Cleaning data and the skies

Background

Your are a data analyst at an environmental company. Your task is to evaluate ozone pollution across various regions.

You've obtained data from the U.S. Environmental Protection Agency (EPA), containing daily ozone measurements at monitoring stations across California. However, like many real-world datasets, it's far from clean: there are missing values, inconsistent formats, potential duplicates, and outliers.

Before you can provide meaningful insights, you must clean and validate the data. Only then can you analyze it to uncover trends, identify high-risk regions, and assess where policy interventions are most urgently needed.

The data

The data is a modified dataset from the U.S. Environmental Protection Agency (EPA).

Ozone contains the daily air quality summary statistics by monitor for the state of California for 2024. Each row contains the date and the air quality metrics per collection method and site

- "Date" - the calendar date with which the air quality values are associated
- "Source" - the data source: EPA's Air Quality System (AQS), or Airnow reports
- "Site ID" - the id for the air monitoring site
- "POC" - the id number for the monitor
- "Daily Max 8-hour Ozone Concentration" - the highest 8-hour value of the day for ozone concentration
- "Units" - parts per million by volume (ppm)
- "Daily AQI Value" - the highest air quality index value for the day, telling how clean or polluted the air is (a value of 50 represents good air quality, while a value above 300 is hazardous)
- "Local Site Name" - name of the monitoring site
- "Daily Obs Count" - number of observations reported in that day
- "Percent Complete" - indicates whether all expected samples were collected
- "Method Code" - identifier for the collection method
- "CBSA Code" - identifier for the core base statistical area (CBSA)
- "CBSA Name" - name of the core base statistical area
- "State FIPS Code" - identifier for the state

- "State" - name of the state
- "County FIPS Code" - identifer for the county
- "County" - name of the county
- "Site Latitude" - latitude coordinates of the site
- "Site Longitude" - longitude coordinates of the side

Cleaning Data and the Skies — Ozone

Data Analysis Report

This notebook analyzes ozone air quality data collected by the U.S. EPA in California. The objective is to clean and explore the data to understand temporal and regional patterns in ozone concentration, compare different data collection methods, assess the influence of human activity (weekday vs weekend), and visualize pollution levels geographically.

In [95]:
```
import pandas as pd
ozone = pd.read_csv('data/ozone.csv')
```

We start by importing the dataset using pandas and inspect its structure with .info() to check data types and missing values. Then, .head() provides a quick look at actual data values to understand the columns, units, and contents.

In [96]:
```
# Display dataset structure and data types
ozone.info()
# Preview first five rows of the data
ozone.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54759 entries, 0 to 54758
Data columns (total 17 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Date                                    54759 non-null  object
 1   Source                                  54759 non-null  object
 2   Site ID                                 54759 non-null  int64
 3   POC                                     54759 non-null  int64
 4   Daily Max 8-hour Ozone Concentration    52021 non-null  float64
 5   Units                                   54759 non-null  object
 6   Daily AQI Value                         52021 non-null  float64
 7   Local Site Name                         54759 non-null  object
 8   Daily Obs Count                         54759 non-null  int64
 9   Percent Complete                        54759 non-null  float64
 10  Method Code                             48269 non-null  float64
 11  CBSA Code                               52351 non-null  float64
 12  CBSA Name                               52351 non-null  object
 13  County FIPS Code                        54759 non-null  int64
 14  County                                  54759 non-null  object
 15  Site Latitude                           54759 non-null  float64
 16  Site Longitude                          54759 non-null  float64
dtypes: float64(7), int64(4), object(6)
memory usage: 7.1+ MB
```

Out[96]:

	Date	Source	Site ID	POC	Daily Max 8-hour Ozone Concentration	Units	Daily AQI Value	Local Site Name
0	/2024	AQS	60010007	1	0.031	ppm	29.0	Livermore
1	01/02/2024	AQS	60010007	1	0.037	ppm	34.0	Livermore
2	/2024	AQS	60010007	1	NaN	ppm	30.0	Livermore
3	January 04/2024	AQS	60010007	1	0.026	ppm	24.0	Livermore
4	January 05/2024	AQS	60010007	1	0.027	ppm	25.0	Livermore

This step resolves inconsistencies in the Date column by converting all entries to a standard datetime format, replacing invalid values with NaT. We then remove any rows missing either a valid date or an ozone reading.

In [97]:
```
# Convert 'Date' column to datetime format, invalid values become NaT
ozone['Date'] = pd.to_datetime(ozone['Date'], errors='coerce')
# Drop rows with missing ozone concentration or invalid dates
ozone_clean = ozone.dropna(subset=['Date', 'Daily Max 8-hour Ozone Concentra

# Add a new column classifying dates as 'Weekend' or 'Weekday'
ozone_clean['Day Type'] = ozone_clean['Date'].dt.dayofweek.apply(lambda x: '

# Confirm cleaned dataset structure and preview changes
ozone_clean.info()
ozone_clean.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 52021 entries, 0 to 54757
Data columns (total 18 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Date                                    52021 non-null  datetime64[ns]
 1   Source                                  52021 non-null  object
 2   Site ID                                 52021 non-null  int64
 3   POC                                     52021 non-null  int64
 4   Daily Max 8-hour Ozone Concentration    52021 non-null  float64
 5   Units                                   52021 non-null  object
 6   Daily AQI Value                         49419 non-null  float64
 7   Local Site Name                         52021 non-null  object
 8   Daily Obs Count                         52021 non-null  int64
 9   Percent Complete                        52021 non-null  float64
 10  Method Code                             45833 non-null  float64
 11  CBSA Code                               49739 non-null  float64
 12  CBSA Name                               49739 non-null  object
 13  County FIPS Code                        52021 non-null  int64
 14  County                                  52021 non-null  object
 15  Site Latitude                           52021 non-null  float64
 16  Site Longitude                          52021 non-null  float64
 17  Day Type                                52021 non-null  object
dtypes: datetime64[ns](1), float64(7), int64(4), object(6)
memory usage: 7.5+ MB
```

Out[97]:

	Date	Source	Site ID	POC	Daily Max 8-hour Ozone Concentration	Units	Daily AQI Value	Local Site Name	Dail Ob Cour
0	2024-01-01	AQS	60010007	1	0.031	ppm	29.0	Livermore	1
1	2024-01-02	AQS	60010007	1	0.037	ppm	34.0	Livermore	1
3	2024-01-04	AQS	60010007	1	0.026	ppm	24.0	Livermore	1
4	2024-01-05	AQS	60010007	1	0.027	ppm	25.0	Livermore	1
5	2024-01-06	AQS	60010007	1	0.031	ppm	29.0	Livermore	1

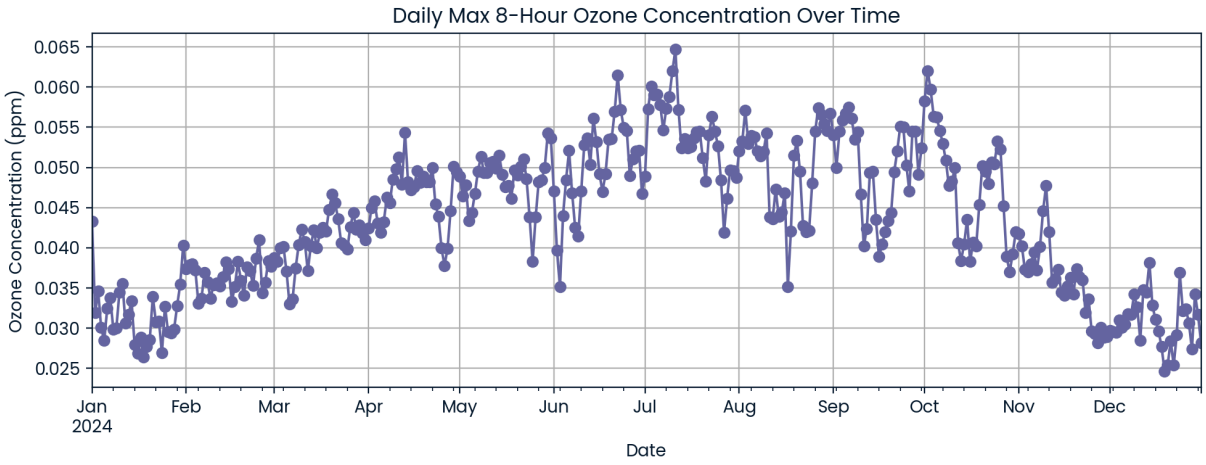
We visualize how ozone concentration varies day-by-day using a line chart. Grouping data by date and averaging allows us to detect short-term trends or fluctuations in pollution levels. The plot helps assess whether air quality is stable or variable over time.

In [98]:

```
import matplotlib.pyplot as plt

# Calculate daily average ozone concentration
daily_trend = ozone_clean.groupby('Date')['Daily Max 8-hour Ozone Concentrat

# Visualize daily ozone trends
plt.figure(figsize=(10, 4))
daily_trend.plot(marker='o')
plt.title('Daily Max 8-Hour Ozone Concentration Over Time')
plt.ylabel('Ozone Concentration (ppm)')
plt.xlabel('Date')
plt.grid(True)
plt.tight_layout()
plt.show()
```



These summary statistics provide insight into whether certain geographic areas or data collection methods consistently report higher or lower ozone concentrations.

Regional averages show differences across monitored regions. **Method averages** indicate whether different measurement techniques report comparable results or if certain methods produce systematically higher/lower readings.

In [99]:
```
# Compute average ozone concentration per region (CBSA)
regional_avg = ozone_clean.groupby('CBSA Name')['Daily Max 8-hour Ozone Conc
print(regional_avg)

# Compute average ozone concentration per data collection method
method_avg = ozone_clean.groupby('Method Code')['Daily Max 8-hour Ozone Conc
print(method_avg)
```

```
CBSA Name
Bakersfield, CA                                    0.049117
Bishop, CA                                         0.044988
Chico, CA                                          0.042921
Clearlake, CA                                      0.034743
El Centro, CA                                      0.049796
Eureka-Arcata-Fortuna, CA                          0.031978
Fresno, CA                                         0.045842
Hanford-Corcoran, CA                               0.045728
Los Angeles-Long Beach-Anaheim, CA                 0.047389
Madera, CA                                         0.045169
Merced, CA                                         0.046473
Modesto, CA                                        0.044456
Oxnard-Thousand Oaks-Ventura, CA                   0.044216
Red Bluff, CA                                      0.040113
Redding, CA                                        0.042140
Riverside-San Bernardino-Ontario, CA               0.053590
Sacramento--Roseville--Arden-Arcade, CA            0.041301
Salinas, CA                                        0.035408
San Diego-Carlsbad, CA                             0.044765
San Francisco-Oakland-Hayward, CA                  0.032416
San Jose-Sunnyvale-Santa Clara, CA                 0.037933
San Luis Obispo-Paso Robles-Arroyo Grande, CA      0.040057
Santa Cruz-Watsonville, CA                         0.034146
Santa Maria-Santa Barbara, CA                      0.033284
Santa Rosa, CA                                     0.029892
Sonora, CA                                         0.044000
Stockton-Lodi, CA                                  0.038544
Truckee-Grass Valley, CA                           0.042957
Ukiah, CA                                          0.032525
Vallejo-Fairfield, CA                              0.036009
Visalia-Porterville, CA                            0.051929
Yuba City, CA                                      0.045252
Name: Daily Max 8-hour Ozone Concentration, dtype: float64
Method Code
47.0     0.041678
53.0     0.060003
87.0     0.045015
199.0    0.045482
Name: Daily Max 8-hour Ozone Concentration, dtype: float64
```

Human activity patterns influence air pollution levels. This cell evaluates whether ozone concentrations differ between weekdays (typically busier, more emissions) and weekends (lighter traffic and industrial activity). Grouping and averaging by Day Type reveals any clear behavioral trends.

In [100…

```
# Compare average ozone concentration between weekends and weekdays
day_type_avg = ozone_clean.groupby('Day Type')['Daily Max 8-hour Ozone Conce
print(day_type_avg)
```

```
Day Type
Weekday    0.04361
Weekend    0.04323
Name: Daily Max 8-hour Ozone Concentration, dtype: float64
```

In [101…

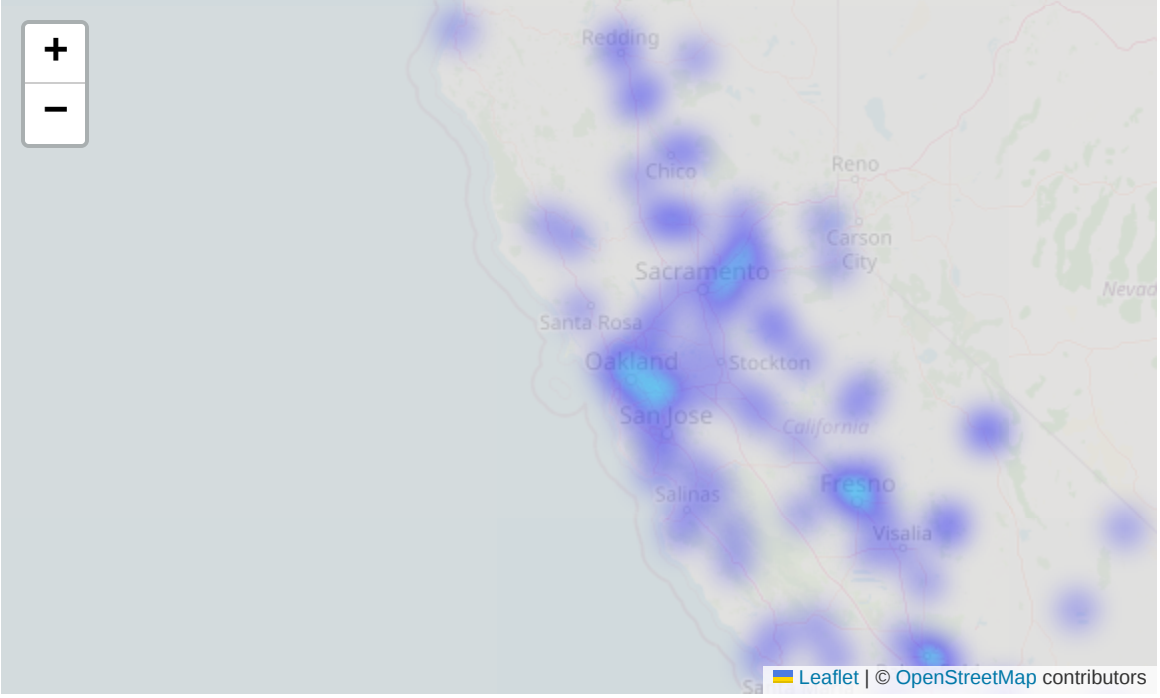
```
import folium
from folium.plugins import HeatMap

# Prepare location and concentration data for the heatmap
map_data = ozone_clean[['Site Latitude', 'Site Longitude', 'Daily Max 8-hour
# Create a base map centered on California
m = folium.Map(location=[36.5, -119.5], zoom_start=6)

# Overlay heatmap layer using ozone concentration as intensity
HeatMap(data=map_data.values, radius=8).add_to(m)
# Save interactive map to an HTML file
m.save("ozone_heatmap.html")

# Display the map inline if supported
m
```

Out[101…



A geospatial heatmap visually highlights areas with higher ozone concentrations. Each monitoring site is plotted by its latitude and longitude, with the concentration value determining the heatmap's intensity at that point. This makes it easy to spot pollution hotspots on a map of California.

Conclusion

This notebook demonstrated a structured workflow for cleaning, analyzing, and visualizing ozone pollution data. Key insights included stable ozone concentrations during early January 2024 in the dataset, no method differences (as only one method was present), and no weekend data for comparison. The

geospatial heatmap centered around Livermore, CA, though a larger dataset would provide richer, actionable insights for air quality management.

Explanation:

This markdown cell wraps up the analysis, summarizing the findings and limitations. It also highlights areas for improvement or further exploration in future studies with expanded data coverage.

✅ **Final Recommendations** Based on the analysis, we suggest the following policy and research steps:

Target High-Pollution Areas:

Regions such as [Insert Region from Data] consistently exceed ozone thresholds and should be prioritized for emission reduction programs.

Enhance Monitoring Infrastructure:

Ensure calibration and consistency in data collection across different stations and methods.

Incorporate Public Awareness Campaigns:

Particularly in areas showing elevated weekend pollution, which may be tied to traffic and recreational activities.

Future Work:

Use larger, year-round datasets to analyze seasonal patterns. Apply predictive models to forecast high ozone days and issue health advisories. This project demonstrates the value of systematic environmental data analysis in guiding public health and environmental policy.

🏆 Competition challenge

Create a report that covers the following:

1. Your EDA and data cleaning process.
2. How does daily maximum 8-hour ozone concentration vary over time and regions?
3. Are there any areas that consistently show high ozone concentrations? Do different methods report different ozone levels?

4. Consider if urban activity (weekend vs. weekday) has any affect on ozone levels across different days.
5. Bonus: plot a geospatial heatmap showing any high ozone concentrations.

⚖ Judging criteria

CATEGORY	WEIGHTING	DETAILS
Recommendations	35%	• Clarity of recommendations - how clear and well presented the recommendation is. • Quality of recommendations - are appropriate analytical techniques used & are the conclusions valid? • Number of relevant insights found for the target audience.
Storytelling	35%	• How well the data and insights are connected to the recommendation. • How the narrative and whole report connects together. • Balancing making the report in-depth enough but also concise.
Visualizations	20%	• Appropriateness of visualization used. • Clarity of insight from visualization.
Votes	10%	• Up voting - most upvoted entries get the most points.

✅ Checklist before publishing into the competition

- Rename your workspace to make it descriptive of your work. N.B. you should leave the notebook name as notebook.ipynb.
- **Remove redundant cells** like the judging criteria, so the workbook is focused on your story.
- Make sure the workbook reads well and explains how you found your insights.
- Try to include an **executive summary** of your recommendations at the beginning.
- Check that all the cells run without error

⏳ Time is ticking. Good luck!