

▼ Projekt - EDA z uwzględnieniem czynnika przestrzennego

Autor: Grzegorz Cichy

Celem projektu jest wykonanie EDA dla danych dotyczących zanieczyszczenia powietrza, pochodzących z bazy WHO. Zbiór danych zawiera pomiary z różnych lokalizacji na świecie, podzielone na kraje oraz miasta.

Szczegółowa instrukcja wykonania:

1. Wczytaj dane zanieczyszczenia powietrza z bazy WHO do odpowiedniej struktury danych, np. DataFrame w Pythonie. Spróbuj również wykorzystać bibliotekę ITTables 2.0 i napisz czy wnosi ona wartość użytkową w przypadku danych używanych w tym projekcie.
2. Przeprowadź eksploracyjną analizę danych (EDA) dla różnych poziomów generalizacji - regionu, kraju i miasta. Wykorzystaj różne techniki analizy danych poznane na wykładzie.
3. Przeprowadź analizę na różnych poziomach generalizacji i skomentuj, jakie są różnice w interpretacji wyników w zależności od tego, czy analizujesz dane na poziomie regionu, kraju czy miasta
4. Przedstaw analizę również za pomocą mapy, aby zobrazować geograficzne rozkłady zanieczyszczenia powietrza w różnych regionach.
5. Na podstawie przeprowadzonej analizy, znajdź potencjalne związki przyczynowo-skutkowe między różnymi zmiennymi. Skomentuj, jakie wnioski można wyciągnąć na temat wpływu zanieczyszczenia powietrza na dane regiony, kraje i miasta.
6. Zastanów się, jakie potencjalne zastosowania ma ten rodzaj danych w uczeniu maszynowym, na przykład jako dane wejściowe do modeli przewidujących jakość powietrza w przyszłości.
7. Przedstaw swoje wnioski w formie podsumowania, uwzględniając interpretacje uzyskanych wyników oraz wnioski dotyczące zastosowań w uczeniu maszynowym.

▼ Import używanych bibliotek:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib as mpl
import numpy as np
#!pip install itables
import itables
from itables import show
from itables import init_notebook_mode
import copy
import geopandas as gpd
from matplotlib.widgets import Slider
```

▼ Wczytanie danych:

Dane pochodzą ze strony: <https://www.who.int/gho/data/themes/air-pollution/who-air-quality-database>. Zawierają informacje o zanieczyszczeniu powietrza w lokacjach na całym świecie jako uśredniona średnia roczna. Badane zanieczyszczenia to stężenie pyłów zawieszonych PM10 oraz PM25, a także stężenie dwutlenku azotu NO2.

Szczegółowy opis kolumn:

- Measurement Year: Rok pomiaru.
- PM2.5 ($\mu\text{g}/\text{m}^3$): Stężenie pyłów PM2.5 w mikrogramach na metr sześcienny.
- PM10 ($\mu\text{g}/\text{m}^3$): Stężenie pyłów PM10 w mikrogramach na metr sześcienny.
- NO2 ($\mu\text{g}/\text{m}^3$): Stężenie dwutlenku azotu (NO2) w mikrogramach na metr sześcienny.
- PM25 temporal coverage (%): Pokrycie czasowe danych dotyczących pyłów PM2.5 (%).
- PM10 temporal coverage (%): Pokrycie czasowe danych dotyczących pyłów PM10 (%).
- NO2 temporal coverage (%): Pokrycie czasowe danych dotyczących dwutlenku azotu (%).
- Reference: Odwołanie do źródła danych.
- Number and type of monitoring stations: Liczba i typ stacji monitorujących.
- Version of the database: Wersja bazy danych.
- Status: Status danych.

```
init_notebook_mode(connected=True)

/usr/local/lib/python3.10/dist-packages/itables/javascript.py:108: UserWarning:
Did you know? init_notebook_mode(all_interactive=False, connected=True) does nothing. Feel free to remove this line, or pass warn_i

df = pd.read_excel('sample_data/who_aap_2021_v9_11august2022.xlsx', sheet_name='AAP_2022_city_v9')

new_cols = {'WHO Region' : 'Region',
            'WHO Country Name' : 'Country',
            'City or Locality': 'City',
            'Measurement Year' : 'Year'}
df.rename(columns = new_cols,inplace = True)
df.head(10)
show(df, column_filters='header', search={"regex": True, "caseInsensitive": True})
```

10 ▾ entries per page		Search: <input type="text"/>
<input type="text"/> Search Region	<input type="text"/> Search ISO3	<input type="text"/> Search Country
Eastern Mediterranean Region	AFG	Afghanistan
European Region	ALB	Albania

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

Użycie biblioteki `Itables` umożliwia interaktywne wyświetlanie danych w jupyter notebooks, filtrację, dostosowanie ilości elementów na stronie, czy wyszukiwanie wg regexu. Zdecydowanie biblioteka `Itables` wnosi wartość użytkową i będę z niej korzystać w przyszłości.

Podstawowe informacje o danych

Typy kolumn:

```
df.dtypes
```

Region	object
ISO3	object
Country	object
City	object
Year	int64
PM2.5 (µg/m³)	float64
PM10 (µg/m³)	float64
NO2 (µg/m³)	float64
PM25 temporal coverage (%)	float64
PM10 temporal coverage (%)	float64
NO2 temporal coverage (%)	float64
Reference	object
Number and type of monitoring stations	object
Version of the database	int64
Status	float64
dtype: object	

Podstawowe statystyki opisowe:

```
df.describe()
```

	Year	PM2.5 ($\mu\text{g}/\text{m}^3$)	PM10 ($\mu\text{g}/\text{m}^3$)	NO2 ($\mu\text{g}/\text{m}^3$)	PM25 temporal coverage (%)	PM10 temporal coverage (%)
count	32191.000000	15048.000000	21109.000000	22200.000000	7275.000000	5381.000000
mean	2015.579354	22.920320	30.533252	20.619336	90.794096	90.583500
std	2.752654	17.925906	29.312756	12.133388	14.872681	13.816311
min	2000.000000	0.010000	1.040000	0.000000	0.000000	2.568493
25%	2014.000000	10.350000	16.980000	12.000000	88.595890	87.945205
50%	2016.000000	16.000000	22.000000	18.800000	97.000000	96.039000

▼ Wartości brakujące:

```
df_null = pd.concat([np.round(df.isnull().sum()/len(df)*100,3), df.isnull().sum()], axis=1)
df_null.columns = ['null percent', 'null count']
df_null
```

	null percent	null count	grid icon
Region	0.003	1	grid icon
ISO3	0.000	0	grid icon
Country	0.000	0	grid icon
City	0.000	0	grid icon
Year	0.000	0	grid icon
PM2.5 ($\mu\text{g}/\text{m}^3$)	53.254	17143	grid icon
PM10 ($\mu\text{g}/\text{m}^3$)	34.426	11082	grid icon
NO2 ($\mu\text{g}/\text{m}^3$)	31.037	9991	grid icon
PM25 temporal coverage (%)	77.401	24916	grid icon
PM10 temporal coverage (%)	83.284	26810	grid icon
NO2 temporal coverage (%)	38.213	12301	grid icon
Reference	0.016	5	grid icon
Number and type of monitoring stations	72.794	23433	grid icon
Version of the database	0.000	0	grid icon
Status	100.000	32191	grid icon

Next steps: [View recommended plots](#)

W zbiorze danych znajduje się bardzo dużo brakujących danych. Kluczowe z punktu widzenia analizy zanieczyszczenia powietrza są kolumny:

- 'PM2.5 ($\mu\text{g}/\text{m}^3$)' - 53.254% brakujących danych
- 'PM10 ($\mu\text{g}/\text{m}^3$)' - 34.426% brakujących danych
- 'NO2 ($\mu\text{g}/\text{m}^3$)' - 31.037% brakujących danych
- 'PM25 temporal coverage (%)' - 77.401% brakujących danych
- 'PM10 temporal coverage (%)' - 83.284% brakujących danych
- 'NO2 temporal coverage (%)' - 38.213% brakujących danych

W każdej z tych kolumn braki są na tyle duże, że nie można pozwolić sobie na usunięcie wierszy z brakami - trzeba będzie w jakiś sposób uzupełnić braki w danych.

Jeśli chodzi o kolumny 'Reference', 'Number and type of monitoring stations', 'Version of the database' - oceniona zostanie ich przydatność w analizie i jeśli okażą się nieprzydatne to zostaną pominięte.

Kolumna 'Status' jest do do pominięcia, ponieważ zawiera jedynie wartości brakujące.

Kolumna 'Version of the database' jest do pominięcia, ponieważ wersja bazy danych nie ma potencjału, żeby wnieść cokolwiek do analizy.

Zbadany zostanie również brak wartości w 1 wierszu dla kolumny region.

```
df_filled = copy.deepcopy(df)
del df_filled['Status']
del df_filled['Version of the database']
```

1. Braki w kolumnie 'Region':

```
print(df_filled['Region'].unique())
df_filled[df_filled['Region'].isnull()]

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

Region	ISO3	Country	City	Year	PM25			
					PM2.5 (µg/m³)	PM10 (µg/m³)	NO2 (µg/m³)	temporal coverage (%)

Lichtenstein jest państwem europejskim, dlatego wartość nan może być zastąpiona przez 'European Region'

```
df_filled.loc[24778, 'Region'] = 'European Region'
df_filled.loc[24778, ]
```

Region	European Region
ISO3	LIE
Country	Liechtenstein
City	Vaduz
Year	2010
PM2.5 (µg/m³)	Nan
PM10 (µg/m³)	17.88
NO2 (µg/m³)	23.59
PM25 temporal coverage (%)	Nan
PM10 temporal coverage (%)	96.164
NO2 temporal coverage (%)	98.265
Reference	European Environment Information and Observati...
Number and type of monitoring stations	Nan
Name: 24778, dtype: object	

2. Braki w kolumnie 'Reference':

```
df_filled[df_filled['Reference'].isnull()]
```

Region	ISO3	Country	City	Year	PM25			
					PM2.5 (µg/m³)	PM10 (µg/m³)	NO2 (µg/m³)	temporal coverage (%)
28209 Eastern Mediterranean Region	QAT	Qatar	Doha	2017	44.0	148.0	29.0	98.0
28210 Eastern Mediterranean Region	QAT	Qatar	Doha	2018	44.0	181.0	47.0	99.0

Nie jest możliwe przewidzenie źródła danych, dlatego te wartości pozostaną jako Nan

3. Braki w kolumnie 'Number and type of monitoring stations':

```
print("\n\nIlość unikalnych wartości w kolumnie 'Number and type of monitoring stations': " + str(len(df_filled['Number and type of monitoring stations']))
show(df_filled[df_filled['Number and type of monitoring stations'].isnull()])
```

Ilość unikalnych wartości w kolumnie 'Number and type of monitoring stations': 635

10 entries per page Search:

	Region	ISO3	Country	City
0	Eastern Mediterranean Region	AFG	Afghanistan	Kabul
1	European Region	ALB	Albania	Durres
2	European Region	ALB	Albania	Durres
3	European Region	ALB	Albania	Elbasan
4	European Region	ALB	Albania	Elbasan
5	European Region	ALB	Albania	Elbasan
6	European Region	ALB	Albania	Korce
7	European Region	ALB	Albania	Korce
8	European Region	ALB	Albania	Vlore
9	European Region	ALB	Albania	Vlore

Showing 1 to 10 of 630 entries (downsampled from 23,433x13 to 630x13 as maxBytes=65536) « < 1 2 3 4 5 > »

```
df_null_natoms = df_filled[df_filled['Number and type of monitoring stations'].isnull()]
df_no_null_natoms = df_filled[df_filled['Number and type of monitoring stations'].notnull()]
df_missing_natoms = pd.concat([np.round(df_null_natoms.isnull().sum()/len(df_null_natoms)*100,3), np.round(df_no_null_natoms.isnull().sum()/len(df_no_null_natoms)*100,3)], axis=1)
df_missing_natoms.columns = ['natoms null - null percent', 'natoms not null - null percent']
df_missing_natoms
```

	natoms null - null percent	natoms not null - null percent	
Region	0.000	0.000	
ISO3	0.000	0.000	
Country	0.000	0.000	
City	0.000	0.000	
Year	0.000	0.000	
PM2.5 (µg/m³)	49.200	64.101	
PM10 (µg/m³)	36.030	30.132	
NO2 (µg/m³)	34.055	22.962	
PM25 temporal coverage (%)	80.719	68.520	
PM10 temporal coverage (%)	96.885	46.894	
NO2 temporal coverage (%)	39.645	34.380	
Reference	0.000	0.057	

Number and type of monitoring

Next steps: [View recommended plots](#)

Porównując braki w kolumnach dla obserwacji gdzie 'Number and type of monitoring stations' jest nullem vs gdzie nie jest, widoczny jest związek braku wartości w kolumnie 'PM10 temporal coverage (%)' z brakiem wartości w kolumnie 'Number and type of monitoring stations'. Jeśli 'Number and type of monitoring stations' jest nullem to w 96% przypadków 'PM10 temporal coverage (%)' jest nullem, natomiast w przeciwnym wypadku, jedynie 46% wartości 'PM10 temporal coverage (%)' jest nullem.

Ta zależność nie pomaga jednak w potencjalnym przewidzeniu wartości 'Number and type of monitoring stations', również ze względu na to, że istnieje aż 635 unikalnych etykiet w tej kolumnie. Z tego powodu wartości brakujące w tej kolumnie nie zostaną uzupełnione.

4. Braki w kolumnach 'PM2.5 (µg/m³)', 'PM10 (µg/m³)', 'NO2 (µg/m³)', 'PM25 temporal coverage (%)', 'PM10 temporal coverage (%)', 'NO2 temporal coverage (%)':

Rozkłady wartości zmiennych:

```
from matplotlib.gridspec import GridSpec

fig, ax = plt.subplots(1,2, figsize=(20,15))
gs = GridSpec(2, 3, figure=fig)

ax1 = plt.subplot(gs[0, :3])
sns.boxplot(df_filled[['PM2.5 (\mu g/m3)', 'PM10 (\mu g/m3)', 'NO2 (\mu g/m3)']], ax=ax1)
ax1.set_title('Boxploty zmennych mierzących zanieczyszczenie powietrza')
ax1.set_ylabel('μg/m3')
ax1.set_ylim([0,200])

# Plot on the second row (3 plots)
ax2 = plt.subplot(gs[1, 0])
sns.histplot(df_filled['PM2.5 (\mu g/m3)'], bins=20, color='skyblue', ax=ax2, kde=True)
ax2.set_title("Histogram zmiennej 'PM25 (\mu g/m3)'")

ax3 = plt.subplot(gs[1, 1])
sns.histplot(df_filled['PM10 (\mu g/m3)'], bins=40, color='lightgreen', ax=ax3, kde=True)
ax3.set_title("Histogram zmiennej 'PM10 (\mu g/m3)'")

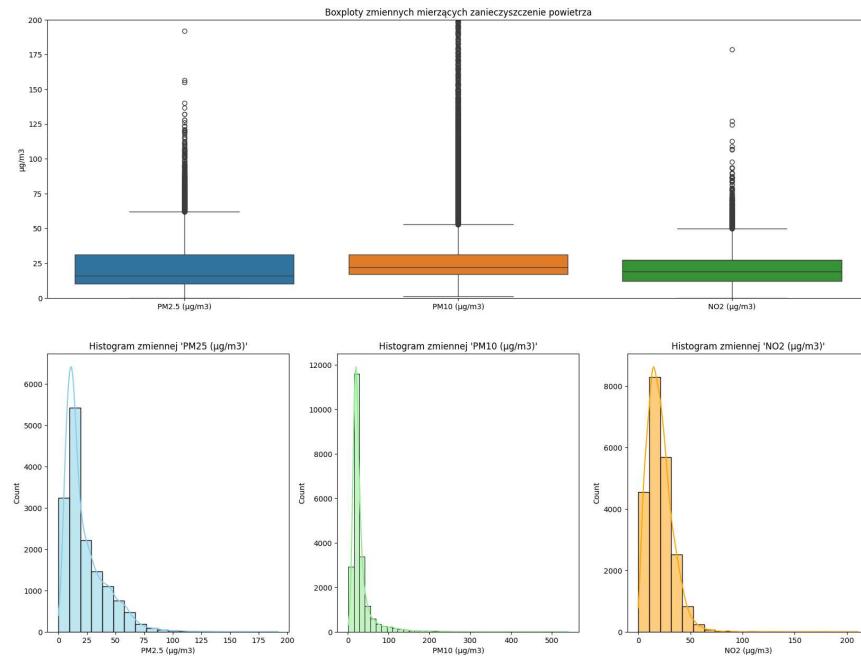
ax4 = plt.subplot(gs[1, 2])
sns.histplot(df_filled['NO2 (\mu g/m3)'], bins=20, color='orange', ax=ax4, kde=True)
ax4.set_title("Histogram zmiennej 'NO2 (\mu g/m3)'")

plt.suptitle('Rozkłady wartości zmennych mierzących zanieczyszczenie powietrza: PM10, PM25, NO2', fontsize=16)
plt.show()
```

```
<ipython-input-130-348204297da7>:6: MatplotlibDeprecationWarning:
```

Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two min

Rozkłady wartości zmiennych mierzących zanieczyszczenie powietrza: PM10, PM25, NO2



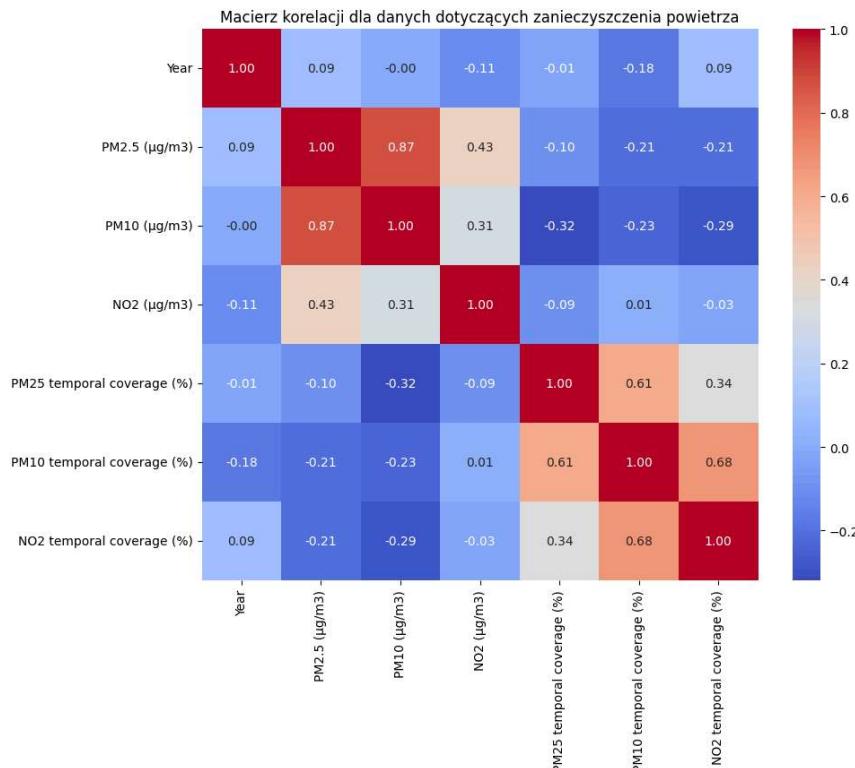
Wszystkie zmienne mają zbliżone rozkłady. Charakteryzują się największą liczbą zliczeń w okolicach wartości 10-20, a potem następuje gwałtowny spadek zliczeń dla kolejnych przedziałów. Są to rozkłady o dużej prawostronnej skośności.

Macierz korelacji

```
correlation_matrix = df_filled.corr(numeric_only = True)

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Macierz korelacji dla danych dotyczących zanieczyszczenia powietrza')

Text(0.5, 1.0, 'Macierz korelacji dla danych dotyczących zanieczyszczenia powietrza')
```



Macierz korelacji wskazuje bardzo dużą korelację między zmiennymi PM10 oraz PM25, a także sporą dodatnią korelację tych zmiennych do zmiennej NO2.

Widoczna jest również duża korelacja zmiennej 'PM10 temporal coverage (%)' z pozostałymi zmiennymi wskazującymi na procent pokrycia pomiarów pozostałych zanieczyszczeń

Ważna obserwacja:

Pył zawieszony PM10 to pyły o średnicy <= 10 mikrometrów, natomiast PM2.5 to pyły o średnicy <= 2.5 mikrometra. W teorii zatem wartości stężeń pyłu PM10 powinny być zawsze większe od wartości stężeń pyłów PM2.5.

```
print('Ilość obserwacji, gdzie stężenie PM10 było mniejsze od stężenie PM2.5:')
temp1 = df_filled[df_filled['PM10 (µg/m³)'] < df_filled['PM2.5 (µg/m³)']]
print(len(temp1))

print('\nIlość obserwacji, gdzie zmierzono zarówno PM10 jak i PM 2.5:')
df_PM_notnull = df_filled[df_filled['PM10 (µg/m³)'].notnull() & df_filled['PM2.5 (µg/m³)'].notnull()]
print(len(df_PM_notnull))

print('\nProcent obserwacji, gdzie stężenie PM10 było większe niż PM2.5 spośród obserwacji gdzie zmierzono oba wskaźniki:')
print(str(np.round(len(temp1)/len(df_PM_notnull) * 100,3))+'%')

df_filled[df_filled['PM10 (µg/m³)'] < df_filled['PM2.5 (µg/m³)']]
```

Ilość obserwacji, gdzie stężenie PM10 było mniejsze od stężenie PM2.5:
32

Ilość obserwacji, gdzie zmierzono zarówno PM10 jak i PM 2.5:
8824

Procent obserwacji, gdzie stężenie PM10 było większe niż PM2.5 spośród obserwacji gd
0.363%

	Region	IS03	Country	City	Year	PM2.5 ($\mu\text{g}/\text{m}^3$)	PM10 ($\mu\text{g}/\text{m}^3$)	NO2 ($\mu\text{g}/\text{m}^3$)	tei co'
2670	Region of the Americas	CAN	Canada	Brandon	2018	11.00	8.00	13.00	99.1
2866	Region of the Americas	CAN	Canada	Flin Flon	2018	13.00	12.00	NaN	97.1
8920	Region of the Americas	CRI	Costa Rica	San Jose	2018	22.00	21.75	40.71	100.1
9519	European Region	CZE	Czechia	Stachy	2013	9.44	8.53	4.92	
9520	European Region	CZE	Czechia	Stachy	2014	9.14	8.98	3.76	
12656	European Region	ESP	Spain	As Pontes De Garcia Rodriguez	2018	9.63	9.11	3.54	
12657	European Region	ESP	Spain	As Pontes De Garcia Rodriguez	2019	8.99	8.96	3.62	
14053	European Region	ESP	Spain	Oural	2015	10.91	10.79	7.47	
14054	European Region	ESP	Spain	Oural	2016	10.34	10.23	12.77	
14055	European Region	ESP	Spain	Oural	2017	11.64	11.42	17.97	
14909	European Region	EST	Estonia	Palmse	2019	5.49	5.46	2.10	99.1
15121	European Region	FIN	Finland	Raahe	2015	6.40	6.38	NaN	
19027	South East Asia Region	IND	India	Daman	2016	68.00	34.00	29.00	57.1
20460	South East Asia Region	IND	India	Silvassa	2016	73.00	37.00	32.00	57.1
21126	European Region	ISL	Iceland	Hafnarfjörður	2017	8.14	4.31	4.12	
22640	European Region	ITA	Italy	Laces	2014	13.16	12.54	NaN	

24883	European Region	LUX	Luxembourg	Bekerich	2010	16.34	14.79	15.03	96.
24918	European Region	LUX	Luxembourg	Uewerpallen	2013	16.42	11.65	14.25	
26168	Western Pacific Region	NZL	New Zealand	South Waikato District	2016	14.66	13.11	NaN	
26534	European Region	POL	Poland	Czestochowa	2010	40.94	38.95	27.61	95.:
26952	European Region	POL	Poland	Lomza	2013	27.90	27.30	12.56	
26954	European Region	POL	Poland	Lomza	2015	26.64	26.48	14.76	
26955	European Region	POL	Poland	Lomza	2016	25.82	23.73	13.29	
26956	European Region	POL	Poland	Lomza	2017	25.58	24.97	13.45	

Jak widać tylko w 0.36% przypadków pomiar PM10 był większy niż PM2.5, może to wynikać z faktu, że oba wskaźniki są zbierane z użyciem różnych metod. Mimo tych 32 odstępstw od reguły wydaje się, że brakujące wartości jednego z tych czynników można uzupełnić w oparciu o pomiar drugiego z nich. Argumentem przemawiającym za sensownością tego rozwiązania jest również wysoka korelacja między zmiennymi.

Scenariusze brakujących wartości:

```
print('Ilość obserwacji, dla których PM10 jest null, a PM2.5 nie jest null:\n')
print(len(df_filled[df_filled['PM10 (\mu g/m³)'].isnull() & df_filled['PM2.5 (\mu g/m³)'].notnull()]))
print('\n\nIlość obserwacji, dla których PM10 nie jest null, a PM2.5 jest null:\n')
print(len(df_filled[df_filled['PM10 (\mu g/m³)'].notnull() & df_filled['PM2.5 (\mu g/m³)'].isnull()]))
print('\n\nIlość obserwacji, dla których PM10 i PM2.5 są nullami:\n')
print(len(df_filled[df_filled['PM10 (\mu g/m³)'].isnull() & df_filled['PM2.5 (\mu g/m³)'].isnull()]))
```

Ilość obserwacji, dla których PM10 jest null, a PM2.5 nie jest null:

6224

Ilość obserwacji, dla których PM10 nie jest null, a PM2.5 jest null:

12285

Ilość obserwacji, dla których PM10 i PM2.5 są nullami:

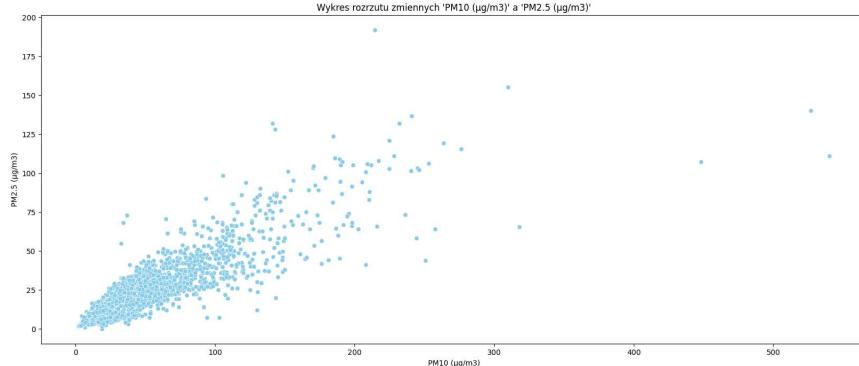
4858

Wniosek: jeśli uda się skutecznie określić wartość jednej ze zmiennych PM na podstawie drugiej, możliwe będzie uzupełnienie 18500 brakujących wartości.

Rozrzut zmiennych PM10 i PM2.5

```
fig, ax = plt.subplots(1, 1, figsize = (20,8))
sns.scatterplot(x= df_filled['PM10 (\mu g/m³)'], y= df_filled['PM2.5 (\mu g/m³)'], color='skyblue', ax=ax)
ax.set_title("Wykres rozrzutu zmiennych 'PM10 (\mu g/m³)' a 'PM2.5 (\mu g/m³)'")
```

Text(0.5, 1.0, "Wykres rozrzutu zmiennych 'PM10 (\mu g/m3)' a 'PM2.5 (\mu g/m3)'")



Miedzy zmiennymi występuje silna liniowa zależność, jednak wraz z rosnącymi wartościami, rośnie wariancja.

Jak prezentują się różnice w wartościach między zmiennymi?

```
df_PM_notnull['difference'] = df_PM_notnull['PM10 (\mu g/m3)'] - df_PM_notnull['PM2.5 (\mu g/m3)']
```

```
<ipython-input-135-b0f656b531b8>:1: SettingWithCopyWarning:
```

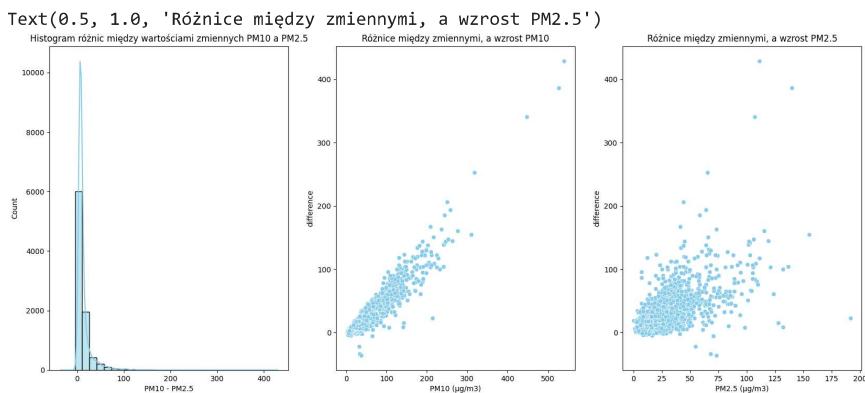
```
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus
```

```
fig, ax = plt.subplots(1, 3, figsize = (20,8))
sns.histplot(df_PM_notnull['difference'], bins=30, color='skyblue', kde=True, ax=ax[0])
ax[0].set_title("Histogram różnic między wartościami zmiennych PM10 a PM2.5")
ax[0].set_xlabel("PM10 - PM2.5")

sns.scatterplot(x = df_PM_notnull['PM10 (\mu g/m3)'], y = df_PM_notnull['difference'], color='skyblue', ax=ax[1])
ax[1].set_title("Różnice między zmiennymi, a wzrost PM10")

sns.scatterplot(x = df_PM_notnull['PM2.5 (\mu g/m3)'], y = df_PM_notnull['difference'], color='skyblue', ax=ax[2])
ax[2].set_title("Różnice między zmiennymi, a wzrost PM2.5")
```

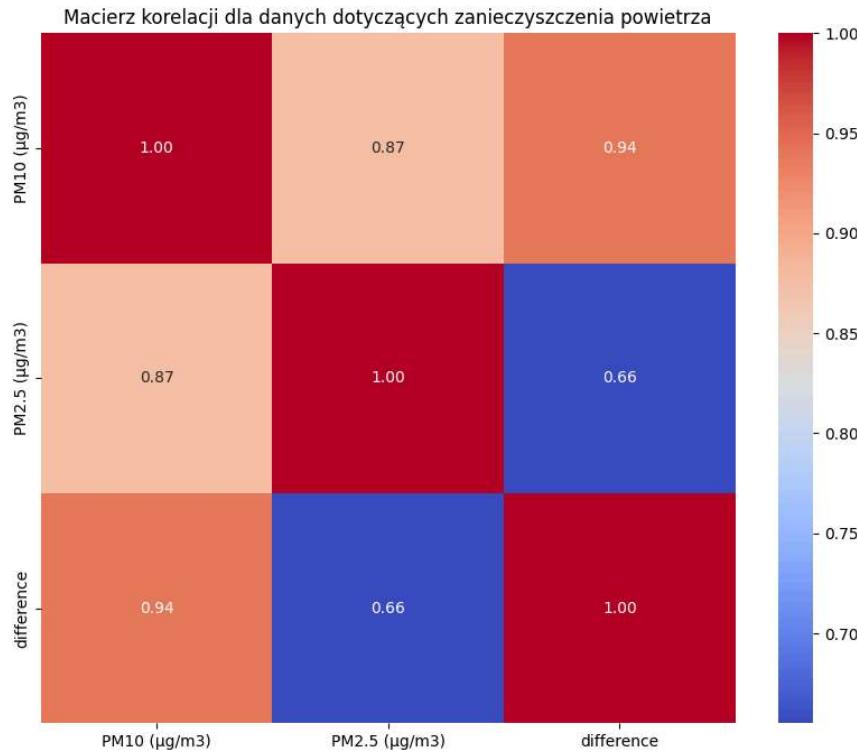


Macierz korelacji różnicy między zmiennymi, a zmiennymi:

```
correlation_matrix = df_PM_notnull[['PM10 (\mu g/m3)', 'PM2.5 (\mu g/m3)', 'difference']].corr()
```

```
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Macierz korelacji dla danych dotyczących zanieczyszczenia powietrza')
```

Text(0.5, 1.0, 'Macierz korelacji dla danych dotyczących zanieczyszczenia powietrza')



Między wzrostem zmiennych PM, a różnicą pomiarów występuje silna dodatnia korelacja. Na podstawie wykresów rozrzutu, widać, że wzrost jest niemal liniowa.

Aby zobrazować ten wzrost różnicy, stworzony zostanie model regresji liniowej, który dopasuje proste do tej zależności. Ich równania posłużą do uzupełnienia brakujących wartości, w 2 scenariuszach:

1. Dane jest stężenie PM10, natomiast stężenie PM2.5 jest nullem - w tym przypadku wykorzystany zostanie model dopasowujący różnicę do wartości PM10. Następnie w celu uzyskania wartości PM2.5 różnica zostanie odjęta od wartości stężenia PM10.
2. Dane jest stężenie PM2.5, natomiast stężenie PM10 jest nullem - w tym przypadku wykorzystany zostanie model dopasowujący różnicę do wartości PM2.5. Następnie w celu uzyskania wartości PM10 różnica zostanie dodana do wartości stężenia PM2.5.

Model regresji w scenariuszu 1

```

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score

X_train, X_test, y_train, y_test = train_test_split(df_PM_notnull['PM10 (\mu g/m3)'], df_PM_notnull['difference'], test_size=0.2, random_s

# Creating a linear regression model
model_1 = LinearRegression()

X_train = X_train.values.reshape(-1,1)
y_train = y_train.values.reshape(-1,1)
X_test = X_test.values.reshape(-1,1)
y_test = y_test.values.reshape(-1,1)
model_1.fit(X_train, y_train)

y_pred = model_1.predict(X_test)

model1_coeff = model_1.coef_
model1_intercept = model_1.intercept_

print("Coefficients:", model1_coeff)
print("Intercept:", model1_intercept)

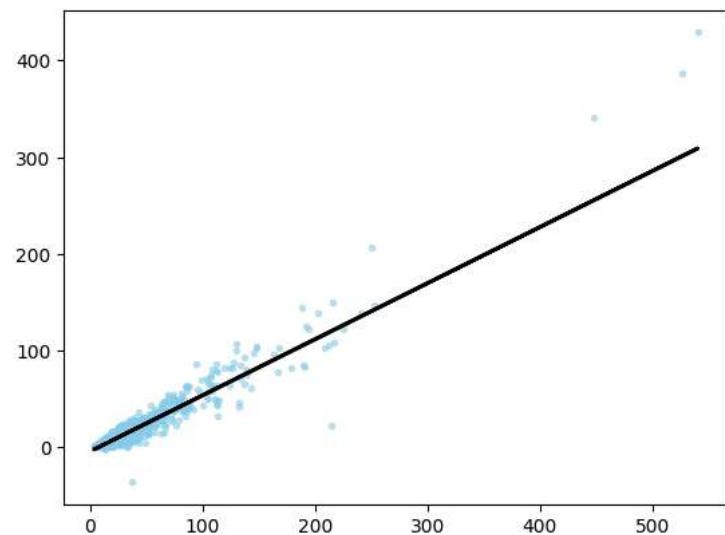
# # Mean squared error
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)

# R-squared score
r2 = r2_score(y_test, y_pred)
print("R-squared:", r2)
# Plotting the data and the regression line
plt.scatter(X_test, y_test, color='skyblue', s=8, alpha=0.5)
plt.plot(X_test, y_pred, color='black', linewidth=2)

plt.show()

```

Coefficients: [[0.58021177]]
 Intercept: [-4.21608103]
 Mean Squared Error: 54.82713789056066
 R-squared: 0.8989620265055162



Model dla 1 scenariusza ma wysoki współczynnik R2 wynoszący więcej niż 0.85 i wydaje się być dobrze dopasowanym do trendu.

Model regresji w scenariuszu 2

```
X_train, X_test, y_train, y_test = train_test_split(df_PM_notnull['PM2.5 (\mu g/m3)'], df_PM_notnull['PM10 (\mu g/m3)'], test_size=0.2, random_state=42)

# Creating a linear regression model
model_2 = LinearRegression()

X_train = X_train.values.reshape(-1,1)
y_train = y_train.values.reshape(-1,1)
X_test = X_test.values.reshape(-1,1)
y_test = y_test.values.reshape(-1,1)
model_2.fit(X_train, y_train)

y_pred = model_2.predict(X_test)

model2_coeff = model_2.coef_
model2_intercept = model_2.intercept_

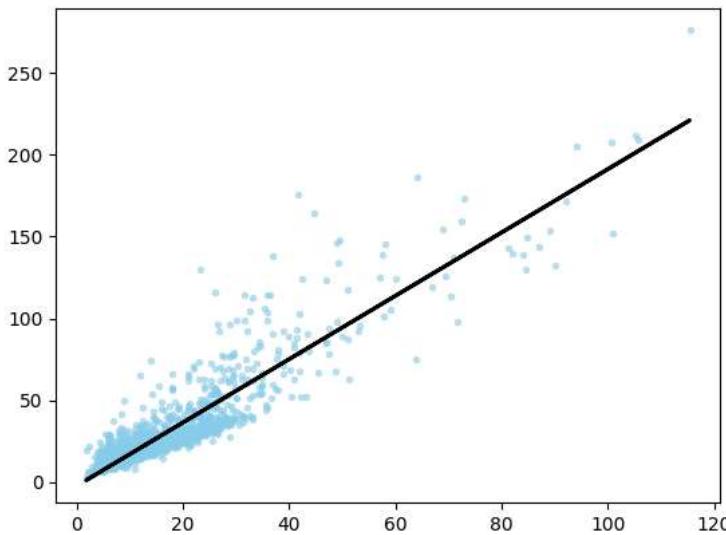
print("Coefficients:", model2_coeff)
print("Intercept:", model2_intercept)

# # Mean squared error
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)

# R-squared score
r2 = r2_score(y_test, y_pred)
print("R-squared:", r2)
# Plotting the data and the regression line
plt.scatter(X_test, y_test, color='skyblue', s=8, alpha=0.5)
plt.plot(X_test, y_pred, color='black', linewidth=2)

plt.show()
```

Coefficients: [[1.93799283]]
 Intercept: [-2.51610007]
 Mean Squared Error: 118.78624164557276
 R-squared: 0.8103982733751854



Ponieważ model 'difference ~ PM2.5 (\mu g/m3)' radził sobie słabo - R2 mniejsze niż 0.5, w 2 scenariuszu użyty zostanie model 'PM10 (\mu g/m3) ~ PM2.5 (\mu g/m3)', który spisuje się gorzej niż model dla pierwszego scenariusza - R2 na poziomie ok. 0.75, ale wciąż wyniki zdają się być satysfakcjonujące.

Uzupełnienie wartości brakujących w kolumnach 'PM10 (\mu g/m3)' i 'PM2.5 (\mu g/m3)':

Scenariusz 1 (PM10 nie jest nullem, PM2.5 tak)

W tym scenariuszu przewidywana jest wartość różnicy między zmiennymi, którą następnie należy odjąć od wartości kolumny 'PM10 (\mu g/m3)'.

```
print("Liczba nulli w kolumnie 'PM2.5 (\mu g/m3)' przed operacją:")
print(len(df_filled[df_filled['PM2.5 (\mu g/m3)'].isnull()]))
df_filled['PM2.5 (\mu g/m3)'] = df_filled.apply(lambda row: row['PM10 (\mu g/m3)'] - (row['PM10 (\mu g/m3)'] * model1_coeff[0][0] + model1_intercept))

print("\nLiczba nulli w kolumnie 'PM2.5 (\mu g/m3)' po operacji:")
print(len(df_filled[df_filled['PM2.5 (\mu g/m3)'].isnull()]))
show(df_filled['PM2.5 (\mu g/m3)'])
```

Liczba nulli w kolumnie 'PM2.5 ($\mu\text{g}/\text{m}^3$)' przed operacją:
17143

Liczba nulli w kolumnie 'PM2.5 ($\mu\text{g}/\text{m}^3$)' po operacji:
4858

10 ▾ entries per page

Search:

PM2.5 ($\mu\text{g}/\text{m}^3$)
119.77
11.625343
14.32
NaN
NaN
NaN
30.34
28.64
10.617852
12.355775

Showing 1 to 10 of 8,192 entries ([downsampled](#) from
32,191x1 to 8,192x1 as maxBytes=65536)

« < 1 2 3 4 5 > »

Scenariusz 2 (PM10 jest nullem, PM2.5 nie)

W tym scenariuszu przewidywana jest bezpośrednio wartość kolumny 'PM10 ($\mu\text{g}/\text{m}^3$)' w zależności od 'PM2.5 ($\mu\text{g}/\text{m}^3$)'.

```
print("Liczba nulli w kolumnie 'PM10 ( $\mu\text{g}/\text{m}^3$ )' przed operacją:")
print(len(df_filled[df_filled['PM10 ( $\mu\text{g}/\text{m}^3$ )'].isnull()]))
df_filled['PM10 ( $\mu\text{g}/\text{m}^3$ )'] = df_filled.apply(lambda row: (row['PM2.5 ( $\mu\text{g}/\text{m}^3$ )'] * model2_coeff[0][0] + model2_intercept[0]) if pd.isnull(
print("\nLiczba nulli w kolumnie 'PM10 ( $\mu\text{g}/\text{m}^3$ )' po operacji:")
print(len(df_filled[df_filled['PM10 ( $\mu\text{g}/\text{m}^3$ )'].isnull()]))
show(df_filled['PM10 ( $\mu\text{g}/\text{m}^3$ )'])
```

Liczba nulli w kolumnie 'PM10 ($\mu\text{g}/\text{m}^3$)' przed operacją:
11082

Liczba nulli w kolumnie 'PM10 ($\mu\text{g}/\text{m}^3$)' po operacji:
4858

10 ▾ entries per page

Search:

PM10 ($\mu\text{g}/\text{m}^3$)
229.597301
17.65
24.56
NaN
NaN
NaN
45.31
40.21
15.25
19.39

Showing 1 to 10 of 8,192 entries ([downsampled](#) from
32,191x1 to 8,192x1 as maxBytes=65536)

« < 1 2 3 4 5 > »

Aby edytować zawartość komórki, kliknij ją dwukrotnie (lub naciśnij klawisz Enter)

Dzięki przeprowadzeniu operacji uzupełniania wartości brakujących w kolumnach 'PM10 ($\mu\text{g}/\text{m}^3$)' oraz 'PM2.5 ($\mu\text{g}/\text{m}^3$)' znaczco zredukowana została ilość nulli a także zwiększyły się możliwości interpretacyjne tego zbioru danych.

Ze względu na brak podobnej zależności dla zmiennej 'NO2 ($\mu\text{g}/\text{m}^3$)', wartości brakujące tej kolumny nie będą uzupełniane.

Podobnie, wartości brakujące nie będą uzupełniane dla kolumn związanych z pokryciem czasowym danych.

Braki w danych po uzupełnianiu:

```
df_null = pd.concat([np.round(df_filled.isnull().sum()/len(df)*100,3), df_filled.isnull().sum()], axis=1)
df_null.columns = ['null percent', 'null count']
df_null
```

	null percent	null count	grid icon
Region	0.000	0	grid icon
ISO3	0.000	0	grid icon
Country	0.000	0	grid icon
City	0.000	0	grid icon
Year	0.000	0	grid icon
PM2.5 ($\mu\text{g}/\text{m}^3$)	15.091	4858	grid icon
PM10 ($\mu\text{g}/\text{m}^3$)	15.091	4858	grid icon
NO2 ($\mu\text{g}/\text{m}^3$)	31.037	9991	grid icon
PM25 temporal coverage (%)	77.401	24916	grid icon
PM10 temporal coverage (%)	83.284	26810	grid icon
NO2 temporal coverage (%)	38.213	12301	grid icon
Reference	0.016	5	grid icon
Number and type of monitoring stations	72.794	23433	grid icon

Next steps: [View recommended plots](#)

- Brakujące wartości w kolumnie 'PM2.5 ($\mu\text{g}/\text{m}^3$)' zredukowano z 53.254% do 15.091%
- Brakujące wartości w kolumnie 'PM10 ($\mu\text{g}/\text{m}^3$)' zredukowano z 34.426% do 15.091%

EDA dla różnych poziomów generalizacji

Dodanie geometrii dla krajów, żeby umożliwić przedstawienie analizy na mapach:

```
import geopandas as gpd
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world
```

<ipython-input-143-6a997cdce43c>:2: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You

	pop_est	continent	name	iso_a3	gdp_md_est	geometry	grid icon
0	889953.0	Oceania	Fiji	FJI	5496	MULTIPOLYGON (((180.000000 -16.06713, 180.000000...))	grid icon
1	58005463.0	Africa	Tanzania	TZA	63177	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...))	grid icon
2	603253.0	Africa	W. Sahara	ESH	907	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...))	grid icon
3	37589262.0	North America	Canada	CAN	1736425	MULTIPOLYGON (((-122.84000 49.00000, -122.9742...)))	grid icon
4	328239523.0	North America	United States of	USA	21433226	MULTIPOLYGON (((-122.84000 49.00000...)))	grid icon

Next steps:

 View recommended plots

```
df_merged = df_filled.merge(world, how = 'left', left_on='ISO3', right_on='iso_a3')
del df_merged['name']
del df_merged['iso_a3']
del df_merged['Reference']
df_merged = gpd.GeoDataFrame(df_merged, geometry = 'geometry')
df_merged[df_merged['geometry'].isnull()]['Country'].unique()

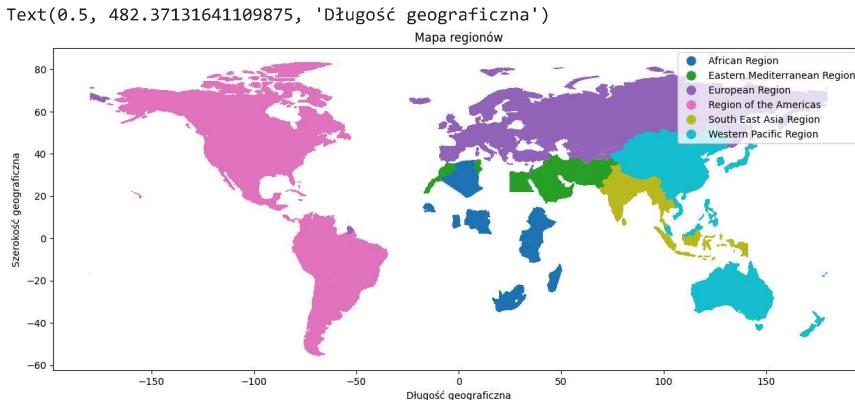
array(['Andorra', 'Bahrain', 'Liechtenstein', 'Monaco', 'Maldives',
       'Malta', 'Mauritius', 'Singapore'], dtype=object)
```

Przy łączeniu usunięto powtarzające się kolumny w danych world oraz kolumnę 'Reference', która nie będzie wykorzystywana w dalszych analizach.

W wyniku merge'u tylko 8 małych krajów pozostało bez poligonów, gdyż nie było ich w zbiorze danych 'naturalearth_lowres'.

▼ Analiza dla regionów:

```
fig, ax = plt.subplots(1, 1, figsize = [15, 15])
df_merged.plot(column='Region', linewidth = 0.8, ax = ax, legend = True)
ax.set_title('Mapa regionów')
ax.set_ylabel('Szerokość geograficzna')
ax.set_xlabel('Długość geograficzna')
```



Regiony pokrywają większość powierzchni lądowej globu. Największe braki występują w Afryce, gdzie dane zebrane zostały jedynie dla kilku krajów. Największym regionem jest ten zawierającym obie Ameryki, a najmniejszym ten z Krajami Bliskiego Wschodu.

Liczebność obserwacji oraz ilość miast dla których zebrano dane wg regionów:

```
region_grouped = df_merged.groupby('Region')
print('Liczebność obserwacji wg regionów:\n')
print(region_grouped.size())

print('\n\nIlość miast dla których zebrano dane wg regionów:\n')

region_grouped['City'].nunique()
```

Liczebność obserwacji wg regionów:

Region	Liczba obserwacji
African Region	1
Eastern Mediterranean Region	1
European Region	1
Region of the Americas	8
South East Asia Region	1
Western Pacific Region	1

```
African Region           191
Eastern Mediterranean Region    438
European Region          20293
Region of the Americas      3957
South East Asia Region      2514
Western Pacific Region      4798
dtype: int64
```

Ilość miast dla których zebrano dane wg regionów:

```
Region
African Region           43
Eastern Mediterranean Region    159
European Region          3736
Region of the Americas      779
South East Asia Region      489
Western Pacific Region      1693
Name: City, dtype: int64
```

```
colors = ['blue', 'green', 'darkorchid', 'hotpink', 'yellow', 'cyan']
fig, ax = plt.subplots(2, 1, figsize = [15, 8])

sns.barplot(region_grouped.size(), ax=ax[0], legend=True, palette=colors)
ax[0].set_title('Wykres słupkowy ilości obserwacji wg regionu')

sns.barplot(region_grouped['City'].nunique(), ax=ax[1], legend=True, palette=colors)
ax[1].set_title('Wykres słupkowy ilości obserwacji wg regionu')

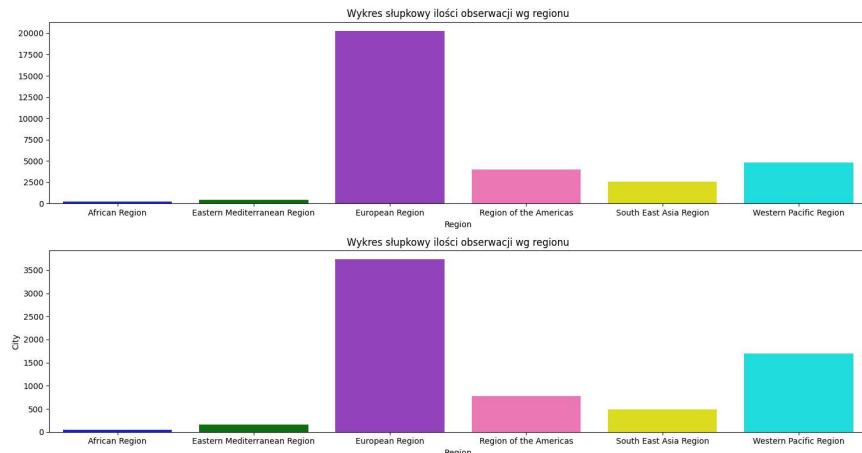
plt.tight_layout()
```

```
<ipython-input-147-c31bd25af112>:4: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

```
<ipython-input-147-c31bd25af112>:7: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14



Najwięcej danych w zbiorze zebrane zostało dla regionu europejskiego. Regiony położone w Afryce i na Bliskim Wschodzie są najmniej reprezentatywne i posiadają mniej niż 500 obserwacji.

Ilości obserwacji mogą wynikać z dostępności danych n.t. zanieczyszczeń w poszczególnych regionach. W Europie kraje przywiązuja największą wagę do czystości środowiska naturalnego, dlatego naturalne jest, że występuje tam najwięcej czujników badających zanieczyszczenie powietrza.

Małe ilości obserwacji w Afryce i regionie Bliskiego Wschodu mogą wynikać z faktu, że w tych regionach poza kilkoma wyjątkami występują kraje rozwijające się. Ze względu na wiele innych problemów o wyższym priorytecie, monitorowanie jakości powietrza ma tam mniejsze znaczenie.

Wiarygodność interpretację ilości pomiarów wzmacnia również wykres przedstawiający liczbę miast, w których zbierano dane. Tutaj również zdecydowaną większość ma region Europy - 3736 miast. Wysoki wynik ma również region Zachodniego Pacyfiku - 1693 miasta. Może to wynikać z faktu, że do tego regionu zaliczane są rozwinięte kraje jak Japonia czy Australia, które podobnie jak kraje Unii Europejskiej duży nacisk kładą na środowisko naturalne, a także Chiny, w których olbrzymia liczba miast może implikować występowanie dużej ilości czujników monitorujących jakość powietrza.

Brakujące wartości wg regionów:

```
region_grouped.apply(lambda x: x.isnull().sum() / len(x) * 100)
```

	Region	IS03	Country	City	Year	PM2.5 ($\mu\text{g}/\text{m}^3$)	PM10 ($\mu\text{g}/\text{m}^3$)	NO2 ($\mu\text{g}/\text{m}^3$)	tem	cov
Region										
African Region	0.0	0.0	0.0	0.0	0.0	6.806283	6.806283	35.078534	20.4	
Eastern Mediterranean Region	0.0	0.0	0.0	0.0	0.0	1.826484	1.826484	65.981735	83.5	
European Region	0.0	0.0	0.0	0.0	0.0	16.113931	16.113931	16.429311	94.8	
Region of the Americas	0.0	0.0	0.0	0.0	0.0	12.029315	12.029315	45.868082	70.9	
South East Asia Region	0.0	0.0	0.0	0.0	0.0	42.243437	42.243437	12.092283	78.0	

Dla kolumn zawierających stężenie pyłów PM największe braki obserwowane są w regionie 'South East Asia Region' - 42.2%, a najmniejsze w 'Western Pacific Region' - 0.6%

Jeśli chodzi o kolumnę NO2, największe braki występują w kolumnie 'Western Pacific Region' - 87.1%, a namniejsze w 'South East Asia Region' - 12.1%.

Ogółem widoczny jest trend - jeśli są małe braki w stężeniach PM, to są duże braki w stężeniu NO2 i odwrotnie. Wyjątkiem tu jest Europa, gdzie braki są równe dla wszystkich stężeń.

Rozkłady wartości zanieczyszczeń wg regionów

1. PM10 ($\mu\text{g}/\text{m}^3$)

```
fig, ax = plt.subplots(figsize = [20, 12])
gs = GridSpec(2, 6, figure=fig)

df_merged = df_merged.sort_values(by='Region')

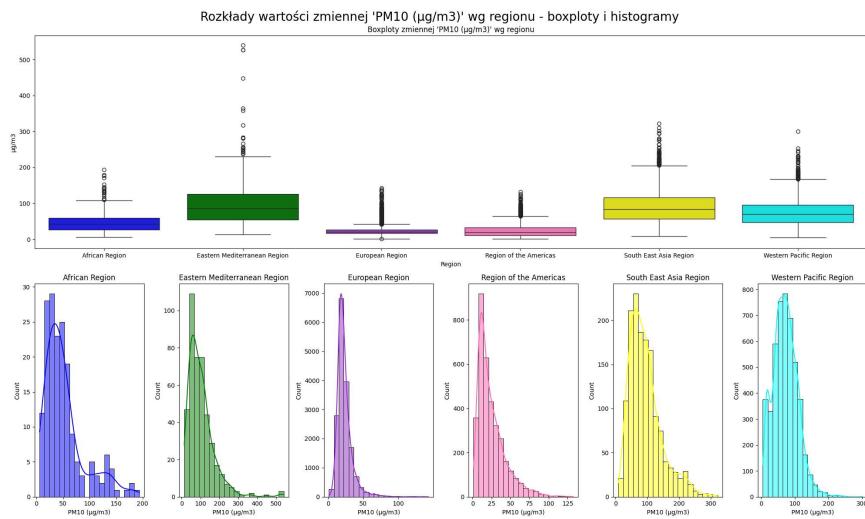
ax1 = plt.subplot(gs[0, :6])
sns.boxplot(x=df_merged['Region'], y=df_merged['PM10 ( $\mu\text{g}/\text{m}^3$ )'], ax=ax1, palette=colors)
ax1.set_title("Boxploty zmiennej 'PM10 ( $\mu\text{g}/\text{m}^3$ )' wg regionu")
ax1.set_ylabel(' $\mu\text{g}/\text{m}^3$ ')

i = 0
for region in df_merged['Region'].unique():
    ax_i = plt.subplot(gs[1, i])
    data = df_merged[df_merged['Region'] == region]
    sns.histplot(data['PM10 ( $\mu\text{g}/\text{m}^3$ )'], ax=ax_i, bins=20, kde=True, color = colors[i])
    ax_i.set_title(region)
    i += 1

plt.suptitle("Rozkłady wartości zmiennej 'PM10 ( $\mu\text{g}/\text{m}^3$ )' wg regionu - boxploty i histogramy", fontsize=20)
plt.tight_layout()
```

```
<ipython-input-149-6325248b9eae>:6: MatplotlibDeprecationWarning:
Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two min
<ipython-input-149-6325248b9eae>:7: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14



2. PM2.5 (µg/m³)

```
fig, ax = plt.subplots(figsize = [20, 12])
gs = GridSpec(2, 6, figure=fig)

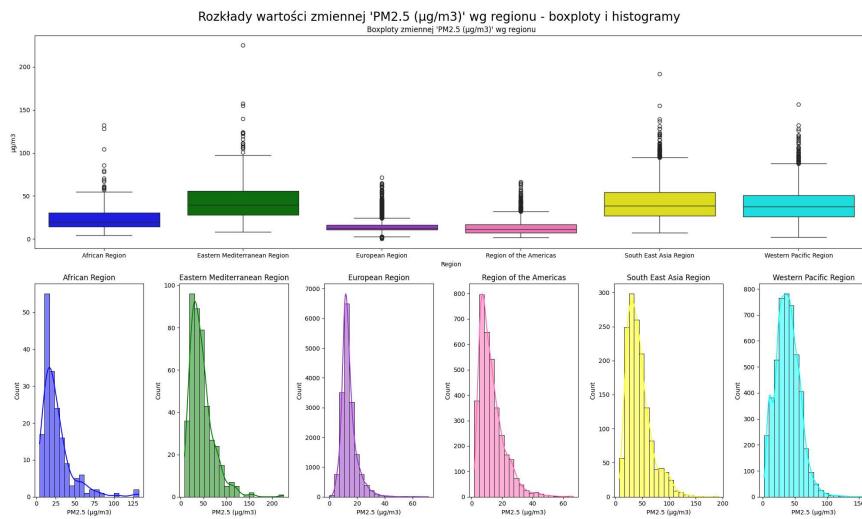
ax1 = plt.subplot(gs[0, :6])
sns.boxplot(x=df_merged['Region'], y=df_merged['PM2.5 (µg/m³)'], ax=ax1, palette=colors)
ax1.set_title("Boxploty zmiennej 'PM2.5 (µg/m³)' wg regionu")
ax1.set_ylabel('µg/m³')

i = 0
for region in df_merged['Region'].unique():
    ax_i = plt.subplot(gs[1, i])
    data = df_merged[df_merged['Region'] == region]
    sns.histplot(data['PM2.5 (µg/m³)'], ax=ax_i, bins=20, kde=True, color = colors[i])
    ax_i.set_title(region)
    i += 1

plt.suptitle("Rozkłady wartości zmiennej 'PM2.5 (µg/m³)' wg regionu - boxploty i histogramy", fontsize=20)
plt.tight_layout()
```

```
<ipython-input-150-b40f5bfdf989>:4: MatplotlibDeprecationWarning:  
Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two min  
<ipython-input-150-b40f5bfdf989>:5: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14



3. NO₂ ($\mu\text{g}/\text{m}^3$)

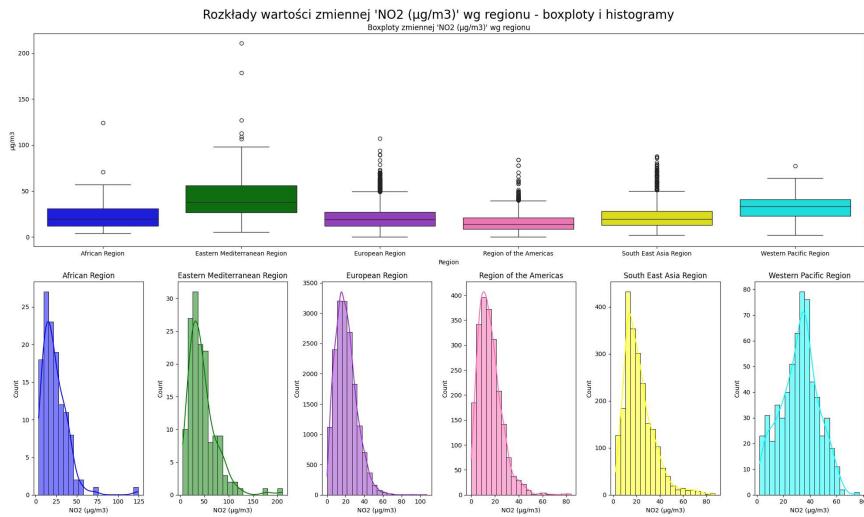
```
fig, ax = plt.subplots(figsize = [20, 12])  
gs = GridSpec(2, 6, figure=fig)  
  
ax1 = plt.subplot(gs[0, :6])  
sns.boxplot(x=df_merged['Region'], y=df_merged['NO2 ( $\mu\text{g}/\text{m}^3$ )'], ax=ax1, palette=colors)  
ax1.set_title("Boxploty zmiennej 'NO2 ( $\mu\text{g}/\text{m}^3$ )' wg regionu")  
ax1.set_ylabel(' $\mu\text{g}/\text{m}^3$ ')  
  
i = 0  
for region in df_merged['Region'].unique():  
    ax_i = plt.subplot(gs[1, i])  
    data = df_merged[df_merged['Region'] == region]  
    sns.histplot(data['NO2 ( $\mu\text{g}/\text{m}^3$ )'], ax=ax_i, bins=20, kde=True, color = colors[i])  
    ax_i.set_title(region)  
    i += 1  
plt.suptitle("Rozkłady wartości zmiennej 'NO2 ( $\mu\text{g}/\text{m}^3$ )' wg regionu - boxploty i histogramy", fontsize=20)  
plt.tight_layout()
```

<ipython-input-151-4d3120bdcb60>:4: MatplotlibDeprecationWarning:

Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two min

<ipython-input-151-4d3120bdcb60>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

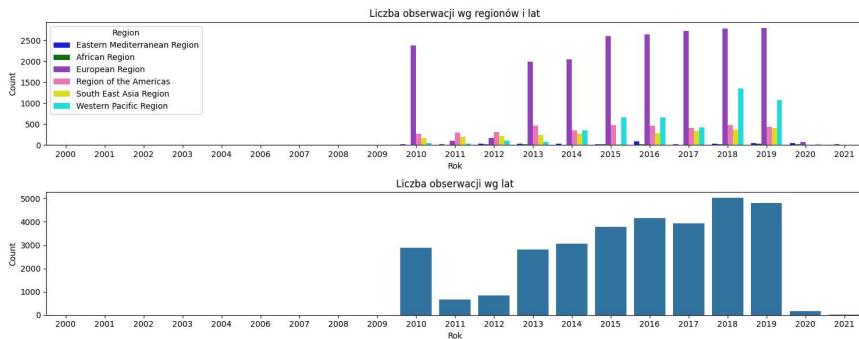


- Jak widać na boxplotach, najbardziej zanieczyszczony region dla każdego z badanych zanieczyszczeń to Bliski Wschód. Ponownie głównym powodem wydaje się być fakt, że większość krajów w tym regionie to kraje rozwijające się, które charakteryzują się wysokoemisyjnym przemysłem.
- Region, który dla stężeń PM2.5 oraz PM10 niemal dorównuje Bliskiemu Wschodowi, to Azja Południowo-Wschodnia, a na trzecim miejscu pod tym względem płasuje się Region Zachodniego Pacyfiku. Przypuszczalnie wynika to z występowania w tych regionach kolejno Indii oraz Chin, których struktura przemysłu oraz gęstość zaludnienia wpływają negatywnie na jakość powietrza.
- Regiony o najmniejszych stężeniach zanieczyszczeń to Europa oraz Ameryki. Interesujące są niskie wartości dla regionu Ameryk, jednak na tym poziomie generalizacji ciężko ocenić skąd biorą się te wartości.
- histogramy oaz estymowane funkcji gęstości prawdopodobieństwa dla każdej ze zmiennych są podobne w większości regionów. Charakteryzują się ostrym wzrostem od wartości 0 aż do maximum zliczeń a następnie nieco łagodniejszym dla rosnących wartości stężeń, który powoduje prawostrońską skośność tych rozkładów.
- Wyjątek stanowi tu region Zachodniego Pacyfiku, dla którego po początkowym wzroście obserwacji dla małych stężeń zanieczyszczeń, później następuje spadek zliczeń, a następnie wzrost zliczeń dla rosnących wartości stężeń. Przypuszczalnie wynika to z faktu, że w tym regionie jest mieszanka krajów wysoko rozwiniętych i dbających o jakość powietrza jak Australia czy Japonia, a także z drugiej strony Chin, które mniej dbają o jakość powietrza.

Ilość obserwacji wg regionu i roku pomiaru

```
grouped_df = df_merged.groupby(['Year', 'Region']).size().reset_index(name='Count')
grouped_df_year = df_merged.groupby('Year').size().reset_index(name='Count')
# Plotting using Seaborn
fig, ax = plt.subplots(2,1, figsize=(15, 6))
sns.barplot(data=grouped_df, x='Year', y='Count', hue='Region', palette=colors, ax=ax[0])
ax[0].set_title('Liczba obserwacji wg regionów i lat')
ax[0].set_xlabel('Rok')
ax[0].set_ylabel('Count')

sns.barplot(data=grouped_df_year, x='Year', y='Count', ax=ax[1])
ax[1].set_title('Liczba obserwacji wg lat')
ax[1].set_xlabel('Rok')
ax[1].set_ylabel('Count')
plt.tight_layout()
plt.show()
```



Jak widać w latach 2000-2009 praktycznie nie ma danych, z tego powodu te obserwacje mogą zostać usunięte ze zbioru danych.

Podobnie, jest bardzo mało obserwacji i tylko dla niektórych regionów w latach 2020 i 2021. Wynika to najprawdopodobniej z faktu, że wszystkie dane nie zostały jeszcze zebrane. Dane te również zostaną usunięte ze zbioru ze względu na niepełny obraz jaki mogą dawać.

Oprócz tego, widoczny jest delikatny trend wzrostowy w kolejnych latach jeśli chodzi o ilość obserwacji. Może to świadczyć o zwiększającej się globalnie świadomości konieczności monitorowania jakości powietrza.

```
df_merged = df_merged[df_merged['Year'].between(2010, 2019)]
```

Zmiany średniego stężenia zanieczyszczeń wg regionów w czasie

1. PM10 ($\mu\text{g}/\text{m}^3$)

```

region_time_groups = df_merged.groupby(['Region', 'Year'])[['PM2.5 (\mu g/m3)', 'PM10 (\mu g/m3)', 'NO2 (\mu g/m3)']].mean()

regions = df_merged['Region'].unique()

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

for j, region in enumerate(regions):
    region_data = region_time_groups.loc[region,:]

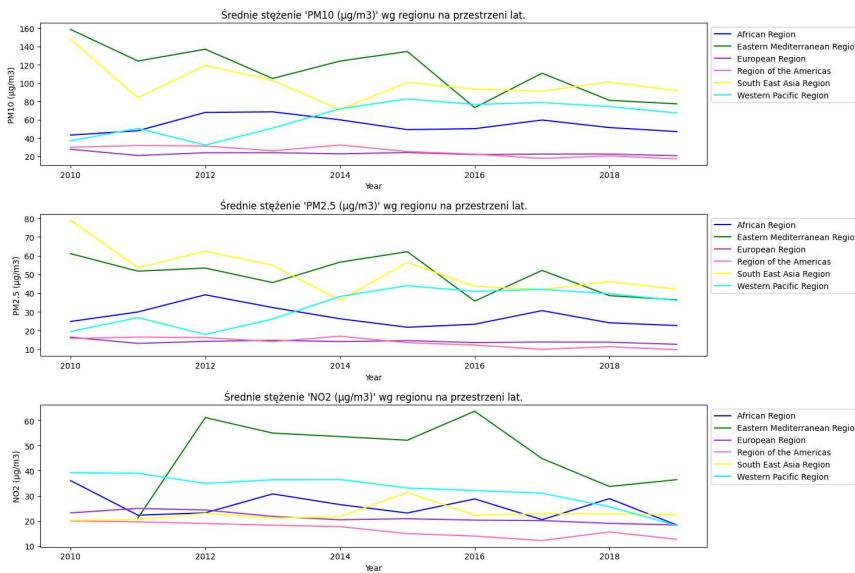
    sns.lineplot(x = region_data.index, y = region_data['PM10 (\mu g/m3)'], ax = ax[0], color = colors[j], label = region)
    ax[0].legend(bbox_to_anchor=(1, 1))
    ax[0].set_title("Średnie stężenie 'PM10 (\mu g/m3)' wg regionu na przestrzeni lat.")

    sns.lineplot(x = region_data.index, y = region_data['PM2.5 (\mu g/m3)'], ax = ax[1], color = colors[j], label = region)
    ax[1].legend(bbox_to_anchor=(1, 1))
    ax[1].set_title("Średnie stężenie 'PM2.5 (\mu g/m3)' wg regionu na przestrzeni lat.")

    sns.lineplot(x = region_data.index, y = region_data['NO2 (\mu g/m3)'], ax = ax[2], color = colors[j], label = region)
    ax[2].legend(bbox_to_anchor=(1, 1))
    ax[2].set_title("Średnie stężenie 'NO2 (\mu g/m3)' wg regionu na przestrzeni lat.")

plt.tight_layout()

```



- Patrząc na wykresy średnich stężeń pyłów PM2.5 i PM10 widać, że trendy dla wszystkich regionów są w ich przypadku bardzo zbliżone do siebie, co wynika z dużej korelacji między tymi zmiennymi.

- Największa zmienność w poszczególnych latach dla wszystkich zmiennych występuje w regionie Bliskiego Wschodu. Być może wynika to z małej ilości obserwacji dla tego regionu.
- W regionach Europy i Ameryki dla wszystkich zmiennych obserwowany jest delikatny trend spadkowy. W Europie może być to spowodowane naciskiem Unii Europejskiej na walkę z zanieczyszczeniem powietrza.
- Pomimo sporej rocznej liczby obserwacji, duże wahania średnich wartości występują dla regionu Azji Południowo-Wschodniej.
- Stężenia pyłów PM2.5 i PM10 w regionie Zachodniego Pacyfiku odnotowały znaczący wzrost w latach 2012-2015. Od tego czasu można zauważać nieznaczny trend spadkowy.
- Widoczny jest globalny trend spadkowy jeśli chodzi o średnie wartości stężenia NO2.

✓ Analiza dla krajów:

Kraje o najwyższych stężeniach zanieczyszczeń

1. PM10

```
stats_by_country = df_merged.groupby('Country').agg({'PM2.5 (\mu g/m³)': ['mean', 'median'],
                                                       'PM10 (\mu g/m³)': ['mean', 'median'],
                                                       'NO2 (\mu g/m³)': ['mean', 'median']})

pm25_stats = stats_by_country['PM2.5 (\mu g/m³)']['mean'].nlargest(10)
pm10_stats = stats_by_country['PM10 (\mu g/m³)']['mean'].nlargest(10)
no2_stats = stats_by_country['NO2 (\mu g/m³)']['mean'].nlargest(10)

stats_by_country.loc[pm10_stats.index, [('PM10 (\mu g/m³)', 'mean'), ('PM10 (\mu g/m³)', 'median')]]

# print("\nTop 5 kraje o najwyższych średnich stężeniach PM10:")
# print(stats_by_country.loc[pm10_stats.index, [('PM10 (\mu g/m³)', 'mean'), ('PM10 (\mu g/m³)', 'median'), ('PM10 (\mu g/m³)', 'std')]])

# print("\nTop 5 kraje o najwyższych średnich stężeniach NO2:")
# print(stats_by_country.loc[no2_stats.index, [('NO2 (\mu g/m³)', 'mean'), ('NO2 (\mu g/m³)', 'median'), ('NO2 (\mu g/m³)', 'std')]])
```

PM10 (\mu g/m³)		
	mean	median
Country		
Afghanistan	229.597301	229.597301
Egypt	227.000000	225.000000
Pakistan	207.750689	149.842785
Bahrain	174.508546	153.140000
Ghana	161.114701	176.400000
Qatar	159.333333	164.500000
Iraq	158.733281	179.380000
Mongolia	150.771779	138.500000
Bangladesh	138.434612	128.500000
Tajikistan	136.089147	136.089147

2. PM2.5

```
stats_by_country.loc[pm25_stats.index, [('PM2.5 (\mu g/m³)', 'mean'), ('PM2.5 (\mu g/m³)', 'median')]]
```

Country	PM2.5 ($\mu\text{g}/\text{m}^3$)	
	mean	median
Afghanistan	119.770000	119.770000
Egypt	99.508009	98.668433
Cameroon	82.666667	67.000000
Saudi Arabia	78.173343	72.000000
Bangladesh	75.127000	70.520000
Mongolia	73.865833	74.500000
Ghana	73.269154	78.266725
Iraq	72.490791	79.517694
Tajikistan	71.520000	71.520000
Pakistan	71.385000	67.050000

- Dla stężeń pyłów PM2.5 i PM10, 7 na 10 najbardziej zanieczyszczonych państw się powtarza - po raz kolejny widoczny jest związek tych 2 zmiennych.
- Cechy wspólne państw o największych stężeniach zanieczyszczeń PM2.5 oraz PM10:

- poza Arabią Saudyjską i Katarzem są to państwa biedne, rozwijające się
- są położone w Afryce lub Azji
- wszystkie poza Ghaną są krajami pustynnymi. Przypuszczalnie, pustynny klimat sprzyja unoszeniu się pyłów i może powodować:

3. NO2

```
stats_by_country.loc[no2_stats.index, [('NO2 ( $\mu\text{g}/\text{m}^3$ )', 'mean'), ('NO2 ( $\mu\text{g}/\text{m}^3$ )', 'median')]]
```

Country	NO2 ($\mu\text{g}/\text{m}^3$)	
	mean	median
Iran (Islamic Republic of)	69.025500	64.495
Lebanon	53.000000	53.000
Iraq	52.562857	50.230
Bahrain	51.625000	51.625
Kuwait	47.877037	47.170
Mongolia	46.400000	49.000
Costa Rica	40.710000	40.710
Qatar	39.333333	42.000
Republic of Korea	37.836343	35.720
China	36.521789	36.000

Kraje z najwyższymi średnimi wartościami stężenia tlenków azotu tworzą mieszankę. Są tu zarówno kraje biedne - Irak, Liban, bogate - kraje Zatoki Perskiej, wysoko rozwinięte Chiny i Korea Płd., a także niepasująca do nich Kostaryka.

Jako, że stężenie NO2 powiązane jest ze spalaniem paliw kopalnych - w dużej mierze w transporcie (również morskim), poszłąkami mogą być tutaj duża gęstość zaludnienia (Chiny, Korea), a także transport morski w zamkniętych lub półotwartych akwenach - Zatoka Perska, Morze Żółte wcinające się między Chiny a Półwysep Koreański jak Zatoka, Morze Karaibskie w przypadku Kostaryki oraz swego rodzaju cieśnina między Libanem a Cyprym. Ta odważna teza wymaga jednak dokładniejszego badania.

Kraje o najniższych stężeniach zanieczyszczeń

1. PM10

```
pm25_stats = stats_by_country['PM2.5 (\mu g/m3)']['mean'].nsmallest(10)
pm10_stats = stats_by_country['PM10 (\mu g/m3)']['mean'].nsmallest(10)
no2_stats = stats_by_country['NO2 (\mu g/m3)']['mean'].nsmallest(10)

stats_by_country.loc[pm10_stats.index, [('PM10 (\mu g/m3)', 'mean'), ('PM10 (\mu g/m3)', 'median')]]
```

PM10 ($\mu\text{g}/\text{m}^3$)		
	mean	median
Country		
Bahamas	5.150000	5.150000
Canada	10.743183	11.000000
Iceland	10.849085	7.420000
Estonia	11.389304	11.020000
Fiji	12.144816	12.144816
Finland	12.555731	12.180000
Ireland	14.125370	13.950000
Monaco	14.748000	14.850000
Norway	15.455238	16.040000
Sweden	15.681437	15.760000

2. PM2.5

```
stats_by_country.loc[pm25_stats.index, [('PM2.5 (\mu g/m3)', 'mean'), ('PM2.5 (\mu g/m3)', 'median')]]
```

PM2.5 ($\mu\text{g}/\text{m}^3$)		
	mean	median
Country		
Bahamas	4.645000	4.645000
Estonia	6.156352	5.750000
Canada	6.722726	6.300000
Iceland	6.785040	5.020000
Fiji	7.565000	7.565000
Finland	7.790597	8.128507
Norway	8.011449	7.850000
Kenya	8.350000	8.350000
Sweden	9.124910	9.816056
Ireland	9.319620	9.321650

- Jedyną różnicą w krajach o najniższych stężeniach pyłów PM2.5 oraz PM10 to Kenia i Monako.
- Wszystkie kraje mają dostęp do morza.
- Większość z nich stanowią wysoko rozwinięte kraje półkuli północnej, których rządy od wielu lat kładą duży nacisk na dbałość o środowisko naturalne. Wyjątek stanowią tu wyspiarskie Fidżi oraz Bahamy - kraje nieposiadające rozbudowanego przemysłu, opierające się na turystyce, a także Kenia.

3. NO2

```
stats_by_country.loc[no2_stats.index, [('NO2 (\mu g/m3)', 'mean'), ('NO2 (\mu g/m3)', 'median')]]
```

	NO2 ($\mu\text{g}/\text{m}^3$)	
	mean	median
Country		
Estonia	6.775179	5.610
Myanmar	6.815000	6.100
Iceland	7.896053	5.150
Trinidad and Tobago	9.600000	8.300
Australia	9.772121	9.055
Canada	11.251689	11.000
New Zealand	11.791818	2.920
Finland	13.281899	13.145
Malaysia	13.470625	12.175
Ireland	13.477049	9.630

W przypadku najniższych stężeń NO2 również dominują wysokorozwinięte kraje nadmorskie. Tak jak w przypadku PM2.5 oraz PM10 jest także wyspiarski przedstawiciel - Trynidad i Tobago. To co wymyka się prostej intuicji to obecność Malezji i Birmy, które pomimo dostępu do morza leżą w regionie narażonym na wysokie stężenia NO2 ze względu na swoich sąsiadów - Chiny oraz Indie.

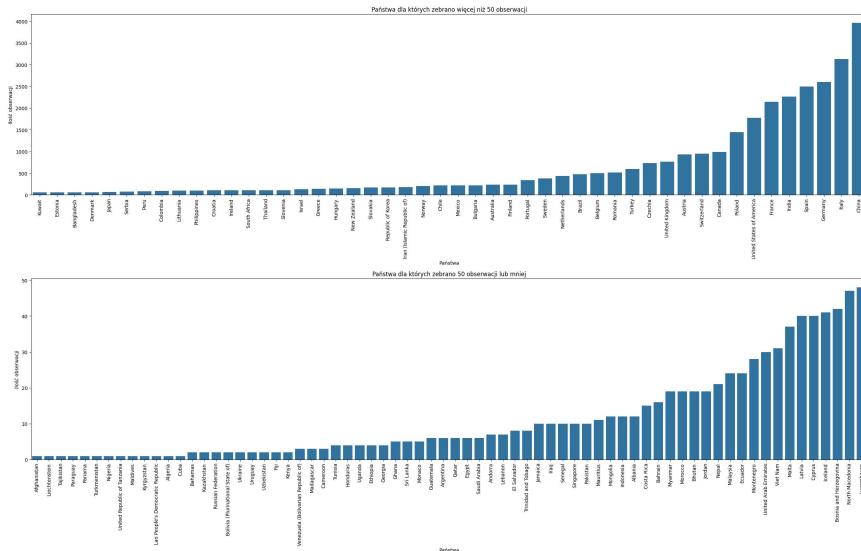
Ilość obserwacji wg kraju

```
fig, ax = plt.subplots(2, 1, figsize = [25, 16])
grouped_by_country = df_merged.groupby('Country').size().sort_values()

sns.barplot(grouped_by_country[grouped_by_country > 50],ax=ax[0])
ax[0].tick_params(axis='x', rotation=90)
ax[0].set_ylabel('Ilość obserwacji')
ax[0].set_xlabel('Państwa')
ax[0].set_title('Państwa dla których zebrano więcej niż 50 obserwacji')

sns.barplot(grouped_by_country[grouped_by_country <= 50],ax=ax[1])
ax[1].tick_params(axis='x', rotation=90)
ax[1].set_ylabel('Ilość obserwacji')
ax[1].set_xlabel('Państwa')
ax[1].set_title('Państwa dla których zebrano 50 obserwacji lub mniej')

plt.tight_layout()
```



```

print("Kraje o największej ilości obserwacji:\n")
print(df_merged.groupby('Country').size().nlargest(15))
print("\n\nKraje o ilości obserwacji równej 1:\n")
print(df_merged.groupby('Country').size().nsmallest(15))

```

Kraje o największej ilości obserwacji:

```
Country          |
China           3967|
Italy            3129|
Germany         2601|
Spain            2497|
India             2265|
France            2142|
United States of America 1776|
Poland            1448|
Canada             986|
Switzerland        951|
Austria            934|
United Kingdom      769|
Czechia            730|
Turkey              597|
Romania            513|
dtype: int64
```

Kraje o ilości obserwacji równej 1:

Country	
Afghanistan	1
Algeria	1
Cuba	1

```

Kyrgyzstan          1
Lao People's Democratic Republic    1
Liechtenstein       1
Maldives           1
Nigeria            1
Panama             1
Paraguay           1
Tajikistan          1
Turkmenistan        1
United Republic of Tanzania      1
Bahamas             2
Bolivia (Plurinational State of) 2
dtype: int64

```

- Widoczna jest bardzo duża dysproporcja w ilości danych zebranych w poszczególnych krajach.
- Aby kraje o najmniejszej ilości obserwacji były jakkolwiek widoczne w porównaniu z krajami o bardzo dużej ilości jak np. Chiny, konieczne było rozdzielenie wykresu na 2 części wg progu 50 obserwacji.
- Jeśli chodzi o kraje z największymi ilościami obserwacji, to są to największe pod względem populacji państwa świata - Chiny, Indie, a także duże kraje Europejskie, Stany i Kanada.
- Dalsze miejsca w tej klasyfikacji zajmują głównie kraje Europejskie lub kraje bardzo duże (populacyjnie i/lub rozmiarowo) jak Brazylia, Australia, Meksyk.
- Kraje o ilości obserwacji równej 1 (oznacza to, że dla 1 miasta zebrano roczną średnią zanieczyszczeń w 1 roku) to kraje globalnego południa - biedne, z wyjątkiem Lichtensteinu, który jest z kolei bardzo mały.
- Kraje o bardzo małej ilości obserwacji na tym poziomie generalizacji nie są reprezentatywne i mogłyby zostać usunięte ze zbioru. Jednak ze względu na to, że dane są analizowane również na wyższym poziomie generalizacji - dla regionów, obserwacje te mogą być przydatne.

Mapa świata obrazująca ilość obserwacji wg kraju

```

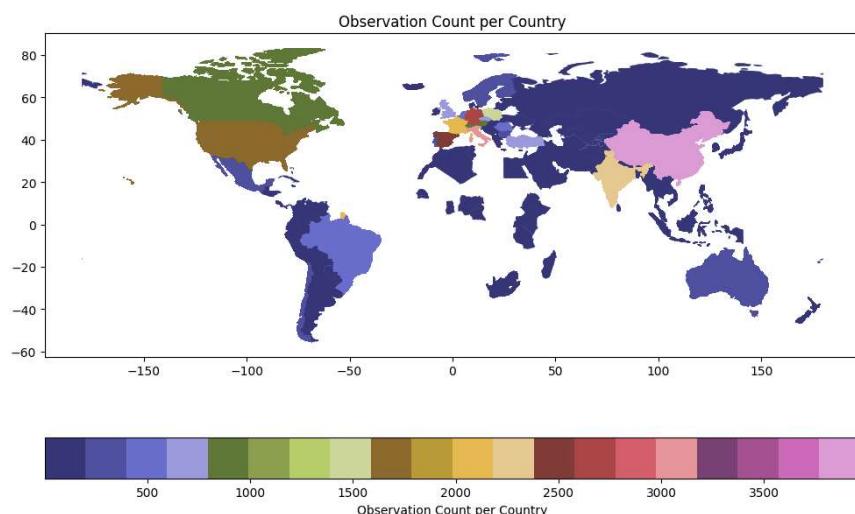
observation_count = df_merged['ISO3'].value_counts().reset_index()
observation_count.columns = ['ISO3', 'count']

# Merge observation count with world GeoDataFrame
df_merge_plot = df_merged.merge(observation_count, how='left', left_on='ISO3', right_on='ISO3')

# Plotting
df_merge_plot.plot(column='count', cmap='tab20b', legend=True,
                    legend_kwds={'label': "Observation Count per Country", 'orientation': "horizontal"}, figsize=(12, 8))

plt.title('Observation Count per Country')
plt.show()

```



Jak widać, kraje o bardzo małej ilości obserwacji to głównie państwa bliskowschodnie oraz afrykańskie, co pokrywa się z analizą dla regionów, gdzie regiony te miały najmniej obserwacji.

Oprócz tego, dzięki zmianie poziomu generalizacji można dostrzec, że niektóre kraje znajdujące się w liczniej reprezentowanych regionach również zawierają małą ilość obserwacji, np. państwa Azji Centralnej, Bałkany, czy spora część krajów Ameryki Południowej i Środkowej.

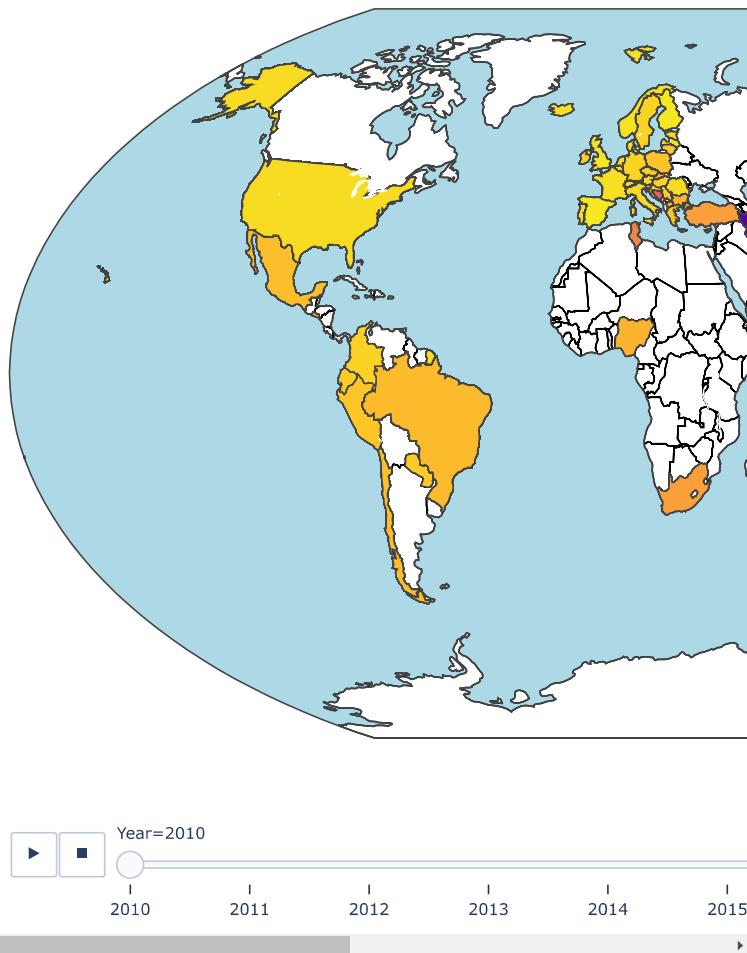
Nietypowa skala kolorów użyta do wykresu została dobrana, aby pokazać zmienność dla krajów o mniejszych ilościach obserwacji. Ze względu na wartości skrajne reprezentowane m.in. przez Chiny, używając skali o ciąglej zmienności kolorów, większość świata miała jeden odcień. Dzięki tej palecie udało się zmniejszyć ten efekt.

```
import plotly.express as px
df_plot = df_merged.sort_values(by='Year')
def plot_map(display_column, frame_column):
    fig = px.choropleth(
        df_plot,
        locations="ISO3",
        color=display_column,
        hover_name="Country",
        animation_frame=frame_column,
        color_continuous_scale=px.colors.sequential.Plasma[::-1],
        labels={display_column: f'{display_column} µg/m³'},
        range_color=[0, data[display_column].max()],
        title=f"Stężenie {display_column} wg kraju na przestrzeni lat",
        width = 1200,
        height = 800
    )
    fig.update_geos(
        projection_type="winkel tripel",
        showocean=True, oceancolor="LightBlue",
        showland=True, landcolor="white",
        showcountries=True, countrycolor="black",
    )
    fig.show()
```

Mapy prezentujące średnie stężenie zanieczyszczeń wg roku

1. PM10 ($\mu\text{g}/\text{m}^3$)

```
plot_map('PM10 (\mu\text{g}/\text{m}^3)', 'Year')
```

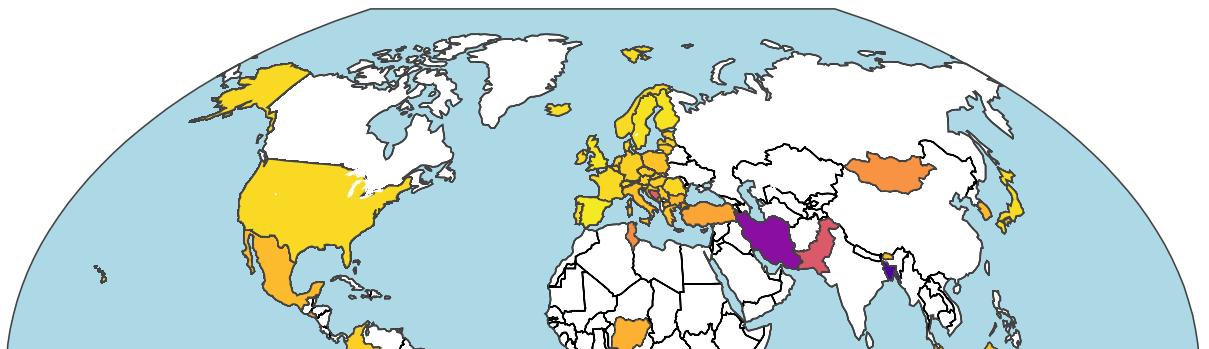
Stężenie PM10 ($\mu\text{g}/\text{m}^3$) wg kraju na przestrzeni lat

- Patrząc na animację zmian stężenia pyłów PM10 wg krajów, widoczny jest wzrost ilości krajów, dla których zbierane są dane o zanieczyszczeniach powietrza, co stanowi przesłankę, że zwiększa się dbałość państw o jakość powietrza
- Animacja dobrze obrazuje powolny progress jaki wykonują Chiny jeśli chodzi o stężenie pyłów PM10 w tym kraju

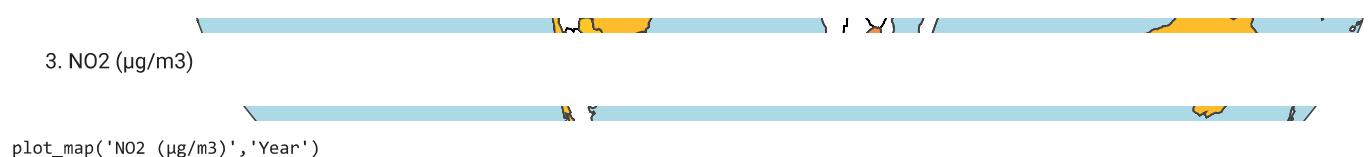
Aby edytować zawartość komórki, kliknij ją dwukrotnie (lub naciśnij klawisz Enter)

2. PM2.5 ($\mu\text{g}/\text{m}^3$)

```
plot_map('PM2.5 ( $\mu\text{g}/\text{m}^3$ )', 'Year')
```

Stężenie PM2.5 ($\mu\text{g}/\text{m}^3$) wg kraju na przestrzeni lat

- Trendy widoczne na animacji dla pyłów PM10 przenoszą się na animację PM2.5, ze względu na podobieństwo tych 2 zmiennych.
- Południowa Azja na przestrzeni lat wyraźnie dominuje jeśli chodzi o stężenie pyłów PM2.5



```
plot_map('NO2 ( $\mu\text{g}/\text{m}^3$ )', 'Year')
```

Stężenie NO2 ($\mu\text{g}/\text{m}^3$) wg kraju na przestrzeni lat