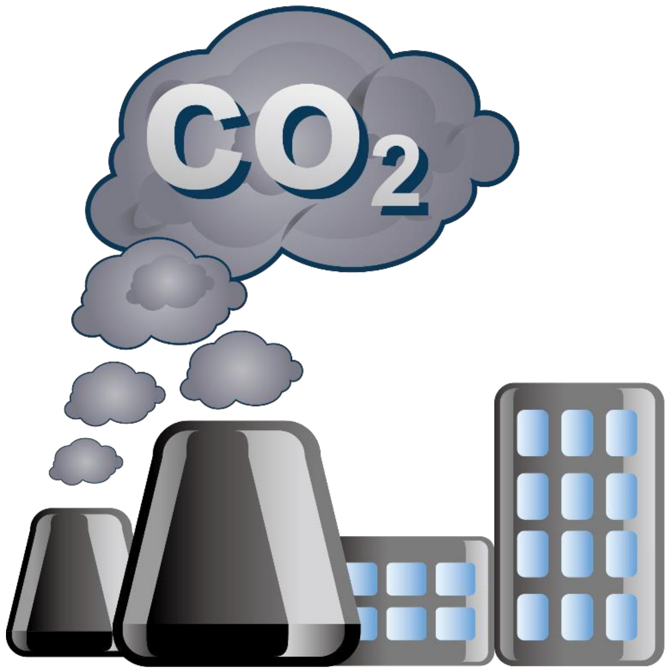
International Information Technology University

Faculty of Computer Technology and Cybersecurity



Air Pollution

Done by: Aibek Shynazbek, Nurkanat Demesin

Group: IT2-2107

Check by: Meruyert T. Aristombayeva

Almaty, 2023

I. Project topic: Air Pollution

II. Project description: Air pollution is one of the most important environmental threats to urban populations, and although all people are exposed to it, pollutant emissions, exposure levels and population vulnerabilities vary from area to area. Exposure to common air pollutants has been linked to respiratory and cardiovascular disease, cancer, and premature death. These indicators provide a measure of air quality and public health in New York City over time and across geographic areas of the city.

III. Dataset description: Dataset contains information on New York City air quality surveillance data in a 16122 row. Link: https://catalog.data.gov/dataset/air-quality.

IV. Research questions:

Descriptive	Diagnostic	Predictive
1. Which area has the most pollution? 2. Which indicator pollutes the air the most?	1. How does geographic type affect the pollution indicator?	1. Predicting future pollution based on time period.

V. Dataset screenshot:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Unique ID	Indicator I	Name	Measure	Measure I	Geo Type I	Geo Join I	Geo Place	Time Perio	Start Date	Data Value	Message
2	216498	386	Ozone (O3	Mean	ppb	CD	313	Coney Isla	Summer 2(	6/1/2013	34.64	
3	216499	386	Ozone (O3	Mean	ppb	CD	313	Coney Isla	Summer 2(	6/1/2014	33.22	
4	219969	386	Ozone (O3	Mean	ppb	Borough	1	Bronx	Summer 2(	6/1/2013	31.25	
5	219970	386	Ozone (O3	Mean	ppb	Borough	1	Bronx	Summer 2(	6/1/2014	31.15	
6	164876	383	Sulfur Dio	Mean	ppb	CD	211	Morris Par	Winter 20(	########	5.89	
7	164877	383	Sulfur Dio	Mean	ppb	CD	212	Williamsbr	Winter 20(	########	5.75	
8	219971	386	Ozone (O3	Mean	ppb	Borough	2	Brooklyn	Summer 2(	6/1/2009	26.27	
9	219972	386	Ozone (O3	Mean	ppb	Borough	2	Brooklyn	Summer 2(	6/1/2010	33.83	
10	164878	383	Sulfur Dio	Mean	ppb	CD	301	Greenpoin	Winter 20(	########	4.33	
11	164879	383	Sulfur Dio	Mean	ppb	CD	302	Fort Green	Winter 20(	########	4.41	
12	164880	383	Sulfur Dio	Mean	ppb	CD	303	Bedford St	Winter 20(	########	4.73	
13	164881	383	Sulfur Dio	Mean	ppb	CD	304	Bushwick (	Winter 20(	########	4.71	
14	164882	383	Sulfur Dio	Mean	ppb	CD	305	East New '	Winter 20(	########	3.78	
15	164883	383	Sulfur Dio	Mean	ppb	CD	306	Park Slope	Winter 20(	########	3.94	
16	164884	383	Sulfur Dio	Mean	ppb	CD	307	Sunset Par	Winter 20(	########	3.78	
17	164885	383	Sulfur Dio	Mean	ppb	CD	308	Crown Hei	Winter 20(	########	4.79	
18	219973	386	Ozone (O3	Mean	ppb	Borough	2	Brooklyn	Summer 2(	6/1/2011	33.19	
19	219974	386	Ozone (O3	Mean	ppb	Borough	2	Brooklyn	Summer 2(	6/1/2012	33.89	
20	219975	386	Ozone (O3	Mean	ppb	Borough	2	Brooklyn	Summer 2(	6/1/2013	31.13	
21	219976	386	Ozone (O3	Mean	ppb	Borough	2	Brooklyn	Summer 2(	6/1/2014	31.29	
22	164930	383	Sulfur Dio	Mean	ppb	CD	206	Belmont a	Winter 20(	########	5.3	
23	164931	383	Sulfur Dio	Mean	ppb	CD	207	Kingsbridg	Winter 20(	########	7.49	
24	130355	639	PM2.5-Att	Estimated	per 100,00	UHF42	101	Kingsbridg	2005-2007	1/1/2005	117.7	
25	130356	639	PM2.5-Att	Estimated	per 100,00	UHF42	102	Northeast	2005-2007	1/1/2005	77.3	
26	130357	639	PM2.5-Att	Estimated	per 100,00	UHF42	103	Fordham -	2005-2007	1/1/2005	67.3	
27	130358	639	PM2.5-Att	Estimated	per 100,00	UHF42	104	Pelham - T	2005-2007	1/1/2005	73.6	
28	130359	639	PM2.5-Att	Estimated	per 100,00	UHF42	105	Crotona -T	2005-2007	1/1/2005	65.8	

VI. Data columns description:

№	Column name	Description	Sample Values
1	Unique ID	Unique Record Identifier is used to marks that one record is unique from any another.	172091, 412533, 667046, 212301 650066.
2	Indicator ID	Identifier of the type of measured value across time and place.	366, 385, 386, 365, 644.
3	Name	Name of the indicator that measured.	NO2, PM2.5, O3, SO2.
4	Measure	How the indicator is measured.	Mean, million miles, Estimated annual rate, Estimated annual rate - children 0 to 17 years old, Estimated annual rate - 18 years old and over.
5	Measure Info	Information (such as units) about the measure.	per 100,000 adults, per km2, per 100,000 children.
6	Geo Type Name	Geography type. For instance, Citywide, Borough, and Community Districts are different geography types. #Offtop: UHF' stands for United Hospital Fund neighborhoods	UHF42, CD, UHF34, Borough, Citywide.
7	Geo Join ID	Identifier of the neighborhood geographic area, used for joining to mapping geography files to make thematic maps.	302, 209, 207, 407, 206.
8	Geo Place Name	Neighborhood name.	West Queens, Downtown - Heights – Slope, Southeast Queens.
9	Time Period	Description of the time that the data applies to; Could be a year, range of years, or season for example.	2005-2007, 2015-2017, Summer 2012.

10	Start Date	Date value for the start of the time period; Always a date value; could be useful for plotting a time series.	1/1/2005.
11	Data Value	The actual data value for this indicator, measure, place, and time.	2.8, 2, 2.1, 80, 71.
12	Message		Notes that apply to the data value; For example, if an estimate is based on small numbers we will detail here.

VII. Dataset description:

1. Info about dataset

```
Ввод [5]: airp.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16122 entries, 0 to 16121
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unique_ID       16122 non-null  int64
 1   Indicator_ID    16122 non-null  int64
 2   Name            16121 non-null  object
 3   Measure         16122 non-null  object
 4   Measure_Info    16122 non-null  object
 5   Geo_Type_Name   16121 non-null  object
 6   Geo_Join_ID     16122 non-null  int64
 7   Geo_Place_Name  16122 non-null  object
 8   Time_Period     16121 non-null  object
 9   Start_Date      16121 non-null  object
 10  Data_Value      16122 non-null  float64
 11  Message         0 non-null      float64
dtypes: float64(2), int64(3), object(7)
memory usage: 1.5+ MB
```

2. Info about columns

```
Ввод [2]: air_pollution.columns

Out[2]: Index(['Unique ID', 'Indicator ID', 'Name', 'Measure', 'Measure Info',
               'Geo Type Name', 'Geo Join ID', 'Geo Place Name', 'Time Period',
               'Start_Date', 'Data Value', 'Message'],
              dtype='object')
```

3. Dataset

Ввод [18]:
```
air_pollution
```

Out[18]:

	Unique ID	Indicator ID	Name	Measure	Measure Info	Geo Type Name	Geo Join ID	Geo Place Name	Time Period	Start_Date	Data Value	Message
0	216498	386	Ozone (O3)	Mean	ppb	CD	313	Coney Island (CD13)	Summer 2013	06/01/2013	34.64	NaN
1	216499	386	Ozone (O3)	Mean	ppb	CD	313	Coney Island (CD13)	Summer 2014	06/01/2014	33.22	NaN
2	219969	386	Ozone (O3)	Mean	ppb	Borough	1	Bronx	Summer 2013	06/01/2013	31.25	NaN
3	219970	386	Ozone (O3)	Mean	ppb	Borough	1	Bronx	Summer 2014	06/01/2014	31.15	NaN
4	164876	383	Sulfur Dioxide (SO2)	Mean	ppb	CD	211	Morris Park and Bronxdale (CD11)	Winter 2008-09	12/01/2008	5.89	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
16117	671118	386	Ozone (O3)	Mean	ppb	CD	306	Park Slope and Carroll Gardens (CD6)	Summer 2020	06/01/2020	28.70	NaN
16118	671119	386	Ozone (O3)	Mean	ppb	CD	305	East New York and Starrett City (CD5)	Summer 2020	06/01/2020	29.56	NaN
16119	671120	386	Ozone (O3)	Mean	ppb	CD	304	Bushwick (CD4)	Summer 2020	06/01/2020	29.65	NaN
16120	671121	386	Ozone (O3)	Mean	ppb	CD	303	Bedford Stuyvesant (CD3)	Summer 2020	06/01/2020	29.28	NaN
16121	671122	386	Ozone (O3)	Mean	ppb	CD	302	Fort Greene and Brooklyn Heights (CD2)	Summer 2020	06/01/2020	28.93	NaN

16122 rows × 12 columns

VIII. Data Cleaning and Researching:

Ввод [6]:
```
airp = airp.drop(columns = ["Unique_ID", "Measure_Info", "Message"])
airp.columns
```

Out[6]:
```
Index(['Indicator_ID', 'Name', 'Measure', 'Geo_Type_Name', 'Geo_Join_ID',
       'Geo_Place_Name', 'Time_Period', 'Start_Date', 'Data_Value'],
      dtype='object')
```

Ввод [7]:
```
airp = airp.dropna()
airp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16118 entries, 0 to 16121
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Indicator_ID    16118 non-null  int64
 1   Name            16118 non-null  object
 2   Measure         16118 non-null  object
 3   Geo_Type_Name   16118 non-null  object
 4   Geo_Join_ID     16118 non-null  int64
 5   Geo_Place_Name  16118 non-null  object
 6   Time_Period     16118 non-null  object
 7   Start_Date      16118 non-null  object
 8   Data_Value      16118 non-null  float64
dtypes: float64(1), int64(2), object(6)
memory usage: 1.2+ MB
```

```
In [80]: time_period = airp["Time_Period"]
         time_period

Out[80]: 0          Summer 2013
         1          Summer 2014
         2          Summer 2013
         3          Summer 2014
         4        Winter 2008-09
                      ...
         16117      Summer 2020
         16118      Summer 2020
         16119      Summer 2020
         16120      Summer 2020
         16121      Summer 2020
         Name: Time_Period, Length: 16118, dtype: object
```

```
In [81]: import re
         year_regex = re.compile(r'\d{4}')

         def extract_year(row):
             match = year_regex.search(row['Time_Period'])
             if match:
                 return match.group()
             else:
                 return None

         airp['ear'] = airp.apply(extract_year, axis=1)

         airp = airp.dropna()
         airp
```

Out[81]:

	Indicator_ID	Name	Measure	Geo_Type_Name	Geo_Join_ID	Geo_Place_Name	Time_Period	Start_Date	Data_Value	Year
0	386	Ozone (O3)	Mean	CD	313	Coney Island (CD13)	Summer 2013	6/1/2013	34.64	2013
1	386	Ozone (O3)	Mean	CD	313	Coney Island (CD13)	Summer 2014	6/1/2014	33.22	2014
2	386	Ozone (O3)	Mean	Borough	1	Bronx	Summer 2013	6/1/2013	31.25	2013
3	386	Ozone (O3)	Mean	Borough	1	Bronx	Summer 2014	6/1/2014	31.15	2014
4	383	Sulfur Dioxide (SO2)	Mean	CD	211	Morris Park and Bronxdale (CD11)	Winter 2008-09	12/1/2008	5.89	2008
...	...	...	...	...	...	...	...	...	...	...
16117	386	Ozone (O3)	Mean	CD	306	Park Slope and Carroll Gardens (CD6)	Summer 2020	6/1/2020	28.70	2020
16118	386	Ozone (O3)	Mean	CD	305	East New York and Starrett City (CD5)	Summer 2020	6/1/2020	29.56	2020
16119	386	Ozone (O3)	Mean	CD	304	Bushwick (CD4)	Summer 2020	6/1/2020	29.65	2020
16120	386	Ozone (O3)	Mean	CD	303	Bedford Stuyvesant (CD3)	Summer 2020	6/1/2020	29.28	2020
16121	386	Ozone (O3)	Mean	CD	302	Fort Greene and Brooklyn Heights (CD2)	Summer 2020	6/1/2020	28.93	2020

16118 rows × 10 columns

In [83]:
```
airp = airp.drop(columns = "Time_Period")
airp
```

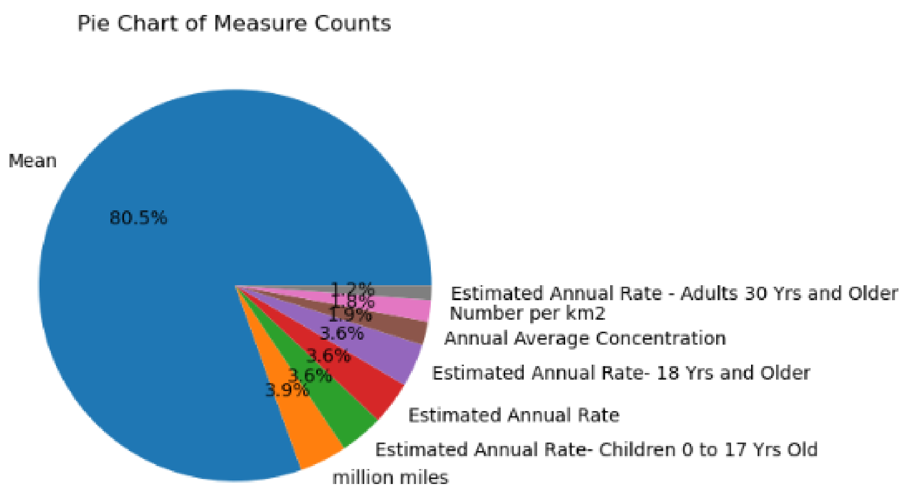
Out[83]:

	Indicator_ID	Name	Measure	Geo_Type_Name	Geo_Join_ID	Geo_Place_Name	Start_Date	Data_Value	Year
0	386	Ozone (O3)	Mean	CD	313	Coney Island (CD13)	6/1/2013	34.64	2013
1	386	Ozone (O3)	Mean	CD	313	Coney Island (CD13)	6/1/2014	33.22	2014
2	386	Ozone (O3)	Mean	Borough	1	Bronx	6/1/2013	31.25	2013
3	386	Ozone (O3)	Mean	Borough	1	Bronx	6/1/2014	31.15	2014
4	383	Sulfur Dioxide (SO2)	Mean	CD	211	Morris Park and Bronxdale (CD11)	12/1/2008	5.89	2008
...	...	...	...	...	...	...	...	...	...
16117	386	Ozone (O3)	Mean	CD	306	Park Slope and Carroll Gardens (CD6)	6/1/2020	28.70	2020
16118	386	Ozone (O3)	Mean	CD	305	East New York and Starrett City (CD5)	6/1/2020	29.56	2020
16119	386	Ozone (O3)	Mean	CD	304	Bushwick (CD4)	6/1/2020	29.65	2020
16120	386	Ozone (O3)	Mean	CD	303	Bedford Stuyvesant (CD3)	6/1/2020	29.28	2020
16121	386	Ozone (O3)	Mean	CD	302	Fort Greene and Brooklyn Heights (CD2)	6/1/2020	28.93	2020

16118 rows × 9 columns

Visualization:

In [84]:
```
measure_counts = airp['Measure'].value_counts()
labels = measure_counts.index
sizes = measure_counts.values
fig = plt.figure()
ax = fig.add_subplot(111)
ax.pie(sizes, labels=labels, autopct='%1.1f%%')
ax.set_title('Pie Chart of Measure Counts')
plt.show()
```

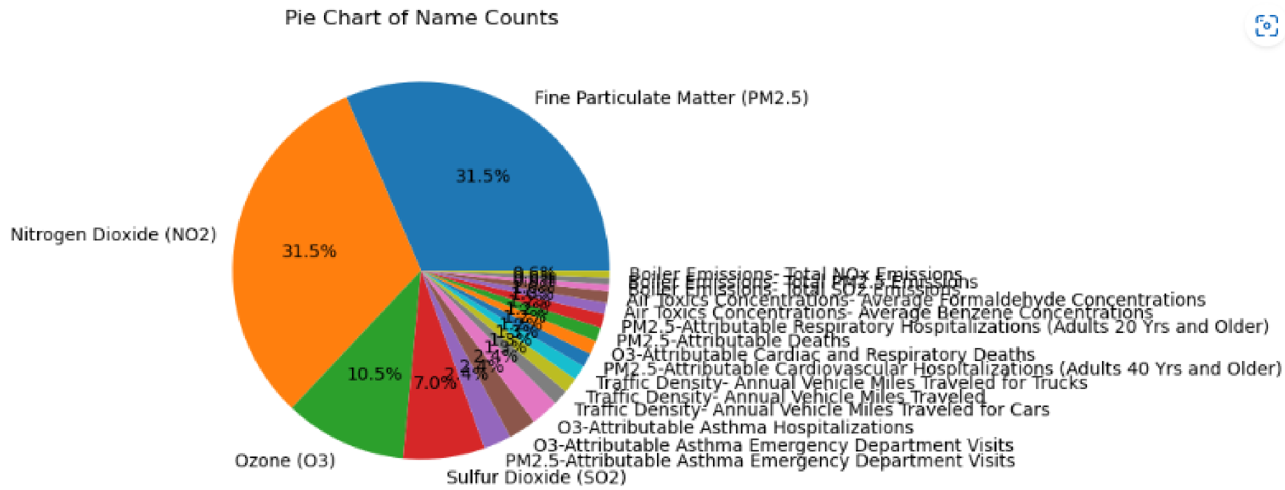


In [85]:
```
name_counts = airp["Name"].value_counts()
name_counts
```

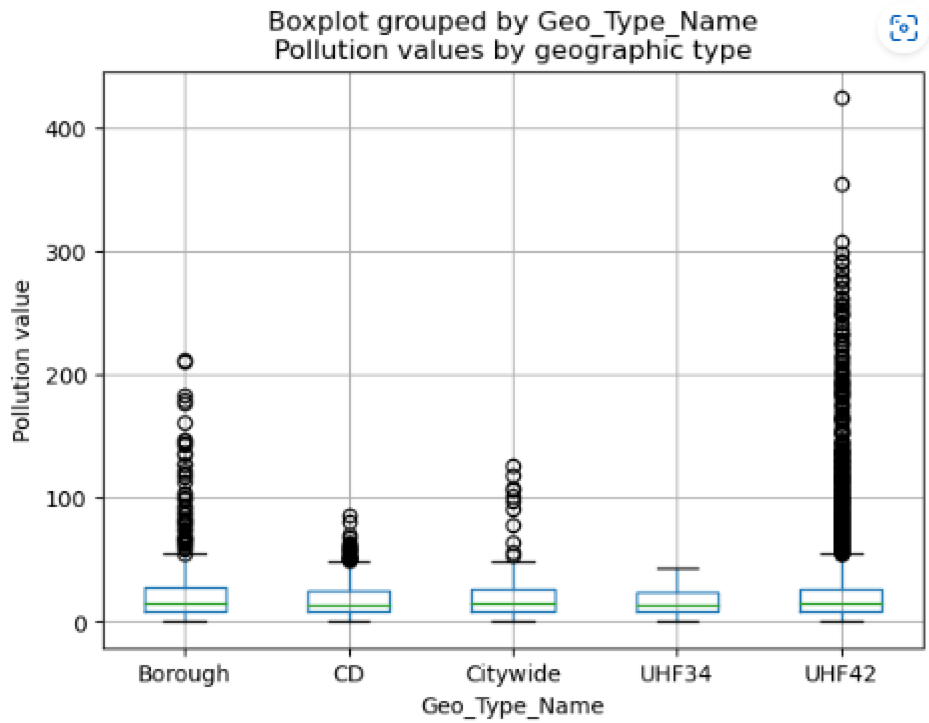
Out[85]:
```
Fine Particulate Matter (PM2.5)                                          5076
Nitrogen Dioxide (NO2)                                                   5075
Ozone (O3)                                                               1692
Sulfur Dioxide (SO2)                                                     1126
PM2.5-Attributable Asthma Emergency Department Visits                     384
O3-Attributable Asthma Emergency Department Visits                        384
O3-Attributable Asthma Hospitalizations                                   384
Traffic Density- Annual Vehicle Miles Traveled for Cars                   213
Traffic Density- Annual Vehicle Miles Traveled                            209
Traffic Density- Annual Vehicle Miles Traveled for Trucks                 209
PM2.5-Attributable Cardiovascular Hospitalizations (Adults 40 Yrs and Older)    192
O3-Attributable Cardiac and Respiratory Deaths                            192
PM2.5-Attributable Deaths                                                 192
PM2.5-Attributable Respiratory Hospitalizations (Adults 20 Yrs and Older)   192
Air Toxics Concentrations- Average Benzene Concentrations                 155
Air Toxics Concentrations- Average Formaldehyde Concentrations            155
Boiler Emissions- Total SO2 Emissions                                      96
Boiler Emissions- Total PM2.5 Emissions                                    96
Boiler Emissions- Total NOx Emissions                                      96
Name: Name, dtype: int64
```

In [86]:

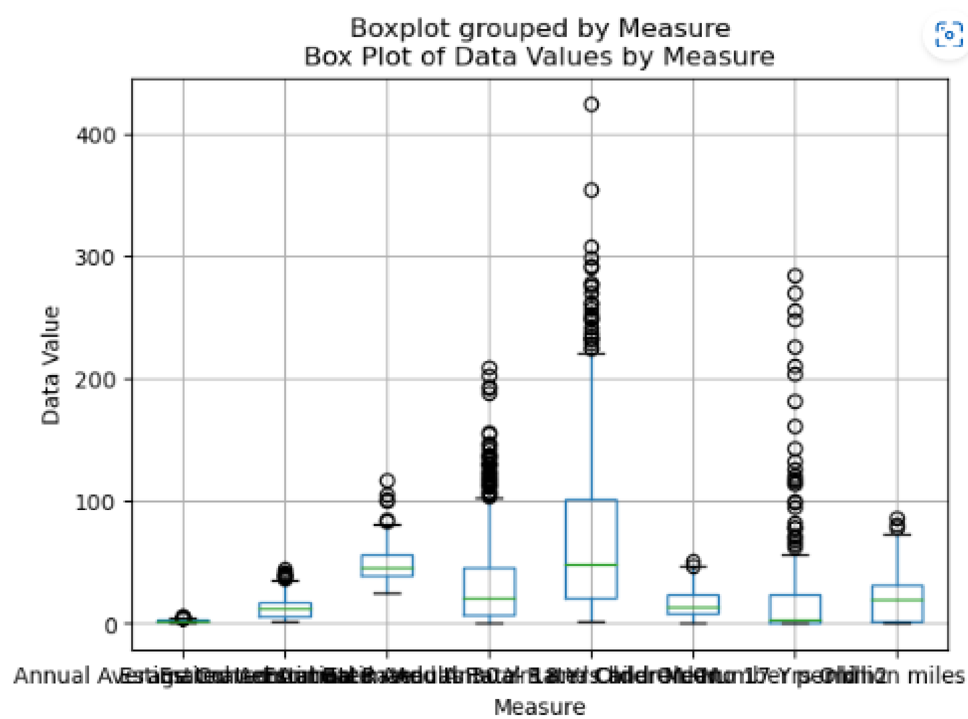
```python
labels = name_counts.index
sizes = name_counts.values
fig = plt.figure()
ax = fig.add_subplot(111)
ax.pie(sizes, labels=labels, autopct='%1.1f%%')
ax.set_title('Pie Chart of Name Counts')
plt.show()
```



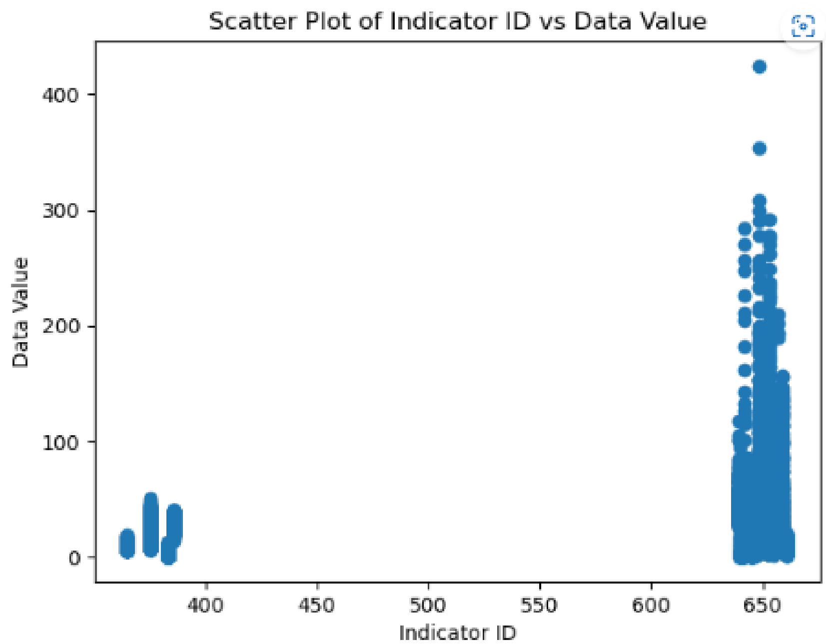
In [91]:
```
fig, ax = plt.subplots()
airp.boxplot(column='Data_Value', by='Geo_Type_Name', ax=ax)
ax.set_ylabel('Pollution value')
ax.set_title('Pollution values by geographic type')
plt.show()
```



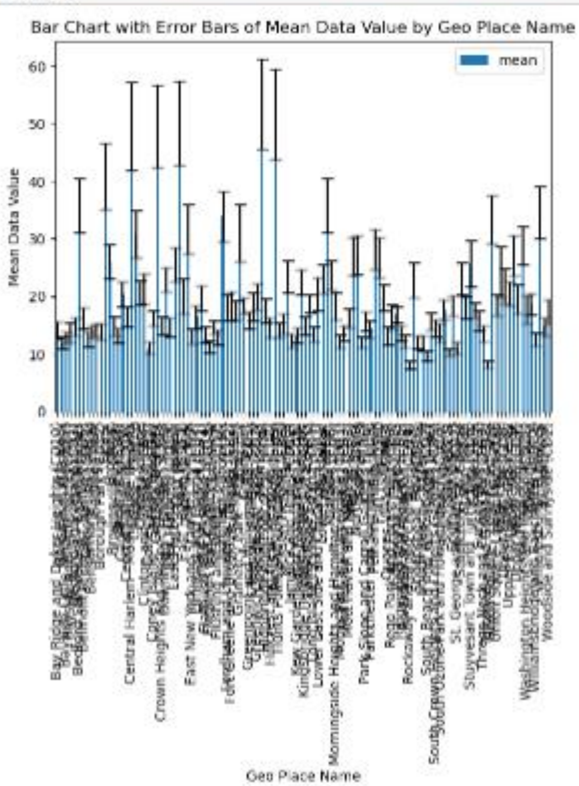
In [93]:
```
airp.boxplot(column='Data_Value', by='Measure')
plt.xlabel('Measure')
plt.ylabel('Data Value')
plt.title('Box Plot of Data Values by Measure')
plt.show()
```



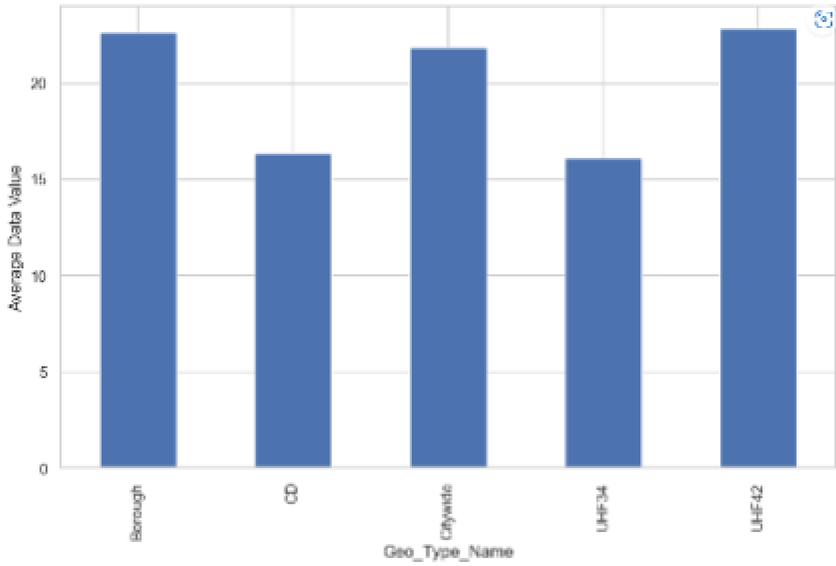
In [92]:
```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.scatter(airp['Indicator_ID'], airp['Data_Value'])
ax.set_xlabel('Indicator ID')
ax.set_ylabel('Data Value')
ax.set_title('Scatter Plot of Indicator ID vs Data Value')
plt.show()
```



In [51]:
```
place_data = airp.groupby('Geo_Place_Name').agg({'Data_Value': ['mean', 'sem']})


place_data.plot(kind='bar', y='Data_Value', yerr='sem', capsize=4)
plt.xlabel('Geo Place Name')
plt.ylabel('Mean Data Value')
plt.title('Bar Chart with Error Bars of Mean Data Value by Geo Place Name')
plt.show()
```



In [107]:
```
fig, ax = plt.subplots(figsize=(10,6))
airp.groupby("Geo_Type_Name")["Data_Value"].mean().plot(kind="bar", ax=ax)
ax.set_ylabel("Average Data Value")
plt.show()
```



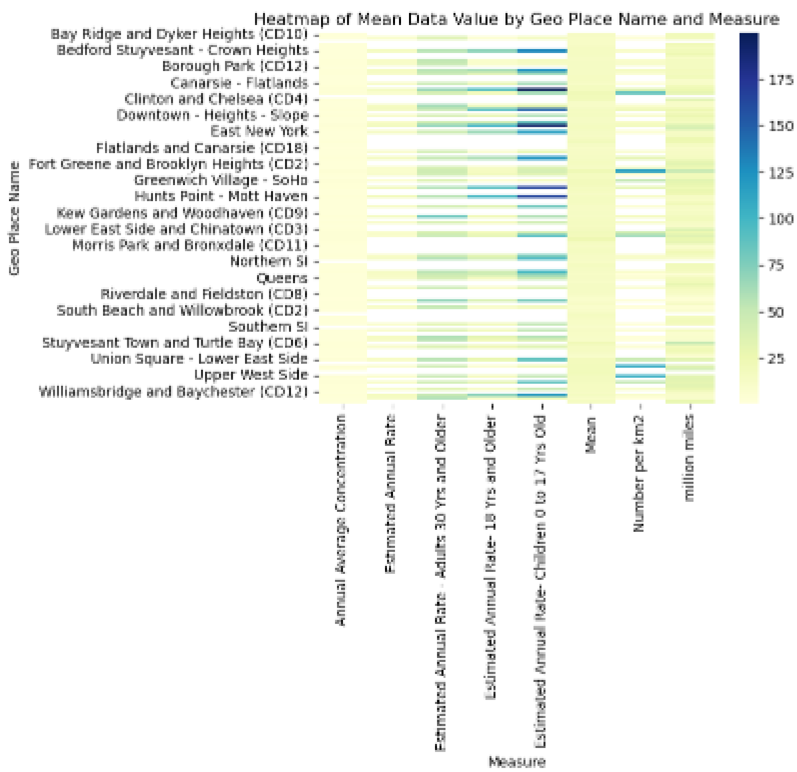
In [94]:
```
mean_by_geo = airp.groupby(['Geo_Type_Name', 'Name'])['Data_Value'].mean()
print('Mean Data_Value by Geo_Type_Name and Name:\n', mean_by_geo)
```

```
Mean Data_Value by Geo_Type_Name and Name:
 Geo_Type_Name  Name
Borough        Air Toxics Concentrations- Average Benzene Concentrations                   2.150000
               Air Toxics Concentrations- Average Formaldehyde Concentrations              2.610000
               Boiler Emissions- Total NOx Emissions                                      46.880000
               Boiler Emissions- Total PM2.5 Emissions                                     1.240000
               Boiler Emissions- Total SO2 Emissions                                       9.930000
                                                                                             ...
UHF42          PM2.5-Attributable Respiratory Hospitalizations (Adults 20 Yrs and Older)  14.855383
               Sulfur Dioxide (SO2)                                                        2.600476
               Traffic Density- Annual Vehicle Miles Traveled                             29.790476
               Traffic Density- Annual Vehicle Miles Traveled for Cars                    27.880723
               Traffic Density- Annual Vehicle Miles Traveled for Trucks                   1.647619
Name: Data_Value, Length: 70, dtype: float64
```

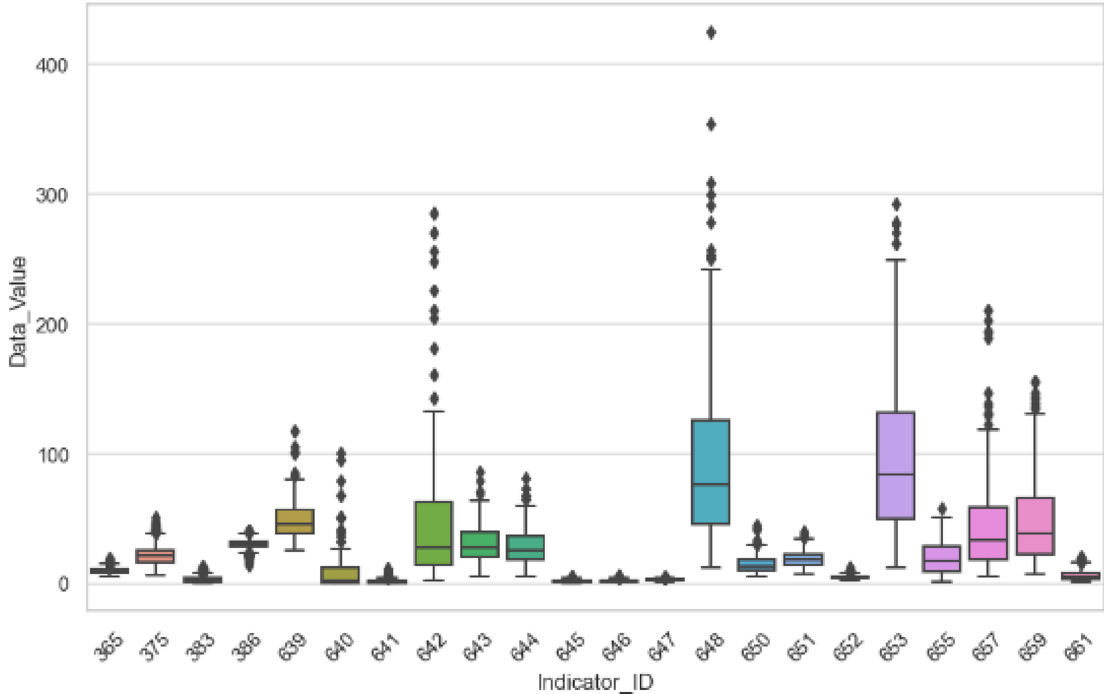
```
In [95]: sns.heatmap(airp[['Data_Value', 'Geo_Join_ID']].corr(), annot=True, cmap='coolwarm')
         plt.title('Correlation between Data_Value and Geo_Join_ID')
         plt.show()
```


```
In [52]: import seaborn as sns
         pivot_data = airp.pivot_table(values='Data_Value', index='Geo_Place_Name', columns='Measure', aggfunc='mean')
```

```
In [53]: sns.heatmap(pivot_data, cmap='YlGnBu')
         plt.xlabel('Measure')
         plt.ylabel('Geo Place Name')
         plt.title('Heatmap of Mean Data Value by Geo Place Name and Measure')
         plt.show()
```



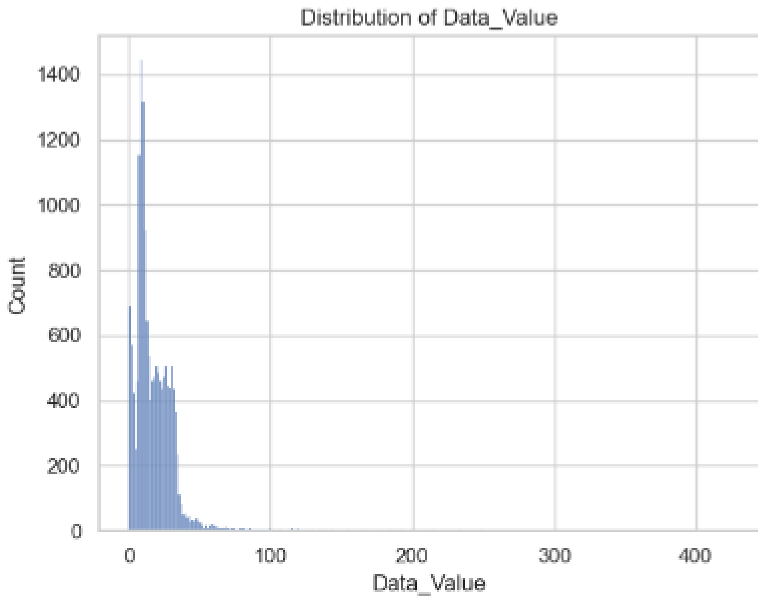
In [97]:
```
sns.set(style="whitegrid")

fig, ax = plt.subplots(figsize=(10,6))
sns.boxplot(x="Indicator_ID", y="Data_Value", data=airp, ax=ax)
plt.xticks(rotation=45)
plt.show()
```



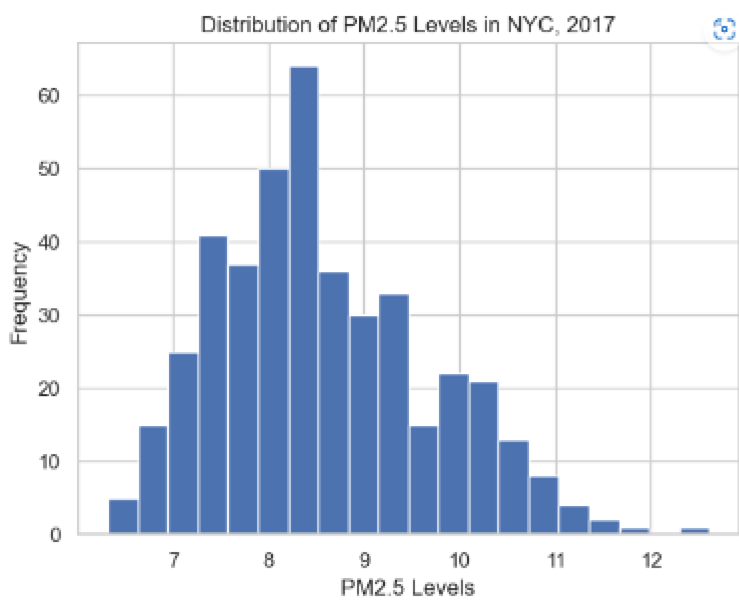
In [98]:
```
airp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16118 entries, 0 to 16121
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Indicator_ID    16118 non-null  int64  
 1   Name            16118 non-null  object 
 2   Measure         16118 non-null  object 
 3   Geo_Type_Name   16118 non-null  object 
 4   Geo_Join_ID     16118 non-null  int64  
 5   Geo_Place_Name  16118 non-null  object 
 6   Start_Date      16118 non-null  object 
 7   Data_Value      16118 non-null  float64
 8   Year            16118 non-null  object 
dtypes: float64(1), int64(2), object(6)
memory usage: 1.2+ MB
```

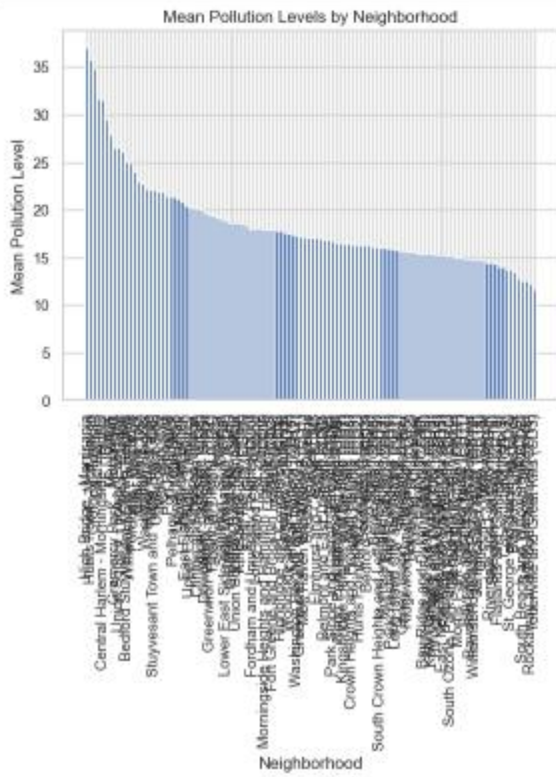
In [99]:
```
sns.histplot(airp['Data_Value'])
plt.title('Distribution of Data_Value')
plt.show()
```



In [101]:
```
pm25_2017 = airp[(airp["Name"] == "Fine Particulate Matter (PM2.5)") & (airp["Year"] == "2017")]

plt.hist(pm25_2017["Data_Value"], bins=20)
plt.xlabel("PM2.5 Levels")
plt.ylabel("Frequency")
plt.title("Distribution of PM2.5 Levels in NYC, 2017")
plt.show()
```



```
In [102]: pollution_by_area = airp.groupby('Geo_Place_Name')['Data_Value'].mean()

          pollution_by_area = pollution_by_area.sort_values(ascending=False)

          plt.bar(pollution_by_area.index, pollution_by_area.values)
          plt.xticks(rotation=90)
          plt.xlabel('Neighborhood')
          plt.ylabel('Mean Pollution Level')
          plt.title('Mean Pollution Levels by Neighborhood')
          plt.show()
```

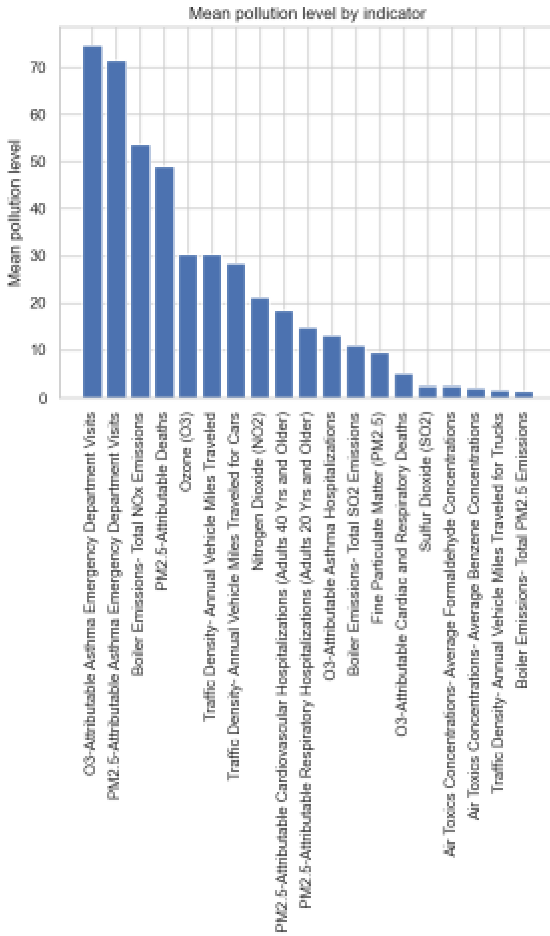



```
In [103]: print("The area with the highest mean pollution level is", pollution_by_area.index[0])

          The area with the highest mean pollution level is High Bridge - Morrisania
```

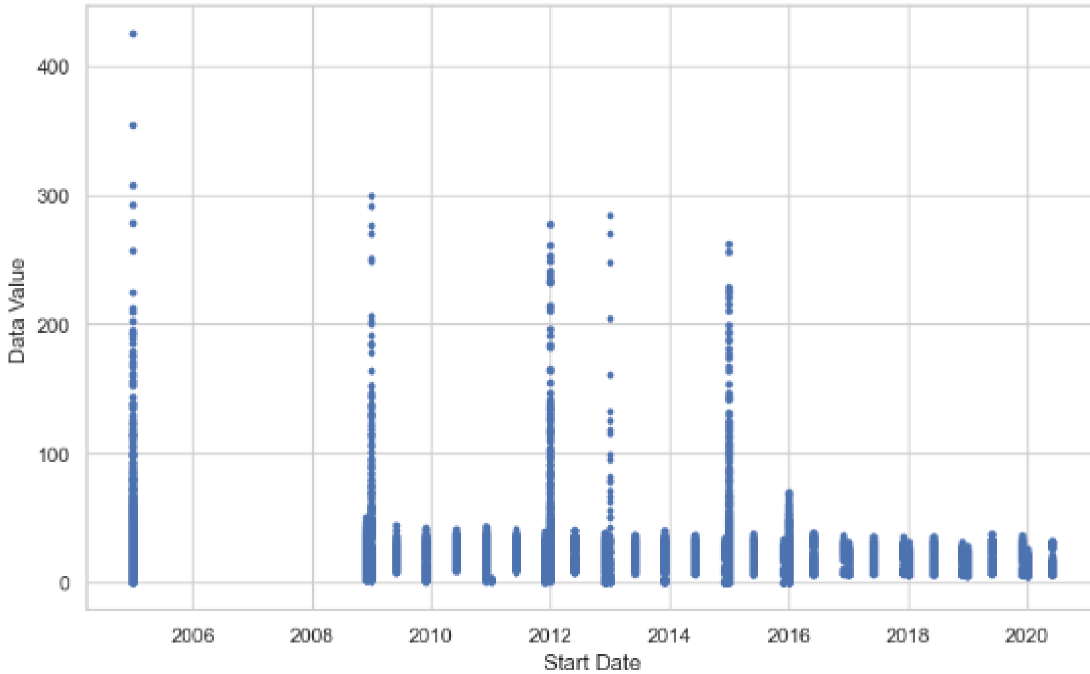
```
In [104]: pollution_by_indicator = airp.groupby('Name')['Data_Value'].mean()

          pollution_by_indicator = pollution_by_indicator.sort_values(ascending=False)

          print(pollution_by_indicator)

Name
O3-Attributable Asthma Emergency Department Visits                        74.718229
PM2.5-Attributable Asthma Emergency Department Visits                     71.417994
Boiler Emissions- Total NOx Emissions                                     53.791667
PM2.5-Attributable Deaths                                                 49.116530
Ozone (O3)                                                                30.367398
Traffic Density- Annual Vehicle Miles Traveled                            30.307177
Traffic Density- Annual Vehicle Miles Traveled for Cars                   28.329577
Nitrogen Dioxide (NO2)                                                    21.275992
PM2.5-Attributable Cardiovascular Hospitalizations (Adults 40 Yrs and Older)  18.554893
PM2.5-Attributable Respiratory Hospitalizations (Adults 20 Yrs and Older)     14.865290
O3-Attributable Asthma Hospitalizations                                   13.119531
Boiler Emissions- Total SO2 Emissions                                     10.991667
Fine Particulate Matter (PM2.5)                                            9.516063
O3-Attributable Cardiac and Respiratory Deaths                             4.995312
Sulfur Dioxide (SO2)                                                       2.614607
Air Toxics Concentrations- Average Formaldehyde Concentrations             2.481290
Air Toxics Concentrations- Average Benzene Concentrations                  2.030201
Traffic Density- Annual Vehicle Miles Traveled for Trucks                  1.679426
Boiler Emissions- Total PM2.5 Emissions                                    1.373958
Name: Data_Value, dtype: float64
```

In [105]:
```python
fig, ax = plt.subplots()
ax.bar(pollution_by_indicator.index, pollution_by_indicator)
ax.set_xticklabels(pollution_by_indicator.index, rotation=90)
ax.set_ylabel('Mean pollution level')
ax.set_title('Mean pollution level by indicator')
plt.show()
```

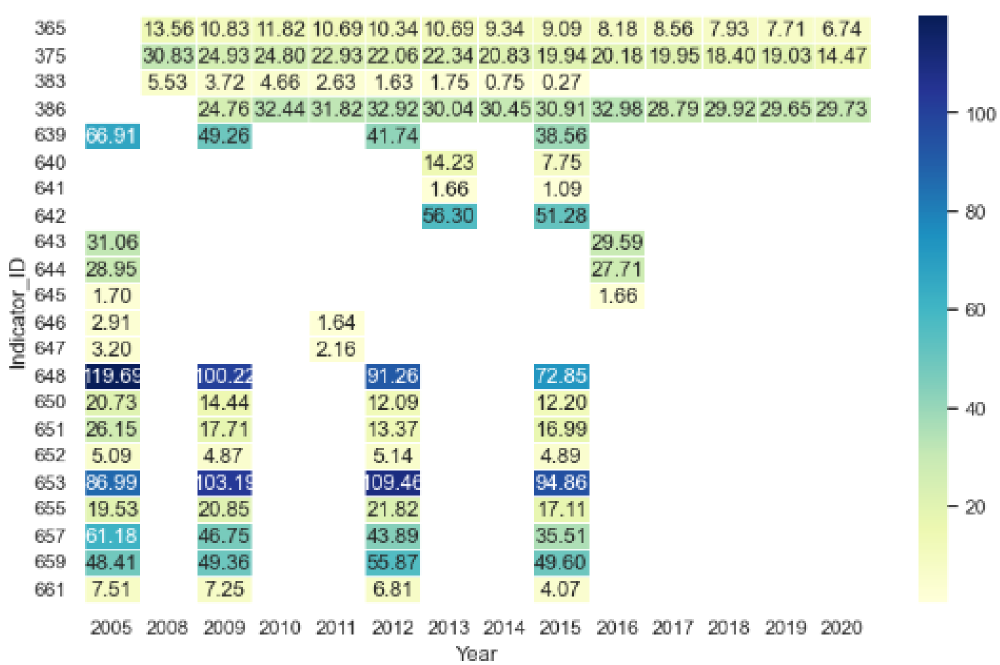
```
C:\Users\aakku\AppData\Local\Temp\ipykernel_2040\1181977917.py:3: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels(pollution_by_indicator.index, rotation=90)
```



In [106]:
```python
airp["Start_Date"] = pd.to_datetime(airp["Start_Date"])

fig, ax = plt.subplots(figsize=(10,6))
ax.scatter(airp["Start_Date"], airp["Data_Value"], s=10)
ax.set_xlabel("Start Date")
ax.set_ylabel("Data Value")
plt.show()
```



In [109]:
```python
data_pivot = airp.pivot_table(index="Indicator_ID", columns="Year", values="Data_Value", aggfunc="mean")

fig, ax = plt.subplots(figsize=(10,6))
sns.heatmap(data_pivot, cmap="YlGnBu", annot=True, fmt=".2f", linewidths=.5, ax=ax)
plt.show()
```



XI . Conclusion

In our project we remove the two columns Unique ID and Message. Check the null value in a row and we write a code how to remove column with null values and replace null values with an average. Then, we experimented with our dataset and see some info about dataset. At the end, we save our changed dataset as a new csv file by the name “Air Pollution”.