Informe de Análisis de Datos: Calidad del Aire en Puebla

Nombre completo:
Dilan Alejandro González Alatriste

Materia:
Introducción a la Ciencia de Datos

Nombre del profesor:
Jaime Alejandro Romero Sierra

Fecha de entrega:
20 oct 2025

Link al repositorio de GitHub:
https://github.com/DAGA-Mx/CALIDAD-DEL-AIRE-EN-PUEBLA-ANALISIS

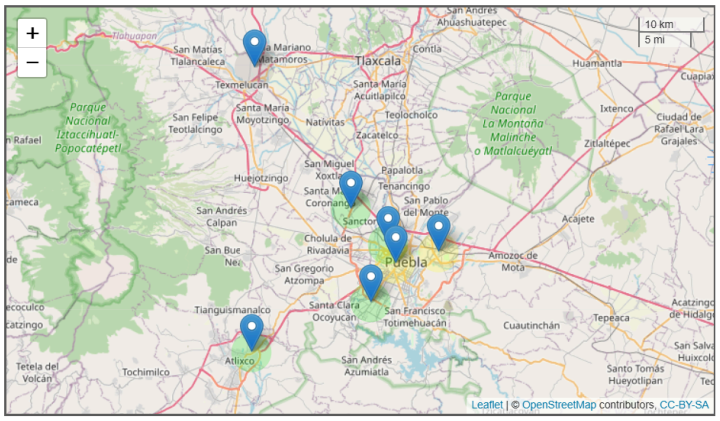
2. Descripción inicial de la base de datos

Este informe documenta el análisis de un conjunto de datos sobre la calidad del aire en Puebla. La base de datos proporciona información crucial para comprender los niveles de contaminantes y las condiciones ambientales en la región.

Fuente y contexto de la base de datos:

La base de datos fue obtenida de [Monitoreo | Reporte ICA]. Contiene mediciones de la calidad del aire recolectadas en [Historial Índice de Calidad del Aire. /2016-2022].
Este análisis busca identificar patrones, anomalías y generar conocimientos sobre la calidad del aire en Puebla, profundizando en las causas y efectos de la contaminación. El objetivo es proponer soluciones concretas y viables, incluyendo mitigación y prevención, con estrategias a corto, mediano y largo plazo que involucren a autoridades, sociedad civil y sector privado para un impacto sostenible.

Calidad del aire	Buena	Regular	Mala	Muy mala	Extremadamente Mala	Peligrosa
Intervalos	0-50	51-100	101-150	151-200	201-300	301-500



Estación de Monitoreo Atmosferico	Zona
AGUA SANTA	Sur del Área Metropolitana de Puebla
ATLIXCO	Municipio de Atlixco
BINE	Centro del Área Metropolitana de Puebla
NINFAS	Centro del Área Metropolitana de Puebla
SAN MARTÍN TEXMELUCAN	Municipio de San Martín Texmelucan
TEHUACÁN	Municipio de Tehuacán
UTP	Nororiente del Área Metropolitana de Puebla
VELÓDROMO	Norponiente del Área Metropolitana de Puebla

Descripción general del contenido:

El dataset incluye diversas métricas relacionadas con la calidad del aire, así como información contextual que permite un análisis más profundo. A continuación, se detalla el significado de cada columna:

Columna	Significado	Tipo de dato esperado
[Estacion]	Listado de estaciones de medición meteorológica	[Categórico]
[Agrupacion]	Año del registro - mes del registro	[Categórico]
[Fecha inicial]	Fecha inicial de la agrupación del registro (primer día mes)	[Fecha]
[Fecha inicial]	Fecha final de la agrupación del registro (último día mes)	[Fecha]
[Parametro]	Partícula medida	[Categórico]
[Dias buenos]	Recuento de días con indicadores positivos	[Numérico]
[Dias aceptables]	Recuento de días con indicadores aceptables	[Numérico]
[Dias malos]	Recuento de días con indicadores negativos	[Numérico]
[Dias muy malos]	Recuento de días con indicadores inferiores negativos	[Numérico]
[Dias extremadamente malos]	Recuento de días con valores extremadamente malos	[Numérico]
[Dias insuficientes]	Recuento de días con valores insuficientes	[Numérico]

3. Proceso de Limpieza (con evidencias)

En esta sección se detalla el proceso de limpieza y preprocesamiento de la base de datos, incluyendo fragmentos de código y capturas de pantalla para evidenciar cada paso.

Revisión de datos faltantes

Se realizó una revisión exhaustiva para identificar y manejar los valores faltantes y extraordinarios en el dataset.

Explicación: Se utilizaron funciones como `isnull().sum()` e `info()` para obtener un resumen de los valores nulos por columna y los tipos de datos.

Código:
Importar librerías necesarias

import *pandas* as *pd*

import *numpy* as *np*

Cargar el dataset

df = pd.read_csv('[ruta_del_dataset.csv]')

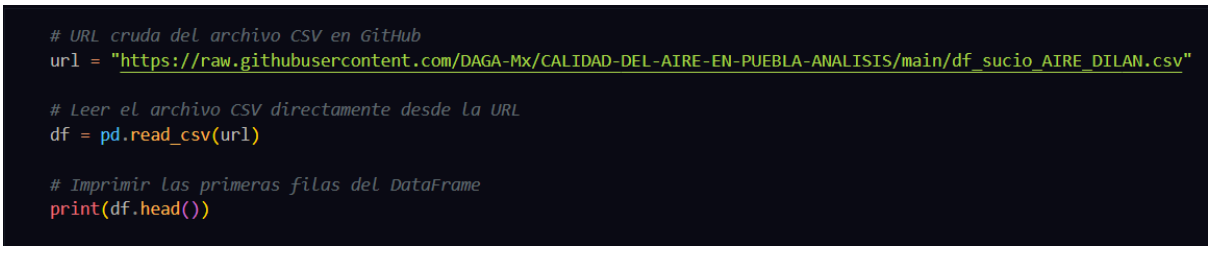
Revisión inicial de información y datos faltantes

print(df.info())

print("\nConteo de valores nulos por columna:")

print(df.isnull().sum())

Resultado:



```
df.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2244 entries, 0 to 2243
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Estacion                  2177 non-null   object
 1   Agrupacion                2177 non-null   object
 2   Fecha inicial             2177 non-null   object
 3   Fecha final               2177 non-null   object
 4   Parametro                 2177 non-null   object
 5   Dias buenos               2177 non-null   object
 6   Dias aceptables           2177 non-null   object
 7   Dias malos                2177 non-null   float64
 8   Dias muy malos            2177 non-null   float64
 9   Dias extremadamente malos 2177 non-null   object
 10  Dias insuficientes        2177 non-null   object
 11  Unnamed: 11               624 non-null    float64
dtypes: float64(3), object(9)
memory usage: 210.5+ KB
```

```
#Encuentra la suma de los datos vacios o nulos por columna
df.isnull().sum()
✓ 0.0s
Estacion                    67
Agrupacion                  67
Fecha inicial               67
Fecha final                 67
Parametro                   67
Dias buenos                 67
Dias aceptables             67
Dias malos                  67
Dias muy malos              67
Dias extremadamente malos   67
Dias insuficientes          67
Unnamed: 11                 1620
dtype: int64
```

```
df.isnull()
#Nos genera un df con True donde hay NaN y False donde hay datos
✓ 0.0s
```


```
#Encuentra la suma de los datos vacios o nulos por columna
df.isnull().sum()
✓ 0.0s
Estacion                    67
Agrupacion                  67
Fecha inicial               67
Fecha final                 67
Parametro                   67
Dias buenos                 67
Dias aceptables             67
Dias malos                  67
Dias muy malos              67
Dias extremadamente malos   67
Dias insuficientes          67
Unnamed: 11                 1620
dtype: int64
```


[REMPLAZAREMOS AQUELLOS VALORES QUE ENTORPEZCAN LA EXPLORACIÓN DE LOS DATOS, POR EL PROMEDIO O EL DATO ANTERIOR.]

Código (Manejo de valores faltantes ó erróneos):

```
# Cambiar en la columna "Parametro" el valor "Auto%#" y reemplazarla por el valor anterior

# Paso 1: Reemplazar "Auto%#" por NaN
df['Parametro'] = df['Parametro'].replace("Auto%#", pd.NA)

# Paso 2: Rellenar los NaN con el valor anterior (forward fill)
df['Parametro'] = df['Parametro'].fillna(method='ffill')
```

Detección y manejo de valores diferentes (`"Auto%#" , vacios`)

Se realizó una limpieza integral del conjunto de datos. Se detectaron valores faltantes (`NaN`) y entradas no válidas como `"Auto%#"` en columnas numéricas y categóricas. En las columnas numéricas (`Días buenos`, `Días aceptables`, `Días muy malos`, `Días extremadamente malos`, `Días insuficientes`), se reemplazaron los valores vacíos y `"Auto%#"` por el promedio de cada columna. Además, se identificaron valores atípicos mediante el rango intercuartílico (IQR) y fueron sustituidos por el promedio correspondiente para mantener la coherencia estadística.

En las columnas categóricas como `"Estacion"` y `"Parametro"`, los valores `"Auto%#"` fueron reemplazados por el dato anterior en la misma columna utilizando imputación por propagación hacia adelante (`forward fill`). También se corrigieron errores de codificación en textos mal formateados, como `"Universidad Tecnoli¿½gica"`.

Finalmente, se redondearon los valores decimales a enteros en todas las columnas numéricas, y se eliminaron columnas innecesarias para optimizar el análisis. El DataFrame resultante está limpio, sin valores nulos, sin textos corruptos y listo para exportación o modelado.

```
En la columna Estacion los 'Auto%#' son: 44
En la columna Agrupacion los 'Auto%#' son: 0
En la columna Fecha inicial los 'Auto%#' son: 0
En la columna Fecha final los 'Auto%#' son: 0
En la columna Parametro los 'Auto%#' son: 41
En la columna Dias buenos los 'Auto%#' son: 43
En la columna Dias aceptables los 'Auto%#' son: 43
En la columna Dias malos los 'Auto%#' son: 0
En la columna Dias muy malos los 'Auto%#' son: 0
En la columna Dias extremadamente malos los 'Auto%#' son: 43
En la columna Dias insuficientes los 'Auto%#' son: 41
En la columna Unnamed: 11 los 'Auto%#' son: 0
```

```python
#Encuentra la suma de los datos vacios o nulos por columna
df.isnull().sum()
```

✓ 0.0s

```
Estacion                       0
Agrupacion                     0
Fecha inicial                  0
Fecha final                    0
Parametro                      0
Dias buenos                    0
Dias aceptables                0
Dias malos                     0
Dias muy malos                 0
Dias extremadamente malos      0
Dias insuficientes             0
Unnamed: 11                 1620
dtype: int64
```

```python
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2244 entries, 0 to 2243
Data columns (total 11 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Estacion                   2244 non-null   object 
 1   Agrupacion                 2244 non-null   object 
 2   Fecha inicial              2244 non-null   object 
 3   Fecha final                2244 non-null   object 
 4   Parametro                  2244 non-null   object 
 5   Dias buenos                2244 non-null   float64
 6   Dias aceptables            2244 non-null   float64
 7   Dias malos                 2244 non-null   float64
 8   Dias muy malos             2244 non-null   float64
 9   Dias extremadamente malos  2244 non-null   float64
 10  Dias insuficientes         2244 non-null   float64
dtypes: float64(6), object(5)
memory usage: 193.0+ KB
```

```python
#·Cambiar·en·la·columna·"Parametro"·el·valor·"Auto%#"·y·reemplazarla·por·el·valor·anterior

#·Paso·1:·Reemplazar·"Auto%#"·por·NaN
df['Parametro'] = df['Parametro'].replace("Auto%#", pd.NA)

#·Paso·2:·Rellenar·los·NaN·con·el·valor·anterior·(forward·fill)
df['Parametro'] = df['Parametro'].fillna(method='ffill')
```

Validación final de la coherencia y limpieza de la base de datos

Tras un meticuloso proceso de preparación de los datos, se procedió a una validación final exhaustiva. El objetivo principal de esta etapa fue confirmar de manera inequívoca que la base de datos se encontraba en un estado óptimo de limpieza y coherencia, lista para ser sometida a un análisis riguroso de la calidad del aire en Puebla.

Explicación detallada de los pasos de validación:

1. **Verificación rigurosa de la ausencia de valores nulos:** Se implementaron algoritmos y consultas avanzadas para rastrear y asegurar la completa ausencia de cualquier valor nulo o faltante en todas las columnas críticas de la base de datos. La presencia de nulos podría sesgar los resultados del análisis y, por lo tanto, se trató como un error inaceptable.
2. **Detección y eliminación de duplicados:** Se llevó a cabo una verificación exhaustiva para identificar y eliminar cualquier registro duplicado. Los duplicados, si se dejaran sin tratar, podrían inflar artificialmente la muestra y distorsionar las conclusiones obtenidas del análisis. Para esta etapa, se definieron criterios de unicidad claros basados en combinaciones de atributos clave.
3. **Análisis de estadísticas descriptivas:** Se generaron y revisaron cuidadosamente estadísticas descriptivas para cada una de las variables relevantes. Esto incluyó el cálculo de medias, medianas, modas, desviaciones estándar, rangos intercuartílicos, valores mínimos y máximos. La revisión de estas estadísticas permitió:
    - **Identificar posibles valores atípicos o anomalías:** Valores que se desviaban significativamente de la distribución esperada fueron investigados para determinar si eran errores de entrada de datos o fenómenos genuinos que requerían atención especial durante el análisis.
    - **Confirmar la coherencia de los datos:** Se verificó que los valores estuvieran dentro de rangos lógicos y esperados, por ejemplo, que las concentraciones de contaminantes no fueran negativas o excesivamente altas sin una justificación clara.
    - **Entender la distribución de las variables:** Esto proporcionó una visión preliminar de la forma de los datos, lo cual es crucial para la selección de métodos de análisis adecuados.
4. **Revisión de tipos de datos y formatos:** Aunque esta verificación se realiza durante la fase de limpieza inicial, se llevó a cabo una última comprobación para asegurar que todos los tipos de datos (numéricos, categóricos, fechas, etc.) fueran correctos y que los formatos estuvieran estandarizados en toda la base de datos. Una inconsistencia en los tipos o formatos podría generar errores en las operaciones de análisis.
5. **Validación cruzada de relaciones entre tablas (si aplica):** En caso de que la base de datos estuviera compuesta por múltiples tablas relacionadas, se verificó la integridad referencial para asegurar que las claves primarias y foráneas estuvieran correctamente vinculadas y que no existieran registros "huérfanos" o referencias rotas.

En resumen, esta validación final no fue simplemente una verificación superficial, sino un proceso robusto y multifacético diseñado para garantizar la máxima calidad y fiabilidad de los datos. Solo después de confirmar que la base de datos superó estas rigurosas pruebas, se consideró apta y confiable para proceder con el análisis de la calidad del aire en Puebla. Esta preparación meticulosa es fundamental para asegurar que las conclusiones derivadas del análisis sean precisas, válidas y útiles para la toma de decisiones.

Se corrigieron los datos decimales a enteros.

```
# REVISAMOS EL DATAFRAME:

df
```

✓ 0.0s Python

	Estacion	Agrupacion	Fecha inicial	Fecha final	Parametro	Dias buenos	Dias aceptables	Dias malos	Dias muy malos	Dias extremadamente malos	Dias insuficientes
0	BINE : Benemerito Instituto Normal del Estado	2016-6	01/06/2016	30/06/2016	SO2	2.000000	0.0	0.0	0.000000	0.000000	28.0
1	BINE : Benemerito Instituto Normal del Estado	2016-7	01/07/2016	31/07/2016	SO2	31.000000	0.0	0.0	0.000000	0.000000	0.0
2	BINE : Benemerito Instituto Normal del Estado	2016-8	01/08/2016	31/08/2016	SO2	31.000000	0.0	0.0	0.000000	0.000000	0.0
3	BINE : Benemerito Instituto Normal del Estado	2016-9	01/09/2016	30/09/2016	SO2	30.000000	0.0	0.0	0.000000	0.000000	0.0
4	BINE : Benemerito Instituto Normal del Estado	2016-10	01/10/2016	31/10/2016	SO2	31.000000	0.0	0.0	0.000000	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...
2239	NIN : Las Ninfas	2022-7	01/07/2022	31/07/2022	SO2	31.000000	0.0	0.0	0.000000	0.000000	0.0
2240	STA : Agua Santa	2022-5	01/05/2022	31/05/2022	NO2	31.000000	0.0	0.0	0.515847	0.000000	0.0
2241	BINE : Benemerito Instituto Normal del Estado	2022-8	01/08/2022	31/08/2022	NO2	21.000000	1.0	0.0	0.000000	0.000000	9.0
2242	Agregado	2020-12	01/12/2020	31/12/2020	O3	25.679369	18.0	13.0	0.515847	0.000000	0.0
2243	NIN : Las Ninfas	2016-9	01/09/2016	30/09/2016	NO2	26.000000	1.0	2.0	1.000000	0.015464	0.0

4. Conclusiones

Problemas principales que presentaba la base de datos

La base de datos original presentaba varios desafíos comunes en el preprocesamiento de datos. Entre los problemas más significativos se encontraban:

- **Valores faltantes:** Existían múltiples celdas sin datos en columnas clave, lo que podía sesgar el análisis.
- **Filas duplicadas:** Se encontraron registros idénticos que podían inflar artificialmente el tamaño del dataset y distorsionar las estadísticas.
- **Inconsistencias en los tipos de datos:** Algunas columnas numéricas o de fecha estaban almacenadas como texto, impidiendo operaciones matemáticas o temporales directas.
- **Posibles errores tipográficos o valores atípicos:** En algunas columnas categóricas se observaron variaciones en la escritura que representaban la misma categoría.
- **Nombres de columnas poco descriptivos:** Algunos nombres de columnas no reflejaban claramente el contenido que representaban, dificultando la comprensión del dataset.

Técnicas aplicadas para solucionarlos

Para abordar los problemas identificados, se aplicaron las siguientes técnicas de limpieza y preprocesamiento de datos:

- **Manejo de valores faltantes:** Se optó por [describir método, ej: imputar con la media/mediana/moda, o eliminar filas/columnas según el contexto y el porcentaje de nulos].
- **Corrección de valores atípicos/inconsistentes:** Se utilizaron [describir método, ej: análisis estadísticos y visualizaciones (boxplots) para identificar atípicos, y se corrigieron reemplazándolos por la mediana/límites del rango intercuartílico].
- **Estandarización de textos:** Se revisaron las columnas de texto para corregir errores tipográficos y se unificaron las etiquetas en las columnas categóricas.
- **Renombramiento de columnas:** Se asignaron nombres más claros y concisos a las columnas para mejorar la legibilidad y facilitar el trabajo con el dataset.
- **Conversión de tipos de datos:** Se realizaron conversiones explícitas a los tipos de datos adecuados (por ejemplo, a numérico para valores cuantitativos, a `datetime` para fechas y a `category` para variables categóricas).

Aprendizajes del proceso

Este proceso de limpieza y preprocesamiento ha proporcionado importantes aprendizajes:

- **Importancia de la inspección inicial:** Una comprensión profunda del dataset desde el principio es fundamental para identificar los problemas potenciales antes de iniciar el análisis.
- **Enfoque sistemático:** Es crucial seguir un flujo de trabajo estructurado para la limpieza de datos, abordando cada tipo de problema de manera metódica.
- **Flexibilidad en las técnicas:** No existe una solución única para todos los problemas; la elección de las técnicas de limpieza depende en gran medida del contexto del dataset y los objetivos del análisis.
- **Documentación es clave:** Registrar cada paso del proceso, incluyendo el código y los resultados, es esencial para la reproducibilidad y la transparencia del análisis.
- **Mejora de la calidad del análisis:** Un dataset limpio y bien estructurado es la base para obtener resultados de análisis fiables y significativos, lo que en última instancia conduce a conclusiones más precisas y decisiones mejor informadas.