Anno accademico 2023/2024
Università Ca' Foscari Venezia

Air pollution

Effect of nuclear energy on air pollution in Europe

Computer Programming and Data Management

Data Analytics for Business and Society

The Bug Slayerz

Objective of the research

Does a positive association exist between nuclear-generated electricity and the corresponding levels of air pollution for countries in Europe?

Do those countries producing and relying on nuclear-generated electricity have lower levels of air pollution?

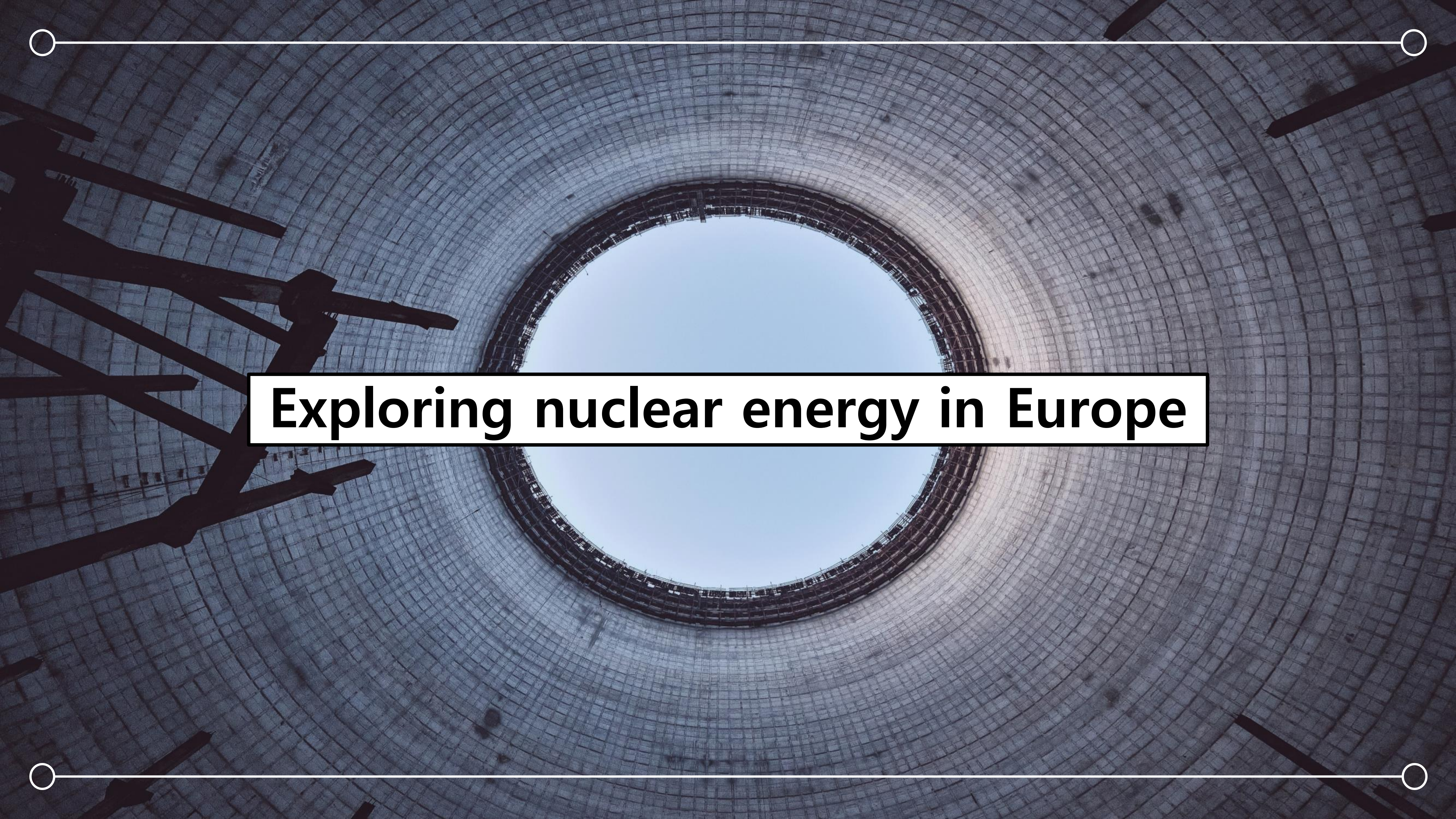
INDEX

- **Exploring nuclear energy in Europe**
  - Share of energy consumption by source
  - Share of electricity production by source
  - The geolocation of all power plants in Europe
- **Exploring air pollution in Europe**
  - CO_2
  - SO_x
  - NO_x
  - PM 2.5
- **Nuclear energy vs. levels of air pollution**
- **Linear regression**
- **Conclusion**

Installing and importing packages

```
#Importing the libraries
import pandas as pd
import numpy as np
import geopandas as gpd
from plotly.subplots import make_subplots
import plotly.express as px
import plotly.graph_objects as go
from google.colab import drive
drive.mount('/content/drive')
from PIL import Image
import glob
from sklearn.impute import SimpleImputer
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
import seaborn as sns
import matplotlib.pyplot as plt
```

Exploring nuclear energy in Europe



Geolocation of power plants in Europe

Importing the datasets

Dataset:
World Resource Institute

Source:

L. Byers, J. Friedrich, R. Hennig, A. Kressig, Li X., C. McCormick, and L. Malaguzzi Valeri. 2021. "A Global Database of Power Plants." Washington, DC: World Resources Institute. Available online at www.wri.org/publication/global-database-power-plants.

Table 3 | Indicator Coverage for Geolocated Plants

INDICATOR	DESCRIPTION	PERCENT OF PLANTS WITH INDICATOR
Name	Power plant name	100%
Fuel type	fuel category	100%
Capacity	installed electrical capacity (MW)	100%
Location	latitude and longitude (xx.xx, xx.xx)	100% (by definition)
Year of capacity	year of reported capacity	100%
Year of generation	year of reported generation	100%
Data source	source of data	100%
URL	URL linking directly to data source	100%
Annual generation	annual generation (calendar year) in gigawatt hours (GWhs), gross	100%: 24% reported, 76% estimated (see Section 6)
Operational status	commissioned/retired/planned	100%
Generator technology	technology used to generate electricity	64%
Owner	primary owner of the power plant	60%
Commissioning year	first year plant generated electricity	45%

Note: Annual generation represents the only operational indicator in the first version of the database. More operational indicators will be included in future versions.
Source: WRI, 2018.

Cleaning the dataset

1. Adding continent information to the dataset by merging it with GeoDataFrame (cont) based on the country names.

```
[ ] # keeping only desired columns
    locs = locs[['country_long','latitude','longitude','primary_fuel','estimated_generation_gwh_2017']]
```

```
[ ] # add the continent of each countries from geopandas
    cont = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
    cont = cont[['name','continent']]

    locs = locs.merge(cont, how='left', left_on='country_long', right_on='name').drop(columns=['name'])
```

2. Keeping only countries in Europe

```
# selecting only european countries
locs = locs[locs['continent']=='Europe'].reset_index(drop=True)
```

3. Checking for countries not assigned to continent and if so adding them manually

```
# check if there are still countries not assigned to the continent
locs.loc[pd.isna(locs.continent),:].country_long.unique()
```

```
# assigning the continent Europe to the detected european missing countries (Macedonia, Bosnia and Herzegovina and Czech Republic)
locs.loc[locs.country_long=='Macedonia','continent'] = 'Europe'
locs.loc[locs.country_long=='Bosnia and Herzegovina','continent'] = 'Europe'
locs.loc[locs.country_long=='Czech Republic','continent'] = 'Europe'
```

Geolocation of power plants in Europe

Importing the datasets

Dataset: World Resource Institute	
Source: WORLD RESOURCES INSTITUTE	L. Byers, J. Friedrich, R. Hennig, A. Kressig, Li X., C. McCormick, and L. Malaguzzi Valeri. 2021. "A Global Database of Power Plants." Washington, DC: World Resources Institute. Available online at www.wri.org/ publication/global-database-power-plants.

Table 3 | Indicator Coverage for Geolocated Plants

INDICATOR	DESCRIPTION	PERCENT OF PLANTS WITH INDICATOR
Name	Power plant name	100%
Fuel type	fuel category	100%
Capacity	installed electrical capacity (MW)	100%
Location	latitude and longitude (xx.xx, xx.xx)	100% (by definition)
Year of capacity	year of reported capacity	100%
Year of generation	year of reported generation	100%
Data source	source of data	100%
URL	URL linking directly to data source	100%
Annual generation	annual generation (calendar year) in gigawatt hours (GWhs), gross	100%: 24% reported, 76% estimated (see Section 6)
Operational status	commissioned/retired/planned	100%
Generator technology	technology used to generate electricity	64%
Owner	primary owner of the power plant	60%
Commissioning year	first year plant generated electricity	45%

Note: Annual generation represents the only operational indicator in the first version of the database. More operational indicators will be included in future versions.
Source: WRI, 2018.

Cleaning the dataset

4. Checking and removing NAs if present

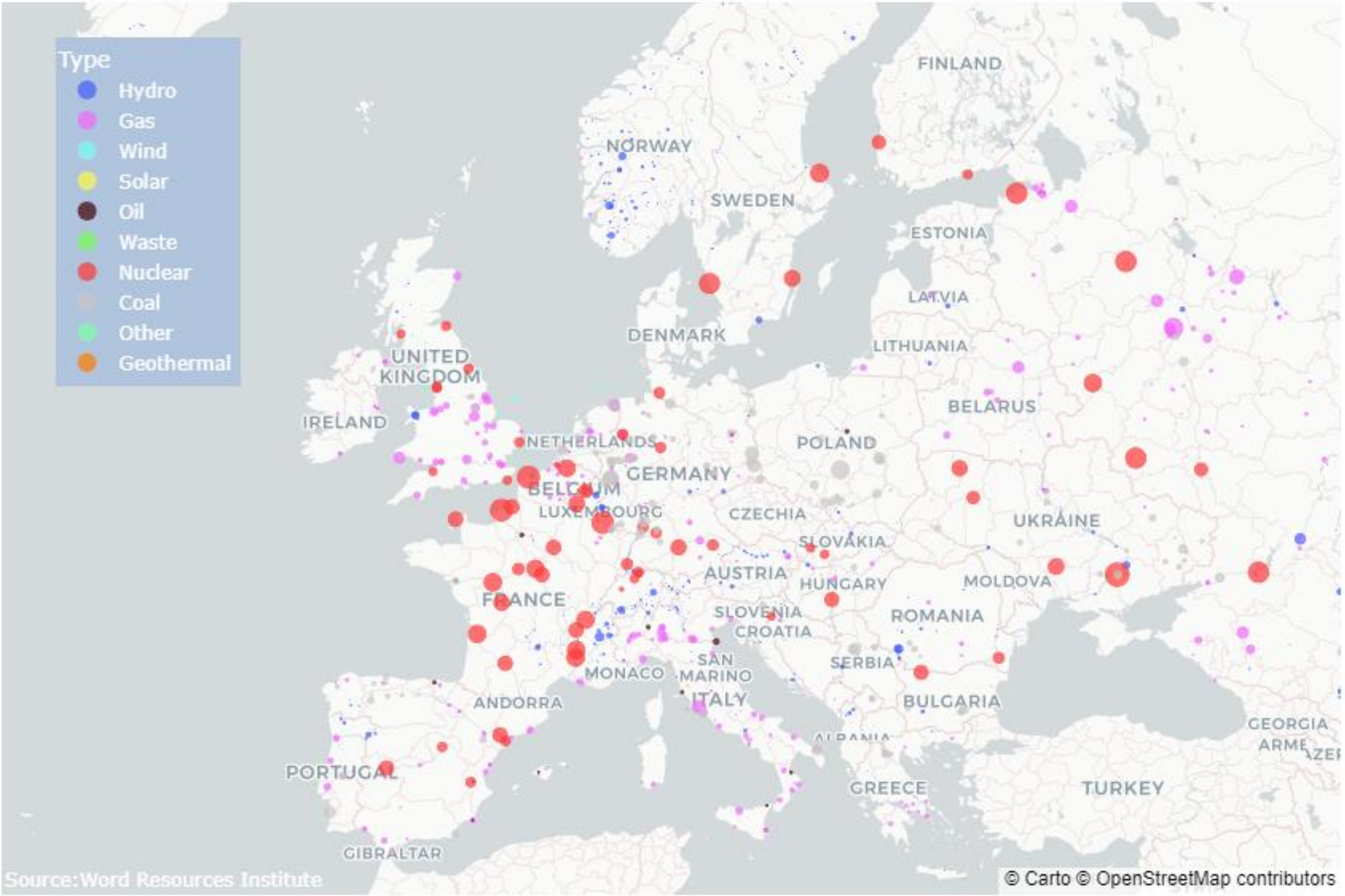
```
# check for missing values for each of the columns in the dataset
locs.isna().sum()
```

```
# removing missing values
locs=locs.dropna()
```

5. Using `px.scatter_mapbox` to geolocate power plants

Geolocating all power plants provides a <u>comprehensive view of the energy landscape in Europe</u>. Understanding the **distribution** and **density** of <u>power plants</u> across the continent is essential for context.

Location and types of power plant in Europe (updated up to 2017)



Importing the datasets

Dataset: Share of direct primary energy consumption by source	
Source:	Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data. “Share of primary energy consumption that comes from oil – Using the substitution method” [dataset]. Energy Institute, “Statistical Review of World Energy” [original data].

Dataset: Share of direct primary energy production by source	
Source:	Ember - Yearly Electricity Data (2023); Ember - European Electricity Review (2022); Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data. “Share of electricity generated by coal” [dataset]. Ember, “Yearly Electricity Data”; Ember, “European Electricity Review”; Energy Institute, “Statistical Review of World Energy” [original data].

Cleaning the dataset

```
shares = pd.read_csv('/content/drive/My Drive/Python_project/Nuclear_energy_generation/share_energy_source_sub.csv', delimiter=';')
```

1. Adding continent information to the dataset by merging it with GeoDataFrame (cont) based on the country names.

```
# adding the continent of each countries from geopandas
cont = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
cont = cont[['name','continent']]
# adding continent column to the dataset
shares = shares.merge(cont, how='left', left_on='Entity', right_on='name').drop(columns=['name'])
```

2. Keeping only countries in Europe.

```
# keep european countries
shares = shares[shares['continent']=='Europe'].reset_index(drop=True)
```

3. Renaming the columns.

```
# rename the columns
shares.rename(columns={
    'Oil (% equivalent primary energy)': 'Oil',
    'Coal (% equivalent primary energy)': 'Coal',
    'Solar (% equivalent primary energy)': 'Solar',
    'Nuclear (% equivalent primary energy)': 'Nuclear',
    'Hydro (% equivalent primary energy)': 'Hydro',
    'Wind (% equivalent primary energy)': 'Wind',
    'Gas (% equivalent primary energy)': 'Gas',
    'Other renewables (% equivalent primary energy)': 'Other renewables',
    }, inplace=True)
```

Importing the datasets

Dataset: Share of direct primary energy consumption by source	
Source: Our World in Data; EMBER	Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data. “Share of primary energy consumption that comes from oil – Using the substitution method” [dataset]. Energy Institute, “Statistical Review of World Energy” [original data].

Dataset: Share of direct primary energy production by source	
Source: Our World in Data; EMBER	Ember - Yearly Electricity Data (2023); Ember - European Electricity Review (2022); Energy Institute - Statistical Review of World Energy (2023) – with major processing by Our World in Data. “Share of electricity generated by coal” [dataset]. Ember, “Yearly Electricity Data”; Ember, “European Electricity Review”; Energy Institute, “Statistical Review of World Energy” [original data].

Cleaning the dataset

3. Normalizing the shares of energy consumed.

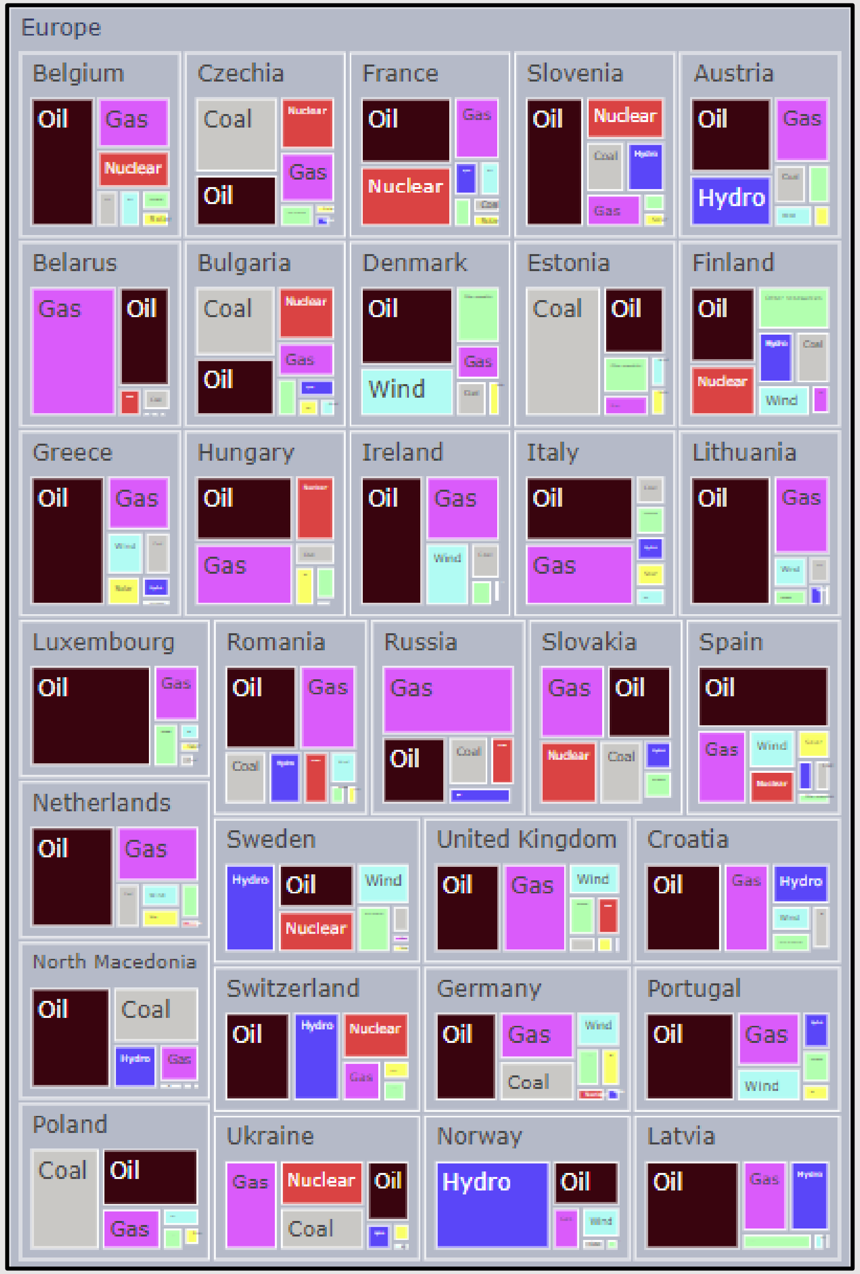
```
# Since the columns is the % share of energy consumption, the sum of sources should be exactly 100%.
# However, for some rows this is not the case (e.g. ~99.9%); below we will slightly modify the values for the 'Other renewables' columns to be the case.

fuel_type=['Oil', 'Coal', 'Solar', 'Nuclear', 'Hydro', 'Wind', 'Gas']

shares['Other renewables'] = 100 - shares[fuel_type].sum(axis=1)

# if the values of 'other renewables' is negative, correct them.
index_=shares.index[shares['Other renewables']<0]
for i in range(len(index_)):
    shares.at[index_[i],'Oil'] = shares['Oil'].iloc[index_[i]]+shares['Other renewables'].iloc[index_[i]]
    shares.at[index_[i],'Other renewables'] = 0
```

4. Create a new DataFrame adapted to the plotly.treemap.

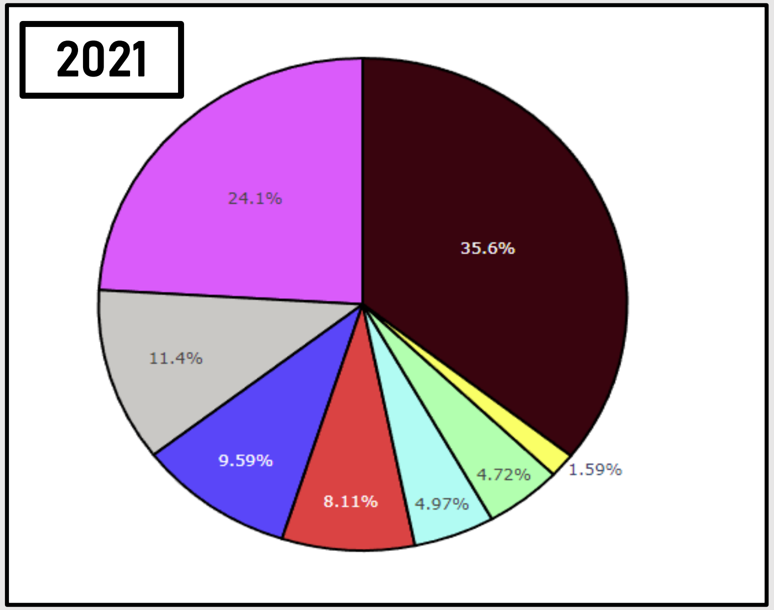
```
# create a new DataFrame adapted to the plotly.treemap
# list of desired columns
columns = ['Country','Year','fuel_type','fuel_type_val']

# list of energy sources
fuel_type = ['Oil', 'Coal', 'Solar', 'Nuclear', 'Hydro', 'Wind', 'Gas', 'Other renewables']

# list of year span
years=shares['Year'].unique()

# prealocate the number of rows of new dataframe
index=len(shares)*len(fuel_type)

# create the dataframe
shares_new=pd.DataFrame(index=range(index),columns=columns)

# insert the values from the old shares
z=0
for i in range(len(shares)):
    for f in range(len(fuel_type)):

        shares_new.at[z,'Country'] = shares['Entity'].iloc[i]
        shares_new.at[z,'Year'] = shares['Year'].iloc[i]
        shares_new.at[z,'fuel_type'] = fuel_type[f]
        shares_new.at[z,'fuel_type_val'] = shares[fuel_type[f]].iloc[i]
        z=z+1
```

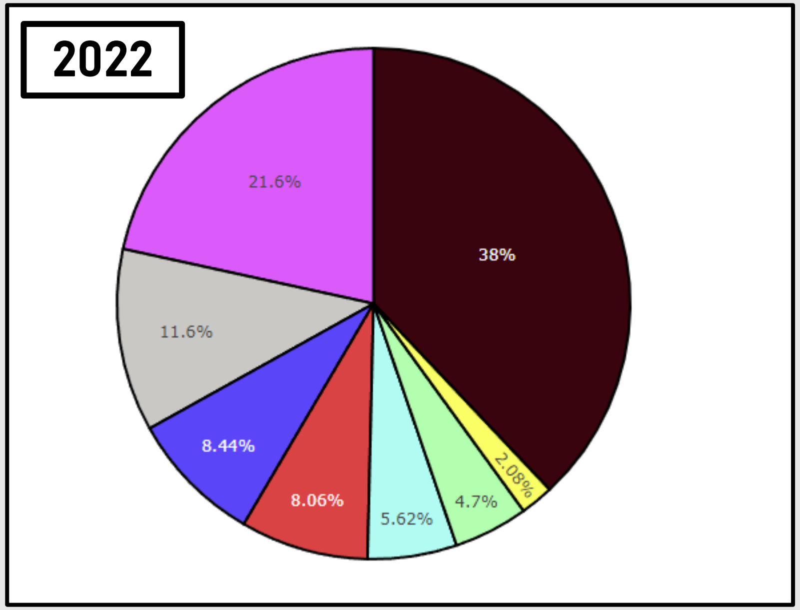
Exploratory analysis

Share of direct primary energy **consumption** by source



The tree map highlights **France**, **Sweden**, **Ukraine**, and **Slovakia** as the primary consumers of nuclear-generated electricity in Europe.





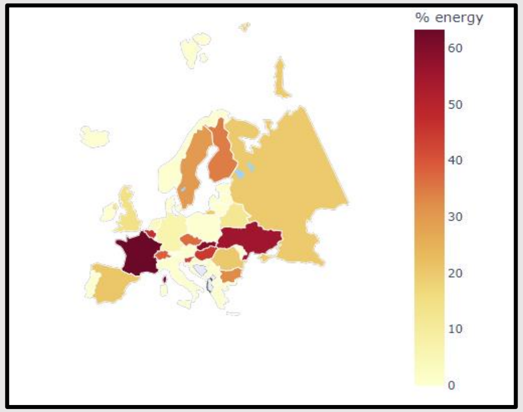
2022 witnessed a slight **drop** in the consumption share of nuclear and hydropower, despite a slight global increase in renewable sources.

Exploratory analysis

Share of direct primary energy **production** by source

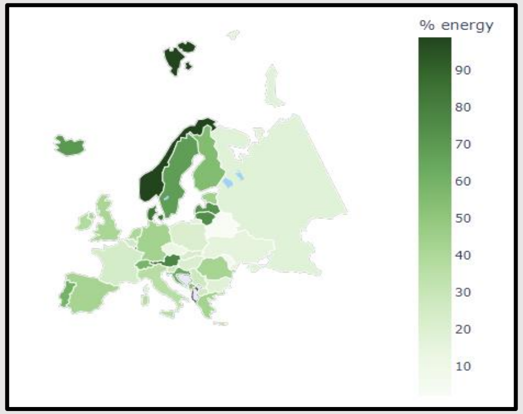
Countries like **France** (63.28%), **Slovakia** (59.45%) and **Ukraine** (55.35%) have embraced nuclear power as their main source of electricity, showcasing a commitment to low-carbon energy solutions.

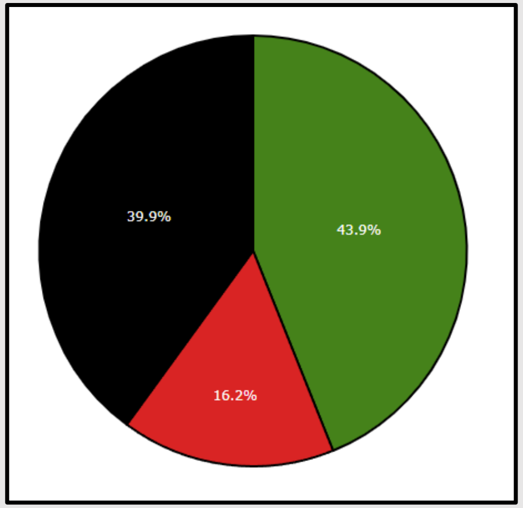
Nuclear



Fossil

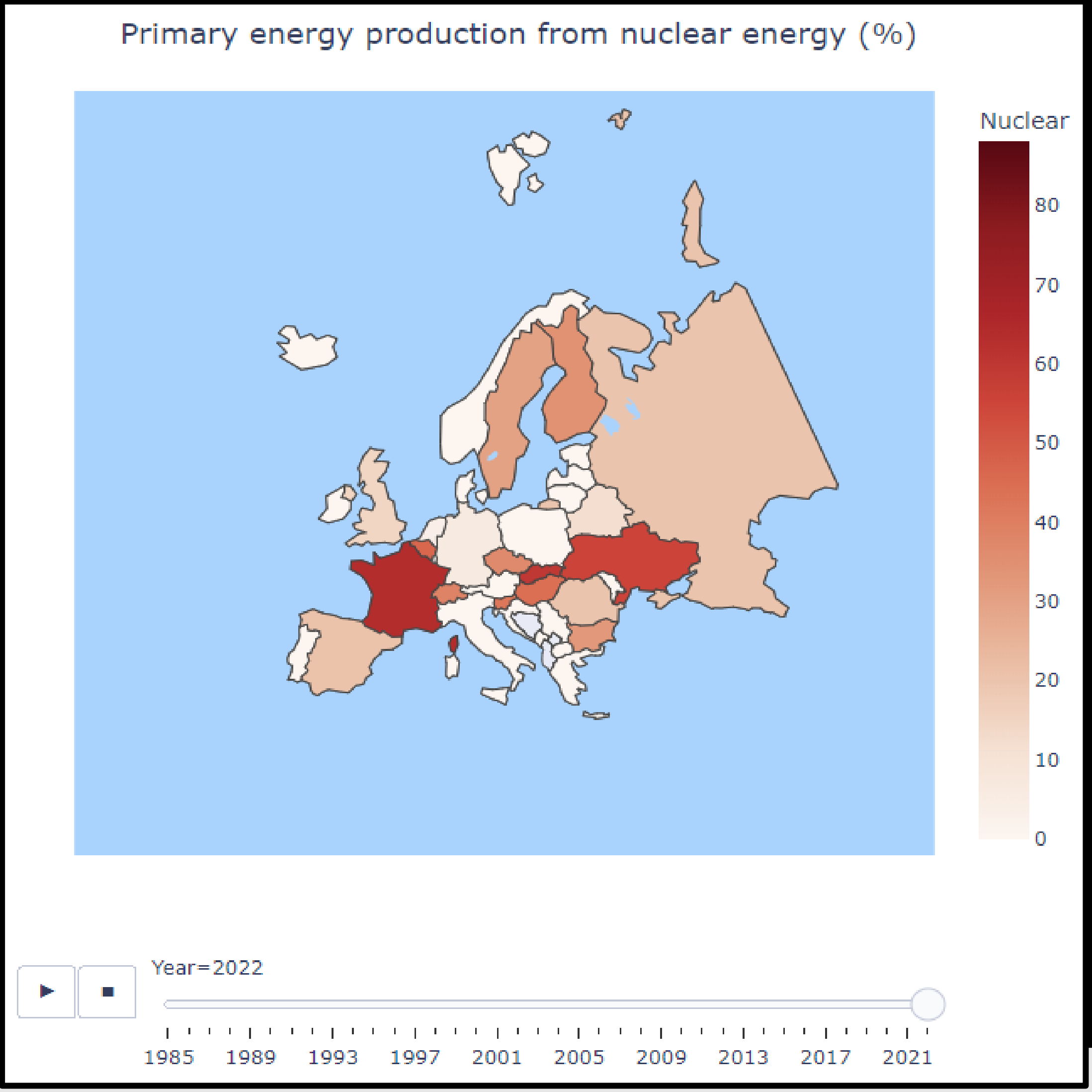
Renewables



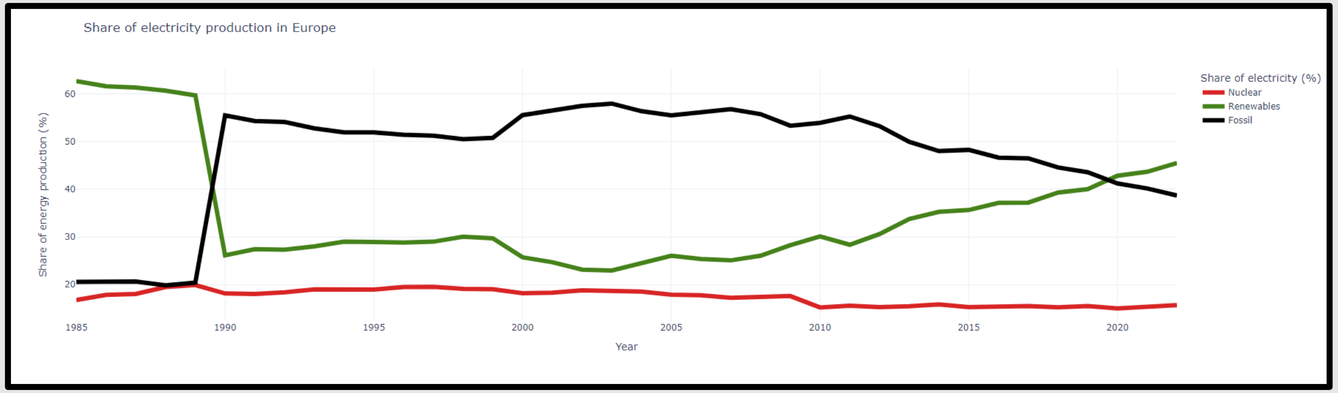


In **2022** the European energy mix comprises a combination of traditional and renewable sources, reflecting the growing commitment towards sustainability and a more resilient energy future.

Nuclear power holds a substantial 16.2% of the total share, contributing significantly to the overall electricity production.



Over the course of time, <u>nuclear power production in Europe has undergone dynamic changes</u>, reflecting the region's evolving energy landscape.
Nuclear energy production of was **not uniform** and consistent across the continent with <u>alternating periods of expansion and decline in each country</u>. In 2022, **France** is by far the country with the largest share of nuclear generated electricity, followed by **Slovakia** and **Ukraine**.



For the past 40 years the average percentage of nuclear power generated across Europe has been consistently **decreasing**.

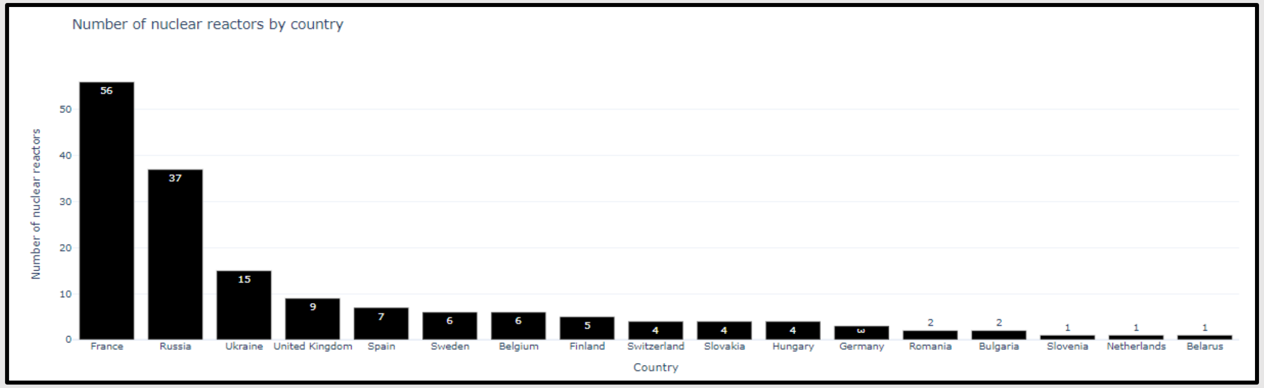
Nuclear power in Europe

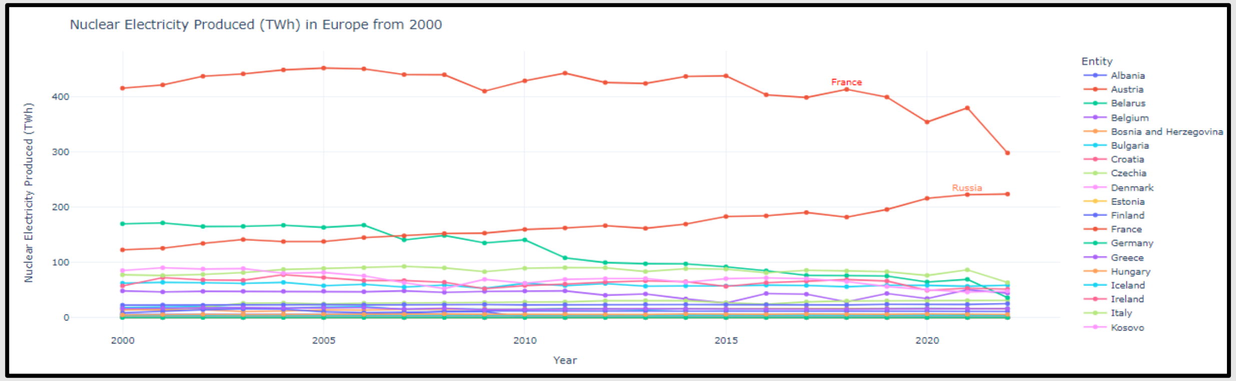
Dataset:
Number nuclear reactors

Source:	INTERNATIONAL ATOMIC ENERGY AGENCY, Nuclear Power Reactors in the World, Reference Data Series No. 2, IAEA, Vienna (2023).

Dataset:
Electricity generated by source

Source:	Ember - Yearly Electricity Data (2023); Ember - European Electricity Review (2022) – with major processing by Our World in Data. "Electricity generation from other renewables, excluding bioenergy" [dataset]. Ember, "Yearly Electricity Data"; Ember, "European Electricity Review" [original data].





Total of 170 nuclear reactors across Europe

France stands as a dominant player in Europe's nuclear landscape, hosting 56 active nuclear reactors, followed by Russia, Ukraine, and the United Kingdom.

France overrules Europe's generation of nuclear energy, consistently generating over 400 terawatt-hours before 2019.
Recent data indicates a notable decline in nuclear power production, with a nearly **21% fall** from 379 TWh to 297 TWh between 2021 and 2022.

Exploring air pollution in Europe

Importing the datasets

Dataset:
CO2, SOx, NOx and PM 2.5

Source: OECD (2024), Air and GHG emissions (indicator). doi: 10.1787/93d10cf7-en

Country Selection
Selecting European countries.

```
[181] # save excel containing the CODE and Country name used in the dataset
      Country_codes = pd.read_excel('/content/drive/My Drive/Python_project/OECD_dataset/Countries.xlsx')
      selected = ['CODE', 'Country']
      Country_codes = Country_codes[selected]
      # Rename the column
      Country_codes.rename(columns={'CODE': 'LOCATION'}, inplace=True)
```

Keeping only countries common to all datasets.

```
[222] # checking and selecting only those countries common to all datasets
      set1 = set(CO2['LOCATION'])
      set2 = set(NOx['LOCATION'])
      set3 = set(PM25['LOCATION'])
      set4 = set(SOx['LOCATION'])

      # get common elements
      common_elements = set1.intersection(set2, set3, set4)
```

```
[224] # saving each pollutant dataframe with only the selected columns
      CO2 = CO2[CO2['LOCATION'].isin(common_elements)]
      NOx = NOx[NOx['LOCATION'].isin(common_elements)]
      PM25 = PM25[PM25['LOCATION'].isin(common_elements)]
      SOx = SOx[SOx['LOCATION'].isin(common_elements)]
```

Cleaning the datasets

```
[182] CO2 = pd.read_csv('/content/drive/My Drive/Python_project/OECD_dataset/CO2.csv', delimiter=',')
```

```
[183] # Converting tonnes per capita to kilograms per capita
      CO2['Kilograms_per_capita'] = CO2['Value'] * 1000
```

```
[184] # Drop the specified columns
      columns_to_drop = ['INDICATOR', 'SUBJECT', 'Flag Codes', 'MEASURE', 'Value']
      CO2 = CO2.drop(columns= columns_to_drop)
```

```
[185] # Use map to substitute LOCATION code with Country name
      CO2['LOCATION'] = CO2['LOCATION'].map(Country_codes.set_index('LOCATION')['Country'])
      # adding continent column to the dataset
      CO2 = CO2.merge(cont, how='left', left_on='LOCATION', right_on='name').drop(columns=['name'])
```

```
[186] # check if there are still countries not yet assigned to a continent
      # using Pandas to identify unique values in the "Entity" column of the DataFrame shares where the "continent" column is NaN (missing or null)
      CO2.loc[pd.isna(CO2.continent),:].LOCATION.unique()
```

```
[187] # assigning the continent Europe to the detected european missing countries (Bosnia and Herzegovina, Czech Republic and Malta)
      CO2.loc[CO2.LOCATION=='Czech Republic','continent'] = 'Europe'
      CO2.loc[CO2.LOCATION=='Bosnia and Herzegovina','continent'] = 'Europe'
      CO2.loc[CO2.LOCATION=='Malta','continent'] = 'Europe'
```

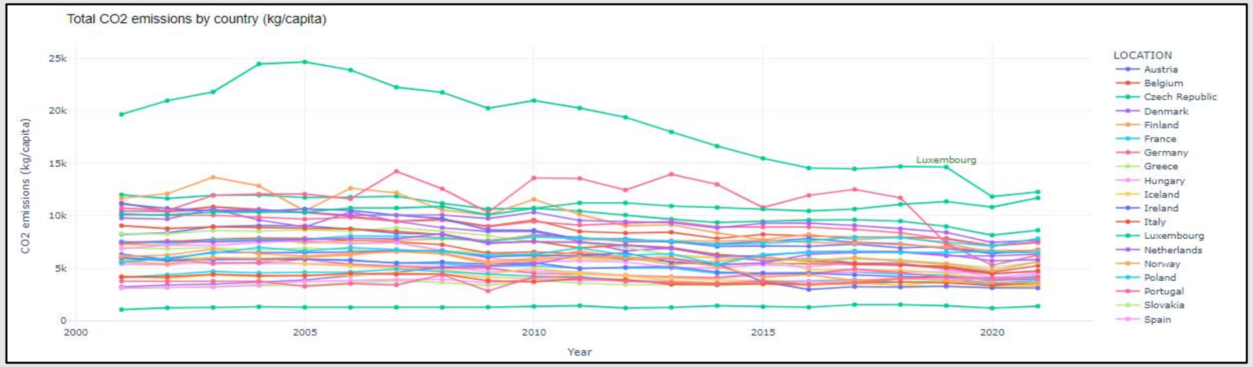
```
[188] # Keep only statistical units collected after the year 2000
      CO2 = CO2[CO2['TIME'] > 2000]
```

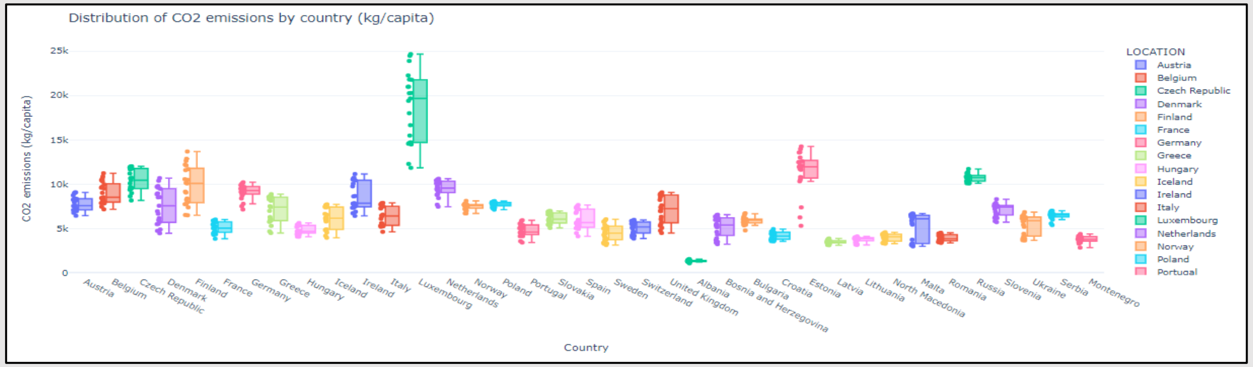
```
[189] # checking na values
      nan_values = CO2.isna().sum()
      nan_values
```

```
[191] # We initialize the SimpleImputer with the desired strategy to fill the NAs ('mean')
      imp = SimpleImputer(strategy='mean')
      # Group by 'LOCATION' and fill missing values with the mean of each group
      CO2['Kilograms_per_capita'] = CO2.groupby('LOCATION')['Kilograms_per_capita'].transform(lambda x: x.fillna(x.mean()))
```

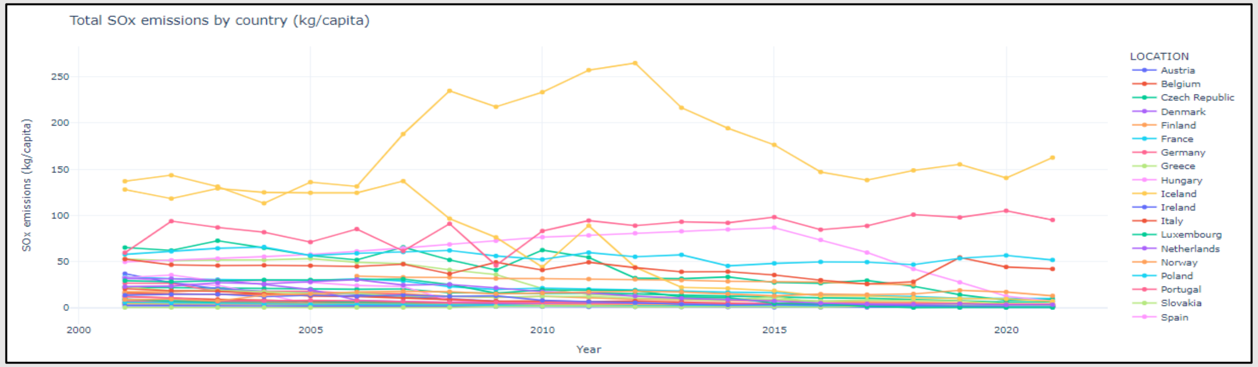
Exploratory analysis

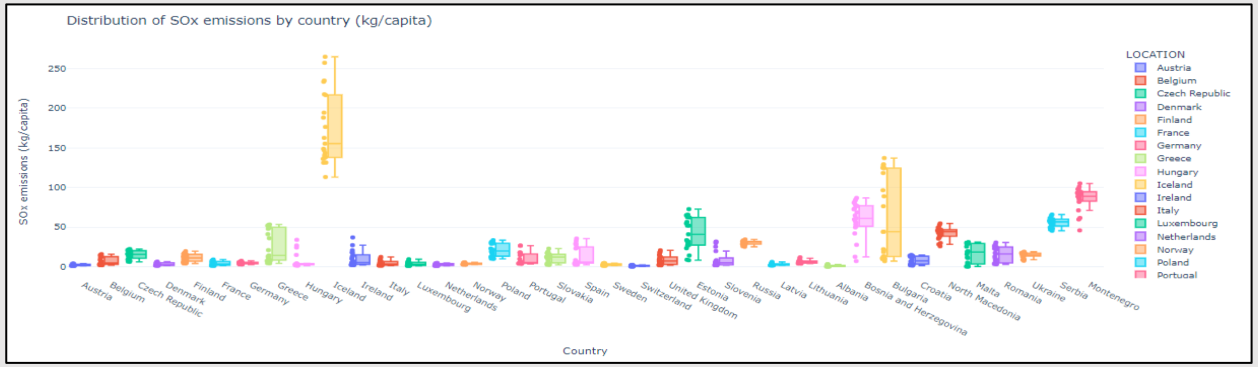
CO2





SOx



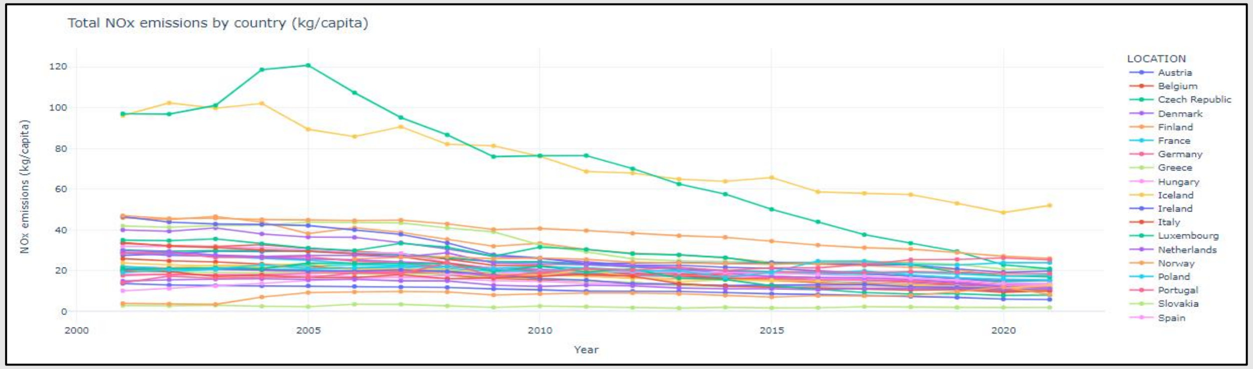


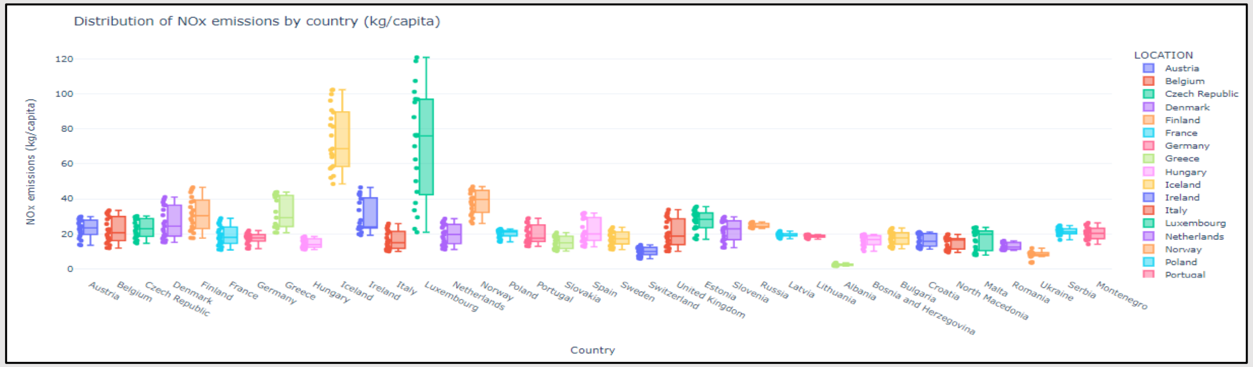
<u>Prior to 2020</u>, the levels of **CO2 kept decreasing** for most european countries. A <u>change of tendency has been registered for the few past years</u>, after 2 years of energy use oscillations, due to the impact of the Covid-19 pandemic and the war in Ukraine, as emissions started growing again.

<u>Since 2000</u>, the emissions of **sulfur oxides (SOx)** in Europe have predominantly followed a **decreasing pattern across most countries**.

Exploratory analysis

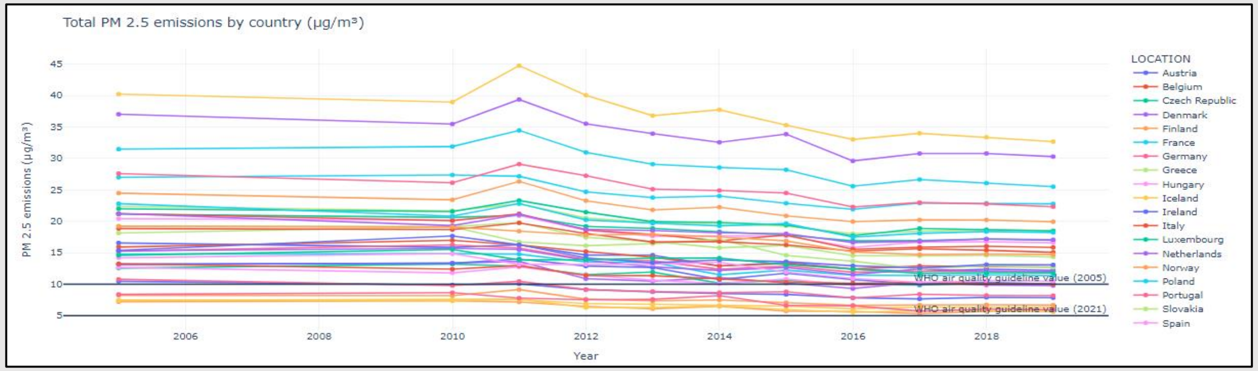
NOx

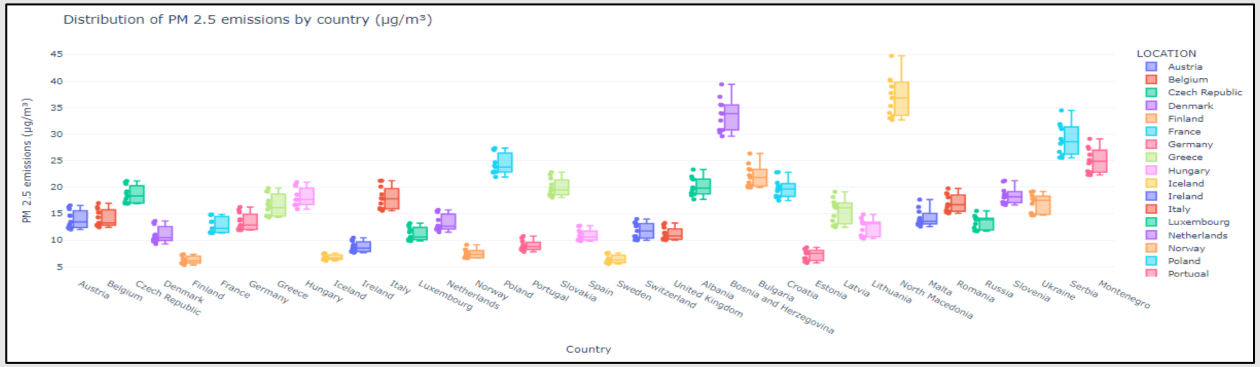




Compared to 2000, **NOx emissions decreased for most european countries**, excluding Iceland and Luxembourg.
Nonetheless, since 2005 Luxembourg has also delivered one of the steepest reductions in the Union, reducing its NOx emissions of 83% between 2005 and 2021.

PM 2.5





PM2.5 emission levels and intensities are also **decreasing in most european countries**.
Inhabitants of most countries are still exposed to levels exceeding the World Health Organization (WHO) air quality guideline value set at 10µg of PM2.5/m3 for 2005, while none of the countries is keeping its pollution levels lower the maximum level set in 2021.

Analyzing the relationship between nuclear energy and levels of air pollution

Preparing the dataframe for the analysis

1. Merging the dataset, combining the air pollution datasets (CO2, NOx, SOx, PM 2.5)

```
[317] # Merge datasets on the common keys 'Country' and 'Year'
      combined_electricity_pollution = pd.merge(electricity_mix, CO2, on=['LOCATION', 'TIME'], how='inner')
      combined_electricity_pollution = pd.merge(combined_electricity_pollution, SOx, on=['LOCATION', 'TIME'], how='inner')
      combined_electricity_pollution = pd.merge(combined_electricity_pollution, NOx, on=['LOCATION', 'TIME'], how='inner')
      combined_electricity_pollution = pd.merge(combined_electricity_pollution, PM25, on=['LOCATION', 'TIME'], how='inner')

[318] combined_electricity_pollution.rename(columns={'LOCATION': 'Country'}, inplace=True)
```

3. Creating dummy variable **Nuclear_reactor** (1 = countries with at least 1 nuclear reactor and 0 = no nuclear reactor)

```
[319] #Create variable with value 1 if the country has nuclear reactors, 0 otherwise
      combined_electricity_pollution['Nuclear_reactors'] = combined_electricity_pollution['Country'].isin(Number_reactors['Country']).astype(int)
```

4. Adding a column **Number_reactors** with the number of nuclear reactors by country

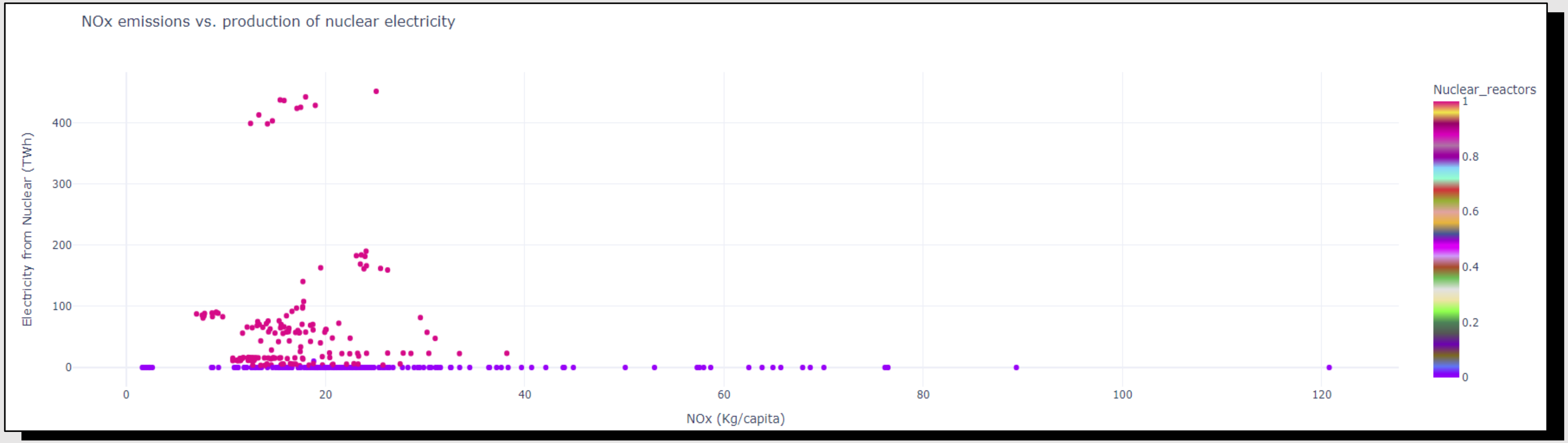
```
[320] # Merge the DataFrames based on the 'Country' column
      combined_electricity_pollution = pd.merge(combined_electricity_pollution, Number_reactors, on='Country', how='left')
```

```
[321] # Fill NaN values (countries not present in df_reactors) with 0
      combined_electricity_pollution['Num_reactors'].fillna(0, inplace=True)
```

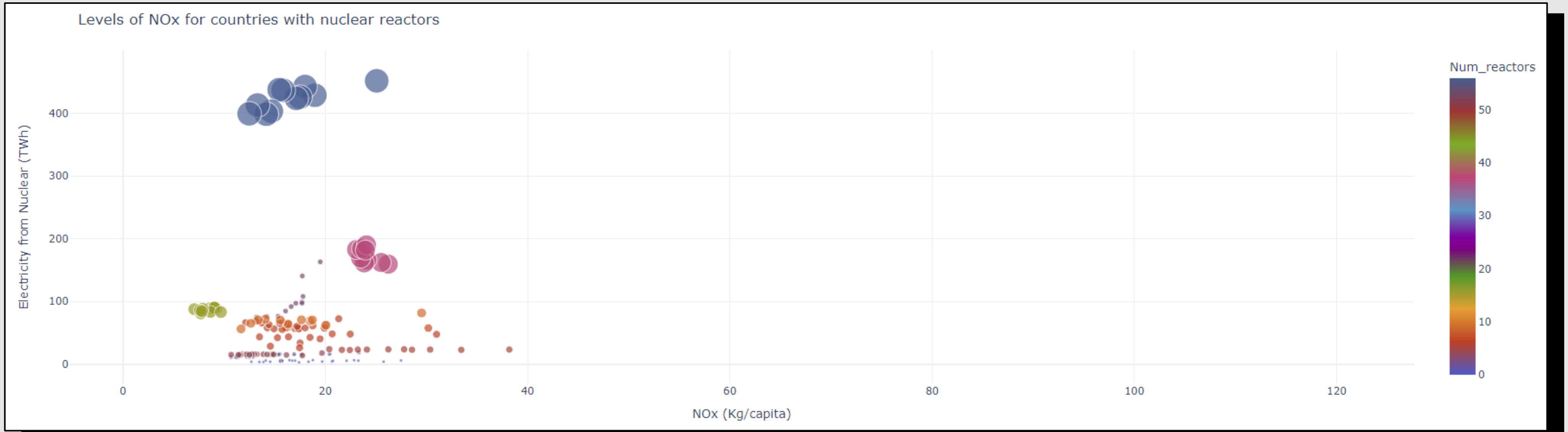
5. Graphically representing the relation between **Nuclear_reactor** by **air pollutant**

6. Graphically representing the relation between **Number_reactors** by **air pollutant**

NOx emissions vs. production of nuclear electricity

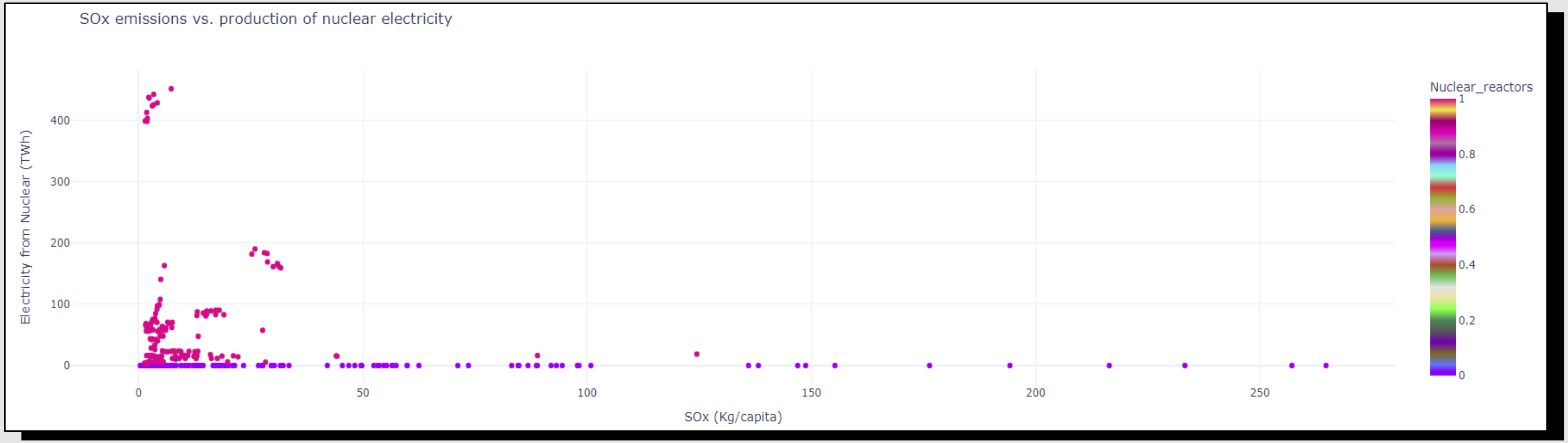


Considering NOx emissions, the distribution of non-nuclear countries takes higher values.

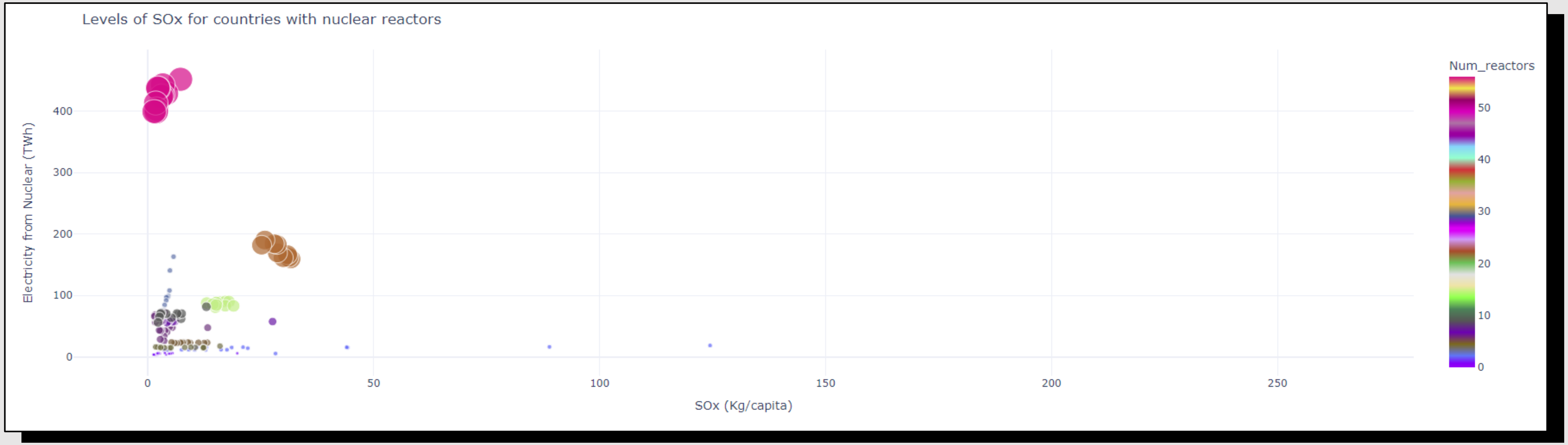


The pattern is supported again by the number of nuclear reactors, as we notice that France, Russia and Ukraine exhibit lower levels of NOx than reactor-free countries.

SOx emissions vs. production of nuclear electricity

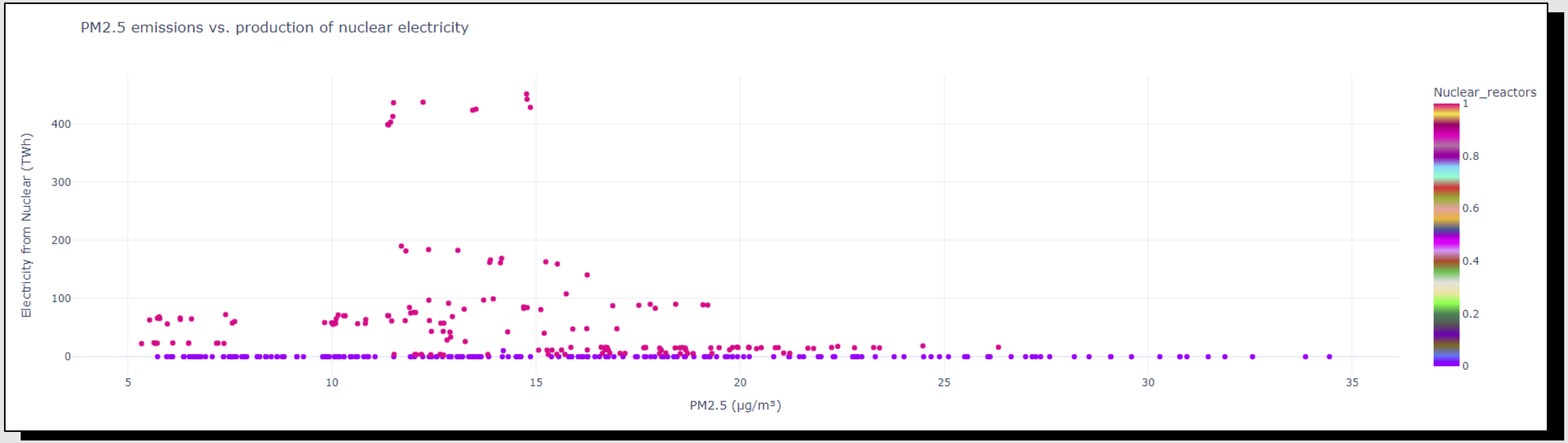


From the graph it can be observed how **countries not producing any nuclear energy** tend to be <u>associated with higher levels of pollution in terms of SOx</u>.

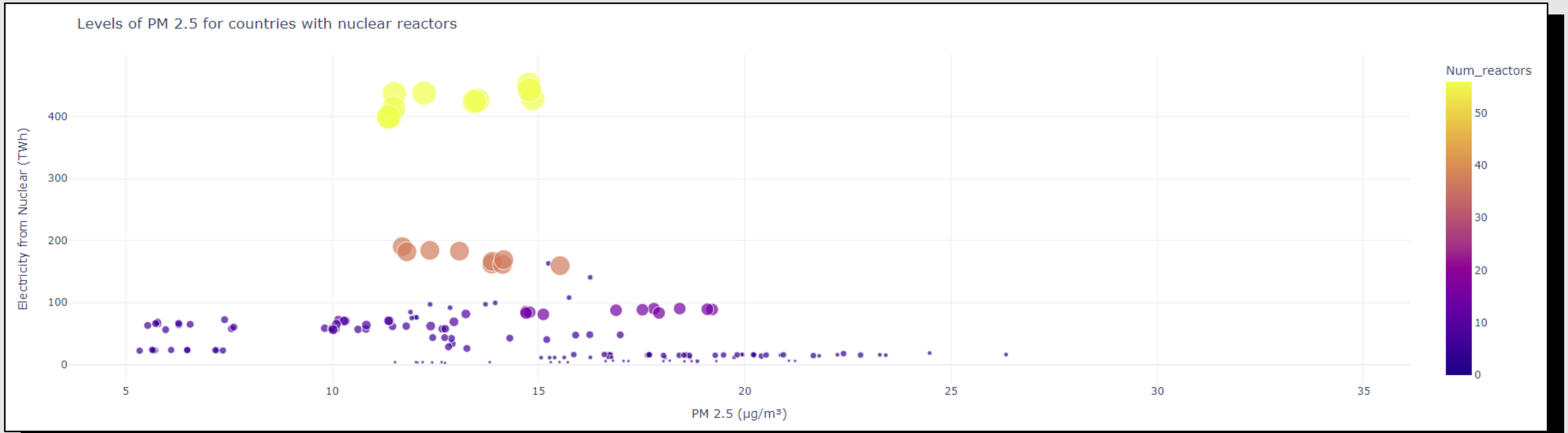


France and *Russia*, the countries holding the highest number of reactors, are <u>linked to lower emissions of SOx than non-nuclear countries</u>, as displayed in the previous graph.

PM 2.5 emissions vs. production of nuclear electricity

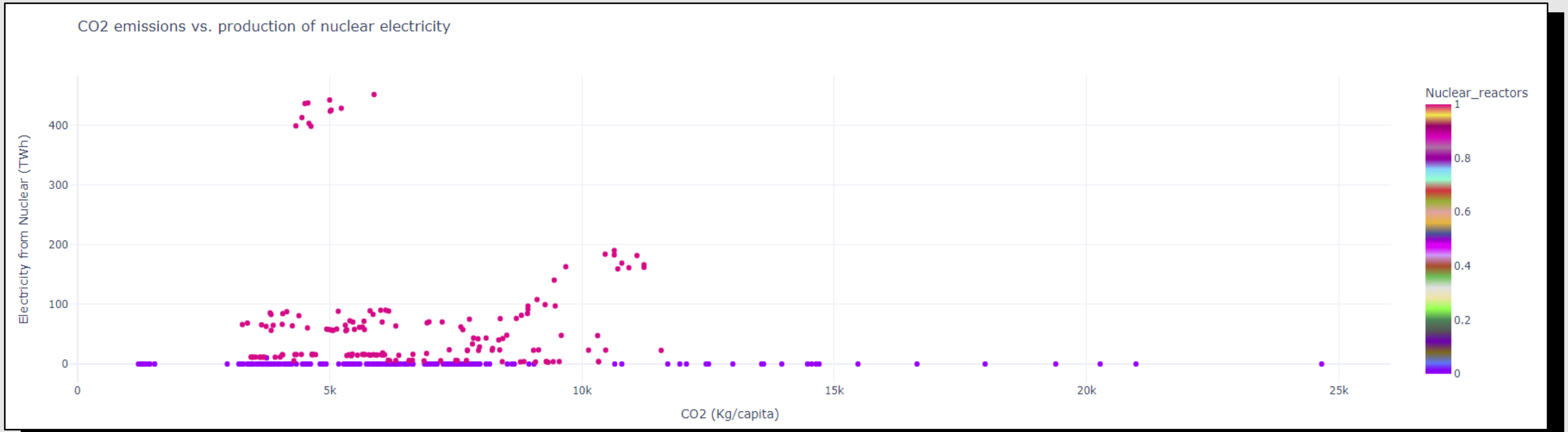


For PM 2.5 emissions the significance of the **relation** starts to lose its strength. The distribution of nuclear-countries cannot be easily set apart from the one of those nuclear-free.

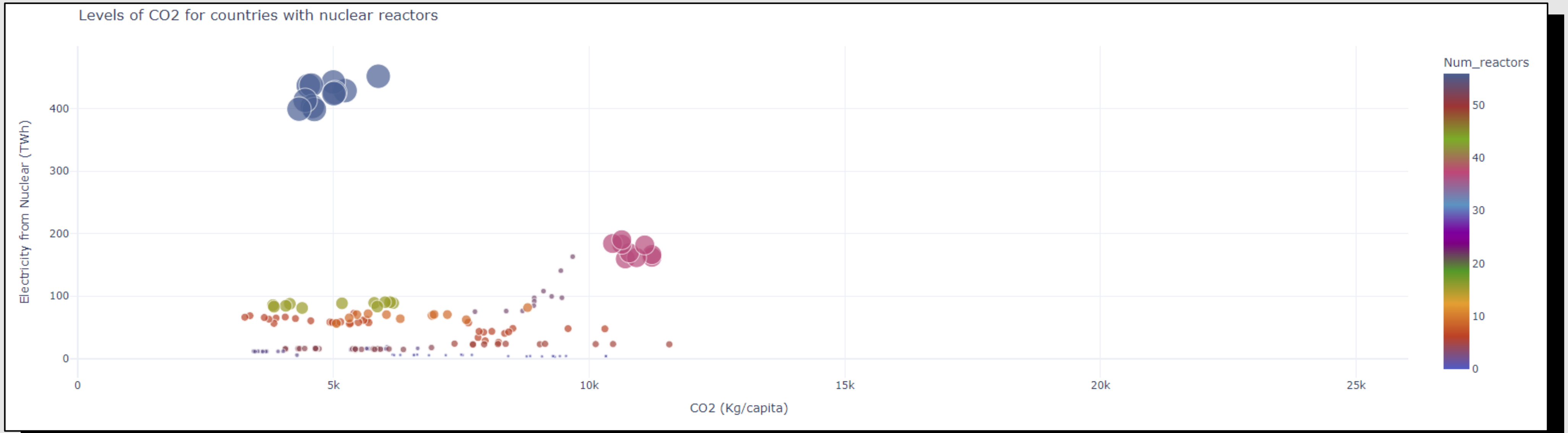


It may still be noticed that **countries with no available nuclear reactor** are associated with higher emissions levels, than those with at least 1.

CO2 emissions vs. production of nuclear electricity



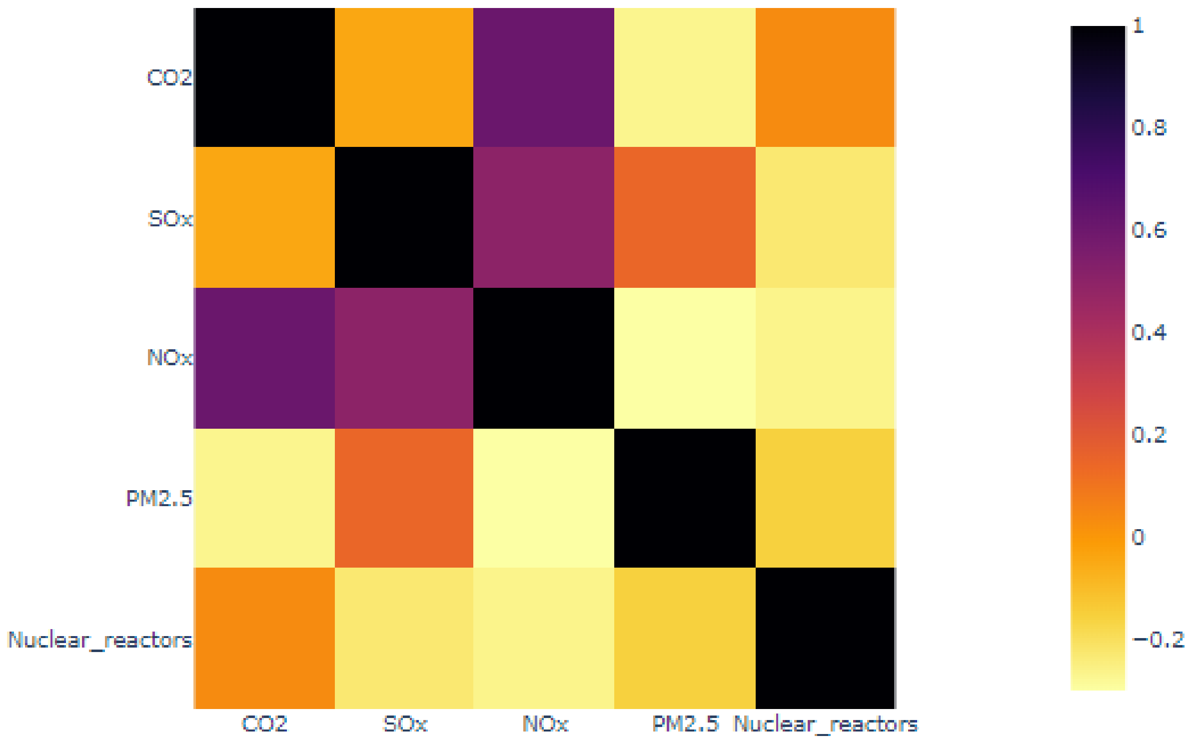
Somekind of pattern may still be observed for CO2 emissions, <u>but the correlation is not as significant as for the previously examined pollutants.</u>



As previously noted, even though not such strong difference can be pointed out, <u>France keeps showcasing lower levels than most european countries</u>.

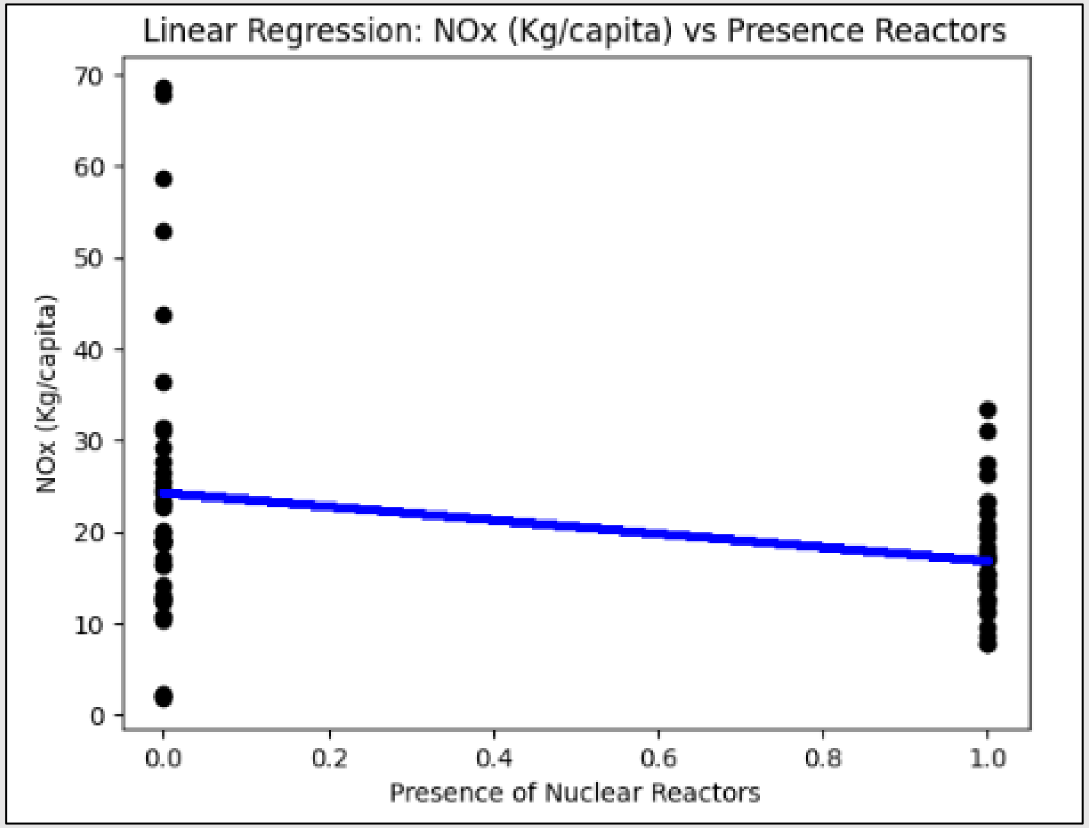
Heatmap Pollutants vs. Presence nuclear reactors

To further understand the kind of **correlation** existing between the presence of nuclear reactors and each air pollutant we decided to plot a heatmap.
The resulting correlation values support our previous analysis, since **NO_x**, **SO_x** and **PM 2.5** (respectively of -0.26 and -0.23, -0.15) are **more strongly negatively related** to the **availability of reactors** than CO_2 (0.034).

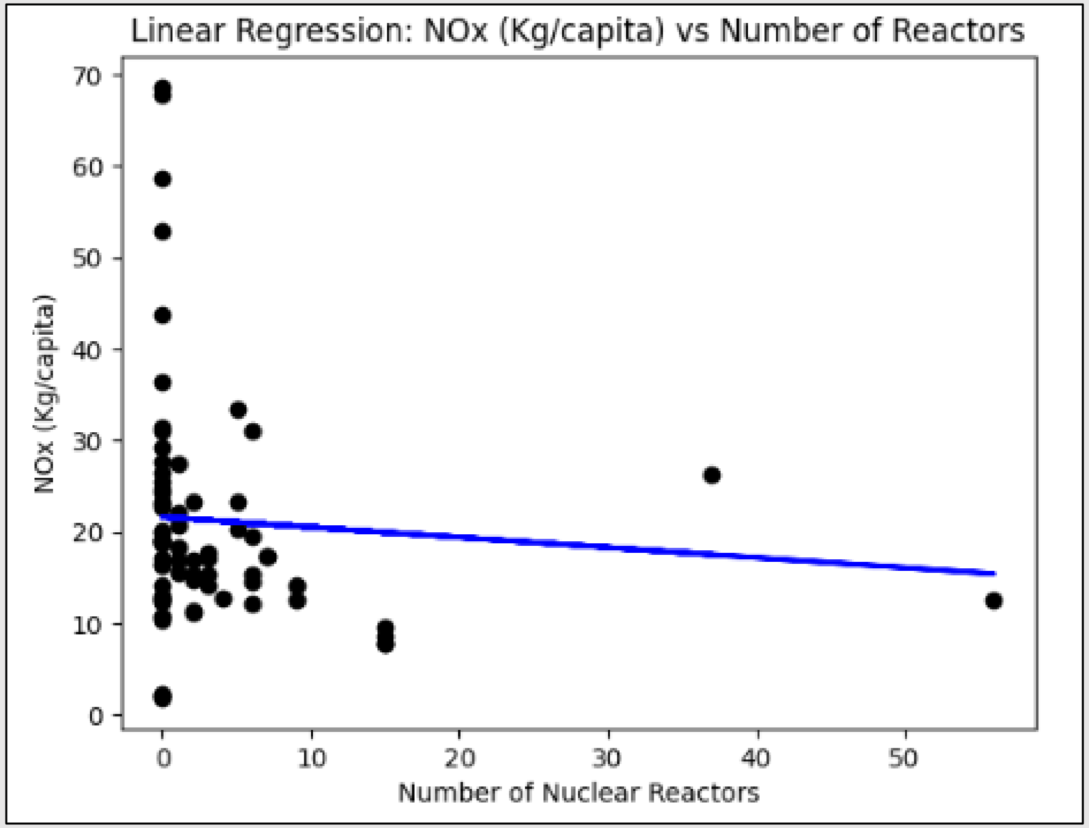


Linear regressions

NOx vs. Presence of reactors

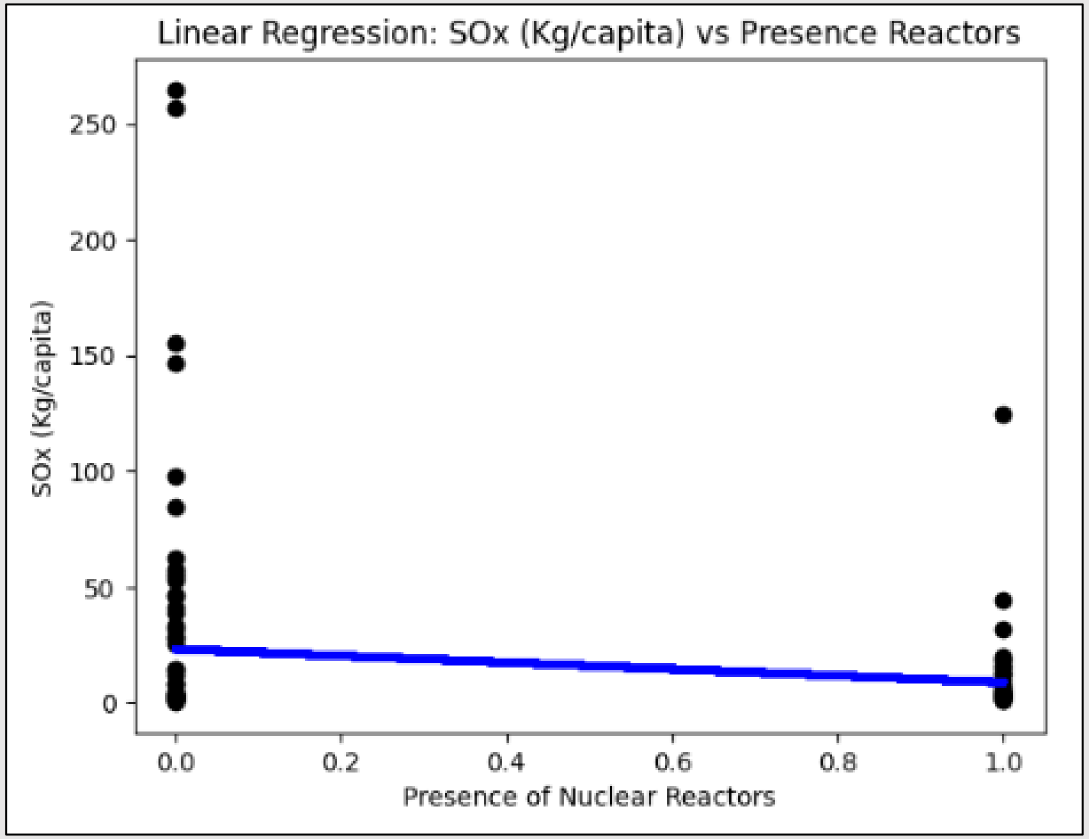


NOx vs. Number of reactors

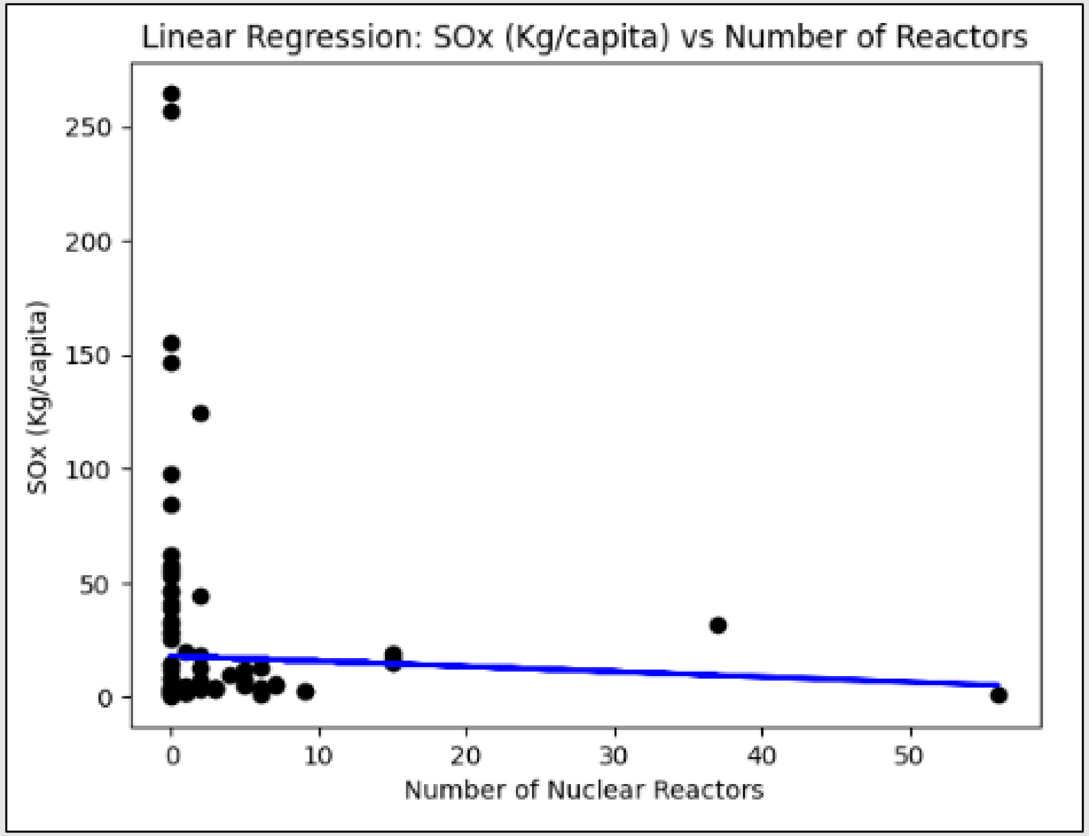


The **presence of nuclear reactors** suggests on average a lower value of NOx concentration. The same result can be deduced from the second graph, as the **slope** of the regression line is **negative**, meaning that a higher number of operative nuclear reactors is associated to a lower average level of NOx.

SOx vs. Presence of reactors

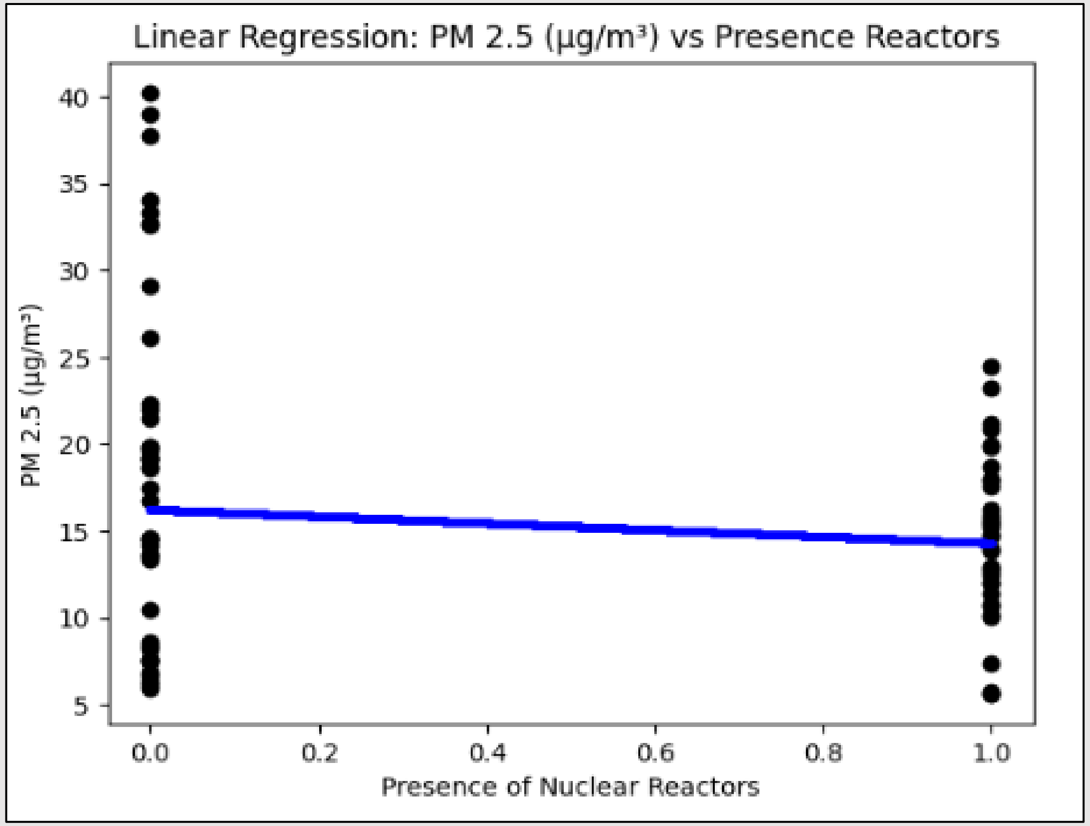


SOx vs. Number of reactors

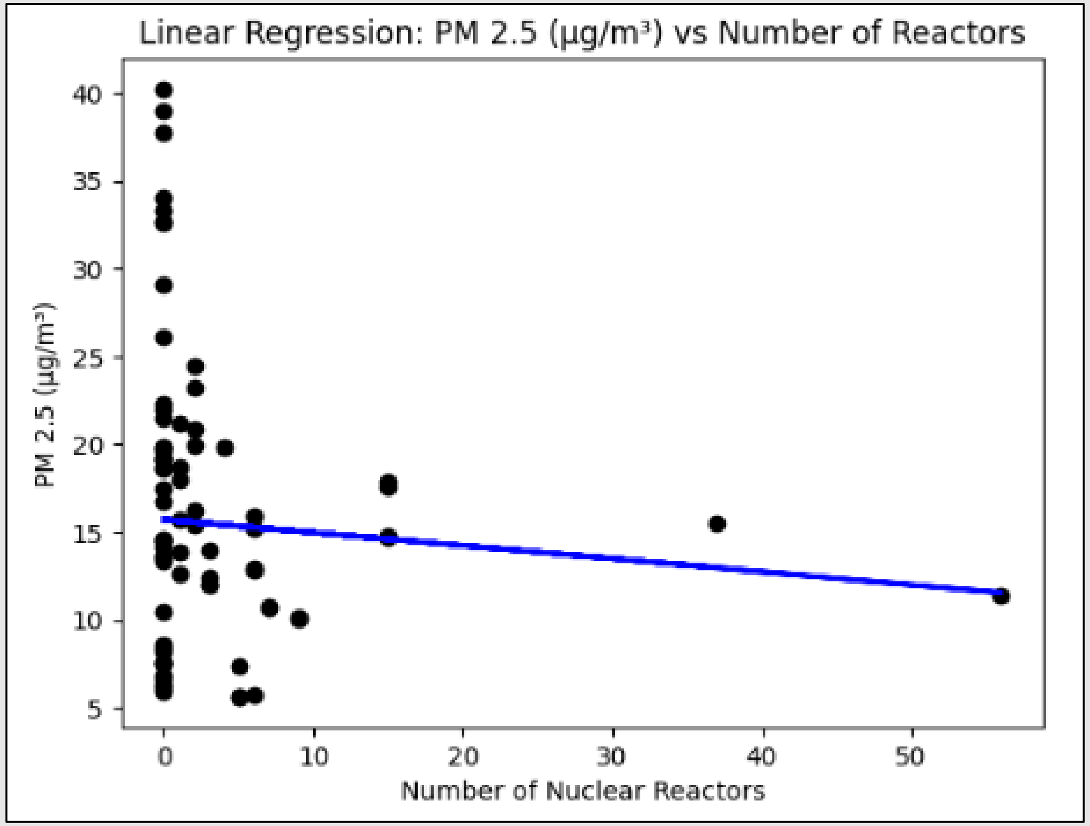


The initial observation of a negative correlation finds further support in the initial linear regression. The graph illustrates a clear trend, showcasing **lower sulfur oxides** (SOx) levels for countries with at least one nuclear reactor. The second regression analysis strengthens this connection, revealing that countries with a higher number of nuclear reactors tend to exhibit **lower levels of SOx emissions**.

PM 2.5 vs. Presence of reactors

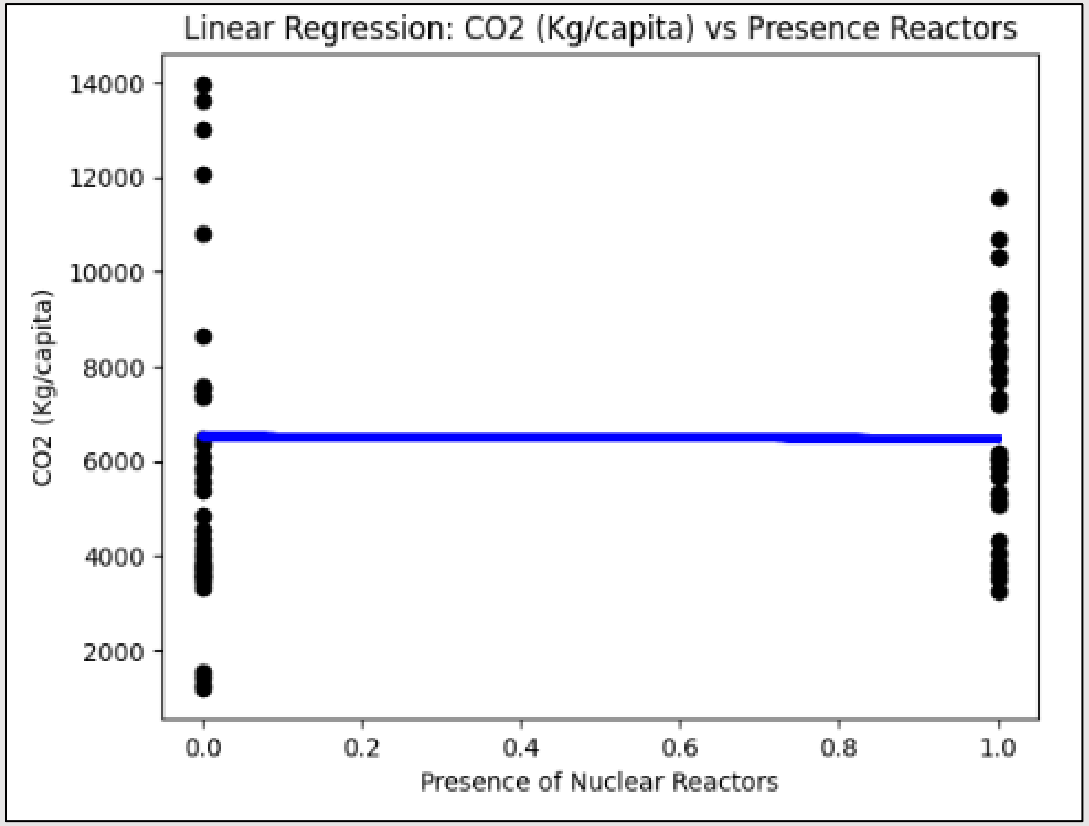


PM 2.5 vs. Number of reactors

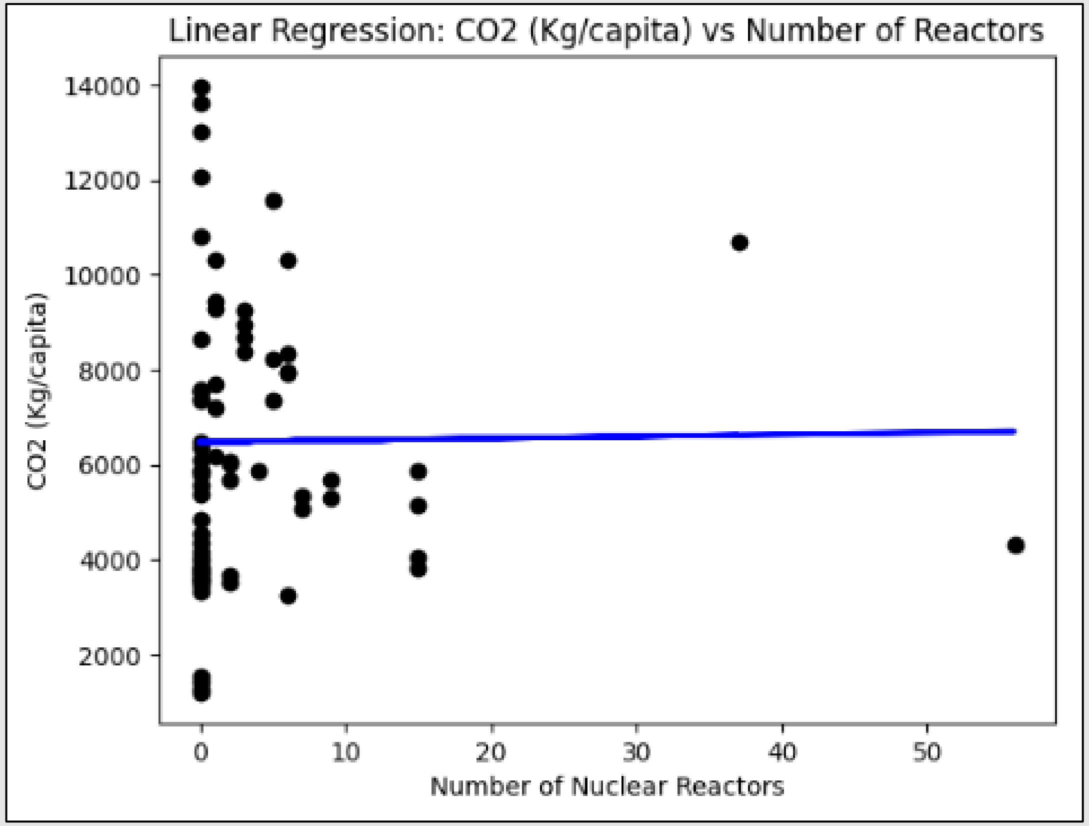


The outcomes derived from the linear regression analysis on PM25 support the previous exploratory analysis performed, by displaying a tendency for **PM2.5 levels to decrease** with the increasing number of nuclear reactors.

CO2 vs. Presence of reactors



CO2 vs. Number of reactors



The slopes of the regression lines, in both cases, are slightly positive, suggesting that the countries with an higher number of reactors seem to be linked to a greater concentration of CO_2. It's crucial to remember that correlation does not imply causation, meaning that the **observed positive correlation** might not necessarily mean that nuclear reactors directly cause increased CO_2.

Conclusion

Conclusions:

The linear regression analyses conducted on various air pollutants, specifically NOx, SOx, PM2.5 and CO2, did not reveal a clear and significant trend associating nuclear energy production with lower pollution levels.

The resulting research provided slightly more evident association between nuclear energy production and SOx, NOx and PM 2.5, while the regression considering CO2 did not provided the wished significance.

These findings suggest that nuclear energy production may not be a major contributing factor to air pollution, at least within the scope of the examined pollutants. Moreover, these findings highlight the importance of further research on the effects of nuclear power on air pollution levels, as well as the need for effective and evidence based air pollution policies.

We thank you for your attention.

Daniele Barbato
883661

Nicolae Bologa
901121

Alice Pantano
883502

Sergiu Nica
874965