TECNOLÓGICO SUPERIOR DE PURÍSIMA DEL RINCÓN

Nombres del Alumno (a):

Alva Zavala Gabriela – RS21110105

Botello Santos Rufino – RS21110220

Márquez Robles Emmanuel – RS21110007

Medina Martínez Oscar Eduardo - RS21110201

Carrera:

Ing. Informática

Grado y grupo:

7-A

Materia:

Inteligencia de Negocios

Nombre del docente:

Valentín Calzada Ledesma

Proyecto 1er Parcial.

Purísima del Rincón, Gto. A 25 de septiembre de 2024

1. Introducción

El área de la informática está enfocada en muchas ramas, desde la programación de algoritmos básicos y complejos, las redes de telecomunicación y también una rama dedicada al análisis de datos. Aunque los analistas de datos comenzaron a realizar esta actividad cientos de años atrás, en la época de 1950 los analistas comenzaron a utilizar las computadoras para automatizar algunos procesos. Y aunque el uso del análisis de datos se dio gracias a la contabilidad mesopotámica, las computadoras permitieron realizar análisis estadísticos y cálculos más complejos o largos. En general, el análisis de datos es usado actualmente con el mismo propósito que hace cientos de años, visualizar las tendencias de los datos y llegar a una conclusión que favorezca a algún hito de la sociedad. La Inteligencia de Negocios es el conjunto de herramientas y estrategias para analizar los datos que son generados por negocios o por las mismas personas, por lo general siempre está enfocada al marketing, a las finanzas y las operaciones empresariales.

La materia de Inteligencia de Negocios abarca diversos temas que van desde la introducción a Business Intelligence (BI) hasta el uso de herramientas informáticas y lenguajes de programación para analizar datos, exponiendo los hallazgos a través de tablas, gráficos o reportes. Específicamente estos temas son: Introducción a Business Intelligence, introducción al análisis de datos con Python, introducción a Pandas, introducción a Jupyter Notebook, extracción de datos de archivos CSV, análisis de datos con la librería Pandas, reporte de resultados en Jupyter Notebook, uso de librerías como MatPlotLib y graficación de resultados.

El proyecto tiene como propósito realizar un análisis de las series de tiempo de COVID-19 a nivel nacional provenientes de la página del gobierno federal del CONAHCYT. Esto con el propósito de identificar tendencias en los datos almacenados por cada estado, además de realizar operaciones estadísticas para evaluar como es que se distribuyen y comportan los datos. Por último, la información solicitada deberá mostrarse en tablas y gráficas para su interpretación.

2. Objetivo general y específicos

Objetivo General:

- Analizar el conjunto de datos de COVID-19 en la república mexicana.

Objetivos Específicos:

- Aplicar herramientas y lenguajes de programación para el análisis.
- Extraer datos de archivos CSV con Pandas.
- Usar DataFrames y Series para el análisis.
- Filtrar y dar formato a los datos.
- Realizar operaciones estadísticas con los datos.
- Visualizar los resultados utilizando MatPlotLib para graficar.
- Analizar los datos y dar una conclusión en base a los datos obtenidos.

3. Marco teórico

3.1 Inteligencia de Negocios.

Business Intelligence (BI) o inteligencia de negocios combina análisis de negocios, minería de datos, visualización de datos, herramientas e infraestructura de datos, y las prácticas recomendadas para ayudar a las organizaciones a tomar decisiones más basadas en los datos. Surgió originalmente en la década de 1960 como un sistema de intercambio de información entre organizaciones. Se desarrolló aún más en la década de 1980 junto con modelos informáticos para la toma de decisiones y convertir los datos en información (Tableau, 2024). Según la organización, la inteligencia de negocios se utiliza para la toma de decisiones en las empresas en base a los datos que generaron durante cierto lapso de tiempo. También sobre cómo es que la BI ha ido evolucionando y adaptándose a las herramientas informáticas.

3.2 Análisis de datos.

El análisis de datos convierte datos sin procesar en información práctica. Incluye una serie de herramientas, tecnologías y procesos para encontrar tendencias y resolver problemas mediante datos. Los análisis de datos pueden dar forma a

procesos empresariales, mejorar la toma de decisiones e impulsar el crecimiento empresarial (Amazon Web Services, 2023). Según la empresa, el análisis de datos puede enforcarse a cualquier hito empresarial o social, esta misma define que herramientas y procesos puede utilizar para realizar el análisis y la toma de decisiones.

3.3 Python.

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes (Amazon Web Services, s.f.). Según la empresa, Python es un lenguaje de programación multifuncional, pero que principalmente se utiliza para la ciencia de datos y el machine learning.

3.4 Pandas.

La biblioteca de software de código abierto Pandas está diseñada específicamente para la manipulación y el análisis de datos en el lenguaje Python. Es potente, flexible y fácil de usar. Gracias a Pandas, se puede utilizar el lenguaje Python para cargar, alinear, manipular o incluso fusionar datos. El rendimiento es realmente impresionante cuando el código fuente del back-end está escrito en C o Python (López, 2022). Según el autor, Pandas es una librería desarrollada con C y Python, la cual le permite a este último el filtrado y control de datos para realizar análisis.

3.5 MatPlotLib.

Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python. Matplotlib hace que las cosas fáciles sean fáciles y las difíciles, posibles. Matplotlib es muy flexible y personalizable para crear gráficos. Requiere mucho código para crear gráficos más básicos con pocas personalizaciones.

3.6 DataMart.

Un data mart es una forma sencilla de un almacén de datos que se centra en un único tema o línea de negocio, como ventas, finanzas o marketing. Dado su enfoque, los data marts obtienen datos de menos orígenes que los almacenes de datos. Las fuentes de datos de un data mart pueden incluir sistemas operativos internos, un almacén de datos central y datos externos.

3.7 Jupyter Notebook.

Jupyter Notebook es una de las herramientas más populares en la ciencia de datos. Con su interfaz interactiva, permite a los usuarios crear y compartir documentos que contienen código, visualizaciones y textos explicativos en un solo lugar. Permite realizar tareas desde limpieza de datos hasta modelización y visualización de resultados. Jupyter Notebook es una herramienta versátil que puede utilizarse en cada paso del proceso de análisis de datos (Fernández, 2024). Según el autor, Jupyter Notebook es una herramienta versátil que se utiliza para el análisis de datos, ya que puede ejecutar códigos por bloques, muestra los resultados y te permite agregar anotaciones.

4. Metodología

Se importan las librerías pandas, matplotlib y numpy. Con la finalidad de realizar el análisis de los datos, realizar operaciones matemáticas y finalmente graficar los resultados.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Se carga el archivo CSV de los casos de COVID-19 por estado de la república mexicana.

```
df = pd.read_csv("Casos_Diarios_Estado_Nacional_Confirmados_20230625.csv")
df
```

Se eliminan las columnas innecesarias como ‘cve_ent’, ‘poblacion’ y ‘nombre’ ya que no registran datos importantes para el análisis. También se eliminan los primeros 15 días ya que existe un retraso según el CONAHCYT.

```
df_sin_nac = df.iloc[:-1]
df_fechasCompl = df_sin_nac.drop(columns=["cve_ent","poblacion","nombre"])
df_fechas = df_fechasCompl.iloc[:, :-15]
df_fechas
```

Pico máximo de casos a nivel nacional

Se calcula el día del pico máximo de casos, sumando el total de casos de cada columna y devolviendo el mayor número de casos con la fecha en que ocurrió.

```
suma_total_casos_por_dia = df_fechas.sum()
pico_maximo = suma_total_casos_por_dia.max()
fecha_pico_maximo = suma_total_casos_por_dia.idxmax()

print(f'El dia de pico máximo fue el {fecha_pico_maximo} con un total de {pico_maximo} casos ')
[3]

El dia de pico máximo fue el 17-01-2022 con un total de 81302 casos
```

Porcentaje de casos acumulados en función de la población a nivel nacional

Se calcula el porcentaje de casos acumulados de COVID-19 dividiendo el total de casos sumados en el DataFrame entre la población total, multiplicando por 100. Luego se imprime el porcentaje resultante, junto con el número total de casos y la población total utilizada para el cálculo.

```
casos_acumulados = df_fechas.sum().sum()
poblacion_total = df_sin_nac["poblacion"].sum()
porcentaje = (casos_acumulados*100)/poblacion_total
print(f'El porcentaje de casos acumulados de COVID-19 es de {casos_acumulados} que equivale a un {porcentaje}%
 según una población de {poblacion_total} personas.')
[52]

El porcentaje de casos acumulados de COVID-19 es de 7628589 que equivale a un 5.969522291822841% según una ↵
 población de 127792286 personas.
```

Fechas de los picos máximos de cada uno de los estados

Se crea un DataFrame con el nombre de cada estado, la fecha de su pico máximo de casos y el número de casos en ese pico.

```
fecha_pico_maximo_estados = df_fechas.idxmax(axis=1)
pico_maximo_estados = df_fechas.max(axis=1)
df_pico_estados = pd.DataFrame({
    "Estado": df_sin_nac["nombre"],
    "Fecha de pico máximo": fecha_pico_maximo_estados,
    "Número de casos": pico_maximo_estados
})
df_pico_estados
```

Estado con más casos acumulados

Se identifica el estado con más casos acumulados sumando los casos por estado y mostrando el nombre del estado con el mayor total.

```
suma_total_casos_estado = df_fechas.sum(axis=1)
id_estado_mas_casos = suma_total_casos_estado.idxmax()
estado_mas_casos = df.loc[id_estado_mas_casos, "nombre"]

print(f'El estado con más casos acumulados: {estado_mas_casos}')

[6]

El estado con más casos acumulados: DISTRITO FEDERAL
```

Estado con porcentaje más alto de casos acumulados en función de la población del estado

Se calcula el porcentaje de casos acumulados en relación con la población de cada estado, identificando el estado con el porcentaje más alto de casos respecto a su población, y mostrando su nombre junto al total de casos acumulados.

```
casos_acumulados_estado = df_fechas.sum(axis=1)
porcentaje_casos_acumulados_estado = (casos_acumulados_estado*100)/(df_sin_nac["poblacion"])
id_estado_porcentaje_mas_alto = porcentaje_casos_acumulados_estado.idxmax()
estado = df.loc[id_estado_porcentaje_mas_alto, "nombre"]

print(f'{estado} es el estado con porcentaje más alto de casos acumulados
 ({casos_acumulados_estado[id_estado_porcentaje_mas_alto]}) según su población')
[53]

 DISTRITO FEDERAL es el estado con porcentaje más alto de casos acumulados (1901128) según su población
```

Estado con menos casos acumulados

Se identifica el estado con el menor número de casos acumulados y se imprime este.

```
id_estado_menos_casos = casos_acumulados_estado.idxmin()
estado_menos_casos = df.loc[id_estado_menos_casos, "nombre"]
print(f'{estado_menos_casos} es el estado con menos casos acumulados')
[8]

 CAMPECHE es el estado con menos casos acumulados
```

Estado con porcentaje más bajo de casos acumulados en función de la población del estado

Se identifica el estado con el porcentaje más bajo de casos acumulados en relación con su población y se imprime su nombre junto con el número total de casos acumulados.

```
id_estado_porcentaje_menos_alto = porcentaje_casos_acumulados_estado.idxmin()
estado2 = df.loc[id_estado_porcentaje_menos_alto, "nombre"]

print(f'{estado2} es el estado con porcentaje más bajo de casos acumulados
 ({casos_acumulados_estado[id_estado_porcentaje_menos_alto]}) según su población')
[54]

 CHIAPAS es el estado con porcentaje más bajo de casos acumulados (59836) según su población
```

Media de casos por estado

Se crea un DataFrame que muestra la media de casos por estado. Primero, se calcula el promedio de casos para cada estado, luego se genera un nuevo DataFrame con el nombre del estado y su media de casos.

```
media_por_estado = df_fechas.mean(axis=1)
df_media_estado = pd.DataFrame({
    "Estado": df_sin_nac["nombre"],
    "Media": media_por_estado
})
df_media_estado
[10]
```

Estados con mediana similar

Se calcula la mediana de casos por estado y luego se determina un rango de valores similares basado en la mediana global y la desviación estándar de las medianas de los estados.

```
mediana_por_estado = df_fechas.median(axis=1)
df_medianas_estados = pd.DataFrame(
    {
        "Estado": df_sin_nac["nombre"],
        "Mediana": mediana_por_estado
    }
)


mediana_global = mediana_por_estado.median()
desv_medianas = mediana_por_estado.std()

rango_abajo = mediana_global - (desv_medianas * 0.5)
rango_arriba = mediana_global + (desv_medianas * 0.5)

mediana_similar = df_medianas_estados[(df_medianas_estados["Mediana"] >= rango_abajo) &
 (df_medianas_estados["Mediana"] <= rango_arriba)]

mediana_similar
```

Exportar datamarts a CSV

Se crean datamarts de los datos de los picos máximos por estado, la media por estado y la mediana similar en más menos a CSV, los cuales pueden servirnos más adelante para un análisis más detallado.

```
df_pico_estados.to_csv("fechas_pico_maximo_estados.csv", index=False)
df_media_estado.to_csv("media_de_casos_por_estado.csv", index=False)
mediana_similar.to_csv("estados_con_mediana_similar", index =False)
[12]
```

Gráfica de casos acumulados diarios a nivel nacional

Se suman los casos diarios de COVID-19 y se identifica la fecha con el mayor total. Luego, se grafica el total acumulado, destacando el punto máximo con un marcador rojo en la gráfica.

```
casos_acumulados_diarios = df_fechas.sum()
fecha_mas_casos = casos_acumulados_diarios.idxmax()
cantidad_casos = casos_acumulados_diarios.max()
casos_acumulados_diarios.index = pd.to_datetime(casos_acumulados_diarios.index, dayfirst=True)

casos_acumulados_diarios.plot(label = 'Casos')
fecha_mas_casos = pd.to_datetime(fecha_mas_casos, dayfirst=True)

plt.plot(fecha_mas_casos, cantidad_casos, 'ro', label=f'Punto máximo: {fecha_mas_casos.date()}')

plt.title('Casos diarios de COVID-19 en México ')
plt.xlabel('Fecha')
plt.ylabel('Número de Casos')
plt.gcf().set_size_inches(12, 8)

plt.legend()
plt.show()
```

Dashboards de Boxplot por Estado

Se crean boxplots de casos diarios de COVID-19 para los estados, dividiendo los estados en 8 grupos. Cada grupo se grafica mostrando la distribución de casos por estado, y luego se muestran todos los gráficos en el dashbord.

```
df_states_cases = df_sin_nac.drop(columns=["cve_ent","poblacion"])
df_box = df_states_cases.iloc[:, :-15]

df_conjunto = df_box.melt(id_vars=['nombre'],
                          var_name='Fecha',
                          value_name = 'Casos')

states = df_conjunto['nombre'].unique()

state_groups = np.array_split(states, 8)

plt.figure(figsize=(7, 65))
```

```
for i, group in enumerate(state_groups):
    data = [df_conjunto[df_conjunto['nombre'] == state]['Casos'].dropna() for state in group]

    plt.subplot(8, 1, i + 1)
    plt.boxplot(data, tick_labels=group)
    plt.title(f'Boxplot de casos diarios para el grupo {i + 1}')
    plt.xlabel('Estado')
    plt.ylabel('Casos diarios')
    plt.xticks(rotation=45)

    max_cases = np.max([np.max(d) for d in data if len(d) > 0])
    plt.ylim(0, max_cases * 1.05)

plt.tight_layout()

plt.show()
```

5. Reporte de resultados

Pico máximo de casos a nivel nacional

Según el análisis realizado, el día con el pico máximo de casos se dio el 17 de enero de 2022 con 81302 casos registrados.

```
El dia de pico máximo fue el 17-01-2022 con un total de 81302 casos
```

Porcentaje de casos acumulados en función de la población a nivel nacional

Como podemos observar, el total de personas en la república mexicana es de 127,792,286 que equivale al 100%, el porcentaje arrojado fue de 5.96% que equivale a 7,628,589 personas.

```
El porcentaje de casos acumulados de COVID-19 es de 7628589 que equivale a un 5.9695222918228 41% según una
población de 127792286 personas.
```

Fechas de los picos máximos de cada uno de los estados

Tras aplicar el filtro de casos máximos hemos podido determinar que el mayor número registrado está en el mes de enero de 2022 en todos los estados.

32 rows | 32 rows × 3 columns

	Estado	Fecha de pico máximo	Número de casos
0	AGUASCALIENTES	17-01-2022	1522
1	BAJA CALIFORNIA	12-01-2022	3065
2	BAJA CALIFORNIA SUR	10-01-2022	2172
3	CAMPECHE	25-01-2022	412
4	CHIAPAS	20-01-2022	830
5	CHIHUAHUA	17-01-2022	2443
6	DISTRITO FEDERAL	12-01-2022	17807
7	COAHUILA	14-01-2022	2313
8	COLIMA	24-01-2022	974
9	DURANGO	10-01-2022	1084
10	GUANAJUATO	13-01-2022	3270
11	GUERRERO	17-01-2022	1103
12	HIDALGO	17-01-2022	1251
13	JALISCO	17-01-2022	4419
14	MEXICO	17-01-2022	8040
15	MICHOACAN	17-01-2022	983
16	MORELOS	13-01-2022	760
17	NAYARIT	17-01-2022	1263
18	NUEVO LEON	17-01-2022	6496
19	OAXACA	24-01-2022	1507

Estado con más casos acumulados

El estado con más casos acumulados es el Distrito Federal o CDMX.

```
El estado con más casos acumulados: DISTRITO FEDERAL
```

Estado con porcentaje más alto de casos acumulados en función de la población del estado

El estado con el porcentaje más alto en base a su población es el Distrito Federal o CDMX. El estado tiene una población de 9,018,645 personas que representa el 100%, mientras que los casos registrados son de 1,901,128 que equivale a un 21% de la población total.

```
DISTRITO FEDERAL es el estado con porcentaje más alto de casos acumulados (1901128) según su población
```

Estado con menos casos acumulados

El estado con menos casos es Campeche, en este estado hay 1,000,617 habitantes y el caso más alto fue de 412. Esto puede deberse a que se respetaron las medidas de salubridad o que, a pesar de ser un lugar turístico, las personas que presentaron los síntomas quizá pertenecían a lugares rurales.

```
CAMPECHE es el estado con menos casos acumulados
```

Estado con porcentaje más bajo de casos acumulados en función de la población del estado.

El estado con menos casos en base a su población es Chiapas, en este estado hay 5,730,000 habitantes y el caso más alto fue de 830. Aunque tiene su cifra de caso más alto que Campeche, también puede deberse a que en este estado hay más comunidades indígenas y rurales, y quizá no hubo muchas instituciones de salud que aplicaran las pruebas.

```
CHIAPAS es el estado con porcentaje más bajo de casos acumulados (59836) según su población
```

Media de casos por estado

Como podemos observar, la mayoría de los estados están entre los 100 y 180 casos. Tenemos estados por debajo de 100 y otros que exceden los 200 e incluso los 1000.

32 rows × 2 columns

	Estado	Media
0	AGUASCALIENTES	76.188333
1	BAJA CALIFORNIA	149.905000
2	BAJA CALIFORNIA SUR	109.084167
3	CAMPECHE	37.082500
4	CHIAPAS	49.863333
5	CHIHUAHUA	145.627500
6	DISTRITO FEDERAL	1584.273333
7	COAHUILA	158.240000
8	COLIMA	60.272500
9	DURANGO	68.463333
10	GUANAJUATO	311.267500
11	GUERRERO	100.791667
12	HIDALGO	108.702500
13	JALISCO	249.296667
14	MEXICO	633.604167
15	MICHOACAN	98.633333
16	MORELOS	85.025000
17	NAYARIT	64.784167
18	NUEVO LEON	358.573333
19	OAXACA	133.888333
20	PUEBLA	188.440833

Estados con mediana similar

Los casos con una mediana similar (más menos) son aquellos que casi no tienen casos acumulados. Ninguno de ellos supera los 90 casos en mediana, como ya se dijo puede deberse a las medidas de salud o a los casos registrados en instituciones.

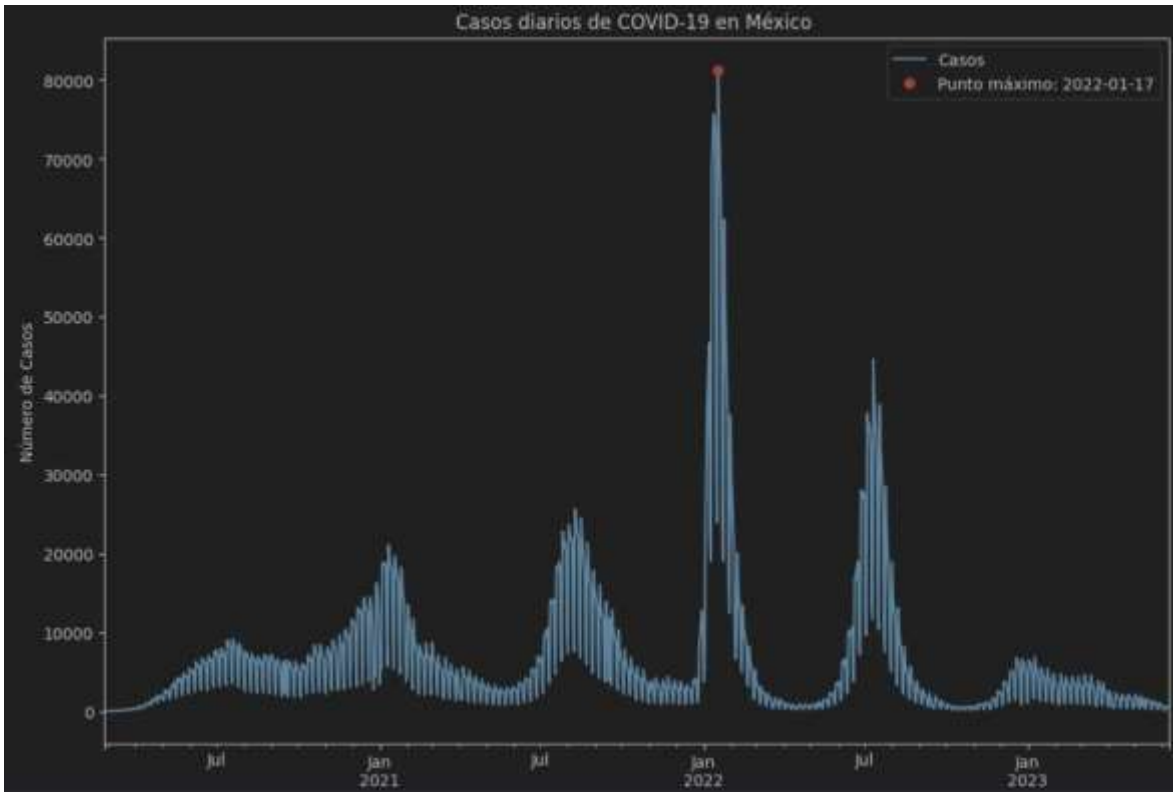
28 rows × 2 columns

	Estado	Mediana
0	AGUASCALIENTES	40.0
1	BAJA CALIFORNIA	82.0
2	BAJA CALIFORNIA SUR	48.0
3	CAMPECHE	13.0
4	CHIAPAS	18.0
5	CHIHUAHUA	66.5
7	COAHUILA	67.5
8	COLIMA	20.0
9	DURANGO	25.0

11	GUERRERO	39.0
12	HIDALGO	62.0
13	JALISCO	111.0
15	MICHOACAN	49.0
16	MORELOS	37.0
17	NAYARIT	22.0
19	OAXACA	70.5
20	PUEBLA	109.0
21	QUERETARO	66.5
22	QUINTANA ROO	41.0
23	SAN LUIS POTOSI	82.0
24	SINALOA	66.0
25	SONORA	97.0

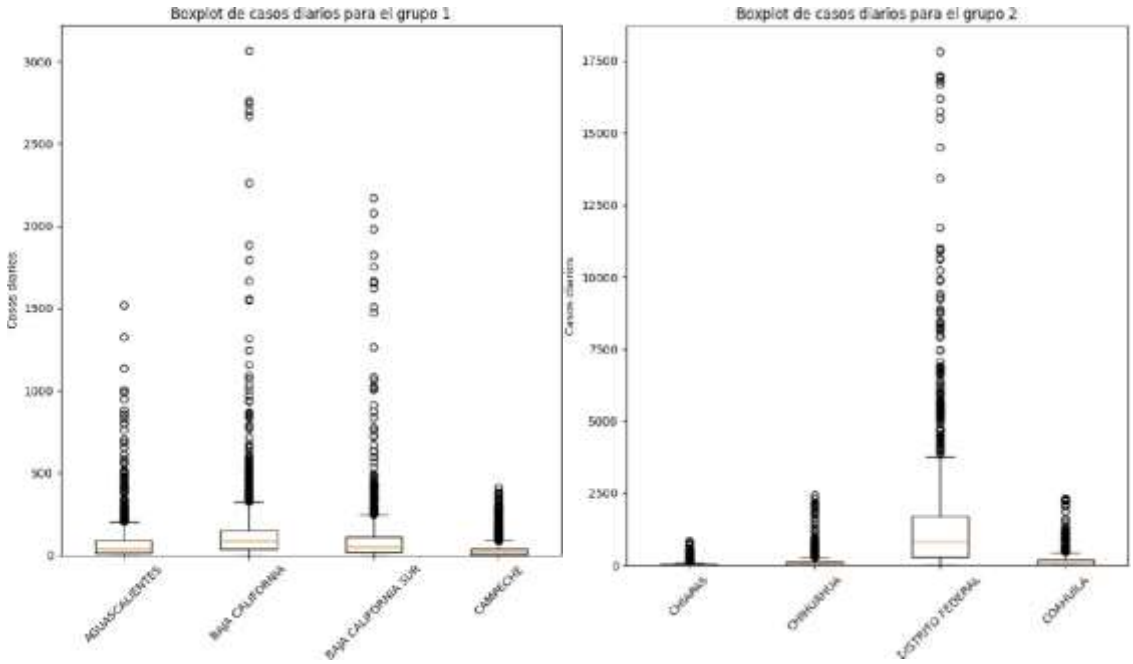
Gráfica de casos acumulados diarios a nivel nacional

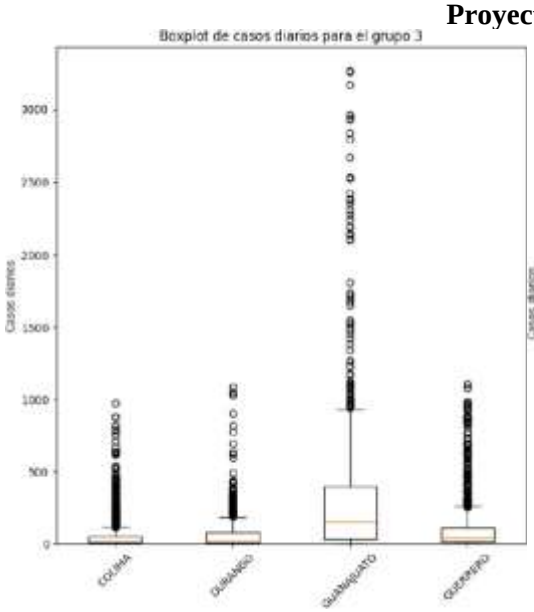
Como podemos ver, el día con más casos se dio el 17 de enero de 2022 con los 81,302 casos como se muestra en el punto rojo de la gráfica. Igual podemos observar picos en enero de 2021 y julio 2022 donde hubo un reapunte COVID-19 quizá debido a las vacaciones de verano.

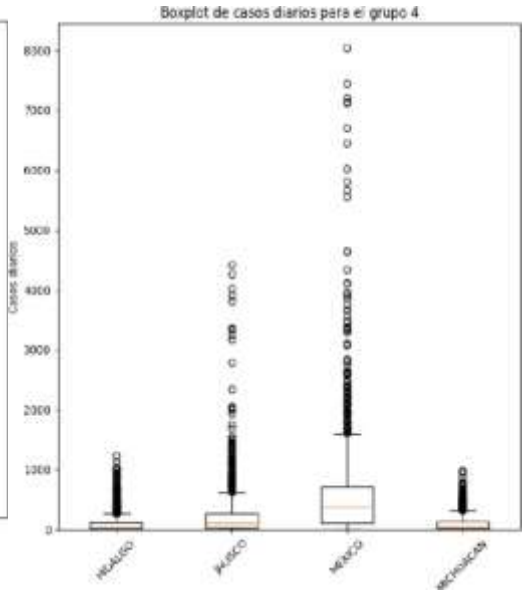


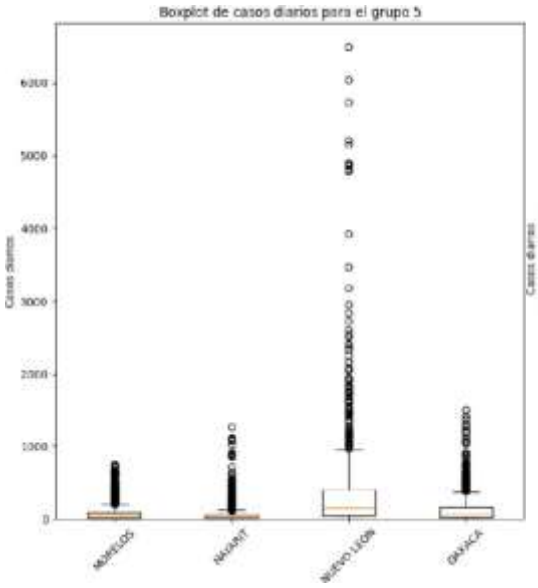
Dashboards de Boxplot por Estado

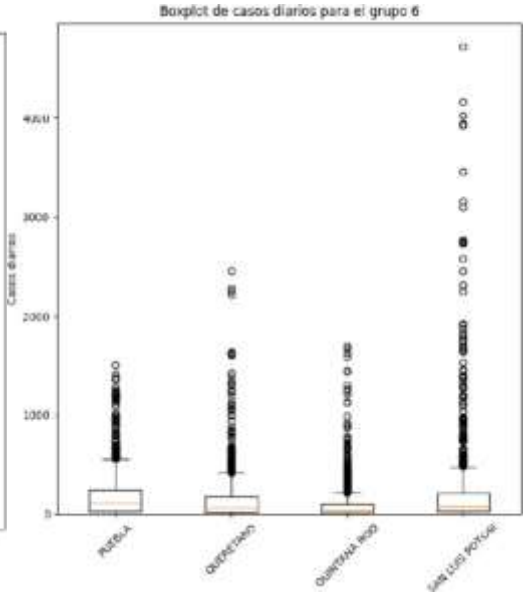
Como podemos ver, en cada estado hay valores atípicos. De las 8 agrupaciones de 4 estados las más notables son de Baja California Norte, CDMX, Guanajuato, Estado de México, Nuevo León, San Luis Potosí, Tabasco, Veracruz y Yucatán, como podemos ver los "bigotes" de las gráficas están muy extendidos, lo que representa mucha variación en los datos.

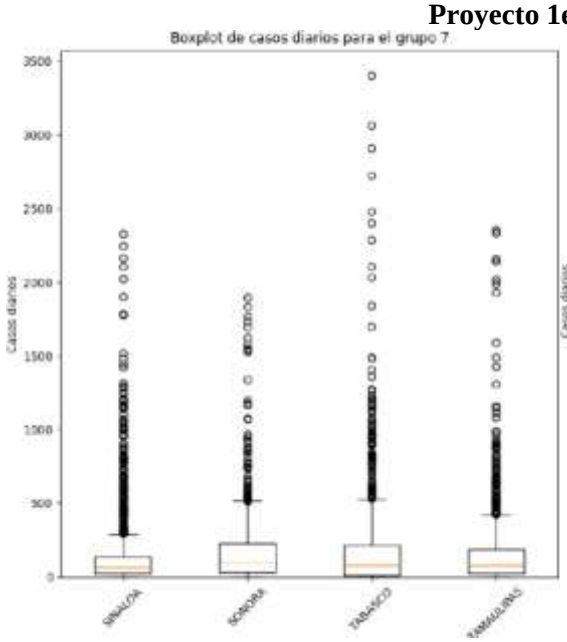


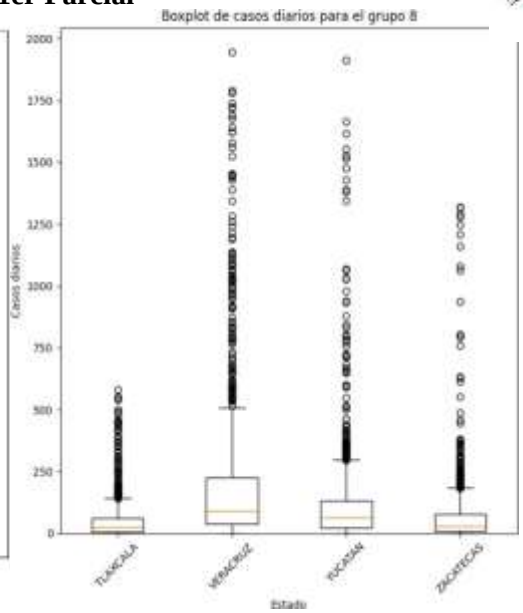

Boxplot de casos diarios para el grupo 3
Casos diarios
COLIMA
DURANGO
GUANAJUATO
GUERRERO


Boxplot de casos diarios para el grupo 4
Casos diarios
HIDALGO
JALISCO
MEXICO
MICHOACAN


Boxplot de casos diarios para el grupo 5
Casos diarios
MORELOS
NAYARIT
NUEVO LEON
OAXACA


Boxplot de casos diarios para el grupo 6
Casos diarios
PUEBLA
QUERETARO
QUINTANA ROO
SAN LUIS POTOSI





6. Análisis de resultados

Como podemos observar, el análisis muestra que la distribución de casos diarios varía significativamente entre los estados, con algunos presentando picos de contagios notables. Además, el gráfico de casos acumulados destaca claramente el día con el mayor número de casos reportados. También se identifican estados con un alto porcentaje de casos acumulados en relación con su población, lo que sugiere áreas de mayor vulnerabilidad. En general, estos resultados reflejan la complejidad y variabilidad de la pandemia a nivel nacional.

7. Conclusión

En conclusión, se comprendió la importancia de la extracción de datos y el análisis de estos para poder generar estimaciones denotar la importancia que tiene la recopilación de datos como en este caso el COVID-19, de este modo saber el impacto que tuvo está pandemia, los picos que llegó a alcanzar y que aún existen fuegos de repuntar los casos de COVID-19.

8. Bibliografía

- **Amazon Web Services. (2023). *Amazon Web Services*. Obtenido de Amazon Web Services: https://aws.amazon.com/es/what-is/data-analytics/**
- **Amazon Web Services. (s.f.). *Amazon Web Services*. Obtenido de Amazon Web Services: https://aws.amazon.com/es/what-is/python/**
- **Fernández, O. (14 de julio de 2024). *Aprender Big Data*. Obtenido de Aprender Big Data: https://aprenderbigdata.com/jupyter-notebook/**
- **López, E. N. (19 de diciembre de 2022). *DataScientist*. Obtenido de DataScientist: https://datascientest.com/es/pandas-python**
- **Tableau. (2024). *Tableau*. Obtenido de Tableau: https://www.tableau.com/es-mx/learn/articles/business-intelligence**
- **DataCamp. (n.d.). Matplotlib tutorial – Introduction to Python plotting. DataCamp. https://www.datacamp.com/tutorial/matplotlib-tutorial-python**
- **Oracle. (n.d.). ¿Qué es un data mart? Oracle. https://www.oracle.com/mx/autonomous-database/what-is-data-mart/**
- **Matplotlib. (n.d.). Matplotlib: Python plotting. Matplotlib. https://matplotlib.org/**