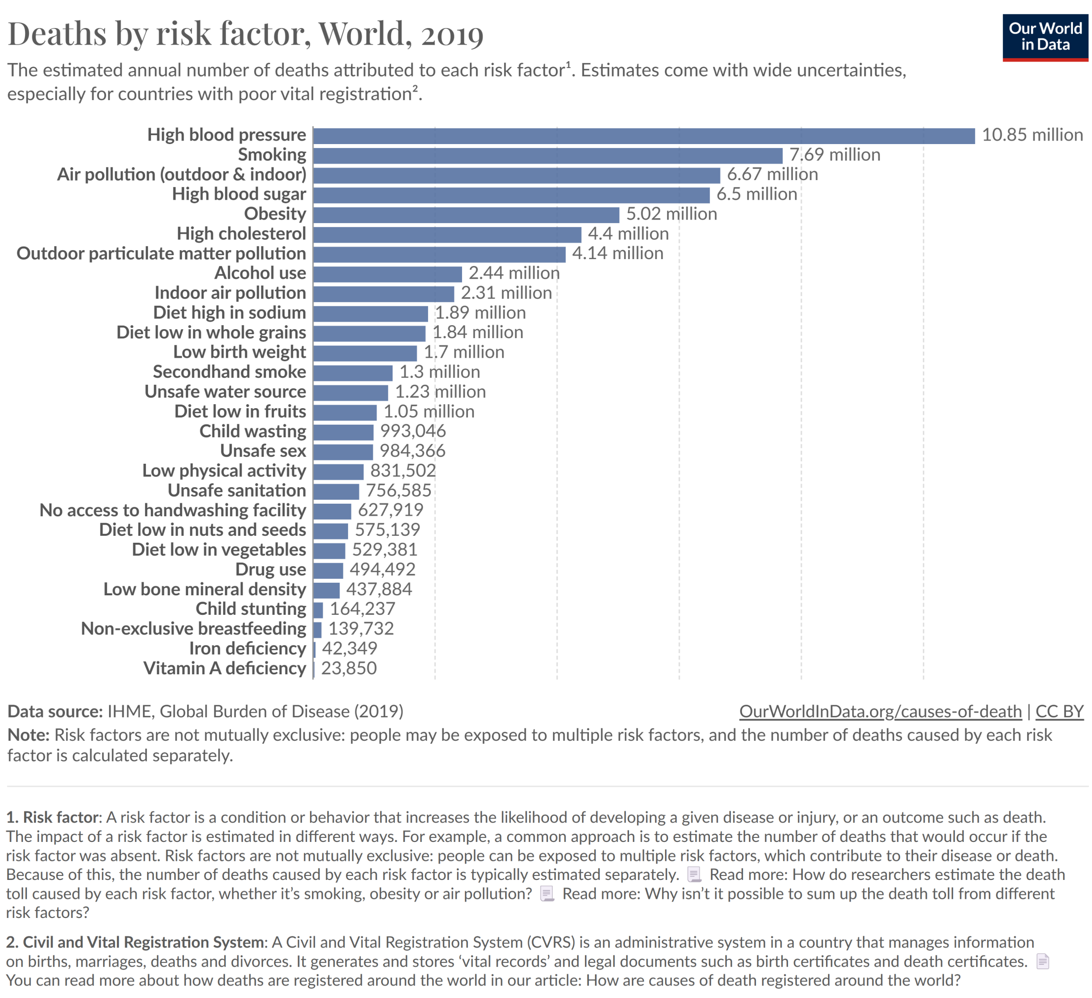
project

January 9, 2024

1 Problématique

Problématique: Étude de l'impact de la pollution de l'air sur les maladies respiratoires:

La qualité de l'air est un enjeu majeur de santé publique, notamment dans les zones urbaines où la concentration de polluants atmosphériques est souvent élevée. La pollution de l'air extérieur est l'un des principaux facteurs de risque de décès prématuré.



Parmi les conséquences néfastes de la pollution de l'air, son impact sur les maladies respiratoires suscite une préoccupation croissante à l'échelle mondialeante.

D'après les analyses précédentes de https://ourworldindata.org/, 7,2 % des décès dans le monde sont attribués à la pollution de l'air extérieur. Dans certains pays, elle est responsable d'un décès sur dix. Les taux de mortalité dus à la pollution de l'air extérieur varient d'un facteur 10 dans le monde. Les taux de mortalité sont généralement plus élevés dans les pays à revenus moyens. Globalement, et dans la plupart des pays, le nombre de décès dus à la pollution de l'air a augmenté.

La pollution de l'air est un enjeu mondial complexe et urgent, nécessitant une compréhension approfondie des facteurs qui contribuent à la variabilité des impacts sur la santé. Alors que des initiatives ont été lancées pour réduire les émissions de polluants dans certains pays, il est impératif de comprendre pourquoi la mortalité attribuée à la pollution de l'air persiste et augmente dans certaines régions, en particulier parmi les populations les plus vulnérables.

En regardant les données sur la pollution de l'air et les taux de maladies respiratoires, ce projet cherche à identifier des tendances, des corrélations et des disparités géographiques qui pourraient contribuer à une meilleure compréhension de cette problématique complexe.

2 Partie 1. Acquisition de données

2.1 1.1 données de la qualité de l'air (par ville)

2.2 #### a) Air quality data from WHO

- WHO Ambient Air quality database:

  Dans le fichier "../data/airquality/who/who_ambient_air_quality_database_version_2023_(v6.0).xlsx"

- Téléchargement: https://www.who.int/data/gho/data/themes/air-pollution/who-air-quality-database

[4]:
```
!pip install openpyxl
```

```
Requirement already satisfied: openpyxl in /opt/conda/lib/python3.11/site-
packages (3.1.2)
Requirement already satisfied: et-xmlfile in /opt/conda/lib/python3.11/site-
packages (from openpyxl) (1.1.0)
```

Les attributs de ce dataset:

[333]:
```
import pandas as pd
who_df = pd.read_excel("../data/airquality/who/
 ↪who_ambient_air_quality_database_version_2023_(v6.0).xlsx", sheet_name=2)
print(who_df.columns.values)
```

```
['who_region' 'iso3' 'country_name' 'city' 'year' 'version'
 'pm10_concentration' 'pm25_concentration' 'no2_concentration'
 'pm10_tempcov' 'pm25_tempcov' 'no2_tempcov' 'type_of_stations'
 'reference' 'web_link' 'population' 'population_source' 'latitude'
 'longitude' 'who_ms']
```

Le dataset nous offre les données sur les particules (PM2.5, PM10) et NO2 en années et en villes.

[207]:

```
who_df
```

[207]:

```
      who_region iso3  country_name       city  year  \
0         3_Sear  IND         India    Chennai  2018
1         3_Sear  IND         India    Solapur  2016
2         3_Sear  IND         India    Chennai  2019
3         3_Sear  IND         India  Hyderabad  2019
4         3_Sear  IND         India       Pune  2017
…              …    …             …          …     …
41359      5_Emr  SAU  Saudi Arabia      Jizan  2014
41360      5_Emr  SAU  Saudi Arabia      Jizan  2013
41361      5_Emr  SAU  Saudi Arabia      Jizan  2012
41362      5_Emr  SAU  Saudi Arabia      Jizan  2011
41363      5_Emr  SAU  Saudi Arabia      Jizan  2010

                            version  pm10_concentration  pm25_concentration  \
0                      version 2022                 NaN                30.0
1       version 2022, version 2018                 NaN                39.0
2                      version 2022                 NaN                39.0
3                      version 2022                 NaN                42.0
4                      version 2022                 NaN                43.0
…                                 …                   …                   …
41359                  version 2023               148.0                 NaN
41360                  version 2023               208.0                 NaN
41361                  version 2023               184.0                 NaN
41362                  version 2023               316.0                 NaN
41363                  version 2023               198.0                 NaN

       no2_concentration  pm10_tempcov  pm25_tempcov  no2_tempcov  \
0                    NaN           NaN          91.0          NaN
1                    NaN           NaN          99.0          NaN
2                    NaN           NaN          85.0          NaN
3                    NaN           NaN          87.0          NaN
4                    NaN           NaN           NaN          NaN
…                      …             …             …            …
41359                NaN           NaN           NaN          NaN
41360                NaN           NaN           NaN          NaN
41361                NaN           NaN           NaN          NaN
41362                NaN           NaN           NaN          NaN
41363                NaN           NaN           NaN          NaN

      type_of_stations                                         reference  \
0                  NaN   U.S. Department of State, United States Enviro…
1                  NaN   Central Pollution Control Board India, Environ…
2                  NaN   U.S. Department of State, United States Enviro…
3                  NaN   U.S. Department of State, United States Enviro…
4                  NaN   Central Pollution Control Board India, Environ…
```

```
…                 …                                                 …
41359              NaN    Ministry of Environment, Water, and Agriculture
41360              NaN    Ministry of Environment, Water, and Agriculture
41361              NaN    Ministry of Environment, Water, and Agriculture
41362              NaN    Ministry of Environment, Water, and Agriculture
41363              NaN    Ministry of Environment, Water, and Agriculture

                                                     web_link  population  \
0      https://www.airnow.gov/index.cfm?action=airnow…   9890427.0
1                                                         NaN    985568.0
2      [[["EPA AirNow DOS","http://airnow.gov/index.c…   9890427.0
3      [[["EPA AirNow DOS","http://airnow.gov/index.c…   8943523.0
4                            http://www.cpcb.gov.in/CAAQM/   5727530.0
…                                                       …          …
41359                                                     NaN    127743.0
41360                                                     NaN    127743.0
41361                                                     NaN    127743.0
41362                                                     NaN    127743.0
41363                                                     NaN    127743.0

       population_source   latitude  longitude  who_ms
0                    NaN  13.087840  80.278470       1
1                    NaN  17.659919  75.906391       1
2                    NaN  13.087840  80.278470       1
3                    NaN  17.384050  78.456360       1
4                    NaN  18.505320  73.823839       1
…                   …         …          …      …
41359                NaN  16.885875  42.573386       1
41360                NaN  16.885875  42.573386       1
41361                NaN  16.885875  42.573386       1
41362                NaN  16.885875  42.573386       1
41363                NaN  16.885875  42.573386       1

[41364 rows x 20 columns]
```

[201]:
```
who_df_nan_percentage = who_df.dropna(subset=['city', 'year'], how='any').iloc[:
 ↪,6:12].isna().mean() * 100
print("Show percentage of NaN values for the air pollution attributes:\n----")
print(who_df_nan_percentage)
```

```
Show percentage of NaN values for the air pollution attributes:
----
pm10_concentration    31.855425
pm25_concentration    47.864346
no2_concentration     35.469795
pm10_tempcov          48.341899
pm25_tempcov          61.024920
```

```
no2_tempcov          44.419665
dtype: float64
```

Visualisation des données manquantes:

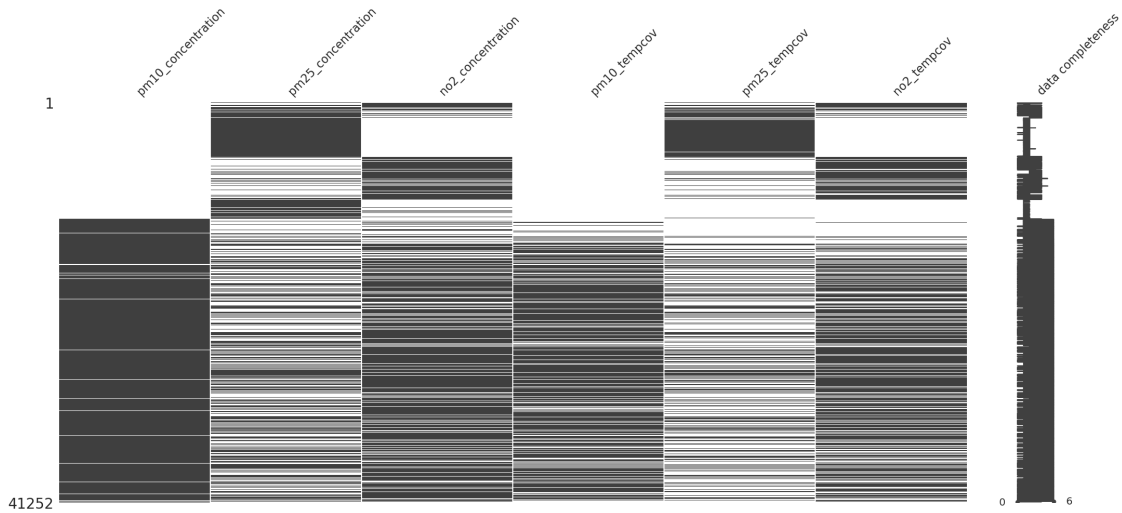
[194]:
```
!pip install missingno
```

```
Collecting missingno
  Downloading missingno-0.5.2-py3-none-any.whl (8.7 kB)
Requirement already satisfied: numpy in /opt/conda/lib/python3.11/site-packages
(from missingno) (1.24.4)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.11/site-
packages (from missingno) (3.8.0)
Requirement already satisfied: scipy in /opt/conda/lib/python3.11/site-packages
(from missingno) (1.11.3)
Requirement already satisfied: seaborn in /opt/conda/lib/python3.11/site-
packages (from missingno) (0.13.0)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->missingno) (1.1.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.11/site-
packages (from matplotlib->missingno) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->missingno) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->missingno) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->missingno) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.11/site-
packages (from matplotlib->missingno) (10.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->missingno) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.11/site-
packages (from seaborn->missingno) (2.0.3)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.2->seaborn->missingno) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.2->seaborn->missingno) (2023.3)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.11/site-
packages (from python-dateutil>=2.7->matplotlib->missingno) (1.16.0)
Installing collected packages: missingno
Successfully installed missingno-0.5.2
```

[335]:
```
import missingno as msno
fig = msno.matrix(who_df.dropna(subset=['city', 'year'], how='any').iloc[:,6:
 ↪12], labels=True)
fig_copy = fig.get_figure()
```

```
fig_copy.savefig('../fig/who_msno.png', bbox_inches = 'tight')
fig
```

[335]: <Axes: >



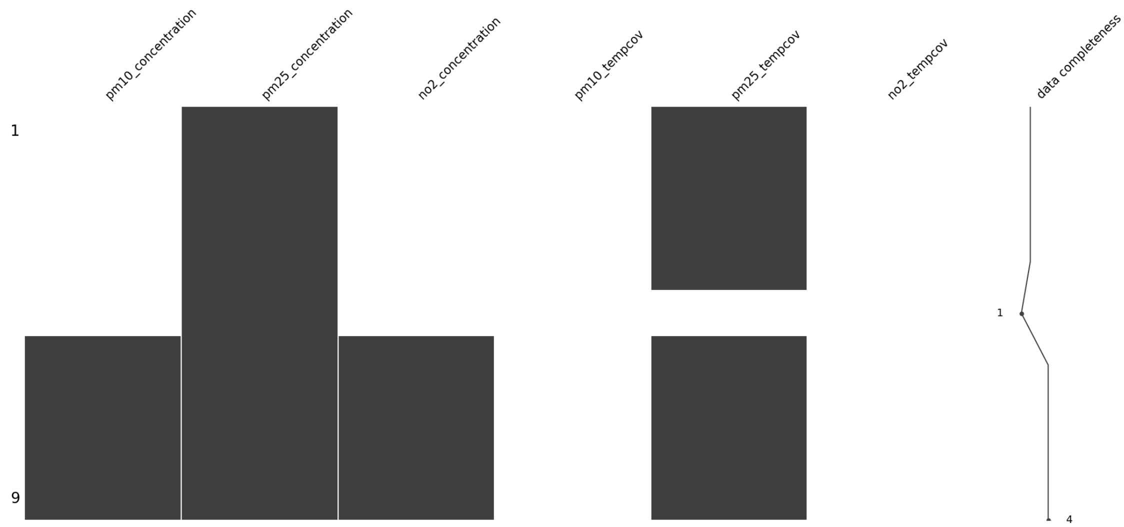
[105]:
```
df_who_sh = who_df[who_df['city']=='Shanghai']
who_df_nan_percentage_sh = df_who_sh.dropna(subset=['year'], how='any').iloc[:
↪,6:12].isna().mean() * 100
print("Show percentage of NaN values for the air pollution attributes in␣
↪Shanghai:\n----")
print(who_df_nan_percentage_sh)
```

```
Show percentage of NaN values for the air pollution attributes in Shanghai:
----
pm10_concentration     55.555556
pm25_concentration      0.000000
no2_concentration      55.555556
pm10_tempcov          100.000000
pm25_tempcov           11.111111
no2_tempcov           100.000000
dtype: float64
```

Visualisation des données manquantes pour Shanghai:

[206]:
```
msno.matrix(who_df[who_df['city']=='Shanghai'].dropna(subset=['year'],␣
↪how='any').iloc[:,6:12], labels=True)
```

[206]: <Axes: >



2.3 #### b) Air quality data from the World Air Quality Index project (WAQI: aqicn.org)

Le WAQI est un projet à but non lucratif lancé en 2007. Sa mission est de sensibiliser les citoyens à la pollution de l'air et de fournir des informations unifiées et mondiales sur la qualité de l'air.

Toutes les données sur la qualité de l'air affichées sur l'Indice mondial de la qualité de l'air sont les données officielles de l'Agence de protection de l'environnement (EPA) de chaque pays.

La liste complète des sources de l'EPA utilisée: https://aqicn.org/sources/fr/

L'indice américain EPA a été choisi pour harmoniser les données.

Ce site (https://aqicn.org/historica permet de télécharger les données de sur la qualité de l'air depuis il y a 121 mois en fonction du nom de la ville.

[32]:
```
aqicn_path = "../data/airquality/aqicn/"

def get_aqicn_city(city):
    path = aqicn_path + city + ".csv"
    return pd.read_csv(path)
```

[33]:
```
df_aqicn_sh = get_aqicn_city("shanghai")
df_aqicn_sh.columns.values
```

[33]:
```
array(['date', ' pm25', ' pm10', ' o3', ' no2', ' so2', ' co'],
      dtype=object)
```

Le dataset fournit les données quotidiennes des attributs suivants: AQI de PM2.5, PM10, O3, NO2, SO2, CO. (Il ne donne pas la valeur de concentration, mais s'agit plutôt de la valeur convertie de l'IQA pour chaque polluant.)

On présente ici quelques lignes de données de la ville de Shanghai:

[34]:
```
df_aqicn_sh.columns = ['date', 'PM2.5', 'PM10', 'O3', 'NO2', 'SO2', 'CO']
df_aqicn_sh['date'] = pd.to_datetime(df_aqicn_sh['date'])
for polluant in ['PM2.5', 'PM10', 'O3', 'NO2', 'SO2', 'CO']:
    df_aqicn_sh[polluant]=  pd.to_numeric(df_aqicn_sh[polluant],errors='coerce')
df_aqicn_sh
```

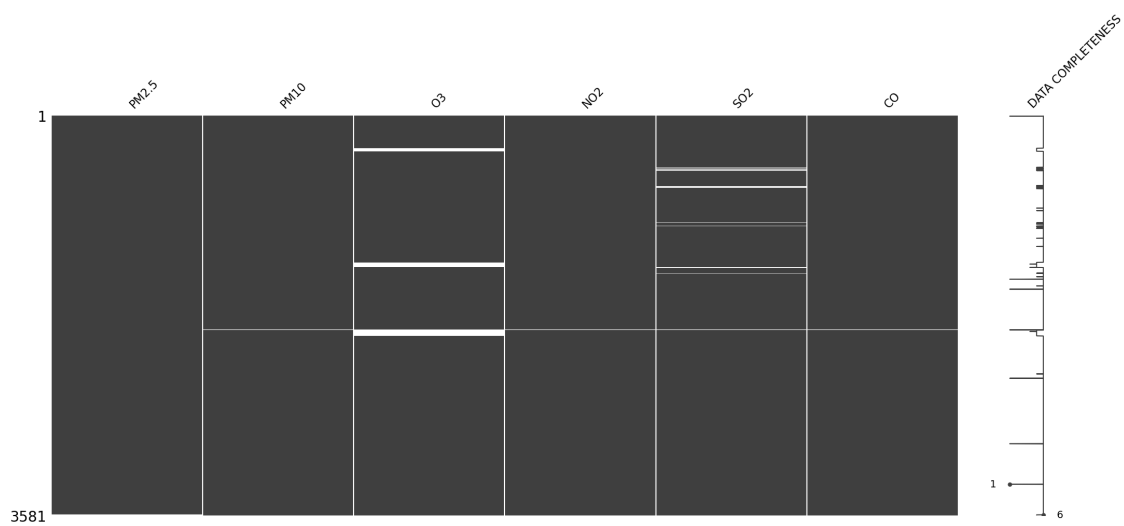
[34]:

	date	PM2.5	PM10	O3	NO2	SO2	CO
0	2024-01-01	151.0	55.0	28.0	23.0	3.0	7.0
1	2024-01-02	113.0	76.0	28.0	23.0	3.0	10.0
2	2024-01-03	165.0	92.0	39.0	34.0	4.0	12.0
3	2024-01-04	179.0	75.0	25.0	38.0	6.0	10.0
4	2024-01-05	165.0	109.0	37.0	45.0	5.0	14.0
…	…	…	…	…	…	…	…
3576	2018-12-31	NaN	34.0	26.0	13.0	4.0	3.0
3577	2017-09-10	NaN	26.0	33.0	16.0	3.0	9.0
3578	2016-03-13	NaN	61.0	51.0	13.0	8.0	7.0
3579	2014-12-31	NaN	55.0	24.0	19.0	15.0	6.0
3580	2013-12-31	NaN	121.0	29.0	57.0	30.0	14.0

[3581 rows x 7 columns]

[218]:
```
msno.matrix(df_aqicn_sh.iloc[:,1:7], labels=True)
```

[218]: <Axes: >



2.4 #### c) Air quality data from berkeleyearth

Ce site (https://berkeleyearth.org/air-quality-location/) permet de télécharger les données horaires passées de la concentration de PM2.5 (g/m³) depuis 2013 selon le nom de ville ou pays.

[1]:
```
# city="Shanghai"

from pyspark import SparkConf, SparkContext
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import *
import pandas as pd
import time
from operator import add

berkeleyearth_path = "../data/airquality/berkeleyearth/"
spark_be = SparkSession.builder.master("local[10]").config("spark.driver.
 ↪memory", "15g")\
.appName("berkeley_earth").getOrCreate()
# spark = SparkSession.builder \
#   .master("local[10]") \
#   .config("spark.driver.memory", "15g") \
#   .appName("musique") \
#   .getOrCreate()
def row_to_dataframe(record):
    return pd.DataFrame([record], columns=fieldnames)

def toCsv_berkeleyearth_by_city(city):
    path = berkeleyearth_path + city + ".txt"
    csv_path = berkeleyearth_path + city + ".csv"
    # sc = spark_be.sparkContext
    # ac = sc.textFile(path, minPartitions=4, use_unicode=True).map(lambda␣
 ↪element: element.split("\t"))
    # # print(ac.count())
    # # print(ac.zipWithIndex().take(10))
    # # remove first 7 lines
    # ac = ac.zipWithIndex().filter(lambda row: row[1] > 7).map(lambda row:␣
 ↪row[0])
    # ac_pd = ac.map(row_to_dataframe)
    # ac_df = ac_pd.toDF().toPandas()
    fieldnames = ['Year', 'Month', 'Day', 'UTC Hour', 'PM2.5', 'PM10_mask',␣
 ↪'Retrospective']

    data=pd.read_csv(path, sep='\t', header=None, names=fieldnames, skiprows=8)
    # data["City"] = city
    data.to_csv(csv_path, index=False)
    # return ac_pd
    return data
```

```
toCsv_berkeleyearth_by_city("Shanghai")
```

[1]:

```
       Year  Month  Day  UTC Hour  PM2.5  PM10_mask  Retrospective
0      2014      5   16         9  55.69        0.0              1
1      2014      5   16        10  57.35        0.0              1
2      2014      5   16        11  56.46        0.0              1
3      2014      5   16        12  57.41        0.0              1
4      2014      5   16        13  60.16        0.0              1
…      …      …    …         …     …          …              …
80293  2023      7    6        14  31.40        0.0              0
80294  2023      7    6        14  31.40        0.0              0
80295  2023      7    6        14  31.40        0.0              0
80296  2023      7    6        14  31.40        0.0              0
80297  2023      7    6        14  31.40        0.0              0

[80298 rows x 7 columns]
```

On vois qu'il y a des répétitions de lignes bizarres dans les dernières lignes.

Ici, seule la colonne de la concentration PM2.5 contient des données valides sur la pollution.

[10]:

```
df_berkeley_sh = pd.read_csv("../data/airquality/berkeleyearth/Shanghai.csv") \
.drop_duplicates()
df_berkeley_sh['date'] = pd.to_datetime(df_berkeley_sh[['Year', 'Month',␣
↪'Day']])
df_berkeley_sh = df_berkeley_sh.drop(columns=['Year', 'Month', 'Day',␣
↪'PM10_mask'])
df_berkeley_sh
```

[10]:

```
       UTC Hour  PM2.5  Retrospective        date
0             9  55.69              1  2014-05-16
1            10  57.35              1  2014-05-16
2            11  56.46              1  2014-05-16
3            12  57.41              1  2014-05-16
4            13  60.16              1  2014-05-16
…           …     …              …           …
77376        11  25.37              0  2023-07-06
77377        12  25.37              0  2023-07-06
77378        13  24.62              0  2023-07-06
77379        14  24.69              0  2023-07-06
77380        14  31.40              0  2023-07-06

[77380 rows x 4 columns]
```

[30]:

```
print(df_berkeley_sh[df_berkeley_sh['Retrospective']==1].count())
print(df_berkeley_sh[df_berkeley_sh['Retrospective']==1])
print(df_berkeley_sh.isna().mean())
```

```
UTC Hour          33676
PM2.5             33676
Retrospective     33676
date              33676
dtype: int64
       UTC Hour  PM2.5  Retrospective       date
0             9  55.69              1 2014-05-16
1            10  57.35              1 2014-05-16
2            11  56.46              1 2014-05-16
3            12  57.41              1 2014-05-16
4            13  60.16              1 2014-05-16
…         …    …            …         …
33672        20  87.76              1 2018-04-30
33673        21  82.18              1 2018-04-30
33674        22  75.16              1 2018-04-30
33675        23  73.79              1 2018-04-30
33676         0  81.03              1 2018-05-01

[33676 rows x 4 columns]
UTC Hour          0.0
PM2.5             0.0
Retrospective     0.0
date              0.0
dtype: float64
```

2.5 #### d) Données de la qualité de l'air des ville en Chine: aqistudy

Ce cite (https://www.aqistudy.cn/historydata/) fournit des données moyennes quoditiennes de l'IQA, PM2.5($\mu g/m^3$), PM10, CO, NO2, SO2, O3 depuis 2014 pour 389 villes en Chine.

[17]:
```python
import requests
from bs4 import BeautifulSoup

header={
    'Referer': "https://www.bing.com/",
    'User-Agent': "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36␣
 ↪(KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0",
}


cn_cities = requests.get("https://www.aqistudy.cn/historydata/", headers=header)
cn_cities = BeautifulSoup(cn_cities.text, 'html.parser')
# cn_cities = cn_cities.findall("div", class_ = "all")
cn_cities = cn_cities.findAll("div", class_= "all")[0].findAll("a")
# cn_cities = cn_cities.findAll("li")
# print(cn_cities)
CNcities = []
for c in cn_cities:
```

```
    CNcities.append(c.contents[0])

print(len(CNcities))
print(CNcities)
```

```
389
['  ', ' ', '   ', '  ', '  ', '   ', ' ', ' ', ' ', '   ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', '  ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
'  ', '  ', ' ', ' ', ' ', ' ', ' ', '  ', ' ', ' ', ' ', ' ', '  ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '  ', '  ', ' ', ' ', ' ',
' ', '    ', ' ', ' ', '  ', ' ', ' ', ' ', '   ', '  ', ' ', '  ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '  ', '  ', ' ',
' ', ' ', '  ', ' ', ' ', ' ', '  ', ' ', ' ', ' ', '   ', ' ',
'  ', ' ', ' ', ' ', ' ', '  ', ' ', '   ', ' ', '   ', '  ', '   ',
' ', '  ', ' ', '  ', ' ', ' ', ' ', '   ', '  ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '  ', ' ', '    ', ' ',
' ', ' ', ' ', ' ', '  ', ' ', ' ', ' ', ' ', ' ', '  ', ' ', '  ',
' ', ' ', ' ', ' ', '  ', ' ', ' ', ' ', ' ', ' ', '  ', ' ',
'   ', '  ', ' ', '   ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', '  ', ' ', ' ', ' ', '  ', ' ',
' ', ' ', '  ', ' ', ' ', ' ', ' ', ' ', ' ', '  ', ' ', ' ', ' ',
' ', ' ', '  ', '  ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', '  ', ' ', ' ', '  ', ' ', ' ', ' ', '  ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', '  ', ' ', ' ', '  ', ' ', ' ', '   ', '  ',
'   ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '  ', ' ', ' ', ' ', '  ',
' ', ' ', '  ', '  ', ' ', ' ', '  ', ' ', ' ', '    ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', '  ', ' ', '   ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', '    ', ' ', '   ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
'  ', ' ', ' ', ' ', ' ', ' ', '   ', '   ', ' ', ' ', '  ', ' ',
' ', ' ', ' ', ' ', '  ', ' ', '  ', ' ', ' ', ' ', ' ', '  ', ' ',
'   ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', '  ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', '    ', ' ', '  ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', '  ', '  ', '  ', ' ', ' ',
'  ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ']
```

[72]:
```
with open('./data/airquality/aqistudy/cities.txt', 'w', encoding='utf-8') as␣
 ↪file:
    for city in CNcities:
        file.write(city + '\n')
```

[19]:
```
import os
import csv
import json
```

```python
aqistudy_path = "../data/airquality/aqistudy/"

def toCsv_aqistudy_by_city(city):
    path = aqistudy_path+city
    files = os.listdir(path)
    csv_filename = aqistudy_path+city+".csv"
    file_exists = os.path.isfile(csv_filename)
    if file_exists and os.stat(csv_filename).st_size != 0:
        return
    print(files)
    for json_filename in files:
        if not json_filename.lower().endswith('.json'):
            continue
        with open(csv_filename, 'a', newline='', encoding='utf-8') as csvfile:
            fieldnames = ['Year', 'Month', 'Day', 'AQI', 'Quality Grade', 'PM2.
 ↪5', 'PM10', 'CO', 'SO2', 'NO2', 'O3_8h']
            writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

            if not file_exists or os.stat(csv_filename).st_size == 0:
                writer.writeheader()

            with open(path+"/"+json_filename, 'r', encoding='utf-8') as␣
 ↪json_file:
                json_data = json.load(json_file)
                for record in json_data:
                    year, month, day = map(int, record['date'].split('-'))

                    writer.writerow({
                        'Year': year,
                        'Month': month,
                        'Day': day,
                        'AQI': int(record['AQI']),
                        'Quality Grade': record['Quality grade'],
                        'PM2.5': int(record['PM2.5']),
                        'PM10': int(record['PM10']),
                        'CO': float(record['CO']),
                        'SO2': int(record['SO2']),
                        'NO2': int(record['NO2']),
                        'O3_8h': int(record['O3_8h'])
                    })


# json_to_csv(json_data, 'output.csv')
#
```

Les données pour Shanghai:

[20]:
```
toCsv_aqistudy_by_city(" ")
```

[58]:
```
df_aqistudy_sh = pd.read_csv("../data/airquality/aqistudy/ .csv")
df_aqistudy_sh.columns = ['Year', 'Month', 'Day', 'AQI', 'Quality Grade', 'PM2.
 ↪5', 'PM10', 'CO', 'SO2', 'NO2', 'O3']
df_aqistudy_sh['date'] = pd.to_datetime(df_aqistudy_sh[['Year', 'Month',␣
 ↪'Day']])
df_aqistudy_sh = df_aqistudy_sh.drop(columns=['Year', 'Month', 'Day', 'Quality␣
 ↪Grade'])
for idx in ['AQI', 'PM2.5', 'PM10', 'CO', 'SO2', 'NO2', 'O3']:
    df_aqistudy_sh[idx] = pd.to_numeric(df_aqistudy_sh[idx],errors='coerce')
# df_aqistudy_sh.dropna(how='any')
df_aqistudy_sh
```

[58]:
```
       AQI  PM2.5  PM10   CO  SO2  NO2  O3        date
0      195    147   181  1.7   63   99  61  2014-01-01
1      147    113   131  1.6   37   95  60  2014-01-02
2      189    142   163  1.4   56   96  45  2014-01-03
3      151    115   125  1.2   36   64  38  2014-01-04
4       65     47    60  1.0   25   63  31  2014-01-05
…     …     …    …   …   …   …   ..         …
3647   83     51    68  0.7    8   66   0  2023-12-27
3648   99     64    85  0.9    9   79  43  2023-12-28
3649   67     48    61  0.6    6   47  60  2023-12-29
3650  115     87   102  0.9    7   62  43  2023-12-30
3651  202    152   180  1.3    8   42  70  2023-12-31

[3652 rows x 8 columns]
```

[27]:
```
df_aqistudy_sh.isna().mean()
```

[27]:
```
AQI      0.0
PM2.5    0.0
PM10     0.0
CO       0.0
SO2      0.0
NO2      0.0
O3       0.0
date     0.0
dtype: float64
```

2.6 1.2 Comapraison entre les données de la pollution de l'air

[99]:
```
import matplotlib.pyplot as plt
min_date = min(df_aqistudy_sh['date'].min(), df_berkeley_sh['date'].min(),␣
 ↪df_aqicn_sh['date'].min())
```

```
max_date = max(df_aqistudy_sh['date'].max(), df_berkeley_sh['date'].max(),␣
↪df_aqicn_sh['date'].max())

df_aqistudy_sh = df_aqistudy_sh.sort_values(by='date')
df_berkeley_sh = df_berkeley_sh.sort_values(by='date')
df_aqicn_sh = df_aqicn_sh.sort_values(by='date')

df_aqistudy_sh_m = df_aqistudy_sh.copy()
df_berkeley_sh_m = df_berkeley_sh.copy()
df_aqicn_sh_m = df_aqicn_sh.copy()

df_aqistudy_sh_m['year_month'] = pd.to_datetime(df_aqistudy_sh_m['date'].dt.
↪to_period('M').dt.strftime('%Y-%m'))
df_berkeley_sh_m['year_month'] = pd.to_datetime(df_berkeley_sh_m['date'].dt.
↪to_period('M').dt.strftime('%Y-%m'))
df_aqicn_sh_m['year_month'] = pd.to_datetime(df_aqicn_sh_m['date'].dt.
↪to_period('M').dt.strftime('%Y-%m'))

df_aqistudy_sh_m = df_aqistudy_sh_m.groupby('year_month', as_index=False).
↪mean().drop(columns=['date']).sort_values(by='year_month')
df_berkeley_sh_m = df_berkeley_sh_m.groupby('year_month', as_index=False).
↪mean().drop(columns=['date']).sort_values(by='year_month')
df_aqicn_sh_m = df_aqicn_sh_m.groupby('year_month', as_index=False).mean().
↪drop(columns=['date']).sort_values(by='year_month')

df_aqistudy_sh_y = df_aqistudy_sh.copy()
df_berkeley_sh_y = df_berkeley_sh.copy()
df_aqicn_sh_y = df_aqicn_sh.copy()
df_aqistudy_sh_y['year'] = pd.to_datetime(df_aqistudy_sh_y['date'].dt.
↪to_period('Y').dt.strftime('%Y'))
df_berkeley_sh_y['year'] = pd.to_datetime(df_berkeley_sh_y['date'].dt.
↪to_period('Y').dt.strftime('%Y'))
df_aqicn_sh_y['year'] = pd.to_datetime(df_aqicn_sh_y['date'].dt.to_period('Y').
↪dt.strftime('%Y'))
df_aqistudy_sh_y = df_aqistudy_sh_y.drop(columns=['date']).groupby('year',␣
↪as_index=False).mean(numeric_only=True)
df_berkeley_sh_y = df_berkeley_sh_y.drop(columns=['date']).groupby('year',␣
↪as_index=False).mean(numeric_only=True)
df_aqicn_sh_y = df_aqicn_sh_y.drop(columns=['date']).groupby('year',␣
↪as_index=False).mean(numeric_only=True)
```

Pour la ville de Shanghai:

a) AQI D'après la définition de l'IQA ($IQA = max(IQA_{polluant1}, IQA_{polluant2}, ...)$), nous pouvons calculer la colonne de l'IQA pour les données de WAQI:

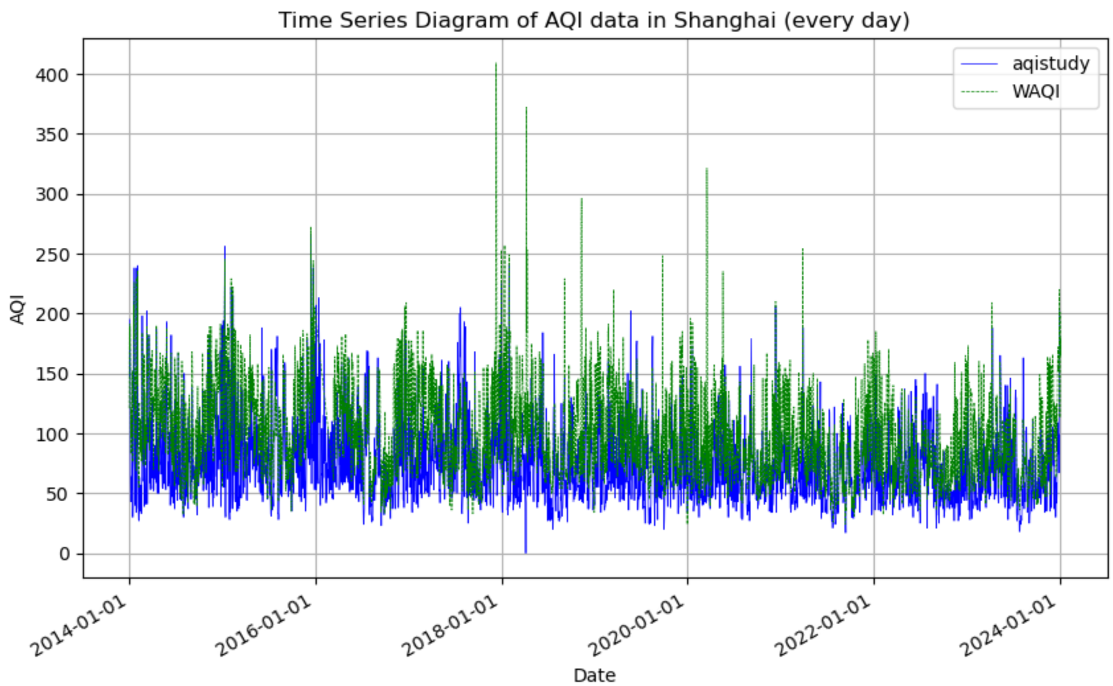
[42]:
```
df_aqicn_sh['AQI'] = df_aqicn_sh[['PM2.5', 'PM10', 'O3', 'NO2', 'SO2', 'CO']].
    ↪max(axis=1)
df_aqicn_sh
```

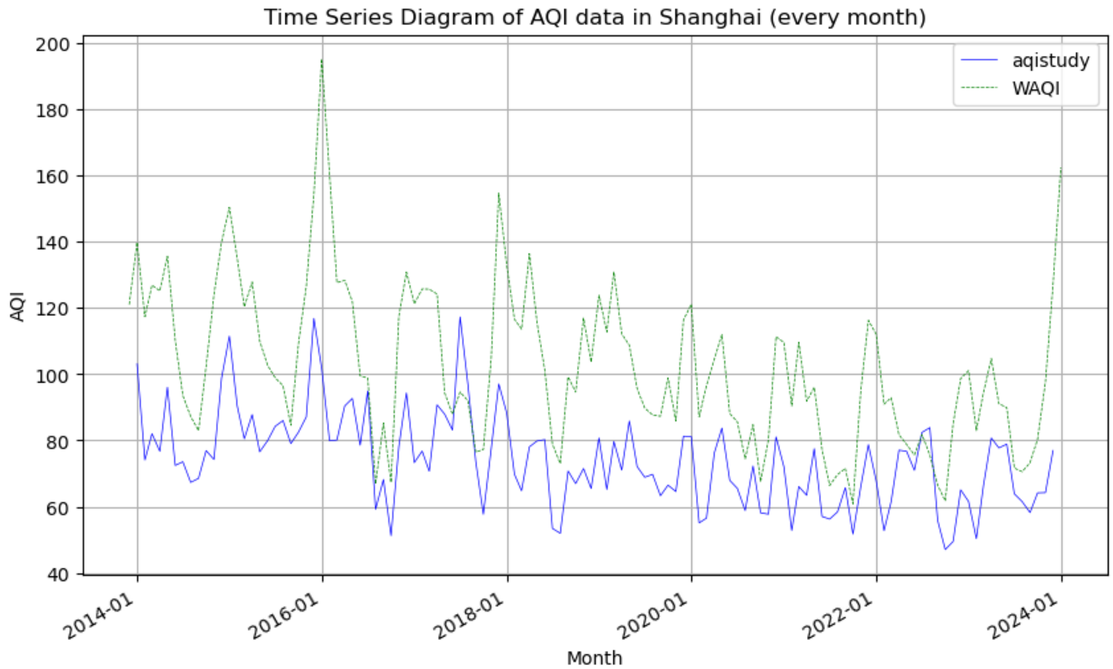
[42]:
```
            date  PM2.5   PM10    O3   NO2   SO2    CO    AQI
0     2024-01-01  151.0   55.0  28.0  23.0   3.0   7.0  151.0
1     2024-01-02  113.0   76.0  28.0  23.0   3.0  10.0  113.0
2     2024-01-03  165.0   92.0  39.0  34.0   4.0  12.0  165.0
3     2024-01-04  179.0   75.0  25.0  38.0   6.0  10.0  179.0
4     2024-01-05  165.0  109.0  37.0  45.0   5.0  14.0  165.0
…            …      …      …     …     …     …     …      …
3576  2018-12-31    NaN   34.0  26.0  13.0   4.0   3.0   34.0
3577  2017-09-10    NaN   26.0  33.0  16.0   3.0   9.0   33.0
3578  2016-03-13    NaN   61.0  51.0  13.0   8.0   7.0   61.0
3579  2014-12-31    NaN   55.0  24.0  19.0  15.0   6.0   55.0
3580  2013-12-31    NaN  121.0  29.0  57.0  30.0  14.0  121.0

[3581 rows x 8 columns]
```

Nous pouvons comparer les IQA de WAQI(aqicn.org) et de aqistudy:

[120]:
```
plt.figure(figsize=(10, 6))
plt.plot(df_aqistudy_sh['date'], df_aqistudy_sh['AQI'], linestyle='-',␣
    ↪linewidth=0.5, color='b', label='aqistudy')
# plt.plot(df_berkeley_sh['date'], df_berkeley_sh['PM2.5'], linestyle='--',␣
    ↪linewidth=0.5, color='r', label='berkeley')
plt.plot(df_aqicn_sh['date'], df_aqicn_sh['AQI'], linestyle='--', linewidth=0.
    ↪5, color='g', label='WAQI')

plt.xlabel('Date')
plt.ylabel('AQI')
# plt.ylim([0, 300])
plt.title('Time Series Diagram of AQI data in Shanghai (every day)')

plt.gca().xaxis.set_major_formatter(plt.matplotlib.dates.
    ↪DateFormatter('%Y-%m-%d'))
plt.gcf().autofmt_xdate()
plt.grid()
plt.legend()
plt.show()

plt.figure(figsize=(10, 6))

plt.plot(df_aqistudy_sh_m['year_month'], df_aqistudy_sh_m['AQI'],␣
    ↪linestyle='-', linewidth=0.5, color='b', label='aqistudy')
# plt.plot(df_berkeley_sh['date'], df_berkeley_sh['PM2.5'], linestyle='--',␣
    ↪linewidth=0.5, color='r', label='berkeley')
```

```python
plt.plot(df_aqicn_sh_m['year_month'], df_aqicn_sh_m['AQI'], linestyle='--',␣
 ↪linewidth=0.5, color='g', label='WAQI')

plt.xlabel('Month')
plt.ylabel('AQI')
# plt.ylim([0, 300])
plt.title('Time Series Diagram of AQI data in Shanghai (every month)')
plt.grid()
plt.gca().xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
plt.gcf().autofmt_xdate()

plt.legend()
plt.show()
```

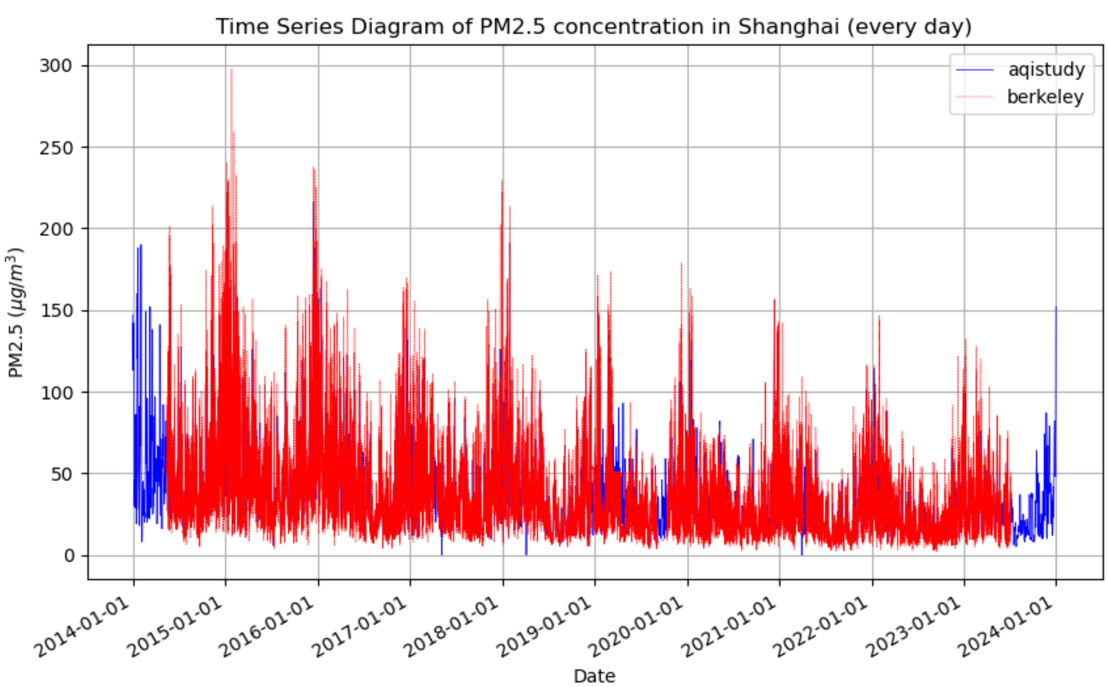


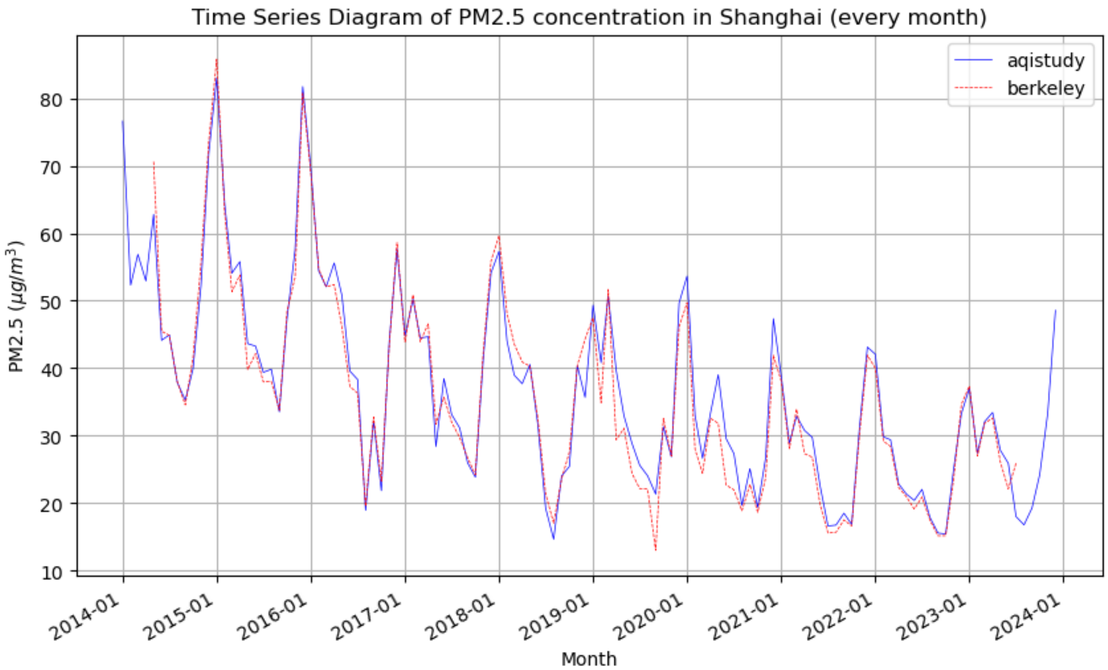


Nous pouvons comparer les concentrations de PM2.5 des données de berkeleyearth et de aqistudy:

[90]:
```
plt.figure(figsize=(10, 6))
plt.plot(df_aqistudy_sh['date'], df_aqistudy_sh['PM2.5'], linestyle='-',␣
↪linewidth=0.5, color='b', label='aqistudy')
plt.plot(df_berkeley_sh['date'], df_berkeley_sh['PM2.5'], linestyle='--',␣
↪linewidth=0.3, color='r', label='berkeley')
# plt.plot(df_aqicn_sh['date'], df_aqicn_sh['PM2.5'], linestyle='--',␣
↪linewidth=0.5, color='g', label='WAQI')

plt.xlabel('Date')
plt.ylabel('PM2.5 ($\mu g/m^3$)')
# plt.ylim([0, 300])
plt.title('Time Series Diagram of PM2.5 concentration in Shanghai (every day)')

plt.gca().xaxis.set_major_formatter(plt.matplotlib.dates.
↪DateFormatter('%Y-%m-%d'))
plt.gcf().autofmt_xdate()
plt.grid()
plt.legend()
plt.show()

plt.figure(figsize=(10, 6))
```

```
plt.plot(df_aqistudy_sh_m['year_month'], df_aqistudy_sh_m['PM2.5'],␣
 ↪linestyle='-', linewidth=0.5, color='b', label='aqistudy')
plt.plot(df_berkeley_sh_m['year_month'], df_berkeley_sh_m['PM2.5'],␣
 ↪linestyle='--', linewidth=0.5, color='r', label='berkeley')
# plt.plot(df_aqicn_sh_m['year_month'], df_aqicn_sh_m['PM2.5'], linestyle='--',␣
 ↪linewidth=0.5, color='g', label='WAQI')

plt.xlabel('Month')
plt.ylabel('PM2.5 ($\mu g/m^3$)')
# plt.ylim([0, 300])
plt.title('Time Series Diagram of PM2.5 concentration in Shanghai (every␣
 ↪month)')
plt.grid()
plt.gca().xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y-%m'))
plt.gcf().autofmt_xdate()

plt.legend()
plt.show()
```

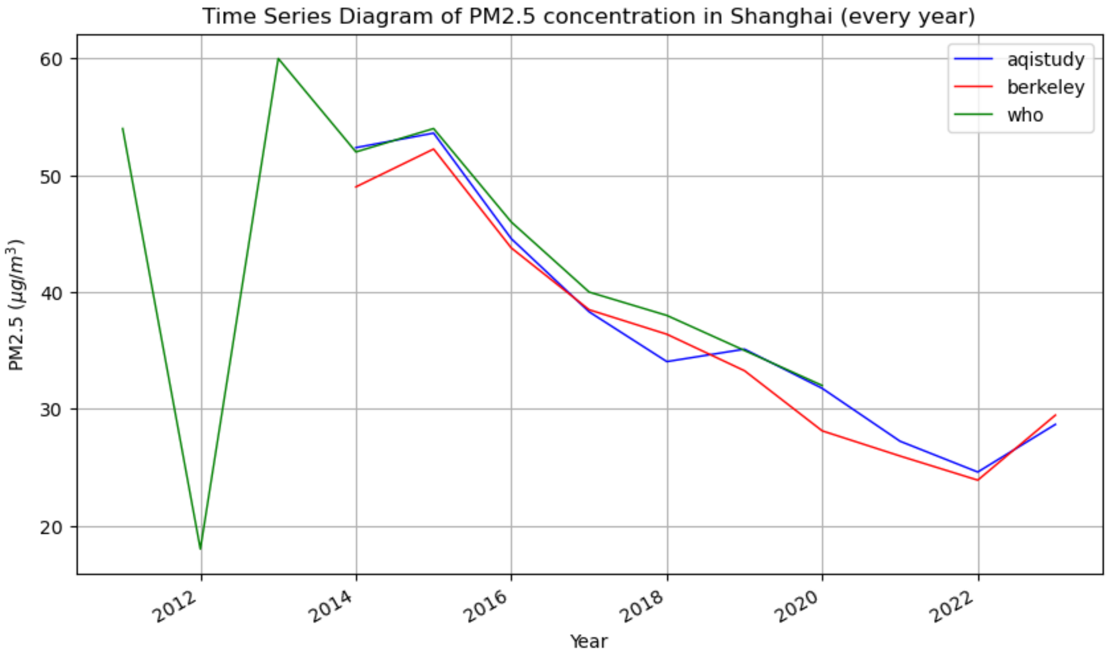




Nous pouvons aussi comparer les concentrations de PM2.5 annuelle de WHO, berkeleyearth et aqistudy:

[118]:
```
df_who_sh.loc[:,'year'] = pd.to_datetime(df_who_sh['year'], format='%Y')
df_who_sh = df_who_sh.dropna(subset=['pm25_concentration']).sort_values('year')

plt.figure(figsize=(10, 6))

plt.plot(df_aqistudy_sh_y['year'], df_aqistudy_sh_y['PM2.5'], linestyle='-',␣
 ↪linewidth=1, color='b', label='aqistudy')
plt.plot(df_berkeley_sh_y['year'], df_berkeley_sh_y['PM2.5'], linestyle='-',␣
 ↪linewidth=1, color='r', label='berkeley')
# plt.plot(df_aqicn_sh_m['year_month'], df_aqicn_sh_m['PM2.5'], linestyle='--',␣
 ↪linewidth=0.5, color='g', label='WAQI')
plt.plot(df_who_sh['year'], df_who_sh['pm25_concentration'], linestyle='-',␣
 ↪linewidth=1, color='g', label='who')

plt.xlabel('Year')
plt.ylabel('PM2.5 ($\mu g/m^3$)')
# plt.ylim([0, 300])
plt.title('Time Series Diagram of PM2.5 concentration in Shanghai (every year)')
plt.grid()
plt.gca().xaxis.set_major_formatter(plt.matplotlib.dates.DateFormatter('%Y'))
plt.gcf().autofmt_xdate()
```

```
plt.legend()
plt.show()
```



2.7 ## 1.3 Données du taux de mortalité maladies respiratoires et problème rencontré

- La plupart des pays ne donne que les données anuelles de décès.
- Les données de décès par ville sont souvent inaccessible.
- et ne précise pas la cause de décès.

[1]:
```
from pyspark import SparkConf, SparkContext
from pyspark.sql.types import *
from pyspark.sql import SparkSession
from pyspark.sql.functions import  expr, split, min, max, stddev, mean,␣
 ↪avg,median, col, broadcast, when
from pyspark.sql.types import IntegerType, StringType
spark = SparkSession.builder \
 .master("local[10]") \
 .config("spark.driver.memory", "20g") \
 .appName("air-pollution") \
 .getOrCreate()
spark.conf.set("spark.sql.caseSensitive", "false")
```

[197]:
```
sc = spark.sparkContext
```

2.7.1 a) Données de taux de décès maladies respiratoires (src: IHME)

Téléchargement: https://vizhub.healthdata.org/gbd-results/

[150]:
```
filepath_IHME = "../data/IHME/IHME-GBD_2019_DATA-d343b043-1.csv"
# df_IHME = pd.read_csv(filepath_IHME)
# df_cols = df_IHME.columns.values
# print(df_cols)
IHMEparsed = spark.read.option("header", "true").csv(filepath_IHME)
# ps_cols = IHMEparsed.columns
# print(ps_cols)
IHMEparsed.show(5)
```

```
+--------+-----------------+----+--------+--------------------+-------+----+-----
-------------+--------------------+--------------------+
|measure|         location| sex|     age|               cause| metric|year|
val|                upper|               lower|
+--------+-----------------+----+--------+--------------------+-------+----+-----
-------------+--------------------+--------------------+
| Deaths|           Guyana|Both|All ages|Respiratory infec…|Percent|1990|
0.05751660152114|0.060875075732561616|0.053794338095241014|
| Deaths|           Guyana|Both|All ages|Respiratory infec…|
Rate|1990|48.524773827237134|   54.12308391476895|  42.849050964854754|
| Deaths|    Guinea-Bissau|Both|All ages|Respiratory
infec…|Percent|1990|0.1532392311976194|  0.1857518042301856|
0.1288624737097962|
| Deaths|    Guinea-Bissau|Both|All ages|Respiratory infec…|
Rate|1990|260.93392193693967|  326.36381444880226|  209.46886866481836|
| Deaths|Brunei Darussalam|Both|All ages|Chronic
respirato…|Percent|1990|0.0661729164170575| 0.07354479349561156|
0.04879569163886584|
+--------+-----------------+----+--------+--------------------+-------+----+-----
-------------+--------------------+--------------------+
only showing top 5 rows
```

[151]:
```
IHMEparsed = IHMEparsed.filter(col("metric") == "Rate").select('location',␣
↪'cause', 'year', 'val')
IHMEparsed = IHMEparsed.withColumnRenamed("val", "Death Rate")
IHMEparsed.show(5)
```

```
+-----------------+--------------------+----+------------------+
|         location|               cause|year|        Death Rate|
+-----------------+--------------------+----+------------------+
|           Guyana|Respiratory infec…|1990|48.524773827237134|
|    Guinea-Bissau|Respiratory infec…|1990|260.93392193693967|
|Brunei Darussalam|Chronic respirato…|1990|28.047355314816443|
|         Honduras|Respiratory infec…|1990| 44.64436203593287|
```

```
|           Kuwait|Chronic respirato…|1990|4.6259092778190425|
+-----------------+------------------+----+------------------+
only showing top 5 rows
```

[260]:
```python
IHMEparsed_ri = IHMEparsed.filter(col("cause") == "Respiratory infections and␣
↪tuberculosis")
IHMEparsed_cr = IHMEparsed.filter(col("cause") == "Chronic respiratory␣
↪diseases")
```

[261]:
```python
IHMEparsed_cr.count()
```

[261]: 6120

2.7.2 b) Données de émise de polluant par pay

Téléchargement: https://ourworldindata.org/explorers/air-pollution

[34]:
```python
import csv
year_from = 1990
country_alias = {
    "egypt":"egypt, arab rep.",
    "europe":"european union",
    "faeroe islands":"faroe islands",
    "gambia":"gambia, the",
    "french guiana":"guinea",
    "high-income countries":"high income",
    "hong kong":"hong kong sar, china",
    "iran":"iran, islamic rep.",
    "north korea":"korea, dem. people's rep.",
    "south korea":"korea, rep.",
    "laos":"lao pdr",
    "low-income countries":"low income",
    "micronesia (country)":"micronesia, fed. sts.",
    "russia":"russian federation",
    "slovakia":"slovak republic",
    "saint kitts and nevis":"st. kitts and nevis",
    "saint lucia":"st. lucia",
    "syria":"syrian arab republic",
    "timor":"timor-leste",
    "turkey":"turkiye",
    "upper-middle-income countries":"upper middle income",
    "venezuela":"venezuela, rb",
    "vietnam":"viet nam",
    "united states virgin islands":"virgin islands (u.s.)",
    "yemen":"yemen, rep.",
}
```

[50]:

```python
df_air_pollution = pd.read_csv("../data/airquality/air-pollution.csv")
df_air_pollution = df_air_pollution[df_air_pollution['Year'].astype(int) >=␣
 ↪1990]
fields_ap = df_air_pollution.columns.values
print(fields_ap)
df_air_pollution
```

```
['Nitrogen oxide (NOx)' 'Sulphur dioxide (SO )' 'Carbon monoxide (CO)'
 'Organic carbon (OC)' 'NMVOCs' 'Black carbon (BC)' 'Ammonia (NH )'
 'Nitrogen oxide (NOx).1' 'Sulphur dioxide (SO ).1'
 'Carbon monoxide (CO).1' 'Organic carbon (OC).1' 'NMVOCs.1'
 'Black carbon (BC).1' 'Ammonia (NH ).1' 'Entity' 'Year']
```

[50]:

```
       Nitrogen oxide (NOx)  Sulphur dioxide (SO )  Carbon monoxide (CO)  \
195           369593.165109           10560.149196          7.660519e+05
196           350497.507709            9881.083158          7.245905e+05
197           224889.350417            5981.234759          4.662375e+05
198           222415.282316            5894.177473          4.633083e+05
199           222376.597095            6251.537337          4.836455e+05
…                       …                    …                   …
47530          83842.096401           67231.291799          1.610636e+06
47531          76234.430113           59452.695878          1.632515e+06
47532          74381.797266           53891.385836          1.657689e+06
47533          73062.525071           51072.778332          1.653665e+06
47534          70779.920495           45896.979630          1.647792e+06

       Organic carbon (OC)         NMVOCs  Black carbon (BC)  Ammonia (NH )  \
195           21148.920979  324790.071352        6524.425310   75722.096012
196           21775.994114  297031.068717        6648.054079   80299.290882
197           22343.021367  183132.832872        6514.724062   86201.798012
198           23349.443296  177369.007446        6739.386545   92924.370095
199           24295.073299  181305.074648        7037.784596   99621.739384
…                      …              …                 …              …
47530        108275.483442  299713.466501       30912.239774  112425.840095
47531        111975.723799  302718.315314       31570.526454  115539.979134
47532        114613.199492  306905.624759       32344.405320  118254.660089
47533        114583.507408  306860.211088       32365.562573  119965.763390
47534        114543.283470  306574.849324       32364.526930  121689.671422

       Nitrogen oxide (NOx).1  Sulphur dioxide (SO ).1  \
195                 29.776338                 0.850780
196                 26.355146                 0.742994
197                 15.525089                 0.412911
198                 14.062142                 0.372658
199                 13.022964                 0.366107
…                        …                 …
47530                6.069075                 4.866669
```

```
47531              5.433542             4.237439
47532              5.224689             3.785412
47533              5.060148             3.537187
47534              4.832887             3.133868

       Carbon monoxide (CO).1  Organic carbon (OC).1   NMVOCs.1  \
195                 61.717105               1.703867  26.166769
196                 54.484521               1.637414  22.334816
197                 32.186403               1.542436  12.642456
198                 29.292532               1.476262  11.214104
199                 28.323565               1.422784  10.617707
…                         …                    …          …
47530              116.589083               7.837734  21.695348
47531              116.356079               7.980971  21.575982
47532              116.438520               8.050602  21.557510
47533              114.529137               7.935799  21.252456
47534              112.512068               7.821071  20.933079

       Black carbon (BC).1  Ammonia (NH ).1       Entity  Year
195               0.525641         6.100564  Afghanistan  1990
196               0.499891         6.037987  Afghanistan  1991
197               0.449740         5.950885  Afghanistan  1992
198               0.426096         5.875116  Afghanistan  1993
199               0.412151         5.834114  Afghanistan  1994
…                      …               …           …    …
47530             2.237643         8.138165     Zimbabwe  2015
47531             2.250162         8.235010     Zimbabwe  2016
47532             2.271919         8.306384     Zimbabwe  2017
47533             2.241567         8.308562     Zimbabwe  2018
47534             2.209866         8.309030     Zimbabwe  2019

[6900 rows x 16 columns]
```

[148]:
```python
lst_structField_ap = []
for field in fields_ap:
    if field == "Entity" or field == "Year":
        lst_structField_ap.append(StructField(field, StringType(), False))
    else:
        lst_structField_ap.append(StructField(field, DoubleType(), True))
schema_ap = StructType(lst_structField_ap)
parsed_ap = spark.read.option("header", "true").schema(schema_ap).csv("../data/
 ↪airquality/air-pollution.csv")
# parsed_ap.count()
parsed_ap = parsed_ap.filter(col("Year") >=1990)
parsed_ap.count()
parsed_ap.show(5)
```

```
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+-----------------+-------------------
---+-----------------------+----------------------+---------------------+-------
-----------+-------------------+-----------------+-----------+----+
|Nitrogen oxide (NOx)|Sulphur dioxide (SO )|Carbon monoxide (CO)|Organic carbon
(OC)|            NMVOCs| Black carbon (BC)|    Ammonia (NH )|Nitrogen oxide
(NOx).1|Sulphur dioxide (SO ).1|Carbon monoxide (CO).1|Organic carbon (OC).1|
NMVOCs.1|Black carbon (BC).1|  Ammonia (NH ).1|     Entity|Year|
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+-----------------+-------------------
---+-----------------------+----------------------+---------------------+-------
-----------+-------------------+-----------------+-----------+----+
|    369593.165109308|     10560.1491957337|   766051.9024026981|
21148.9209786874|324790.07135184004|  6524.42531036713| 75722.0960124105|
29.77633779151264|     0.8507802612852433|    61.717105090478164|
1.703866506300672|26.166768730806055|
0.525641462767661|6.100563868598725|Afghanistan|1990|
|  350497.50770859997|     9881.08315843062|    724590.512212164|
21775.9941142059|  297031.068717025|6648.0540791168505| 80299.2908819285|
26.35514595279831|     0.7429935536907858|     54.48452067522619|
1.6374139345501877|22.334815501915703|
0.4998906745519255|6.037987388084088|Afghanistan|1991|
|    224889.350416567|     5981.23475867664|    466237.519516482|
22343.0213674859|183132.83287231598|6514.7240623324205| 86201.7980122156|
15.525089423059047|     0.4129106350156594|    32.186402644103985|
1.542435887110749|
12.64245550700557|0.44973972065337287|5.950884824422226|Afghanistan|1992|
|    222415.282315777|      5894.1774732922|    463308.283652427|
23349.4432961168|177369.00744575102|  6739.38654489345|92924.37009491949|
14.06214156352411|    0.37265765718514365|    29.292531540273856|   1.4762617642
132339|11.214103930784562|0.42609575501673524|5.875116284144709|Afghanistan|1993
|
|    222376.597095223|    6251.537336636859|  483645.49216481904|
24295.0732989028|  181305.074647839|  7037.78459638022| 99621.7393839683|
13.022964355910506|    0.36610663607647415|     28.32356501373289|
1.422784041705443|10.617706878900801|0.41215136457902235|
5.83411374226436|Afghanistan|1994|
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+-----------------+-------------------
---+-----------------------+----------------------+---------------------+-------
-----------+-------------------+-----------------+-----------+----+
only showing top 5 rows
```

2.7.3 c) Données de population, surface, GDP par pays (src: The World Bank)

- World Development Indicators (WDI) téléchargement: https://datacatalog.worldbank.org/search/dataset/00 Development-Indicators

[85]:
```
df_wdi = pd.read_csv("../data/WDI_CSV/WDIData.csv")
selected_indicators = {
    "NY.GDP.MKTP.CD": "GDP",                # GDP
    "EN.ATM.PM25.MC.M3": "PM2.5",           # PM2.5
    "SP.POP.TOTL": "Population",            # Population, total
    "AG.SRF.TOTL.K2": "Surface",            # Surface area (sq. km)
}
df_wdi.columns.values
```

[85]:
```
array(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
       '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967',
       '1968', '1969', '1970', '1971', '1972', '1973', '1974', '1975',
       '1976', '1977', '1978', '1979', '1980', '1981', '1982', '1983',
       '1984', '1985', '1986', '1987', '1988', '1989', '1990', '1991',
       '1992', '1993', '1994', '1995', '1996', '1997', '1998', '1999',
       '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007',
       '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015',
       '2016', '2017', '2018', '2019', '2020', '2021', '2022',
       'Unnamed: 67'], dtype=object)
```

[336]:
```
df_wdi
```

[336]:
```
                       Country Name Country Code  \
0       Africa Eastern and Southern          AFE
1       Africa Eastern and Southern          AFE
2       Africa Eastern and Southern          AFE
3       Africa Eastern and Southern          AFE
4       Africa Eastern and Southern          AFE
…                               …          …
395271                     Zimbabwe          ZWE
395272                     Zimbabwe          ZWE
395273                     Zimbabwe          ZWE
395274                     Zimbabwe          ZWE
395275                     Zimbabwe          ZWE

                                           Indicator Name     Indicator Code  \
0       Access to clean fuels and technologies for coo…     EG.CFT.ACCS.ZS
1       Access to clean fuels and technologies for coo…  EG.CFT.ACCS.RU.ZS
2       Access to clean fuels and technologies for coo…  EG.CFT.ACCS.UR.ZS
3                 Access to electricity (% of population)     EG.ELC.ACCS.ZS
4       Access to electricity, rural (% of rural popul…  EG.ELC.ACCS.RU.ZS
…                                                     …                 …
395271  Women who believe a husband is justified in be…     SG.VAW.REFU.ZS
395272  Women who were first married by age 15 (% of w…  SP.M15.2024.FE.ZS
395273  Women who were first married by age 18 (% of w…  SP.M18.2024.FE.ZS
395274  Women's share of population ages 15+ living wi…  SH.DYN.AIDS.FE.ZS
395275  Young people (ages 15-24) newly infected with HIV     SH.HIV.INCD.YG
```

```
        1960  1961  1962  1963  1964  1965  …          2014          2015  \
0        NaN   NaN   NaN   NaN   NaN   NaN  …     17.392349     17.892005
1        NaN   NaN   NaN   NaN   NaN   NaN  …      6.720331      7.015917
2        NaN   NaN   NaN   NaN   NaN   NaN  …     38.184152     38.543180
3        NaN   NaN   NaN   NaN   NaN   NaN  …     31.859257     33.903515
4        NaN   NaN   NaN   NaN   NaN   NaN  …     17.623956     16.516633
…        …     …     …     …     …     …    …            …            …
395271   NaN   NaN   NaN   NaN   NaN   NaN  …           NaN     14.500000
395272   NaN   NaN   NaN   NaN   NaN   NaN  …           NaN      3.700000
395273   NaN   NaN   NaN   NaN   NaN   NaN  …           NaN     32.400000
395274   NaN   NaN   NaN   NaN   NaN   NaN  …     59.400000     59.500000
395275   NaN   NaN   NaN   NaN   NaN   NaN  …  19000.000000  17000.000000

                2016          2017          2018         2019         2020  \
0          18.359993     18.795151     19.295176    19.788156    20.279599
1           7.281390      7.513673      7.809566     8.075889     8.366010
2          38.801719     39.039014     39.323186    39.643848    39.894830
3          38.851444     40.197332     43.028332    44.389773    46.268621
4          24.594474     25.389297     27.041743    29.138285    30.998687
…                …             …             …            …            …
395271           NaN           NaN           NaN          NaN          NaN
395272           NaN           NaN           NaN     5.400000          NaN
395273           NaN           NaN           NaN    33.700000          NaN
395274     59.700000     59.900000     60.100000    60.300000    60.500000
395275  15000.000000  13000.000000  10000.000000  8600.000000  7700.000000

               2021  2022  Unnamed: 67
0         20.773627   NaN          NaN
1          8.684137   NaN          NaN
2         40.213891   NaN          NaN
3         48.103609   NaN          NaN
4         32.772690   NaN          NaN
…               …     …            …
395271          NaN   NaN          NaN
395272          NaN   NaN          NaN
395273          NaN   NaN          NaN
395274    60.700000   NaN          NaN
395275  6800.000000   NaN          NaN

[395276 rows x 68 columns]
```

[118]:

```
selected_columns = ['Country Name', 'Country Code', 'Indicator Code'] +
 ↪[str(year) for year in range(1990, 2023)]
df_wdi_selected = df_wdi[df_wdi['Indicator Code'].isin(list(selected_indicators.
 ↪keys()))]
df_wdi_selected = df_wdi_selected[selected_columns]
```

```
# [df_wdi['Indicator Name'].isin(list(selected_indicators.keys()))]
df_pivoted = df_wdi_selected.pivot(index=['Country Name', 'Country Code'],␣
 ↪columns=['Indicator Code'] )
df_pivoted.reset_index(inplace=True)
df_pivoted
```

[118]:

```
                           Country Name Country Code           1990  \
Indicator Code                                          AG.SRF.TOTL.K2
0                           Afghanistan          AFG   6.528600e+05
1                Africa Eastern and Southern          AFE   1.510674e+07
2                 Africa Western and Central          AFW   9.166270e+06
3                               Albania          ALB   2.875000e+04
4                               Algeria          DZA   2.381740e+06
..                                  …            …              …
261                  West Bank and Gaza          PSE   6.020000e+03
262                               World          WLD   1.339735e+08
263                         Yemen, Rep.          YEM   5.279700e+05
264                              Zambia          ZMB   7.526100e+05
265                            Zimbabwe          ZWE   3.907600e+05

                                                                   1991  \
Indicator Code EN.ATM.PM25.MC.M3 NY.GDP.MKTP.CD   SP.POP.TOTL AG.SRF.TOTL.K2
0                      49.282398            NaN  1.069480e+07   6.528600e+05
1                      30.132449   2.546735e+11  3.098907e+08   1.510674e+07
2                      64.258847   1.218036e+11  2.067390e+08   9.166270e+06
3                      24.947482   2.028554e+09  3.286542e+06   2.875000e+04
4                      30.068900   6.204851e+10  2.551807e+07   2.381740e+06
..                           …              …            …              …
261                    30.043128            NaN  1.978248e+06   6.020000e+03
262                    40.860853   2.293504e+13  5.293395e+09   1.339736e+08
263                    47.262359   1.264382e+10  1.337512e+07   5.279700e+05
264                    26.123078   3.285217e+09  7.686401e+06   7.526100e+05
265                    24.227920   8.783817e+09  1.011389e+07   3.907600e+05

                                                             …  \
Indicator Code EN.ATM.PM25.MC.M3 NY.GDP.MKTP.CD   SP.POP.TOTL  …
0                            NaN            NaN  1.074517e+07  …
1                            NaN   2.756220e+11  3.185441e+08  …
2                            NaN   1.279390e+11  2.121729e+08  …
3                            NaN   1.099559e+09  3.266790e+06  …
4                            NaN   4.571568e+10  2.613390e+07  …
..                           …              …            …  …
261                          NaN            NaN  2.068845e+06  …
262                          NaN   2.393251e+13  5.382537e+09  …
263                          NaN   1.466545e+10  1.389585e+07  …
264                          NaN   3.376791e+09  7.880466e+06  …
265                          NaN   8.641482e+09  1.037782e+07  …
```

```
                            2020                                    2021  \
Indicator Code NY.GDP.MKTP.CD   SP.POP.TOTL AG.SRF.TOTL.K2 EN.ATM.PM25.MC.M3
0                 1.995593e+10  3.897223e+07   6.528600e+05              NaN
1                 9.288802e+11  6.851130e+08   1.516201e+07              NaN
2                 7.869624e+11  4.661891e+08   9.166260e+06              NaN
3                 1.516273e+10  2.837849e+06   2.875000e+04              NaN
4                 1.457435e+11  4.345167e+07   2.381741e+06              NaN
..                           …             …              …                …
261               1.553170e+10  4.803269e+06   6.025000e+03              NaN
262               8.525774e+13  7.820206e+09   1.404869e+08              NaN
263                        NaN  3.228405e+07   5.279700e+05              NaN
264               1.811064e+10  1.892772e+07   7.526100e+05              NaN
265               2.150970e+10  1.566967e+07   3.907600e+05              NaN

                                                                    2022  \
Indicator Code NY.GDP.MKTP.CD   SP.POP.TOTL AG.SRF.TOTL.K2 EN.ATM.PM25.MC.M3
0                 1.426650e+10  4.009946e+07            NaN              NaN
1                 1.086531e+12  7.029771e+08            NaN              NaN
2                 8.449275e+11  4.781859e+08            NaN              NaN
3                 1.793057e+10  2.811666e+06            NaN              NaN
4                 1.634724e+11  4.417797e+07            NaN              NaN
..                           …             …              …                …
261               1.810900e+10  4.922749e+06            NaN              NaN
262               9.752968e+13  7.888306e+09            NaN              NaN
263                        NaN  3.298164e+07            NaN              NaN
264               2.209642e+10  1.947312e+07            NaN              NaN
265               2.837124e+10  1.599352e+07            NaN              NaN


Indicator Code NY.GDP.MKTP.CD   SP.POP.TOTL
0                          NaN  4.112877e+07
1                 1.185138e+12  7.208591e+08
2                 8.753937e+11  4.903309e+08
3                 1.891638e+10  2.777689e+06
4                 1.949984e+11  4.490322e+07
..                           …             …
261               1.911190e+10  5.043612e+06
262               1.013257e+14  7.950947e+09
263                        NaN  3.369661e+07
264               2.916378e+10  2.001768e+07
265               2.736663e+10  1.632054e+07

[266 rows x 134 columns]
```

[131]:
```
df_stacked = df_pivoted.set_index(['Country Name', 'Country Code']).
↪stack(level=0).reset_index()
```

```
df_stacked
```

[131]:
```
Indicator Code Country Name Country Code level_2  AG.SRF.TOTL.K2  \
0               Afghanistan          AFG    1990        652860.0
1               Afghanistan          AFG    1991        652860.0
2               Afghanistan          AFG    1992        652860.0
3               Afghanistan          AFG    1993        652860.0
4               Afghanistan          AFG    1994        652860.0
…                         …            …       …               …
8740               Zimbabwe          ZWE    2018        390760.0
8741               Zimbabwe          ZWE    2019        390760.0
8742               Zimbabwe          ZWE    2020        390760.0
8743               Zimbabwe          ZWE    2021        390760.0
8744               Zimbabwe          ZWE    2022             NaN

Indicator Code  EN.ATM.PM25.MC.M3  NY.GDP.MKTP.CD  SP.POP.TOTL
0                       49.282398             NaN   10694796.0
1                             NaN             NaN   10745167.0
2                             NaN             NaN   12057433.0
3                             NaN             NaN   14003760.0
4                             NaN             NaN   15455555.0
…                               …               …            …
8740                    22.085555    3.415607e+10   15052184.0
8741                    20.834700    2.183223e+10   15354608.0
8742                          NaN    2.150970e+10   15669666.0
8743                          NaN    2.837124e+10   15993524.0
8744                          NaN    2.736663e+10   16320537.0

[8745 rows x 7 columns]
```

[144]:
```
df_wdi_filtered = df_stacked.rename(columns=selected_indicators)\
.rename(columns={"level_2": "Year", "Country Name":"Entity", "Country Code":
 ↪"Code"})
df_wdi_filtered.to_csv("../data/WDI_CSV/WDIDataFiltered.csv")
fields_wdi = df_wdi_filtered.columns.values
print(fields_wdi)
df_wdi_filtered
```

```
['Entity' 'Code' 'Year' 'Surface' 'PM2.5' 'GDP' 'Population']
```

[144]:
```
Indicator Code       Entity Code  Year   Surface      PM2.5           GDP  \
0               Afghanistan  AFG  1990  652860.0  49.282398           NaN
1               Afghanistan  AFG  1991  652860.0        NaN           NaN
2               Afghanistan  AFG  1992  652860.0        NaN           NaN
3               Afghanistan  AFG  1993  652860.0        NaN           NaN
4               Afghanistan  AFG  1994  652860.0        NaN           NaN
…                         …    …     …         …          …
```

```
8740              Zimbabwe  ZWE  2018  390760.0  22.085555  3.415607e+10
8741              Zimbabwe  ZWE  2019  390760.0  20.834700  2.183223e+10
8742              Zimbabwe  ZWE  2020  390760.0        NaN  2.150970e+10
8743              Zimbabwe  ZWE  2021  390760.0        NaN  2.837124e+10
8744              Zimbabwe  ZWE  2022       NaN        NaN  2.736663e+10

Indicator Code  Population
0               10694796.0
1               10745167.0
2               12057433.0
3               14003760.0
4               15455555.0
…                      …
8740            15052184.0
8741            15354608.0
8742            15669666.0
8743            15993524.0
8744            16320537.0

[8745 rows x 7 columns]
```

[146]:

```python
lst_structField_wdi = [StructField("Index", StringType(), False)]
for field in fields_wdi:
    if field == "Entity" or field == "Year" or field == "Code":
        lst_structField_wdi.append(StructField(field, StringType(), False))
    else:
        lst_structField_wdi.append(StructField(field, DoubleType(), True))
schema_wdi = StructType(lst_structField_wdi)
parsed_wdi = spark.read.option("header", "true").schema(schema_wdi).csv("../
 ↪data/WDI_CSV/WDIDataFiltered.csv").drop("Index")

print(parsed_wdi.columns)
parsed_wdi.show(5)
```

```
['Entity', 'Code', 'Year', 'Surface', 'PM2.5', 'GDP', 'Population']
+-----------+----+----+--------+-----------+----+-----------+
|     Entity|Code|Year| Surface|      PM2.5| GDP| Population|
+-----------+----+----+--------+-----------+----+-----------+
|Afghanistan| AFG|1990|652860.0|49.28239771|NULL|1.0694796E7|
|Afghanistan| AFG|1991|652860.0|       NULL|NULL|1.0745167E7|
|Afghanistan| AFG|1992|652860.0|       NULL|NULL|1.2057433E7|
|Afghanistan| AFG|1993|652860.0|       NULL|NULL| 1.400376E7|
|Afghanistan| AFG|1994|652860.0|       NULL|NULL|1.5455555E7|
+-----------+----+----+--------+-----------+----+-----------+
only showing top 5 rows
```

3 Partie 2. Analyse de données

3.1 2.1 join

[262]:
```
fields_air = df_air_pollution.columns.values
fields_air

# datafile ="../data/airquality/air-pollution.csv"
# lst_structField_air = []
# for field in fields_air:
#     if field == "Entity" or field == "Year" or field == "Code":
#         lst_structField_air.append(StructField(field, StringType(), False))
#     else:
#         lst_structField_air.append(StructField(field, DoubleType(), True))
# schema_air = StructType(lst_structField_air)
# parsed_air = spark.read.option("header", "true").schema(schema_wdi).
 ↪csv(datafile)
parsed_air = spark.createDataFrame(df_air_pollution)
parsed_air.show(5)
parsed_air.createOrReplaceTempView("pollutions");
# print(parsed_air.columns)
# parsed_air.show(5)

IHMEparsed_ri.createOrReplaceTempView('IHMEparsed_ri')
IHMEparsed_ri.show(4)
IHMEparsed_cr.createOrReplaceTempView('IHMEparsed_cr')
IHMEparsed_cr.show(4)
```

```
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+-----------------+-------------------
---+----------------------+---------------------+---------------------+-------
-----------+-------------------+-----------------+-----------+----+
|Nitrogen oxide (NOx)|Sulphur dioxide (SO )|Carbon monoxide (CO)|Organic carbon
(OC)|            NMVOCs| Black carbon (BC)|    Ammonia (NH )|Nitrogen oxide
(NOx).1|Sulphur dioxide (SO ).1|Carbon monoxide (CO).1|Organic carbon (OC).1|
NMVOCs.1|Black carbon (BC).1|  Ammonia (NH ).1|     Entity|Year|
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+-----------------+-------------------
---+----------------------+---------------------+---------------------+-------
-----------+-------------------+-----------------+-----------+----+
|    369593.165109308|     10560.1491957337|   766051.9024026981|
21148.9209786874|324790.07135184004|  6524.42531036713| 75722.0960124105|
29.77633779151264|     0.8507802612852433|     61.71710509047816|
1.703866506300672| 26.16676873080605|
0.5256414627676611|6.100563868598725|Afghanistan|1990|
|      350497.5077086|     9881.08315843062|    724590.512212164|
21775.9941142059|  297031.068717025|6648.0540791168505| 80299.2908819285|
26.35514595279831|     0.7429935536907858|     54.48452067522619|
```

```
1.6374139345501877|22.334815501915703|
0.4998906745519255|6.037987388084088|Afghanistan|1991|
|    224889.350416567|     5981.23475867664|    466237.519516482|
22343.0213674859|  183132.832872316|6514.7240623324205| 86201.7980122156|
15.525089423059049|     0.4129106350156594|    32.186402644103985|
1.542435887110749| 12.64245550700557|
0.4497397206533728|5.950884824422226|Afghanistan|1992|
|    222415.282315777|      5894.1774732922|    463308.283652427|
23349.4432961168|177369.00744575102|  6739.38654489345|92924.37009491948|
14.06214156352411|     0.3726576571851436|    29.292531540273856|
1.476261764213234|11.214103930784562|
0.4260957550167352|5.875116284144709|Afghanistan|1993|
|    222376.597095223|     6251.537336636859|  483645.49216481904|
24295.0732989028|  181305.074647839|  7037.78459638022| 99621.7393839683|
13.022964355910506|     0.3661066360764741|     28.32356501373289|
1.422784041705443|  10.6177068789008| 0.4121513645790223|
5.83411374226436|Afghanistan|1994|
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+-----------------+-------------------
---+-----------------------+----------------------+---------------------+-------
-----------+-------------------+-----------------+-----------+----+
only showing top 5 rows

+-------------+--------------------+----+------------------+
|     location|               cause|year|        Death Rate|
+-------------+--------------------+----+------------------+
|       Guyana|Respiratory infec…|1990|48.524773827237134|
|Guinea-Bissau|Respiratory infec…|1990|260.93392193693967|
|     Honduras|Respiratory infec…|1990| 44.64436203593287|
|       Kuwait|Respiratory infec…|1990|12.763978594785975|
+-------------+--------------------+----+------------------+
only showing top 4 rows

+-----------------+--------------------+----+------------------+
|         location|               cause|year|        Death Rate|
+-----------------+--------------------+----+------------------+
|Brunei Darussalam|Chronic respirato…|1990|28.047355314816443|
|           Kuwait|Chronic respirato…|1990|4.6259092778190425|
|            Haiti|Chronic respirato…|1990| 38.07623894197776|
|       Bangladesh|Chronic respirato…|1990| 56.06083285386855|
+-----------------+--------------------+----+------------------+
only showing top 4 rows
```

[263]:
```
merged_ri_x = spark.sql(""" SELECT distinct a.*, b.cause, b.`Death Rate`
FROM pollutions a INNER JOIN IHMEparsed_ri b ON ((lower(a.Entity) = lower(b.
↪Location)) AND (a.Year = b.Year))
```

```
""")
merged_ri_x.createOrReplaceTempView("merged_ri_x")
merged_ri_x.show(5)
```

```
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+------------------+------------------
----+----------------------+---------------------+---------------------+------
------------+-------------------+------------------+------------------+----+---
-----------------+------------------+
|Nitrogen oxide (NOx)|Sulphur dioxide (SO )|Carbon monoxide (CO)|Organic carbon
(OC)|            NMVOCs| Black carbon (BC)|     Ammonia (NH )|Nitrogen oxide
(NOx).1|Sulphur dioxide (SO ).1|Carbon monoxide (CO).1|Organic carbon (OC).1|
NMVOCs.1|Black carbon (BC).1|   Ammonia (NH ).1|            Entity|Year|
cause|        Death Rate|
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+------------------+------------------
----+----------------------+---------------------+---------------------+------
------------+-------------------+------------------+------------------+----+---
-----------------+------------------+
|    3167.88457235479|     631.160920359483|    8584.33329427952|
81.136821419267|  2305.82962110336|  29.7602352311426|  443.723242937342|
38.298791904186544|      7.630549723260389|     103.7820624346191|
0.9809202855499848|27.876801319027503|  0.359792482997553|
5.364483381942114|Antigua and Barbuda|2006|Respiratory infec…|
30.52505142327932|
|    806884.234471647|     149533.387667925|    2529191.98280047|
52905.5526083757|  623954.441692257|36189.159356150696|  445450.180365389|
21.88410129446749|      4.055605082893383|     68.59607757859283|
1.4348904376345903|16.922727724464014| 0.9815128308092588|12.081382250749048|
Argentina|2000|Respiratory infec…| 51.41609581321555|
|    10079.5241807752|     16377.7678479015|    46764.47308828741|
3124.38797238794|  28513.9759580697|  428.841004447109|  14576.7551840914|
3.304018893053956|      5.368552465872102|     15.32916632137408|
1.0241591472829192| 9.346742325519472| 0.1405719908398009|  4.77818929384781|
Armenia|2001|Respiratory infec…|18.161339739060384|
|    9054.84239691955|     1806.02988556892|    20181.5296223171|
280.903695320677|5764.8299559385605|  94.8274836857312|  1832.85449401107|
32.89284989236368|      6.560629917462829|     73.31193579813174|
1.0204178802202717|20.941467347923997| 0.3444727196584285|    6.658073669682|
Barbados|2004|Respiratory infec…| 58.29728369937335|
|    4101.03469125606|      739.87817565696|    18451.3606734246|
330.76178790906204|  4108.02386728081|  73.0276186525865|1682.2881087102298|
21.104108042527223|     3.8074462014828847|     94.95152772392808|
1.7021149621717442|21.140074655116248| 0.3758033935725206| 8.657129889824365|
Belize|1992|Respiratory infec…| 42.84084381164416|
+--------------------+---------------------+--------------------+---------------
----+------------------+------------------+------------------+------------------
```

```
----+-----------------------+----------------------+---------------------+------
------------+-------------------+-------------------+-------------------+----+---
-----------------+------------------+
only showing top 5 rows
```

[264]:

```python
merged_cr_x = spark.sql(""" SELECT distinct a.*, b.cause, b.`Death Rate`
FROM pollutions a INNER JOIN IHMEparsed_cr b ON ((lower(a.Entity) = lower(b.
 ↪Location)) AND (a.Year = b.Year))
""")
merged_cr_x.createOrReplaceTempView("merged_cr_x")
merged_cr_x.show(5)
```

```
+--------------------+---------------------+--------------------+---------------
----+----------------+-----------------+----------------+----------------------
+-----------------------+----------------------+---------------------+----------
--------+-------------------+-------------------+-------------------+----+-------
-------------+------------------+
|Nitrogen oxide (NOx)|Sulphur dioxide (SO )|Carbon monoxide (CO)|Organic carbon
(OC)|          NMVOCs|Black carbon (BC)|   Ammonia (NH )|Nitrogen oxide
(NOx).1|Sulphur dioxide (SO ).1|Carbon monoxide (CO).1|Organic carbon (OC).1|
NMVOCs.1|Black carbon (BC).1|   Ammonia (NH ).1|             Entity|Year|
cause|        Death Rate|
+--------------------+---------------------+--------------------+---------------
----+----------------+-----------------+----------------+----------------------
+-----------------------+----------------------+---------------------+----------
--------+-------------------+-------------------+-------------------+----+-------
-------------+------------------+
|    1165043.39059467|     1295551.99629241|    2096402.82862448|
29416.0056269422|805467.891824373|  25740.379343289|566361.469648237|
50.09880882777239|       55.71089653198472|     90.14881796260852|
1.264937301286808|34.636462689016966| 1.1068792409659087|24.354487759327448|
Australia|2013|Chronic respirato…|41.376906115400715|
|    77168.4595604618|     94038.1052737655|    141373.970758497|
32061.250263080296|103678.305950474| 4603.99438005617|55284.2897057991|
9.323167035713965|      11.361286310276109|    17.080205454288084|
3.8735047100729902|12.525974599749306| 0.5562351364953845| 6.679214154509975|
Azerbaijan|2002|Chronic respirato…| 19.09138586318142|
|     464362.23784385|     636524.8526003689|    1595471.94968175|
12588.0493542503|578233.646895256| 16469.4359765098|296801.910053225|
45.74485886000432|       62.70479632084184|    157.17177928199655|
1.2400632396525413| 56.96246251234527| 1.6224231060378769|29.238297988670723|
Belarus|1990|Chronic respirato…| 63.49405772574148|
|    98452.2777170772|      3321.85088681692|    325589.662718754|
32482.294712499|94423.9507151521| 9049.26162902681|110906.927634647|
4.55650086402944|      0.1537396268215825|    15.068717696510188|
1.503323309070462|  4.37006459368116| 0.4188117267330321| 5.132918438376198|
Afghanistan|2001|Chronic respirato…| 37.37684246541206|
```

```
|    5835.83654939336|    779.9656962652069|    13658.3339236729|
76.5324431257261|3336.83059291867| 34.9782010501062|371.128146643245|
62.36800450346111|      8.335549435885124|     145.9675959824401|
0.8179077184782262| 35.66094829507721| 0.3738145477776897|
3.966273168430871|Antigua and Barbuda|2015|Chronic
respirato…|11.226358566238698|
+--------------------+---------------------+-------------------+---------------
----+----------------+----------------+----------------+---------------------
+----------------------+---------------------+--------------------+----------
--------+-------------------+-----------------+------------------+----+-------
-------------+------------------+
only showing top 5 rows
```

[265]:
```
parsed_wdi = spark.createDataFrame(df_wdi_filtered)
parsed_wdi.createOrReplaceTempView("wdidata")
parsed_wdi.show(5)
```

```
+-----------+----+----+--------+-----------+---+-----------+
|     Entity|Code|Year| Surface|      PM2.5|GDP| Population|
+-----------+----+----+--------+-----------+---+-----------+
|Afghanistan| AFG|1990|652860.0|49.28239771|NaN|1.0694796E7|
|Afghanistan| AFG|1991|652860.0|        NaN|NaN|1.0745167E7|
|Afghanistan| AFG|1992|652860.0|        NaN|NaN|1.2057433E7|
|Afghanistan| AFG|1993|652860.0|        NaN|NaN| 1.400376E7|
|Afghanistan| AFG|1994|652860.0|        NaN|NaN|1.5455555E7|
+-----------+----+----+--------+-----------+---+-----------+
only showing top 5 rows
```

[266]:
```
merged_ri = spark.sql(""" SELECT distinct a.GDP, a.Surface, a.`PM2.5`, a.
 ↪Population, b.*
FROM wdidata a INNER JOIN merged_ri_x b ON ((lower(a.Entity) = lower(b.Entity))␣
 ↪AND (a.Year = b.Year))
""")
merged_ri.createOrReplaceTempView("merged_ri")
merged_ri.show(5)
```

```
+------------------+---------+-----------+-----------+--------------------+-----
----------------+--------------------+-------------------+------------------+---
---------------+-------------------+----------------------+----------------------
-+----------------------+---------------------+-------------------+--------------
-----+------------------+-------------------+----+--------------------+---------
---------+
|                GDP|  Surface|      PM2.5| Population|Nitrogen oxide
(NOx)|Sulphur dioxide (SO )|Carbon monoxide (CO)|Organic carbon (OC)|
NMVOCs| Black carbon (BC)|     Ammonia (NH )|Nitrogen oxide (NOx).1|Sulphur
dioxide (SO ).1|Carbon monoxide (CO).1|Organic carbon (OC).1|
```

```
NMVOCs.1|Black carbon (BC).1|   Ammonia (NH ).1|            Entity|Year|
cause|        Death Rate|
+------------------+---------+-----------+-----------+--------------------+-----
----------------+--------------------+-------------------+------------------+---
---------------+------------------+----------------------+---------------------
-+----------------------+---------------------+------------------+--------------
-----+------------------+-------------------+----+-------------------+---------
---------+
|1.15766296296296E9|    440.0|        NaN|    80895.0|    3167.88457235479|
631.160920359483|    8584.33329427952|    81.136821419267|  2305.82962110336|
29.7602352311426|  443.723242937342|    38.298791904186544|
7.630549723260389|     103.7820624346191|
0.9809202855499848|27.876801319027503|  0.359792482997553|
5.364483381942114|Antigua and Barbuda|2006|Respiratory infec…|
30.52505142327932|
|      2.8420375E11|2780400.0|14.16360006|3.7070774E7|    806884.234471647|
149533.387667925|    2529191.98280047|   52905.5526083757|
623954.441692257|36189.159356150696|  445450.180365389|     21.88410129446749|
4.055605082893383|     68.59607757859283|
1.4348904376345903|16.922727724464014| 0.9815128308092588|12.081382250749048|
Argentina|2000|Respiratory infec…| 51.41609581321555|
|2.11846791337873E9|  29740.0|        NaN|  3133133.0|    10079.5241807752|
16377.7678479015|   46764.47308828741|   3124.38797238794|  28513.9759580697|
428.841004447109|  14576.7551840914|     3.304018893053956|
5.368552465872102|     15.32916632137408|   1.0241591472829192|
9.346742325519472| 0.1405719908398009|  4.77818929384781|
Armenia|2001|Respiratory infec…|18.161339739060384|
|          3.4445E9|    430.0|        NaN|   268505.0|    9054.84239691955|
1806.02988556892|    20181.5296223171|   280.903695320677|5764.8299559385605|
94.8274836857312|  1832.85449401107|     32.89284989236368|
6.560629917462829|     73.31193579813174|
1.0204178802202717|20.941467347923997| 0.3444727196584285|    6.658073669682|
Barbados|2004|Respiratory infec…| 58.29728369937335|
| 6.9362516824559E8|  22970.0|        NaN|   190299.0|    4101.03469125606|
739.87817565696|    18451.3606734246| 330.76178790906204|  4108.02386728081|
73.0276186525865|1682.2881087102298|    21.104108042527223|
3.8074462014828847|     94.95152772392808|
1.7021149621717442|21.140074655116248| 0.3758033935725206| 8.657129889824365|
Belize|1992|Respiratory infec…| 42.84084381164416|
+------------------+---------+-----------+-----------+--------------------+-----
----------------+--------------------+-------------------+------------------+---
---------------+------------------+----------------------+---------------------
-+----------------------+---------------------+------------------+--------------
-----+------------------+-------------------+----+-------------------+---------
---------+
only showing top 5 rows
```

[267]:
```
merged_cr = spark.sql(""" SELECT distinct a.GDP, a.Surface, a.`PM2.5`, a.
↪Population, b.*
FROM wdidata a INNER JOIN merged_cr_x b ON ((lower(a.Entity) = lower(b.Entity))␣
↪AND (a.Year = b.Year))
""")
merged_cr.createOrReplaceTempView("merged_cr")
merged_cr.show(5)
```

```
+------------------+---------+-----------+-----------+-------------------+----
-----------------+--------------------+-------------------+----------------+----
-------------+----------------+---------------------+----------------------+--
--------------------+---------------------+------------------+------------------
-+------------------+-------------------+----+-------------------+-------------
-----+
|                GDP|  Surface|      PM2.5| Population|Nitrogen oxide
(NOx)|Sulphur dioxide (SO )|Carbon monoxide (CO)|Organic carbon (OC)|
NMVOCs|Black carbon (BC)|   Ammonia (NH )|Nitrogen oxide (NOx).1|Sulphur dioxide
(SO ).1|Carbon monoxide (CO).1|Organic carbon (OC).1|          NMVOCs.1|Black
carbon (BC).1|   Ammonia (NH ).1|             Entity|Year|               cause|
Death Rate|
+------------------+---------+-----------+-----------+-------------------+----
-----------------+--------------------+-------------------+----------------+----
-------------+----------------+---------------------+----------------------+--
--------------------+---------------------+------------------+------------------
-+------------------+-------------------+----+-------------------+-------------
-----+
|1.57730184020001E12|7741220.0|6.964302296|2.3128129E7|    1165043.39059467|
1295551.99629241|    2096402.82862448|   29416.0056269422|805467.891824373|
25740.379343289|566361.469648237|     50.09880882777239|      55.71089653198472|
90.14881796260852|    1.264937301286808|34.636462689016966|
1.1068792409659087|24.354487759327448|          Australia|2013|Chronic
respirato…|41.376906115400715|
| 6.23608773828284E9|  86600.0|        NaN|  8171950.0|    77168.4595604618|
94038.1052737655|    141373.970758497| 32061.250263080296|103678.305950474|
4603.99438005617|55284.2897057991|     9.323167035713965|
11.361286310276109|    17.080205454288084|
3.8735047100729902|12.525974599749306| 0.5562351364953845| 6.679214154509975|
Azerbaijan|2002|Chronic respirato…| 19.09138586318142|
|                NaN| 207600.0| 25.2079681|1.0189348E7|     464362.23784385|
636524.8526003689|    1595471.94968175|   12588.0493542503|578233.646895256|
16469.4359765098|296801.910053225|     45.74485886000432|
62.70479632084184|    157.17177928199655|   1.2400632396525413|
56.96246251234527| 1.6224231060378769|29.238297988670723|
Belarus|1990|Chronic respirato…| 63.49405772574148|
|                NaN| 652860.0|        NaN|1.9688632E7|    98452.2777170772|
3321.85088681692|    325589.662718754|    32482.294712499|94423.9507151521|
9049.26162902681|110906.927634647|      4.55650086402944|
```

```
0.1537396268215825|    15.068717696510188|    1.503323309070462|
4.37006459368116| 0.4188117267330321| 5.132918438376198|
Afghanistan|2001|Chronic respirato…| 37.37684246541206|
| 1.43775555555556E9|    440.0|18.14588457|    89941.0|    5835.83654939336|
779.9656962652069|    13658.3339236729|   76.5324431257261|3336.83059291867|
34.9782010501062|371.128146643245|     62.36800450346111|
8.335549435885124|     145.9675959824401|   0.8179077184782262|
35.66094829507721| 0.3738145477776897| 3.966273168430871|Antigua and
Barbuda|2015|Chronic respirato…|11.226358566238698|
+-------------------+---------+-----------+-----------+--------------------+----
-----------------+--------------------+-------------------+----------------+----
-------------+----------------+----------------------+-----------------------+--
--------------------+---------------------+------------------+------------------
-+-------------------+-------------------+----+-------------------+-------------
-----+
only showing top 5 rows
```

[277]:
```
merged_cr.columns
```

[277]:
```
['GDP',
 'Surface',
 'PM2.5',
 'Population',
 'Nitrogen oxide (NOx)',
 'Sulphur dioxide (SO )',
 'Carbon monoxide (CO)',
 'Organic carbon (OC)',
 'NMVOCs',
 'Black carbon (BC)',
 'Ammonia (NH )',
 'Nitrogen oxide (NOx).1',
 'Sulphur dioxide (SO ).1',
 'Carbon monoxide (CO).1',
 'Organic carbon (OC).1',
 'NMVOCs.1',
 'Black carbon (BC).1',
 'Ammonia (NH ).1',
 'Entity',
 'Year',
 'cause',
 'Death Rate']
```

3.2 2.2 Analyse

[281]:
```
df_cr = merged_cr.toPandas().drop(columns=['Entity', 'Year','cause'])
df_cr['Death Rate'] = pd.to_numeric(df_cr['Death Rate'])
df_ri = merged_ri.toPandas().drop(columns=['Entity', 'Year','cause'])
df_ri['Death Rate'] = pd.to_numeric(df_ri['Death Rate'])
```

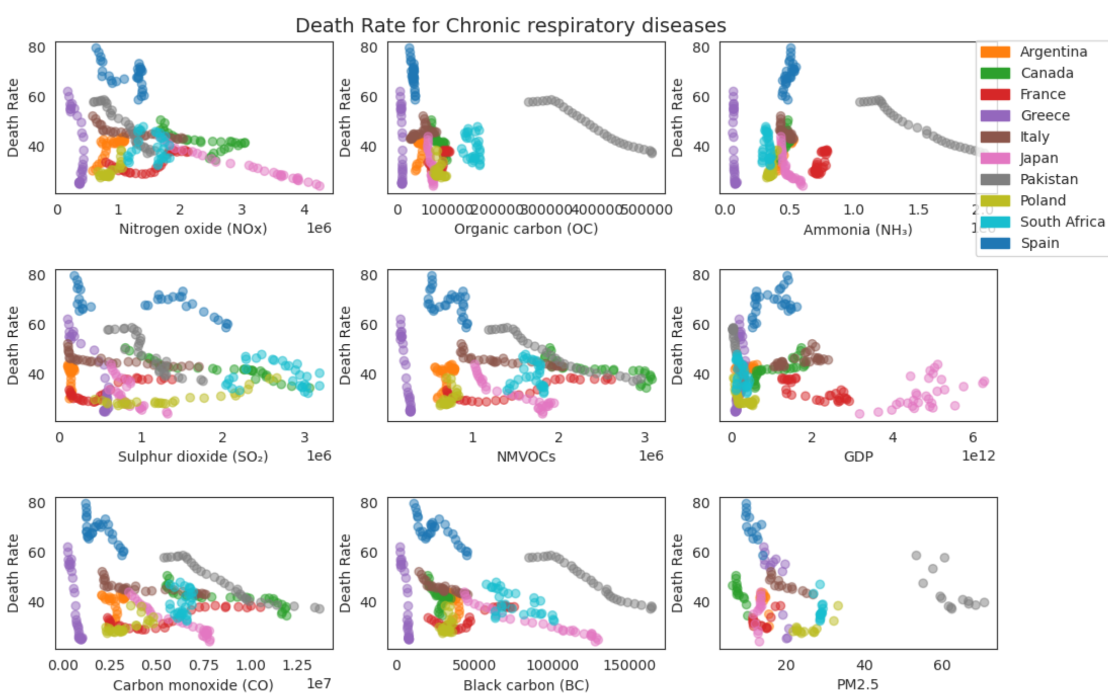
[339]:
```
import numpy as np
import matplotlib.patches as mpatches

from matplotlib import colors as mcolors
from pyspark.sql.functions import  expr, split, min, max, stddev, mean,␣
↪avg,median, col, broadcast, when, lower

indicators = [
    "Nitrogen oxide (NOx)",        "Sulphur dioxide (SO )",
    "Carbon monoxide (CO)",        "Organic carbon (OC)",
    "NMVOCs",        "Black carbon (BC)",        "Ammonia (NH )",
    "GDP", "PM2.5"
  ]

def draw_pigure(title):
    fig = plt.figure(constrained_layout=True)
    colors = list(mcolors.TABLEAU_COLORS.values())

    fig, axs = plt.subplots(3,3, figsize=(12, 8))
    plt.subplots_adjust(bottom=0.2, top=0.95, hspace=0.5)
    m=0
    n=0
    labels = {}
    for indicator in indicators:
        xx = selected_merged_p.toPandas()[[indicator, 'Death Rate', 'Entity']].
↪dropna()
        l = xx.values.tolist()
        x = [x[0] for x in l]
        y = [float(x[1]) for x in l]
        last = ''
        c = []
        k = 0
        for i, a in enumerate([x[2] for x in l]):
            if last != a:
                k += 1
                last = a
                labels[a] = colors[k%len(colors)]
            c.append(colors[k%len(colors)])

        axs[m,n].scatter(x, y, alpha=0.5, c=c)
```

```
        axs[m,n].set(xlabel=indicator, ylabel="Death Rate")

        m += 1
        if m > 2:
            m = 0
            n += 1


    handles = []
    for label, color in labels.items():
        handles.append(mpatches.Patch(color=color, label=label))
    fig.legend(handles=handles, bbox_to_anchor=(1,0.96))

    plt.suptitle(title, fontsize=14)
    plt.savefig("../fig/"+title+".png")
    plt.show()
```

[283]:
```
# selected_countries = ['poland', 'pakistan', 'france', 'japan']
selected_countries = ['poland', 'japan', 'pakistan', 'italy', 'france', 'south␣
↪africa', 'argentina', 'spain', 'canada', 'greece']
```

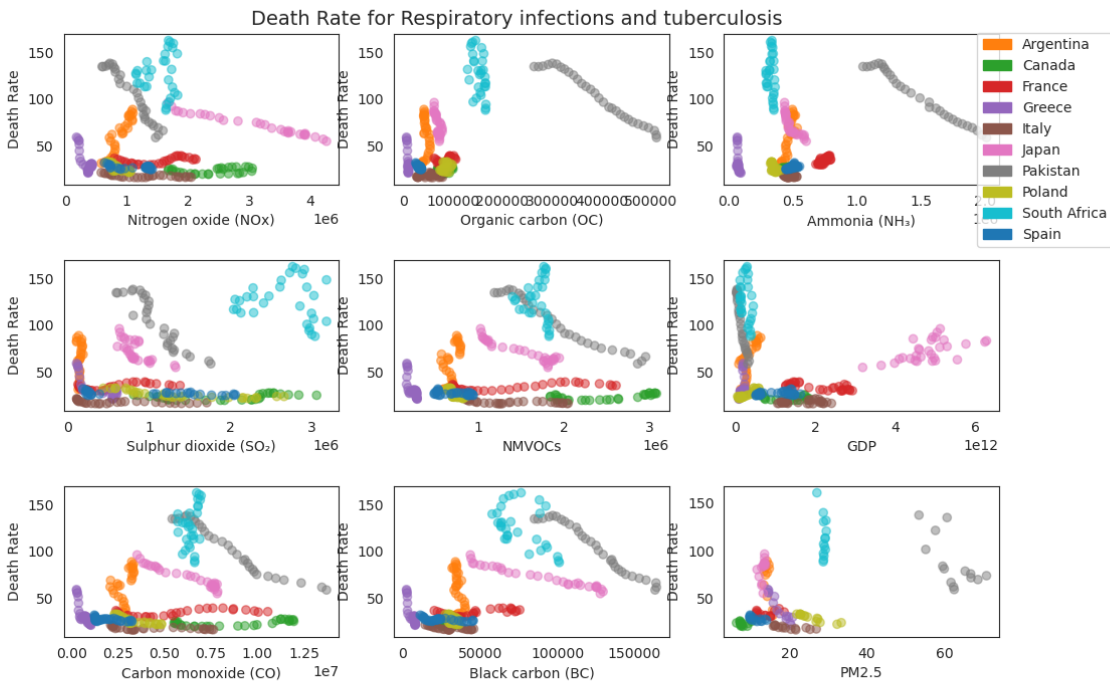
[340]:
```
selected_merged_p = merged_cr.filter(lower(merged_cr["Entity"]).
↪isin(selected_countries)).sort('Entity')
selected_merged_p.count()
draw_pigure("Death Rate for Chronic respiratory diseases")
```

```
<Figure size 640x480 with 0 Axes>
```



[341]:
```python
selected_merged_p = merged_ri.filter(lower(merged_ri["Entity"]).isin(selected_countries)).sort('Entity')
selected_merged_p.count()
draw_pigure("Death Rate for Respiratory infections and tuberculosis")
```

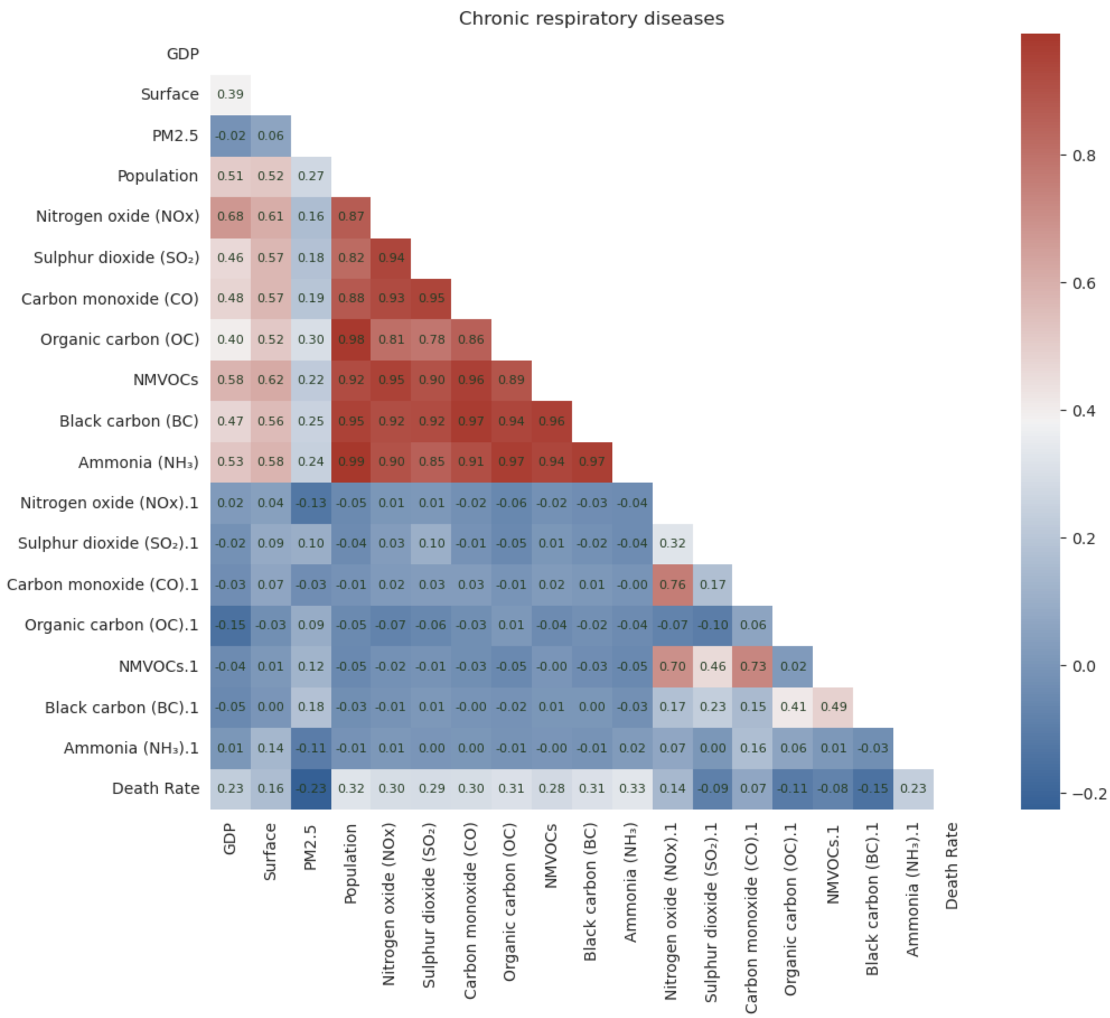
```
<Figure size 640x480 with 0 Axes>
```



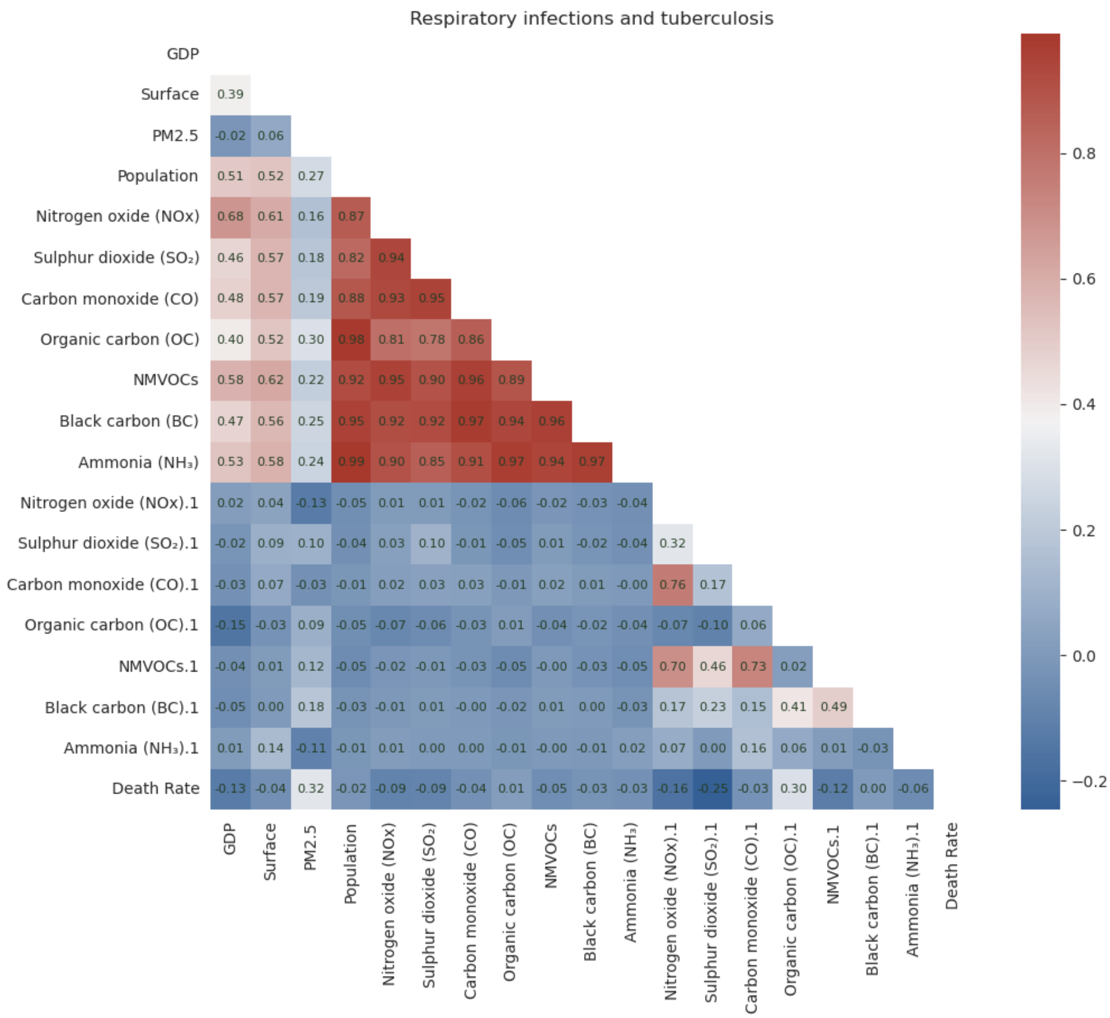
[354]:
```python
import numpy as np
def draw_dcorr(data, title):
    dcorr = data.corr(method='pearson')
    plt.figure(figsize=(11, 9),dpi=100)
    cmap = 'RdBu'
    cmap = sns.diverging_palette(250, 15, s=75, l=40, n=9, center="light", as_cmap=True)
    plt.title(title)
    sns.heatmap(data=dcorr,
                annot=True,#
                fmt=".2f",#
                annot_kws={'size':8,'weight':'normal', 'color':'#253D24'},
                mask=np.triu(np.ones_like(dcorr,dtype=np.bool_)),
                cmap = cmap
               )
```

```
    plt.savefig("../fig/dcorr_"+title+".png")
    # sns.clustermap(data=dcorr,
    #                vmax=0.7,
    #                cmap=cmap,
    #                linewidths=.75,

    #               )
def draw_corr_cluster(data, title):
    dcorr = data.corr(method='pearson')
    plt.figure(figsize=(11, 9),dpi=100)
    cmap = 'RdBu'
    cmap = sns.diverging_palette(250, 15, s=75, l=40, n=9, center="light",␣
 ↪as_cmap=True)
    rel = sns.clustermap(data=dcorr,
                   vmax=0.7,
                   cmap=cmap,
                   linewidths=.75,
                  )
    # rel.fig.suptitle(title)

    plt.savefig("../fig/corrCluster_"+title+".png")
```

[343]:
```
draw_dcorr(df_cr, "Chronic respiratory diseases")
```

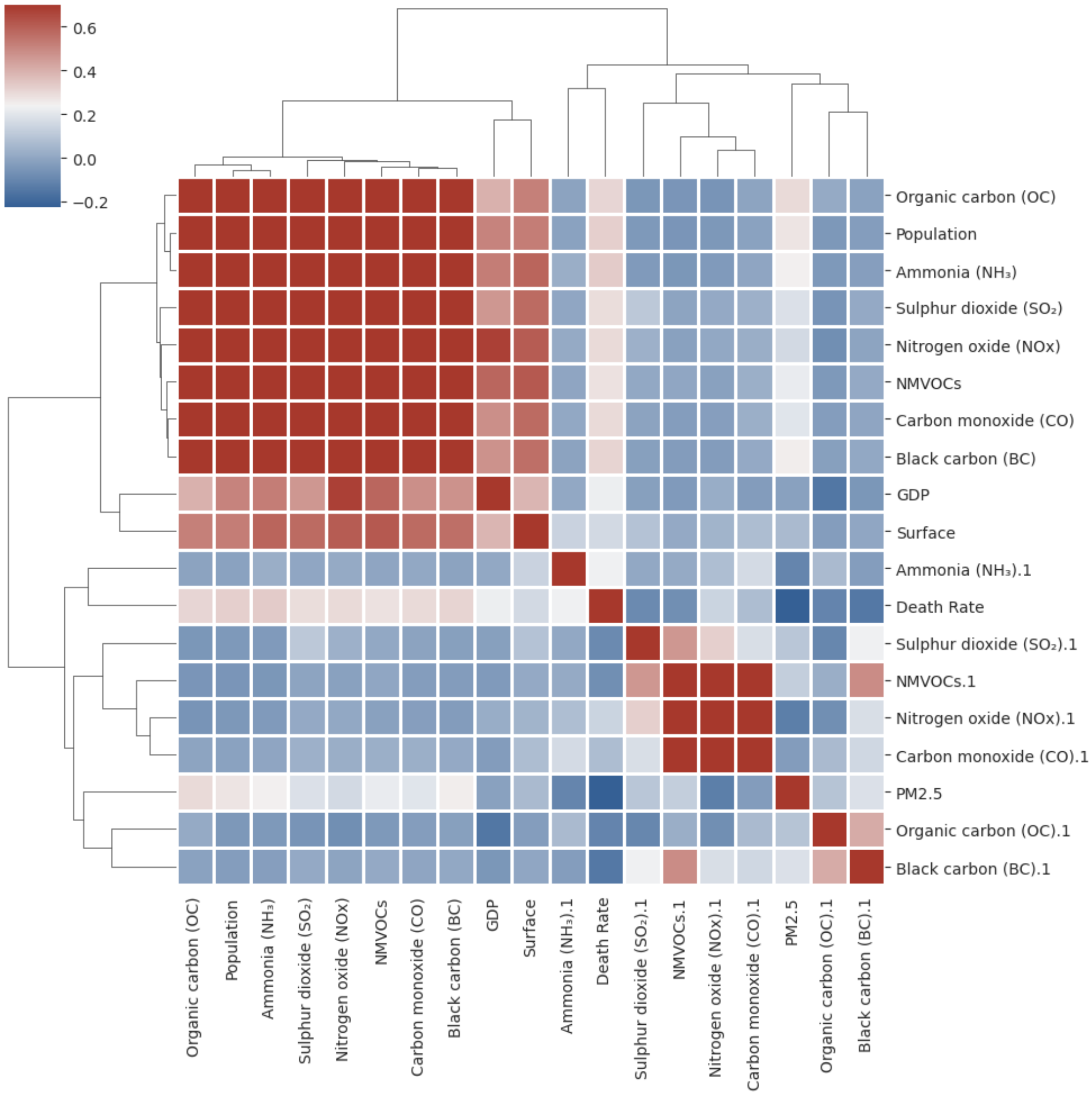


[344]:
```
draw_dcorr(df_ri, "Respiratory infections and tuberculosis")
```



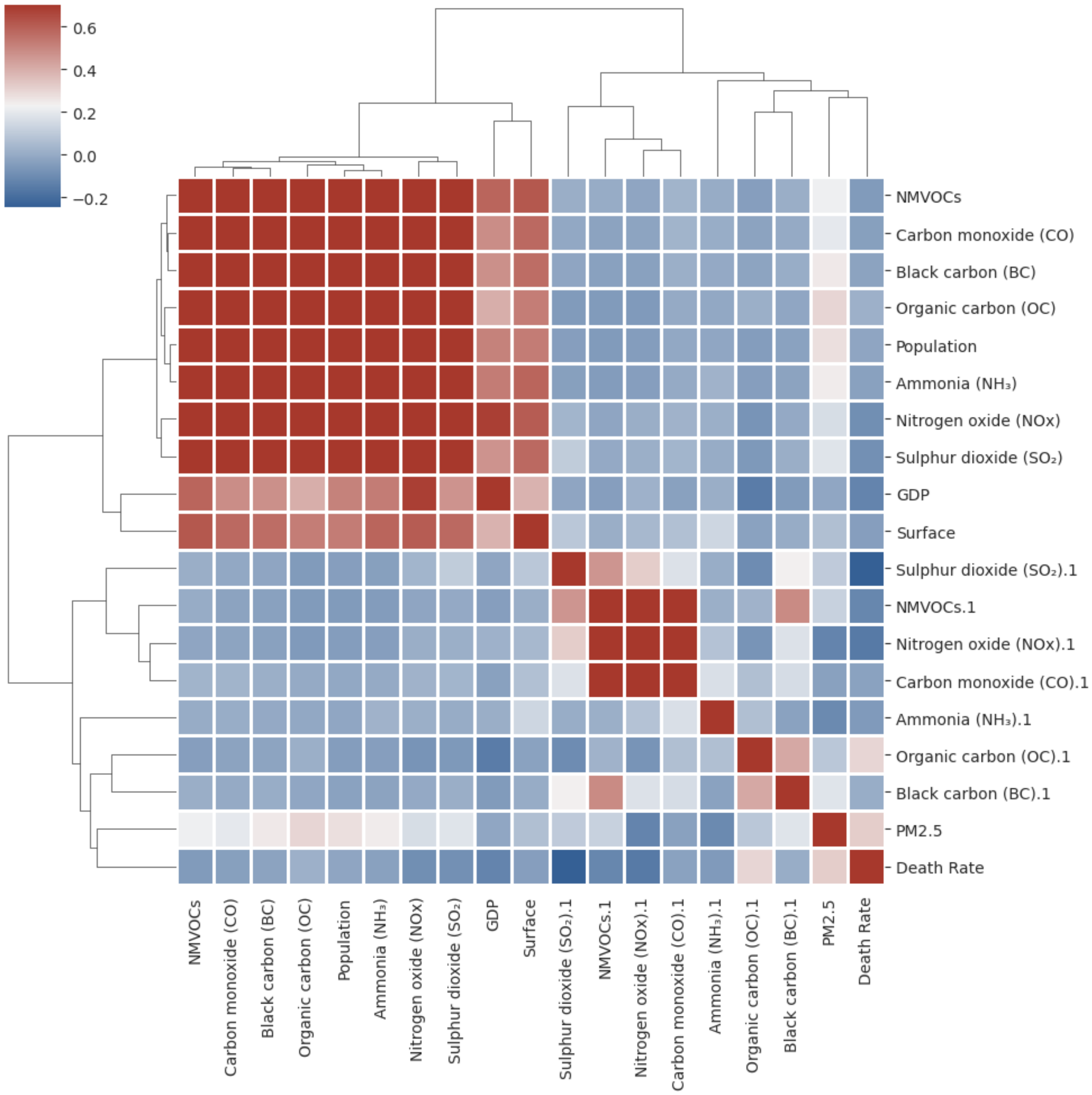

```
[355]: draw_corr_cluster(df_cr, "Chronic respiratory diseases")
```

```
<Figure size 1100x900 with 0 Axes>
```



[356]:
```
draw_corr_cluster(df_ri, "Respiratory infections and tuberculosis")
```

```
<Figure size 1100x900 with 0 Axes>
```



[ ]:

```

```

[ ]:

```

```

4 Partie 3 Machine Learning

[315]:

```
!pip install seaborn
!pip install xgboost
```

```
Requirement already satisfied: seaborn in /opt/conda/lib/python3.11/site-
packages (0.13.0)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
```

```
/opt/conda/lib/python3.11/site-packages (from seaborn) (1.24.4)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.11/site-
packages (from seaborn) (2.0.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.3 in
/opt/conda/lib/python3.11/site-packages (from seaborn) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.3->seaborn)
(1.1.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.11/site-
packages (from matplotlib!=3.6.1,>=3.3->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.3->seaborn)
(4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.3->seaborn)
(1.4.5)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.3->seaborn)
(23.2)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.11/site-
packages (from matplotlib!=3.6.1,>=3.3->seaborn) (10.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.3->seaborn)
(3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.3->seaborn)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.2->seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.11/site-
packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.3->seaborn) (1.16.0)
Requirement already satisfied: xgboost in /opt/conda/lib/python3.11/site-
packages (2.0.3)
Requirement already satisfied: numpy in /opt/conda/lib/python3.11/site-packages
(from xgboost) (1.24.4)
Requirement already satisfied: scipy in /opt/conda/lib/python3.11/site-packages
(from xgboost) (1.11.3)
```

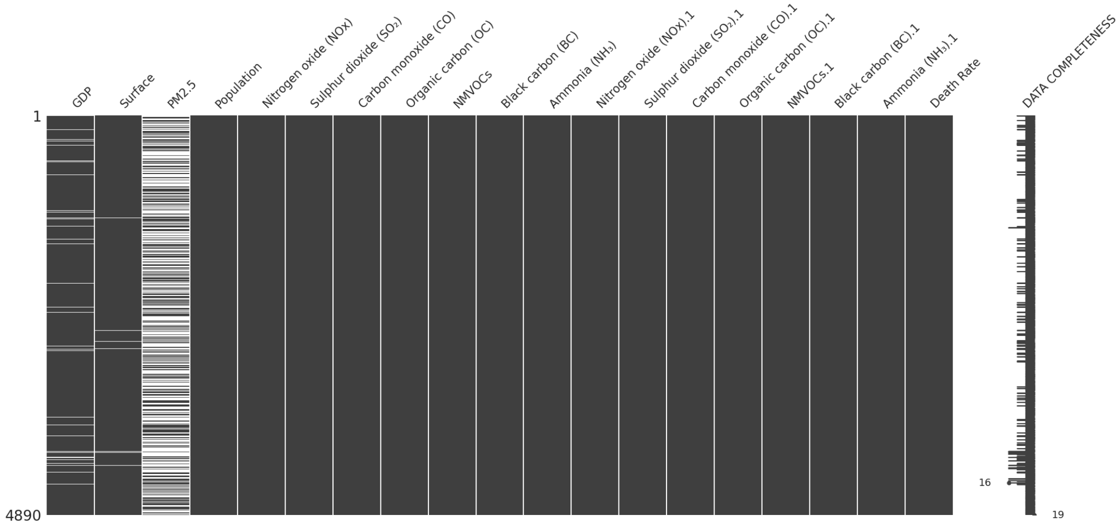
[307]:
```
print("cr:\n---------\n", df_cr.isna().mean())
print("ri:\n---------\n", df_ri.isna().mean())
```

```
cr:
---------
 GDP                          0.036196
Surface                      0.007157
PM2.5                        0.533333
```

```
Population                  0.000000
Nitrogen oxide (NOx)        0.000000
Sulphur dioxide (SO )      0.000000
Carbon monoxide (CO)        0.000000
Organic carbon (OC)         0.000000
NMVOCs                      0.000000
Black carbon (BC)           0.000000
Ammonia (NH )              0.000000
Nitrogen oxide (NOx).1      0.000000
Sulphur dioxide (SO ).1    0.000000
Carbon monoxide (CO).1      0.000000
Organic carbon (OC).1       0.000000
NMVOCs.1                    0.000000
Black carbon (BC).1         0.000000
Ammonia (NH ).1            0.000000
Death Rate                  0.000000
dtype: float64
ri:
---------
 GDP                        0.036196
Surface                     0.007157
PM2.5                       0.533333
Population                  0.000000
Nitrogen oxide (NOx)        0.000000
Sulphur dioxide (SO )      0.000000
Carbon monoxide (CO)        0.000000
Organic carbon (OC)         0.000000
NMVOCs                      0.000000
Black carbon (BC)           0.000000
Ammonia (NH )              0.000000
Nitrogen oxide (NOx).1      0.000000
Sulphur dioxide (SO ).1    0.000000
Carbon monoxide (CO).1      0.000000
Organic carbon (OC).1       0.000000
NMVOCs.1                    0.000000
Black carbon (BC).1         0.000000
Ammonia (NH ).1            0.000000
Death Rate                  0.000000
dtype: float64
```

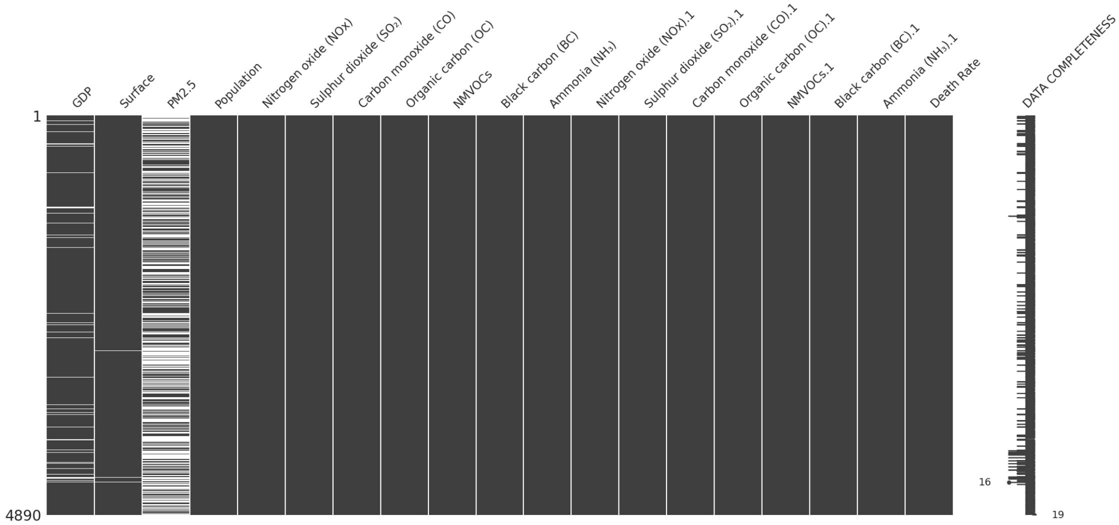
[313]:
```
import missingno as msno
msno.matrix(df_cr.dropna(subset=['Death Rate'], how='any'), labels=True)
```

[313]:
```
<Axes: >
```




```
[314]: msno.matrix(df_ri.dropna(subset=['Death Rate'], how='any'), labels=True)
```

```
[314]: <Axes: >
```



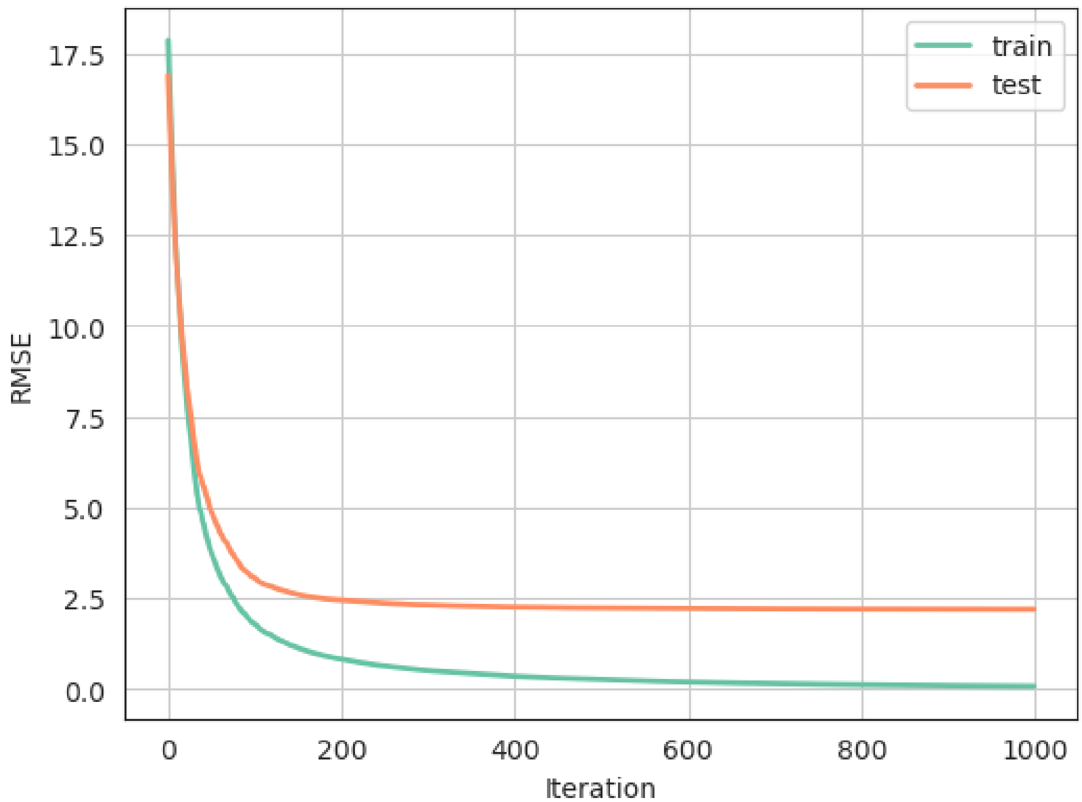

```
[371]: import matplotlib.pyplot as plt
       # from sklearn.preprocessing import MinMaxScaler
       import xgboost as xgb
       import seaborn as sns
       from sklearn.model_selection import train_test_split

       def train_xgb(df):
```

```
    X = df.loc[:, df.columns != 'Death Rate']
    # X = X.drop(columns = ['Entity', 'Year', 'Code'])
    y = df.loc[:, df.columns =='Death Rate']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,␣
 ↪random_state=42)
    reg_mod = xgb.XGBRegressor(
        n_estimators=1000,
        learning_rate=0.08,
        subsample=0.75,
        colsample_bytree=1,
        max_depth=7,
        gamma=0,
    )

    eval_set = [(X_train, y_train), (X_test, y_test)]
    reg_mod.fit(X_train, y_train, eval_set=eval_set, eval_metric='rmse',␣
 ↪verbose=False)
    #
    sns.set_style("white")
    palette = sns.color_palette("Set2", n_colors=2)
    return reg_mod, X_test, y_test

def draw_loss(reg_mod, name):
    plt.plot(reg_mod.evals_result()['validation_0']['rmse'], label='train',␣
 ↪color=palette[0], linewidth=2)
    plt.plot(reg_mod.evals_result()['validation_1']['rmse'], label='test',␣
 ↪color=palette[1], linewidth=2)
    plt.xlabel('Iteration')
    plt.ylabel('RMSE')
    plt.legend()
    plt.grid()
    plt.savefig("../fig/Loss_"+name+".png")
    plt.show()
```

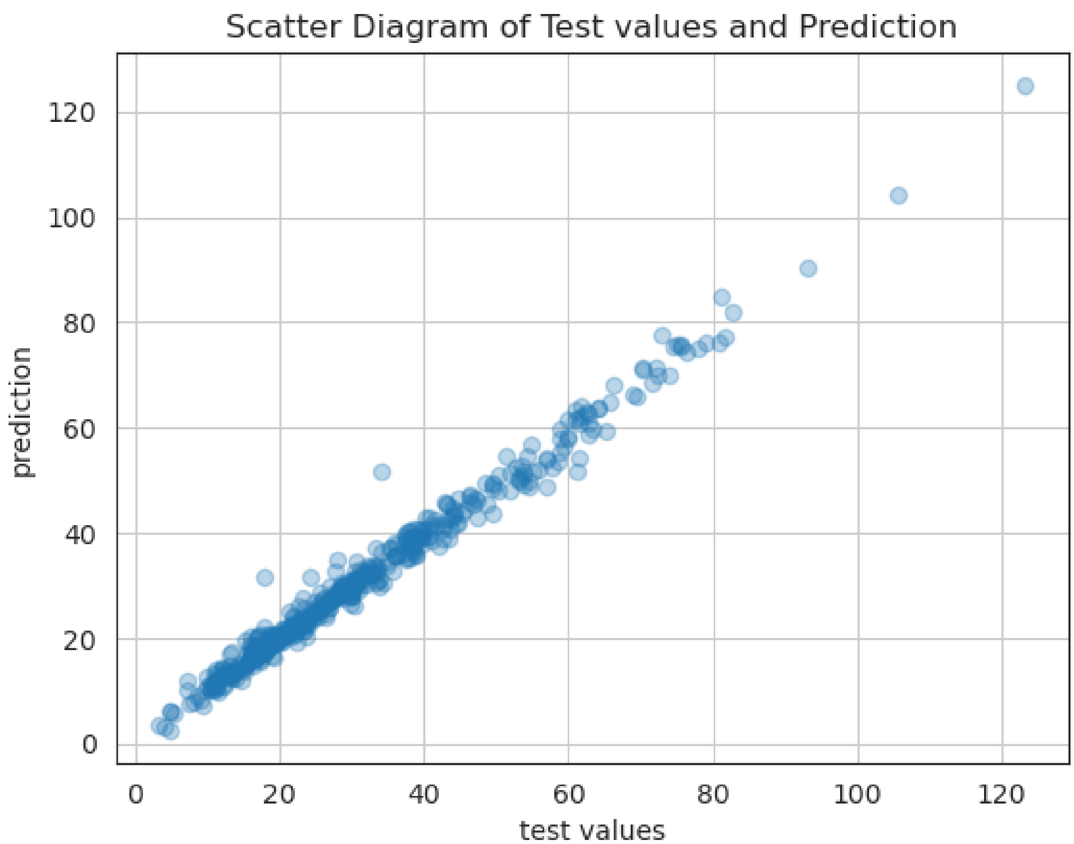
[323]:
```
model_cr, X_test_cr, y_test_cr=train_xgb(df_cr)
```

```
/opt/conda/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning:
`eval_metric` in `fit` method is deprecated for better compatibility with
scikit-learn, use `eval_metric` in constructor or`set_params` instead.
  warnings.warn(
```

[372]:
```
draw_loss(model_cr, "CR")
```



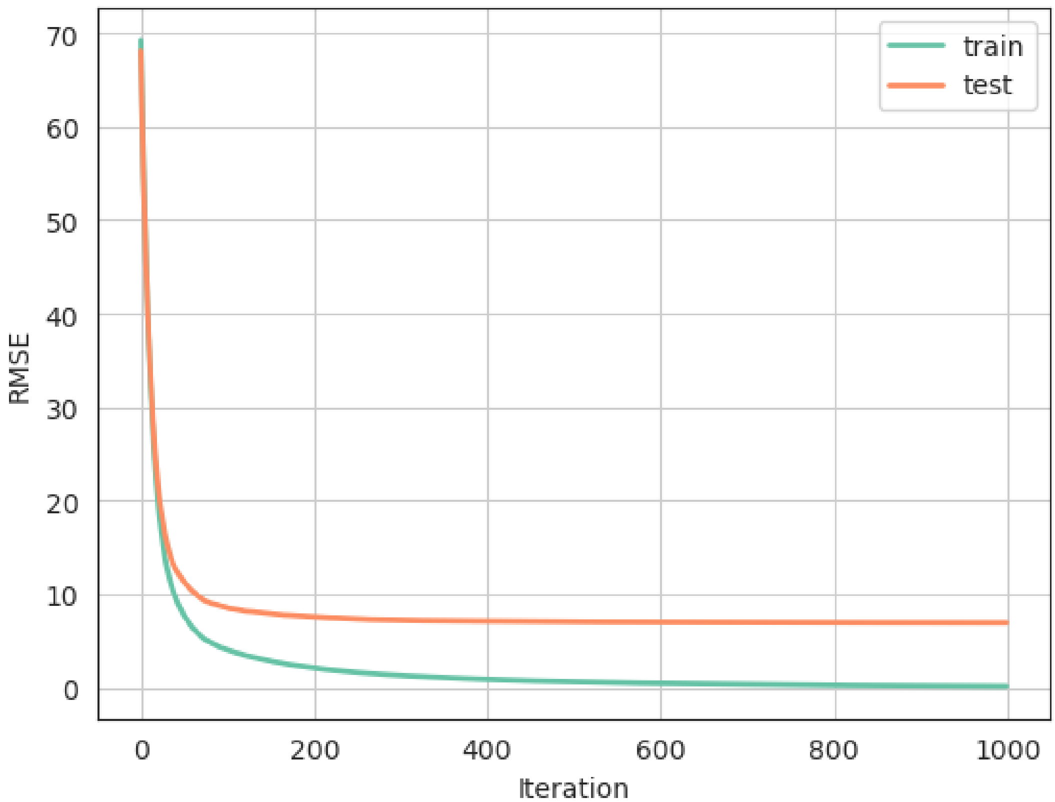
[369]:
```
y_pred_cr = model_cr.predict(X_test_cr)
plt.scatter(y_test_cr, y_pred_cr, alpha=0.3)
plt.grid()
plt.xlabel("test values")
plt.ylabel("prediction")
plt.title("Scatter Diagram of Test values and Prediction")
plt.savefig("../fig/yyplot_CR.png")
plt.show()
```



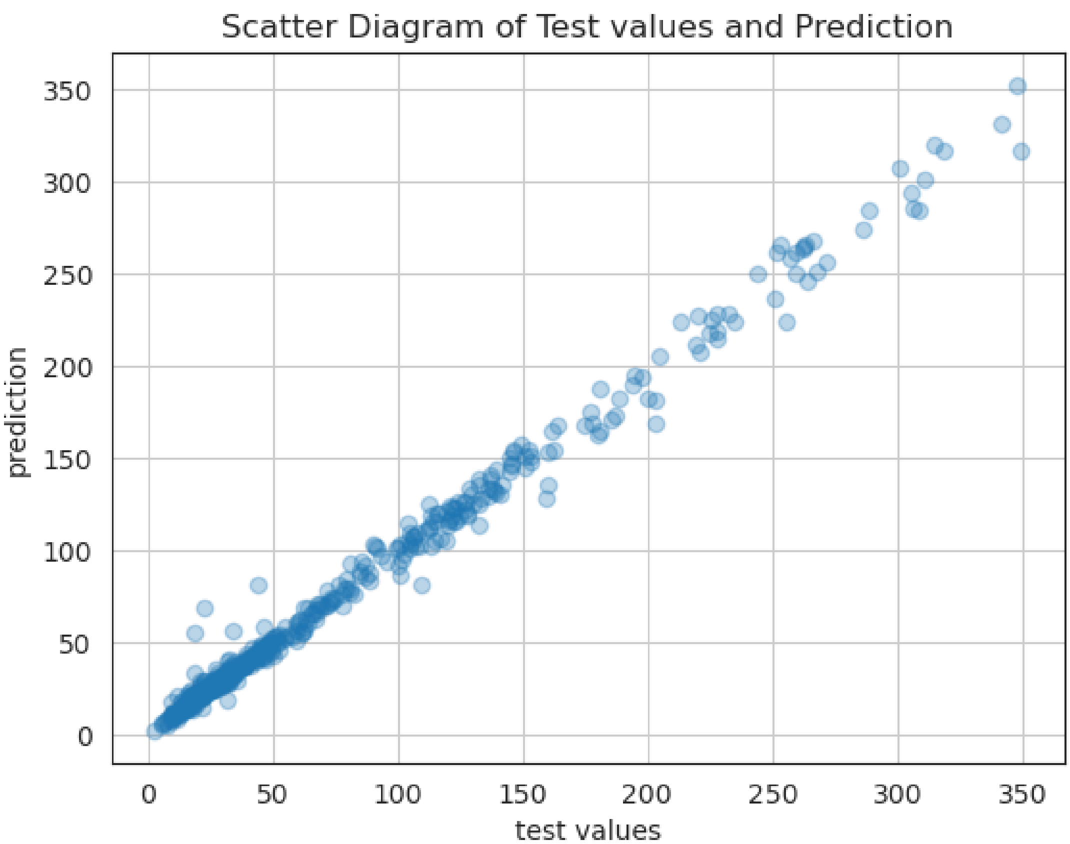
[325]:
```
model_ri, X_test_ri, y_test_ri=train_xgb(df_ri)
```

```
/opt/conda/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning:
`eval_metric` in `fit` method is deprecated for better compatibility with
scikit-learn, use `eval_metric` in constructor or`set_params` instead.
  warnings.warn(
```

[373]:
```
draw_loss(model_ri, "RI")
```



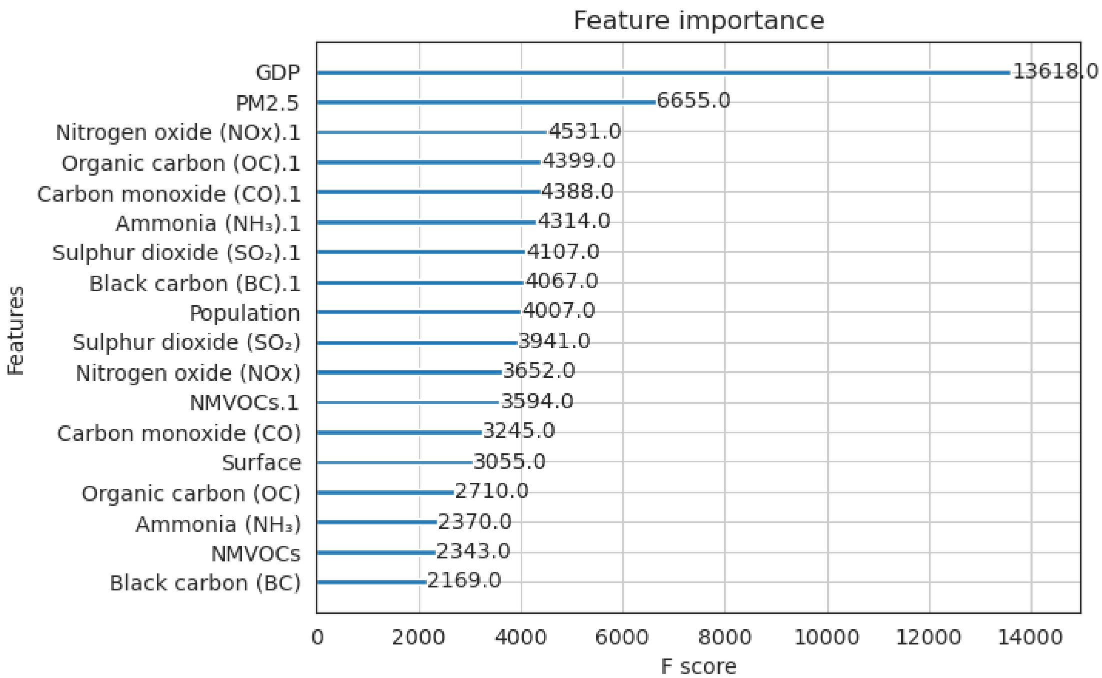
[370]:
```
y_pred_ri = model_ri.predict(X_test_ri)
plt.scatter(y_test_ri, y_pred_ri, alpha=0.3)
plt.grid()
plt.xlabel("test values")
plt.ylabel("prediction")
plt.title("Scatter Diagram of Test values and Prediction")
plt.savefig("../fig/yyplot_RI.png")

plt.show()
```



[375]:
```python
from xgboost import plot_importance

print(model_ri.feature_importances_)
plot_importance(model_ri, )
```

```
[0.02353315 0.11311804 0.0038979  0.02654205 0.01618798 0.01823453
 0.01891412 0.04647917 0.0222679  0.02698714 0.01105512 0.46500936
 0.01357463 0.02344412 0.02287523 0.02072789 0.09967233 0.02747937]
```

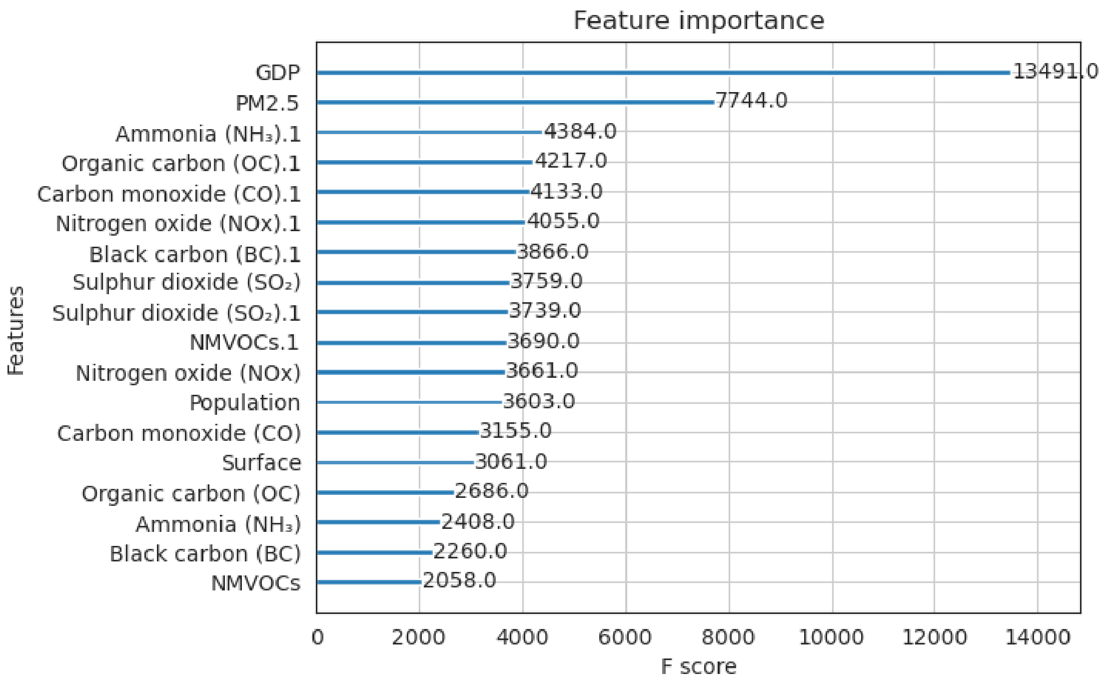
[375]:
```
<Axes: title={'center': 'Feature importance'}, xlabel='F score',
ylabel='Features'>
```



[376]:
```python
from xgboost import plot_importance

print(model_cr.feature_importances_)
plot_importance(model_cr)
```

```
[0.02514318 0.18057905 0.03767888 0.12197781 0.02979152 0.04131323
 0.0404816  0.06039368 0.03939407 0.02948503 0.10309768 0.03331529
 0.0205866  0.0451732  0.05338199 0.05071318 0.02105185 0.06644212]
```

[376]:
```
<Axes: title={'center': 'Feature importance'}, xlabel='F score',
ylabel='Features'>
```


Feature importance
GDP
PM2.5
Ammonia (NH₃).1
Organic carbon (OC).1
Carbon monoxide (CO).1
Nitrogen oxide (NOx).1
Black carbon (BC).1
Sulphur dioxide (SO₂)
Sulphur dioxide (SO₂).1
NMVOCs.1
Nitrogen oxide (NOx)
Population
Carbon monoxide (CO)
Surface
Organic carbon (OC)
Ammonia (NH₃)
Black carbon (BC)
NMVOCs
13491.0
7744.0
4384.0
4217.0
4133.0
4055.0
3866.0
3759.0
3739.0
3690.0
3661.0
3603.0
3155.0
3061.0
2686.0
2408.0
2260.0
2058.0
Features
0
2000
4000
6000
8000
10000
12000
14000
F score

[ ]: