In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv("D:\\AIT614\\Project\\Air_Quality.csv")
print(df)
```

```
       Unique ID  Indicator ID                                   Name  \
0         172653           375                 Nitrogen dioxide (NO2)
1         172585           375                 Nitrogen dioxide (NO2)
2         336637           375                 Nitrogen dioxide (NO2)
3         336622           375                 Nitrogen dioxide (NO2)
4         172582           375                 Nitrogen dioxide (NO2)
...          ...           ...                                   ...
16213     130750           647       Outdoor Air Toxics - Formaldehyde
16214     130780           647       Outdoor Air Toxics - Formaldehyde
16215     131020           652  Cardiac and respiratory deaths due to Ozone
16216     131026           652  Cardiac and respiratory deaths due to Ozone
16217     325247           643           Annual vehicle miles traveled

                          Measure Measure Info Geo Type Name  Geo Join ID  \
0                            Mean          ppb         UHF34          203
1                            Mean          ppb         UHF34          203
2                            Mean          ppb         UHF34          204
3                            Mean          ppb         UHF34          103
4                            Mean          ppb         UHF34          104
...                           ...          ...           ...          ...
16213  Annual average concentration        µg/m3         UHF42          211
16214  Annual average concentration        µg/m3       Borough            5
16215         Estimated annual rate  per 100,000         UHF42          504
16216         Estimated annual rate  per 100,000       Borough            5
16217                 million miles      per km2            CD          107

                         Geo Place Name          Time Period Start_Date  \
0      Bedford Stuyvesant - Crown Heights  Annual Average 2011  12/1/2010
1      Bedford Stuyvesant - Crown Heights  Annual Average 2009  12/1/2008
2                          East New York  Annual Average 2015   1/1/2015
3                    Fordham - Bronx Pk  Annual Average 2015   1/1/2015
4                   Pelham - Throgs Neck  Annual Average 2009  12/1/2008
...                                  ...                 ...        ...
16213             Williamsburg - Bushwick                2005   1/1/2005
16214                      Staten Island                2005   1/1/2005
16215           South Beach - Tottenville           2005-2007   1/1/2005
16216                      Staten Island           2005-2007   1/1/2005
16217               Upper West Side (CD7)                2016   1/1/2016

       Data Value  Message
0           25.30      NaN
1           26.93      NaN
2           19.09      NaN
3           19.76      NaN
4           22.83      NaN
...           ...      ...
16213        3.10      NaN
16214        2.30      NaN
16215        7.50      NaN
16216        7.80      NaN
16217       50.00      NaN

[16218 rows x 12 columns]
```

In [2]:

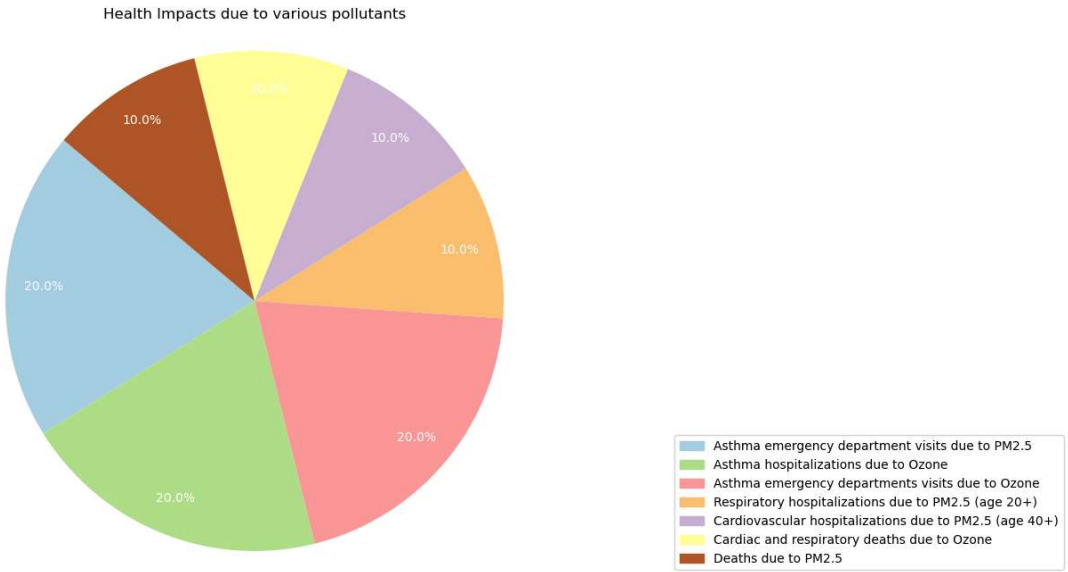
```
# List of keywords
keywords = ["Asthma", "Cardiovascular", "Respiratory", "Death", "Cardiac and respiratc

# Filter rows based on keywords in the "Name" column
filtered_data = df[df['Name'].str.contains('|'.join(keywords))]
```

```
# Display the filtered data
print(filtered_data)
```

```
       Unique ID  Indicator ID  \
766       518895           648
767       628444           648
768       518888           648
769       518906           648
770       518926           648
...          ...           ...
16189     628484           657
16190     131266           657
16212     130834           648
16215     131020           652
16216     131026           652

                                                  Name  \
766    Asthma emergency department visits due to PM2.5
767    Asthma emergency department visits due to PM2.5
768    Asthma emergency department visits due to PM2.5
769    Asthma emergency department visits due to PM2.5
770    Asthma emergency department visits due to PM2.5
...                                                ...
16189  Asthma emergency department visits due to PM2.5
16190  Asthma emergency department visits due to PM2.5
16212  Asthma emergency department visits due to PM2.5
16215      Cardiac and respiratory deaths due to Ozone
16216      Cardiac and respiratory deaths due to Ozone

                                    Measure          Measure Info  \
766    Estimated annual rate (under age 18)  per 100,000 children
767    Estimated annual rate (under age 18)  per 100,000 children
768    Estimated annual rate (under age 18)  per 100,000 children
769    Estimated annual rate (under age 18)  per 100,000 children
770    Estimated annual rate (under age 18)  per 100,000 children
...                                     ...                  ...
16189       Estimated annual rate (age 18+)    per 100,000 adults
16190       Estimated annual rate (age 18+)    per 100,000 adults
16212  Estimated annual rate (under age 18)  per 100,000 children
16215                Estimated annual rate           per 100,000
16216                Estimated annual rate           per 100,000

      Geo Type Name  Geo Join ID             Geo Place Name Time Period  \
766           UHF42          201                 Greenpoint   2012-2014
767           UHF42          201                 Greenpoint   2015-2017
768           UHF42          101    Kingsbridge - Riverdale   2012-2014
769           UHF42          301         Washington Heights   2012-2014
770           UHF42          501               Port Richmond   2012-2014
...             ...          ...                        ...         ...
16189       Borough            5               Staten Island   2015-2017
16190       Borough            5               Staten Island   2005-2007
16212       Borough            5               Staten Island   2005-2007
16215         UHF42          504  South Beach - Tottenville   2005-2007
16216       Borough            5               Staten Island   2005-2007

      Start_Date  Data Value  Message
766     1/2/2012       43.52      NaN
767     1/1/2015       27.40      NaN
768     1/2/2012       82.30      NaN
769     1/2/2012      111.60      NaN
770     1/2/2012       85.94      NaN
...          ...         ...      ...
16189   1/1/2015       20.80      NaN
```

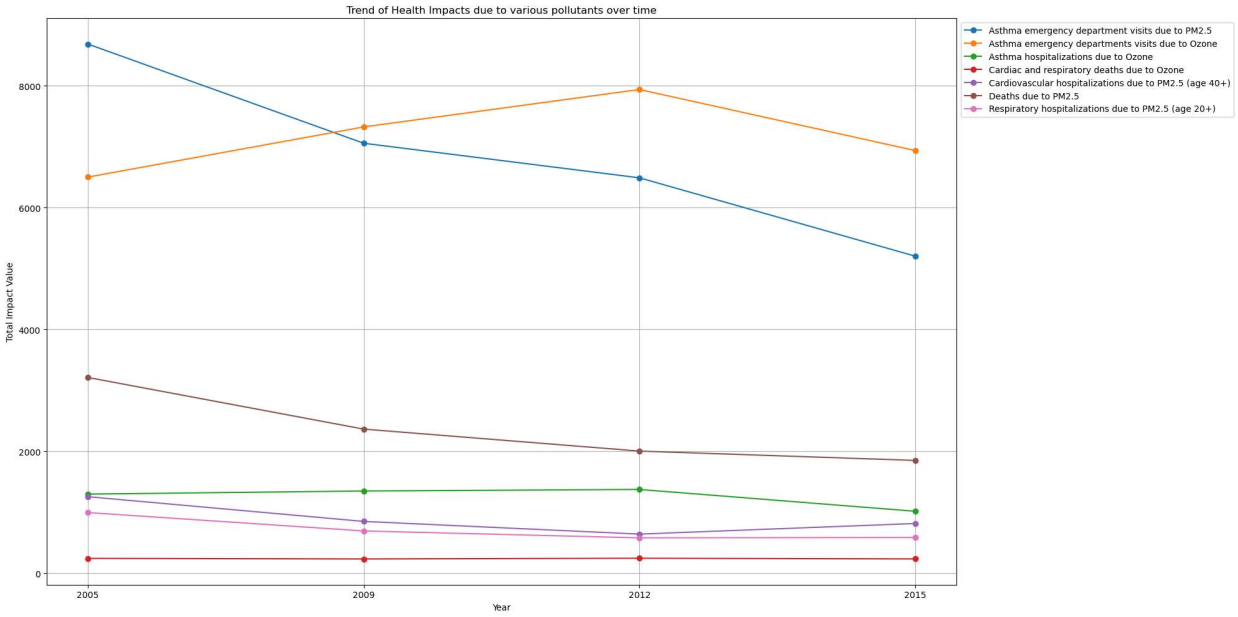
```
16190   1/1/2005       29.60      NaN
16212   1/1/2005       55.30      NaN
16215   1/1/2005        7.50      NaN
16216   1/1/2005        7.80      NaN

[1920 rows x 12 columns]
```

In [3]:
```
# Calculate the frequency of each health impact
name_frequency = filtered_data['Name'].value_counts().reset_index()
name_frequency.columns = ['Name', 'Frequency']

# Define custom colors
colors = plt.cm.Paired(np.linspace(0, 1, len(name_frequency)))

# Pie chart
plt.figure(figsize=(12, 8))
wedges, texts, autotexts = plt.pie(name_frequency['Frequency'], startangle=140, autopc

plt.title('Health Impacts due to various pollutants')
plt.axis('equal')

# Add color boxes on the side
legend_patches = []
for i, autotext in enumerate(name_frequency['Name']):
    # Add color box
    legend_patches.append(mpatches.Patch(color=colors[i], label=autotext))

plt.legend(handles=legend_patches, loc='lower left', bbox_to_anchor=(1, 0))
plt.show()
```



In [4]:
```
# Aggregate data by 'Time Period' and 'Name'
line_data = filtered_data.groupby(['Time Period', 'Name']).sum().reset_index()

# Extract years from 'Time Period' for sorting and plotting
line_data['Year'] = line_data['Time Period'].str.extract('(\d{4})')

# Sort the data by 'Year' and 'Name'
line_data = line_data.sort_values(by=['Year', 'Name'])

# Line plot
plt.figure(figsize=(20, 10))
```

```
for name in line_data['Name'].unique():
    plt_data = line_data[line_data['Name'] == name]
    plt.plot(plt_data['Year'], plt_data['Data Value'], marker='o', label=name)

plt.title('Trend of Health Impacts due to various pollutants over time')
plt.xlabel('Year')
plt.ylabel('Total Impact Value')
plt.grid(True)
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.tight_layout()
plt.show()
```

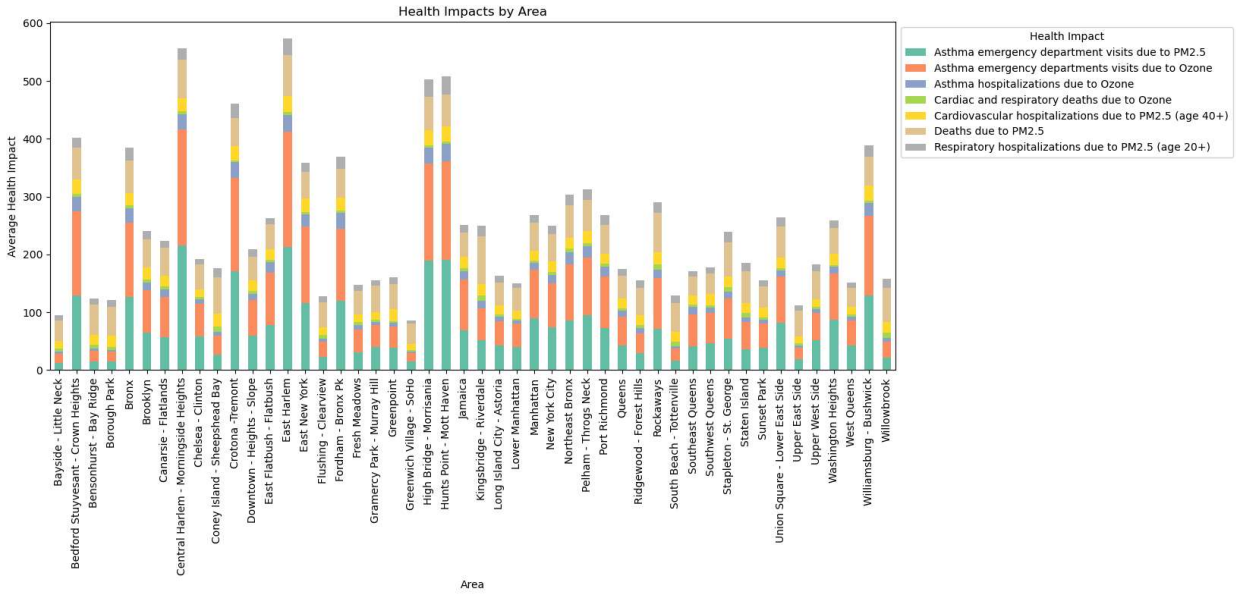


In [7]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame containing the relevant data

# List of keywords
keywords = ["Asthma", "Cardiovascular", "Respiratory", "Death", "Cardiac and respirato

# Filter rows based on keywords in the "Name" column
filtered_data = df[df['Name'].str.contains('|'.join(keywords))]

# Convert 'Data Value' to numeric if needed
filtered_data['Data Value'] = pd.to_numeric(filtered_data['Data Value'], errors='coerc

# Aggregate data by 'Geo Place Name' and 'Name', calculating the average of 'Data Valu
agg_data = filtered_data.groupby(['Geo Place Name', 'Name'])['Data Value'].mean().rese

# Pivot the data to create the stacked bar chart
pivot_data = agg_data.pivot(index='Geo Place Name', columns='Name', values='Data Value

# Stacked Bar plot
plt.figure(figsize=(12, 8))
pivot_data.plot(x='Geo Place Name', kind='bar', stacked=True, colormap='Set2', figsize

plt.title('Health Impacts by Area')
plt.xlabel('Area')
plt.ylabel('Average Health Impact')
```

```
plt.xticks(rotation=90)
plt.legend(title='Health Impact', bbox_to_anchor=(1, 1))
plt.tight_layout()
plt.show()
```

```
C:\Users\ojadh\AppData\Local\Temp\ipykernel_6672\2494543913.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_data['Data Value'] = pd.to_numeric(filtered_data['Data Value'], errors='coerce')
<Figure size 1200x800 with 0 Axes>
```



In [ ]: