ANALYSE AND PREDICT CLIMATE CHANGES

Project Description :--

Here we are analsying the climate of Rwanda. well we go into the analysis we know about the country, Rwanda is a landlocked country in the Great Rift Valley of East Africa, where the African Great Lakes region and Southeast Africa converge. Located a few degrees south of the Equator. Rwanda with a distribution around farm lands, cities and power plants. The years 2019 - 2021 data we have taken for analyze, and we have task to predict the CO2 emissions data for 2022 through November.

Discussing about the some terms on which Climate change is depended :

we have taken the
Density : Amount of gas in a vertical column of the atmosphere above a given area.

Density_Amf : Air Mass Factor (AMF) is a correction factor that relates the amount of gas actually measured by a satellite (slant column) to the amount that would be directly above a point on the ground (vertical column).

Slant Column : The slant column refers to the total amount of a gas measured along the path from the satellite to the Earth's surface (which is at an angle, not straight down).

Vertical Column=(Slant Column/AMF)

cloud_fraction : cloud_fraction is a value between 0 and 1 that indicates how much of the sky is covered by clouds in a specific area.

0 → Clear sky (no clouds) 1 → Fully overcast (100% cloud cover)

azimuth_angle : The azimuth_angle describes the compass direction from which the sunlight or satellite sensor is viewing a point on Earth.

sensor_zenith_angle : The zenith_angle is the angle between:Directly overhead (the zenith, 0°) And the incoming light (from the Sun or satellite sensor)

solar_zenith_angle : The solar_zenith_angle is the angle between the Sun and the vertical direction (zenith) at a specific location.

0° → Sun is directly overhead

90° → Sun is on the horizon (sunrise/sunset)

lessthan 90° → Sun is below the horizon (nighttime)

Ozone : Ozone is a gas made of three oxygen atoms (O_3)

Cloud : Clouds are made of tiny water droplets or ice crystals suspended in the atmosphere. They form when moist air rises and cools, condensing water vapor.

UV Aerosol Index :The UV Aerosol Index (UVAI) is a satellite-derived measure that detects UV-absorbing aerosols (smoke,dust,ash)

In [1]:
```
## import files and libraries
## Data collection process
```

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
train = pd.read_csv("train_data.csv")
```

In [2]:
```python
## Data cleaning process
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 79023 entries, 0 to 79022
Data columns (total 69 columns):
 #   Column                                    Non-Null Count  Dtype  
---  ------                                    --------------  -----  
 0   ID LAT LON YEAR WEEK                      79023 non-null  object 
 1   latitude                                  79023 non-null  float64
 2   longitude                                 79023 non-null  float64
 3   year                                      79023 non-null  int64  
 4   week no                                   79023 non-null  int64  
 5   SO2 density                               64414 non-null  float64
 6   SO2 density amf                           64414 non-null  float64
 7   SO2 slant density                         64414 non-null  float64
 8   SO2 cloud fraction                        64414 non-null  float64
 9   SO2 sensor azimuth angle                  64414 non-null  float64
 10  SO2 sensor zenith angle                   64414 non-null  float64
 11  SO2 solar azimuth angle                   64414 non-null  float64
 12  SO2 solar zenith angle                    64414 non-null  float64
 13  SO2 density 15km                          64414 non-null  float64
 14  CO density                                76901 non-null  float64
 15  CO H2O  density                           76901 non-null  float64
 16  CO cloud height                           76901 non-null  float64
 17  CO sensor altitude                        76901 non-null  float64
 18  CO sensor azimuth angle                   76901 non-null  float64
 19  CO sensor zenith angle                    76901 non-null  float64
 20  CO solar azimuth angle                    76901 non-null  float64
 21  CO solar zenith angle                     76901 non-null  float64
 22  NO2  density                              60703 non-null  float64
 23  NO2 tropospheric density                  60703 non-null  float64
 24  NO2 stratospheric density                 60703 non-null  float64
 25  NO2 slant  density                        60703 non-null  float64
 26  NO2 tropopause pressure                   60703 non-null  float64
 27  NO2 absorbing aerosol index               60703 non-null  float64
 28  NO2 cloud fraction                        60703 non-null  float64
 29  NO2 sensor altitude                       60703 non-null  float64
 30  NO2 sensor azimuth angle                  60703 non-null  float64
 31  NO2 sensor zenith angle                   60703 non-null  float64
 32  NO2 solar azimuth angle                   60703 non-null  float64
 33  NO2 solar zenith angle                    60703 non-null  float64
 34  Formaldehyde tropospheric HCHO  density     71746 non-null  float64
 35  Formaldehyde tropospheric HCHO  density amf 71746 non-null  float64
 36  Formaldehyde HCHO slant  density          71746 non-null  float64
 37  Formaldehyde cloud fraction               71746 non-null  float64
 38  Formaldehyde solar zenith angle           71746 non-null  float64
 39  Formaldehyde solar azimuth angle          71746 non-null  float64
 40  Formaldehyde sensor zenith angle          71746 non-null  float64
 41  Formaldehyde sensor azimuth angle         71746 non-null  float64
 42  UvAerosolIndex absorbing aerosol index    78484 non-null  float64
 43  UvAerosolIndex sensor altitude            78484 non-null  float64
 44  UvAerosolIndex sensor azimuth angle       78484 non-null  float64
 45  UvAerosolIndex sensor zenith angle        78484 non-null  float64
 46  UvAerosolIndex solar azimuth angle        78484 non-null  float64
 47  UvAerosolIndex solar zenith angle         78484 non-null  float64
 48  Ozone O3  density                         78475 non-null  float64
 49  Ozone O3  density amf                     78475 non-null  float64
 50  Ozone O3 slant  density                   78475 non-null  float64
```

```
 51  Ozone O3 effective temperature                 78475 non-null  float64
 52  Ozone cloud fraction                           78475 non-null  float64
 53  Ozone sensor azimuth angle                     78475 non-null  float64
 54  Ozone sensor zenith angle                      78475 non-null  float64
 55  Ozone solar azimuth angle                      78475 non-null  float64
 56  Ozone solar zenith angle                       78475 non-null  float64
 57  Cloud cloud fraction                           78539 non-null  float64
 58  Cloud cloud top pressure                       78539 non-null  float64
 59  Cloud cloud top height                         78539 non-null  float64
 60  Cloud cloud base pressure                      78539 non-null  float64
 61  Cloud cloud base height                        78539 non-null  float64
 62  Cloud cloud optical depth                      78539 non-null  float64
 63  Cloud surface albedo                           78539 non-null  float64
 64  Cloud sensor azimuth angle                     78539 non-null  float64
 65  Cloud sensor zenith angle                      78539 non-null  float64
 66  Cloud solar azimuth angle                      78539 non-null  float64
 67  Cloud solar zenith angle                       78539 non-null  float64
 68  emission                                       79023 non-null  float64
dtypes: float64(66), int64(2), object(1)
memory usage: 41.6+ MB
```

In [3]:
```python
## Remove NAN from data
train =train.dropna()
```

In [4]:
```python
train.shape
```

Out[4]:
```
(57209, 69)
```

In [5]:
```python
desc =train.describe()
```

In [6]:
```python
desc.to_csv('describe_info.csv')
```

In [7]:
```python
print(train['longitude'].nunique())
print(train['latitude'].nunique())
```

```
450
449
```

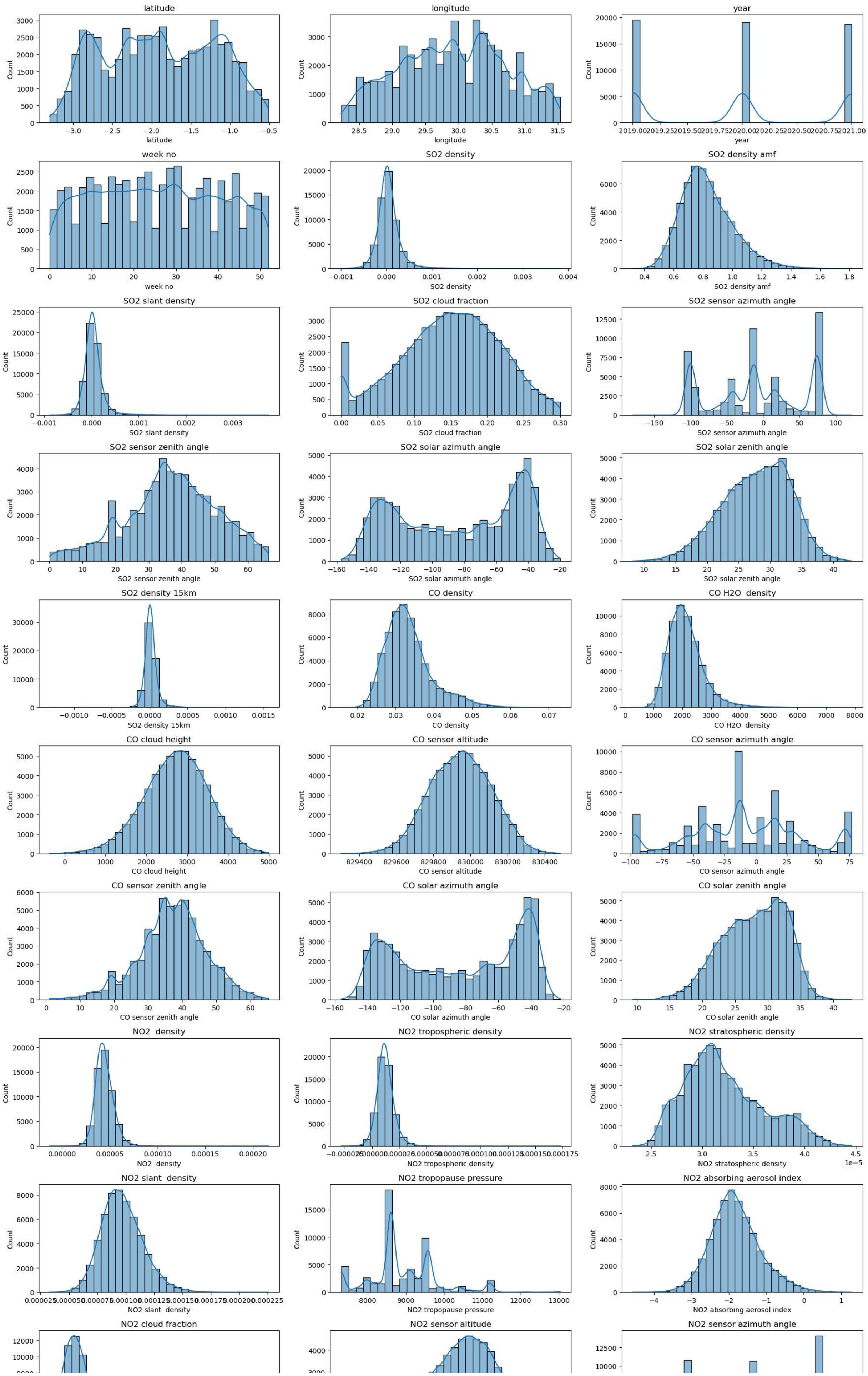
In [8]:
```python
import folium

# Create base map centered around the mean Location
map_center = [train['latitude'].mean(), train['longitude'].mean()]
dot_map = folium.Map(location=map_center, zoom_start=6)

# Add dots
for _, row in train.iterrows():
    folium.CircleMarker(
        location=[row['latitude'], row['longitude']],
        radius=3,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.7
    ).add_to(dot_map)
```

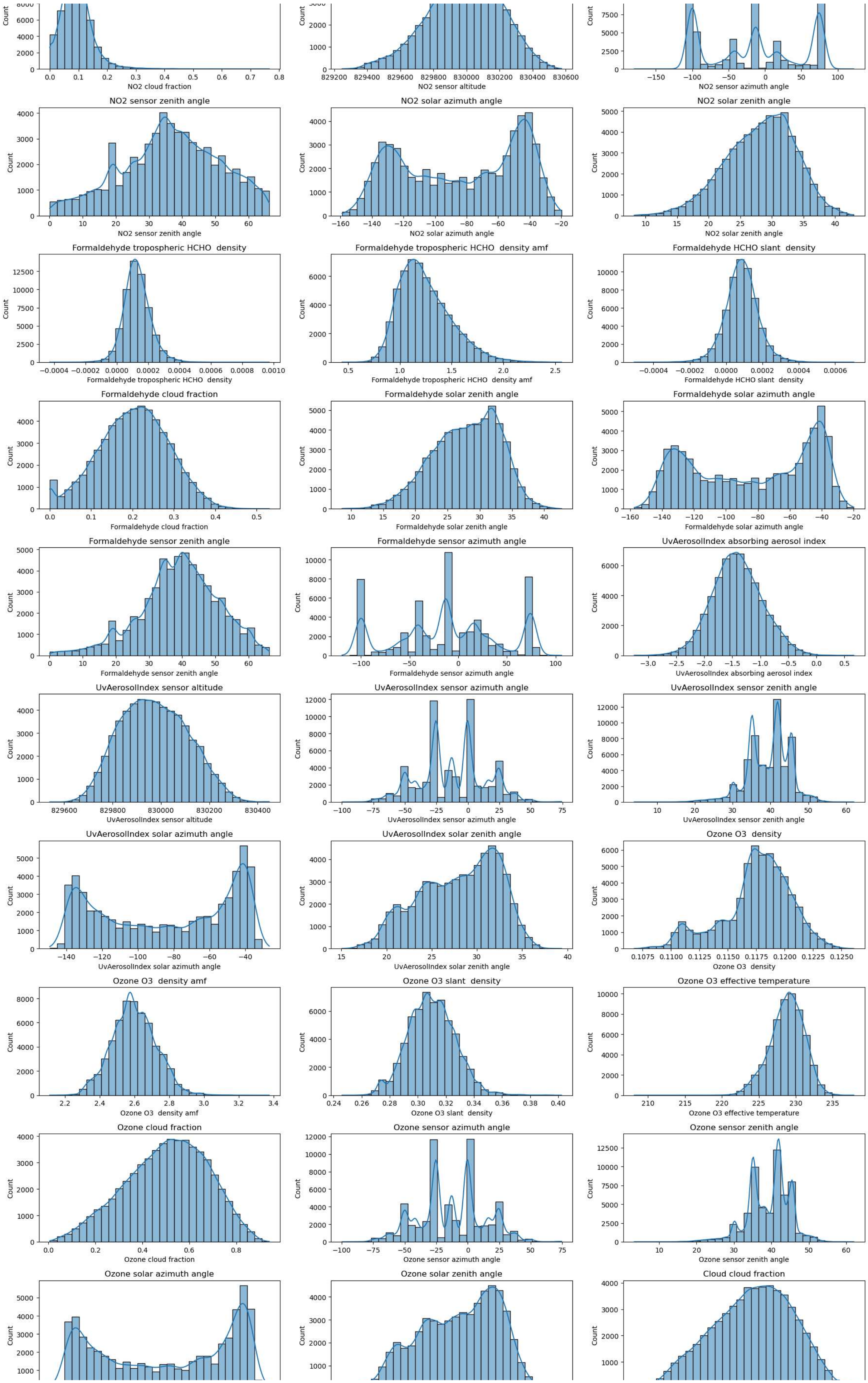
```
# Show the map
dot_map
```

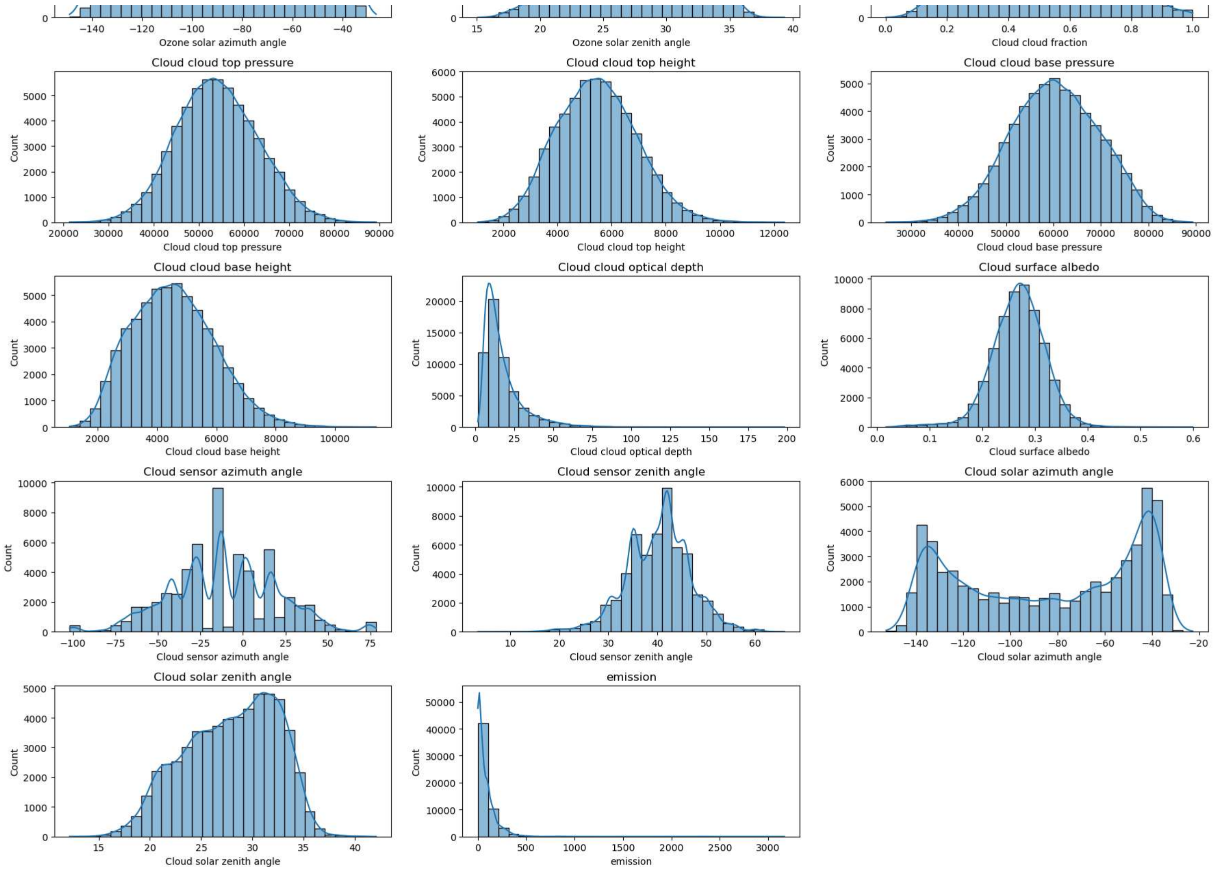
Out[8]:

In [9]:

```
numeric_cols = train.select_dtypes(include='number').columns
n = len(numeric_cols)
cols = 3
rows = int(np.ceil(n / cols))

fig, axes = plt.subplots(rows, cols, figsize=(18, rows * 3))

for i, col in enumerate(numeric_cols):
    ax = axes[i // cols, i % cols]
    sns.histplot(train[col], kde=True, ax=ax, bins=30)
    ax.set_title(col)

# Hide empty subplots
for j in range(i + 1, rows * cols):
    fig.delaxes(axes[j // cols, j % cols])

plt.tight_layout()
plt.savefig('distribution.png')
plt.show()
```


latitude
longitude
year
week no
SO2 density
SO2 density amf
SO2 slant density
SO2 cloud fraction
SO2 sensor azimuth angle
SO2 sensor zenith angle
SO2 solar azimuth angle
SO2 solar zenith angle
SO2 density 15km
CO density
CO H2O density
CO cloud height
CO sensor altitude
CO sensor azimuth angle
CO sensor zenith angle
CO solar azimuth angle
CO solar zenith angle
NO2 density
NO2 tropospheric density
NO2 stratospheric density
NO2 slant density
NO2 tropopause pressure
NO2 absorbing aerosol index
NO2 cloud fraction
NO2 sensor altitude
NO2 sensor azimuth angle
Count


NO2 cloud fraction
NO2 sensor altitude
NO2 sensor azimuth angle
NO2 sensor zenith angle
NO2 solar azimuth angle
NO2 solar zenith angle
Formaldehyde tropospheric HCHO density
Formaldehyde tropospheric HCHO density amf
Formaldehyde HCHO slant density
Formaldehyde cloud fraction
Formaldehyde solar zenith angle
Formaldehyde solar azimuth angle
Formaldehyde sensor zenith angle
Formaldehyde sensor azimuth angle
UvAerosolIndex absorbing aerosol index
UvAerosolIndex sensor altitude
UvAerosolIndex sensor azimuth angle
UvAerosolIndex sensor zenith angle
UvAerosolIndex solar azimuth angle
UvAerosolIndex solar zenith angle
Ozone O3 density
Ozone O3 density amf
Ozone O3 slant density
Ozone O3 effective temperature
Ozone cloud fraction
Ozone sensor azimuth angle
Ozone sensor zenith angle
Ozone solar azimuth angle
Ozone solar zenith angle
Cloud cloud fraction
Count

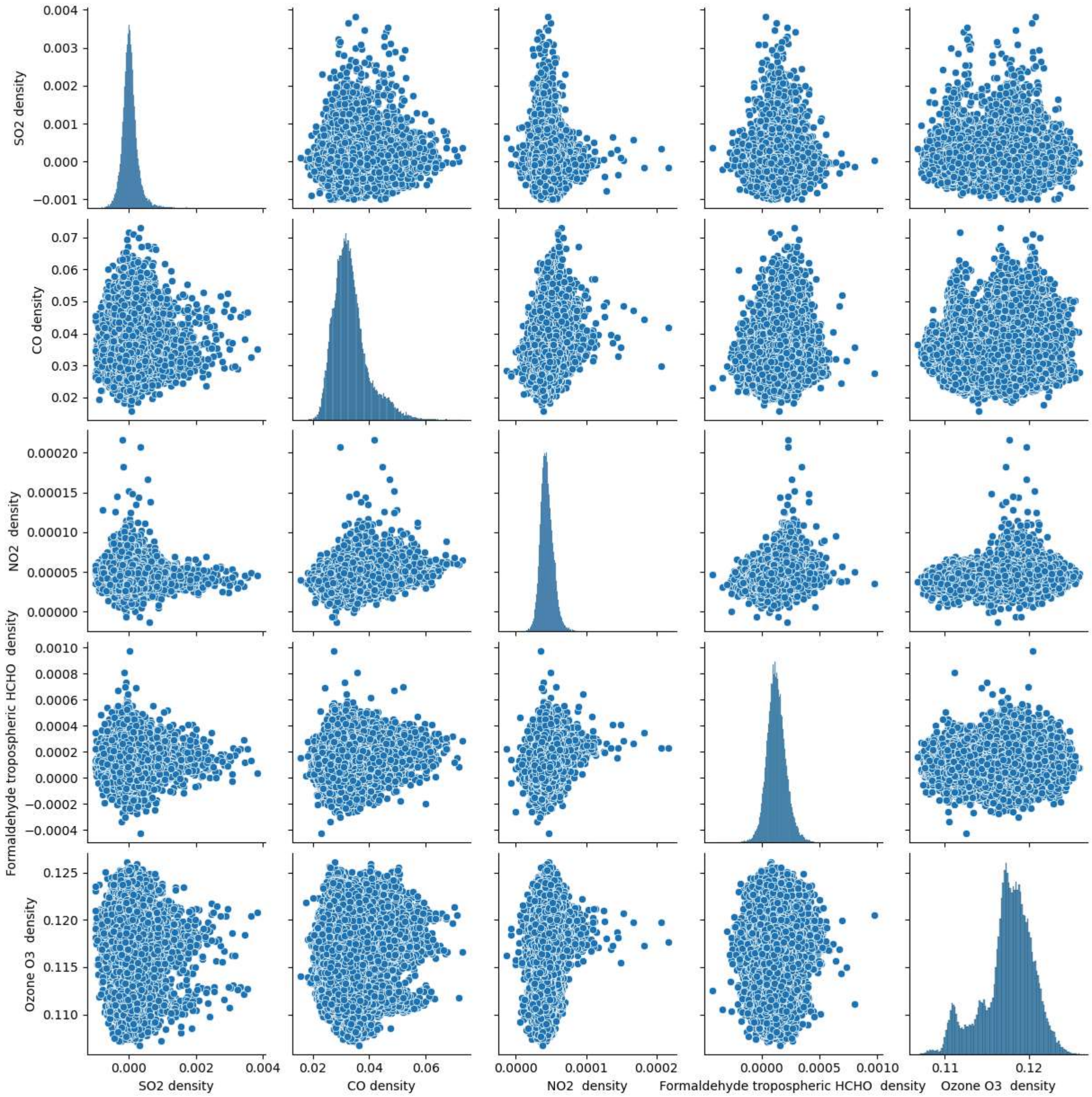


From the above graph i have we can clearly say that reacording are taken in different daytimings 1.Recording taken from the Azimuth angle(sun position to object) are almost Day and night or twilight timings. the angle is frenquency distributed

2.Recording taken from the zenith angle(sun rays project on object) are mostly taken on day timings.The reading angles are normaly distrubted.

Here we are taking data, where AMF and Slaint density are calculated and final density we get Vertical Column Density (VCD) = Slant Column Density / AMF

In [10]:
```
## relation between densities


plot = sns.pairplot(train[['SO2 density','CO density','NO2  density','Formaldehyde

plot.fig.savefig("plot.jpeg")
```



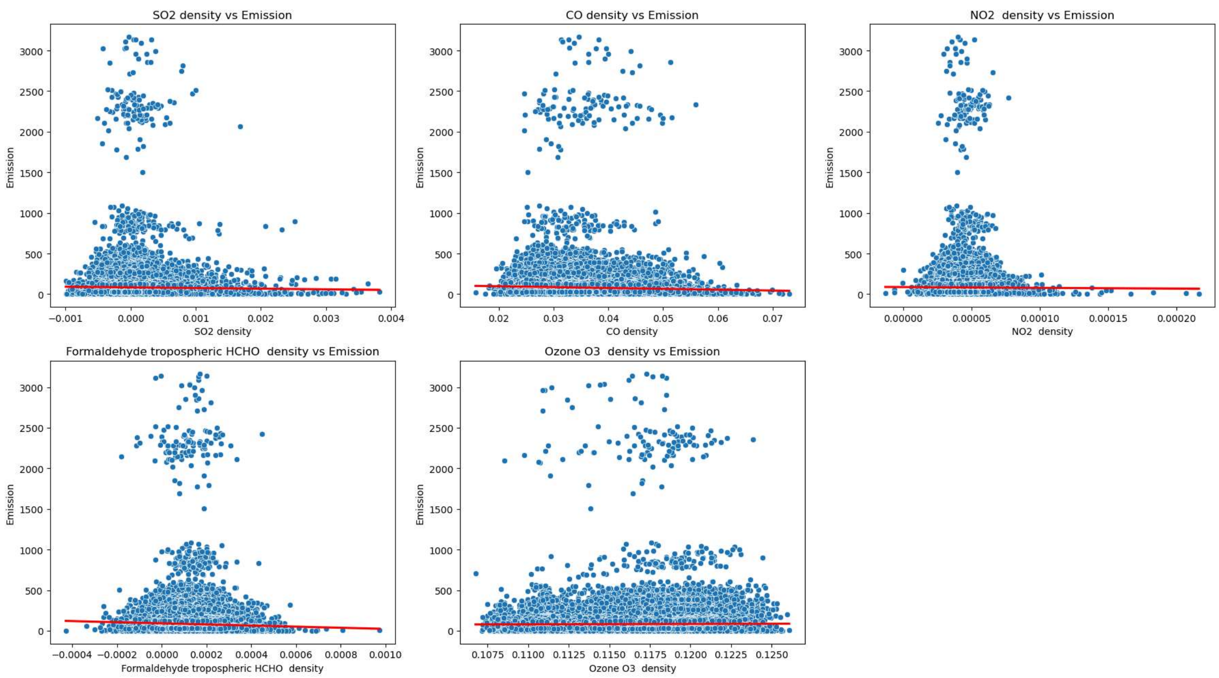
We know that
---> Weak Positive Relationships: SO2 shows a weak positive correlation with all other variables.

---> Moderate Positive Relationships: CO, NO2, and HCHO show moderate positive correlations with each other and with O3.

---> O3 as a Central Variable: Ozone (O3) density tends to increase as CO, NO2, and HCHO densities increase.

In [11]:
```
densities = [
    'SO2 density',
    'CO density',
    'NO2  density',
    'Formaldehyde tropospheric HCHO  density',
    'Ozone O3  density']



fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(18, 10))
```

```
axes = axes.flatten()

# Plot each subplot
for i, gas in enumerate(densities):
    sns.scatterplot(data=train, x=gas, y='emission', ax=axes[i])
    sns.regplot(data=train, x=gas, y='emission', scatter=False, ax=axes[i], color='
    axes[i].set_title(f'{gas} vs Emission')
    axes[i].set_xlabel(gas)
    axes[i].set_ylabel('Emission')

# Remove unused subplot if any
if len(densities) < len(axes):
    for j in range(len(densities), len(axes)):
        fig.delaxes(axes[j])

plt.tight_layout()
plt.savefig("densities_vs_emission_subplot.png", dpi=300)
plt.show()
```



we can only interpret the patterns in the data.
--->CO and NO2 are the strongest predictors: CO and NO2 densities show the strongest positive relationships with "Emission," suggesting they might be more influential factors.

--->SO2, HCHO, and O3 have weak relationships: SO2, HCHO, and O3 densities have weak positive relationships with "Emission," indicating they might not be strongly associated with the emission process being measured.

---> need to be more analysis

Reading taken in different areas how many area reaings has taken and each area increases more co2 emission

In [12]:
```
Weekwise = train.groupby(['year','week no'])['emission'].mean().reset_index()
```

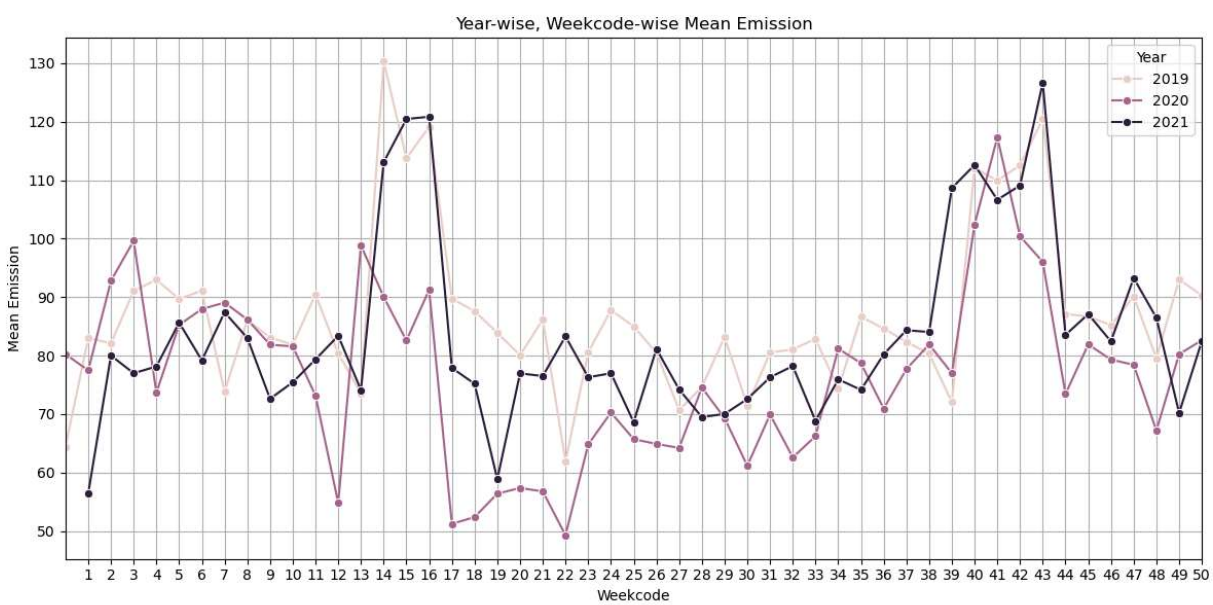
In [13]:
```
Weekwise
```

Out[13]:

	year	week no	emission
0	2019	0	64.415420
1	2019	1	82.998669
2	2019	2	82.105929
3	2019	3	91.057897
4	2019	4	92.981316
...	...	...	...
153	2021	48	86.449145
154	2021	49	70.184358
155	2021	50	82.500927
156	2021	51	71.679088
157	2021	52	72.307177

158 rows × 3 columns

In [36]:
```
plt.figure(figsize=(12, 6))
sns.lineplot(data=Weekwise, x='week no', y='emission', hue='year', marker='o')
plt.title("Year-wise, Weekcode-wise Mean Emission")
plt.xlabel("Weekcode")
plt.xlim(0,50)
plt.xticks(ticks=range(1, 51))
plt.ylabel("Mean Emission")
plt.legend(title='Year')
plt.grid(True)
plt.tight_layout()
plt.savefig("Trend chart.png")
plt.show()
```

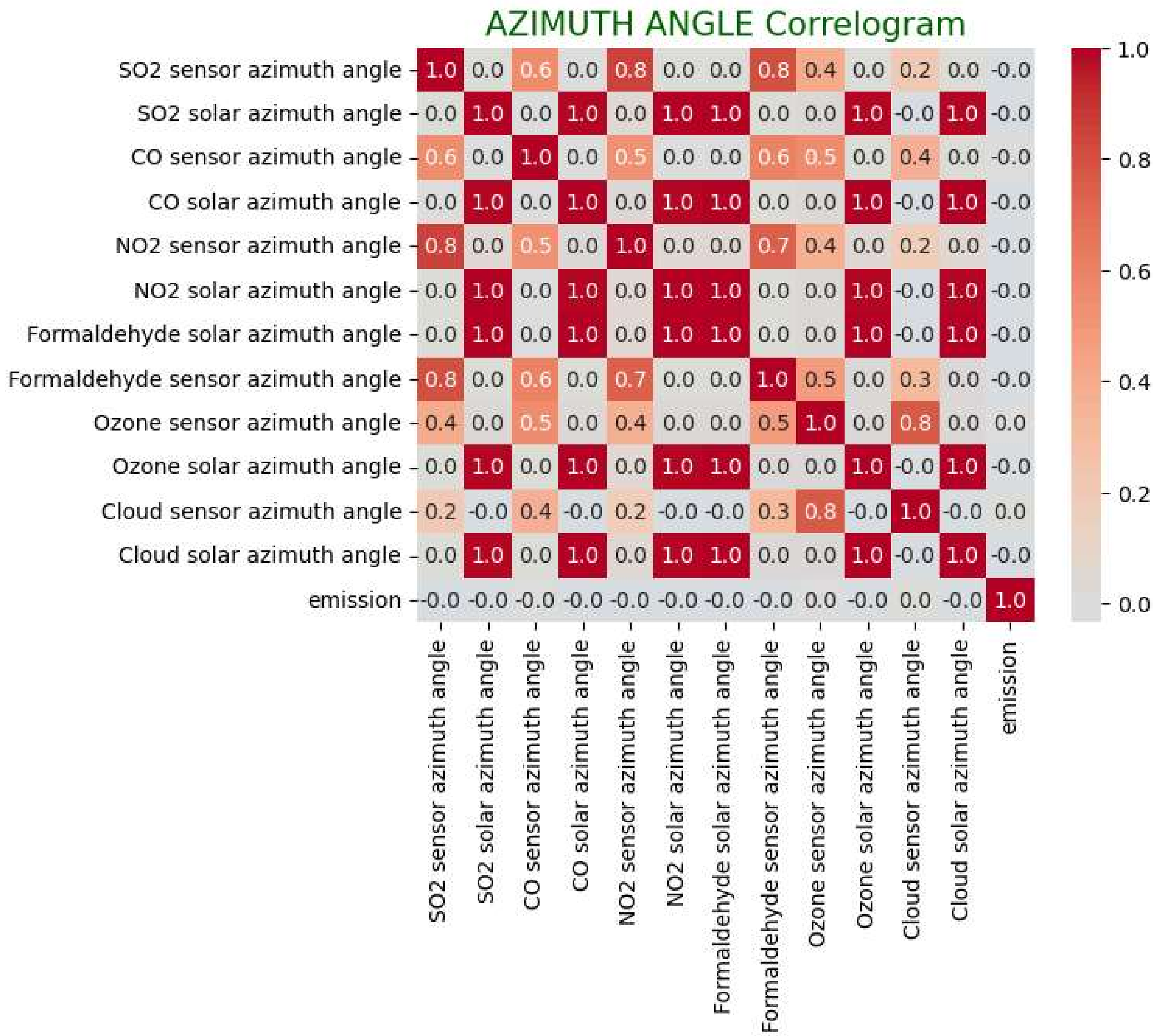


There is a clear seasonal and event-based emission pattern. Year 2020 stands out with a significant drop during mid-year, indicating the impact of the pandemic. Peaks during Weeks 14–16 and 40–44 are consistent, indicating possible recurring high-emission activities.

MODEL

In [15]:
```
col1 =['SO2 sensor azimuth angle','SO2 solar azimuth angle','CO sensor azimuth angl
      ,'NO2 solar azimuth angle','Formaldehyde solar azimuth angle','Formaldehyde s
      ,'Ozone sensor azimuth angle','Ozone solar azimuth angle','Cloud sensor azimu
      'Cloud solar azimuth angle','emission']
col2 =['SO2 sensor zenith angle','SO2 solar zenith angle','CO sensor zenith angle',
       'NO2 solar zenith angle','Formaldehyde solar zenith angle','Ozone sensor zen
       'Ozone solar zenith angle','Cloud sensor zenith angle','Cloud solar zenith a
corr1 = train[col1].corr()
```

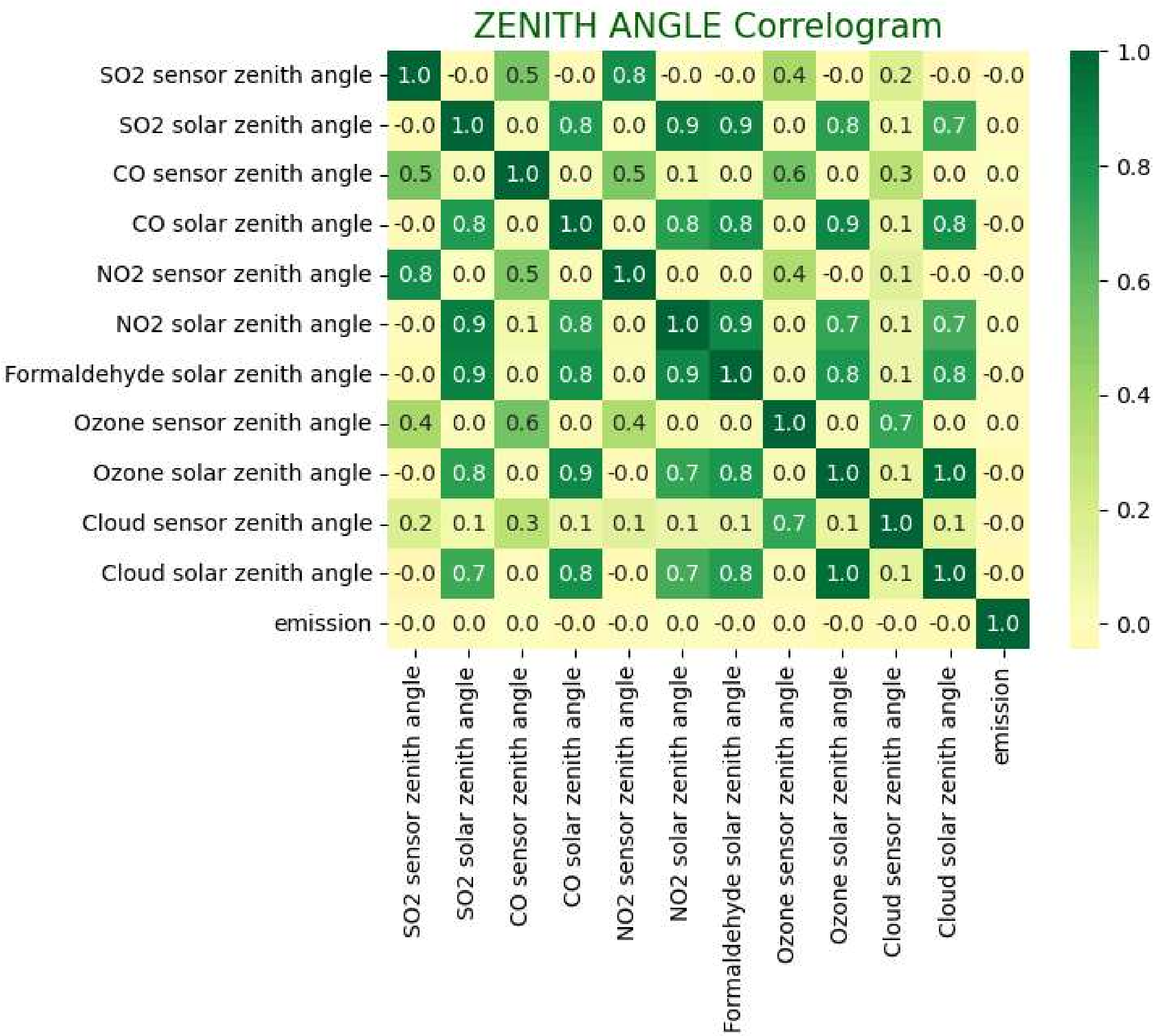
In [39]:
```
sns.heatmap(corr1, cmap='coolwarm', annot = True, fmt=".1f",center = 0)
plt.title('AZIMUTH ANGLE Correlogram', fontsize = 15, color = 'darkgreen')
plt.savefig("Azimuth Angle Correlogram.jpeg")
plt.show()
```



Here all solar azimuth angles are related (appox 1 correlation) each other,

In [17]:
```
corr2 = train[col2].corr()
```

In [40]:
```
sns.heatmap(corr2, cmap='RdYlGn', annot = True,fmt=".1f", center = 0)
plt.title('ZENITH ANGLE Correlogram', fontsize = 15, color = 'darkgreen')
plt.savefig("Zenith Angle Correlogram")
plt.show()
```



CO sensor zenith angle and Ozone sensor zenith angle has 0.6 cor.
CO solar ↔ Ozone solar has 0.9.

other parameter futures are strongly correlated with each other so we can remove them.

Removing the Angle feature which is less importance in predicting the emission, but mostly related to the Density of each Gas parameter, Here we are removing the angles from data for elimating noise in data

In [19]:
```
features_to_drop = ['SO2 solar zenith angle','CO solar zenith angle','NO2 solar zen
                    'Ozone solar zenith angle','Cloud solar zenith angle','SO2 sens
                    'NO2 sensor azimuth angle','Formaldehyde sensor azimuth angle',
                    'Cloud sensor azimuth angle','ID LAT LON YEAR WEEK','year','wee



cleaned_train = train.drop(features_to_drop,axis=1)
```

In [20]:
```
cleaned_train.head()
```

Out[20]:

	latitude	longitude	SO2 density	SO2 density amf	SO2 slant density	SO2 cloud fraction	SO2 sensor zenith angle	SO2 solar azimuth angle	
1	-0.51	29.29	0.000021	0.728214	0.000014	0.130988	39.137194	-140.874435	0
2	-0.51	29.29	0.000514	0.748199	0.000385	0.110018	52.868816	-150.191757	0
4	-0.51	29.29	-0.000079	0.676296	-0.000048	0.121164	35.515587	-137.409159	-0
5	-0.51	29.29	0.000294	0.871713	0.000242	0.227656	57.097124	-136.616859	0
6	-0.51	29.29	-0.000285	0.791956	-0.000226	0.119397	58.496368	-143.726913	-0

5 rows × 54 columns

In [21]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

In [41]:
```
x = cleaned_train.drop(['emission'],axis=1)
y =cleaned_train['emission']
```

In [42]:
```
x_train,x_test,y_train,y_test =train_test_split(x,y, random_state= 43,test_size =0.
```

In [43]:
```
model = RandomForestRegressor(random_state=42)
```

In [44]:
```
model.fit(x_train, y_train)
```

Out[44]:
```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [45]:
```
y_pred = model.predict(x_test)
```

In [46]:
```
from sklearn.metrics import mean_squared_error, r2_score
print("R² Score:", r2_score(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
```

```
R² Score: 0.9379918427237676
MSE: 1350.3968644999889
```

In this project, we analyzed and modeled environmental emission data using various sensor and solar angle parameters. After performing thorough preprocessing—such as handling missing values, analyzing correlations, and exploring trends—we selected the Random Forest Regressor for prediction modeling.

The model demonstrated exceptionally high performance, achieving:

R^2 Score: 0.9379 Mean Squared Error (MSE): 1350.39