Imports

In [ ]:
```
import pandas as pd
import numpy as np
import geopandas as gpd
import seaborn as sns
import matplotlib.pyplot as plt
from itables import init_notebook_mode
import itables
import warnings
warnings.filterwarnings("ignore", "use_inf_as_na")
init_notebook_mode(all_interactive=True)
import plotly.express as px
```

Ładowanie danych

In [ ]:
```
df = pd.read_excel('data/who_aap_2021_v9_11august2022.xlsx', sheet_name='AAP_202
```

In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['WHO Region', 'ISO3', 'WHO Country Name', 'City or Locality',
       'Measurement Year', 'PM2.5 (µg/m3)', 'PM10 (µg/m3)', 'NO2 (µg/m3)',
       'PM25 temporal coverage (%)', 'PM10 temporal coverage (%)',
       'NO2 temporal coverage (%)', 'Reference',
       'Number and type of monitoring stations', 'Version of the database',
       'Status'],
      dtype='object')
```

Przegląd danych

In [ ]:
```
itables.show(df)
```

10 entries per page

WHO Region	ISO3	WHO Country Name
Eastern Mediterranean Region	AFG	Afghanistan
European Region	ALB	Albania
European Region	ALB	Albania
European Region	ALB	Albania
European Region	ALB	Albania
European Region	ALB	Albania
European Region	ALB	Albania
European Region	ALB	Albania
European Region	ALB	Albania
European Region	ALB	Albania

Showing 1 to 10 of 546 entries (downsampled from 32,191x15 to 546x15 as maxBytes=65536)

Biblioteka ITables 2.0 stanowi użyteczne narzędzie - można przeglądać dane łatwiej niż za pomocą scrollowania i zadecydować ile chcemy ich widzieć na raz. Nie jest ona bezwzględnie potrzebna do analizy danych, ale na pewno stanowi ułatwienie pod względem czytelności i przejrzystości.

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32191 entries, 0 to 32190
Data columns (total 15 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   WHO Region                              32190 non-null  object 
 1   ISO3                                    32191 non-null  object 
 2   WHO Country Name                        32191 non-null  object 
 3   City or Locality                        32191 non-null  object 
 4   Measurement Year                        32191 non-null  int64  
 5   PM2.5 (µg/m3)                           15048 non-null  float64
 6   PM10 (µg/m3)                            21109 non-null  float64
 7   NO2 (µg/m3)                             22200 non-null  float64
 8   PM25 temporal coverage (%)              7275 non-null   float64
 9   PM10 temporal coverage (%)              5381 non-null   float64
 10  NO2 temporal coverage (%)               19890 non-null  float64
 11  Reference                               32186 non-null  object 
 12  Number and type of monitoring stations  8758 non-null   object 
 13  Version of the database                 32191 non-null  int64  
 14  Status                                  0 non-null      float64
dtypes: float64(7), int64(2), object(6)
memory usage: 3.7+ MB
```

In [ ]:
```
df.describe()
```

Out[ ]:

	Measurement Year	PM2.5 (µg/m3)	PM10 (µg/m3)	NO2 (µg/m3)	P
count	32191	15048	21109	22200	
mean	2015.579354	22.92032	30.533252	20.619336	
std	2.752654	17.925906	29.312756	12.133388	
min	2000	0.01	1.04	0	
25%	2014	10.35	16.98	12	
50%	2016	16	22	18.8	
75%	2018	31	31.3	27.16	
max	2021	191.9	540	210.68	

Mimo widocznych wartości odstających zdecydowano o nieusuwaniu ich, ponieważ mogłoby to uniemożliwić zaobserwowanie ważnych wniosków - zanieczyszczenia mogą być na pewnych terenach ekstremalne.

Wartości NaN

In [ ]:
```
print("Ilość wszystkich wierszy:", len(df))
df.isna().sum()
```

```
Ilość wszystkich wierszy: 32191
```

Out[ ]:

10 entries per page Search:

	0
WHO Region	1
ISO3	0
WHO Country Name	0
City or Locality	0
Measurement Year	0
PM2.5 (µg/m3)	17143
PM10 (µg/m3)	11082
NO2 (µg/m3)	9991
PM25 temporal coverage (%)	24916
PM10 temporal coverage (%)	26810

Showing 1 to 10 of 15 entries « ‹ 1 2 › »

Kolumna Status składa się tylko z NaN - można ją bezpiecznie usunąć. Kolumny WHO Region i Reference mają bardzo małą ilość NaN (<5%), te wiersze można wypełnić odpowiednimi informacjami (WHO Region - European Region dla Liechtensteinu) lub usunąć (Reference - brak informacji o tym, jak je wypełnić). Inne kolumny mają wiele wartości NaN (około > 10 000), więc zostaną pozostawione bez zmian. Są to najważniejsze kolumny dotyczące parametrów PM2.5, PM10 oraz NO2, więc można domyślać się, że istnieją braki w odczytywanych wartościach na czujnikach. Niestety sporo danych dotyczących pokrycia (części roku, w której zbierane są dane) również brakuje.

In [ ]:
```
df.drop(columns=['Status'], inplace=True)
```

In [ ]:
```
print(df["WHO Region"].unique())
df.loc[df["WHO Region"].isna(), "WHO Region"] = 'European Region'
```

```
['Eastern Mediterranean Region' 'European Region' 'Region of the Americas'
 'Western Pacific Region' 'South East Asia Region' 'African Region' nan]
```

In [ ]:
```
df.loc[df["Reference"].isna(), :] # ISO3 = QAT
```

Out[ ]:

	WHO Region	ISO3	WHO Country Name	City or Locality
28209	Eastern Mediterranean Region	QAT	Qatar	Doha
28210	Eastern Mediterranean Region	QAT	Qatar	Doha
28211	Eastern Mediterranean Region	QAT	Qatar	Doha
28212	Eastern Mediterranean Region	QAT	Qatar	Doha
28213	Eastern Mediterranean Region	QAT	Qatar	Doha

In [ ]:
```
df.loc[df["ISO3"]=="QAT", :]
```

Out[ ]:

	WHO Region	ISO3	WHO Country Name	City or Locality
28208	Eastern Mediterranean Region	QAT	Qatar	Al-Bidda
28209	Eastern Mediterranean Region	QAT	Qatar	Doha
28210	Eastern Mediterranean Region	QAT	Qatar	Doha
28211	Eastern Mediterranean Region	QAT	Qatar	Doha
28212	Eastern Mediterranean Region	QAT	Qatar	Doha
28213	Eastern Mediterranean Region	QAT	Qatar	Doha
28214	Eastern Mediterranean Region	QAT	Qatar	Madinat Khaifa
28215	Eastern Mediterranean Region	QAT	Qatar	Muaither

In [ ]:
```
df.dropna(subset=["Reference"],inplace=True)
```

In [ ]:
```
# Double-check
print("Number of rows in the dataframe:", len(df))
df.isna().sum()
```

```
Number of rows in the dataframe: 32186
```

Out[ ]:

10 entries per page Search:

	0
WHO Region	0
ISO3	0
WHO Country Name	0
City or Locality	0
Measurement Year	0
PM2.5 (µg/m3)	17143
PM10 (µg/m3)	11082
NO2 (µg/m3)	9991
PM25 temporal coverage (%)	24916
PM10 temporal coverage (%)	26810

Showing 1 to 10 of 14 entries « ‹ 1 2 › »

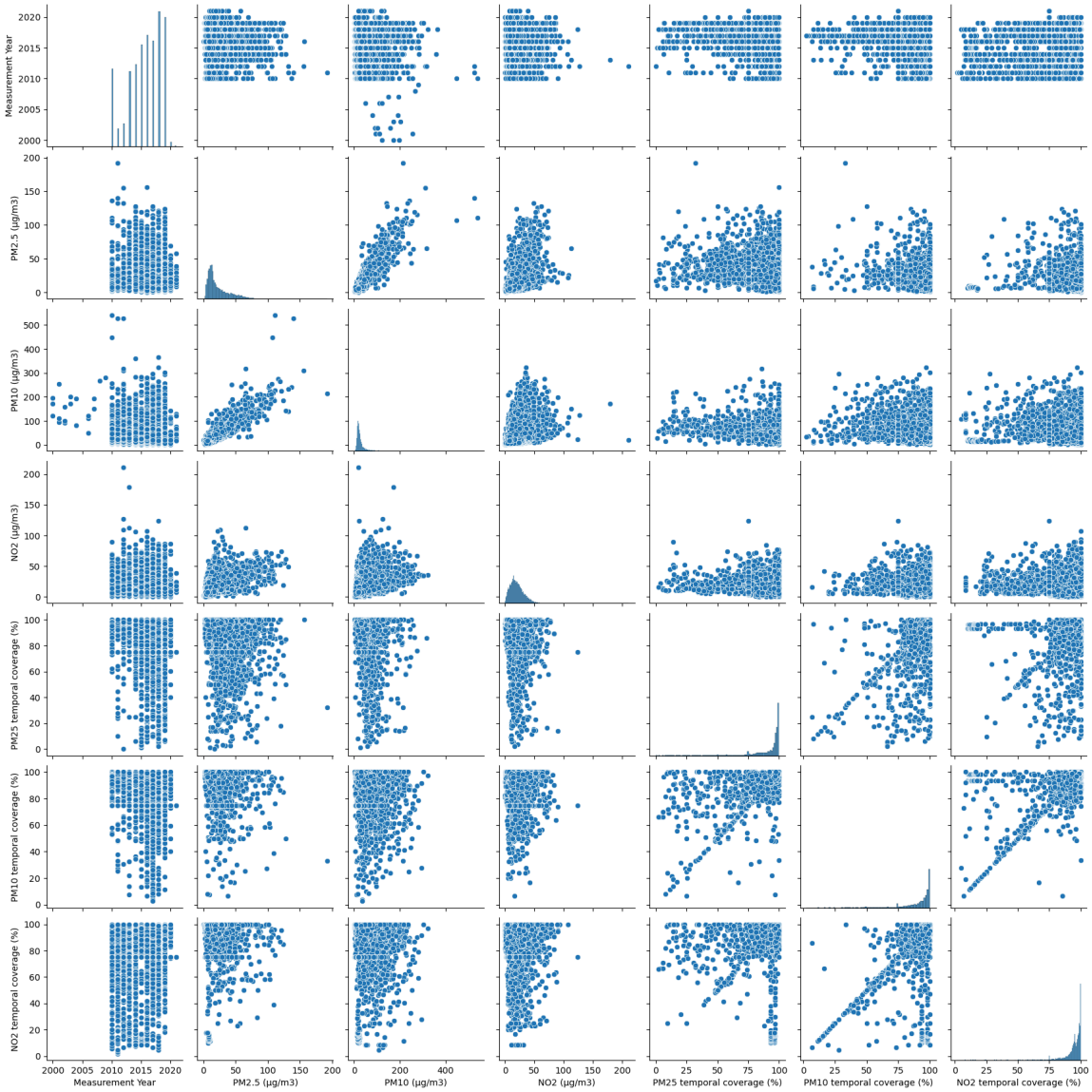
Dodatkowa selekcja cech - zdecydowano odrzucić kolumnę Version Of The Database, ponieważ nie wnosi ona nic merytorycznego do analizy.

In [ ]:
```
df.drop(columns=["Version of the database"], inplace=True)
```

Wizualizacje

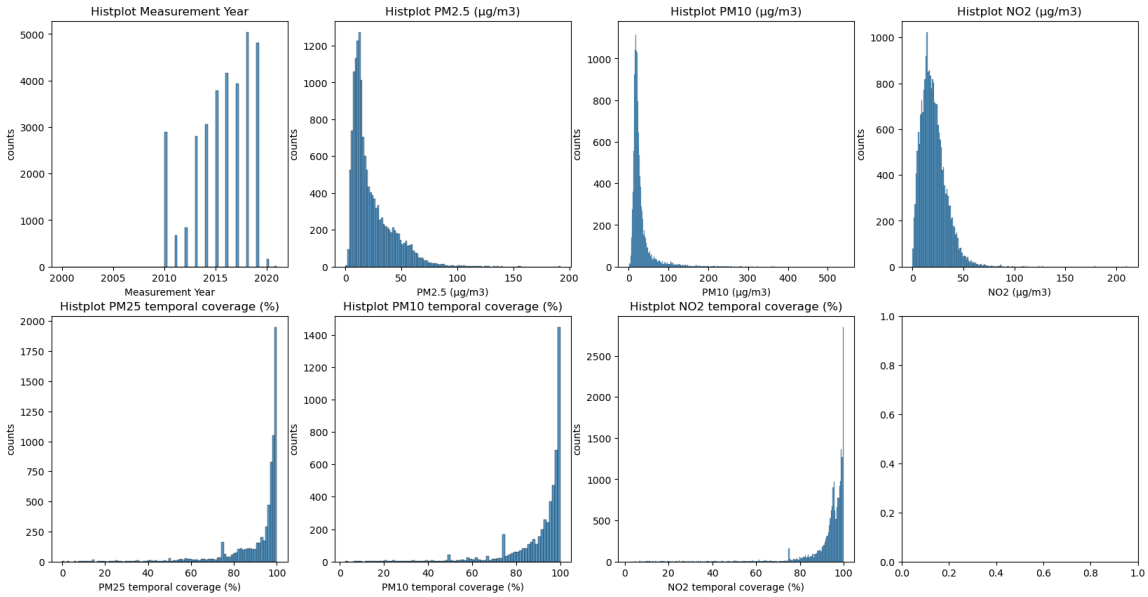
Poniżej przedstawiono podstawowe wizualizacje danych (pairplot, histplot, heatmap), aby lepiej zrozumieć ich charakter przed dalszą analizą. Przedstawiono również jak wyglądają zanieczyszczenia dla poszczególnych lat i jak wyglądał coverage.

In [ ]:
```
sns.pairplot(df)
plt.show()
```



Na pairplotcie widać relacje między zmiennymi oraz histogram każdej z nich. Zależność liniową widać pomiędzy PM2.5 oraz PM10. NO2 również posiada w jakimś stopniu zależność do PM2.5 jak i PM10, ale tego typu zależności zostaną zbadane dokładniej w dalszej kolejności. Histogramom przyjrzyjmy się dokładniej poniżej.

In [ ]:
```
numeric_columns = df.select_dtypes(include=['float64', 'int64'])

cols = numeric_columns.columns.to_list()
fig, ax = plt.subplots(2,4,figsize=(20,10))
ax = ax.flatten()

for i in range(len(cols)):
    sns.histplot(df[cols[i]], ax=ax[i])
    ax[i].set(title=f'Histplot {cols[i]}', ylabel='counts')

plt.show()
```

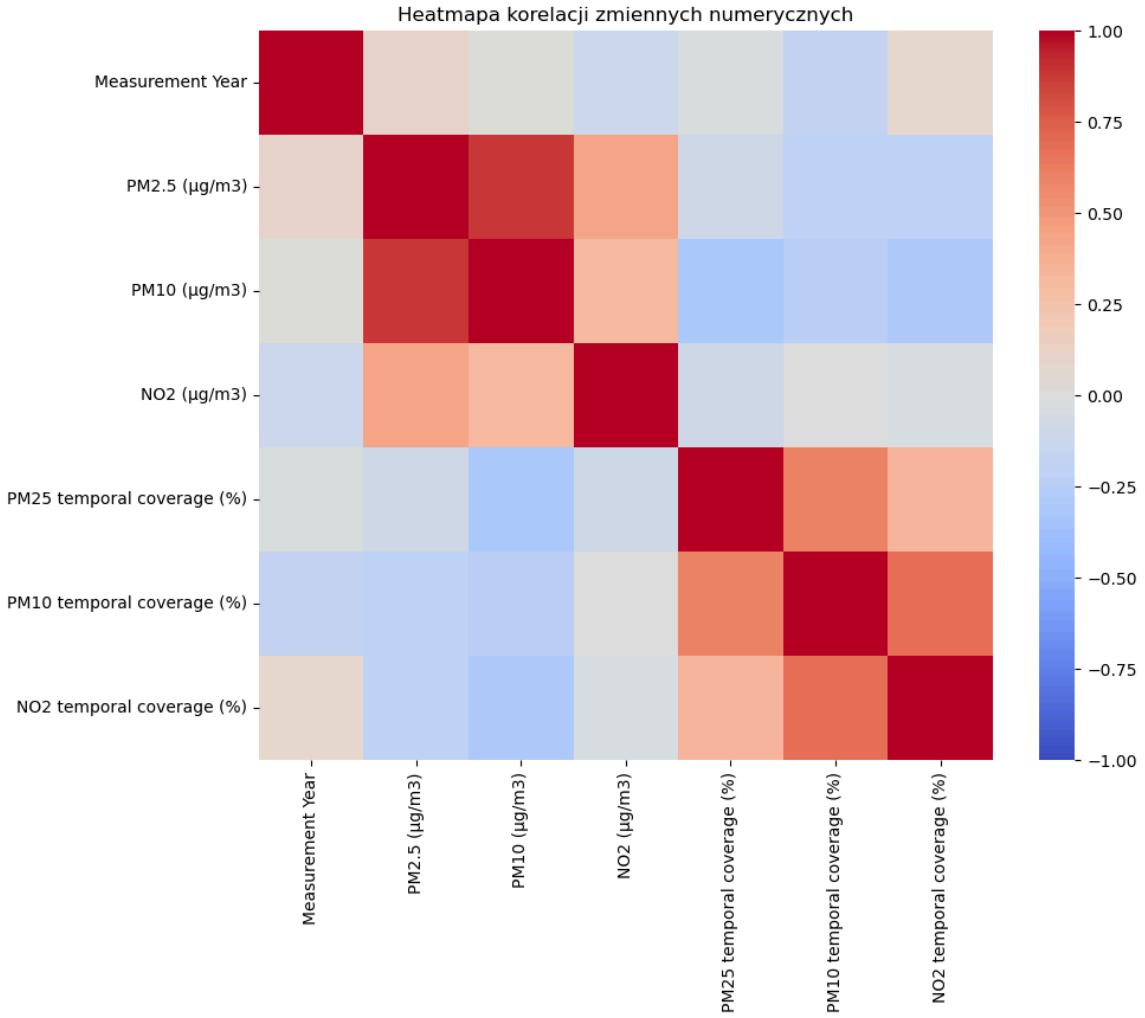


Pierwszym wnioskiem jest obserwacja na temat czasu kiedy były zbierane dane - sensowna ilość obserwacji zaczyna pojawiać się dopiero od 2010 roku (być może wtedy nastąpił skok technologiczny dotyczący czujników lub pojawiła się większa ich dostępność na świecie). Obserwacje sprzed 2010 zostaną usunięte. Warto zauważyć, że w latach 2011-12 zanotowano spadek obserwacji a następnie nagły skok aż do 2020 kiedy wybuchła pandemia (spadek obserwacji może wynikać właśnie z niej i powodanego przez nią kryzysu na świecie - były wtedy ważniejsze sprawy do monitorowania niż jakość powietrza). Jeśli chodzi o zmiennie dotyczące PM2.5, PM10 oraz NO2 mają one rozkłady prawoskośne, natomiast wykresy dotyczące pokrycia mają rozkłady lewoskośne. To dobrze, ponieważ im mniejsze zanieczyszczenie tym lepiej dla świata, a im większe pokrycie tym lepiej, bo dane są dokładniejsze i pozwalają na bardziej wiarygodne analizy.

In [ ]:
```
df2010 = df[df['Measurement Year']>=2010].reset_index(drop=True)
print(len(df), len(df2010)) # niewielka róznica około 20 rekordów
```

```
32186 32165
```

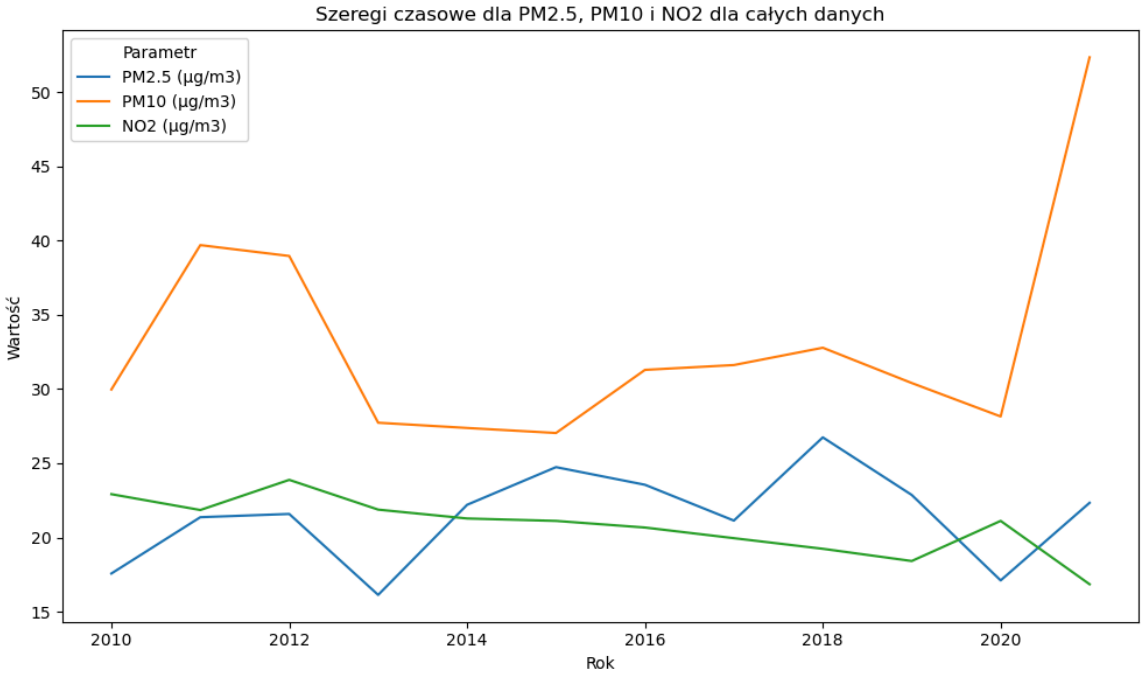
In [ ]:
```
numeric_columns = df2010.select_dtypes(include=['float64', 'int64'])

correlation = numeric_columns.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation, cmap='coolwarm', fmt=".2f", vmin=-1, vmax=1)
plt.title('Heatmapa korelacji zmiennych numerycznych')
plt.show()
```



Z heatmapy wypływa podobny wniosek jak z pairplotu - występuje związek między PM2.5 oraz PM10, a w mniejszym stopniu między NO2 oraz PM2.5 jak i NO2 oraz PM10. Podobnie dla tych samych par wygląda sytuacja dla zmiennych dotyczących pokrycia.

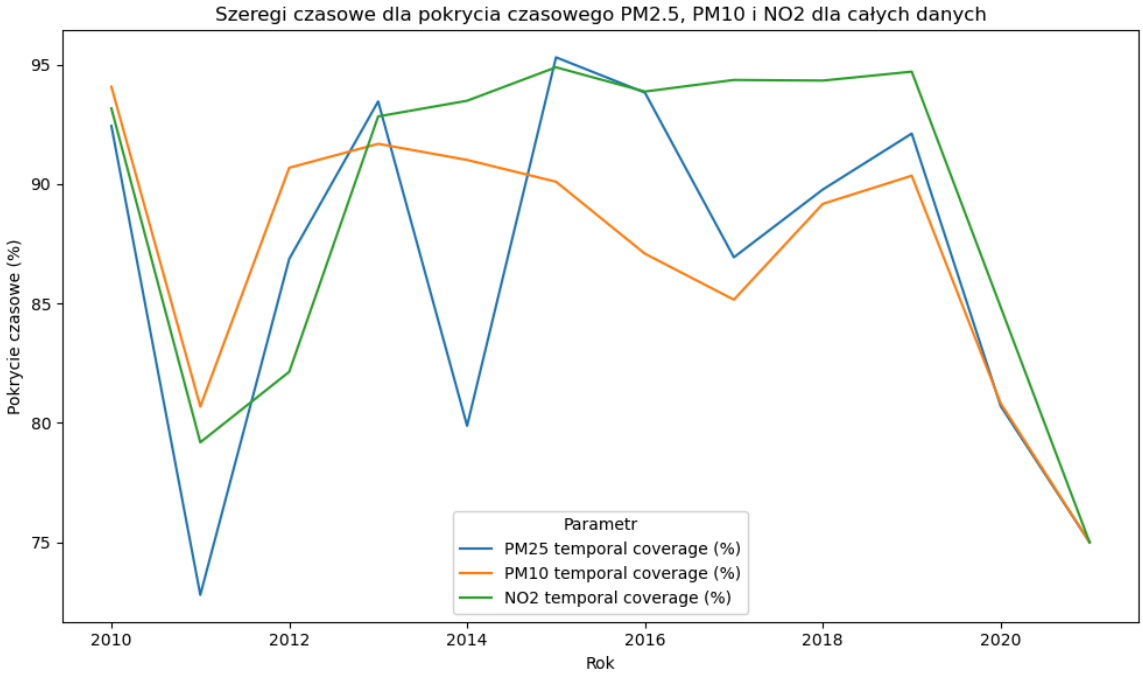
Poniżej przedstawiono jak wygląda zanieczyszczenie na przestrzeni lat.

In [ ]:
```
columns = ['PM2.5 (µg/m3)', 'PM10 (µg/m3)', 'NO2 (µg/m3)']
plt.figure(figsize=(10, 6))

for column in columns:
    sns.lineplot(data=df2010, x='Measurement Year', y=column, label=column, erro

plt.title('Szeregi czasowe dla PM2.5, PM10 i NO2 dla całych danych')
plt.xlabel('Rok')
plt.ylabel('Wartość')
plt.legend(title='Parametr')
plt.tight_layout()
plt.show()
```



Największe zanieczyszczenie powoduje PM10, natomiast PM2.5 oraz NO2 wypadają dosyć podobnie. Można zauważyć, że nie ma wielu wspólnych tendencji dla wszystkich 3 parametrów na raz - ewentualnie stosunkowa stabilność w latach 2014-2017.

In [ ]:
```
columns_cov = ['PM25 temporal coverage (%)', 'PM10 temporal coverage (%)', 'NO2

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

for column in columns_cov:
    sns.lineplot(data=df2010, x='Measurement Year', y=column, label=column, erro

plt.title('Szeregi czasowe dla pokrycia czasowego PM2.5, PM10 i NO2 dla całych d
plt.xlabel('Rok')
plt.ylabel('Pokrycie czasowe (%)')
plt.legend(title='Parametr')
plt.tight_layout()
plt.show()
```



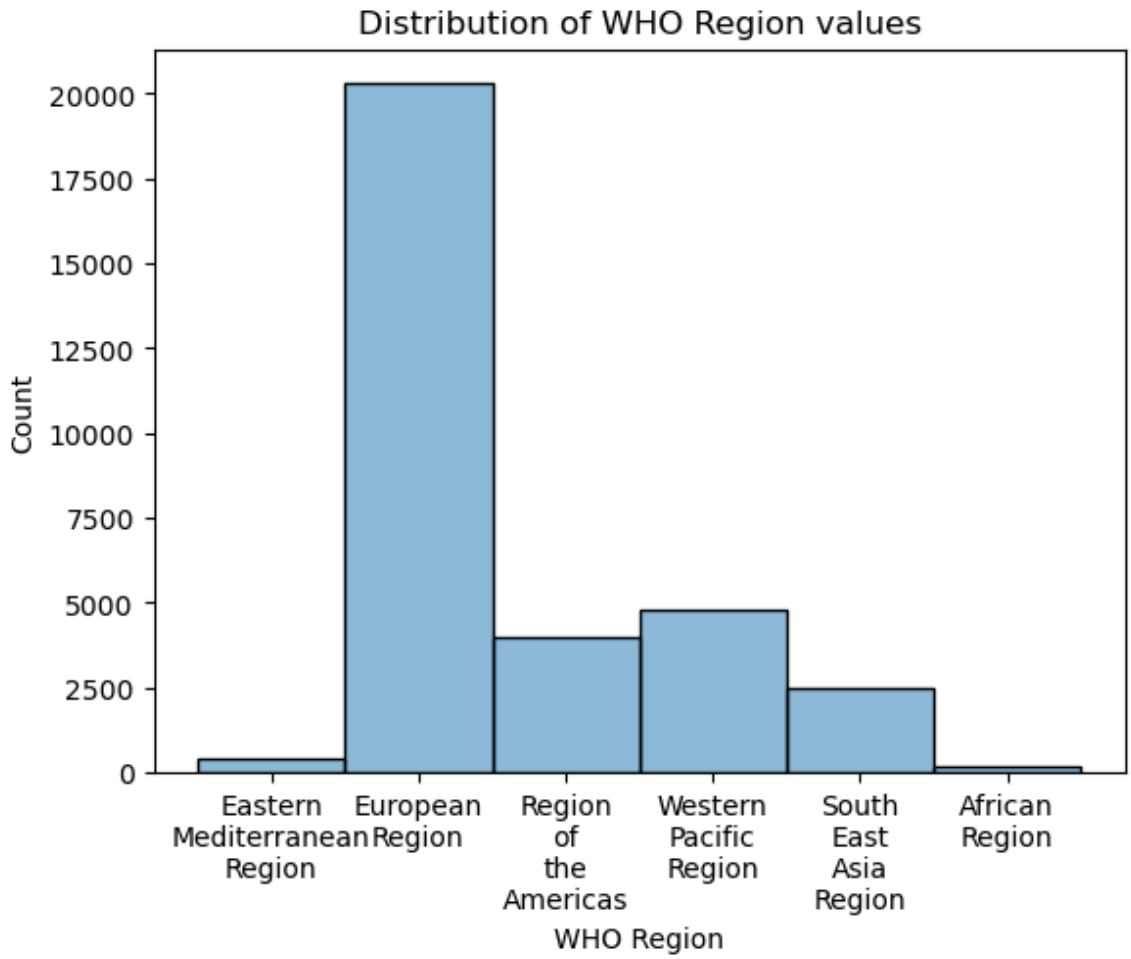
Dla NO2 pokrycie wygląda najstabilniej - dane po podejrzanych latach 2011-12 utrzymywał się aż do pandemii na stałym wysokim poziomie, dane były zbierane przez więcej niż 90% czasu w roku. Najmniej stabilnie przedstawia się PM2.5, może to wynikać z jakości czujników lub problemów technicznych.

Analiza na poziomie regionów

In [ ]:
```
len(df2010['WHO Region'].unique()) # 6 regionów - wizualizacje będą czytelne
```

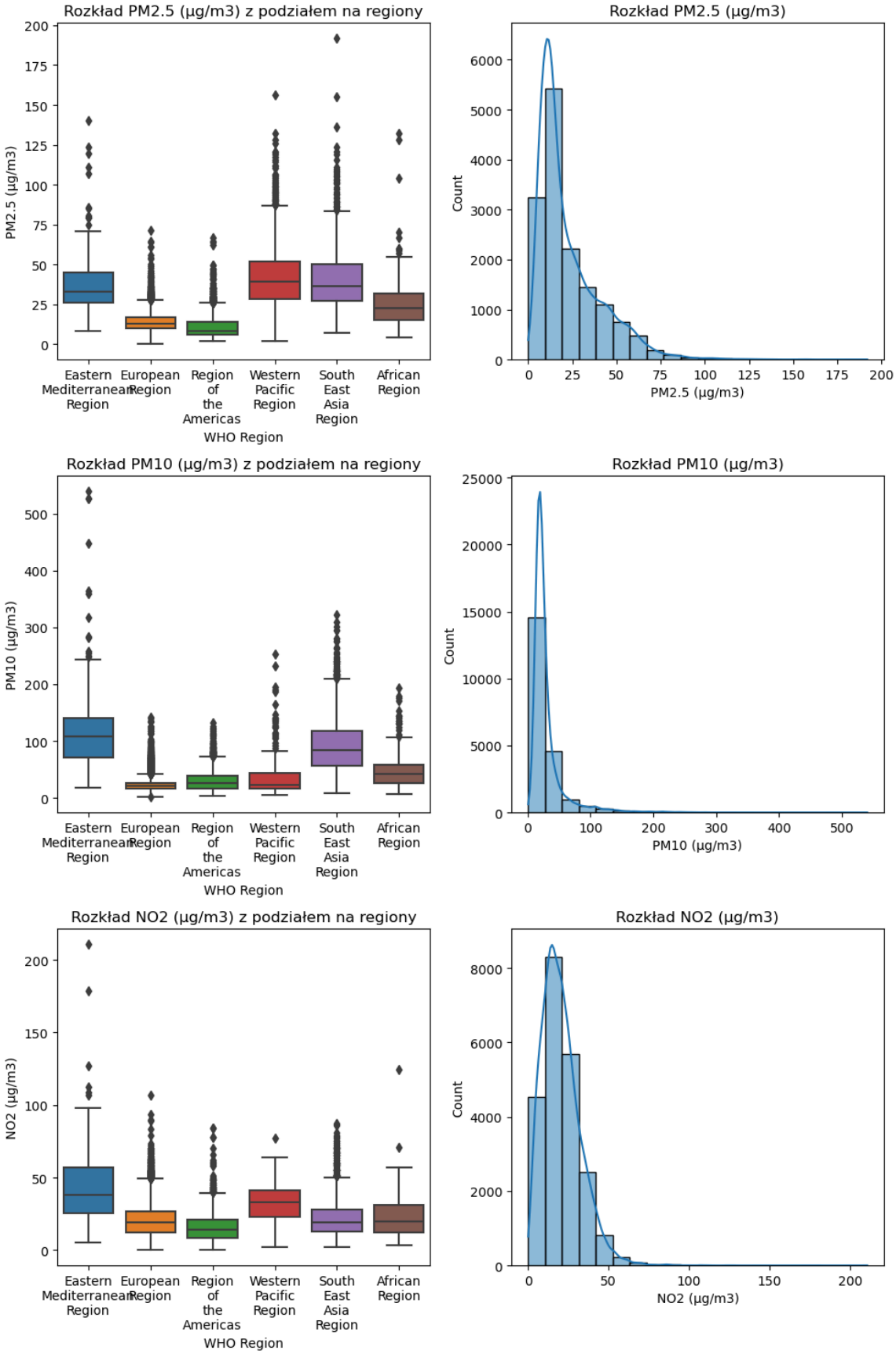
Out[ ]: 6

In [ ]:
```
x_ticks = [x.replace(" ", "\n") for x in df2010["WHO Region"].unique()]
sns.histplot(data=df2010, x="WHO Region",bins=20, alpha=0.5)
plt.xticks(ticks=np.arange(len(x_ticks)),labels=x_ticks)
plt.title(f'Distribution of WHO Region values')
plt.show()
```



W regionie europejskim zebrano zdecydowanie najwięcej danych, należy pamiętać o tym podczas dalszych analiz - gdy patrzymy na dane całościowo może wystąpić "europejski bias". Poniżej przedstawionio wykresy pudełkowe zmiennych PM2.5, PM10, NO2 z podziałem na regiony oraz rozkładem danej zmiennej dla odniesienia.

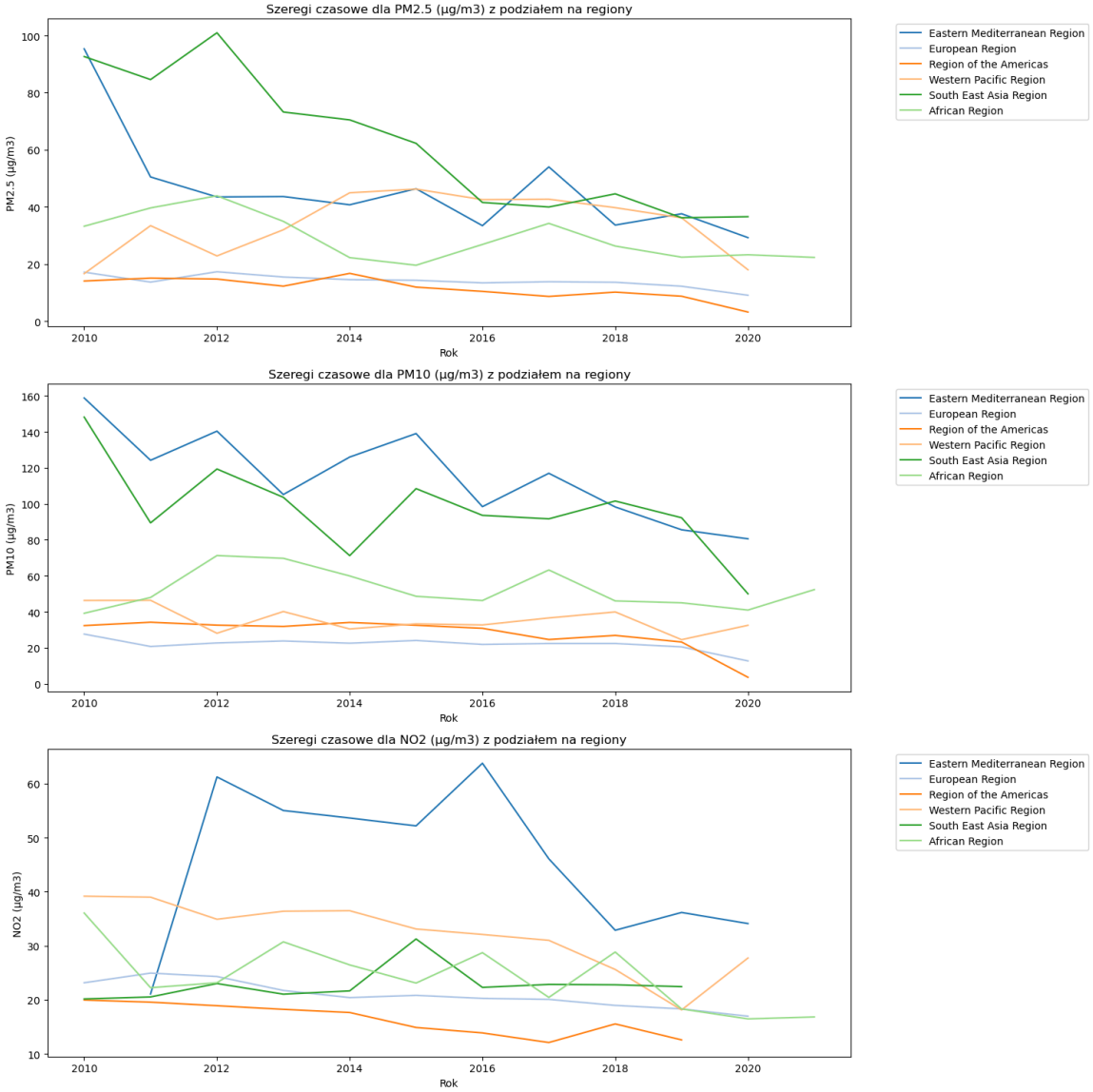
In [ ]:
```
fig, ax = plt.subplots(3,2, figsize=(10,15))

for i, column in enumerate(columns):
    sns.boxplot(data=df2010, y=column, x='WHO Region', ax=ax[i,0])
    ax[i,0].set_xticks(ticks=np.arange(len(x_ticks)), labels=x_ticks)
    ax[i,0].set_title(f'Rozkład {column} z podziałem na regiony')

    sns.histplot(data=df2010, x=column, kde=True, bins=20, ax=ax[i,1], alpha=0.5
    ax[i,1].set_title(f'Rozkład {column}')

plt.tight_layout()
plt.show()
```



Z wizualizacji można zauważyć, że większość obserwacji skupia się wokół mniejszych wartości, ale widać sporo outlierów w górnej części wykresów pudełkowych.

In [ ]:
```
fig, ax = plt.subplots(3, 1, figsize=(15, 15))
for i, column in enumerate(columns):
    sns.lineplot(data=df2010, x='Measurement Year', y=column, hue='WHO Region',
    ax[i].set_title(f'Szeregi czasowe dla {column} z podziałem na regiony')
```

```
    ax[i].set_ylabel(column)
    ax[i].set_xlabel('Rok')
    ax[i].legend(bbox_to_anchor=(1.05, 1), loc='upper left')


plt.tight_layout()
plt.show()
```



Napawającym optymizmem wnioskiem jest to, że z czasem zanieczyszczenie ma tendencje do spadku (wszystkie 3 parametry).

Analiza na poziomie krajów

In [ ]:
```
len(df2010['WHO Country Name'].unique())
```

Out[ ]:
```
118
```

Krajów jest za dużo, aby czytelnie zwizualizować ich cechy. Postanowiono wybrać po 2 kraje dla każdego regionu z największą ilością rekordów, aby były reprezentatywne.

In [ ]:
```
def top_countries_with_counts(group):
    counts = group['WHO Country Name'].value_counts().nlargest(2)
```

```
    return list(zip(counts.index, counts.values))

top_countries_by_region = df2010.groupby('WHO Region').apply(top_countries_with_
top_countries = [country[0] for sublist in top_countries_by_region['Top Countrie
top_countries = list(set(top_countries))

df_countries = df2010[df2010['WHO Country Name'].isin(top_countries)]
print(len(df_countries))
top_countries_by_region
```

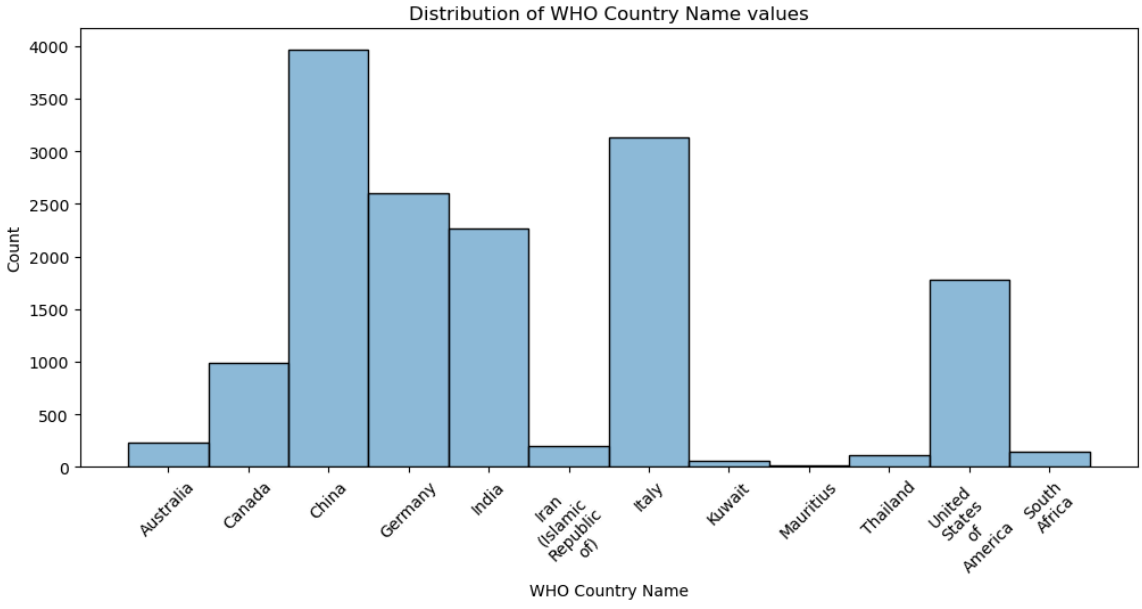
15492

Out[ ]:

WHO Region	Top Countries
African Region	[(South Africa, 145), (Mauritius, 11)]
Eastern Mediterranean Region	[(Iran (Islamic Republic of), 204), (Kuwait, 62)]
European Region	[(Italy, 3129), (Germany, 2601)]
Region of the Americas	[(United States of America, 1776), (Canada, 986)]
South East Asia Region	[(India, 2265), (Thailand, 110)]
Western Pacific Region	[(China, 3967), (Australia, 236)]

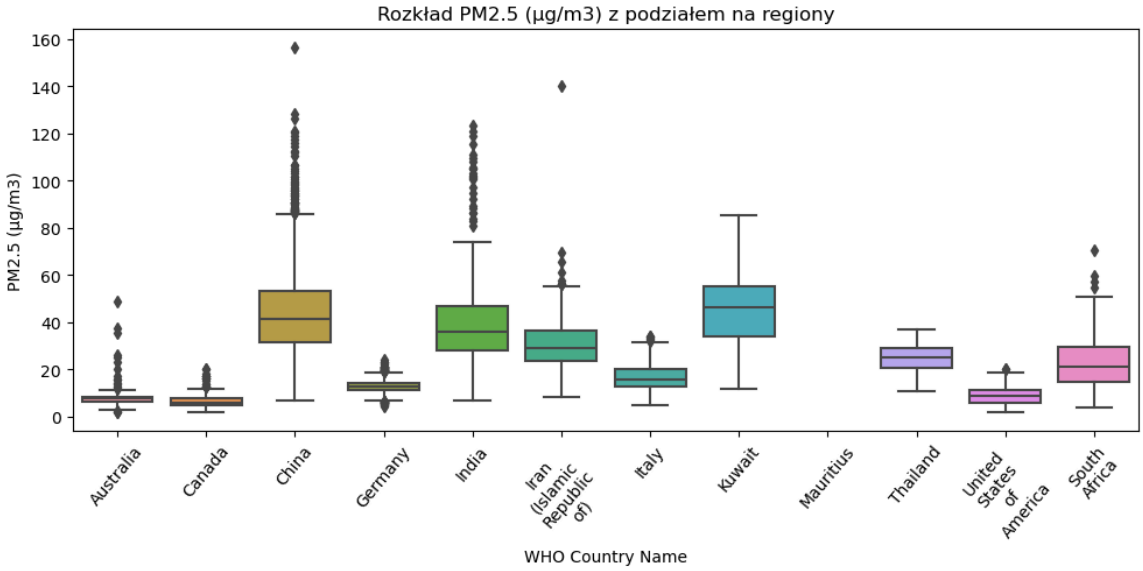
Warto zauważyć, że dla Afryki mamy bardzo mało obserwacji w porównaniu do innych regionów - aby o tym pamiętać pozostawiono wybrane kraje do dalszej analizy. Poniżej rozkład wartości dla wybranych państw.

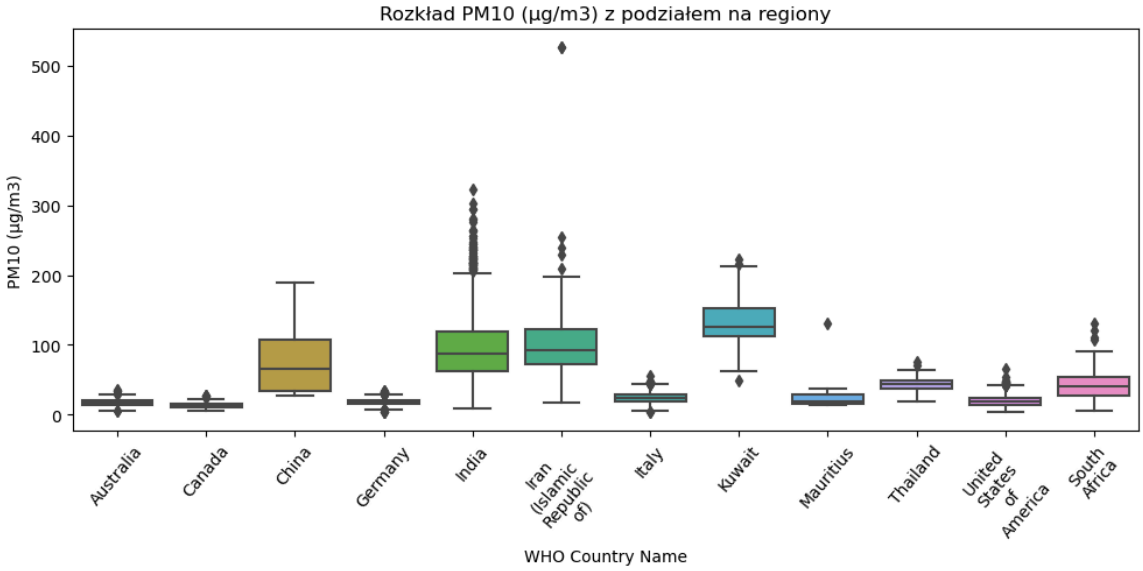
In [ ]:
```
x_ticks = [x.replace(" ", "\n") for x in df_countries["WHO Country Name"].unique
fig, ax = plt.subplots(1,1,figsize=(12,5))
sns.histplot(ax=ax,data=df_countries, x="WHO Country Name",bins=20, alpha=0.5)
plt.xticks(ticks=np.arange(len(x_ticks)),labels=x_ticks, rotation=45)
plt.title(f'Distribution of WHO Country Name values')
plt.show()
```

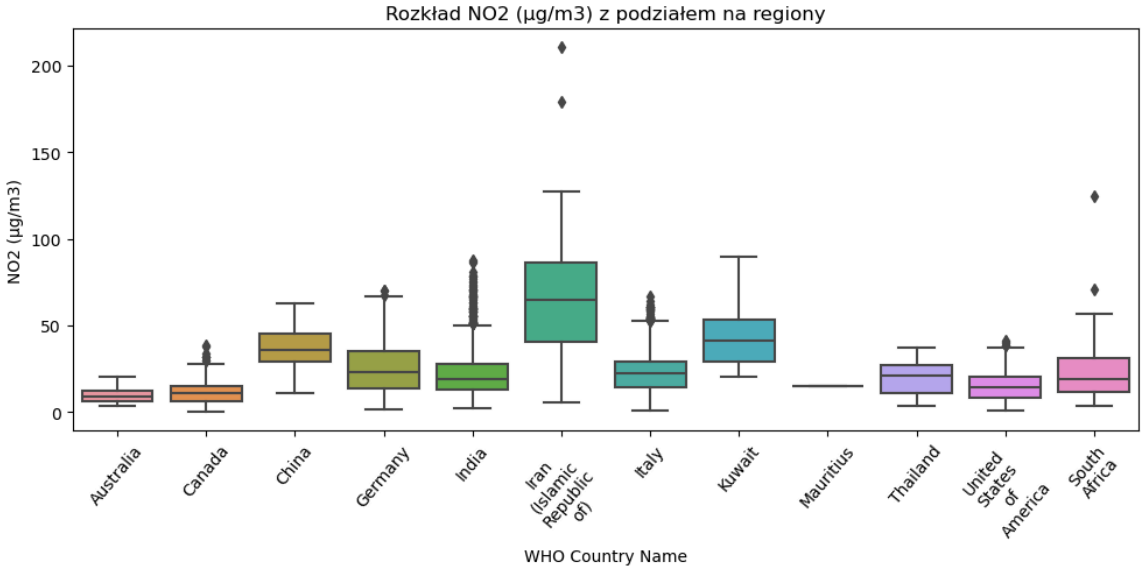


In [ ]:
```
columns = ['PM2.5 (µg/m3)', 'PM10 (µg/m3)', 'NO2 (µg/m3)']

fig, ax = plt.subplots(3,1, figsize=(10,15))
```

```
for i, column in enumerate(columns):
    sns.boxplot(data=df_countries, y=column, x='WHO Country Name', ax=ax[i])
    ax[i].set_xticks(ticks=np.arange(len(x_ticks)), labels=x_ticks)
    ax[i].set_title(f'Rozkład {column} z podziałem na regiony')
    ax[i].set_xticklabels(x_ticks, rotation = 50)

plt.tight_layout()
plt.show()
```



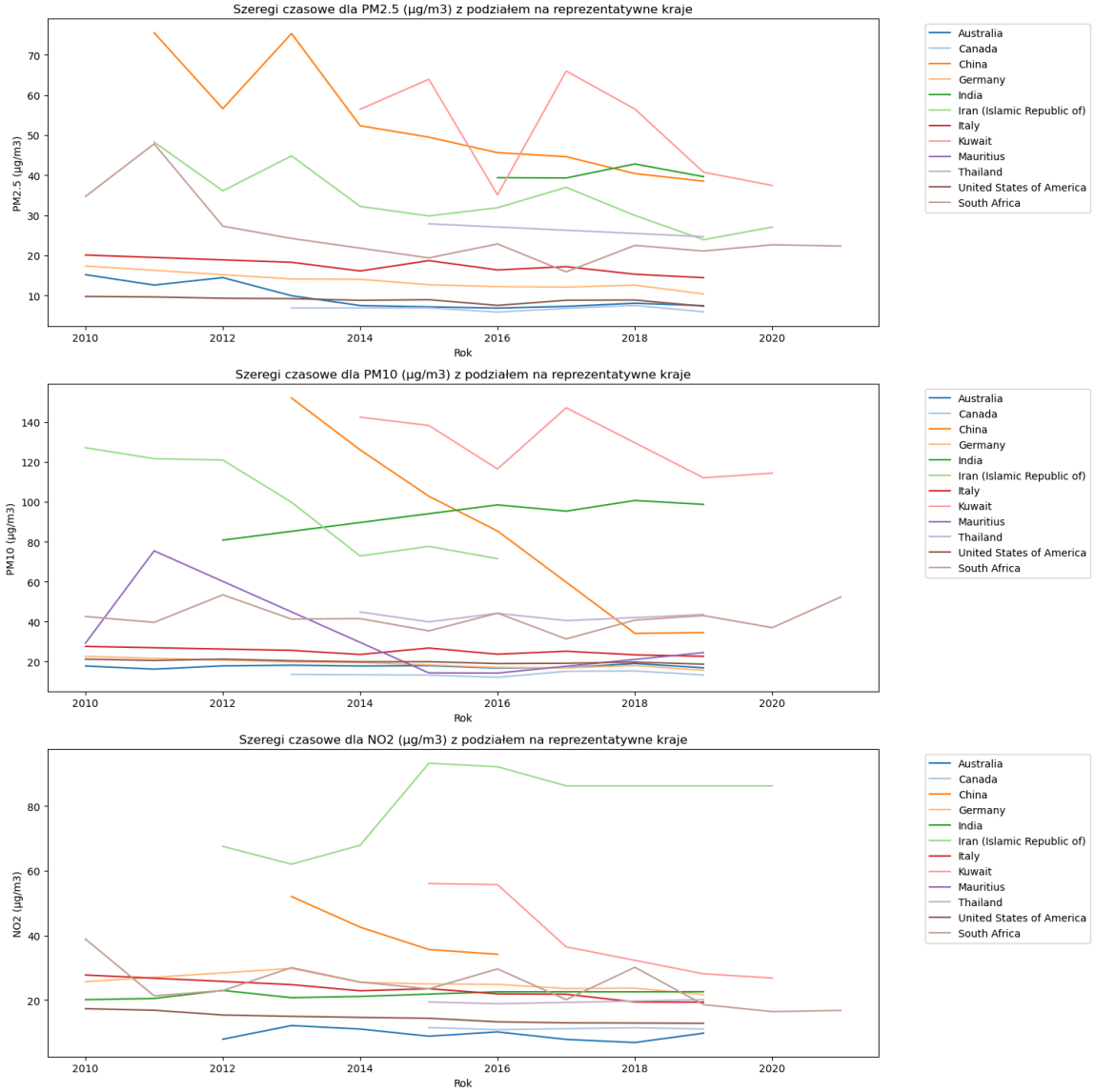




Podobnie jak przy analizie regionów widać, że większość wartości skupia się w dolnej części boxplotów. Natomiast na tym poziomie granularności widać rozbieżności w ilości pomiarów dla poszczególnych regionów.

In [ ]:
```
fig, ax = plt.subplots(3, 1, figsize=(15, 15))
for i, column in enumerate(columns):
    sns.lineplot(data=df_countries, x='Measurement Year', y=column, hue='WHO Cou
```

```
    ax[i].set_title(f'Szeregi czasowe dla {column} z podziałem na reprezentatywn
    ax[i].set_ylabel(column)
    ax[i].set_xlabel('Rok')
    ax[i].legend(bbox_to_anchor=(1.05, 1), loc='upper left')

plt.tight_layout()
plt.show()
```



Analiza na poziomie miast

In [ ]:
```
len(df2010['City or Locality'].unique())
```

Out[ ]: 6871

Analogicznie jak dla miast zostaną wybrane tylko reprezentatywne państwa w oparciu o regiony.

In [ ]:
```
def top_cities_with_counts(group):
    counts = group.groupby(['City or Locality', 'WHO Country Name']).size().nlar
    return list(zip(counts.index.get_level_values(0), counts.index.get_level_val

top_cities_by_region = df2010.groupby('WHO Region').apply(top_cities_with_counts
top_cities = [city[0] for sublist in top_cities_by_region['Top Cities'] for city
```

```
top_cities = list(set(top_cities))

df_cities = df2010[df2010['City or Locality'].isin(top_cities)]
print(len(df_cities))
top_cities_by_region
```

146

Out[ ]:

WHO Region	Top Cities
African Region	[(Ethekwini, South Africa, 13), (Gert Sibande, South Africa, 1
Eastern Mediterranean Region	[(Tehran, Iran (Islamic Republic of), 11), (Abu Dhabi, United Arab Emirates,
European Region	[(Avully, Switzerland, 17), (Basel, Switzerland, 1
Region of the Americas	[(Lima, Peru, 12), (Albuquerque (Nm), United States of America, 1
South East Asia Region	[(Bhubneshwar, India, 10), (Chittagong, Bangladesh, 1
Western Pacific Region	[(Seoul, Republic of Korea, 13), (Busan, Republic of Korea, 1

Jako, że dane są zbierane co rok, a widać wartości np. 17 to widać, że coś jest nie w porządku, postanowiono przyjrzeć się danym.

In [ ]:
```
df_cities
```

Out[ ]:

10 entries per page

	WHO Region	ISO3	WHO Country Name	City or Locality
20	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
21	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
22	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
23	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
24	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
25	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
26	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
27	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
28	Eastern Mediterranean Region	ARE	United Arab Emirates	Abu Dhabi
1742	South East Asia Region	BGD	Bangladesh	Chittagong

Showing 1 to 10 of 146 entries

Dane zawierają duplikaty z różnicą jedynie w kolumnie Number and type of monitoring stations. Jako, że kluczowe parametry są te same postanowiono pozbyć się tej kolumny a

następnie usunąć duplikaty.

In [ ]:
```
df_unique = df2010.drop(columns=['Number and type of monitoring stations'])
df_unique = df_unique.drop_duplicates()
print(len(df2010),len(df_unique))
```

```
32165 32069
```

Widać, że zniknęło około 100 rekordów - nie ma to większego wpływu na poprzednie analizy, więc nie wprowadzano żadnych korekt.

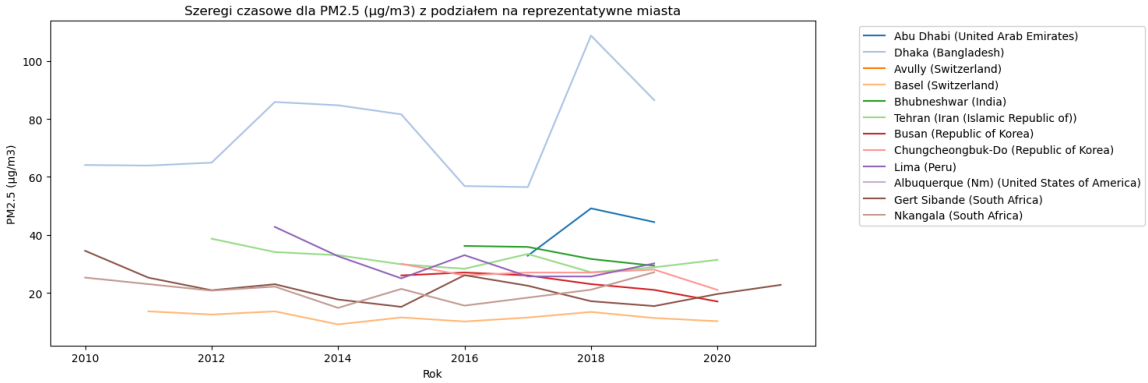
In [ ]:
```
top_cities_by_region = df_unique.groupby('WHO Region').apply(top_cities_with_cou
top_cities = [city[0] for sublist in top_cities_by_region['Top Cities'] for city
top_cities = list(set(top_cities))

df_cities = df_unique[df_unique['City or Locality'].isin(top_cities)]
print(len(df_cities))
top_cities_by_region
```

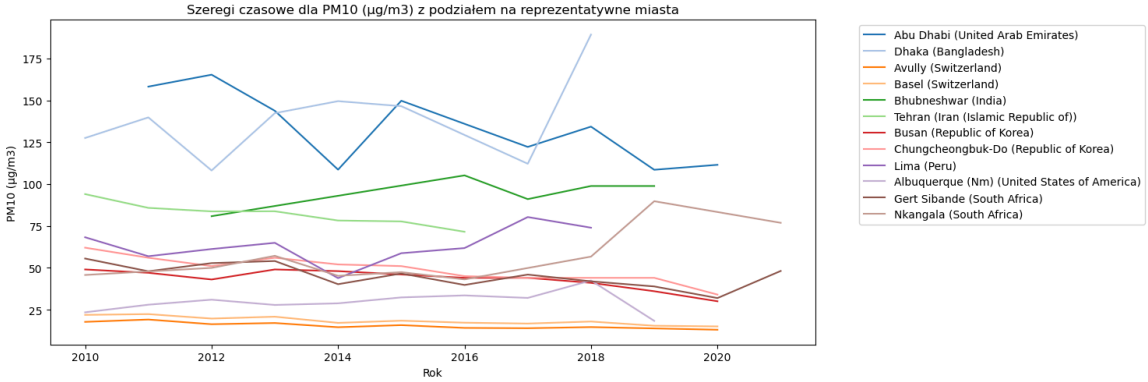
```
130
```

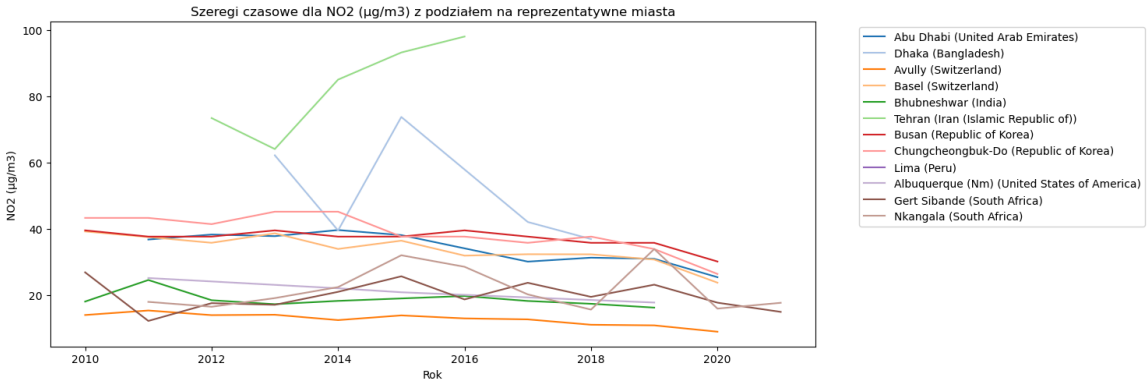
Out[ ]:

WHO Region	Top Cities
African Region	[(Gert Sibande, South Africa, 12), (Nkangala, South Africa,
Eastern Mediterranean Region	[(Tehran, Iran (Islamic Republic of), 11), (Abu Dhabi, United Arab Emirates
European Region	[(Avully, Switzerland, 11), (Basel, Switzerland,
Region of the Americas	[(Lima, Peru, 12), (Albuquerque (Nm), United States of America,
South East Asia Region	[(Bhubneshwar, India, 10), (Dhaka, Bangladesh,
Western Pacific Region	[(Busan, Republic of Korea, 11), (Chungcheongbuk-Do, Republic of Korea,

In [ ]:
```
fig, ax = plt.subplots(3, 1, figsize=(15, 15))

for i, column in enumerate(columns):
    sns.lineplot(data=df_cities, x='Measurement Year', y=column, hue='City or Lo
    handles, labels = ax[i].get_legend_handles_labels()
    new_labels = [f"{label} ({df_cities[df_cities['City or Locality'] == label][
    ax[i].legend(handles, new_labels,  bbox_to_anchor=(1.05, 1), loc='upper left
    ax[i].set_title(f'Szeregi czasowe dla {column} z podziałem na reprezentatywn
    ax[i].set_ylabel(column)
    ax[i].set_xlabel('Rok')

plt.tight_layout()
plt.show()
```







Mapa

In [ ]:
```
mean_pollution = df_unique.groupby(['WHO Country Name','ISO3'])[['PM2.5 (µg/m3)'

for col in columns:
    fig = px.choropleth(
        mean_pollution,
        locations='ISO3',
        locationmode='ISO-3',
        color=col,
        hover_name='WHO Country Name',
        projection='natural earth',
        title=f'Średnie wartości {col} dla całości danych (lata 2010-2021)'
    )

    fig.show()
```

Na pytania wylistowane na UPELu częściowo odpowiedzinao w komentarzach i analizach powyżej. Poniżej jednak znajduje się podsumowanie i wypunktowane najważniejsze wnioski:

- dane obejmują (po oczyszczeniu) okres czasu 2010-2021
- ilość czujników z czasem zwiększa się co jest dość logiczne, z czasem kraje rozwijają się - mimo wszystko kraje biedne są w dużo gorszej sytuacji, np. Afryka jest opisana bardzo małą ilością danych
- podczas interpretacji należy pamiętać o coverage, jeśli jest niski mamy do czynienia z obciążeniem i możliwymi błędnymi wnioskami
- w Europie mamy najwięcej czujników, a tym samym zebranych pomiarów
- w krajach wysoko rozwiniętych zanieczyszczenia mają z czasem tendencje spadkowe - jakość powietrza poprawia się
- najgorzej sytuacja wygląda w krajach takich jak Iran, Chiny, Mongolia, Irak, Indie - jakość powietrza jest tam wyraźnie gorsza

Związki przyczynowo-skutkowe:

- parametry dotyczące zanieczyszczenia powietrza są ze sobą silnie dodatnio skorelowane (tak należało się spodziewać)
- analogicznie dla kolumn dotyczących pokrycia owych parametrów
- mała ilość danych dla poszczególnych rejonów może powodować gorszą jakość analiz, obciążenie, błędne wnioski
- jeżeli kraj jest bogaty/rozwijający się to jakość powietrza polepsza się (ogólne stiwerdzenie, zawsze znajdą się wyjątki od reguły) oraz są tam dokładnie zbierane dane, jest dużo czujników
- kraje biedniejsze mają gorszą jakość powietrza oraz mniejszą ilość czujników

Kontekst ML:

- dane mogłyby być materiałem treningowym dla modelu przewidującego jakość powietrza, ale jakość takiego działania mogłaby być wątpliwa
- w danych znajduje się dużo wartości brakujących, a więc model nie byłby się w stanie dobrze nauczyć
- agregacja na poziomie roku również nie daje możliwości dokładnej eksploracji danych
- dane pochodzą z ogranicznego okresu, a rejony z których są zbierane różnią się między sobą ilością i jakością - utrudnienie dla treningu modelu