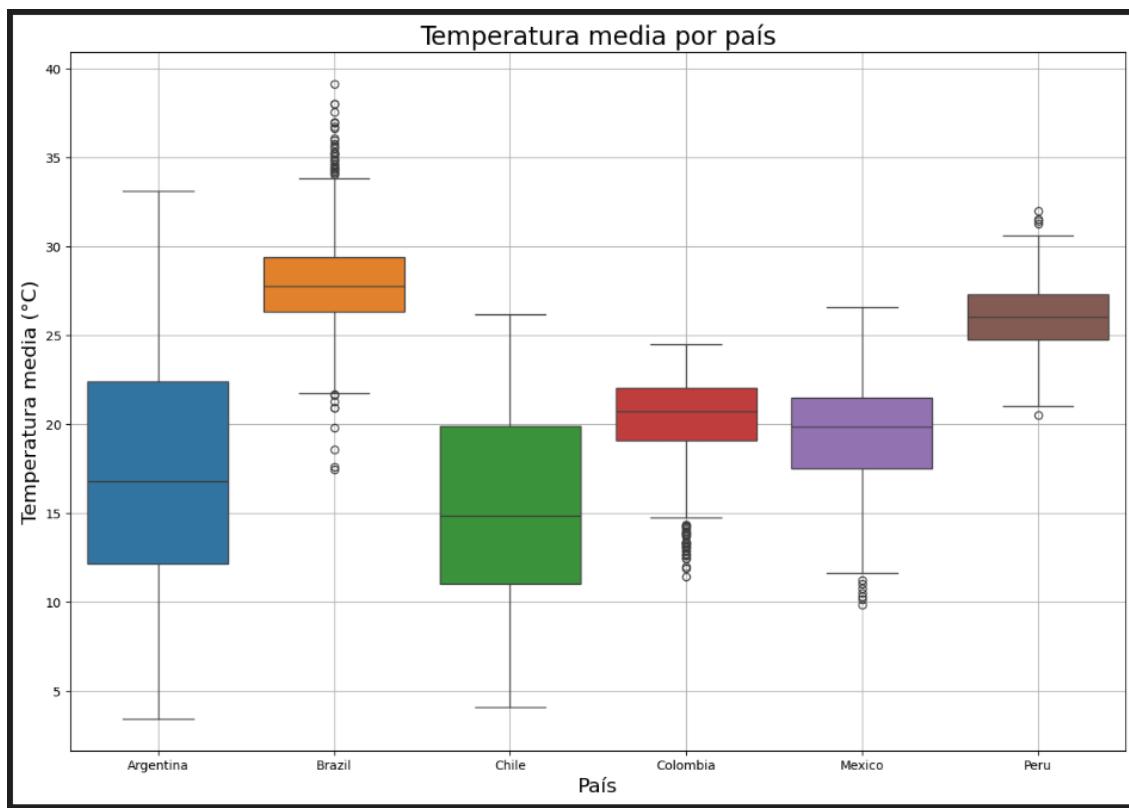
# Se genera un boxplot con 'country_name' en el eje x y 'average_temperature_celsius' en el eje y
sns.boxplot(x='country_name', y='average_temperature_celsius', data=df_paises_covid, hue='country_name')

# Se establece el título del gráfico
plt.title('Temperatura media por país', fontsize=20)

# Se añaden etiquetas a los ejes x e y
plt.xlabel('País', fontsize=16)
plt.ylabel('Temperatura media (°C)', fontsize=16)

# Se añade una cuadrícula
plt.grid(True)

# Se muestra el gráfico
plt.show()
```

**Hallazgos Clave:****1. Comparación de Temperaturas:**

- El boxplot permite comparar la distribución de la temperatura media entre diferentes países. Podemos ver las diferencias en las temperaturas medias y cómo se distribuyen en cada país.

2. Mediana y Cuartiles:

- La línea dentro de cada caja representa la mediana de la temperatura media. Los bordes de la caja representan el primer y tercer cuartil (Q1 y Q3), mostrando la dispersión central de los datos.
- Países con una mediana más alta tienen una temperatura media mayor en comparación con otros países.

3. Rango Intercuartílico (IQR) y Valores Atípicos:

- El rango intercuartílico (IQR), que es la longitud de la caja, muestra la variabilidad de la temperatura media. Una caja más grande indica una mayor variabilidad.

Módulo 4

- Los puntos fuera de las "bigotes" (líneas que se extienden desde las cajas) son valores atípicos y representan temperaturas que son significativamente diferentes del resto de los datos.

14. Violinplot de las variables que "Si" tenga cambios de valores

```
# Configuramos el estilo de Seaborn para mejorar la estética del gráfico
sns.set_style("whitegrid")

# Creamos una lista para almacenar las columnas con más de 6 valores únicos
columnas_mayores = []

# Iteramos sobre todas las columnas del DataFrame
for i in df_paises_covid.columns:
    # Excluimos las columnas que no queremos considerar
    if i not in ['location_key','date','country_code','latitude','longitude']:
        # Si la columna tiene más de 6 valores únicos, la añadimos a la lista
        if df_paises_covid[i].nunique() > 6:
            columnas_mayores.append(i)

# Añadimos la columna 'country_name' a la lista
columnas_mayores.append('country_name')

# Creamos un nuevo DataFrame con solo las columnas de la lista
df_paises_covid_mayores = df_paises_covid[columnas_mayores]

# Iteramos sobre todas las columnas del nuevo DataFrame
for i in df_paises_covid_mayores.columns:
    # Excluimos las columnas que no queremos considerar
    if i not in ['location_key','date','country_code','latitude','longitude','country_name']:
        # Creamos una nueva figura con un tamaño más grande
        plt.figure(figsize=(20, 10))

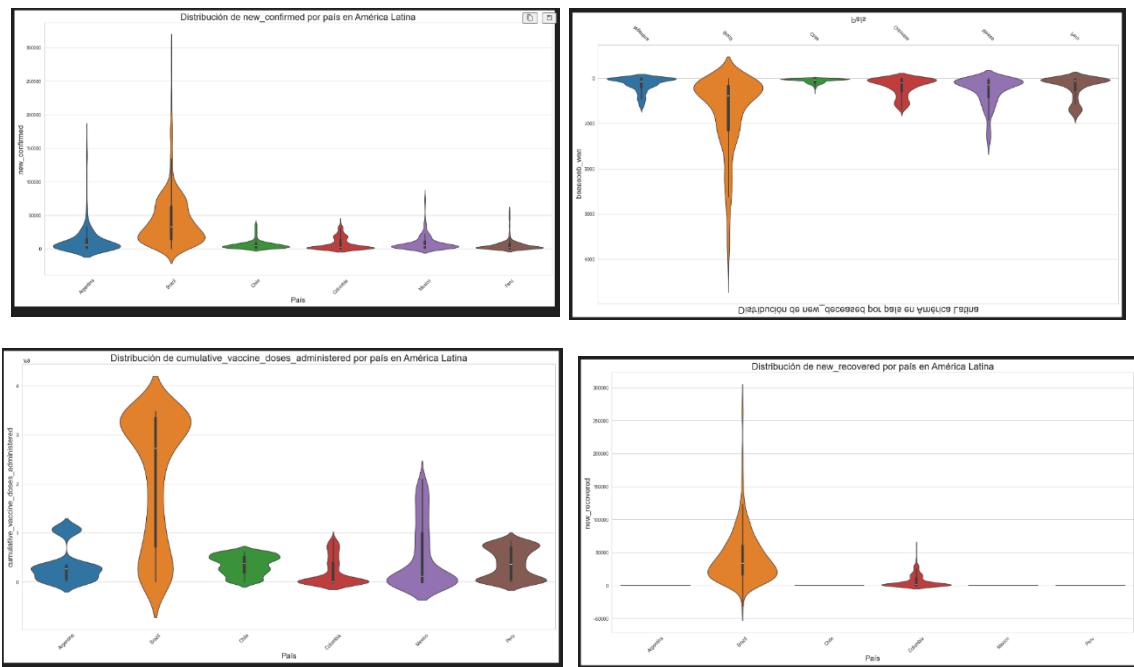
        # Creamos un gráfico de violín con Seaborn
        sns.violinplot(x='country_name', y=i, data=df_paises_covid_mayores, hue='country_name')

        # Añadimos un título y aumentamos el tamaño de la fuente
        plt.title(f'Distribución de {i} por país en América Latina', fontsize=20)

        # Aumentamos el tamaño de la fuente de las etiquetas de los ejes
        plt.xlabel('País', fontsize=16)
        plt.ylabel(i, fontsize=16)

        # Rotamos las etiquetas del eje x para evitar la superposición
        plt.xticks(rotation=45)

        # Mostramos el gráfico
        plt.show()
```



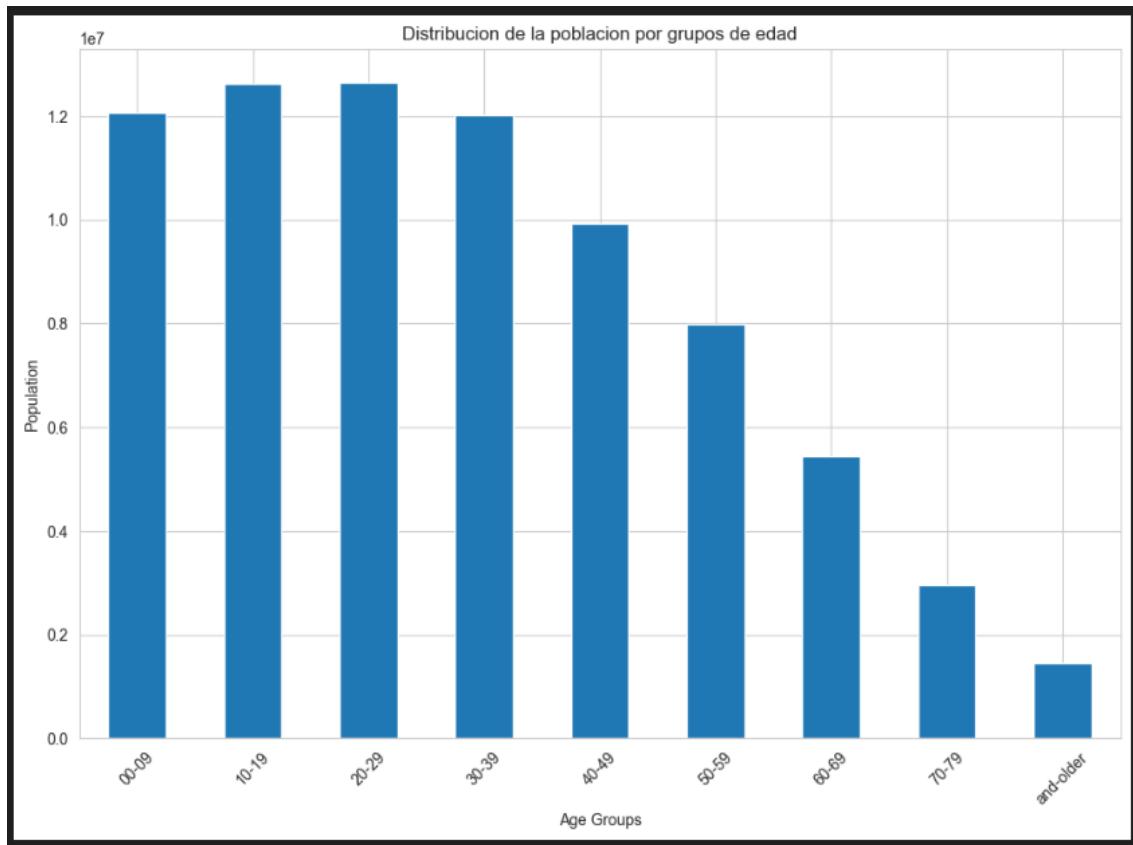
- Densidad de datos:** Las partes más anchas de los violines indican una mayor densidad de datos en esos rangos de valores.
- Comparaciones entre países:** Los gráficos ayudan a identificar diferencias y similitudes en la distribución de las variables entre los países, lo que puede ser útil para análisis comparativos.
- Análisis de variables:** Al analizar distintas variables (como prevalencia de enfermedades, tasas de vacunación, etc.), se pueden obtener insights sobre cómo cada país está manejando aspectos relacionados con el COVID-19.

15. Distribución de la población por grupos de edad

```
#calculamos la distribucion de la poblacion por grupos de edad
age_columns=['population_age_00_09','population_age_10_19','population_age_20_29','population_age_30_39',
             ['population_age_40_49','population_age_50_59','population_age_60_69','population_age_70_79','population_age_80_and_older']]
age_distribution = df_paises_covid[age_columns].mean()

plt.figure(figsize=(12,8))
age_distribution.plot(kind='bar', stacked=True)
plt.xlabel('Age Groups')
plt.ylabel('Population')
plt.title('Distribucion de la poblacion por grupos de edad')
plt.xticks(range(len(age_columns)), [col.split('_')[-2] + '-' + col.split('_')[-1] for col in age_columns], rotation=45)
plt.show()
```

Módulo 4



- Los grupos de edad más jóvenes (00-09, 10-19, 20-29) tienen una población significativamente alta. Esto sugiere que la población de América Latina es relativamente joven.
- La población disminuye a medida que aumenta el grupo de edad, con los grupos de edad más avanzada (70-79, 80 y más) teniendo la menor población. Esto es consistente con la expectativa de vida general y las tasas de mortalidad.
- Este gráfico puede ser útil para entender la distribución de la población que podría ser más vulnerable a la COVID-19, ya que se ha demostrado que los grupos de edad más avanzada son más susceptibles a complicaciones graves de la enfermedad.

Módulo 4

16. Mapa de calor de métricas por país

```
# Seleccionar las métricas a incluir en el mapa de calor
metricas = ['population_density', 'human_development_index', 'smoking_prevalence', 'diabetes_prevalence', 'life_expectancy']

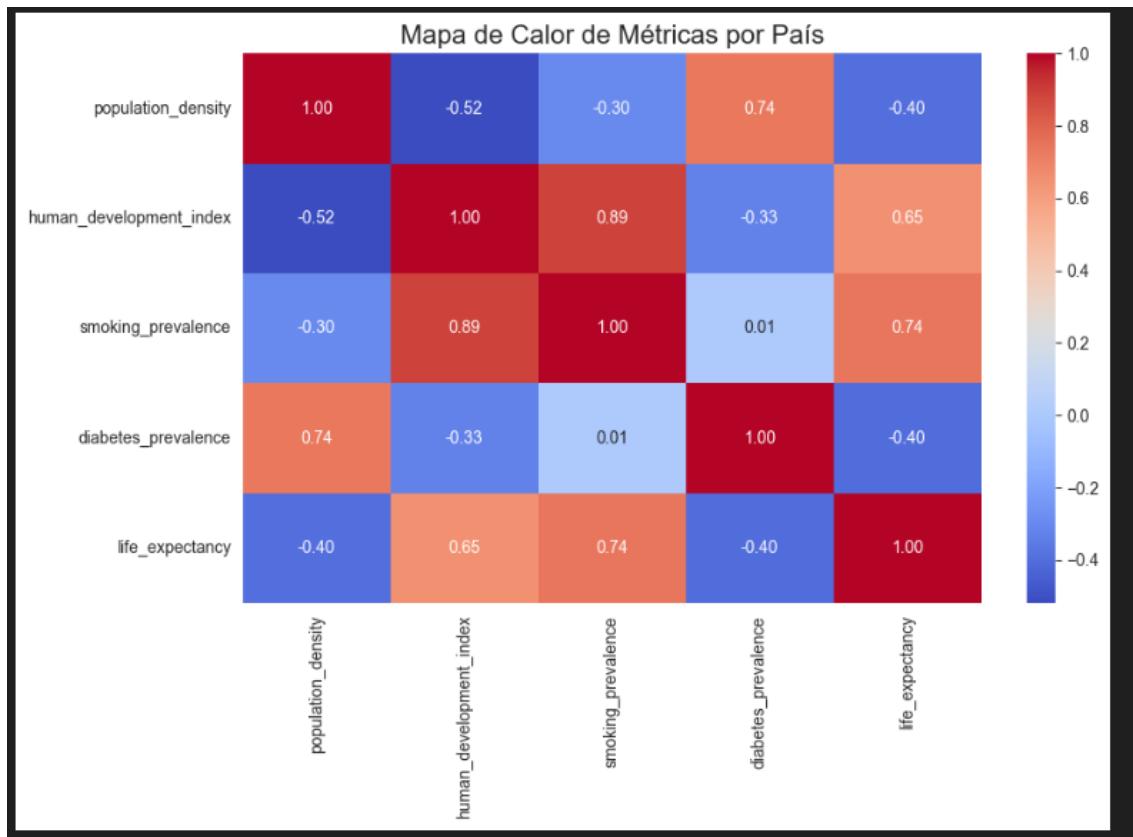
# Filtrar los datos para los países seleccionados
datos_filtrados = df_paises_covid[df_paises_covid['country_name'].isin(paises_seleccionados)]

# Calcular la matriz de correlación entre las métricas seleccionadas
correlacion = datos_filtrados[metricas].corr()

# Crear el mapa de calor
plt.figure(figsize=(10, 5))
sns.heatmap(correlacion, annot=True, cmap='coolwarm', fmt='.2f', annot_kws={"size": 10})

# Configurar el título
plt.title('Mapa de Calor de Métricas por País', fontsize=16)

# Mostrar el mapa de calor
plt.show()
```



- Los valores en el mapa de calor representan el coeficiente de correlación de Pearson entre las métricas. Un valor cercano a 1 indica una fuerte correlación positiva, un valor cercano a -1 indica una fuerte correlación negativa, y un valor cercano a 0 indica una débil o ninguna correlación.

Módulo 4

- Si dos métricas tienen una fuerte correlación positiva, significa que tienden a aumentar o disminuir juntas. Por ejemplo, si la esperanza de vida y el índice de desarrollo humano tienen una fuerte correlación positiva, podría sugerir que en los países donde la esperanza de vida es alta, también lo es el índice de desarrollo humano.
- Si dos métricas tienen una fuerte correlación negativa, significa que una tiende a aumentar cuando la otra disminuye. Por ejemplo, si la prevalencia de la diabetes y la esperanza de vida tienen una fuerte correlación negativa, podría sugerir que en los países donde la prevalencia de la diabetes es alta, la esperanza de vida tiende a ser baja.
- Las métricas que no tienen una correlación fuerte pueden no estar directamente relacionadas entre sí, o su relación puede ser influenciada por otros factores.

17. Diagrama de barras apilado de la comparación de la tasa de mortalidad masculina vs. femenina por país

```
# Establecemos el tamaño de la figura
plt.figure(figsize=(14,8))

# Creamos un gráfico de barras para la tasa de mortalidad masculina
sns.barplot(data=df_paises_covid, x='country_name', y='adult_male_mortality_rate', color='blue', label='Male Mortality Rate')

# Creamos un gráfico de barras para la tasa de mortalidad femenina
sns.barplot(data=df_paises_covid, x='country_name', y='adult_female_mortality_rate', color='red', alpha=0.8, label='Female Mortality Rate')

# Establecemos la etiqueta del eje y
plt.ylabel('Mortality rate', fontsize=16)
plt.xlabel('País', fontsize=16)

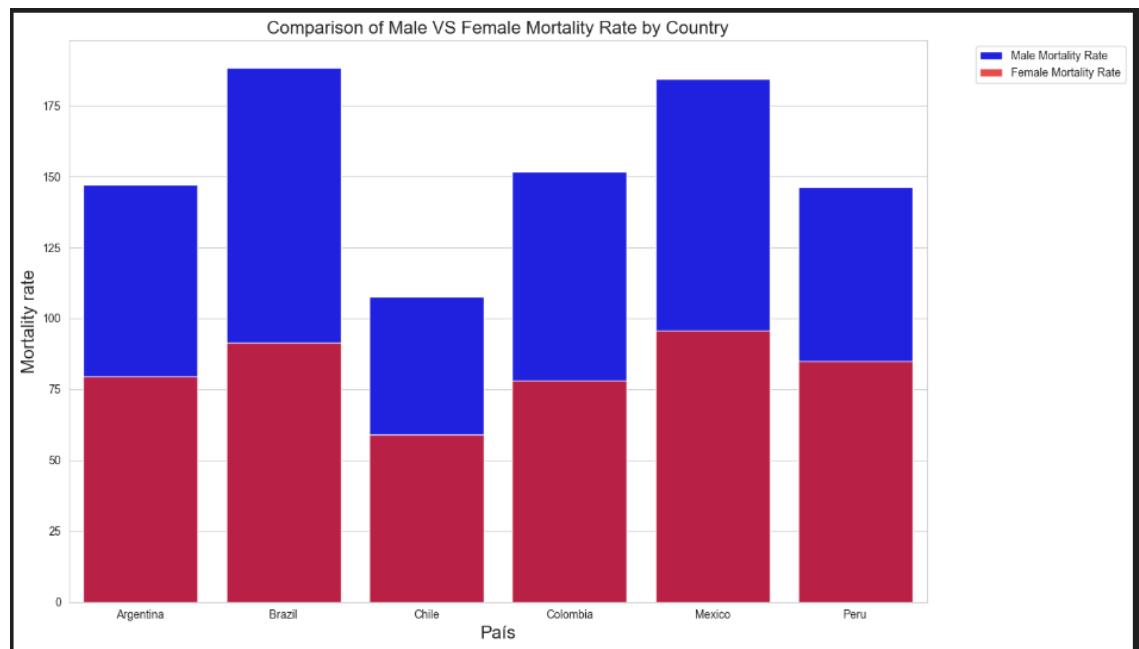
# Establecemos el título del gráfico
plt.title('Comparison of Male VS Female Mortality Rate by Country', fontsize=16)

# Ajustamos la leyenda del gráfico
plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')

# Mejoramos el layout para una mejor presentación
plt.tight_layout()

# Mostramos el gráfico
plt.show()
```

Módulo 4



- Comparación de género: En algunos países, la tasa de mortalidad masculina parece ser más alta que la femenina, mientras que en otros es al revés. Esto podría indicar diferencias en la salud, el acceso a la atención médica, o los factores de riesgo entre hombres y mujeres en estos países.
- Comparación entre países: Algunos países tienen tasas de mortalidad generalmente más altas que otros, tanto para hombres como para mujeres. Esto podría reflejar diferencias en la calidad de la atención médica, la prevalencia de enfermedades, los estilos de vida y otros factores de salud pública entre los países.
- Superposición de barras: Donde las barras rojas y azules se superponen, indica que las tasas de mortalidad masculina y femenina son similares.

Módulo 4

18. Matriz de Correlación de indicadores demográficos y de salud

```

# Este código genera un mapa de calor que muestra la mitad inferior de la matriz de correlación de ciertos indicadores demográficos y de salud.

# Se seleccionan las columnas para el análisis de correlación
columnas=['population_density','population_rural', 'population_urban','population_largest_city',
          'life_expectancy', 'smoking_prevalence','diabetes_prevalence','cumulative_deceased','human_development_index']

# Se calcula la matriz de correlación de las columnas seleccionadas
matriz_correlacion=df_paises_covid[columnas].corr()

# Se obtienen los valores absolutos de la matriz de correlación y se filtran aquellos que son mayores a 0
matriz_correlacion = matriz_correlacion[abs(matriz_correlacion)>0]

# Se crea una máscara para ocultar la mitad superior de la matriz de correlación
mask = np.triu(np.ones_like(matriz_correlacion, dtype=bool))

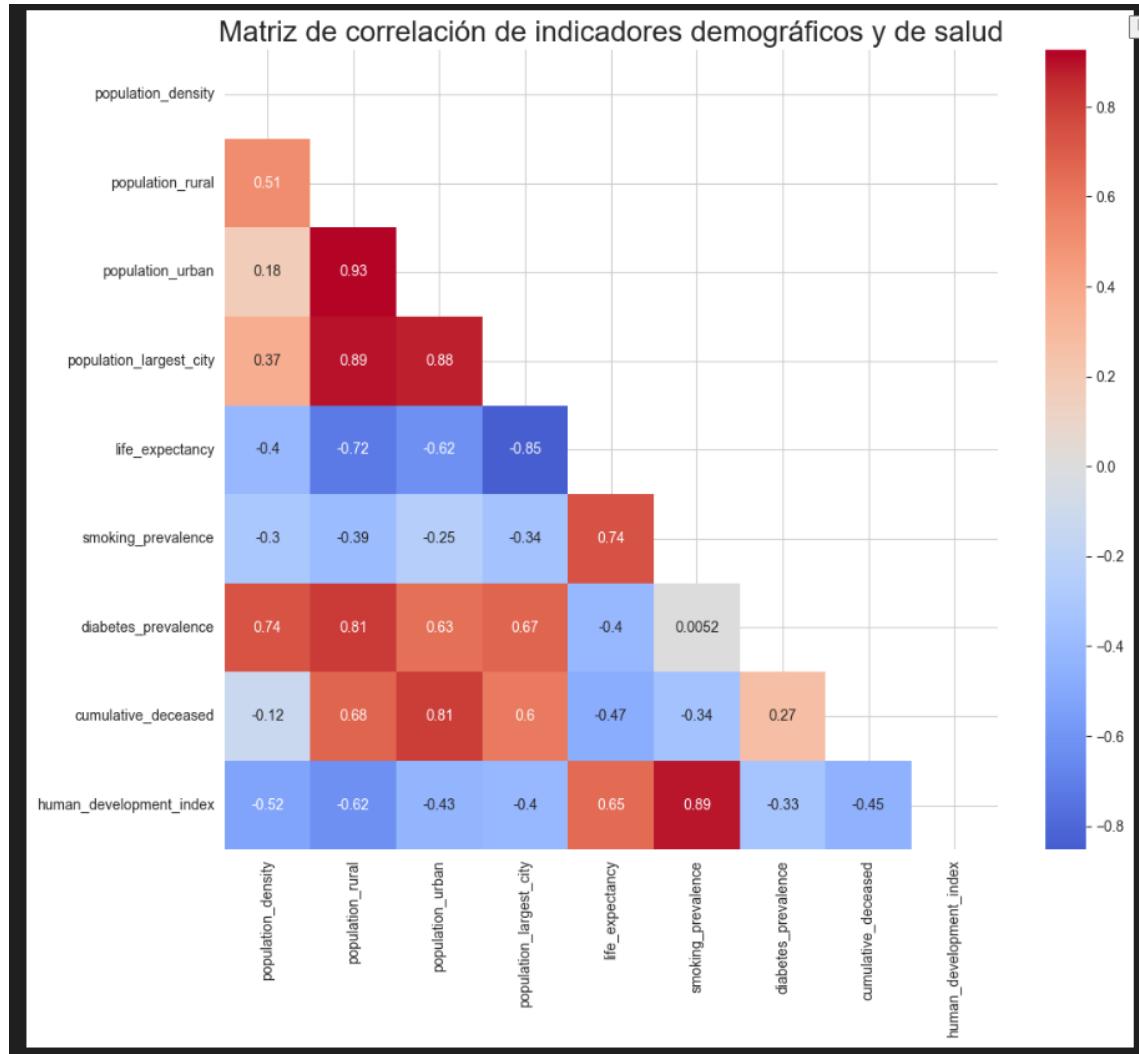
# Se crea una nueva figura con un tamaño más grande
plt.figure(figsize=(12,10))

# Se genera el mapa de calor usando seaborn, aplicando la máscara
sns.heatmap(matriz_correlacion, mask=mask, annot=True, cmap='coolwarm', center=0)

# Se establece el título del gráfico como 'Matriz de correlación de indicadores demográficos y de salud' con un tamaño de fuente mayor
plt.title('Matriz de correlación de indicadores demográficos y de salud', fontsize=20)

# Se muestra el gráfico
plt.show()

```



Módulo 4

- Los valores en el mapa de calor representan el coeficiente de correlación de Pearson entre los indicadores. Un valor cercano a 1 indica una fuerte correlación positiva, un valor cercano a -1 indica una fuerte correlación negativa, y un valor cercano a 0 indica una débil o ninguna correlación.
- Si dos indicadores tienen una fuerte correlación positiva, significa que tienden a aumentar o disminuir juntos. Por ejemplo, si la esperanza de vida y el índice de desarrollo humano tienen una fuerte correlación positiva, podría sugerir que en los países donde la esperanza de vida es alta, también lo es el índice de desarrollo humano.
- Si dos indicadores tienen una fuerte correlación negativa, significa que una tiende a aumentar cuando la otra disminuye. Por ejemplo, si la prevalencia de la diabetes y la esperanza de vida tienen una fuerte correlación negativa, podría sugerir que en los países donde la prevalencia de la diabetes es alta, la esperanza de vida tiende a ser baja.
- Los indicadores que no tienen una correlación fuerte pueden no estar directamente relacionados entre sí, o su relación puede ser influenciada por otros factores.

19. Comparación de la Prevalencia de Diabetes y la Tasa de Mortalidad

```
# Se crea una nueva figura con dos subtramas
fig,(ax,ax2)= plt.subplots(1,2,figsize=(15,10))

# Se genera el primer gráfico de barras
sns.barplot(data=df_paises_covid, x=df_paises_covid['country_name'], y=df_paises_covid['diabetes_prevalence'],
            color='yellow', label='Prevalencia de Diabetes')

ax2=ax.twinx()

# Se genera el segundo gráfico de barras
sns.barplot(data=df_paises_covid, x=df_paises_covid['country_name'], y=df_paises_covid['cumulative_deceased']/df_paises_covid['population'],
            color='Red', label = 'Tasa de Mortalidad por COVID-19', ax=ax2, alpha=0.8)

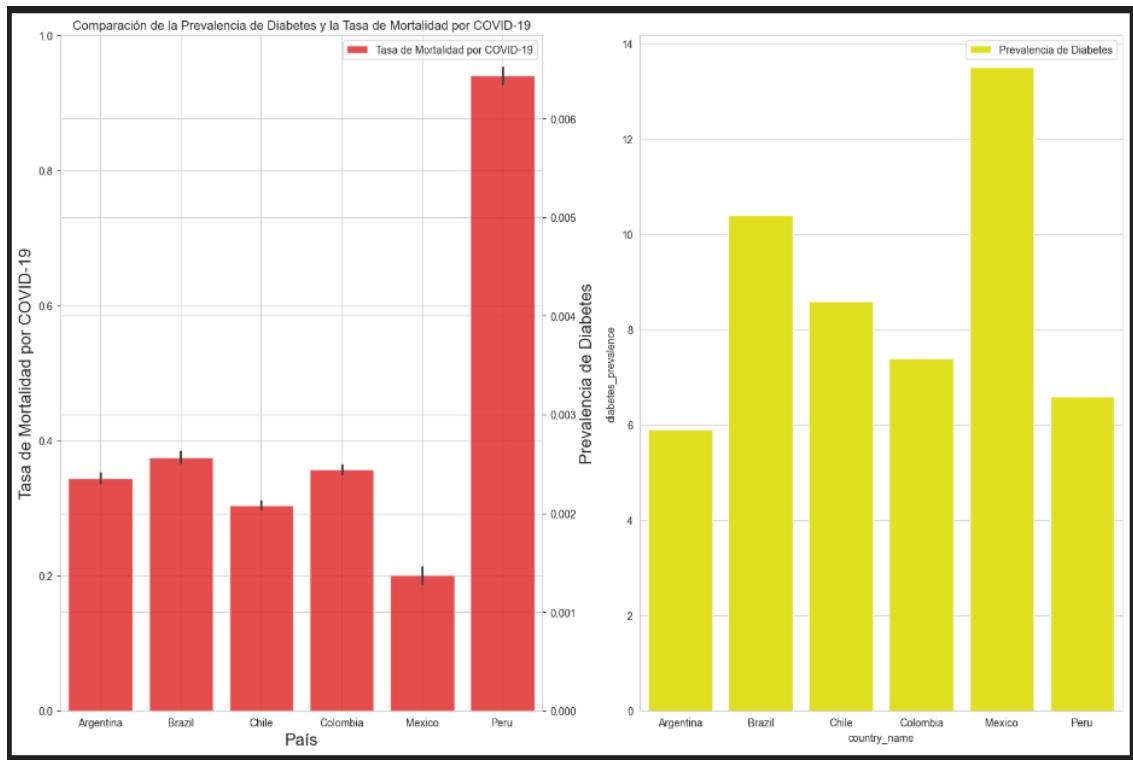
plt.xticks(rotation=90)

#Se establecen etiquetas en los ejes
ax.set_ylabel('Prevalencia de Diabetes', fontsize=16)
ax.set_xlabel('País', fontsize=16)
ax2.set_xlabel('País', fontsize=16)
ax.set_ylabel('Tasa de Mortalidad por COVID-19', fontsize=16)

#Se establece el título
plt.title('Comparación de la Prevalencia de Diabetes y la Tasa de Mortalidad por COVID-19')
# Se ajusta el diseño para evitar la superposición de elementos
plt.tight_layout()

# Se muestra el gráfico
plt.show()
```

Módulo 4



- Comparación entre países: Algunos países tienen una prevalencia de diabetes más alta que otros, y algunos países tienen una tasa de mortalidad por COVID-19 más alta que otros. Esto podría reflejar diferencias en la salud de la población, los estilos de vida, la calidad de la atención médica y otros factores de salud pública entre los países.
- Relación entre la diabetes y la COVID-19: Si las barras amarillas y rojas para un país son ambas altas, esto podría sugerir una posible relación entre la prevalencia de la diabetes y la tasa de mortalidad por COVID-19. Sin embargo, se necesitaría un análisis más detallado para confirmar cualquier relación causal.
- Superposición de barras: Donde las barras amarillas y rojas se superponen, indica que la prevalencia de la diabetes y la tasa de mortalidad por COVID-19 son similares.

EDA e insights

Se realizaron en Python una serie adicional de gráficos para descubrir Insights para tomar la mejor decisión en el proyecto que es identificar a ubicación para su expansión con nuevos laboratorios en América Latina.

1. Evolución de Casos Activos vs. Recuperados

```
# Calculamos los casos activos en cada país de América Latina
# Los casos activos se calculan como el total de casos confirmados menos los casos fallecidos y recuperados
df_paises_covid['active_cases'] = df_paises_covid['cumulative_confirmed'] - df_paises_covid['cumulative_deceased'] - df_paises_covid['cumulative_recovered']

# Creamos una nueva figura con un tamaño específico
plt.figure(figsize=(16, 9))

# Creamos un gráfico de líneas para los casos activos, resample('ME') se usa para agrupar los datos por el final de cada mes y calcular la media
# Usamos seaborn (sns) para el gráfico de líneas porque proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos
sns.lineplot(data=df_paises_covid.resample('ME').mean(numeric_only=True), x='date', y='active_cases', label='Casos Activos', linewidth=2.5, color='red')

# Creamos un gráfico de líneas para los casos recuperados
sns.lineplot(data=df_paises_covid.resample('ME').mean(numeric_only=True), x='date', y='cumulative_recovered', label='Casos Recuperados', linewidth=2.5, color='blue')

# Configuramos las etiquetas de los ejes y el título del gráfico
plt.xlabel('Fecha', fontsize=14)
plt.ylabel('Número de Casos', fontsize=14)
plt.title('Evolución de Casos Activos vs. Recuperados ', fontsize=16)

# Añadimos una leyenda para indicar qué línea representa a qué grupo de casos
plt.legend(fontsize=12)

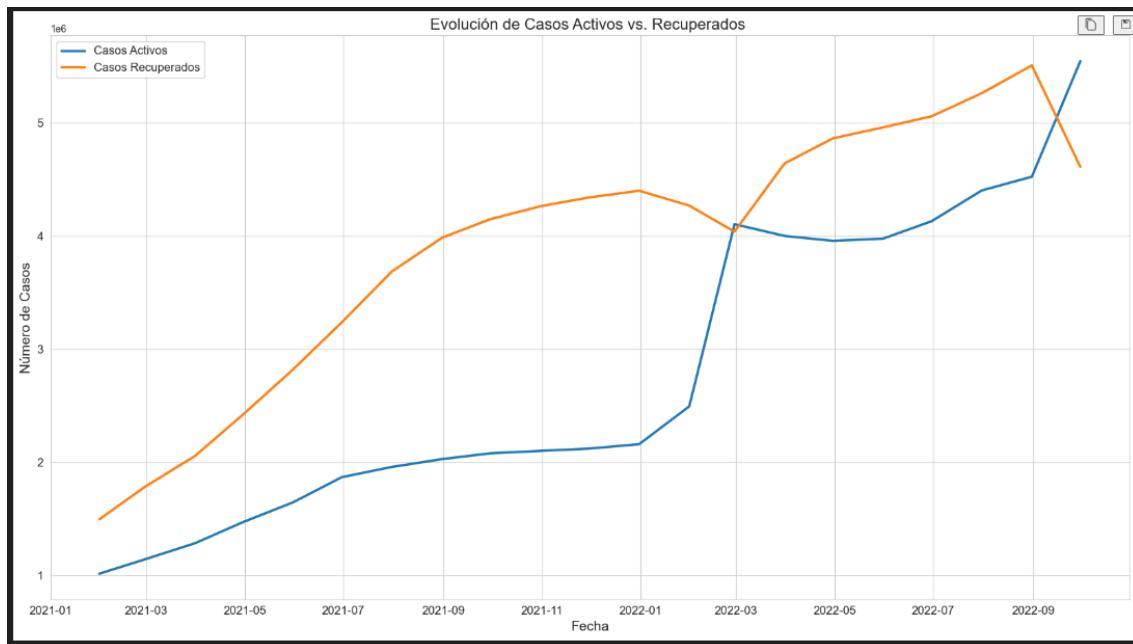
plt.yticks(fontsize=12)
plt.xticks(fontsize=12)

# Creamos un gráfico de líneas para los casos activos y lo configuramos en color rojo
sns.lineplot(data=df_paises_covid.resample('ME').mean(numeric_only=True), x='date', y='active_cases', label='Casos Activos', linewidth=2.5, color='red')

# Creamos un gráfico de líneas para los casos recuperados y lo configuramos en color azul
sns.lineplot(data=df_paises_covid.resample('ME').mean(numeric_only=True), x='date', y='cumulative_recovered', label='Casos Recuperados', linewidth=2.5, color='blue')

# Mejoramos el layout para una mejor presentación
plt.tight_layout()

# Mostramos el gráfico
plt.show()
```



Módulo 4

- Tendencias de los casos activos: La línea amarilla muestra cómo ha cambiado el número de casos activos de COVID-19 con el tiempo. Un aumento en la línea roja indica un aumento en los casos activos, mientras que una disminución indica una disminución en los casos activos.
- Tendencias de los casos recuperados: La línea azul muestra cómo ha cambiado el número de casos recuperados de COVID-19 con el tiempo. Un aumento en la línea azul indica un aumento en los casos recuperados, mientras que una disminución indica una disminución en los casos recuperados.
- Comparación entre casos activos y recuperados: Al comparar las dos líneas, podemos ver cómo se relacionan los casos activos y recuperados. Por ejemplo, si la línea roja es más alta que la azul, esto indica que hay más casos activos que recuperados.

2. Tasa de Crecimiento (%)

```
# Calculamos la tasa de crecimiento mensual de los nuevos casos confirmados
# Resample('ME') se usa para agrupar los datos por el final de cada mes
# pct_change() se usa para calcular el cambio porcentual de un mes al siguiente
# Multiplicamos por 100 para convertir la tasa a porcentaje
tasa_crecimiento_mensual = df_paises_covid.resample('ME')[['new_confirmed']].sum().pct_change() * 100

# Creamos una nueva figura con un tamaño específico
plt.figure(figsize=(16, 9))

# Creamos un gráfico de líneas para la tasa de crecimiento mensual
# Usamos seaborn (sns) para el gráfico de líneas porque proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos
sns.lineplot(x=tasa_crecimiento_mensual.index, y=tasa_crecimiento_mensual, color='blue', linewidth=2.5)

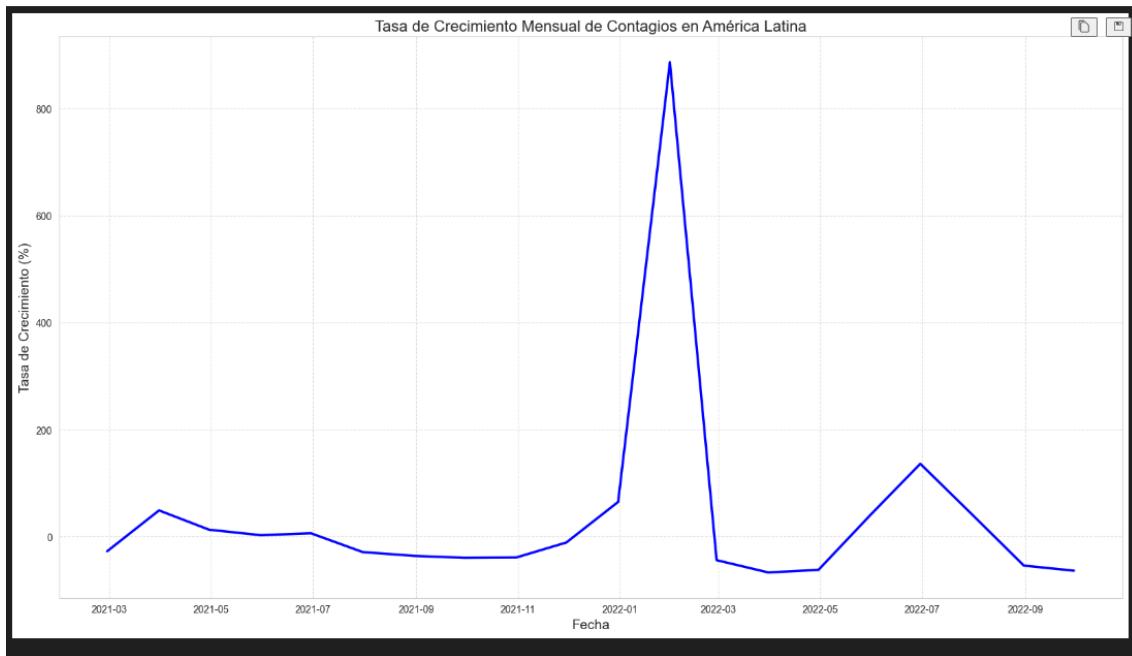
# Configuramos las etiquetas de los ejes y el título del gráfico
plt.xlabel('Fecha', fontsize=14)
plt.ylabel('Tasa de Crecimiento (%)', fontsize=14)
plt.title('Tasa de Crecimiento Mensual de Contagios en América Latina', fontsize=16)

# Añadimos una cuadrícula para facilitar la lectura de los datos
plt.grid(True, linestyle='--', alpha=0.6)

# Mejoraremos el layout para una mejor presentación
plt.tight_layout()

# Mostramos el gráfico
plt.show()
```

Módulo 4



- Tendencias de crecimiento: La línea azul muestra cómo ha cambiado la tasa de crecimiento mensual con el tiempo. Un aumento en la línea indica un aumento en la tasa de crecimiento, lo que significa que el número de nuevos casos está aumentando más rápidamente. Una disminución en la línea indica una disminución en la tasa de crecimiento, lo que significa que el número de nuevos casos está aumentando más lentamente o incluso disminuyendo.
- Picos y valles: Los picos en la línea indican meses en los que la tasa de crecimiento fue particularmente alta, lo que podría corresponder a brotes de la enfermedad. Los valles indican meses en los que la tasa de crecimiento fue particularmente baja, lo que podría corresponder a períodos de control efectivo de la enfermedad.
- Variabilidad a lo largo del tiempo: La variabilidad en la línea a lo largo del tiempo puede indicar cómo la situación de la COVID-19 ha cambiado en América Latina a lo largo del tiempo, con períodos de rápido crecimiento seguidos de períodos de crecimiento más lento.

Módulo 4

3. Relación de la cobertura de Vacunación y la Reducción de casos

```
# Como evoluciona la vacunación en América Latina y la reducción de casos de COVID-19
plt.figure(figsize=(14, 7))
ax1 = plt.gca()
ax2 = ax1.twinx()

# Crea porcentajes de vacunación
df_paises_covid['percentage_vaccinated'] = (df_paises_covid['cumulative_vaccine_doses_administered'] / df_paises_covid['population']) * 100

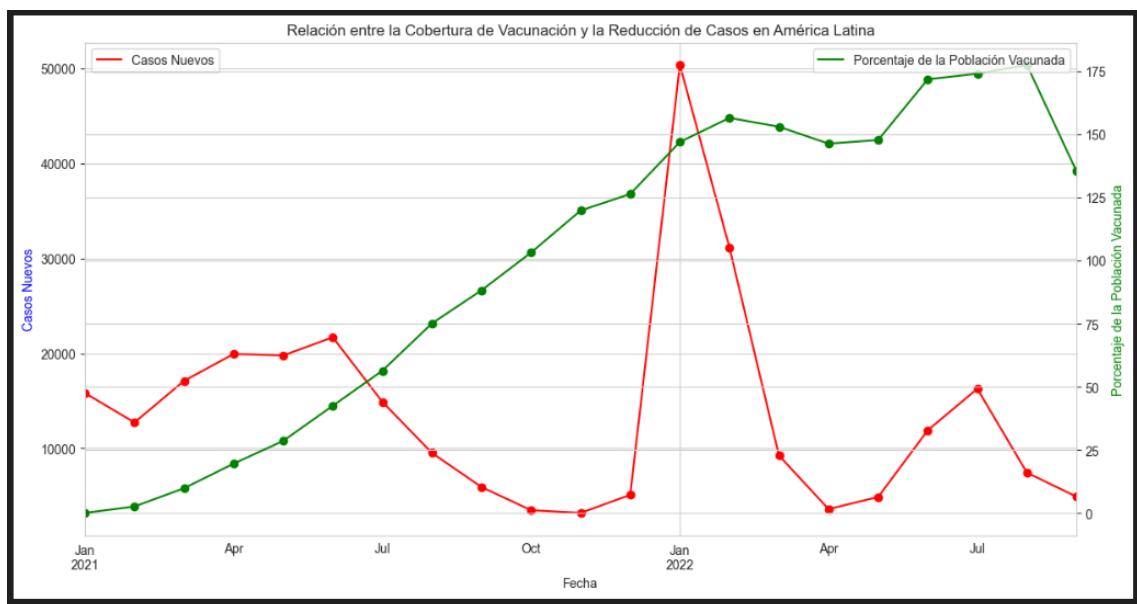
# Gráfico de los casos nuevos
df_paises_covid.resample('ME').mean(numeric_only=True)[['new_confirmed']].plot(ax=ax1, color='red', label='Casos Nuevos')

# Gráfico del porcentaje de vacunación
df_paises_covid.resample('ME').mean(numeric_only=True)[['percentage_vaccinated']].plot(ax=ax2, color='green', label='Porcentaje de la Población Vacunada')

# Agregamos puntos a cada cambio de mes
ax1.scatter(df_paises_covid.resample('ME').mean(numeric_only=True).index, df_paises_covid.resample('ME').mean(numeric_only=True)[['new_confirmed']], color='red')
ax2.scatter(df_paises_covid.resample('ME').mean(numeric_only=True).index, df_paises_covid.resample('ME').mean(numeric_only=True)[['percentage_vaccinated']], color='green')

# Configuración del gráfico
ax1.set_xlabel('Fecha')
ax1.set_ylabel('Casos Nuevos', color='blue')
ax2.set_ylabel('Porcentaje de la Población Vacunada', color='green')
ax1.set_title('Relación entre la Cobertura de Vacunación y la Reducción de Casos en América Latina')
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Mostrar el gráfico
plt.show()
```



- Tendencias de los casos activos: La línea de casos activos muestra cómo ha cambiado el número de casos activos de COVID-19 con el tiempo para cada país. Un aumento en la línea indica un aumento en los casos activos, mientras que una disminución indica una disminución en los casos activos.
- Tendencias de los casos recuperados: La línea de casos recuperados muestra cómo ha cambiado el número de casos recuperados de COVID-19 con el tiempo para cada país. Un aumento en la línea indica un aumento en los casos recuperados, mientras que una disminución indica una disminución en los casos recuperados.

Módulo 4

- Comparación entre casos activos y recuperados: Al comparar las dos líneas, podemos ver cómo se relacionan los casos activos y recuperados para cada país. Por ejemplo, si la línea de casos activos es más alta que la de casos recuperados, esto indica que hay más casos activos que recuperados en ese país.
- Comparación entre países: Al generar un gráfico para cada país, podemos comparar las tendencias de los casos activos y recuperados entre diferentes países.

4. Progreso de la vacunación por país

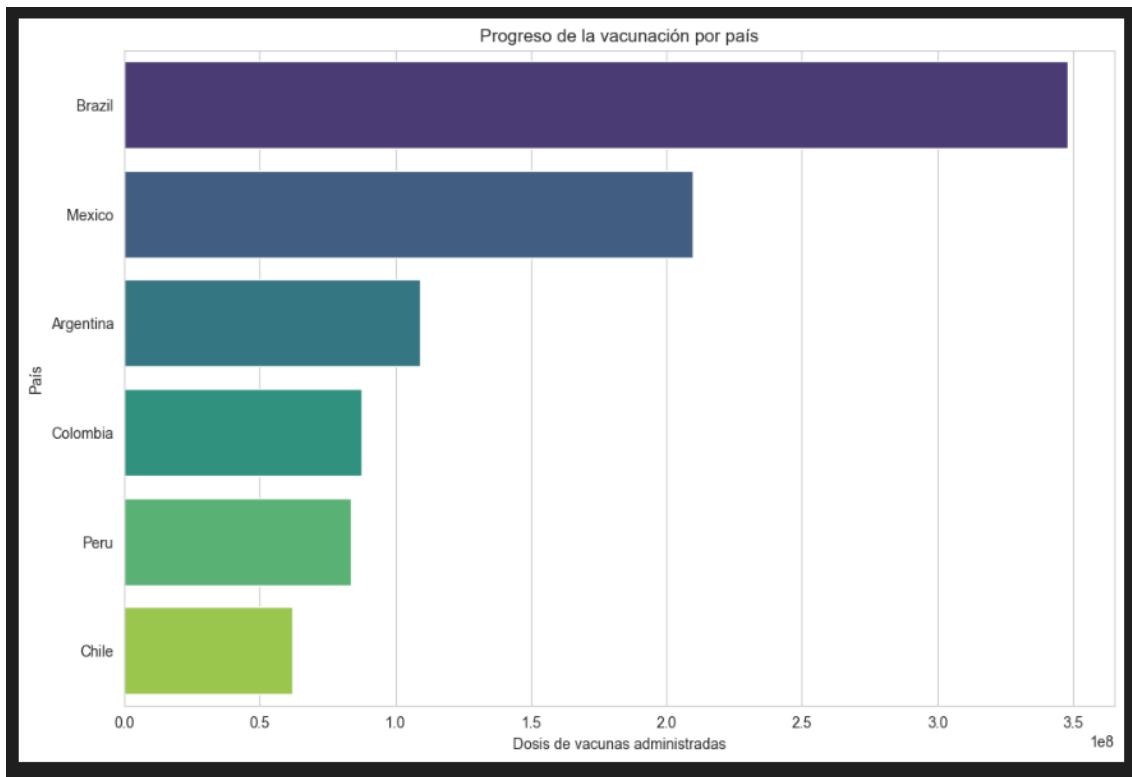
```
# Agrupar los datos de vacunación por país, calculando el máximo de dosis de vacunas administradas y ordenando de mayor a menor
vaccination_data = df_paises_covid.groupby('country_name')['cumulative_vaccine_doses_administered'].max().sort_values(ascending=False)

# Crear una figura para la gráfica con tamaño especificado
plt.figure(figsize=(12, 8))

# Graficar un gráfico de barras con los datos de vacunación
sns.barplot(x=vaccination_data.values, y=vaccination_data.index, palette='viridis')

# Etiquetas de los ejes y título de la gráfica
plt.xlabel('Dosis de vacunas administradas')
plt.ylabel('País')
plt.title('Progreso de la vacunación por país')

# Mostrar la gráfica
plt.show()
```



Módulo 4

- Comparación entre países: El gráfico permite comparar el progreso de la vacunación entre diferentes países. Los países en la parte superior del gráfico han administrado más dosis de vacunas que los países en la parte inferior.
- Progreso de la vacunación: La longitud de las barras indica el número de dosis de vacunas administradas, que puede ser un indicador del progreso de la vacunación en cada país. Sin embargo, este número no necesariamente representa el porcentaje de la población que ha sido vacunada, ya que algunas vacunas requieren más de una dosis por persona.
- Diferencias en la vacunación: Las diferencias en la longitud de las barras pueden reflejar diferencias en la disponibilidad de vacunas, las estrategias de vacunación, la aceptación de la vacuna por parte de la población y otros factores.

5. Nuevos casos y temperatura promedio

```
# Este código genera un gráfico de línea que muestra la relación entre la temperatura promedio y los nuevos casos confirmados de COVID-19.

# Se crea una nueva figura con un tamaño más grande
plt.figure(figsize=(20, 10))

# Se crea el primer eje (ax1) para los casos confirmados
ax1 = plt.gca()

# Se crea un segundo eje (ax2) que comparte el mismo eje x que ax1 para la temperatura
ax2 = ax1.twinx()

# Se grafican los nuevos casos confirmados, promediados mensualmente, en ax1
ax1.plot(
    df_paises_covid.resample('ME').mean(numeric_only=True).index,
    df_paises_covid.resample('ME').mean(numeric_only=True)[['new_confirmed']],
    color='red',
    label='Nuevos casos confirmados',
    linewidth=2
)
# Se grafica la temperatura promedio, promediada mensualmente, en ax2
ax2.plot(
    df_paises_covid.resample('ME').mean(numeric_only=True).index,
    df_paises_covid.resample('ME').mean(numeric_only=True)[['average_temperature_celsius']],
    color='blue',
    label='Temperatura promedio (°C)',
    linestyle='--',
    linewidth=2
)

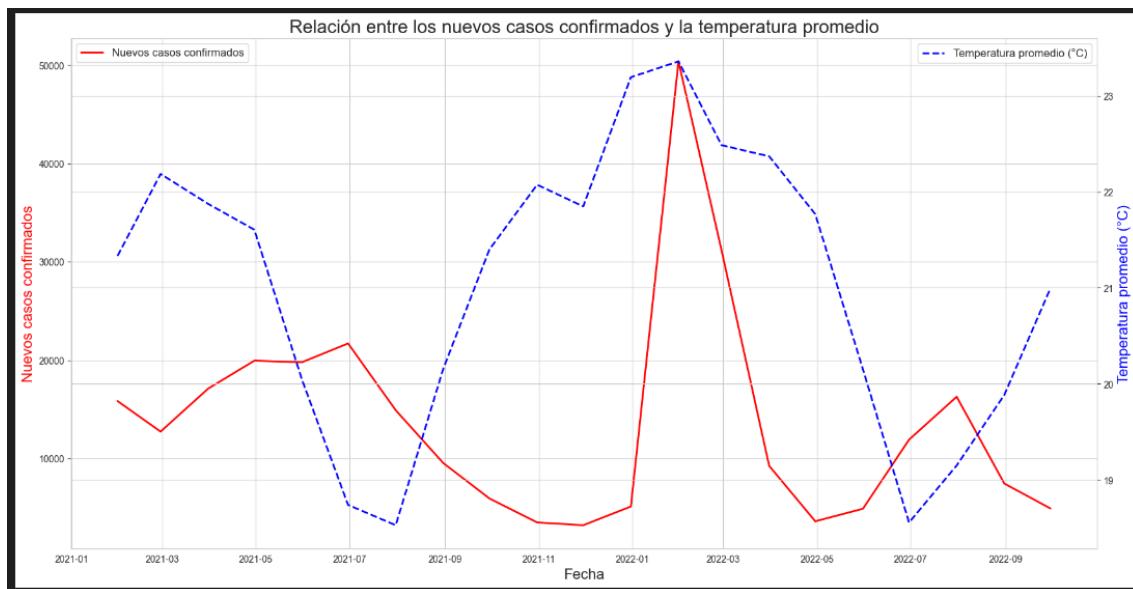
# Se configuran las etiquetas de los ejes con un tamaño de fuente mayor
ax1.set_xlabel('Fecha', fontsize=16)
ax1.set_ylabel('Nuevos casos confirmados', color='red', fontsize=16)
ax2.set_ylabel('Temperatura promedio (°C)', color='blue', fontsize=16)

# Se establece el título del gráfico como 'Relación entre la temperatura promedio y los nuevos casos confirmados de COVID-19' con un tamaño de fuente mayor
plt.title('Relación entre los nuevos casos confirmados y la temperatura promedio', fontsize=20)

# Se configuran las leyendas para ambos ejes
ax1.legend(loc='upper left', fontsize=12)
ax2.legend(loc='upper right', fontsize=12)

# Se muestra la gráfica
plt.show()
```

Módulo 4



- Tendencias de los casos nuevos: La línea roja muestra cómo ha cambiado el número de nuevos casos confirmados de COVID-19 con el tiempo. Un aumento en la línea roja indica un aumento en los casos nuevos, mientras que una disminución indica una disminución en los casos nuevos.
- Tendencias de la temperatura: La línea azul muestra cómo ha cambiado la temperatura promedio con el tiempo. Un aumento en la línea azul indica un aumento en la temperatura promedio, mientras que una disminución indica una disminución en la temperatura promedio.
- Relación entre la temperatura y los casos nuevos: Al comparar las dos líneas, podemos ver cómo se relacionan la temperatura promedio y los casos nuevos de COVID-19. Por ejemplo, si las dos líneas tienden a moverse juntas, esto podría sugerir una posible relación entre la temperatura y la propagación del COVID-19. Sin embargo, se necesitaría un análisis más detallado para confirmar cualquier relación causal.

Módulo 4**6. Impacto de la Urbanización en la Propagación del Covid-19**

```

# Calculamos el porcentaje de población urbana
df_paises_covid['urban_population_percentage'] = df_paises_covid['population_urban'] / df_paises_covid['population'] * 100

# Agrupamos los datos por país y tomamos el último valor de cada grupo, luego ordenamos por porcentaje de población urbana
urban_cases = df_paises_covid.groupby('country_name').last().sort_values(by='urban_population_percentage')

# Configuramos el tamaño de la figura
plt.figure(figsize=(12, 8))

# Creamos un gráfico de barras con los porcentajes de población urbana
# Asignamos la variable 'y' a 'hue' y establecemos 'legend=False' para el mismo efecto que 'palette'
sns.barplot(x=urban_cases['urban_population_percentage'], y=urban_cases.index, hue=urban_cases.index, legend=False)

# Añadimos etiquetas y título
plt.xlabel('Porcentaje de Población Urbana (%)', fontsize=14)
plt.ylabel('País', fontsize=14)

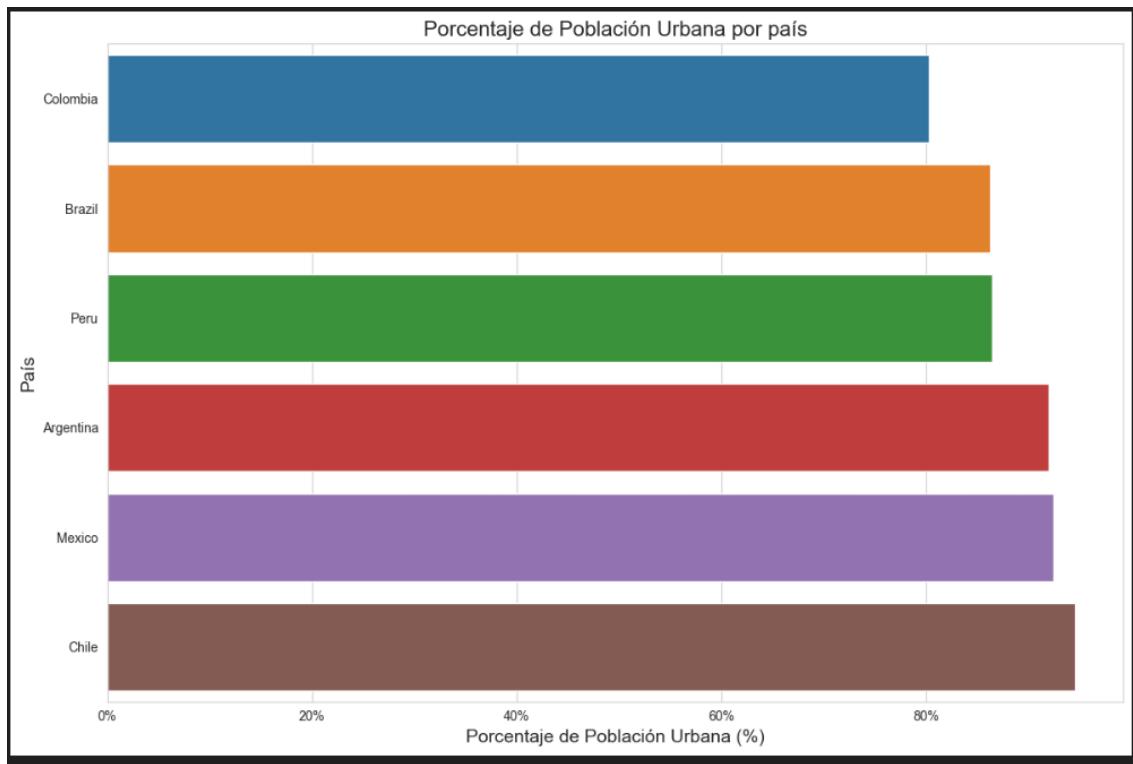
plt.title('Porcentaje de Población Urbana por país', fontsize=16)

# Formateamos el eje x para mostrar porcentajes
fmt = '%.0f%%'
xticks = mtick.FormatStrFormatter(fmt)
plt.gca().xaxis.set_major_formatter(xticks)

# Mejoramos el layout para una mejor presentación
plt.tight_layout()

# Mostramos el gráfico
plt.show()

```



- Comparación entre países: El gráfico permite comparar el porcentaje de población urbana entre diferentes países. Los países en la parte superior del

Módulo 4

gráfico tienen un mayor porcentaje de población urbana que los países en la parte inferior.

- **Urbanización:** La longitud de las barras indica el grado de urbanización en cada país. Un país con una barra más larga tiene una mayor proporción de su población viviendo en áreas urbanas.
- **Diferencias en la urbanización:** Las diferencias en la longitud de las barras pueden reflejar diferencias en el grado de urbanización entre los países, que pueden estar influenciadas por factores como el desarrollo económico, las políticas de planificación urbana y las tendencias demográficas.

7. Evolución Semanal de Casos Nuevos y

8. Evolución Anual de Casos Nuevos

```
# Este código genera dos gráficos de líneas que muestran la evolución de los casos nuevos de COVID-19 en América Latina a lo largo del tiempo, por semana y por mes.

# Se añaden las columnas 'week' y 'year' al DataFrame
df_paises_covid['week'] = df_paises_covid.index.isocalendar().week
df_paises_covid['year'] = df_paises_covid.index.isocalendar().year

# Se calcula la suma de los casos nuevos de COVID-19 por semana para cada país
weekly_cases = df_paises_covid.groupby(['country_name', 'week'])['new_confirmed'].sum().reset_index()

# Se crea una nueva figura con un tamaño más grande
plt.figure(figsize=(16, 10))

# Se genera un gráfico de líneas para cada país
for country in weekly_cases['country_name'].unique():
    country_weekly_cases = weekly_cases[weekly_cases['country_name'] == country]
    plt.plot(country_weekly_cases['week'], country_weekly_cases['new_confirmed'], label=country)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Semana del Año', fontsize=16)
plt.ylabel('Casos Nuevos de COVID-19', fontsize=16)

# Se establece el título del gráfico como 'Evolución Semanal de Casos Nuevos de COVID-19 en América Latina' con un tamaño de fuente mayor
plt.title('Evolución Semanal de Casos Nuevos de COVID-19 en América Latina', fontsize=20)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()

# Se añade la columna 'month' al DataFrame
df_paises_covid['month'] = df_paises_covid.index.month

# Se calcula la suma de los casos nuevos de COVID-19 por mes para cada país
yearly_cases = df_paises_covid.groupby(['country_name', 'month'])['new_confirmed'].sum().reset_index()

# Se crea una nueva figura con un tamaño más grande
plt.figure(figsize=(16, 10))

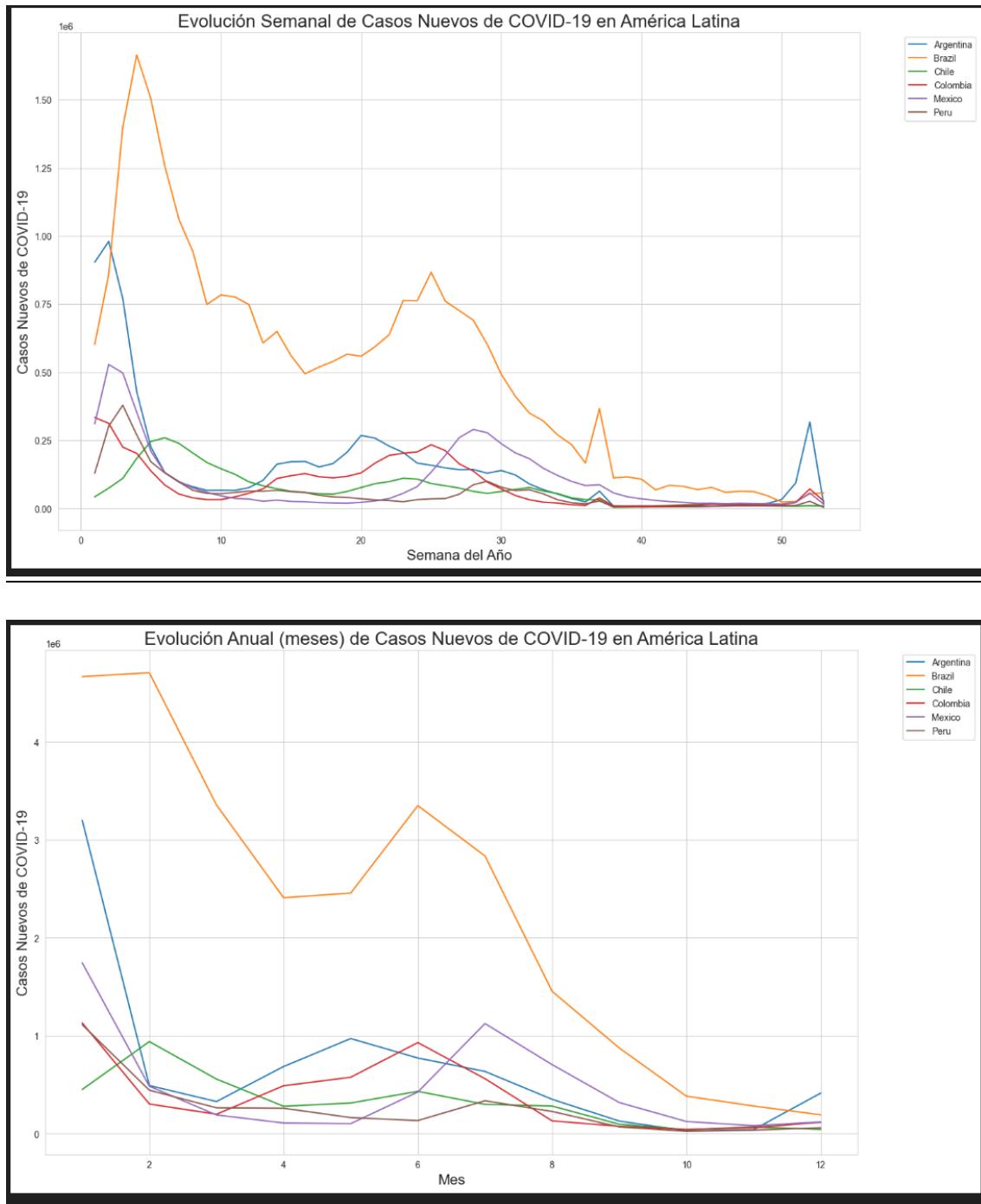
# Se genera un gráfico de líneas para cada país
for country in yearly_cases['country_name'].unique():
    country_yearly_cases = yearly_cases[yearly_cases['country_name'] == country]
    plt.plot(country_yearly_cases['month'], country_yearly_cases['new_confirmed'], label=country)

# Se añaden etiquetas a los ejes x e y con un tamaño de fuente mayor
plt.xlabel('Mes', fontsize=16)
plt.ylabel('Casos Nuevos de COVID-19', fontsize=16)

# Se establece el título del gráfico como 'Evolución Anual (meses) de Casos Nuevos de COVID-19 en América Latina' con un tamaño de fuente mayor
plt.title('Evolución Anual (meses) de Casos Nuevos de COVID-19 en América Latina', fontsize=20)

# Se mueve la leyenda fuera del área de la gráfica
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Se muestra el gráfico
plt.show()
```

Módulo 4

- Tendencias de los casos nuevos: Las líneas en los gráficos muestran cómo ha cambiado el número de casos nuevos de COVID-19 con el tiempo para cada país. Un aumento en la línea indica un aumento en los casos nuevos, mientras que una disminución indica una disminución en los casos nuevos.

Módulo 4

- Comparación entre países: Los gráficos permiten comparar la evolución de los casos nuevos de COVID-19 entre diferentes países. Esto puede ayudar a identificar qué países han tenido un mayor número de casos nuevos en diferentes momentos.
- Tendencias semanales vs. mensuales: Al comparar los dos gráficos, podemos ver cómo las tendencias de los casos nuevos de COVID-19 pueden variar en diferentes escalas de tiempo. Por ejemplo, puede haber fluctuaciones semanales que se suavizan cuando se mira a una escala mensual.
- Picos y valles: Los picos en las líneas indican semanas o meses en los que hubo un número particularmente alto de casos nuevos. Los valles indican semanas o meses en los que hubo un número particularmente bajo de casos nuevos.

9. Distribución de la Población por edad en países

```
# Se podría tomar los países con altas tasas de mortalidad por COVID-19,
#high_mortality_countries = df_paises_covid.groupby('country_name')['cumulative_deceased'].max().sort_values(ascending=False).head(3).index

#pero se graficaron todos los países para ver la distribución completa de los países por la edad de su población
high_mortality_countries = df_paises_covid.groupby('country_name')['cumulative_deceased'].max().sort_values(ascending=False).index

# Definimos los grupos de edad que queremos analizar
age_groups = [
    'population_age_00_09',
    'population_age_10_19',
    'population_age_20_29',
    'population_age_30_39',
    'population_age_40_49',
    'population_age_50_59',
    'population_age_60_69',
    'population_age_70_79',
    'population_age_80_and_older'
]

# Filtramos los datos para obtener la distribución de la población por edad en los países con alta mortalidad
age_distribution = df_paises_covid[df_paises_covid['country_name'].isin(high_mortality_countries)].groupby('country_name')[age_groups].sum()

# Graficamos la distribución de la población por edad
age_distribution.plot(kind='bar', stacked=False, figsize=(12, 8))

# Etiqueta del eje x
plt.xlabel('País')

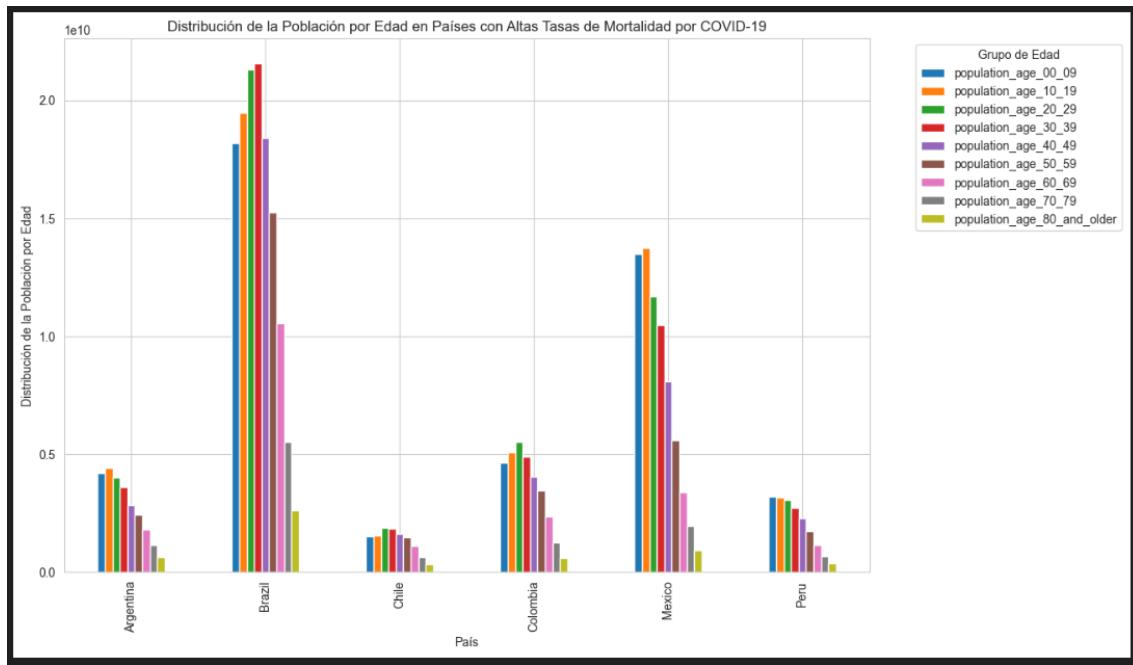
# Etiqueta del eje y
plt.ylabel('Distribución de la Población por Edad')

# Título de la gráfica
plt.title('Distribución de la Población por Edad en Países con Altas Tasas de Mortalidad por COVID-19')

# Configuración de la leyenda
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', title='Grupo de Edad')

# Mostrar la gráfica
plt.show()
```

Módulo 4



- Comparación entre países: El gráfico permite comparar la distribución de la población por grupos de edad entre diferentes países. Esto puede ayudar a identificar diferencias en la estructura de edad de la población que podrían estar relacionadas con las altas tasas de mortalidad por COVID-19.
- Distribución de la población por edad: Los segmentos de las barras indican la proporción de la población que se encuentra en cada grupo de edad. Esto puede proporcionar información sobre la demografía de cada país, como la proporción de personas mayores que podrían estar en mayor riesgo de enfermedad grave por COVID-19.
- Relación entre la edad y la mortalidad por COVID-19: Al observar la distribución de la población por edad en relación con las tasas de mortalidad por COVID-19, podemos obtener insights sobre cómo la edad puede estar relacionada con la mortalidad por COVID-19. Por ejemplo, si los países con una mayor proporción de personas mayores tienden a tener tasas de mortalidad más altas, esto podría sugerir una posible relación entre la edad y la mortalidad por COVID-19.

Módulo 4**10. Comparación de Estrategias de Vacunación en América Latina**

```
# Este código genera un gráfico de barras que muestra la cantidad de dosis de vacunas administradas por cada 100 personas en cada país.

# Calculamos la estrategia de vacunación por país
vaccination_strategy = df_paises_covid.groupby('country_name')['cumulative_vaccine_doses_administered'].max() / df_paises_covid.groupby('country_name')['population'].max()

# Ordenamos los valores de manera descendente
vaccination_strategy = vaccination_strategy.sort_values(ascending=False)

# Generamos una lista de colores aleatorios
colors = plt.cm.viridis(np.linspace(0, 1, len(vaccination_strategy)))

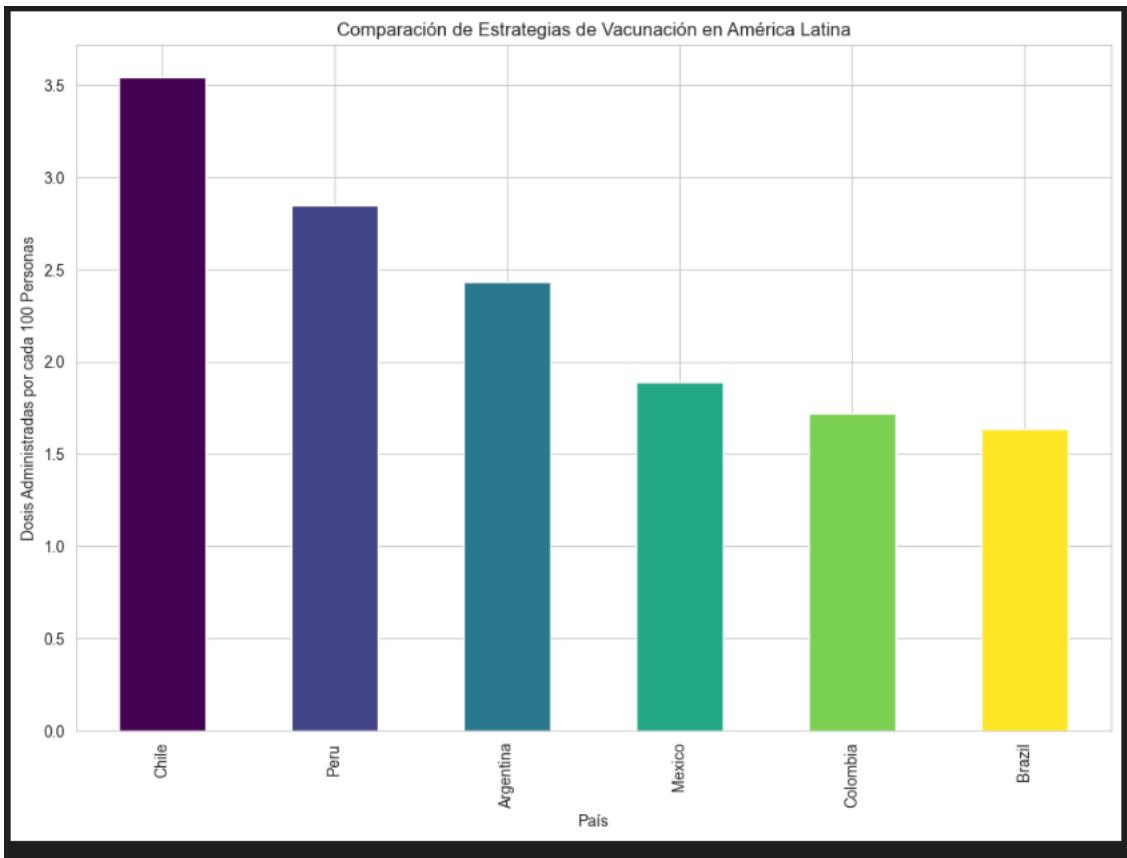
# Graficamos los datos usando un gráfico de barras con colores diferentes para cada barra
vaccination_strategy.plot(kind='bar', figsize=(12, 8), color=colors)

# Etiqueta del eje x
plt.xlabel('País')

# Etiqueta del eje y
plt.ylabel('Dosis Administradas por cada 100 Personas')

# Título del gráfico
plt.title('Comparación de Estrategias de Vacunación en América Latina')

# Mostramos el gráfico
plt.show()
```



- Comparación entre países: El gráfico permite comparar la cantidad de dosis de vacunas administradas por cada 100 personas entre diferentes países. Los países en la parte superior del gráfico han administrado más dosis de vacunas por cada 100 personas que los países en la parte inferior.

- Estrategias de vacunación: La longitud de las barras indica la cantidad de dosis de vacunas administradas por cada 100 personas, que puede ser un indicador de la agresividad de la estrategia de vacunación en cada país.
- Diferencias en la vacunación: Las diferencias en la longitud de las barras pueden reflejar diferencias en la disponibilidad de vacunas, las estrategias de vacunación, la aceptación de la vacuna por parte de la población y otros factores.

11. Evolución del Número de muertes diarias

```
# Ordenar los datos por fecha
df_paises_covid.sort_index(inplace=True)

# Crear una nueva figura
plt.figure(figsize=(20, 10))

# Calcular los promedios mensuales
df_resample = df_paises_covid.resample('ME').mean(numeric_only=True)

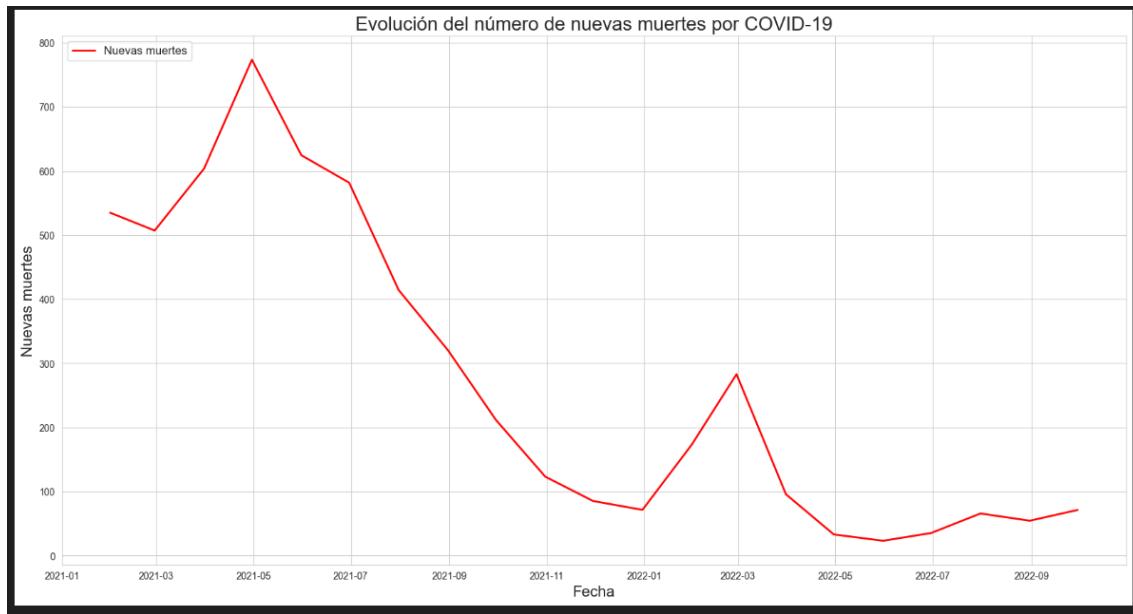
# Graficar los nuevos casos confirmados
plt.plot(
    df_resample.index,
    df_resample['new_deceased'],
    color='red',
    label='Nuevas muertes',
    linewidth=2
)

# Configurar las etiquetas de los ejes con un tamaño de fuente mayor
plt.xlabel('Fecha', fontsize=16)
plt.ylabel('Nuevas muertes', fontsize=16)

# Establecer el título del gráfico
plt.title('Evolución del número de nuevas muertes por COVID-19', fontsize=20)

# Configurar la leyenda
plt.legend(loc='upper left', fontsize=12)

# Mostrar la gráfica
plt.show()
```

Módulo 4

- Tendencias de las nuevas muertes: La línea roja muestra cómo ha cambiado el número de nuevas muertes por COVID-19 con el tiempo. Un aumento en la línea roja indica un aumento en las nuevas muertes, mientras que una disminución indica una disminución en las nuevas muertes.
- Picos y valles: Los picos en la línea indican momentos en los que hubo un número particularmente alto de nuevas muertes. Los valles indican momentos en los que hubo un número particularmente bajo de nuevas muertes.
- Impacto de la pandemia: Este gráfico puede proporcionar una visión general del impacto de la pandemia en términos de mortalidad. Sin embargo, para una interpretación más completa, sería útil comparar estos datos con otros indicadores, como el número de casos confirmados, la capacidad del sistema de salud, entre otros.

Módulo 4

12. Prevalencia de condiciones preexistentes en países con altas y bajas tasas de mortalidad

```

# Prevalencia de diabetes

# Obtener los 5 países con mayor número de fallecimientos acumulados
top_countries = df_paises_covid.groupby('country_name')['cumulative_deceased'].max().nlargest(5).index

# Obtener los 5 países con menor número de fallecimientos acumulados
bottom_countries = df_paises_covid.groupby('country_name')['cumulative_deceased'].max().nsmallest(5).index

# Filtrar los datos para incluir solo los países con las tasas más altas y más bajas de mortalidad
conditions_data = df_paises_covid[df_paises_covid['country_name'].isin(top_countries.union(bottom_countries))]

# Configuración del tamaño de la figura para el gráfico de diabetes
plt.figure(figsize=(14, 8))

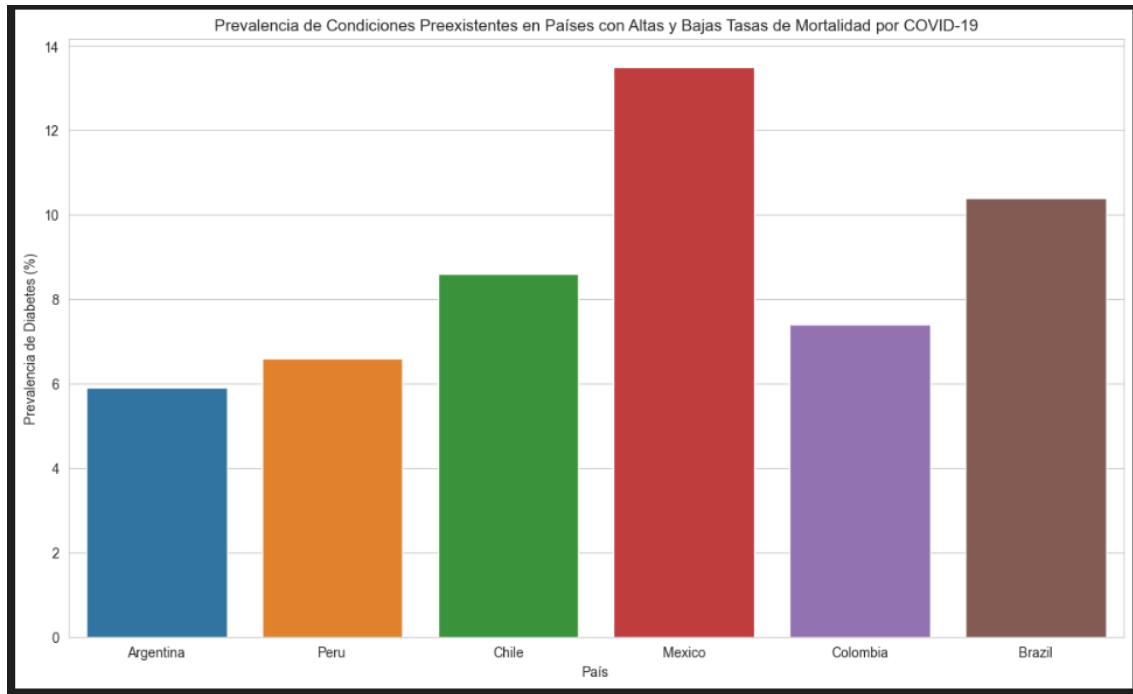
# Graficar la prevalencia de diabetes
sns.barplot(data=conditions_data, x='country_name', y='diabetes_prevalence', hue='country_name')
plt.xlabel('País')
plt.ylabel('Prevalencia de Diabetes (%)')
plt.title('Prevalencia de Condiciones Preexistentes en Países con Altas y Bajas Tasas de Mortalidad por COVID-19')
plt.show()

# Prevalencia de fumar

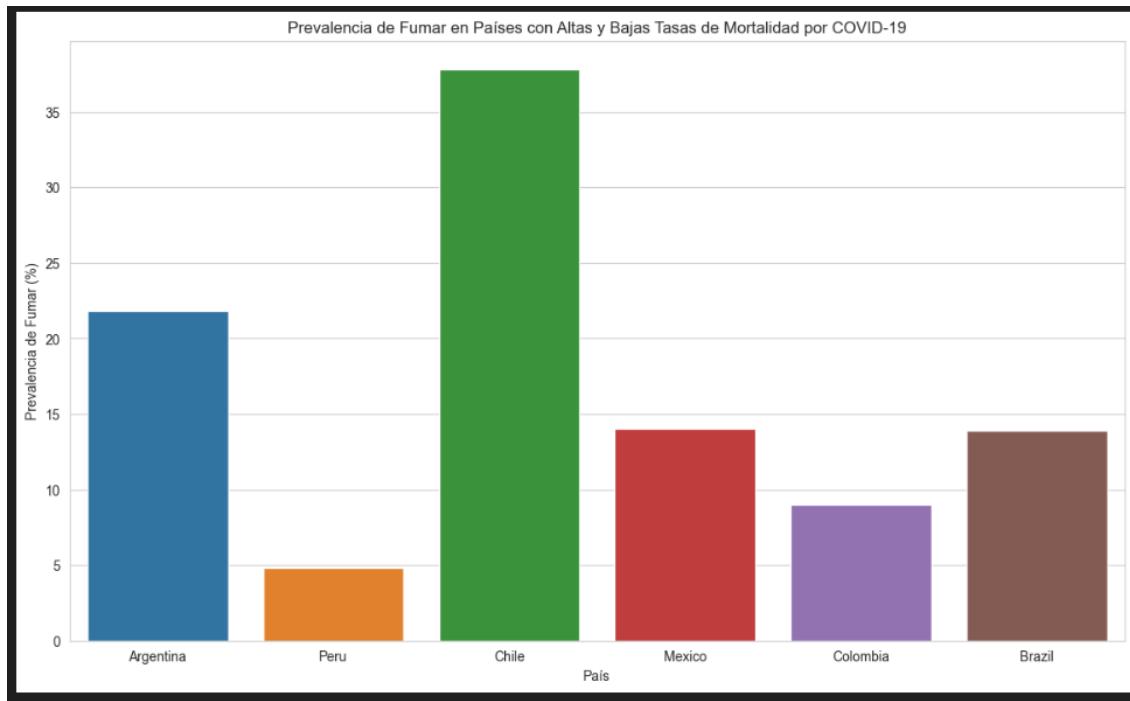
# Configuración del tamaño de la figura para el gráfico de fumar
plt.figure(figsize=(14, 8))

# Graficar la prevalencia de fumar
sns.barplot(data=conditions_data, x='country_name', y='smoking_prevalence', hue='country_name')
plt.xlabel('País')
plt.ylabel('Prevalencia de Fumar (%)')
plt.title('Prevalencia de Fumar en Países con Altas y Bajas Tasas de Mortalidad por COVID-19')
plt.show()

```



Módulo 4



- Comparación entre países: Los gráficos permiten comparar la prevalencia de diabetes y fumar entre diferentes países. Esto puede ayudar a identificar diferencias en la prevalencia de estas condiciones que podrían estar relacionadas con las tasas de mortalidad por COVID-19.
- Prevalencia de condiciones preexistentes: Las barras indican la prevalencia de diabetes y fumar en cada país. Esto puede proporcionar información sobre la salud de la población en cada país, que podría estar relacionada con la vulnerabilidad a la enfermedad grave por COVID-19.
- Relación entre condiciones preexistentes y mortalidad por COVID-19: Al observar la prevalencia de estas condiciones en relación con las tasas de mortalidad por COVID-19, podemos obtener insights sobre cómo las condiciones preexistentes pueden estar relacionadas con la mortalidad por COVID-19. Por ejemplo, si los países con una mayor prevalencia de diabetes o fumar tienden a tener tasas de mortalidad más altas, esto podría sugerir una posible relación entre estas condiciones y la mortalidad por COVID-19.

Módulo 4**13. Variación Mensual de Casos y Muertes**

```
# Calcula la tasa de letalidad por país
df_paises_covid['case_fatality_rate'] = df_paises_covid['cumulative_deceased'] / df_paises_covid['cumulative_confirmed']
fatality_rates = df_paises_covid.groupby('country_name')['case_fatality_rate'].last().sort_values()

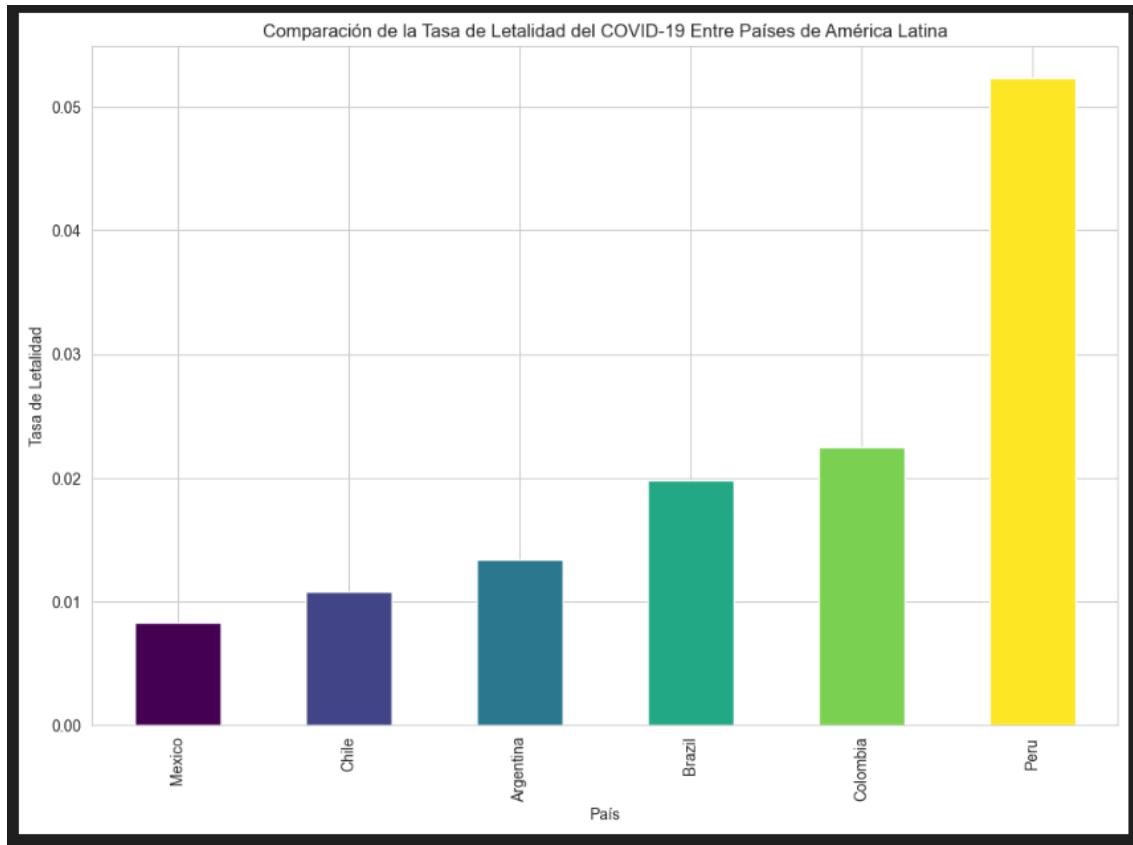
# Crea el gráfico de barras
plt.figure(figsize=(12, 8))

# Generamos una lista de colores aleatorios
colors = plt.cm.viridis(np.linspace(0, 1, len(fatality_rates)))

# Graficamos los datos usando un gráfico de barras con colores diferentes para cada barra
bars = fatality_rates.plot(kind='bar', color=colors)

# Configura los ejes y el título del gráfico
plt.xlabel('País')
plt.ylabel('Tasa de Letalidad')
plt.title('Comparación de la Tasa de Letalidad del COVID-19 Entre Países de América Latina')

# Muestra el gráfico
plt.show()
```



- Comparación entre países: El gráfico permite comparar la tasa de letalidad del COVID-19 entre diferentes países. Los países en la parte superior del gráfico tienen una mayor tasa de letalidad que los países en la parte inferior.
- Impacto del COVID-19: La longitud de las barras indica la tasa de letalidad del COVID-19, que puede ser un indicador del impacto de la enfermedad en cada país.

Módulo 4

- Diferencias en la letalidad: Las diferencias en la longitud de las barras pueden reflejar diferencias en la letalidad del COVID-19 entre los países, que pueden estar influenciadas por factores como la capacidad del sistema de salud, la prevalencia de condiciones preexistentes, las estrategias de mitigación y otros factores.

14. Análisis temporal de la Mortalidad

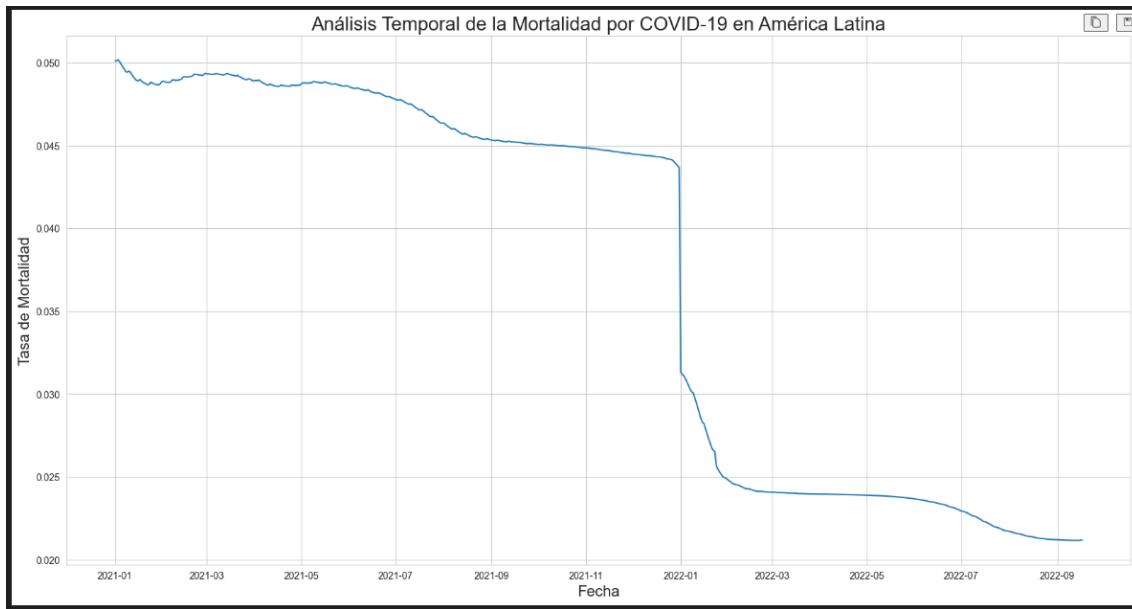
```
# Calculamos la tasa de mortalidad como el número acumulado de fallecidos dividido por el número acumulado de confirmados
df_paises_covid['mortality_rate'] = df_paises_covid['cumulative_deceased'] / df_paises_covid['cumulative_confirmed']

# Creamos una nueva figura con un tamaño más grande para mejorar la legibilidad
plt.figure(figsize=(20, 10))

# Creamos un gráfico de líneas con Seaborn
# Establecemos 'estimator' en 'mean' para mostrar solo la media de la tasa de mortalidad
sns.lineplot(data=df_paises_covid, x='date', y='mortality_rate', estimator='mean', errorbar=None)

# Aumentamos el tamaño de la fuente del título y las etiquetas de los ejes para mejorar la legibilidad
plt.title('Análisis Temporal de la Mortalidad por COVID-19 en América Latina', fontsize=20)
plt.xlabel('Fecha', fontsize=16)
plt.ylabel('Tasa de Mortalidad', fontsize=16)

# Mostramos el gráfico
plt.show()
```



- Tendencias de la tasa de mortalidad: La línea en el gráfico muestra cómo ha cambiado la tasa de mortalidad por COVID-19 con el tiempo. Un aumento en la línea indica un aumento en la tasa de mortalidad, mientras que una disminución indica una disminución en la tasa de mortalidad.

Módulo 4

- Impacto de la pandemia: Este gráfico puede proporcionar una visión general del impacto de la pandemia en términos de mortalidad. Sin embargo, para una interpretación más completa, sería útil comparar estos datos con otros indicadores, como el número de casos confirmados, la capacidad del sistema de salud, entre otros.
- Picos y valles: Los picos en la línea indican momentos en los que hubo una tasa de mortalidad particularmente alta. Los valles indican momentos en los que hubo una tasa de mortalidad particularmente baja.

```
# Calculamos la tasa de mortalidad como el número acumulado de fallecidos dividido por el número acumulado de confirmados
df_paises_covid['mortality_rate'] = df_paises_covid['cumulative_deceased'] / df_paises_covid['cumulative_confirmed']

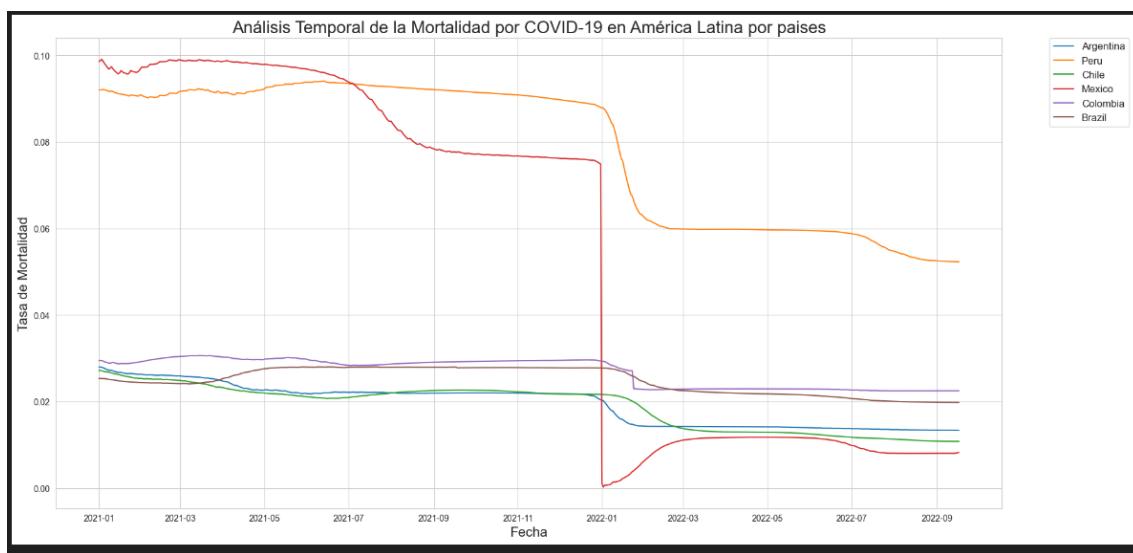
# Creamos una nueva figura con un tamaño más grande para mejorar la legibilidad
plt.figure(figsize=(20, 10))

# Creamos un gráfico de líneas con Seaborn, donde cada línea representa un país
# Usamos el argumento 'hue' para diferenciar los países por color
sns.lineplot(data=df_paises_covid, x='date', y='mortality_rate', hue='country_name', estimator='mean', errorbar=None)

# Aumentamos el tamaño de la fuente del título y las etiquetas de los ejes para mejorar la legibilidad
plt.title('Análisis Temporal de la Mortalidad por COVID-19 en América Latina por países', fontsize=20)
plt.xlabel('Fecha', fontsize=16)
plt.ylabel('Tasa de Mortalidad', fontsize=16)

# Movemos la leyenda fuera del área del gráfico para evitar la superposición
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0., fontsize=12)

# Mostramos el gráfico
plt.show()
```



15. Comparación de Situación Actual (Contexto de los datos)

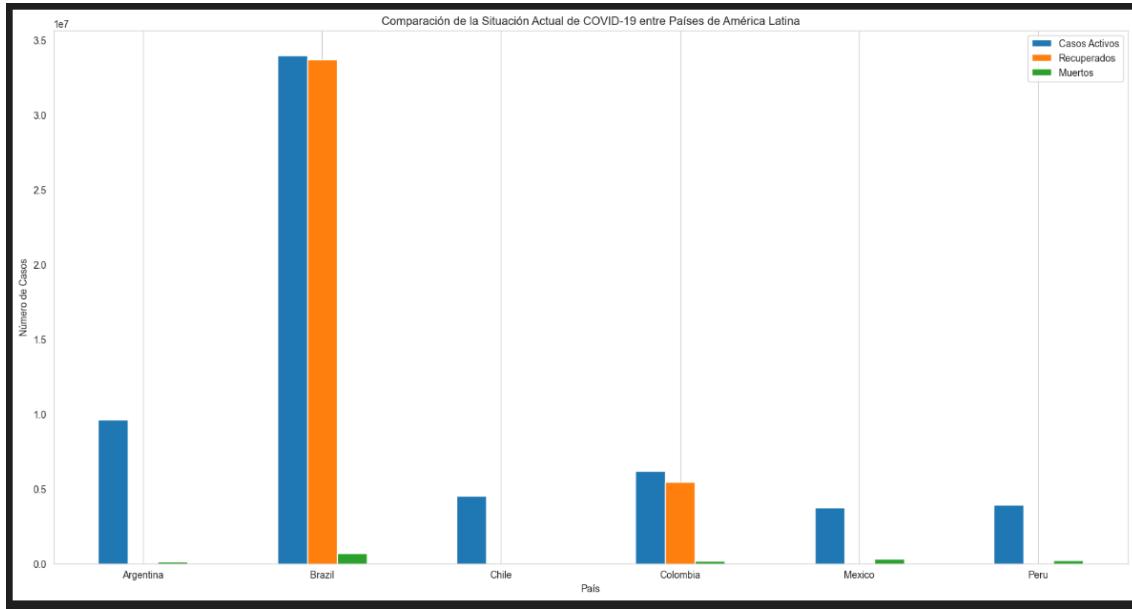
```
# No hay datos de recuperados y muertes en algunos países

# Agrupa los datos por nombre del país y obtiene el valor máximo para 'active_cases', 'cumulative_recovered' y 'cumulative_deceased'
situacion_actual = df_paises_covid.groupby('country_name').max()[['active_cases', 'cumulative_recovered', 'cumulative_deceased']]

# Crea el gráfico de barras
plt.figure(figsize=(6, 4)) # Configura el tamaño de la figura del gráfico
situacion_actual.plot(kind='bar', stacked=False, figsize=(20, 10)) # Grafica los datos en un gráfico de barras no apiladas con colores específicos

# Configura el título y etiquetas del gráfico
plt.title('Comparación de la Situación Actual de COVID-19 entre Países de América Latina')
# Etiqueta del eje X
plt.xlabel('País')
# Etiqueta del eje Y
plt.ylabel('Número de Casos')
# Leyenda para cada barra
plt.legend(['Casos Activos', 'Recuperados', 'Muertos'])
# Agregar líneas solo en el eje Y
plt.grid(axis='y')
# Rotación de las etiquetas en el eje X
plt.xticks(rotation=0)

# Muestra el gráfico en pantalla
plt.show()
```



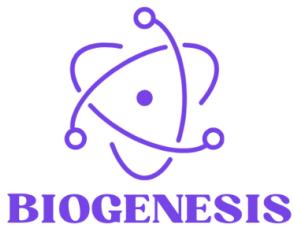
- Comparación entre países: El gráfico permite comparar la situación de COVID-19 entre diferentes países. Esto puede ayudar a identificar qué países tienen más casos activos, más recuperados o más fallecidos.
- Situación actual de la pandemia: Las barras indican la situación actual de la pandemia en cada país. Esto puede proporcionar información sobre la gravedad de la pandemia en cada país.

Módulo 4

- Impacto de la pandemia: Al observar el número de casos activos, recuperados y fallecidos, podemos obtener insights sobre el impacto de la pandemia en cada país. Por ejemplo, un alto número de fallecidos en relación con el número de casos activos y recuperados podría indicar una alta tasa de mortalidad.

Análisis del Dashboard

A continuación analizaremos el Dashboard elaborado mediante Power Bi



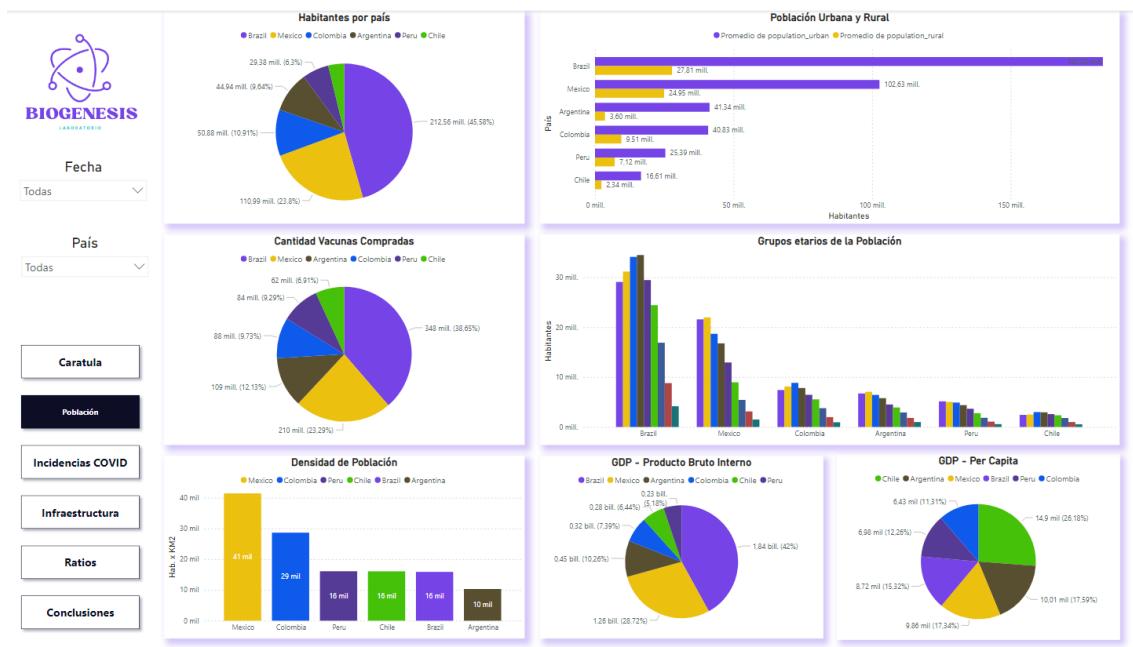
EXPANSIÓN ESTRATÉGICA DE LABORATORIOS PARA INVESTIGACIÓN Y VACUNACIÓN DE COVID-19

OBJETIVO: Identificar las ubicaciones optimas para la expansión de laboratorios farmacéuticos, basándose en el análisis de datos de incidencia de COVID-19, tasa de vacunación y la disponibilidad de infraestructuras sanitarias.

Elaborado por : Arturo Ponce de León - Data Analyst



En la carátula mostramos el nombre del proyecto y el objetivo principal del desarrollo del mismo. Así mismo contamos con un menú de navegación inferior para ir recorriendo las diferentes páginas del Dashboard.

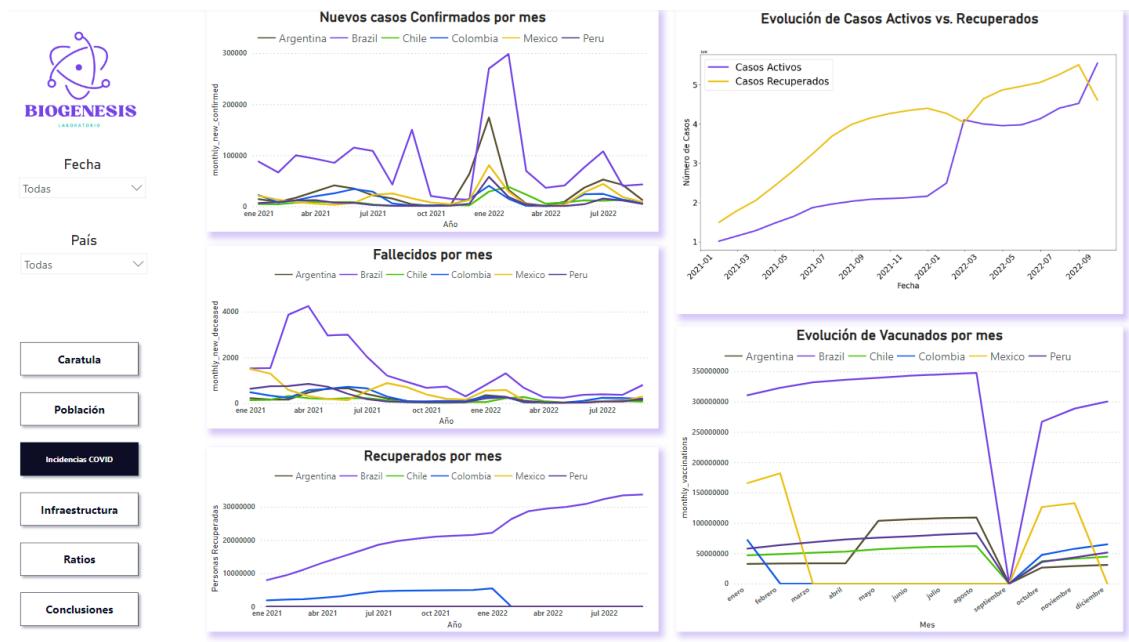
Módulo 4

En esta primera página realizamos un análisis detallado de la situación demográfica y económica de cada país, siendo este uno de los principales factores para determinar donde expandir un laboratorio de Biogenesis.

Observamos que entre los países de Brasil y México representan casi el 70% de la población entre todos los países seleccionados para este estudio. Por esta razón, también ambos países son los que más vacunas administraron (compararon), más del 60% del total de lo que compraron todos los países del estudio.

Así mismo destaca, que entre Brasil y México tienen el 70% del PBI de los países en estudio, lo cual es muy importante para la capacidad económica que puedan tener los países para comprar las vacunas.

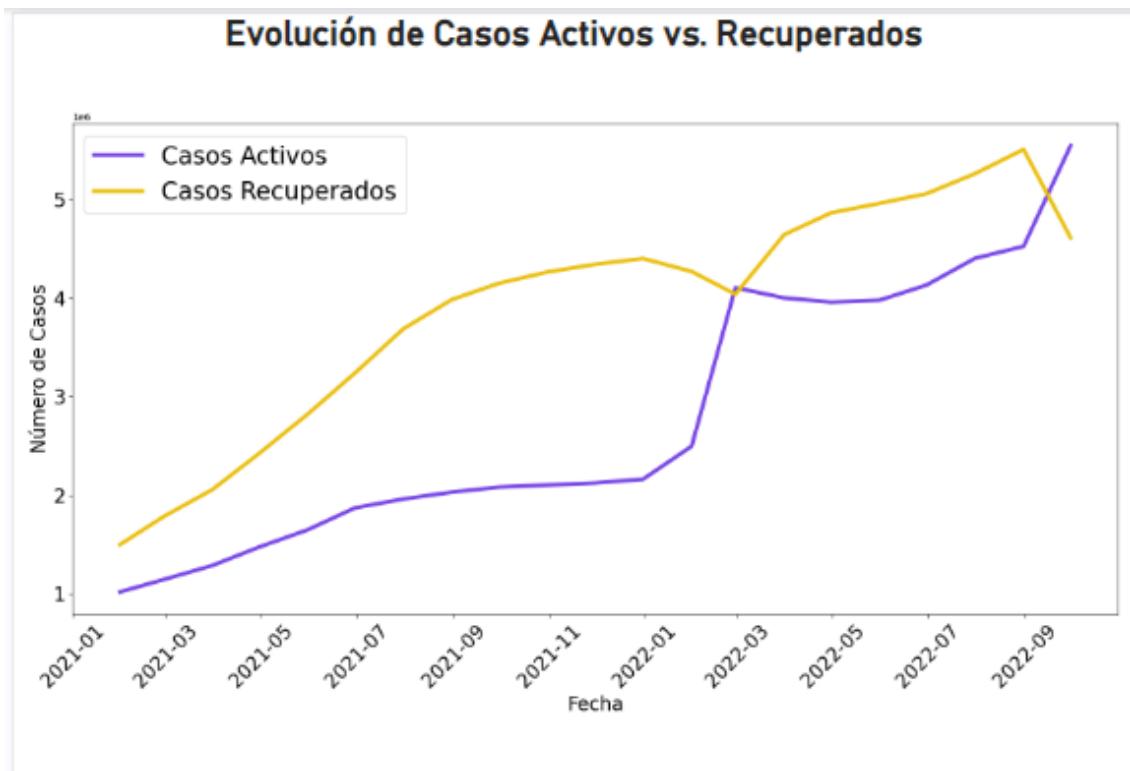
Módulo 4



En esta página de Incidencias COVID analizamos como se comportaron los países con respecto a las 4 variables principales del estudio con respecto al COVID-19, como son los Nuevos Casos Confirmados, los Fallecidos, los Recuperados y los Vacunados.

Llama poderosamente la atención la cantidad de fallecidos que se dieron al inicio en Perú, superando ampliamente a los países de la región.

También observamos que igual tenemos muchos datos faltantes para el estudio en algunas de las variables como por ejemplo en el caso de las personas recuperadas por país, solo se cuenta con los datos completos para Brasil, datos parciales para Colombia; y para el resto de países ese dato esta en cero, lo que no permite poder usar este dato para sacar conclusiones de como afronto cada país la situación de la pandemia. Así mismo en el gráfico de la Evolución de vacunados por mes se observa claramente que no existen datos en un mes.



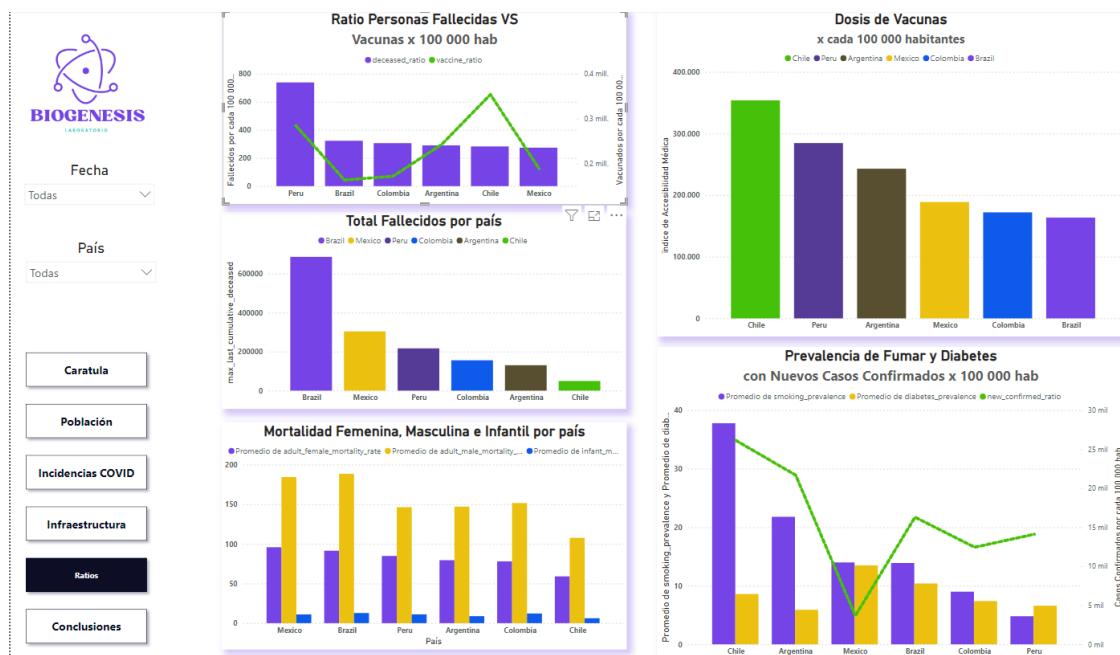
Este gráfico de la Evolución de Casos Activos vs Recuperados se importo mediante un script de Python desde el análisis previo que se realizo en Visual Studio Code mediante Python. Con esto podemos ver la flexibilidad que tiene la herramienta de Power BI, para insertar gráficos que se pueden realizar bajo código Python para desarrollar gráficos que podría ser muy difícil o no pueda ser posible realizar en Power BI. Cabe destacar que este tipo de gráficos pierden la interactividad que puedan tener el resto de gráficos y se pueda analizar por ejemplo los datos por país o un periodo específico de tiempo.

Módulo 4



En esta tercera página de Infraestructura, queremos ver como puede afectar la capacidad instalada de un país para enfrentar una pandemia como se dio el COVID-19.

Cabe resaltar que en los datos del análisis solo contamos con los ratios de cantidad de enfermeras y médicos por cada 1000 personas en los países. No se cuenta por ejemplo con un dato muy importante para este análisis como la cantidad de camas hospitalarias por cada 1000 personas, o de centros de salud u hospitales en cada país. Pero igual con esta información de Enfermeras y Médicas, se creo un índice de Disponibilidad con el que se puede inferir la respuesta en infraestructura médica de cada país. En este sentido se observa que Chile, Brasil y Argentina son los países que tienen los mayores índices de Disponibilidad medica para enfrentar los problemas de salud.

Módulo 4

En esta última página damos mayor énfasis a la importancia que tienen las vacunas en una situación similar de pandemia.

Observamos que hay países como Chile fue el país que en términos comparativos compro más vacunas por habitante. Aunque al ser el país más pequeño en términos de población esto hace que pierda valor para un posible establecimiento de un laboratorio. Muy por el contrario, Brasil, que es el país con mayor población de los estudiados, es el que menos vacunas compro en la proporción por persona; solo 163 mil por cada 100 mil personas VS Chile que compró 352 mil por cada 100 mil personas, lo que amplia una gran oportunidad de que puedan obtener mucho más vacunas en una eventual situación similar. Algo similar también pasa con México y Colombia, que tienen mucha población, pero pocas vacunas compradas por persona.

Conclusiones y Recomendaciones

Estos son los hallazgos claves encontrados en el análisis:

Tasa de Vacunación por País:

- **Insight:** Los países con tasas de vacunación más bajas indican una mayor necesidad de centros de vacunación y, por lo tanto, pueden beneficiarse más de la apertura de nuevos laboratorios.
- **Hallazgo:** Entre los países analizados, Brasil y Colombia muestran tasas de vacunación relativamente más bajas en comparación con Chile, Perú y Argentina. Esto sugiere que abrir laboratorios en Brasil y Colombia podría tener un mayor impacto positivo en aumentar la cobertura de vacunación.

Incidencia de COVID-19 por País:

- **Insight:** Los países con una alta incidencia de COVID-19 necesitan más recursos para controlar la pandemia.
- **Hallazgo:** Brasil y México tienen una alta incidencia de COVID-19. En términos absolutos son los países con mayor población y por lo tanto pueden beneficiarse de tener laboratorios adicionales para producir vacunas y distribuirlas de manera más eficiente.

Disponibilidad de Infraestructuras Médicas por País:

- **Insight:** La infraestructura médica existente es crucial para la distribución efectiva de las vacunas.
- **Hallazgo:** Chile, Brasil y Argentina tienen una mejor infraestructura médica en términos de enfermeras y médicos por cada 1000 habitantes. Sin embargo, países con menor infraestructura como Perú y Colombia podrían necesitar más apoyo en términos de nuevos laboratorios y centros de vacunación.

Relación entre Infraestructura Médica y Tasa de Vacunación:

Módulo 4

- **Insight:** Una infraestructura médica robusta puede correlacionarse con una tasa de vacunación más alta.
- **Hallazgo:** Hay una correlación positiva entre el número de enfermeras/médicos por 1000 habitantes y la tasa de vacunación en países como Chile y Argentina.
- Cabe resaltar que Brasil, a pesar de tener una infraestructura positiva, su tasa de vacunación es relativamente baja por la gran cantidad de población que tiene comparada con otros países de mucho menor población como puede ser Chile o Perú.
- Esto sugiere que mejorar la infraestructura médica en países con bajas tasas de vacunación podría ayudar a aumentar la cobertura de vacunación.

Tendencia de Casos Activos y Recuperados:

- **Insight:** La tendencia de casos activos y recuperados proporciona una visión de la evolución de la pandemia en cada país.
- **Hallazgo:** México y Brasil continúan mostrando altas cifras de casos activos, lo que indica una demanda continua de vacunas y tratamientos. Abrir laboratorios en estos países podría ser crucial para mantener el suministro de vacunas y controlar la pandemia.

Recomendaciones para la Expansión de Laboratorios

1. Brasil y México:

- **Motivo:** Alta incidencia de COVID-19 y necesidad continua de vacunas.
Estos países se beneficiarían de la proximidad de laboratorios para asegurar un suministro constante de vacunas y mejorar la logística de distribución. Además, que entre los dos países representan el 70% de la población y el 60% del PBI de todos los países analizados.

Como una segunda alternativa seria:

2. Perú y Colombia:

- **Motivo:** En países con la necesidad de mejorar la infraestructura médica y con bajas tasas de vacunación como es el caso de Colombia; la apertura de laboratorios en estos países podría aumentar significativamente la cobertura de vacunación y fortalecer la respuesta a la pandemia. Además, considerar que por la ubicación estratégica de estos países (en el centro de todos los países analizados), facilitaría mucho la distribución de las vacunas a toda la región.

Reflexión personal

Durante el desarrollo del proyecto de Expansión Estratégica de Laboratorios para Investigación y Vacunación de COVID-19, he enriquecido y fortalecido una serie de competencias vitales para mi función como Analista de Datos. Este proyecto me ha permitido comprender profundamente el proceso integral de análisis de datos, desde la recopilación hasta la visualización de percepciones que apoyan la toma de decisiones estratégicas para las empresas.

Competencias Adquiridas:

- **Recopilación y Selección de Datos:**

He aprendido a identificar y seleccionar fuentes de datos pertinentes, garantizando que los datos recolectados sean precisos, coherentes y actuales.

- **Limpieza y Transformación de Datos:**

La importancia de la calidad de los datos se ha vuelto evidente. Adquirí habilidades avanzadas en la depuración de datos, manejo de valores ausentes, eliminación de duplicados y estandarización de formatos.

Implementación de técnicas de ETL (Extracción, Transformación y Carga) para optimizar la eficacia del análisis.

- **Análisis Exploratorio de Datos (EDA):**

Desarrollo de capacidades para realizar análisis exploratorio de datos, utilizando estadísticas descriptivas y visualizaciones para identificar tendencias y patrones.

- **Visualización de Datos:**

Creación de dashboards interactivos en Power BI, que permiten explorar los datos desde múltiples perspectivas y apoyar la toma de decisiones informadas.

Utilización de diversas técnicas de visualización, como gráficos de barras, mapas coropléticos y gráficos de líneas.

Módulo 4

- **Modelado y Medidas DAX:**

Aplicación de fórmulas DAX en Power BI para crear medidas que permitan un análisis detallado y preciso.

Reflexión sobre el Proceso y Mejoras Futuras

Si tuvieras que volver a empezar este proyecto, ¿lo harías de la misma manera?
¿Cambiarías algo?

Lo que Haría de la Misma Manera

- **Enfoque en la Calidad de los Datos:**

Continuaría priorizando la limpieza y transformación de datos para asegurar que el análisis se base en información precisa y confiable.

- **Metodología de Análisis:**

Mantendría el enfoque estructurado en el análisis exploratorio de datos y la creación de medidas específicas para obtener percepciones claras y accionables.

- **Visualización Interactiva:**

Seguiría utilizando Power BI para la creación de dashboards interactivos, ya que son herramientas poderosas para comunicar hallazgos y apoyar la toma de decisiones.

Lo que Cambiaría

- **Automatización del Proceso ETL:**

Implementaría un flujo de trabajo más automatizado para la extracción, transformación y carga de datos, utilizando scripts de automatización en Python o en SQL para reducir el tiempo y el esfuerzo manual.

Módulo 4

La automatización permitiría actualizaciones más rápidas y frecuentes de los datos, asegurando que el análisis esté siempre basado en la información más reciente.

- **Ampliación de Fuentes de Datos:**

Buscaría integrar más fuentes de datos para enriquecer el análisis. Por ejemplo, datos sobre la movilidad de la población o indicadores de infraestructura médica que podrían influir en la propagación del COVID-19 y la efectividad de la vacunación.

- **Validación y Verificación de Datos:**

Implementaría procedimientos adicionales de validación y verificación de datos para asegurar aún más la precisión de los datos utilizados en el análisis.

En resumen, este proyecto ha sido una experiencia enriquecedora que ha mejorado mis competencias como Analista de Datos. La recopilación, limpieza, transformación y visualización de datos son componentes esenciales para cualquier análisis eficaz. Si tuviera que empezar el proyecto nuevamente, integraría procesos de automatización y validación más robustos y buscaría ampliar la diversidad de las fuentes de datos para obtener una visión aún más completa. Estos ajustes no solo mejorarían la eficiencia del análisis, sino que también fortalecerían la calidad y profundidad de las conclusiones obtenidas.