```
from google.colab import files
uploaded = files.upload()
```

Choose files final_dataset.csv

final_dataset.csv(text/csv) - 78368 bytes, last modified: 30/01/2026 - 100% done

```
Saving final_dataset.csv to final_dataset (1).csv
```

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

# Loading the CSV file
df = pd.read_csv('final_dataset.csv')

# View the first 5 rows to confirm it loaded correctly
df.head()
```

	Date	Month	Year	Holidays_Count	Days	PM2.5	PM10	NO2	SO2	CO	Ozone	AQI
0	1	1	2021	0	5	408.80	442.42	160.61	12.95	2.77	43.19	462
1	2	1	2021	0	6	404.04	561.95	52.85	5.18	2.60	16.43	482
2	3	1	2021	1	7	225.07	239.04	170.95	10.93	1.40	44.29	263
3	4	1	2021	0	1	89.55	132.08	153.98	10.42	1.01	49.19	207
4	5	1	2021	0	2	54.06	55.54	122.66	9.70	0.64	48.88	149

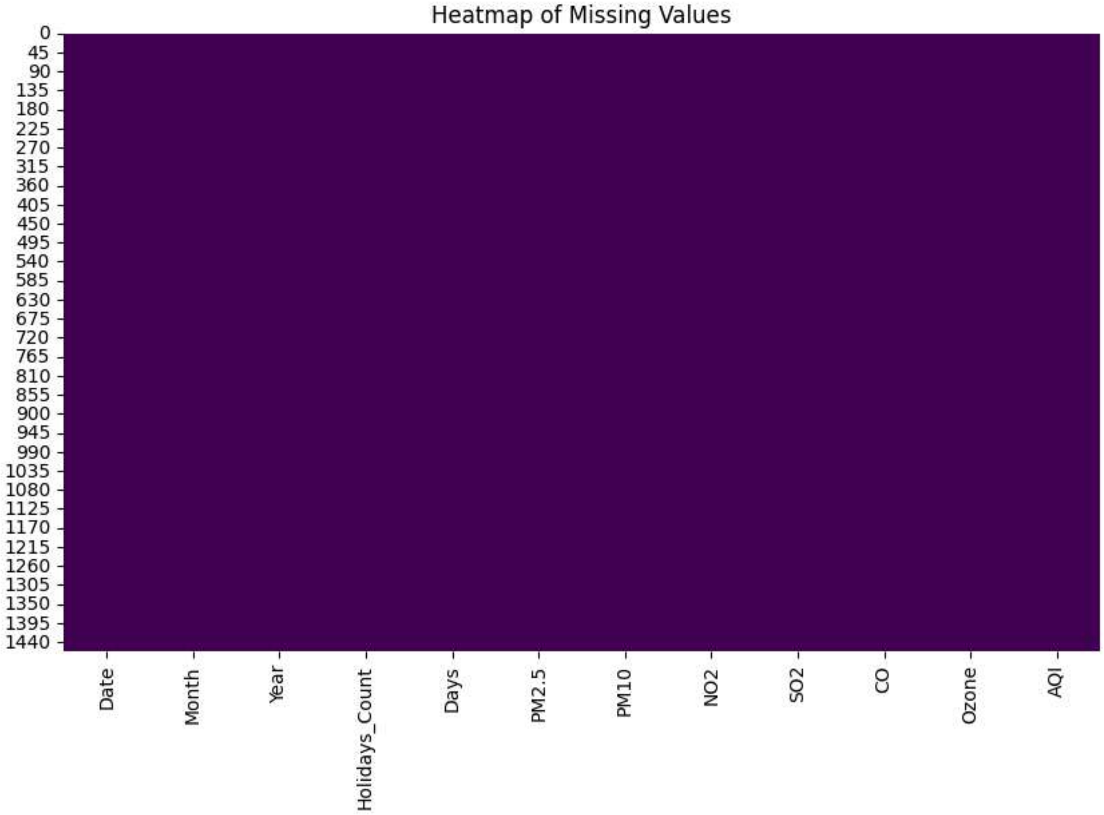
Next steps: Generate code with df New interactive sheet

```
# Check total rows and columns
print(f"Dataset Shape: {df.shape}")

# See column names and data types (int, float, object)
df.info()
```

```
Dataset Shape: (1461, 12)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Date            1461 non-null   int64  
 1   Month           1461 non-null   int64  
 2   Year            1461 non-null   int64  
 3   Holidays_Count  1461 non-null   int64  
 4   Days            1461 non-null   int64  
 5   PM2.5           1461 non-null   float64
 6   PM10            1461 non-null   float64
 7   NO2             1461 non-null   float64
 8   SO2             1461 non-null   float64
 9   CO              1461 non-null   float64
 10  Ozone           1461 non-null   float64
 11  AQI             1461 non-null   int64  
dtypes: float64(6), int64(6)
memory usage: 137.1 KB
```

```
# Display total null values for each column
null_counts = df.isnull().sum()
print("Null Values per Column:")
print(null_counts)

# Calculate percentage of missing values (useful for beginners)
print("\nPercentage of Missing Values:")
print((df.isnull().sum() / len(df)) * 100)

# Optional: Visualize missing data with a heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.title("Heatmap of Missing Values")
plt.show()
```

```
Null Values per Column:
Date              0
Month             0
Year              0
Holidays_Count    0
Days              0
PM2.5             0
PM10              0
NO2               0
SO2               0
CO                0
Ozone             0
AQI               0
dtype: int64

Percentage of Missing Values:
Date              0.0
Month             0.0
Year              0.0
Holidays_Count    0.0
Days              0.0
PM2.5             0.0
PM10              0.0
NO2               0.0
SO2               0.0
CO                0.0
Ozone             0.0
AQI               0.0
dtype: float64
```



```
# Statistical summary of numerical columns
df.describe()
```

	Date	Month	Year	Holidays_Count	Days	PM2.5	PM10	NO2	SO2	
count	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.
mean	15.729637	6.522930	2022.501027	0.189596	4.000684	90.774538	218.219261	37.184921	20.104921	1.
std	8.803105	3.449884	1.118723	0.392116	2.001883	71.650579	129.297734	35.225327	16.543659	0.
min	1.000000	1.000000	2021.000000	0.000000	1.000000	0.050000	9.690000	2.160000	1.210000	0.
25%	8.000000	4.000000	2022.000000	0.000000	2.000000	41.280000	115.110000	17.280000	7.710000	0.
50%	16.000000	7.000000	2023.000000	0.000000	4.000000	72.060000	199.800000	30.490000	15.430000	0.
75%	23.000000	10.000000	2024.000000	0.000000	6.000000	118.500000	297.750000	45.010000	26.620000	1.
max	31.000000	12.000000	2024.000000	1.000000	7.000000	1000.000000	1000.000000	433.980000	113.400000	4.

```
# 1. Structural Overview
print("--- Dataset Information ---")
print(df.info())  # Shows non-null counts and data types (int, float, object)

print("\n--- Descriptive Statistics ---")
# Provides Mean, Median (50%), Std Dev, Min, and Max for numerical columns
display(df.describe())

# 2. Null Value Distribution
print("\n--- Missing Value Report ---")
null_counts = df.isnull().sum()
null_percent = (df.isnull().sum() / len(df)) * 100
missing_report = pd.concat([null_counts, null_percent], axis=1, keys=['Total Nulls', '% Missing'])
display(missing_report[missing_report['Total Nulls'] > 0].sort_values(by='% Missing', ascending=False))
```

```
--- Dataset Information ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            1461 non-null   int64
 1   Month           1461 non-null   int64
 2   Year            1461 non-null   int64
 3   Holidays_Count  1461 non-null   int64
 4   Days            1461 non-null   int64
 5   PM2.5           1461 non-null   float64
 6   PM10            1461 non-null   float64
 7   NO2             1461 non-null   float64
 8   SO2             1461 non-null   float64
 9   CO              1461 non-null   float64
 10  Ozone           1461 non-null   float64
 11  AQI             1461 non-null   int64
dtypes: float64(6), int64(6)
memory usage: 137.1 KB
None

--- Descriptive Statistics ---
```

	Date	Month	Year	Holidays_Count	Days	PM2.5	PM10	NO2	SO2	
count	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000	1461.
mean	15.729637	6.522930	2022.501027	0.189596	4.000684	90.774538	218.219261	37.184921	20.104921	1.
std	8.803105	3.449884	1.118723	0.392116	2.001883	71.650579	129.297734	35.225327	16.543659	0.
min	1.000000	1.000000	2021.000000	0.000000	1.000000	0.050000	9.690000	2.160000	1.210000	0.
25%	8.000000	4.000000	2022.000000	0.000000	2.000000	41.280000	115.110000	17.280000	7.710000	0.
50%	16.000000	7.000000	2023.000000	0.000000	4.000000	72.060000	199.800000	30.490000	15.430000	0.
75%	23.000000	10.000000	2024.000000	0.000000	6.000000	118.500000	297.750000	45.010000	26.620000	1.
max	31.000000	12.000000	2024.000000	1.000000	7.000000	1000.000000	1000.000000	433.980000	113.400000	4.

```
--- Missing Value Report ---
```

Total Nulls	% Missing

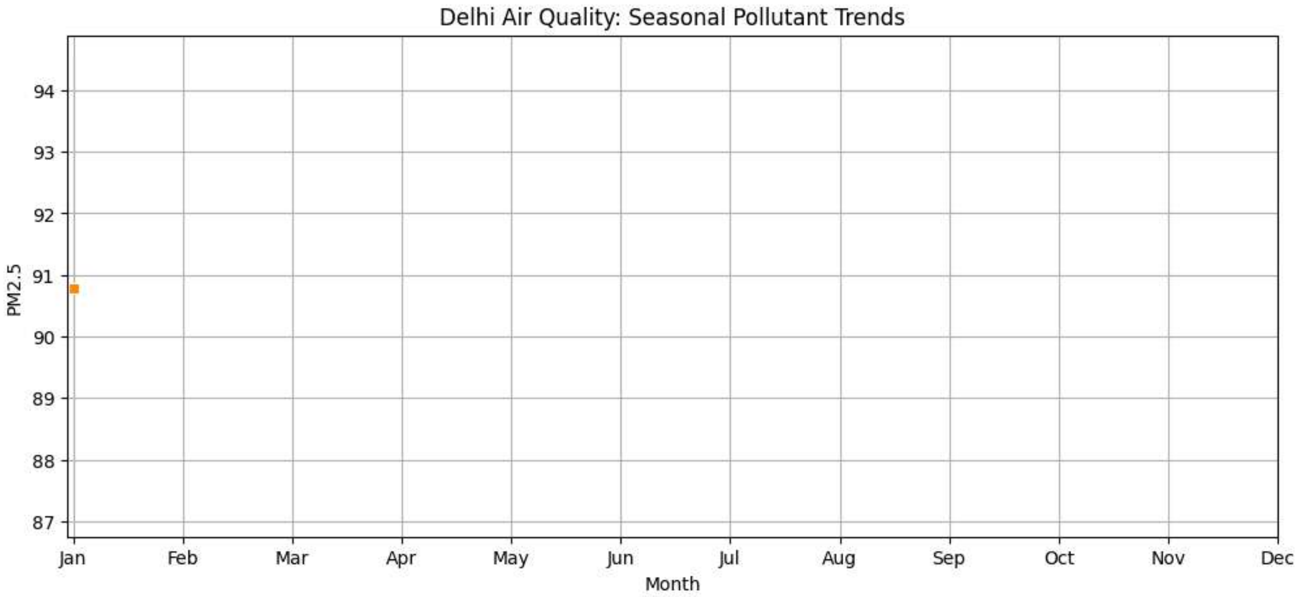
```
# 1. Formatting Dates
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

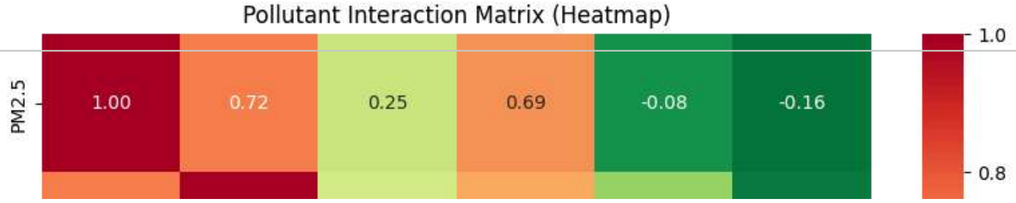
```
# 2. Imputation (Handling Missing Values)
# We use the median because environmental data is often skewed by extreme pollution events
pollutant_list = ['PM2.5', 'PM10', 'NO2', 'NH3', 'CO', 'SO2', 'O3']
for pollutant in pollutant_list:
    if pollutant in df.columns:
        df[pollutant] = df[pollutant].fillna(df[pollutant].median())

print("Preprocessing complete. All missing pollutant values handled.")
```

```
Preprocessing complete. All missing pollutant values handled.
```

```
# Linear Segmented Formula for PM2.5 (Indian Standard Breakpoints)
def calculate_pm25_subindex(pm25):
    if pm25 <= 30: return pm25 * 50 / 30
    elif pm25 <= 60: return 50 + (pm25 - 30) * 50 / 30
    elif pm25 <= 90: return 100 + (pm25 - 60) * 100 / 30
    elif pm25 <= 120: return 200 + (pm25 - 90) * 100 / 30
    elif pm25 <= 250: return 300 + (pm25 - 120) * 100 / 130
    else: return 400 + (pm25 - 250) * 100 / 250

df['PM2.5_SubIndex'] = df['PM2.5'].apply(calculate_pm25_subindex)
```

```
# 1. Seasonal Trends (Monthly Averages)
df['Month'] = df['Date'].dt.month
monthly_trends = df.groupby('Month')['PM2.5'].mean()

plt.figure(figsize=(12, 5))
sns.lineplot(data=df, x='Month', y='PM2.5', color='darkorange', marker='s')
plt.title('Delhi Air Quality: Seasonal Pollutant Trends')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.grid(True)
plt.show()

# 2. Correlation Matrix
# Redefine pollutant_list to only include columns present in df
pollutant_columns_for_corr = ['PM2.5', 'PM10', 'NO2', 'CO', 'SO2', 'Ozone']
plt.figure(figsize=(10, 8))
sns.heatmap(df[pollutant_columns_for_corr].corr(), annot=True, cmap='RdYlGn_r', fmt='.2f')
plt.title('Pollutant Interaction Matrix (Heatmap)')
plt.show()
```


Delhi Air Quality: Seasonal Pollutant Trends
PM2.5
94
93
92
91
90
89
88
87
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month


Pollutant Interaction Matrix (Heatmap)
PM2.5
1.00
0.72
0.25
0.69
-0.08
-0.16
1.0
0.8