Pollution_data_analysis

May 7, 2023

[100]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go

# Importing required libraries for data visualization and analysis

import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# Importing required Plotly libraries for creating interactive plots
```

1 1. Data Collecting

1.1 1.1 Importing the dataset from a CSV file

[101]:
```
# Load the CSV file into a pandas dataframe
df = pd.read_csv(r"C:\Users\faizb\OneDrive\Bureau\dataset\global air pollution␣
↪dataset.csv")

# Get the shape of the dataframe (number of rows, number of columns)
df.shape
```

[101]:
```
(23463, 12)
```

[3]:
```
#Display the first 5 rows of the dataframe


df.head(5)
```

[3]:

	Country	City	AQI Value	AQI Category	CO AQI Value	\
0	Russian Federation	Praskoveya	51	Moderate	1	
1	Brazil	Presidente Dutra	41	Good	1	
2	Italy	Priolo Gargallo	66	Moderate	1	
3	Poland	Przasnysz	34	Good	1	

```
4              France          Punaauia         22         Good              0

  CO AQI Category  Ozone AQI Value Ozone AQI Category  NO2 AQI Value  \
0            Good               36               Good              0
1            Good                5               Good              1
2            Good               39               Good              2
3            Good               34               Good              0
4            Good               22               Good              0

  NO2 AQI Category  PM2.5 AQI Value PM2.5 AQI Category
0             Good               51           Moderate
1             Good               41               Good
2             Good               66           Moderate
3             Good               20               Good
4             Good                6               Good
```

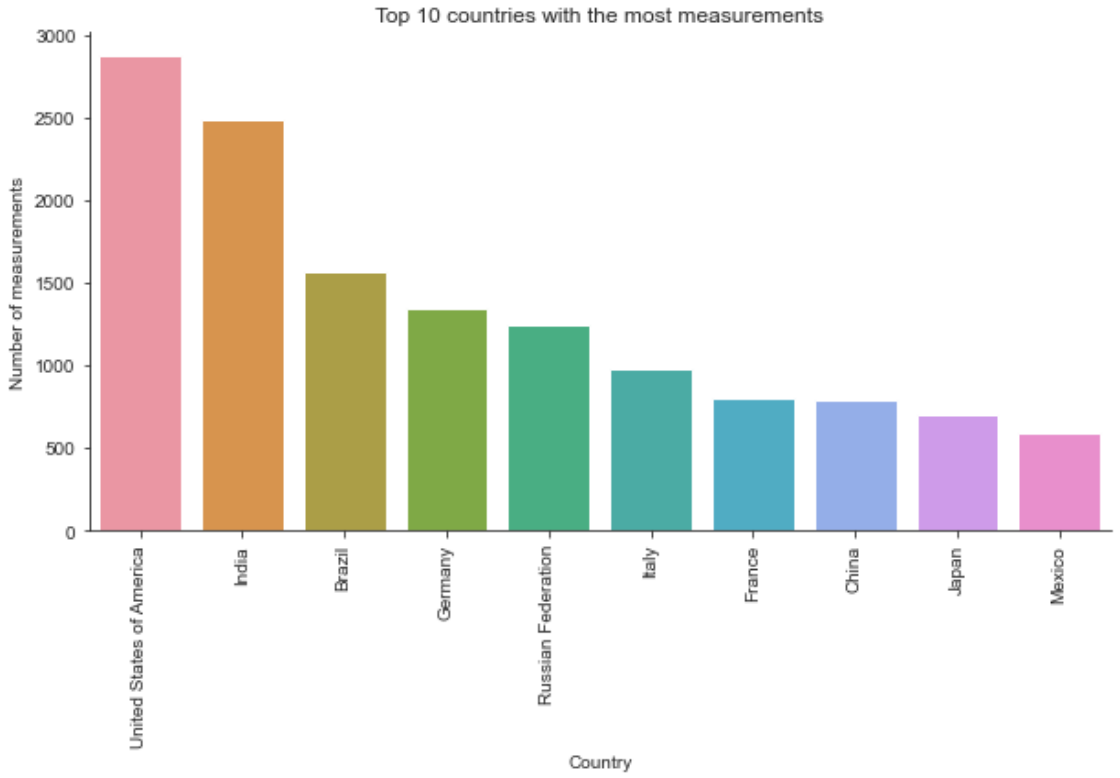
2 2. Data Analysis

[4]:
```
#Checking of Missing or Null value in dataset:

df.isnull().sum()
```

[4]:
```
Country               427
City                    1
AQI Value               0
AQI Category            0
CO AQI Value            0
CO AQI Category         0
Ozone AQI Value         0
Ozone AQI Category      0
NO2 AQI Value           0
NO2 AQI Category        0
PM2.5 AQI Value         0
PM2.5 AQI Category      0
dtype: int64
```

[102]:
```
# Get a count of measurements by country
country_counts = df['Country'].value_counts()

# Get the top 10 countries with the highest number of measurements
top_countries = country_counts.head(10)

# Set the figure size and plot a bar chart of the top 10 countries
plt.figure(figsize=(10,5))
sns.barplot(x=top_countries.index, y=top_countries.values)
```

```
# Rotate the x-axis labels by 90 degrees for better readability
plt.xticks(rotation=90)

# Set the x and y axis labels and the plot title
plt.xlabel('Country')
plt.ylabel('Number of measurements')
plt.title('Top 10 countries with the most measurements')

# Remove the top and right spines of the plot
sns.despine()

# Show the plot
plt.show()
```



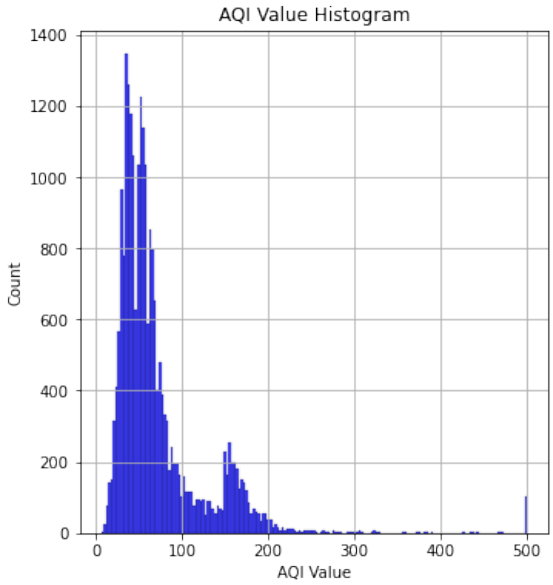
[103]:
```
#Create a list of all column names in the dataframe

metrics = list(df.columns)

# Calculate the descriptive statistics of each metric in the dataframe and␣
 ↪transpose the result

df[metrics].describe().T
```

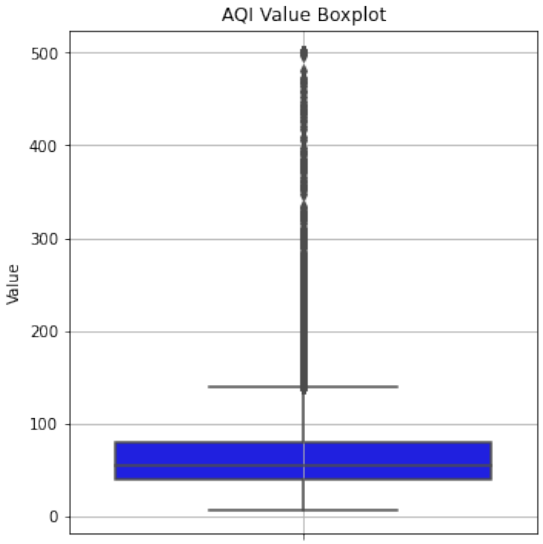
[103]:

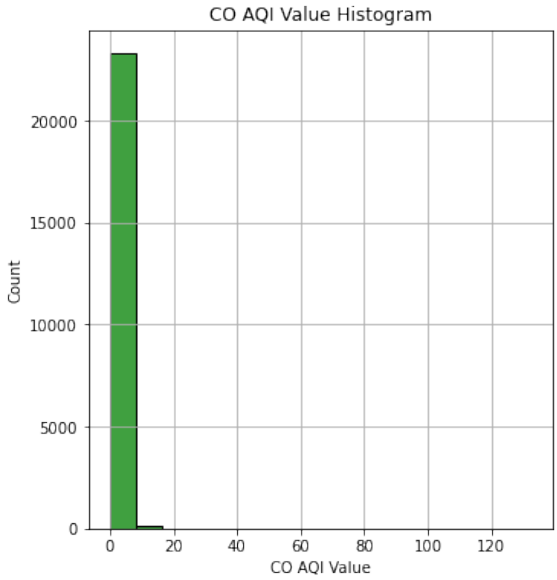
	count	mean	std	min	25%	50%	75%	max
AQI Value	23463.0	72.010868	56.055220	6.0	39.0	55.0	79.0	500.0
CO AQI Value	23463.0	1.368367	1.832064	0.0	1.0	1.0	1.0	133.0
Ozone AQI Value	23463.0	35.193709	28.098723	0.0	21.0	31.0	40.0	235.0
NO2 AQI Value	23463.0	3.063334	5.254108	0.0	0.0	1.0	4.0	91.0
PM2.5 AQI Value	23463.0	68.519755	54.796443	0.0	35.0	54.0	79.0	500.0

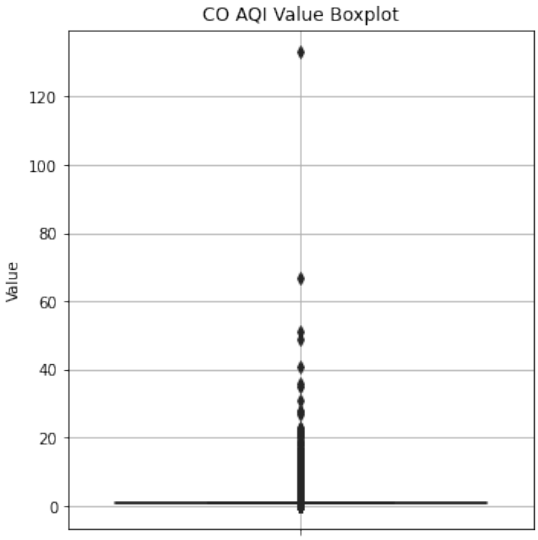
[9]:

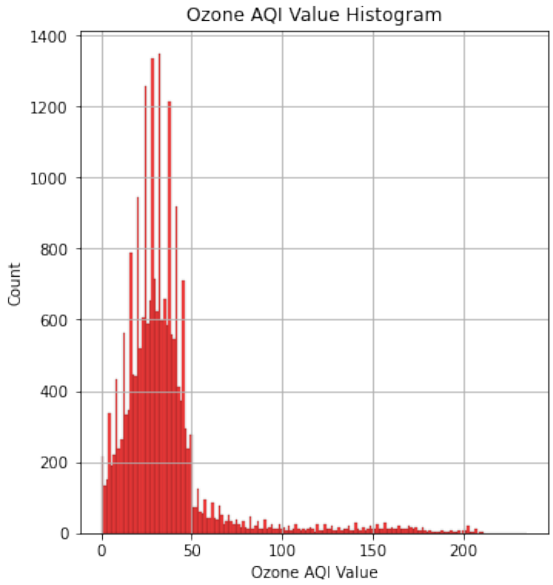
```
# Define a list of colors to use for the plots
colors = ['blue', 'green', 'red', 'purple', 'orange', 'yellow', 'pink']

# Loop over the columns of the DataFrame
for i, col in enumerate(df.select_dtypes(['float64', 'int64'])):

    # Create a figure with two subplots
    fig, axs = plt.subplots(ncols=2, figsize=(12, 6))

    # Create a histogram of the column in the first subplot
    sns.histplot(df[col], ax=axs[0], kde=False, color=colors[i % len(colors)])

    # Create a box plot of the column in the second subplot
    sns.boxplot(y=df[col], ax=axs[1], color=colors[i % len(colors)])

    # Set the titles of the subplots
    axs[0].set_title(f"{col} Histogram")
    axs[0].grid(True)
    axs[1].set_title(f"{col} Boxplot")
    axs[1].grid(True)
    # Set the y-axis label for the second subplot
    axs[1].set_ylabel('Value')

    # Show the figure
    plt.show()
```

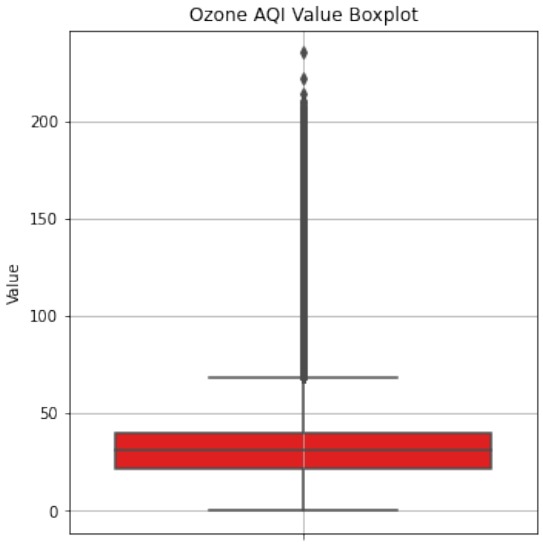

AQI Value Histogram
Count
AQI Value
0
200
400
600
800
1000
1200
1400
0
100
200
300
400
500

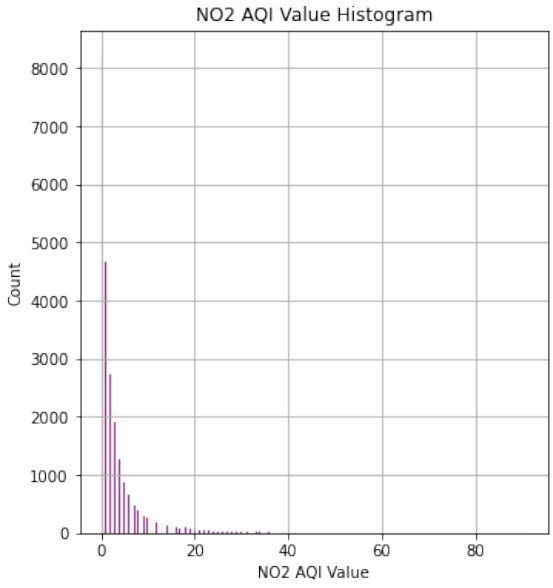

AQI Value Boxplot
Value
0
100
200
300
400
500

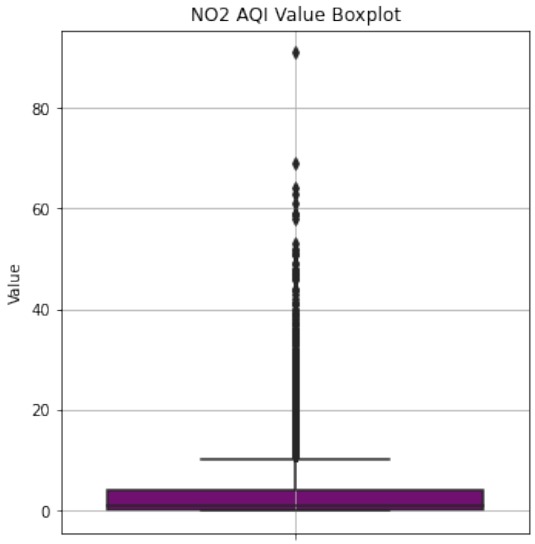

CO AQI Value Histogram
Count
CO AQI Value
0
5000
10000
15000
20000
0
20
40
60
80
100
120

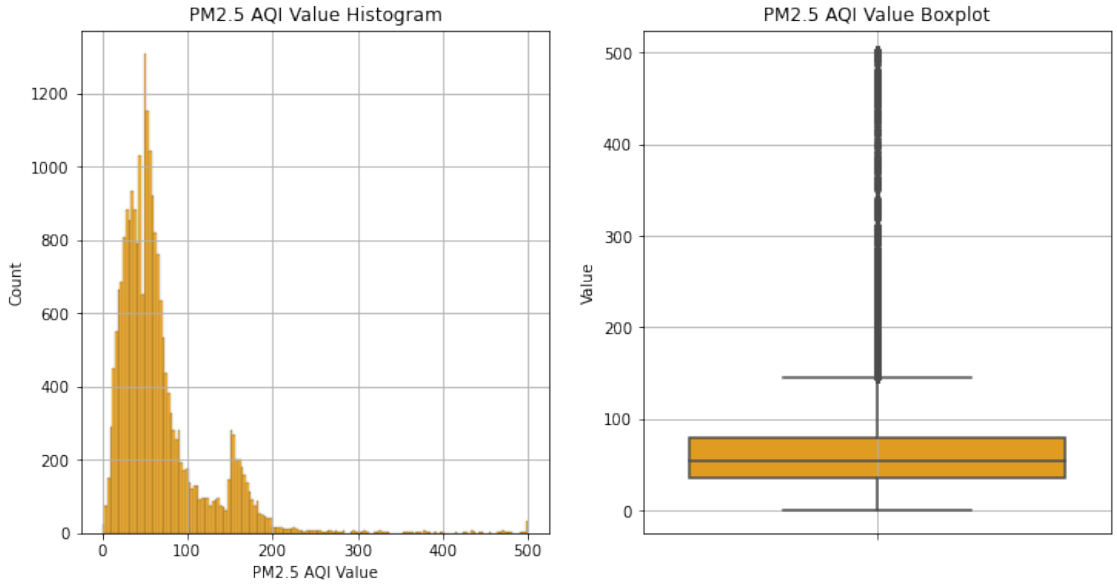

CO AQI Value Boxplot
Value
0
20
40
60
80
100
120


Ozone AQI Value Histogram
Count
Ozone AQI Value
0
200
400
600
800
1000
1200
1400
0
50
100
150
200


Ozone AQI Value Boxplot
Value
0
50
100
150
200


NO2 AQI Value Histogram
Count
NO2 AQI Value
0
1000
2000
3000
4000
5000
6000
7000
8000
0
20
40
60
80


NO2 AQI Value Boxplot
Value
0
20
40
60
80



[105]:
```
# Group the data by country and aggregate the values for different AQI␣
 ↪parameters
countries = (
    df.groupby('Country')
    .agg(
        avg_aqi_value=('AQI Value', 'median'),
        max_aqi_value=('AQI Value','max'),
        avg_co_aqi_value=('CO AQI Value', 'median'),
        max_co_aqi_value=('CO AQI Value', 'max'),
        avg_ozone_aqi_value=('Ozone AQI Value', 'median'),
        max_ozone_aqi_value=('Ozone AQI Value', 'max'),
        avg_no2_aqi_value=('NO2 AQI Value', 'median'),
        max_no2_aqi_value=('NO2 AQI Value', 'max'),
        avg_pm2_5_aqi_value=('PM2.5 AQI Value', 'median'),
        max_pm2_5_aqi_value=('PM2.5 AQI Value', 'max')
    )
)

# Show the aggregated data for each country
countries
```

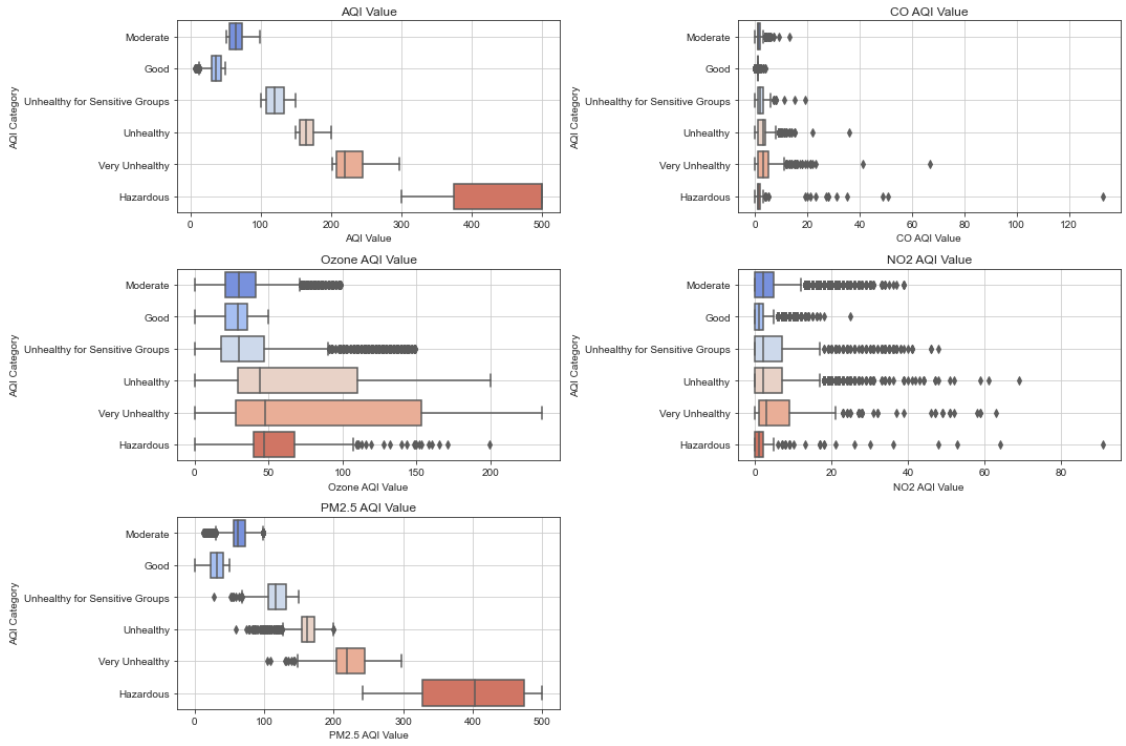
[105]:

Country	avg_aqi_value	max_aqi_value	\
Afghanistan	87.0	198	
Albania	66.0	115	
Algeria	82.5	164	
Andorra	29.0	32	

```
Angola                                    58.0            285
…                                         …               …
Venezuela (Bolivarian Republic of)        60.0            165
Viet Nam                                  69.0            194
Yemen                                    151.0            179
Zambia                                    36.5            125
Zimbabwe                                  41.0             93

                                  avg_co_aqi_value  max_co_aqi_value  \
Country
Afghanistan                                  1.0                 2
Albania                                      1.0                 1
Algeria                                      1.0                10
Andorra                                      1.0                 1
Angola                                       1.0                23
…                                            …                  …
Venezuela (Bolivarian Republic of)           1.0                 4
Viet Nam                                     2.0                10
Yemen                                        1.0                 2
Zambia                                       1.0                 2
Zimbabwe                                     1.0                 5

                                  avg_ozone_aqi_value  max_ozone_aqi_value  \
Country
Afghanistan                                      41.0                   64
Albania                                          42.0                   49
Algeria                                          40.0                  117
Andorra                                          29.0                   32
Angola                                           21.0                   49
…                                                …                      …
Venezuela (Bolivarian Republic of)               16.0                   52
Viet Nam                                         32.0                  194
Yemen                                            44.0                   93
Zambia                                           20.0                   27
Zimbabwe                                         17.0                   26

                                  avg_no2_aqi_value  max_no2_aqi_value  \
Country
Afghanistan                                     0.0                  1
Albania                                         1.0                  2
Algeria                                         1.0                 69
Andorra                                         0.0                  0
Angola                                          0.0                 14
…                                               …                   …
Venezuela (Bolivarian Republic of)              3.0                 18
Viet Nam                                        1.0                 22
Yemen                                           1.0                  2
```

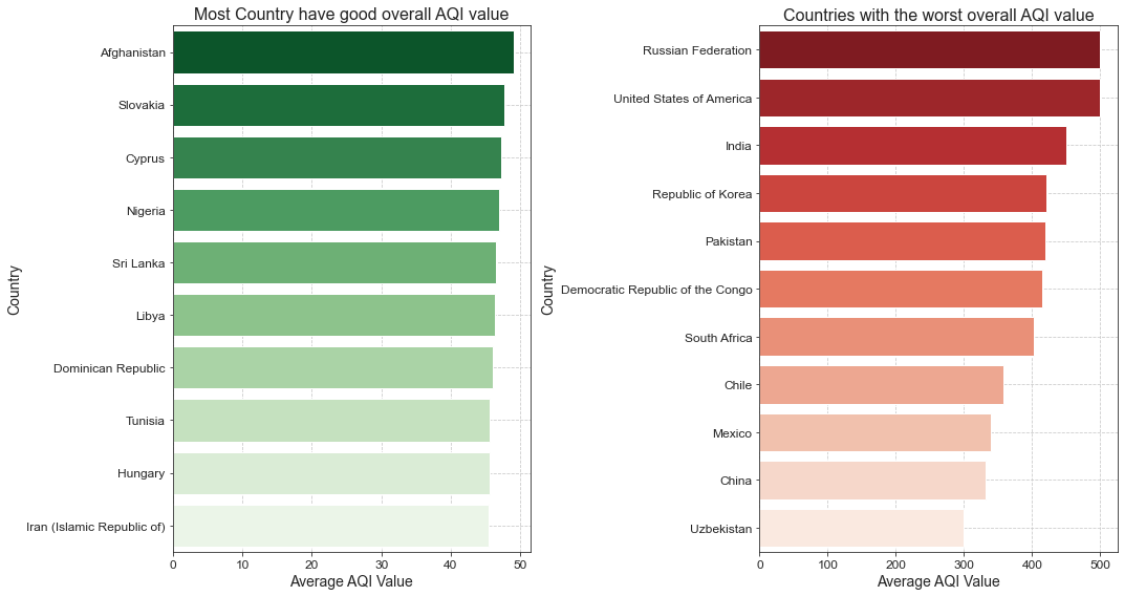
```
Zambia                                          0.0                   2
Zimbabwe                                        0.0                   3

                                  avg_pm2_5_aqi_value  max_pm2_5_aqi_value
Country
Afghanistan                                      87.0                  198
Albania                                          66.0                  115
Algeria                                          72.0                  164
Andorra                                          22.0                   24
Angola                                           58.0                  285
…                                                …                     …
Venezuela (Bolivarian Republic of)               60.0                  165
Viet Nam                                         69.0                  179
Yemen                                           151.0                  179
Zambia                                           36.5                  125
Zimbabwe                                         41.0                   93

[175 rows x 10 columns]
```

[104]:
```
# columns to plot
columns = ['AQI Value', 'CO AQI Value', 'Ozone AQI Value', 'NO2 AQI Value',␣
 ↪'PM2.5 AQI Value']

# create a 3x2 subplot grid with given size
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(15, 10))
axes = axes.ravel()

# set seaborn style and palette
sns.set_style('ticks')
sns.set_palette('coolwarm')

# loop through each column and plot boxplot for AQI category
for i, col in enumerate(columns):
    sns.boxplot(data=df, x=col, y='AQI Category', ax=axes[i])
    axes[i].set_title(col)
    axes[i].grid(True)

# delete the last subplot
fig.delaxes(axes[-1])

# adjust subplot layout
plt.tight_layout()

# display the plot
plt.show()
```



[12]:
```
# Define the data sets
data_sets = [
    {'title': 'Most Country have good overall AQI value', 'color': 'Greens_r',
     'data': df[df['AQI Category'] == 'Good'].groupby('Country',␣
 ↪as_index=False)['AQI Value'].mean().sort_values(by='AQI␣
 ↪Value',ascending=False).head(10)},
    {'title': 'Countries with the worst overall AQI value', 'color': 'Reds_r',
     'data': df[df['AQI Category'] == 'Hazardous'].groupby('Country',␣
 ↪as_index=False)['AQI Value'].mean().sort_values(by='AQI Value',␣
 ↪ascending=False)}
]

# Create the subplots
fig, axs = plt.subplots(ncols=len(data_sets), figsize=(15, 8))

# Loop through the data sets and create the corresponding graph for each set
for i, data in enumerate(data_sets):
    sns.barplot(x='AQI Value', y='Country', data=data['data'],␣
 ↪palette=data['color'], ax=axs[i])
    axs[i].set_title(data['title'], fontsize=16)
    axs[i].set_xlabel('Average AQI Value', fontsize=14)
    axs[i].set_ylabel('Country', fontsize=14)
    axs[i].tick_params(labelsize=12)
```

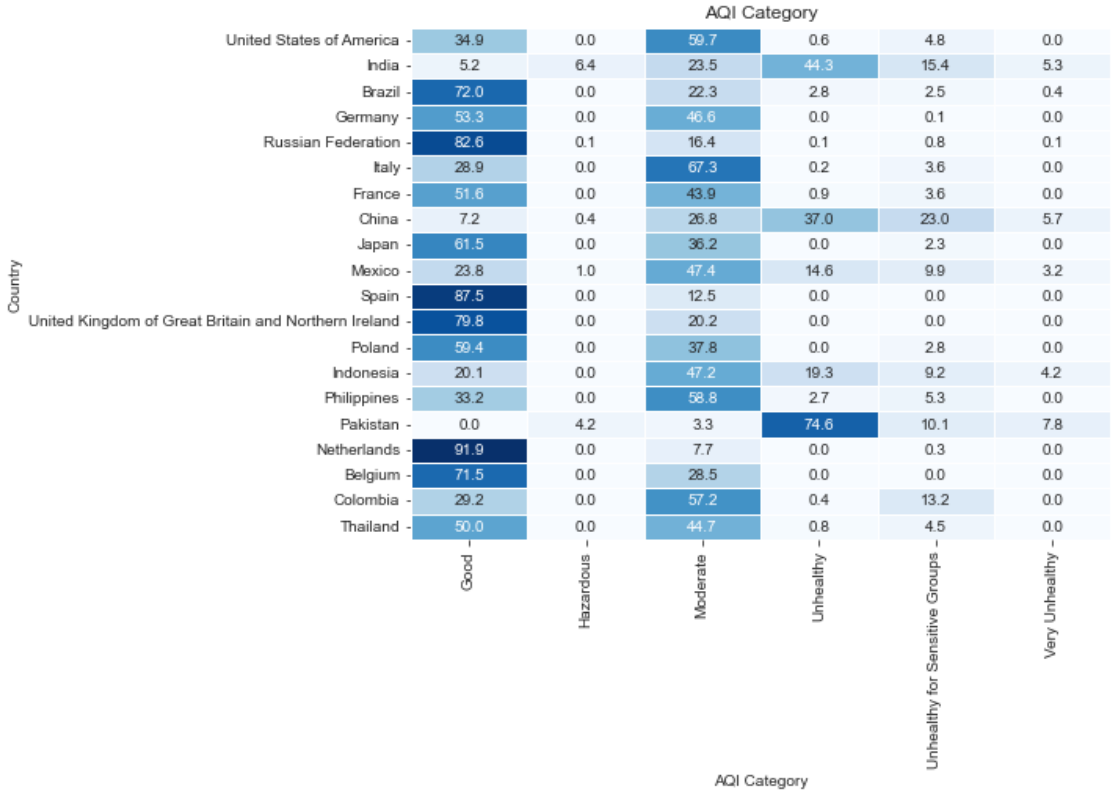
```
    axs[i].grid(linestyle='--')

# Adjust the layout and display the plot
plt.tight_layout()
plt.show()
```

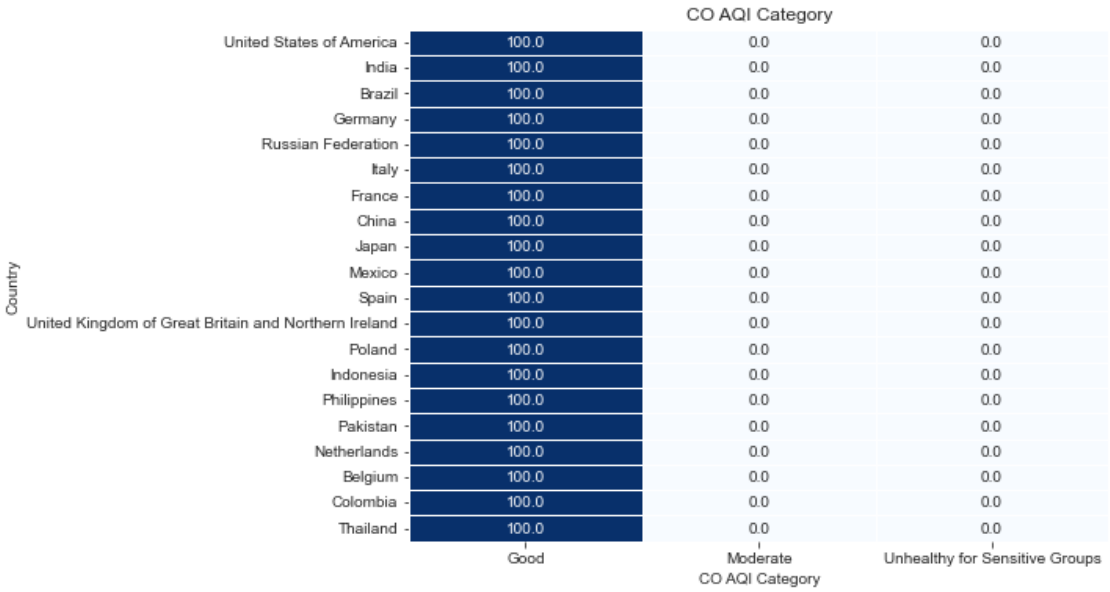


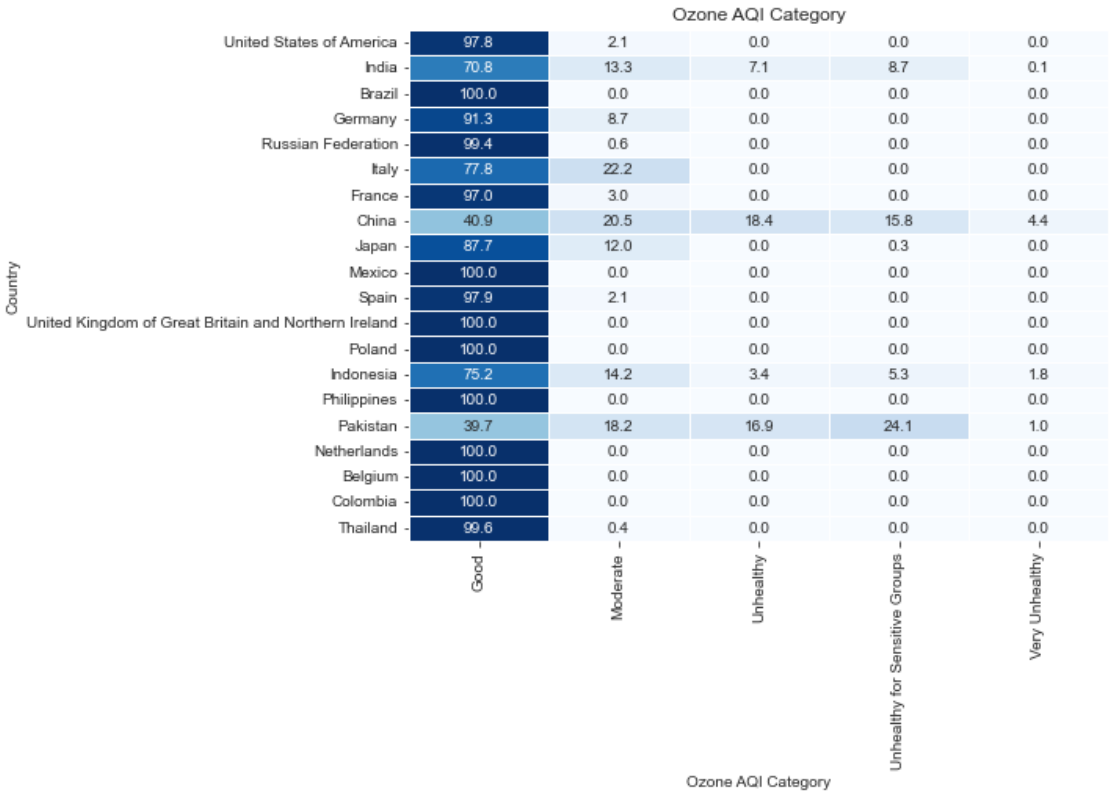
India and China have the highest percentage of locations with unhealthy air quality, while Indonesia and Mexico have more 'good' areas. Spain has the best air quality with no locations marked as risky. No country has locations marked as hazardous for carbon monoxide, and the USA has the fewest 'moderate' areas. China has the worst ground-level ozone conditions, but over 60% of areas are within normal limits. India has the most 'good' locations for this category. Brazil, Mexico, Philippines, Poland, and the UK have all locations marked as 'Good' for ground-level ozone. Indonesia and China have relatively worse nitrogen dioxide conditions, with the USA having the fewest 'moderate' areas. India, China, Indonesia, and Mexico have the worst atmospheric particulate matter conditions, with less than one-third of Indian locations being 'Good' to 'Moderate'.

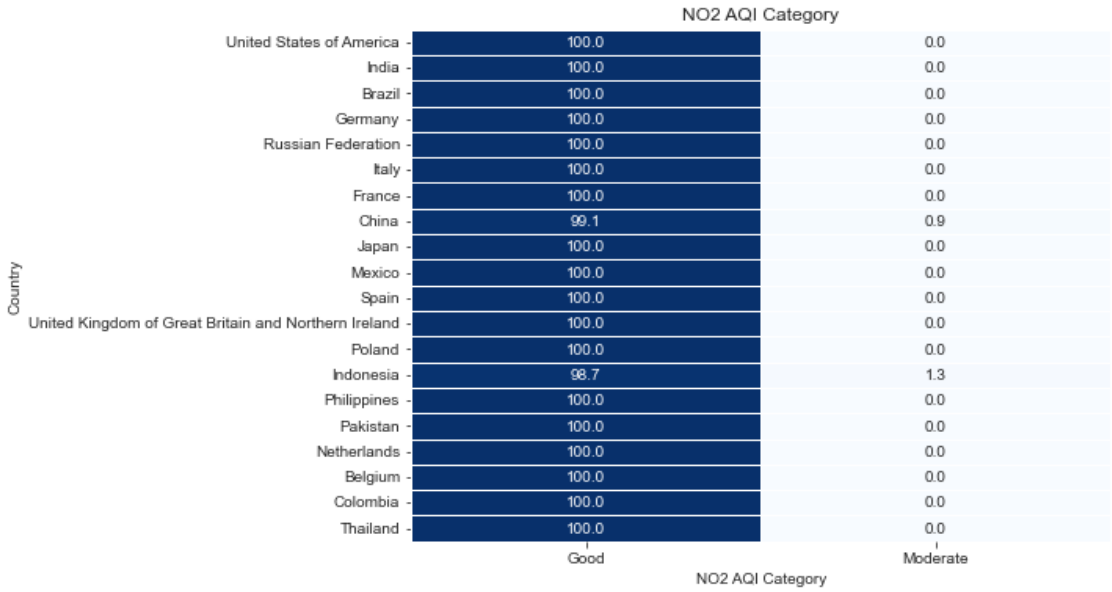
[108]:
```
# Define the columns to process
cols = ['AQI Category', 'CO AQI Category', 'Ozone AQI Category', 'NO2 AQI␣
 ↪Category', 'PM2.5 AQI Category' ]

# Loop over each column
for col in cols:
    # Create a pivot table
    pivot = pd.pivot_table(df,
                           index=['Country'],
                           columns=[col],
                           aggfunc='size',
                           fill_value=0)
```

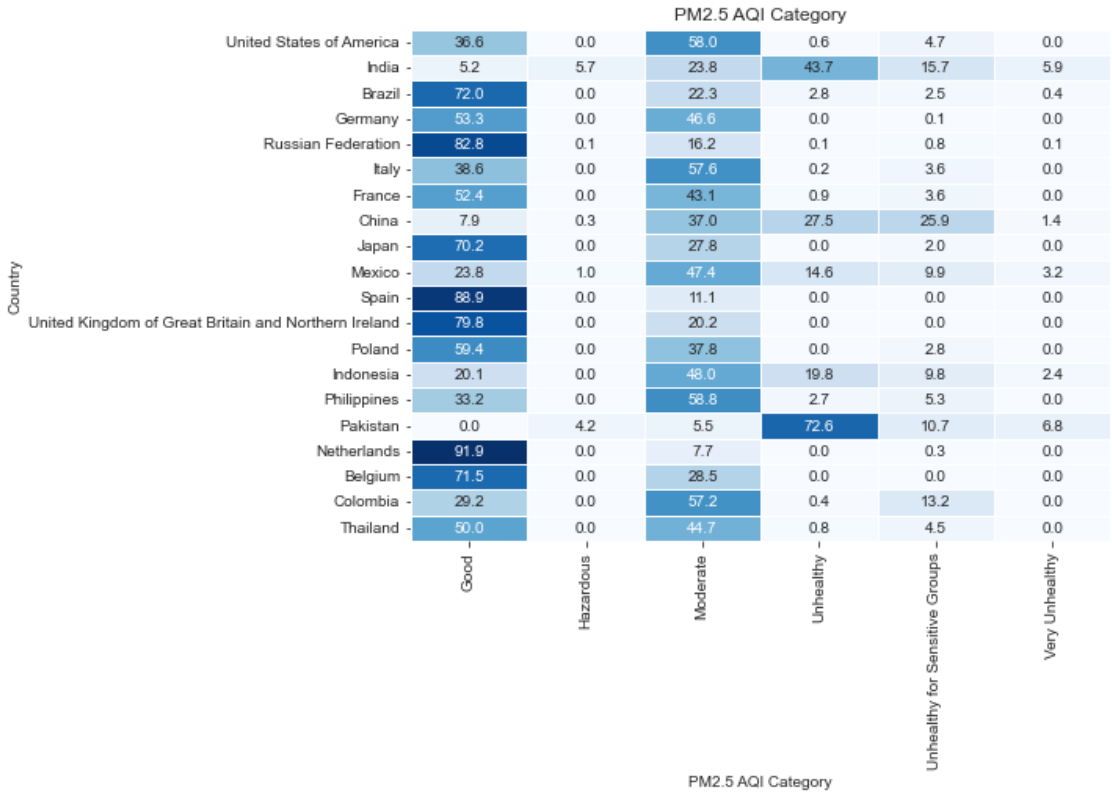
```
   # Select the top countries by the total count of cities
   top_countries = df['Country'].value_counts().head(20).index

   # Calculate the percentages for each cell
   percentages = pivot.loc[top_countries].apply(lambda x: x/x.sum()*100,␣
↪axis=1)

   # Create a heatmap using seaborn
   plt.figure(figsize=(8,6))
   ax = sns.heatmap(percentages, annot=True, cmap="Blues", fmt='.1f',␣
↪linewidths=.5, cbar=False)

   # Customize the heatmap
   ax.set_title(col)
   ax.set_xlabel(col)
   ax.set_ylabel('Country')

   # Show the plot
   plt.show()
```




CO AQI Category
United States of America 100.0 0.0 0.0
India 100.0 0.0 0.0
Brazil 100.0 0.0 0.0
Germany 100.0 0.0 0.0
Russian Federation 100.0 0.0 0.0
Italy 100.0 0.0 0.0
France 100.0 0.0 0.0
China 100.0 0.0 0.0
Japan 100.0 0.0 0.0
Mexico 100.0 0.0 0.0
Spain 100.0 0.0 0.0
United Kingdom of Great Britain and Northern Ireland 100.0 0.0 0.0
Poland 100.0 0.0 0.0
Indonesia 100.0 0.0 0.0
Philippines 100.0 0.0 0.0
Pakistan 100.0 0.0 0.0
Netherlands 100.0 0.0 0.0
Belgium 100.0 0.0 0.0
Colombia 100.0 0.0 0.0
Thailand 100.0 0.0 0.0
Country
Good
Moderate
Unhealthy for Sensitive Groups
CO AQI Category


Ozone AQI Category
United States of America 97.8 2.1 0.0 0.0 0.0
India 70.8 13.3 7.1 8.7 0.1
Brazil 100.0 0.0 0.0 0.0 0.0
Germany 91.3 8.7 0.0 0.0 0.0
Russian Federation 99.4 0.6 0.0 0.0 0.0
Italy 77.8 22.2 0.0 0.0 0.0
France 97.0 3.0 0.0 0.0 0.0
China 40.9 20.5 18.4 15.8 4.4
Japan 87.7 12.0 0.0 0.3 0.0
Mexico 100.0 0.0 0.0 0.0 0.0
Spain 97.9 2.1 0.0 0.0 0.0
United Kingdom of Great Britain and Northern Ireland 100.0 0.0 0.0 0.0 0.0
Poland 100.0 0.0 0.0 0.0 0.0
Indonesia 75.2 14.2 3.4 5.3 1.8
Philippines 100.0 0.0 0.0 0.0 0.0
Pakistan 39.7 18.2 16.9 24.1 1.0
Netherlands 100.0 0.0 0.0 0.0 0.0
Belgium 100.0 0.0 0.0 0.0 0.0
Colombia 100.0 0.0 0.0 0.0 0.0
Thailand 99.6 0.4 0.0 0.0 0.0
Country
Good
Moderate
Unhealthy
Unhealthy for Sensitive Groups
Very Unhealthy
Ozone AQI Category

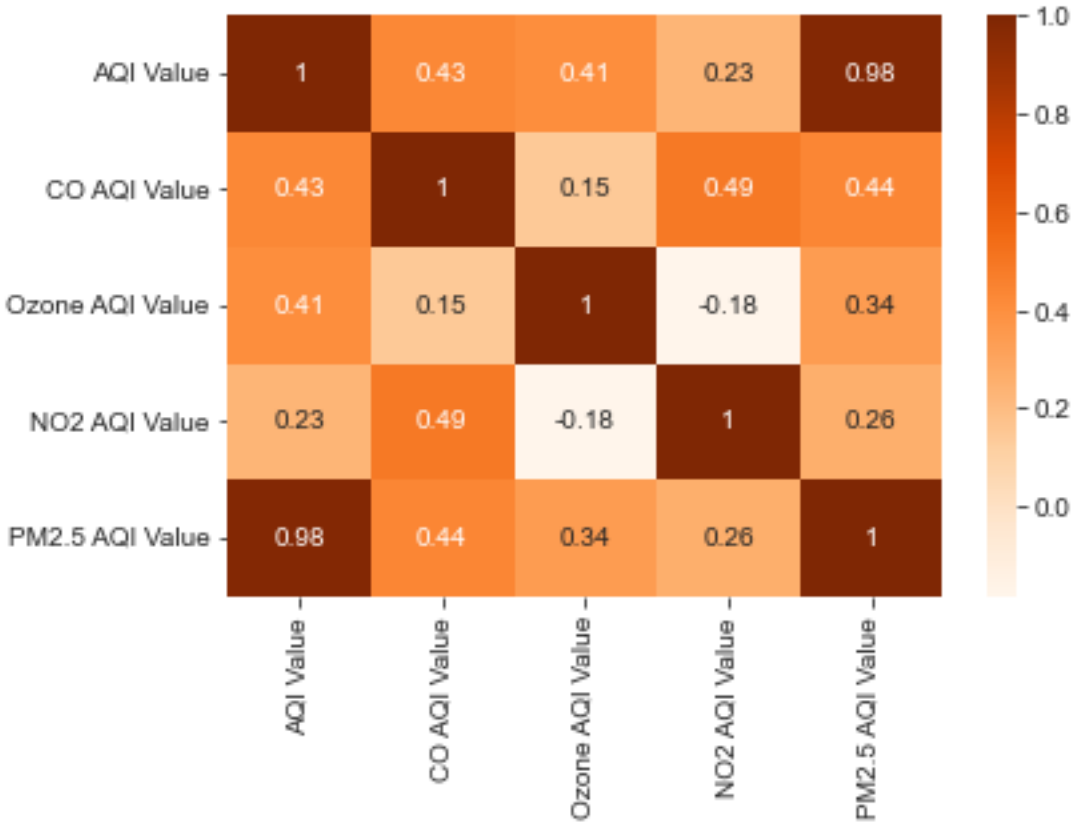




[93]:
```
#Create a heatmap of the correlation matrix using Seaborn's heatmap function
```

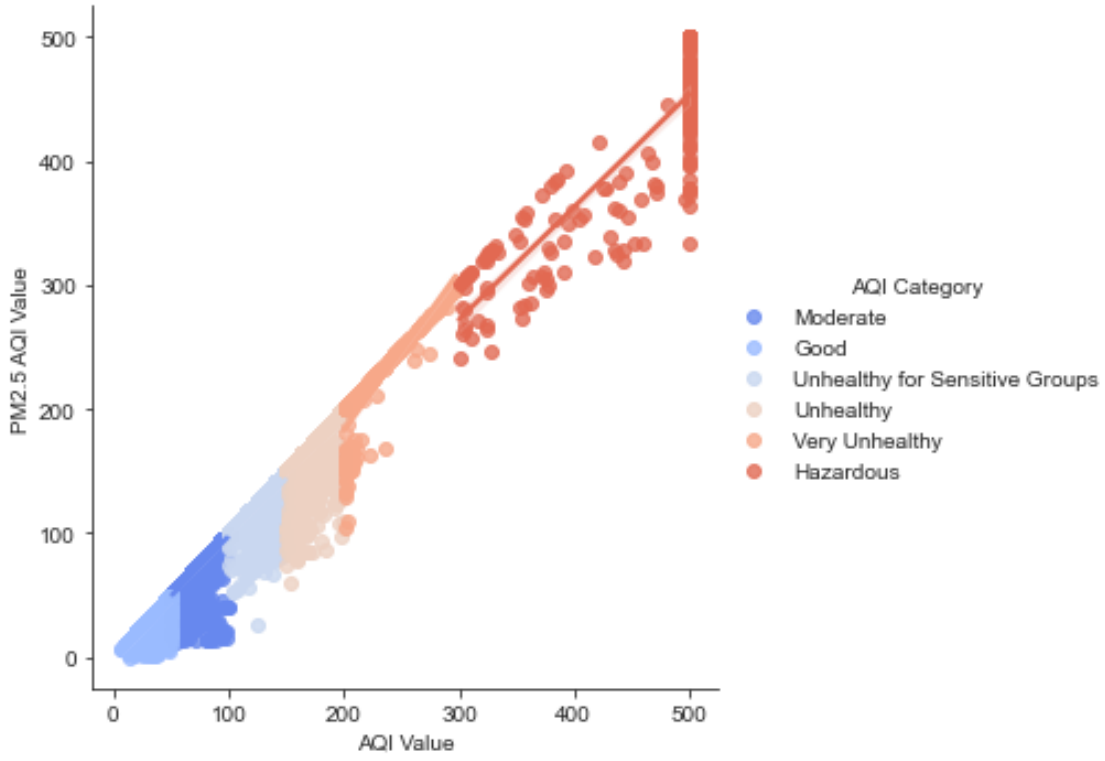
```
sns.heatmap(df.corr(), annot=True, cmap="Oranges")
```

[93]: <AxesSubplot:>



A Pearson correlation coefficient of 0.984 indicates a very strong positive correlation between AQI Value and PM2.5 AQI Value. This suggests that as the AQI Value increases, so does the PM2.5 AQI Value, and vice versa. The correlation coefficient value of 0.984 is very close to 1, which suggests that the relationship between these two variables is almost perfectly linear.

[99]:
```
sns.lmplot(data=df,y='PM2.5 AQI Value',x='AQI Value',hue="AQI Category");
```



[98]:
```
import scipy.stats as stats

# datas
x = df["AQI Value"]
y = df["PM2.5 AQI Value"]

# regression linear computing
slope, intercept, r_value, p_value, std_err = stats.linregress(x, y)

# displaying results
print("Pente :", slope)
print("Ordonnée à l'origine :", intercept)
print("Coefficient de corrélation de Pearson :", r_value)
print("P-value :", p_value)

# plots
sns.lmplot(data=df, y="PM2.5 AQI Value", x="AQI Value", line_kws={"color":␣
 ↪"red"});
plt.grid(True)
plt.show()
```

```
Pente : 0.9622225325525673
Ordonnée à l'origine : -0.7707254399703629
```

```
Coefficient de corrélation de Pearson : 0.9843265891583604
P-value : 0.0
```