CS4132 Data Analytics

Effect of Air Quality and Pollution on Health Issues

by Teoh Yu Xin (M21405)

Table of Content

Motivation and background

Over the years, the environmental situation of Earth has been closely monitored and investigated closely. Air pollution, in particular, is an important environmental factor. Therefore, I would like to investigate how the relationship between different types of pollutants (i.e. PSI index, SO2, NO2, different factors affecting air quality) in relation to number of respiratory related diseases (tendency of increasing risk of respiratory diseases, life expectancy, getting lung cancer etc) based on the number of deaths or the number of lives lost as a result of these air pollution related causes, also known as DALYS (Disability adjusted life years).

Knowing the relationship between the heatlh of people with air pollution, and the extent to which air pollution relates to the health of people plays a significant role in coming up with solutions to improve the health of people.

Summary of research questions and Results

1. How does different kinds of air pollution affect the health of people in the region?
    - Different pollutant possibly have different effect on health, where PM2.5, PM10 show moderate positive correlation to both the average deaths and average dalys per 100 000 of the country population. Particularly, the effect of AQI of PM2.5 on the average deaths per country population shows a moderately strong linear positive correlation.
2. How does worsened air pollution affect the risk of different respiratory related diseases differently?
    - The AQI value of each country have varying effects on different death and dalys causes, some showing little to no relation, however some causes like air pollution, household pollution as death cause, and air pollution pollution as dalys cause show moderate positive correlation (though linear regression may not be the best model).
3. Does the effect of air pollution in a region affect the health of people of different genders differently?
    - The AQI value appears to affect the health of females slightly less than males, with females having a lower median average death value as compared to males for the same country. Females are less affected by air pollution than males.
4. How does the air quality effect on health of people vary from 2014 to 2017?
    - The extent to which health of peopple is correlated to AQI value is different for different years. For a smaller increase in AQI index, the extent to which the number of deaths increased in 2014 is the highest, followed by 2015, 2016 and 2017. For a smaller increase in AQI index, the extent to which the number of dalys increased in 2014 is the highest, followed by 2015, 2016 and 2017, which both happen to follow chronological order.
5. How does the air quality effect on health of people vary across different geographical locations?
    - Air pollution may have different effects the health of people in different countries around the world differently, where North and South America have relatively low index values for deaths/dalys over aqi value. Countries in Africa has a higher deaths/dalys over aqi value. Some outstanding countries like Australia, has the highest index for dalys of 1.49 (high number of dalys per 100 000 of country population to AQI index ratio), has a significantly high index (though slightly lower than that for dalys) of 1.04. Brunei (country code of BRN) has the highest index for deaths of 2.31, indicating it has a high number of deaths per 100 000 of country population to AQI index ratio.
6. How are individual factors varied over the years? (i.e. Air pollution and emissions, Health causes)

- Many individual factors have decreased over the years, which include average number of deaths/dalys and AQI value. Papau New Guinea has the highest average deaths and several African countries have high average dalys. There is a jump from 2016 to 2017 and from 2020 to 2021, with the increase from 2016 to 2017 being a sharper increase than from 2020 to 2021.

Dataset

This is the list of datasets used in this project, with dataset links found under References Section.

1. air_pollutant_co2.csv (Singapore Data for CO2 Emissions from fossil fuel combustion)
2. air_pollutant_lead.csv (Singaopre Data for Lead)
3. air_pollution_exposure.csv (amounnt of pm2.5 pollutants by country over the years)
4. aqi_breakpoints.csv (Standard table for AQI Breakpoints of different pollutants)
5. death-rates-from-air-pollution.csv (number of deaths per 100 000 based on different types of pollution)
6. disease-burden-by-risk-factor.csv (number of DALYS (disability adjusted life years) per 100 000 based on different types of pollution)
7. parameters.csv (standard units for AQI breakpoints)
8. pneumonia-death-rates-age-standardized.csv (number of deaths per 100 000 for lower respiratory infections)
9. respiratory-disease-death-rate.csv (number of deaths per 100 000 for chronic respiratory disease)
10. singstat_subcollation.csv (collation of pollutant emissions in Singapore)
11. stats_oecd_pollutants.csv (amount of pollutant emissions by year, by country)
12. who_respiratory_pollution_caused_rate.csv (number of attributed deaths per 100 000 due to different respiratory diseases)
13. waqi-covid19-airqualitydata-2015H1.csv (AQI Data from Quarter 1 of 2015)
14. waqi-covid19-airqualitydata-2016H1.csv (AQI Data from Quarter 1 of 2016)
15. waqi-covid19-airqualitydata-2017H1.csv (AQI Data from Quarter 1 of 2017)
16. waqi-covid19-airqualitydata-2018H1.csv (AQI Data from Quarter 1 of 2018)
17. waqi-covid19-airqualitydata-2019Q1.csv (AQI Data from Quarter 1 of 2019)
18. waqi-covid19-airqualitydata-2019Q2.csv (AQI Data from Quarter 2 of 2019)
19. waqi-covid19-airqualitydata-2019Q3.csv (AQI Data from Quarter 3 of 2019)
20. waqi-covid19-airqualitydata-2019Q1.csv (AQI Data from Quarter 4 of 2019)
21. waqi-covid19-airqualitydata-2020Q1.csv (AQI Data from Quarter 1 of 2020)
22. waqi-covid19-airqualitydata-2020Q2.csv (AQI Data from Quarter 2 of 2020)
23. waqi-covid19-airqualitydata-2020Q3.csv (AQI Data from Quarter 3 of 2020)
24. waqi-covid19-airqualitydata-2020Q1.csv (AQI Data from Quarter 4 of 2020)
25. waqi-covid19-airqualitydata-2021.csv (AQI Data from 2021)
26. https://en.wikipedia.org/wiki/List_of_countries_by_past_and_projected_future_population (population by country from 1950 to 2050, only used till 2020)
27. https://en.wikipedia.org/wiki/List_of_ISO_3166_country_codes (country codes corresponding to countries)

Methodology

Since the data required is not unique to each research question, data acquisition and data cleaning will be done for all datasets before data exploration and analysis pertaining to each research question.

Data Acquisition

Relevant Imports

In [1]:
```
import requests, pandas as pd, numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
import plotly.express as px
from scipy import stats
import matplotlib.patches as mpatches
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

Singapore Data

In [2]:
```
#singapore dataset links (dataset no. 10, 1, 2)
singapore_subcollation = "singstat_subcollation.csv"
singapore_lead = "air_pollutant_lead.csv"
singapore_co2 = "air_pollutant_co2.csv"
```

Global Data

In [3]:
```
#world dataset links (dataset no. 3, 11, 12)
world_oecd_pm25 = "air_pollution_exposure.csv"
world_oecd_pollutants = "stats_oecd_pollutants.csv"
who_attributeddeaths = "who_respiratory_pollution_caused_rate.csv"

#owid stands for our world in data (dataset no. 5, 6, 8, 9)
owid_death_rates_air_pollution = "death-rates-from-air-pollution.csv"
```

```
owid_dalys_air_pollution_risk = "disease-burden-by-risk-factor.csv"
owid_death_pneumonia = "pneumonia-death-rates-age-standardized.csv"
owid_death_rate_respiratory_disease = "respiratory-disease-death-rate.csv"

#world air quality index (dataset no. 13 - 25)
waqi_2015 = "waqi-covid19-airqualitydata-2015H1.csv"
waqi_2016 = "waqi-covid19-airqualitydata-2016H1.csv"
waqi_2017 = "waqi-covid19-airqualitydata-2017H1.csv"
waqi_2018 = "waqi-covid19-airqualitydata-2018H1.csv"
waqi_2019Q1 = "waqi-covid19-airqualitydata-2019Q1.csv"
waqi_2019Q2 = "waqi-covid19-airqualitydata-2019Q2.csv"
waqi_2019Q3 = "waqi-covid19-airqualitydata-2019Q3.csv"
waqi_2019Q4 = "waqi-covid19-airqualitydata-2019Q1.csv"
waqi_2020Q1 = "waqi-covid19-airqualitydata-2020Q1.csv"
waqi_2020Q2 = "waqi-covid19-airqualitydata-2020Q2.csv"
waqi_2020Q3 = "waqi-covid19-airqualitydata-2020Q3.csv"
waqi_2020Q4 = "waqi-covid19-airqualitydata-2020Q1.csv"
waqi_2021 = "waqi-covid19-airqualitydata-2021.csv"
```

Calculation Data

In [4]:

```
#global categorical data for calculation and categorising (datasets 4, 7, 26, 27)
aqi_breakpoints = "aqi_breakpoints.csv"
aqi_breakpoints_units = "parameters.csv"
country_population = "https://en.wikipedia.org/wiki/List_of_countries_by_past_and_projected_future_population"
country_code = "https://en.wikipedia.org/wiki/List_of_ISO_3166_country_codes"

aqi_breakpoints_data = pd.read_csv(aqi_breakpoints)
aqi_breakpoints_units_data = pd.read_csv(aqi_breakpoints_units)
country_population_url = requests.get(country_population)
country_population_data = pd.read_html(country_population_url.text)
country_population_data_1950_1980 = country_population_data[0]
country_population_data_1985_2015 = country_population_data[1]
country_population_data_2020_2050 = country_population_data[2]
country_code_url = requests.get(country_code)
country_code_data = pd.read_html(country_code_url.text)
country_code_data = country_code_data[0]
```

Processing and Reading Raw Data

In [5]:

```
#reading all of the datasets is performed
singapore_subcollation_data = pd.read_csv(singapore_subcollation,skiprows=4,skipfooter=25,engine='python')
singapore_lead_data = pd.read_csv(singapore_lead)
singapore_co2_data = pd.read_csv(singapore_co2)
world_oecd_pm25_data = pd.read_csv(world_oecd_pm25)
world_oecd_pollutants_data = pd.read_csv(world_oecd_pollutants,dtype={"Value":"float64"},low_memory=False)
who_attributeddeaths_data = pd.read_csv(who_attributeddeaths)
owid_death_rates_air_pollution_data = pd.read_csv(owid_death_rates_air_pollution)
owid_dalys_air_pollution_risk_data = pd.read_csv(owid_dalys_air_pollution_risk)
owid_death_pneumonia_data = pd.read_csv(owid_death_pneumonia)
owid_death_rate_respiratory_disease_data = pd.read_csv(owid_death_rate_respiratory_disease)

waqi_2015_data = pd.read_csv(waqi_2015,skiprows=4)
waqi_2016_data = pd.read_csv(waqi_2016,skiprows=4)
waqi_2017_data = pd.read_csv(waqi_2017,skiprows=4)
waqi_2018_data = pd.read_csv(waqi_2018,skiprows=4)
waqi_2019Q1_data = pd.read_csv(waqi_2019Q1,skiprows=4)
waqi_2019Q2_data = pd.read_csv(waqi_2019Q2,skiprows=4)
waqi_2019Q3_data = pd.read_csv(waqi_2019Q3,skiprows=4)
waqi_2019Q4_data = pd.read_csv(waqi_2019Q4,skiprows=4)
waqi_2020Q1_data = pd.read_csv(waqi_2020Q1,skiprows=4)
waqi_2020Q2_data = pd.read_csv(waqi_2020Q2,skiprows=4)
waqi_2020Q3_data = pd.read_csv(waqi_2020Q3,skiprows=4)
waqi_2020Q4_data = pd.read_csv(waqi_2020Q4,skiprows=4)
waqi_2021_data = pd.read_csv(waqi_2021,skiprows=4)
```

The next few cells are the printouts for all of the raw datasets.

In [6]:

```
singapore_subcollation_data.head()
```

Out[6]:

	Variables	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	Unnamed: 15
0	Sulphur Dioxide (Maximum 24-hour Mean) * (Mic...	84.0	80.0	93.0	104.0	80.0	98.0	75.0	83.0	75.0	61.0	59.0	65.0	57.0	30.0	NaN
1	Nitrogen Dioxide (Annual Mean) * (Microgram P...	22.0	22.0	22.0	23.0	25.0	25.0	25.0	24.0	22.0	26.0	25.0	26.0	23.0	20.0	NaN
2	Nitrogen Dioxide (Maximum 1-hour Mean) * (Mic...	177.0	126.0	147.0	153.0	189.0	154.0	132.0	121.0	99.0	123.0	158.0	147.0	156.0	118.0	NaN
3	Particulate Matter (PM10) (Annual Mean) * (Mi...	27.0	25.0	29.0	26.0	27.0	29.0	31.0	30.0	37.0	26.0	25.0	29.0	30.0	25.0	NaN
4	Particulate Matter (PM10) (99th Percentile 24...	53.0	49.0	59.0	76.0	55.0	57.0	215.0	75.0	186.0	61.0	57.0	59.0	90.0	43.0	NaN

In [7]:

```
singapore_lead_data.head()
```

Out[7]:

	year	lead_mean
0	2006	<0.1
1	2007	<0.1
2	2008	<0.1
3	2009	<0.1
4	2010	<0.1

In [8]:
```
singapore_co2_data.head()
#this one is from fossil fuel combustion
```

Out[8]:

	year	co2_emissions
0	2007	39905
1	2008	38524
2	2009	39465
3	2010	43122
4	2011	45281

In [9]:
```
world_oecd_pm25_data.head()
```

Out[9]:

	LOCATION	INDICATOR	SUBJECT	MEASURE	FREQUENCY	TIME	Value	Flag Codes
0	AUS	POLLUTIONEXP	EXPOS2PM25	MICGRCUBM	A	1990	7.60250	NaN
1	AUS	POLLUTIONEXP	EXPOS2PM25	MICGRCUBM	A	1995	7.49591	NaN
2	AUS	POLLUTIONEXP	EXPOS2PM25	MICGRCUBM	A	2000	7.36613	NaN
3	AUS	POLLUTIONEXP	EXPOS2PM25	MICGRCUBM	A	2005	6.90976	NaN
4	AUS	POLLUTIONEXP	EXPOS2PM25	MICGRCUBM	A	2010	6.78718	NaN

In [10]:
```
world_oecd_pollutants_data.head()
```

Out[10]:

	COU	Country	POL	Pollutant	VAR	Variable	YEA	Year	Unit Code	Unit	PowerCode Code	PowerCode	Reference Period Code	Reference Period	Value	Flag Codes	Flags
0	AUS	Australia	SOX	Sulphur Oxides	TOT	Total man-made emissions	1990	1990	TONNE	Tonnes	3	Thousands	NaN	NaN	1585.754	NaN	NaN
1	AUS	Australia	SOX	Sulphur Oxides	TOT	Total man-made emissions	1991	1991	TONNE	Tonnes	3	Thousands	NaN	NaN	1570.777	NaN	NaN
2	AUS	Australia	SOX	Sulphur Oxides	TOT	Total man-made emissions	1992	1992	TONNE	Tonnes	3	Thousands	NaN	NaN	1652.946	NaN	NaN
3	AUS	Australia	SOX	Sulphur Oxides	TOT	Total man-made emissions	1993	1993	TONNE	Tonnes	3	Thousands	NaN	NaN	1743.161	NaN	NaN
4	AUS	Australia	SOX	Sulphur Oxides	TOT	Total man-made emissions	1994	1994	TONNE	Tonnes	3	Thousands	NaN	NaN	1764.906	NaN	NaN

In [11]:
```
who_attributeddeaths_data.head()
```

Out[11]:

	IndicatorCode	Indicator	ValueType	ParentLocationCode	ParentLocation	Location type	SpatialDimValueCode	Location	Period type	Period	...	FactValueUoM
0	AIR_5	Ambient air pollution attributable death rate ...	numeric	AFR	Africa	Country	MWI	Malawi	Year	2016	...	NaN
1	AIR_5	Ambient air pollution attributable death rate ...	numeric	AFR	Africa	Country	NER	Niger	Year	2016	...	NaN

	IndicatorCode	Indicator	ValueType	ParentLocationCode	ParentLocation	Location type	SpatialDimValueCode	Location	Period type	Period	...	FactValueUoM
2	AIR_5	Ambient air pollution attributable death rate ...	numeric	AFR	Africa	Country	TZA	United Republic of Tanzania	Year	2016	...	NaN
3	AIR_5	Ambient air pollution attributable death rate ...	numeric	AFR	Africa	Country	TZA	United Republic of Tanzania	Year	2016	...	NaN
4	AIR_5	Ambient air pollution attributable death rate ...	numeric	AFR	Africa	Country	MWI	Malawi	Year	2016	...	NaN

5 rows × 34 columns

In [12]:
```
owid_death_rates_air_pollution_data.head()
```

Out[12]:

	Entity	Code	Year	Deaths - Air pollution - Sex: Both - Age: Age-standardized (Rate)	Deaths - Household air pollution from solid fuels - Sex: Both - Age: Age-standardized (Rate)	Deaths - Ambient particulate matter pollution - Sex: Both - Age: Age-standardized (Rate)	Deaths - Ambient ozone pollution - Sex: Both - Age: Age-standardized (Rate)
0	Afghanistan	AFG	1990	299.477309	250.362910	46.446589	5.616442
1	Afghanistan	AFG	1991	291.277967	242.575125	46.033841	5.603960
2	Afghanistan	AFG	1992	278.963056	232.043878	44.243766	5.611822
3	Afghanistan	AFG	1993	278.790815	231.648134	44.440148	5.655266
4	Afghanistan	AFG	1994	287.162923	238.837177	45.594328	5.718922

In [13]:
```
owid_dalys_air_pollution_risk_data.head()
```

Out[13]:

	Entity	Code	Year	DALYs (Disability-Adjusted Life Years) - Air pollution - Sex: Both - Age: All Ages (Number)	DALYs (Disability-Adjusted Life Years) - Child wasting - Sex: Both - Age: All Ages (Number)	DALYs (Disability-Adjusted Life Years) - Child stunting - Sex: Both - Age: All Ages (Number)	DALYs (Disability-Adjusted Life Years) - Secondhand smoke - Sex: Both - Age: All Ages (Number)	DALYs (Disability-Adjusted Life Years) - Unsafe sanitation - Sex: Both - Age: All Ages (Number)	DALYs (Disability-Adjusted Life Years) - Unsafe water source - Sex: Both - Age: All Ages (Number)	DALYs (Disability-Adjusted Life Years) - Low physical activity - Sex: Both - Age: All Ages (Number)	...	DALYs (Disability-Adjusted Life Years) - Diet low in fruits - Sex: Both - Age: All Ages (Number)	A A
0	Afghanistan	AFG	1990	1.396895e+06	1.986646e+06	9.051669e+05	211581.261894	521511.931474	6.691367e+05	98784.241096	...	244040.521758	7
1	Afghanistan	AFG	1991	1.372209e+06	1.945396e+06	8.934228e+05	208935.686668	508979.975154	6.534517e+05	99448.821844	...	246864.964365	7
2	Afghanistan	AFG	1992	1.427254e+06	2.020055e+06	9.246678e+05	218153.716392	531882.119225	6.833615e+05	102373.508494	...	258888.740805	9
3	Afghanistan	AFG	1993	1.684234e+06	2.443222e+06	1.069002e+06	255940.120953	711012.533845	9.143711e+05	106006.567727	...	274598.209012	11
4	Afghanistan	AFG	1994	1.906674e+06	2.883149e+06	1.238041e+06	288758.236266	790743.749812	1.017927e+06	108725.626545	...	285333.186410	12

5 rows × 25 columns

In [14]:
```
owid_death_pneumonia_data.head()
```

Out[14]:

	Entity	Code	Year	Deaths - Lower respiratory infections - Sex: Both - Age: Age-standardized (Rate)
0	Afghanistan	AFG	1990	164.811829
1	Afghanistan	AFG	1991	151.460290
2	Afghanistan	AFG	1992	127.896225
3	Afghanistan	AFG	1993	124.725141
4	Afghanistan	AFG	1994	134.410918

In [15]:
```
owid_death_rate_respiratory_disease_data.head()
```

Out[15]:

	Entity	Code	Year	Deaths - Chronic respiratory diseases - Sex: Both - Age: Age-standardized (Rate)
0	Afghanistan	AFG	1990	95.273780
1	Afghanistan	AFG	1991	95.270656

	Entity	Code	Year	Deaths - Chronic respiratory diseases - Sex: Both - Age: Age-standardized (Rate)
2	Afghanistan	AFG	1992	95.584266
3	Afghanistan	AFG	1993	96.581362
4	Afghanistan	AFG	1994	98.105844

In [16]:
```
aqi_breakpoints_data.head()
```

Out[16]:

	Parameter	Parameter Code	Duration Code	Duration Description	AQI Category	Low AQI	High AQI	Low Breakpoint	High Breakpoint
0	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	GOOD	0	50	0.0	12.0
1	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	MODERATE	51	100	12.1	35.4
2	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	UNHEALTHY FOR SENSITIVE	101	150	35.5	55.4
3	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	UNHEALTHY	151	200	55.5	150.4
4	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	VERY UNHEALTHY	201	300	150.5	250.4

In [17]:
```
aqi_breakpoints_units_data.head()
```

Out[17]:

	Parameter Code	Parameter	Parameter Abbreviation	Parameter Alternate Name	CAS Number	Standard Units	Still Valid	Round or Truncate
0	43834	1,1,1,2,2-Pentafluoroethane	NaN	HFC-125	354-33-6	Parts per billion Carbon	YES	R
1	43837	1,1,1,2-Tetrachloroethane	NaN	NaN	630-20-6	Parts per billion Carbon	YES	R
2	43162	1,1,1-Trichloro-2,2-bis (p-chlorophenyl) ethane	TRICH	NaN	50-29-3	Nanograms/cubic meter (25 C)	YES	R
3	43818	1,1,2,2-Tetrachloroethane	4CLET	Ethane,1,1,2,2-tetrachloro-	79-34-5	Parts per billion Carbon	YES	R
4	43821	1,1,2-Trichloro-1,2,2-trifluoroethane	NaN	NaN	76-13-1	Parts per billion Carbon	YES	R

In [18]:
```
country_population_data_1950_1980.head()
```

Out[18]:

	Country (or dependent territory)	1950	1955	%	1960	%.1	1965	%.2	1970	%.3	1975	%.4	1980	%.5
0	Afghanistan	8151	8892	1.76	9830	2.03	10998	2.27	12431	2.48	14133	2.60	15045	1.26
1	Albania	1228	1393	2.56	1624	3.12	1884	3.02	2157	2.74	2402	2.17	2672	2.16
2	Algeria	8893	9842	2.05	10910	2.08	11964	1.86	13932	3.09	16141	2.99	18807	3.10
3	American Samoa	20	20	0.72	21	0.20	25	4.23	28	2.08	30	1.68	33	1.81
4	Andorra	7	7	0.04	9	6.28	14	10.17	20	7.49	27	6.32	34	4.81

In [19]:
```
country_code_data.head()
```

Out[19]:

	ISO 3166[1]	Unnamed: 1_level_0	Unnamed: 2_level_0			ISO 3166-1[2]	ISO 3166-2[3]	Unnamed: 7_level_0
	Country name[5]	Official state name[6]	Sovereignty[6][7][8]	Alpha-2 code[5]	Alpha-3 code[5]	Numeric code[5]	Subdivision code links[3]	Internet ccTLD[9]
0	Afghanistan	The Islamic Republic of Afghanistan	UN member state	.mw-parser-output .monospaced{font-family:mono...	AFG	004	ISO 3166-2:AF	.af
1	Akrotiri and Dhekelia – See United Kingdom, The	Akrotiri and Dhekelia – See United Kingdom, The	Akrotiri and Dhekelia – See United Kingdom, The	Akrotiri and Dhekelia – See United Kingdom, The	Akrotiri and Dhekelia – See United Kingdom, The	Akrotiri and Dhekelia – See United Kingdom, The	Akrotiri and Dhekelia – See United Kingdom, The	Akrotiri and Dhekelia – See United Kingdom, The
2	Åland Islands	Åland	Finland	AX	ALA	248	ISO 3166-2:AX	.ax
3	Albania	The Republic of Albania	UN member state	AL	ALB	008	ISO 3166-2:AL	.al
4	Algeria	The People's Democratic Republic of Algeria	UN member state	DZ	DZA	012	ISO 3166-2:DZ	.dz

In [20]:
```
waqi_2015_data.head()
```

Out[20]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2015-01-06	KR	Jeonju	co	124	0.1	12.3	4.5	55.74
1	2015-01-22	KR	Jeonju	co	116	4.5	10.0	6.7	16.09
2	2015-03-30	KR	Jeonju	co	118	1.2	11.2	5.6	35.98
3	2015-05-27	KR	Jeonju	co	93	2.3	5.6	3.4	6.54
4	2015-02-03	KR	Jeonju	co	133	4.5	13.4	7.8	39.24

In [21]:
```
waqi_2016_data.head()
```

Out[21]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2016-02-03	SE	Stockholm	o3	24	15.0	23.1	17.6	68.93
1	2016-02-04	SE	Stockholm	o3	24	9.2	25.7	17.8	252.52
2	2016-02-16	SE	Stockholm	o3	24	3.8	21.1	15.4	193.79
3	2016-03-11	SE	Stockholm	o3	24	1.0	29.2	15.6	1085.12
4	2016-04-02	SE	Stockholm	o3	24	22.8	27.0	25.6	16.40

In [22]:
```
waqi_2017_data.head()
```

Out[22]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2016-12-27	CN	Beijing	co	420	1.0	34.4	7.3	217.26
1	2017-01-28	CN	Beijing	co	434	1.9	49.7	23.5	508.57
2	2017-02-24	CN	Beijing	co	427	1.0	37.1	6.4	374.07
3	2017-03-18	CN	Beijing	co	425	1.0	29.0	11.8	334.03
4	2017-04-16	CN	Beijing	co	407	1.0	17.2	9.1	126.15

In [23]:
```
waqi_2018_data.head()
```

Out[23]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2018-04-19	HR	Zagreb	pm10	72	12.0	66.0	19.0	1034.64
1	2018-05-03	HR	Zagreb	pm10	72	5.0	46.0	20.0	740.53
2	2018-05-08	HR	Zagreb	pm10	69	7.0	33.0	17.0	286.35
3	2018-05-31	HR	Zagreb	pm10	48	15.0	60.0	25.0	704.61
4	2018-06-22	HR	Zagreb	pm10	62	1.0	60.0	7.0	670.06

In [24]:
```
waqi_2019Q1_data.head()
```

Out[24]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2019-01-16	AE	Abu Dhabi	pm10	24	86.0	99.0	97.0	179.40
1	2019-01-22	AE	Abu Dhabi	pm10	24	51.0	57.0	55.0	23.75
2	2019-01-26	AE	Abu Dhabi	pm10	24	136.0	173.0	160.0	941.96
3	2019-01-07	AE	Abu Dhabi	pm10	24	60.0	91.0	72.0	1006.88
4	2019-01-10	AE	Abu Dhabi	pm10	24	82.0	93.0	87.0	57.97

In [25]:
```
waqi_2019Q2_data.head()
```

Out[25]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2019-05-10	MO	Macau	o3	117	1.4	42.9	15.7	1129.11
1	2019-05-14	MO	Macau	o3	114	1.5	39.4	10.1	734.48
2	2019-05-15	MO	Macau	o3	117	1.7	24.1	9.8	367.55
3	2019-05-22	MO	Macau	o3	118	10.3	70.9	36.0	1859.40
4	2019-05-29	MO	Macau	o3	114	1.2	42.0	18.8	1410.13

In [26]:
```
waqi_2019Q3_data.head()
```

Out[26]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2019-08-20	SK	Košice	dew	20	15.0	18.5	17.0	9.94

	Date	Country	City	Specie	count	min	max	median	variance
1	2019-08-22	SK	Košice	dew	19	11.0	14.5	13.0	11.70
2	2019-09-20	SK	Košice	dew	24	-0.5	4.5	2.5	21.52
3	2019-09-27	SK	Košice	dew	23	8.0	12.5	11.5	28.22
4	2019-10-01	SK	Košice	dew	72	4.5	8.0	7.0	12.84

In [27]:
```
waqi_2019Q4_data.head()
```

Out[27]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2019-01-16	AE	Abu Dhabi	pm10	24	86.0	99.0	97.0	179.40
1	2019-01-22	AE	Abu Dhabi	pm10	24	51.0	57.0	55.0	23.75
2	2019-01-26	AE	Abu Dhabi	pm10	24	136.0	173.0	160.0	941.96
3	2019-01-07	AE	Abu Dhabi	pm10	24	60.0	91.0	72.0	1006.88
4	2019-01-10	AE	Abu Dhabi	pm10	24	82.0	93.0	87.0	57.97

In [28]:
```
waqi_2020Q1_data.head()
```

Out[28]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2020-01-13	CO	Bogotá	so2	316	0.2	2.8	0.5	1.47
1	2020-02-25	CO	Bogotá	so2	335	0.1	4.1	0.8	3.25
2	2020-03-09	CO	Bogotá	so2	308	0.2	5.4	0.6	3.51
3	2020-03-13	CO	Bogotá	so2	333	0.2	3.7	0.8	3.45
4	2020-03-27	CO	Bogotá	so2	323	0.2	13.8	0.7	27.21

In [29]:
```
waqi_2020Q2_data.head()
```

Out[29]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2020-04-15	IN	Thrissur	wind-gust	3	10.2	10.2	10.2	0.0
1	2020-04-26	IN	Thrissur	wind-gust	3	12.8	12.8	12.8	0.0
2	2020-05-02	IN	Thrissur	wind-gust	3	8.7	8.7	8.7	0.0
3	2020-05-11	IN	Thrissur	wind-gust	3	11.3	11.3	11.3	0.0
4	2020-06-21	IN	Thrissur	wind-gust	3	10.2	10.2	10.2	0.0

In [30]:
```
waqi_2020Q3_data.head()
```

Out[30]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2020-08-16	UA	Odessa	pm10	23	3.0	13.0	7.0	76.13
1	2020-08-18	UA	Odessa	pm10	21	3.0	9.0	5.0	31.48
2	2020-08-24	UA	Odessa	pm10	7	3.0	8.0	5.0	36.19
3	2020-08-25	UA	Odessa	pm10	21	4.0	12.0	6.0	42.14
4	2020-08-26	UA	Odessa	pm10	24	3.0	7.0	5.0	16.07

In [31]:
```
waqi_2020Q4_data.head()
```

Out[31]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2020-01-13	CO	Bogotá	so2	316	0.2	2.8	0.5	1.47
1	2020-02-25	CO	Bogotá	so2	335	0.1	4.1	0.8	3.25
2	2020-03-09	CO	Bogotá	so2	308	0.2	5.4	0.6	3.51
3	2020-03-13	CO	Bogotá	so2	333	0.2	3.7	0.8	3.45
4	2020-03-27	CO	Bogotá	so2	323	0.2	13.8	0.7	27.21

In [32]:
```
waqi_2021_data.head()
```

Out[32]:

	Date	Country	City	Specie	count	min	max	median	variance
0	2021-06-21	EC	Quito	so2	112	0.9	27.3	4.3	84.02
1	2021-04-14	EC	Quito	so2	187	0.3	10.3	3.3	32.19

	Date	Country	City	Specie	count	min	max	median	variance
2	2021-05-07	EC	Quito	so2	132	0.7	49.1	3.8	410.57
3	2021-03-19	EC	Quito	so2	188	0.5	17.1	3.9	49.32
4	2021-09-10	EC	Quito	so2	119	1.0	21.5	4.7	156.31

Data Cleaning

singapore_lead_data is obtained from air_pollutant_lead.csv, which is exported from **Singapore Lead Data**.

singapore_co2_data is obtained from air_pollutant_co2.csv, which is exported from **Singapore CO2 Data**.

singapore_subcollation_data is obtained from air_pollutant_exposure.csv, which contains a collation of other pollutant values, which is exported from **Singapore Subcollation Data**. (Side note: this link does not work if clicked on directly, will have to paste the actual link to view the dataset "https://www.tablebuilder.singstat.gov.sg/publicfacing/createDataTable.action?refId=14589".

singapore_co2_data and singapore_lead_data are formatted to match formatting of singapore_subcollation_data to combine into singapore_collated_data which contains all of the values from the various pollutants. Several renaming, dropping of columns and converting of column types are done to make the naming concise and consitent.

In [33]:
```
#preliminary combination of Singapore data
#formatting lead_data to subcollation_data format
singapore_lead_data = singapore_lead_data.rename(columns={"year":"Variables","lead_mean":"LEAD (MICGRCUBM)"})
singapore_lead_data.index = singapore_lead_data["Variables"]
singapore_lead_data.drop(columns=["Variables"],inplace=True)
singapore_lead_data = singapore_lead_data.T
singapore_lead_data.columns

#dropping, striping column strings, change to int
singapore_subcollation_data.drop(columns=["Unnamed: 15"], inplace=True)
singapore_subcollation_data = singapore_subcollation_data.rename(columns={" Variables ":"Variables"})
singapore_subcollation_data.index = singapore_subcollation_data["Variables"]
singapore_subcollation_data.drop(columns=["Variables"], inplace=True)
singapore_subcollation_data.columns = np.arange(2007,2021)

#renaming of columns for easier reference and consistency with other datasets
singapore_co2_data = singapore_co2_data.rename(columns={"year":"Variables","co2_emissions":"CO2 (FOSSIL FUEL, KILOTON)"})
singapore_co2_data.index = singapore_co2_data["Variables"]
singapore_co2_data.drop(columns=["Variables"],inplace=True)
singapore_co2_data = singapore_co2_data.T
singapore_subcollation_data = singapore_subcollation_data.rename(
    index={" Sulphur Dioxide (Maximum 24-hour Mean) * (Microgram Per Cubic Metre) ": "SOX (24H,MICGRCUBM)",
           " Nitrogen Dioxide (Annual Mean) * (Microgram Per Cubic Metre) ": "NOX (YR,MICGRCUBM)",
           " Nitrogen Dioxide (Maximum 1-hour Mean) * (Microgram Per Cubic Metre) ":"NOX (1H,MICRCUBM)",
           " Particulate Matter (PM10) (Annual Mean) * (Microgram Per Cubic Metre) ":"PM10 (YR,MICGRCUBM)",
           " Particulate Matter (PM10) (99th Percentile 24-hour Mean) * (Microgram Per Cubic Metre) ":"PM10 (24H,MICGRCUBM)",
           " Particulate Matter (PM2.5) (Annual Mean) * (Microgram Per Cubic Metre) ":"PM2.5 (YR,MICGRCUBM)",
           " Particulate Matter (PM2.5) (99th Percentile 24-hour Mean) * (Microgram Per Cubic Metre) ":"PM2.5 (24H,MICGRCUBM)",
           " Carbon Monoxide (Maximum 8-hour Mean) * (Milligram Per Cubic Metre) ":"CO (8H,MILGRCUBM)",
           " Carbon Monoxide (Maximum 1-hour Mean) * (Milligram Per Cubic Metre) ":"CO (1H,MILGRCUBM)",
           " Ozone (Maximum 8-hour Mean) * (Microgram Per Cubic Metre) ":"O3 (8H,MICGRCUBM)"})

singapore_collated_data = singapore_subcollation_data.append([singapore_lead_data,singapore_co2_data])
singapore_collated_data = singapore_collated_data.dropna(how="any",thresh=5,axis=1)
singapore_collated_data.head()
```

Out[33]:

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
SOX (24H,MICGRCUBM)	84.0	80.0	93.0	104.0	80.0	98.0	75.0	83.0	75.0	61.0	59.0	65.0	57.0	30.0
NOX (YR,MICGRCUBM)	22.0	22.0	22.0	23.0	25.0	25.0	25.0	24.0	22.0	26.0	25.0	26.0	23.0	20.0
NOX (1H,MICRCUBM)	177.0	126.0	147.0	153.0	189.0	154.0	132.0	121.0	99.0	123.0	158.0	147.0	156.0	118.0
PM10 (YR,MICGRCUBM)	27.0	25.0	29.0	26.0	27.0	29.0	31.0	30.0	37.0	26.0	25.0	29.0	30.0	25.0
PM10 (24H,MICGRCUBM)	53.0	49.0	59.0	76.0	55.0	57.0	215.0	75.0	186.0	61.0	57.0	59.0	90.0	43.0

In [34]:
```
#data cleaning of world_oecd_pm25_data
world_oecd_pm25_data.drop(columns=["INDICATOR","SUBJECT","MEASURE","FREQUENCY","Flag Codes"],inplace=True)
world_oecd_pm25_data = world_oecd_pm25_data.rename(columns={"Value": "PM2.5 (MICGRCUBM)","LOCATION":"COUNTRY","TIME":"YEAR"})
world_oecd_pm25_data.head()
```

Out[34]:

	COUNTRY	YEAR	PM2.5 (MICGRCUBM)
0	AUS	1990	7.60250
1	AUS	1995	7.49591
2	AUS	2000	7.36613
3	AUS	2005	6.90976
4	AUS	2010	6.78718

In [35]:
```
#pivot table for subsequent use in later parts of the project
```

```
world_oecd_pm25_data_year_column = pd.pivot_table(world_oecd_pm25_data,index=["COUNTRY"],values=["PM2.5 (MICGRCUBM)"],columns=["YE
world_oecd_pm25_data_year_column.head()
```

Out[35]:

COUNTRY	PM2.5 (MI												
YEAR	1990	1995	2000	2005	2010	2011	2012	2013	2014	2015	2016	2017	2018
AFG	74.672740	74.294640	76.117000	73.301755	76.247925	78.549860	77.733055	79.811385	81.360960	80.705785	78.983170	77.015585	76.716105
AGO	64.937710	64.181785	64.069875	64.253540	64.477005	64.523585	64.636180	64.753190	64.234300	64.634435	64.628930	64.022560	64.154935
ALB	62.253760	61.912895	62.083730	60.988540	60.772870	61.651165	60.722460	59.963815	59.902945	59.706855	58.839635	59.428940	59.324230
ARE	70.446650	70.638535	70.840765	70.549345	70.926015	72.164580	72.941715	71.343930	70.016630	72.246415	71.047655	71.166765	71.256645
ARG	57.563955	57.542040	56.887270	52.780560	55.709350	56.593930	54.167495	57.224095	56.662425	57.157390	56.771105	56.251870	56.268085

In [36]:

```
#data cleaning of world_oecd_pollutants
world_oecd_pollutants_data.drop(columns=["Country","Pollutant","Variable","Year","Unit","Reference Period Code",
                                         "Reference Period","PowerCode Code","Flag Codes","Flags"],inplace=True)
world_oecd_pollutants_data = world_oecd_pollutants_data.rename(
    columns={"COU": "COUNTRY","POL":"POLLUTANT","YEA":"YEAR","Value":"VALUE"})
world_oecd_pollutants_data["Unit Code"] = "KILOTON"
world_oecd_pollutants_data.drop(columns=["PowerCode"],inplace=True)
world_oecd_pollutants_data = world_oecd_pollutants_data[world_oecd_pollutants_data["COUNTRY"].str.len()==3]
world_oecd_pollutants_data_grouped = world_oecd_pollutants_data.groupby(by=["COUNTRY","POLLUTANT"])[["VALUE"]].sum()
world_oecd_pollutants_data_grouped = pd.pivot_table(world_oecd_pollutants_data,index=["COUNTRY","POLLUTANT"],
                                                    values=["VALUE"],columns=["YEAR"],aggfunc="sum")
world_oecd_pollutants_data.head()
```

Out[36]:

	COUNTRY	POLLUTANT	VAR	YEAR	Unit Code	VALUE
0	AUS	SOX	TOT	1990	KILOTON	1585.754
1	AUS	SOX	TOT	1991	KILOTON	1570.777
2	AUS	SOX	TOT	1992	KILOTON	1652.946
3	AUS	SOX	TOT	1993	KILOTON	1743.161
4	AUS	SOX	TOT	1994	KILOTON	1764.906

In [37]:

```
#dataframe is pivoted to display the AQI value over the years as columns
world_oecd_pollutants_data_grouped.head()
#value in thousands
```

Out[37]:

	YEAR	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	...	2009	
COUNTRY	POLLUTANT													
AUS	**CO**	18956.307	19031.699	19516.703	20222.270	20642.617	20132.705	20647.341	20673.745	19515.949	19040.313	...	10566.750	1004
	NMVOC	4665.703	4620.486	4631.471	4688.312	4723.063	4668.638	4584.423	4567.901	4490.598	4371.424	...	4135.876	414
	NOX	5751.195	5662.502	5804.043	5952.182	5964.240	5994.335	6172.417	6371.196	6492.953	6607.677	...	8021.174	832
	SOX	5151.136	5096.456	5352.882	5634.453	5698.658	5409.994	5681.019	5864.542	5622.729	5906.158	...	8304.664	761
AUT	**CO**	4803.458	4836.337	4647.530	4431.931	4188.424	3782.165	3807.266	3523.292	3338.441	2869.847	...	2264.266	233

5 rows × 29 columns

In [38]:

```
#data cleaning of who_attributeddeaths_data
who_attributeddeaths_data.drop(columns=["IndicatorCode","Indicator","ValueType","ParentLocation","Location type",
                                        "Location","Period type","IsLatestYear","Dim1 type","Dim1","Dim2 type",
                                         "Dim2ValueCode","Dim3 type","Dim3","Dim3ValueCode","DataSourceDimValueCode",
                                         "DataSource","FactValueUoM","FactValueNumericLowPrefix","FactValueNumericPrefix",
                                        "FactValueNumericHighPrefix","Value","FactValueTranslationID","FactComments","Language",
                                        "DateModified"],inplace=True)
who_attributeddeaths_data = who_attributeddeaths_data.rename(
    columns={"ParentLocationCode":"REGION","SpatialDimValueCode":"COUNTRY","Period":"YEAR","FactValueNumericLow":"MIN",
             "FactValueNumericHigh":"MAX","FactValueNumeric":"VALUE","Dim1ValueCode":"SEX","Dim2":"CAUSE"})
who_attributeddeaths_data.drop(columns="COUNTRY")
who_attributeddeaths_data.head()
```

Out[38]:

	REGION	COUNTRY	YEAR	SEX	CAUSE	VALUE	MIN	MAX
0	AFR	MWI	2016	FMLE	Trachea, bronchus, lung cancers	0.040	0.022	0.061
1	AFR	NER	2016	FMLE	Trachea, bronchus, lung cancers	0.047	0.032	0.062
2	AFR	TZA	2016	FMLE	Trachea, bronchus, lung cancers	0.051	0.029	0.076
3	AFR	TZA	2016	BTSX	Trachea, bronchus, lung cancers	0.055	0.031	0.083

	REGION	COUNTRY	YEAR	SEX	CAUSE	VALUE	MIN	MAX
4	AFR	MWI	2016	BTSX	Trachea, bronchus, lung cancers	0.056	0.030	0.086

In [39]:
```
#data cleaning of owid_death_rates_air_pollution_data
owid_death_rates_air_pollution_data.drop(columns=["Entity"],inplace=True)
owid_death_rates_air_pollution_data = owid_death_rates_air_pollution_data.rename(
    columns={"Code":"COUNTRY",
             "Deaths - Air pollution - Sex: Both - Age: Age-standardized (Rate)":"DEATH (AIR POLLUTION PER 100 000)",
            "Deaths - Household air pollution from solid fuels - Sex: Both - Age: Age-standardized (Rate)":"DEATH (HOUSEHOLD PER 1
            "Deaths - Ambient particulate matter pollution - Sex: Both - Age: Age-standardized (Rate)":"DEATH (PM PER 100 000)",
            "Deaths - Ambient ozone pollution - Sex: Both - Age: Age-standardized (Rate)":"DEATH (OZONE PER 100 000)",
            "Year":"YEAR"})
owid_death_rates_air_pollution_data.head()
```

Out[39]:

	COUNTRY	YEAR	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)
0	AFG	1990	299.477309	250.362910	46.446589	5.616442
1	AFG	1991	291.277967	242.575125	46.033841	5.603960
2	AFG	1992	278.963056	232.043878	44.243766	5.611822
3	AFG	1993	278.790815	231.648134	44.440148	5.655266
4	AFG	1994	287.162923	238.837177	45.594328	5.718922

In [40]:
```
#data cleaning of owid_dalys_air_pollution_risk_data
owid_dalys_air_pollution_risk_data = owid_dalys_air_pollution_risk_data[["Entity","Code","Year",
                                                                         "DALYs (Disability-Adjusted Life Years) - Air pollution -
                                                                        "DALYs (Disability-Adjusted Life Years) - Ambient particula
                                                                        "DALYs (Disability-Adjusted Life Years) - Household air pol
owid_dalys_air_pollution_risk_data = owid_dalys_air_pollution_risk_data.rename(
    columns={"DALYs (Disability-Adjusted Life Years) - Air pollution - Sex: Both - Age: All Ages (Number)":"DALYS (Air pollution)"
            "DALYs (Disability-Adjusted Life Years) - Ambient particulate matter pollution - Sex: Both - Age: All Ages (Number)":"
            "DALYs (Disability-Adjusted Life Years) - Household air pollution from solid fuels - Sex: Both - Age: All Ages (Number
owid_dalys_air_pollution_risk_data["DALYS (Air pollution)"] = owid_dalys_air_pollution_risk_data["DALYS (Air pollution)"].astype(i
owid_dalys_air_pollution_risk_data["DALYS (PM)"] = owid_dalys_air_pollution_risk_data["DALYS (PM)"].astype(int)
owid_dalys_air_pollution_risk_data["DALYS (HOUSEHOLD, SOLID FUEL)"] = owid_dalys_air_pollution_risk_data["DALYS (HOUSEHOLD, SOLID
owid_dalys_air_pollution_risk_data.drop(columns=["Entity"],inplace=True)
owid_dalys_air_pollution_risk_data = owid_dalys_air_pollution_risk_data.rename(columns={"Code":"COUNTRY","Year":"YEAR"})
#to give country codes to those without
owid_dalys_air_pollution_risk_data.head()
```

Out[40]:

	COUNTRY	YEAR	DALYS (Air pollution)	DALYS (PM)	DALYS (HOUSEHOLD, SOLID FUEL)
0	AFG	1990	1396894	207108	1186388
1	AFG	1991	1372209	207126	1161641
2	AFG	1992	1427254	215298	1208403
3	AFG	1993	1684234	254329	1426202
4	AFG	1994	1906673	286731	1616132

In [41]:
```
#data cleaning of owid_death_pneumonia_data
owid_death_pneumonia_data.drop(columns=["Entity"],inplace=True)
owid_death_pneumonia_data = owid_death_pneumonia_data.rename(
    columns={"Code":"COUNTRY","Year":"YEAR",
             "Deaths - Lower respiratory infections - Sex: Both - Age: Age-standardized (Rate)":"DEATH (LOWER RESPIRATORY INFECTIO
owid_death_pneumonia_data.head()
```

Out[41]:

	COUNTRY	YEAR	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)
0	AFG	1990	164.811829
1	AFG	1991	151.460290
2	AFG	1992	127.896225
3	AFG	1993	124.725141
4	AFG	1994	134.410918

In [42]:
```
#data cleaning of owid_death_rate_respiratory_disease_data
owid_death_rate_respiratory_disease_data.drop(columns=["Entity"],inplace=True)
owid_death_rate_respiratory_disease_data = owid_death_rate_respiratory_disease_data.rename(
    columns={"Code":"COUNTRY","Year":"YEAR",
             "Deaths - Chronic respiratory diseases - Sex: Both - Age: Age-standardized (Rate)":"DEATH (CHRONIC RESPIRATORY DISEAS
owid_death_rate_respiratory_disease_data.head()
```

Out[42]:

	COUNTRY	YEAR	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)
0	AFG	1990	95.273780
1	AFG	1991	95.270656
2	AFG	1992	95.584266

	COUNTRY	YEAR	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)
3	AFG	1993	96.581362
4	AFG	1994	98.105844

country_population_data_1950_1980, country_population_1985_2015, country_population_2020_2050 tables are obtained from the same website **Country Population Over the Years** in separate tables. Note that this table has the population values in thousand.

country_code_data gives the country codes corresponding to the respective country names which are obtained from **Country Code Data**. The 3-letter code is obtained from the table, and then joined with the country_population_data to match the country name to the country code (majority of the other datasets contain the 3-letter code, which is more standardised and easier to compare, and useful for plotting choropleth maps in EDA for the respective research questions).

In [43]:
```
#data cleaning of country_population_data and merging to change country name to country code
country_population_data_1950_1980.rename(columns={"Country (or dependent territory)":"COUNTRY NAME","%":"1950 %",
                                                  "%.1":"1955 %","%.2":"1965 %","%.3":"1970 %","%.4":"1975 %",
                                                  "%.5":"1980 %"},inplace=True)
country_population_data_1985_2015.rename(columns={"Country (or dependent territory)":"COUNTRY NAME","%":"1985 %",
                                                  "%.1":"1990 %","%.2":"1995 %","%.3":"2000 %","%.4":"2005 %",
                                                  "%.5":"2010 %","%.6":"2015 %"},inplace=True)
country_population_data_2020_2050.rename(columns={"Country (or dependent territory)":"COUNTRY NAME","%":"2020 %",
                                                  "%.1":"2025 %","%.2":"2030 %","%.3":"2035 %","%.4":"2040 %",
                                                  "%.5":"2045 %","%.6":"2050 %"},inplace=True)
country_code_data_2 = country_code_data.copy()
country_code_data = pd.concat([country_code_data["ISO 3166[1]","Country name[5]"],
                               country_code_data["ISO 3166-1[2]","Alpha-3 code[5]"]],axis=1)
country_code_data.columns=country_code_data.columns.droplevel()
country_code_data = country_code_data.rename(columns={"Country name[5]":"COUNTRY NAME","Alpha-3 code[5]":"COUNTRY"})
#formatting of the country name to match that of country_code_data
country_code_data = country_code_data[country_code_data["COUNTRY"].str.len()==3]
country_code_data_str_name = country_code_data["COUNTRY NAME"].str.split("[")
country_code_data["COUNTRY NAME"] = country_code_data_str_name.str.get(0)
country_code_data_str_name = country_code_data["COUNTRY NAME"].str.split("(")
country_code_data["COUNTRY NAME"] = country_code_data_str_name.str.get(0)
country_code_data["COUNTRY NAME"] = country_code_data["COUNTRY NAME"].str.strip()

country_population_data = country_population_data_1950_1980.set_index("COUNTRY NAME").join(
    [country_population_data_1985_2015.set_index('COUNTRY NAME'),country_population_data_2020_2050.set_index('COUNTRY NAME'),
     country_code_data.set_index("COUNTRY NAME")])
country_population_data.drop(index=["World"],inplace=True)
country_population_data.drop(columns=["2020 %","2025","2025 %","2030","2030 %","2035","2035 %","2040","2040 %",
                                      "2045","2045 %","2050","2050 %"],inplace=True)
country_population_data.head()
```

Out[43]:

COUNTRY NAME	1950	1955	1950 %	1960	1955 %	1965	1965 %	1970	1970 %	1975	...	2000	2000 %	2005	2005 %	2010	2010 %	2015	2015 %	2020	COU
Afghanistan	8151	8892	1.76	9830	2.03	10998	2.27	12431	2.48	14133	...	22462	2.93	26335	3.23	29121	2.03	32565	2.26	36644	
Albania	1228	1393	2.56	1624	3.12	1884	3.02	2157	2.74	2402	...	3159	0.00	3025	-0.86	2987	-0.25	3030	0.28	3075	
Algeria	8893	9842	2.05	10910	2.08	11964	1.86	13932	3.09	16141	...	30639	1.58	32918	1.45	35950	1.78	39543	1.92	42973	
American Samoa	20	20	0.72	21	0.20	25	4.23	28	2.08	30	...	58	1.39	57	-0.28	56	-0.53	55	-0.41	54	
Andorra	7	7	0.04	9	6.28	14	10.17	20	7.49	27	...	66	0.58	77	3.18	85	2.12	86	0.25	86	

5 rows × 29 columns

Manual intervention is done to countries stated below as the country name cannot be matched to the country code due to spelling error or additional words and cannot be done through other ways.

In [44]:
```
country_population_data.loc["British Virgin Islands","COUNTRY"] = "VGB"
country_population_data.loc["Brunei","COUNTRY"] = "BRN"
country_population_data.loc["Cape Verde","COUNTRY"] = "CPV"
country_population_data.loc["Czech Republic","COUNTRY"] = "CZE"
country_population_data.loc["Democratic Republic of the Congo","COUNTRY"] = "COD"
country_population_data.loc["Federated States of Micronesia","COUNTRY"] = "FSM"
country_population_data.loc["Ivory Coast","COUNTRY"] = "CIV"
country_population_data.loc["Laos","COUNTRY"] = "LAO"
country_population_data.loc["Macau","COUNTRY"] = "MAC"
country_population_data.loc["North Korea","COUNTRY"] = "PRK"
country_population_data.loc["Palestine","COUNTRY"] = "PSE"
country_population_data.loc["Republic of the Congo","COUNTRY"] = "COG"
country_population_data.loc["Russia","COUNTRY"] = "RUS"
country_population_data.loc["Saint Helena, Ascension and Tristan da Cunha","COUNTRY"] = "SHN"
country_population_data.loc["South Korea","COUNTRY"] = "KOR"
country_population_data.loc["Syria","COUNTRY"] = "SYR"
country_population_data.loc["São Tomé and Príncipe","COUNTRY"] = "STP"
country_population_data.loc["Tanzania","COUNTRY"] = "TZA"
country_population_data.loc["United Kingdom","COUNTRY"] = "GBR"
country_population_data.loc["United States","COUNTRY"] = "USA"
country_population_data.loc["United States Virgin Islands","COUNTRY"] = "VIR"
```

```
country_population_data.loc["Vietnam","COUNTRY"] = "VNM"
country_population_data.drop(index={"Kosovo"},inplace=True)

country_population_data.reset_index()
country_population_data = country_population_data.set_index("COUNTRY")
country_population_data.head()
```

Out[44]:

COUNTRY	1950	1955	1950 %	1960	1955 %	1965	1965 %	1970	1970 %	1975	...	1995 %	2000	2000 %	2005	2005 %	2010	2010 %	2015	2015 %	2020
AFG	8151	8892	1.76	9830	2.03	10998	2.27	12431	2.48	14133	...	7.46	22462	2.93	26335	3.23	29121	2.03	32565	2.26	36644
ALB	1228	1393	2.56	1624	3.12	1884	3.02	2157	2.74	2402	...	-0.54	3159	0.00	3025	-0.86	2987	-0.25	3030	0.28	3075
DZA	8893	9842	2.05	10910	2.08	11964	1.86	13932	3.09	16141	...	2.37	30639	1.58	32918	1.45	35950	1.78	39543	1.92	42973
ASM	20	20	0.72	21	0.20	25	4.23	28	2.08	30	...	2.69	58	1.39	57	-0.28	56	-0.53	55	-0.41	54
AND	7	7	0.04	9	6.28	14	10.17	20	7.49	27	...	3.70	66	0.58	77	3.18	85	2.12	86	0.25	86

5 rows × 28 columns

In [45]:

```
aqi_breakpoints_units_data = aqi_breakpoints_units_data[["Parameter Code","Standard Units"]]
aqi_breakpoints_data = aqi_breakpoints_data.join(aqi_breakpoints_units_data.set_index("Parameter Code"),on="Parameter Code")
aqi_breakpoints_data.head()
```

Out[45]:

	Parameter	Parameter Code	Duration Code	Duration Description	AQI Category	Low AQI	High AQI	Low Breakpoint	High Breakpoint	Standard Units
0	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	GOOD	0	50	0.0	12.0	Micrograms/cubic meter (LC)
1	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	MODERATE	51	100	12.1	35.4	Micrograms/cubic meter (LC)
2	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	UNHEALTHY FOR SENSITIVE	101	150	35.5	55.4	Micrograms/cubic meter (LC)
3	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	UNHEALTHY	151	200	55.5	150.4	Micrograms/cubic meter (LC)
4	Acceptable PM2.5 AQI & Speciation Mass	88502	7	24 HOUR	VERY UNHEALTHY	201	300	150.5	250.4	Micrograms/cubic meter (LC)

All of the waqi datasets are obtained from **WAQI (World AQI Data)** (numbered from 13 to 25 above) which are all in the same format, though the year (period of time of the year) is different. Therefore, all of the datasets are appended to each other before doing data cleaning. Waqi datasets also use the 2-letter country code instead of 3-letter code, therefore data from country_code_data is obtained to match the 2-letter to 3-letter country codes for standardisation.

The assumption is made that the value provided under Specie column of raw waqi dataset is the AQI value of the pollutant. (A simple visual cross check between the country PM2.5 value of waqi_data_total and world_oecd_pm25_data_total shows that the values are quite close to each other)

In [46]:

```
#data cleaning of waqi data
country_code_data_2 = pd.concat([country_code_data_2["ISO 3166-1[2]","Alpha-3 code[5]"],
                                 country_code_data_2["ISO 3166-1[2]","Alpha-2 code[5]"]],axis=1)
country_code_data_2.columns=country_code_data_2.columns.droplevel()
country_code_data_2 = country_code_data_2.rename(columns={"Alpha-3 code[5]":"3 CODE","Alpha-2 code[5]":"2 CODE"})
country_code_data_2 = country_code_data_2[country_code_data_2["3 CODE"].str.len()==3]
country_code_data_2 = country_code_data_2[country_code_data_2["2 CODE"].str.len()==2]

#combining all waqi datasets into one
waqi_data_total = waqi_2015_data.append([waqi_2016_data,waqi_2017_data,waqi_2018_data,waqi_2019Q1_data,waqi_2019Q2_data,waqi_2019Q
              waqi_2019Q4_data,waqi_2020Q1_data,waqi_2020Q2_data,waqi_2020Q3_data,waqi_2020Q4_data,waqi_2021_data])
waqi_data_total = waqi_data_total.set_index("Country").join([country_code_data_2.set_index("2 CODE")])
waqi_data_total = waqi_data_total.reset_index()
waqi_data_total_str_data = waqi_data_total["Date"].str.split("-")
waqi_data_total["Date"] = waqi_data_total_str_data.str.get(0)
waqi_data_total["Date"] = waqi_data_total["Date"].str.strip()
waqi_data_total.drop(columns=["index","City","count","min","max","variance"],inplace=True)
waqi_data_total = waqi_data_total.rename(columns={"Date":"YEAR","Specie":"POLLUTANT","median":"AQI","3 CODE":"COUNTRY"})
waqi_data_total = waqi_data_total[(waqi_data_total["POLLUTANT"].astype(str)=="pm25")|
                                  (waqi_data_total["POLLUTANT"].astype(str)=="pm10")|
                                  (waqi_data_total["POLLUTANT"].astype(str)=="co")|
                                  (waqi_data_total["POLLUTANT"].astype(str)=="no2")|
                                  (waqi_data_total["POLLUTANT"].astype(str)=="o3")|
                                   (waqi_data_total["POLLUTANT"].astype(str)=="so2")]
waqi_data_total.loc[:,"POLLUTANT"] = waqi_data_total.loc[:,"POLLUTANT"].replace(
    {"pm25":"PM2.5","pm10":"PM10","co":"CO","no2":"NOX","o3":"O3","so2":"SOX"})
#waqi_data_total = waqi_data_total.groupby(["COUNTRY","YEAR","POLLUTANT"])[["AQI"]].mean()
waqi_data_total = waqi_data_total.reset_index()
waqi_data_total = waqi_data_total.set_index("COUNTRY")
waqi_data_total.drop(columns=["index"],inplace=True)
waqi_data_total["YEAR"] = waqi_data_total["YEAR"].astype(int)
waqi_data_total.head()
```

Out[46]:

COUNTRY	YEAR	POLLUTANT	AQI

COUNTRY	YEAR	POLLUTANT	AQI
ARE	2015	PM2.5	129.0
ARE	2015	PM2.5	87.0
ARE	2015	PM2.5	158.0
ARE	2015	PM2.5	127.0
ARE	2015	PM2.5	154.0

In [47]:
```
waqi_data_total_mean = waqi_data_total.groupby(["COUNTRY","YEAR","POLLUTANT"])[["AQI"]].mean()
waqi_data_total_mean.head()
```

Out[47]:

			AQI
COUNTRY	**YEAR**	**POLLUTANT**	
ARE	**2015**	**CO**	3.242500
		NOX	32.685185
		O3	30.575000
		PM10	50.450000
		PM2.5	117.712500

In [48]:
```
aqi_breakpoints_data_processed = aqi_breakpoints_data.copy()
aqi_breakpoints_data_processed = aqi_breakpoints_data_processed.set_index("Parameter")
# to merge Parameter code to units
aqi_breakpoints_data_processed = aqi_breakpoints_data_processed.drop(columns={"Parameter Code","Duration Code","AQI Category"},ind
aqi_breakpoints_data_processed = aqi_breakpoints_data_processed.rename(
    index={"Acceptable PM2.5 AQI & Speciation Mass":"PM2.5","Carbon monoxide":"CO","Nitrogen dioxide (NO2)":"NOX","Ozone":"O3",
           "PM10 Total 0-10um STP":"PM10","Sulfur dioxide":"SOX"})
aqi_breakpoints_data_processed = aqi_breakpoints_data_processed[aqi_breakpoints_data_processed["Low AQI"]!=-1]
aqi_breakpoints_data_processed = aqi_breakpoints_data_processed[aqi_breakpoints_data_processed["Duration Description"]!="24-HR BLK
aqi_breakpoints_data_processed.head()
```

Out[48]:

Parameter	Duration Description	Low AQI	High AQI	Low Breakpoint	High Breakpoint	Standard Units
PM2.5	24 HOUR	0	50	0.0	12.0	Micrograms/cubic meter (LC)
PM2.5	24 HOUR	51	100	12.1	35.4	Micrograms/cubic meter (LC)
PM2.5	24 HOUR	101	150	35.5	55.4	Micrograms/cubic meter (LC)
PM2.5	24 HOUR	151	200	55.5	150.4	Micrograms/cubic meter (LC)
PM2.5	24 HOUR	201	300	150.5	250.4	Micrograms/cubic meter (LC)

In [49]:
```
death_by_cause_data = owid_death_rate_respiratory_disease_data.join(
    owid_death_pneumonia_data.set_index(["COUNTRY","YEAR"]),on=["COUNTRY","YEAR"])
death_by_cause_data = death_by_cause_data.join(
    owid_death_rates_air_pollution_data.set_index(["COUNTRY","YEAR"]),on=["COUNTRY","YEAR"])
death_by_cause_data.head()
```

Out[49]:

	COUNTRY	YEAR	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)
0	AFG	1990	95.273780	164.811829	299.477309	250.362910	46.446589	5.616442
1	AFG	1991	95.270656	151.460290	291.277967	242.575125	46.033841	5.603960
2	AFG	1992	95.584266	127.896225	278.963056	232.043878	44.243766	5.611822
3	AFG	1993	96.581362	124.725141	278.790815	231.648134	44.440148	5.655266
4	AFG	1994	98.105844	134.410918	287.162923	238.837177	45.594328	5.718922

The function below divides the number of dalys for each country by the respective country's population to standardise the number per 100 000 of the population to mitigate the effect of the population of the country on the number of dalys years.

As the data for country populations are only available every 5 years, the percentage growth is used to project the population in a particular year based on population growth or shrink from the nearest available population data. For example to get the population of the country in 1996, the country's population in 1995 is taken, with will then be multiplied by the percentage growth of the population as indicated by '1995 %' in this case.

In [50]:
```
def calculate_per_100000(row):
    country_population,dalys_air_pollution,dalys_pm,dalys_household = 0,0,0,0
    #country population is already in thousands
    if (row.YEAR%5==0):
        country_population = country_population_data.loc[row.COUNTRY,str(row.YEAR)]
    else:
        year_remainder = row.YEAR%5
```

```
        #country population is estimateed by population growth
        year_to_retrieve = row.YEAR-year_remainder
        country_population = country_population_data.loc[row.COUNTRY,str(year_to_retrieve)]*(1+
            (country_population_data.loc[row.COUNTRY,(str(year_to_retrieve)+" %")])/100)**year_remainder
        if (country_population<0):
            print("Error"+row.COUNTRY+str(country_population))
    if (country_population!=0):
        dalys_air_pollution = row["DALYS (Air pollution)"]/country_population*100
        dalys_pm = row["DALYS (PM)"]/country_population*100
        dalys_household = row["DALYS (HOUSEHOLD, SOLID FUEL)"]/country_population*100
        row["DALYS (AIR POLLUTION PER 100 000)"] = dalys_air_pollution
        row["DALYS (PM PER 100 000)"] = dalys_pm
        row["DALYS (HOUSEHOLD, SOLID FUEL PER 100 000)"] = dalys_household
    else:
        print("Error"+row.COUNTRY+str(row.YEAR))

    return row
```

In [51]:

```
#further data organisation for owid_death_rates_air_pollution_data,owid_death_pneumonia_data,
#owid_death_rate_respiratory_disease_data to the same format and merge into one dataframe containing all the average dalys
#due to different respiratory / pollution related causes
owid_death_rates_air_pollution_data_total = owid_death_rates_air_pollution_data.copy()
owid_death_rates_air_pollution_data_total = owid_death_rates_air_pollution_data_total.groupby(
    by=["COUNTRY"])[["DEATH (AIR POLLUTION PER 100 000)","DEATH (HOUSEHOLD PER 100 000)",
                     "DEATH (PM PER 100 000)","DEATH (OZONE PER 100 000)"]].mean()

owid_death_pneumonia_data_total = owid_death_pneumonia_data.copy()
owid_death_pneumonia_data_total = owid_death_pneumonia_data_total.groupby(
    by=["COUNTRY","YEAR"])[["DEATH (LOWER RESPIRATORY INFECTION PER 100 000)"]].sum()

owid_death_rate_respiratory_disease_data_total = owid_death_rate_respiratory_disease_data.copy()
owid_death_rate_respiratory_disease_data_total = owid_death_rate_respiratory_disease_data_total.groupby(
    by=["COUNTRY","YEAR"])[["DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)"]].sum()
owid_death_rate_respiratory_disease_data_total = owid_death_rate_respiratory_disease_data_total.groupby(by=["COUNTRY"])[["DEATH (C
owid_dalys_air_pollution_risk_data_by_country = owid_dalys_air_pollution_risk_data.dropna(subset=["COUNTRY"],axis=0)

owid_dalys_air_pollution_risk_data_by_country = owid_dalys_air_pollution_risk_data_by_country.set_index("COUNTRY")
owid_dalys_air_pollution_risk_data_by_country.drop(index=["OWID_WRL"],inplace=True)
owid_dalys_air_pollution_risk_data_by_country = owid_dalys_air_pollution_risk_data_by_country.reset_index()
owid_dalys_air_pollution_risk_data_by_country = owid_dalys_air_pollution_risk_data_by_country.apply(calculate_per_100000,axis='col
owid_dalys_air_pollution_risk_data_by_country.drop(columns=["DALYS (Air pollution)","DALYS (PM)","DALYS (HOUSEHOLD, SOLID FUEL)"],
owid_dalys_air_pollution_risk_data_by_country.head()
```

Out[51]:

	COUNTRY	YEAR	DALYS (AIR POLLUTION PER 100 000)	DALYS (PM PER 100 000)	DALYS (HOUSEHOLD, SOLID FUEL PER 100 000)
0	AFG	1990	10294.745375	1526.332080	8743.370919
1	AFG	1991	10045.518374	1516.305489	8504.015066
2	AFG	1992	10378.946878	1565.640387	8787.469185
3	AFG	1993	12166.184045	1837.163614	10302.271547
4	AFG	1994	13681.323112	2057.436937	11596.547538

In [52]:

```
world_oecd_pm25_data_total = world_oecd_pm25_data.copy()
world_oecd_pm25_data_total = world_oecd_pm25_data_total.groupby(by=["COUNTRY"])[["PM2.5 (MICGRCUBM)"]].mean()

world_oecd_pollutants_data_total = world_oecd_pollutants_data.copy()
world_oecd_pollutants_data_total = world_oecd_pollutants_data_total.groupby(by=["COUNTRY","YEAR","POLLUTANT"])[["VALUE"]].sum()
world_oecd_pollutants_data_total = world_oecd_pollutants_data_total.groupby(by=["COUNTRY","POLLUTANT"])[["VALUE"]].mean()
world_oecd_pollutants_data_total = pd.pivot_table(world_oecd_pollutants_data,index=["COUNTRY"],
                                                  values=["VALUE"],columns=["POLLUTANT"],aggfunc="mean")
world_oecd_pollutants_data_total.columns = world_oecd_pollutants_data_total.columns.droplevel()
world_oecd_pollutants_data_total = world_oecd_pollutants_data_total.rename(
    columns={"CO":" CO (KIILOTON)","NMVOC":"NMVOC (KILOTON)","NOX":"NOX (KILOTON)","PM10":"PM10 (KILOTON)",
             "PM2-5":"PM 2.5 (KILOTON)","SOX":"SOX (KILOTON)"})

world_oecd_pollutants_data_total.head()
```

Out[52]:

POLLUTANT	CO (KIILOTON)	NMVOC (KILOTON)	NOX (KILOTON)	PM10 (KILOTON)	PM 2.5 (KILOTON)	SOX (KILOTON)
COUNTRY						
AUS	859.384791	257.033154	443.022199	NaN	NaN	418.012110
AUT	184.601067	44.648801	54.816435	16.181534	13.340493	15.165105
BEL	157.937688	48.434994	68.307550	13.429882	11.266879	41.482949
CAN	1747.873402	470.856396	463.042310	1299.853592	302.837720	393.038079
CHE	82.595957	35.964318	29.029860	14.044797	11.833351	12.354225

The picture below displays how the AQI value is calculated based on the concentration of a certain pollution based on the standard table of AQI breakpoints. The function calculate_aqi takes in the relevant parameters (row of dataframe, pollutant, pollutant name) and calculates the corresponding AQI value using the dataframe aqi_breakpoints_data_processed.

$$I = \frac{I_{high} - I_{low}}{C_{high} - C_{low}}(C - C_{low}) + I_{low}$$

where:

I = the (Air Quality) index,

C = the pollutant concentration,

C_{low} = the concentration breakpoint that is ≤ C,

C_{high} = the concentration breakpoint that is ≥ C,

I_{low} = the index breakpoint corresponding to C_{low},

I_{high} = the index breakpoint corresponding to C_{high}.

In [53]:
```
def calculate_aqi(row,pollutant,pollutant_column_name):
    pollutant_df = aqi_breakpoints_data_processed.loc[pollutant]
    pollutant_value = row[pollutant_column_name]
    pollutant_df_row = pollutant_df[(pollutant_df["Low Breakpoint"]<=pollutant_value) &
                                    (pollutant_df["High Breakpoint"]>=pollutant_value)]
    aqi_value = (pollutant_df_row["High AQI"]-pollutant_df_row["Low AQI"])/(
        pollutant_df_row["High Breakpoint"]-pollutant_df_row["Low Breakpoint"])*(
        pollutant_value-pollutant_df_row["Low Breakpoint"])+pollutant_df_row["Low AQI"]
    #corner case wehere pollutant is between breakpoints and does not fall under any category, just round off to threshold
    if pollutant_df_row.empty:
        pollutant_value = round(pollutant_value,1)
        pollutant_df_row = pollutant_df[(pollutant_df["Low Breakpoint"]<=pollutant_value) &
                                    (pollutant_df["High Breakpoint"]>=pollutant_value)]
        aqi_value = (pollutant_df_row["High AQI"]-pollutant_df_row["Low AQI"])/(
            pollutant_df_row["High Breakpoint"]-pollutant_df_row["Low Breakpoint"])*(
            pollutant_value-pollutant_df_row["Low Breakpoint"])+pollutant_df_row["Low AQI"]
    row["AQI"] = float(aqi_value)
    return row
world_oecd_pm25_data_total = world_oecd_pm25_data_total.apply(calculate_aqi,pollutant='PM2.5',pollutant_column_name='PM2.5 (MICGRC
world_oecd_pm25_data_total.head()
```

Out[53]:

COUNTRY	PM2.5 (MICGRCUBM)	AQI
AFG	77.286359	162.249016
AGO	64.393955	155.592242
ALB	60.483999	153.573403
ARE	71.198538	159.105673
ARG	56.268959	151.397039

Data Exploration and Analysis (Preliminary Results)

In [54]:
```
#to beautify graphs
sns.set()
```

Q1. Effect of type of air pollutant on health of people

Air pollution can be factored by different kinds of air pollutants, namely particulate matter (PM 2.5 and PM 10), ground level ozone (O3), carbon monoxide (CO), nitrogen dioxide (NO2), sulfur dioxide (SO2) in the environment. There are many factors to which an increase in respiratory related diseases can be brought about, therefore I would like to find out whether there is an impact and the extent different air pollutants worsens the health of people. Health of people can be defined as the number of cases of respiratory related diseases or mortality rate. In this project, the average number of deaths and average number of dalys (disability adjusted life years) are provided / calculated.

I first collate the level of different pollutants and calculate their air quality index categorised by country, then add up the number of associated deaths due to respiratory diseases. I can compare the countries having the highest AQI / highest average dalys / associated deaths, AQI values for different types of pollutants and observe the relationship of the concentration of pollutants against the number of associated average deaths per 100 000 of population.

Disclaimer: Data for x axis and y axis are spanding across different years, therefore average is taken to mitigate the effect of year on the results obtained.

The AQI value for each country is obtained, which is merged with the data containing the average dalys per 100 000 and average deaths per 100 000. The mean value of the AQI and average dalys is taken to match it into one data entry.

In [55]:
```
owid_dalys_air_pollution_risk_qn1 = owid_dalys_air_pollution_risk_data_by_country.copy()
owid_dalys_air_pollution_risk_qn1 = owid_dalys_air_pollution_risk_qn1.set_index(["COUNTRY","YEAR"])
owid_dalys_air_pollution_risk_qn1["AVERAGE DALYS PER 100 000"] = owid_dalys_air_pollution_risk_qn1.mean(axis=1)
owid_dalys_air_pollution_risk_qn1.drop(columns=["DALYS (AIR POLLUTION PER 100 000)",
                                "DALYS (PM PER 100 000)","DALYS (HOUSEHOLD, SOLID FUEL PER 100 000)"],inplace=True)
owid_dalys_air_pollution_risk_qn1_temp = owid_dalys_air_pollution_risk_qn1.reset_index()
owid_dalys_air_pollution_risk_qn1 = owid_dalys_air_pollution_risk_qn1_temp.groupby(["COUNTRY"])[["AVERAGE DALYS PER 100 000"]].mea
health_dalys_pollution_qn1 = waqi_data_total.join(owid_dalys_air_pollution_risk_qn1,on="COUNTRY")
health_dalys_pollution_qn1 = health_dalys_pollution_qn1.reset_index()
health_dalys_pollution_qn1 = health_dalys_pollution_qn1.groupby(["COUNTRY","POLLUTANT"])[["AQI","AVERAGE DALYS PER 100 000"]].mean
health_dalys_pollution_qn1 = health_dalys_pollution_qn1.reset_index()
health_dalys_pollution_qn1.head()
```

Out[55]:

	COUNTRY	POLLUTANT	AQI	AVERAGE DALYS PER 100 000
0	ARE	CO	0.336595	597.45297
1	ARE	NOX	12.645674	597.45297
2	ARE	O3	21.508303	597.45297
3	ARE	PM10	77.799667	597.45297
4	ARE	PM2.5	104.595389	597.45297

In [56]:
```
plt.figure(figsize=(15,8))
plt.title("Graph of average DALYS (disability-adjusted life years) per 100 000 of "+
          "country population against AQI index value for different pollutants")
sns.scatterplot(x="AQI",y="AVERAGE DALYS PER 100 000",hue="POLLUTANT",data=health_dalys_pollution_qn1)
caption = "Figure 1.1: DALYS against AQI"
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

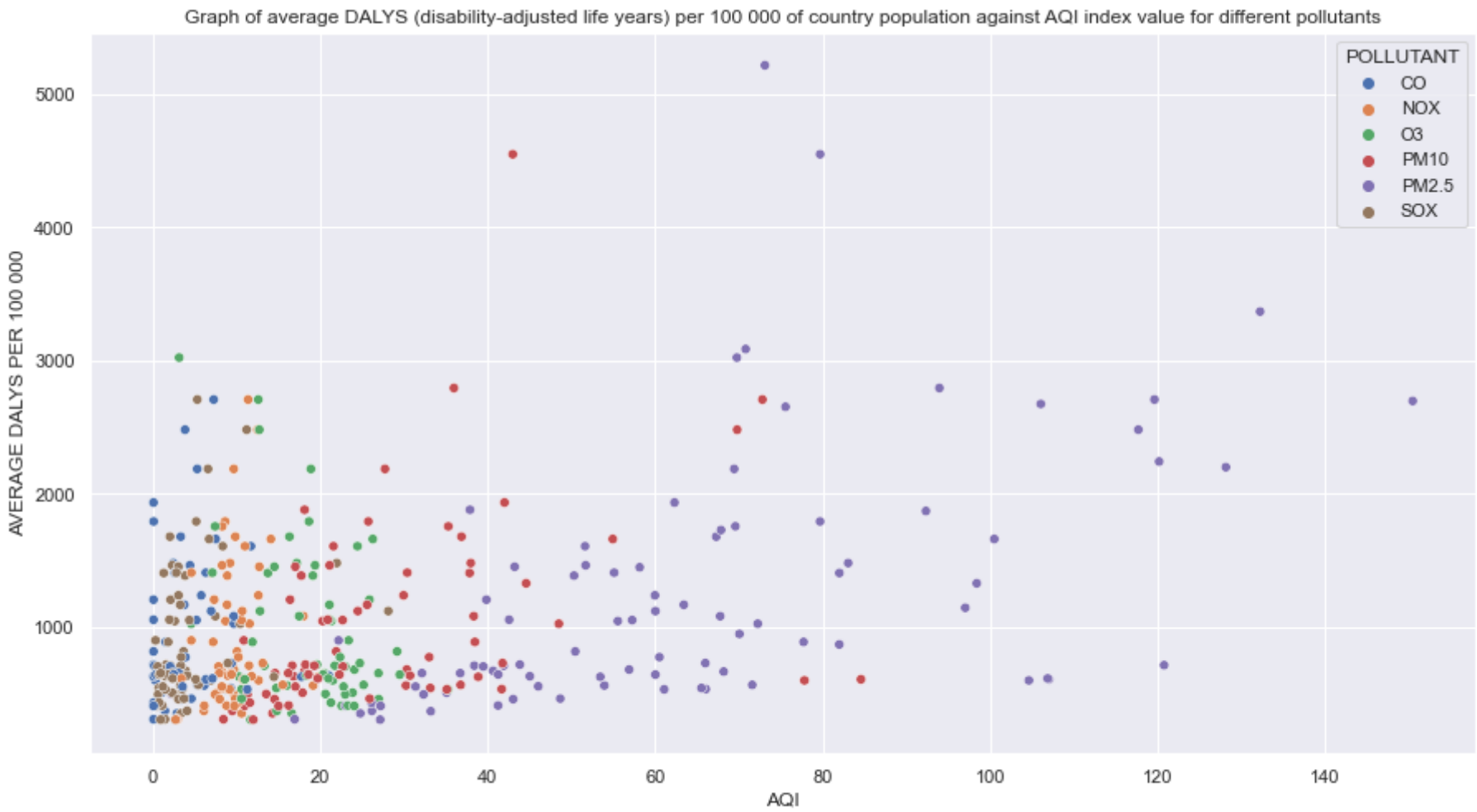


Figure 1.1: DALYS against AQI

This graph shows a scatterplot of average dalys per 100 000 against aqi value. It can be noted that majority of the datapoints cluster at the bottom left corner (low average deaths and low AQI). It can be observed that there is a weak positive correlation that may not necessarily be linear. Further investigation and EDA needs to be done to investigate the correlation for each pollutant individually as it is difficult to tell from this combined scatterplot alone.

In [57]:
```
death_by_cause_qn1 = death_by_cause_data.copy()
death_by_cause_qn1 = death_by_cause_qn1.set_index(["COUNTRY","YEAR"])
death_by_cause_qn1["AVERAGE DEATHS PER 100 000"] = death_by_cause_qn1.mean(axis=1)
death_by_cause_qn1.drop(columns=["DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)",
                                 "DEATH (LOWER RESPIRATORY INFECTION PER 100 000)",
```

```
                                  "DEATH (AIR POLLUTION PER 100 000)", "DEATH (HOUSEHOLD PER 100 000)",
                                   "DEATH (PM PER 100 000)","DEATH (OZONE PER 100 000)"],inplace=True)
death_by_cause_qn1_temp = death_by_cause_qn1.reset_index()
death_by_cause_qn1 = death_by_cause_qn1_temp.groupby(["COUNTRY"])[["AVERAGE DEATHS PER 100 000"]].mean()
health_deaths_pollution_qn1 = waqi_data_total.join(death_by_cause_qn1)
health_deaths_pollution_qn1 = health_deaths_pollution_qn1.reset_index()
health_deaths_pollution_qn1 = health_deaths_pollution_qn1.groupby(["COUNTRY","POLLUTANT"])[["AQI","AVERAGE DEATHS PER 100 000"]].m
health_deaths_pollution_qn1 = health_deaths_pollution_qn1.reset_index()
health_deaths_pollution_qn1.head()
```

Out[57]:

	COUNTRY	POLLUTANT	AQI	AVERAGE DEATHS PER 100 000
0	ARE	CO	0.336595	44.900843
1	ARE	NOX	12.645674	44.900843
2	ARE	O3	21.508303	44.900843
3	ARE	PM10	77.799667	44.900843
4	ARE	PM2.5	104.595389	44.900843

In [58]:

```
plt.figure(figsize=(15,8))
plt.title("Graph of average deaths associated with air pollution per 100 000 of country population against "+
          "AQI index value for different pollutants")
sns.scatterplot(x="AQI",y="AVERAGE DEATHS PER 100 000",hue="POLLUTANT",data=health_deaths_pollution_qn1)
caption = "Figure 1.2: DEATHS against AQI"
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

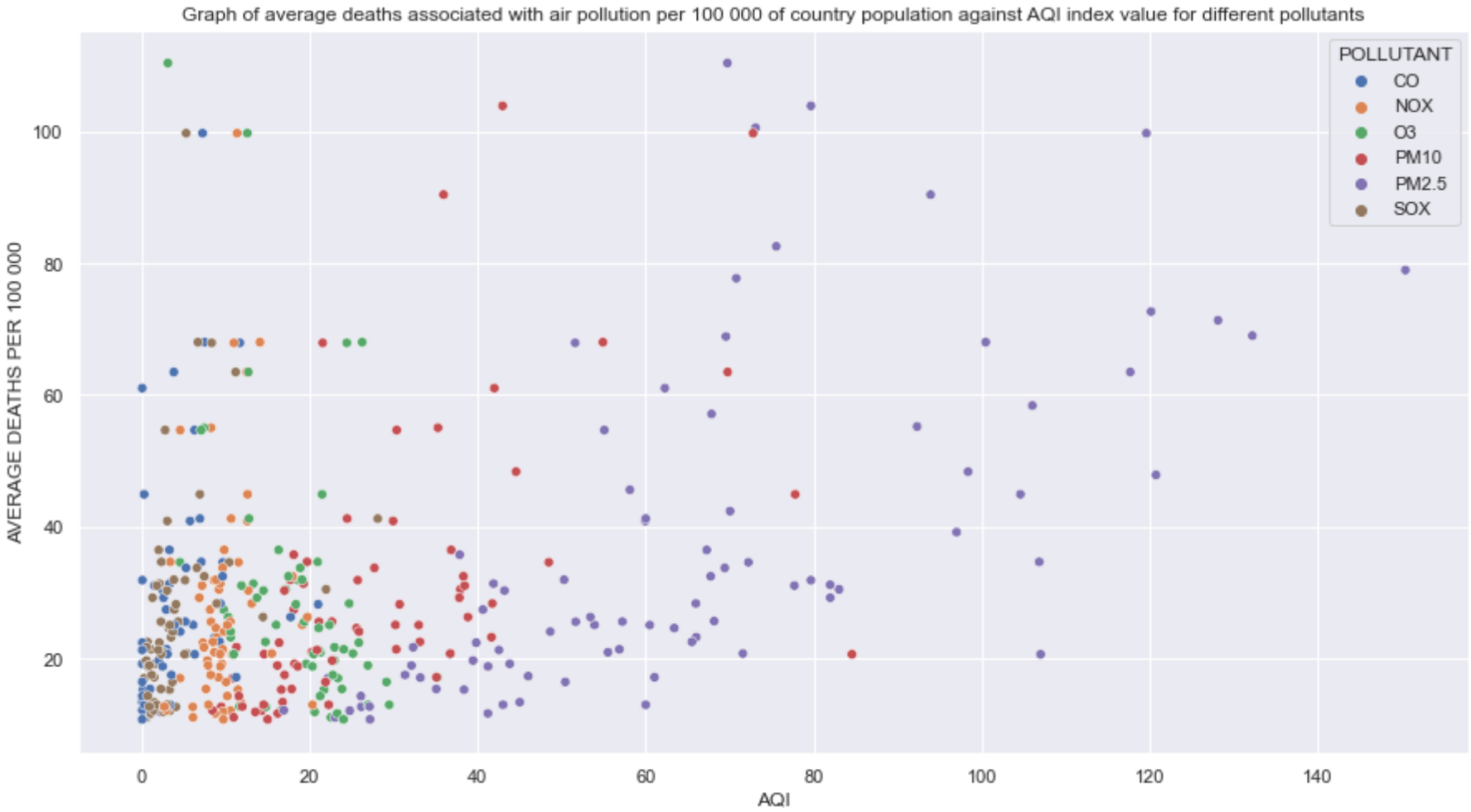


Figure 1.2: DEATHS against AQI

This graph shows a scatterplot of average deaths per 100 000 against AQI value. It can be observed that there is a weak positive correlation that may not necessarily be linear. Further investigation and EDA needs to be done to investigate the correlation for each pollutant individually.

In [59]:

```
qn1_dalys = sns.FacetGrid(health_dalys_pollution_qn1, col="POLLUTANT",height=5,aspect=1.2,col_wrap=2)
qn1_dalys.map_dataframe(sns.regplot, x="AQI", y="AVERAGE DALYS PER 100 000")
qn1_dalys.fig.subplots_adjust(top=0.9)
qn1_dalys.fig.suptitle("FacetGrid of average dalys per 100 000 against AQI for different pollutant types")
caption = "Figure 1.3.1: DALYS against AQI for different pollutants"
plt.figtext(0.5, -0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
```

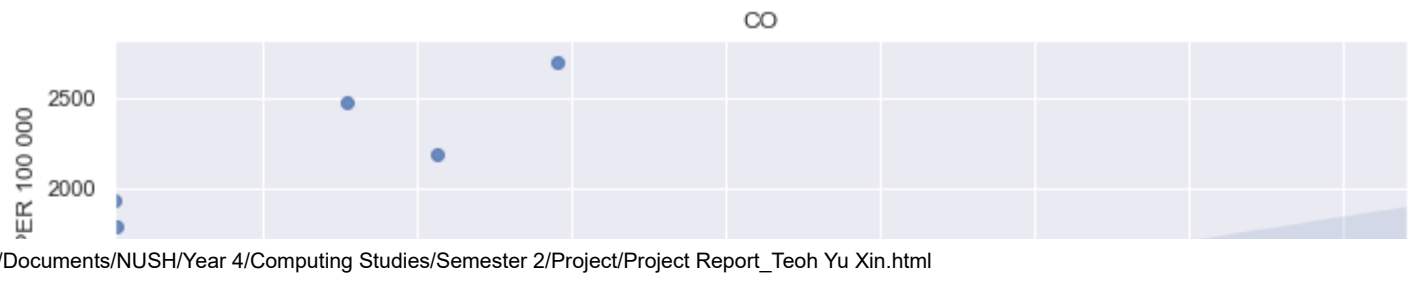
Out[59]:

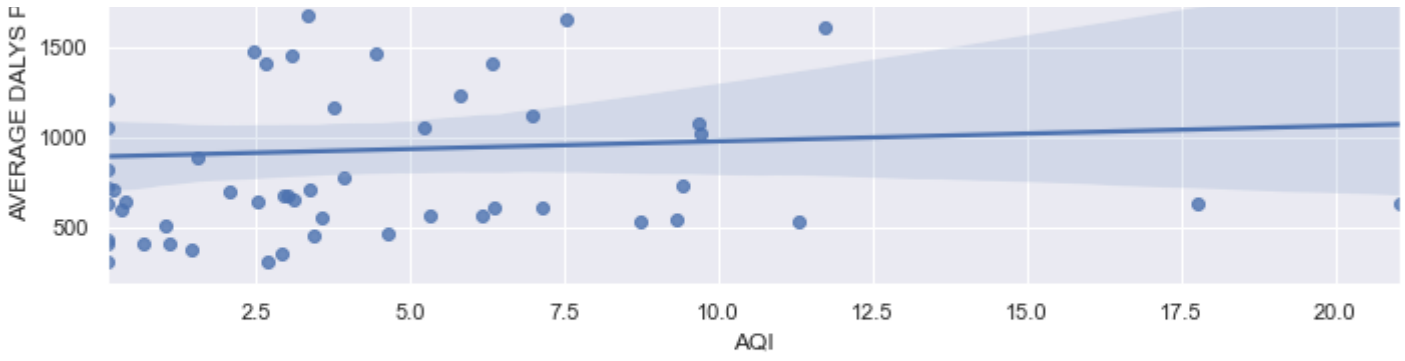
```
Text(0.5, -0.01, 'Figure 1.3.1: DALYS against AQI for different pollutants')
```

Figure 1.3.1: DALYS against AQI for different pollutants

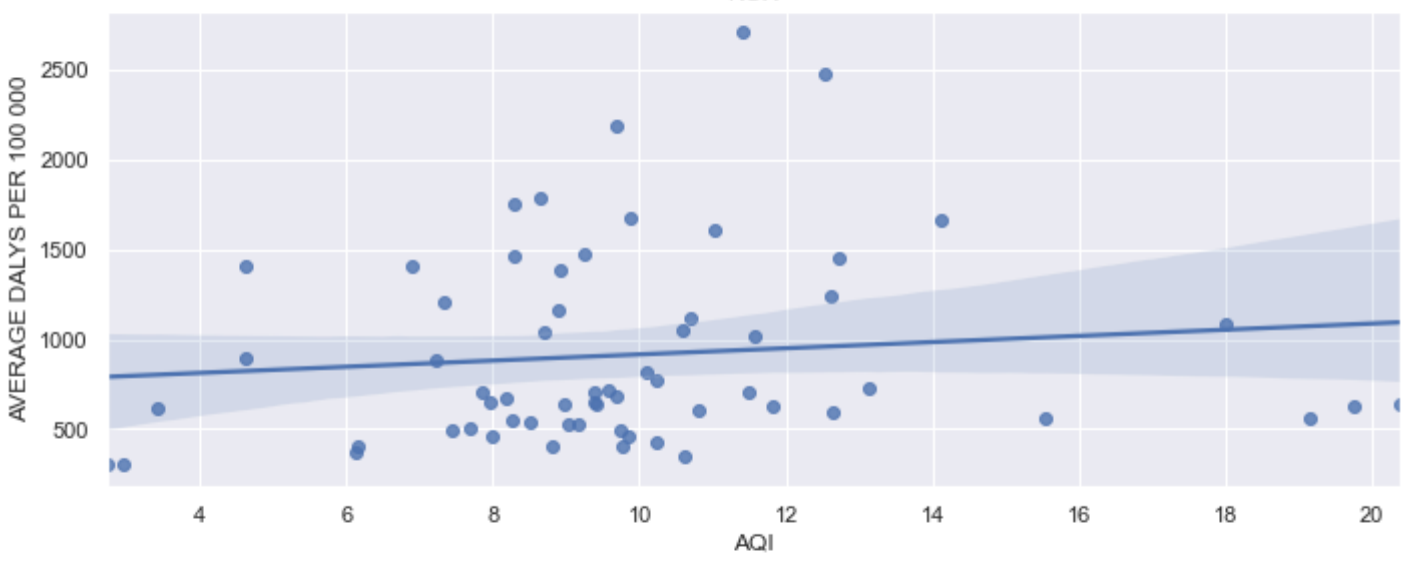
By using the same x-axis, it is unable to show a good regression plot for pollutants= CO, NOX, O3, therefore small multiples is used instead to allow for scaling of the x axis for different regression plots.

In [60]:
```
fig, axes = plt.subplots(nrows=6, ncols=1,  tight_layout=True,figsize=(10,25))

count=0
pollutant_list = health_dalys_pollution_qn1["POLLUTANT"].unique()
for r in range(6):
    ax = axes[r]
    sns.regplot(x="AQI",y="AVERAGE DALYS PER 100 000",
                data=health_dalys_pollution_qn1[health_dalys_pollution_qn1["POLLUTANT"]==pollutant_list[r]],ax=ax)
    ax.set_title(pollutant_list[r])
    count+=1
caption = "Figure 1.3.2: DALYS against AQI for different pollutants (small multiple)"
plt.figtext(0.5, -0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

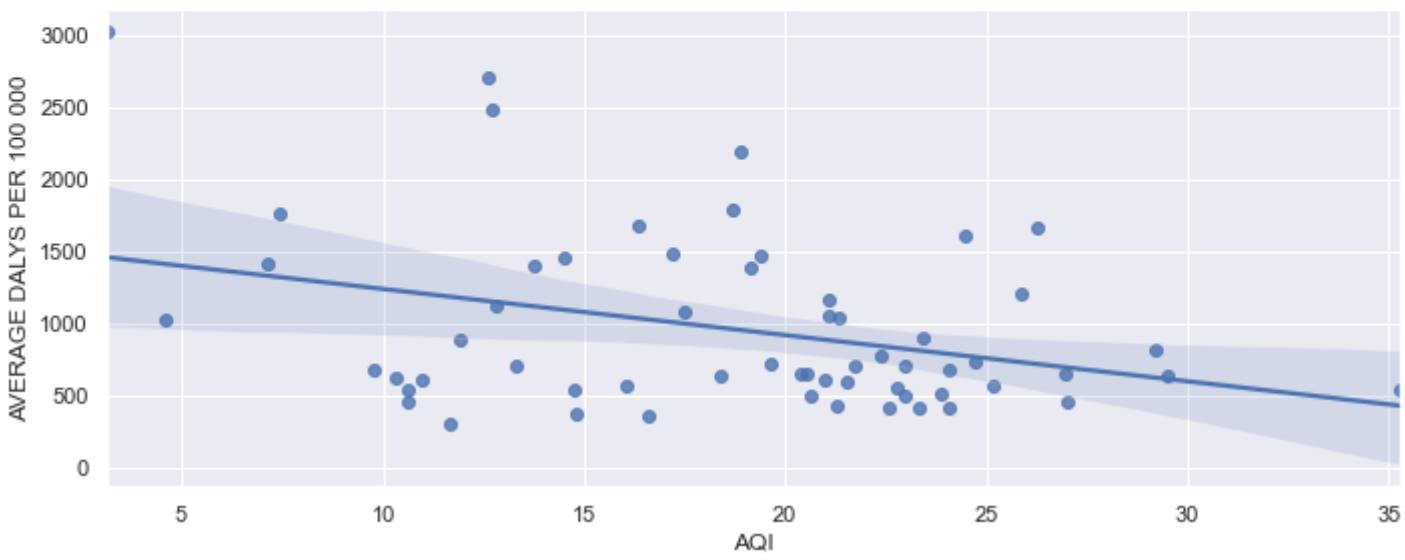




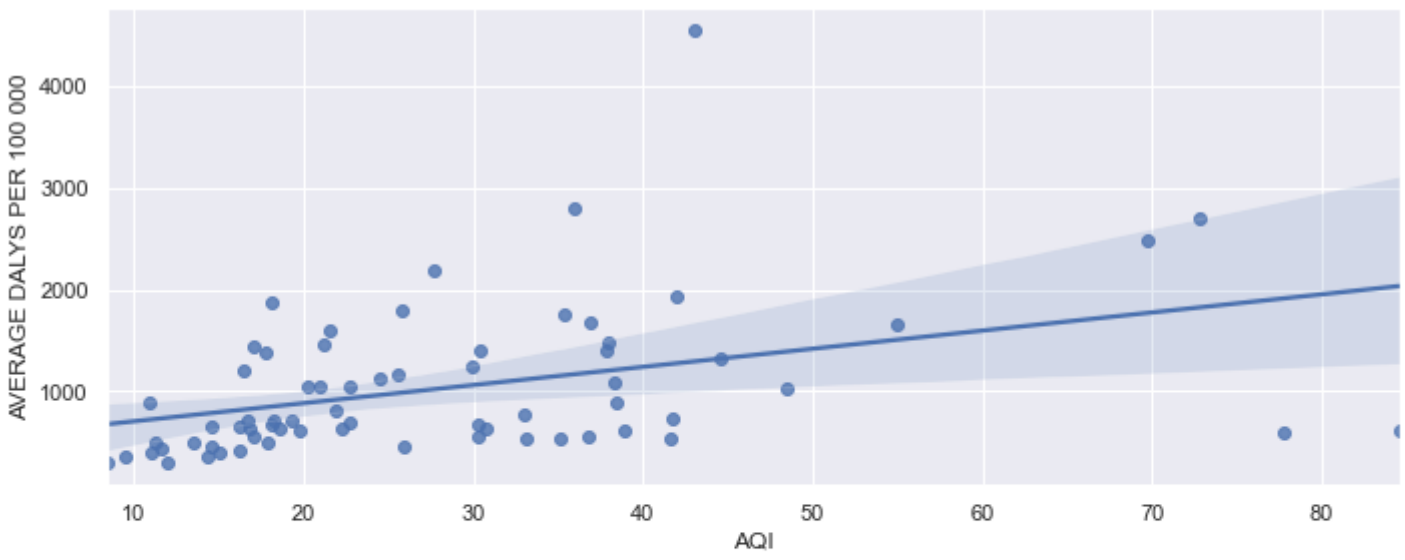
NOX



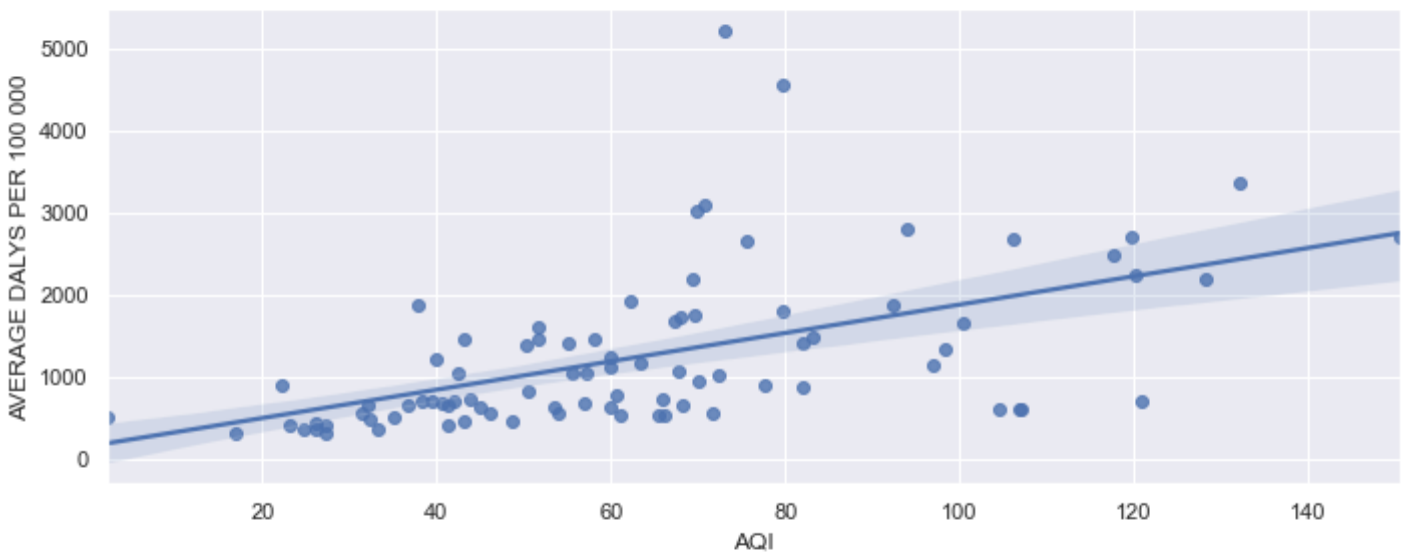
O3



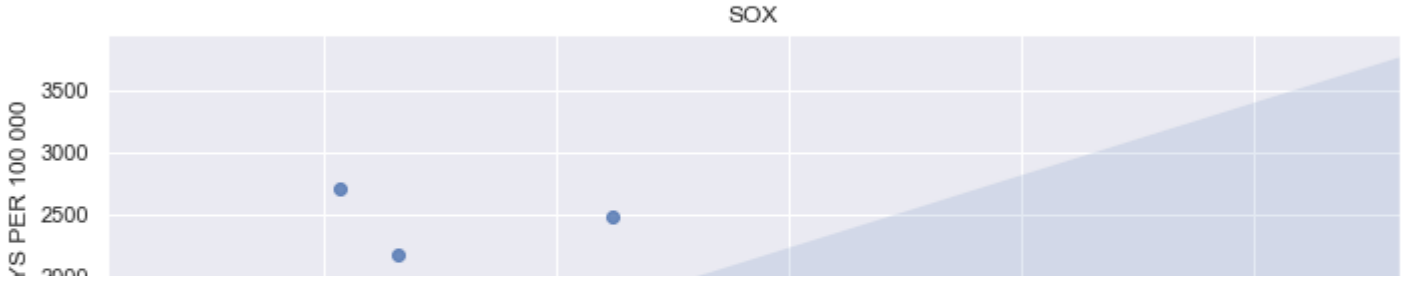
PM10



PM2.5



SOX



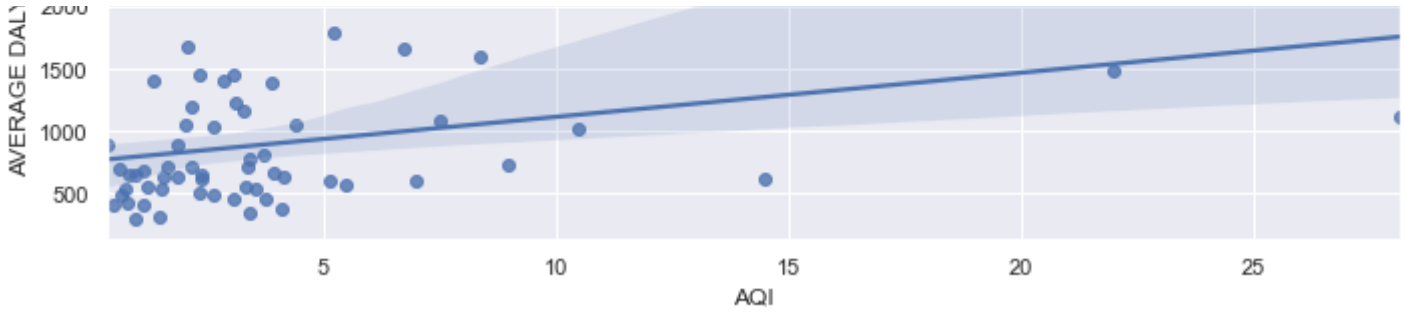


Figure 1.3.2: DALYS against AQI for different pollutants (small multiple)

From the plot of small multiples, it can be observed that PM 10 and PM 2.5 display the greatest linear positive correlation among the different pollutants. Pollutants CO, NOX and SOX show little to no positive or negative correlation. Further investigation by performing linear regression model on these pollutants will be carried out.

Interestingly, the pollutant O3 seems to indicate a weak negative correlation, meaning that a higher AQI value (worsened air quality) indicates a lower average number of dalys. This may already be due to O3 AQI values being low (ranging from 0 to 35) such that they do not have much effect on the dalys, such that the weak correlation may be insignificant.

In [61]:
```
qn1_deaths = sns.FacetGrid(health_deaths_pollution_qn1, col="POLLUTANT",height=5,aspect=1.2,col_wrap=2)
qn1_deaths.map_dataframe(sns.regplot, x="AQI", y="AVERAGE DEATHS PER 100 000")
qn1_deaths.fig.subplots_adjust(top=0.9)
qn1_deaths.fig.suptitle("FacetGrid of average deaths per 100 000 against AQI for different pollutant types")
caption = "Figure 1.4.1: DEATHS against AQI for different pollutants"
plt.figtext(0.5, -0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
```

Out[61]: Text(0.5, -0.01, 'Figure 1.4.1: DEATHS against AQI for different pollutants')

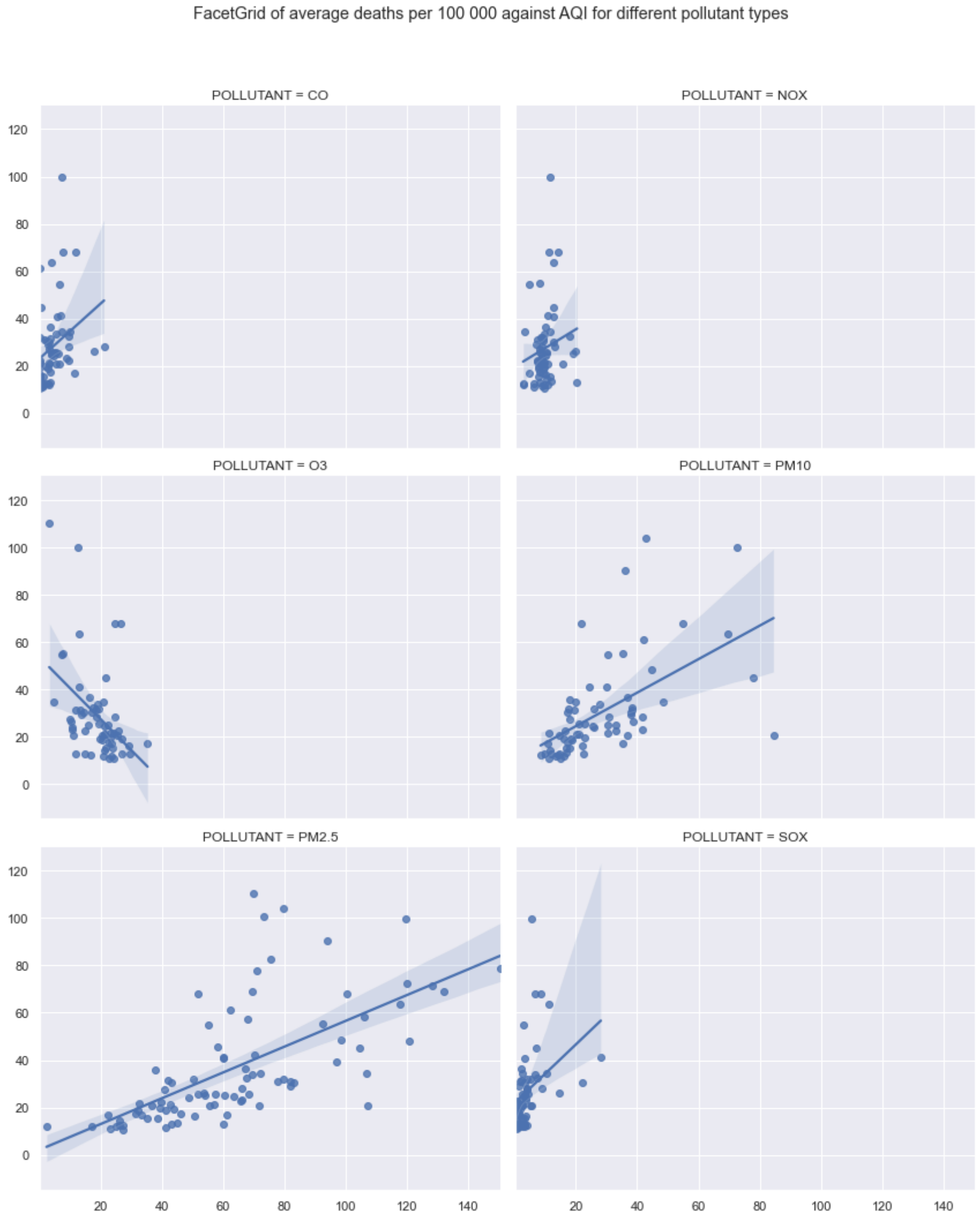
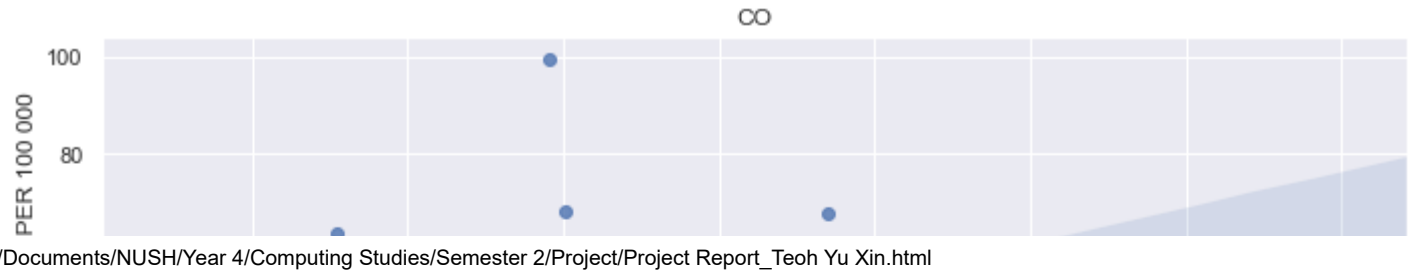
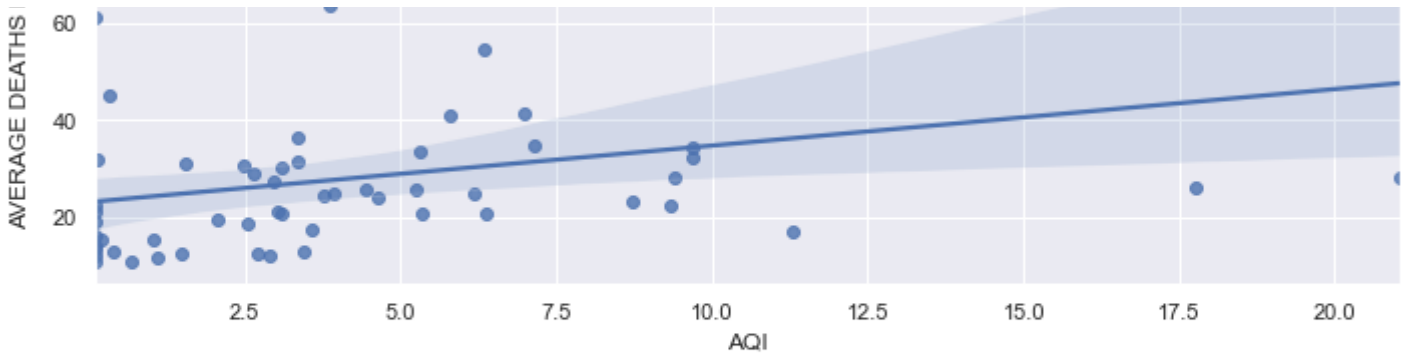


Figure 1.4.1: DEATHS against AQI for different pollutants

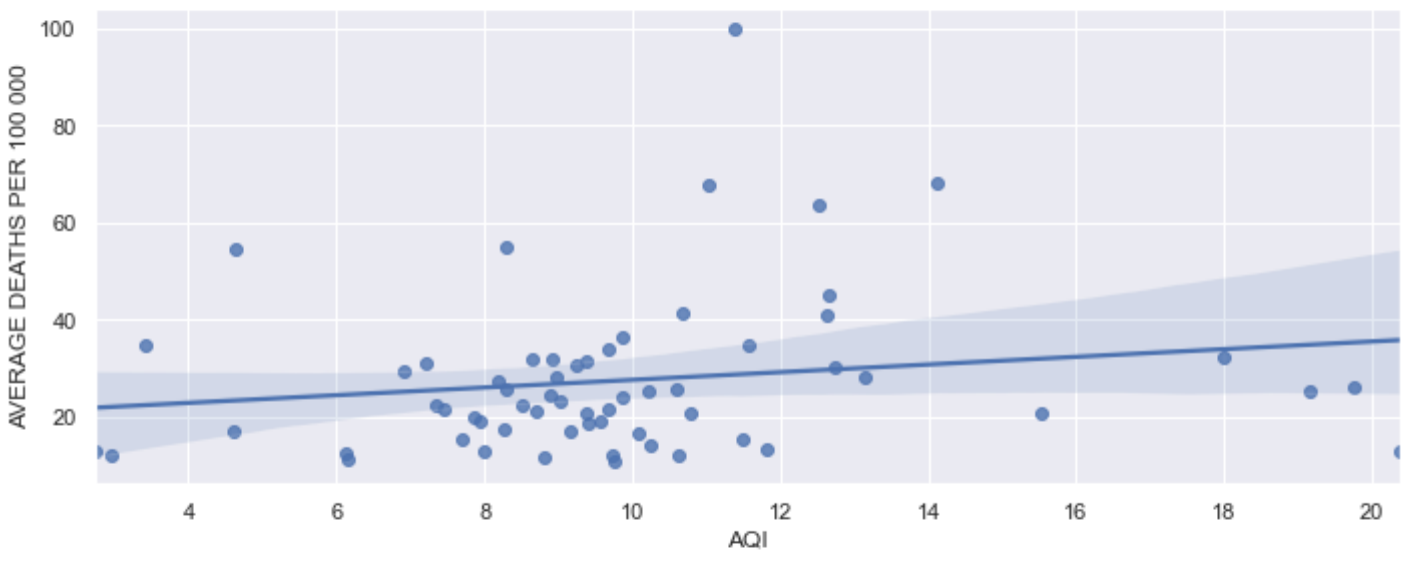
By using the same x-axis, it is unable to show a good regression plot for pollutants= CO, NOX, O3, therefore small multiples is used instead to allow for scaling of the x axis for different regression plots.

In [62]:
```
fig, axes = plt.subplots(nrows=6, ncols=1,  tight_layout=True,figsize=(10,25))

count=0
pollutant_list = health_deaths_pollution_qn1["POLLUTANT"].unique()
for r in range(6):
    ax = axes[r]
    sns.regplot(x="AQI",y="AVERAGE DEATHS PER 100 000",
                data=health_deaths_pollution_qn1[health_deaths_pollution_qn1["POLLUTANT"]==pollutant_list[r]],ax=ax)
    ax.set_title(pollutant_list[r])
    count+=1
caption = "Figure 1.4.2: DEATHS against AQI for different pollutants"
plt.figtext(0.5, -0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

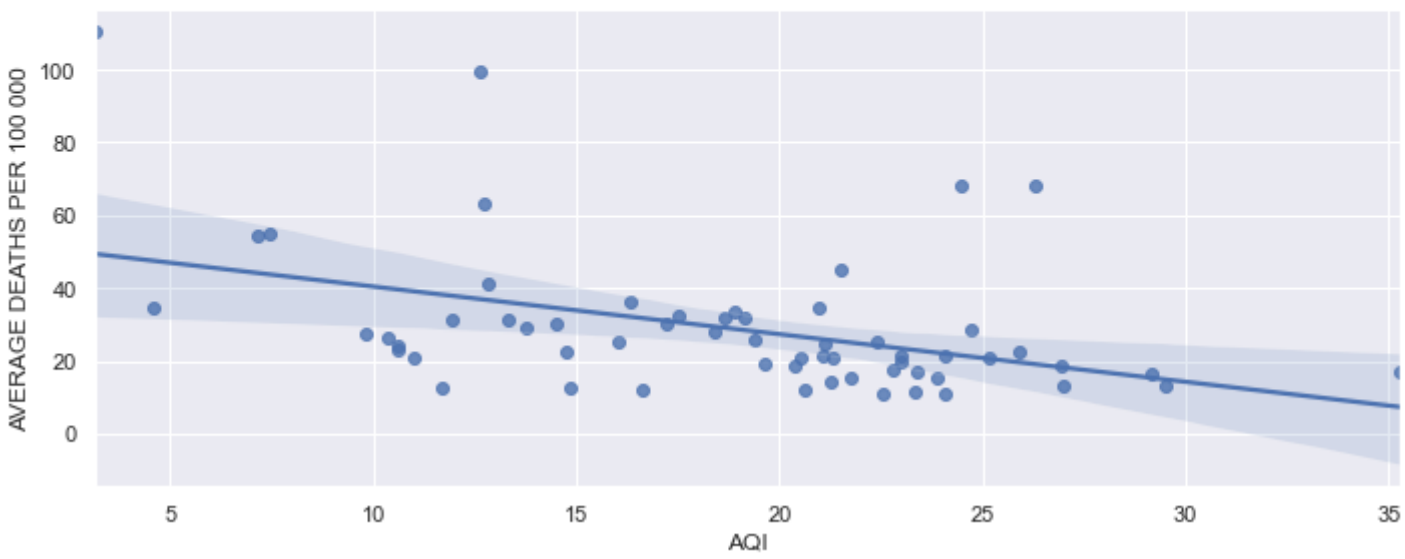




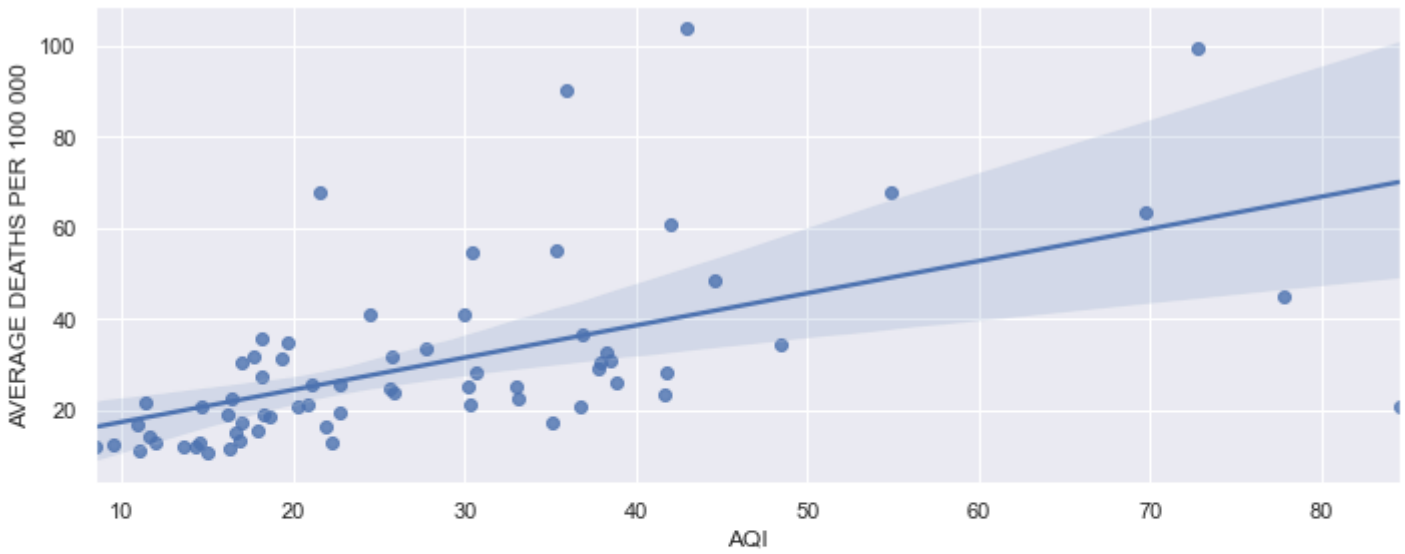
NOX



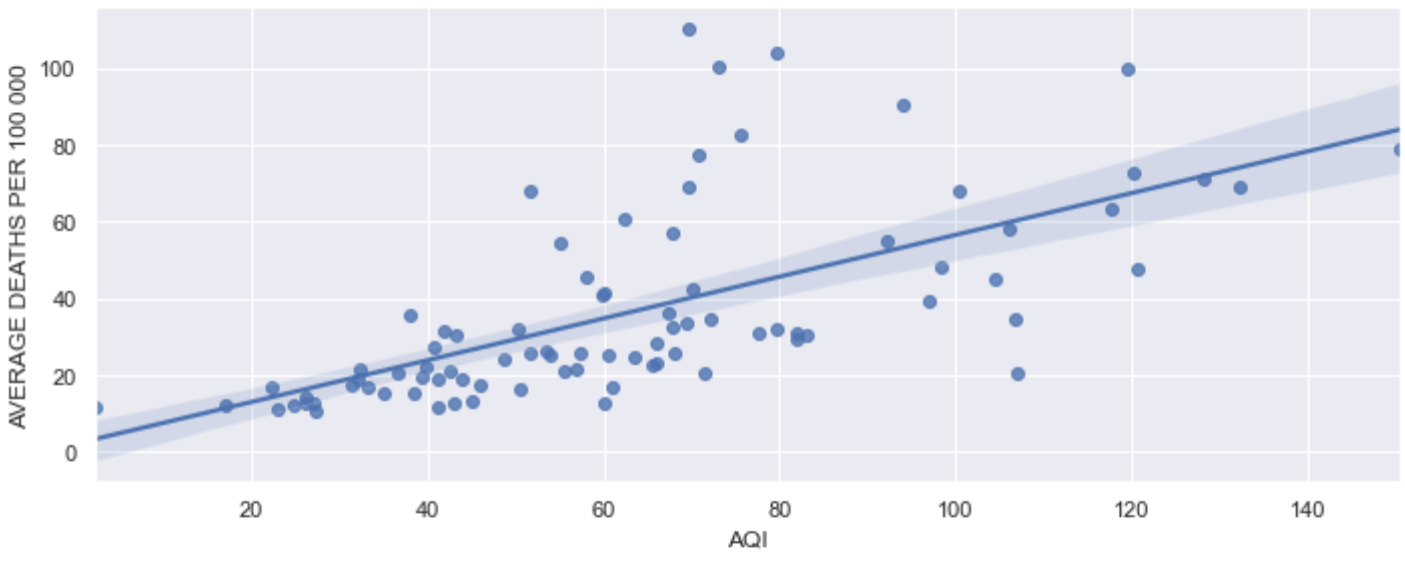
O3



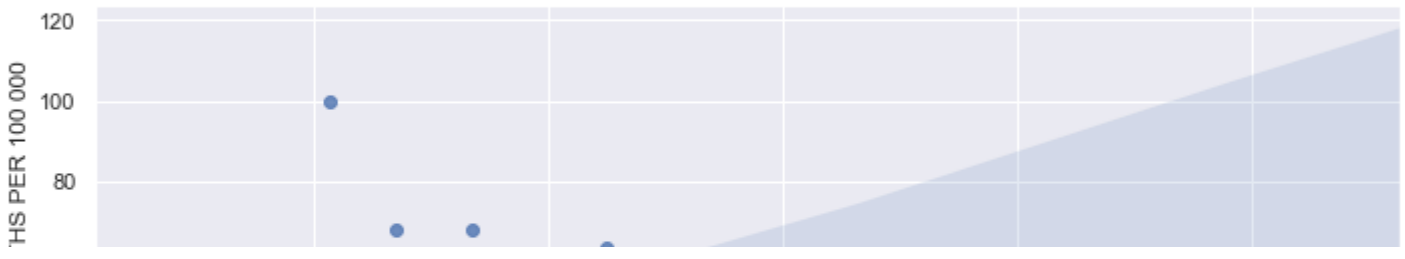
PM10



PM2.5



SOX



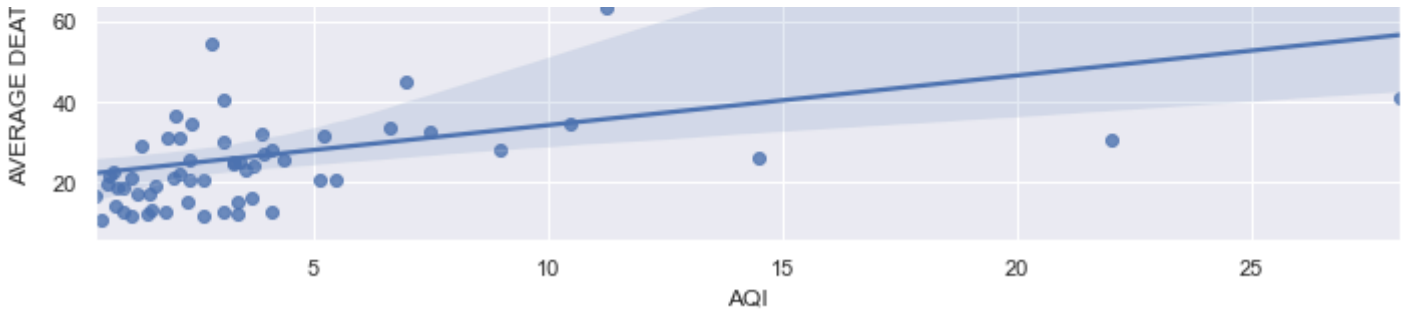


Figure 1.4.2: DEATHS against AQI for different pollutants

From the plot of small multiples, it can be observed that PM 10 and PM 2.5 display the greatest linear positive correlation among the different pollutants. Pollutants CO, SOX and NOX show very weak to negligible positive correlation. Further investigation by performing linear regression model on these pollutants will be carried out.

Interestingly, the pollutant O3 seems to indicate a weak negative correlation, meaning that a higher AQI value (worsened air quality) indicates a lower average number of deaths. This may already be due to O3 AQI values being low (ranging from 0 to 35) such that they do not have much effect on the deaths, such that the weak correlation may be insignificant.

Q2. Effect of air pollution on death / dalys cause (respiratory-related diseases/air-pollution related)

There are many respiratory related diseases such as respiratory infections, pulmonary heart disease and lung cancer which may be affected by air pollution to different extents. I would like to find out which respiratory related disease is the most adversely affected by air pollution if any (i.e. a small decrease in air quality can significantly increase the risk of having a particular respiratory related disease).

I first collate the level of different pollutants based on country and their respective AQI levels, and merge them to form one single overall pollutant index (mean AQI value), then collate the number of associated deaths due to the respective respiratory diseases. This data is compared against the number of deaths / dalys per 100 000 of population for each cause.

In [63]:
```
#formatting of who attributed data to suitable format, where each column is one death cause for question 2
who_attributeddeaths_data_copy = who_attributeddeaths_data.drop(columns=["REGION","MIN","MAX"])
who_attributeddeaths_data_copy= who_attributeddeaths_data_copy[(who_attributeddeaths_data_copy["SEX"]=="BTSX")]
who_attributeddeaths_data_copy = who_attributeddeaths_data.drop(columns=["SEX"])
who_attributeddeaths_data_copy = who_attributeddeaths_data_copy.groupby(["COUNTRY","YEAR","CAUSE"])[["VALUE"]].mean()
who_attributeddeaths_data_copy.reset_index()
who_attributeddeaths_data_copy = pd.pivot_table(who_attributeddeaths_data_copy,index=["COUNTRY","YEAR"],
                                                values=["VALUE"],columns=["CAUSE"],aggfunc="mean")
who_attributeddeaths_data_copy = who_attributeddeaths_data_copy["VALUE"]
who_attributeddeaths_data_copy = who_attributeddeaths_data_copy.drop(columns=["Total"])
who_attributeddeaths_data_qn2 = who_attributeddeaths_data_copy.drop(columns=["Lower respiratory infections"])
who_attributeddeaths_data_qn2 = who_attributeddeaths_data_qn2.rename(
    columns={"Chronic obstructive pulmonary disease":"DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)",
            "Ischaemic heart disease":"DEATH (ISCHAEMIC HEART DISEASE PER 100 000)",
            "Stroke":"DEATH (STROKE PER 100 000)",
             "Trachea, bronchus, lung cancers":"DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)"})

#taking the average over the years
death_by_cause_data_qn2 = death_by_cause_data.groupby(["COUNTRY"])[["DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)",
                                                                   "DEATH (LOWER RESPIRATORY INFECTION PER 100 000)",
                                                                   "DEATH (AIR POLLUTION PER 100 000)",
                                                                   "DEATH (HOUSEHOLD PER 100 000)",
                                                                   "DEATH (PM PER 100 000)",
                                                                   "DEATH (OZONE PER 100 000)"]].mean()
who_attributeddeaths_data_qn2 = who_attributeddeaths_data_qn2.reset_index()
who_attributeddeaths_data_qn2.drop(columns=["YEAR"],inplace=True)
death_by_cause_data_qn2 = death_by_cause_data_qn2.join(who_attributeddeaths_data_qn2.set_index("COUNTRY"),on=["COUNTRY"])
death_by_cause_data_qn2.head()
```

Out[63]:

COUNTRY	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)	DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)	DEATH (ISCHAEMIC HEART DISEASE PER 100 000)	DEATH (STROKE PER 100 000)	DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)
AFG	90.176087	109.232365	252.842551	203.967377	45.979634	5.797506	3.586667	20.926667	7.930000	0.703333
AGO	66.860067	151.220258	163.970719	131.054799	28.843601	7.204439	1.696667	5.623333	3.646667	0.103000
ALB	27.764644	30.495357	59.291351	35.145038	22.319736	2.600926	3.643333	33.930000	19.136667	4.183333
AND	23.423892	21.767718	21.150254	0.397801	18.356823	2.874569	NaN	NaN	NaN	NaN
ARE	46.630824	52.147546	84.592507	1.472571	79.261157	5.300451	1.216667	8.490000	3.096667	0.406667

In [64]:
```
#combining death cause with dalys cause
owid_dalys_air_pollution_risk_data_by_country_qn2 = owid_dalys_air_pollution_risk_data_by_country.groupby(
    ["COUNTRY"])[["DALYS (AIR POLLUTION PER 100 000)","DALYS (PM PER 100 000)","DALYS (HOUSEHOLD, SOLID FUEL PER 100 000)"]].mean(
death_by_cause_data_qn2 = death_by_cause_data_qn2.join(owid_dalys_air_pollution_risk_data_by_country_qn2,on=["COUNTRY"])
death_by_cause_data_qn2.head()
```

Out[64]:

COUNTRY	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)	DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)	DEATH (ISCHAEMIC HEART DISEASE PER 100 000)	DEATH (STROKE PER 100 000)	DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)	DALYS (AI POLLUTIOI PER 10 000
AFG	90.176087	109.232365	252.842551	203.967377	45.979634	5.797506	3.586667	20.926667	7.930000	0.703333	7435.33046
AGO	66.860067	151.220258	163.970719	131.054799	28.843601	7.204439	1.696667	5.623333	3.646667	0.103000	7235.16728
ALB	27.764644	30.495357	59.291351	35.145038	22.319736	2.600926	3.643333	33.930000	19.136667	4.183333	1596.48199
AND	23.423892	21.767718	21.150254	0.397801	18.356823	2.874569	NaN	NaN	NaN	NaN	619.00522
ARE	46.630824	52.147546	84.592507	1.472571	79.261157	5.300451	1.216667	8.490000	3.096667	0.406667	914.90956

In [65]:
```
#taking average AQI over the years
waqi_data_total_by_country = waqi_data_total_mean.groupby(["COUNTRY"])[["AQI"]].mean()
waqi_data_total_by_country.head()
```

Out[65]:

COUNTRY	AQI
ARE	37.280818
ARG	14.292142
AUS	13.681081
AUT	16.799100
BEL	16.505927

In [66]:
```
health_by_cause_pollution = death_by_cause_data_qn2.join(waqi_data_total_by_country,on="COUNTRY")
health_by_cause_pollution.head()
```

Out[66]:

COUNTRY	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)	DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)	DEATH (ISCHAEMIC HEART DISEASE PER 100 000)	DEATH (STROKE PER 100 000)	DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)	DALYS (AI POLLUTIOI PER 10 000
AFG	90.176087	109.232365	252.842551	203.967377	45.979634	5.797506	3.586667	20.926667	7.930000	0.703333	7435.33046
AGO	66.860067	151.220258	163.970719	131.054799	28.843601	7.204439	1.696667	5.623333	3.646667	0.103000	7235.16728
ALB	27.764644	30.495357	59.291351	35.145038	22.319736	2.600926	3.643333	33.930000	19.136667	4.183333	1596.48199
AND	23.423892	21.767718	21.150254	0.397801	18.356823	2.874569	NaN	NaN	NaN	NaN	619.00522
ARE	46.630824	52.147546	84.592507	1.472571	79.261157	5.300451	1.216667	8.490000	3.096667	0.406667	914.90956

In [67]:
```
health_by_cause_pollution_dropnoaqi = health_by_cause_pollution.dropna(subset=["AQI"], axis=0)
#NA values for certain death causes are not dropped as the country still has data from other death causes
#which are useful and significant, but AQI values are dropped as AQI is a required variable to compare with,
#therefore cannot be done without
health_by_cause_pollution_dropnoaqi.head()
```

Out[67]:

COUNTRY	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)	DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)	DEATH (ISCHAEMIC HEART DISEASE PER 100 000)	DEATH (STROKE PER 100 000)	DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)	DALYS (AIR POLLUTION PER 100 000)
ARE	46.630824	52.147546	84.592507	1.472571	79.261157	5.300451	1.216667	8.490000	3.096667	0.406667	914.909562
ARG	36.860913	46.649300	40.277936	7.524100	31.725161	1.270309	6.256667	12.646667	4.300000	1.750000	1012.937429
AUS	27.161935	9.463523	17.768147	0.248587	17.207894	0.360452	3.183333	8.103333	2.850000	1.543333	528.476989
AUT	19.658749	6.744753	26.575058	0.815802	23.826135	2.365360	5.440000	23.610000	4.630000	3.523333	967.855188
BEL	33.454472	22.257121	29.328649	0.302120	26.673759	2.885367	8.230000	14.660000	5.530000	4.666667	1105.054690

In [68]:
```
health_by_cause_pollution_dropnoaqi_pivot = health_by_cause_pollution_dropnoaqi.reset_index()
health_by_cause_pollution_dropnoaqi_pivot = pd.melt(health_by_cause_pollution_dropnoaqi_pivot,id_vars=["COUNTRY","AQI"],value_vars
                                                    health_by_cause_pollution_dropnoaqi_pivot.columns[1:-1],value_name="VALUE",
                                                   var_name="CAUSE OF HEALTH")
health_by_cause_pollution_dropnoaqi_pivot.head()
```

Out[68]:

	COUNTRY	AQI	CAUSE OF HEALTH	VALUE
0	ARE	37.280818	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	46.630824
1	ARG	14.292142	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	36.860913
2	AUS	13.681081	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	27.161935
3	AUT	16.799100	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	19.658749
4	BEL	16.505927	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	33.454472

In [69]:
```
plt.figure(figsize=(15,8))
plt.title("Graph of deaths associated with different causes per 100 000 "+
          "of country population against overall AQI index value")
sns.scatterplot(x="AQI",y="VALUE",hue="CAUSE OF HEALTH",data=
                health_by_cause_pollution_dropnoaqi_pivot[health_by_cause_pollution_dropnoaqi_pivot[
                    "CAUSE OF HEALTH"].str.startswith("DEATH")],style="CAUSE OF HEALTH")
plt.ylabel("DEATHS per 100 000 of population")
plt.legend(bbox_to_anchor=(1.01,1))
caption = "Figure 2.1: DEATHS different causes against AQI"
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

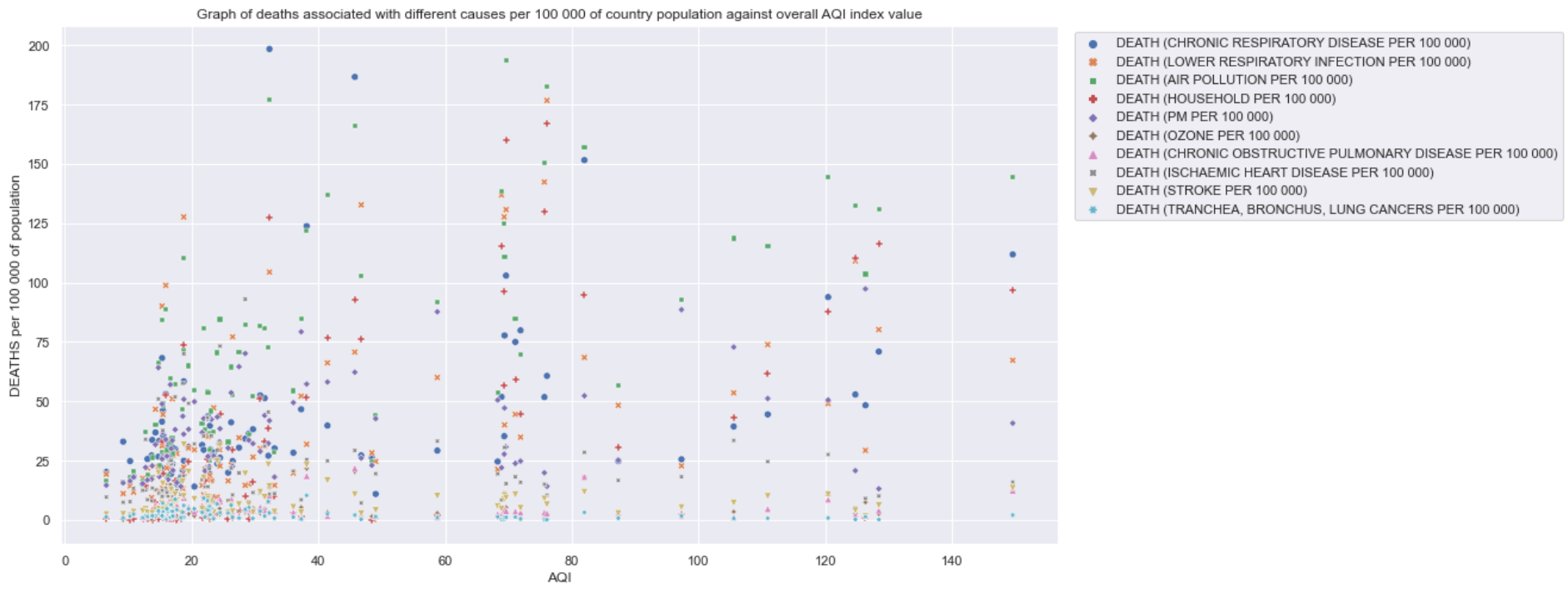


Figure 2.1: DEATHS different causes against AQI

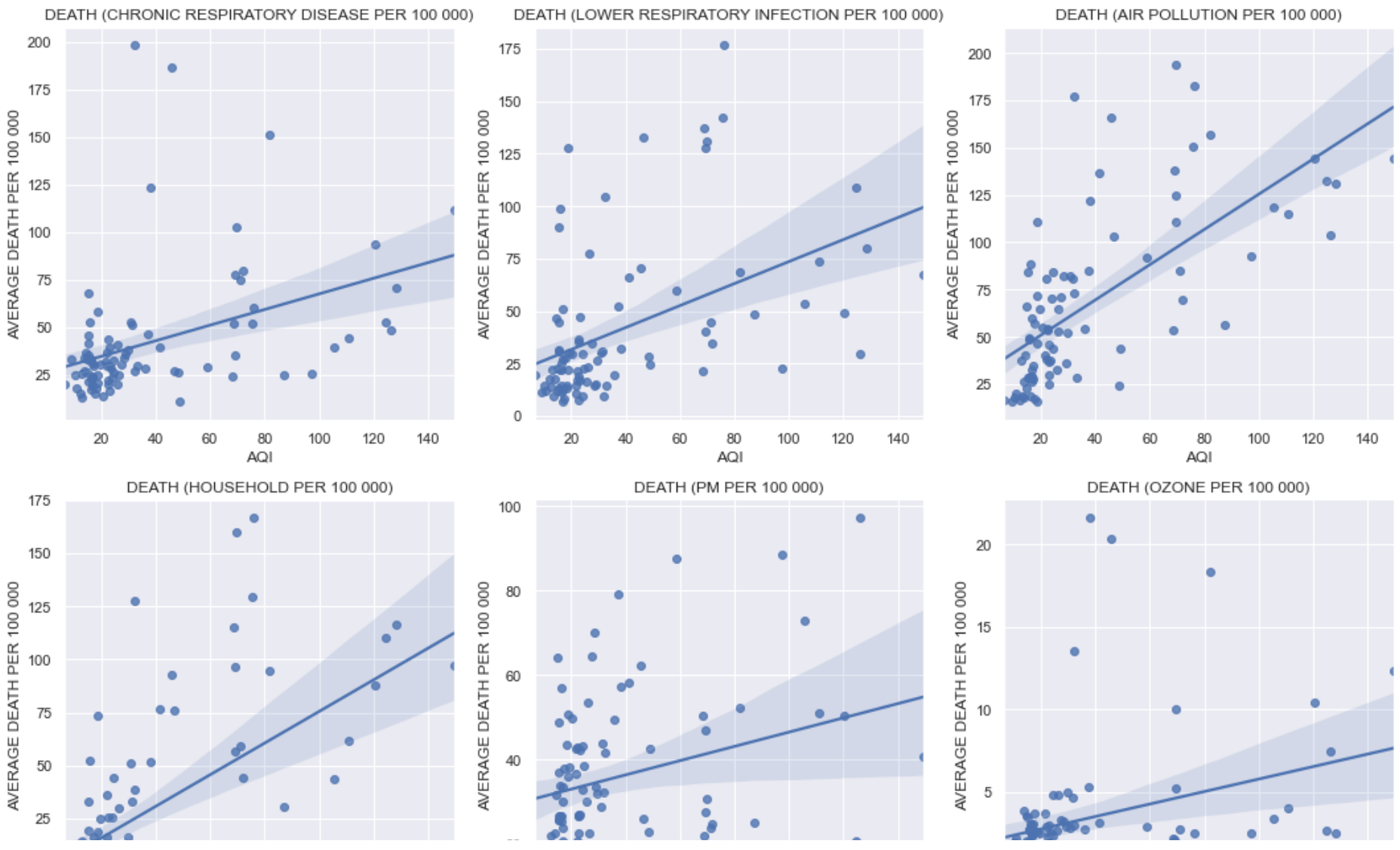
The graph shows the number of deaths per 100 000 of the country population against the AQI value, which is difficult to tell from this scatterplot any positive correlation as the points seem to be randomly scattered, with clustering at low AQI and death values. Therefore, further investigation is required to see each death cause respectively to determine whether there is a relationship.

In [70]:
```
plt.figure(figsize=(15,8))
plt.title("Graph of dalys associated with different causes per 100 000 "+
          "of country population against overall AQI index value")
sns.scatterplot(x="AQI",y="VALUE",hue="CAUSE OF HEALTH",data=
                health_by_cause_pollution_dropnoaqi_pivot[health_by_cause_pollution_dropnoaqi_pivot[
                    "CAUSE OF HEALTH"].str.startswith("DALYS")],style="CAUSE OF HEALTH")
plt.ylabel("DALYS per 100 000 of population")
plt.legend(bbox_to_anchor=(1.01,1))
caption = "Figure 2.2: DALYS different causes against AQI"
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

Figure 2.2: DALYS different causes against AQI

The graph shows the number of dalys per 100 000 of the country population against the AQI value, which is difficult to tell from this scatterplot any positive correlation as the points seem to be randomly scattered, with clustering at low AQI and dalys values. Therefore, further investigation is required to see each dalys cause respectively to determine whether there is a relationship.

In [71]:
```
fig, axes = plt.subplots(nrows=5, ncols=3,  tight_layout=True,figsize=(15,25))

cause_health_list = health_by_cause_pollution_dropnoaqi_pivot["CAUSE OF HEALTH"].unique()
for r in range(5):
    for c in range(3):
        if (r*3+c)==13:
            break
        ax = axes[r,c]
        sns.regplot(x="AQI",y="VALUE",
                    data=health_by_cause_pollution_dropnoaqi_pivot[
                        health_by_cause_pollution_dropnoaqi_pivot["CAUSE OF HEALTH"]==cause_health_list[r*3+c]],ax=ax)
        ax.set_ylabel("AVERAGE "+cause_health_list[r*3+c].split(" ")[0]+" PER 100 000")
        ax.set_title(cause_health_list[r*3+c])

fig.delaxes(axes[4,1])
fig.delaxes(axes[4,2])
caption = "Figure 2.3: DALYS/DEATHS (different causes) against AQI (small multiples)"
plt.figtext(0.5, -0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```



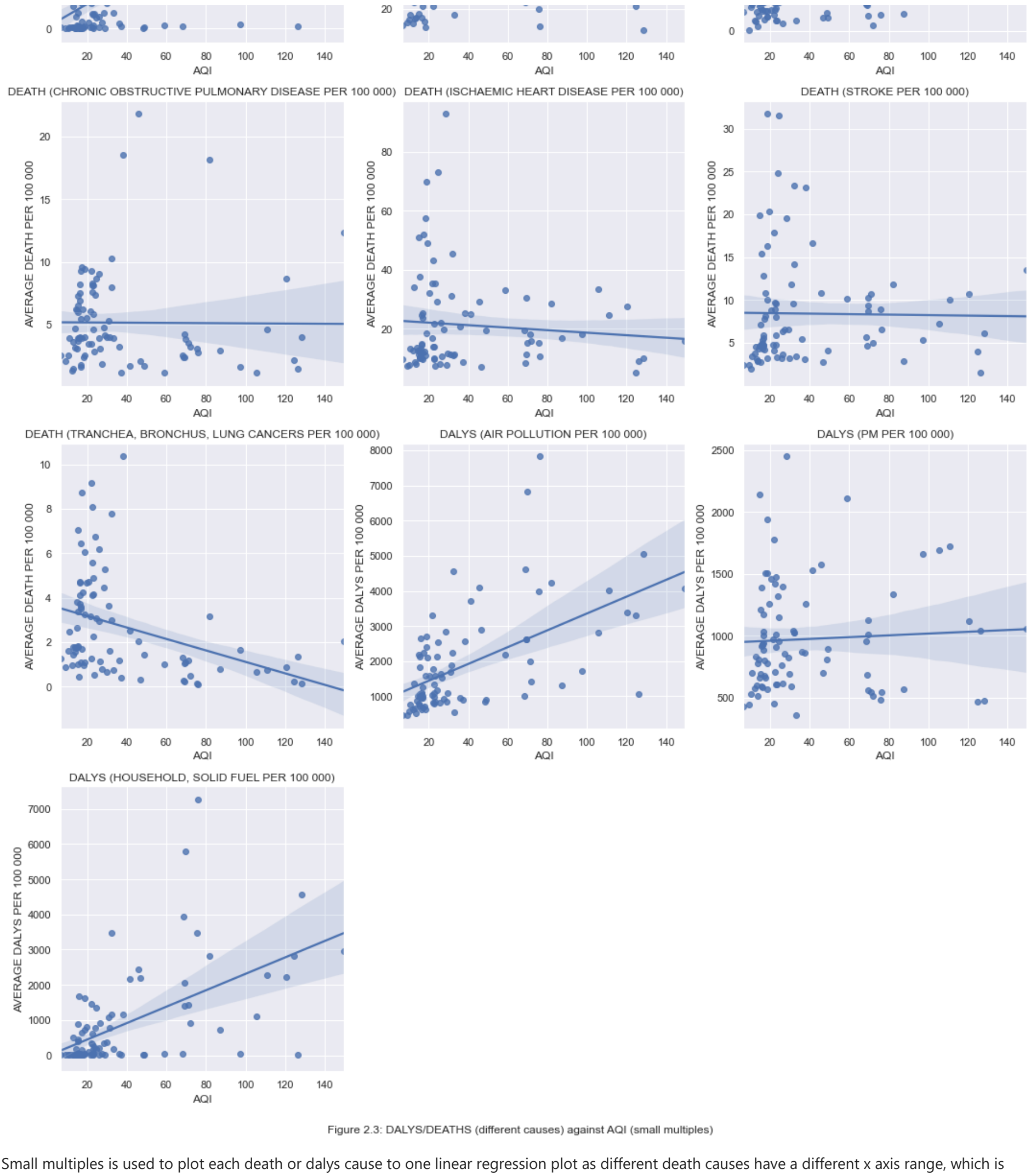


Figure 2.3: DALYS/DEATHS (different causes) against AQI (small multiples)

Small multiples is used to plot each death or dalys cause to one linear regression plot as different death causes have a different x axis range, which is also significantly different for dalys and deaths therefore a more suitable representation to scale the x axis to each death cause, showing a clearer relationship.

From the graph above, it can be observed that death (air pollution per 100 000) (sum of household, pm and ozone) against AQI, death (household per 100 000) which means death associated with household air pollution, and dalys (air pollution per 100 000) shows the most positive linear correlation. Further investigation for these causes are to be investigated by performing a linear regression model on each of them.

For death causes chronic respiratory disease, lower respiratory infection, pm, ozone, they show a very weak positive correlation between average number of deaths per 100 000 of country population against AQI.

For death causes chronic obstructive pulmonary disease, ischaemic heart disease, stroke and dalys causes pm show negligible correlation, showing that there is no relationship between AQI value and number of deaths or dalys per 100 000 of country population.

Interestingly, death causes tranchea, bronchus, lung cancers shows a negative correlation (though weakly linear), indicating that an increase in AQI value (higher AQI, worse pollution) is related to decreased number of deaths per 100 000 of a country's population. This may be due to other factors like smoking (not taken into account in air pollution), being a greater factor to the death associated with tranchea, bronchus and lung cancers.

Q3. Effect of air pollution on health of different genders

I want to find out how air pollution affects different genders differently or equally, whether there is a significant difference to which the extent of health is adversely affected based on gender alone.

I can first collate the number of males and females respectively per country affected by these respiratory related diseases. Then I can draw appropriate graphical representations for females and males for different respective countries to compare the general relation and difference between genders (if any).

In [72]:
```
#takes only data that is differentiated by genders only unlike other research questions
who_attributeddeaths_data_qn3 = who_attributeddeaths_data.drop(columns=["REGION","MIN","MAX"])
who_attributeddeaths_data_qn3= who_attributeddeaths_data_qn3[(who_attributeddeaths_data_qn3["SEX"]=="FMLE")|
                                                             (who_attributeddeaths_data_qn3["SEX"]=="MLE")]
who_attributeddeaths_data_qn3 = who_attributeddeaths_data_qn3.groupby(["COUNTRY","YEAR","SEX","CAUSE"])[["VALUE"]].mean()
who_attributeddeaths_data_qn3.reset_index()
who_attributeddeaths_data_qn3 = pd.pivot_table(who_attributeddeaths_data_qn3,index=["COUNTRY","YEAR","SEX"],
                                               values=["VALUE"],columns=["CAUSE"],aggfunc="mean")
who_attributeddeaths_data_qn3 = who_attributeddeaths_data_qn3["VALUE"]
who_attributeddeaths_data_qn3 = who_attributeddeaths_data_qn3.rename(
    columns={"Chronic obstructive pulmonary disease":"DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)",
            "Ischaemic heart disease":"DEATH (ISCHAEMIC HEART DISEASE PER 100 000)",
            "Stroke":"DEATH (STROKE PER 100 000)",
             "Lower respiratory infections":"DEATH (LOWER RESPIRATORY INFECTIONS PER 100 000)",
             "Trachea, bronchus, lung cancers":"DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)"})
who_attributeddeaths_data_qn3 = who_attributeddeaths_data_qn3.reset_index()

who_attributeddeaths_data_qn3.drop(columns=["Total","YEAR"],inplace=True)
who_attributeddeaths_data_qn3 = who_attributeddeaths_data_qn3.set_index("COUNTRY")

#only data from 2016 is taken for waqi as the dataset for who_attributeddeaths based on gender is only from the year 2016 to
#use more accurate AQI values
waqi_data_total_2016 = waqi_data_total.reset_index()
waqi_data_total_2016 = waqi_data_total_2016[waqi_data_total_2016["YEAR"]==2016]
waqi_data_total_2016.drop(columns=["YEAR"],inplace=True)
waqi_data_total_2016 = waqi_data_total_2016.groupby(["COUNTRY"])[["AQI"]].mean()

health_gender_pollution_qn3 = who_attributeddeaths_data_qn3.join(waqi_data_total_2016,on="COUNTRY")
health_gender_pollution_qn3_dropnoaqi = health_gender_pollution_qn3.dropna(subset=["AQI"], axis=0)
health_gender_pollution_qn3_dropnoaqi.head()
```

Out[72]:

COUNTRY	SEX	DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)	DEATH (ISCHAEMIC HEART DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTIONS PER 100 000)	DEATH (STROKE PER 100 000)	DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)	AQI
ARE	FMLE	0.67	4.92	0.97	2.78	0.34	18.200473
ARE	MLE	1.62	11.12	1.09	3.33	0.46	18.200473
ARG	FMLE	5.95	11.31	9.29	4.17	1.10	15.060193
ARG	MLE	6.57	14.00	8.44	4.43	2.41	15.060193
AUS	FMLE	2.93	7.02	1.30	3.38	1.24	10.120175

In [73]:
```
health_gender_pollution_qn3_dropnoaqi_pivot = health_gender_pollution_qn3_dropnoaqi.reset_index()
health_gender_pollution_qn3_dropnoaqi_pivot = pd.melt(health_gender_pollution_qn3_dropnoaqi_pivot,id_vars=["COUNTRY","AQI","SEX"],
                                                      health_gender_pollution_qn3_dropnoaqi_pivot.columns[2:-1],value_name="VALUE",
                                                     var_name="CAUSE OF HEALTH")
q3_gender_cause = sns.FacetGrid(health_gender_pollution_qn3_dropnoaqi_pivot, col="SEX",height=5,aspect=1.2,col_wrap=2)
q3_gender_cause.map_dataframe(sns.regplot, x="AQI", y="VALUE")
q3_gender_cause.fig.subplots_adjust(top=0.85)
q3_gender_cause.fig.suptitle("FacetGrid of average deaths per 100 000 against AQI for different genders")
caption = "Figure 3.1: DEATHS against AQI for different genders"
plt.figtext(0.5, -0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
```

Out[73]: Text(0.5, -0.01, 'Figure 3.1: DEATHS against AQI for different genders')

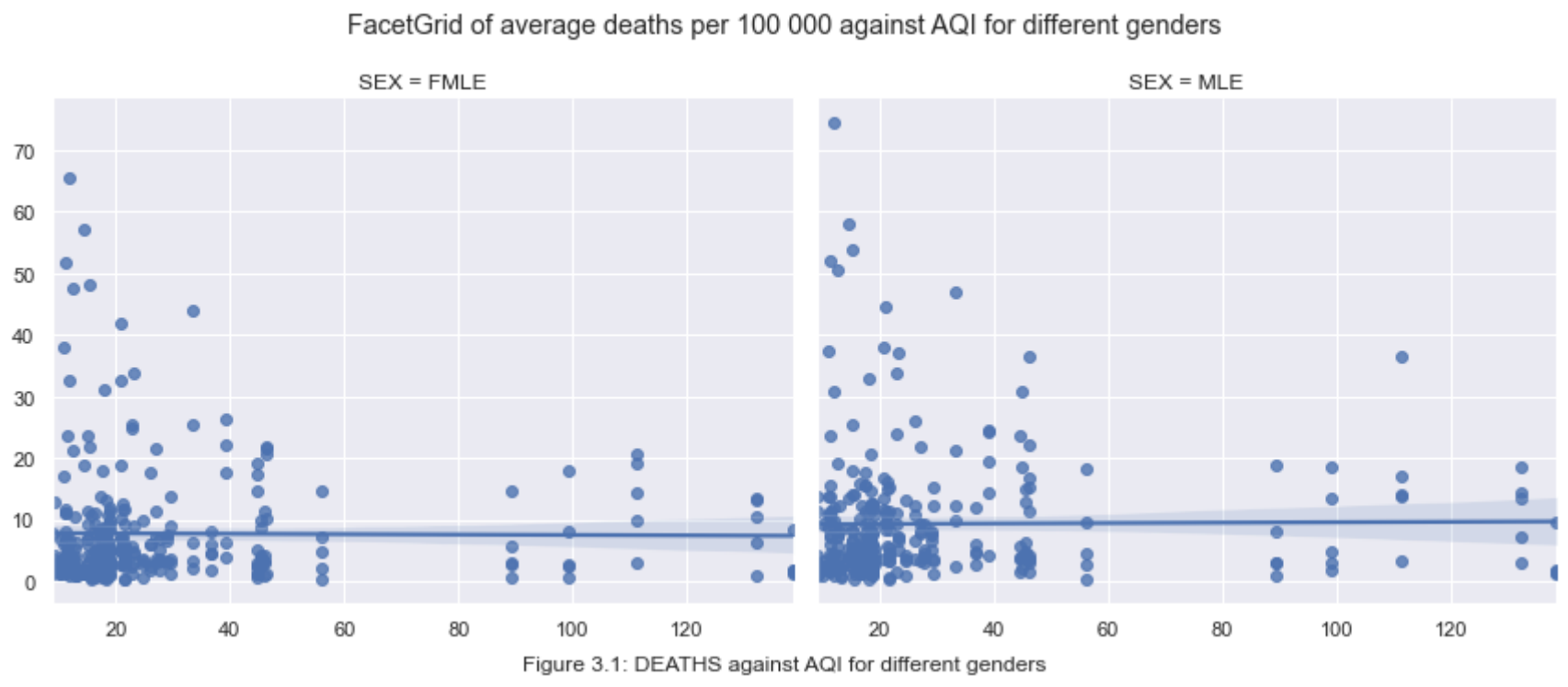


Figure 3.1: DEATHS against AQI for different genders

From this facetgrid, it can be observed that a lot of points are clustered at low AQI values and unable to see any positive correlation value. Therefore, no relationship can be seen for average number od deaths against AQI. However, since the AQI value for each country is the same (i.e. the AQI value

for each corresponding datapoint for the same country is the same), a categorical scatterplot can be plotted just for the average deaths per 100 000 for each country for further investigation and analysis.

In [74]:
```
plt.figure(figsize=(15,8))  #to set figure size to specific dimensionplt.figure(figsize=(6,7))
ax = sns.boxplot( y="VALUE", x="SEX",data = health_gender_pollution_qn3_dropnoaqi_pivot ) #x denotes the cateogry
ax = sns.stripplot( y="VALUE", x="SEX",  data=health_gender_pollution_qn3_dropnoaqi_pivot, color="tab:green",jitter=0.2, alpha=0.9
plt.title("Graph of number of deaths per 100 000 due to air pollution related causes/diseases")
plt.ylabel("Average deaths per 100 000")
#since AQI values are the same for the same datapoints
caption = "Figure 3.2: DEATHS for different genders"
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

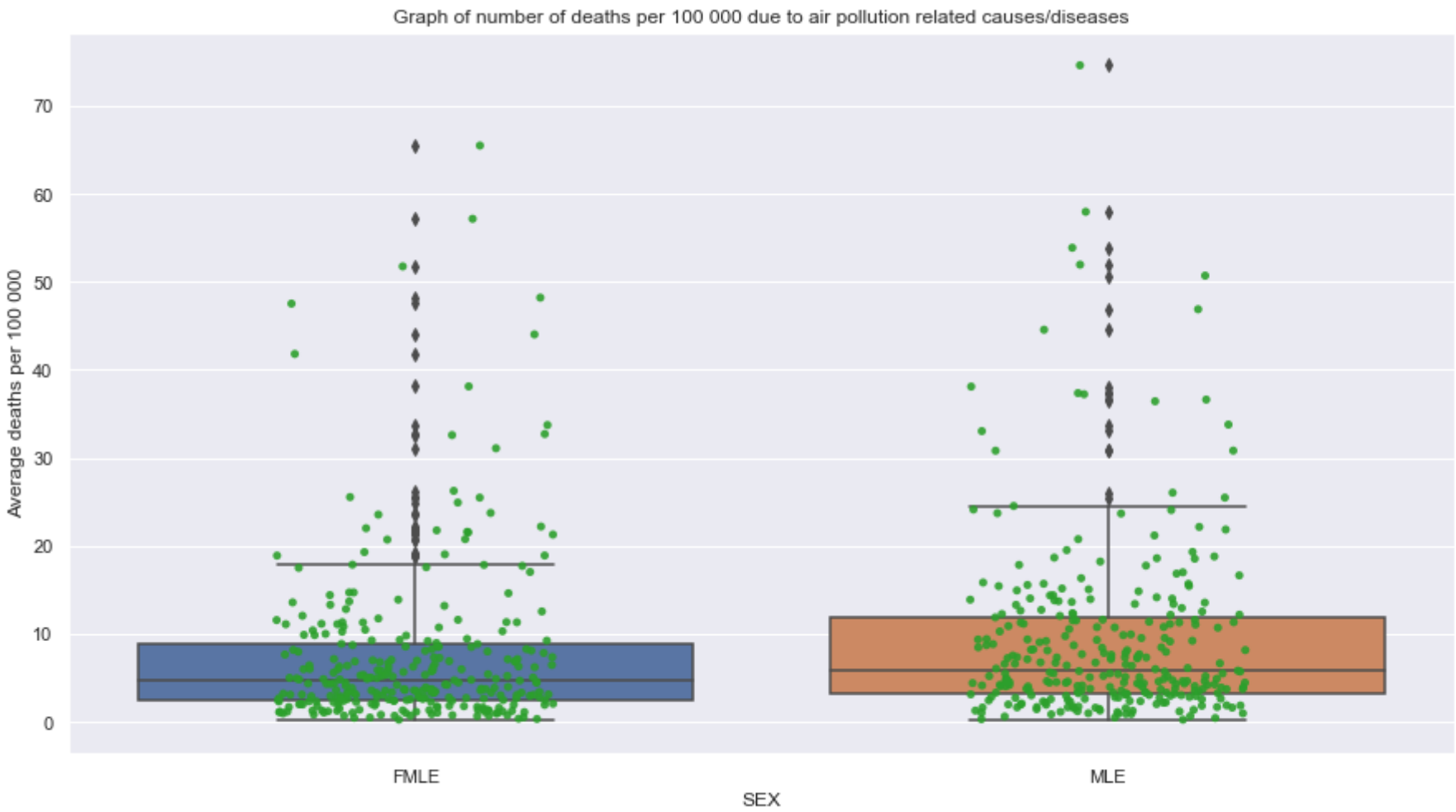


Figure 3.2: DEATHS for different genders

Based on this graph, it can be observed that the median value for females is slightly lower than males. This shows that death of females are less affected by air pollution as compared to men. Both the range and interquartile range of females is smaller than male, showing that the average number of deaths per 100 000 for females are more clustered and spread over a smaller range of values. Both females and males show several outliers above Q3 + 1.5 IQR, showing that there are few exceptions with extremely high average deaths per 100 000.

Q4. Effect of air pollution on health of people from 2014 to 2017

With the improvement of healthcare provision over the years, the health of people is expected to improve over the years. However, with the advancement of society, there has been increased air pollution. Therefore, I want to find out how air quality effect on the health of people has varied across the past few years, whether the correlation (if any) between the number of respiratory related diseases and concentration of air pollutants is stronger or weaker across different years.

I can first collate the level of different pollutants based on country, and merge them to form one single overall pollutant index, then plot a suitable graphical representation like line graph for each country to see the general trend of pollution index by year as well as overall effect on prevalence of respiratory diseases.

In [75]:
```
health_by_cause_data_q4 = death_by_cause_data.join(
    owid_dalys_air_pollution_risk_data_by_country.set_index(["COUNTRY","YEAR"]),on=["COUNTRY","YEAR"])
health_by_cause_data_q4.head()
```

Out[75]:

	COUNTRY	YEAR	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)	DALYS (AIR POLLUTION PER 100 000)	DALYS (PM PER 100 000)	DALYS (HOUSEHOLD, SOLID FUEL PER 100 000)
0	AFG	1990	95.273780	164.811829	299.477309	250.362910	46.446589	5.616442	10294.745375	1526.332080	8743.370919
1	AFG	1991	95.270656	151.460290	291.277967	242.575125	46.033841	5.603960	10045.518374	1516.305489	8504.015066
2	AFG	1992	95.584266	127.896225	278.963056	232.043878	44.243766	5.611822	10378.946878	1565.640387	8787.469185
3	AFG	1993	96.581362	124.725141	278.790815	231.648134	44.440148	5.655266	12166.184045	1837.163614	10302.271547
4	AFG	1994	98.105844	134.410918	287.162923	238.837177	45.594328	5.718922	13681.323112	2057.436937	11596.547538

In [76]:
```
death_by_cause_qn4 = death_by_cause_qn1_temp.copy()
owid_dalys_air_pollution_risk_qn4 = owid_dalys_air_pollution_risk_qn1_temp.copy()
```

```
death_by_cause_qn4 = death_by_cause_qn4.groupby(["COUNTRY","YEAR"])[["AVERAGE DEATHS PER 100 000"]].mean()
death_by_cause_qn4 = death_by_cause_qn4.reset_index()
health_by_cause_qn4 = death_by_cause_qn4.merge(owid_dalys_air_pollution_risk_qn4,on=["COUNTRY","YEAR"])

#getting the AQI data for the respective years for the respective countries
waqi_data_total_byyear = waqi_data_total.groupby(["COUNTRY","YEAR"])[["AQI"]].mean()
waqi_data_total_byyear = waqi_data_total_byyear.reset_index()

health_by_cause_qn4 = health_by_cause_qn4.merge(waqi_data_total_byyear,on=["COUNTRY","YEAR"])
health_by_cause_qn4.head()
```

Out[76]:

	COUNTRY	YEAR	AVERAGE DEATHS PER 100 000	AVERAGE DALYS PER 100 000	AQI
0	ARE	2015	41.911732	970.449827	46.011019
1	ARE	2016	41.339482	1009.697357	18.200473
2	ARG	2015	26.657648	578.053048	12.732283
3	ARG	2016	26.059127	560.352462	15.060193
4	ARG	2017	25.485870	554.612531	19.957674

In [77]:

```
plt.figure(figsize=(15,8))
sns.lmplot(x="AQI",y="AVERAGE DEATHS PER 100 000",hue="YEAR",data=health_by_cause_qn4,palette="deep",height=10)
plt.ylabel("DEATHS per 100 000 of population")
plt.title("Graph of deaths associated with different causes per 100 000 "+
          "of country population against AQI index from 2014 to 2017")
caption = "Figure 4.1: DEATHS against AQI from 2014 to 2017"
plt.figtext(0.5, -0.05, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

```
<Figure size 1080x576 with 0 Axes>
```

Figure 4.1: DEATHS against AQI from 2014 to 2017

From the graph above, it can be observed that for all years, the number of deaths per 100 000 of population against aqi show a positive correlation (i.e. higher AQI is associated with higher number of deaths per 100 000 of population). However, the linear regression plot for different years can be observed to be slightly different, showing that the extent to which AQI is positively correlated to deaths per 100 000 of population is different. It can also be observed that the datapoints are clustered around the low AQI and low number of deaths per 100 000 population for all years.

From the graph, the year 2014 shows the steepest linear regression plot, followed by 2015, 2016 then 2017. This indicates that for a smaller increase in AQI index, the extent to which the number of deaths increased in 2014 is the highest, followed by 2015, 2016 and 2017, which happens to follow chronological order.

To further observe the individual trends for the different years, a FacetGrid is plotted for each year.

In [78]:
```
q4_deaths_years = sns.FacetGrid(health_by_cause_qn4, col="YEAR",height=5,aspect=1.2,col_wrap=2)
q4_deaths_years.map_dataframe(sns.regplot, x="AQI", y="AVERAGE DEATHS PER 100 000")
q4_deaths_years.fig.subplots_adjust(top=0.9)
q4_deaths_years.set_ylabels("Average deaths per 100 000 population")
q4_deaths_years.set_xlabels("AQI")
q4_deaths_years.fig.suptitle("FacetGrid of average deaths per 100 000 against AQI from 2014 to 2017")
caption = "Figure 4.2: DEATHS against AQI from 2014 to 2017 (small multiples)"
plt.figtext(0.5, -0.05, caption, wrap=True, horizontalalignment='center', fontsize=12)
```

Out[78]: Text(0.5, -0.05, 'Figure 4.2: DEATHS against AQI from 2014 to 2017 (small multiples)')

Figure 4.2: DEATHS against AQI from 2014 to 2017 (small multiples)

From the graphical representation above, it can be seen that the datapoints for all years are clustered at the lower AQI and lower average deaths per 100 000 population value. In 2016 and 2017, there are a few datapoints with very high AQI values above 120 unlike in 2014 and 2015.

In [79]:
```
plt.figure(figsize=(15,8))
sns.lmplot(x="AQI",y="AVERAGE DALYS PER 100 000",hue="YEAR",data=health_by_cause_qn4,palette="deep",height=10)
plt.ylabel("DALYS per 100 000 of population")
plt.title("Graph of dalys associated with different causes per 100 000 "+
          "of country population against AQI index from 2014 to 2017")
caption = "Figure 4.3: DALYS against AQI from 2014 to 2017"
plt.figtext(0.5, -0.05, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

<Figure size 1080x576 with 0 Axes>

Figure 4.3: DALYS against AQI from 2014 to 2017

From the graph above, it can be observed that for all years, the number of dalys per 100 000 of population against aqi show a positive correlation (i.e. higher AQI is associated with higher number of deaths per 100 000 of population). However, the linear regression plot for different years can be observed to be slightly different, showing that the extent to which AQI is positively correlated to dalys per 100 000 of population is different.

From the graph, the year 2014 shows the steepest linear regression plot, followed by 2015, 2016 then 2017. This indicates that for a smaller increase in AQI index, the extent to which the number of dalys increased in 2014 is the highest, followed by 2015, 2016 and 2017, which happens to follow chronological order.

In [80]:
```
q4_dalys_years = sns.FacetGrid(health_by_cause_qn4, col="YEAR",height=5,aspect=1.2,col_wrap=2)
q4_dalys_years.map_dataframe(sns.regplot, x="AQI", y="AVERAGE DALYS PER 100 000")
q4_dalys_years.fig.subplots_adjust(top=0.9)
q4_dalys_years.fig.suptitle("FacetGrid of average dalys per 100 000 against AQI from 2014 to 2017")
caption = "Figure 4.4: DALYS against AQI from 2014 to 2017 (small multiples)"
plt.figtext(0.5, -0.05, caption, wrap=True, horizontalalignment='center', fontsize=12)
```

Out[80]:
```
Text(0.5, -0.05, 'Figure 4.4: DALYS against AQI from 2014 to 2017 (small multiples)')
```

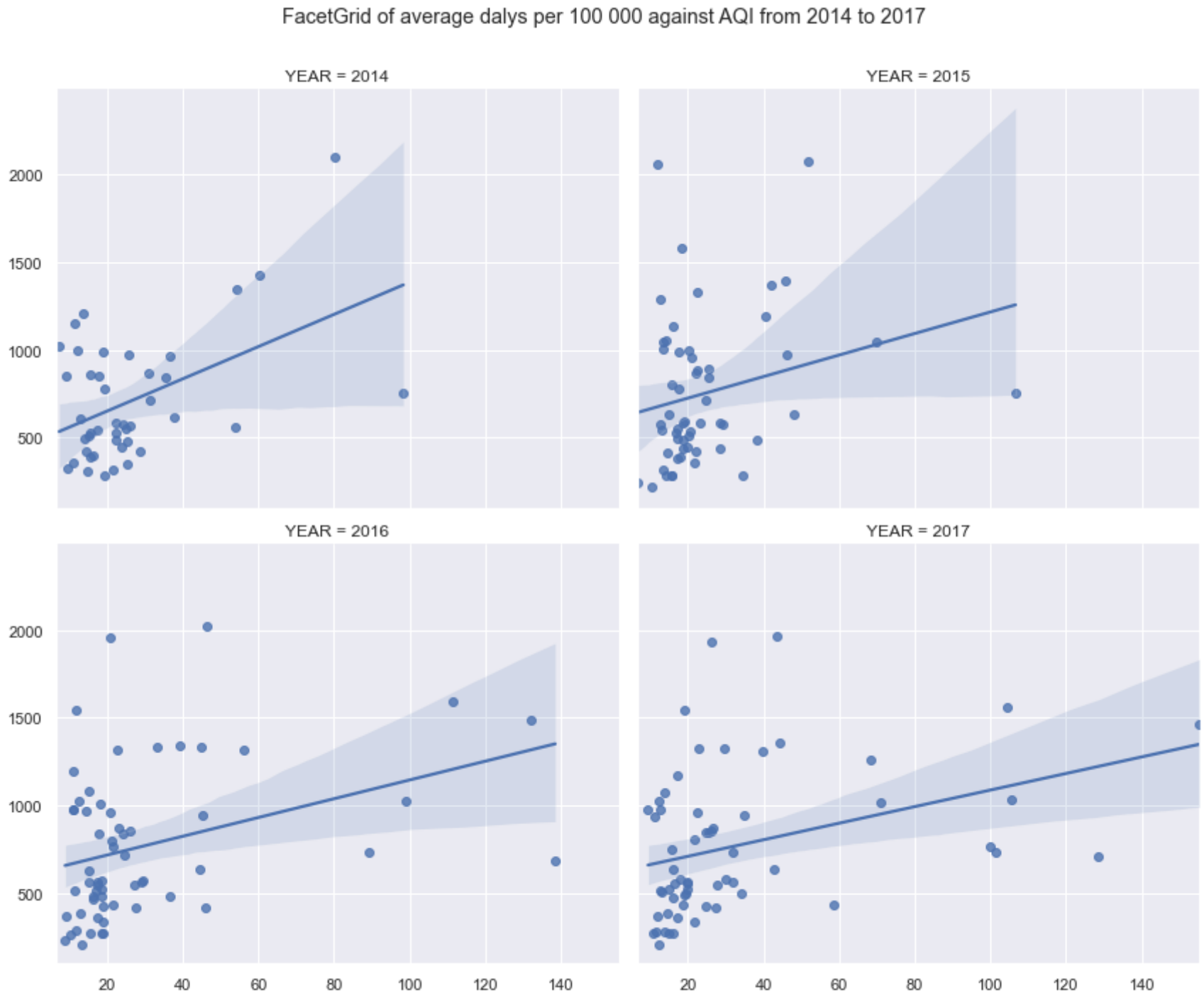


Figure 4.4: DALYS against AQI from 2014 to 2017 (small multiples)

From the graphical representation above, It can also be observed that the datapoints are clustered around the low AQI and low number of dalys per 100 000 population for all years, however with a larger variation of dalys per 100 000 of the population for the lower AQI values.

In 2016 and 2017, there are a few datapoints with very high AQI values above 120 unlike in 2014 and 2015.

To show how the "extent of effect" of average deaths per 100 000 population against AQI population over the years, a geographical representation is plotted to investigate the its effect over the years for the same country using animation to compare directly for country to country.

In [81]:
```
health_by_cause_qn4_map = health_by_cause_qn4.copy()
health_by_cause_qn4_map = health_by_cause_qn4_map.dropna(subset=["AQI"])
health_by_cause_qn4_map_index = health_by_cause_qn4_map.copy()
health_by_cause_qn4_map_index = health_by_cause_qn4_map_index.sort_values(by="YEAR",ascending=True)

# applying min max data normalization such that the AQI and average deaths scale does not affect the correlation index
# disproportionately
health_by_cause_qn4_map_index["AQI"] = (health_by_cause_qn4_map_index["AQI"] -
                                        health_by_cause_qn4_map_index["AQI"].min()
                                       ) / (health_by_cause_qn4_map_index["AQI"].max() -
                                            health_by_cause_qn4_map_index["AQI"].min())

health_by_cause_qn4_map_index["AVERAGE DEATHS PER 100 000"] = (health_by_cause_qn4_map_index["AVERAGE DEATHS PER 100 000"] -
                                        health_by_cause_qn4_map_index["AVERAGE DEATHS PER 100 000"].min()
                                       ) / (health_by_cause_qn4_map_index["AVERAGE DEATHS PER 100 000"].max() -
                                            health_by_cause_qn4_map_index["AVERAGE DEATHS PER 100 000"].min())


health_by_cause_qn4_map_index["index_deaths"] = (health_by_cause_qn4_map_index["AVERAGE DEATHS PER 100 000"]/
                                        health_by_cause_qn4_map_index["AQI"])


fig = px.choropleth(health_by_cause_qn4_map_index, locations='COUNTRY', color="index_deaths",
                            color_continuous_scale="deep",
                            locationmode='ISO-3',
                            scope="world",
                        #overriding of range color is done as certain years have 1 or 2 index_deaths outliers which skew the colour
                        #scale such that it differs greatly in shade for the same index over different years
                        #doing so also keeps the color range consistent throughout all years
                            range_color=(0,15),
                            animation_frame = "YEAR",
                            title = "Average deaths (attributed to air pollution related causes) per 100 000 across the world",
                            width = 1100,
                            height = 600
                           )
fig.update_layout(margin={"r":10,"t":30,"l":10,"b":10})
```

```
caption = "Figure 4.5: DEATHS index across world from 2014 to 2017"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.1, showarrow=False)
fig.show()
```

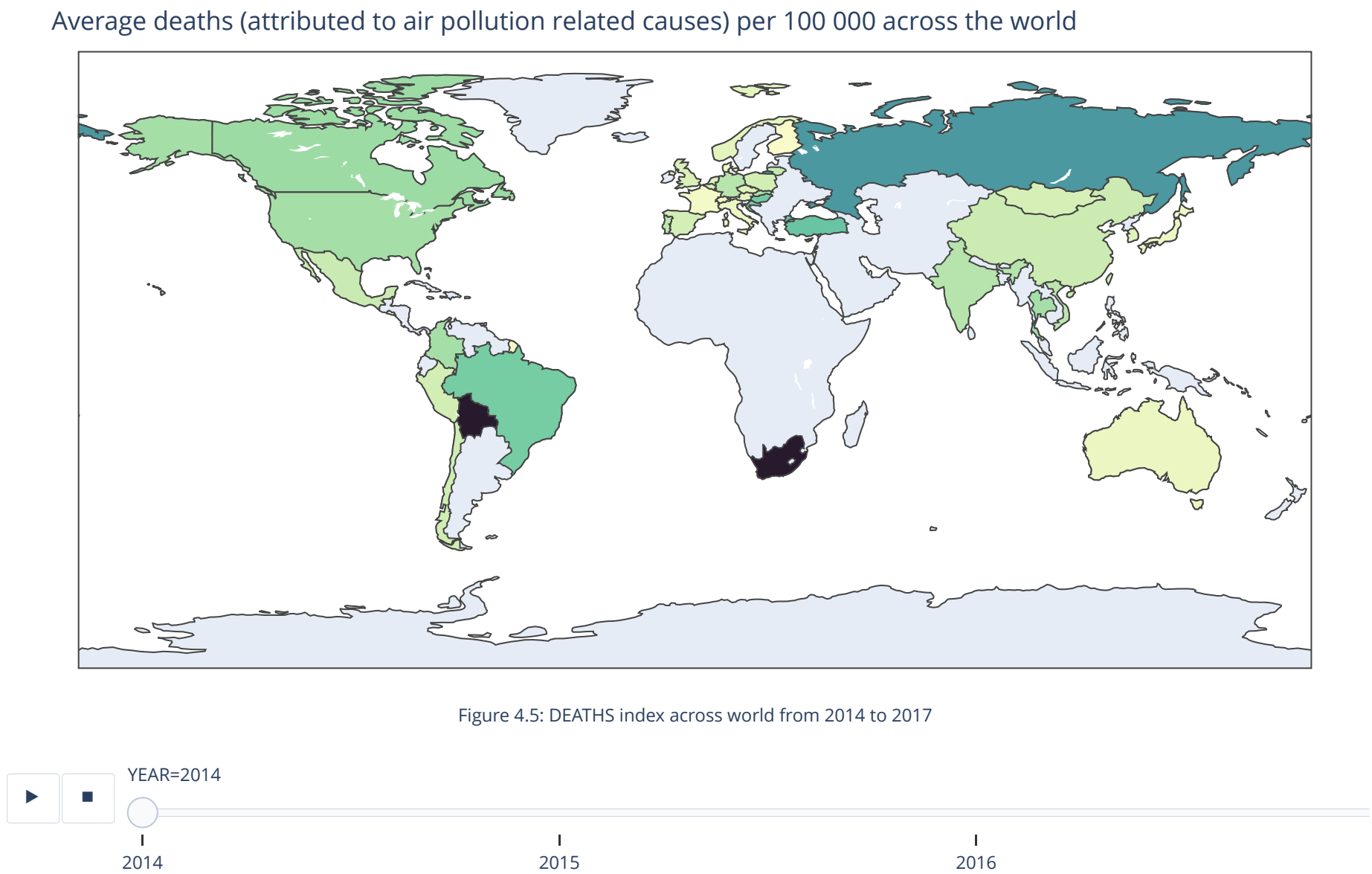


Figure 4.5: DEATHS index across world from 2014 to 2017

In 2014, the country with country code ZAF (Republic of South Africa) shows significantly darker shade, with a index of 107.12 which is calculated by the normalized number of deaths per 100 000 divided by the normalized AQI value. The index of 107.12 is exceptionally high as the typical range of index values is 0 to 10 for this graph. Country of country code BOL (Bolivia) also shows a high index (though not as high as ZAF) index of 24.

In 2015, the countries with country codes ZAF, SRB, ARG and BOL (Republic of South Africa, Serbia, Argentina and Bolivia respectively) show significantly darker shade, with ZAF having an index of 10.64, BOL having an index of 6.90, SRB having an index of 8.52 and ARG having an index of 6.71.

In 2016, country with country code ZAF remains the country with the darkest shade, having an index of 15.06.

In 2017, country with country code ZAF remains the country showing a significantly darker shade as compared to the other countries with an index of 13.46.

A higher index (darker colour shade) indicates that there is a high average number of deaths for the same AQI compared to other countries.

In [82]:

```
health_by_cause_qn4_map_index["AVERAGE DALYS PER 100 000"] = (health_by_cause_qn4_map_index["AVERAGE DALYS PER 100 000"] -
                                                              health_by_cause_qn4_map_index["AVERAGE DALYS PER 100 000"].min()
                                                             ) / (health_by_cause_qn4_map_index["AVERAGE DALYS PER 100 000"].max() -
                                                                  health_by_cause_qn4_map_index["AVERAGE DALYS PER 100 000"].min())

health_by_cause_qn4_map_index["index_dalys"] = (health_by_cause_qn4_map_index["AVERAGE DALYS PER 100 000"]/
                                                health_by_cause_qn4_map_index["AQI"])



fig = px.choropleth(health_by_cause_qn4_map_index, locations='COUNTRY', color="index_dalys",
                            color_continuous_scale="deep",
                            locationmode='ISO-3',
                            scope="world",
                            animation_frame = "YEAR",
                            title = "Average dalys (attributed to air pollution related causes) per 100 000 across the world",
                            width = 1100,
                        #overriding of range color is done as certain years have 1 or 2 index_deaths outliers which skew the colour
                        #scale such that it differs greatly in shade for the same index over different years
                        #doing so also keeps the color range consistent throughout all years
                            range_color = (0,25),
                            height = 600
                           )
caption = "Figure 4.6: DALYS index across world from 2014 to 2017"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.1, showarrow=False)
```

```
fig.update_layout(margin={"r":10,"t":30,"l":10,"b":10})
fig.show()
```

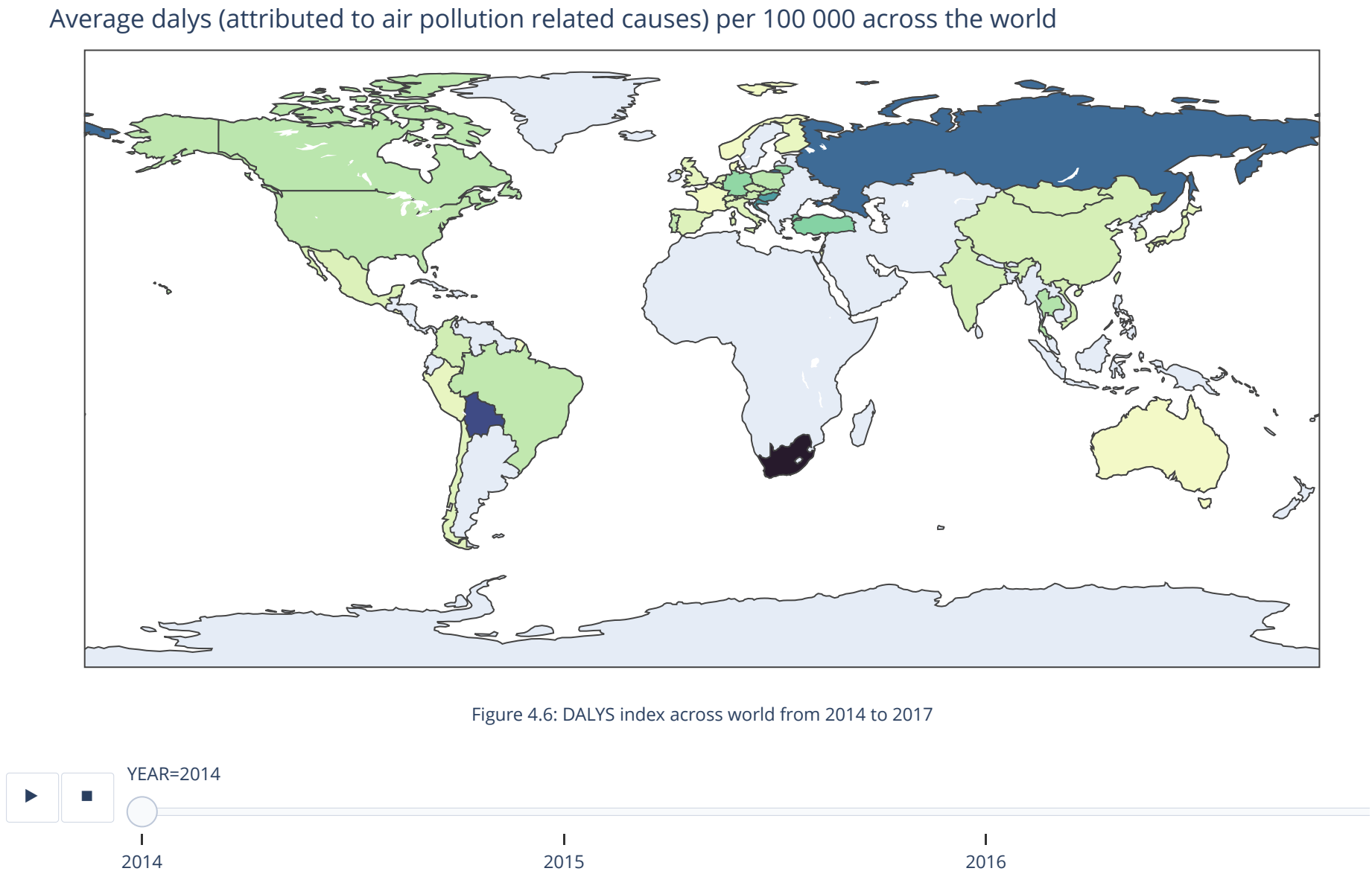


Figure 4.6: DALYS index across world from 2014 to 2017

In 2014, the country with country code ZAF (Republic of South Africa) shows significantly darker shade, with a index of 93.31 which is calculated by the normalized number of deaths per 100 000 divided by the normalized AQI value. Other significant darker shade countries (though not as high as BOL) include RUS (Russia) and BOL (Bolivia) of 15.98 and 18.95 respectively.

In 2015, the country with country codes SRB (Serbia) show significantly darker shade, with SRB having an index of 26.77.

In 2016, there country with country code BGR, ZAF, HUN and HRV (Bulgaria, Republic of South Africa, Hungary and Croatia respectively), with index of 20.13, 13.20, 16.86 and 13.77 respectively.

In 2017, the country with country code LTU and ZAF (Lithuania and Republic of South Africa) shows a significantly darker shade as compared to the other countries with an index of 12.32 and 21.07 respectively.

A higher index indicates that there is a high average number of deaths for the same AQI compared to other countries.

Q5. Effect of air pollution on health of people across different geographical location

The relationship (if any) between the number of respiratory related diseases and concentration of air pollutants may be stronger or weaker based on the geographical locations. Therefore, I would like to find out if this is true and how large of a variety it is.

I collate the level of different pollutants based on country and merge them to form one single overall pollutant index, then sum the total number of people affected by respiratory related diseases and divide it such that it is a value per 100 000 of country population. By grouping the countries and calculating the ratio of the amount of deaths / dalys per 100 000 of country population over the AQI value, we can observe which countries (based on geographical positions) have the highest index, which indicates that for a low AQI value it has a high number of deaths / dalys, which may indicate that the citizen's health is heavily affected by air pollution.

For this question, the AQI value used is not the average AQI value obtained from waqi_data_total and instead uses world_oecd_pm25_data_total which contains the pm2.5 value for the different countries instead due to:

1. The number of countries covered by waqi_data_total is insufficent to cover a good representation of the entire world map
2. From previous research questions, it can be observed that the pm2.5 aqi value is seen to correlate the most with air pollution related deaths and lives lost (dalys), therefore an appriopriate measure to be used in this question

To investigate the extent to which the air pollution affects the average number of deaths and average number of lives lost (dalys) due to air pollution related causes, an index is taken (calculated by AVERAGE DEATHS or DALYS per 100 000 / AQI value). Min-max data normalisation is then done to the respective variables (average deaths / dalys and aqi) to a value between 0 and 1 before taking the ratio.

In [83]:
```
owid_dalys_air_pollution_risk_qn5 = owid_dalys_air_pollution_risk_qn4.groupby(["COUNTRY"])[["AVERAGE DALYS PER 100 000"]].mean()
death_by_cause_qn5 = death_by_cause_qn4.groupby(["COUNTRY"])[["AVERAGE DEATHS PER 100 000"]].mean()
health_by_year_data = death_by_cause_qn5.join([owid_dalys_air_pollution_risk_qn5,world_oecd_pm25_data_total["AQI"]])
health_by_year_data = health_by_year_data.reset_index()
health_by_year_data_dropna = health_by_year_data.dropna(subset=["AQI"])
health_by_year_data_dropna_index = health_by_year_data_dropna.copy()
```

```
#applying min max data normalization
health_by_year_data_dropna_index["AQI"] = (health_by_year_data_dropna_index["AQI"] -
                                           health_by_year_data_dropna_index["AQI"].min()
                                          ) / (health_by_year_data_dropna_index["AQI"].max() -
                                               health_by_year_data_dropna_index["AQI"].min())

health_by_year_data_dropna_index["AVERAGE DALYS PER 100 000"] = (health_by_year_data_dropna_index["AVERAGE DALYS PER 100 000"] -
                                           health_by_year_data_dropna_index["AVERAGE DALYS PER 100 000"].min()
                                          ) / (health_by_year_data_dropna_index["AVERAGE DALYS PER 100 000"].max() -
                                               health_by_year_data_dropna_index["AVERAGE DALYS PER 100 000"].min())


health_by_year_data_dropna_index["index_dalys"] = (health_by_year_data_dropna_index["AVERAGE DALYS PER 100 000"]/
                                           health_by_year_data_dropna_index["AQI"])


fig = px.choropleth(health_by_year_data_dropna_index, locations='COUNTRY', color="index_dalys",
                            color_continuous_scale="deep",
                            locationmode='ISO-3',
                            scope="world",
                            title = "Average dalys (attributed to air pollution related causes) per 100 000 across the world",
                            width = 1100,
                            height = 600
                           )
caption = "Figure 5.1: DALYS index across world"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.01, showarrow=False)
fig.update_layout(margin={"r":10,"t":30,"l":10,"b":10})
fig.show()
```

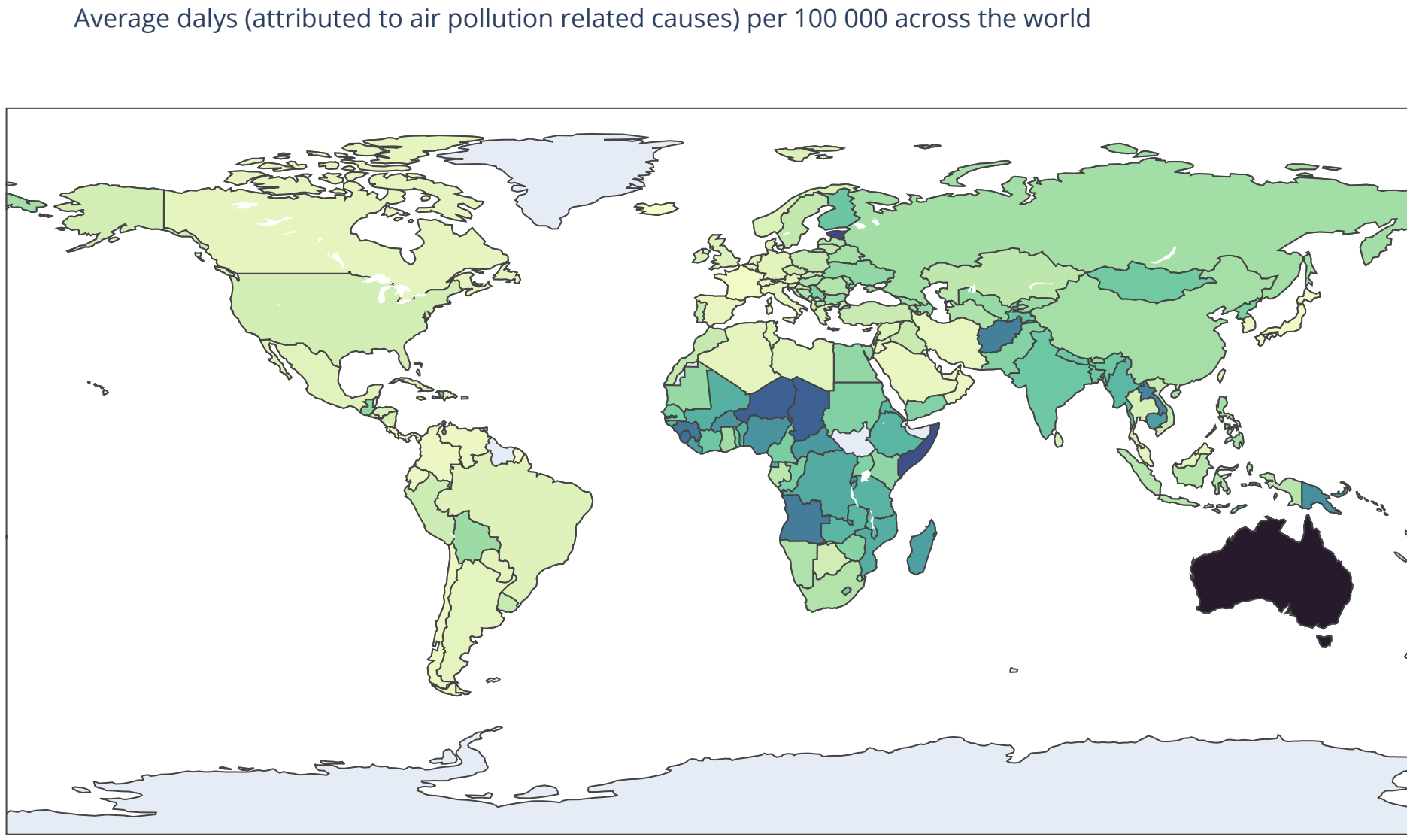


Figure 5.1: DALYS index across world

From this geographical graph, it can be seen that North and South America has relatively low index as shown by the light colour. Countries in Africa show a darker shade of colour, indicating that the index is relatively higher.

From the graph, it can also be seen that (coutry code of AUS) Australia has the highest index for dalys of 1.48, indicating it has a high number of dalys per 100 000 of country population to AQI index ratio.

In [84]:

```
health_by_year_data_dropna = health_by_year_data.dropna(subset=["AQI"])
health_by_year_data_dropna_index = health_by_year_data_dropna.copy()
#applying min max data normalization
health_by_year_data_dropna_index["AQI"] = (health_by_year_data_dropna_index["AQI"] -
                                           health_by_year_data_dropna_index["AQI"].min()
                                          ) / (health_by_year_data_dropna_index["AQI"].max() -
                                               health_by_year_data_dropna_index["AQI"].min())

health_by_year_data_dropna_index["AVERAGE DEATHS PER 100 000"] = (health_by_year_data_dropna_index["AVERAGE DEATHS PER 100 000"] -
                                           health_by_year_data_dropna_index["AVERAGE DEATHS PER 100 000"].min()
                                          ) / (health_by_year_data_dropna_index["AVERAGE DEATHS PER 100 000"].max() -
                                               health_by_year_data_dropna_index["AVERAGE DEATHS PER 100 000"].min())
```

```
health_by_year_data_dropna_index["index_deaths"] = (health_by_year_data_dropna_index["AVERAGE DEATHS PER 100 000"]/
                                                    health_by_year_data_dropna_index["AQI"])


fig = px.choropleth(health_by_year_data_dropna_index, locations='COUNTRY', color="index_deaths",
                            color_continuous_scale="deep",
                            locationmode='ISO-3',
                            scope="world",
                       #animation_frame =
                            title = "Average deaths (attributed to air pollution related causes) per 100 000 across the world",
                            width = 1100,
                            height = 600
                           )
caption = "Figure 5.2: DEATHS index across world"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.01, showarrow=False)
fig.update_layout(margin={"r":10,"t":30,"l":10,"b":10})
fig.show()
```

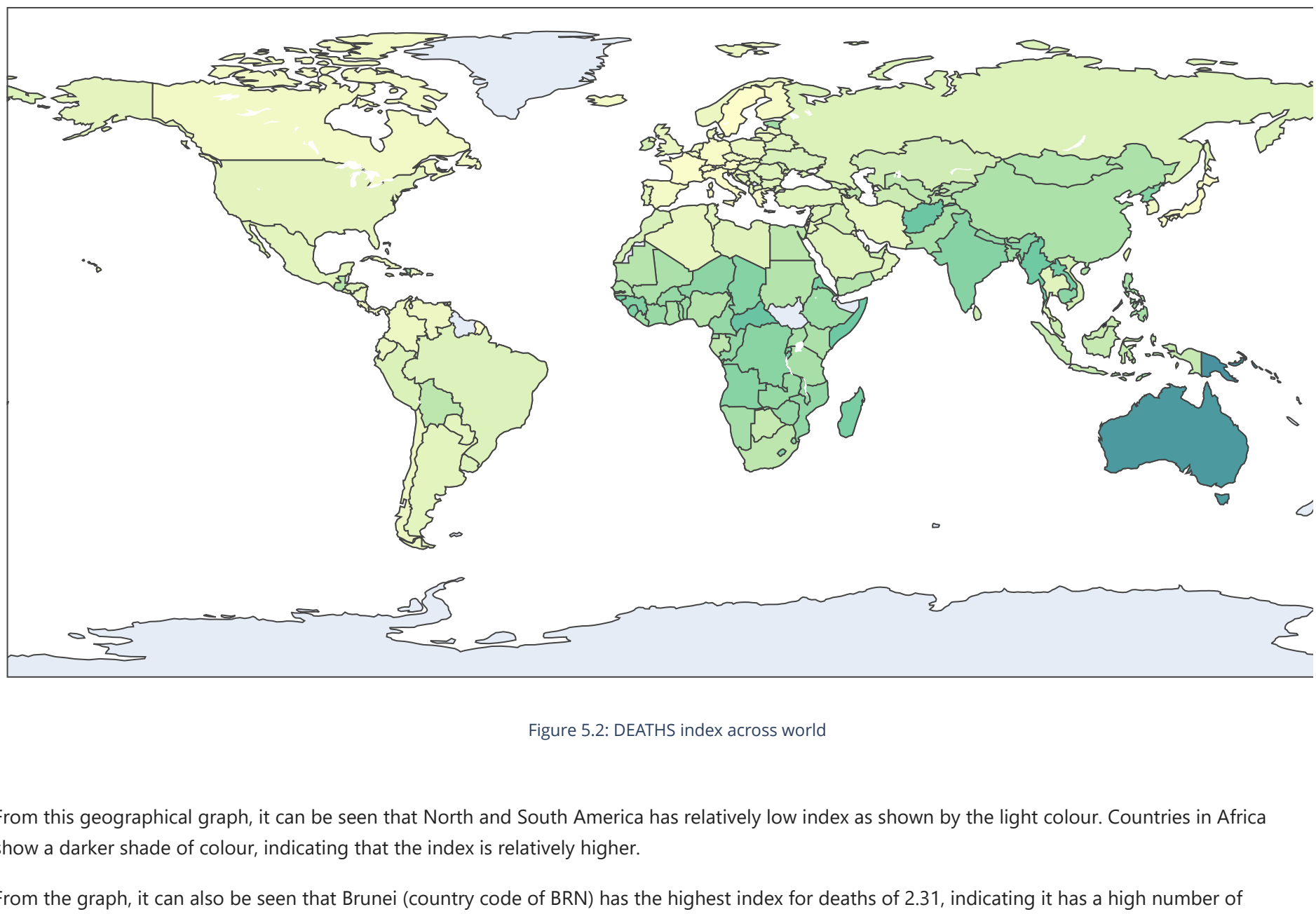


Figure 5.2: DEATHS index across world

From this geographical graph, it can be seen that North and South America has relatively low index as shown by the light colour. Countries in Africa show a darker shade of colour, indicating that the index is relatively higher.

From the graph, it can also be seen that Brunei (country code of BRN) has the highest index for deaths of 2.31, indicating it has a high number of deaths per 100 000 of country population to AQI index ratio.

Australia (country code of AUS) still has a significantly high index (though slightly lower than that for dalys) of 1.04.

Q6. Trend of air pollution/health factors across different years

In [85]:

```
death_by_cause_data_qn6 = pd.melt(death_by_cause_data,id_vars=["COUNTRY","YEAR"],value_vars=
                                                    death_by_cause_data.columns[2:],value_name="DEATH",
                                                   var_name="CAUSE OF DEATH")
death_by_cause_data_qn6 = death_by_cause_data_qn6.groupby(["COUNTRY","YEAR","CAUSE OF DEATH"],as_index=False)[["DEATH"]].mean()

death_by_year_qn6 = death_by_cause_data_qn6.groupby(["COUNTRY","YEAR"],as_index=False)[["DEATH"]].sum()
death_by_year_qn6 = death_by_year_qn6.sort_values(by="YEAR",ascending=True)
fig = px.choropleth(death_by_year_qn6, locations='COUNTRY', color="DEATH",
                            color_continuous_scale="RdBu_r",
                            locationmode='ISO-3',
                            scope="world",
                            animation_frame="YEAR",
                            title = "Average deaths (attributed to air pollution related causes) per 100 000 across the world",
                            labels={'DEATH':'Average deaths per 100 000'},
                            range_color = (0,1300),
                            width = 1100,
                            height = 600
                           )
caption = "Figure 6.1: DEATHS across world over the years"
```

```
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.1, showarrow=False)
fig.update_layout(margin={"r":10,"t":30,"l":10,"b":10})
fig.show()
```

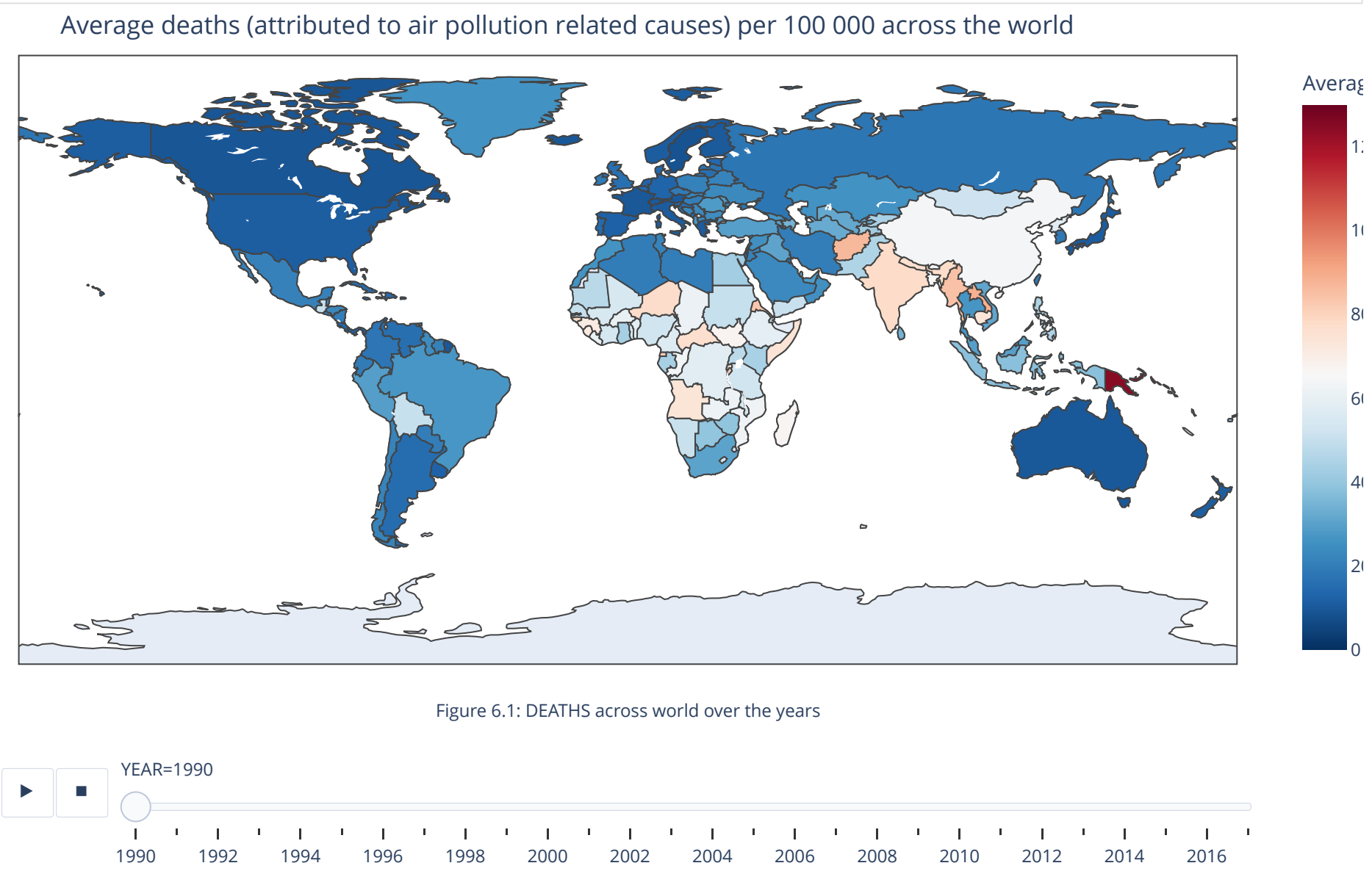
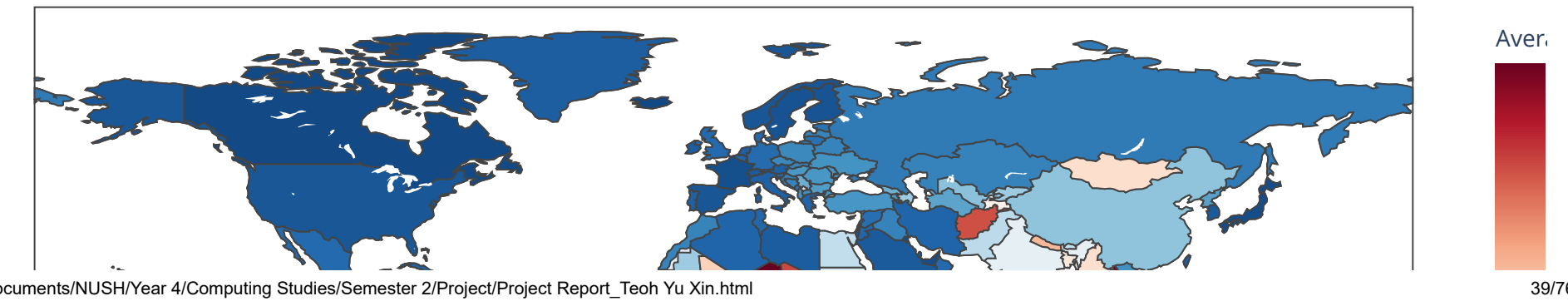


Figure 6.1: DEATHS across world over the years

This shows the average deaths per 100 000 of country population over the years. By playing the animation, the colour shade of each country can be observed to change. When dragging the animation bar over the years, it can be observed that in general, the colours turned a darker shade of blue / lighter shade of red as the years progress, indicating that the average number of deaths per 100 000 of the country population has been decreasing progressively.

It can also be noted that Papau New Guinea (country code of PNG) has stayed the country with the highest average deaths per 100 000 of population over the years, having the darkest shade of red while majority of the other countries remain different shades of blue.

In [86]:
```
dalys_by_cause_data_qn6 = pd.melt(owid_dalys_air_pollution_risk_data_by_country,id_vars=["COUNTRY","YEAR"],value_vars=
                                                owid_dalys_air_pollution_risk_data_by_country.columns[2:],value_name="DALYS",
                                               var_name="CAUSE OF DALYS")

dalys_by_year_qn6 = dalys_by_cause_data_qn6.groupby(["COUNTRY","YEAR"],as_index=False)[["DALYS"]].sum()
dalys_by_year_qn6 = dalys_by_year_qn6.sort_values(by="YEAR",ascending=True)

fig = px.choropleth(dalys_by_year_qn6, locations='COUNTRY', color="DALYS",
                            color_continuous_scale="RdBu_r",
                            locationmode='ISO-3',
                            scope="world",
                            animation_frame="YEAR",
                            title = "Average dalys (attributed to air pollution related causes) per 100 000 across the world",
                            labels={'DALYS':'Average dalys per 100 000'},
                            range_color = (0,25000),
                            width = 1100,
                            height = 600
                           )
caption = "Figure 6.2: DALYS across world over the years"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.1, showarrow=False)
fig.update_layout(margin={"r":10,"t":30,"l":10,"b":10})
fig.show()
```



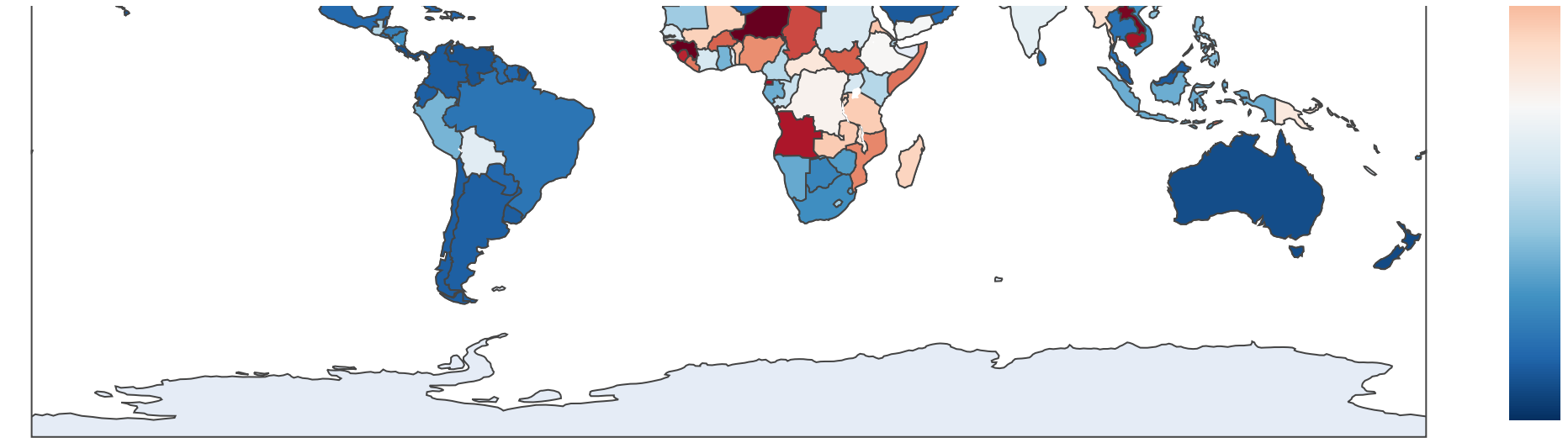
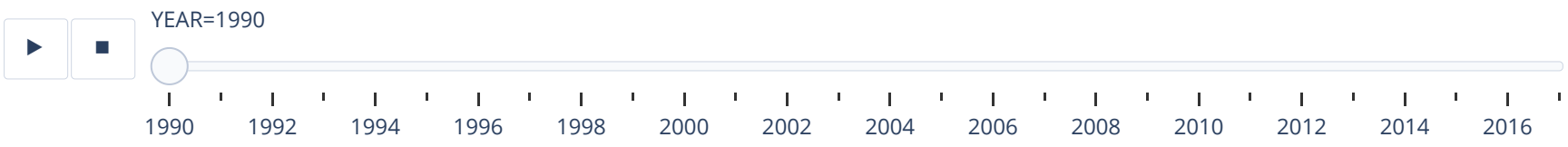


Figure 6.2: DALYS across world over the years



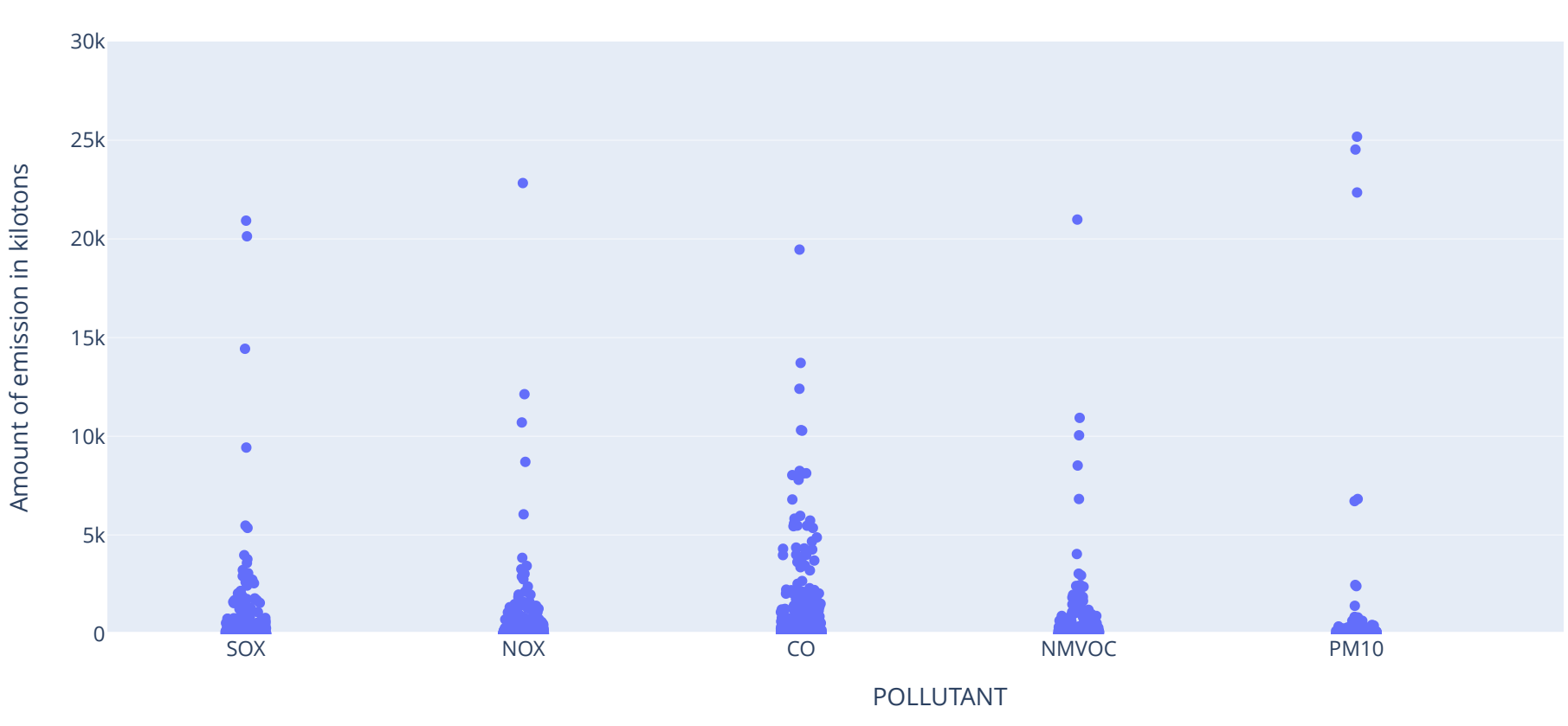
This shows the average dalys per 100 000 of country population over the years. By playing the animation, the colour shade of each country can be observed to change. When dragging the animation bar over the years, it can be observed that in general, the colours turned a darker shade of blue / lighter shade of red as the years progress (a more obvious change in shade as compared to average deaths), indicating that the average number of dalys per 100 000 of the country population has been decreasing.

It can also be noted several African countries have high average dalys per 100 000 in 1990, with countries SSD (South Sudan), TCD (Chad), AGO (Angola), GNQ (Equatorial Guinea), NER (Niger), GIN (Guinea), SLE (Sierra Leone) and other countries like AFG (Afghanistan), KHM (Cambodia), LAO (Laos) having average dalys of above 20k.

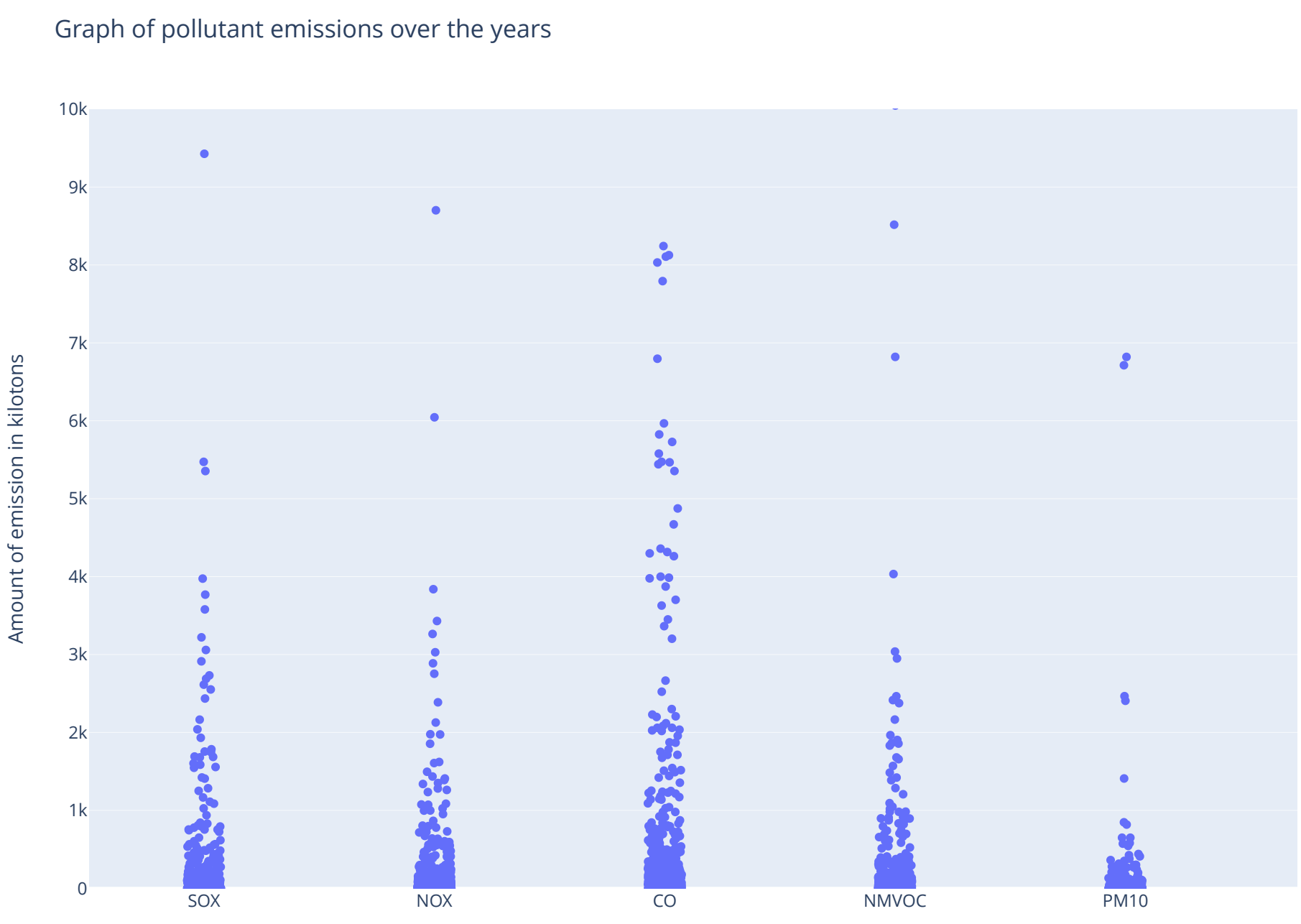
The graph below has been scaled to fit a range of y from 0 to 30k as for auto scaling, it will scale up to 80k in the earlier years due to outliers from the pollutant CO that make it hard to observe the trend of actual amount of emissions for the lower values as they are all clustered at too low values.

In [87]:
```
world_oecd_pollutants_data_qn6 = world_oecd_pollutants_data.drop(columns=["VAR","Unit Code"])
fig = px.strip(data_frame=world_oecd_pollutants_data_qn6, x="POLLUTANT", y="VALUE",
               animation_frame="YEAR",labels={
                      "VALUE": "Amount of emission in kilotons"},
                 title="Graph of pollutant emissions over the years",
                  range_y=(0,30000),
                 height=600,
                 hover_name="COUNTRY")
caption = "Figure 6.3.1: Emissions for different pollutants over the years"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.2, showarrow=False)
fig.show()
```

Graph of pollutant emissions over the years



In [88]:
```
world_oecd_pollutants_data_qn6 = world_oecd_pollutants_data.drop(columns=["VAR","Unit Code"])
fig = px.strip(data_frame=world_oecd_pollutants_data_qn6, x="POLLUTANT", y="VALUE",
               animation_frame="YEAR",labels={
                      "VALUE": "Amount of emission in kilotons"},
                 title="Graph of pollutant emissions over the years",
                height=800,
                range_y=(0,10000),
                hover_name="COUNTRY")
caption = "Figure 6.3.2: Emissions for different pollutants over the years (expanded)"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.15, showarrow=False)
fig.show()
```



From the animation above, it can be observed that the number of outliers (able to see distinct points for amount of emissions above the cluster) decreases over the years.

Generally, the pollutant CO has the largest range of amount of emissions. By hovering the mouse over the points, it can be observed that USA (United sates of America) is the country with one of the most amount of emissions for all different pollutants (from figure 6.3.1).

Majority of the pollutants are clustered in the 0k to 5k kilotons of emissions.

In [89]:
```
waqi_data_total_mean_q6 = waqi_data_total_mean.reset_index()
waqi_data_total_mean_q6 = waqi_data_total_mean_q6.sort_values(by=["POLLUTANT","YEAR"])
fig = px.strip(data_frame=waqi_data_total_mean_q6, x="POLLUTANT", y="AQI",
               animation_frame="YEAR",
                 title="Graph of aqi of pollutants over the years",
                hover_name="COUNTRY")
caption = "Figure 6.4: AQI of different pollutants over the years"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.25, showarrow=False)
fig.show()
```

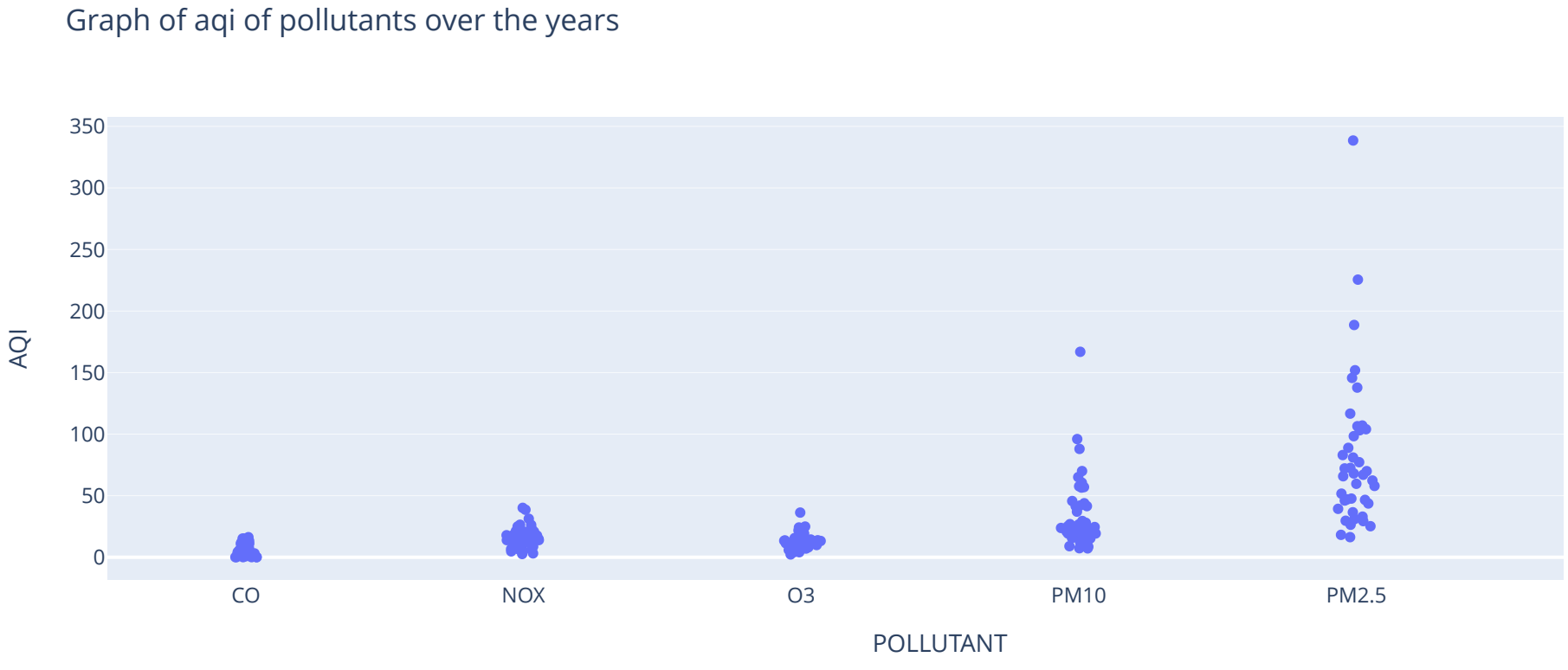


Figure 6.4: AQI of different pollutants over the years

From the graph above, it can be observed that the pollutant PM 2.5 has more spread out AQI values, followed by PM10, while O3, NOX and CO are mostly clustered at very low AQI values ranging between 0 to 50.

In 2014, it is evident that there is an outlier for pollutant PM 2.5 of extremely high AQI value of 338.5 for the country Denmark (country code of DNK) and an outlier for pollutant PM10 of high AQI value of 166.875 for the country India (country code of IND).

In [90]:
```
death_by_cause_data_qn6_xaxis = death_by_cause_data_qn6["CAUSE OF DEATH"].str.split("PER|\\(")
death_by_cause_data_qn6["CAUSE OF DEATH"] = death_by_cause_data_qn6_xaxis.str.get(1)
fig = px.strip(data_frame=death_by_cause_data_qn6, x="CAUSE OF DEATH", y="DEATH",
               animation_frame="YEAR",labels={
                      "DEATH": "Number of deaths per 100 000 due to different causes"},
                title="Graph of average deaths per 100 000 of country population for various death causes over time",height=800,
                hover_name="COUNTRY")
caption = "Figure 6.5.1: DEATH for different causes over the years"
# plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=1.05, showarrow=False)
fig.show()
```

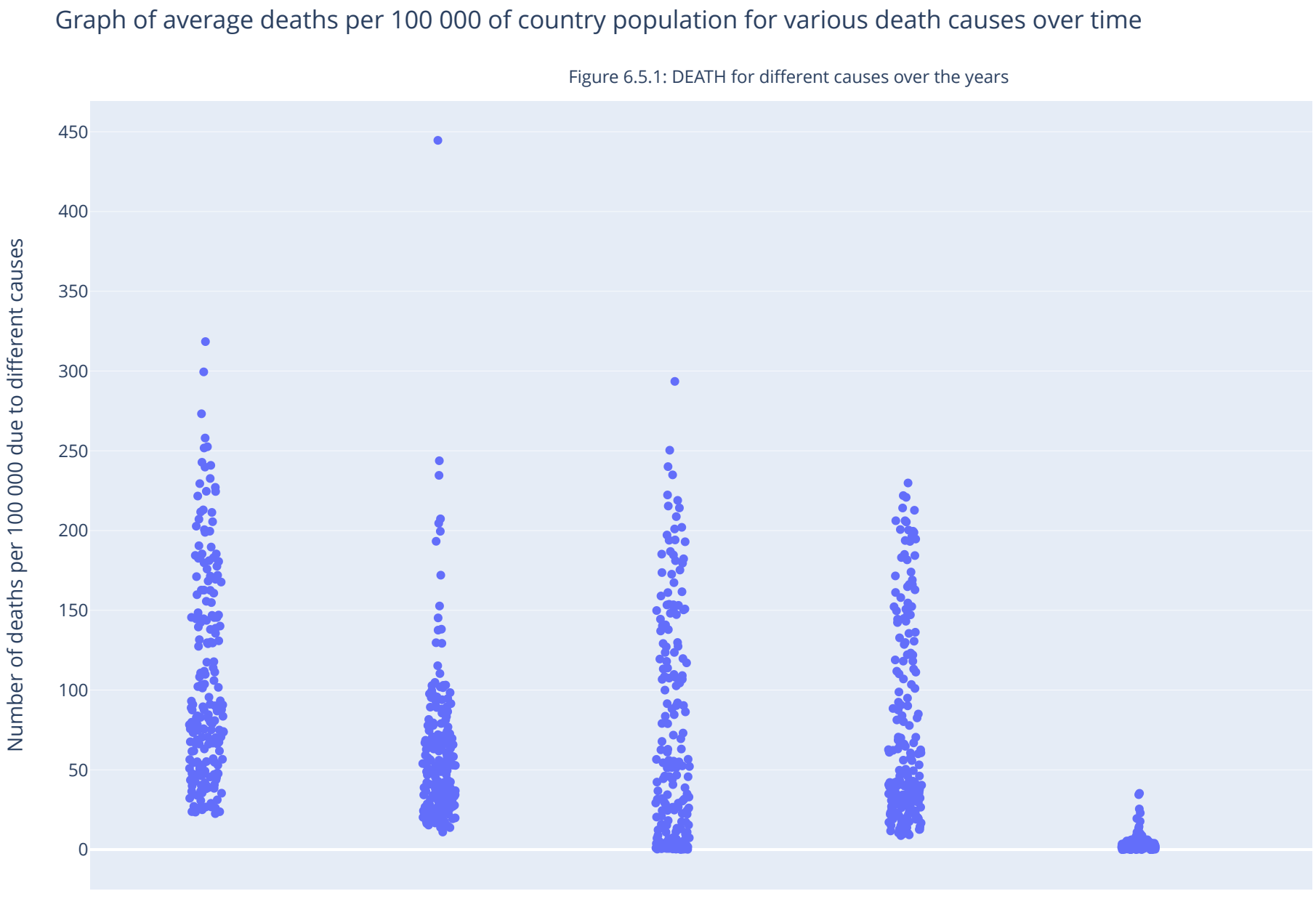


Figure 6.5.1: DEATH for different causes over the years

AIR POLLUTION CHRONIC RESPIRATORY DISEASE HOUSEHOLD LOWER RESPIRATORY INFECTION OZONE

From the graph above, it can be observed that the death causes ozone and pm have lower number of deaths per 100 000 generally, with the majority of datapoints clustered at 0 to 50 number of deaths per 100 000 of country population for ozone, and between 0 to 150 deaths per 100 000 of country population over the years. For causes air pollution, chronic respiratory disease, household and lower respiratory infection, the range of number of deaths per 100 000 have a larger range.

For the chronic respiratory disease cause, there is one consistent outlier from 1995 to 2016 which remains the highest number of deaths which is the country Papau New Guinea (PNG). It can also be observed the Papau New Guinea also has the highest number of deaths for causes air pollution and household over the years.

In [91]:
```
plt.figure(figsize=(32,8))
year_death_cause_qn4 = health_by_cause_data_q4.drop(columns=["COUNTRY"])
year_death_cause_qn4 = year_death_cause_qn4.groupby(["YEAR"])[year_death_cause_qn4.columns[1:]].mean()
year_death_cause_qn4 = year_death_cause_qn4.T
year_death_cause_qn4 = year_death_cause_qn4.reset_index()
year_death_cause_qn4_yaxis = year_death_cause_qn4["index"].str.split("PER|\\(")
year_death_cause_qn4["index"] = year_death_cause_qn4_yaxis.str.get(1)
year_death_cause_qn4 = year_death_cause_qn4.set_index("index")
sns.heatmap(year_death_cause_qn4.iloc[:6],annot=True,fmt=".2f",cmap="YlGnBu")
plt.ylabel("Cause of death")
plt.title("Heatmap of average deaths per 100 000 for different death causes over the years")
caption = "Figure 6.5.2: DEATH for different causes over the years (heatmap)"
plt.figtext(0.425, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

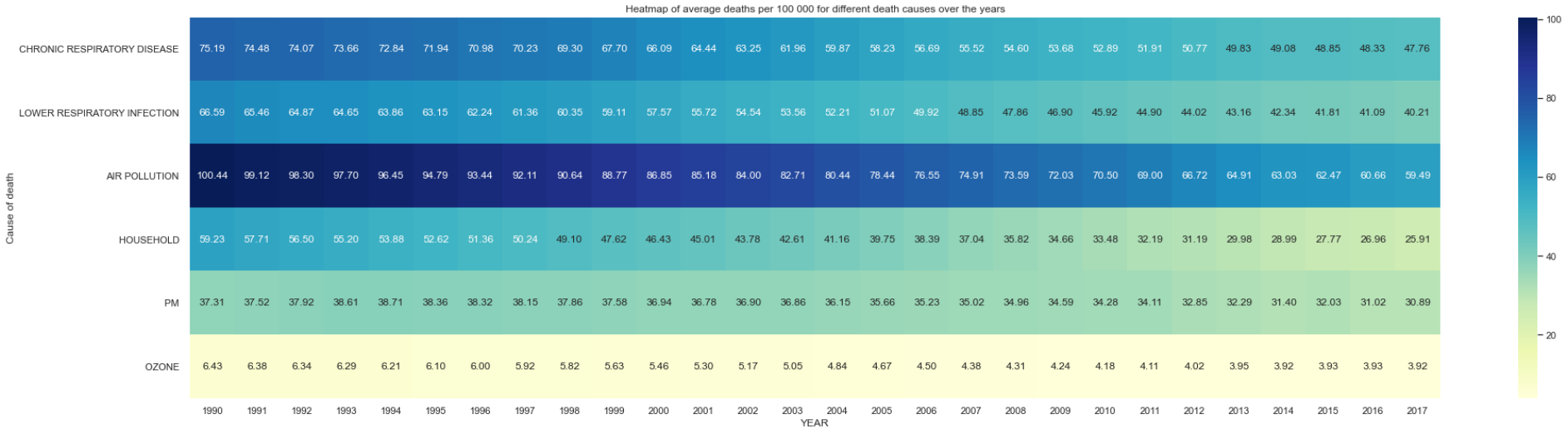
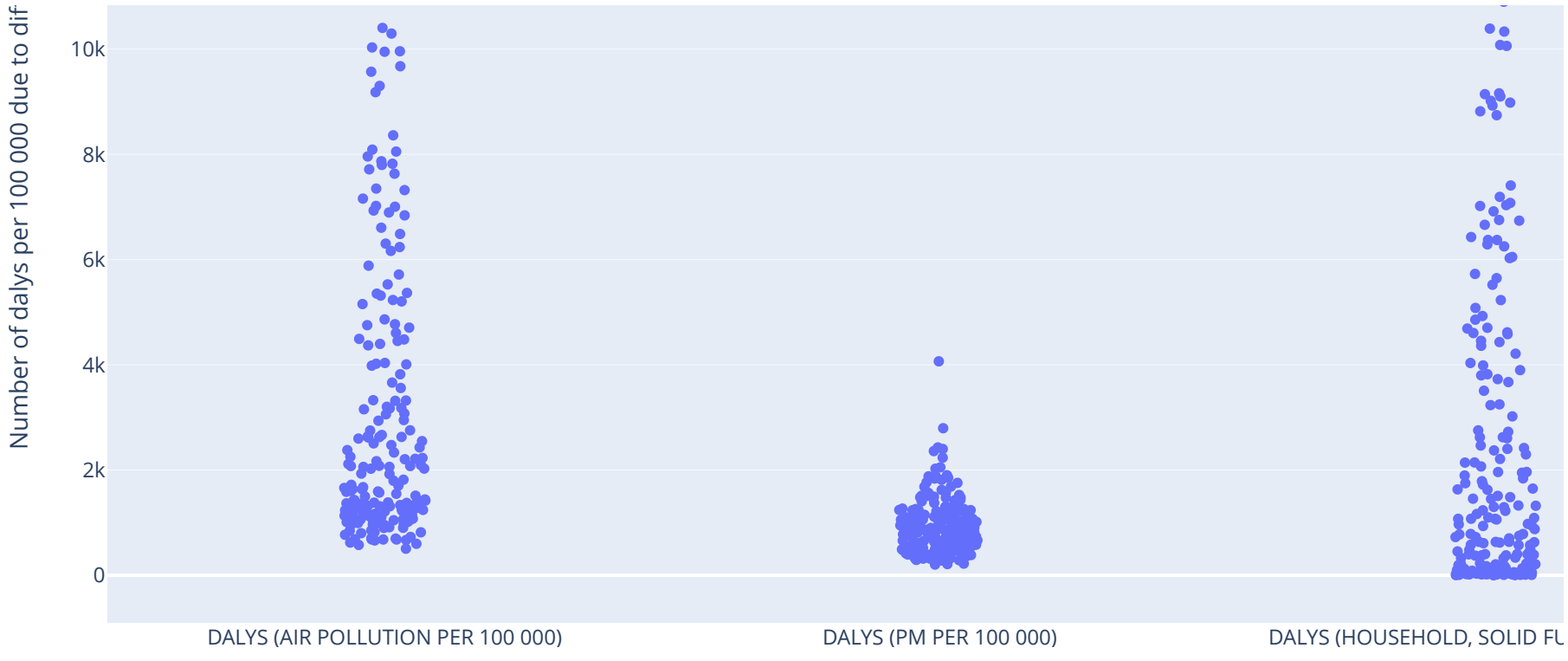


Figure 6.5.2: DEATH for different causes over the years (heatmap)

From the heatmap above, it can be observed that the number of average deaths per 100 000 for air pollution (improved the most), chronic respiratory disease, lower respiratory infection, household, pm (improved slightly) all improved. Ozone already has low number of average deaths from 1990.

It can also be observed that the death cause that contributes the most average number of deaths is air polution, followed by chronic respiratory disease, lower respiratory infection, household, PM then ozone.

In [92]:
```
fig = px.strip(data_frame=dalys_by_cause_data_qn6, x="CAUSE OF DALYS", y="DALYS",
               animation_frame="YEAR",labels={
                      "DALYS": "Number of dalys per 100 000 due to different causes"},
                 title="Graph of average dalys per 100 000 of country population for various dalys causes over time",height=800,
                 hover_name="COUNTRY")
caption = "Figure 6.6.1: DALYS for different causes over the years"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.15, showarrow=False)
fig.show()
```



From the graph above, it can be observed that the range of number of dalys per 100 000 decreases over the years, with the datapoints clustering to the smaller values. It can also be observed that the cause PM has a smaller range of number of dalys per 100 000 value as compared to air pollution and household, solid fuel causes.

Egypt (country code EGY) remains the country with the highest dalys per 100 000 for the cause PM.

From 2005 onwards, Chad (country code TCD) remains the country with the highest dalys per 100 000 for the cause air pollution.

From 2006 onwards, Chad (country code TCD) remains the country with the highest dalys per 100 000 for the cause of household, solid fuel.

In [93]:
```
plt.figure(figsize=(32,8))
sns.heatmap(year_death_cause_qn4.iloc[6:],annot=True,fmt=".2f",cmap="YlGnBu")
plt.ylabel("Cause of dalys")
plt.title("Heatmap of average dalys per 100 000 for different dalys causes over the years.")
caption = "Figure 6.6.2: DALYS for different causes over the years (heatmap)"
plt.figtext(0.425, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

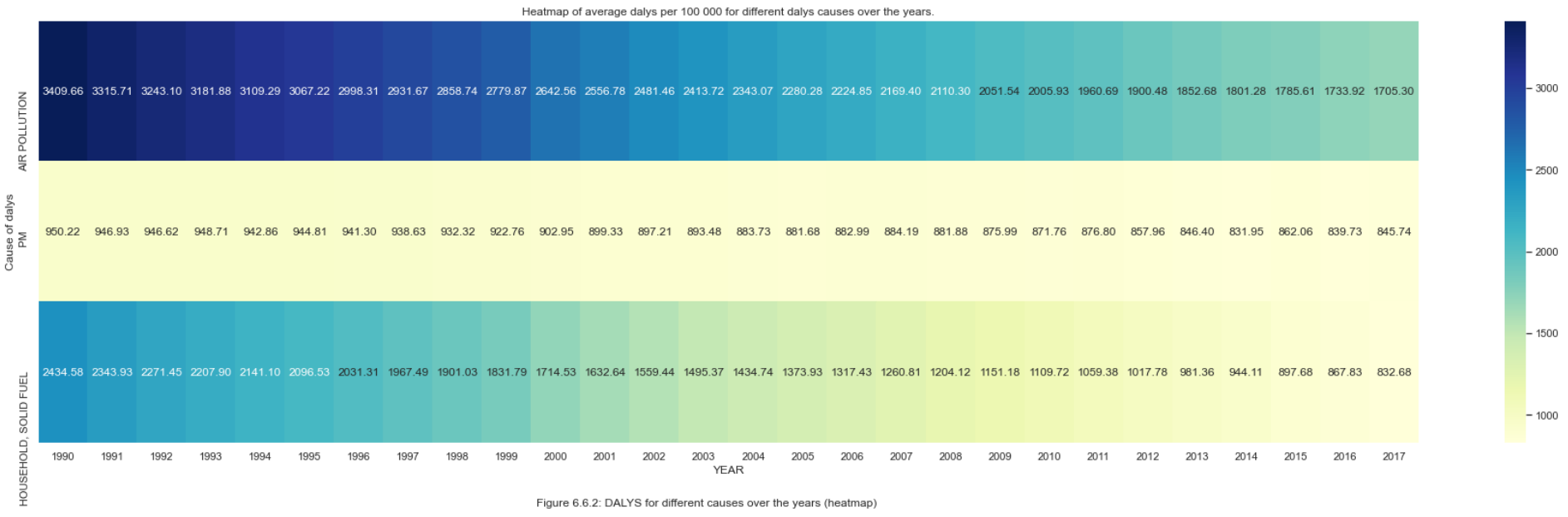


Figure 6.6.2: DALYS for different causes over the years (heatmap)

From the heatmap above, it can be observed that the number of average dalys per 100 000 for air pollution (improved the most), followed by hosuehold, solid fuel then PM (improved slightly) which all improved and decreased over the years. PM already has low number of average dalys from 1990.

It can also be observed that the dalys cause that contributes the most average number of dalys is air polution, followed by household, solid fuel then PM.

In [94]:
```
world_oecd_pm25_data_qn6 = world_oecd_pm25_data.apply(calculate_aqi,pollutant='PM2.5',pollutant_column_name='PM2.5 (MICGRCUBM)', a
world_oecd_pm25_data_qn6 = world_oecd_pm25_data_qn6.groupby(["COUNTRY","YEAR"],as_index=False)[["PM2.5 (MICGRCUBM)","AQI"]].mean()
fig = px.choropleth(world_oecd_pm25_data_qn6, locations='COUNTRY', color="AQI",
                    color_continuous_scale="deep",
                    locationmode='ISO-3',
                    scope="world",
                    animation_frame="YEAR",
                    title = "AQI for PM2.5 across the different years",
                    range_color = (0,180),
                    width = 1100,
```

```
                            height = 600
                           )
fig.update_layout(margin={"r":10,"t":30,"l":10,"b":10})
caption = "Figure 6.7 AQI for PM2.5 across world over the years"
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.1, showarrow=False)
fig.show()
```

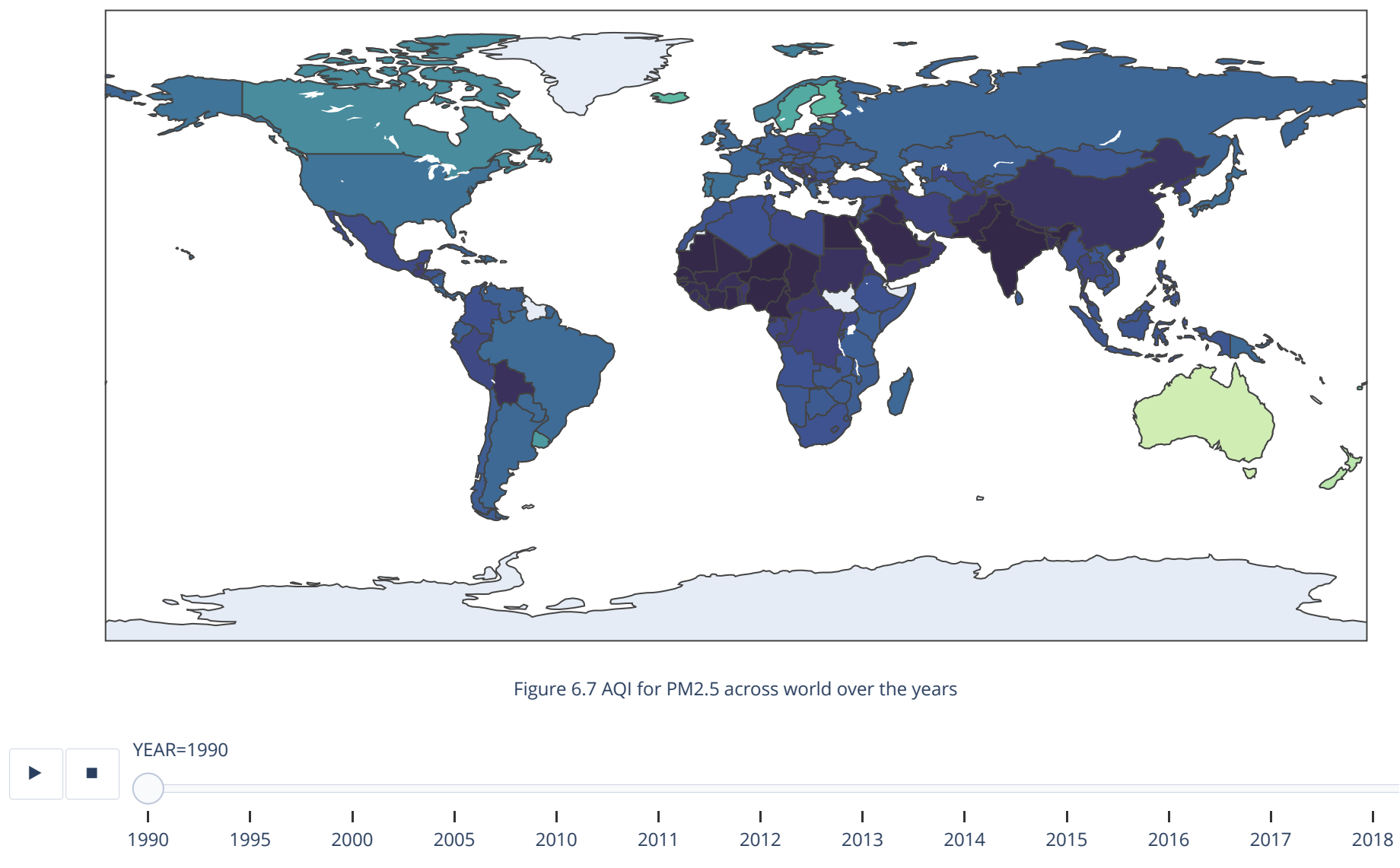


Figure 6.7 AQI for PM2.5 across world over the years

From the graph above, it can be observed that for North and South America, the colour shade turns lighter over the years, indicating that the AQI value decreased over the years. As for the African continent, majority of the countries remain having high AQIs over the years.

Australia (country code AUS) and New Zealand (country code NZL) remain light coloured from the start, having the lowest values of 15.83 and 22.90 respectively in 1990 all the way to 2011, and remain one of the countries with the lowest AQi from 2011 to 2019.

USA, Canada (country code CAN), Finland (country code FIN), Sweden (country code SWE) are among the countries that have the best improvement in AQI over the years, with the largest change in shades of color from dark blue to light green.

In [95]:

```
plt.figure(figsize=(15,8))
sns.lineplot(x="YEAR",y="DEATH",data=death_by_year_qn6)
plt.ylabel("Average deaths per 100 000")
plt.title("Graph of average deaths per 100 000 of population over the years")
caption = "Figure 6.8: DEATH trend over the years"
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

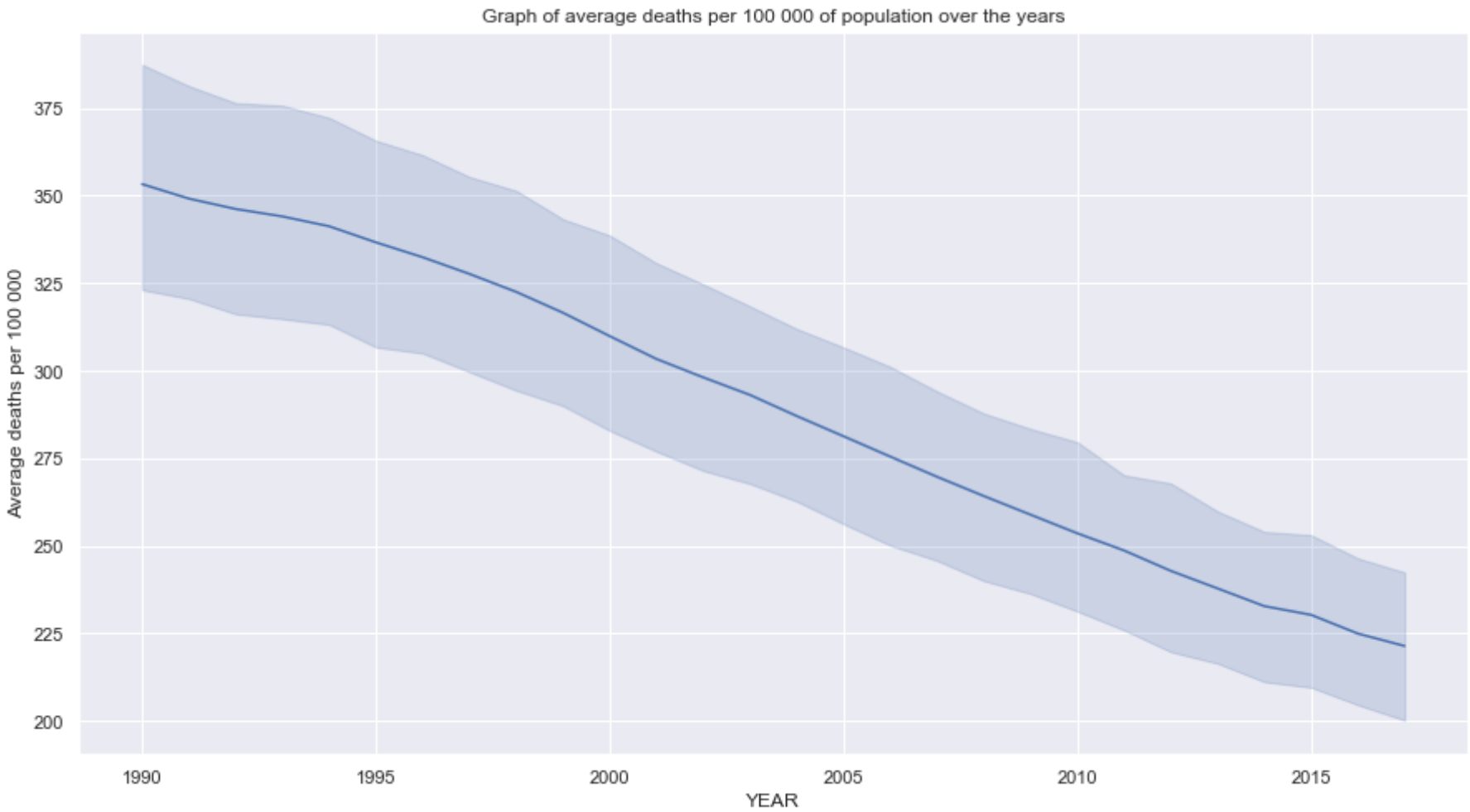


Figure 6.8: DEATH trend over the years

The graph above shows the average number of deaths per 100 000 of population over the years. It can be observed clearly that there is a decreasing linear trend for the average deaths per 100 000 of the population over the years, indicating that the number of average deaths per 100 000 is decreasing as time progresses.

In [96]:
```
plt.figure(figsize=(15,8))
sns.lineplot(x="YEAR",y="DALYS",data=dalys_by_year_qn6)
plt.ylabel("Average dalys per 100 000")
plt.title("Graph of average dalys per 100 000 of population over the years")
caption = "Figure 6.9: DALYS over the years"
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

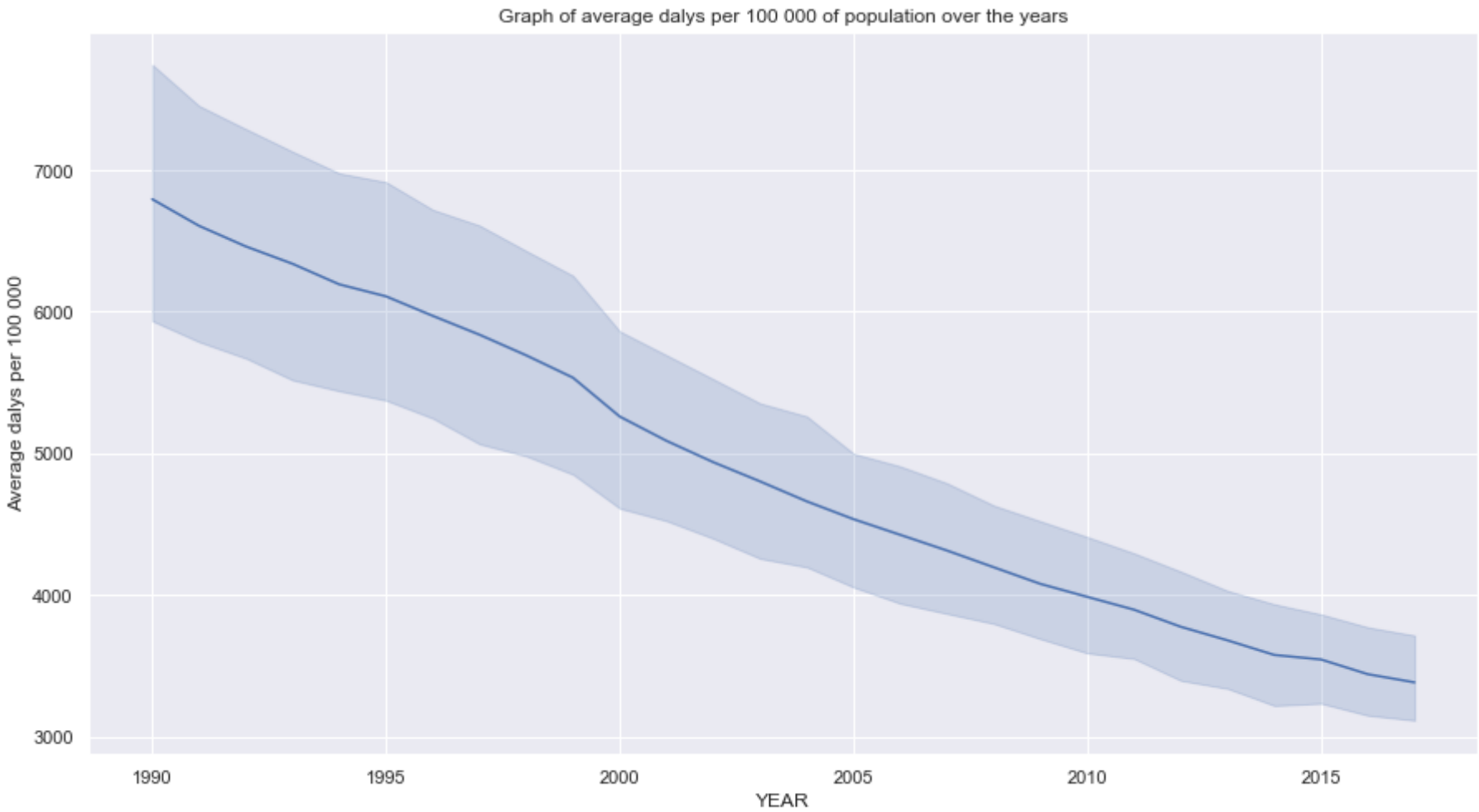


Figure 6.9: DALYS over the years

The graph above shows the average number of dalys per 100 000 of population over the years. It can be observed clearly that there is a decreasing linear trend for the average dalys per 100 000 of the population over the years, indicating that the number of average dalys per 100 000 is decreasing as time progresses.

In [97]:
```
plt.figure(figsize=(15,8))
sns.lineplot(x="YEAR",y="AQI",data=waqi_data_total)
plt.title("Graph of AQI over the years")
caption = "Figure 6.10: AQI over the years"
```

```
plt.figtext(0.5, 0.01, caption, wrap=True, horizontalalignment='center', fontsize=12)
plt.show()
```

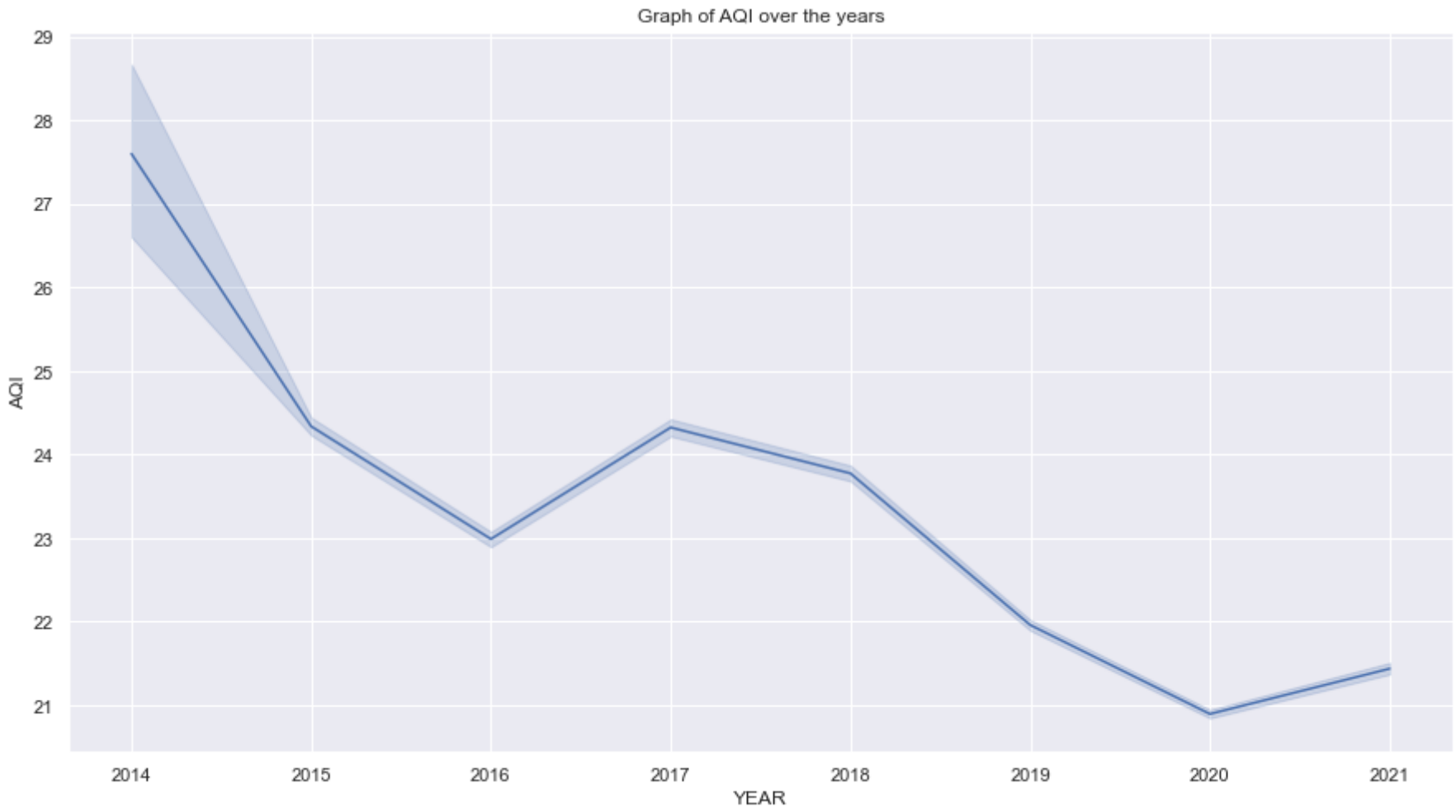


Figure 6.10: AQI over the years

The graph above shows a general decreasing trend for the AQI value over the years. From 2014 to 2015, the translucent error bars are larger, showing that there is a greater range of AQI values for the different countries. Interestingly, there is a sudden increase in AQI values from 2016 to 2017 and from 2020 to 2021, with the increase from 2016 to 2017 being a sharper increase than from 2020 to 2021.

Results Discussion

Results

1. Research Question 1 - Effect of type of air pollutant on health of people
2. Research Question 2 - Effect of air pollution on death / dalys cause (respiratory-related diseases/air-pollution related)
3. Research Question 3 - Effect of air pollution on health of different genders
4. Research Question 4 - Effect of air pollution on health of people from 2014 to 2017
5. Research Question 5 - Effect of air pollution on health of people across different geographical location
6. Research Question 6 - Trend of air pollution/health factors across different years

The function below results_linregmodel shows a full display of graphs for a single linear regression model given the inputs - dataframe, x-axis, y-axis and title as subplots. The first subplot displayed will be the normal linear regression model, with indication of the test data and train data. The second subplot will be a bargraph comparing the fitted values vs their actual values to see how well they match to each other / how far the disparity is. The third and fourth subplots are a residual plot and KDE plot respectively for the actual vs fitted values to evaluate how suitable a single linear regression model is. At the side of the graphs, the relevant key variables are displayed, including the slope, P value, R value (correlation), MSE and R^2 value.

In [98]:

```
def results_linregmodel(dataframe,x_axis,y_axis,title):
    fig = plt.figure() # create figure
    fig.tight_layout()
    fig.subplots_adjust(left=None, bottom=None, right=0.75, top=1.75, wspace=None, hspace=None)

    ax0 = fig.add_subplot(3, 1, 1)
    ax1 = fig.add_subplot(3, 1, 2)
    ax2 = fig.add_subplot(3, 2, 5)
    ax3 = fig.add_subplot(3, 2, 6)

    lm = LinearRegression()
    x_train, x_test, y_train, y_test = train_test_split(dataframe[[x_axis]], dataframe[y_axis],test_size=0.2, random_state=1)
    lm.fit(x_train, y_train)

    yhat = lm.predict(x_test)
    df_act_pred = pd.DataFrame({'Actual': y_test, 'Predicted': yhat})

    # Subplot 1: Linear Regression Plot
    sns.regplot(x=x_train,y=y_train,ax=ax0,color="tab:blue")
    sns.scatterplot(x=x_test[x_axis],y=y_test,ax=ax0,color="tab:orange")
    ax0.set_title("Graph of "+title)
    ax0.set_xlabel(x_axis)
    ax0.set_ylabel(y_axis)


    orange_test = mpatches.Patch(color='tab:orange', label='Test Data')
```

```
blue_train = mpatches.Patch(color='tab:blue', label='Train Data')

ax0.legend(handles=[orange_test, blue_train])

gradient, intercept, rvalue, pvalue, stderr = stats.linregress(x=x_train[x_axis].values, y=y_train)
mse_train = (((gradient*x_train[x_axis].values + intercept) - y_train)**2).mean()
mse_test = (((gradient*x_test[x_axis].values + intercept) - y_test)**2).mean()
pred = gradient*(dataframe[x_axis].values) + intercept
mse = mean_squared_error(y_test, yhat)
r2 = lm.score(x_train, y_train)

text = f"\nSlope: {gradient:.5f}\nP-value: {pvalue:.5f}\nR-value: {rvalue:.5f}"
text += f"\nMSE Train: {mse_train:.5f}\nMSE Test: {mse_test:.5f}"
plt.text(1.3, 1.75, text, size=15, ha='center', va='center', transform=plt.gca().transAxes)

# Subplot 2: Bar Plot for Predicted Vs Actual
df_act_pred.plot(kind='bar',figsize=(16,8),ax=ax1)
plt.grid(which='major', linestyle='-', linewidth='0.5')
ax1.set_title("Graph of Predicted Vs Actual Value for "+title)

text = f"MSE: {mse:.4f}\nR²: {r2:.4f}"
plt.text(1.3, 0.55, text, size=15, ha='center', va='center', transform=plt.gca().transAxes)

 # Subplot 3: Residual Plot
sns.residplot(x=x_axis, y=y_axis,data=dataframe,ax=ax2)
ax2.set_title("Residual plot of "+title)

# Subplot 4: KDE Plot
sns.kdeplot(yhat,  color='b', label='Fitted Value',ax=ax3)
sns.kdeplot(y_test,  color='r', label='Actual Value',ax=ax3)
ax3.set_title("KDE plot of fitted (blue) and actual (red)")

    return None
```

The function below results_multilinregmodel shows a full display of graphs for a multi linear regression model given the inputs - dataframe, x-axis list, y-axis and title as subplots. Since for multi linear regression models, it is difficult to plot the actual graph, only two subplots are plotted to evaluate the suitability of a multi linear regression model. The first and second subplot display the barchart of the actual vs fitted values and KDE plot of the actual vs fitted values respectively. At the side, the equation of the line is stated.

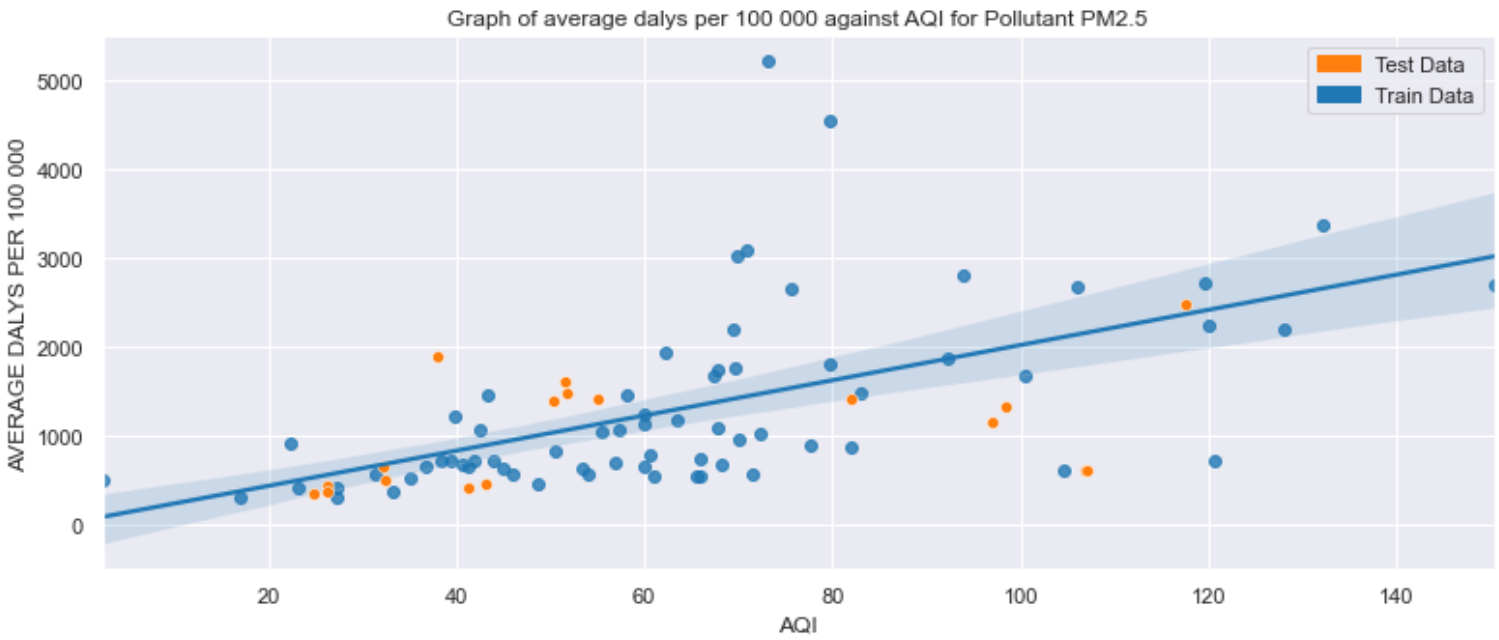
In [99]:

```
def results_multilinregmodel(dataframe,x_axis_list ,y_axis,title):
    fig = plt.figure() # create figure
    fig.tight_layout()
    fig.subplots_adjust(left=None, bottom=None, right=0.75, top=1.75, wspace=None, hspace=None)
    ax1 = fig.add_subplot(2, 1, 1)
    ax2 = fig.add_subplot(2, 1, 2)

    mlm = LinearRegression()
    x_train, x_test, y_train, y_test = train_test_split(dataframe[x_axis_list], dataframe[y_axis],test_size=0.2, random_state=1)
    mlm.fit(x_train, y_train)

    yhat = mlm.predict(x_test)
    df_act_pred = pd.DataFrame({'Actual': y_test, 'Predicted': yhat})

    equation_of_line = "y = "
    for i in range(len(x_axis_list)):
        equation_of_line += str(round(mlm.coef_[i],3)) +"* ("+x_axis_list[i]+") \n"
    equation_of_line += "+ "+str(round(mlm.intercept_,3))

    r2 = mlm.score(x_train, y_train)

    text = "\nEquation of Line: "+equation_of_line+"\n"
    plt.text(1.3, 1.75, text, size=15, ha='center', va='center', transform=plt.gca().transAxes)

    # Subplot 1: Bar Plot for Predicted Vs Actual
    df_act_pred.plot(kind='bar',figsize=(16,8),ax=ax1)
    plt.grid(which='major', linestyle='-', linewidth='0.5')
    ax1.set_title("Graph of Predicted Vs Actual Value for "+title)


    # Subplot 2: KDE Plot
    sns.kdeplot(yhat,  color='b', label='Fitted Value',ax=ax2)
    sns.kdeplot(y_test,  color='r', label='Actual Value',ax=ax2)
    ax2.set_title("KDE plot of fitted (blue) and actual (red)")

    return None
```

Q1. Research Question 1 - Effect of type of air pollutant on health of people

In [100…

```
health_dalys_pollution_qn1_dropna = health_dalys_pollution_qn1.dropna(how='any',axis=0)
health_dalys_pollution_qn1_dropna_pm25 = health_dalys_pollution_qn1_dropna[health_dalys_pollution_qn1_dropna["POLLUTANT"]
                                                                           =="PM2.5"]
results_linregmodel(dataframe=health_dalys_pollution_qn1_dropna_pm25,x_axis="AQI",y_axis="AVERAGE DALYS PER 100 000",
                    title="average dalys per 100 000 against AQI for Pollutant PM2.5")
```



Slope: 19.78709
P-value: 0.00000
R-value: 0.56777
MSE Train: 679054.44579
MSE Test: 458440.53385

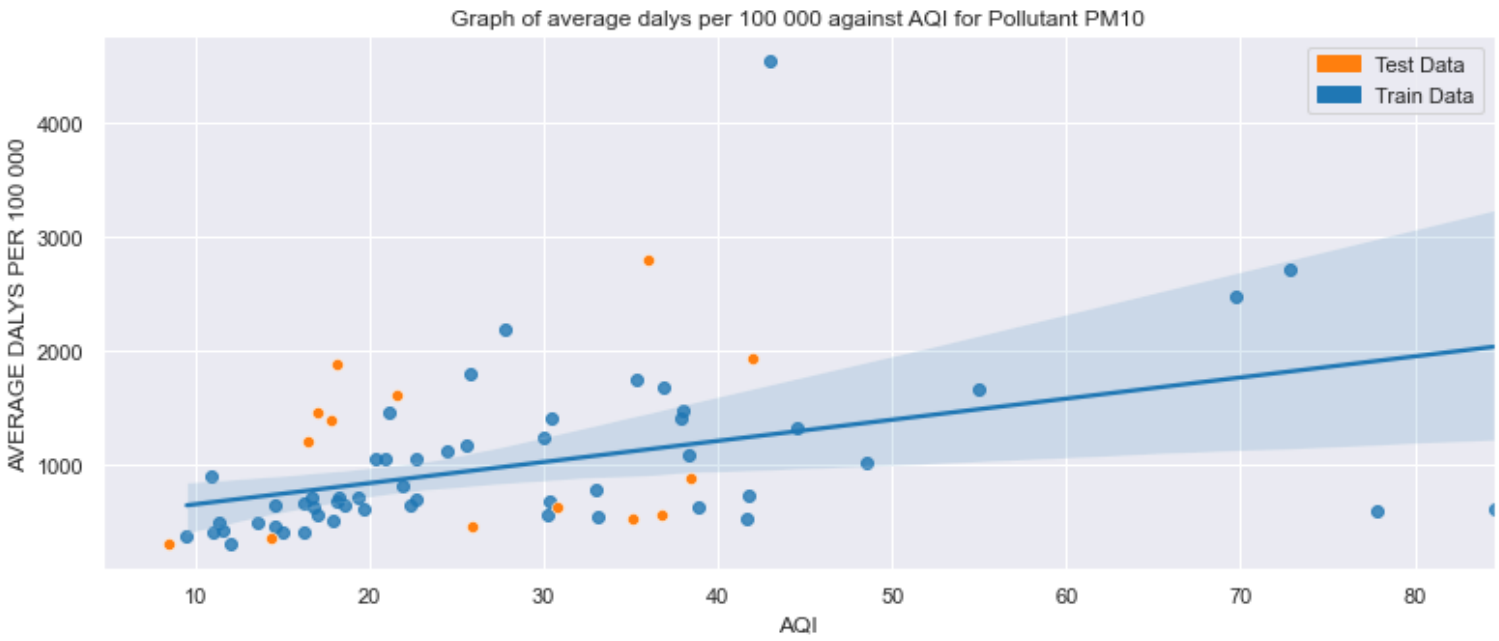
MSE: 458440.5338
R^2: 0.3224

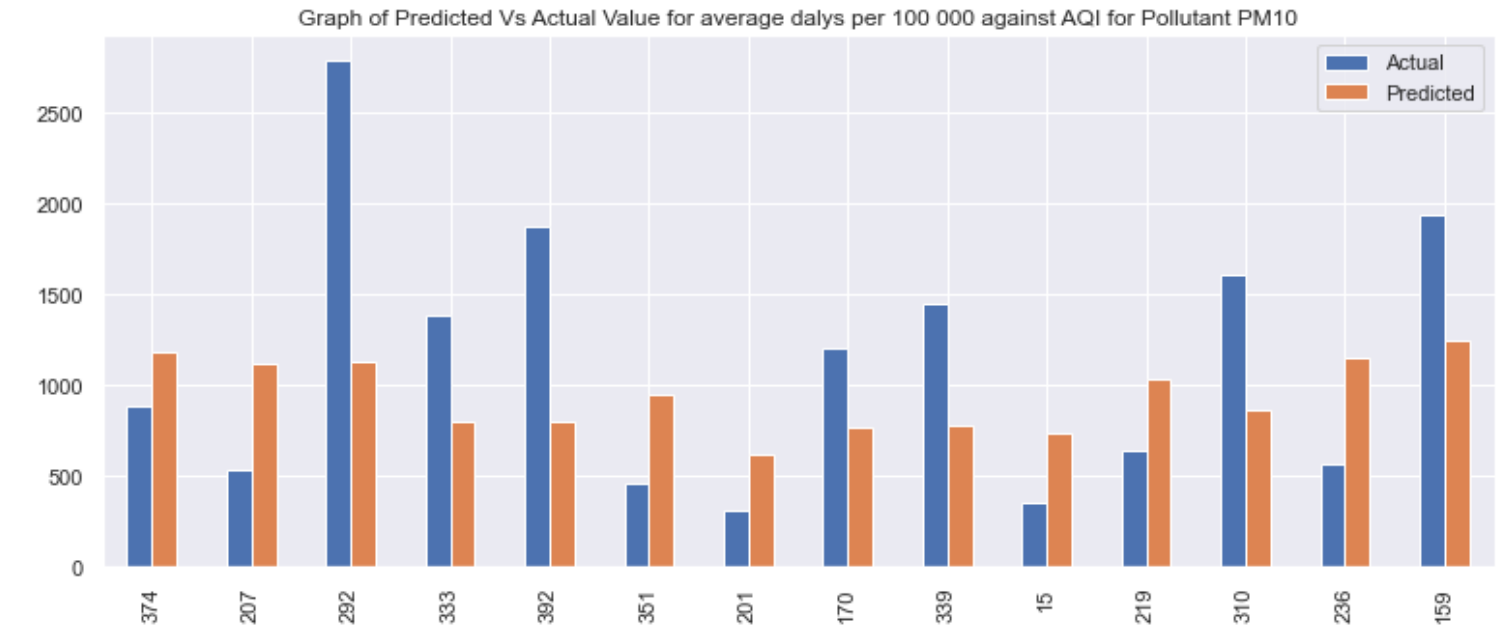
From figure 1.3.2, further investigation is done by performing the linear regression model on the pollutant PM2.5 for average dalys per 100 000 against AQI.The P value is very small while the R value (correlation value) is 0.5677, indicating there is a moderate positive correlation. From the residual plot, it can be observed that the points are roughly randomly separated around the x axis, however there are some points which are very high above the x axis. The KDE plot of fitted vs actual value does not coincide very nicely, therefore a linear regression model may not be the best model.

However, it is still evident that there is moderate correlation between average dalys per 100 000 against AQI for pollutant PM2.5.

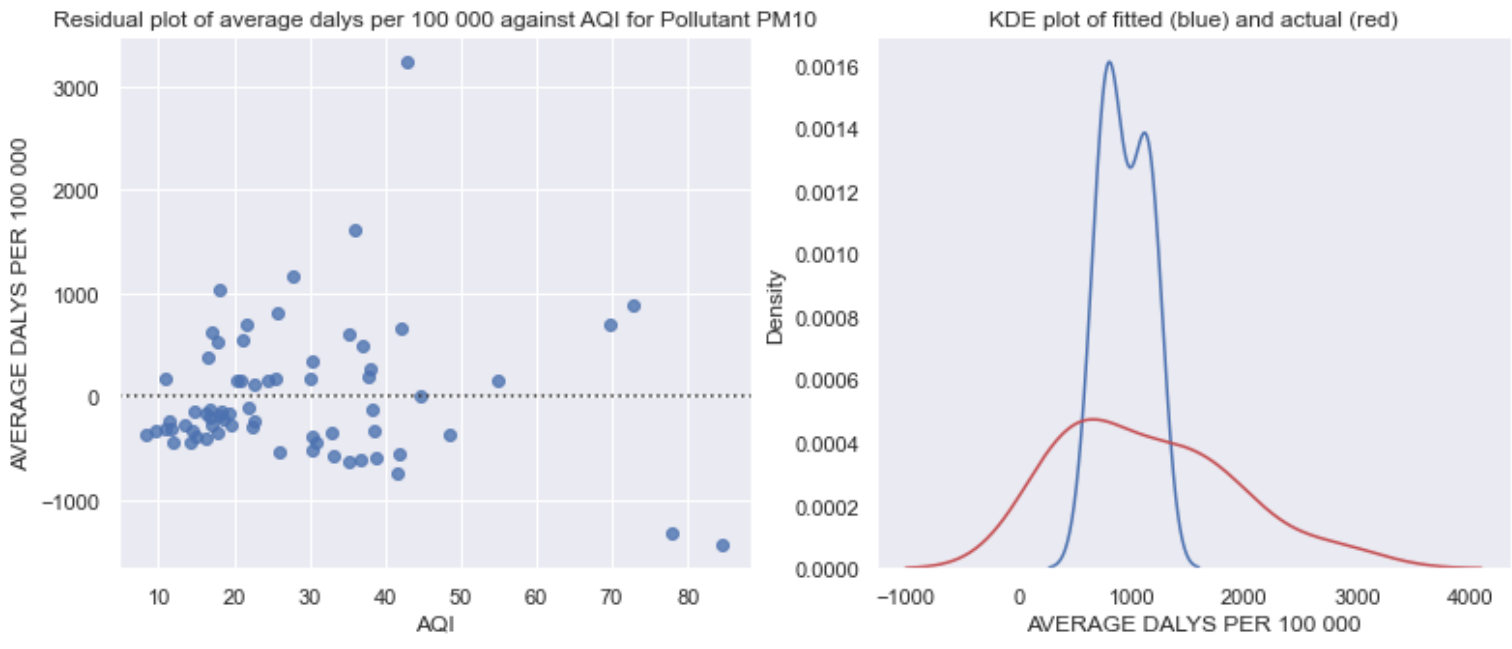
In [101…

```
health_dalys_pollution_qn1_dropna = health_dalys_pollution_qn1.dropna(how='any',axis=0)
health_dalys_pollution_qn1_dropna_pm10 = health_dalys_pollution_qn1_dropna[health_dalys_pollution_qn1_dropna["POLLUTANT"]
                                                                           =="PM10"]
results_linregmodel(dataframe=health_dalys_pollution_qn1_dropna_pm10,x_axis="AQI",y_axis="AVERAGE DALYS PER 100 000",
                    title="average dalys per 100 000 against AQI for Pollutant PM10")
```





Slope: 18.54112
P-value: 0.00085
R-value: 0.44091
MSE Train: 421269.38555
MSE Test: 523187.05193



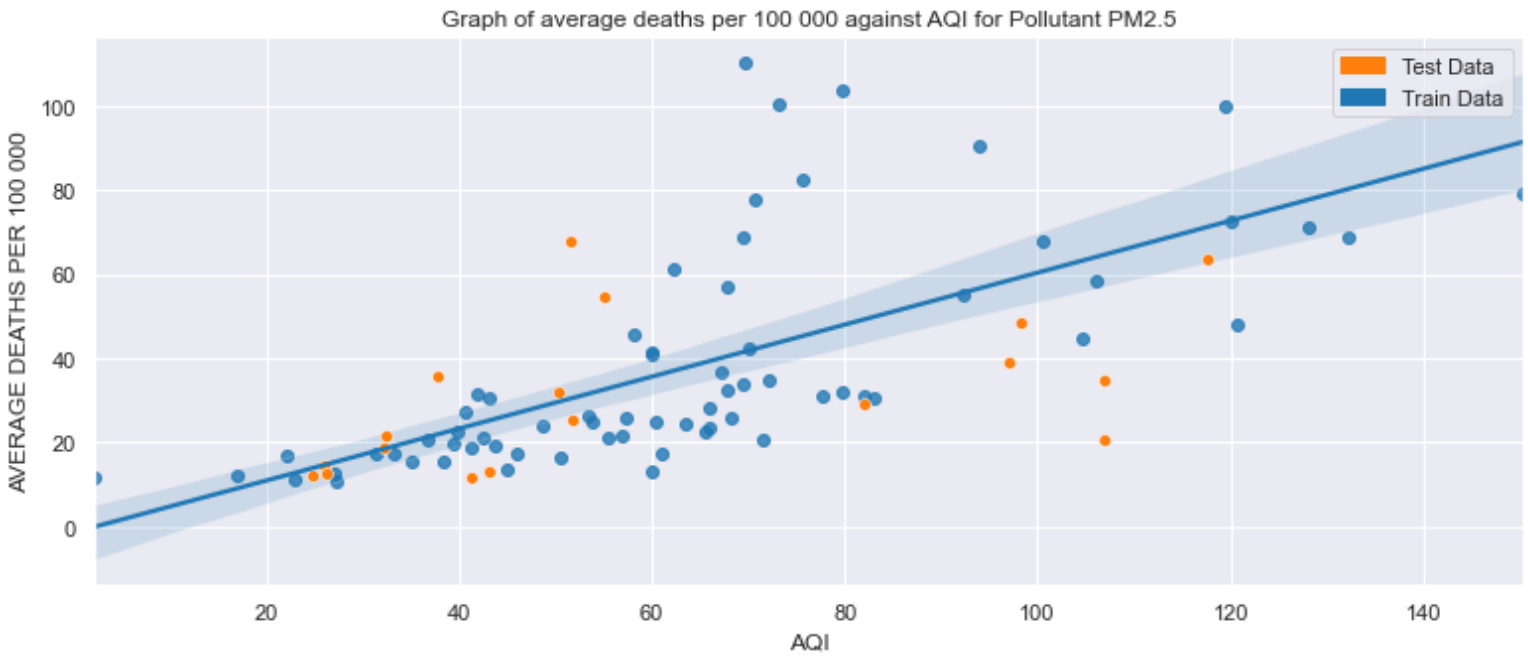
MSE: 523187.0519
R^2: 0.1944

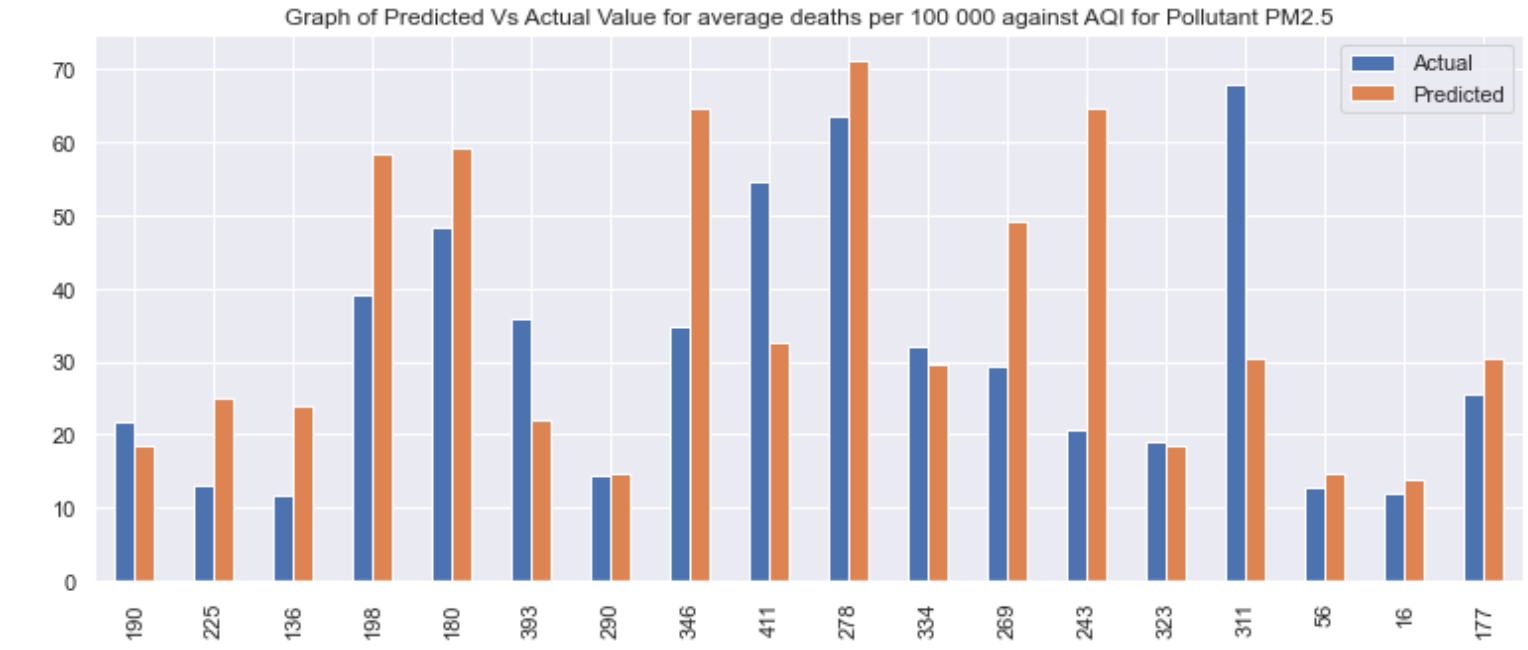
From figure 1.3.2, further investigation is done by performing the linear regression model on the pollutant PM10 for average dalys per 100 000 against AQI.The P value is small of 0.00085 while the R value (correlation value) is 0.44091, indicating there is a moderate positive correlation (slightly less than for Pollutant PM2.5). From the residual plot, it can be observed that the points are roughly randomly separated around the x axis, however there are some points which are very high above the x axis. The KDE plot of fitted vs actual value does not coincide and are very far apart, therefore a linear regression model is not a suitable model.

With the R^2 value being 0.1944, a linear regression model is not suitable for the graph of average dalys per 100 000 against AQI for PM10.

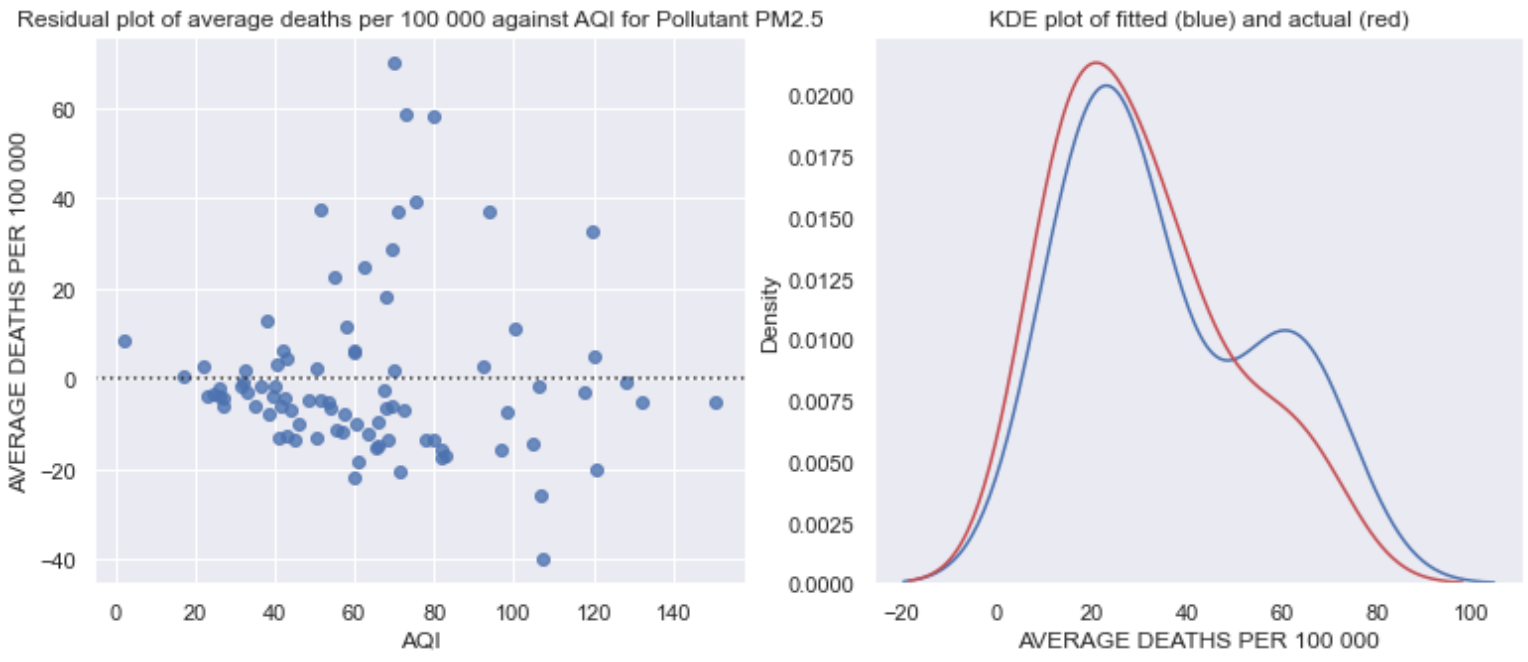
In [102…

```
health_deaths_pollution_qn1_dropna = health_deaths_pollution_qn1.dropna(how='any',axis=0)
health_deaths_pollution_qn1_dropna_pm25 = health_deaths_pollution_qn1_dropna[health_deaths_pollution_qn1_dropna["POLLUTANT"]
                                                                             =="PM2.5"]
results_linregmodel(dataframe=health_deaths_pollution_qn1_dropna_pm25,x_axis="AQI",y_axis="AVERAGE DEATHS PER 100 000",
                    title="average deaths per 100 000 against AQI for Pollutant PM2.5")
```





Slope: 0.61832
P-value: 0.00000
R-value: 0.68594
MSE Train: 354.96625
MSE Test: 345.45607



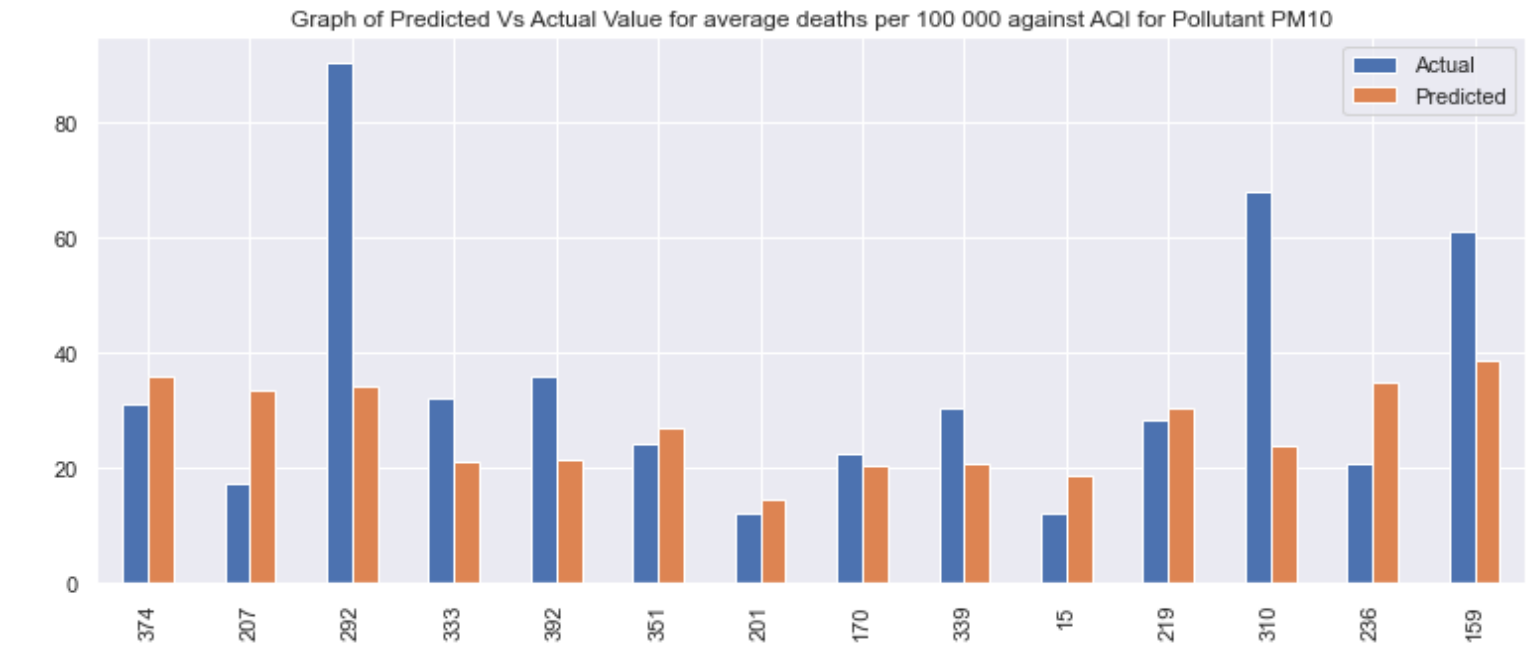
MSE: 345.4561
R^2: 0.4705

From figure 1.3.2, further investigation is done by performing the linear regression model on the pollutant PM2.5 for average deaths per 100 000 against AQI.The P value is very small while the R value (correlation value) is 0.68594, indicating there is a moderate strong positive correlation. From the residual plot, it can be observed that the points are roughly randomly separated around the x axis, with very little points which are very high above the x axis. The KDE plot of fitted vs actual value does coincide quite well, therefore a linear regression model may be a suitable model.

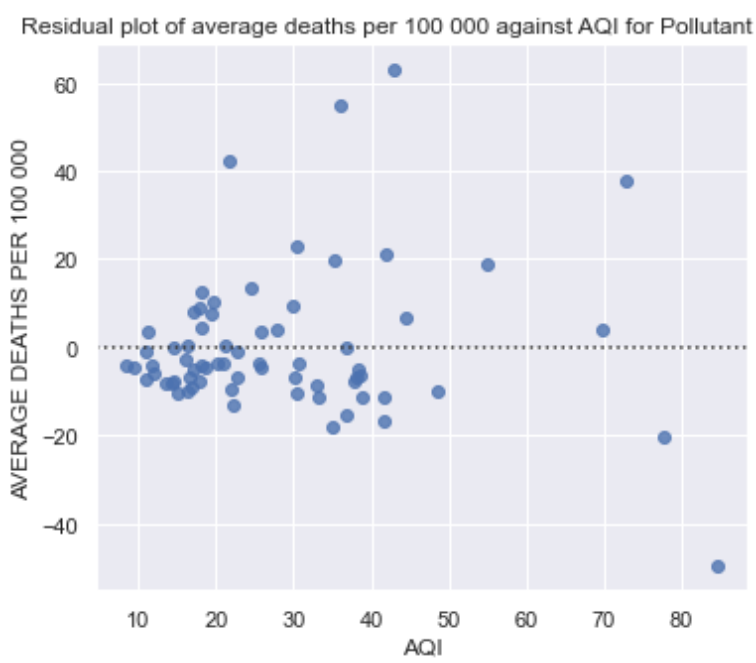
Therefore, we can conclude that the graph of average deaths per 100 000 against AQI for pollutant PM2.5 has a moderately strong linear correlation.

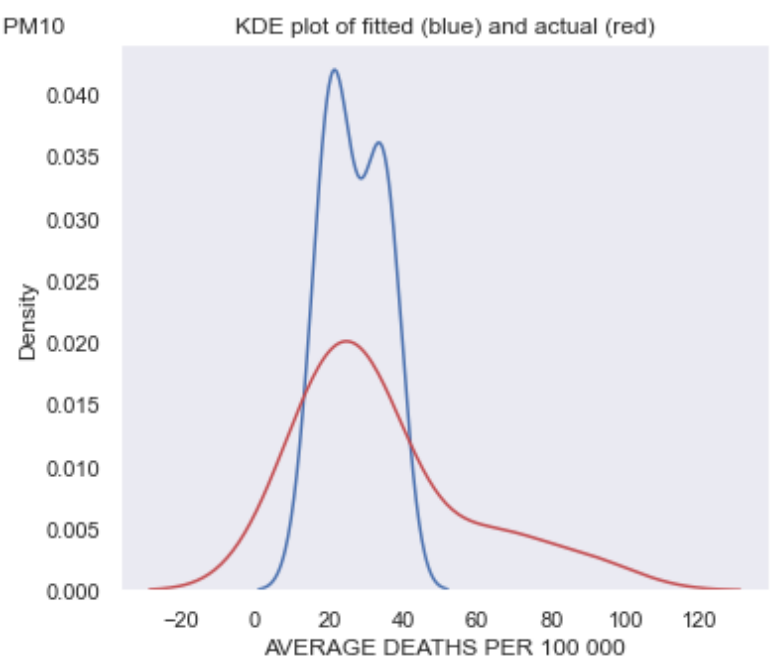
In [103…

```
health_deaths_pollution_qn1_dropna = health_deaths_pollution_qn1.dropna(how='any',axis=0)
health_deaths_pollution_qn1_dropna_pm10 = health_deaths_pollution_qn1_dropna[health_deaths_pollution_qn1_dropna["POLLUTANT"]
                                                                             =="PM10"]
results_linregmodel(dataframe=health_deaths_pollution_qn1_dropna_pm10,x_axis="AQI",y_axis="AVERAGE DEATHS PER 100 000",
                    title="average deaths per 100 000 against AQI for Pollutant PM10")
```



Slope: 0.71190
P-value: 0.00000
R-value: 0.63207
MSE Train: 225.25273
MSE Test: 470.10381





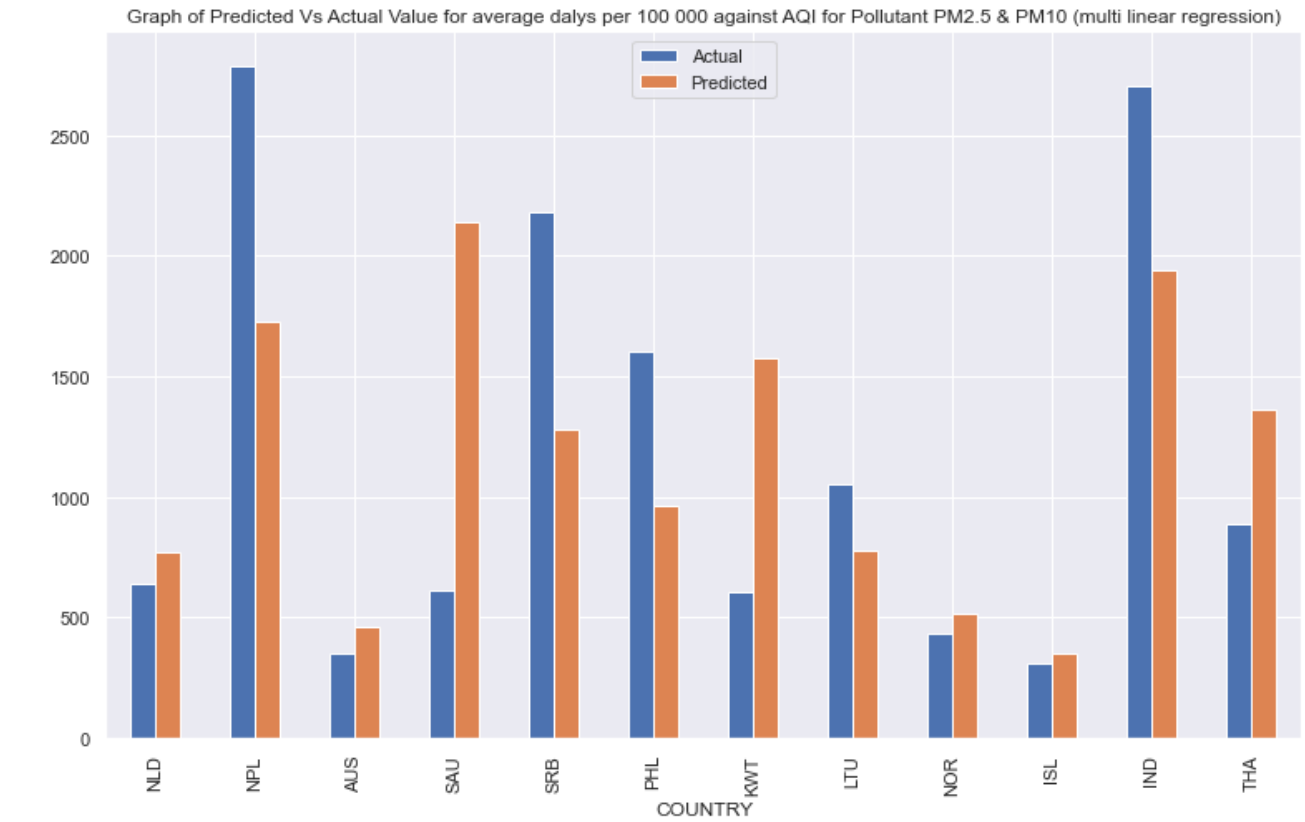
MSE: 470.1038
R²: 0.3995

From figure 1.3.2, further investigation is done by performing the linear regression model on the pollutant PM10 for average deaths per 100 000 against AQI.The P value is very small while the R value (correlation value) is 0.63207, indicating there is a moderate strong positive correlation. From the residual plot, it can be observed that the points are roughly randomly separated around the x axis, with very little points which are very high above and below the x axis. The KDE plot of fitted vs actual value does not really coincide well, therefore a linear regression model may not be a suitable model.

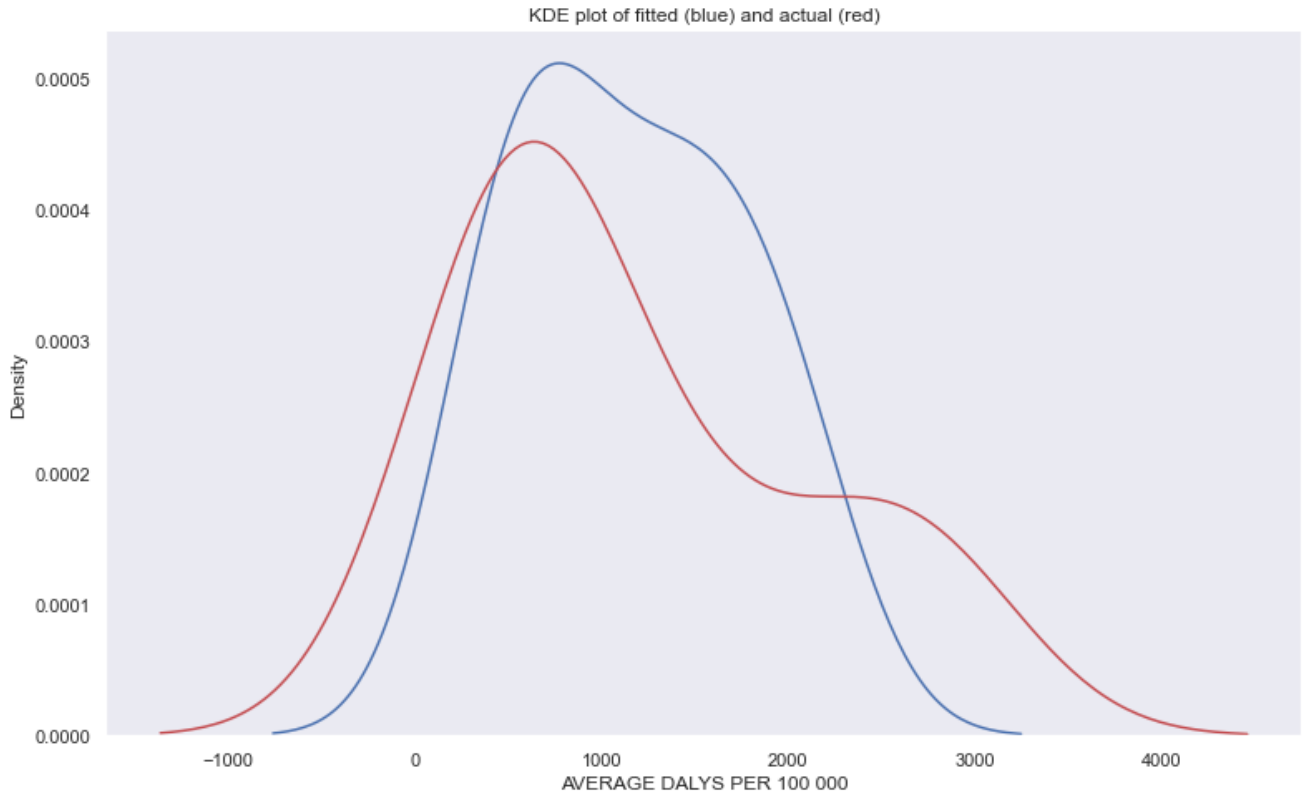
However, we can conclude that the graph of average deaths per 100 000 against AQI for pollutant PM2.5 has a moderately strong positive correlation.

In [104…

```
health_dalys_pollution_qn1_dropna_pm25pm10 = health_dalys_pollution_qn1[(health_dalys_pollution_qn1["POLLUTANT"]
                                                                         =="PM2.5")|(health_dalys_pollution_qn1["POLLUTANT"]
                                                                         =="PM10")]
health_dalys_pollution_qn1_dropna_pm25pm10 = pd.pivot_table(health_dalys_pollution_qn1_dropna_pm25pm10,index=["COUNTRY","AVERAGE D
                                                            values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_dalys_pollution_qn1_dropna_pm25pm10 = health_dalys_pollution_qn1_dropna_pm25pm10["AQI"]
health_dalys_pollution_qn1_dropna_pm25pm10 = health_dalys_pollution_qn1_dropna_pm25pm10.reset_index()
health_dalys_pollution_qn1_dropna_pm25pm10 = health_dalys_pollution_qn1_dropna_pm25pm10.set_index("COUNTRY")
health_dalys_pollution_qn1_dropna_pm25pm10 = health_dalys_pollution_qn1_dropna_pm25pm10.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_dalys_pollution_qn1_dropna_pm25pm10,x_axis_list=["PM2.5","PM10"],y_axis="AVERAGE DALYS P
                         title="average dalys per 100 000 against AQI for Pollutant PM2.5 & PM10 (multi linear regression)")
```



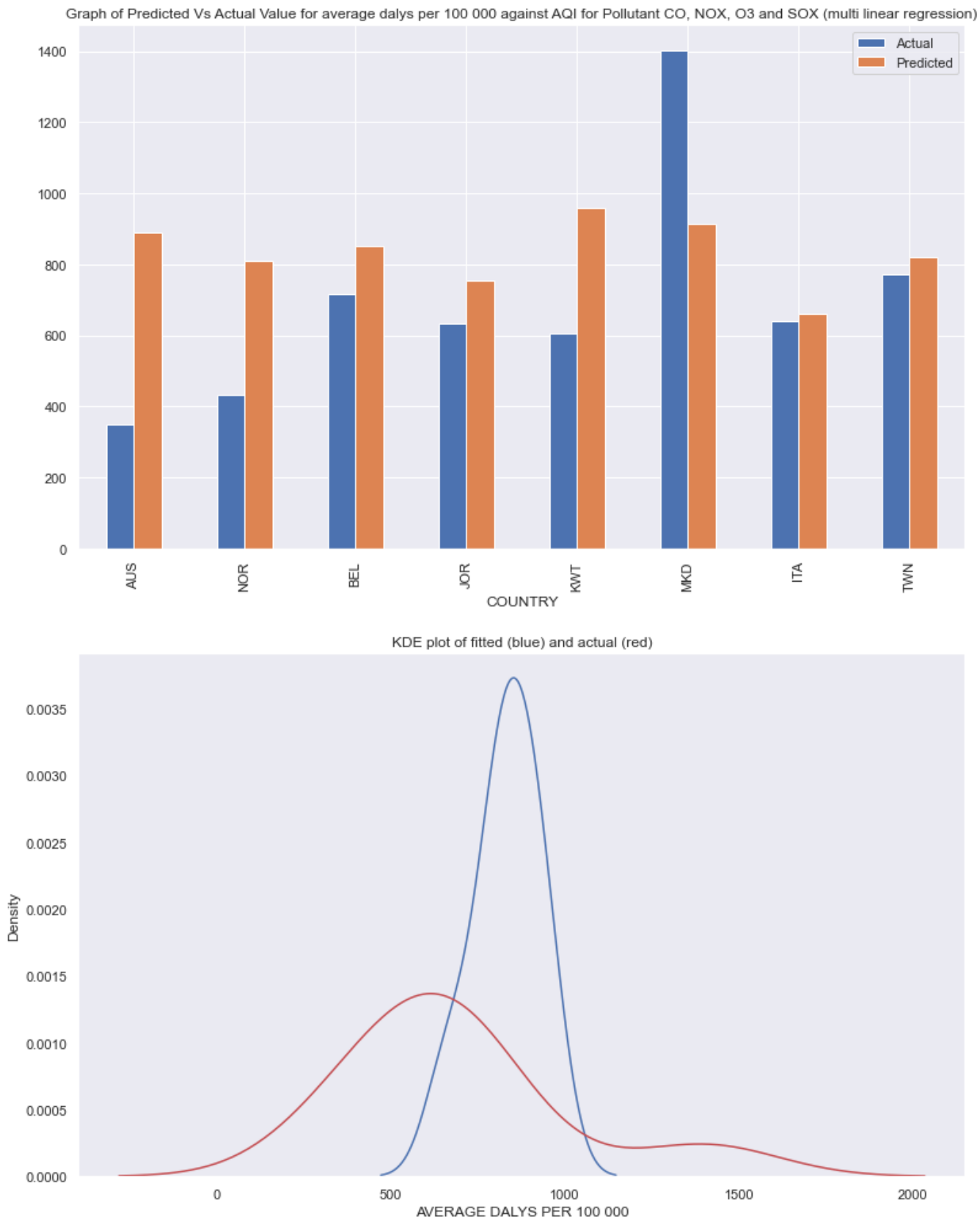
Equation of Line: y = 21.073* (PM2.5)
-8.817* (PM10)
+ 64.078



A multi linear regression model is done for AQI for PM2.5 and PM10 to investigate whether the combined trend will have a better linear correlation for average dalys per 100 000. Form the residual plot, it can be observed that the actual and fitted values do not coincide very well though the general trend follows, therefore a multi linear regression model may not suitable.

In [105…

```
health_dalys_pollution_qn1_dropna_notpm25pm10 = health_dalys_pollution_qn1[(health_dalys_pollution_qn1["POLLUTANT"]
                                                                            =="CO")|(health_dalys_pollution_qn1["POLLUTANT"]
                                                                            =="NOX")|(health_dalys_pollution_qn1["POLLUTANT"]
                                                                            =="O3")|(health_dalys_pollution_qn1["POLLUTANT"]
                                                                            =="SOX")]
health_dalys_pollution_qn1_dropna_notpm25pm10 = pd.pivot_table(health_dalys_pollution_qn1_dropna_notpm25pm10,index=["COUNTRY","AVE
                                                               values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_dalys_pollution_qn1_dropna_notpm25pm10 = health_dalys_pollution_qn1_dropna_notpm25pm10["AQI"]
health_dalys_pollution_qn1_dropna_notpm25pm10 = health_dalys_pollution_qn1_dropna_notpm25pm10.reset_index()
health_dalys_pollution_qn1_dropna_notpm25pm10 = health_dalys_pollution_qn1_dropna_notpm25pm10.set_index("COUNTRY")
health_dalys_pollution_qn1_dropna_notpm25pm10 = health_dalys_pollution_qn1_dropna_notpm25pm10.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_dalys_pollution_qn1_dropna_notpm25pm10,x_axis_list=["CO","NOX","O3","SOX"],y_axis="AVERA
                         title="average dalys per 100 000 against AQI for Pollutant CO, NOX, O3 and SOX (multi linear regression)")
```

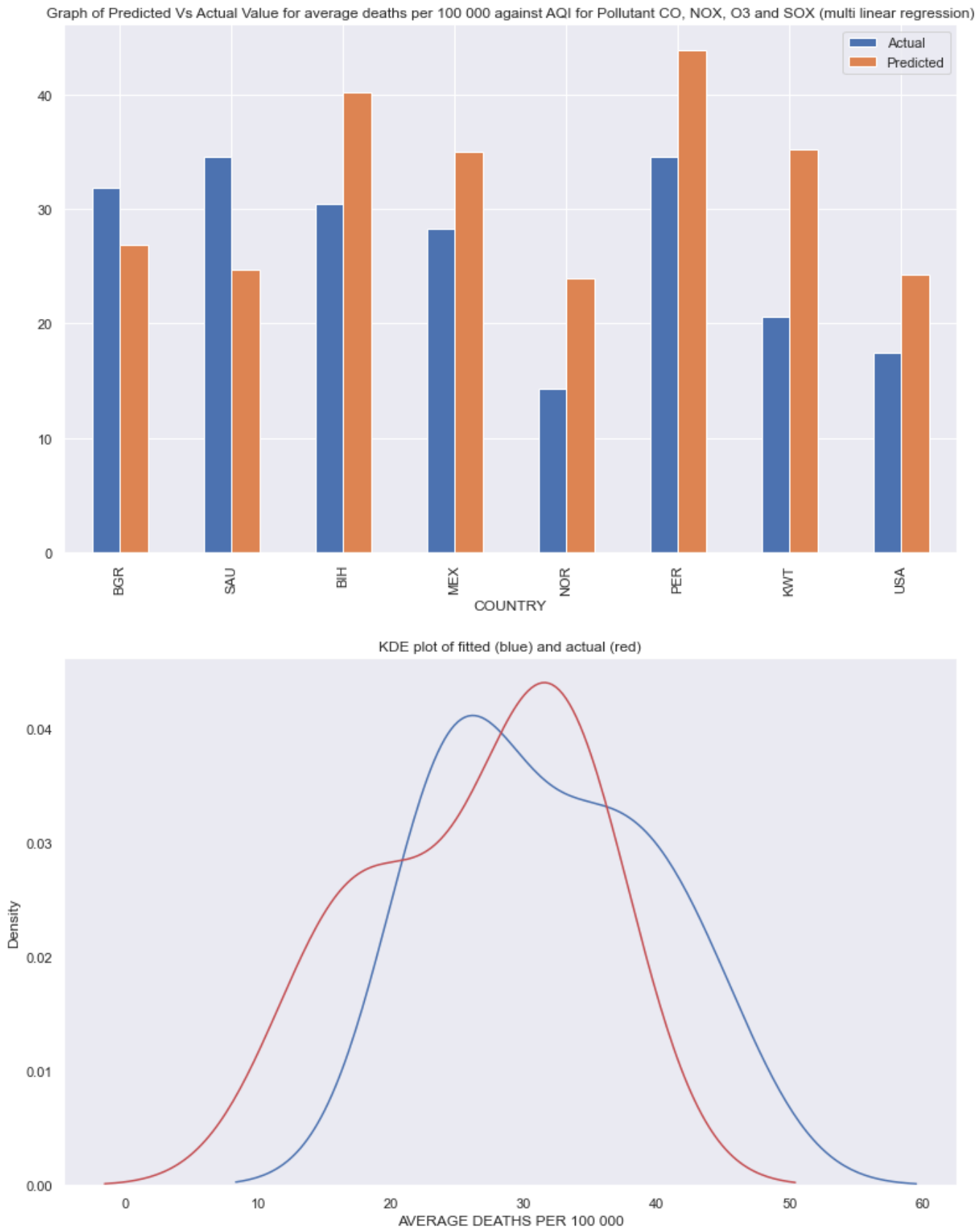


Equation of Line: y = -7.883* (CO)
-7.727* (NOX)
-11.254* (O3)
19.858* (SOX)
+ 1114.765

A multi linear regression model is done for AQI for NOX, SOX, O3 and CO to investigate whether the combined trend will have a better linear correlation for average dalys per 100 000. Form the residual plot, it can be observed that the actual and fitted values do not coincide very well, therefore a multi linear regression model is not suitable.

In [106…

```
health_deaths_pollution_qn1_dropna_notpm25pm10 = health_deaths_pollution_qn1[(health_deaths_pollution_qn1["POLLUTANT"]
                                                                              =="CO")|(health_deaths_pollution_qn1["POLLUTANT"]
                                                                              =="NOX")|(health_deaths_pollution_qn1["POLLUTANT"]
                                                                              =="O3")|(health_deaths_pollution_qn1["POLLUTANT"]
                                                                              =="SOX")]
health_deaths_pollution_qn1_dropna_notpm25pm10 = pd.pivot_table(health_deaths_pollution_qn1_dropna_notpm25pm10,index=["COUNTRY","A
                                                                values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_deaths_pollution_qn1_dropna_notpm25pm10 = health_deaths_pollution_qn1_dropna_notpm25pm10["AQI"]
health_deaths_pollution_qn1_dropna_notpm25pm10 = health_deaths_pollution_qn1_dropna_notpm25pm10.reset_index()
health_deaths_pollution_qn1_dropna_notpm25pm10 = health_deaths_pollution_qn1_dropna_notpm25pm10.set_index("COUNTRY")
health_deaths_pollution_qn1_dropna_notpm25pm10 = health_deaths_pollution_qn1_dropna_notpm25pm10.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_deaths_pollution_qn1_dropna_notpm25pm10,x_axis_list=["CO","NOX","O3","SOX"],y_axis="AVER
                         title="average deaths per 100 000 against AQI for Pollutant CO, NOX, O3 and SOX (multi linear regression)")
```

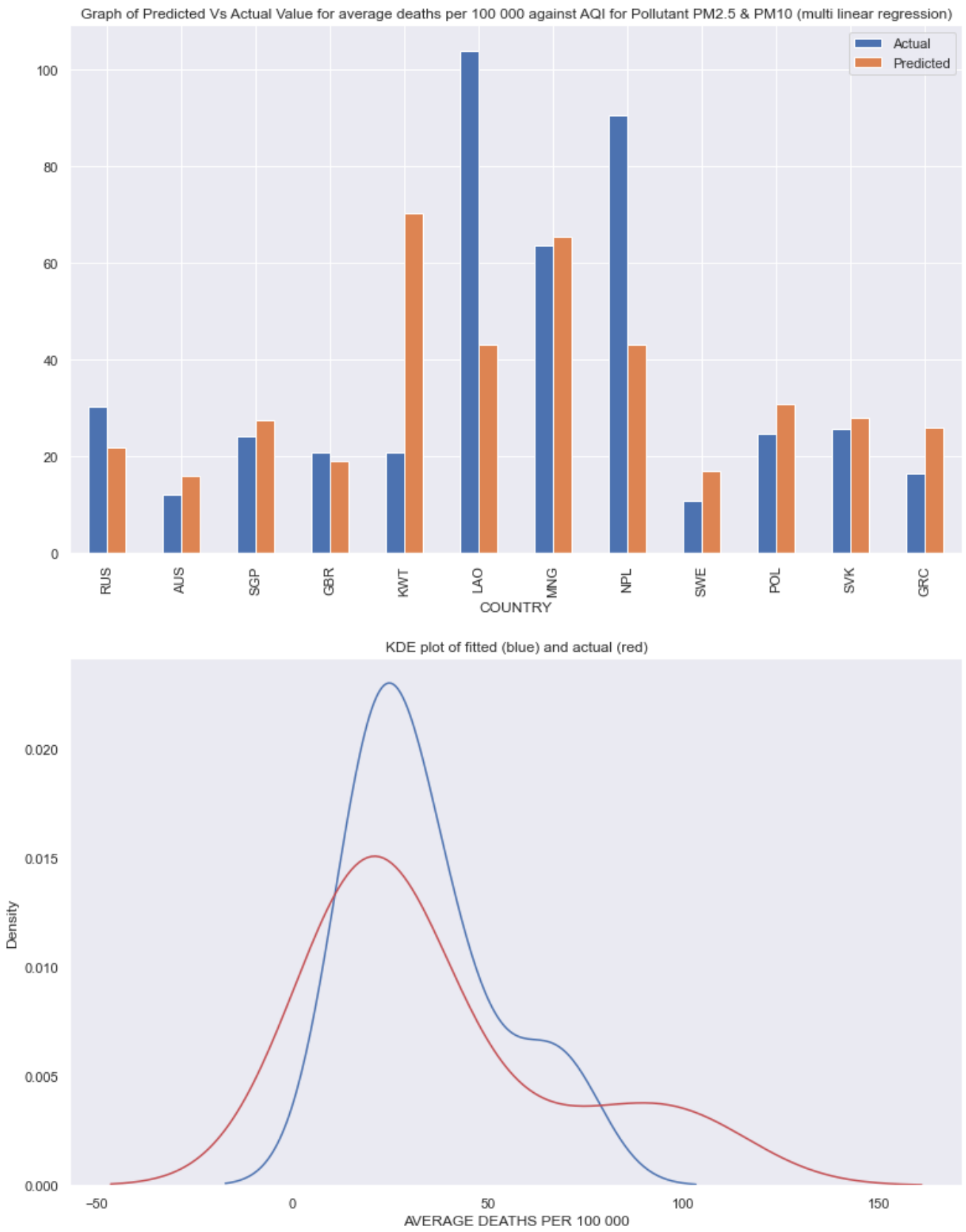


Equation of Line: y = 0.579* (CO)
0.649* (NOX)
-0.441* (O3)
0.648* (SOX)
+ 26.085

A multi linear regression model is done for AQI for NOX, SOX, O3 and CO to investigate whether the combined trend will have a better linear correlation for average deaths per 100 000. Form the residual plot, it can be observed that the actual and fitted values do not coincide very well, therefore a multi linear regression model is not suitable.

In [107…

```
health_deaths_pollution_qn1_dropna_pm25pm10 = health_deaths_pollution_qn1[(health_deaths_pollution_qn1["POLLUTANT"]
                                                                           =="PM2.5")|(health_deaths_pollution_qn1["POLLUTANT"]
                                                                           =="PM10")]
health_deaths_pollution_qn1_dropna_pm25pm10 = pd.pivot_table(health_deaths_pollution_qn1_dropna_pm25pm10,index=["COUNTRY","AVERAGE
                                                             values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_deaths_pollution_qn1_dropna_pm25pm10 = health_deaths_pollution_qn1_dropna_pm25pm10["AQI"]
health_deaths_pollution_qn1_dropna_pm25pm10 = health_deaths_pollution_qn1_dropna_pm25pm10.reset_index()
health_deaths_pollution_qn1_dropna_pm25pm10 = health_deaths_pollution_qn1_dropna_pm25pm10.set_index("COUNTRY")
health_deaths_pollution_qn1_dropna_pm25pm10 = health_deaths_pollution_qn1_dropna_pm25pm10.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_deaths_pollution_qn1_dropna_pm25pm10,x_axis_list=["PM2.5","PM10"],y_axis="AVERAGE DEATHS
                         title="average deaths per 100 000 against AQI for Pollutant PM2.5 & PM10 (multi linear regression)")
```

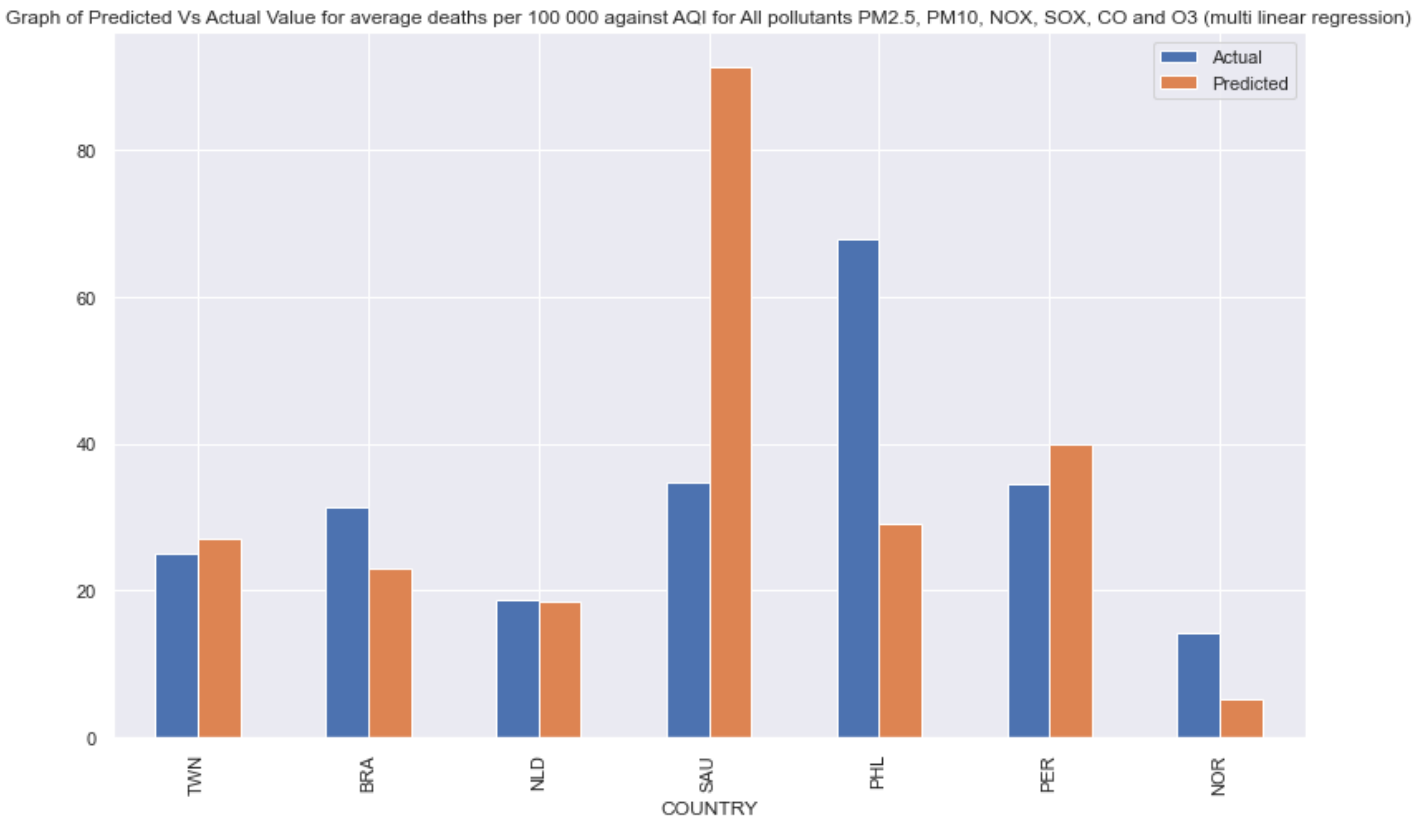


Equation of Line: y = 0.237* (PM2.5)
0.495* (PM10)
+ 3.015

A multi linear regression model is done for AQI for PM2.5 and PM10 to investigate whether the combined trend will have a better linear correlation for average deaths per 100 000. Form the residual plot, it can be observed that the actual and fitted values do not coincide very well, therefore a multi linear regression model may not suitable.

In [108…

```
health_deaths_pollution_qn1_dropna_all = health_deaths_pollution_qn1.copy()
health_deaths_pollution_qn1_dropna_all = pd.pivot_table(health_deaths_pollution_qn1_dropna_all,index=["COUNTRY","AVERAGE DEATHS PE
                                                        values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_deaths_pollution_qn1_dropna_all = health_deaths_pollution_qn1_dropna_all["AQI"]
health_deaths_pollution_qn1_dropna_all = health_deaths_pollution_qn1_dropna_all.reset_index()
health_deaths_pollution_qn1_dropna_all = health_deaths_pollution_qn1_dropna_all.set_index("COUNTRY")
health_deaths_pollution_qn1_dropna_all = health_deaths_pollution_qn1_dropna_all.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_deaths_pollution_qn1_dropna_all,x_axis_list=["PM2.5","PM10","NOX","CO","O3","SOX"],y_axi
                         title="average deaths per 100 000 against AQI for All pollutants PM2.5, PM10, NOX, SOX, CO and O3 (multi line
```

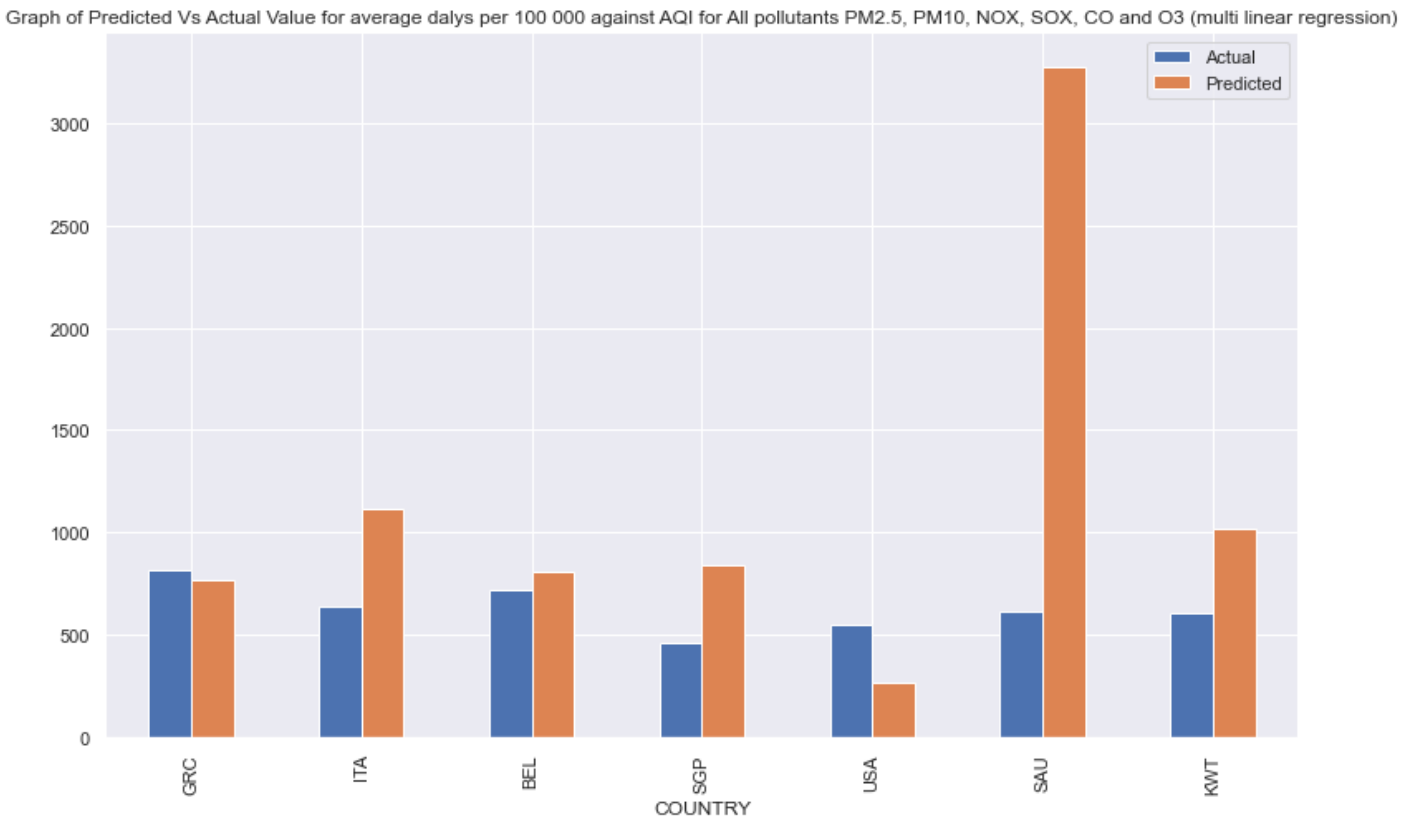


Equation of Line: y = 1.047* (PM2.5)
-0.731* (PM10)
-0.385* (NOX)
0.77* (CO)
-0.57* (O3)
-0.293* (SOX)
+ 2.482

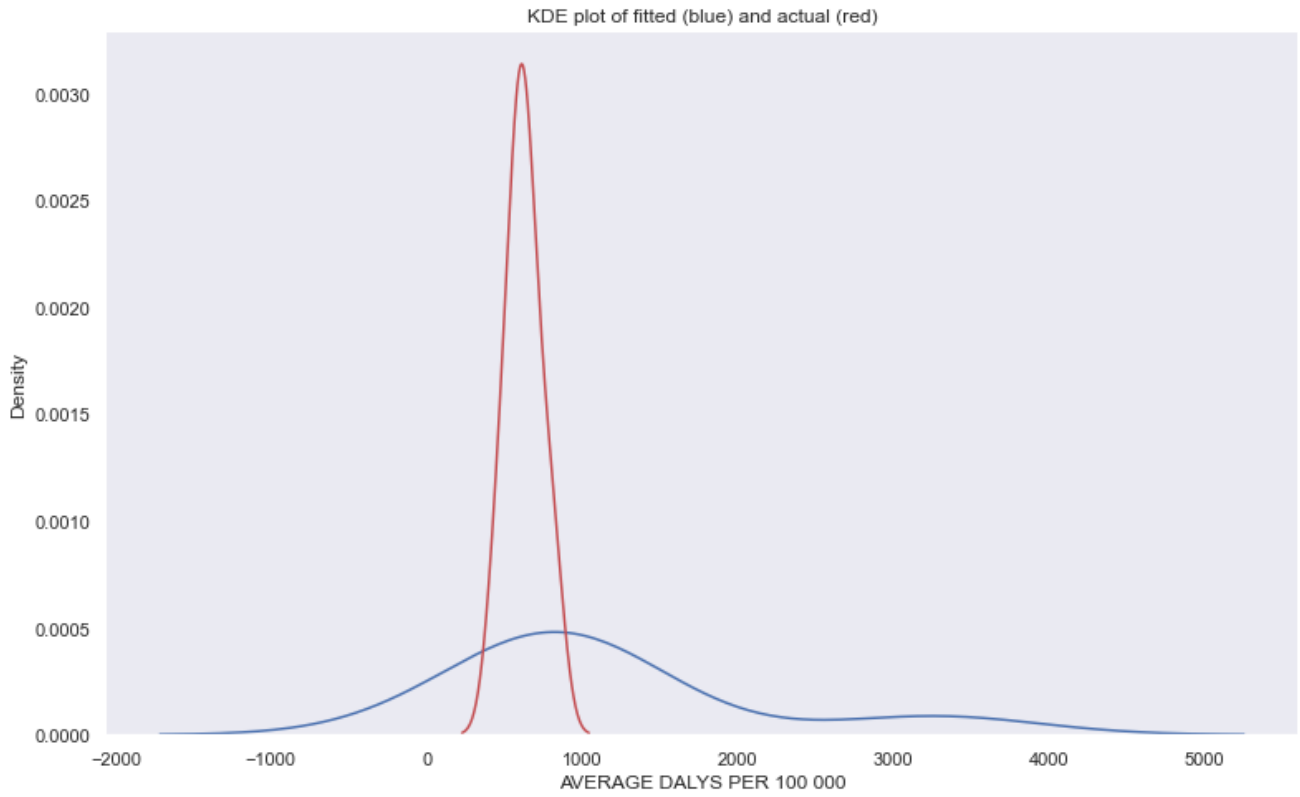
A multi linear regression model is done for AQI for all pollutants to investigate whether the combined trend will have a better linear correlation for average deaths per 100 000. Form the residual plot, it can be observed that the actual and fitted values do not coincide very well, therefore a multi linear regression model may not suitable.

In [109…

```
health_dalys_pollution_qn1_dropna_all = health_dalys_pollution_qn1.copy()
health_dalys_pollution_qn1_dropna_all = pd.pivot_table(health_dalys_pollution_qn1_dropna_all,index=["COUNTRY","AVERAGE DALYS PER 1
                                                       values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_dalys_pollution_qn1_dropna_all = health_dalys_pollution_qn1_dropna_all["AQI"]
health_dalys_pollution_qn1_dropna_all = health_dalys_pollution_qn1_dropna_all.reset_index()
health_dalys_pollution_qn1_dropna_all = health_dalys_pollution_qn1_dropna_all.set_index("COUNTRY")
health_dalys_pollution_qn1_dropna_all = health_dalys_pollution_qn1_dropna_all.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_dalys_pollution_qn1_dropna_all,x_axis_list=["PM2.5","PM10","NOX","CO","O3","SOX"],y_axis
                         title="average dalys per 100 000 against AQI for All pollutants PM2.5, PM10, NOX, SOX, CO and O3 (multi linea
```



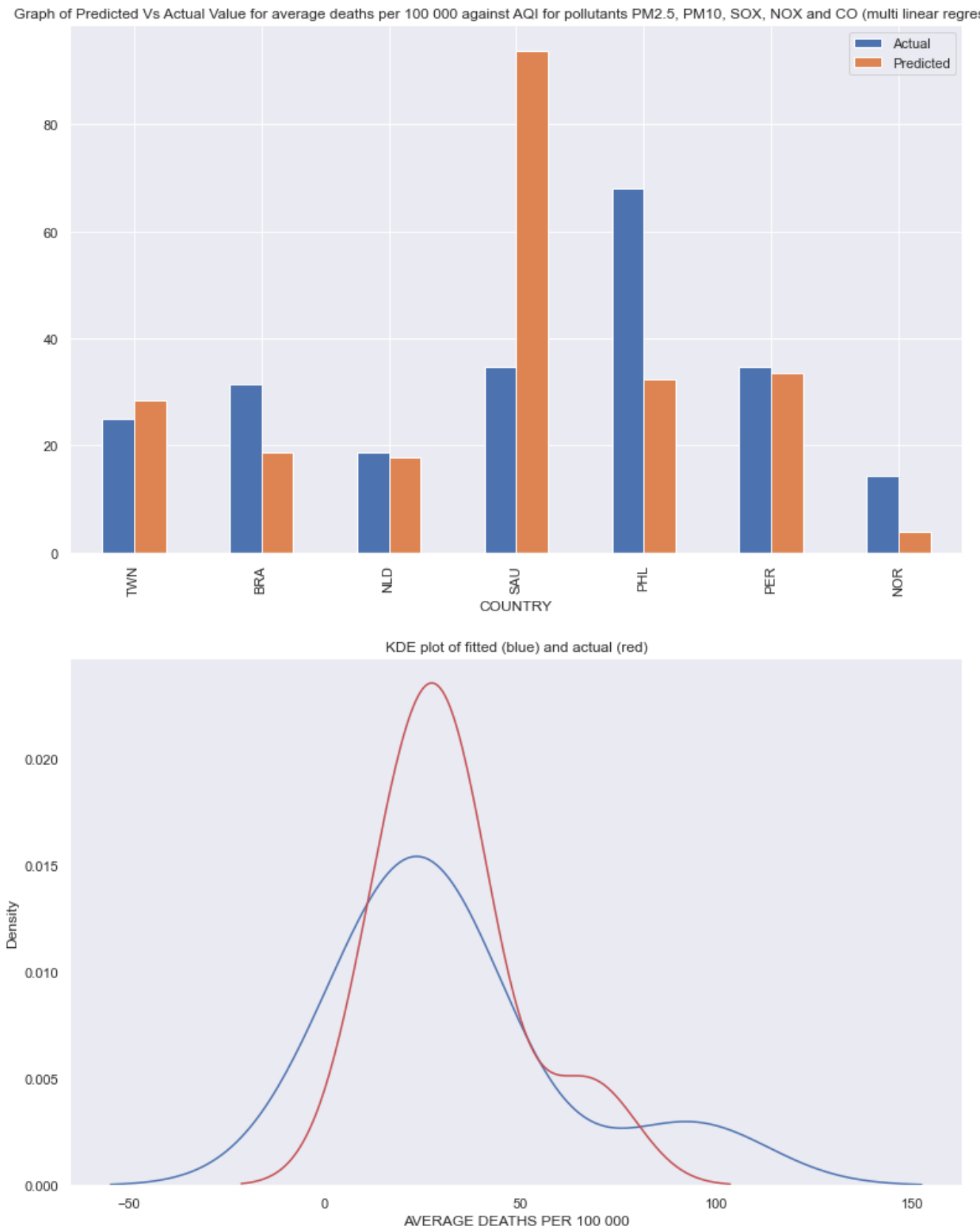
Equation of Line: y = 41.061* (PM2.5)
-37.366* (PM10)
-2.293* (NOX)
-6.897* (CO)
-17.695* (O3)
-3.847* (SOX)
+ 65.755



A multi linear regression model is done for AQI for all pollutants to investigate whether the combined trend will have a better linear correlation for average dalys per 100 000. Form the residual plot, it can be observed that the actual and fitted values do not coincide well, therefore a multi linear regression model is not suitable.

In [110…

```
health_deaths_pollution_qn1_dropna_noo3 = health_deaths_pollution_qn1[(health_deaths_pollution_qn1["POLLUTANT"]
                                                                       !="O3")]
health_deaths_pollution_qn1_dropna_noo3 = pd.pivot_table(health_deaths_pollution_qn1_dropna_noo3,index=["COUNTRY","AVERAGE DEATHS
                                                         values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_deaths_pollution_qn1_dropna_noo3 = health_deaths_pollution_qn1_dropna_noo3["AQI"]
health_deaths_pollution_qn1_dropna_noo3 = health_deaths_pollution_qn1_dropna_noo3.reset_index()
health_deaths_pollution_qn1_dropna_noo3 = health_deaths_pollution_qn1_dropna_noo3.set_index("COUNTRY")
health_deaths_pollution_qn1_dropna_noo3 = health_deaths_pollution_qn1_dropna_noo3.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_deaths_pollution_qn1_dropna_all,x_axis_list=["PM2.5","PM10","NOX","CO","SOX"],y_axis="AV
                         title="average deaths per 100 000 against AQI for pollutants PM2.5, PM10, SOX, NOX and CO (multi linear regre
```

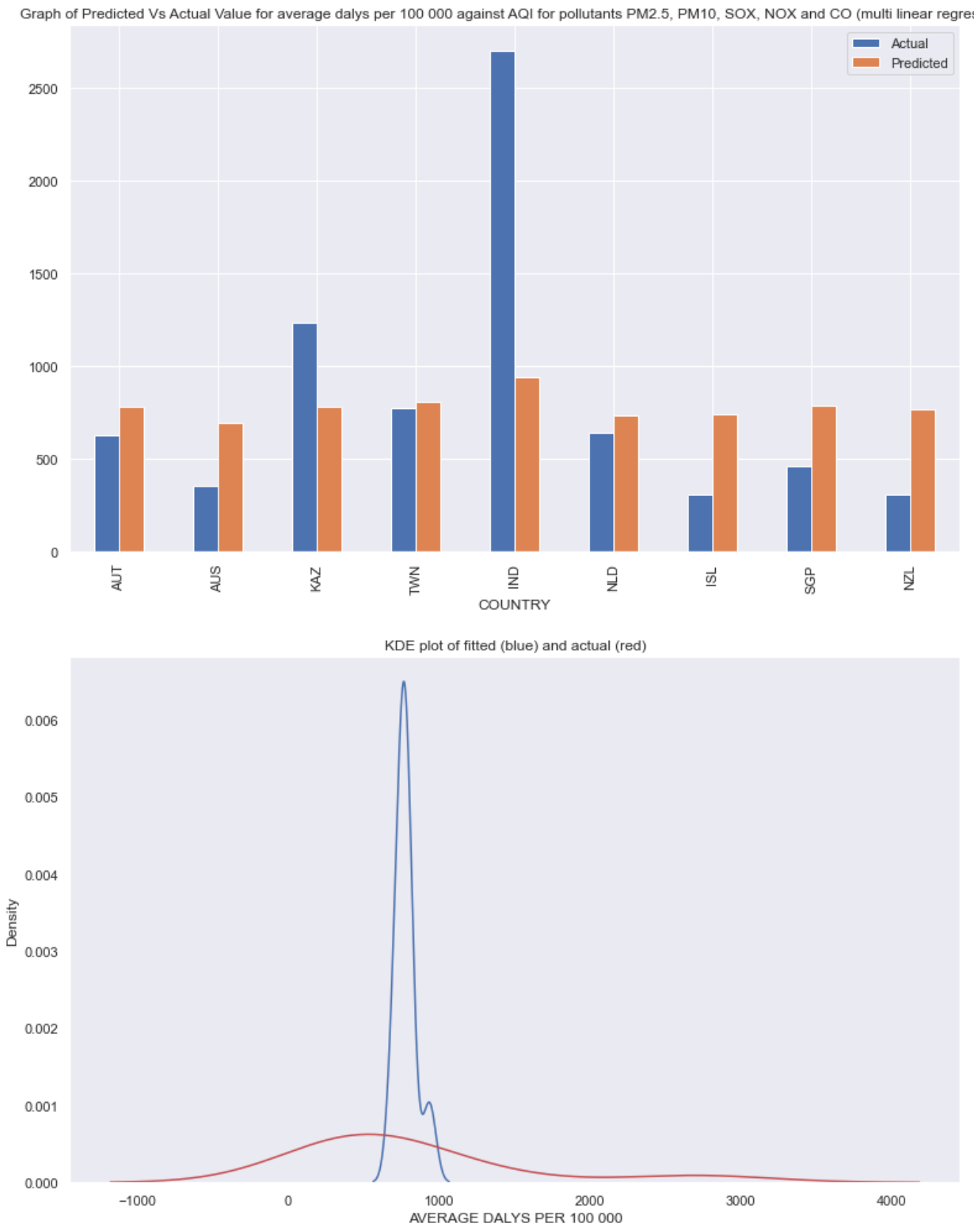


Equation of Line: y = 1.021* (PM2.5)
-0.64* (PM10)
-0.965* (NOX)
0.893* (CO)
-0.117* (SOX)
+ -5.418

A multi linear regression model is done for AQI for pollutants with non negative correlation (PM2.5, PM10, NOX, SOX and CO) to investigate whether the combined trend will have a better linear correlation for average deaths per 100 000. Form the residual plot, it can be observed that the actual and fitted values does follow a similar trend though did not coincide well at the break, therefore a multi linear regression model may not suitable.

In [111…

```
health_dalys_pollution_qn1_dropna_noo3 = health_dalys_pollution_qn1[(health_dalys_pollution_qn1["POLLUTANT"]
                                                                    !="O3")]
health_dalys_pollution_qn1_dropna_noo3 = pd.pivot_table(health_dalys_pollution_qn1_dropna_noo3,index=["COUNTRY","AVERAGE DALYS PER
                                                        values=["AQI"],columns=["POLLUTANT"],aggfunc='mean')
health_dalys_pollution_qn1_dropna_noo3 = health_dalys_pollution_qn1_dropna_noo3["AQI"]
health_dalys_pollution_qn1_dropna_noo3 = health_dalys_pollution_qn1_dropna_noo3.reset_index()
health_dalys_pollution_qn1_dropna_noo3 = health_dalys_pollution_qn1_dropna_noo3.set_index("COUNTRY")
health_dalys_pollution_qn1_dropna_noo3 = health_dalys_pollution_qn1_dropna_noo3.dropna(how='any',axis=0)
results_multilinregmodel(dataframe=health_dalys_pollution_qn1_dropna_noo3,x_axis_list=["PM2.5","PM10","NOX","CO","SOX"],y_axis="AV
                         title="average dalys per 100 000 against AQI for pollutants PM2.5, PM10, SOX, NOX and CO (multi linear regres
```



Equation of Line: y = 7.235* (PM2.5)
-7.811* (PM10)
-11.508* (NOX)
-5.297* (CO)
23.321* (SOX)
+ 685.944

A multi linear regression model is done for AQI for pollutants with non negative correlation (PM2.5, PM10, SOX, NOX and CO) to investigate whether the combined trend will have a better linear correlation for average dalys per 100 000. Form the residual plot, it can be observed that the actual and fitted values do not coincide well, therefore a multi linear regression model is not suitable.

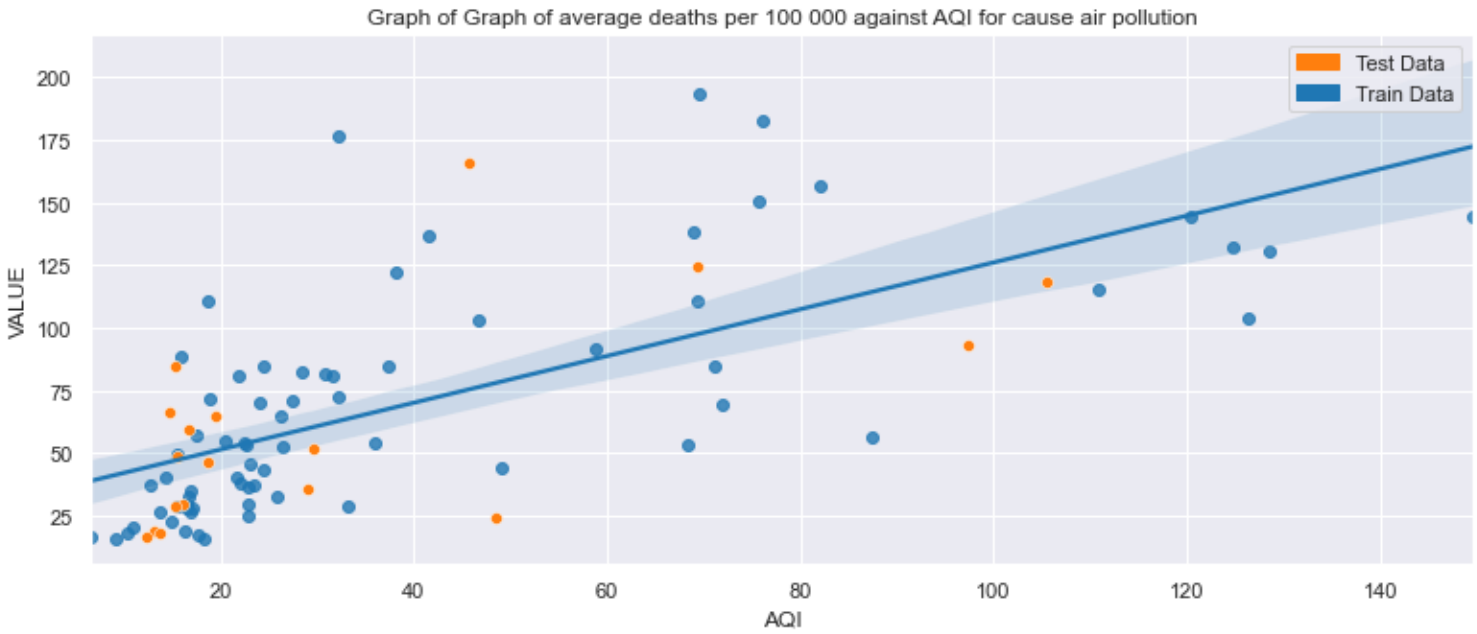
All SLRM below show moderate positive correlation. This is expected as with greater AQI value (i.e. worsened air quality), there will be more respiratory related / pollution related health problems, therefore an increase in average deaths / dalys per 100 000 of country population.

1. Suitable positive linear correlation model
    A. SLRM Average deaths per 100 000 against AQI for pollutant PM 2.5
2. May not be suitable linear correlation model (show some common trend for KDE though does not coincide fully)
    A. SLRM Average dalys per 100 000 against AQI for pollutant PM 2.5
    B. SLRM Average deaths per 100 000 against AQI for pollutant PM 10
    C. MLRM Average dalys per 100 000 against AQI for pollutant PM2.5 and PM 10
    D. MLRM Average deaths per 100 000 against AQI for pollutant PM2.5 and PM 10
    E. MLRM Average deaths per 100 000 against AQI for all pollutants
    F. MLRM Average deaths per 100 000 against AQI for pollutants PM2.5, PM10, SOX, NOX and CO
3. Not suitable linear correlation model
    A. SLRM Average dalys per 100 000 against AQI for pollutant PM 10
    B. MRLM Average dalys per 100 000 against AQI for pollutant CO, NOX, SOX and O3
    C. MLRM Average dalys per 100 000 against AQI for all pollutants
    D. MLRM Average dalys per 100 000 against AQI for pollutants PM2.5, PM10, SOX, NOX and CO
    E. MRLM Average deaths per 100 000 against AQI for pollutant CO, NOX, SOX and O3

Q2. Research Question 2 - Effect of air pollution on death cause (respiratory-related diseases/air-pollution related)

In [112…

```
health_by_cause_pollution_dropnoaqi_pivot_airpol = health_by_cause_pollution_dropnoaqi_pivot[
    health_by_cause_pollution_dropnoaqi_pivot["CAUSE OF HEALTH"]=="DEATH (AIR POLLUTION PER 100 000)"]
results_linregmodel(dataframe=health_by_cause_pollution_dropnoaqi_pivot_airpol,x_axis="AQI",y_axis="VALUE",
                    title="Graph of average deaths per 100 000 against AQI for cause air pollution")
```



Slope: 0.93354
P-value: 0.00000
R-value: 0.68452
MSE Train: 1104.28676
MSE Test: 1029.82659

MSE: 1029.8266
R²: 0.4686

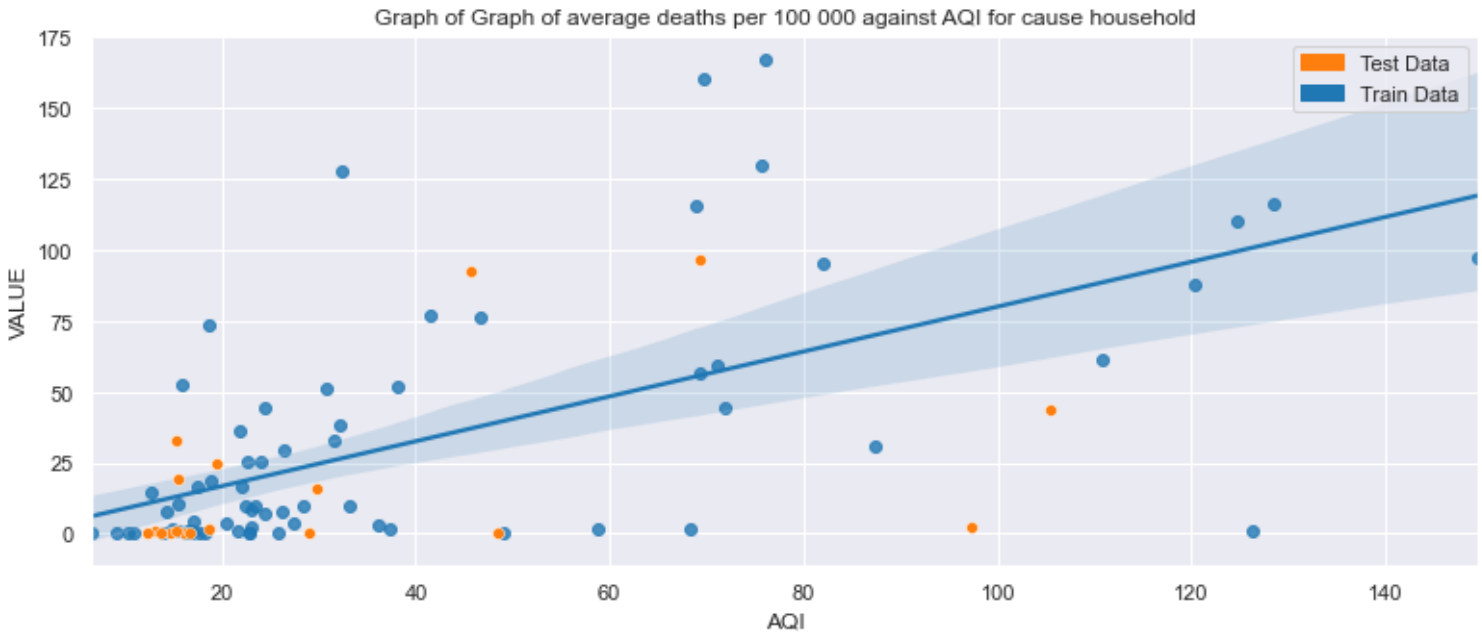
From figure 1.3.2, further investigation is done by performing the linear regression model on the death cause air pollution for average deaths per 100 000 against AQI. The P value is very small while the R value (correlation value) is 0.68018, indicating there is a moderate strong positive correlation.

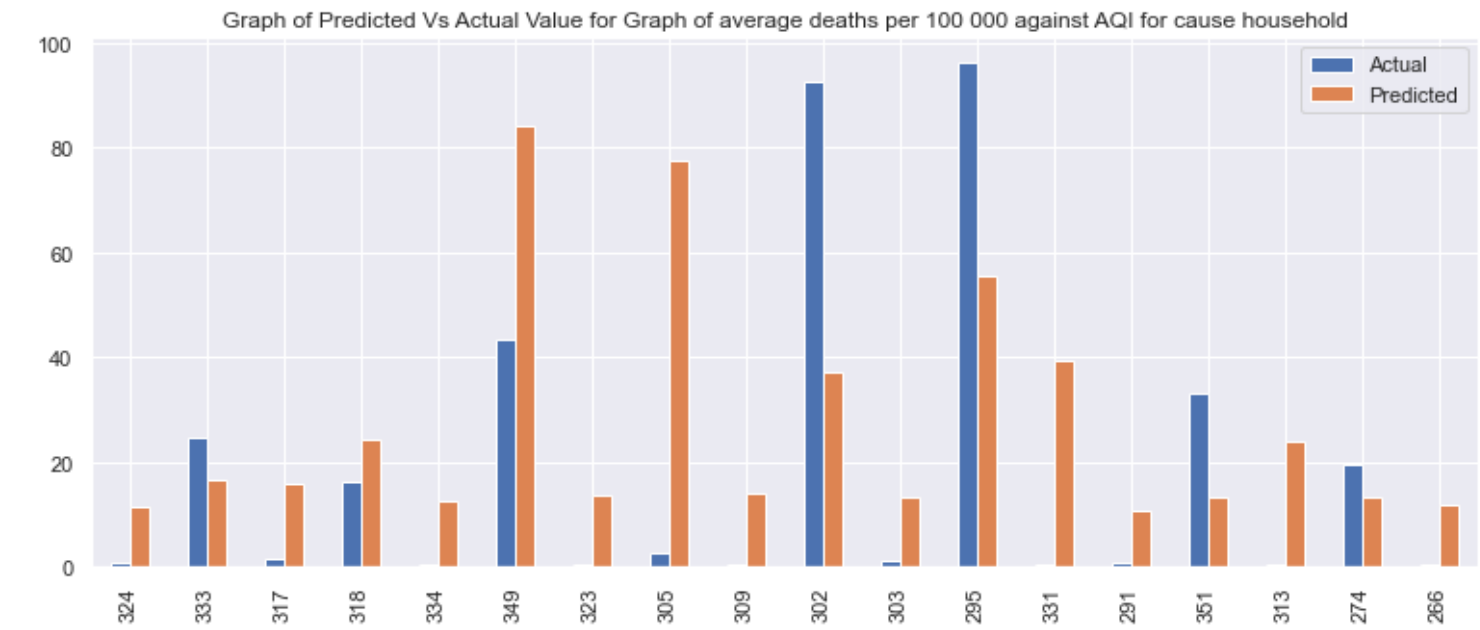
From the residual plot, it can be observed that the points are roughly randomly separated around the x axis, though the points are clustered around the x axis for smaller values of AQI. The KDE plot of fitted vs actual value does not coincide quite well, therefore a linear regression model may not be a suitable model.

However, we can still conclude that the graph of average deaths per 100 000 against AQI for death cause air pollution has a moderately strong positive correlation.

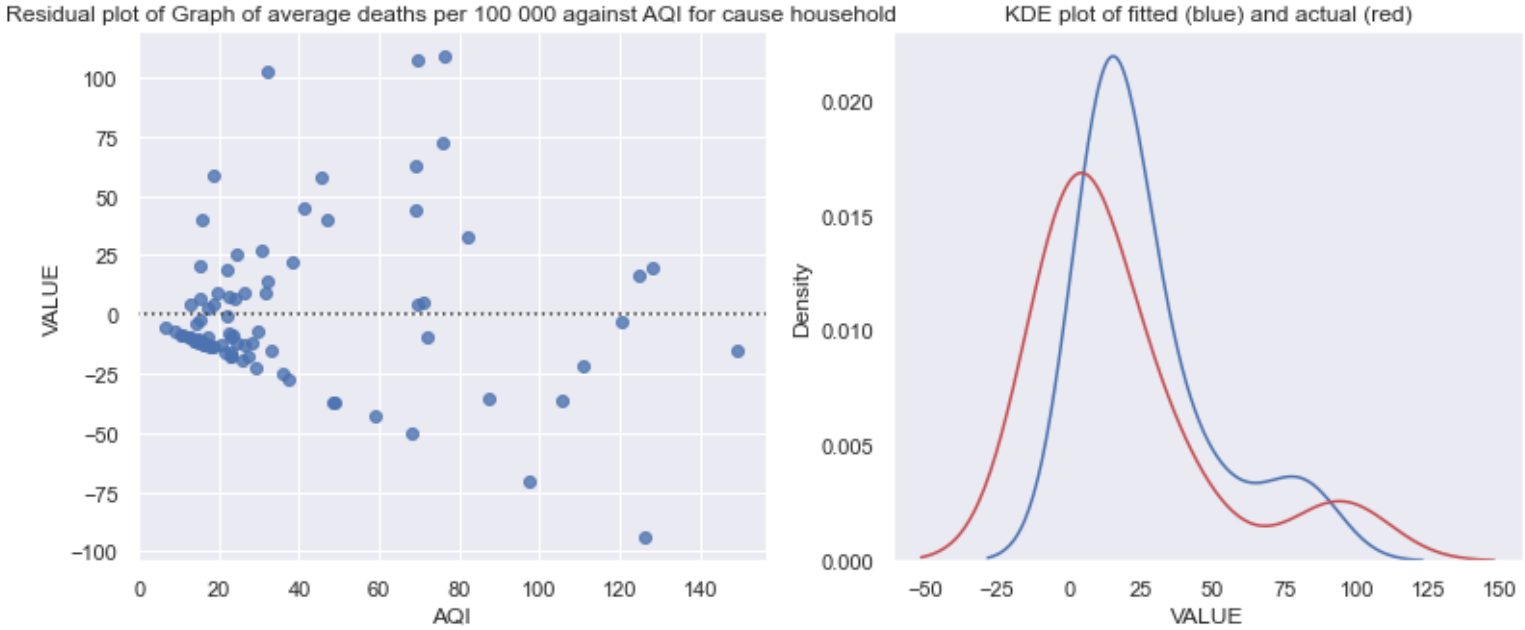
In [113…

```
health_by_cause_pollution_dropnoaqi_pivot_airpol = health_by_cause_pollution_dropnoaqi_pivot[
    health_by_cause_pollution_dropnoaqi_pivot["CAUSE OF HEALTH"]=="DEATH (HOUSEHOLD PER 100 000)"]
results_linregmodel(dataframe=health_by_cause_pollution_dropnoaqi_pivot_airpol,x_axis="AQI",y_axis="VALUE",
                    title="Graph of average deaths per 100 000 against AQI for cause household")
```





Slope: 0.78922
P-value: 0.00000
R-value: 0.61781
MSE Train: 1127.26673
MSE Test: 885.39029



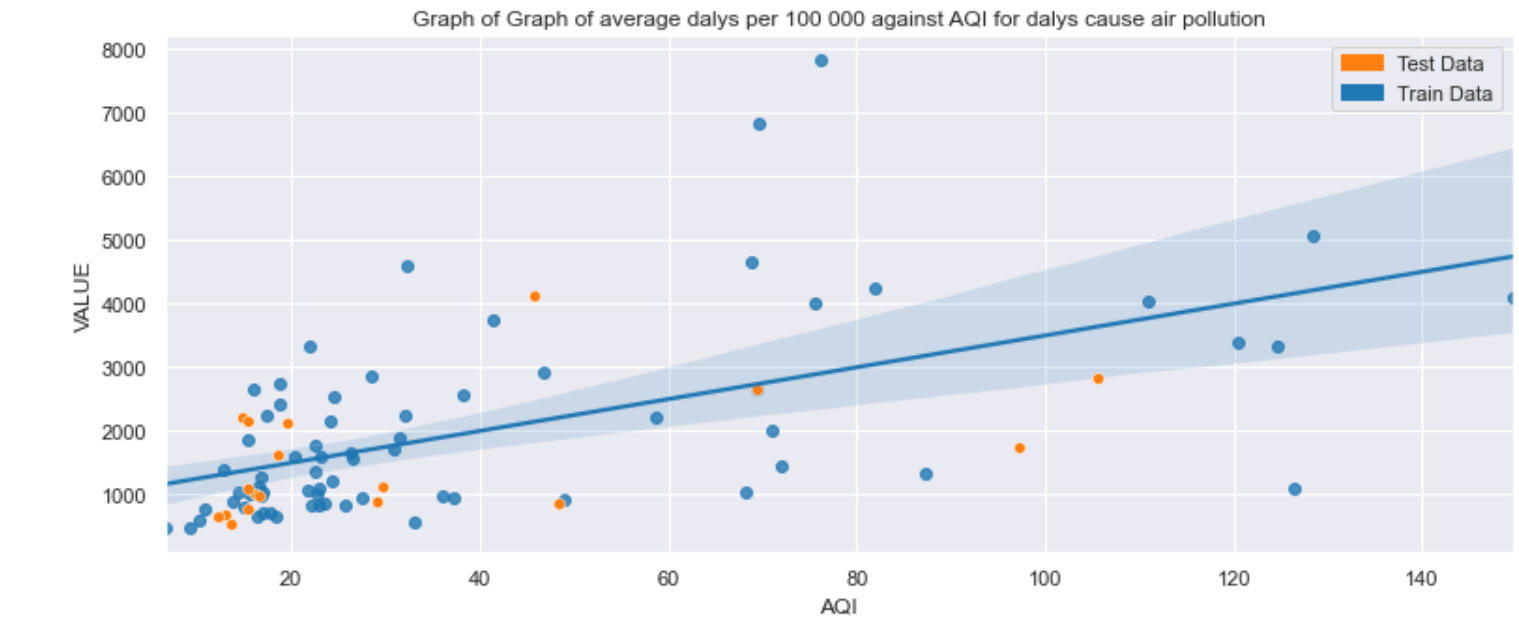
MSE: 885.3903
R²: 0.3817

From figure 1.3.2, further investigation is done by performing the linear regression model on the death cause household for average deaths per 100 000 against AQI. The P value is very small while the R value (correlation value) is 0.60683, indicating there is a moderate positive correlation. From the residual plot, it can be observed that the points are roughly randomly separated around the x axis, though the points are clustered around the x axis for smaller values of AQI. The KDE plot of fitted vs actual value follows the same trend though does not coincide nicely, therefore a linear regression model may not be a suitable model.

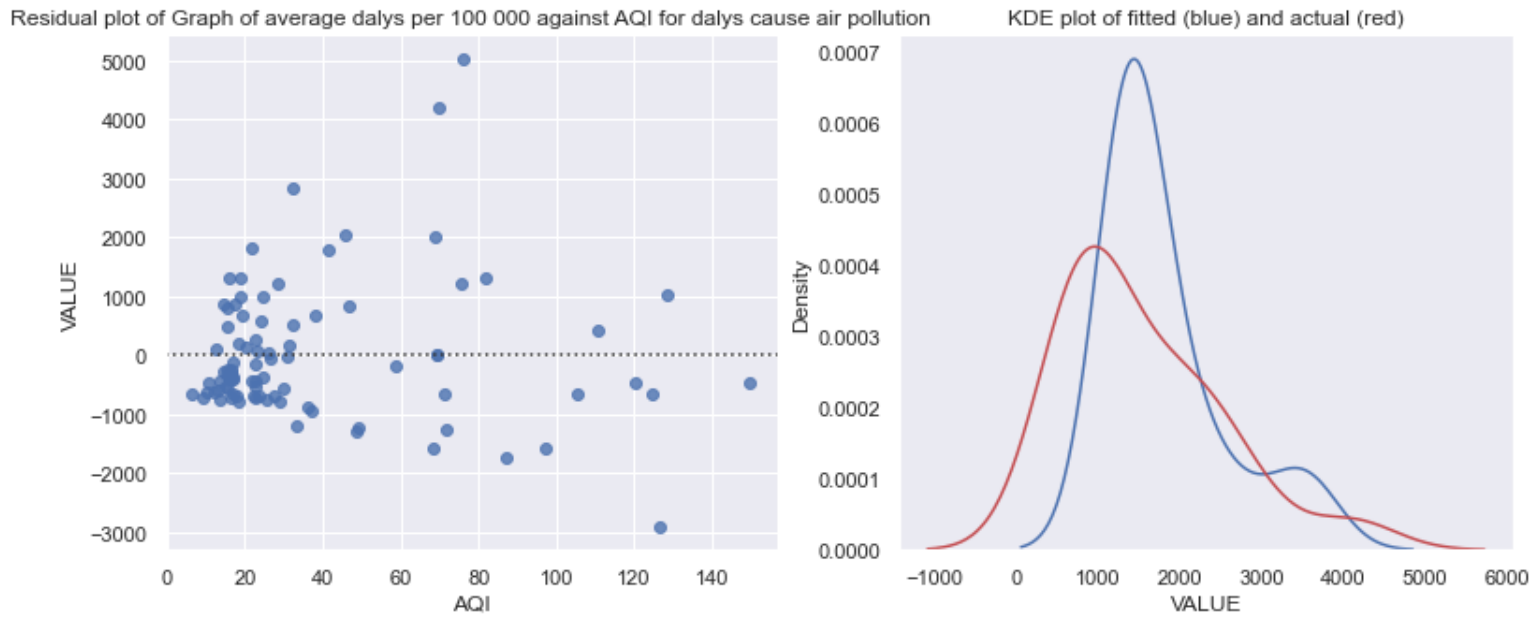
However, we can still conclude that the graph of average deaths per 100 000 against AQI for death cause household has a moderate positive correlation.

In [114…

```
health_by_cause_pollution_dropnoaqi_pivot_airpol = health_by_cause_pollution_dropnoaqi_pivot[
    health_by_cause_pollution_dropnoaqi_pivot["CAUSE OF HEALTH"]=="DALYS (AIR POLLUTION PER 100 000)"]
results_linregmodel(dataframe=health_by_cause_pollution_dropnoaqi_pivot_airpol,x_axis="AQI",y_axis="VALUE",
                    title="Graph of average dalys per 100 000 against AQI for dalys cause air pollution")
```



Slope: 25.03413
P-value: 0.00000
R-value: 0.56440
MSE Train: 1497824.83263
MSE Test: 798234.86867



MSE: 798234.8687
R^2: 0.3185

From figure 1.3.2, further investigation is done by performing the linear regression model on the dalys cause air pollution for average dalys per 100 000 against AQI.The P value is very small while the R value (correlation value) is 0.68018, indicating there is a moderate strong positive correlation. From the residual plot, it can be observed that the points are roughly randomly separated around the x axis, though the points are clustered around the x axis for smaller values of AQI. The KDE plot of fitted vs actual value does not coincide quite well, therefore a linear regression model may not be a suitable model.

Therefore, we can conclude that a linear regression model is not suitable for the graph of average dalys per 100 000 against AQI for dalys cause air pollution has a moderate positive correlation.

All SLRM below show moderate positive correlation.

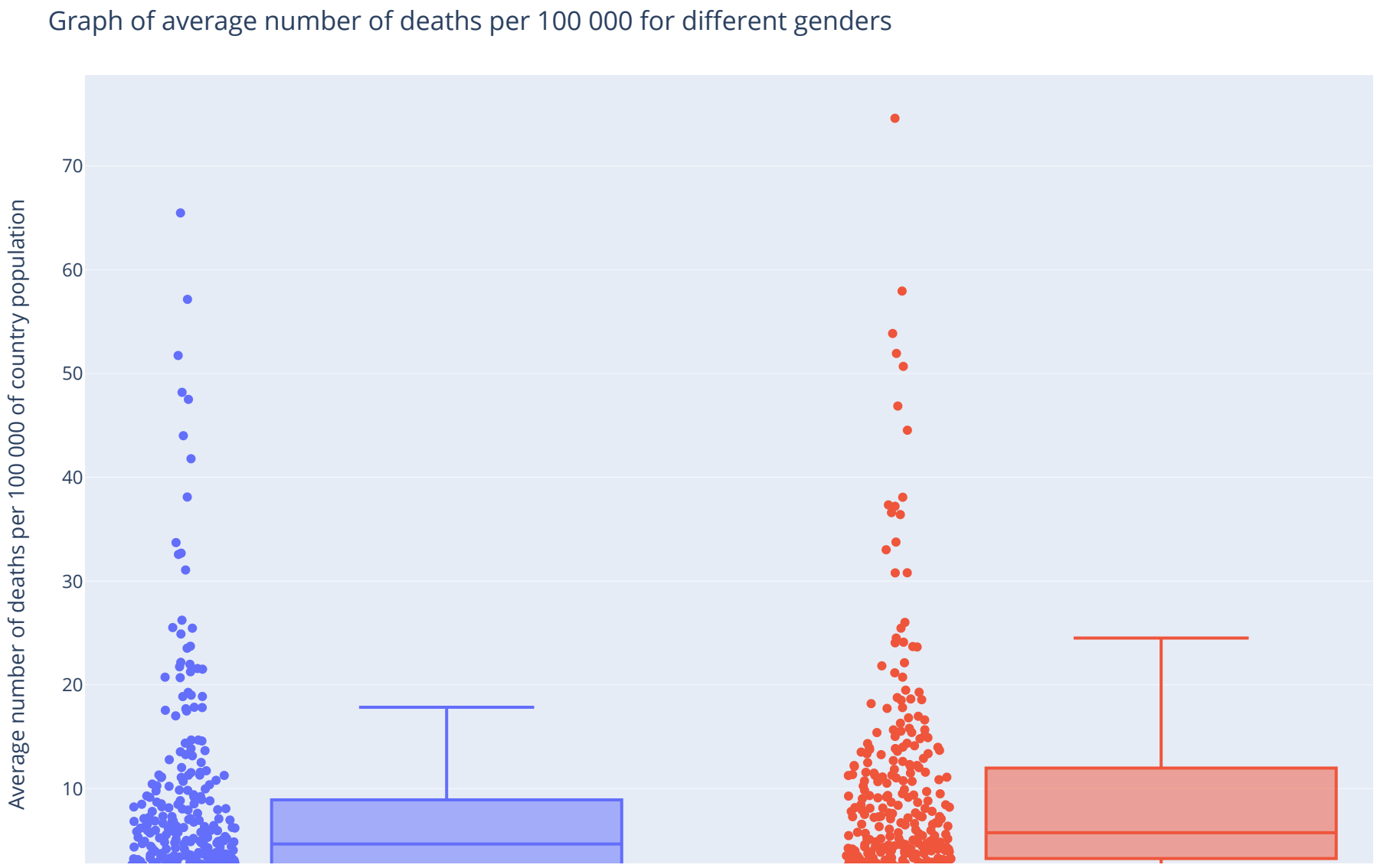
1. May not be suitable linear correlation model
    A. SLRM Average deaths per 100 000 against AQI for death cause pollution
    B. SLRM Average deaths per 100 000 against AQI for death cause household
    C. SLRM Average dalys per 100 000 against AQI for dalys cause air pollution

Q3. Effect of air pollution on health of different genders

In [115…

```
gender_pollution_qn6 = health_gender_pollution_qn3_dropnoaqi_pivot.copy()

fig = px.box(data_frame=gender_pollution_qn6,y="VALUE",x="SEX", color="SEX",points="all",hover_name="COUNTRY")

fig.update_layout(
    title="Graph of average number of deaths per 100 000 for different genders",
    xaxis_title="SEX",
    yaxis_title="Average number of deaths per 100 000 of country population",
    legend_title="SEX",
    height=700
)
caption = "Figure 3.3: DEATHS for different genders (plotly)"
```

```
fig.add_annotation(text=caption,
                   xref="paper", yref="paper",
                   x=0.5, y=-0.2, showarrow=False)
fig.show()
```



Based on this graph above and figure 3.2, it can be observed that the median value for females (of 4.66) is slightly lower than males (of 5.75), with the median value of This shows that death of females are less affected by air pollution as compared to men. Both the range and interquartile range of females (17.62 and 6.49 respectively) is smaller than male (24.29 and 8.7175 respectively), showing that the average number of deaths per 100 000 for females are more clustered and spread over a smaller range of values. Both females and males show several outliers above Q3 + 1.5 IQR, showing that there are few exceptions with extremely high average deaths per 100 000.

For both females and males, the country Bulgaria (country code BGR) has the highest average number of deaths which are 65.48 and 74.6 respectively. This may be due to males being more exposed to air pollution outdoors, or a more plausible reason that females already having a longer life expectancy and therefore less average number of deaths than males.

Q4. Effect of air pollution on health of people from 2014 to 2017

From figure 4.1, it can be observed that all years show a positive correlation between average number of deaths per 100 000 against AQI value. The year 2014 shows the steepest linear regression plot, followed by 2015, 2016 then 2017. This indicates that for a smaller increase in AQI index, the extent to which the number of deaths increased in 2015 is the highest, followed by 2015, 2016 and 2017, which happens to follow chronological order, which may be due to healthcare improving faster than air pollution quality from 2014 to 2017.

From figure 4.2, in 2016 and 2017, there are a few datapoints with very high AQI values above 120 unlike in 2014 and 2015.

From 2014 to 2017, it can be seen that the datapoints are generally clustered at the lower AQI and lower average deaths per 100 000 population value.

From figure 4.3, it can be observed that all years show a positive correlation between average number of dalys per 100 000 against AQI value. The year 2014 shows the steepest linear regression plot, followed by 2015, 2016 then 2017. This indicates that for a smaller increase in AQI index, the extent to which the number of dalys increased in 2014 is the highest, followed by 2015, 2016 and 2017, which happens to follow chronological order.

From figure 4.4, for years 2014 to 2017 the datapoints are clustered around the low AQI and low number of dalys per 100 000 population for all years, however with a larger variation of dalys per 100 000 of the population for the lower AQI values.

In 2016 and 2017, there are a few datapoints with very high AQI values above 120 unlike in 2014 and 2015.

Q5. Effect of air pollution on health of people across different geographical location

From figure 5.1, it can be seen different countries have varying index for average that North and South America has relatively low index as shown by the light colour. Countries in Africa show a darker shade of colour, indicating that the index is relatively higher.

From the graph, it can also be seen that (coutry code of AUS) Australia has the highest index for dalys of 1.49, indicating it has a high number of dalys per 100 000 of country population to AQI index ratio.

From figure 5.2, it can be seen that North and South America has relatively low index as shown by the light colour. Countries in Africa show a darker shade of colour, indicating that the index is relatively higher.

From the graph, it can also be seen that Brunei (country code of BRN) has the highest index for deaths of 2.31, indicating it has a high number of deaths per 100 000 of country population to AQI index ratio.

Australia (country code of AUS) still has a significantly high index (though slightly lower than that for dalys) of 1.04.

Q6. Trend of air pollution/health factors across different years

From figure 6.1, in general the average number of deaths per 100 000 of the country population has been decreasing progressively. It can also be noted that Papau New Guinea (country code of PNG) has stayed the country with the highest average deaths per 100 000 of population over the years as compared to other countries which have much lower average deaths per 100 000 of population.

From figure 6.2, in general, indicating that the average number of dalys per 100 000 of the country population has been decreasing, with the decrease more evident than in figure 6.1.

It can also be noted several African countries have high average dalys per 100 000 in 1990, with countries SSD (South Sudan), TCD (Chad), AGO (Angola), GNQ (Equatorial Guinea), NER (Niger), GIN (Guinea), SLE (Sierra Leone) and other countries like AFG (Afghanistan), KHM (Cambodia), LAO (Laos) having average dalys of above 20k.

From figure 6.3.1 and figure 6.3.2, it can be observed that the number of outliers with very high amount of emissions decreases over the years. Generally, the pollutant CO has the largest range of amount of emissions. It can also be noted that USA (United sates of America) is the country with one of the most amount of emissions for all the different pollutants. Majority of the pollutants are clustered in the 0k to 5k kilotons of emissions.

From figure 6.4, pollutant PM 2.5 has a larger range of AQI values, followed by PM10, while O3, NOX and CO are mostly clustered at very low AQI values ranging between 0 to 50.

In 2014, it is evident that there is an outlier for pollutant PM 2.5 of extremely high AQI value of 338.5 for the country Denmark (country code of DNK) and an outlier for pollutant PM10 of high AQI value of 166.875 for the country India (country code of IND).

From figure 6.5.1, it can be observed that the death causes ozone and pm have lower number of deaths per 100 000 generally. For causes air pollution, chronic respiratory disease, household and lower respiratory infection, the range of number of deaths per 100 000 is larger.

For the chronic respiratory disease cause, there is one consistent outlier from 1995 to 2016 which remains the highest number of deaths which is the country Papau New Guinea (PNG). It can also be observed the Papau New Guinea also has the highest number of deaths for causes air pollution and household over the years.

From figure 6.5.2, the death cause that contributes the most average number of deaths is air polution, followed by chronic respiratory disease, lower respiratory infection, household, PM then ozone.

From figure 6.6.1, the range of number of dalys per 100 000 decreases over the years, with majority of countries having small number of dalys per 100 000 of country population. It can also be observed that the cause PM has a smaller range of number of dalys per 100 000 value as compared to air pollution and household, solid fuel causes.

Egypt (country code EGY) remains the country with the highest dalys per 100 000 for the cause PM.

From 2005 onwards, Chad (country code TCD) remains the country with the highest dalys per 100 000 for the cause air pollution. From 2006 onwards, Chad (country code TCD) remains the country with the highest dalys per 100 000 for the cause of household, solid fuel.

From figure 6.6.2, the dalys cause that contributes the most average number of dalys is air polution, followed by household, solid fuel then PM.

From figure 6.7, it can be observed that for North and South America, the colour shade turns lighter over the years, indicating that the AQI value decreased over the years. As for the African continent, majority of the countries remain having high AQIs over the years.

Australia (country code AUS) and New Zealand (country code NZL) remain light coloured from the start, having the lowest values of 15.83 and 22.90 respectively in 1990 all the way to 2011, and remain one of the countries with the lowest AQI from 2011 to 2019.

USA, Canada (country code CAN), Finland (country code FIN), Sweden (country code SWE) are among the countries that have the best improvement in AQI over the years, with the largest change in shades of color from dark blue to light green.

Figure 6.8 shows a clear decreasing linear trend, indicating that the number of average deaths per 100 000 is decreasing as time progresses.

Figure 6.9 shows a clear decreasing linear trend, indicating that the number of average dalys per 100 000 is decreasing as time progresses.

Figure 6.10 shows a general decreasing trend for the AQI value over the years. From 2014 to 2015, there is a greater range of AQI values for the different countries. Interestingly, there is a sudden increase in AQI values from 2016 to 2017 and from 2020 to 2021, with the increase from 2016 to 2017 being a sharper increase than from 2020 to 2021. The sudden decrease in AQI value from 2016 to 2017 may be accounted by the initial commencement of the Paris Climate Agreement in 2016, and the increase in AQI from 2020 to 2021 by the slight reopening of the economy due to the COVID-19 Pandemic in 2020.

Testing and Verification of Results

Testing and Verification of Results is done for the linear regression models plotted for Research Question 1 and 2 by plotting the Residual and KDE plots to check whether the linear regression models are suitable for the dataset, as well as barcharts on top of KDE plots for multi linear regression models.

Different representations of the same data are also done to observe the data from different visual points of view. For example, a heatmap and stripplot were used to investigate the same data of how the average number of deaths/dalys per 100 000 of country population for different deaths/dalys causes vary over the different years.

Conclusion and recommendations

In conclusion, there are many factors of air pollution that possibly effect different aspects of health of people.

These factors include type of pollutants, chronological time series, geographical location and different aspects include different causes and gender. The extent to which different factors affect and different aspects are affected vary from pollutant type to pollutant type / death cause to death cause. These trends show a correlation and may not necessarily be a direct causation as there are many factors in play. However, the postiive correlation for many of the different pollutant to health still signals the importance of monitoring the levels of air pollution, which can play a role in improving the health of people across the world.

It is heartening to note that the number of deaths/dalys as well as AQI values have decreased over the years and therefore we should continue doing so, not only protecting the people but also protecting the world with improved control over air pollution.

Further investigation can be done on other factors including types of air pollution (i.e. industrial air pollution, household air pollution, burning of forest pollutions), and alternative models can be used to investigate whether the models can be better fitted (if linear regression model is not appropriate, as a further investigation maybe logarithm functions, exponential functions, curvature quadratic models).

References

The numbered list of datasets below are hyperlinked to the website for which the respective csv data is obtained.

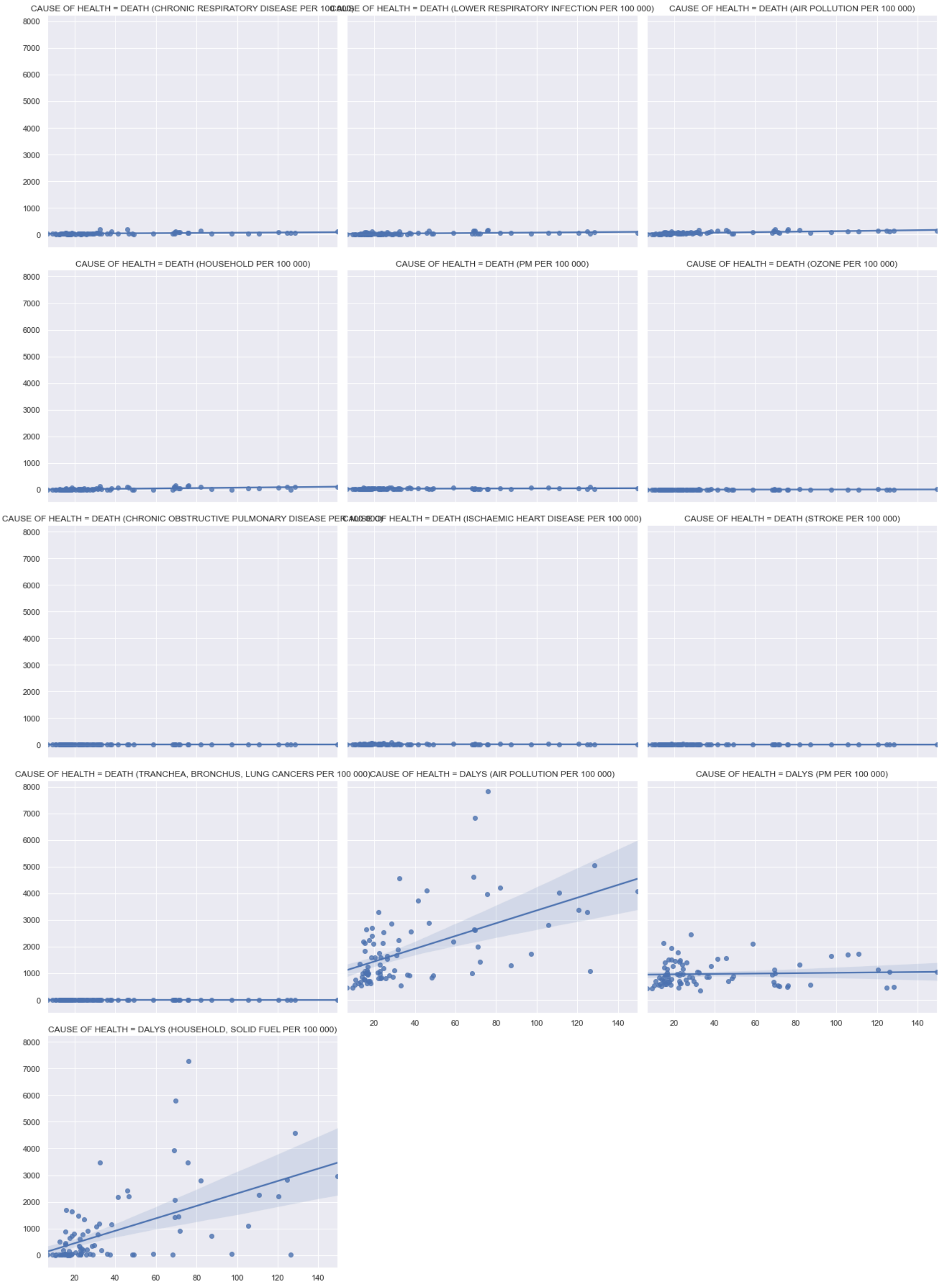
1. **air_pollutant_co2.csv**
2. **air_pollutant_lead.csv**
3. **air_pollution_exposure.csv**
4. **aqi_breakpoints.csv**
5. **death-rates-from-air-pollution.csv**
6. **disease-burden-by-risk-factor.csv**
7. **parameters.csv**
8. **pneumonia-death-rates-age-standardized.csv**
9. **respiratory-disease-death-rate.csv**
10. **singstat_subcollation.csv** (actual link needs to be pasted in to work: "https://www.tablebuilder.singstat.gov.sg/publicfacing/createDataTable.action?refId=14589")
11. **stats_oecd_pollutants.csv**
12. **who_respiratory_pollution_caused_rate.csv)**
13. **waqi-covid19-airqualitydata-2015H1.csv**
14. **waqi-covid19-airqualitydata-2016H1.csv**
15. **waqi-covid19-airqualitydata-2017H1.csv**
16. **waqi-covid19-airqualitydata-2018H1.csv**
17. **waqi-covid19-airqualitydata-2019Q1.csv**
18. **waqi-covid19-airqualitydata-2019Q2.csv**
19. **waqi-covid19-airqualitydata-2019Q3.csv**
20. **waqi-covid19-airqualitydata-2019Q1.csv**
21. **waqi-covid19-airqualitydata-2020Q1.csv**
22. **waqi-covid19-airqualitydata-2020Q2.csv**
23. **waqi-covid19-airqualitydata-2020Q3.csv**
24. **waqi-covid19-airqualitydata-2020Q1.csv**
25. **waqi-covid19-airqualitydata-2021.csv**
26. https://en.wikipedia.org/wiki/List_of_countries_by_past_and_projected_future_population
27. https://en.wikipedia.org/wiki/List_of_ISO_3166_country_codes

Appendix

In [116…

```
q2_health_cause = sns.FacetGrid(health_by_cause_pollution_dropnoaqi_pivot, col="CAUSE OF HEALTH",height=5,aspect=1.2,col_wrap=3)
q2_health_cause.map_dataframe(sns.regplot, x="AQI", y="VALUE")
```

Out[116… <seaborn.axisgrid.FacetGrid at 0x1b92eb54070>



In [117…

```
health_by_cause_pollution.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 196 entries, AFG to ZWE
Data columns (total 14 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)         196 non-null    float64
 1   DEATH (LOWER RESPIRATORY INFECTION PER 100 000)         196 non-null    float64
 2   DEATH (AIR POLLUTION PER 100 000)                       196 non-null    float64
 3   DEATH (HOUSEHOLD PER 100 000)                           196 non-null    float64
 4   DEATH (PM PER 100 000)                                  196 non-null    float64
 5   DEATH (OZONE PER 100 000)                               196 non-null    float64
```

```
 6   DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)  183 non-null    float64
 7   DEATH (ISCHAEMIC HEART DISEASE PER 100 000)                183 non-null    float64
 8   DEATH (STROKE PER 100 000)                                 183 non-null    float64
 9   DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)       183 non-null    float64
 10  DALYS (AIR POLLUTION PER 100 000)                          195 non-null    float64
 11  DALYS (PM PER 100 000)                                     195 non-null    float64
 12  DALYS (HOUSEHOLD, SOLID FUEL PER 100 000)                  195 non-null    float64
 13  AQI                                                        88 non-null     float64
dtypes: float64(14)
memory usage: 23.0+ KB
```

After joining the waqi_data_total to the dataset, it can be observed that there are a few columns wehere AQI for the respective countries is not present. An attempt to fill the remaining AQI data with only PM 2.5 data from world_oecd_pm25_data_total was made. However, it showed that the data was skewed withe the AQI coming from only PM 2.5 deviating a lot from the other AQI values, therefore not taken as a valid result.

In [118…

```
def fill_na_pm25_data(row):
    if (str(row.AQI)=="nan"):
        if (row.name in world_oecd_pm25_data_total.index):
            aqi_value = world_oecd_pm25_data_total.loc[row.name,"AQI"]
            row.AQI = aqi_value
    return row
```

In [119…

```
health_by_cause_pollution_replaceaqi25 = health_by_cause_pollution.apply(fill_na_pm25_data, axis='columns')
health_by_cause_pollution_dropnoaqipm25 = health_by_cause_pollution_replaceaqi25.dropna(subset=["AQI"], axis=0)
health_by_cause_pollution_replaceaqi25.head()
```

Out[119…

COUNTRY	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTION PER 100 000)	DEATH (AIR POLLUTION PER 100 000)	DEATH (HOUSEHOLD PER 100 000)	DEATH (PM PER 100 000)	DEATH (OZONE PER 100 000)	DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)	DEATH (ISCHAEMIC HEART DISEASE PER 100 000)	DEATH (STROKE PER 100 000)	DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)	DALYS (AI POLLUTIOI PER 10 000
AFG	90.176087	109.232365	252.842551	203.967377	45.979634	5.797506	3.586667	20.926667	7.930000	0.703333	7435.33046
AGO	66.860067	151.220258	163.970719	131.054799	28.843601	7.204439	1.696667	5.623333	3.646667	0.103000	7235.16728
ALB	27.764644	30.495357	59.291351	35.145038	22.319736	2.600926	3.643333	33.930000	19.136667	4.183333	1596.48199
AND	23.423892	21.767718	21.150254	0.397801	18.356823	2.874569	NaN	NaN	NaN	NaN	619.00522
ARE	46.630824	52.147546	84.592507	1.472571	79.261157	5.300451	1.216667	8.490000	3.096667	0.406667	914.90956

In [120…

```
health_by_cause_pollution_replaceaqi25.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 196 entries, AFG to ZWE
Data columns (total 14 columns):
 #   Column                                                     Non-Null Count  Dtype
---  ------                                                     --------------  -----
 0   DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)            196 non-null    float64
 1   DEATH (LOWER RESPIRATORY INFECTION PER 100 000)            196 non-null    float64
 2   DEATH (AIR POLLUTION PER 100 000)                          196 non-null    float64
 3   DEATH (HOUSEHOLD PER 100 000)                              196 non-null    float64
 4   DEATH (PM PER 100 000)                                     196 non-null    float64
 5   DEATH (OZONE PER 100 000)                                  196 non-null    float64
 6   DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)  183 non-null    float64
 7   DEATH (ISCHAEMIC HEART DISEASE PER 100 000)                183 non-null    float64
 8   DEATH (STROKE PER 100 000)                                 183 non-null    float64
 9   DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)       183 non-null    float64
 10  DALYS (AIR POLLUTION PER 100 000)                          195 non-null    float64
 11  DALYS (PM PER 100 000)                                     195 non-null    float64
 12  DALYS (HOUSEHOLD, SOLID FUEL PER 100 000)                  195 non-null    float64
 13  AQI                                                        164 non-null    float64
dtypes: float64(14)
memory usage: 23.0+ KB
```

In [121…

```
health_gender_pollution_qn3_replaceaqipm25 = health_gender_pollution_qn3.apply(fill_na_pm25_data, axis='columns')
health_gender_pollution_qn3_replaceaqipm25 = health_gender_pollution_qn3_replaceaqipm25.dropna(subset=["AQI"], axis=0)
health_gender_pollution_qn3_replaceaqipm25.head()
```

Out[121…

COUNTRY	SEX	DEATH (CHRONIC OBSTRUCTIVE PULMONARY DISEASE PER 100 000)	DEATH (ISCHAEMIC HEART DISEASE PER 100 000)	DEATH (LOWER RESPIRATORY INFECTIONS PER 100 000)	DEATH (STROKE PER 100 000)	DEATH (TRANCHEA, BRONCHUS, LUNG CANCERS PER 100 000)	AQI
AFG	FMLE	3.47	18.23	17.22	8.57	0.380	162.249016
AFG	MLE	3.70	23.57	15.38	7.30	1.020	162.249016
AGO	FMLE	1.82	5.50	19.13	4.06	0.079	155.592242
AGO	MLE	1.57	5.75	19.14	3.23	0.130	155.592242
ALB	FMLE	2.47	27.78	2.10	18.63	2.860	153.573403

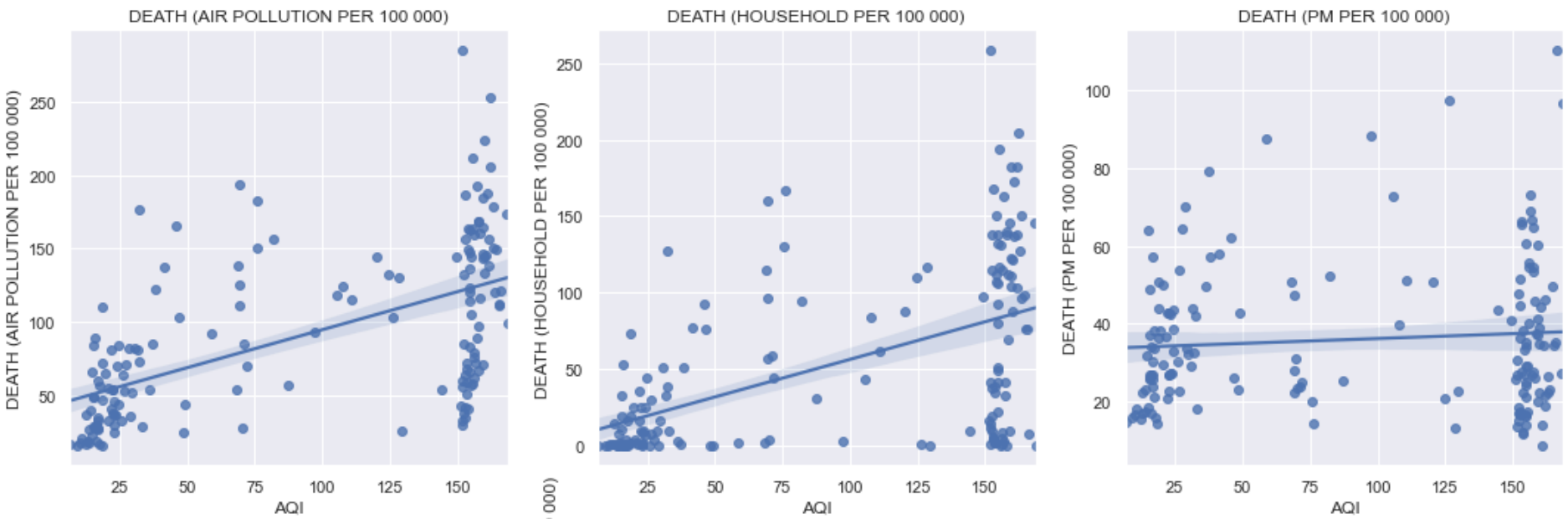
In [122…
```
health_by_cause_pollution_dropnoaqipm25_pivot = health_by_cause_pollution_dropnoaqipm25.reset_index()
health_by_cause_pollution_dropnoaqipm25_pivot = pd.melt(health_by_cause_pollution_dropnoaqipm25_pivot,id_vars=["COUNTRY","AQI"],va
                                                        health_by_cause_pollution_dropnoaqipm25_pivot.columns[1:-1],value_name="VALUE"
                                                       var_name="CAUSE OF HEALTH")
health_by_cause_pollution_dropnoaqipm25_pivot.head()
```

Out[122…

	COUNTRY	AQI	CAUSE OF HEALTH	VALUE
0	AFG	162.249016	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	90.176087
1	AGO	155.592242	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	66.860067
2	ALB	153.573403	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	27.764644
3	ARE	37.280818	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	46.630824
4	ARG	14.292142	DEATH (CHRONIC RESPIRATORY DISEASE PER 100 000)	36.860913

In [123…
```
plt.figure(figsize=(15,8))
plt.title("Graph of deaths associated with different causes per 100 000 "+
          "of country population against overall AQI index value \n(Countries without AQI Values from WAQI are filled in with othe
sns.scatterplot(x="AQI",y="VALUE",hue="CAUSE OF HEALTH",data=health_by_cause_pollution_dropnoaqipm25_pivot,style="CAUSE OF HEALTH"
plt.ylabel("DEATHS/DALYS per 100 000 of population")
plt.legend(bbox_to_anchor=(1.01,1))
plt.show()
```

In [124…
```
fig, axes = plt.subplots(nrows=3, ncols=3,  tight_layout=True,figsize=(15,15))

i = 0
for ax, c in zip(axes.ravel(), health_by_cause_pollution_dropnoaqipm25.iloc[:,2:].index):
    sns.regplot(x="AQI",y=health_by_cause_pollution_dropnoaqipm25.columns[2+i],
                data=health_by_cause_pollution_dropnoaqipm25.loc[:,[health_by_cause_pollution_dropnoaqipm25.columns[2+i],"AQI"]],a
    ax.set_title(health_by_cause_pollution_dropnoaqipm25.columns[2+i])
    i = i+1

plt.show()
```



In [125…

```
world_oecd_pollutants_data_byyear = world_oecd_pollutants_data.groupby(by=["COUNTRY","YEAR","POLLUTANT"])[["VALUE"]].mean()
world_oecd_pollutants_data_byyear = world_oecd_pollutants_data_byyear.groupby(by=["COUNTRY","YEAR"])[["VALUE"]].sum()
world_oecd_pollutants_data_byyear = world_oecd_pollutants_data_byyear.reset_index()
world_oecd_pollutants_data_byyear.head()
```

Out[125…

	COUNTRY	YEAR	VALUE
0	AUS	1990	2030.843588
1	AUS	1991	2024.184882
2	AUS	1992	2076.770529
3	AUS	1993	2146.895118
4	AUS	1994	2178.151647

In [126…

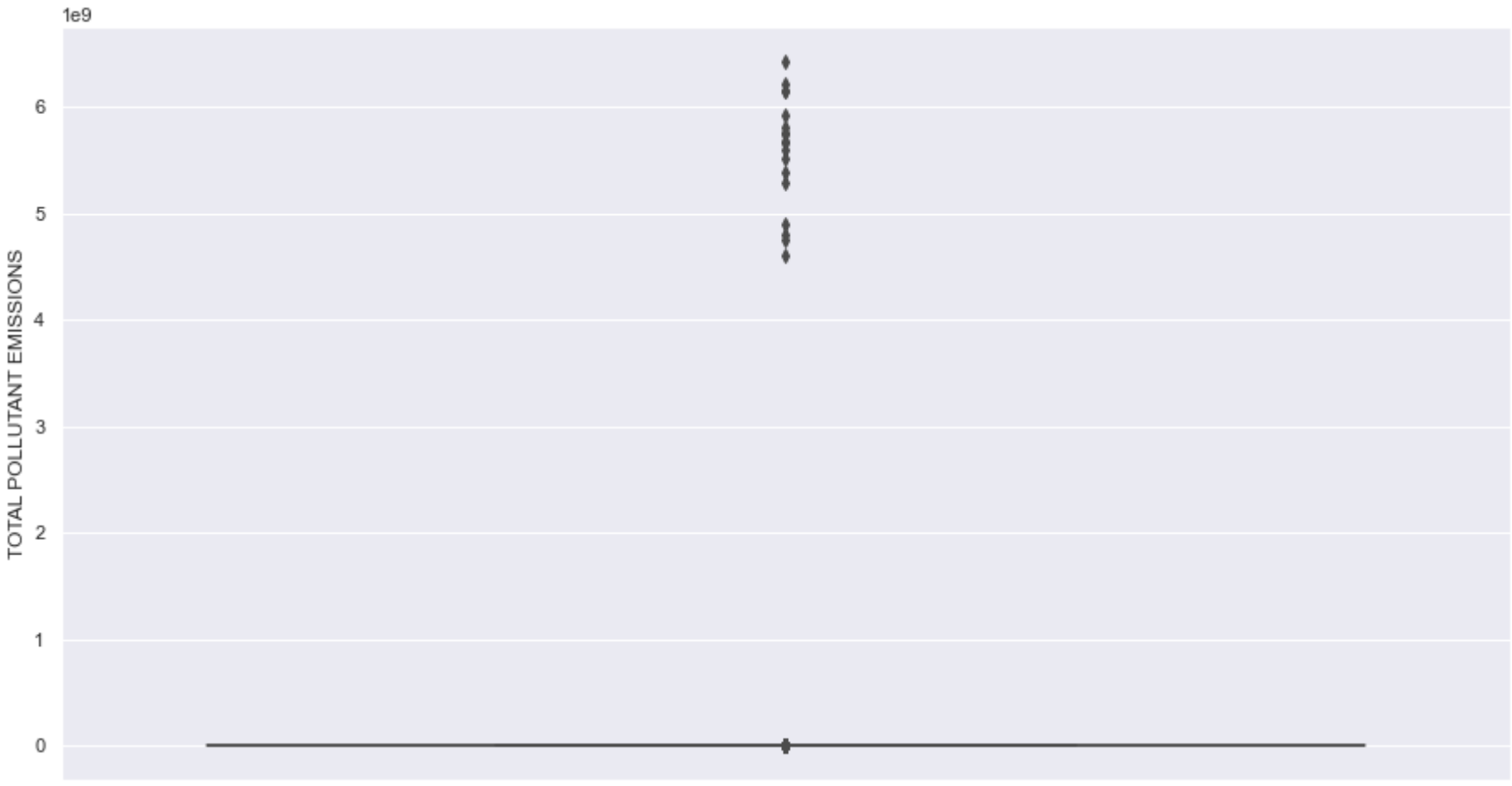
```
health_by_cause_qn4_appendix = death_by_cause_qn4.groupby(["COUNTRY","YEAR"])[["AVERAGE DEATHS PER 100 000"]].mean()
health_by_cause_qn4_appendix = health_by_cause_qn4_appendix.reset_index()
health_by_cause_qn4_appendix = death_by_cause_qn4.merge(owid_dalys_air_pollution_risk_qn4,on=["COUNTRY","YEAR"])
health_by_cause_qn4_appendix = health_by_cause_qn4_appendix.merge(world_oecd_pollutants_data_byyear,on=["COUNTRY","YEAR"])
health_by_cause_qn4_appendix = health_by_cause_qn4_appendix.rename(columns={"VALUE":"TOTAL POLLUTANT EMISSIONS"})
health_by_cause_qn4_appendix.head()
```

Out[126…

	COUNTRY	YEAR	AVERAGE DEATHS PER 100 000	AVERAGE DALYS PER 100 000	TOTAL POLLUTANT EMISSIONS
0	AUS	1990	16.272516	445.483674	2030.843588
1	AUS	1991	15.803590	436.463370	2024.184882
2	AUS	1992	15.798743	433.176282	2076.770529

	COUNTRY	YEAR	AVERAGE DEATHS PER 100 000	AVERAGE DALYS PER 100 000	TOTAL POLLUTANT EMISSIONS
3	AUS	1993	15.261984	416.578658	2146.895118
4	AUS	1994	15.288664	411.408594	2178.151647

In [127…
```
plt.figure(figsize=(15,8))
sns.boxplot(y=health_by_cause_qn4_appendix["TOTAL POLLUTANT EMISSIONS"])
plt.show()
#this shows that there are a few extreme outliers
```



In [128…
```
health_by_cause_qn4_appendix[health_by_cause_qn4_appendix["TOTAL POLLUTANT EMISSIONS"]>1000000000]
#interestingly, hungary has exceptionally high air pollution emissions as compared to all the other countries
```

Out[128…

	COUNTRY	YEAR	AVERAGE DEATHS PER 100 000	AVERAGE DALYS PER 100 000	TOTAL POLLUTANT EMISSIONS
417	HUN	2000	25.192450	1449.957299	5.665823e+09
418	HUN	2001	23.886336	1404.324611	6.143295e+09
419	HUN	2002	23.873878	1412.812703	4.597544e+09
420	HUN	2003	24.214176	1432.958579	5.599101e+09
421	HUN	2004	23.932793	1422.932979	5.521437e+09
422	HUN	2005	24.595117	1450.000000	5.281389e+09
423	HUN	2006	24.060483	1424.782412	4.895140e+09
424	HUN	2007	24.173908	1430.860653	4.801081e+09
425	HUN	2008	23.355606	1389.744051	4.745974e+09
426	HUN	2009	23.381915	1391.245338	5.805509e+09
427	HUN	2010	22.920057	1360.905968	5.737906e+09
428	HUN	2011	22.542163	1345.175794	6.141487e+09
429	HUN	2012	21.730943	1290.266686	6.208326e+09
430	HUN	2013	20.906217	1247.070136	6.431099e+09
431	HUN	2014	20.109380	1210.109658	5.763777e+09
432	HUN	2015	21.185479	1287.239846	5.921199e+09
433	HUN	2016	19.383672	1190.643714	5.684806e+09
434	HUN	2017	18.671147	1167.066188	5.390867e+09

In [129…
```
health_by_cause_qn4_appendix_nooutlier = health_by_cause_qn4_appendix[health_by_cause_qn4_appendix["COUNTRY"]!="HUN"]
health_by_cause_qn4_appendix_nooutlier.head()
```

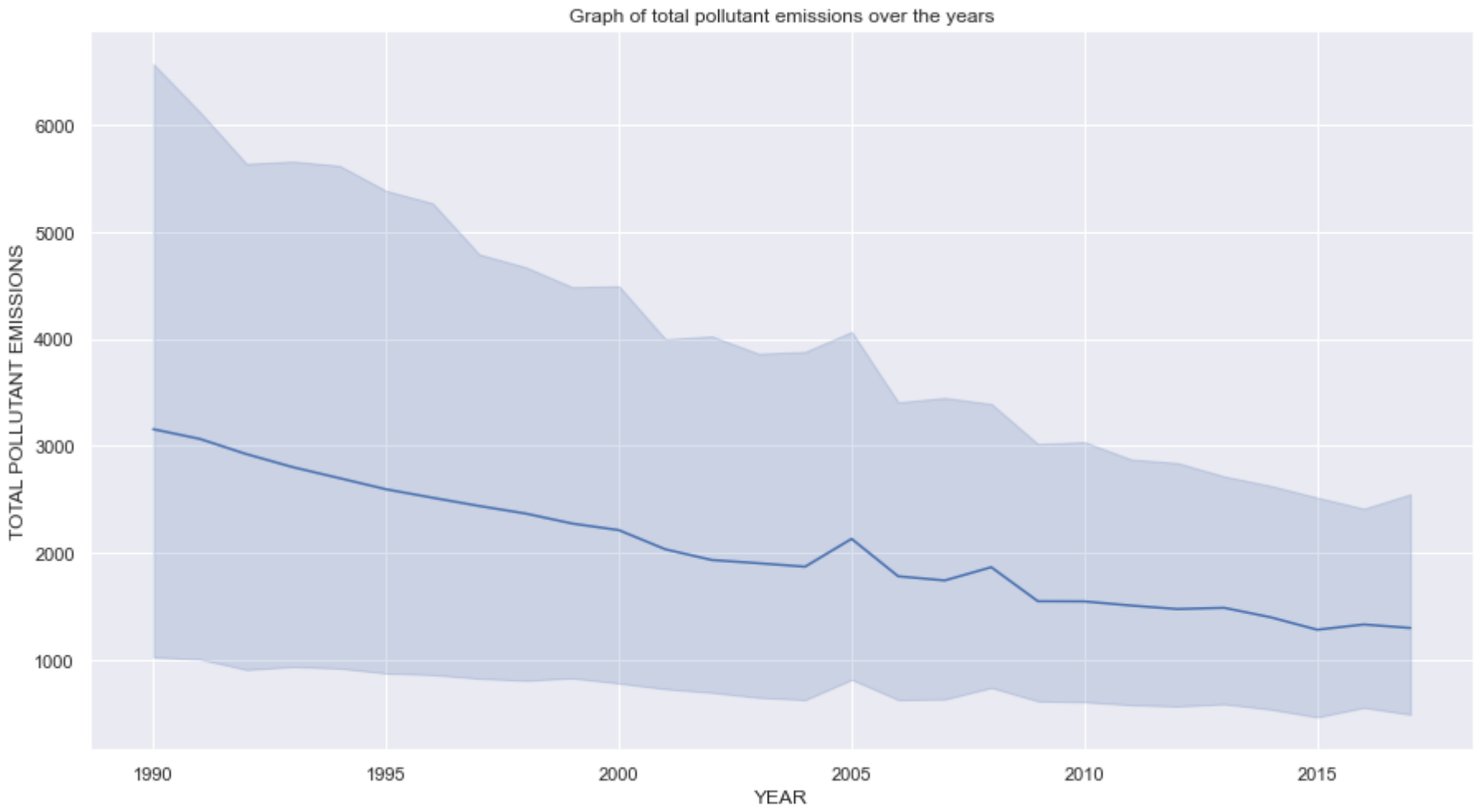
Out[129…

	COUNTRY	YEAR	AVERAGE DEATHS PER 100 000	AVERAGE DALYS PER 100 000	TOTAL POLLUTANT EMISSIONS
0	AUS	1990	16.272516	445.483674	2030.843588
1	AUS	1991	15.803590	436.463370	2024.184882
2	AUS	1992	15.798743	433.176282	2076.770529
3	AUS	1993	15.261984	416.578658	2146.895118

	COUNTRY	YEAR	AVERAGE DEATHS PER 100 000	AVERAGE DALYS PER 100 000	TOTAL POLLUTANT EMISSIONS
4	AUS	1994	15.288664	411.408594	2178.151647

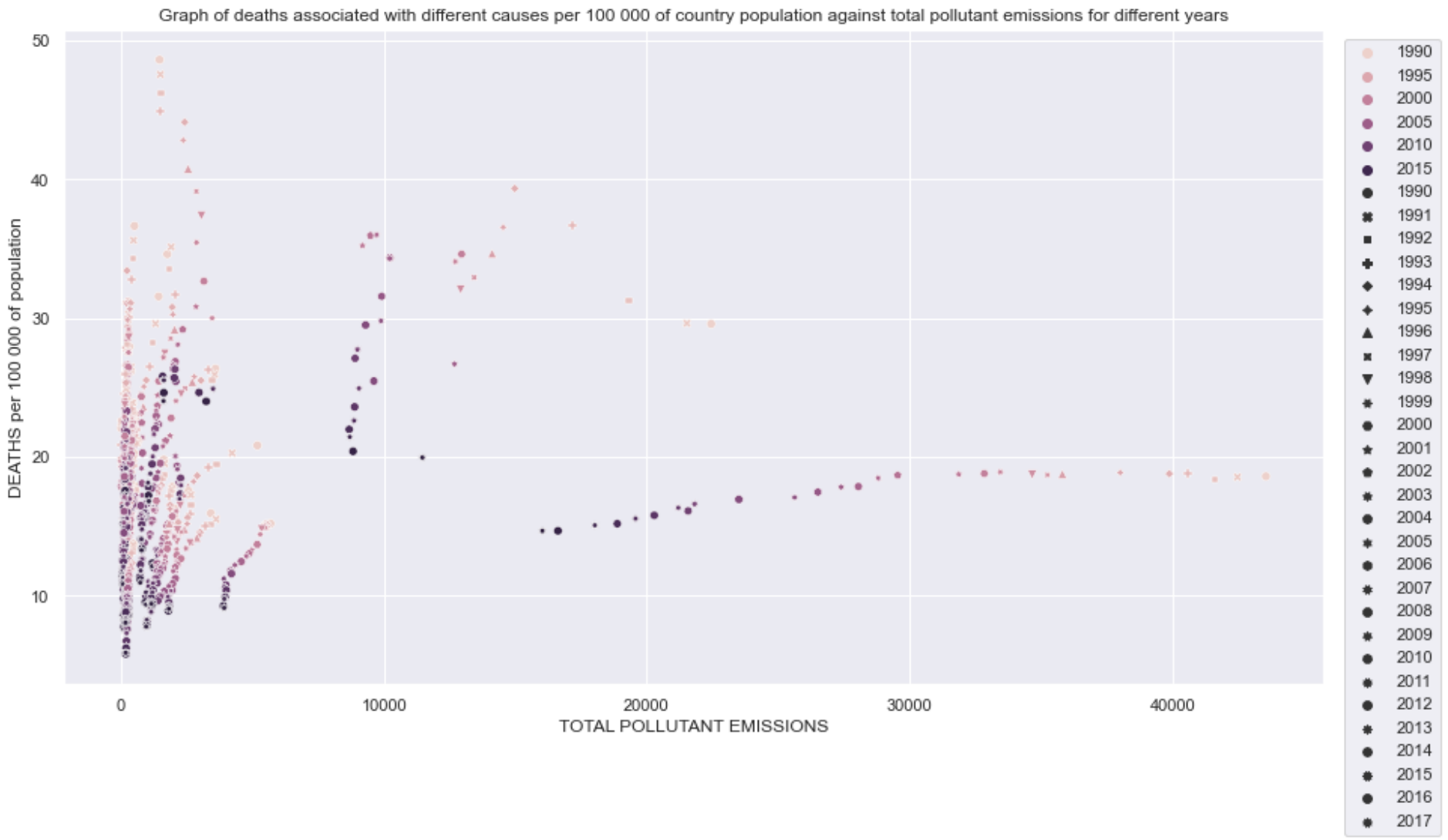
In [130…

```
plt.figure(figsize=(15,8))
sns.lineplot(x="YEAR",y="TOTAL POLLUTANT EMISSIONS",data=health_by_cause_qn4_appendix_nooutlier)
plt.title("Graph of total pollutant emissions over the years")
plt.show()
```



In [131…

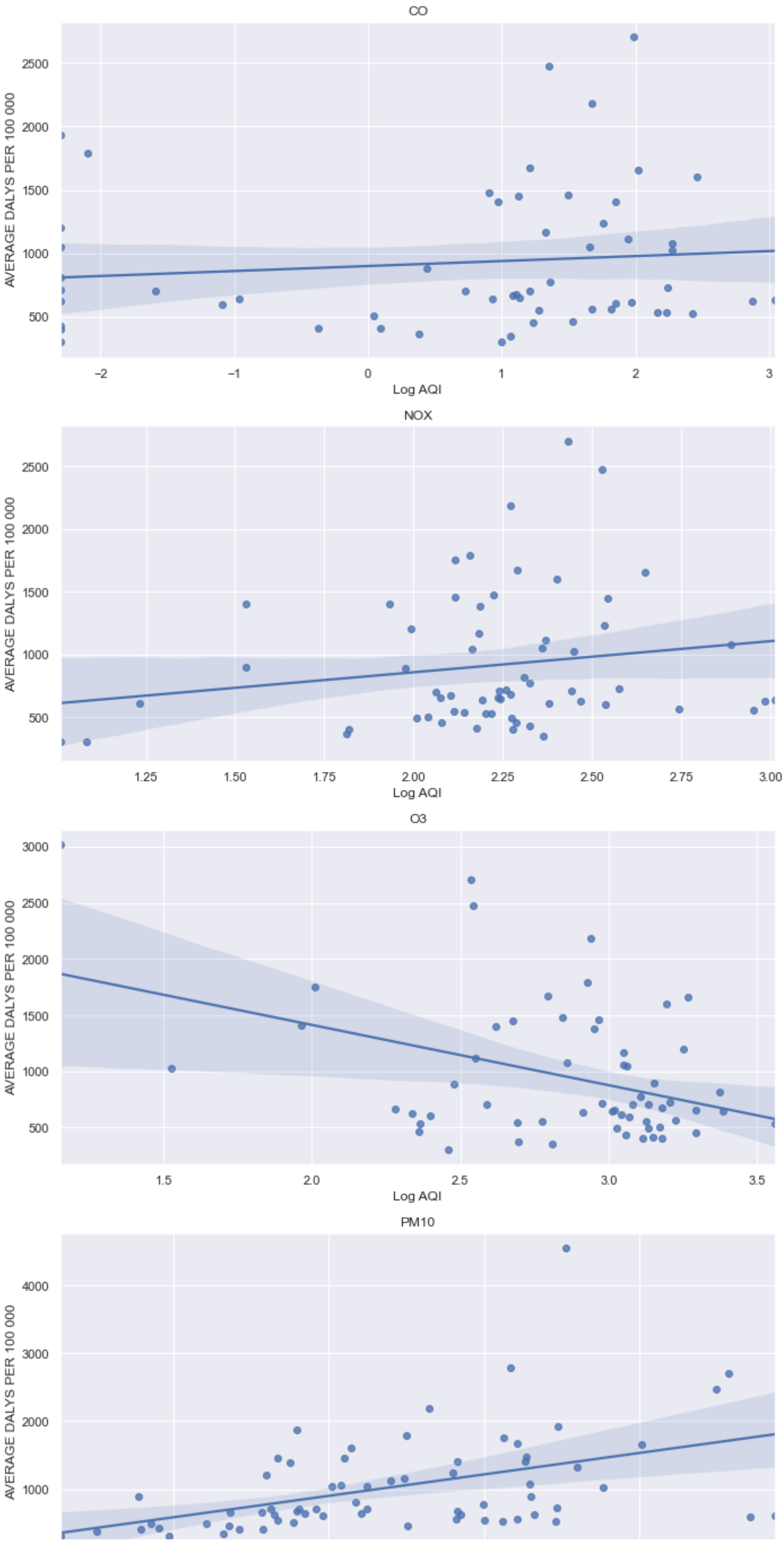
```
plt.figure(figsize=(15,8))
plt.title("Graph of deaths associated with different causes per 100 000 "+
          "of country population against total pollutant emissions for different years")
sns.scatterplot(x="TOTAL POLLUTANT EMISSIONS",y="AVERAGE DEATHS PER 100 000",hue="YEAR",data=health_by_cause_qn4_appendix_nooutlie
plt.ylabel("DEATHS per 100 000 of population")
plt.legend(bbox_to_anchor=(1.01,1))
plt.show()
```

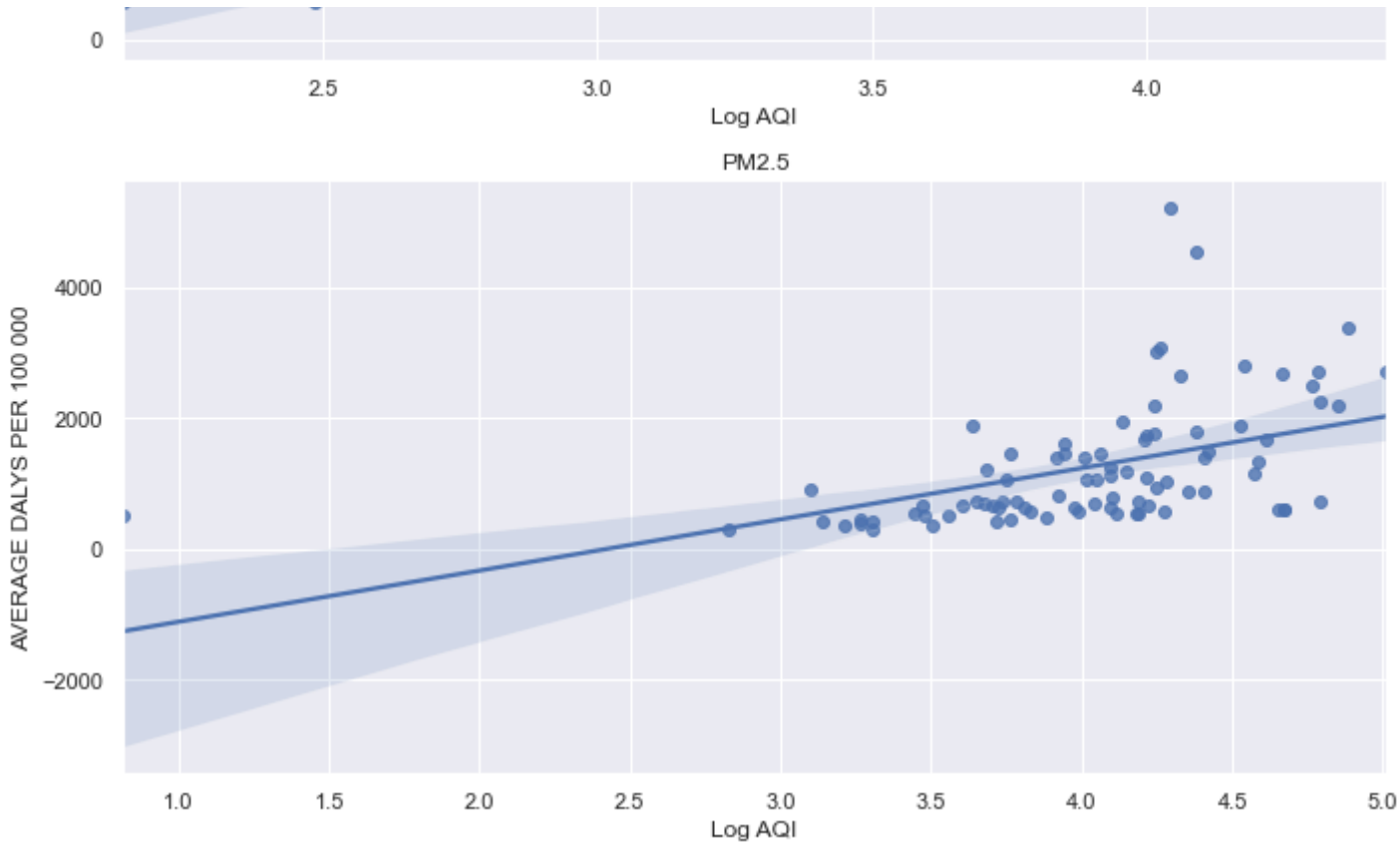


In [132…

```
fig, axes = plt.subplots(nrows=5, ncols=1,  tight_layout=True,figsize=(10,25))

count=0
health_dalys_pollution_qn1_log = health_dalys_pollution_qn1.copy()
health_dalys_pollution_qn1_log["Log AQI"] = np.log(health_dalys_pollution_qn1["AQI"])
pollutant_list = health_dalys_pollution_qn1_log["POLLUTANT"].unique()
```

```
for r in range(5):
    ax = axes[r]
    sns.regplot(x="Log AQI",y="AVERAGE DALYS PER 100 000",
                data=health_dalys_pollution_qn1_log[health_dalys_pollution_qn1_log["POLLUTANT"]==pollutant_list[r]],ax=ax)
    ax.set_title(pollutant_list[r])
    count+=1

plt.show()
```

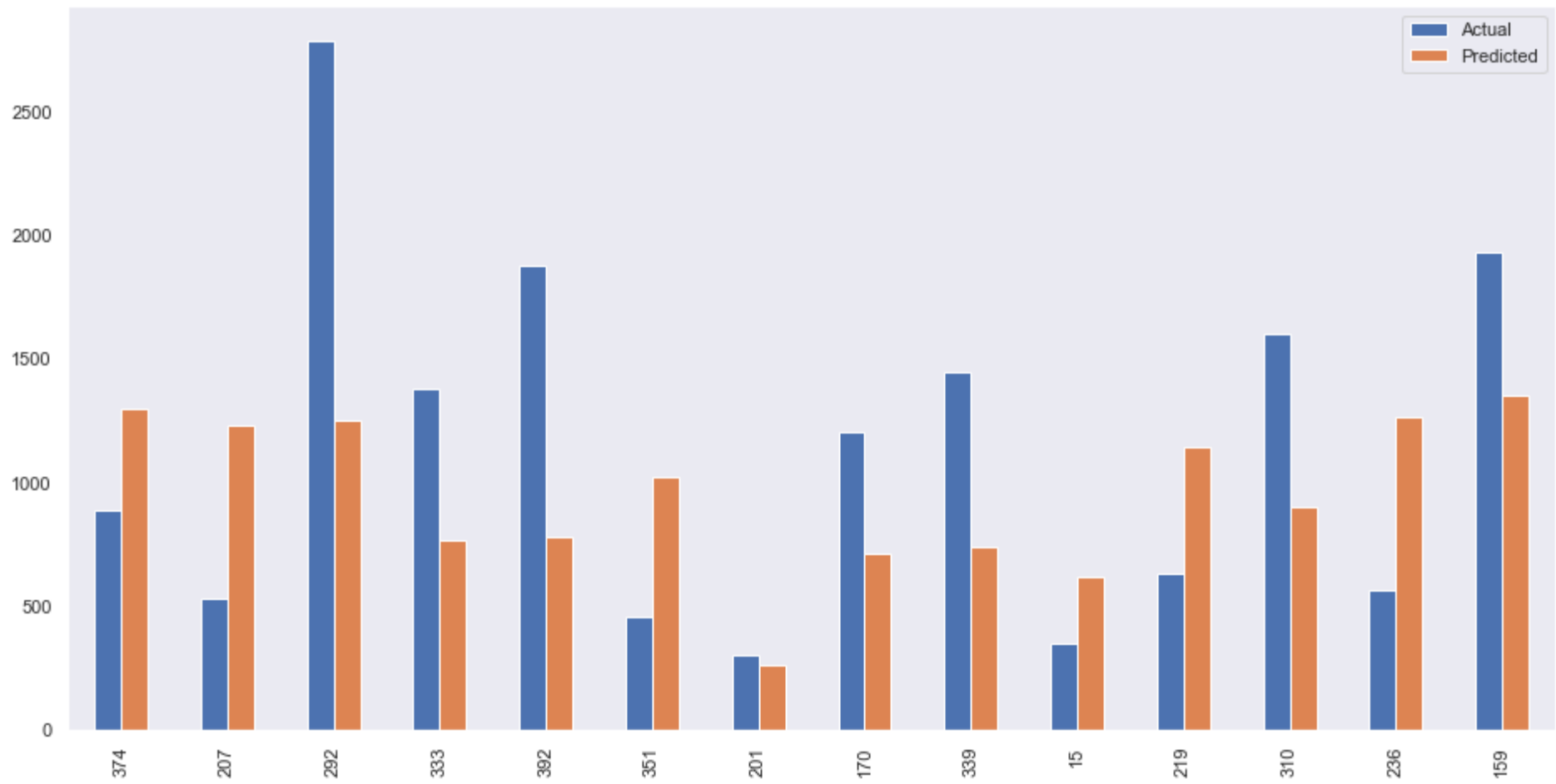




In [133…

```
lm_1 = LinearRegression()
# sns.regplot(x="AQI",y="AVERAGE DEATHS PER 100 000",
#                 data=health_deaths_pollution_qn1[health_deaths_pollution_qn1["POLLUTANT"]==pollutant_list[r]],ax=ax)
# ax.set_title(pollutant_list[r])
health_dalys_pollution_qn1_log_lim_pm10 = health_dalys_pollution_qn1_log[health_dalys_pollution_qn1_log["POLLUTANT"]=="PM10"]
health_dalys_pollution_qn1_log_lim_pm10 = health_dalys_pollution_qn1_log_lim_pm10.dropna(axis=0,how='any')
x_train_1, x_test_1, y_train_1, y_test_1 = train_test_split(health_dalys_pollution_qn1_log_lim_pm10[['Log AQI']],
                                                            health_dalys_pollution_qn1_log_lim_pm10['AVERAGE DALYS PER 100 000'],
                                                            test_size=0.2, random_state=1)
lm_1.fit(x_train_1, y_train_1)
print(lm_1.intercept_, lm_1.coef_)
yhat_1 = lm_1.predict(x_test_1)
df_act_pred_1 = pd.DataFrame({'Actual': y_test_1, 'Predicted': yhat_1})
df_act_pred_1.plot(kind='bar',figsize=(16,8))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.show()
```

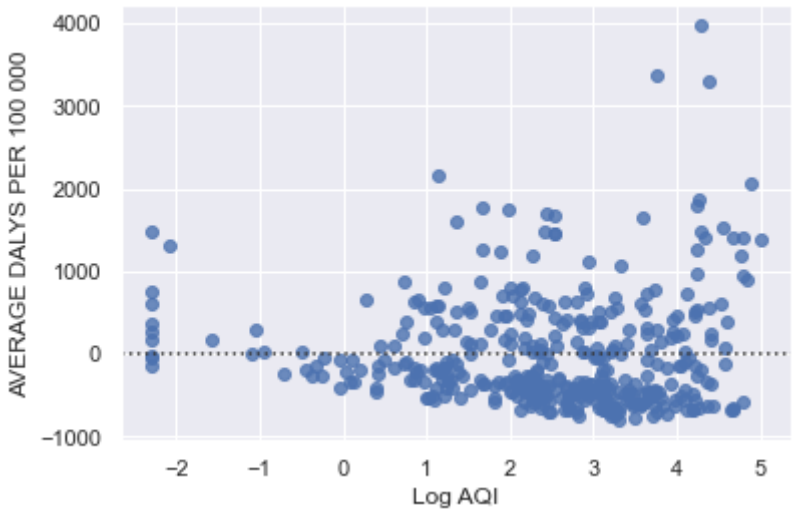
```
-1199.3548325633658 [683.97443752]
```



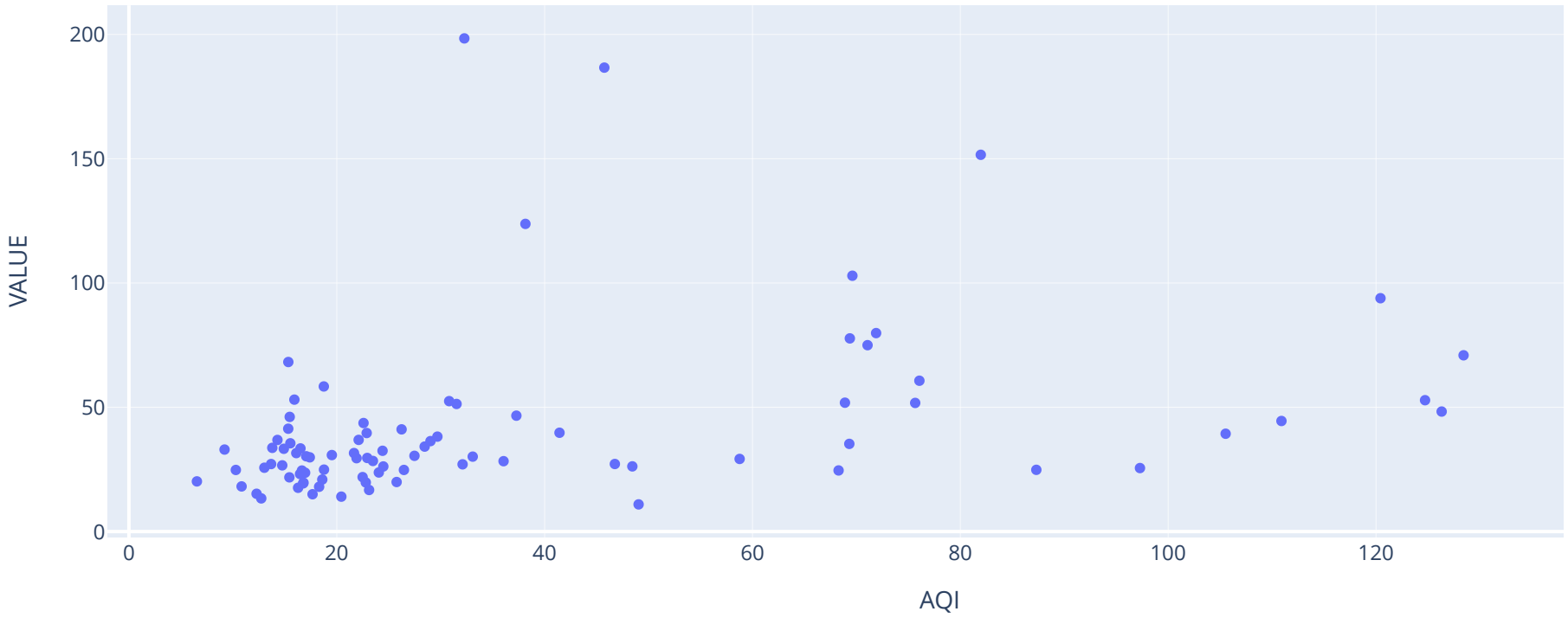
In [134…

```
sns.residplot(x='Log AQI', y= 'AVERAGE DALYS PER 100 000',data =  health_dalys_pollution_qn1_log)
lm_1.score(x_train_1, y_train_1)
#sns.kdeplot(yhat_2,  color='b', label='Fitted Value')
#sns.kdeplot(yhat_2,  color='r', label='Actual Value')
```

Out[134… 0.2531281035509406



In [135…
```
px.scatter(data_frame=health_by_cause_pollution_dropnoaqi_pivot,x="AQI",y="VALUE",animation_frame="CAUSE OF HEALTH",
           hover_name="COUNTRY")
```



As this animation graph uses the same y axis, the data cannot be observed clearly for certain causes of health, therefore an unsuitable representation.

In [136…
```
singapore_collated_data.head()
```

Out[136…

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
SOX (24H,MICGRCUBM)	84.0	80.0	93.0	104.0	80.0	98.0	75.0	83.0	75.0	61.0	59.0	65.0	57.0	30.0
NOX (YR,MICGRCUBM)	22.0	22.0	22.0	23.0	25.0	25.0	25.0	24.0	22.0	26.0	25.0	26.0	23.0	20.0
NOX (1H,MICRCUBM)	177.0	126.0	147.0	153.0	189.0	154.0	132.0	121.0	99.0	123.0	158.0	147.0	156.0	118.0
PM10 (YR,MICGRCUBM)	27.0	25.0	29.0	26.0	27.0	29.0	31.0	30.0	37.0	26.0	25.0	29.0	30.0	25.0
PM10 (24H,MICGRCUBM)	53.0	49.0	59.0	76.0	55.0	57.0	215.0	75.0	186.0	61.0	57.0	59.0	90.0	43.0

Singapore collated data is not used in the final project after further consideration of the research questions.

Originally, it was assumed that the values provided by WAQI website were the breakpoint values, therefore a revised AQI Value was converted using those values as breakpoints. However when the same EDA was performed on the revised AQI, it was noted that the graphs gave very contrasting values and results. In addition, the website mentioned "All air pollutant species are converted to the US EPA standard (i.e. no raw concentrations)", which was unclear whether it was the AQI value or breakpoints values from, unable to clearly come to a conclusion whether the value was the breakpoint (concentration) or the aqi value for the respective pollutant. Therefore, a final conclusion and assumption was made that the value provided is already the AQI value.

To note the original type of "aqi" found in Specie column of raw data of waqi datasets, it was assumed to be the mean aqi of all the different "Specie", therefore the assumption made that the values for the respective pollutants was the AQI value for each pollutant type.

In [137…
```
def calculate_aqi_waqi(row):
    pollutant_df = aqi_breakpoints_data_processed.loc[row.POLLUTANT]
    pollutant_value = row["VALUE"]
    pollutant_df_row = pollutant_df[(pollutant_df["Low Breakpoint"]<=pollutant_value) &
                                    (pollutant_df["High Breakpoint"]>=pollutant_value)]
    aqi_value = (pollutant_df_row["High AQI"]-pollutant_df_row["Low AQI"])/(
        pollutant_df_row["High Breakpoint"]-pollutant_df_row["Low Breakpoint"])*(
```

```
        pollutant_value-pollutant_df_row["Low Breakpoint"])+pollutant_df_row["Low AQI"]
    #corner case wehere pollutant is between breakpoints and does not fall under any category, just round off to threshold
    if pollutant_df_row.empty:
        pollutant_value = round(pollutant_value,0)
        pollutant_df_row = pollutant_df[(pollutant_df["Low Breakpoint"]<=pollutant_value) &
                                        (pollutant_df["High Breakpoint"]>=pollutant_value)]
        aqi_value = (pollutant_df_row["High AQI"]-pollutant_df_row["Low AQI"])/(
            pollutant_df_row["High Breakpoint"]-pollutant_df_row["Low Breakpoint"])*(
            pollutant_value-pollutant_df_row["Low Breakpoint"])+pollutant_df_row["Low AQI"]
    row["AQI_REVISED"] = float(aqi_value)
    return row
```

In [138…

```
waqi_data_total = waqi_data_total.rename(columns={"AQI":"VALUE"})
waqi_data_total = waqi_data_total.groupby(["COUNTRY","YEAR","POLLUTANT"])[["VALUE"]].mean().reset_index()
waqi_data_total = waqi_data_total.apply(calculate_aqi_waqi,axis='columns')
waqi_data_total.drop(columns=["VALUE"],inplace=True)
waqi_data_total = waqi_data_total.set_index("COUNTRY")
waqi_data_total.head()
```

Out[138…

COUNTRY	YEAR	POLLUTANT	AQI_REVISED
ARE	2015	CO	36.846591
ARE	2015	NOX	30.835080
ARE	2015	O3	501.149253
ARE	2015	PM10	46.712963
ARE	2015	PM2.5	183.122366