AIR QUALITY TEAM E - EXTENSION

Zipcode - Analysis

In [44]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from tabulate import tabulate
from scipy import stats
import geopandas as gpd
from shapely import wkt
import plotly.graph_objects as go
from ipywidgets import widgets, Output
import pandas as pd
from ipywidgets import interact, Dropdown
```

In [45]:
```
# Load the datasets
health_data_2022 = pd.read_csv('../Merged_Data/2022_Health_Zipcode.csv')
health_data_2023 = pd.read_csv('../Merged_Data/2023_Health_Zipcode.csv')
```

In [46]:
```
health_data_2022
```

Out[46]:

	Zipcode	TotalPopulation	ACCESS2_CrudePrev	ARTHRITIS_CrudePrev	BINGE_Crud
0	2111	7383	7.8	16.1	
1	2113	6915	4.2	11.3	
2	2118	26498	8.9	16.6	
3	2124	47783	10.8	21.9	
4	2127	31799	6.4	15.7	
5	2128	40508	16.3	17.5	
6	2130	35401	7.2	17.3	
7	2135	42780	6.1	13.4	
8	2139	36349	5.5	12.1	

9 rows × 35 columns

In [47]:
```
health_data_2023
```

Out[47]:

	Zipcode	TotalPopulation	ACCESS2_CrudePrev	ARTHRITIS_CrudePrev	BINGE_Crud
0	2111	7383	5.5	15.8	
1	2113	6915	2.6	12.3	
2	2118	26498	6.8	17.1	
3	2124	47783	7.7	22.4	
4	2127	31799	4.6	16.7	
5	2128	40508	13.2	17.9	
6	2130	35401	5.1	18.2	
7	2135	42780	4.1	14.0	
8	2139	36349	3.6	13.2	

9 rows × 42 columns

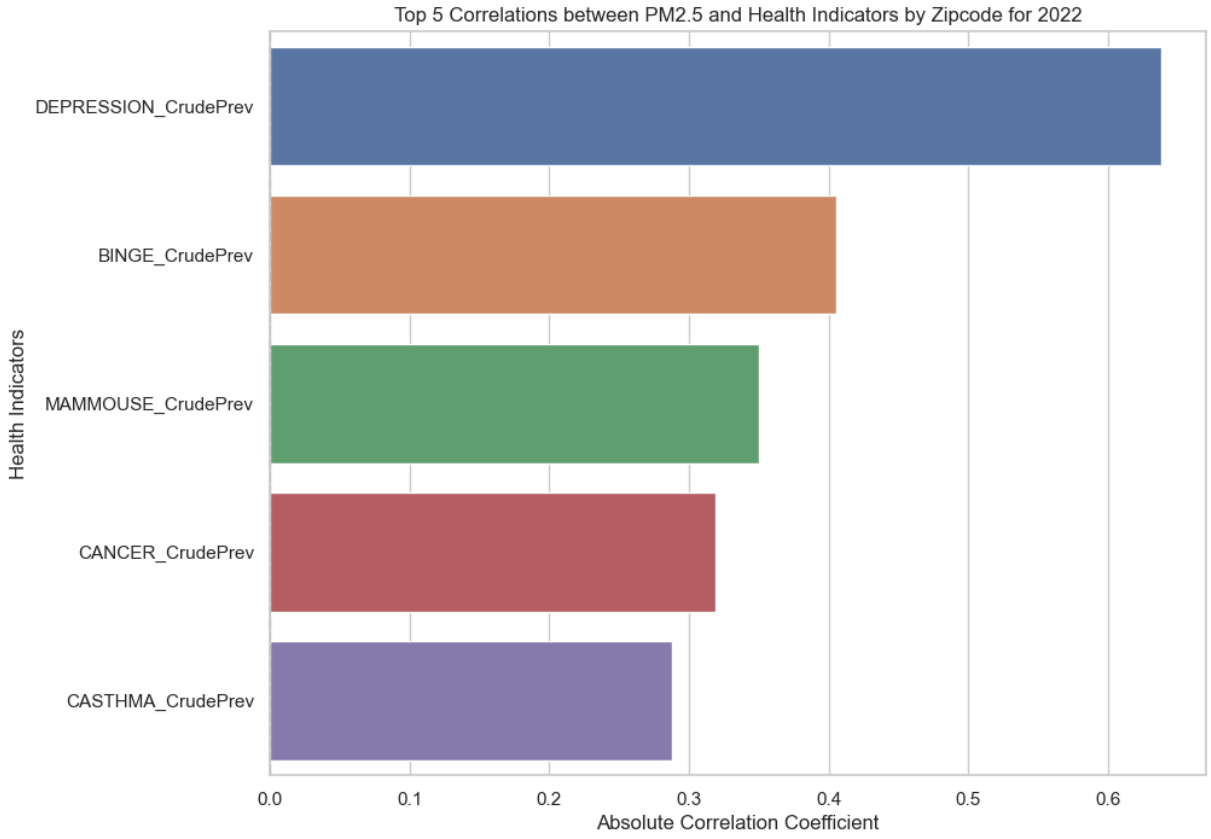
In [48]:

```
pm25_2022 = health_data_2022[['Zipcode', 'PM2.5']]
pm25_2023 = health_data_2023[['Zipcode', 'PM2.5']]

pm25_comparison = pd.merge(pm25_2022,
                           pm25_2023,
                           on='Zipcode',
                           suffixes=('_2022', '_2023'))

pm25_comparison['PM2.5_Change'] = pm25_comparison[
    'PM2.5_2023'] - pm25_comparison['PM2.5_2022']

table = tabulate(pm25_comparison,
                 headers='keys',
                 tablefmt='fancy_grid',
                 showindex=False)
print(table)
```

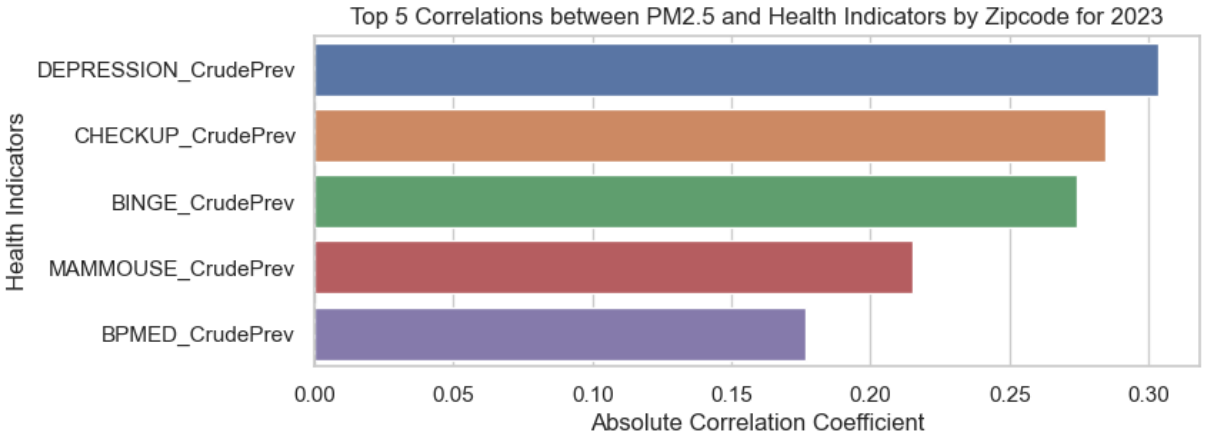
Zipcode	PM2.5_2022	PM2.5_2023	PM2.5_Change
2111	6.73753	9.4645	2.72697
2113	7.9084	9.96361	2.05521
2118	8.30109	10.5777	2.27656
2124	7.21404	9.10869	1.89465
2127	7.75032	10.5325	2.78215
2128	7.08768	9.64689	2.55921
2130	7.12675	8.7106	1.58385
2135	6.94934	8.87072	1.92138
2139	5.53174	9.11105	3.57931

The comparison of air quality between 2022 and 2023, as indicated by PM2.5 levels across common zip codes, reveals a notable trend. For the zip codes analyzed, there is a consistent increase in PM2.5 levels from 2022 to 2023. This upward shift suggests a deterioration in air quality over the year. The changes vary across different areas, with some zip codes experiencing more significant increases than others.

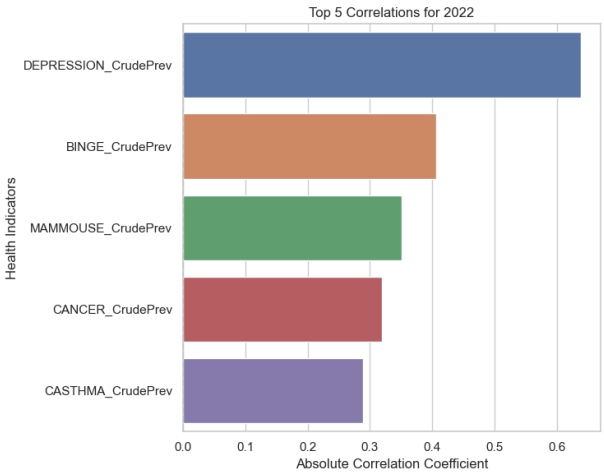
In [53]:
```
import matplotlib.pyplot as plt
import seaborn as sns
crude_columns_2022 = [
    col for col in health_data_2022.columns if 'Crude' in col
]
crude_columns_2022.append('PM2.5')
correlation_matrix_2022 = health_data_2022[crude_columns_2022].corr()

correlations_2022 = correlation_matrix_2022['PM2.5'].drop(
    'PM2.5')  # excluding the PM2.5 self-correlation


# Getting the top 5 correlations by absolute value
top_5_correlations = correlations_2022.abs().nlargest(5)

# Creating a bar plot to visualize the top 5 correlations by zipcode
plt.figure(figsize=(10, 8))
sns.barplot(x=top_5_correlations.values, y=top_5_correlations.index)
plt.title('Top 5 Correlations between PM2.5 and Health Indicators by Zipcode
plt.xlabel('Absolute Correlation Coefficient')
plt.ylabel('Health Indicators')
plt.axvline(x=0, color='gray', linestyle='--')
plt.show()
```

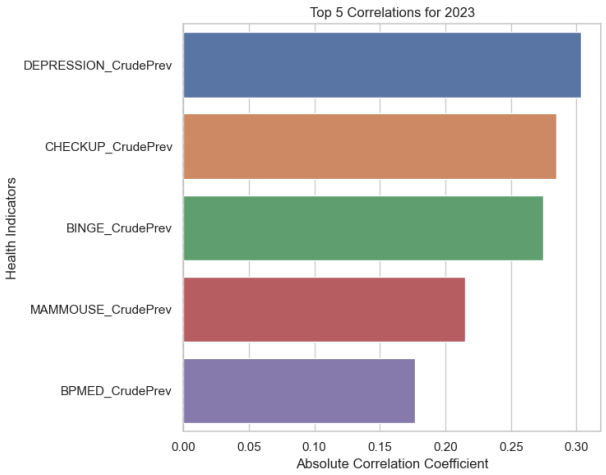


In [62]:
```
crude_columns_2023 = [
    col for col in health_data_2023.columns if 'Crude' in col
]
crude_columns_2023.append('PM2.5')
correlation_matrix_2023 = health_data_2023[crude_columns_2023].corr()

correlations_2023 = correlation_matrix_2023['PM2.5'].drop(
    'PM2.5')  # excluding the PM2.5 self-correlation

# Getting the top 5 correlations by absolute value
top_5_correlations = correlations_2023.abs().nlargest(5)

# Creating a bar plot to visualize the top 5 correlations by zipcode
plt.figure(figsize=(8, 3))
sns.barplot(x=top_5_correlations.values, y=top_5_correlations.index)
plt.title('Top 5 Correlations between PM2.5 and Health Indicators by Zipcode
plt.xlabel('Absolute Correlation Coefficient')
plt.ylabel('Health Indicators')
plt.axvline(x=0, color='gray', linestyle='--')
plt.show()
```

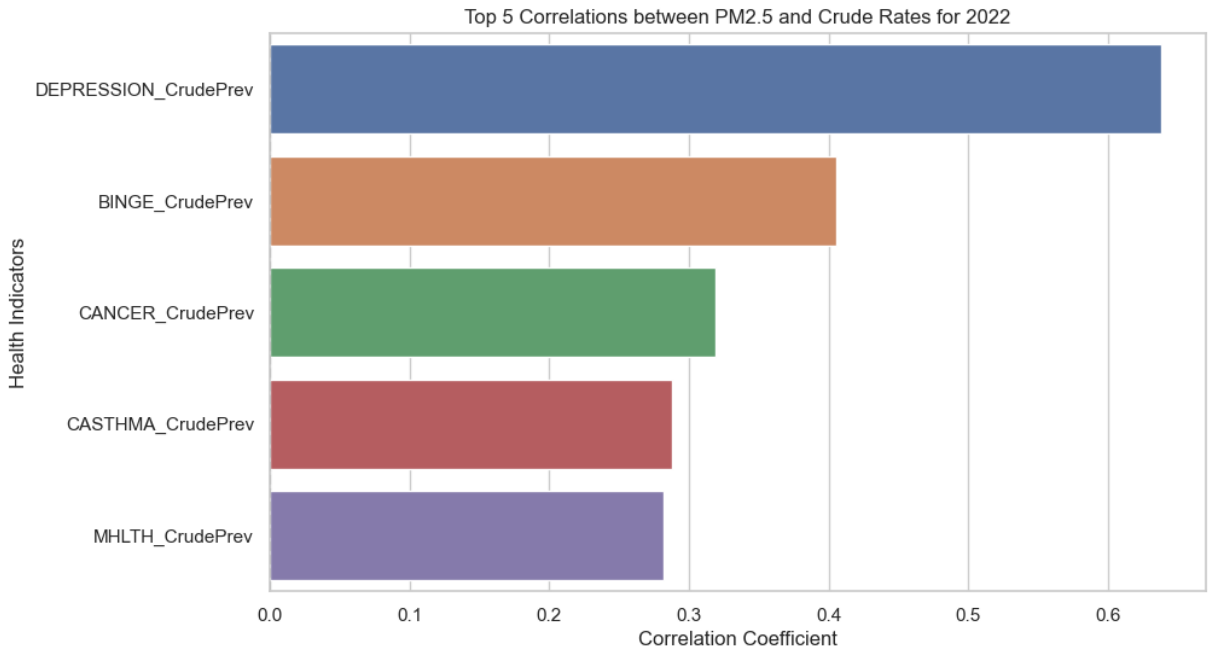


In [63]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming you have defined correlations_2022 and correlations_2023

# Getting the top 5 correlations by absolute value for 2022
top_5_correlations_2022 = correlations_2022.abs().nlargest(5)

# Getting the top 5 correlations by absolute value for 2023
top_5_correlations_2023 = correlations_2023.abs().nlargest(5)

# Create subplots to display side-by-side bar plots for 2022 and 2023
fig, axes = plt.subplots(1, 2, figsize=(15, 6))

# Plotting top 5 correlations for 2022
sns.barplot(ax=axes[0], x=top_5_correlations_2022.values, y=top_5_correlatic
axes[0].set_title('Top 5 Correlations for 2022')
axes[0].set_xlabel('Absolute Correlation Coefficient')
axes[0].set_ylabel('Health Indicators')
axes[0].axvline(x=0, color='gray', linestyle='--')

# Plotting top 5 correlations for 2023
sns.barplot(ax=axes[1], x=top_5_correlations_2023.values, y=top_5_correlatic
axes[1].set_title('Top 5 Correlations for 2023')
axes[1].set_xlabel('Absolute Correlation Coefficient')
axes[1].set_ylabel('Health Indicators')
axes[1].axvline(x=0, color='gray', linestyle='--')

plt.tight_layout()
plt.show()
```

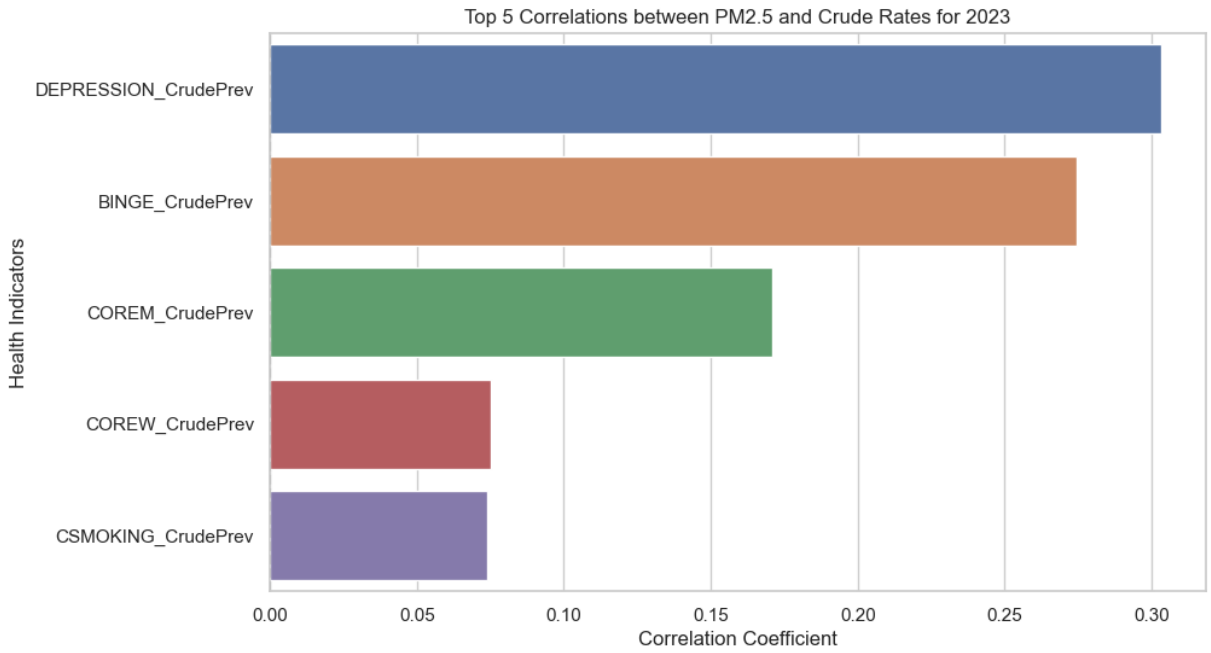




In [28]:
```
import matplotlib.pyplot as plt
import seaborn as sns

crude_columns_2022 = [
    col for col in health_data_2022.columns if 'Crude' in col
]
crude_columns_2022.append('PM2.5')
correlation_matrix_2022 = health_data_2022[crude_columns_2022].corr()

correlations_2022 = correlation_matrix_2022['PM2.5'].drop('PM2.5')

# Getting the top 3 highest coefficients
top_3_correlations = correlations_2022.nlargest(5)

# Creating a bar plot to visualize the top 3 correlations
plt.figure(figsize=(10, 6))
sns.barplot(x=top_3_correlations.values, y=top_3_correlations.index)
plt.title('Top 5 Correlations between PM2.5 and Crude Rates for 2022')
plt.xlabel('Correlation Coefficient')
plt.ylabel('Health Indicators')
plt.axvline(x=0, color='gray', linestyle='--')
plt.show()
```



In [27]:
```
import matplotlib.pyplot as plt
import seaborn as sns

crude_columns_2023 = [
    col for col in health_data_2023.columns if 'Crude' in col
]
crude_columns_2023.append('PM2.5')
correlation_matrix_2023 = health_data_2023[crude_columns_2023].corr()

correlations_2023 = correlation_matrix_2023['PM2.5'].drop('PM2.5')

# Getting the top 3 highest coefficients
top_3_correlations = correlations_2023.nlargest(5)

# Creating a bar plot to visualize the top 3 correlations
plt.figure(figsize=(10, 6))
sns.barplot(x=top_3_correlations.values, y=top_3_correlations.index)
plt.title('Top 5 Correlations between PM2.5 and Crude Rates for 2023')
plt.xlabel('Correlation Coefficient')
plt.ylabel('Health Indicators')
plt.axvline(x=0, color='gray', linestyle='--')
plt.show()
```



In [36]:
```
common_columns = set(health_data_2022.columns).intersection(set(health_data_

# Filtering out non-disease related columns
non_disease_columns = {'Zipcode', 'TotalPopulation', 'Geolocation', 'Year',
disease_columns = common_columns - non_disease_columns

# Merging the two datasets on Zipcode to calculate the changes
merged_data = pd.merge(health_data_2022[['Zipcode'] + list(disease_columns)
                       health_data_2023[['Zipcode'] + list(disease_columns)
                       on='Zipcode', suffixes=('_2022', '_2023'))

# Calculating the change for each disease and PM2.5
for disease in disease_columns:
    merged_data[f'{disease}_Change'] = merged_data[f'{disease}_2023'] - merg

# Calculating PM2.5 change
merged_data['PM2.5_Change'] = merged_data['PM2.5_2023'] - merged_data['PM2.5

# Selecting only the change columns along with the Zipcode for the final tab
change_columns = [col for col in merged_data.columns if 'Change' in col]
final_table = merged_data[['Zipcode'] + change_columns]

final_table
```

Out[36]:

	Zipcode	DIABETES_CrudePrev_Change	BINGE_CrudePrev_Change	STROKE_CrudeP
0	2111	0.9	0.1	
1	2113	0.1	0.1	
2	2118	0.5	-0.2	
3	2124	0.3	0.2	
4	2127	0.2	0.0	
5	2128	0.2	-0.1	
6	2130	0.2	0.2	
7	2135	0.3	-0.1	
8	2139	0.1	2.0	

9 rows × 33 columns

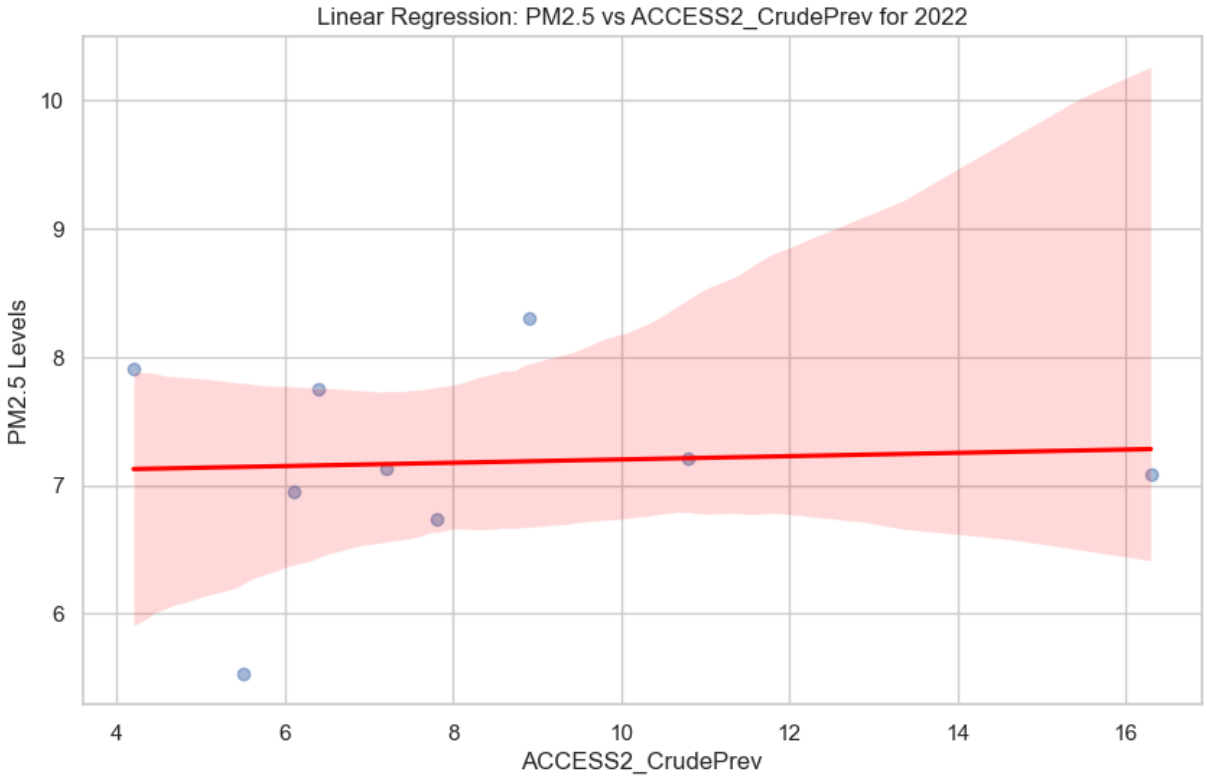
In [38]:
```
# Assuming the 'final_table' DataFrame structure looks like the provided dat

selected_columns = ['DIABETES_CrudePrev_Change', 'BINGE_CrudePrev_Change', '
                    'COREM_CrudePrev_Change', 'CHECKUP_CrudePrev_Change', 'C
                    'CSMOKING_CrudePrev_Change', 'CHD_CrudePrev_Change', 'CH
                    # Add other columns related to the top 5 diseases here
                    'PM2.5_Change']

final_table = final_table[selected_columns]
final_table
```

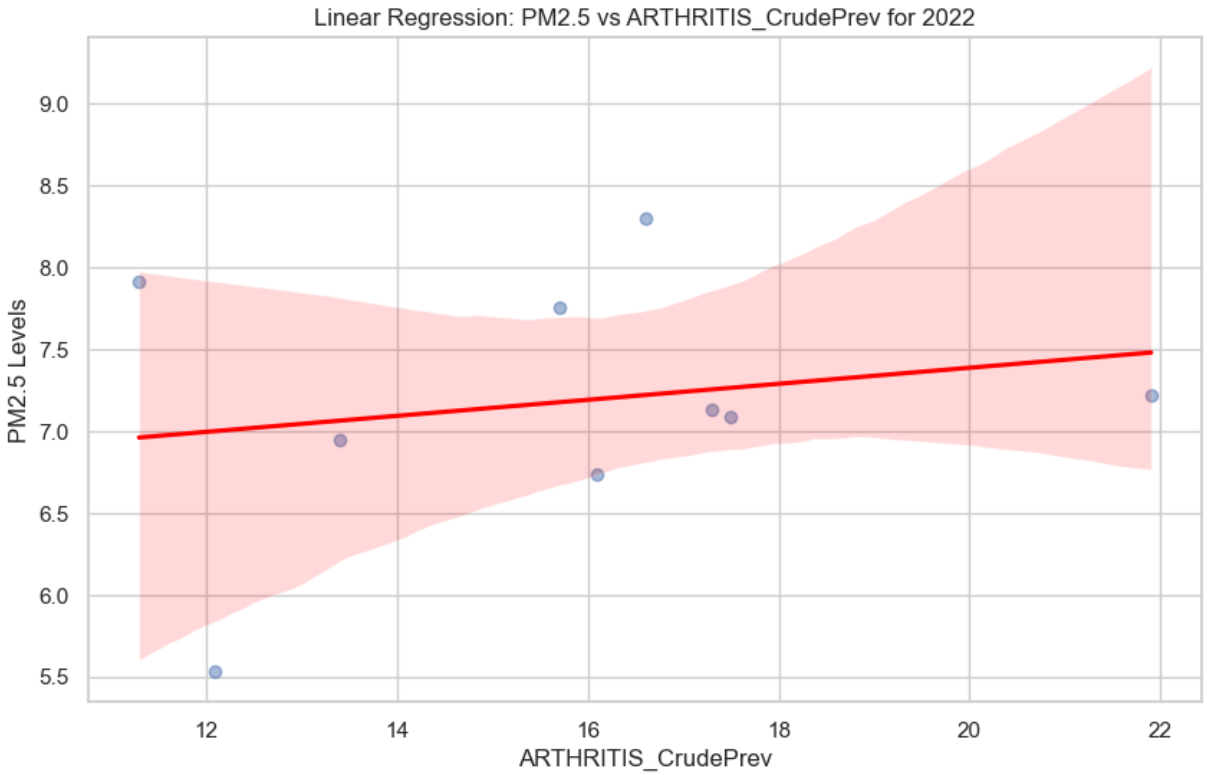
Out[38]:

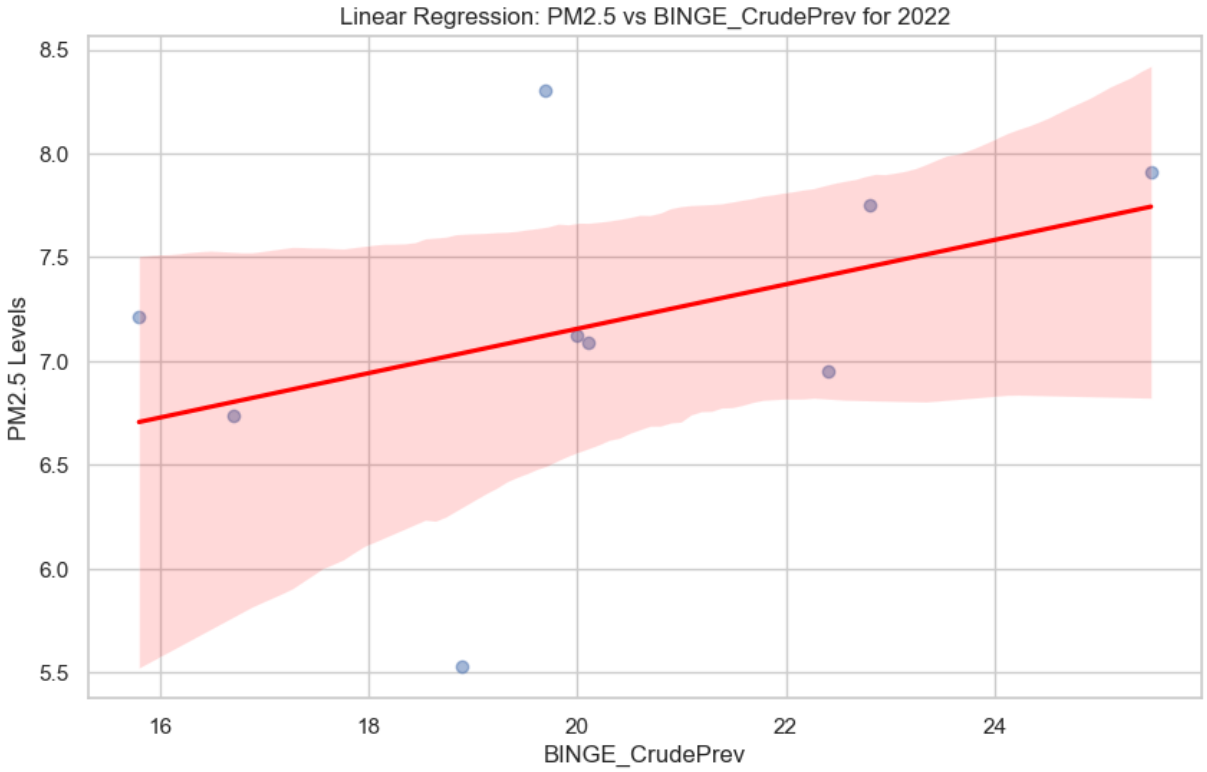
	DIABETES_CrudePrev_Change	BINGE_CrudePrev_Change	STROKE_CrudePrev_Chang
0	0.9	0.1	0
1	0.1	0.1	0
2	0.5	-0.2	0
3	0.3	0.2	0
4	0.2	0.0	0
5	0.2	-0.1	0
6	0.2	0.2	0
7	0.3	-0.1	0
8	0.1	2.0	0

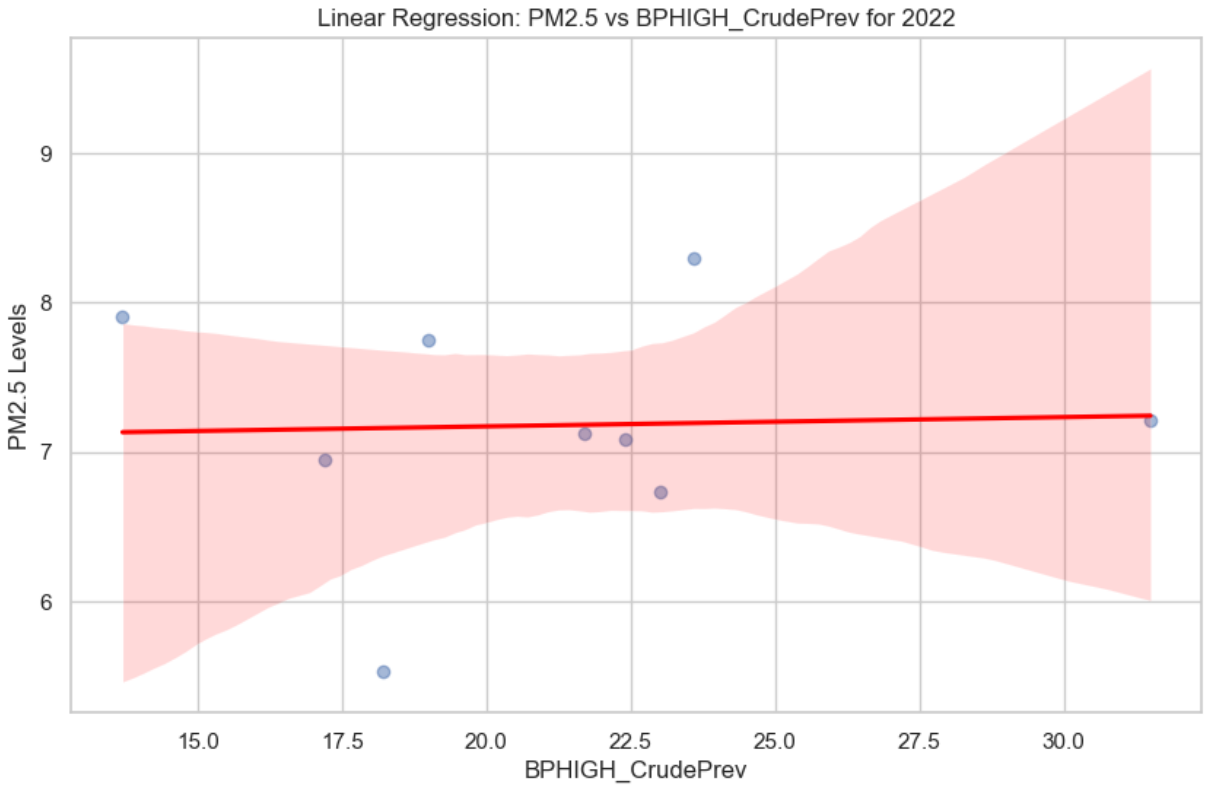
In [39]:
```
def plot_linear_regression(data, crude_columns, year):
    for column in crude_columns:
        if column != 'PM2.5':
            # Performing linear regression
            slope, intercept, r_value, p_value, std_err = stats.linregress(d
```

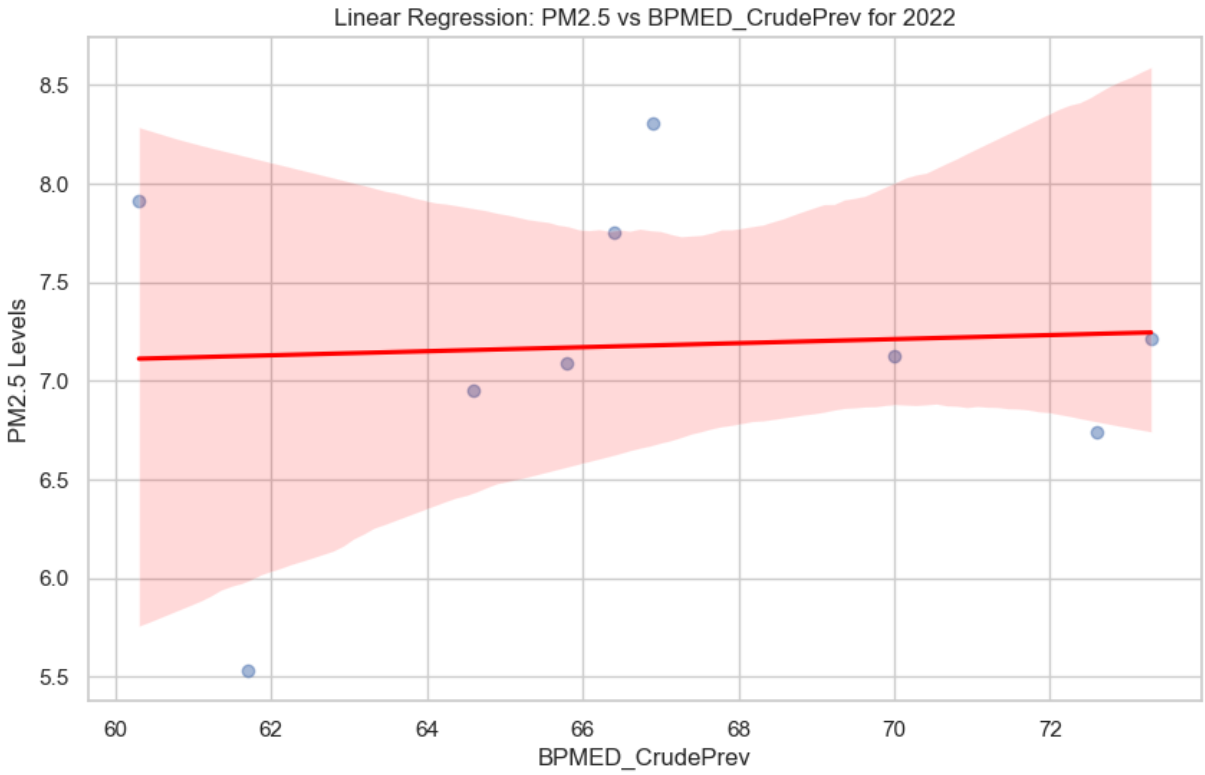
```
            # Plotting
            plt.figure(figsize=(10, 6))
            sns.regplot(x=column, y='PM2.5', data=data, scatter_kws={'alpha'
            plt.title(f'Linear Regression: PM2.5 vs {column} for {year}')
            plt.xlabel(column)
            plt.ylabel('PM2.5 Levels')
            plt.show()

# Extracting crude columns from 2022 dataset
crude_columns_2022 = [col for col in health_data_2022.columns if 'Crude' in

# Making linear regression plots for each crude rate against PM2.5 level for
plot_linear_regression(health_data_2022, crude_columns_2022, '2022')
```

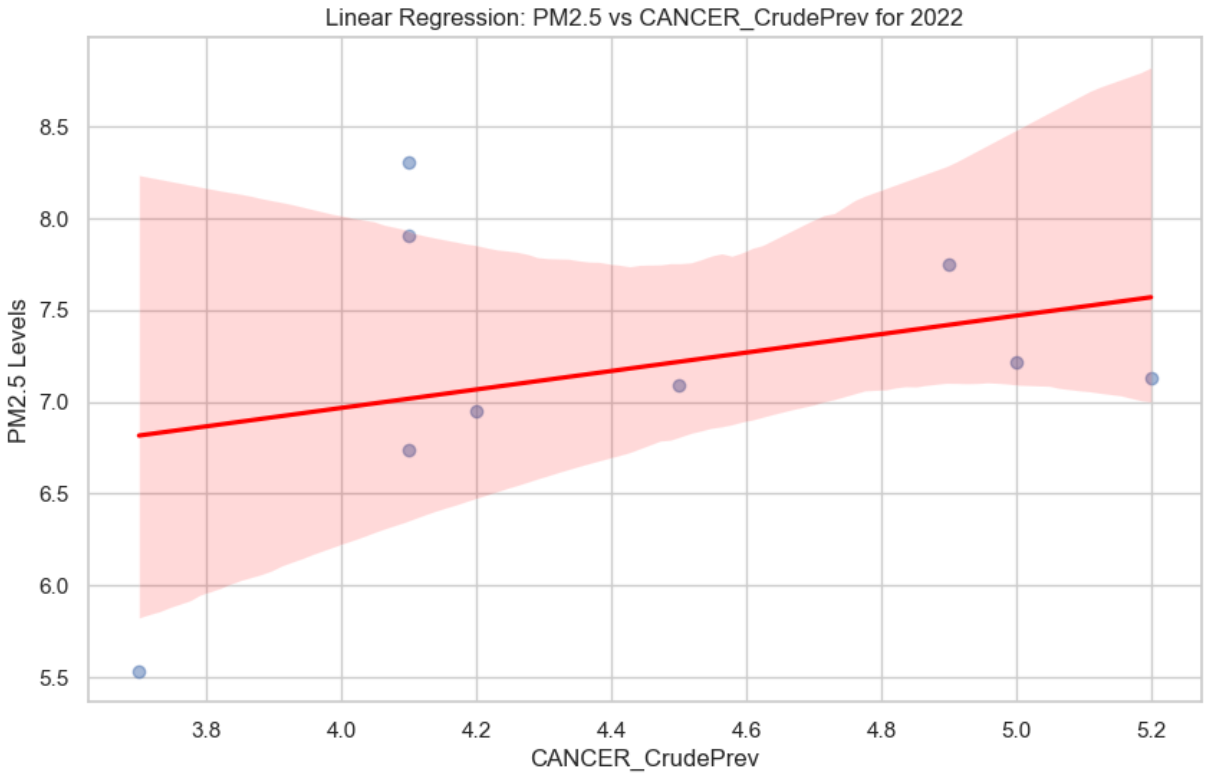


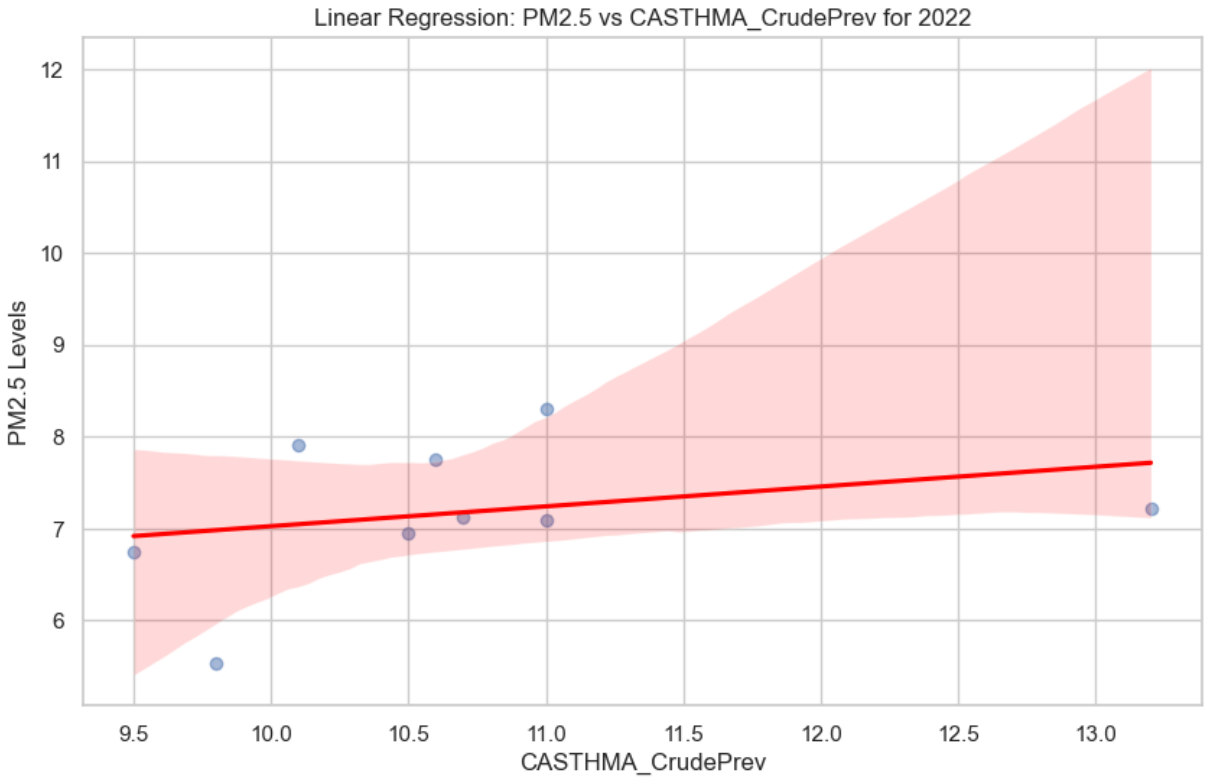

Linear Regression: PM2.5 vs ARTHRITIS_CrudePrev for 2022
PM2.5 Levels
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
12
14
16
18
20
22
ARTHRITIS_CrudePrev

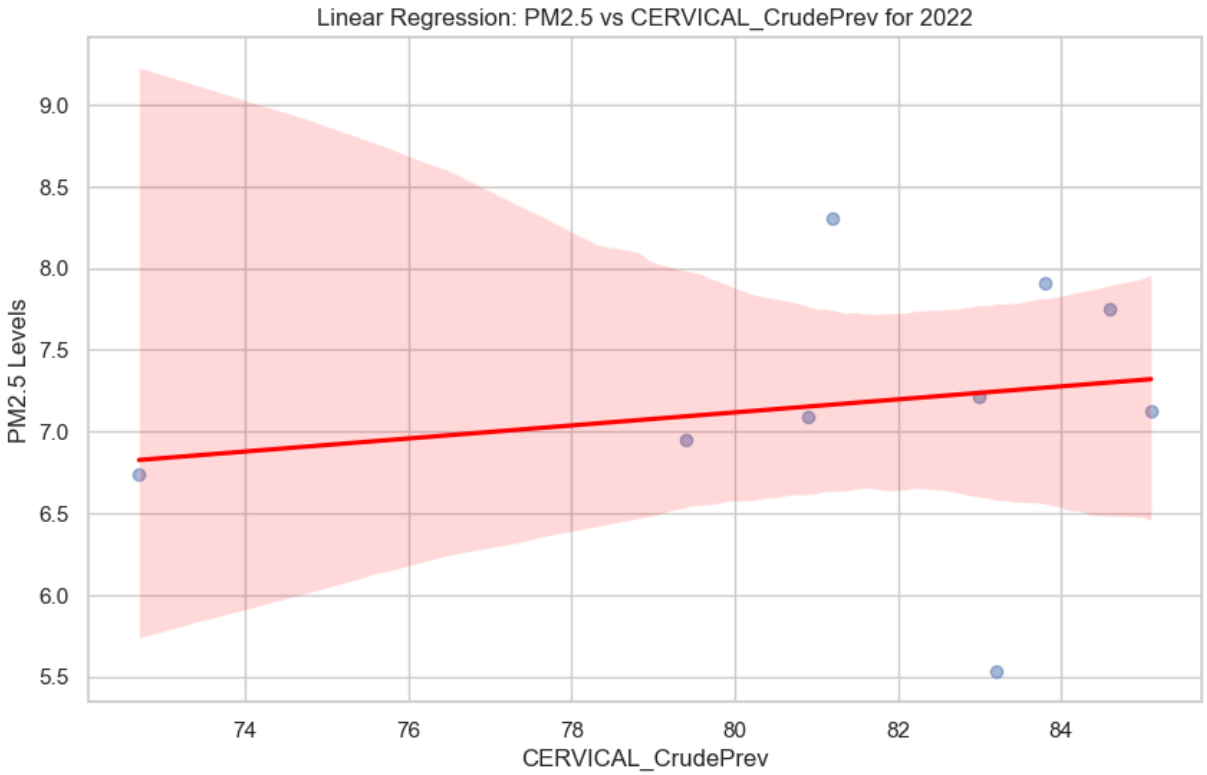

Linear Regression: PM2.5 vs BINGE_CrudePrev for 2022
PM2.5 Levels
8.5
8.0
7.5
7.0
6.5
6.0
5.5
16
18
20
22
24
BINGE_CrudePrev

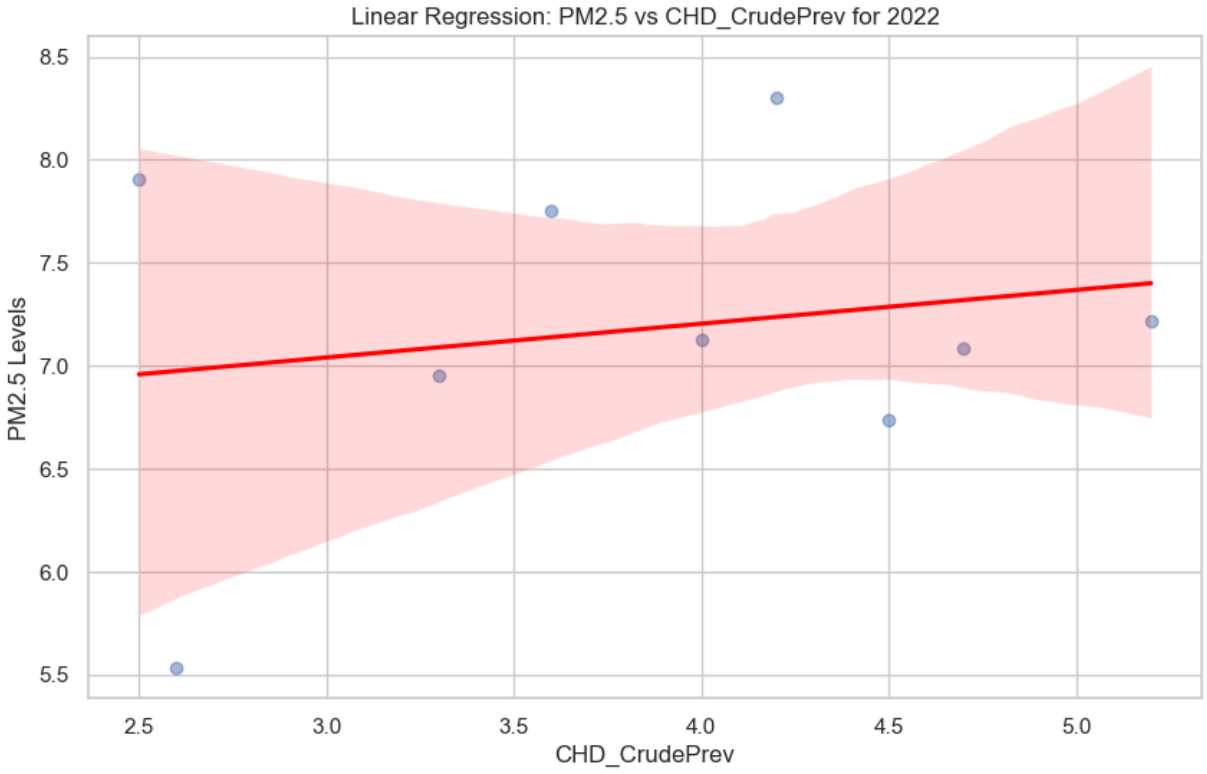

Linear Regression: PM2.5 vs BPHIGH_CrudePrev for 2022
PM2.5 Levels
9
8
7
6
15.0
17.5
20.0
22.5
25.0
27.5
30.0
BPHIGH_CrudePrev

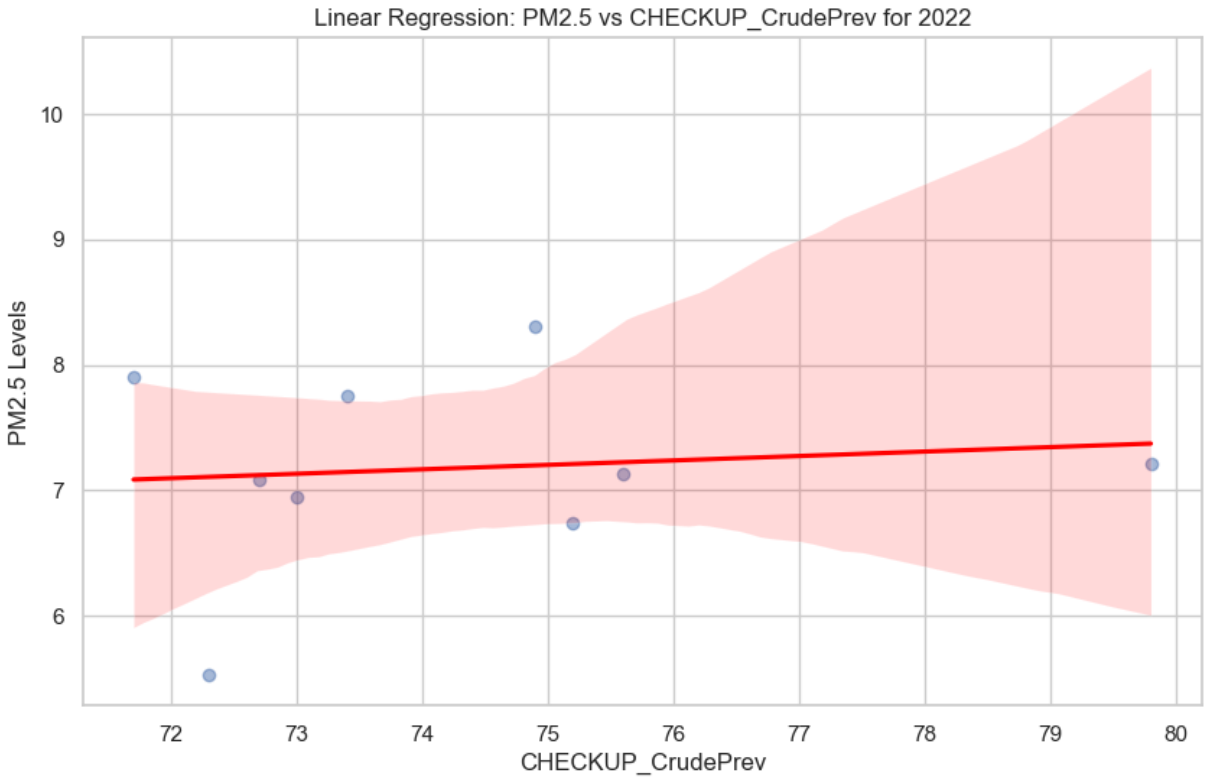

Linear Regression: PM2.5 vs BPMED_CrudePrev for 2022
PM2.5 Levels
8.5
8.0
7.5
7.0
6.5
6.0
5.5
60
62
64
66
68
70
72
BPMED_CrudePrev

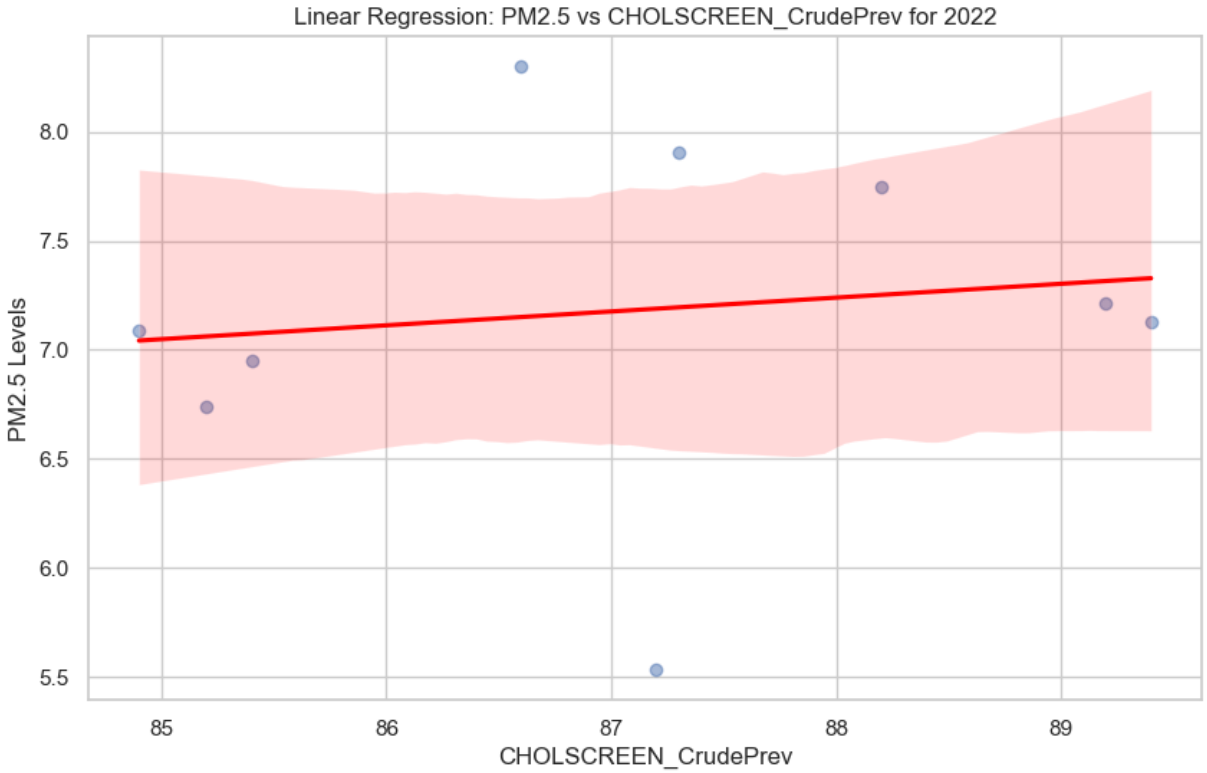

Linear Regression: PM2.5 vs CANCER_CrudePrev for 2022
PM2.5 Levels
8.5
8.0
7.5
7.0
6.5
6.0
5.5
3.8
4.0
4.2
4.4
4.6
4.8
5.0
5.2
CANCER_CrudePrev

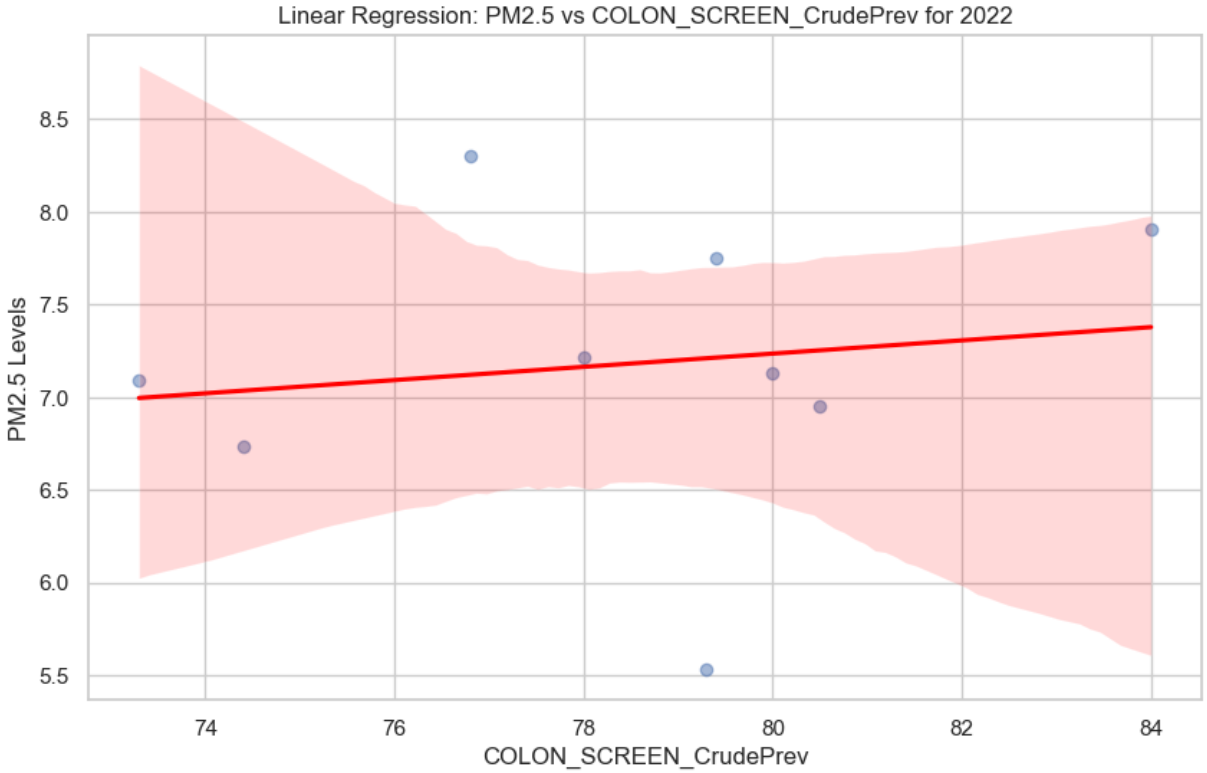

Linear Regression: PM2.5 vs CASTHMA_CrudePrev for 2022
PM2.5 Levels
12
11
10
9
8
7
6
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
CASTHMA_CrudePrev

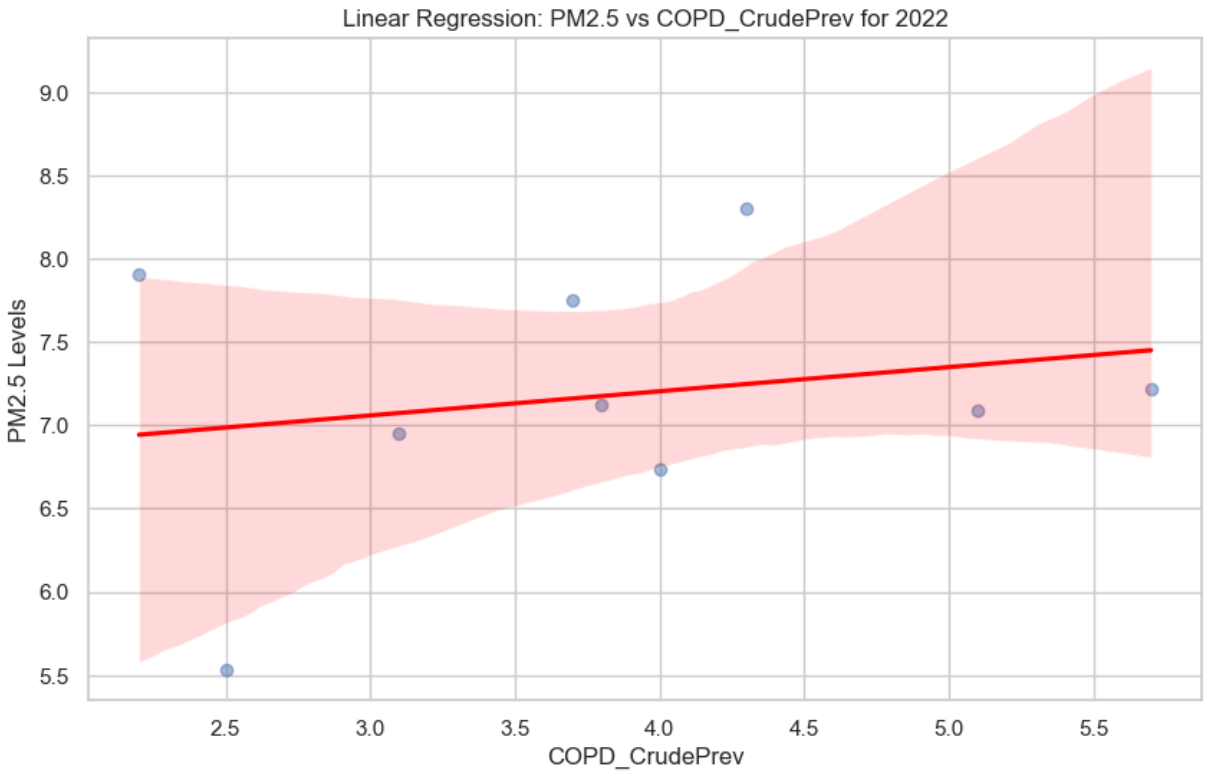

Linear Regression: PM2.5 vs CERVICAL_CrudePrev for 2022
PM2.5 Levels
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
74
76
78
80
82
84
CERVICAL_CrudePrev

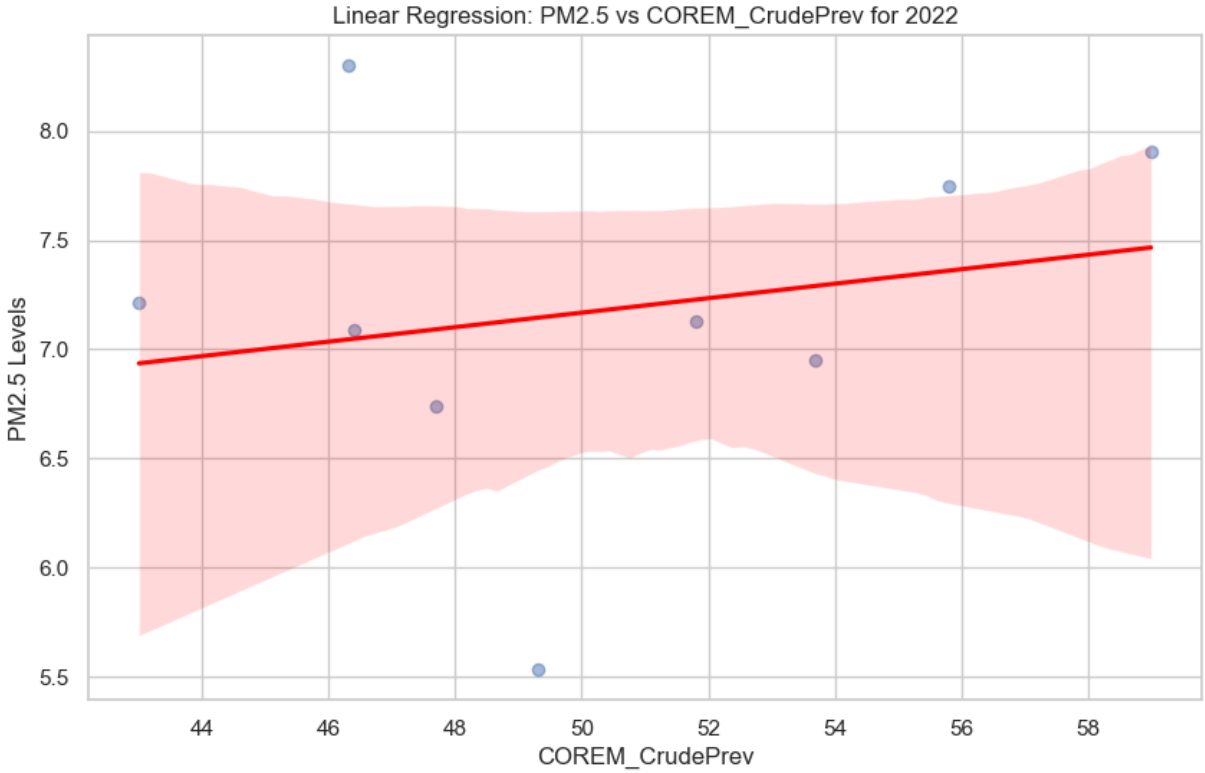

Linear Regression: PM2.5 vs CHD_CrudePrev for 2022
PM2.5 Levels
8.5
8.0
7.5
7.0
6.5
6.0
5.5
2.5
3.0
3.5
4.0
4.5
5.0
CHD_CrudePrev

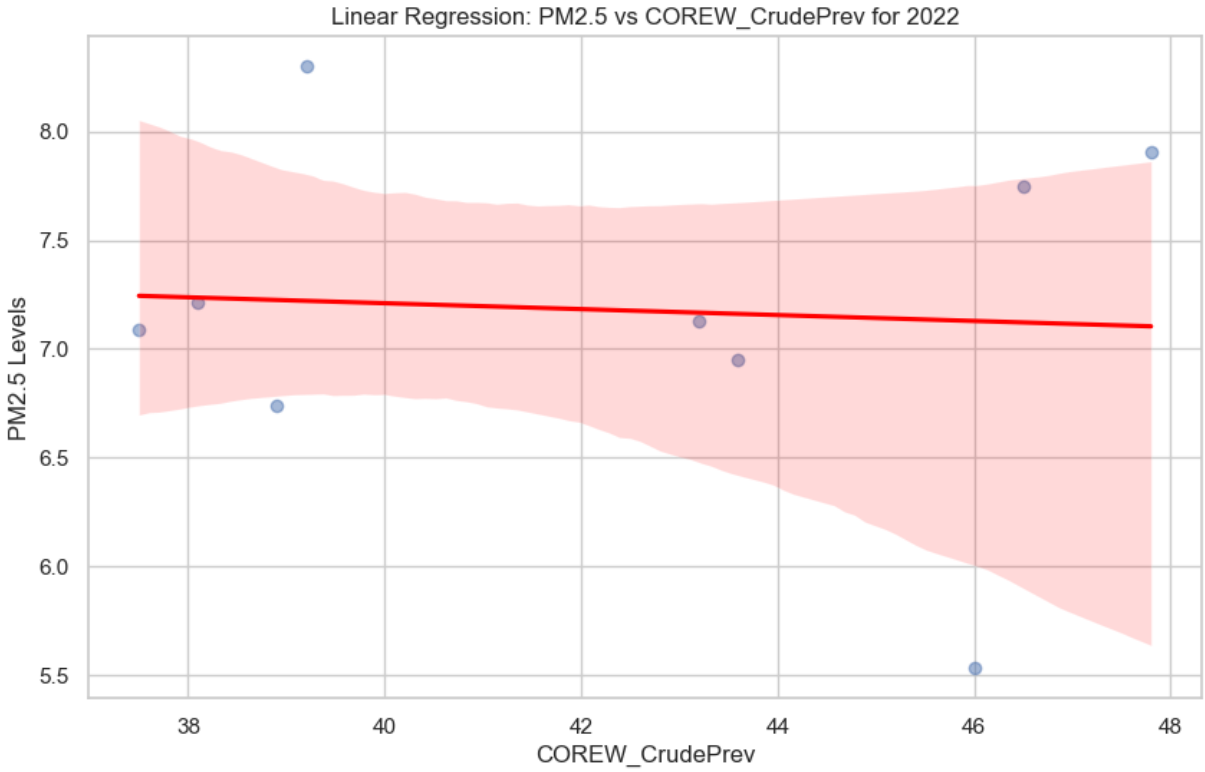

Linear Regression: PM2.5 vs CHECKUP_CrudePrev for 2022
PM2.5 Levels
CHECKUP_CrudePrev
10
9
8
7
6
72
73
74
75
76
77
78
79
80

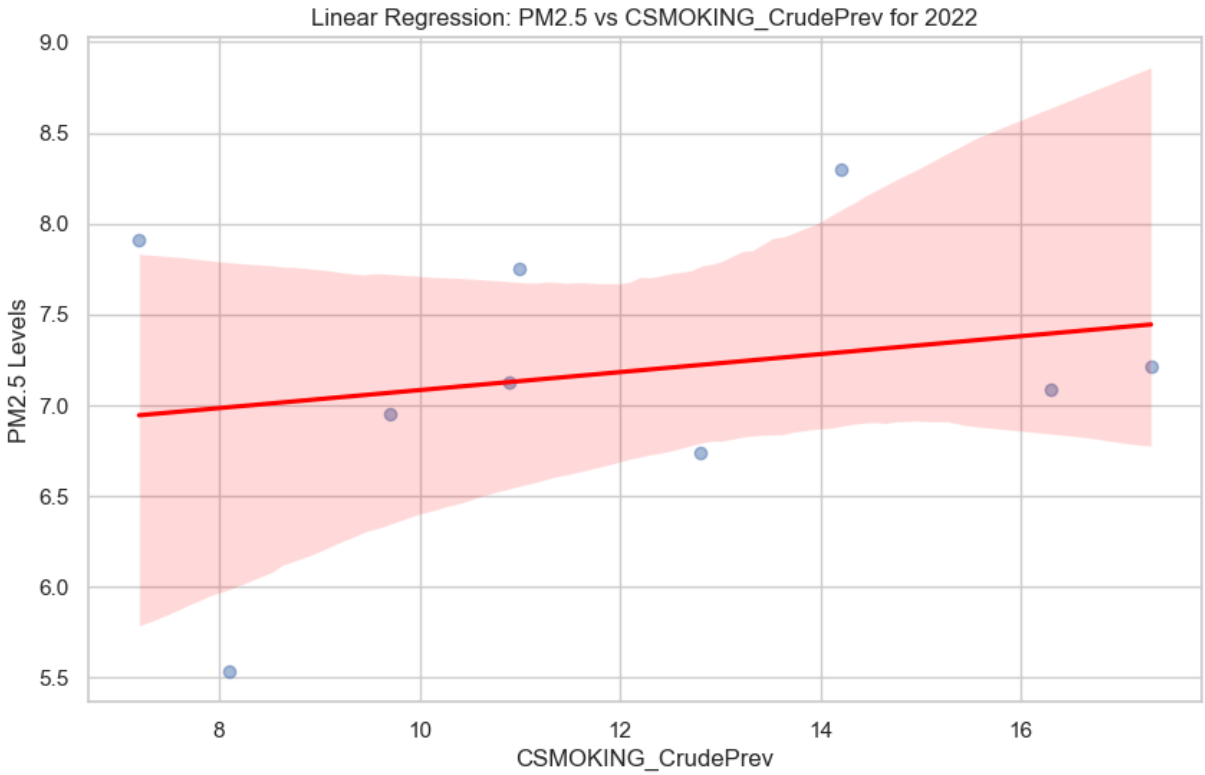

Linear Regression: PM2.5 vs CHOLSCREEN_CrudePrev for 2022
PM2.5 Levels
CHOLSCREEN_CrudePrev
8.0
7.5
7.0
6.5
6.0
5.5
85
86
87
88
89

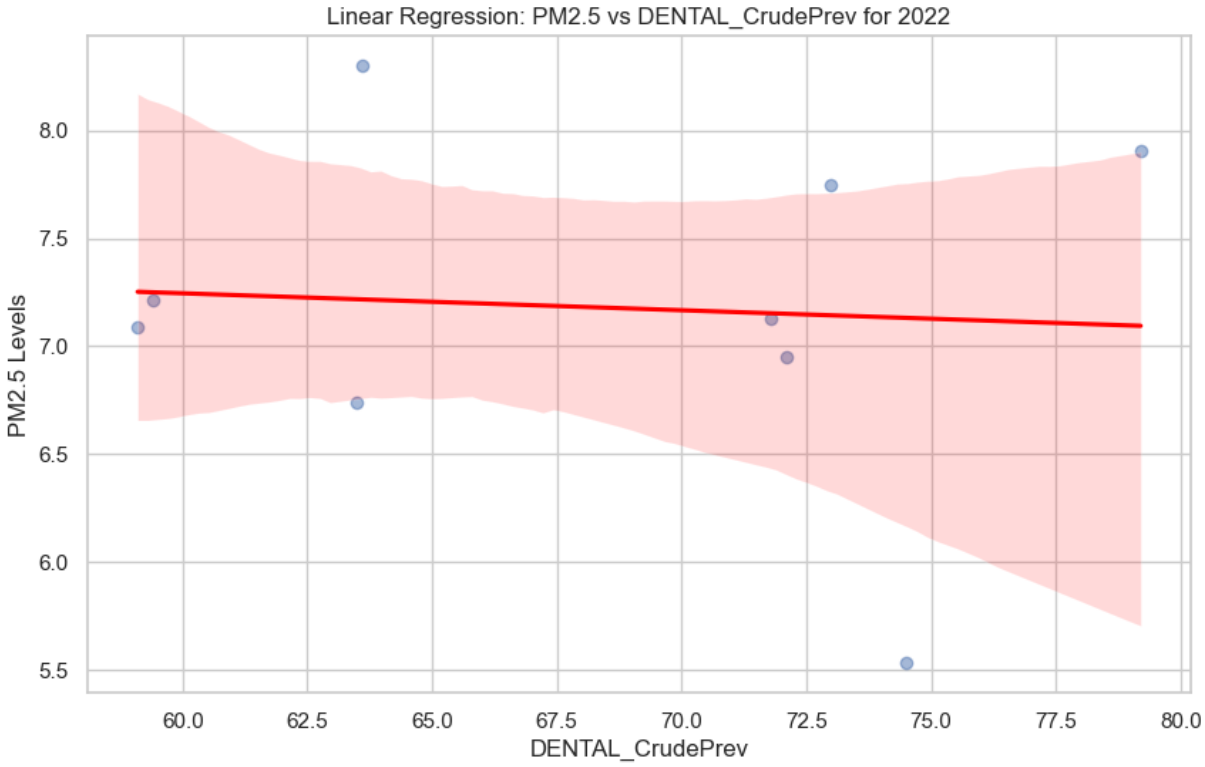

Linear Regression: PM2.5 vs COLON_SCREEN_CrudePrev for 2022
PM2.5 Levels
8.5
8.0
7.5
7.0
6.5
6.0
5.5
74
76
78
80
82
84
COLON_SCREEN_CrudePrev

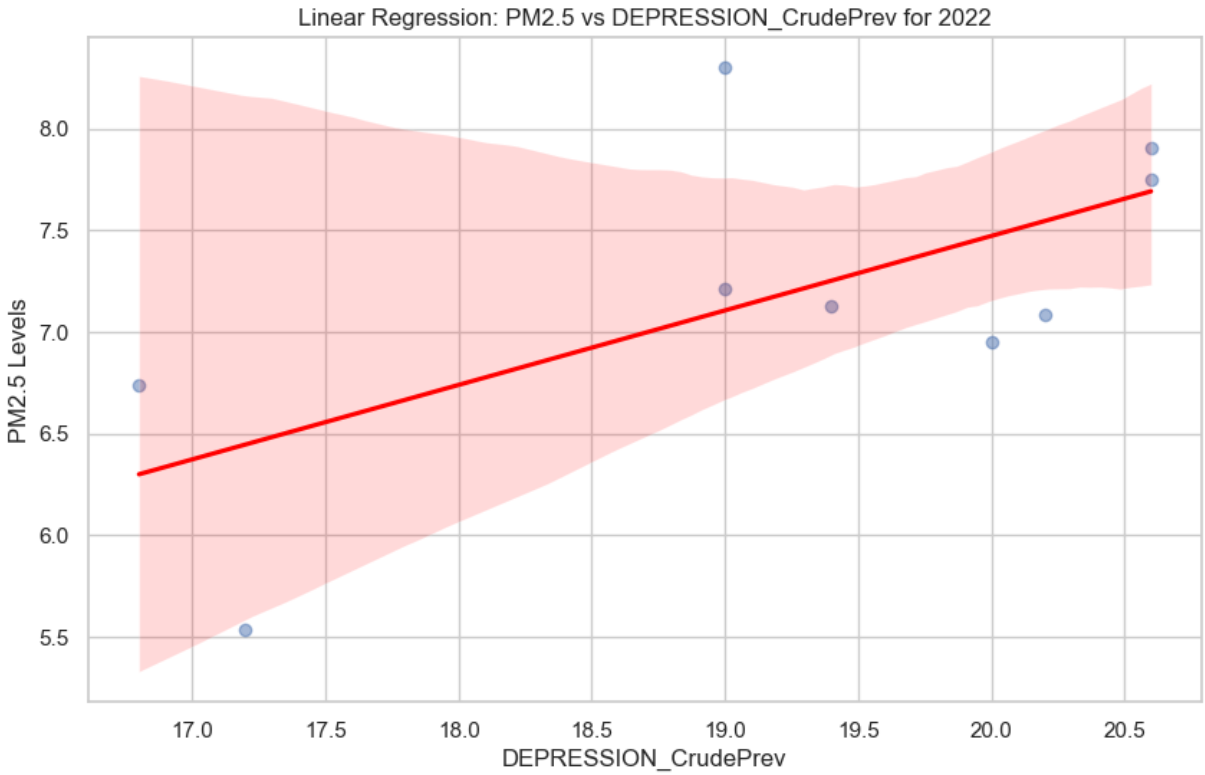

Linear Regression: PM2.5 vs COPD_CrudePrev for 2022
PM2.5 Levels
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
2.5
3.0
3.5
4.0
4.5
5.0
5.5
COPD_CrudePrev

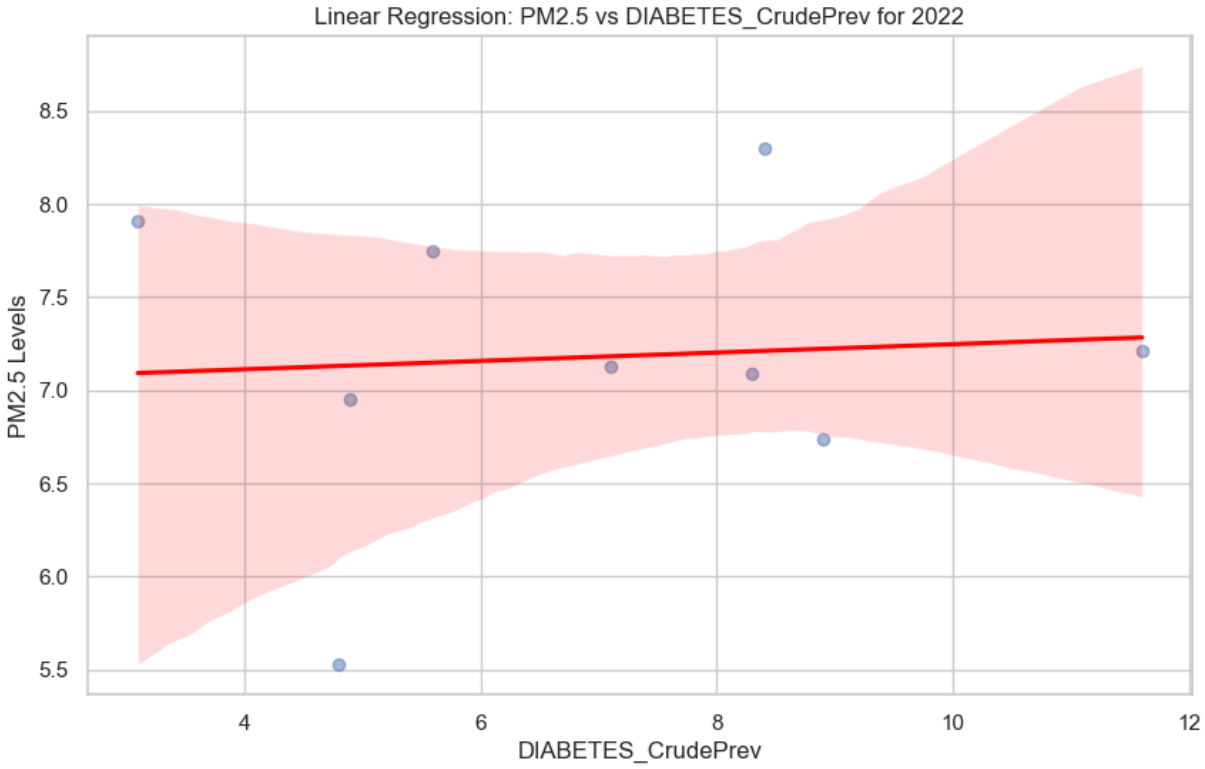

Linear Regression: PM2.5 vs COREM_CrudePrev for 2022
PM2.5 Levels
8.0
7.5
7.0
6.5
6.0
5.5
44
46
48
50
52
54
56
58
COREM_CrudePrev

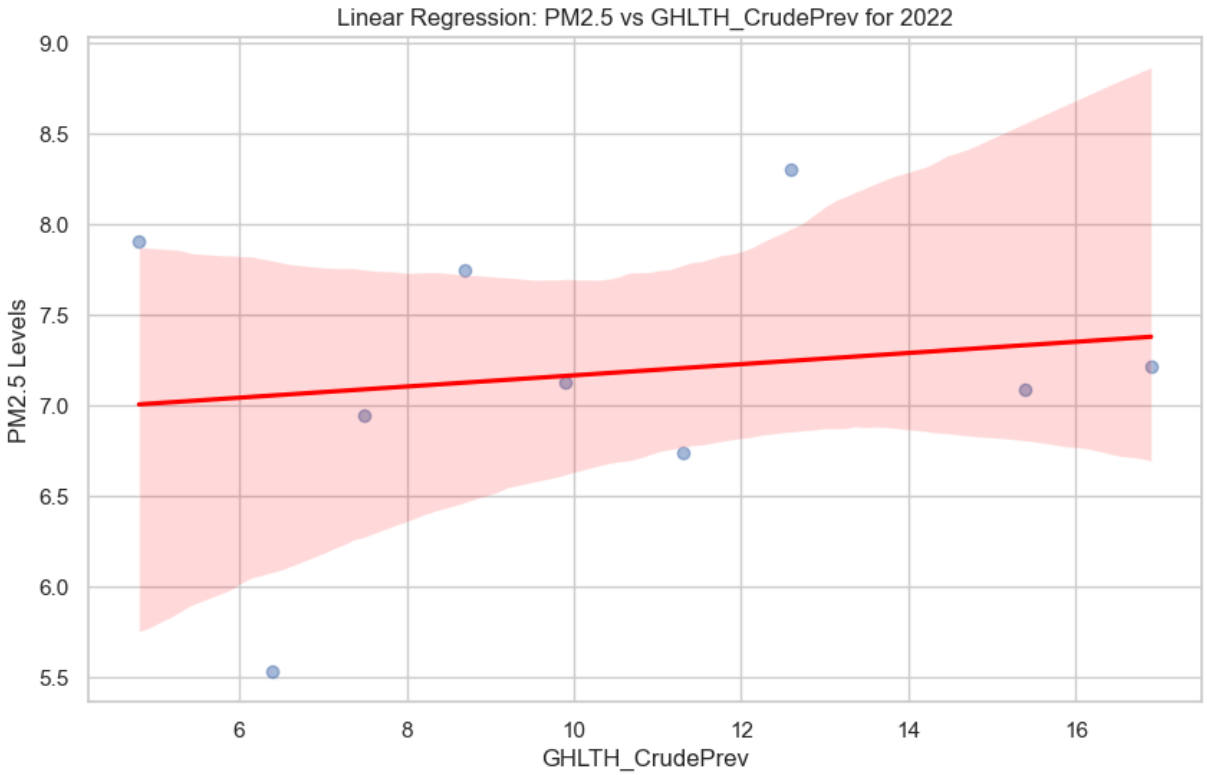

Linear Regression: PM2.5 vs COREW_CrudePrev for 2022
PM2.5 Levels
8.0
7.5
7.0
6.5
6.0
5.5
38
40
42
44
46
48
COREW_CrudePrev

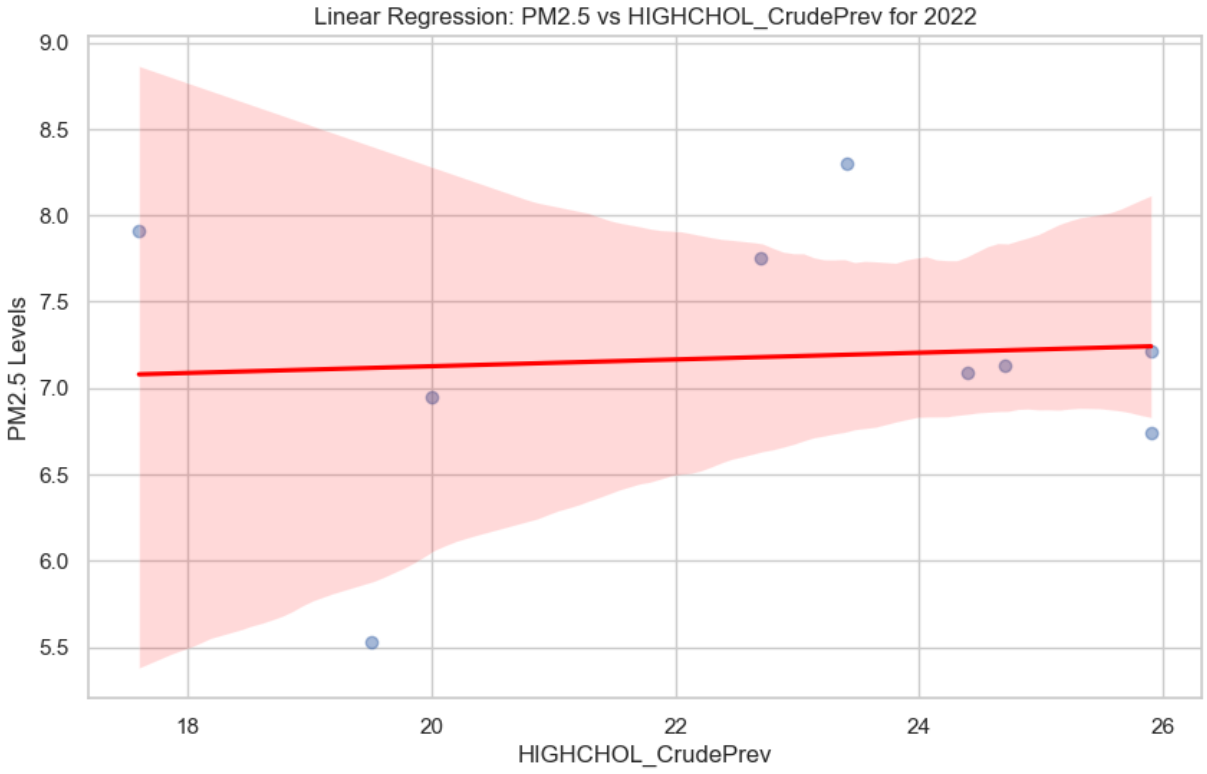

Linear Regression: PM2.5 vs CSMOKING_CrudePrev for 2022
PM2.5 Levels
CSMOKING_CrudePrev
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
8
10
12
14
16

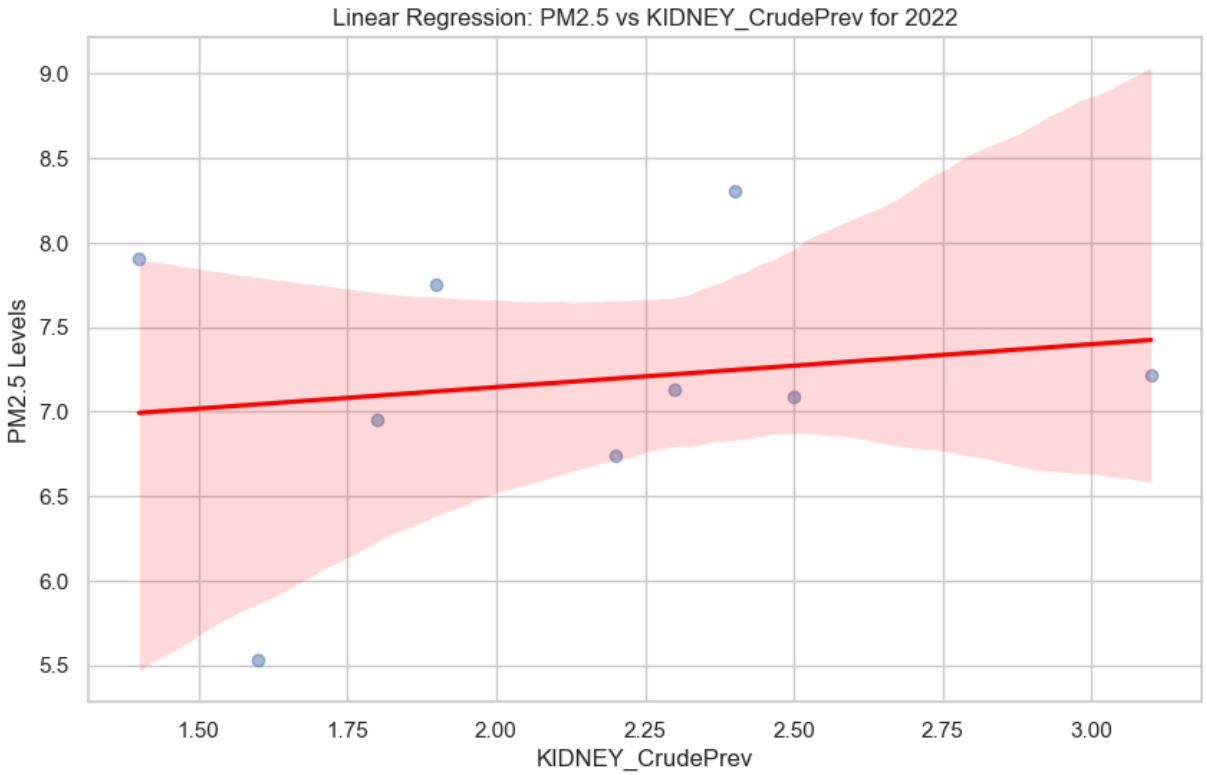

Linear Regression: PM2.5 vs DENTAL_CrudePrev for 2022
PM2.5 Levels
DENTAL_CrudePrev
8.0
7.5
7.0
6.5
6.0
5.5
60.0
62.5
65.0
67.5
70.0
72.5
75.0
77.5
80.0

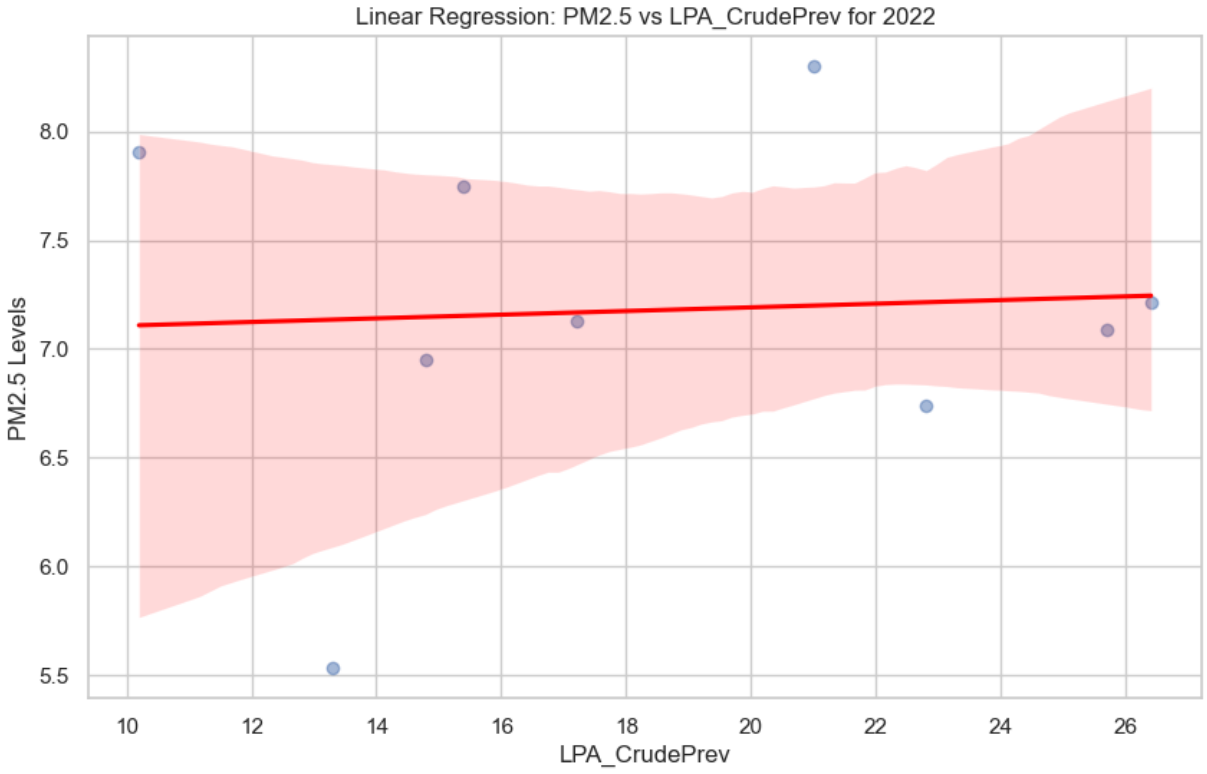

Linear Regression: PM2.5 vs DEPRESSION_CrudePrev for 2022
PM2.5 Levels
8.0
7.5
7.0
6.5
6.0
5.5
17.0
17.5
18.0
18.5
19.0
19.5
20.0
20.5
DEPRESSION_CrudePrev

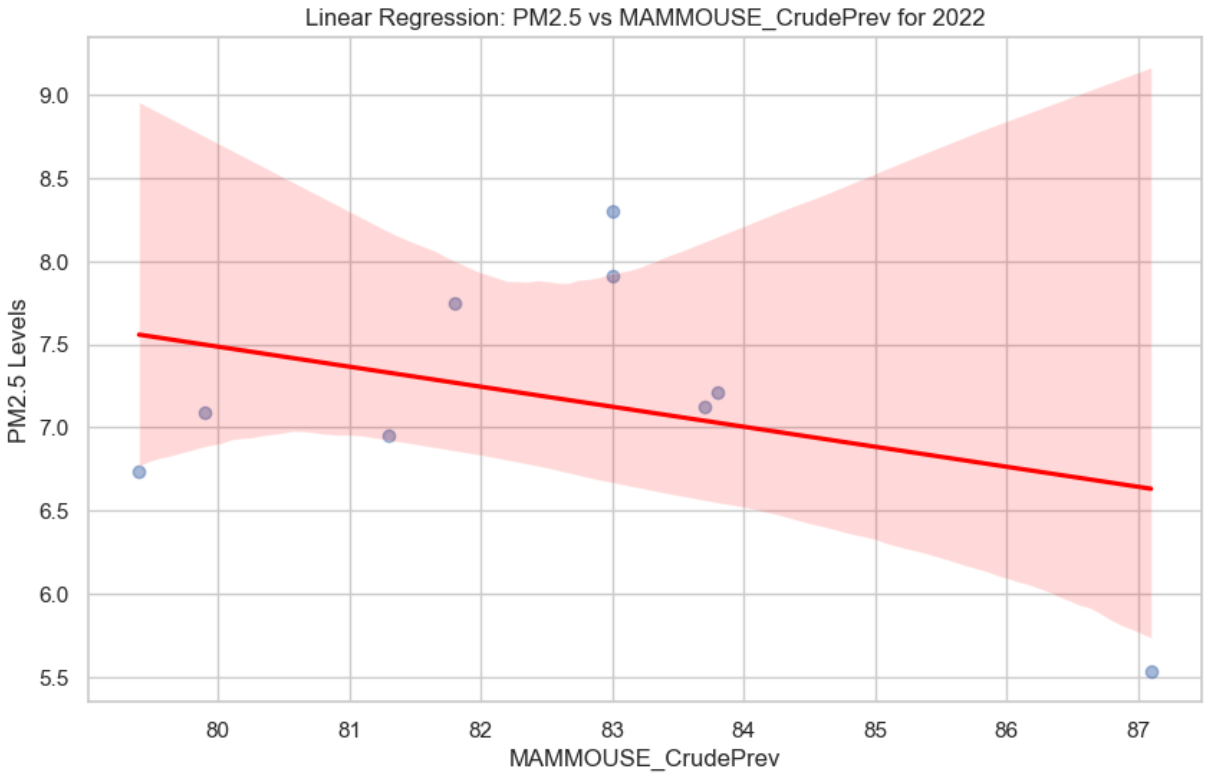

Linear Regression: PM2.5 vs DIABETES_CrudePrev for 2022
PM2.5 Levels
8.5
8.0
7.5
7.0
6.5
6.0
5.5
4
6
8
10
12
DIABETES_CrudePrev

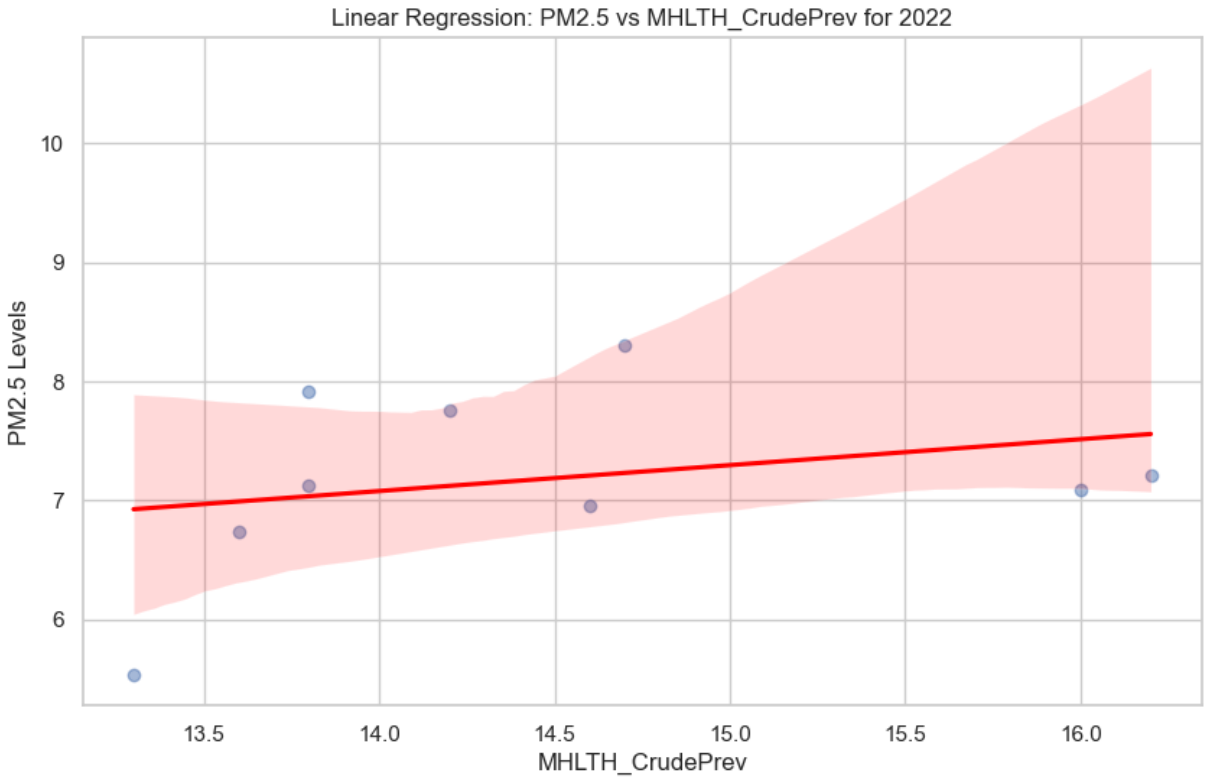

Linear Regression: PM2.5 vs GHLTH_CrudePrev for 2022
PM2.5 Levels
GHLTH_CrudePrev
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
6
8
10
12
14
16

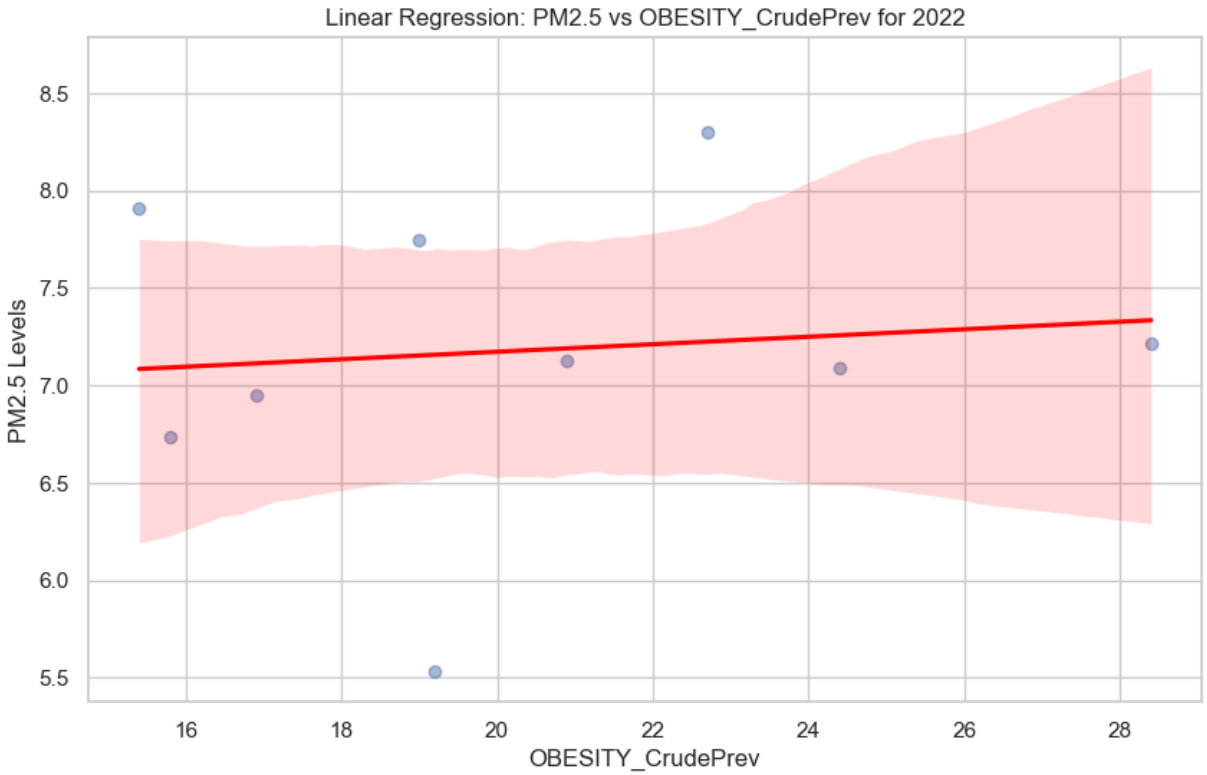

Linear Regression: PM2.5 vs HIGHCHOL_CrudePrev for 2022
PM2.5 Levels
HIGHCHOL_CrudePrev
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
18
20
22
24
26

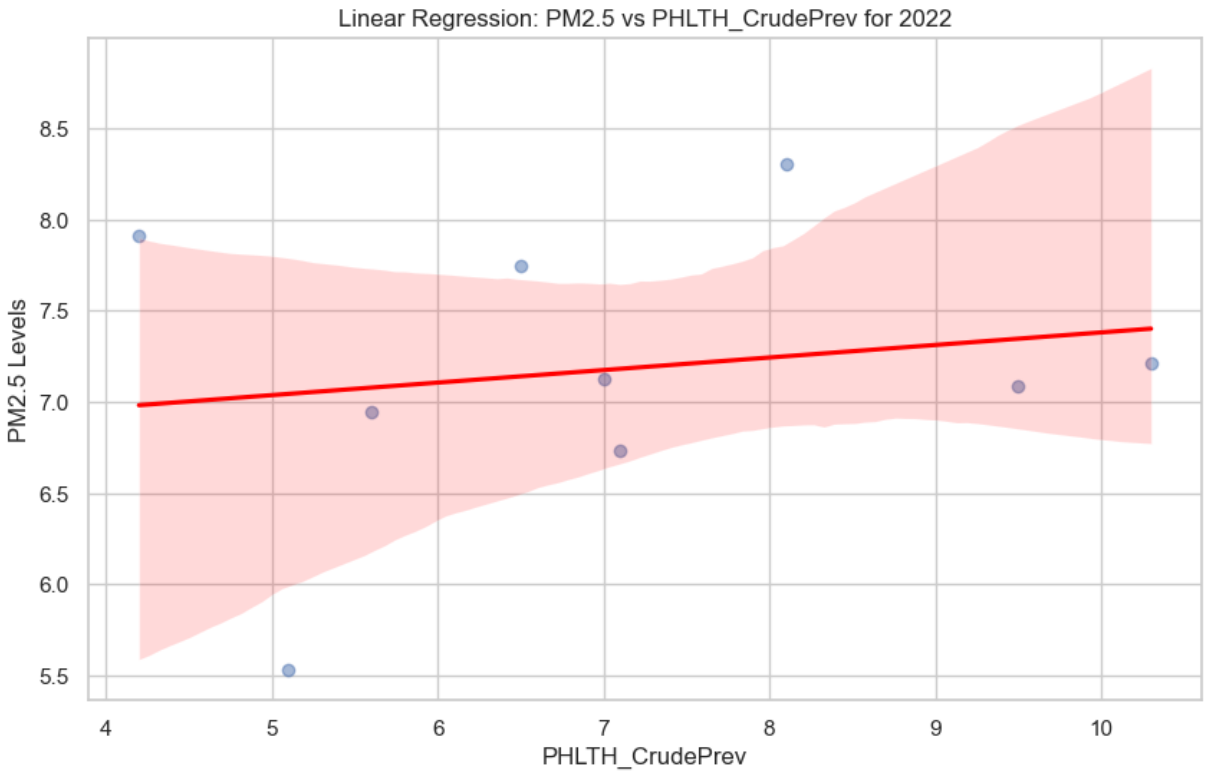

Linear Regression: PM2.5 vs KIDNEY_CrudePrev for 2022
PM2.5 Levels
KIDNEY_CrudePrev
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
1.50
1.75
2.00
2.25
2.50
2.75
3.00

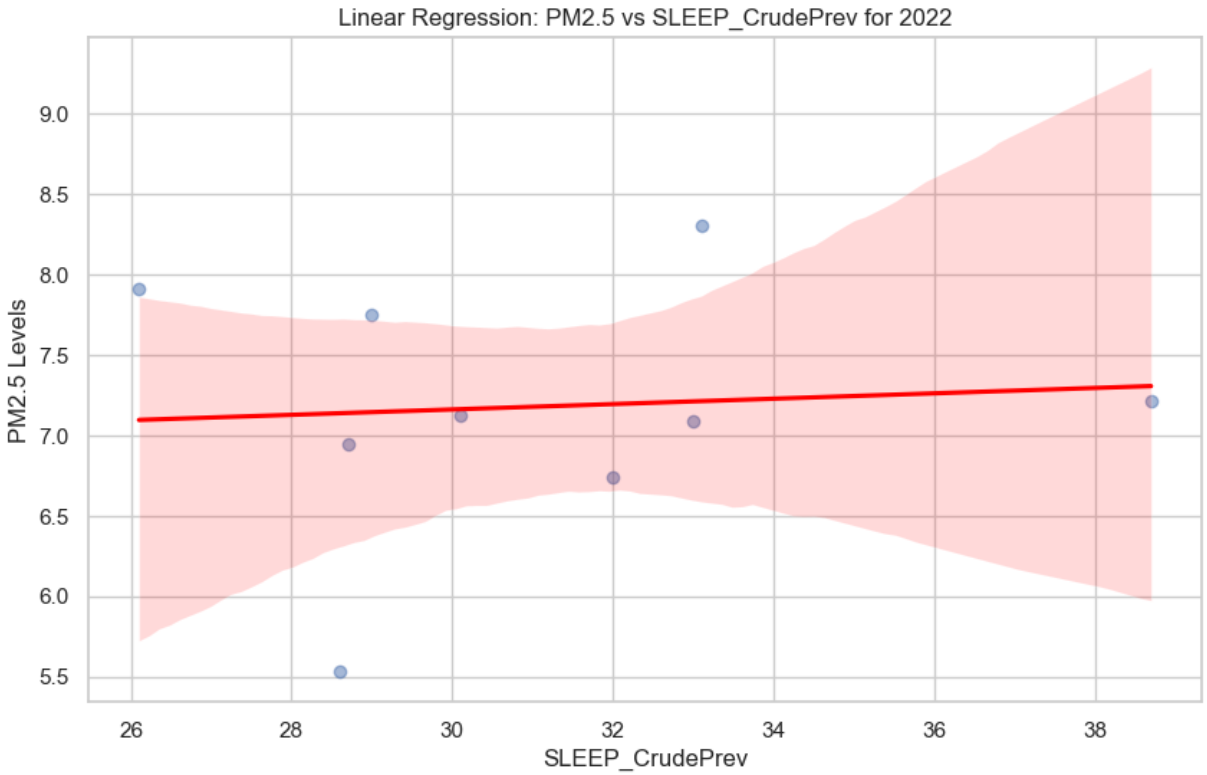

Linear Regression: PM2.5 vs LPA_CrudePrev for 2022
PM2.5 Levels
LPA_CrudePrev
8.0
7.5
7.0
6.5
6.0
5.5
10
12
14
16
18
20
22
24
26

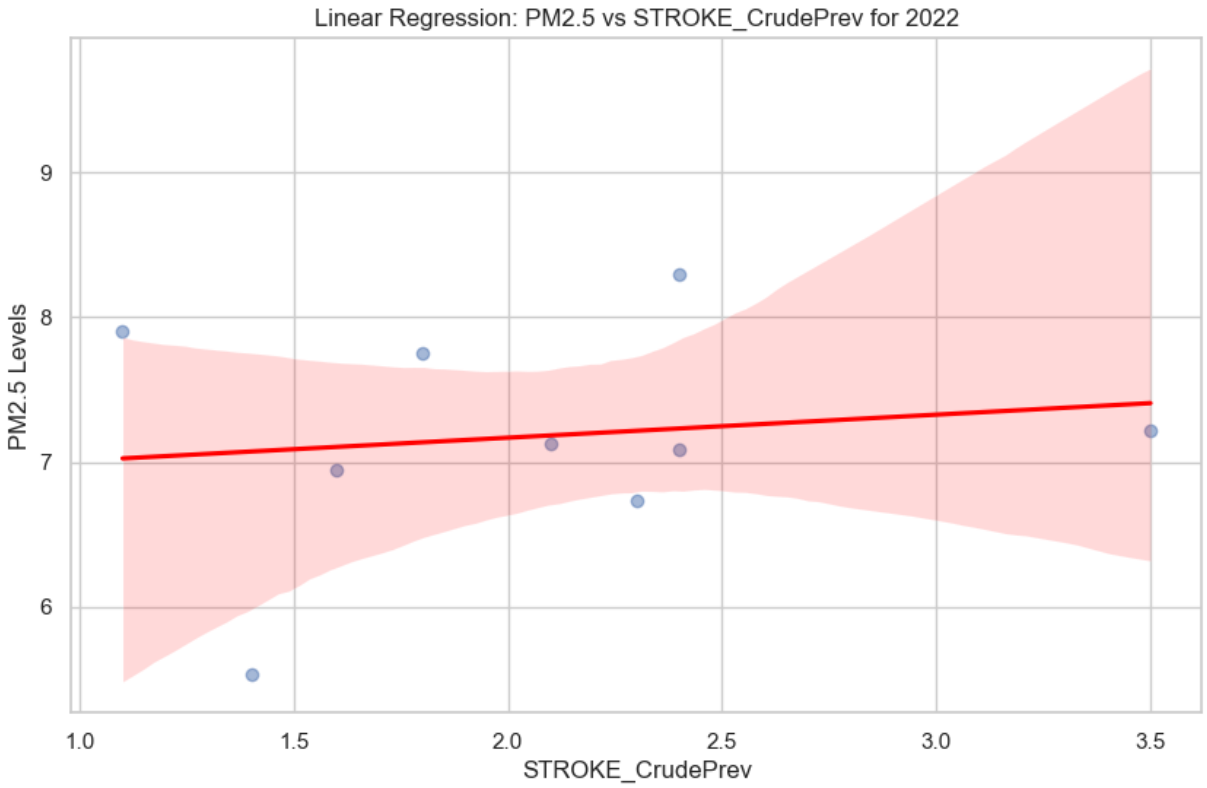

Linear Regression: PM2.5 vs MAMMOUSE_CrudePrev for 2022
PM2.5 Levels
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
80
81
82
83
84
85
86
87
MAMMOUSE_CrudePrev

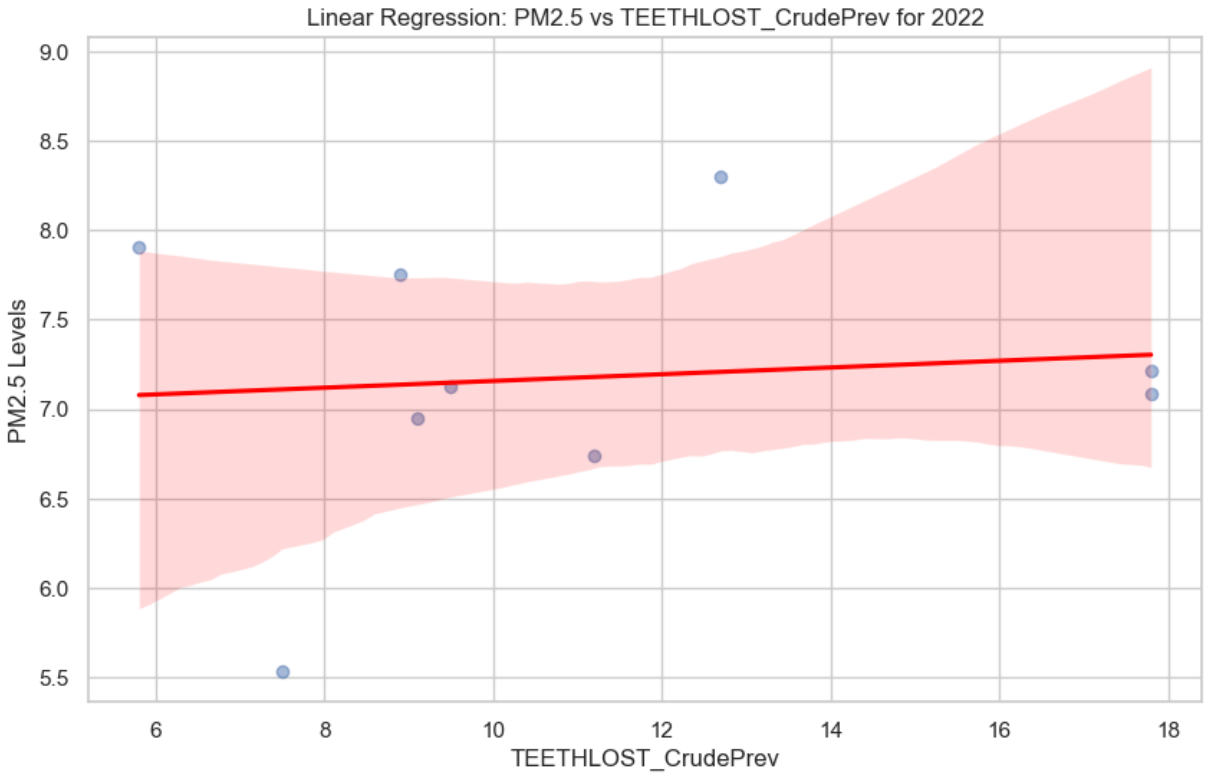

Linear Regression: PM2.5 vs MHLTH_CrudePrev for 2022
PM2.5 Levels
10
9
8
7
6
13.5
14.0
14.5
15.0
15.5
16.0
MHLTH_CrudePrev


Linear Regression: PM2.5 vs OBESITY_CrudePrev for 2022
PM2.5 Levels
OBESITY_CrudePrev
8.5
8.0
7.5
7.0
6.5
6.0
5.5
16
18
20
22
24
26
28

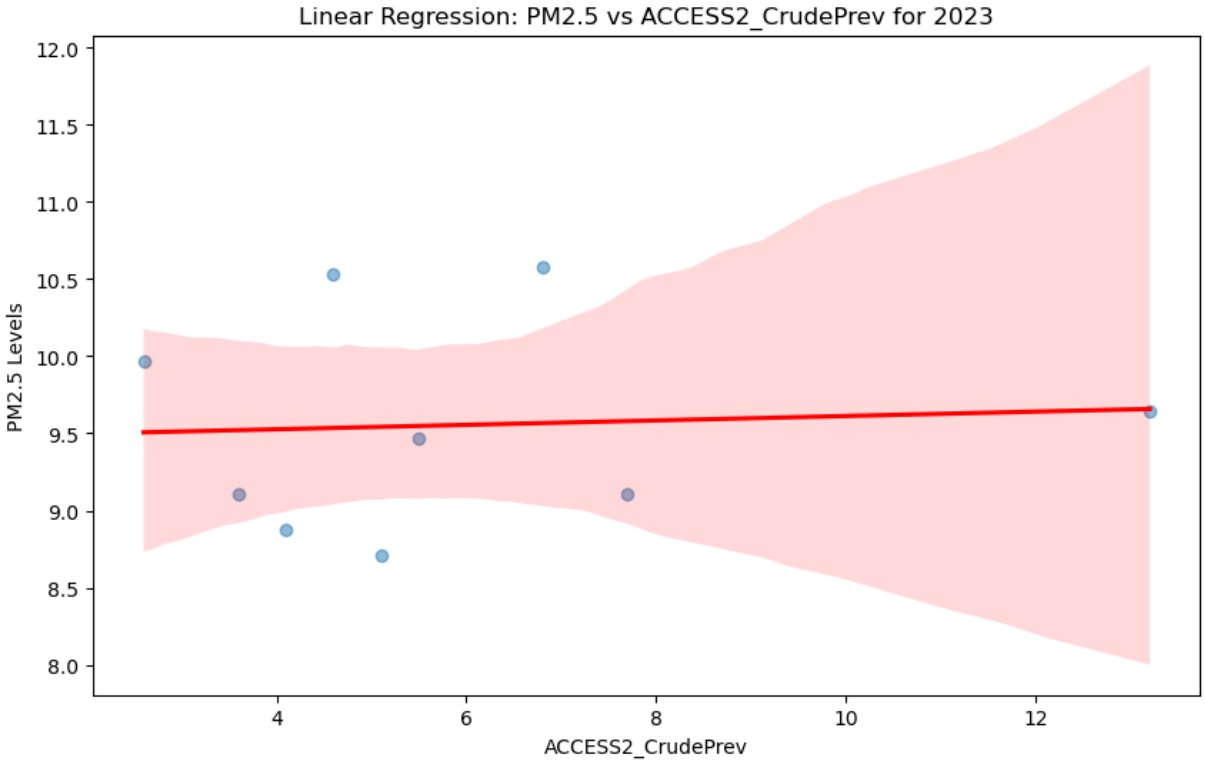

Linear Regression: PM2.5 vs PHLTH_CrudePrev for 2022
PM2.5 Levels
PHLTH_CrudePrev
8.5
8.0
7.5
7.0
6.5
6.0
5.5
4
5
6
7
8
9
10

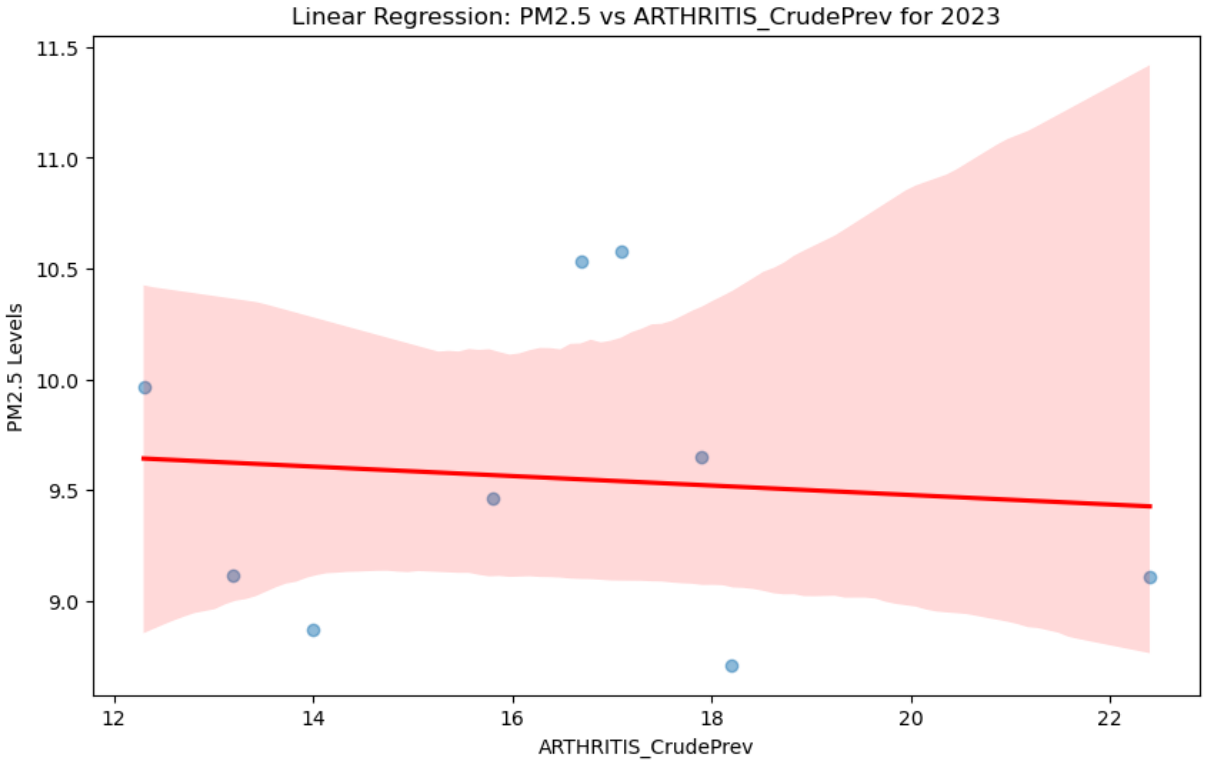

Linear Regression: PM2.5 vs SLEEP_CrudePrev for 2022
PM2.5 Levels
SLEEP_CrudePrev
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
26
28
30
32
34
36
38

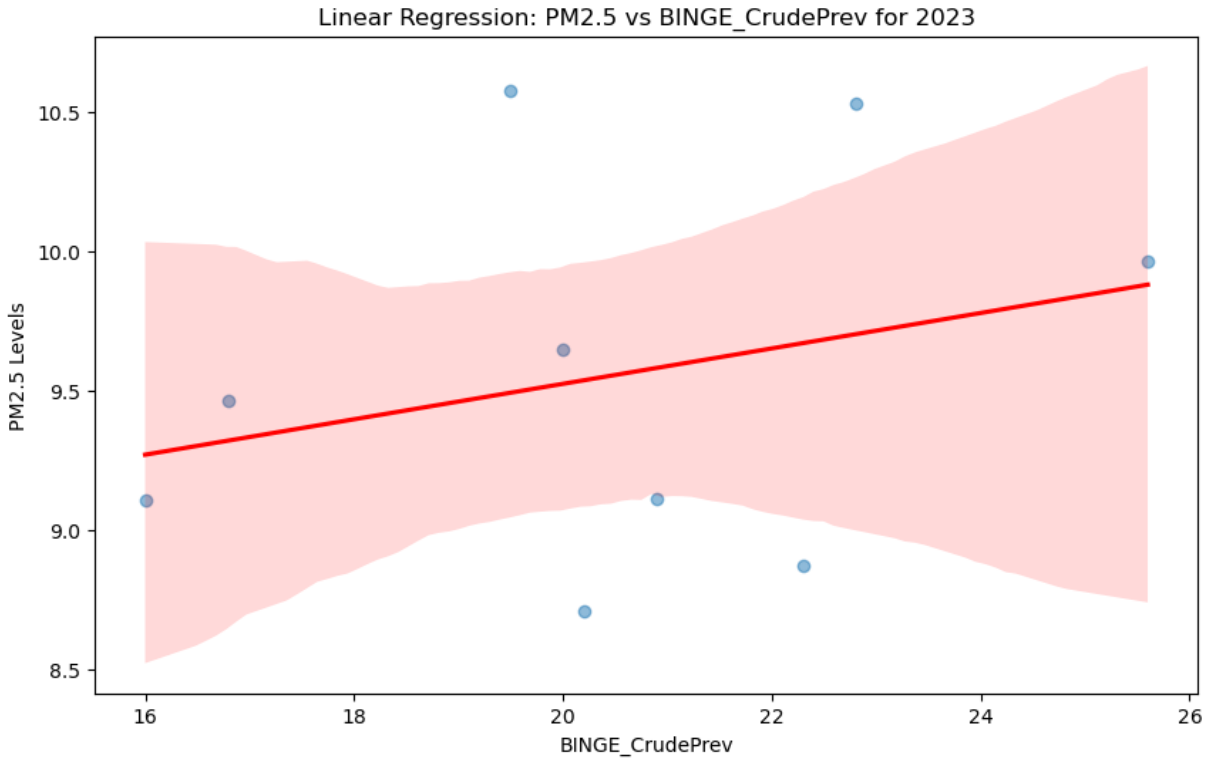

Linear Regression: PM2.5 vs STROKE_CrudePrev for 2022
PM2.5 Levels
STROKE_CrudePrev
9
8
7
6
1.0
1.5
2.0
2.5
3.0
3.5

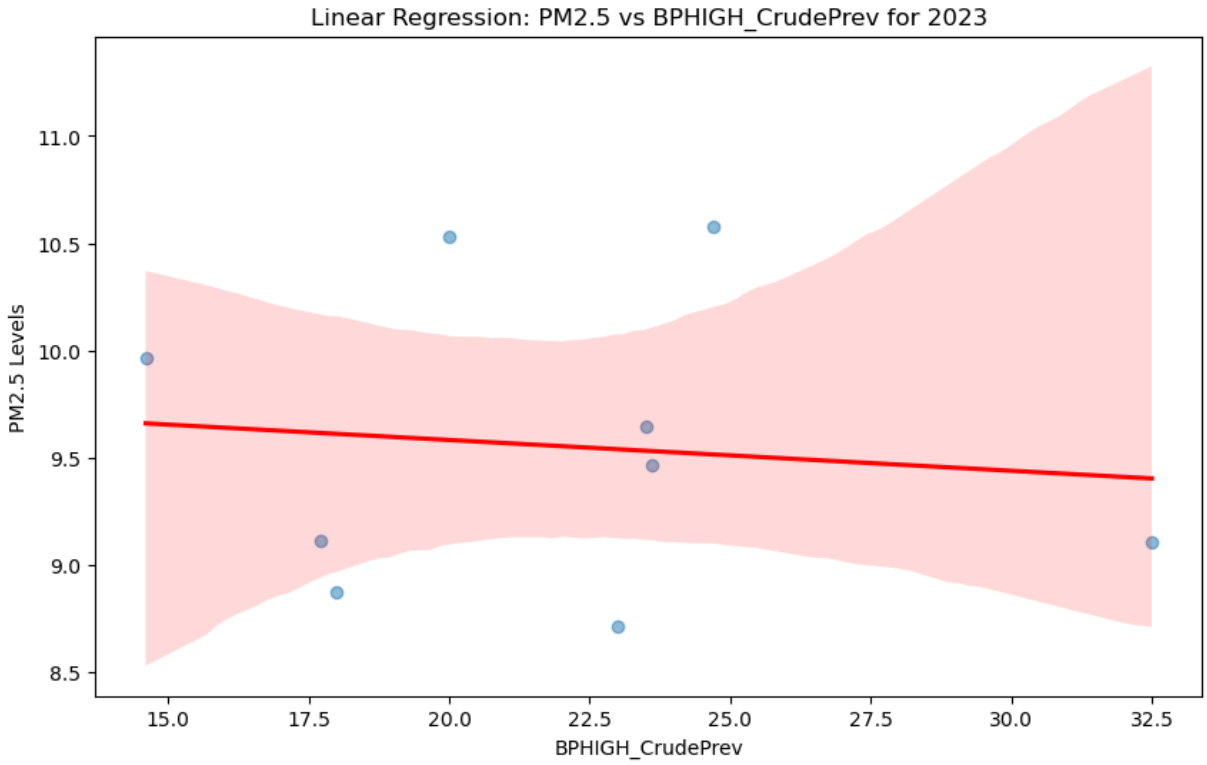


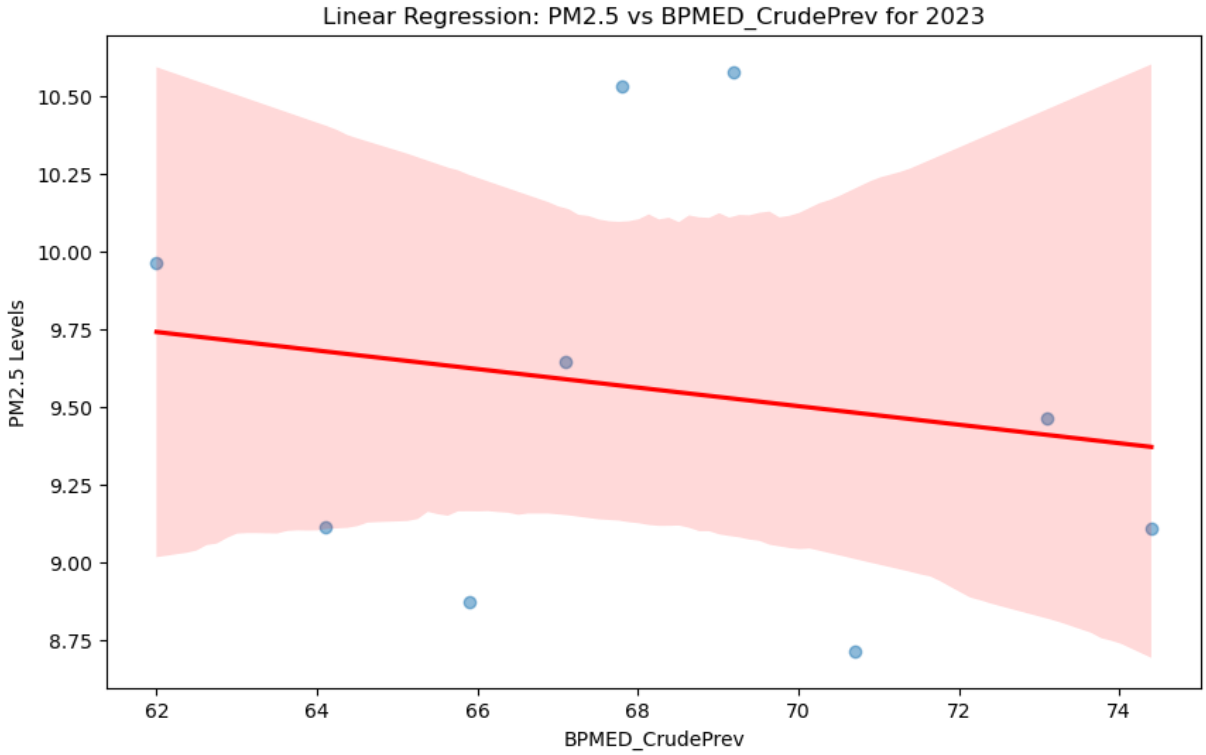
In [10]:
```
def plot_linear_regression(data, crude_columns, year):
    for column in crude_columns:
        if column != 'PM2.5':
            # Performing linear regression
            slope, intercept, r_value, p_value, std_err = stats.linregress(d

            # Plotting
            plt.figure(figsize=(10, 6))
            sns.regplot(x=column, y='PM2.5', data=data, scatter_kws={'alpha'
            plt.title(f'Linear Regression: PM2.5 vs {column} for {year}')
            plt.xlabel(column)
            plt.ylabel('PM2.5 Levels')
            plt.show()


crude_columns_2023 = [col for col in health_data_2023.columns if 'Crude' in

plot_linear_regression(health_data_2023, crude_columns_2023, '2023')
```

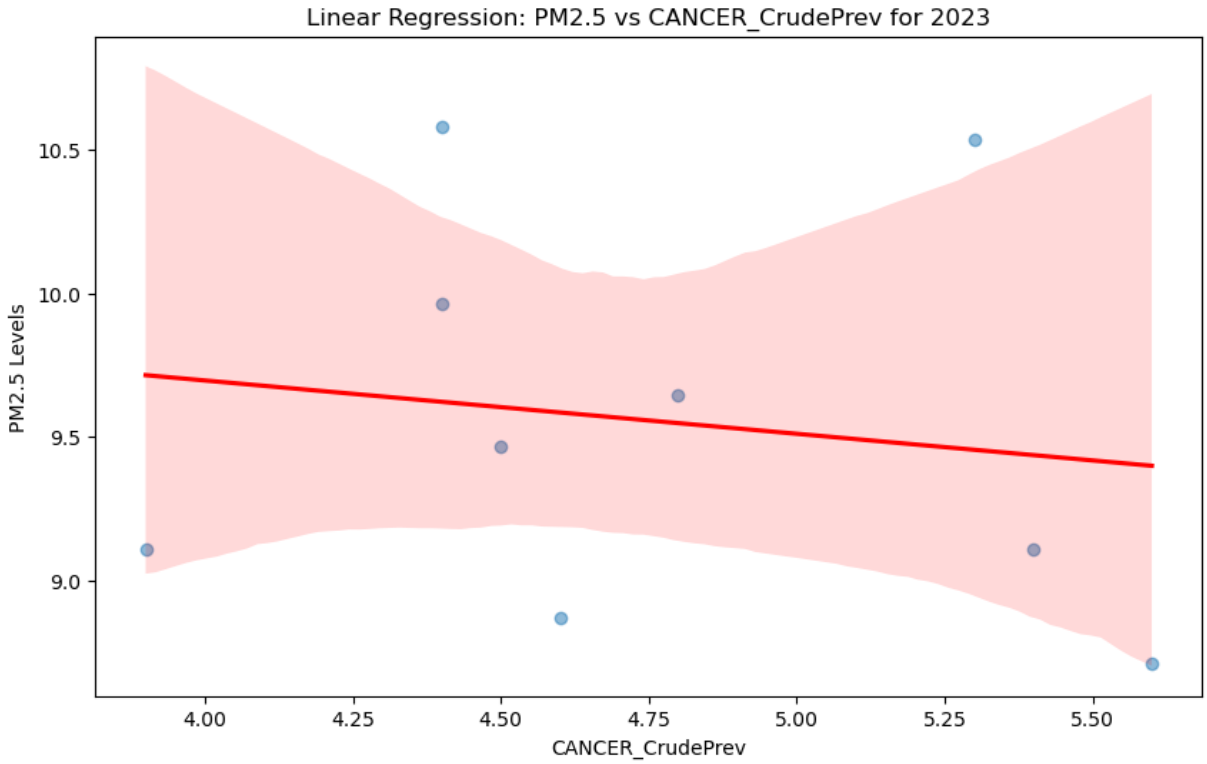

Linear Regression: PM2.5 vs ACCESS2_CrudePrev for 2023
PM2.5 Levels
ACCESS2_CrudePrev
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
8.0
4
6
8
10
12

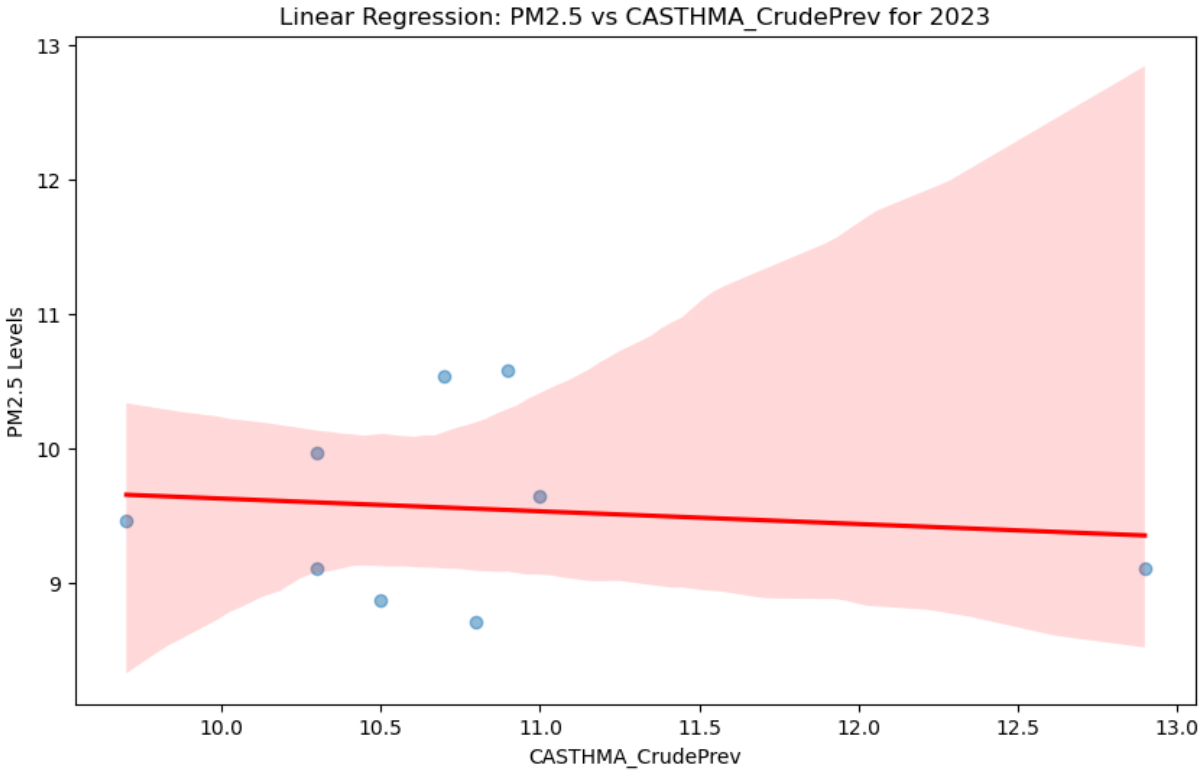

Linear Regression: PM2.5 vs ARTHRITIS_CrudePrev for 2023
PM2.5 Levels
ARTHRITIS_CrudePrev
11.5
11.0
10.5
10.0
9.5
9.0
12
14
16
18
20
22

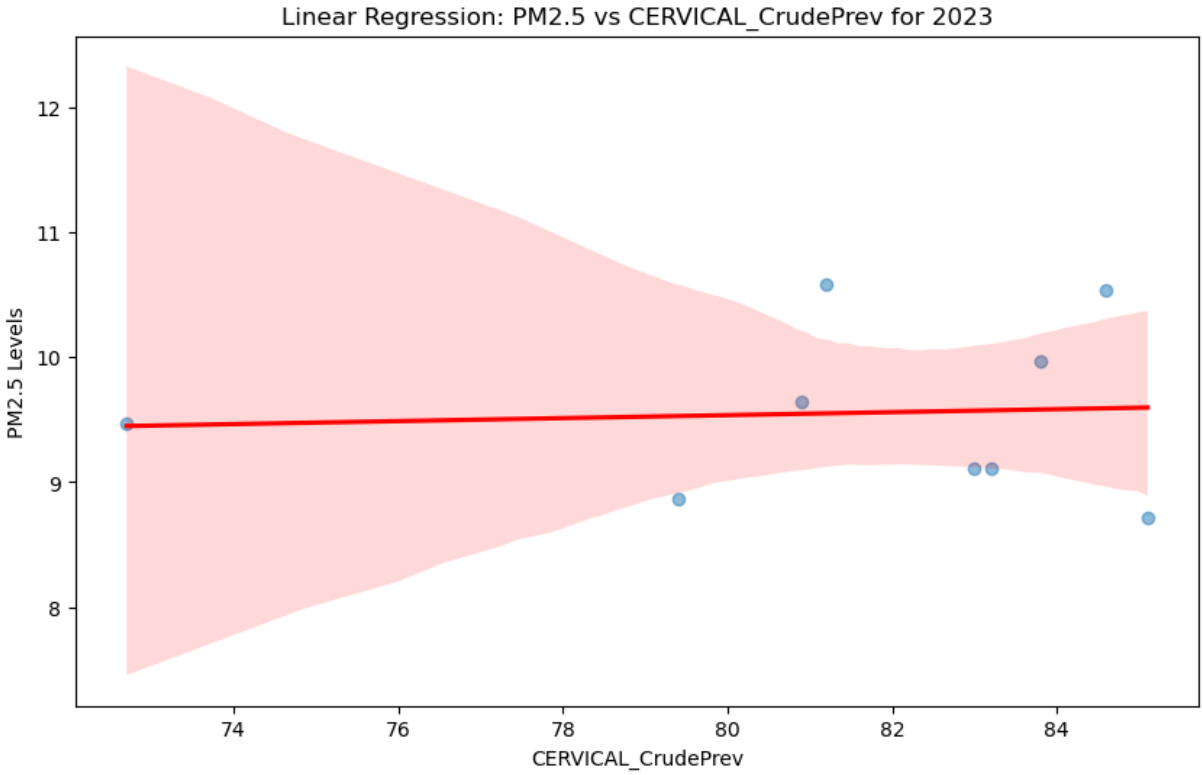

Linear Regression: PM2.5 vs BINGE_CrudePrev for 2023
PM2.5 Levels
BINGE_CrudePrev
10.5
10.0
9.5
9.0
8.5
16
18
20
22
24
26

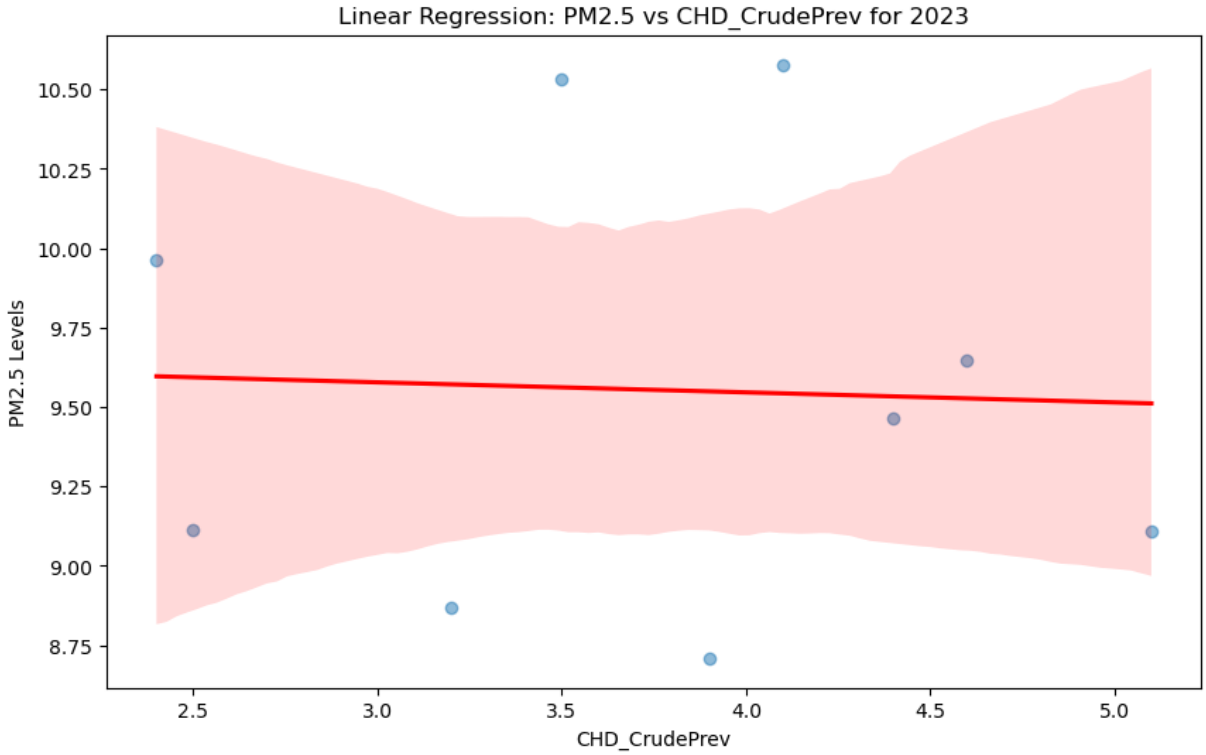

Linear Regression: PM2.5 vs BPHIGH_CrudePrev for 2023
PM2.5 Levels
BPHIGH_CrudePrev
11.0
10.5
10.0
9.5
9.0
8.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0
32.5

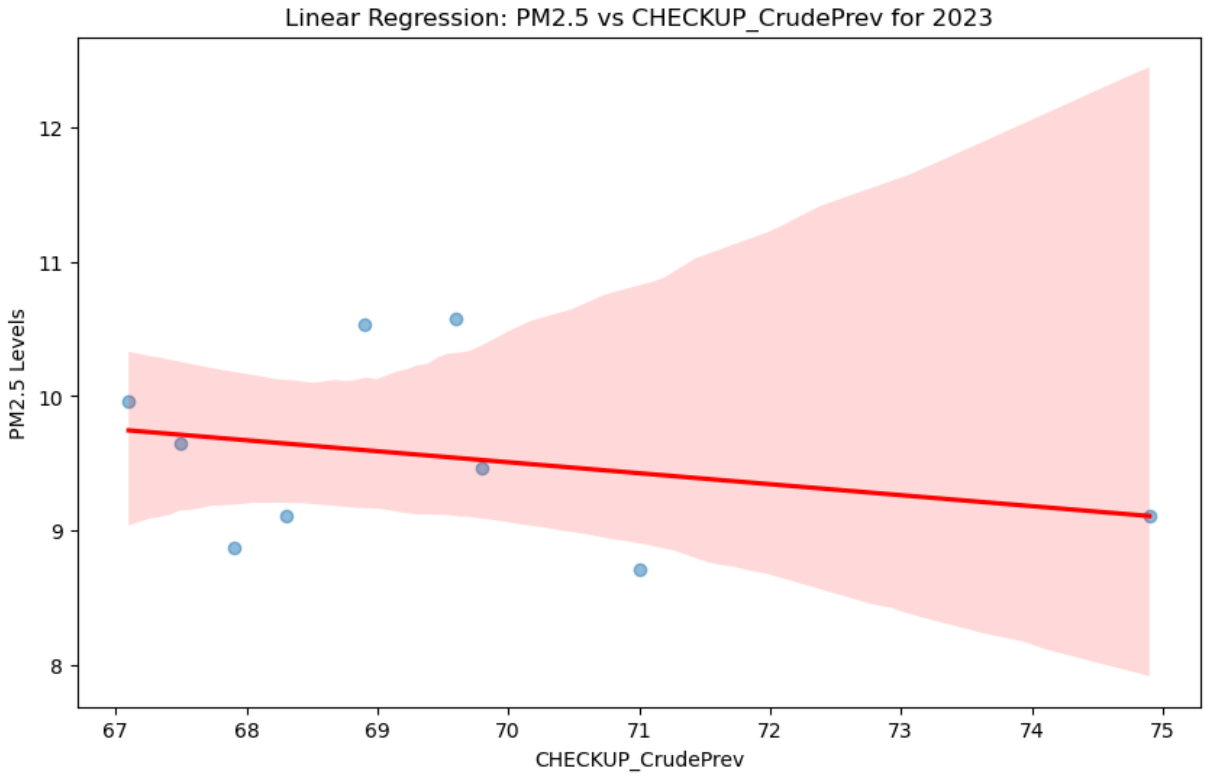

Linear Regression: PM2.5 vs BPMED_CrudePrev for 2023
PM2.5 Levels
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
62
64
66
68
70
72
74
BPMED_CrudePrev

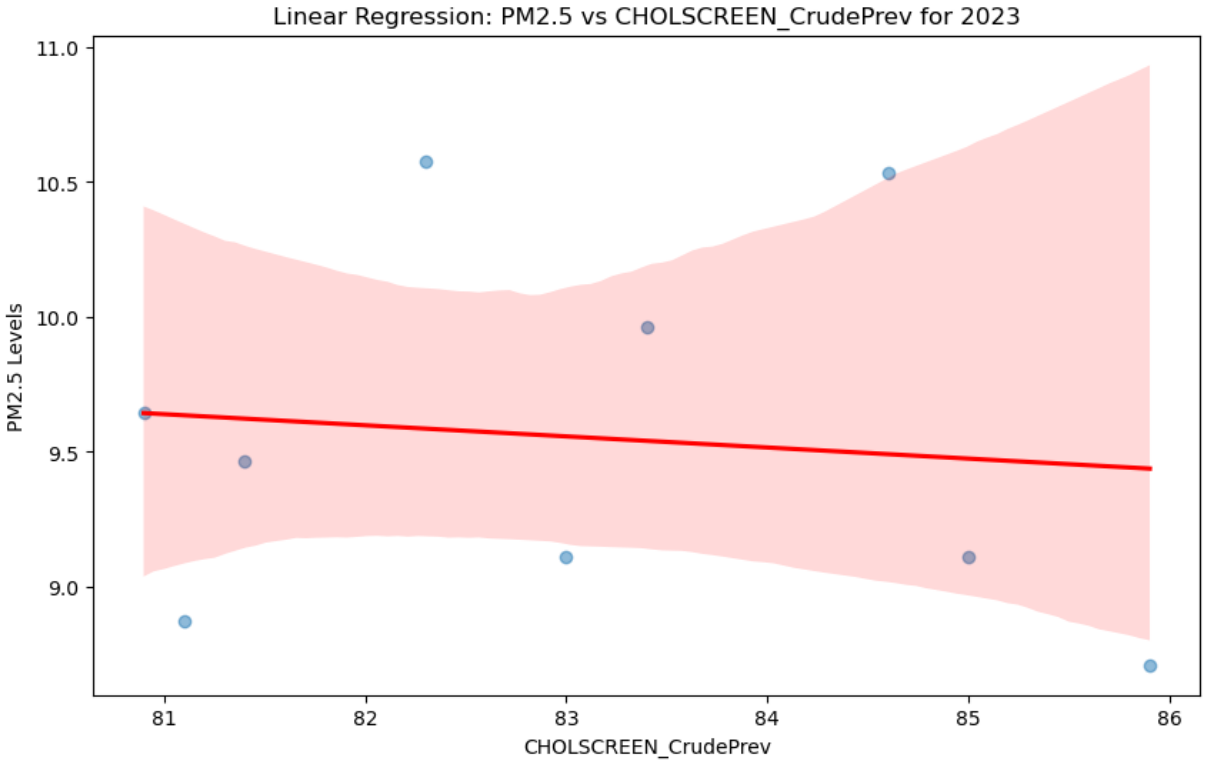

Linear Regression: PM2.5 vs CANCER_CrudePrev for 2023
PM2.5 Levels
10.5
10.0
9.5
9.0
4.00
4.25
4.50
4.75
5.00
5.25
5.50
CANCER_CrudePrev

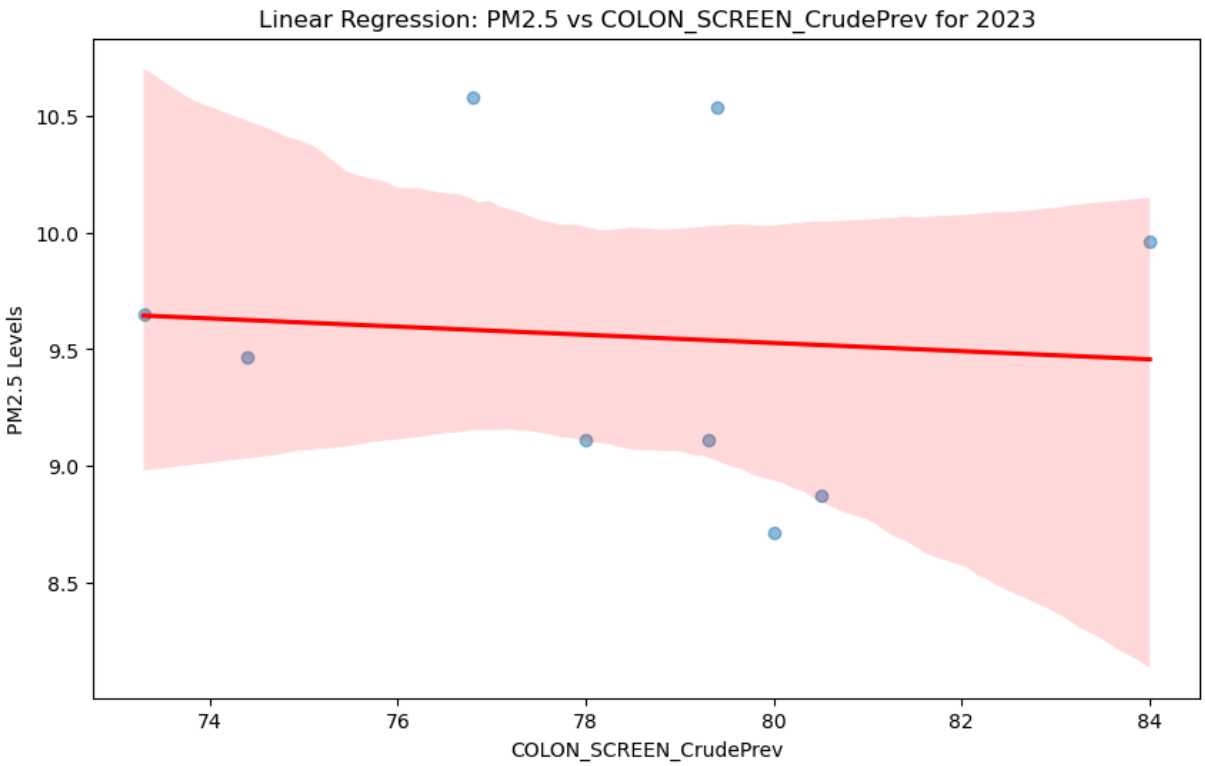

Linear Regression: PM2.5 vs CASTHMA_CrudePrev for 2023
PM2.5 Levels
CASTHMA_CrudePrev
13
12
11
10
9
10.0
10.5
11.0
11.5
12.0
12.5
13.0

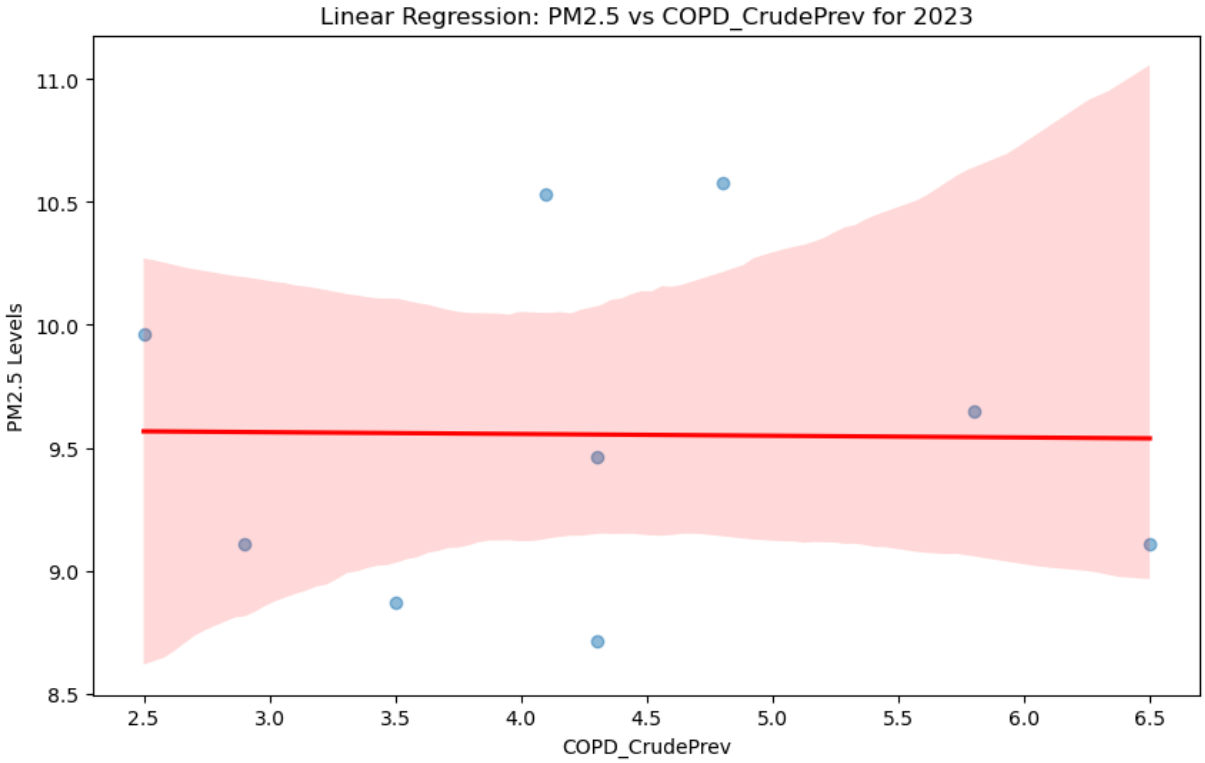

Linear Regression: PM2.5 vs CERVICAL_CrudePrev for 2023
PM2.5 Levels
CERVICAL_CrudePrev
12
11
10
9
8
74
76
78
80
82
84

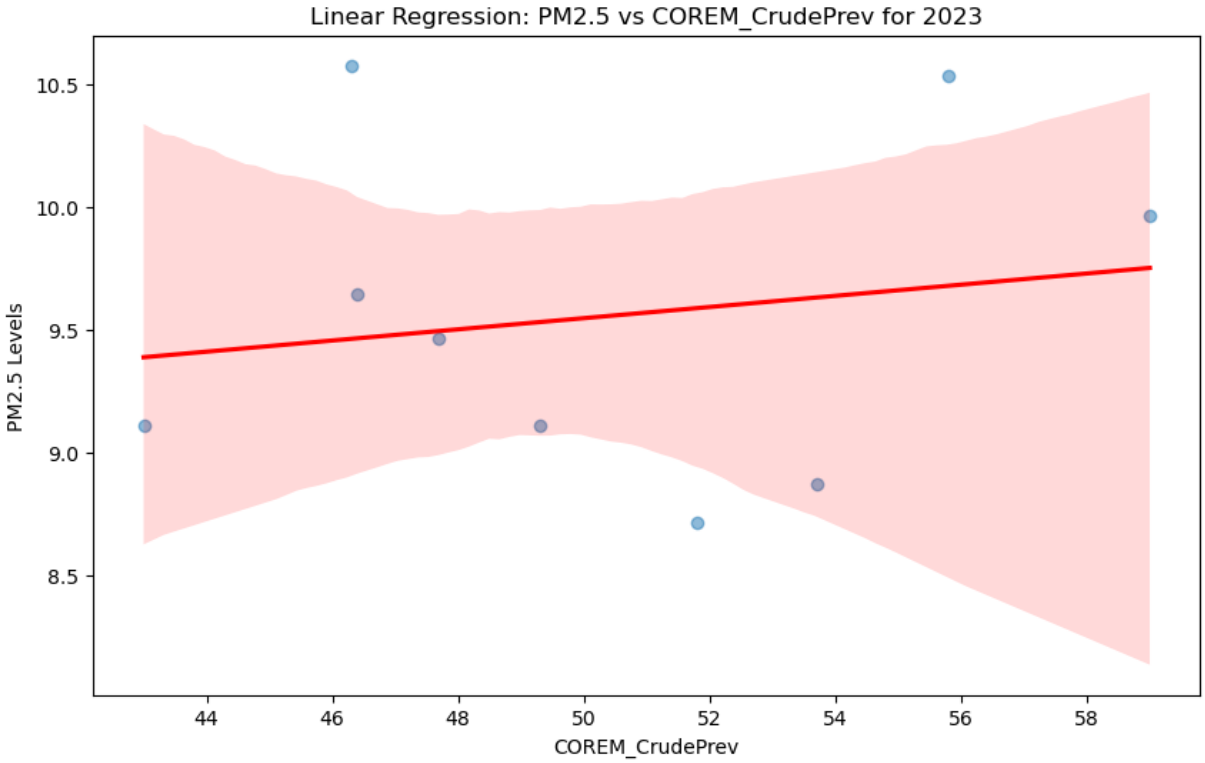

Linear Regression: PM2.5 vs CHD_CrudePrev for 2023
PM2.5 Levels
CHD_CrudePrev
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
2.5
3.0
3.5
4.0
4.5
5.0

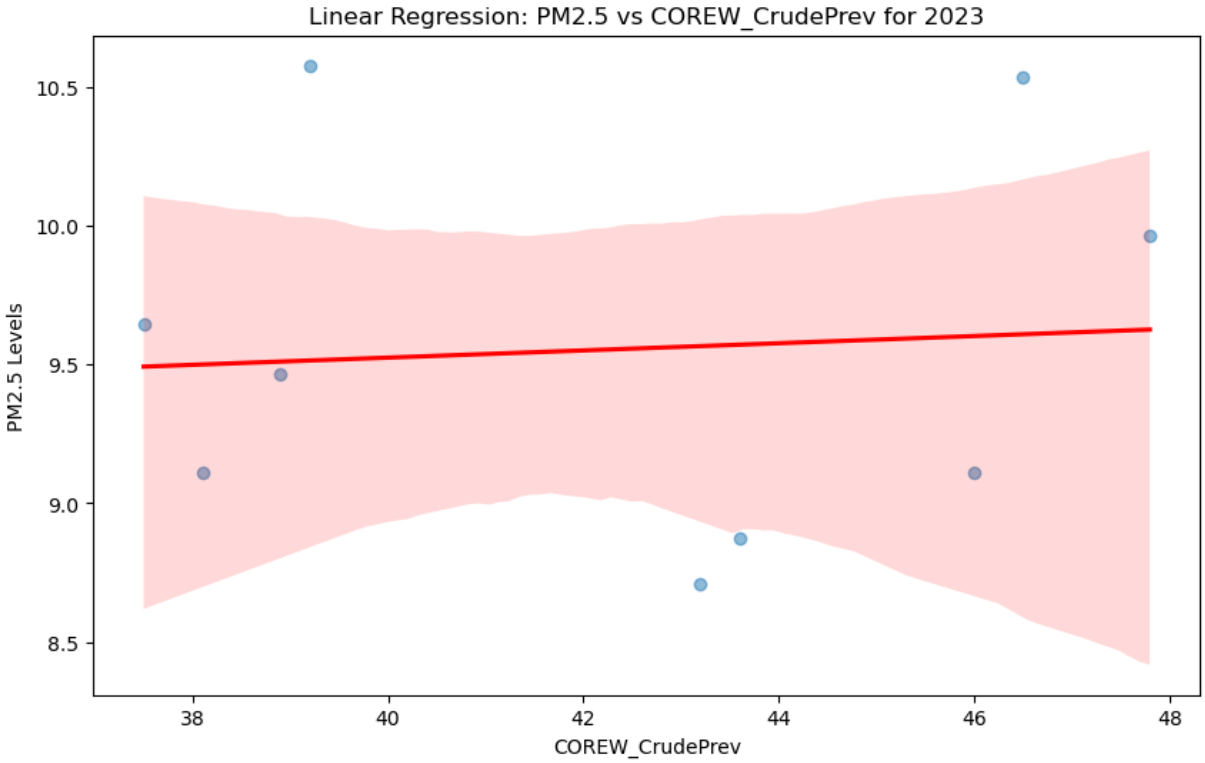

Linear Regression: PM2.5 vs CHECKUP_CrudePrev for 2023
PM2.5 Levels
CHECKUP_CrudePrev
12
11
10
9
8
67
68
69
70
71
72
73
74
75

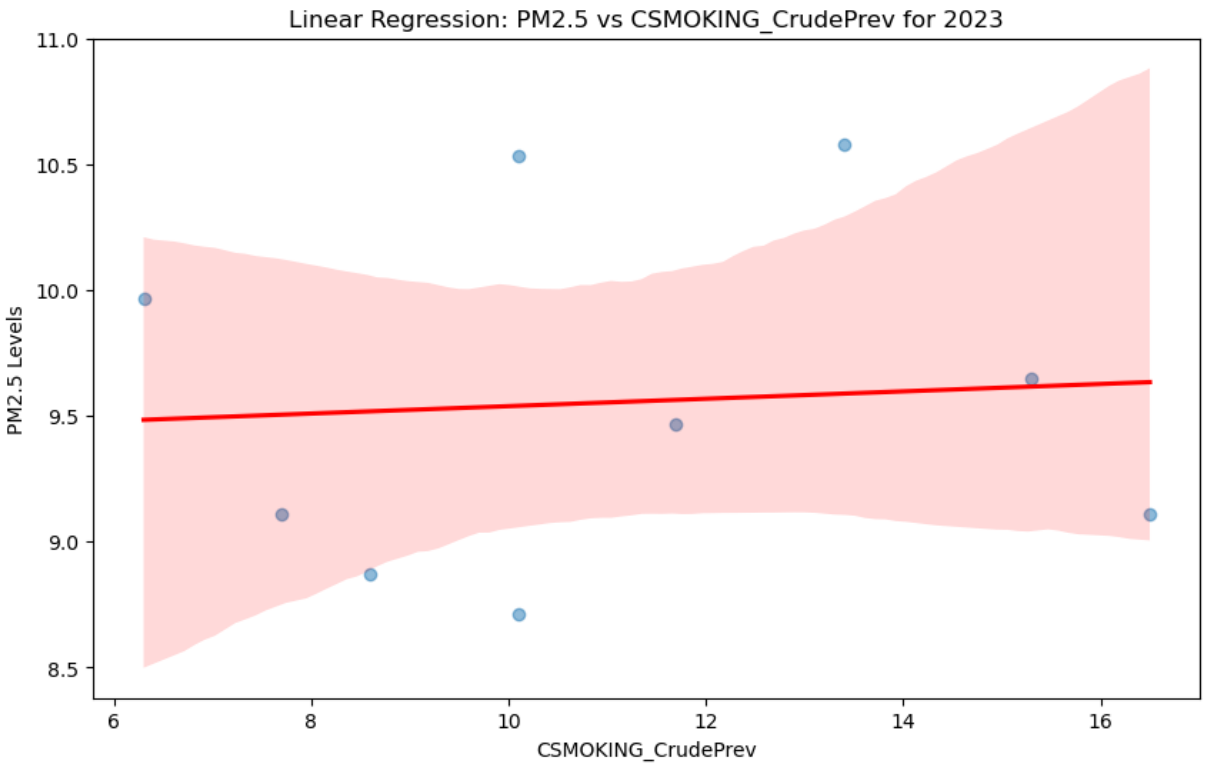

Linear Regression: PM2.5 vs CHOLSCREEN_CrudePrev for 2023
PM2.5 Levels
CHOLSCREEN_CrudePrev
11.0
10.5
10.0
9.5
9.0
81
82
83
84
85
86

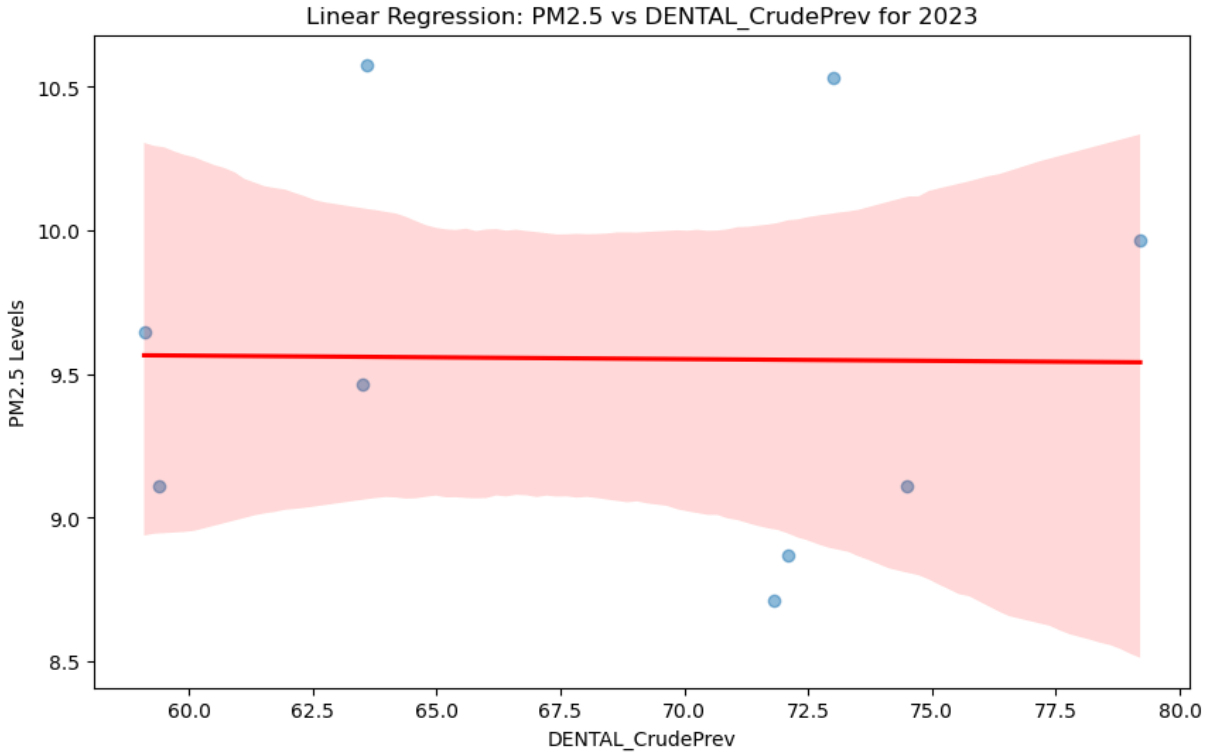

Linear Regression: PM2.5 vs COLON_SCREEN_CrudePrev for 2023
PM2.5 Levels
COLON_SCREEN_CrudePrev
10.5
10.0
9.5
9.0
8.5
74
76
78
80
82
84

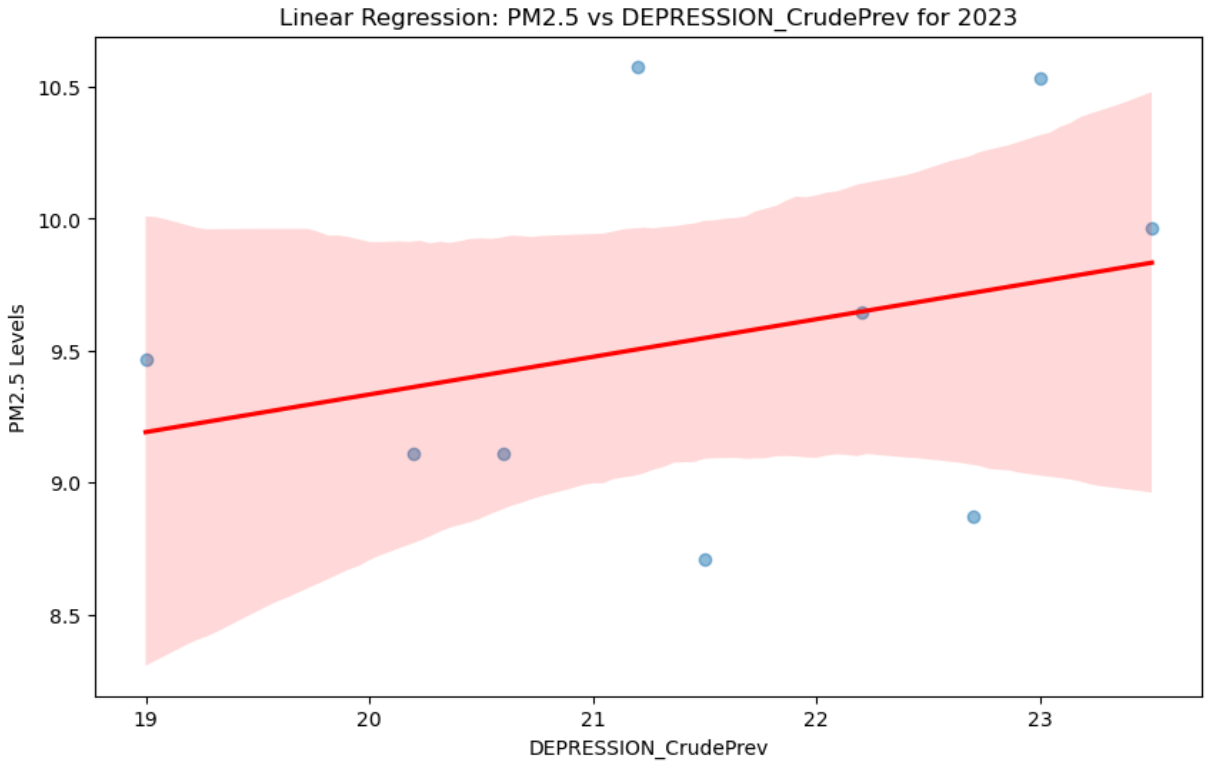

Linear Regression: PM2.5 vs COPD_CrudePrev for 2023
PM2.5 Levels
COPD_CrudePrev
11.0
10.5
10.0
9.5
9.0
8.5
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5

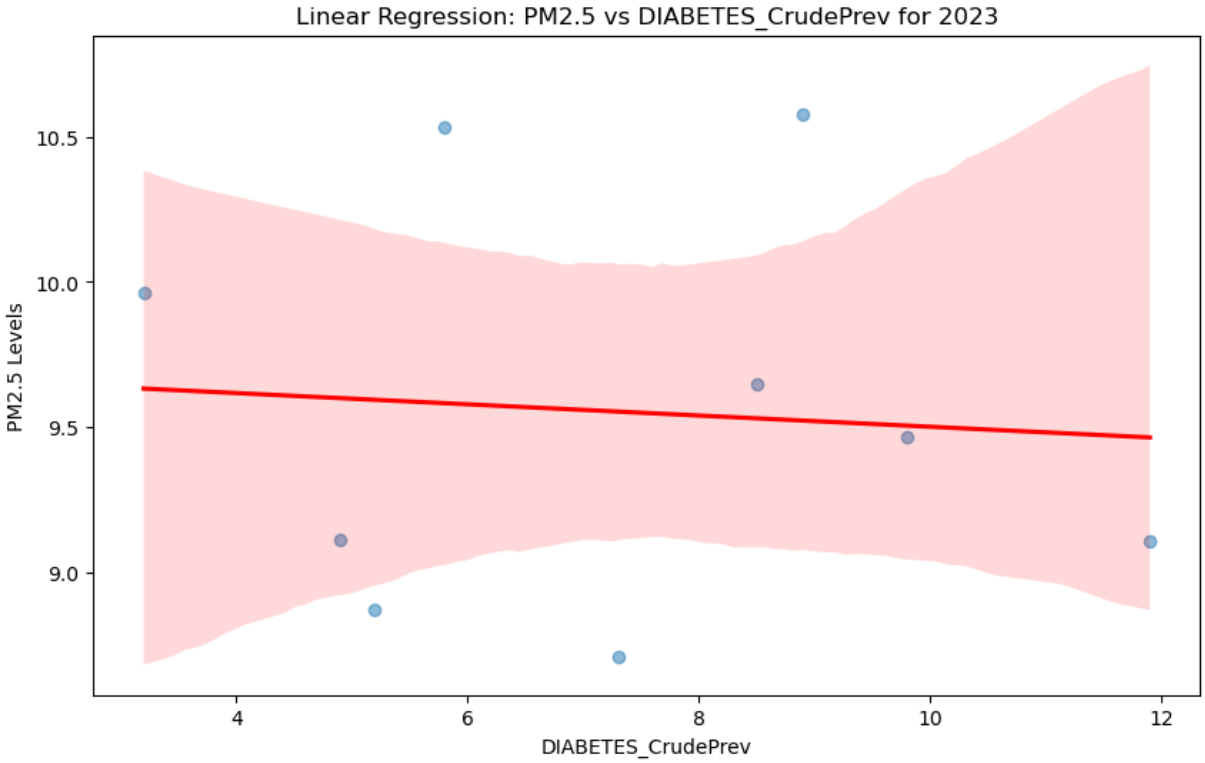

Linear Regression: PM2.5 vs COREM_CrudePrev for 2023
PM2.5 Levels
COREM_CrudePrev
10.5
10.0
9.5
9.0
8.5
44
46
48
50
52
54
56
58

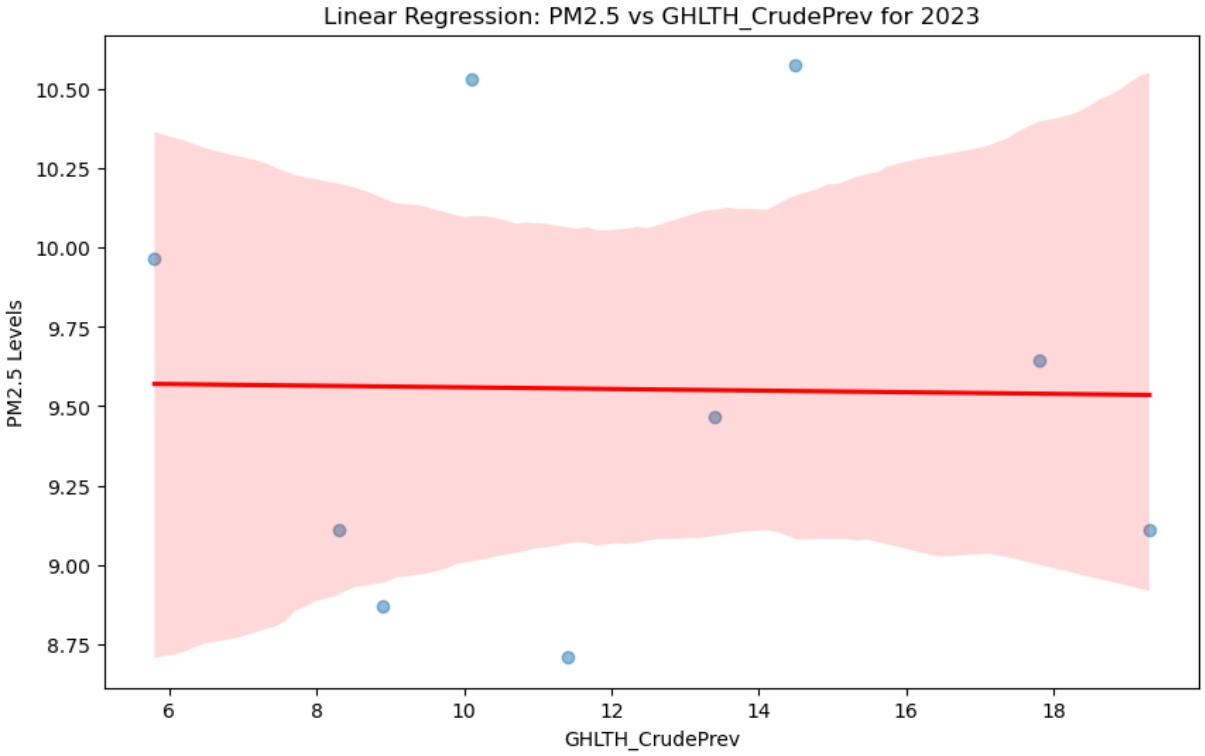

Linear Regression: PM2.5 vs COREW_CrudePrev for 2023
PM2.5 Levels
10.5
10.0
9.5
9.0
8.5
38
40
42
44
46
48
COREW_CrudePrev

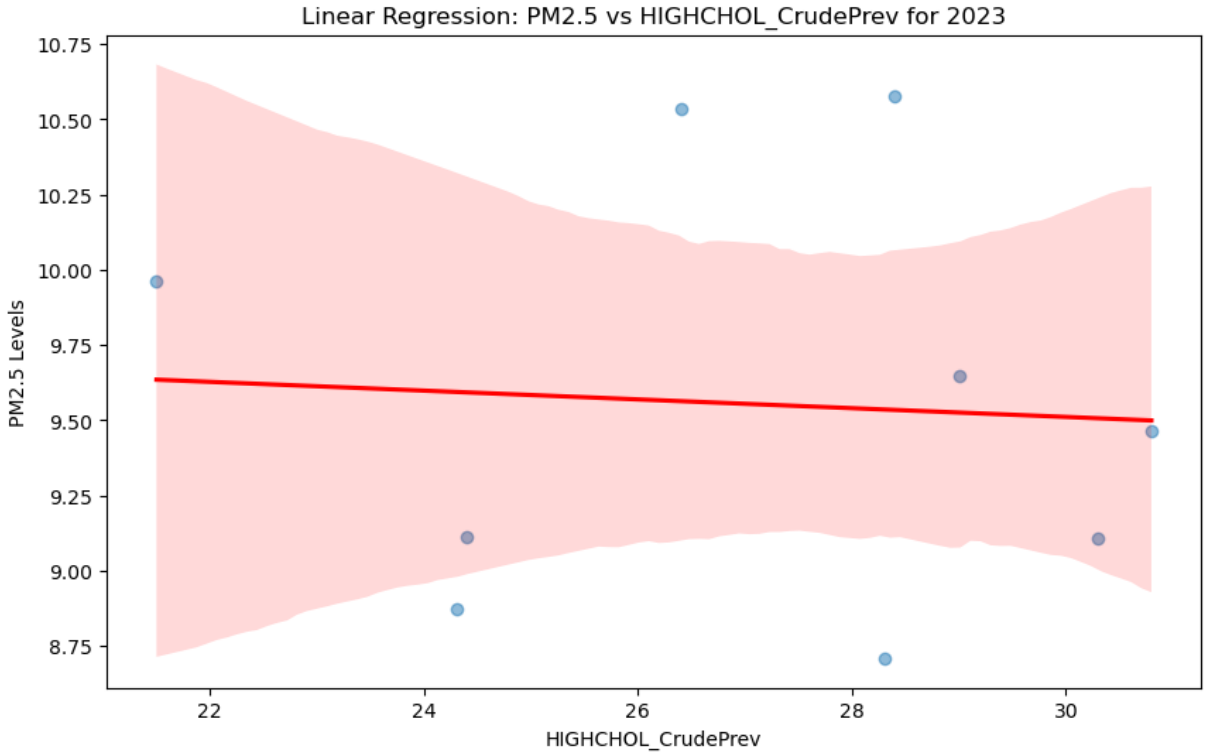

Linear Regression: PM2.5 vs CSMOKING_CrudePrev for 2023
PM2.5 Levels
11.0
10.5
10.0
9.5
9.0
8.5
6
8
10
12
14
16
CSMOKING_CrudePrev

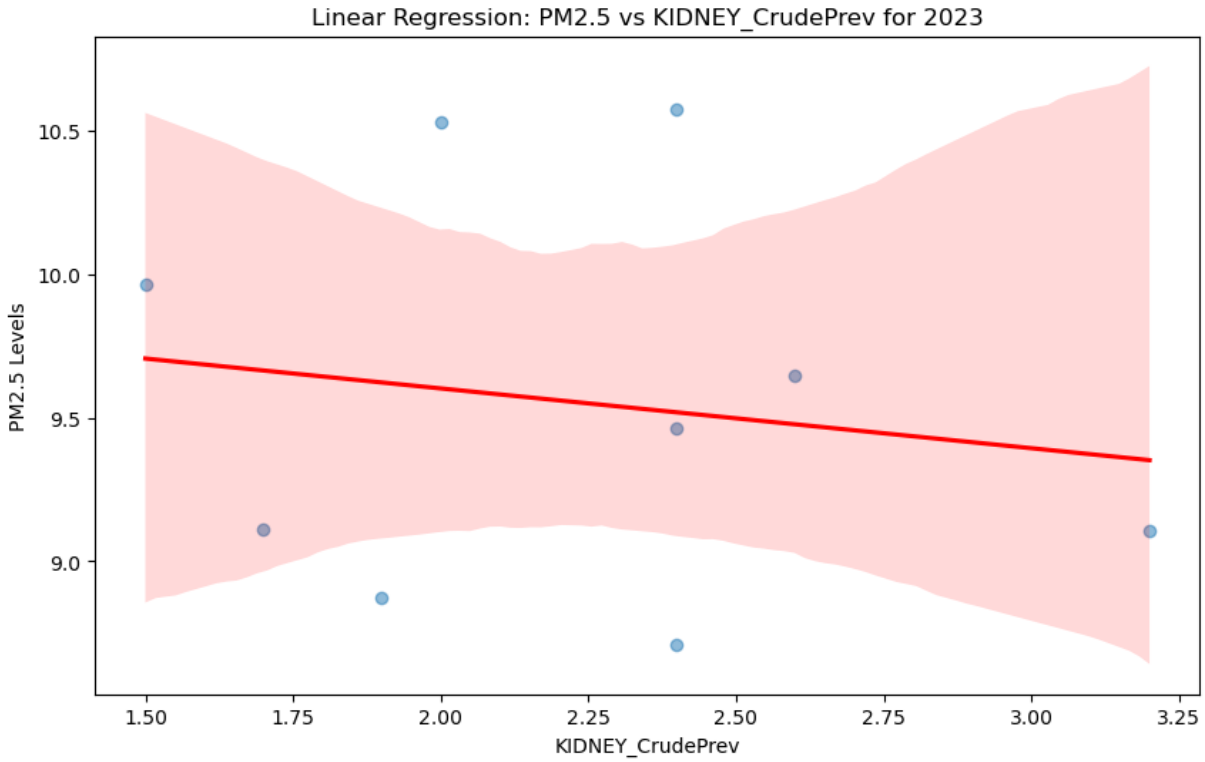

Linear Regression: PM2.5 vs DENTAL_CrudePrev for 2023
PM2.5 Levels
DENTAL_CrudePrev
10.5
10.0
9.5
9.0
8.5
60.0
62.5
65.0
67.5
70.0
72.5
75.0
77.5
80.0

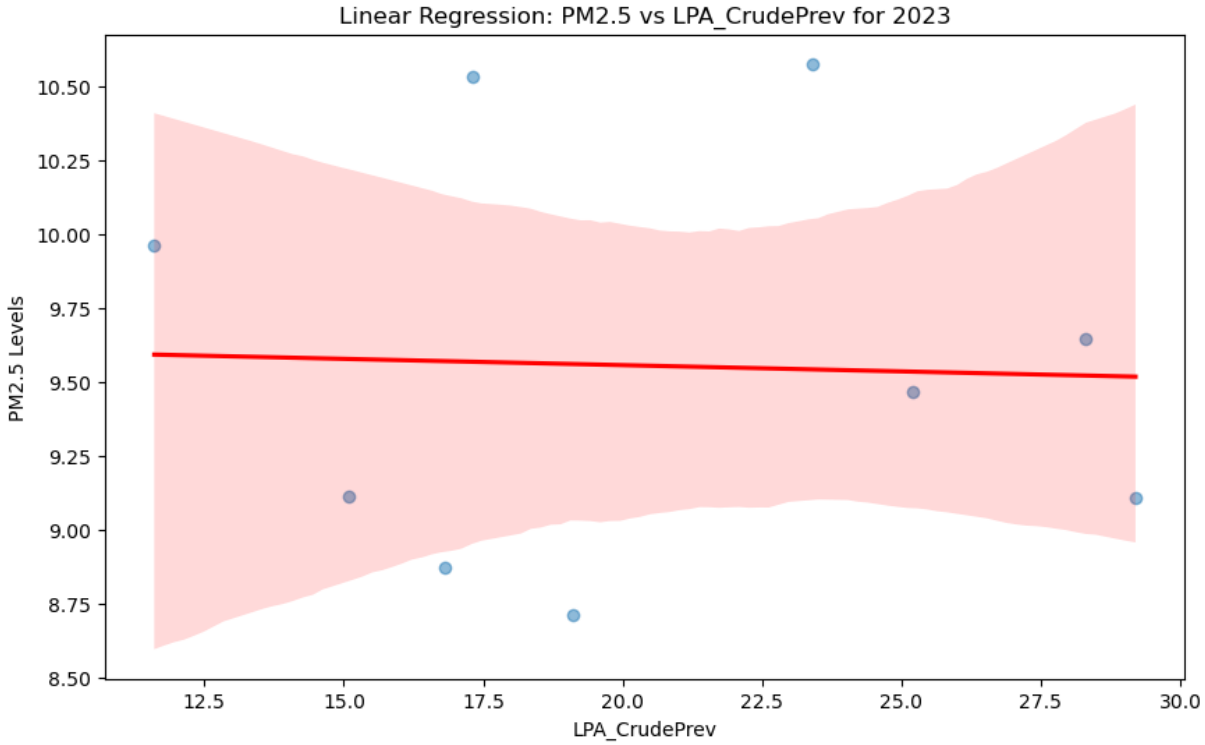

Linear Regression: PM2.5 vs DEPRESSION_CrudePrev for 2023
PM2.5 Levels
DEPRESSION_CrudePrev
10.5
10.0
9.5
9.0
8.5
19
20
21
22
23

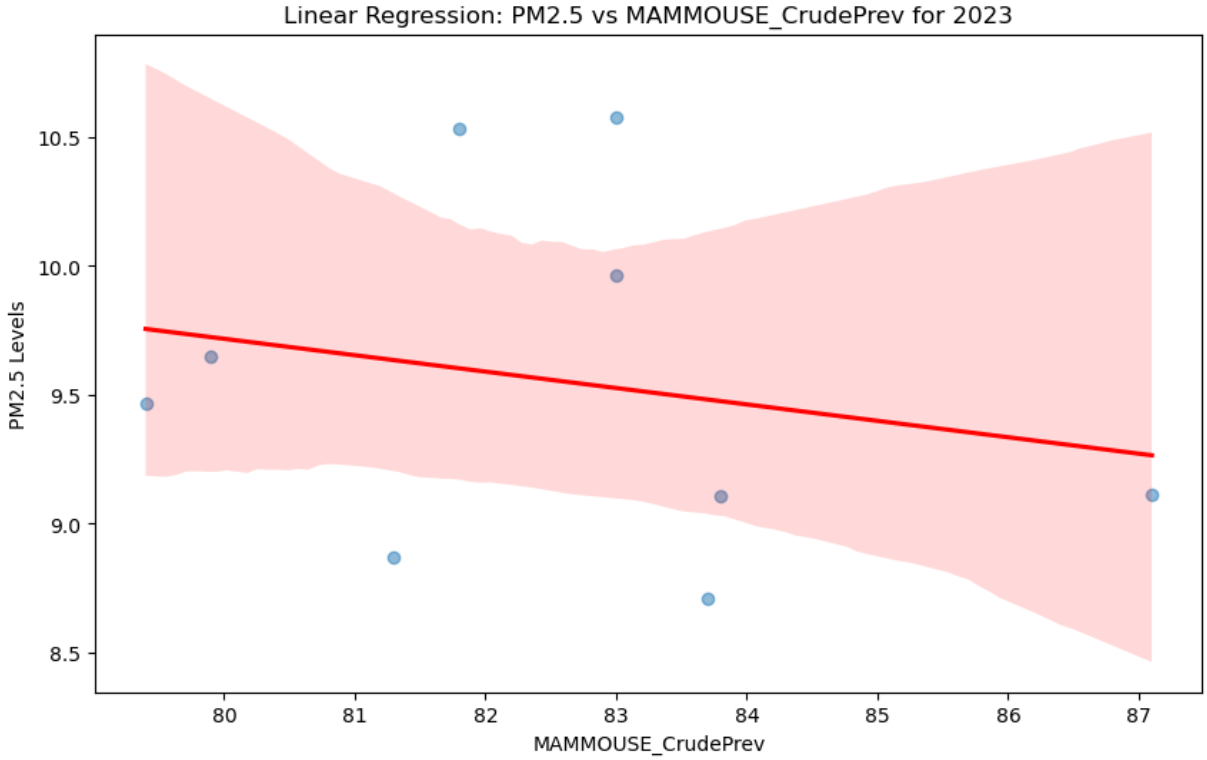

Linear Regression: PM2.5 vs DIABETES_CrudePrev for 2023
PM2.5 Levels
10.5
10.0
9.5
9.0
4
6
8
10
12
DIABETES_CrudePrev

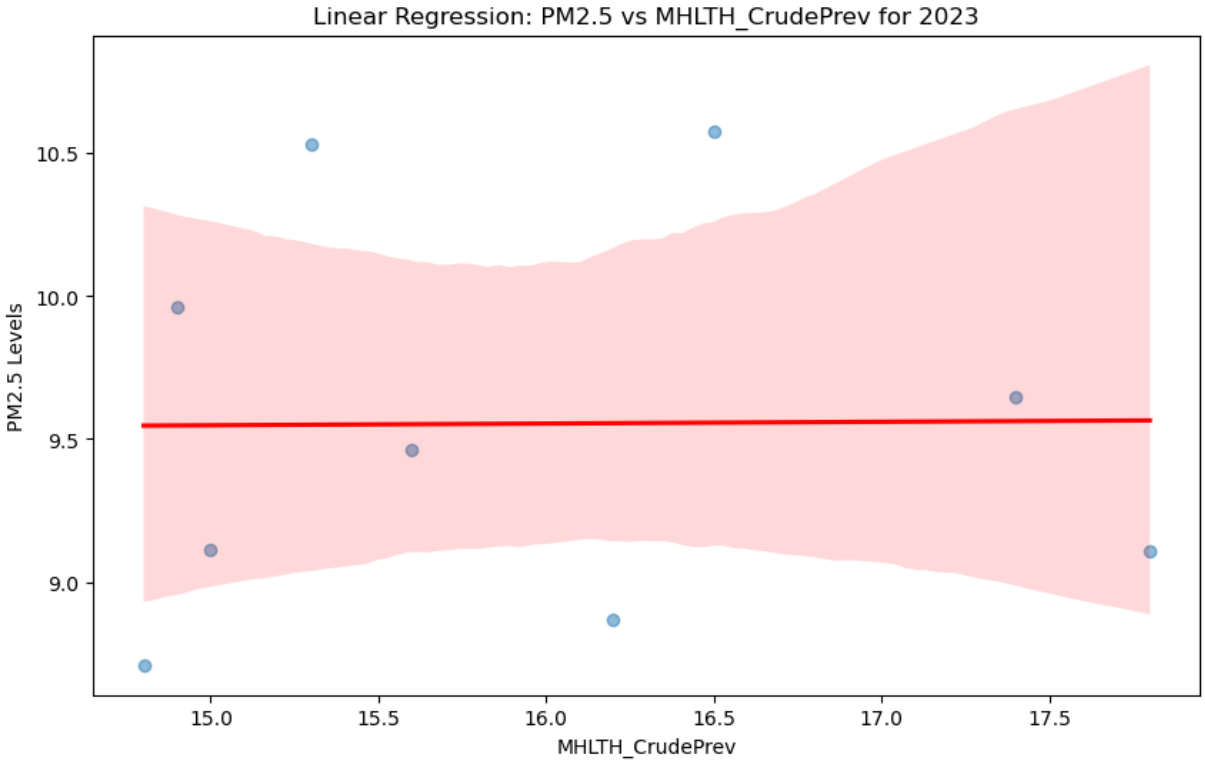

Linear Regression: PM2.5 vs GHLTH_CrudePrev for 2023
PM2.5 Levels
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
6
8
10
12
14
16
18
GHLTH_CrudePrev

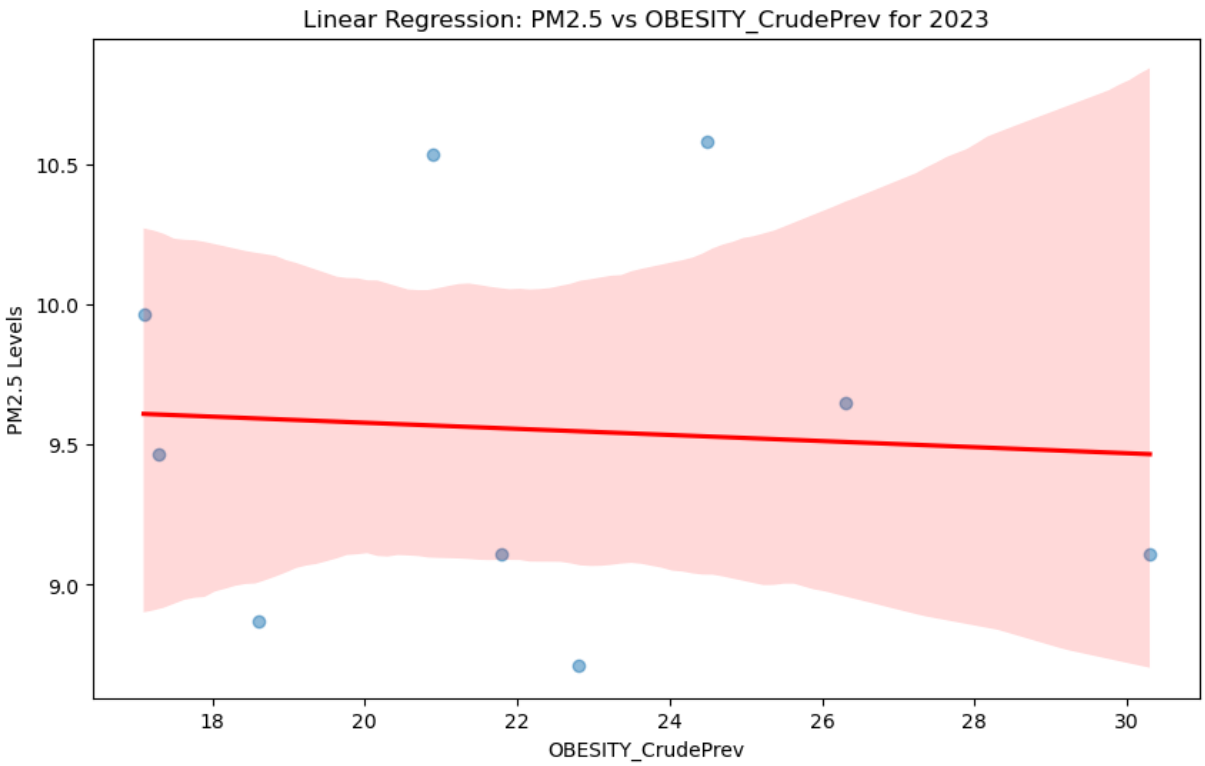

Linear Regression: PM2.5 vs HIGHCHOL_CrudePrev for 2023
PM2.5 Levels
HIGHCHOL_CrudePrev
10.75
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
22
24
26
28
30

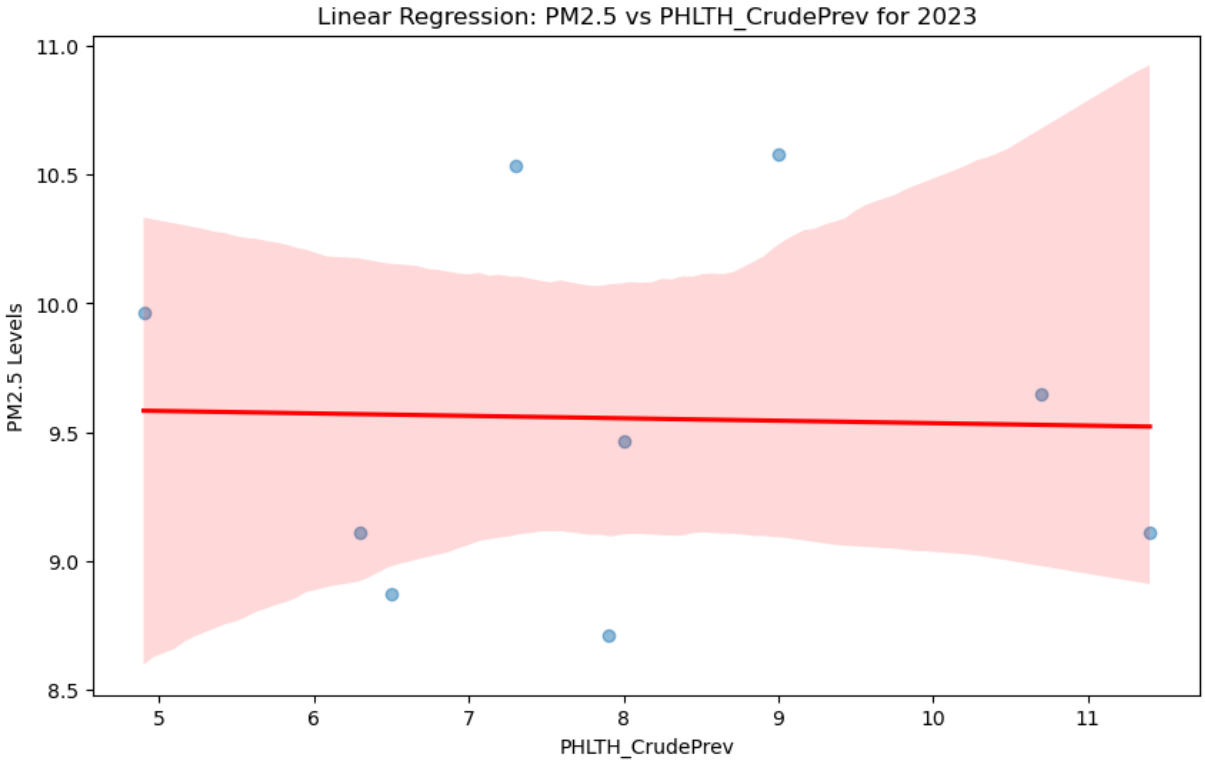

Linear Regression: PM2.5 vs KIDNEY_CrudePrev for 2023
PM2.5 Levels
KIDNEY_CrudePrev
10.5
10.0
9.5
9.0
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25

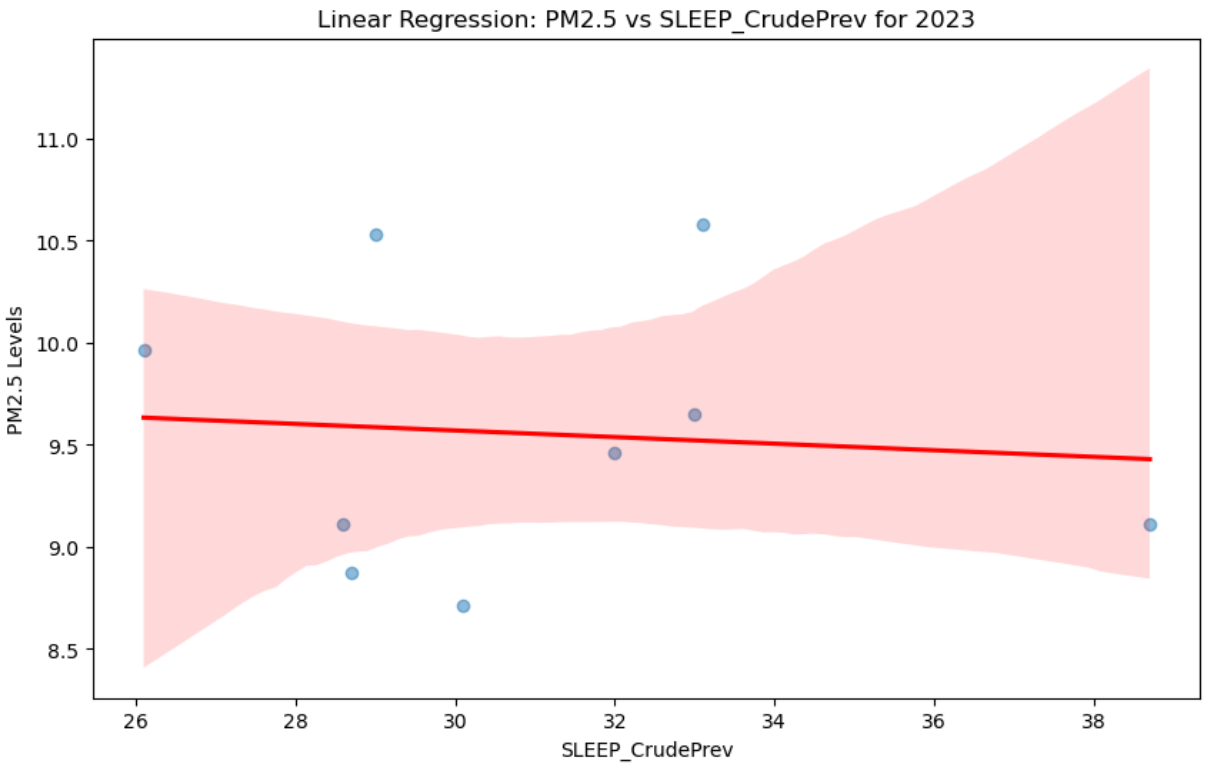

Linear Regression: PM2.5 vs LPA_CrudePrev for 2023
PM2.5 Levels
LPA_CrudePrev
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
8.50
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

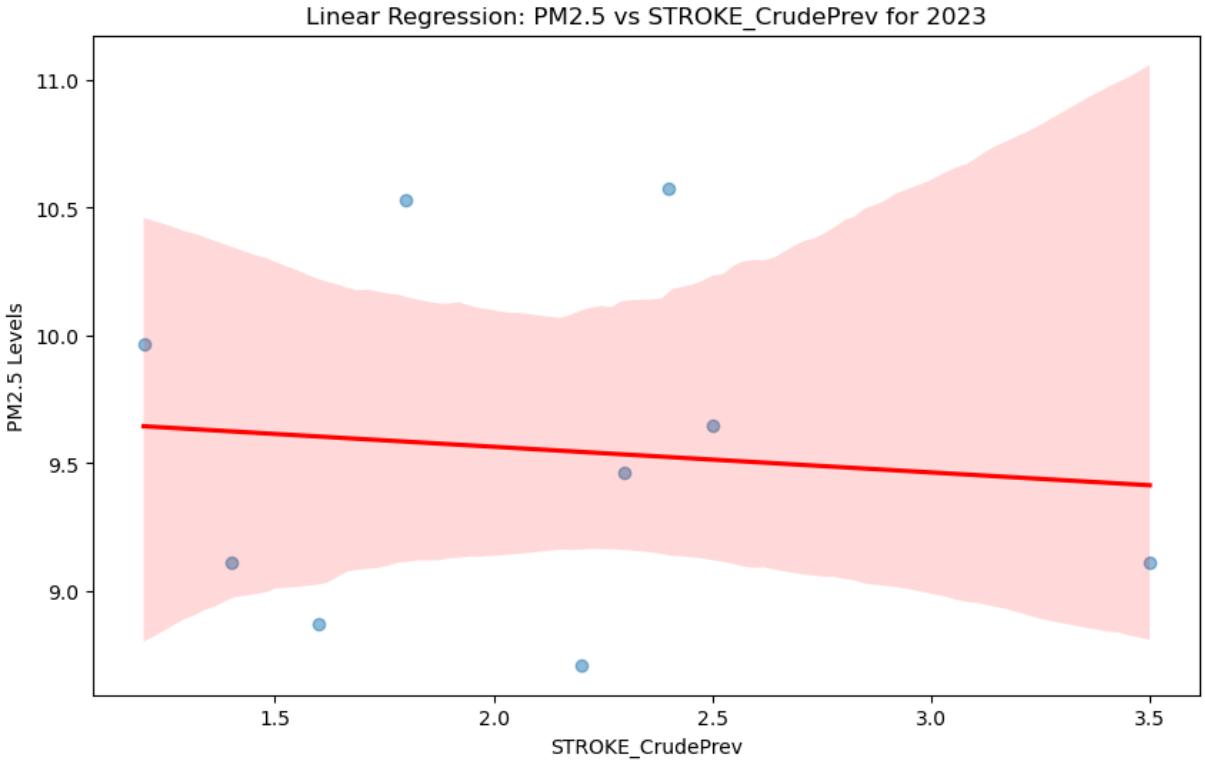

Linear Regression: PM2.5 vs MAMMOUSE_CrudePrev for 2023
PM2.5 Levels
MAMMOUSE_CrudePrev
10.5
10.0
9.5
9.0
8.5
80
81
82
83
84
85
86
87

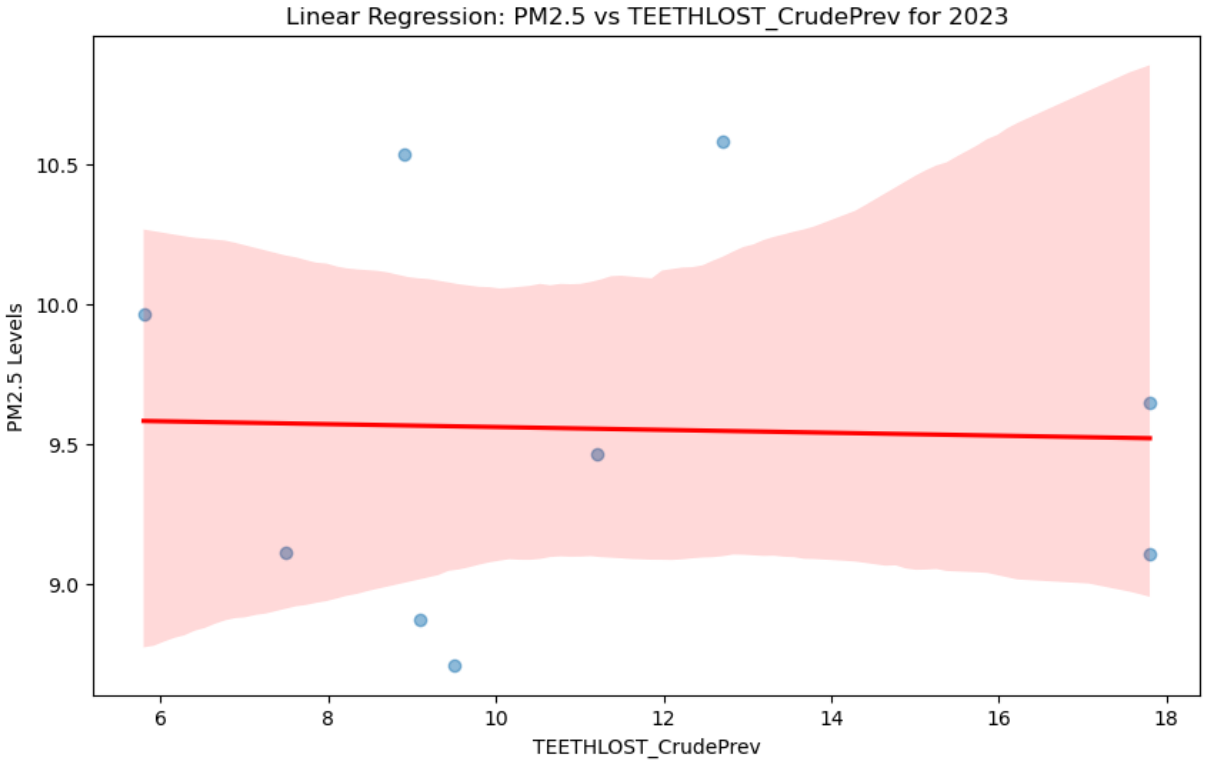

Linear Regression: PM2.5 vs MHLTH_CrudePrev for 2023
PM2.5 Levels
10.5
10.0
9.5
9.0
15.0
15.5
16.0
16.5
17.0
17.5
MHLTH_CrudePrev

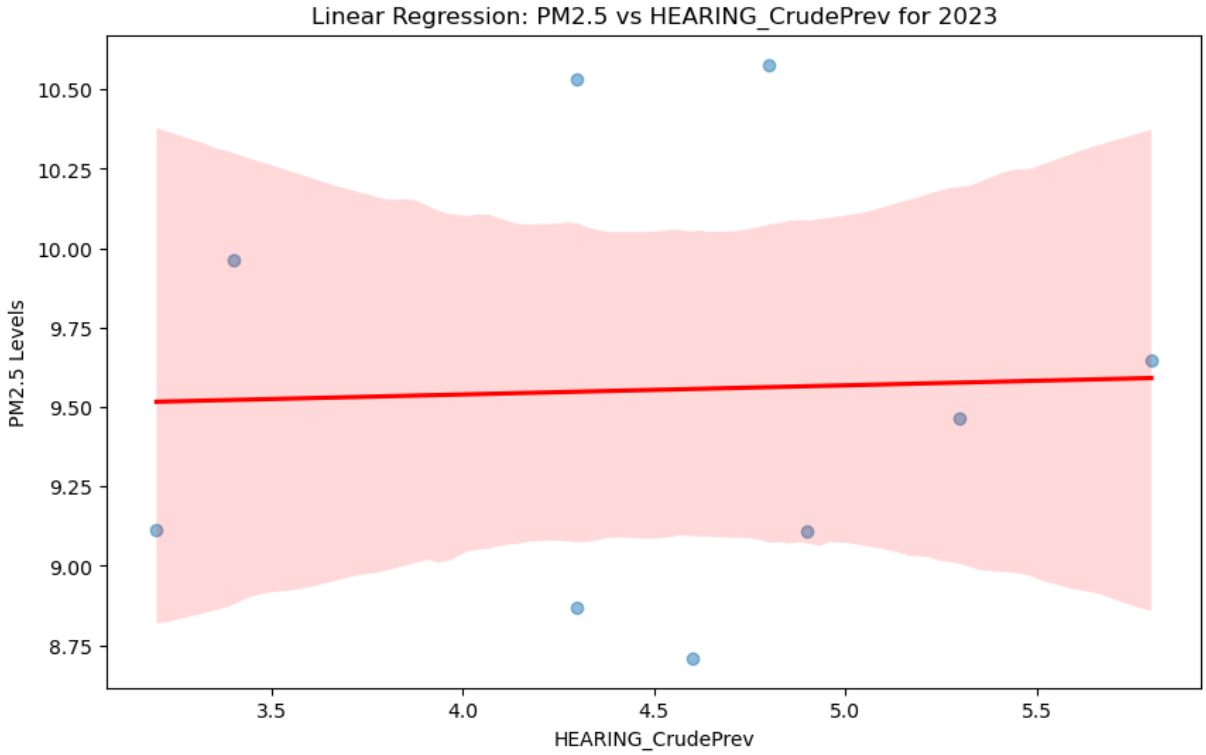

Linear Regression: PM2.5 vs OBESITY_CrudePrev for 2023
PM2.5 Levels
10.5
10.0
9.5
9.0
18
20
22
24
26
28
30
OBESITY_CrudePrev

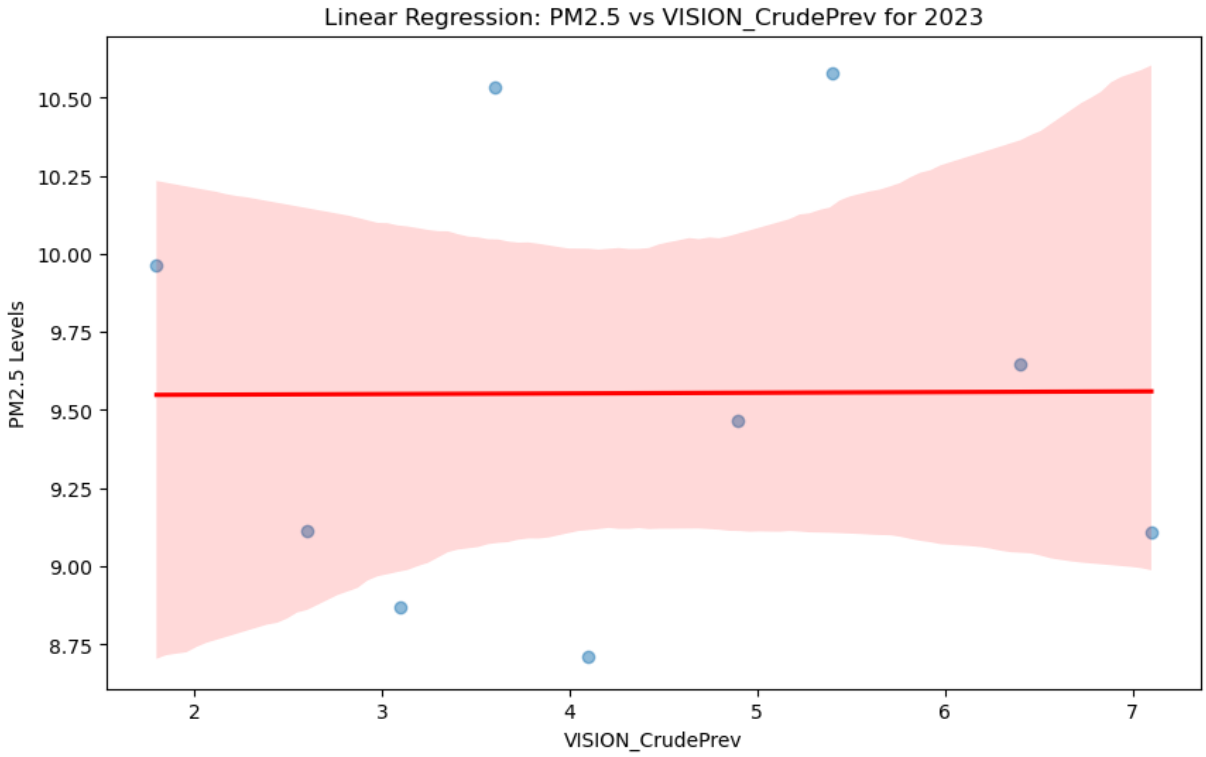

Linear Regression: PM2.5 vs PHLTH_CrudePrev for 2023
PM2.5 Levels
PHLTH_CrudePrev
11.0
10.5
10.0
9.5
9.0
8.5
5
6
7
8
9
10
11

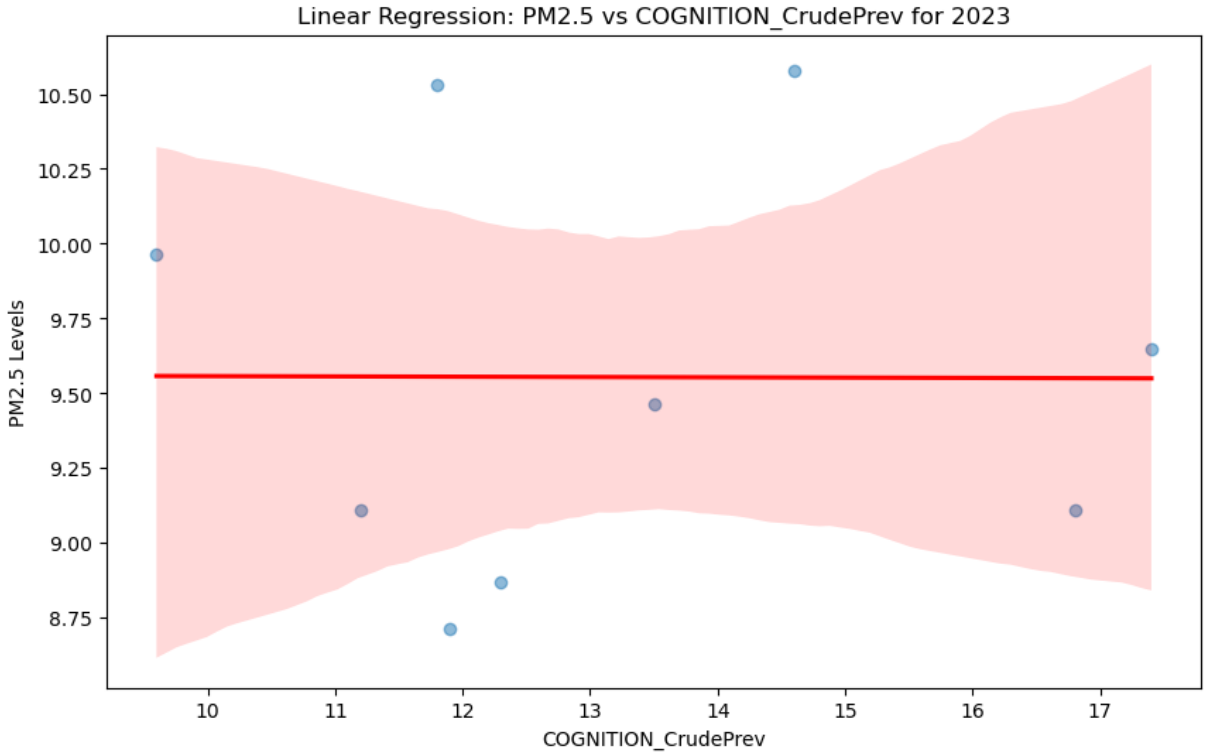

Linear Regression: PM2.5 vs SLEEP_CrudePrev for 2023
PM2.5 Levels
SLEEP_CrudePrev
11.0
10.5
10.0
9.5
9.0
8.5
26
28
30
32
34
36
38

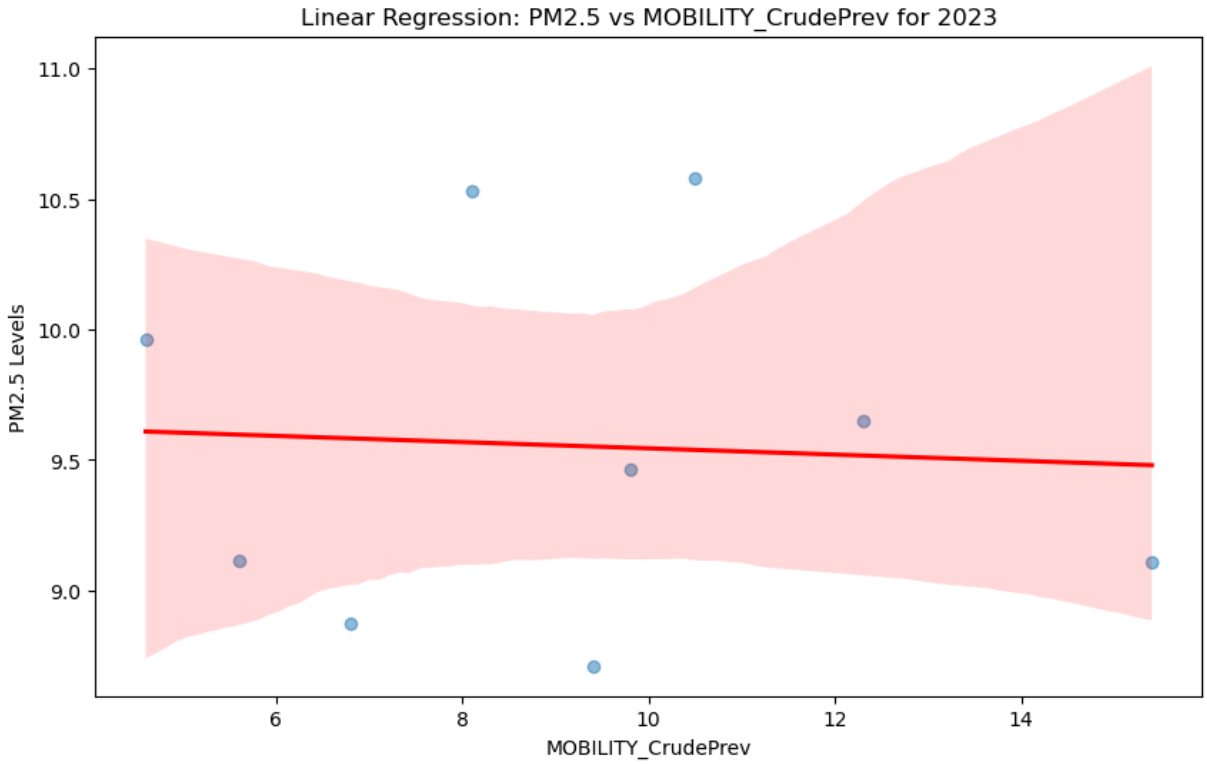

Linear Regression: PM2.5 vs STROKE_CrudePrev for 2023
PM2.5 Levels
STROKE_CrudePrev
11.0
10.5
10.0
9.5
9.0
1.5
2.0
2.5
3.0
3.5

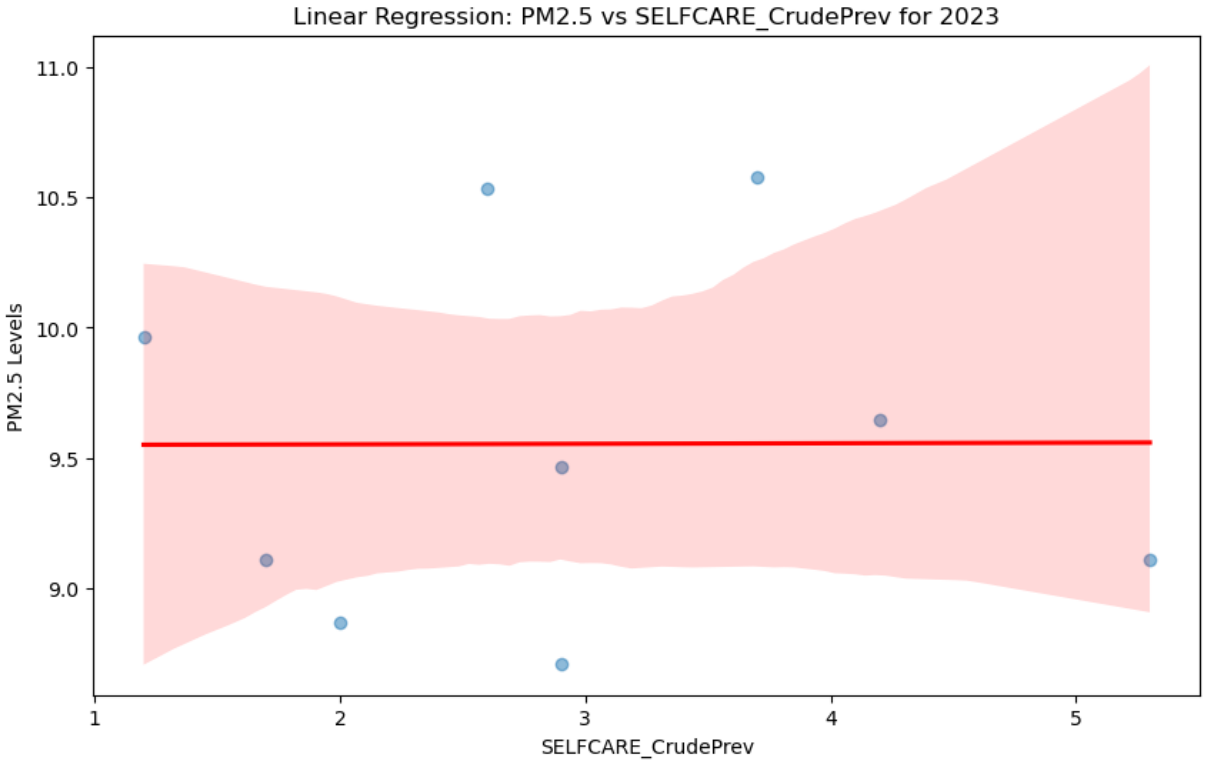

Linear Regression: PM2.5 vs TEETHLOST_CrudePrev for 2023
PM2.5 Levels
TEETHLOST_CrudePrev
10.5
10.0
9.5
9.0
6
8
10
12
14
16
18

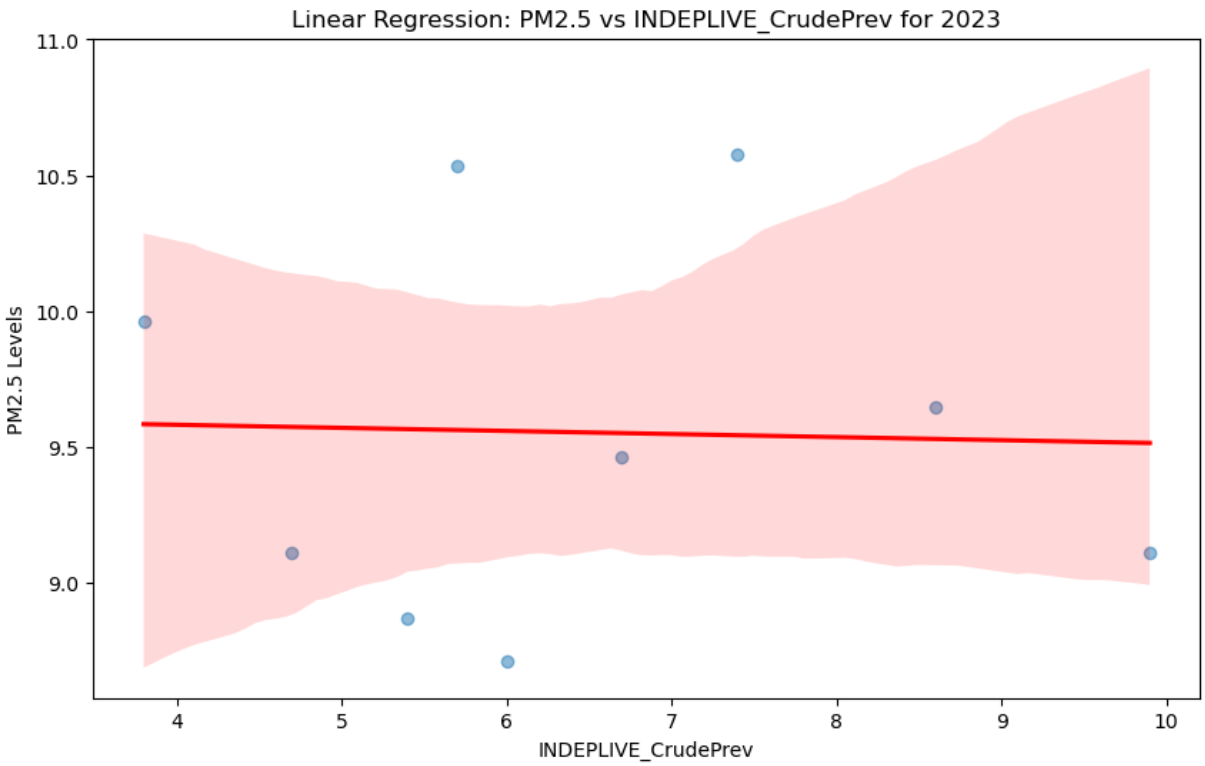

Linear Regression: PM2.5 vs HEARING_CrudePrev for 2023
PM2.5 Levels
HEARING_CrudePrev
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
3.5
4.0
4.5
5.0
5.5

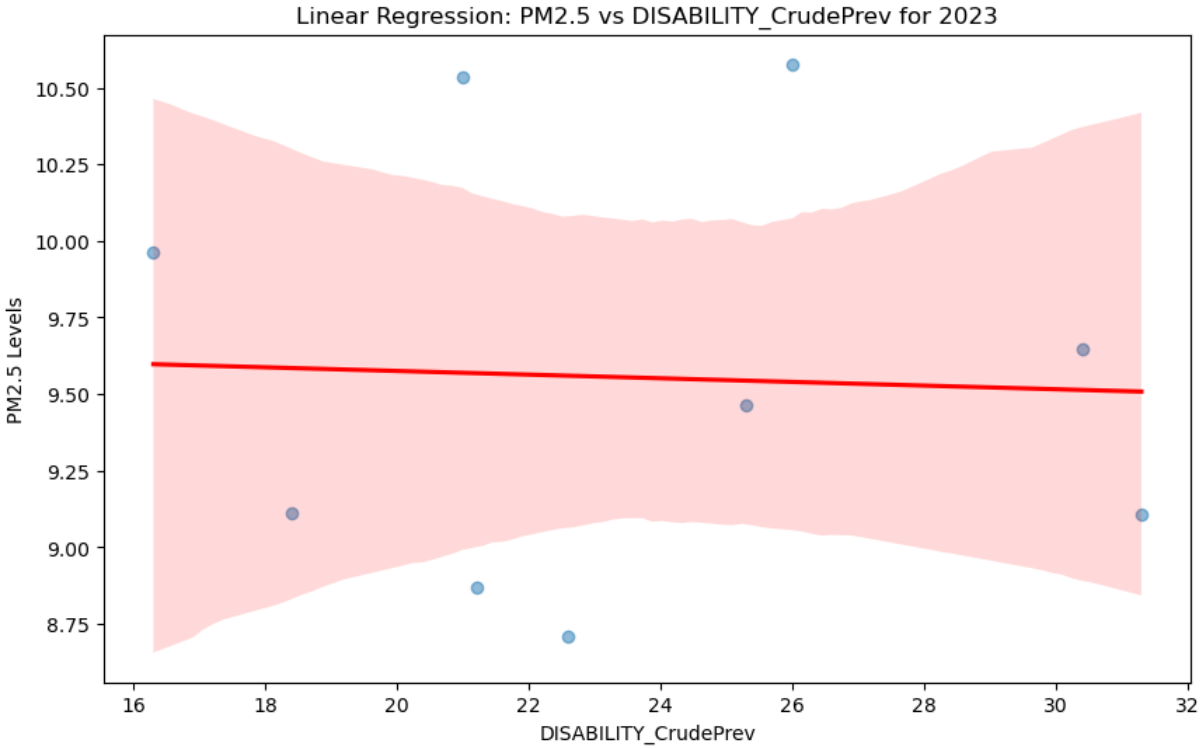

Linear Regression: PM2.5 vs VISION_CrudePrev for 2023
PM2.5 Levels
VISION_CrudePrev
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
2
3
4
5
6
7


Linear Regression: PM2.5 vs COGNITION_CrudePrev for 2023
PM2.5 Levels
10.50
10.25
10.00
9.75
9.50
9.25
9.00
8.75
10
11
12
13
14
15
16
17
COGNITION_CrudePrev

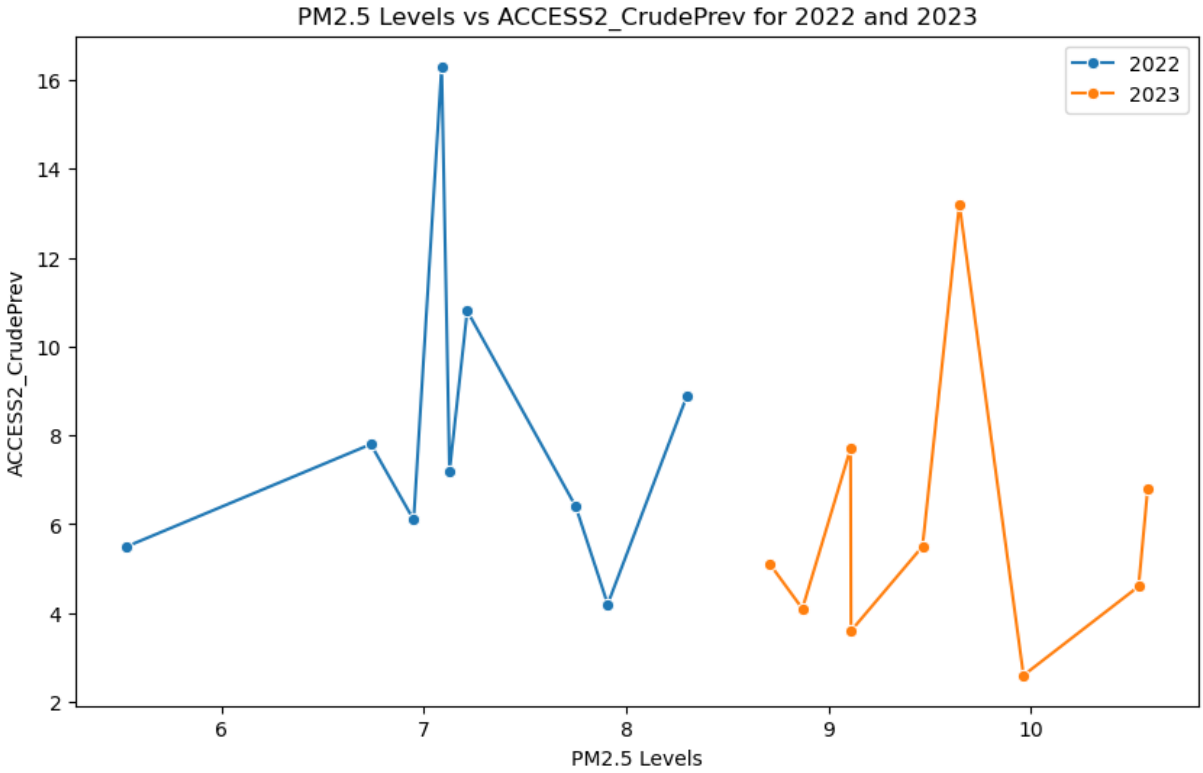

Linear Regression: PM2.5 vs MOBILITY_CrudePrev for 2023
PM2.5 Levels
11.0
10.5
10.0
9.5
9.0
6
8
10
12
14
MOBILITY_CrudePrev

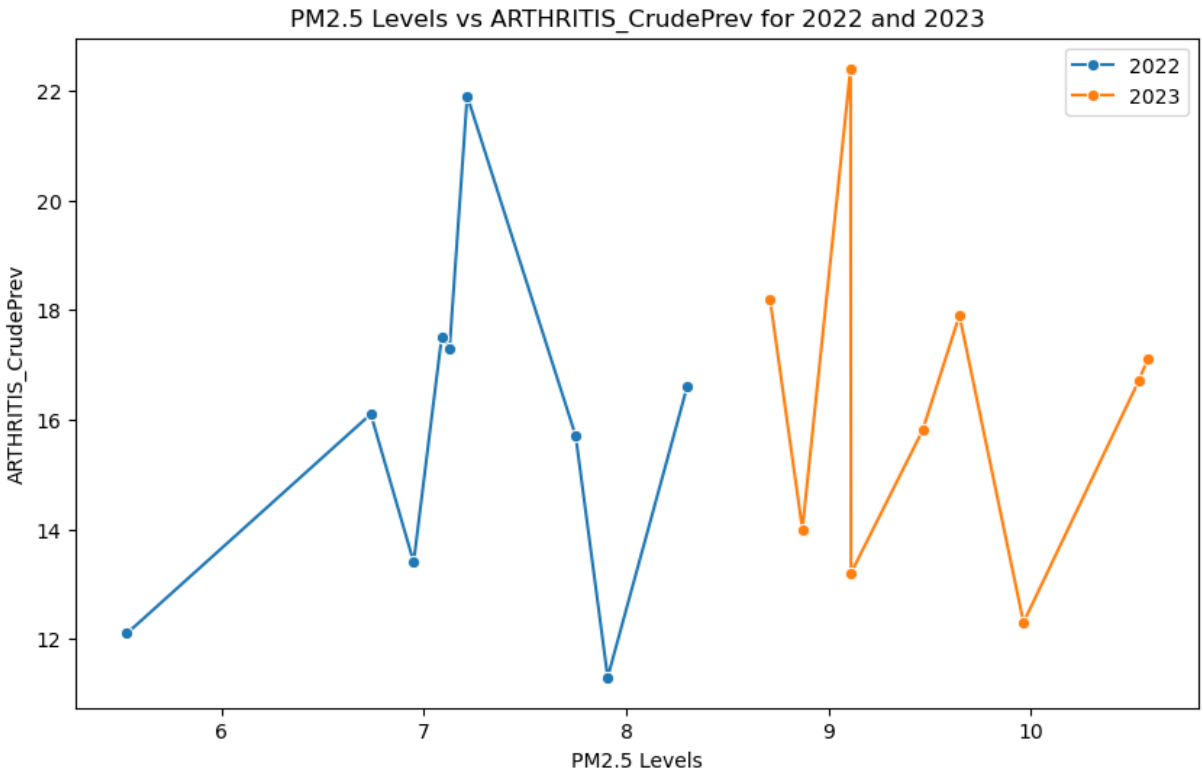

Linear Regression: PM2.5 vs SELFCARE_CrudePrev for 2023
PM2.5 Levels
SELFCARE_CrudePrev
11.0
10.5
10.0
9.5
9.0
1
2
3
4
5

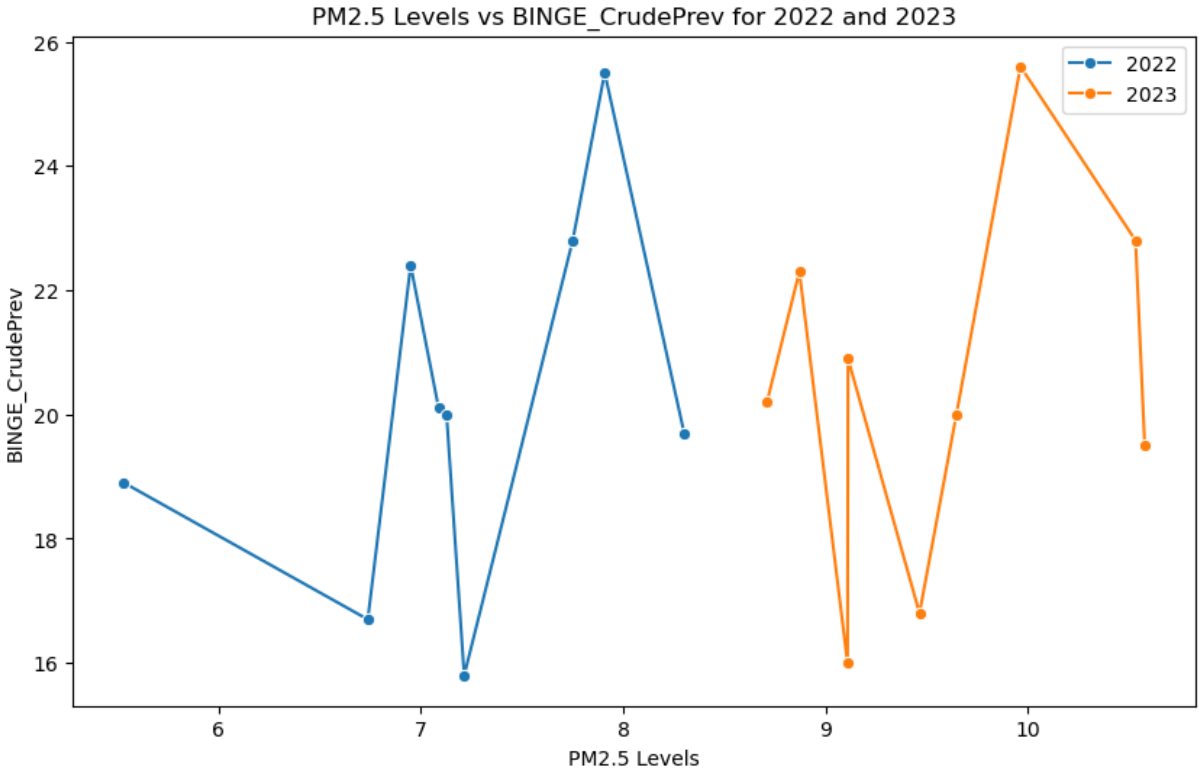

Linear Regression: PM2.5 vs INDEPLIVE_CrudePrev for 2023
PM2.5 Levels
INDEPLIVE_CrudePrev
11.0
10.5
10.0
9.5
9.0
4
5
6
7
8
9
10

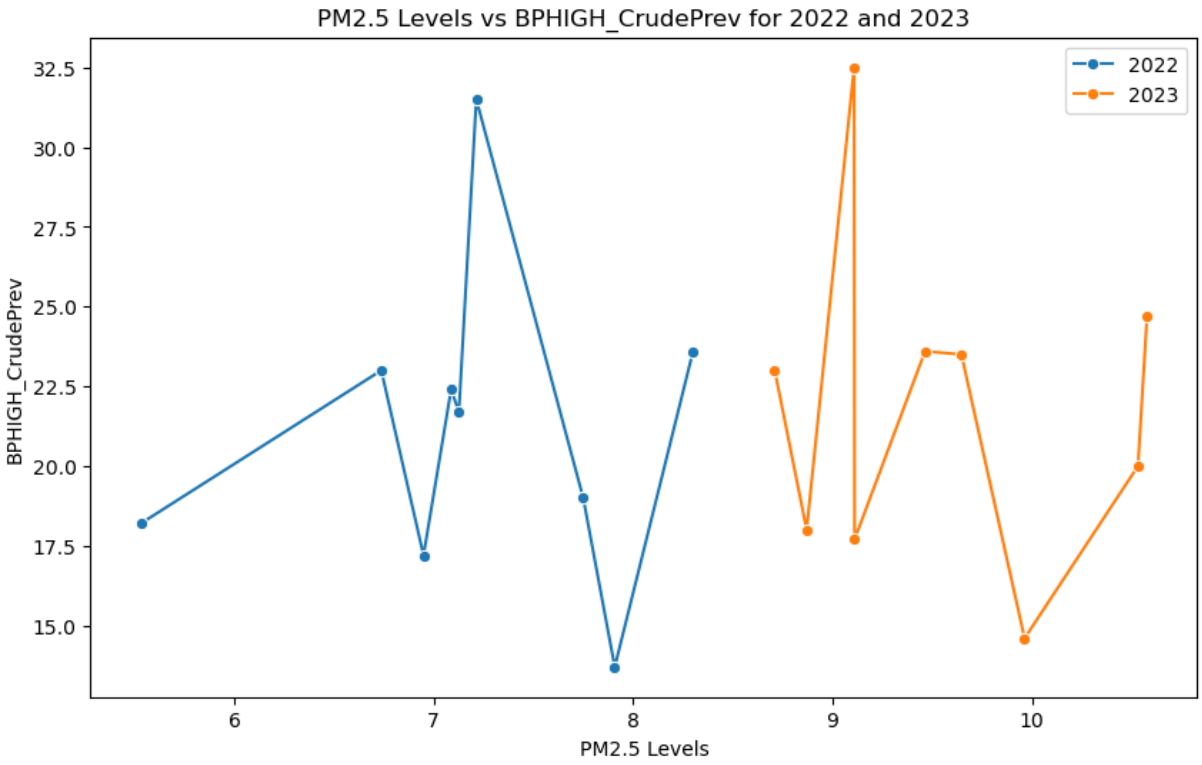


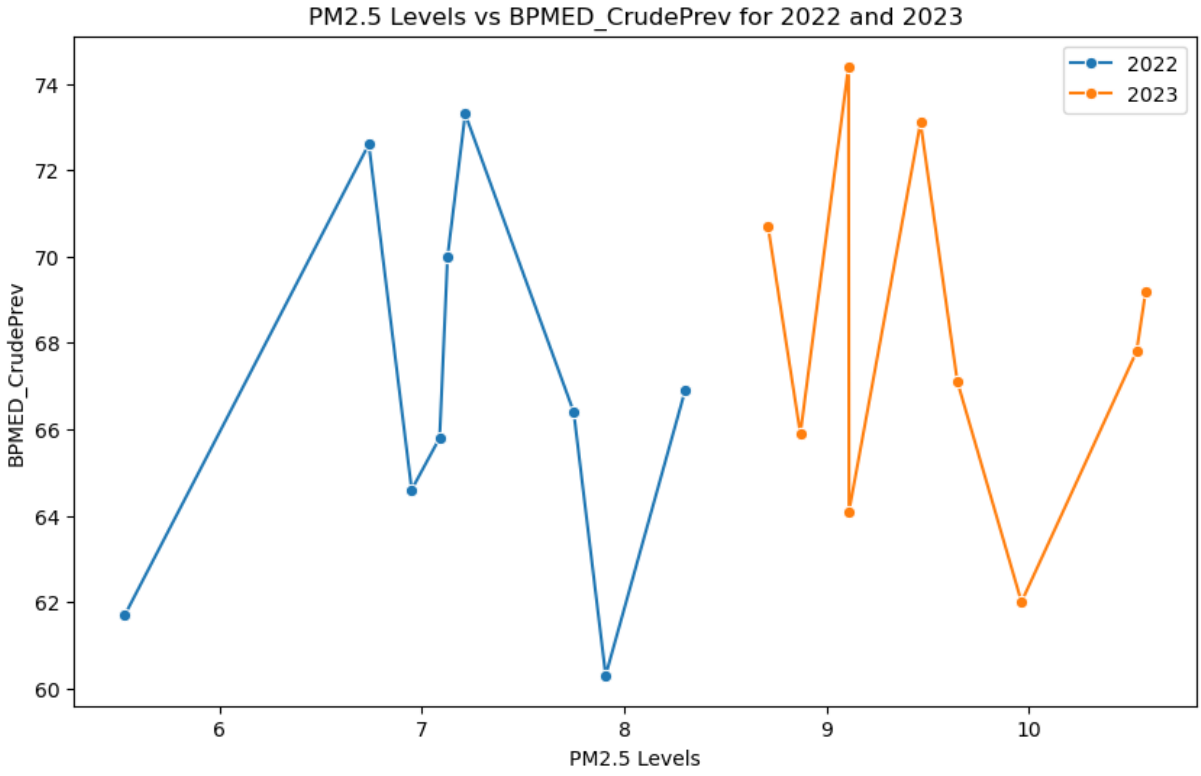
In [11]:
```
health_data_2023_renamed = health_data_2023.rename(columns=lambda x: x.repla

# Merging the datasets again with corrected column names
merged_health_data_corrected = pd.merge(health_data_2022, health_data_2023_r

# Plotting line graphs for each crude rate against PM2.5 level for 2022 and
for column in crude_columns_2022:
    if 'Crude' in column:
        plt.figure(figsize=(10, 6))

        # Plotting for 2022
        sns.lineplot(x=merged_health_data_corrected['PM2.5_2022'], y=merged_

        # Plotting for 2023
        sns.lineplot(x=merged_health_data_corrected['PM2.5_2023'], y=merged_

        plt.title(f'PM2.5 Levels vs {column} for 2022 and 2023')
        plt.xlabel('PM2.5 Levels')
        plt.ylabel(column)
        plt.legend()
        plt.show()
```

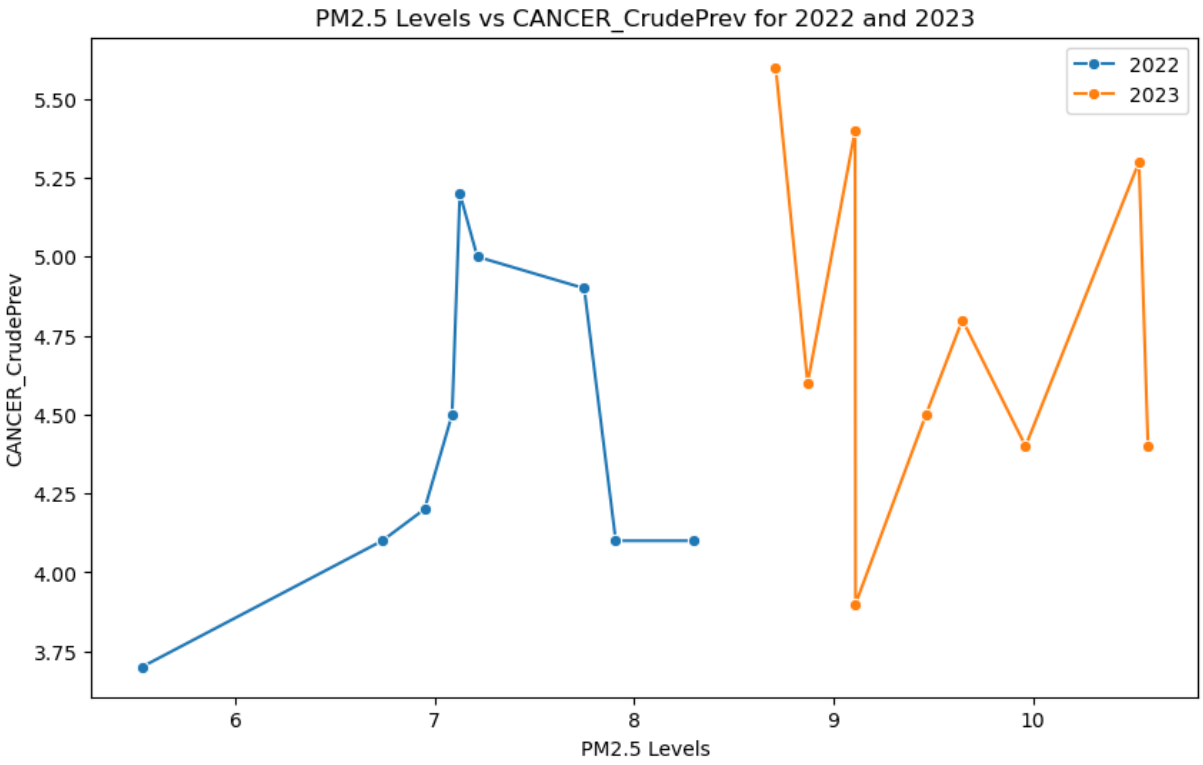

PM2.5 Levels vs ACCESS2_CrudePrev for 2022 and 2023
2022
2023
ACCESS2_CrudePrev
PM2.5 Levels
2
4
6
8
10
12
14
16
6
7
8
9
10

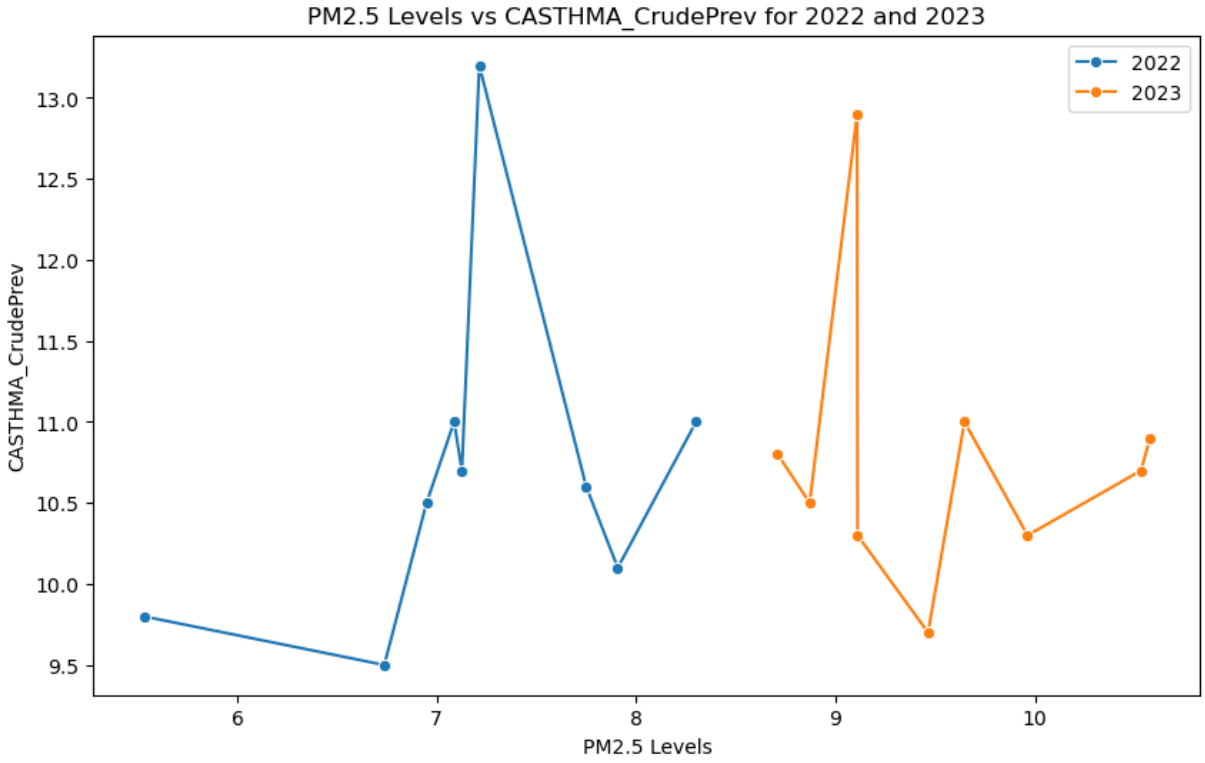

PM2.5 Levels vs ARTHRITIS_CrudePrev for 2022 and 2023
2022
2023
ARTHRITIS_CrudePrev
PM2.5 Levels
12
14
16
18
20
22
6
7
8
9
10

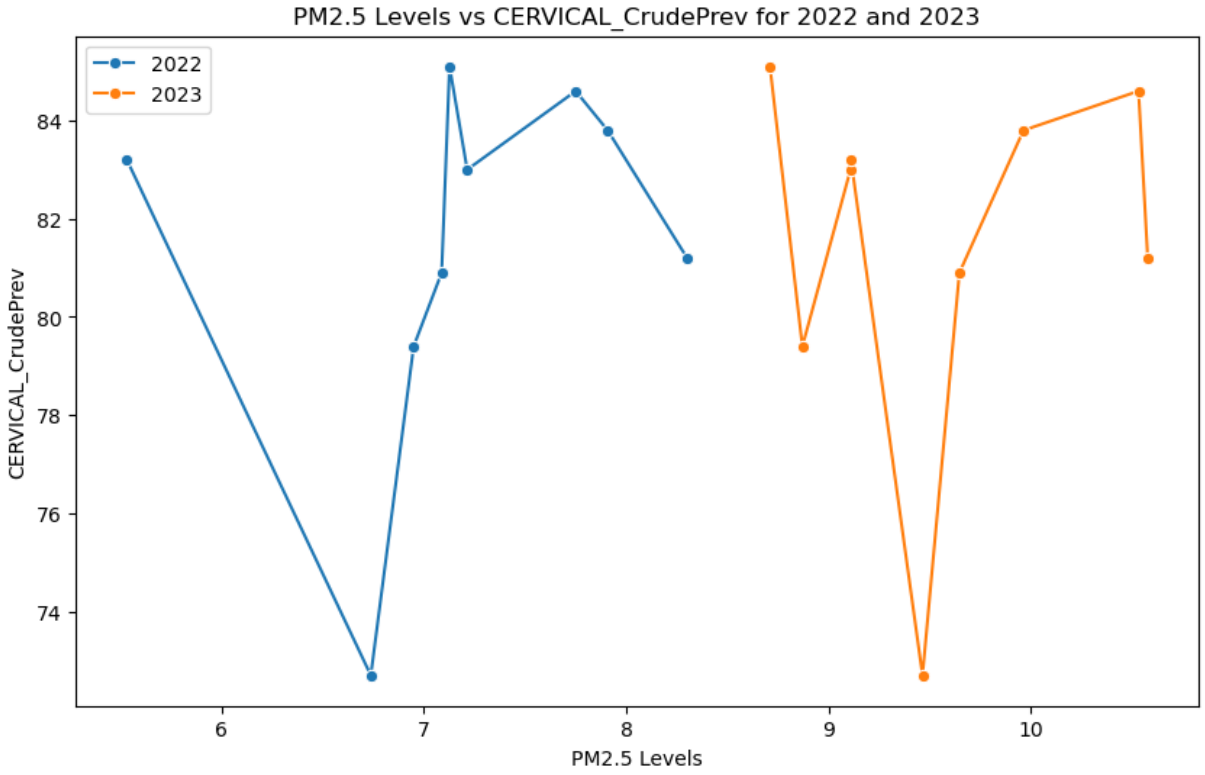

PM2.5 Levels vs BINGE_CrudePrev for 2022 and 2023
2022
2023
BINGE_CrudePrev
PM2.5 Levels
26
24
22
20
18
16
6
7
8
9
10

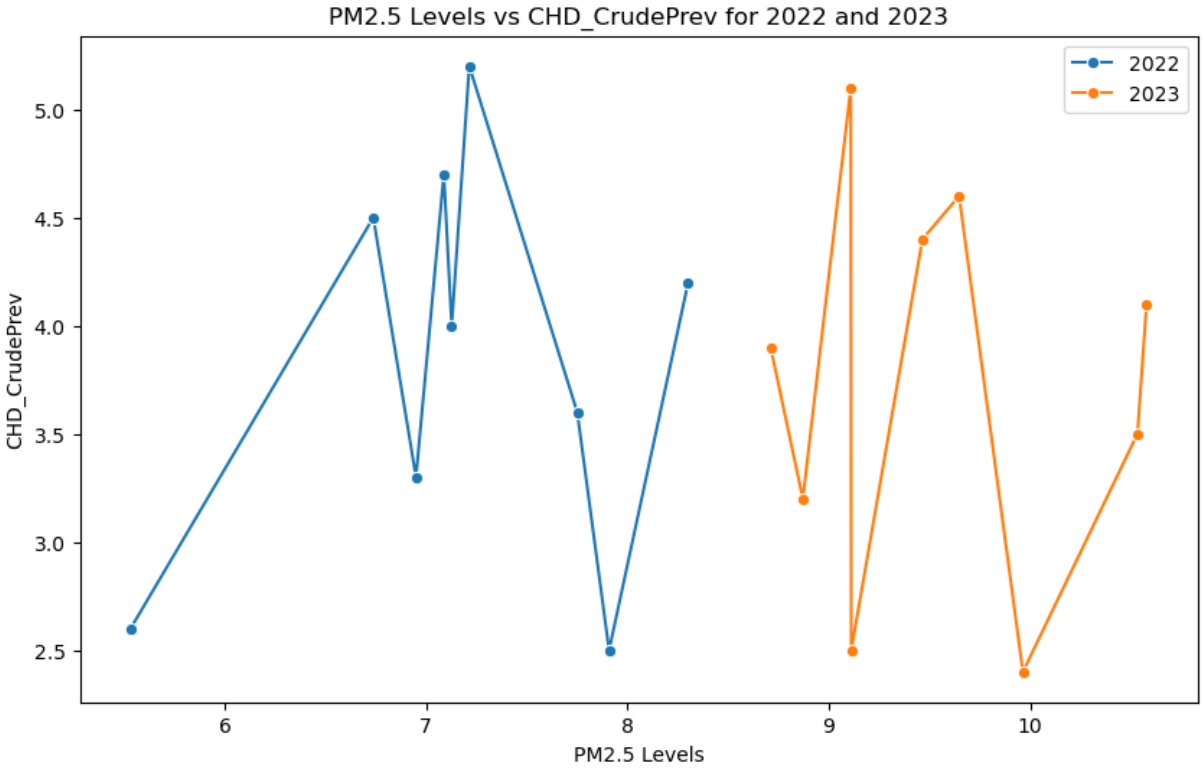

PM2.5 Levels vs BPHIGH_CrudePrev for 2022 and 2023
2022
2023
BPHIGH_CrudePrev
PM2.5 Levels
32.5
30.0
27.5
25.0
22.5
20.0
17.5
15.0
6
7
8
9
10

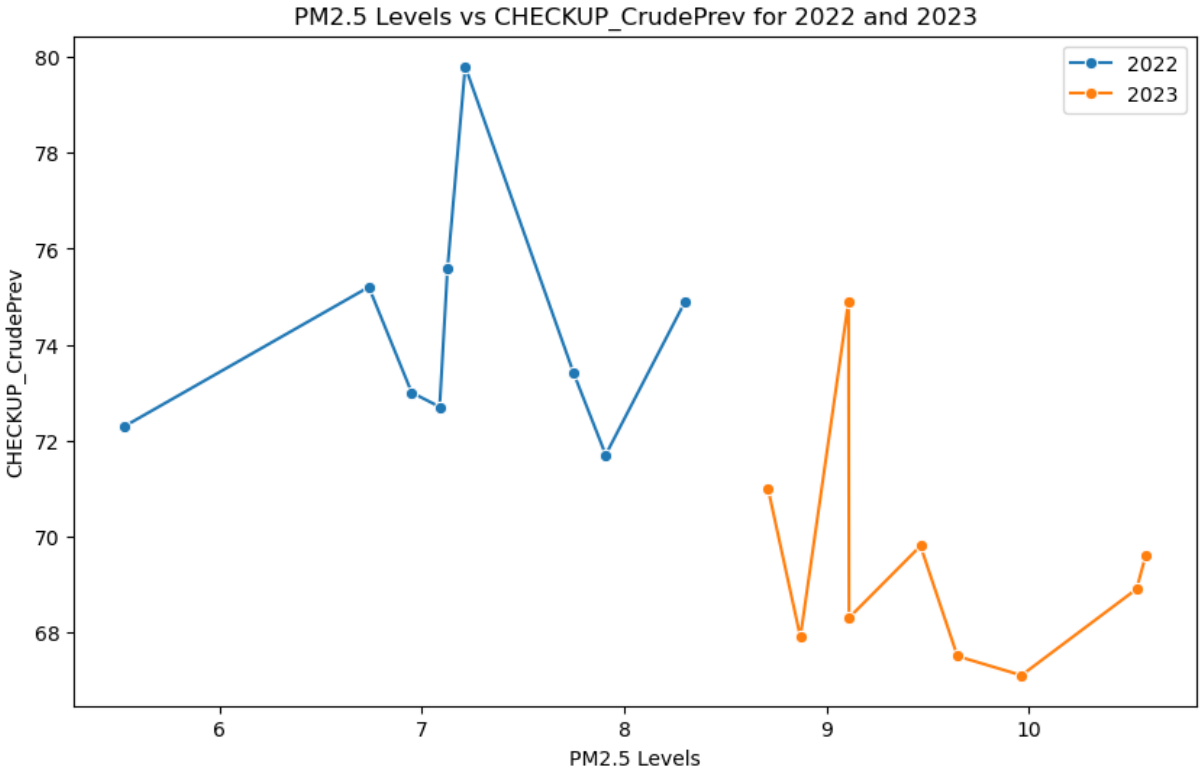

PM2.5 Levels vs BPMED_CrudePrev for 2022 and 2023
2022
2023
BPMED_CrudePrev
PM2.5 Levels
74
72
70
68
66
64
62
60
6
7
8
9
10


PM2.5 Levels vs CANCER_CrudePrev for 2022 and 2023
2022
2023
CANCER_CrudePrev
PM2.5 Levels
5.50
5.25
5.00
4.75
4.50
4.25
4.00
3.75
6
7
8
9
10

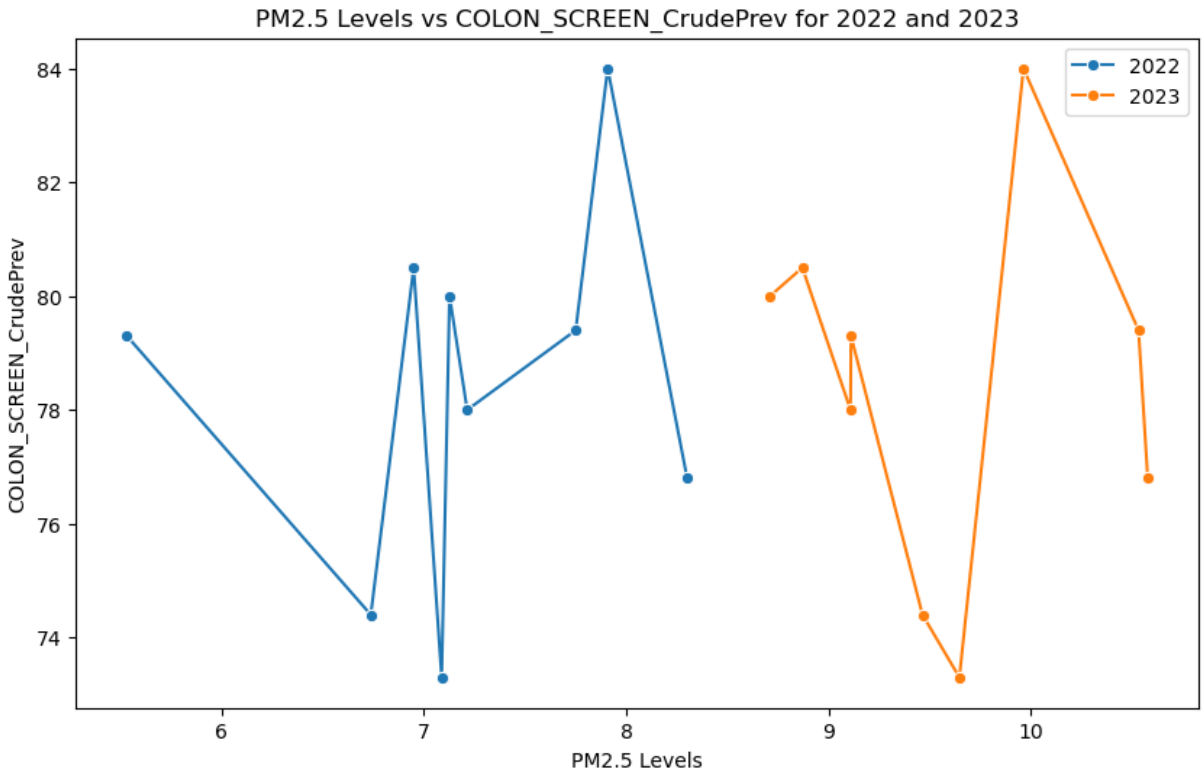

PM2.5 Levels vs CASTHMA_CrudePrev for 2022 and 2023
2022
2023
CASTHMA_CrudePrev
PM2.5 Levels
13.0
12.5
12.0
11.5
11.0
10.5
10.0
9.5
6
7
8
9
10

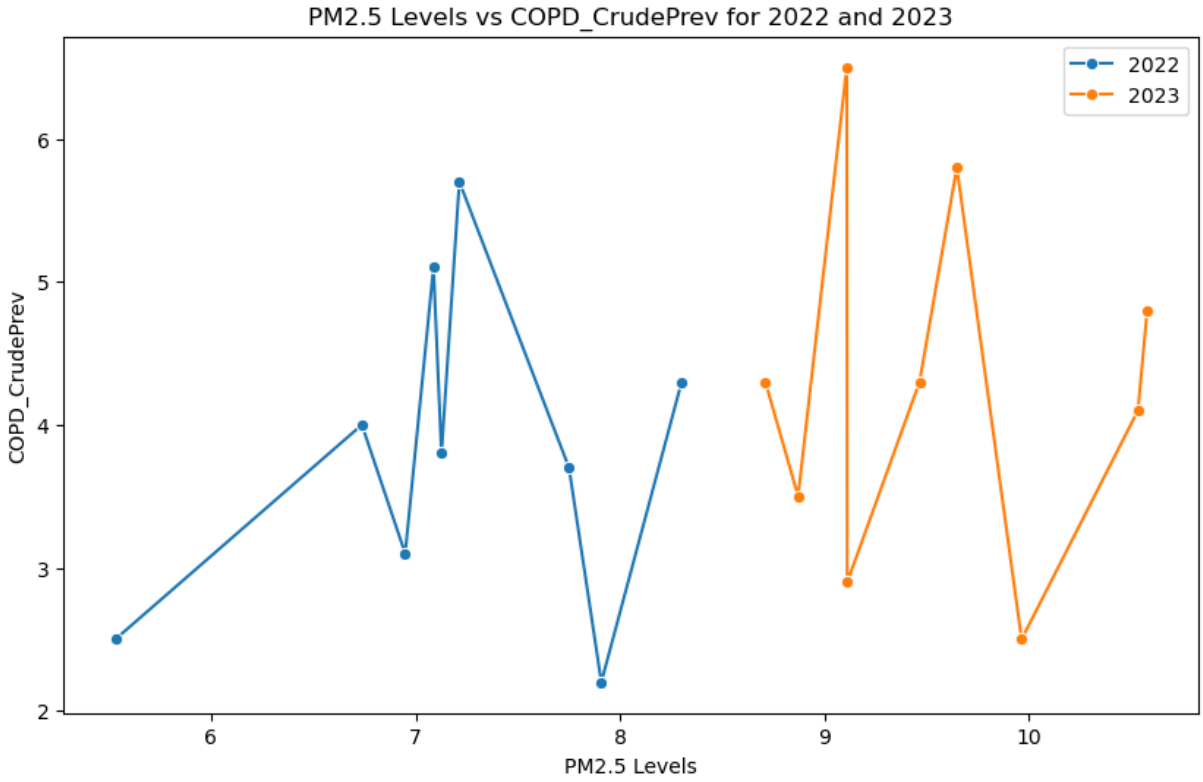

PM2.5 Levels vs CERVICAL_CrudePrev for 2022 and 2023
2022
2023
CERVICAL_CrudePrev
PM2.5 Levels
84
82
80
78
76
74
6
7
8
9
10

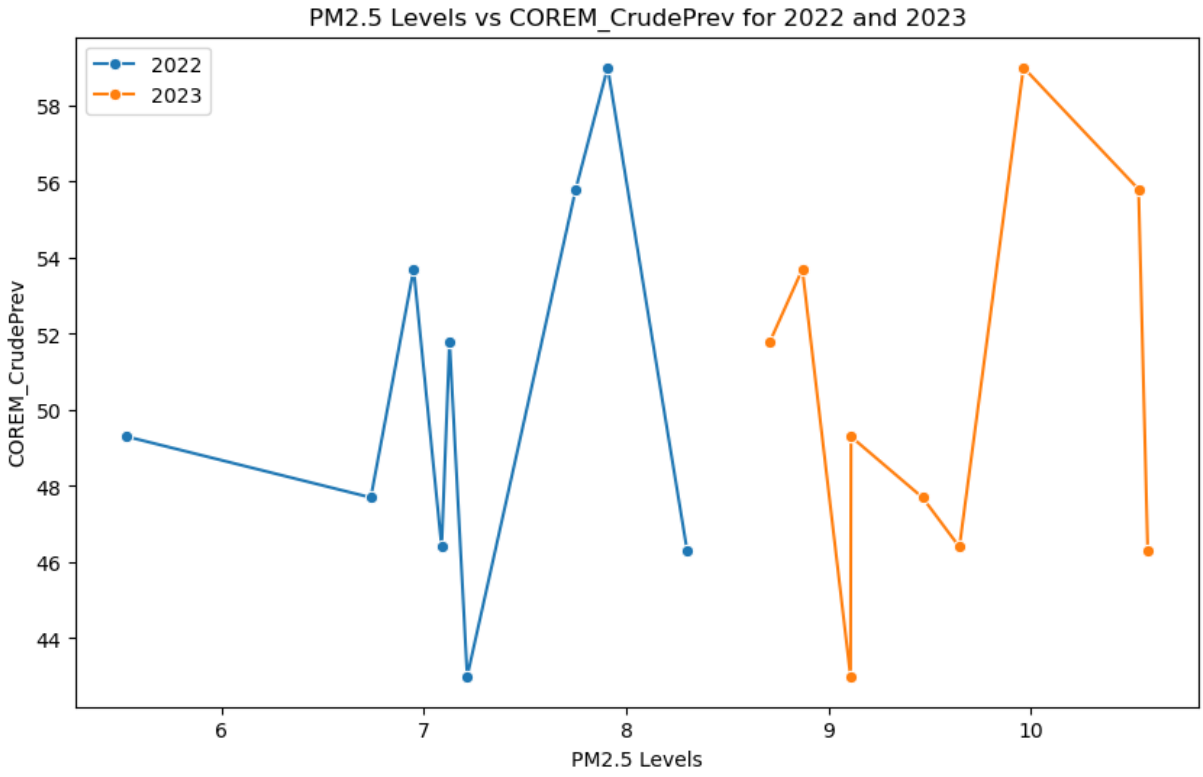

PM2.5 Levels vs CHD_CrudePrev for 2022 and 2023
2022
2023
CHD_CrudePrev
5.0
4.5
4.0
3.5
3.0
2.5
6
7
8
9
10
PM2.5 Levels

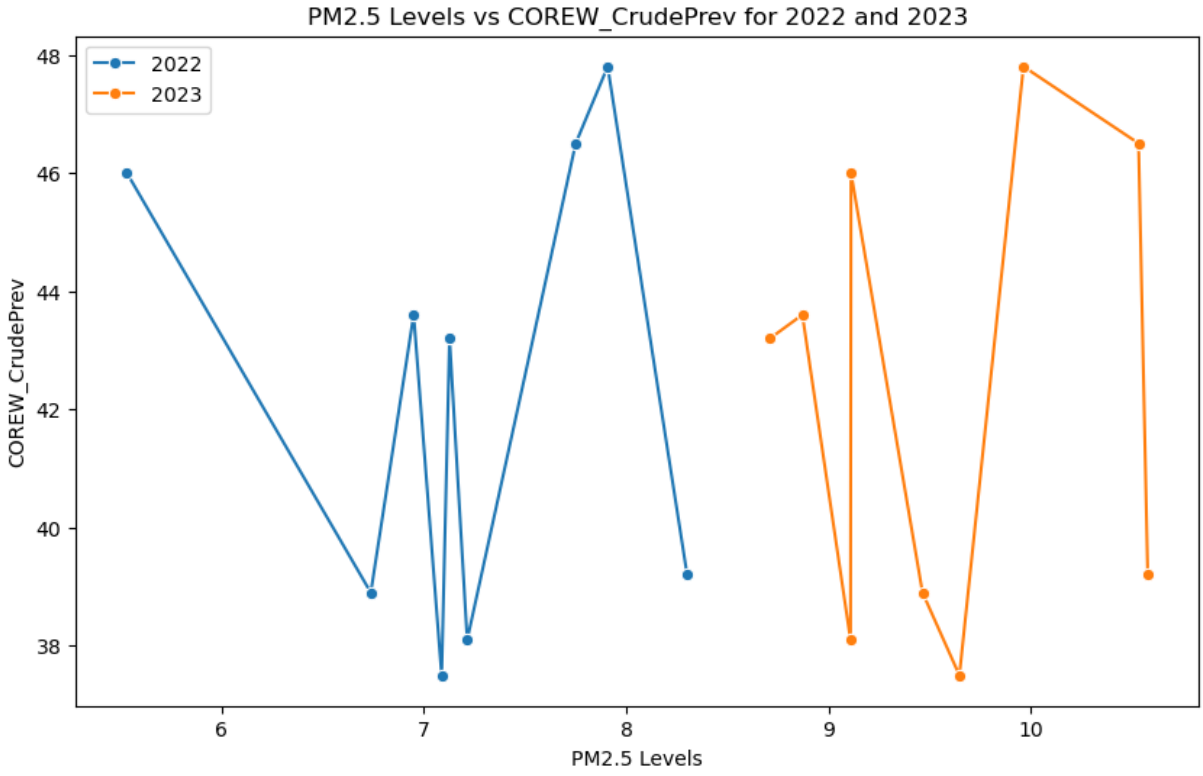

PM2.5 Levels vs CHECKUP_CrudePrev for 2022 and 2023
2022
2023
CHECKUP_CrudePrev
80
78
76
74
72
70
68
6
7
8
9
10
PM2.5 Levels

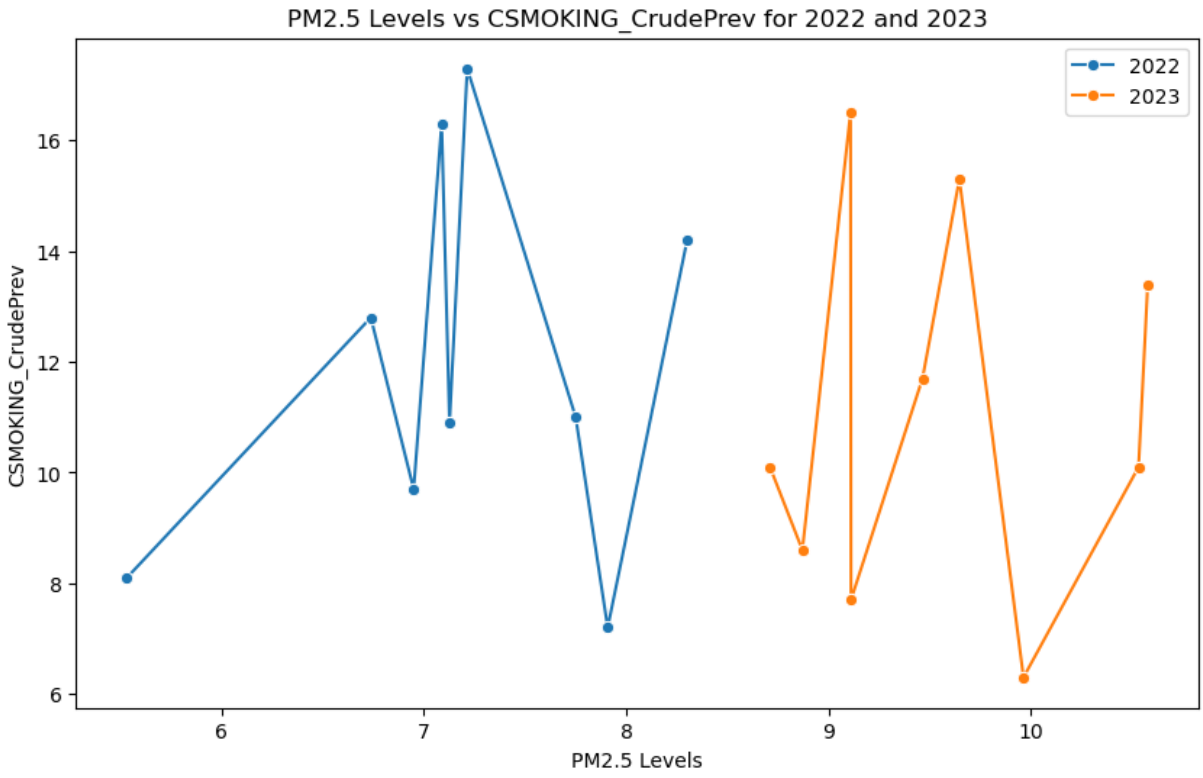

PM2.5 Levels vs CHOLSCREEN_CrudePrev for 2022 and 2023
2022
2023
CHOLSCREEN_CrudePrev
88
86
84
82
6
7
8
9
10
PM2.5 Levels

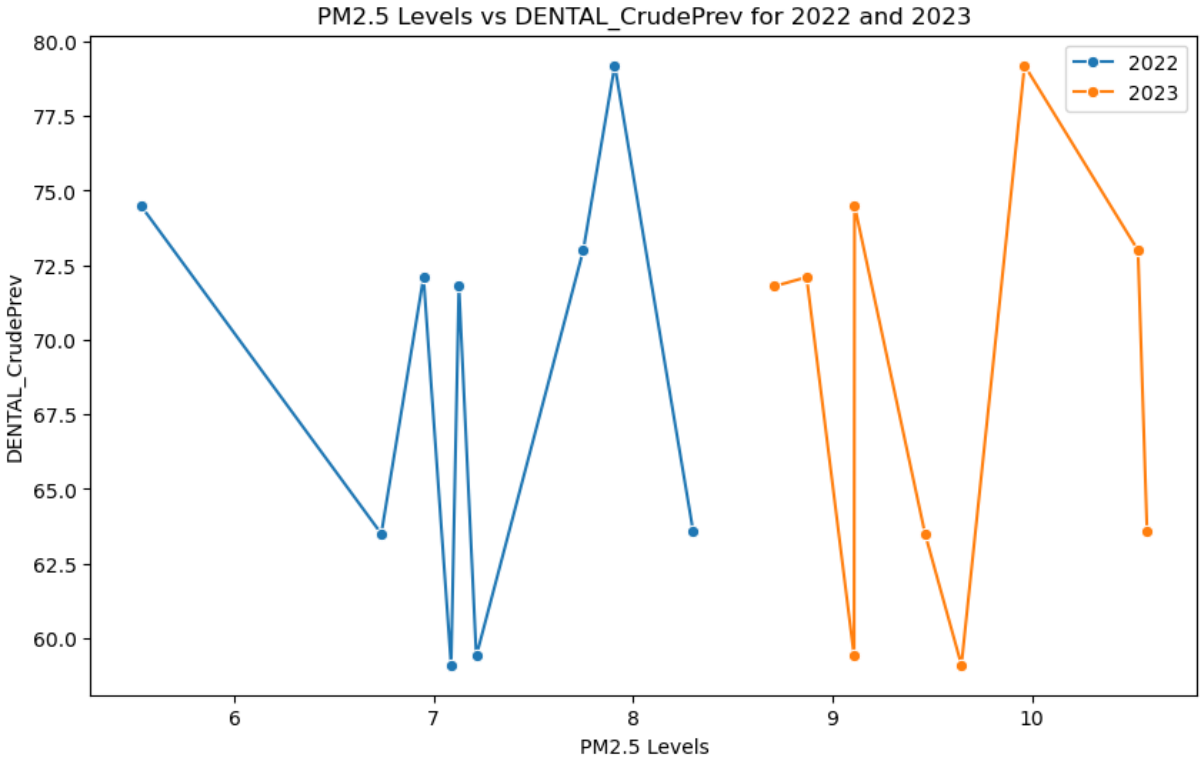

PM2.5 Levels vs COLON_SCREEN_CrudePrev for 2022 and 2023
2022
2023
COLON_SCREEN_CrudePrev
84
82
80
78
76
74
6
7
8
9
10
PM2.5 Levels

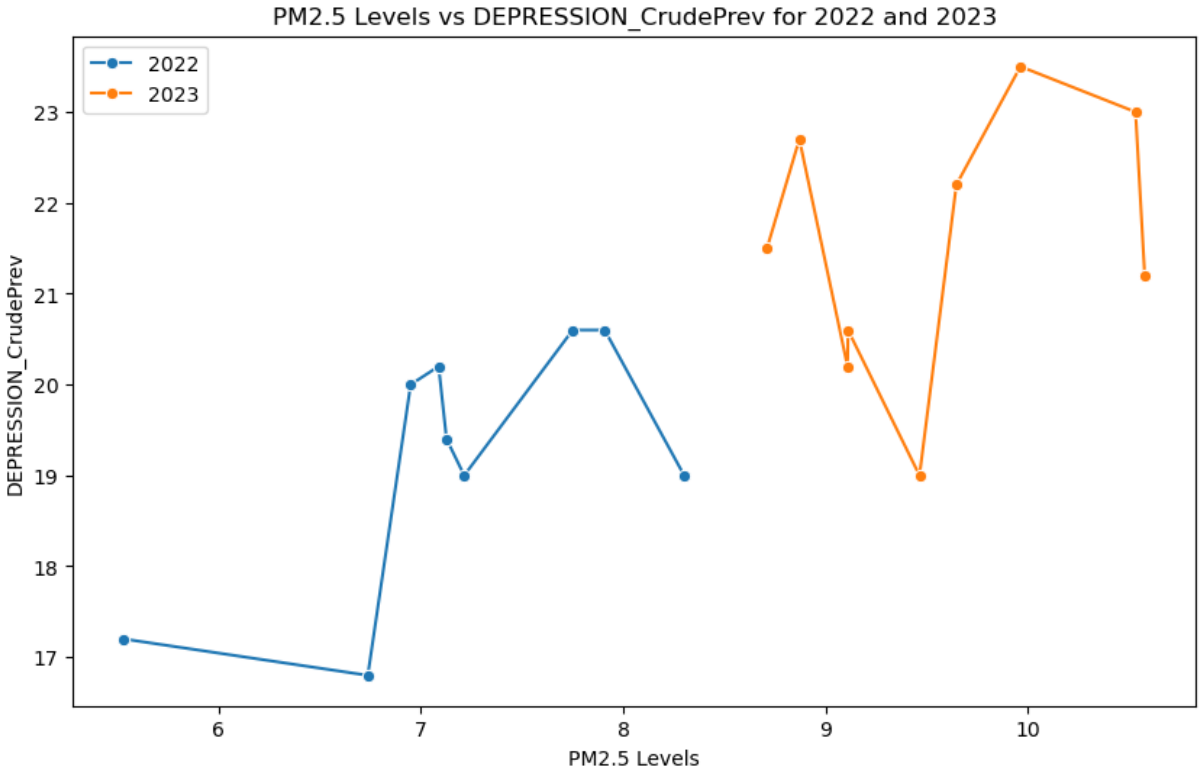

PM2.5 Levels vs COPD_CrudePrev for 2022 and 2023
2022
2023
COPD_CrudePrev
PM2.5 Levels
2
3
4
5
6
6
7
8
9
10

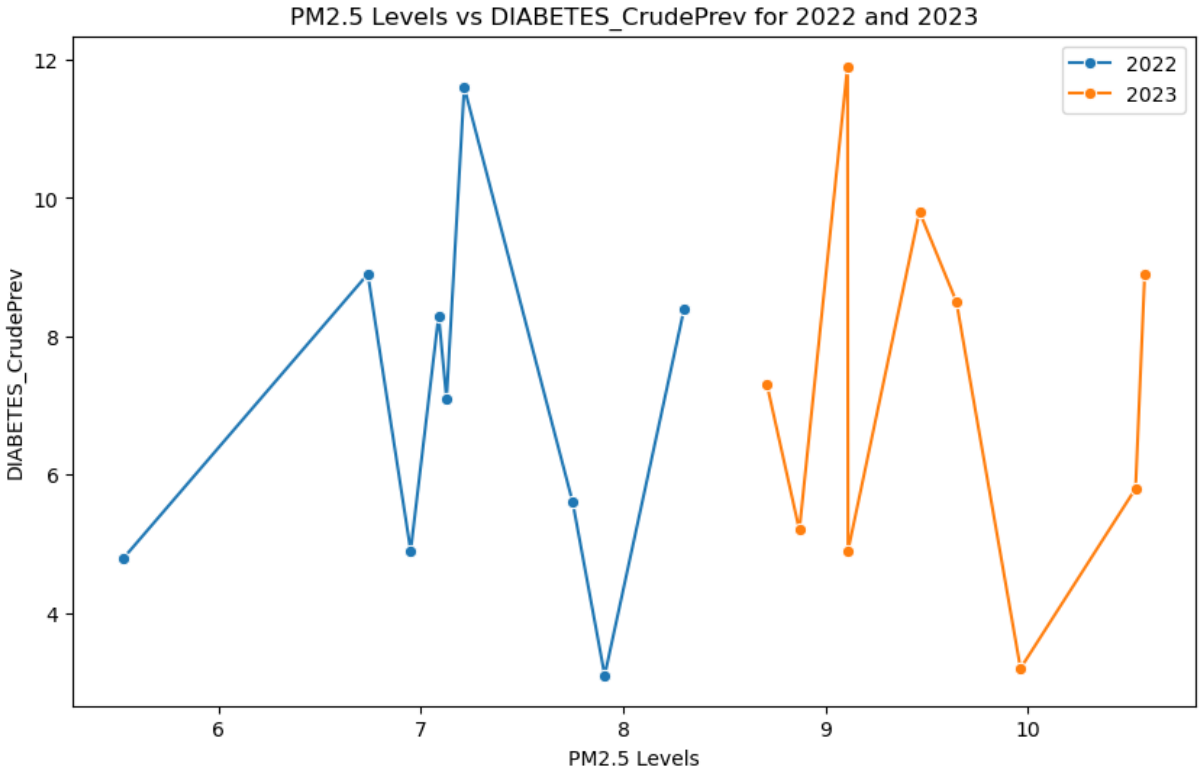

PM2.5 Levels vs COREM_CrudePrev for 2022 and 2023
2022
2023
COREM_CrudePrev
PM2.5 Levels
44
46
48
50
52
54
56
58
6
7
8
9
10

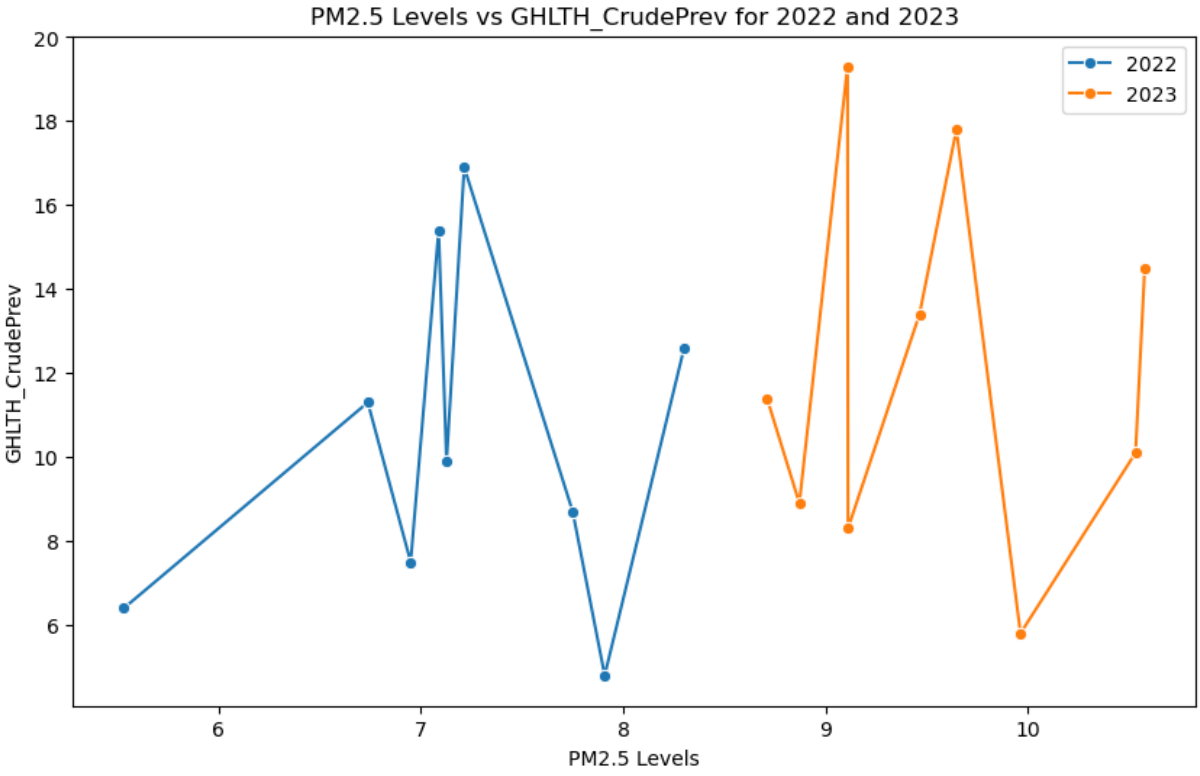

PM2.5 Levels vs COREW_CrudePrev for 2022 and 2023
2022
2023
COREW_CrudePrev
PM2.5 Levels
48
46
44
42
40
38
6
7
8
9
10

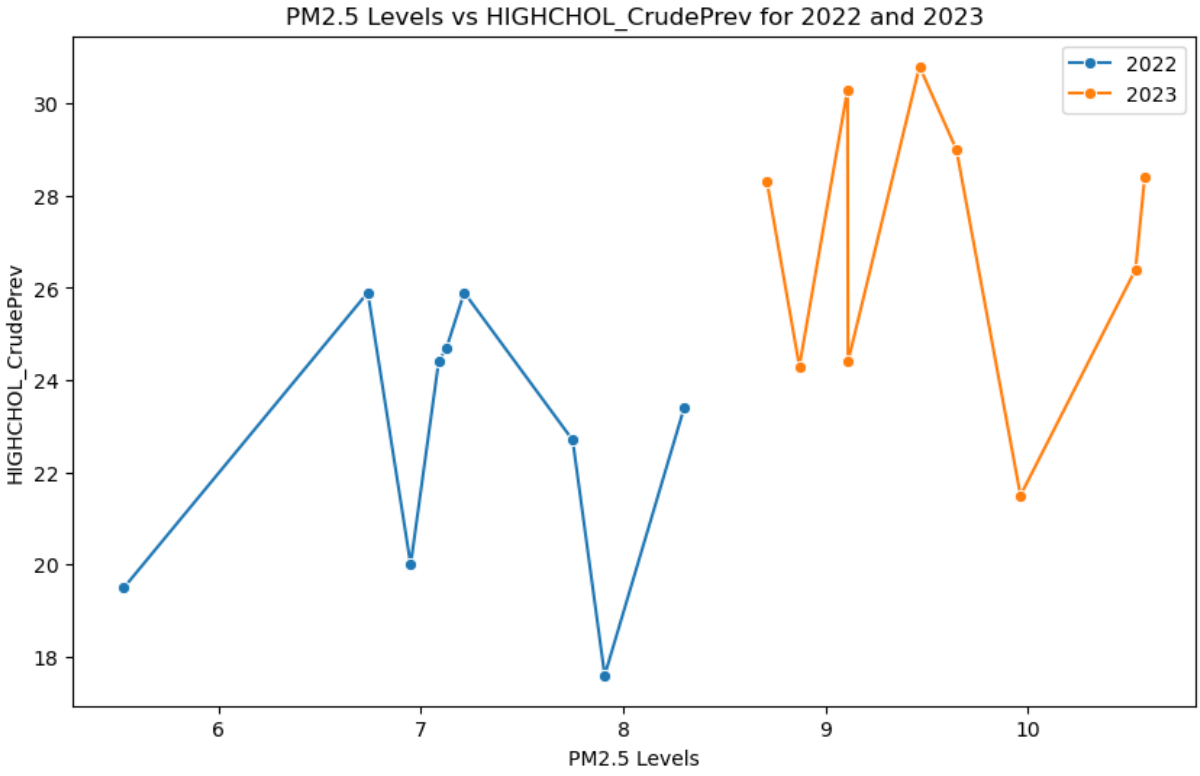

PM2.5 Levels vs CSMOKING_CrudePrev for 2022 and 2023
2022
2023
CSMOKING_CrudePrev
PM2.5 Levels
16
14
12
10
8
6
6
7
8
9
10

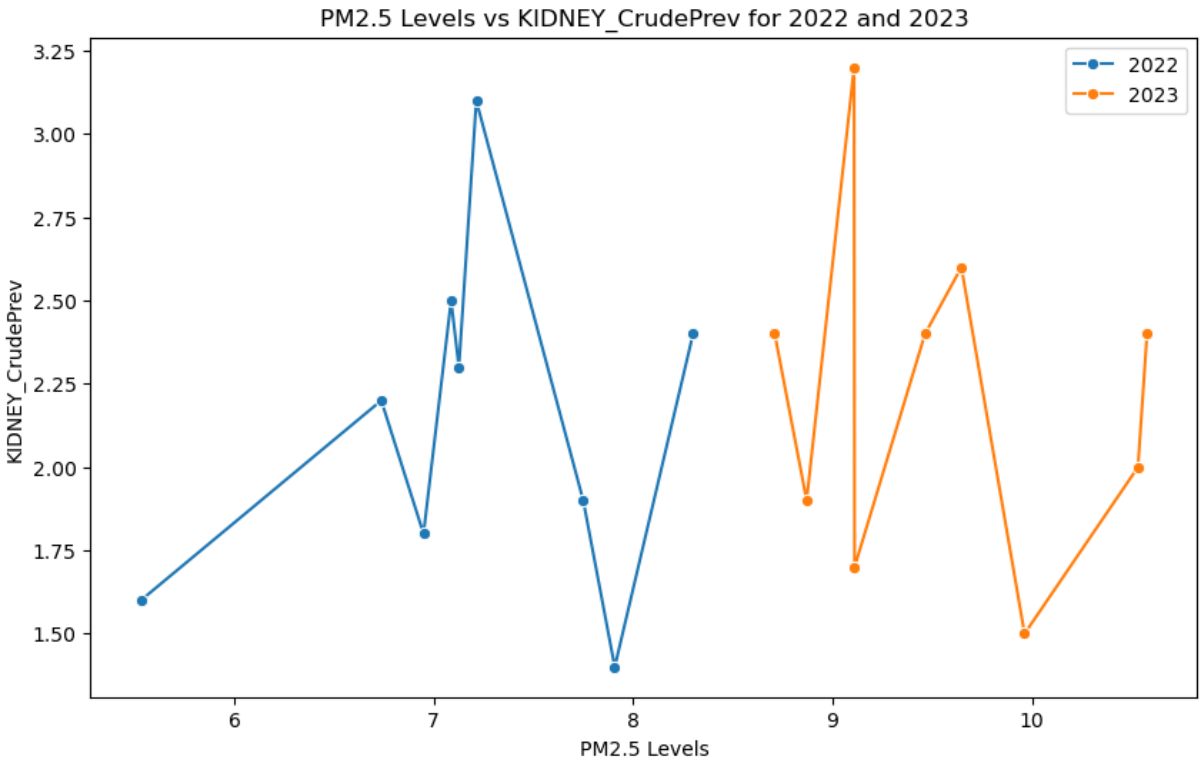

PM2.5 Levels vs DENTAL_CrudePrev for 2022 and 2023
2022
2023
DENTAL_CrudePrev
PM2.5 Levels
80.0
77.5
75.0
72.5
70.0
67.5
65.0
62.5
60.0
6
7
8
9
10

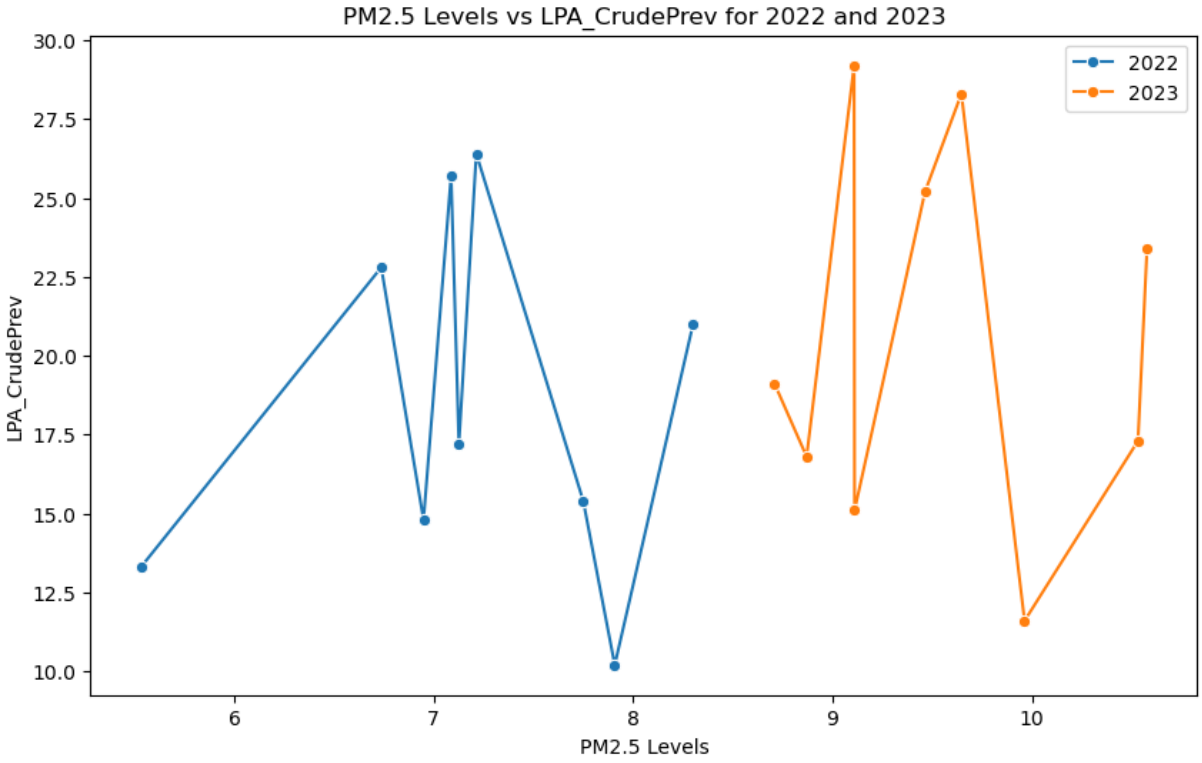

PM2.5 Levels vs DEPRESSION_CrudePrev for 2022 and 2023
2022
2023
DEPRESSION_CrudePrev
PM2.5 Levels
23
22
21
20
19
18
17
6
7
8
9
10

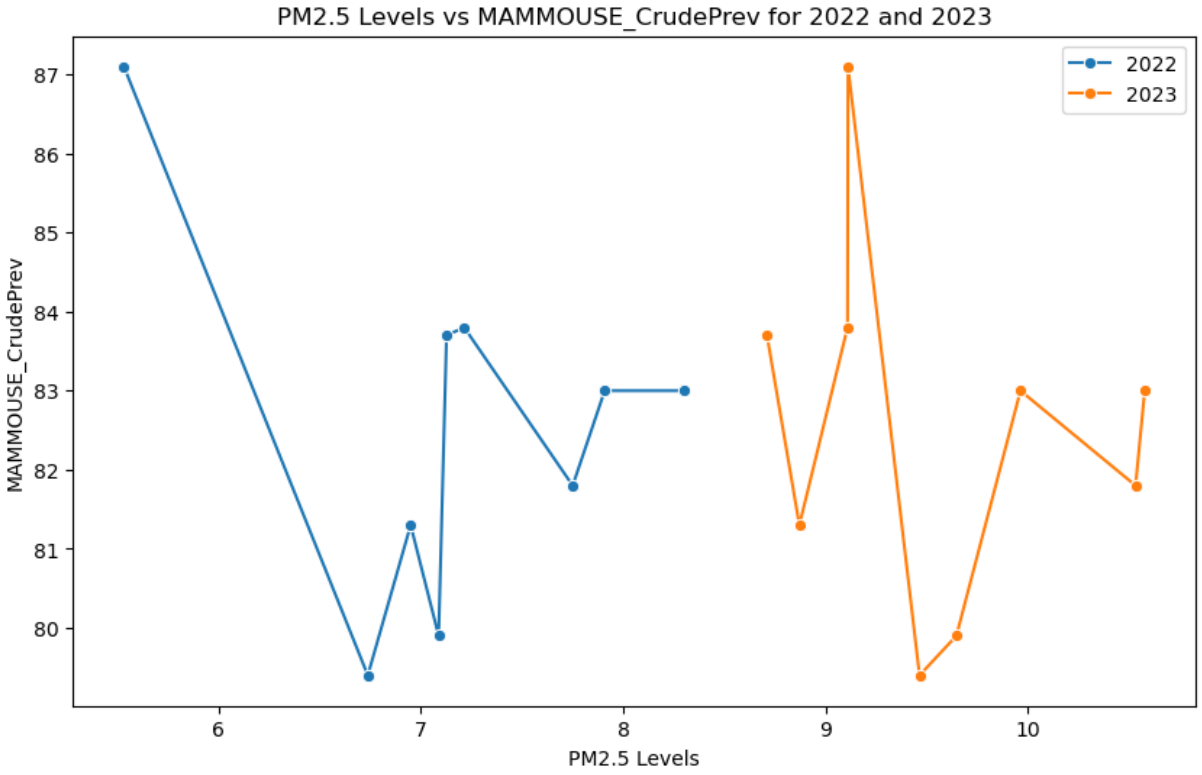

PM2.5 Levels vs DIABETES_CrudePrev for 2022 and 2023
2022
2023
DIABETES_CrudePrev
PM2.5 Levels
12
10
8
6
4
6
7
8
9
10

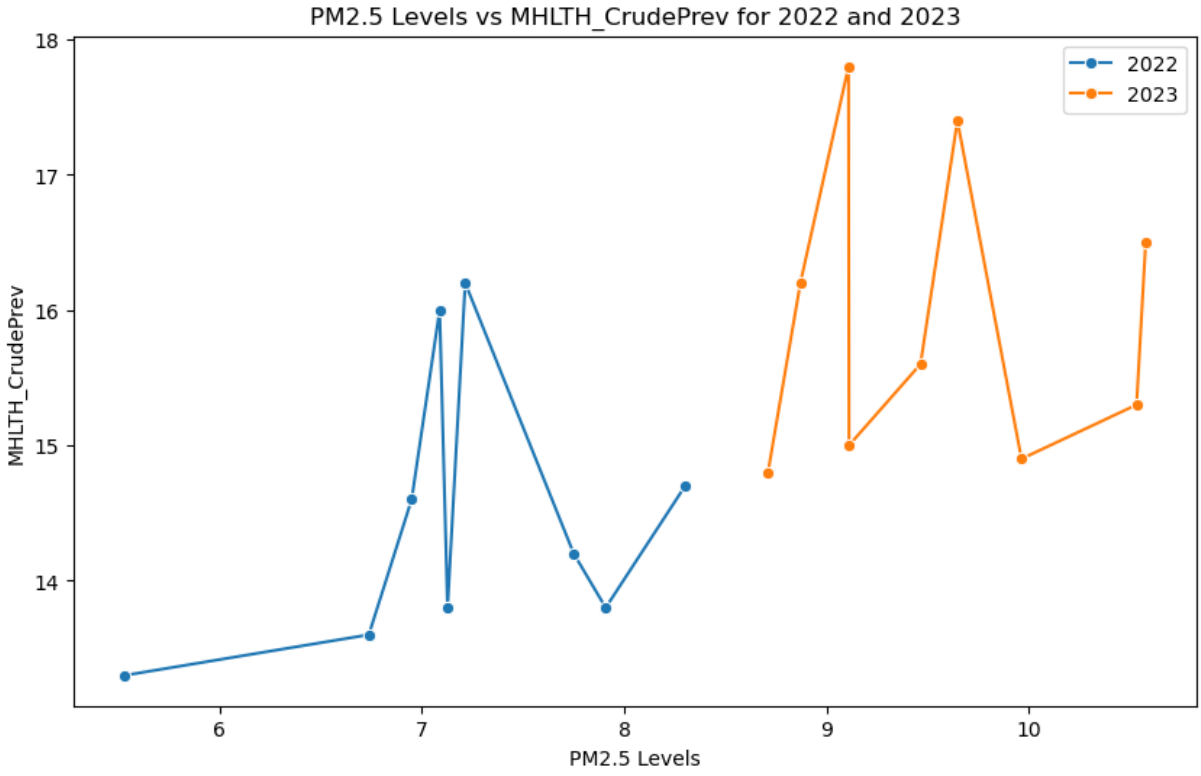

PM2.5 Levels vs GHLTH_CrudePrev for 2022 and 2023
2022
2023
GHLTH_CrudePrev
PM2.5 Levels
20
18
16
14
12
10
8
6
6
7
8
9
10

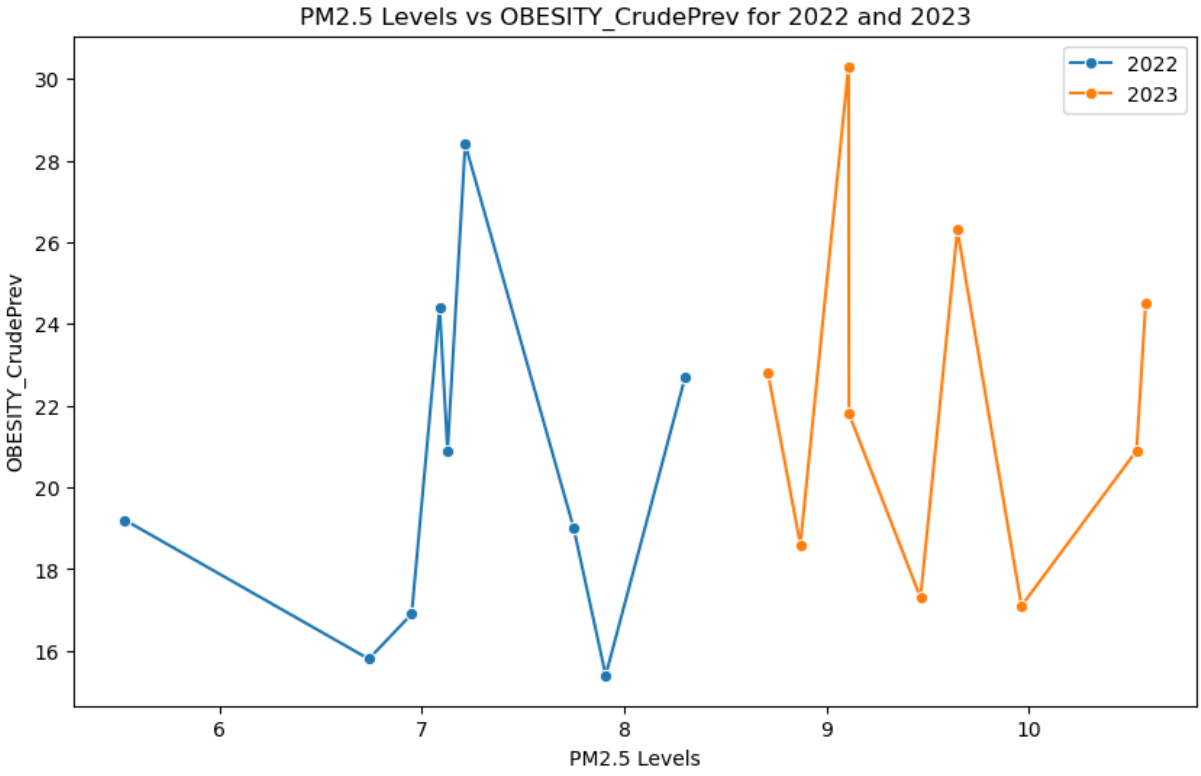

PM2.5 Levels vs HIGHCHOL_CrudePrev for 2022 and 2023
2022
2023
HIGHCHOL_CrudePrev
PM2.5 Levels
30
28
26
24
22
20
18
6
7
8
9
10

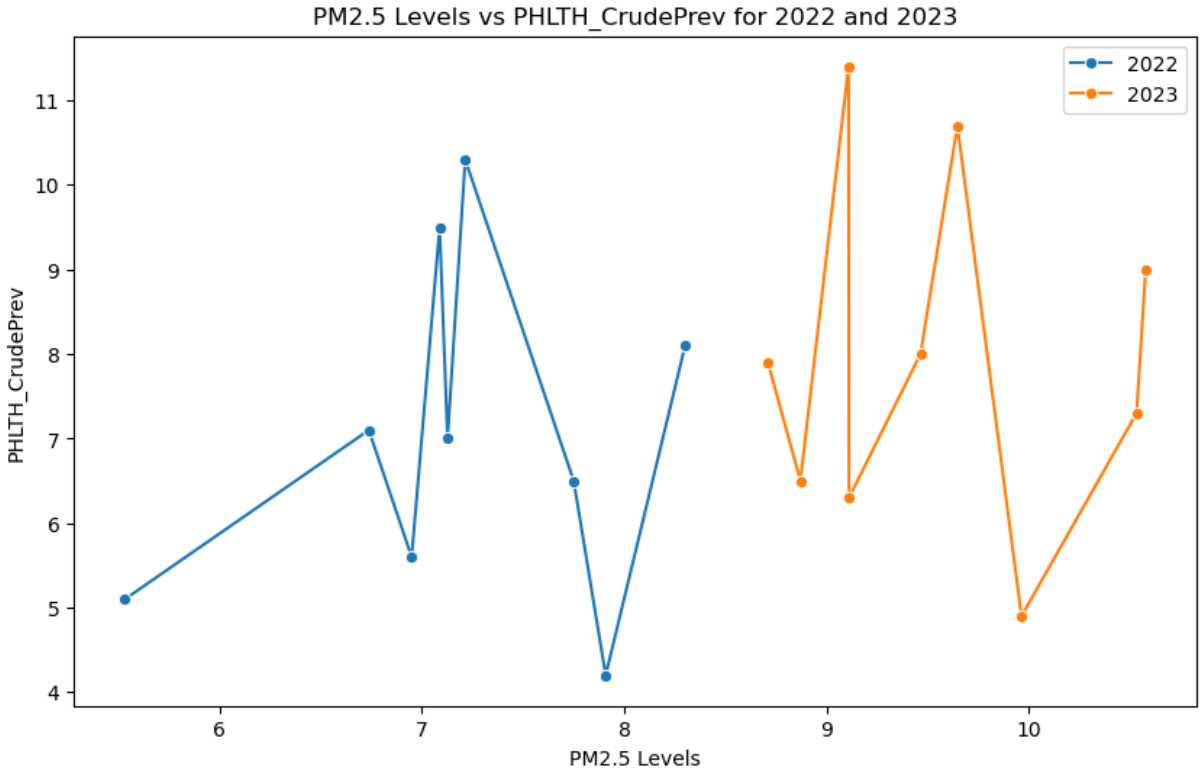

PM2.5 Levels vs KIDNEY_CrudePrev for 2022 and 2023
2022
2023
KIDNEY_CrudePrev
PM2.5 Levels
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
6
7
8
9
10


PM2.5 Levels vs LPA_CrudePrev for 2022 and 2023
2022
2023
LPA_CrudePrev
PM2.5 Levels
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
6
7
8
9
10

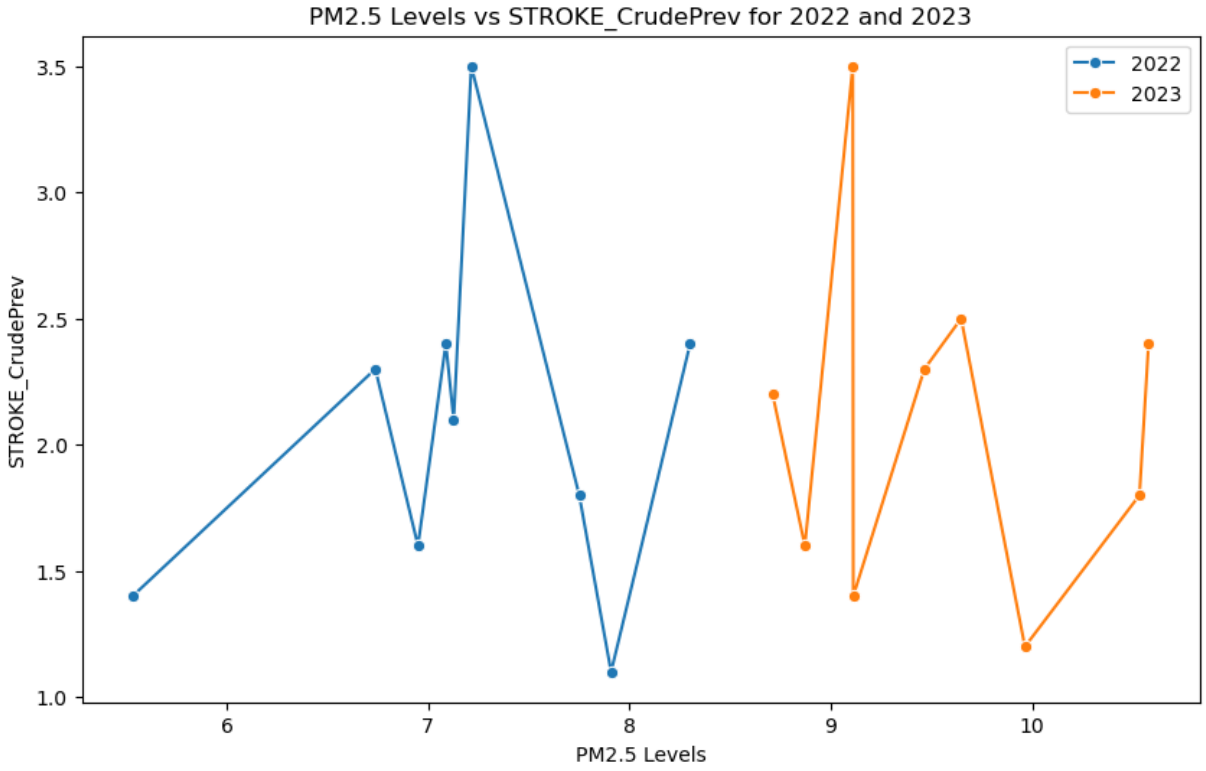

PM2.5 Levels vs MAMMOUSE_CrudePrev for 2022 and 2023
2022
2023
MAMMOUSE_CrudePrev
PM2.5 Levels
87
86
85
84
83
82
81
80
6
7
8
9
10

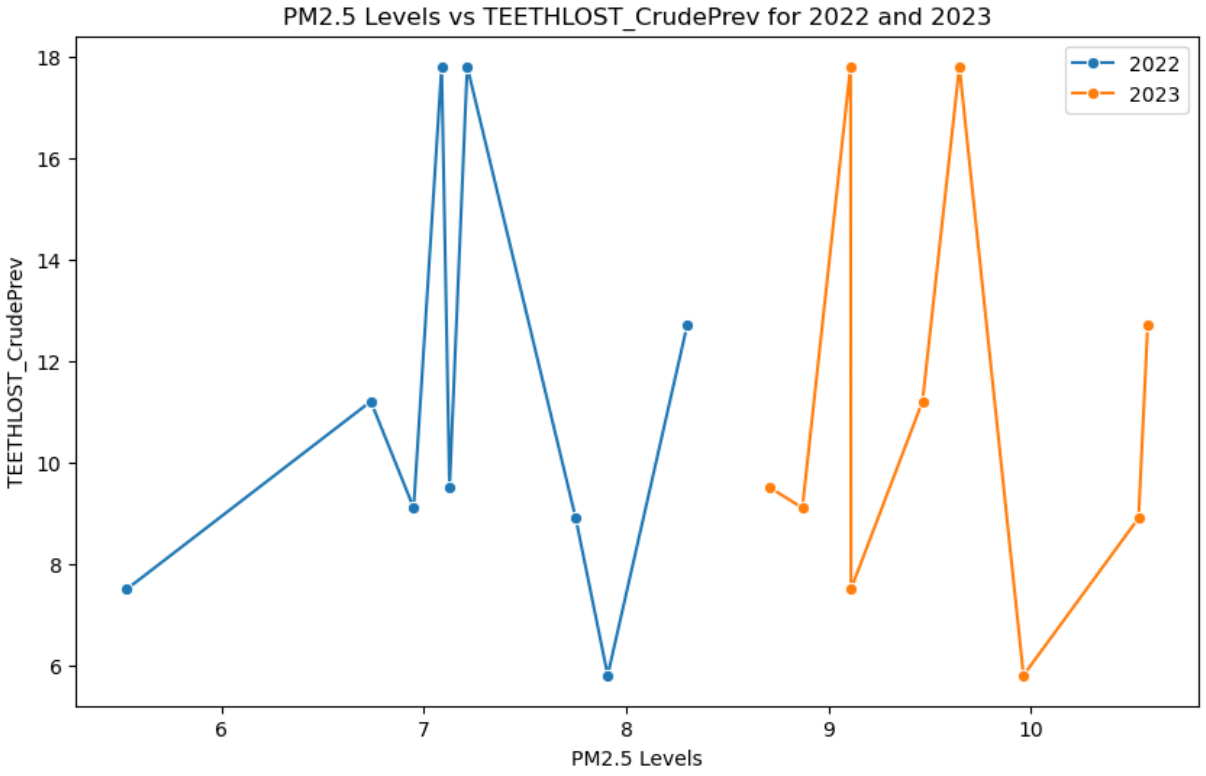

PM2.5 Levels vs MHLTH_CrudePrev for 2022 and 2023
2022
2023
MHLTH_CrudePrev
PM2.5 Levels
18
17
16
15
14
6
7
8
9
10


PM2.5 Levels vs OBESITY_CrudePrev for 2022 and 2023
2022
2023
OBESITY_CrudePrev
PM2.5 Levels
30
28
26
24
22
20
18
16
6
7
8
9
10


PM2.5 Levels vs PHLTH_CrudePrev for 2022 and 2023
2022
2023
PHLTH_CrudePrev
PM2.5 Levels
4
5
6
7
8
9
10
11
6
7
8
9
10

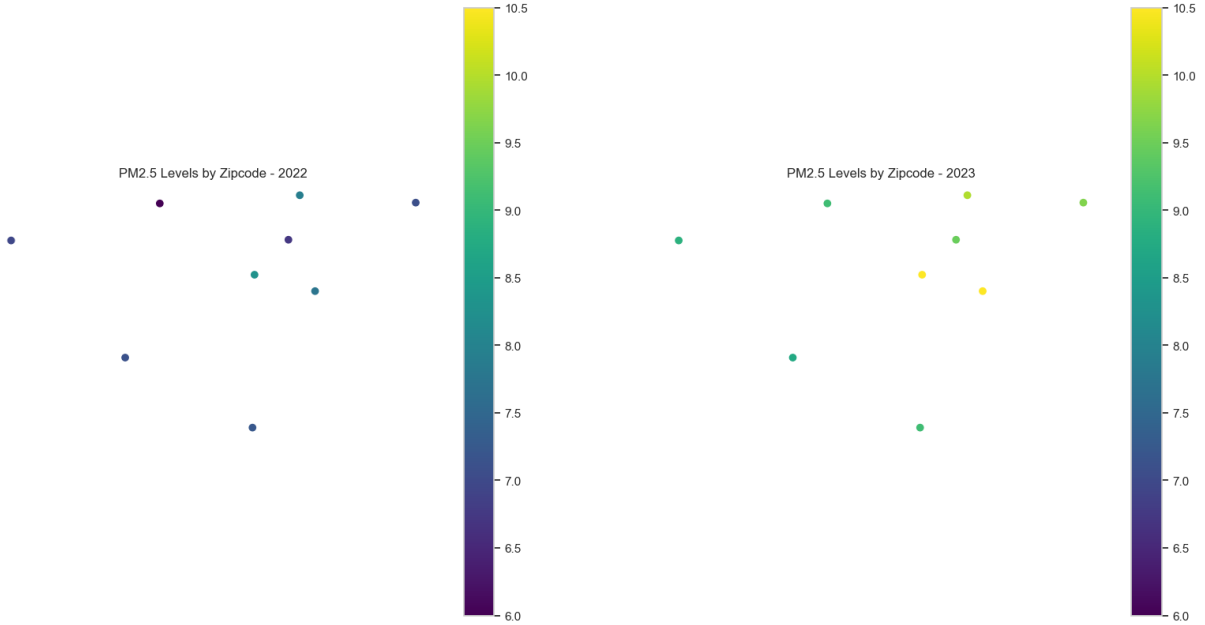

PM2.5 Levels vs SLEEP_CrudePrev for 2022 and 2023
2022
2023
SLEEP_CrudePrev
PM2.5 Levels
26
28
30
32
34
36
38
6
7
8
9
10





In [65]:
```
health_data_2022['geometry'] = health_data_2022['Geolocation'].apply(wkt.loa
health_data_2023['geometry'] = health_data_2023['Geolocation'].apply(wkt.loa

# Creating GeoDataFrame
geo_data_2022 = gpd.GeoDataFrame(health_data_2022, geometry='geometry')
geo_data_2023 = gpd.GeoDataFrame(health_data_2023, geometry='geometry')

fig, ax = plt.subplots(1, 2, figsize=(20, 10))
```

```
# Map for 2022
geo_data_2022.plot(column='PM2.5', cmap='viridis', legend=True, ax=ax[0], vm
ax[0].set_title('PM2.5 Levels by Zipcode - 2022')
ax[0].set_axis_off()

# Map for 2023
geo_data_2023.plot(column='PM2.5', cmap='viridis', legend=True, ax=ax[1], vm
ax[1].set_title('PM2.5 Levels by Zipcode - 2023')
ax[1].set_axis_off()

plt.show()
```



In [13]:
```
cohort_data = pd.merge(health_data_2022, health_data_2023, on='Zipcode', suf

# Preparing data for heatmap
cohort_data_comparison = cohort_data.copy()
for col in health_data_2022.columns:
    if 'Crude' in col or col == 'PM2.5':
        cohort_data_comparison[f'{col}_Change'] = cohort_data_comparison[f'{

# Function to plot bar graphs for a selected health indicator
def plot_bar_graph_for_crude_rate(selected_rate):
    # Extracting relevant data
    data = cohort_data_comparison[['Zipcode', f'{selected_rate}_Change', 'PM

    # Plotting
    fig, ax = plt.subplots(figsize=(15, 8))
    data.set_index('Zipcode').plot(kind='bar', ax=ax)
    ax.set_title(f'Change in {selected_rate} and PM2.5 Levels by Zipcode (20
    ax.set_xlabel('Zipcode')
    ax.set_ylabel('Change')
    plt.xticks(rotation=45)
    plt.show()

# Dropdown menu for selecting the crude rate
crude_rates = [col.replace('_2022', '').replace('_Change', '') for col in cc
```

```
dropdown = Dropdown(options=crude_rates, description='Select Crude Rate:')

# Interactive widget
interact(plot_bar_graph_for_crude_rate, selected_rate=dropdown)
```

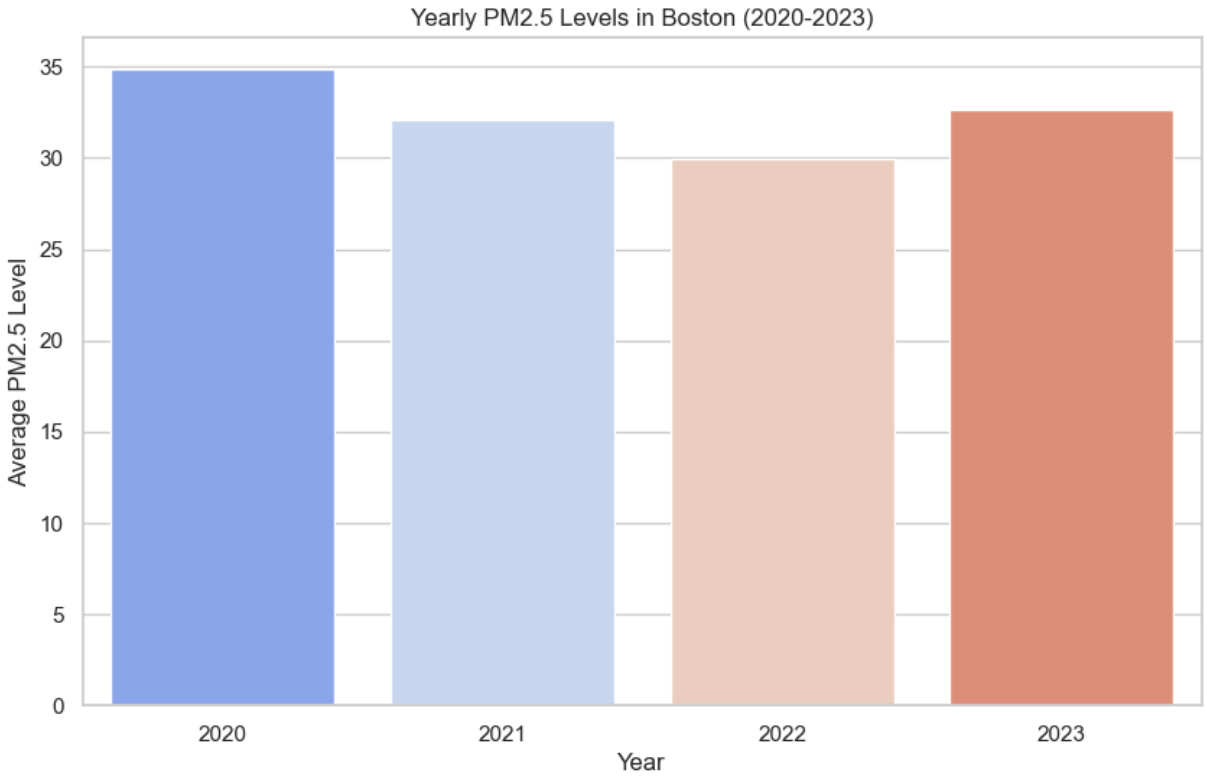
```
interactive(children=(Dropdown(description='Select Crude Rate:', options=('ACCESS2_CrudePrev', 'ARTHRITIS_Crud…
```

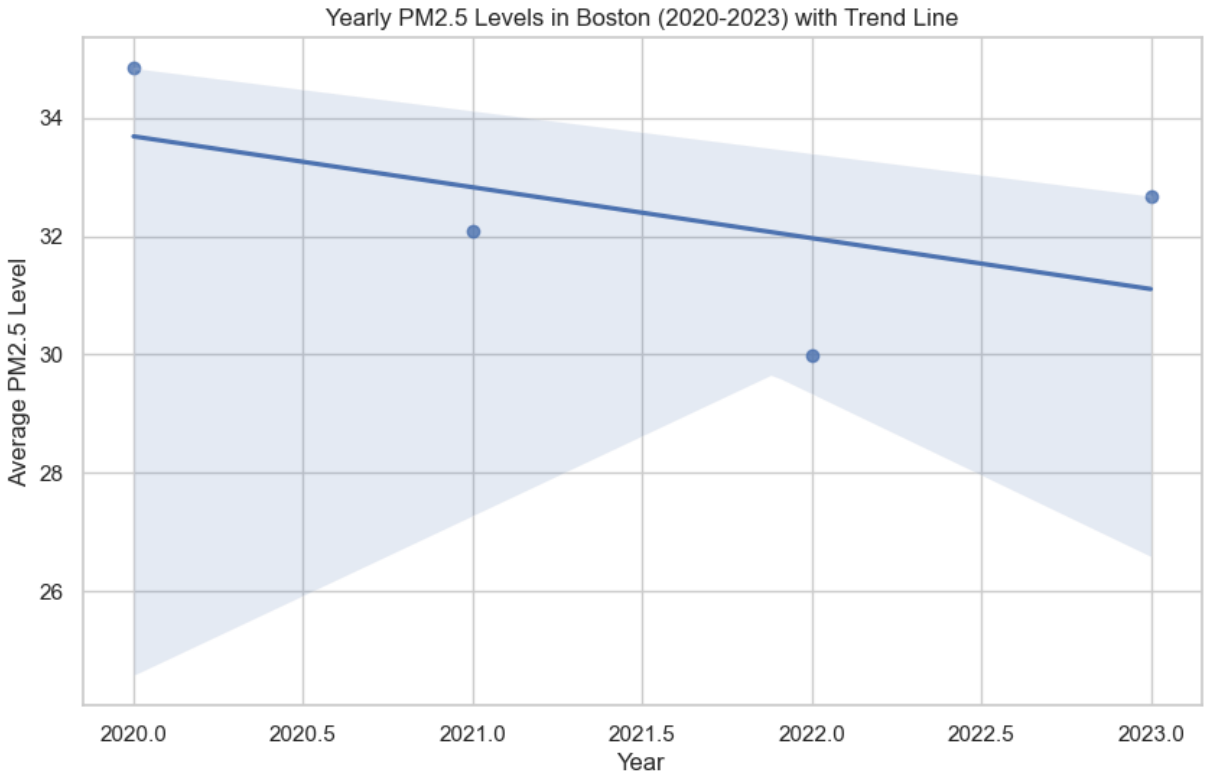
Out[13]: <function __main__.plot_bar_graph_for_crude_rate(selected_rate)>

Boston - Analysis

In [14]:
```
boston_df = pd.read_csv('../Merged_Data/Boston_Health_20_23.csv')
```

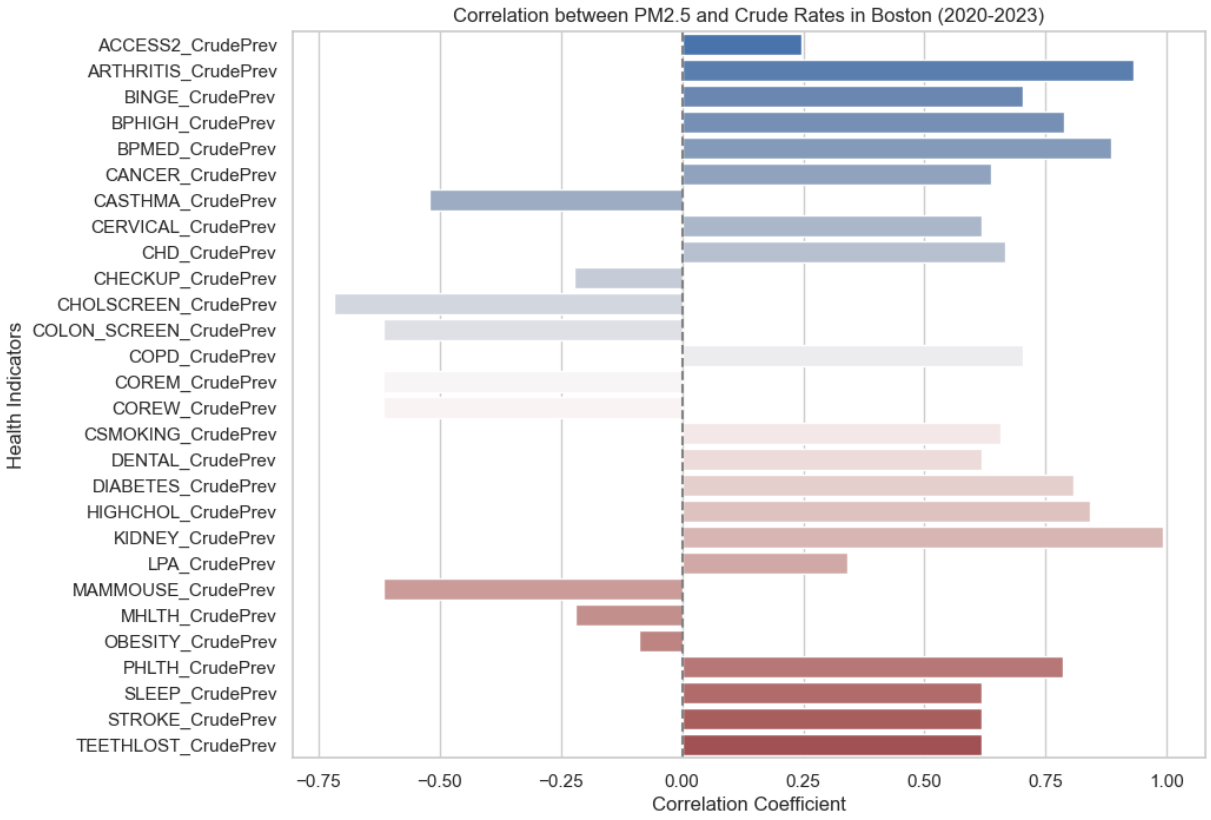
In [15]:
```
sns.set(style="whitegrid")

# Plotting PM2.5 levels for each year
plt.figure(figsize=(10, 6))
sns.barplot(x="Year", y="pm25", data=boston_df, palette="coolwarm")
plt.title("Yearly PM2.5 Levels in Boston (2020-2023)")
plt.xlabel("Year")
plt.ylabel("Average PM2.5 Level")
plt.show()

plt.figure(figsize=(10, 6))
sns.regplot(x="Year", y="pm25", data=boston_df, color='b', marker='o')

# Enhancing the plot
plt.title("Yearly PM2.5 Levels in Boston (2020-2023) with Trend Line")
plt.xlabel("Year")
plt.ylabel("Average PM2.5 Level")
plt.show()
```





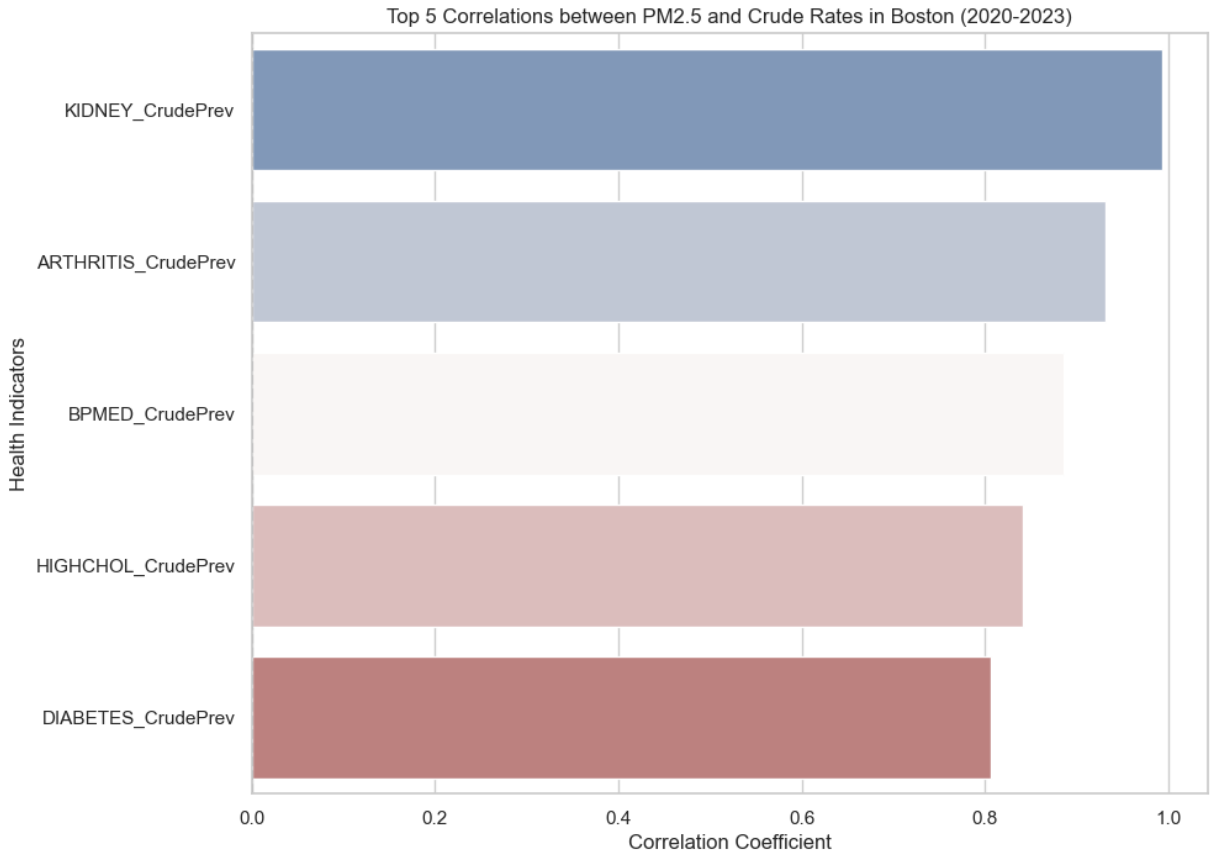
In [16]:
```
crude_columns = [col for col in boston_df.columns if 'Crude' in col]
crude_columns.append('pm25')

# Calculating the correlation matrix for the specified columns in boston_df
correlation_matrix = boston_df[crude_columns].corr()

# Extracting correlations with pm25, excluding the self-correlation of pm25
correlations_pm25 = correlation_matrix['pm25'].drop('pm25')
```

```
# Creating a bar plot to visualize the correlations
plt.figure(figsize=(10, 8))
sns.barplot(x=correlations_pm25.values, y=correlations_pm25.index, palette="
plt.title('Correlation between PM2.5 and Crude Rates in Boston (2020-2023)')
plt.xlabel('Correlation Coefficient')
plt.ylabel('Health Indicators')
plt.axvline(x=0, color='gray', linestyle='--')
plt.show()
```

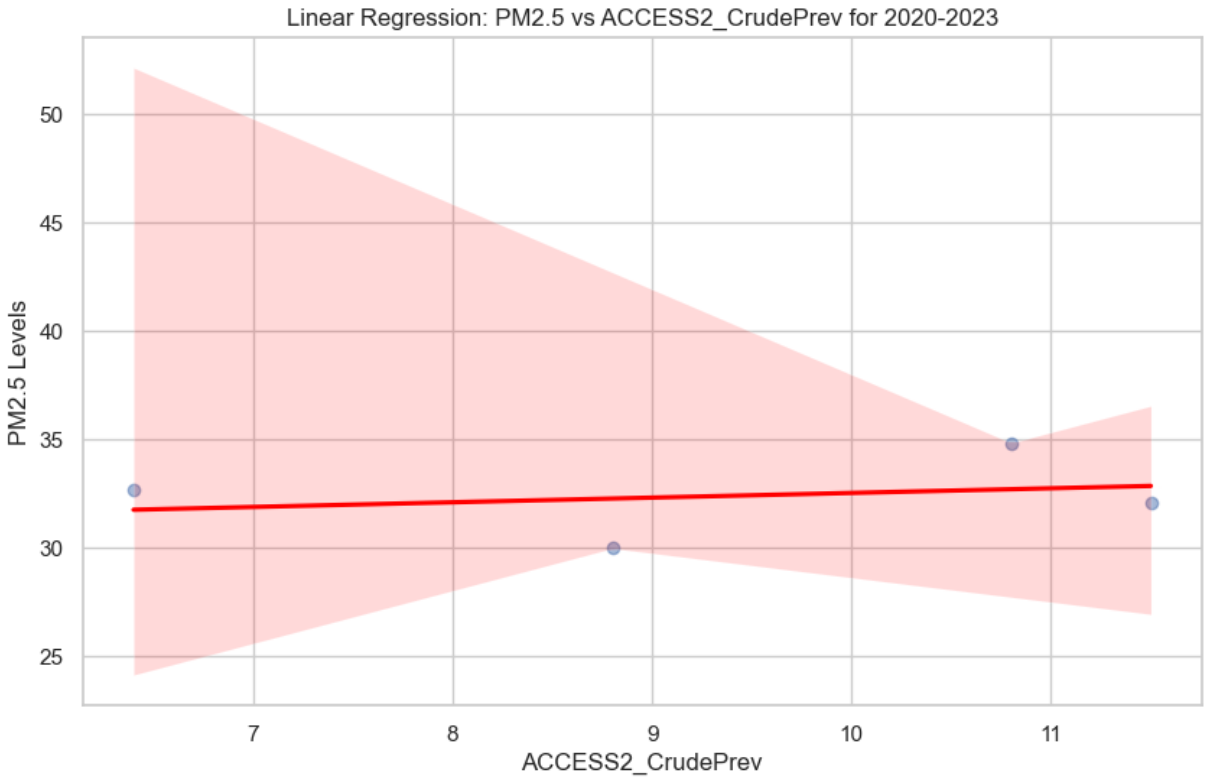


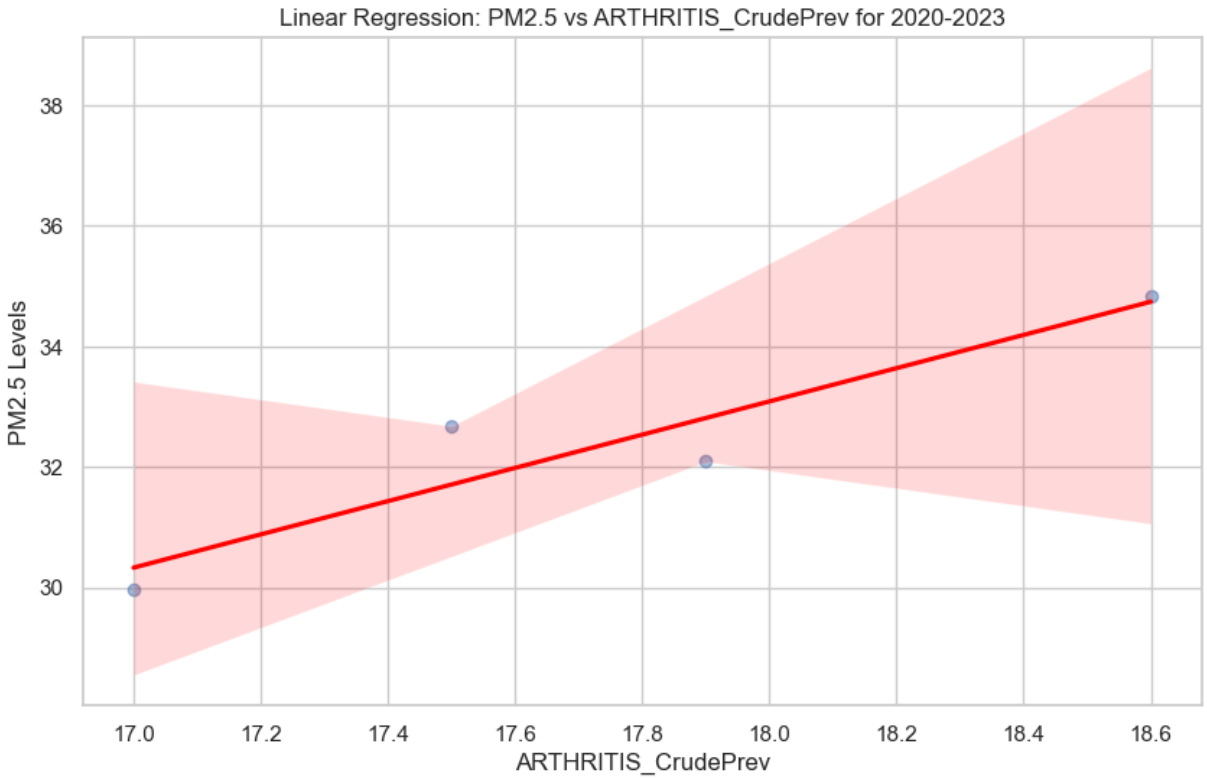
In [68]:

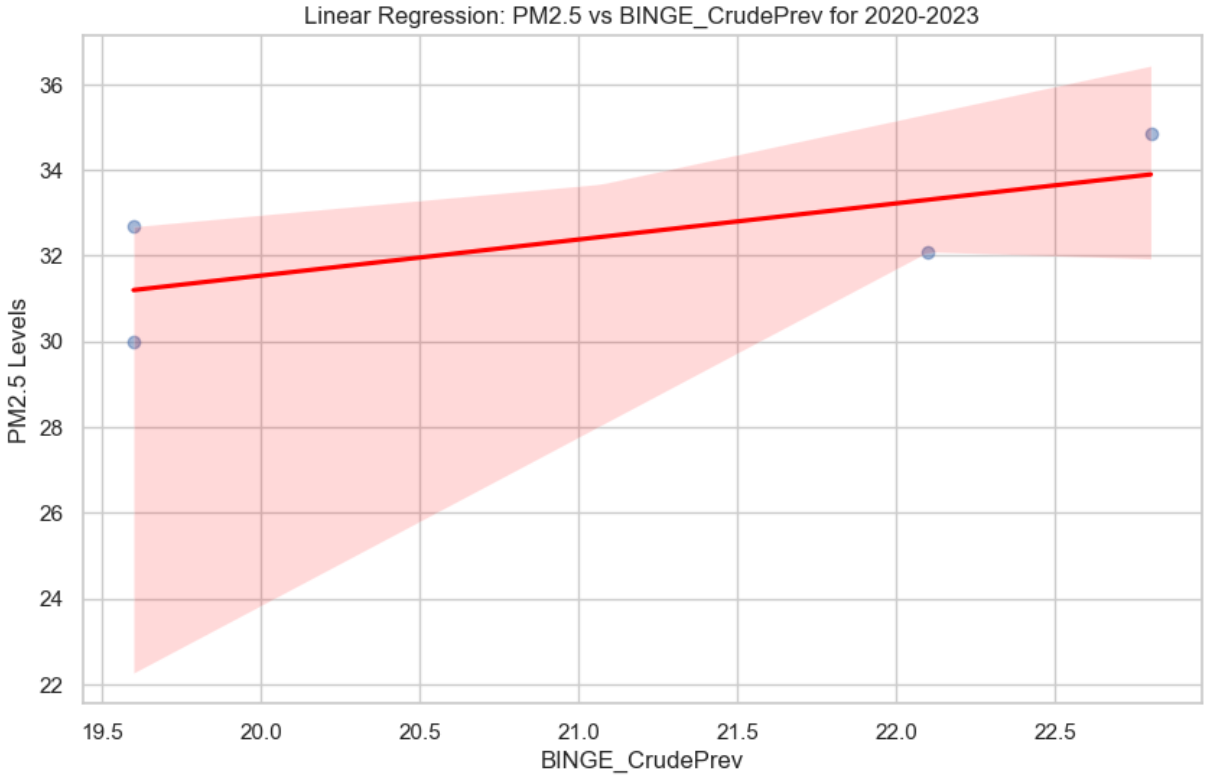
```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming you've already calculated 'correlations_pm25' as in your provided

# Sort correlations in descending order and get the top 5
top_correlations = correlations_pm25.abs().sort_values(ascending=False).head

# Subset correlation matrix to the top 5 correlated columns with PM2.5
top_correlation_matrix = correlation_matrix.loc[top_correlations.index, 'pm2

# Create a bar plot for the top 5 correlations
plt.figure(figsize=(10, 8))
sns.barplot(x=top_correlations.values, y=top_correlations.index, palette="vl
plt.title('Top 5 Correlations between PM2.5 and Crude Rates in Boston (2020-
plt.xlabel('Correlation Coefficient')
plt.ylabel('Health Indicators')
plt.axvline(x=0, color='gray', linestyle='--')
plt.show()
```

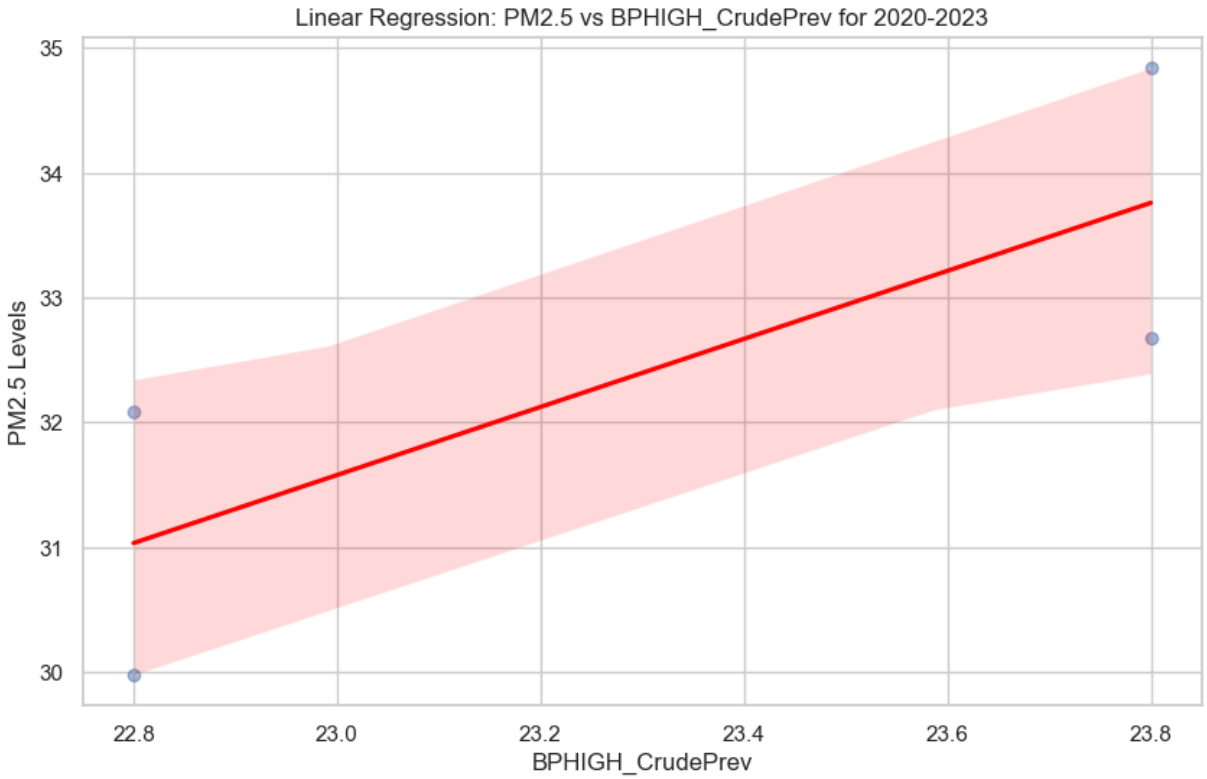


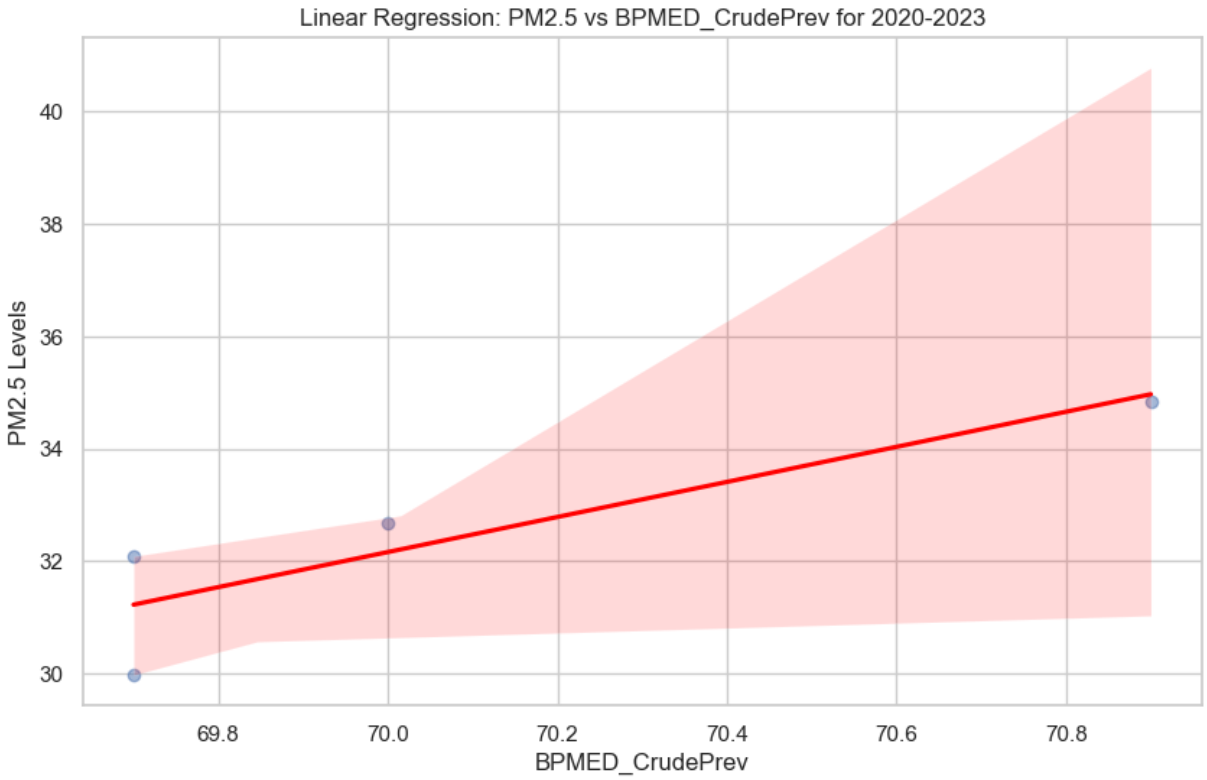
In [17]:
```
def plot_linear_regression(data, crude_columns, year):
    for column in crude_columns:
        if column != 'pm25':
            # Performing linear regression
            slope, intercept, r_value, p_value, std_err = stats.linregress(d

            # Plotting
            plt.figure(figsize=(10, 6))
            sns.regplot(x=column, y='pm25', data=data, scatter_kws={'alpha':
            plt.title(f'Linear Regression: PM2.5 vs {column} for {year}')
            plt.xlabel(column)
            plt.ylabel('PM2.5 Levels')
            plt.show()

# Extracting crude columns from the boston_df dataset
crude_columns_boston = [col for col in boston_df.columns if 'Crude' in col]

# Making linear regression plots for each crude rate against PM2.5 level for
plot_linear_regression(boston_df, crude_columns_boston, '2020-2023')
```

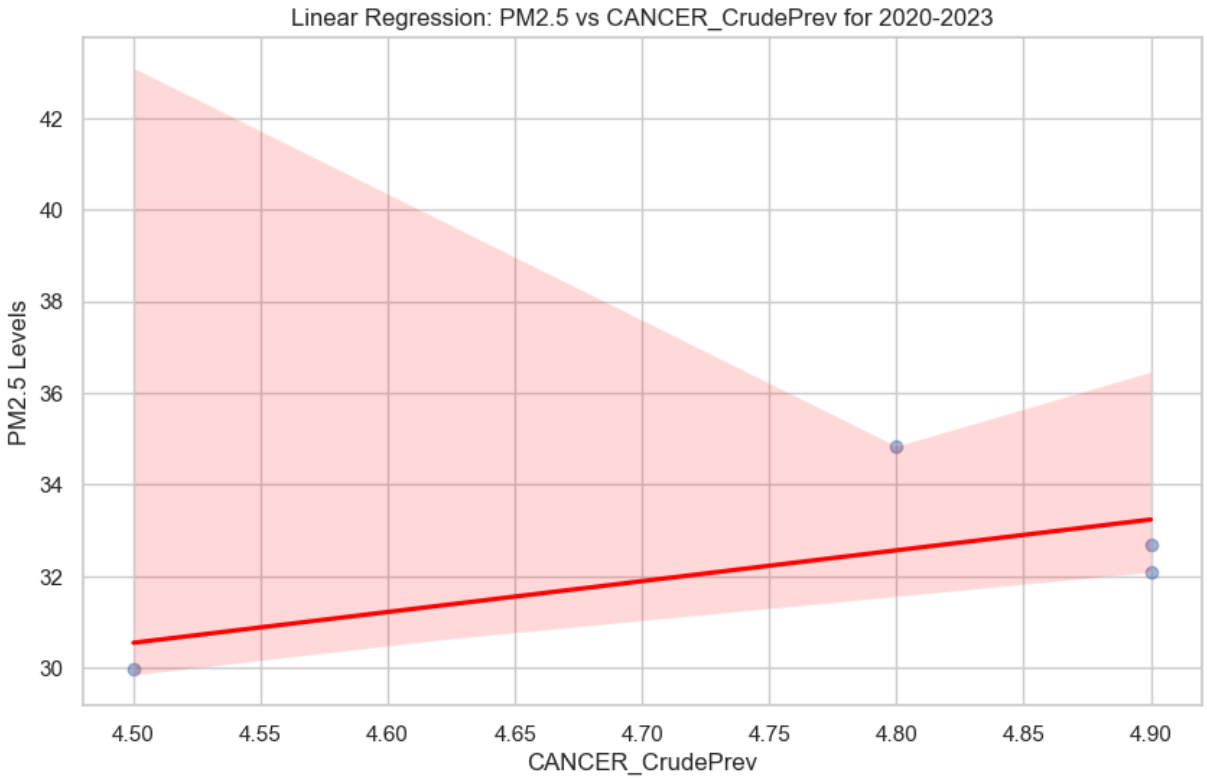

Linear Regression: PM2.5 vs ACCESS2_CrudePrev for 2020-2023
PM2.5 Levels
50
45
40
35
30
25
7
8
9
10
11
ACCESS2_CrudePrev

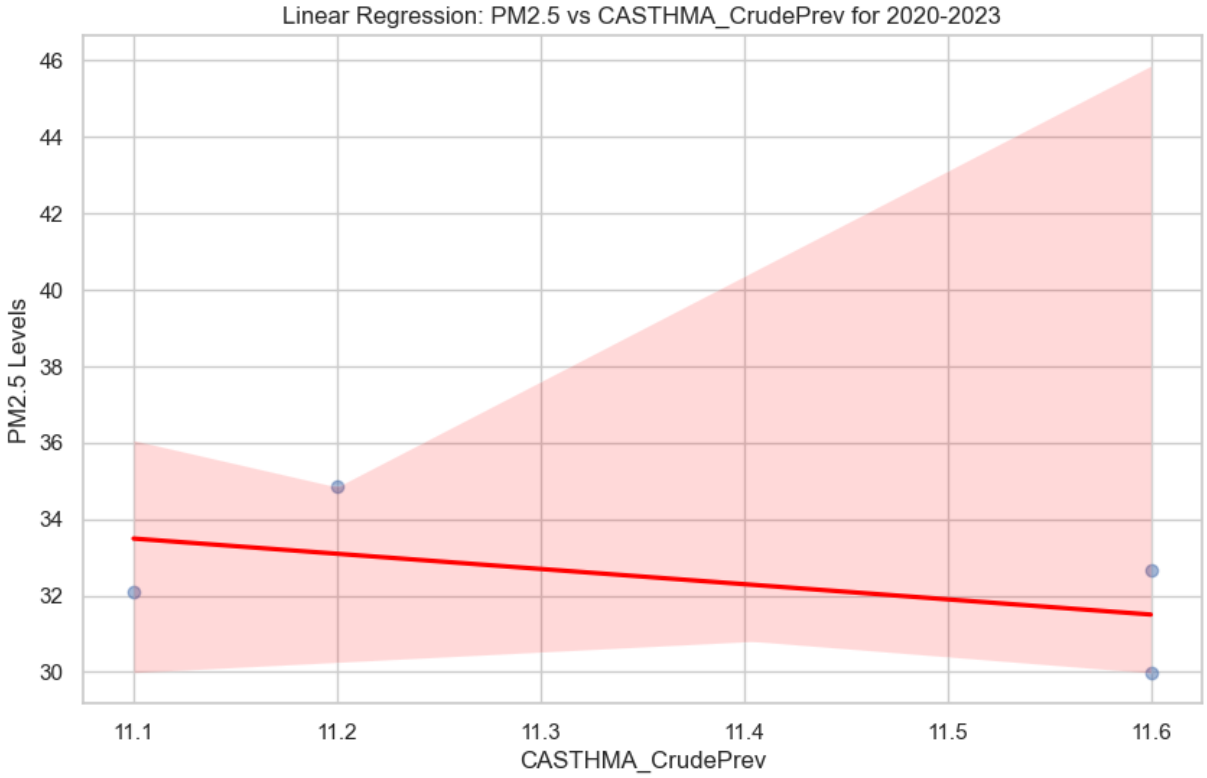

Linear Regression: PM2.5 vs ARTHRITIS_CrudePrev for 2020-2023
PM2.5 Levels
38
36
34
32
30
17.0
17.2
17.4
17.6
17.8
18.0
18.2
18.4
18.6
ARTHRITIS_CrudePrev

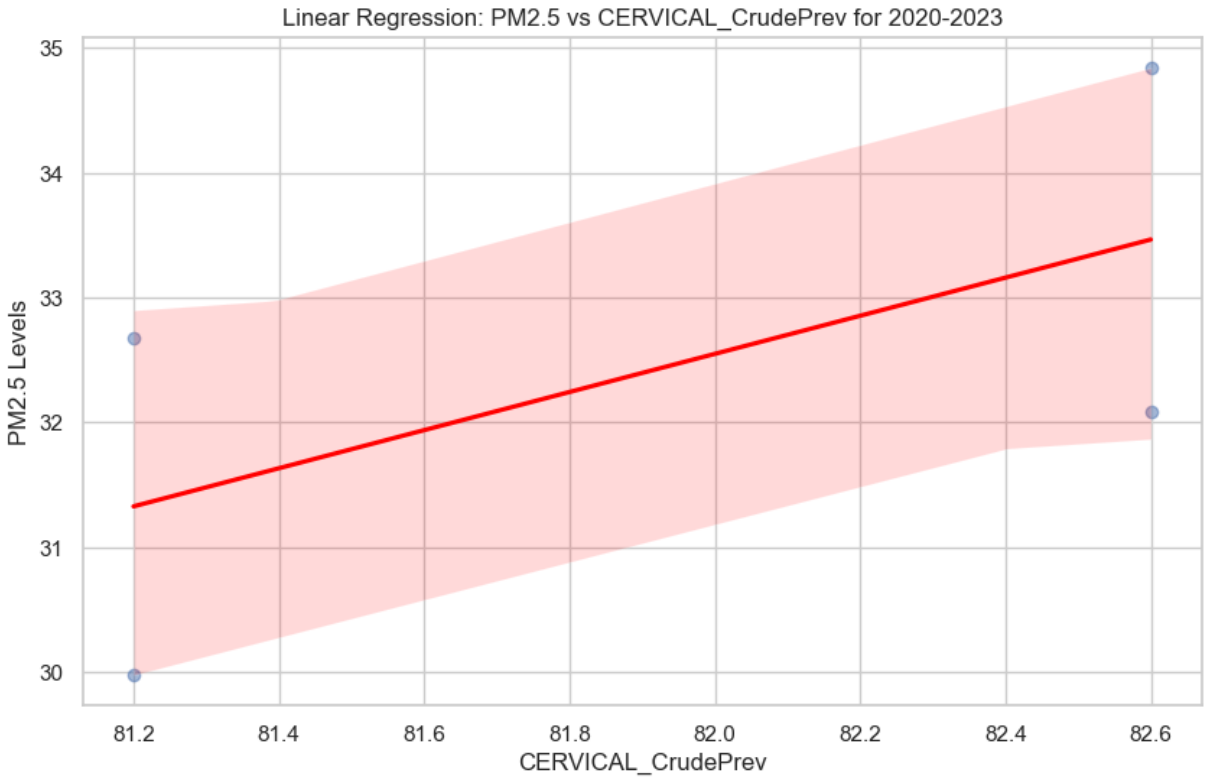

Linear Regression: PM2.5 vs BINGE_CrudePrev for 2020-2023
PM2.5 Levels
BINGE_CrudePrev
36
34
32
30
28
26
24
22
19.5
20.0
20.5
21.0
21.5
22.0
22.5

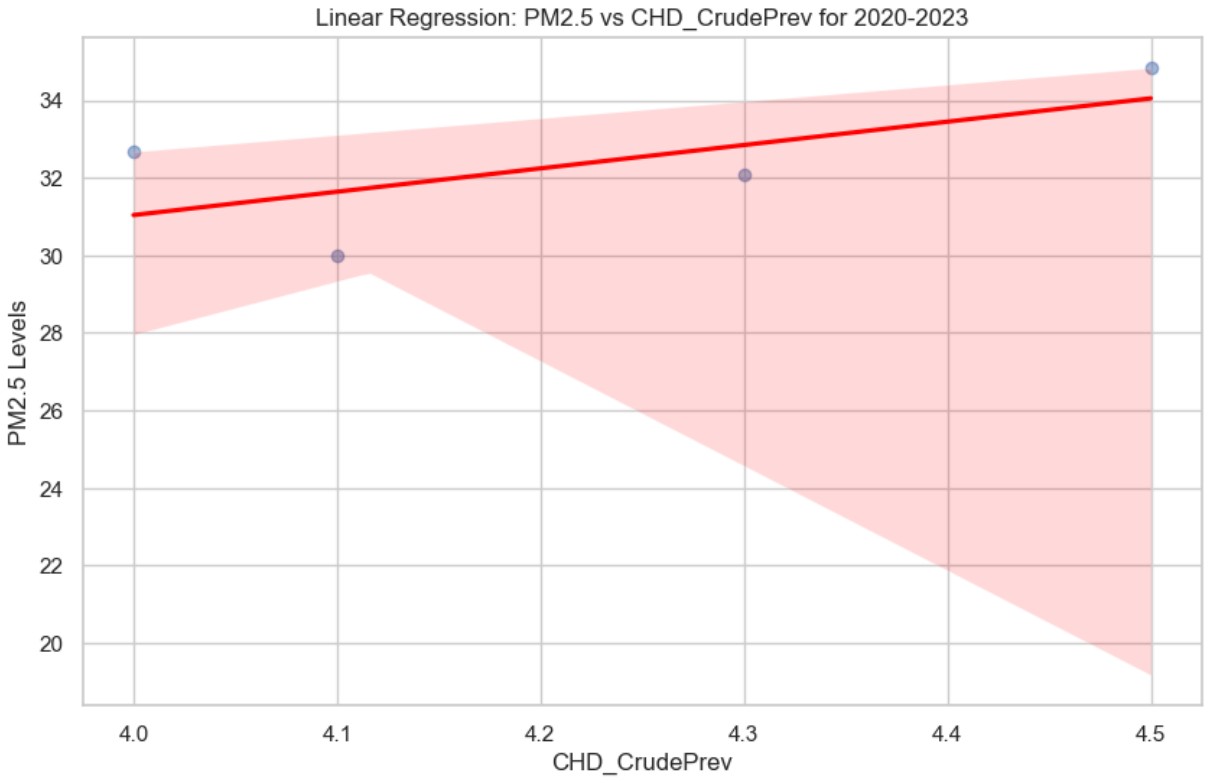

Linear Regression: PM2.5 vs BPHIGH_CrudePrev for 2020-2023
PM2.5 Levels
BPHIGH_CrudePrev
35
34
33
32
31
30
22.8
23.0
23.2
23.4
23.6
23.8

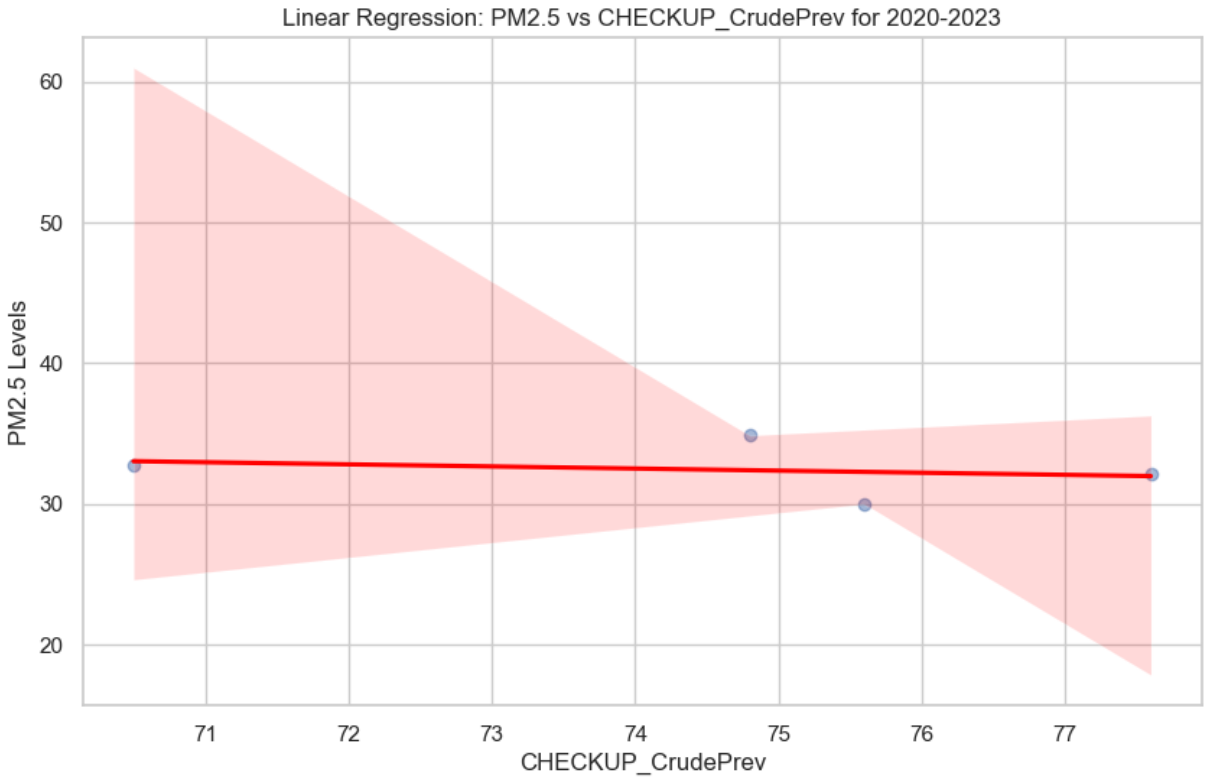

Linear Regression: PM2.5 vs BPMED_CrudePrev for 2020-2023
PM2.5 Levels
40
38
36
34
32
30
69.8
70.0
70.2
70.4
70.6
70.8
BPMED_CrudePrev

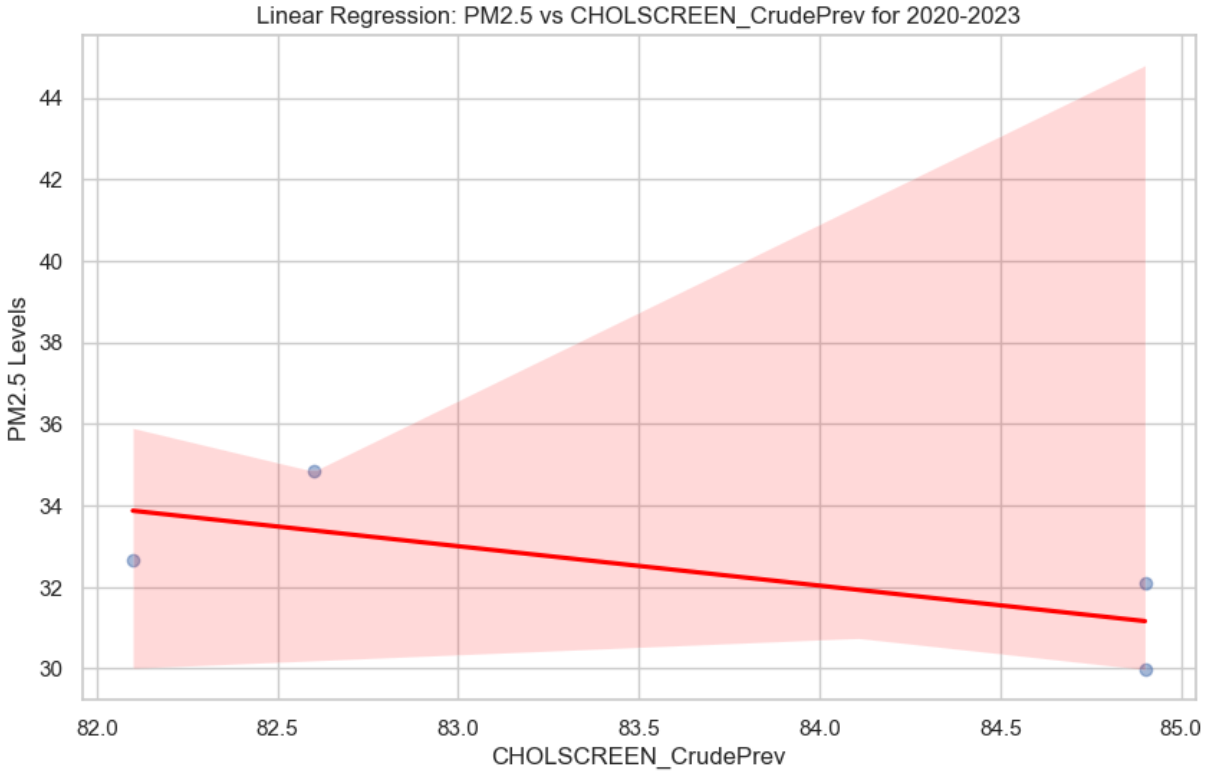

Linear Regression: PM2.5 vs CANCER_CrudePrev for 2020-2023
PM2.5 Levels
42
40
38
36
34
32
30
4.50
4.55
4.60
4.65
4.70
4.75
4.80
4.85
4.90
CANCER_CrudePrev

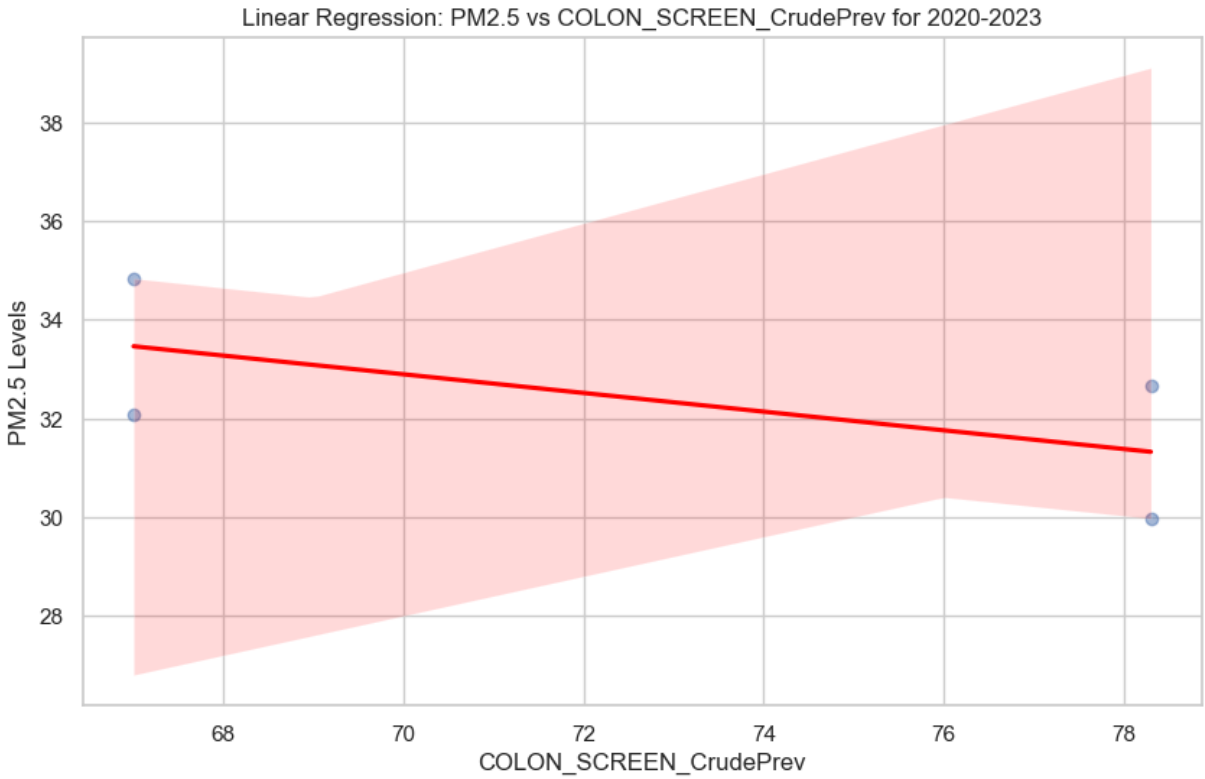

Linear Regression: PM2.5 vs CASTHMA_CrudePrev for 2020-2023
PM2.5 Levels
CASTHMA_CrudePrev
46
44
42
40
38
36
34
32
30
11.1
11.2
11.3
11.4
11.5
11.6

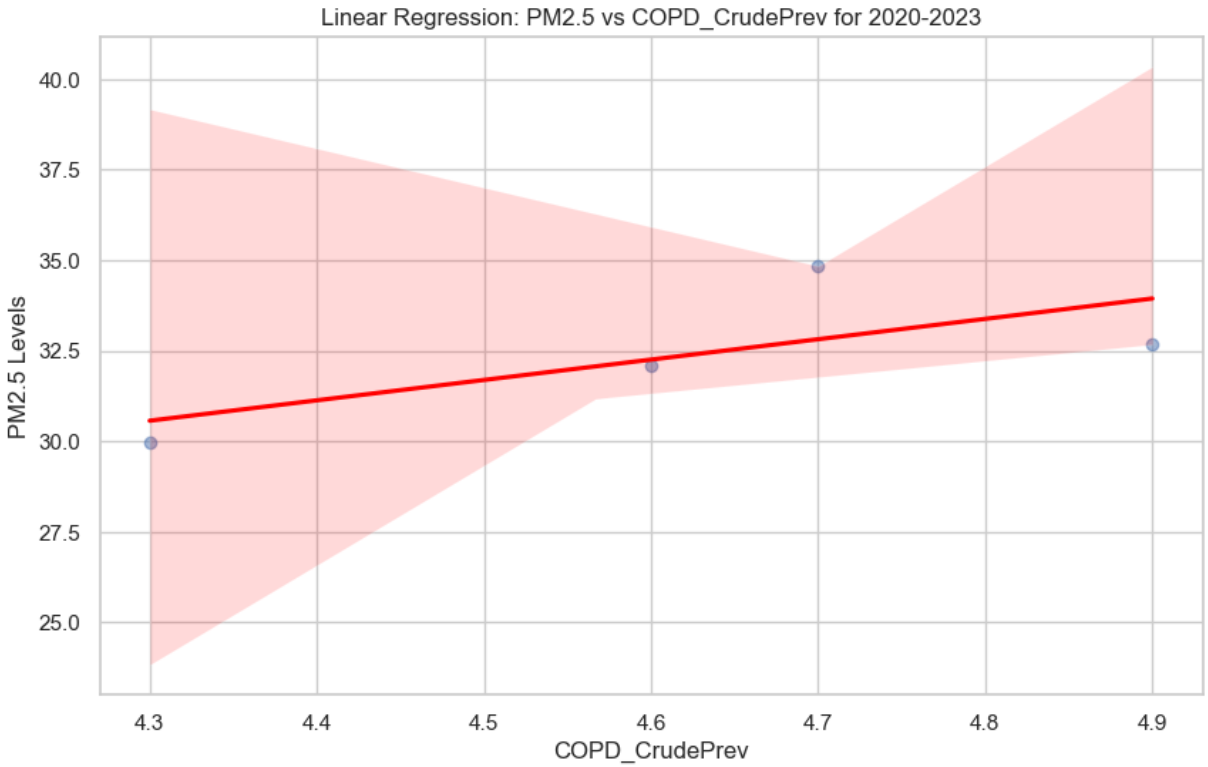

Linear Regression: PM2.5 vs CERVICAL_CrudePrev for 2020-2023
PM2.5 Levels
CERVICAL_CrudePrev
35
34
33
32
31
30
81.2
81.4
81.6
81.8
82.0
82.2
82.4
82.6

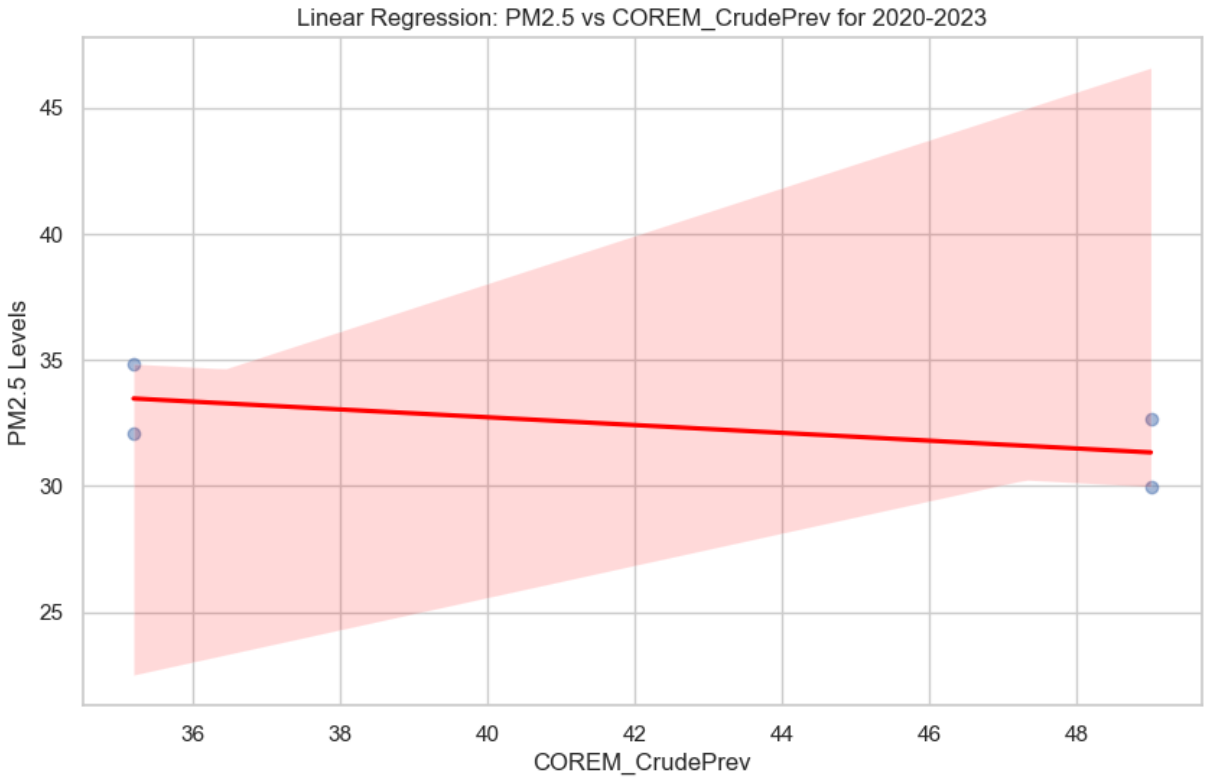

Linear Regression: PM2.5 vs CHD_CrudePrev for 2020-2023
PM2.5 Levels
34
32
30
28
26
24
22
20
4.0
4.1
4.2
4.3
4.4
4.5
CHD_CrudePrev

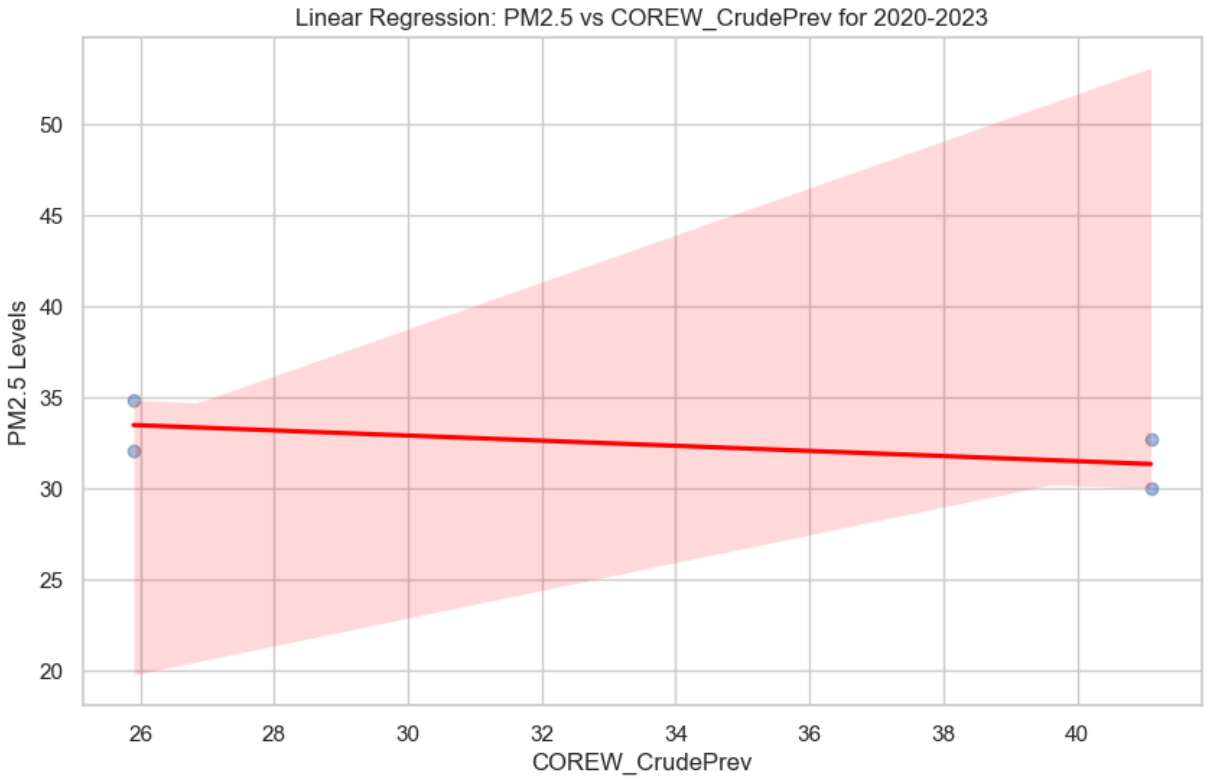

Linear Regression: PM2.5 vs CHECKUP_CrudePrev for 2020-2023
PM2.5 Levels
60
50
40
30
20
71
72
73
74
75
76
77
CHECKUP_CrudePrev

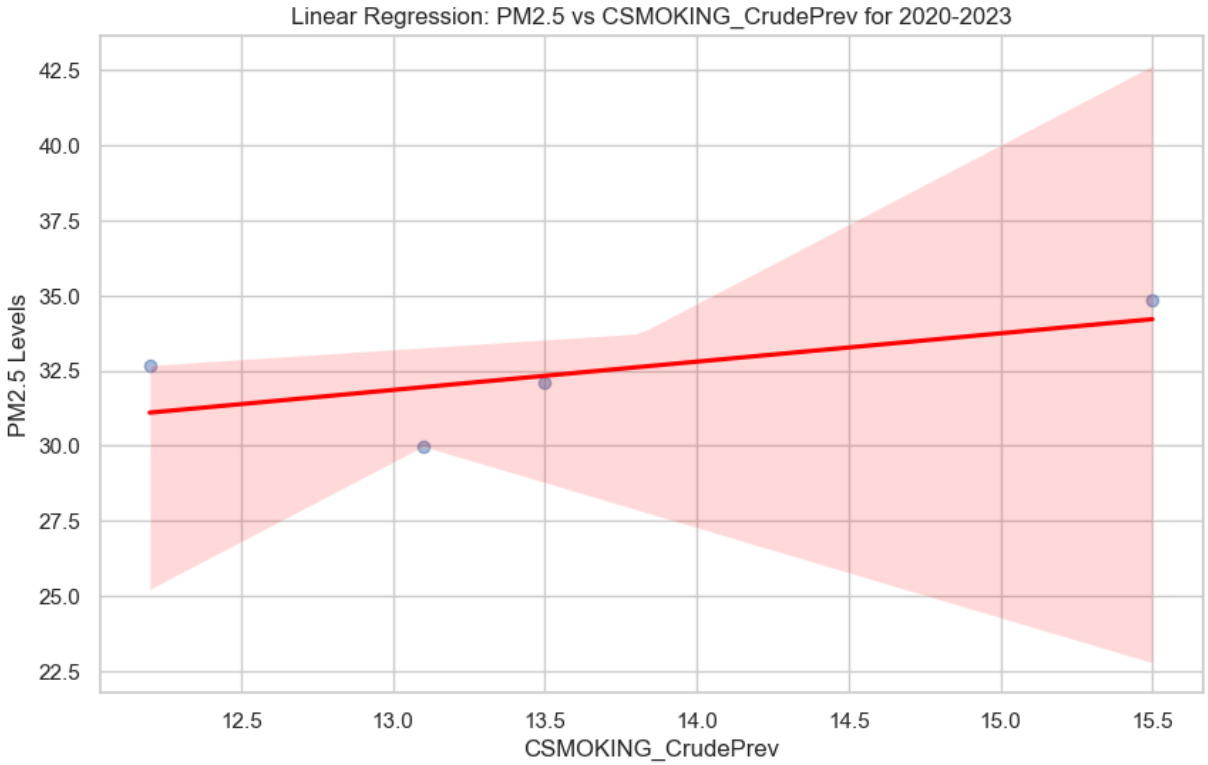

Linear Regression: PM2.5 vs CHOLSCREEN_CrudePrev for 2020-2023
PM2.5 Levels
CHOLSCREEN_CrudePrev
44
42
40
38
36
34
32
30
82.0
82.5
83.0
83.5
84.0
84.5
85.0

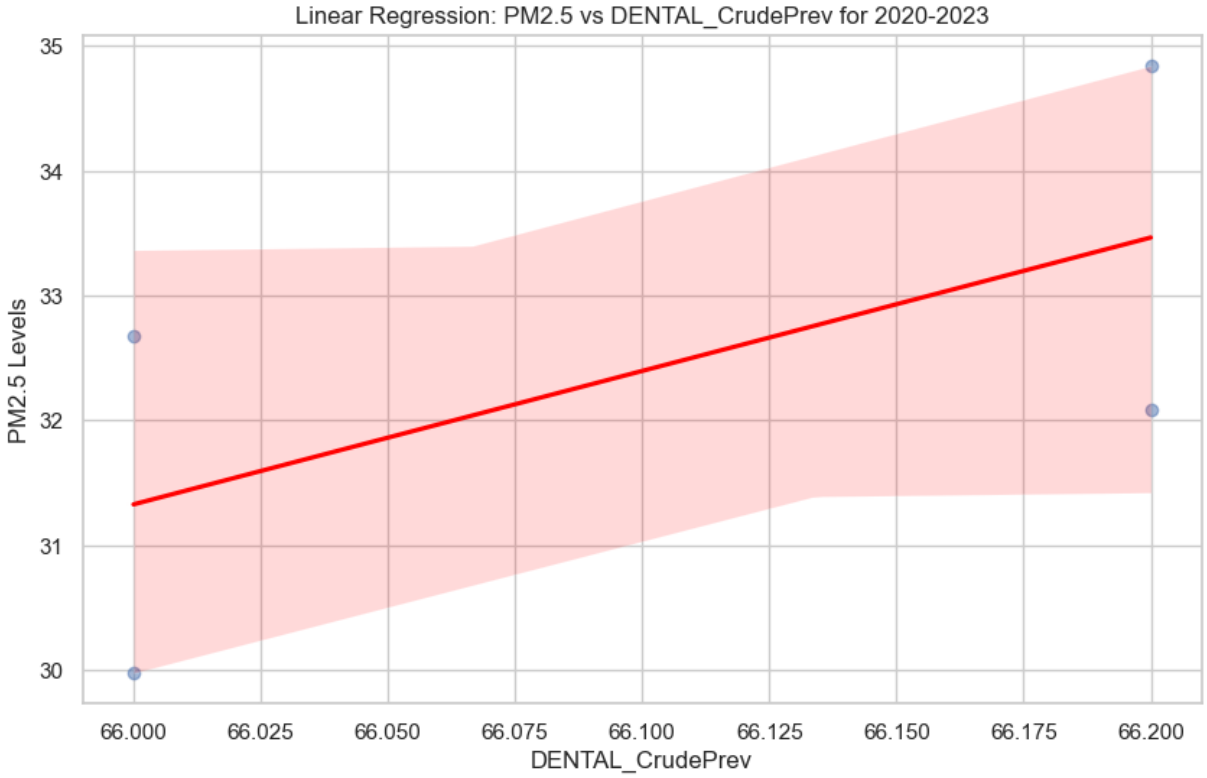

Linear Regression: PM2.5 vs COLON_SCREEN_CrudePrev for 2020-2023
PM2.5 Levels
COLON_SCREEN_CrudePrev
38
36
34
32
30
28
68
70
72
74
76
78

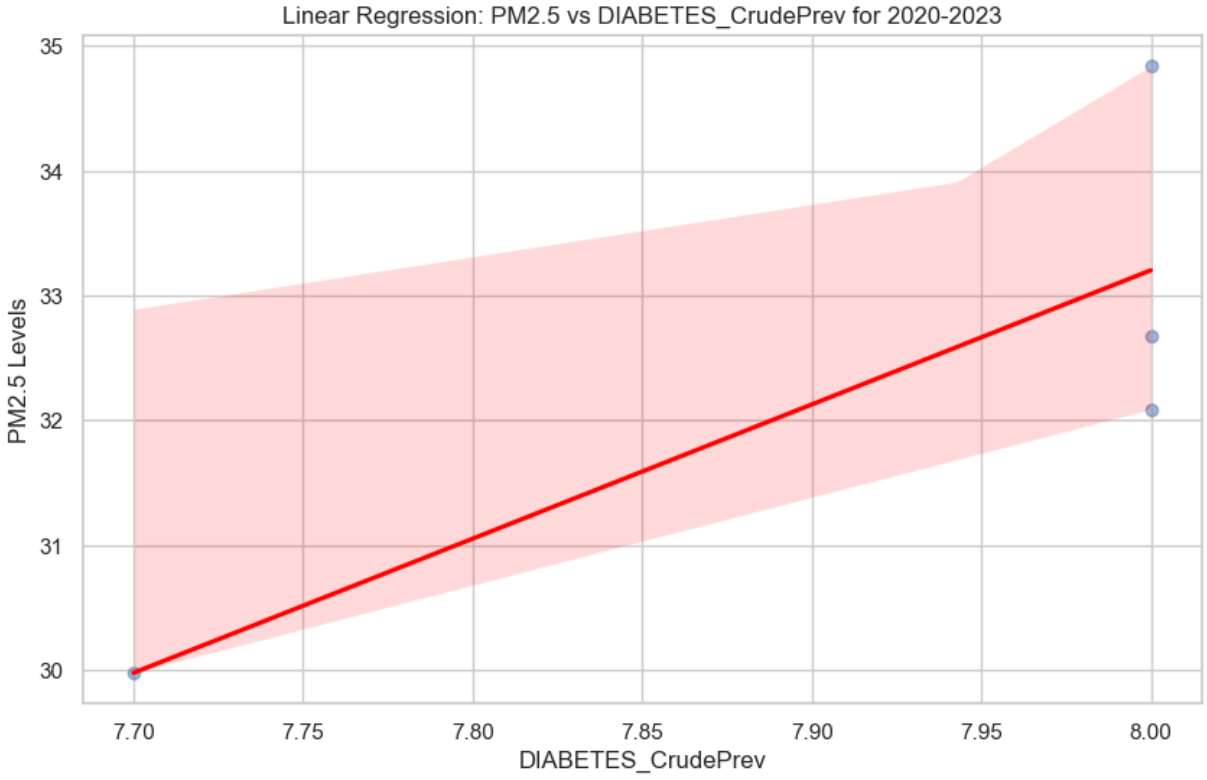

Linear Regression: PM2.5 vs COPD_CrudePrev for 2020-2023
PM2.5 Levels
40.0
37.5
35.0
32.5
30.0
27.5
25.0
4.3
4.4
4.5
4.6
4.7
4.8
4.9
COPD_CrudePrev

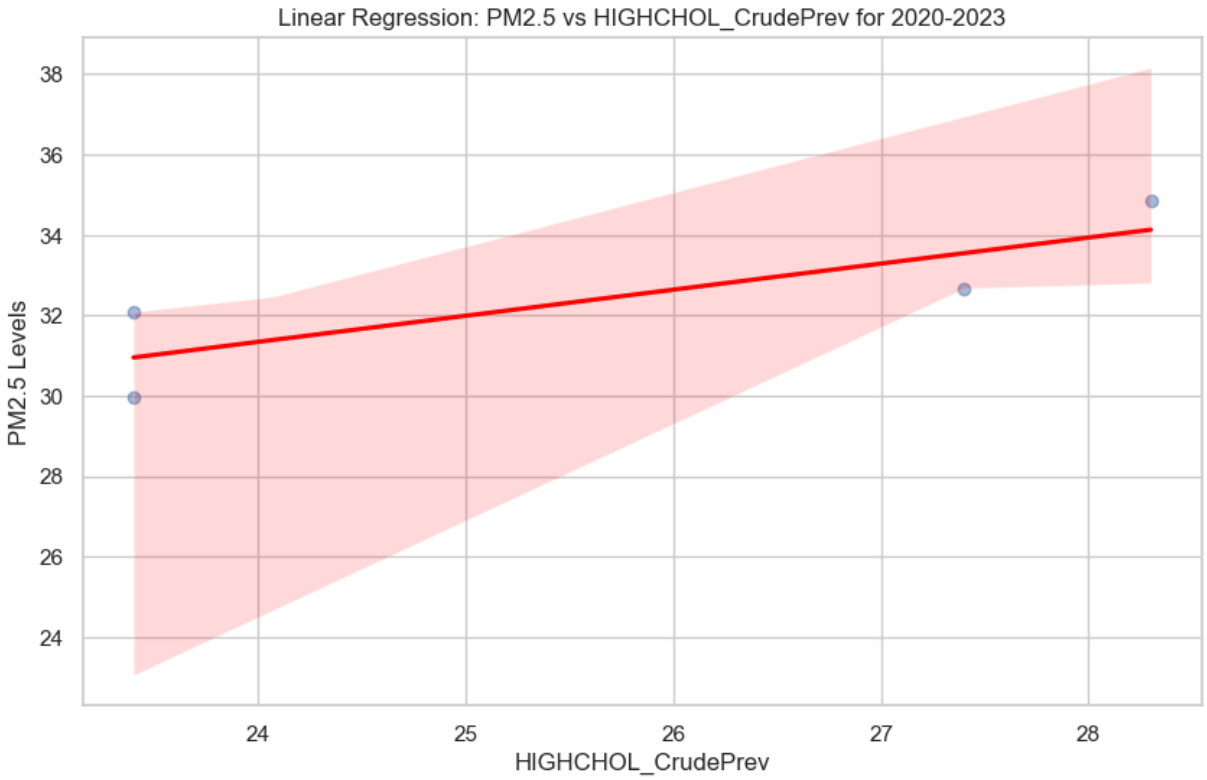

Linear Regression: PM2.5 vs COREM_CrudePrev for 2020-2023
PM2.5 Levels
45
40
35
30
25
36
38
40
42
44
46
48
COREM_CrudePrev

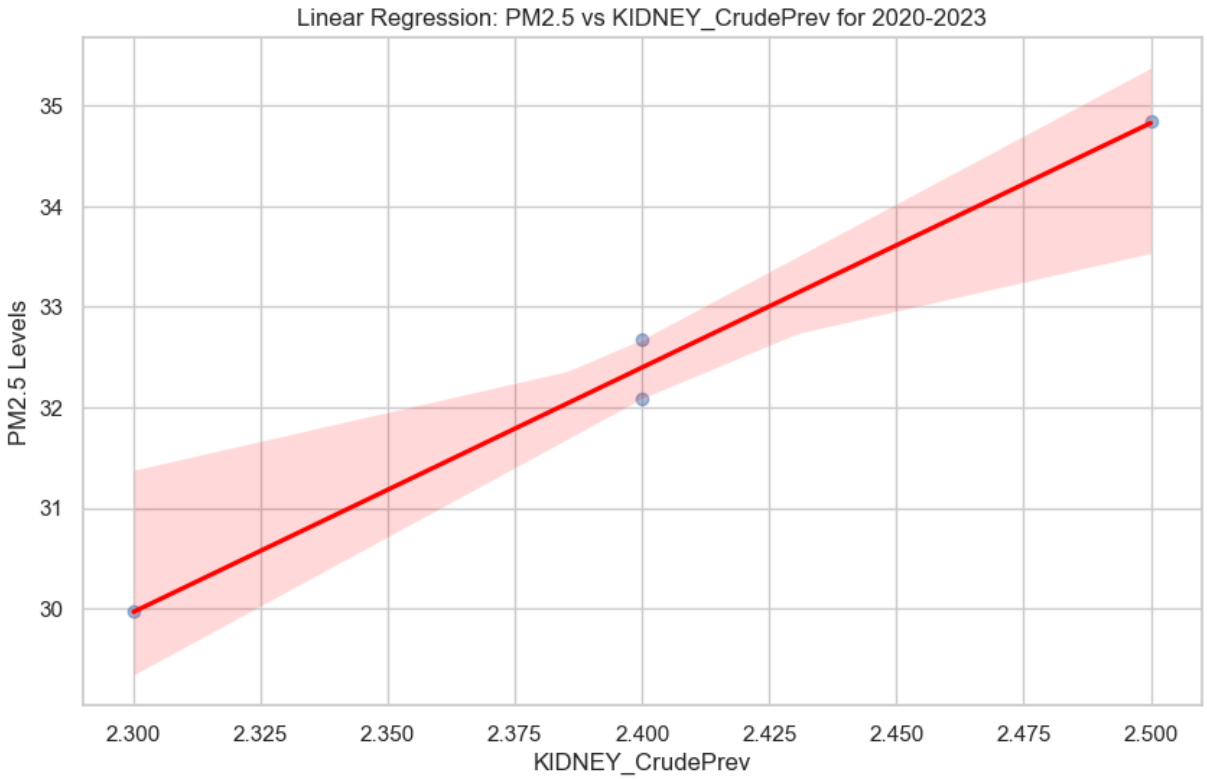

Linear Regression: PM2.5 vs COREW_CrudePrev for 2020-2023
PM2.5 Levels
50
45
40
35
30
25
20
26
28
30
32
34
36
38
40
COREW_CrudePrev

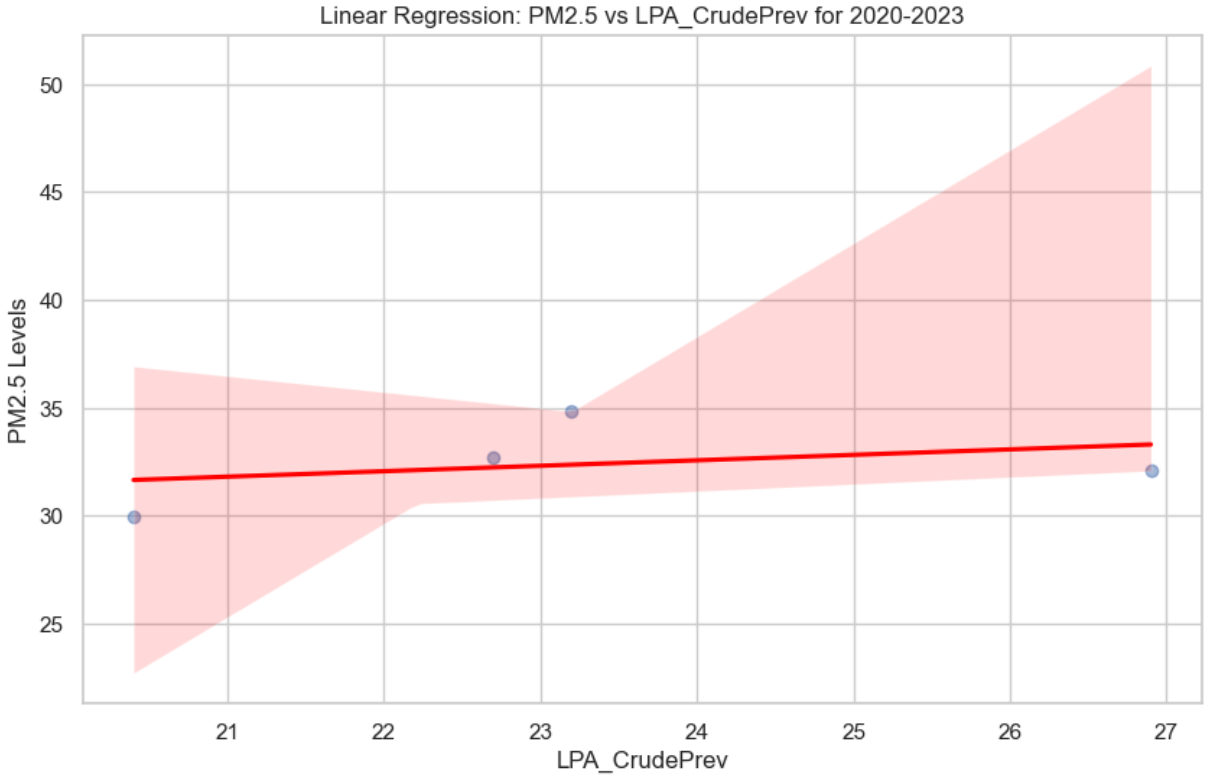

Linear Regression: PM2.5 vs CSMOKING_CrudePrev for 2020-2023
PM2.5 Levels
42.5
40.0
37.5
35.0
32.5
30.0
27.5
25.0
22.5
12.5
13.0
13.5
14.0
14.5
15.0
15.5
CSMOKING_CrudePrev

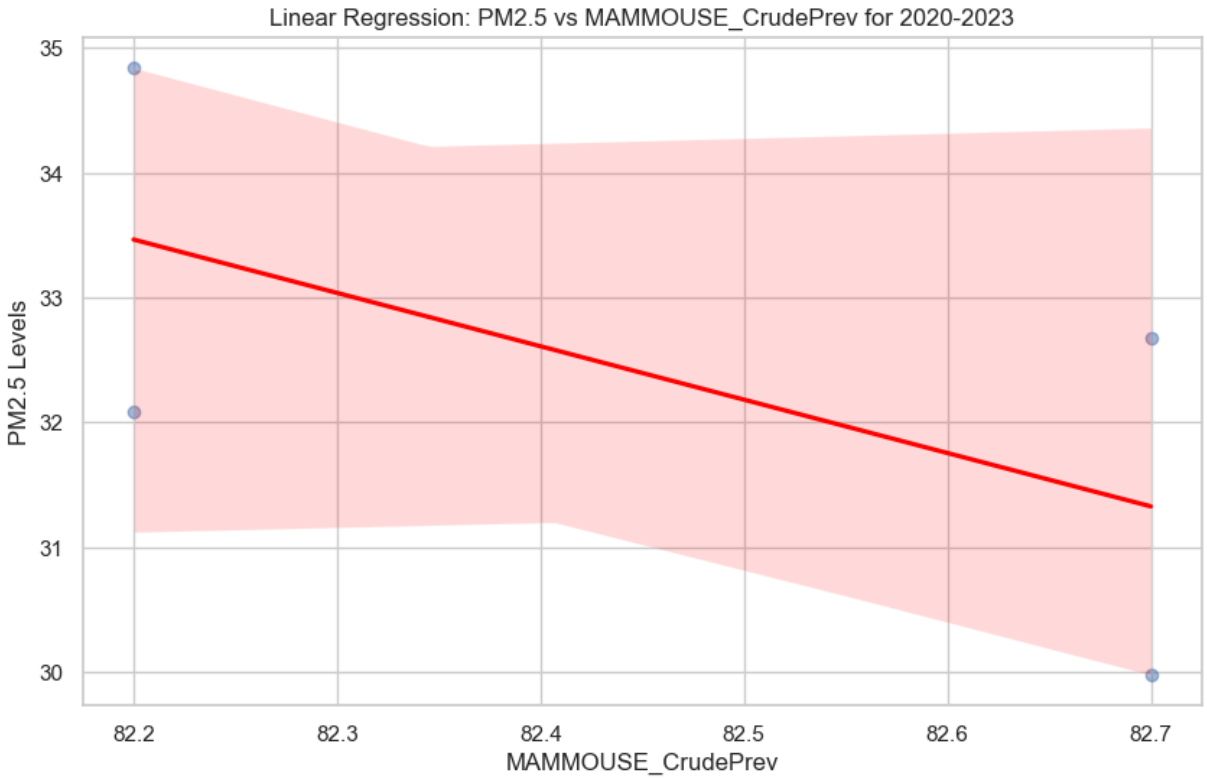

Linear Regression: PM2.5 vs DENTAL_CrudePrev for 2020-2023
PM2.5 Levels
DENTAL_CrudePrev
35
34
33
32
31
30
66.000
66.025
66.050
66.075
66.100
66.125
66.150
66.175
66.200

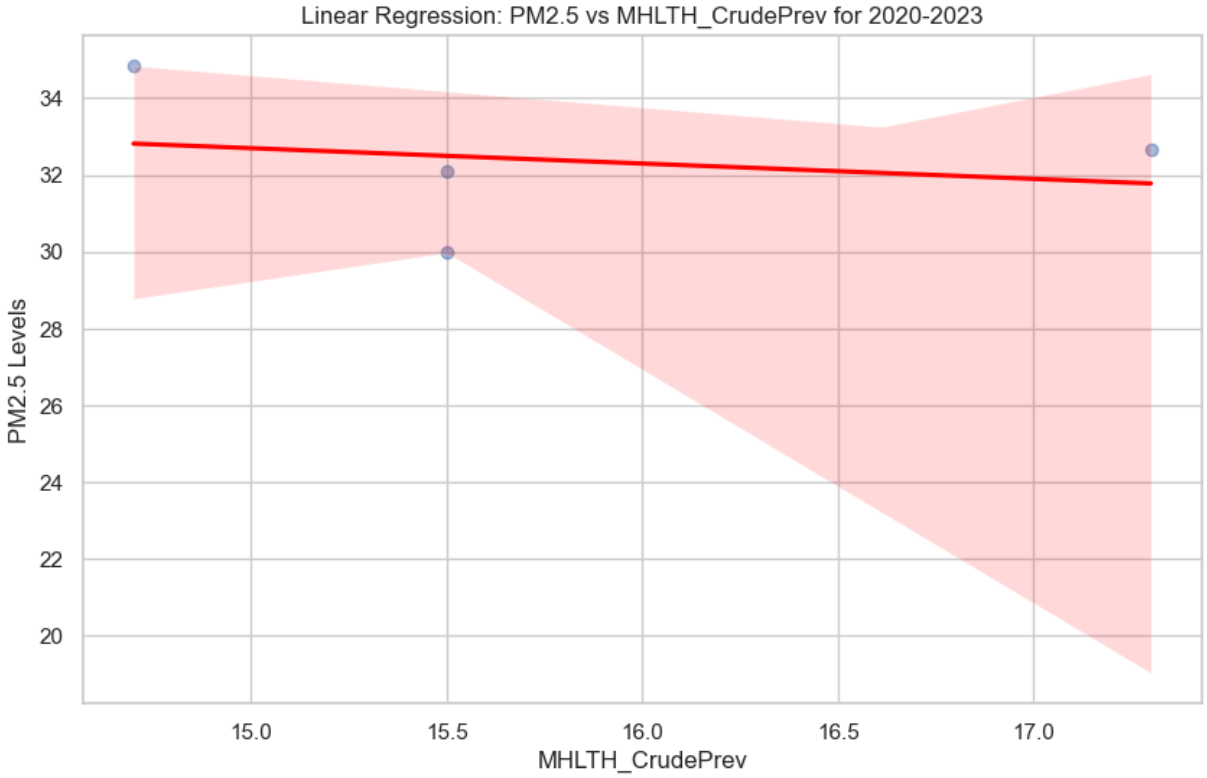

Linear Regression: PM2.5 vs DIABETES_CrudePrev for 2020-2023
PM2.5 Levels
DIABETES_CrudePrev
35
34
33
32
31
30
7.70
7.75
7.80
7.85
7.90
7.95
8.00

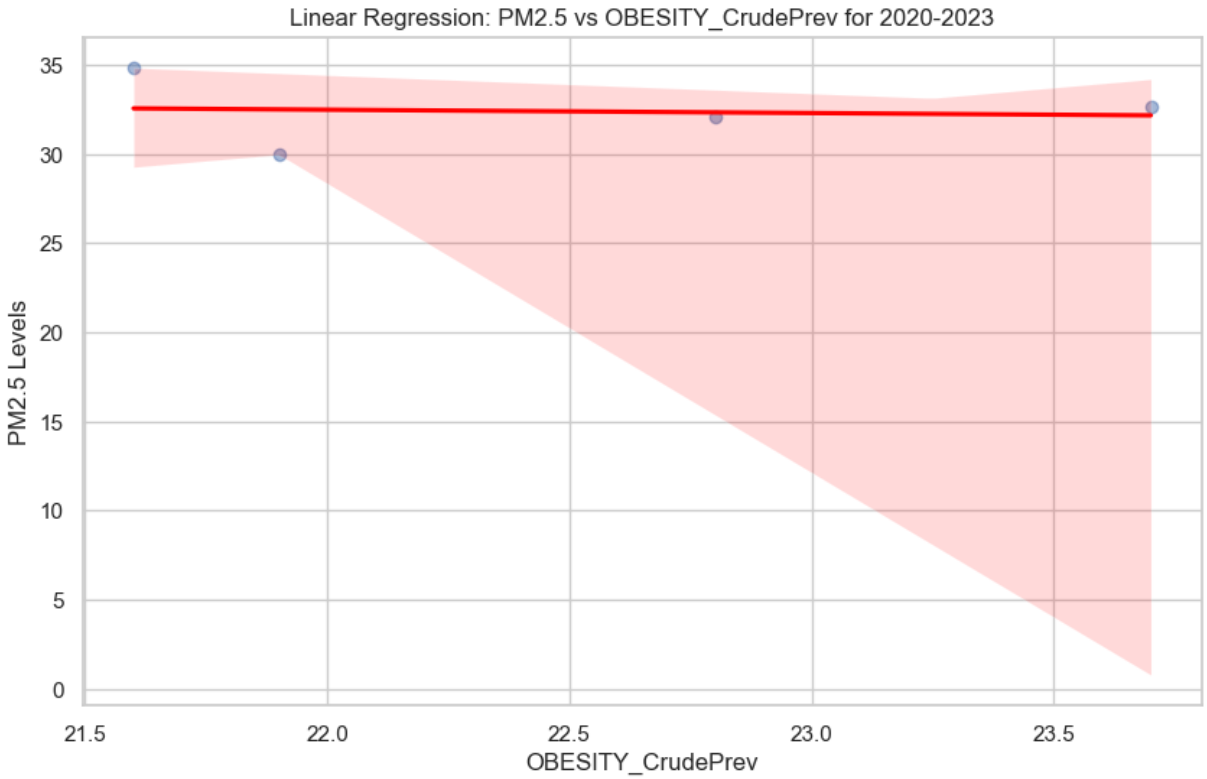

Linear Regression: PM2.5 vs HIGHCHOL_CrudePrev for 2020-2023
PM2.5 Levels
38
36
34
32
30
28
26
24
24
25
26
27
28
HIGHCHOL_CrudePrev


Linear Regression: PM2.5 vs KIDNEY_CrudePrev for 2020-2023
PM2.5 Levels
35
34
33
32
31
30
2.300
2.325
2.350
2.375
2.400
2.425
2.450
2.475
2.500
KIDNEY_CrudePrev

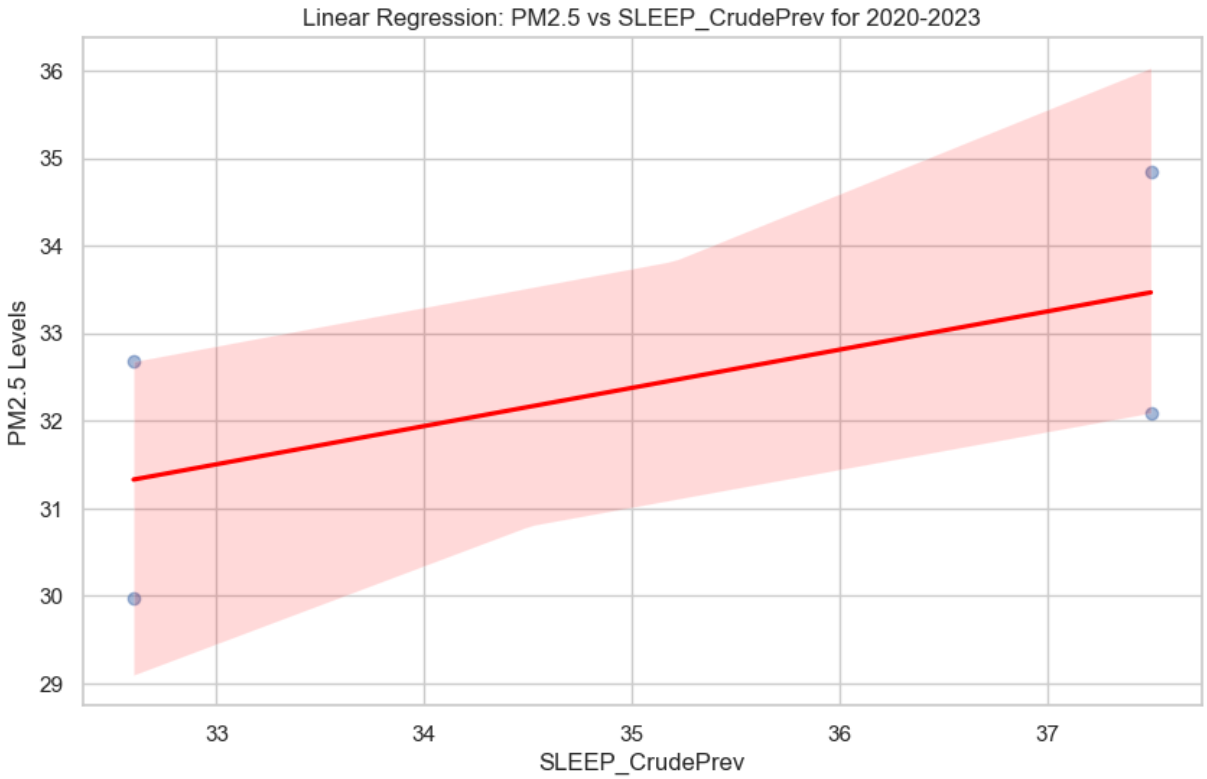

Linear Regression: PM2.5 vs LPA_CrudePrev for 2020-2023
PM2.5 Levels
50
45
40
35
30
25
21
22
23
24
25
26
27
LPA_CrudePrev

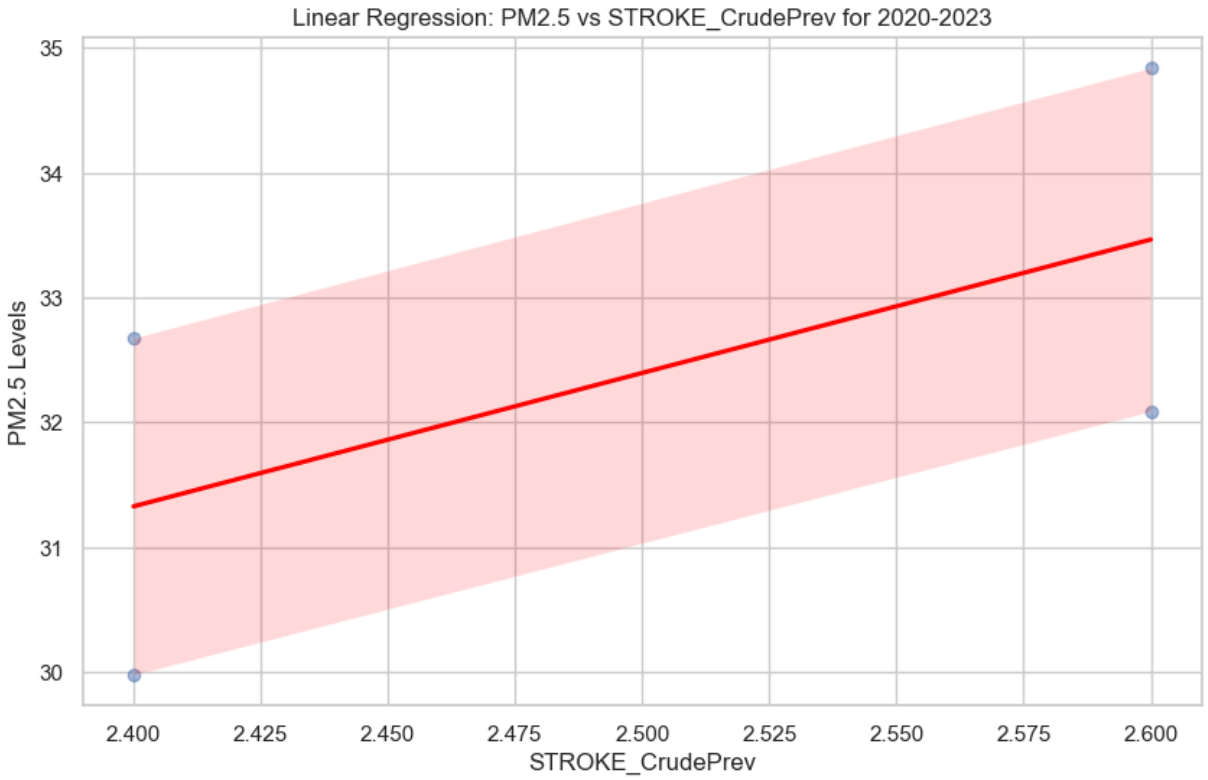

Linear Regression: PM2.5 vs MAMMOUSE_CrudePrev for 2020-2023
PM2.5 Levels
35
34
33
32
31
30
82.2
82.3
82.4
82.5
82.6
82.7
MAMMOUSE_CrudePrev

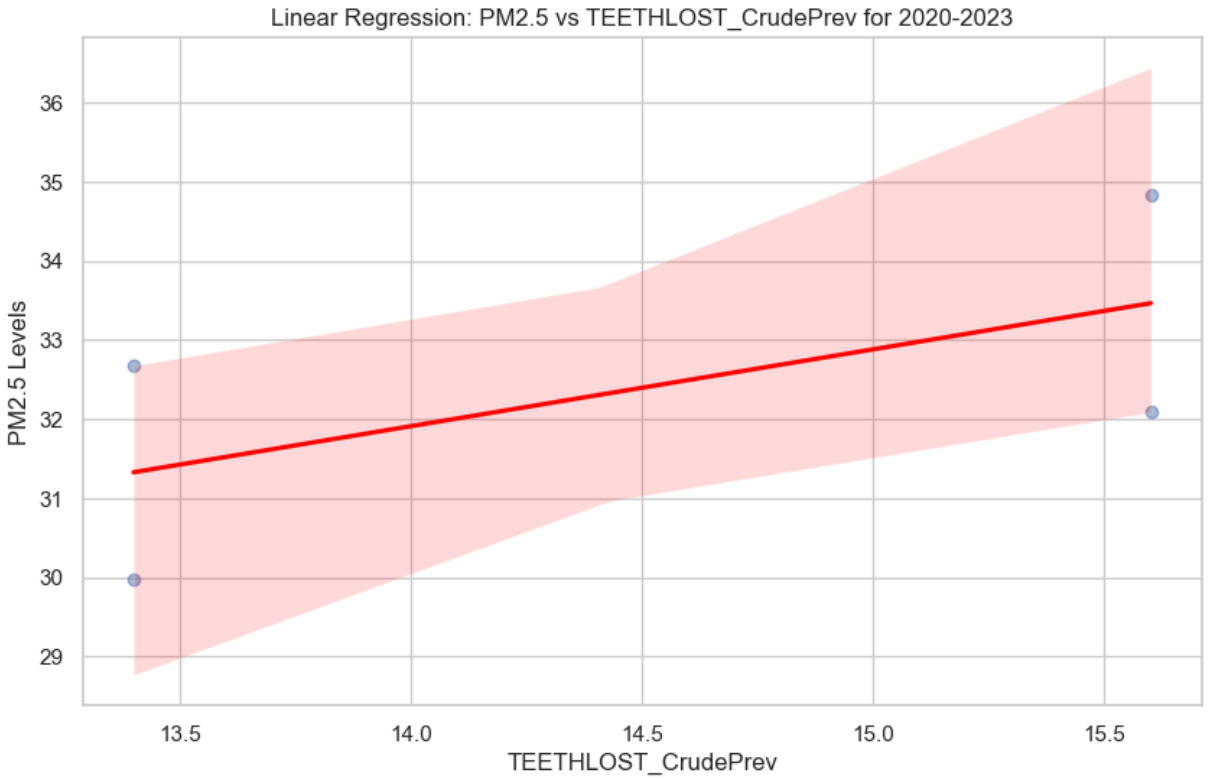

Linear Regression: PM2.5 vs MHLTH_CrudePrev for 2020-2023
PM2.5 Levels
34
32
30
28
26
24
22
20
15.0
15.5
16.0
16.5
17.0
MHLTH_CrudePrev


Linear Regression: PM2.5 vs OBESITY_CrudePrev for 2020-2023
PM2.5 Levels
35
30
25
20
15
10
5
0
21.5
22.0
22.5
23.0
23.5
OBESITY_CrudePrev

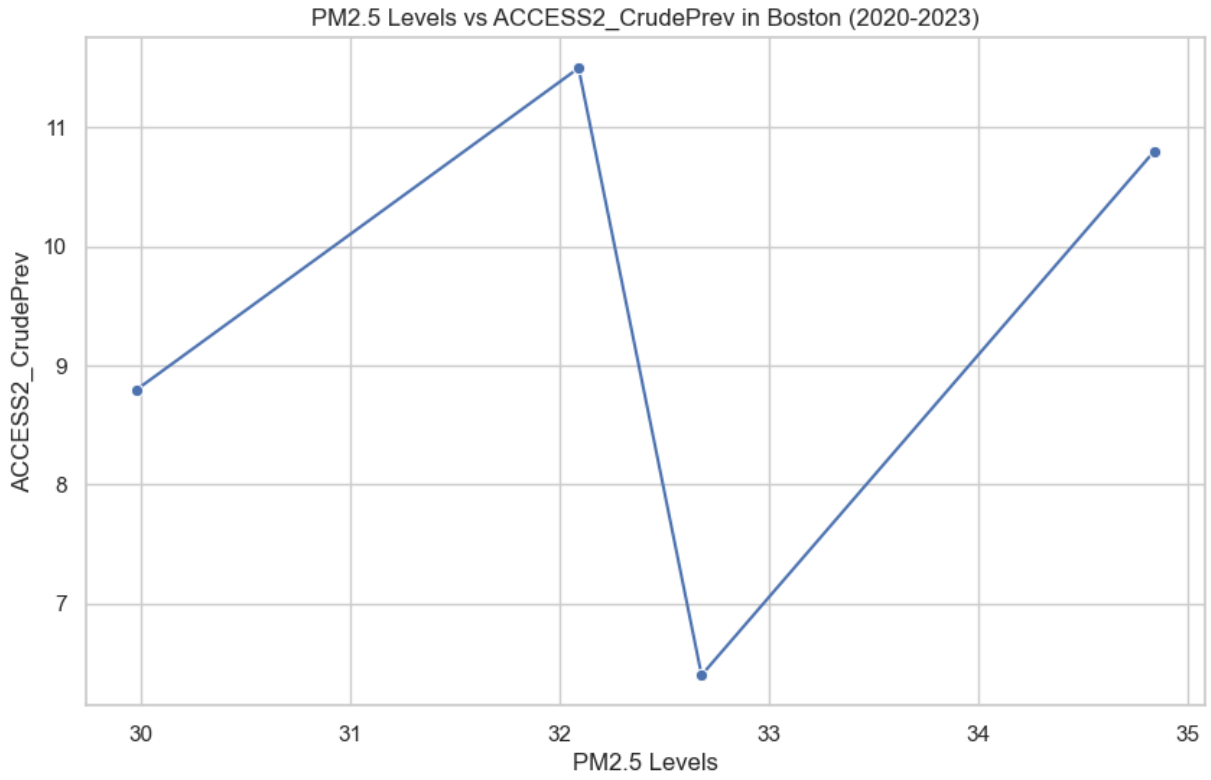

Linear Regression: PM2.5 vs PHLTH_CrudePrev for 2020-2023
PM2.5 Levels
PHLTH_CrudePrev
40
35
30
25
20
15
10
5
8.0
8.5
9.0
9.5
10.0
10.5
11.0

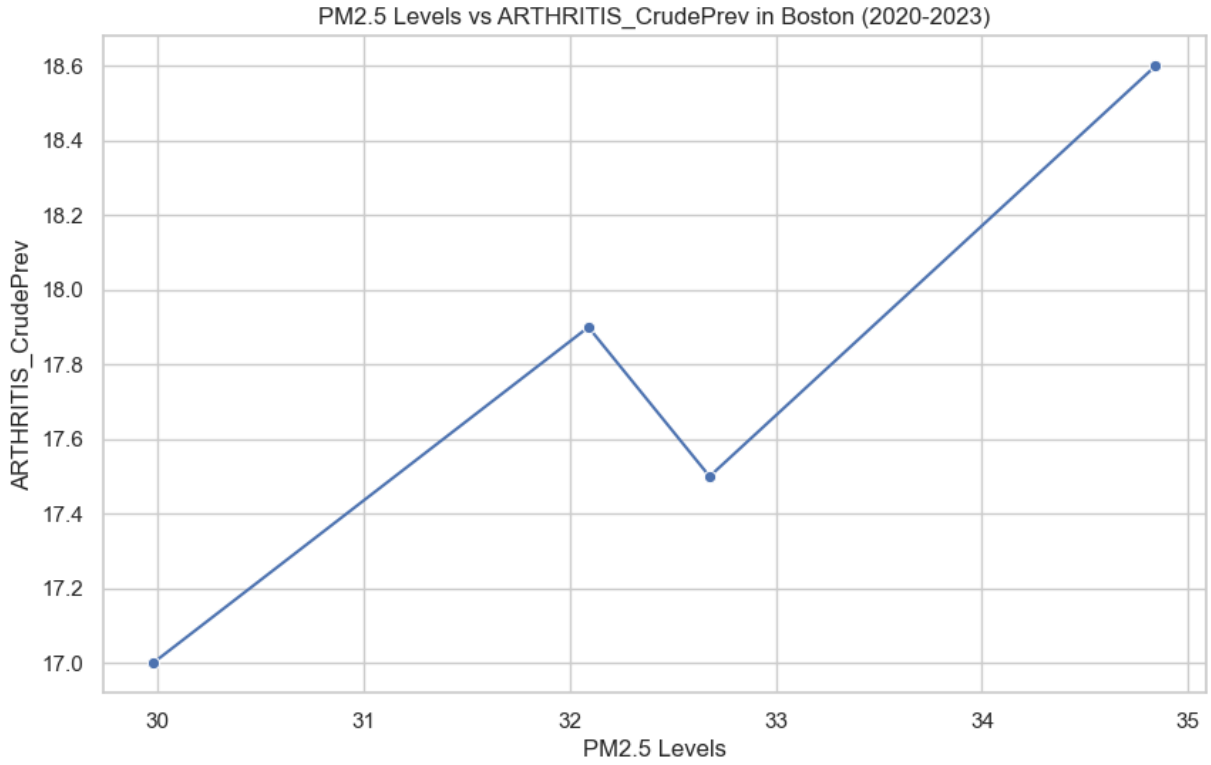

Linear Regression: PM2.5 vs SLEEP_CrudePrev for 2020-2023
PM2.5 Levels
SLEEP_CrudePrev
36
35
34
33
32
31
30
29
33
34
35
36
37

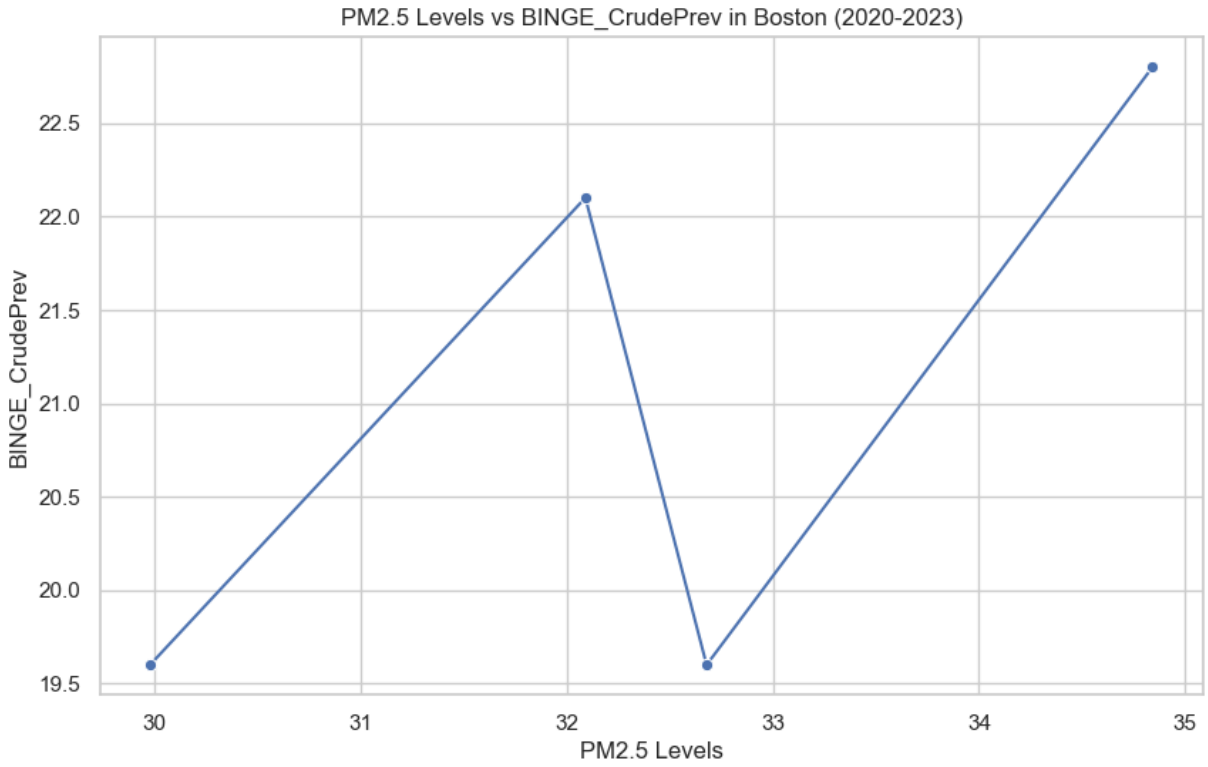


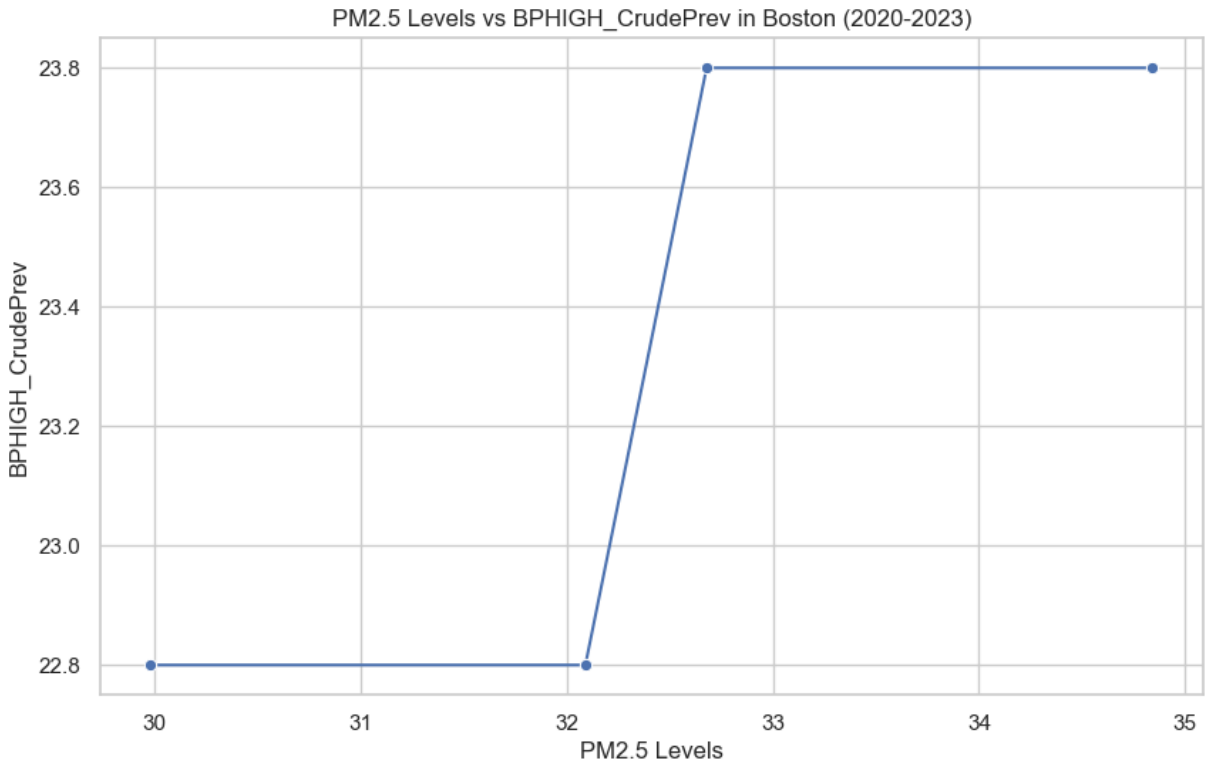


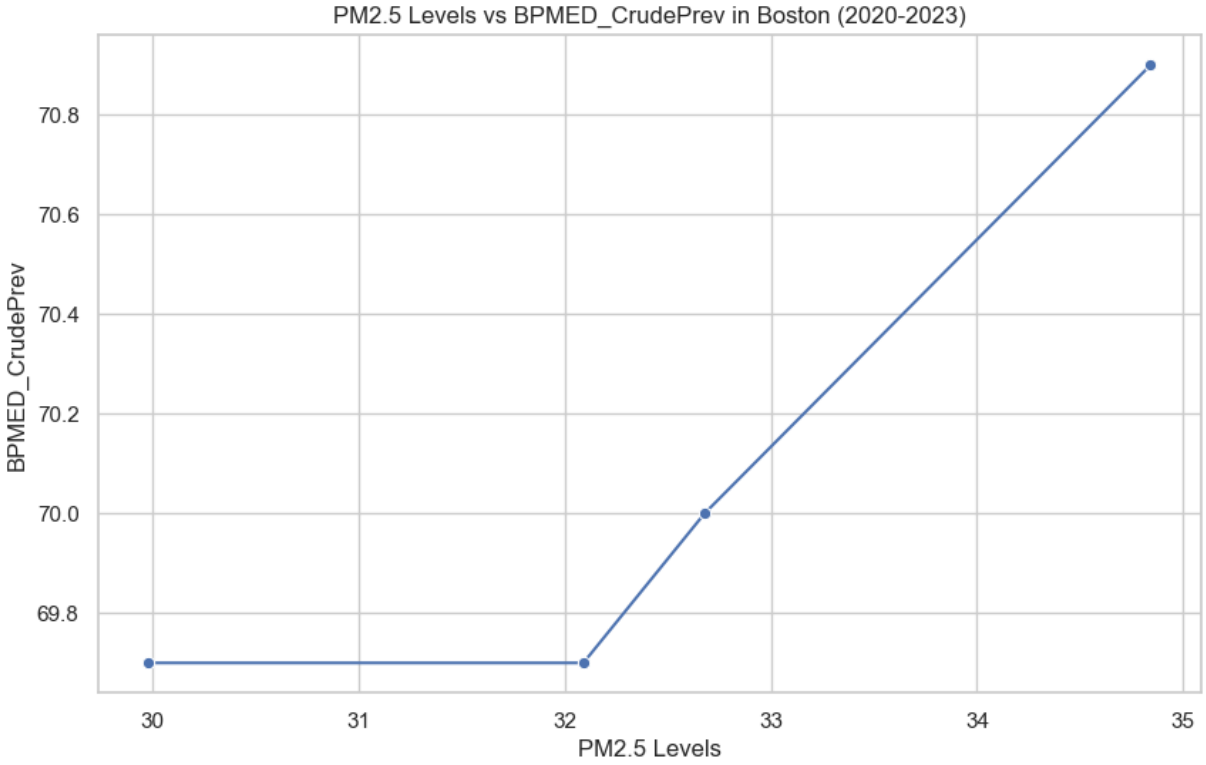
In [18]:
```
for column in crude_columns_boston:
    if column != 'pm25':
        plt.figure(figsize=(10, 6))
        sns.lineplot(x='pm25', y=column, data=boston_df, marker='o')
        plt.title(f'PM2.5 Levels vs {column} in Boston (2020-2023)')
        plt.xlabel('PM2.5 Levels')
        plt.ylabel(column)
        plt.show()
```

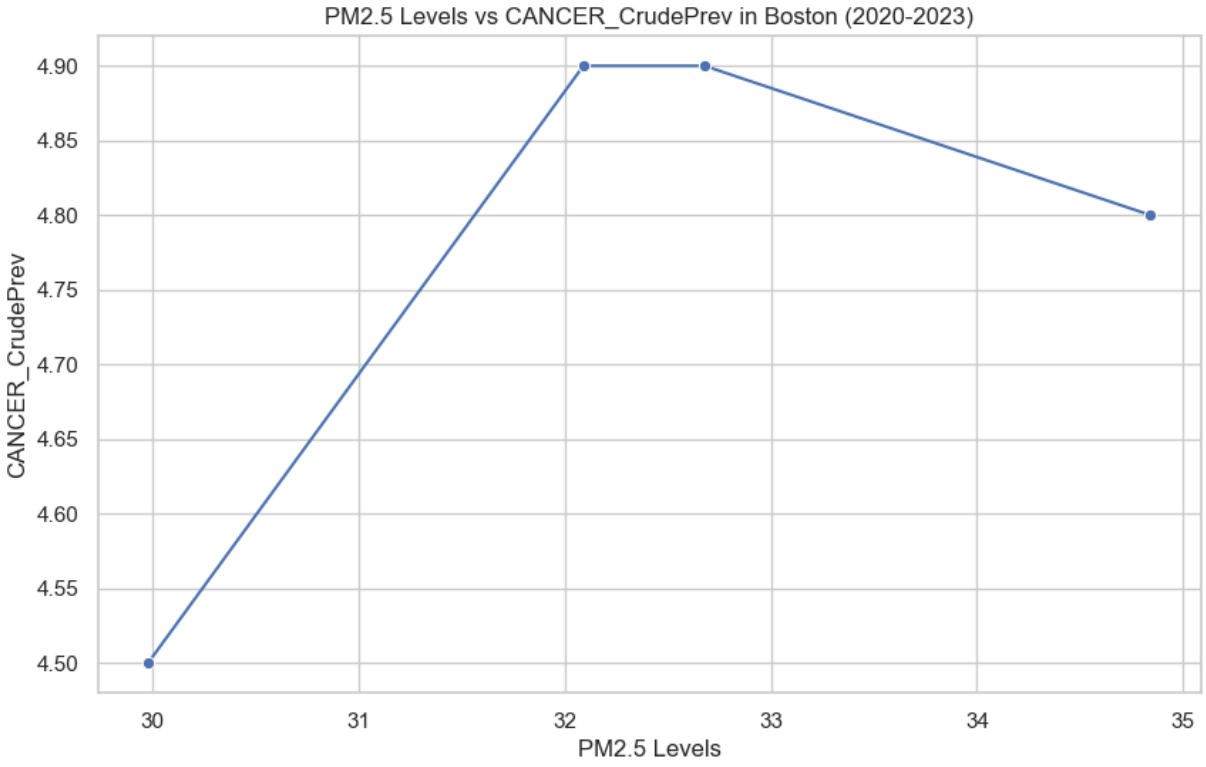

PM2.5 Levels vs ACCESS2_CrudePrev in Boston (2020-2023)
ACCESS2_CrudePrev
11
10
9
8
7
30
31
32
33
34
35
PM2.5 Levels

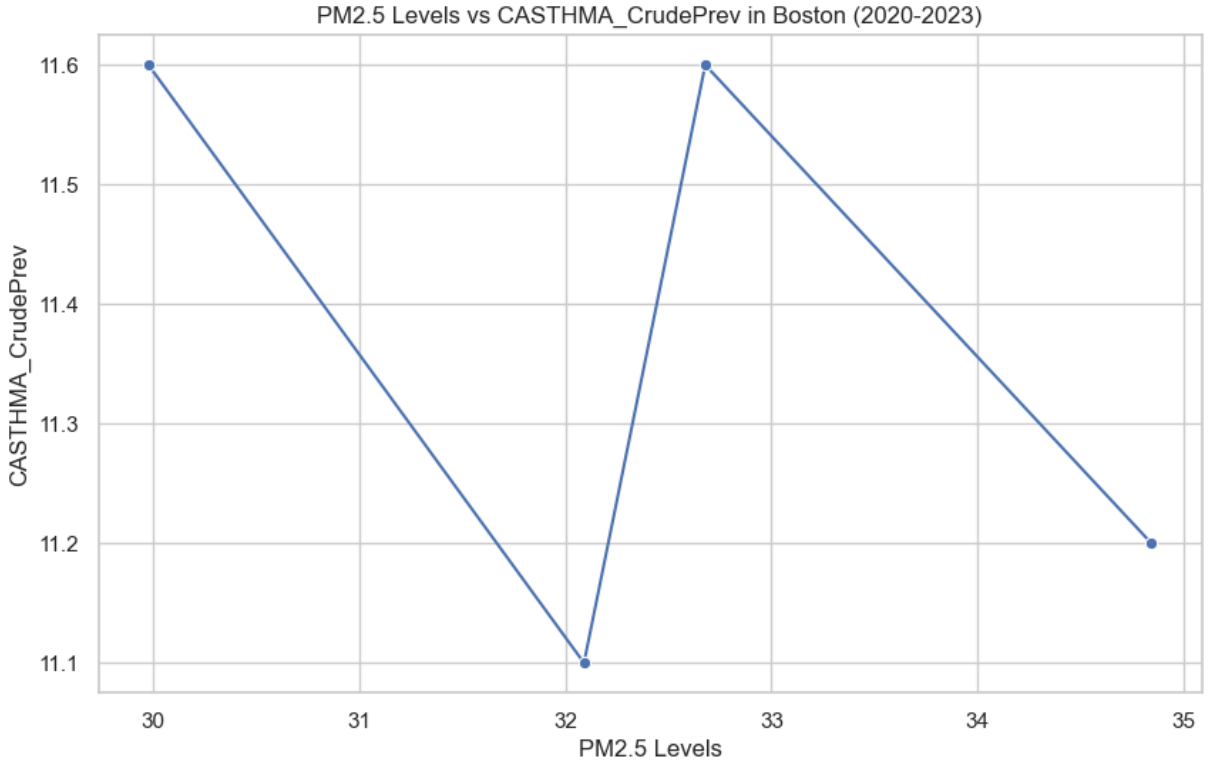

PM2.5 Levels vs ARTHRITIS_CrudePrev in Boston (2020-2023)
ARTHRITIS_CrudePrev
18.6
18.4
18.2
18.0
17.8
17.6
17.4
17.2
17.0
30
31
32
33
34
35
PM2.5 Levels

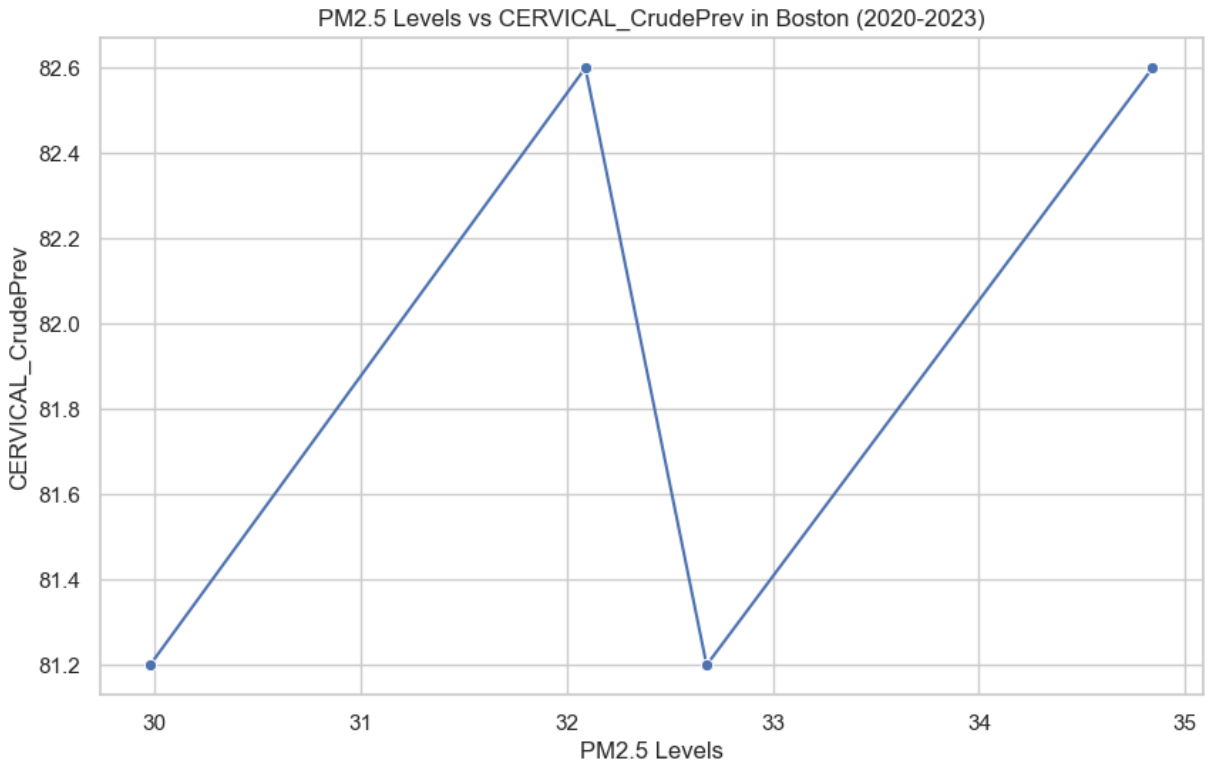

PM2.5 Levels vs BINGE_CrudePrev in Boston (2020-2023)
BINGE_CrudePrev
22.5
22.0
21.5
21.0
20.5
20.0
19.5
30
31
32
33
34
35
PM2.5 Levels

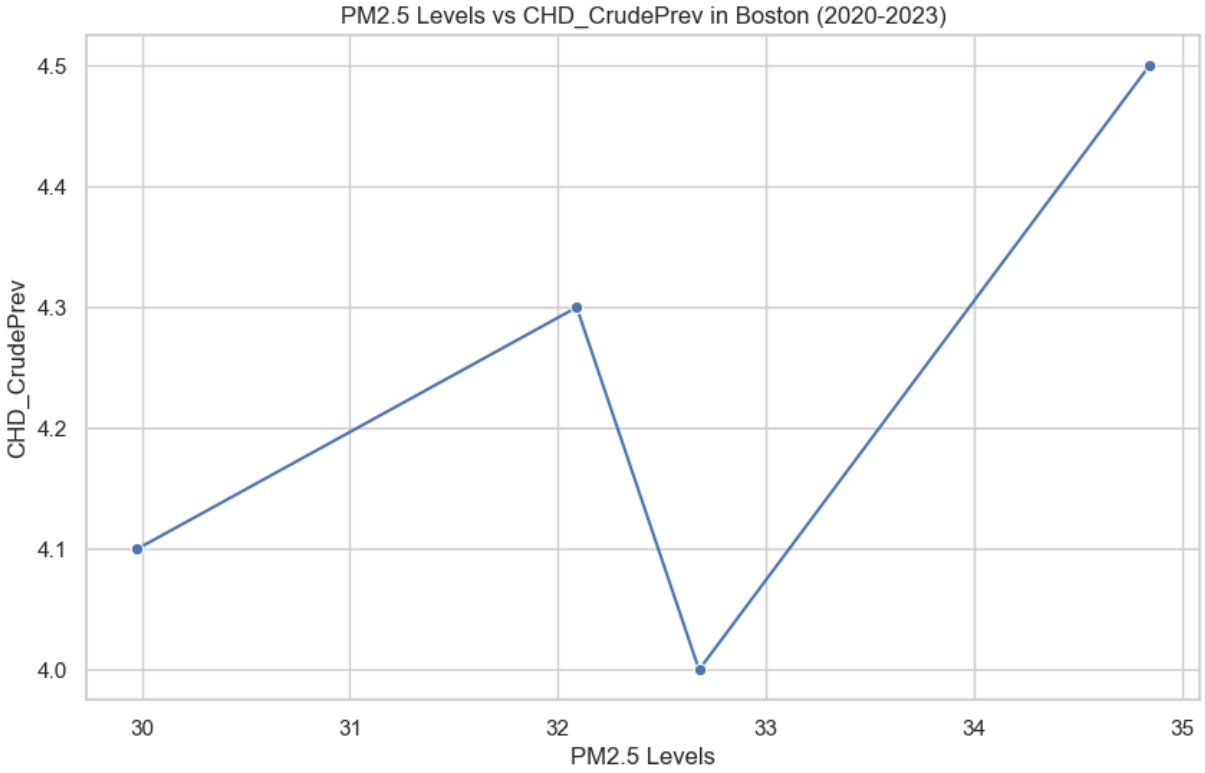

PM2.5 Levels vs BPHIGH_CrudePrev in Boston (2020-2023)
BPHIGH_CrudePrev
23.8
23.6
23.4
23.2
23.0
22.8
30
31
32
33
34
35
PM2.5 Levels

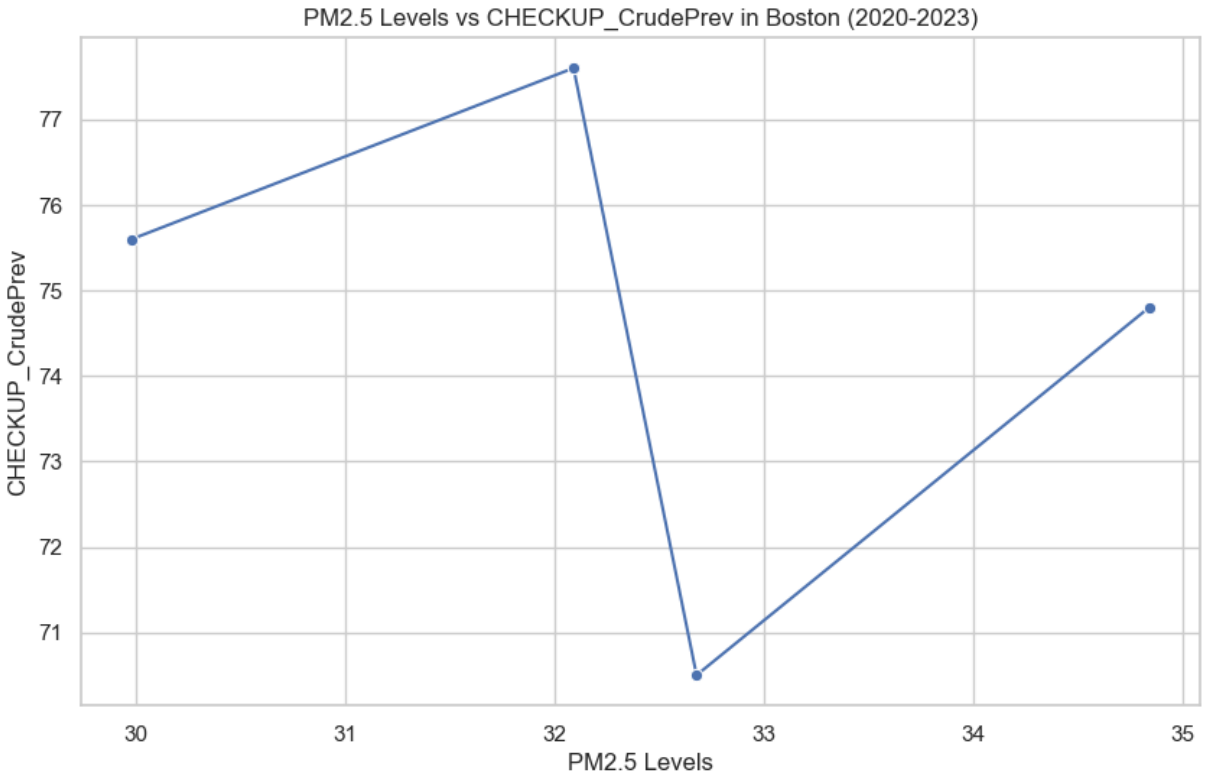

PM2.5 Levels vs BPMED_CrudePrev in Boston (2020-2023)
BPMED_CrudePrev
70.8
70.6
70.4
70.2
70.0
69.8
30
31
32
33
34
35
PM2.5 Levels

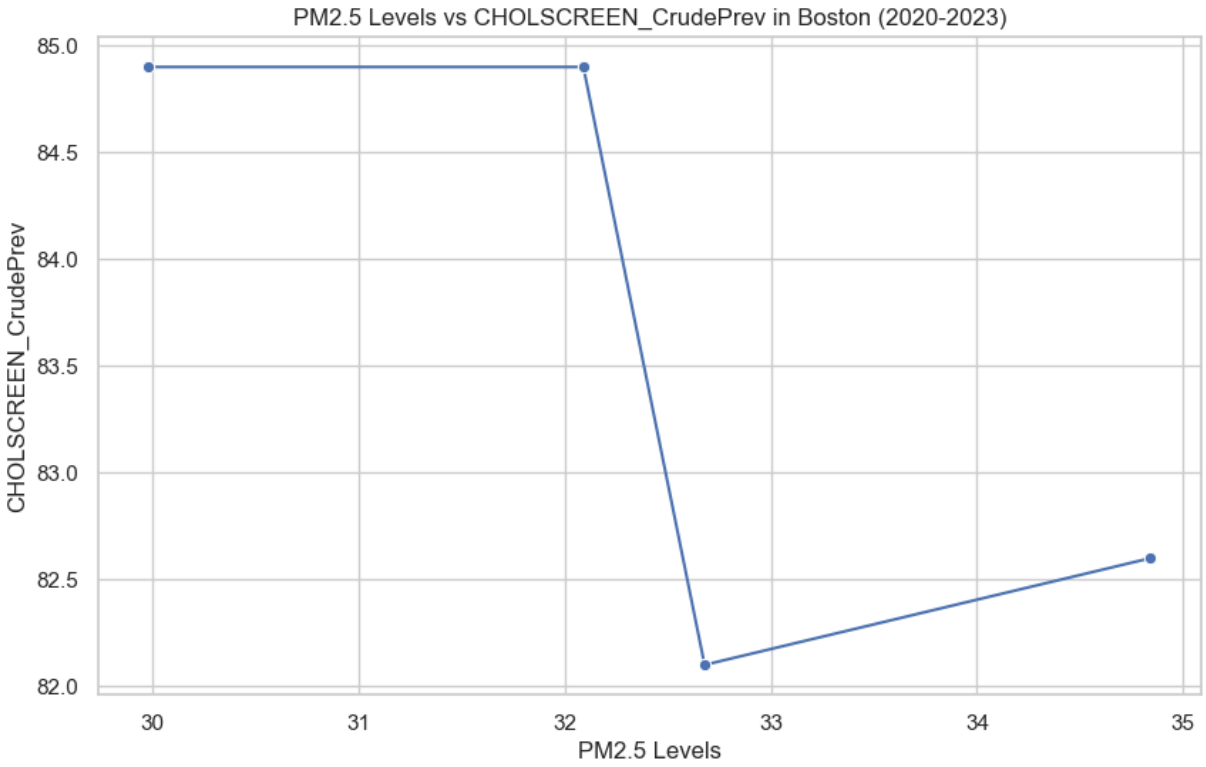

PM2.5 Levels vs CANCER_CrudePrev in Boston (2020-2023)
CANCER_CrudePrev
4.90
4.85
4.80
4.75
4.70
4.65
4.60
4.55
4.50
30
31
32
33
34
35
PM2.5 Levels

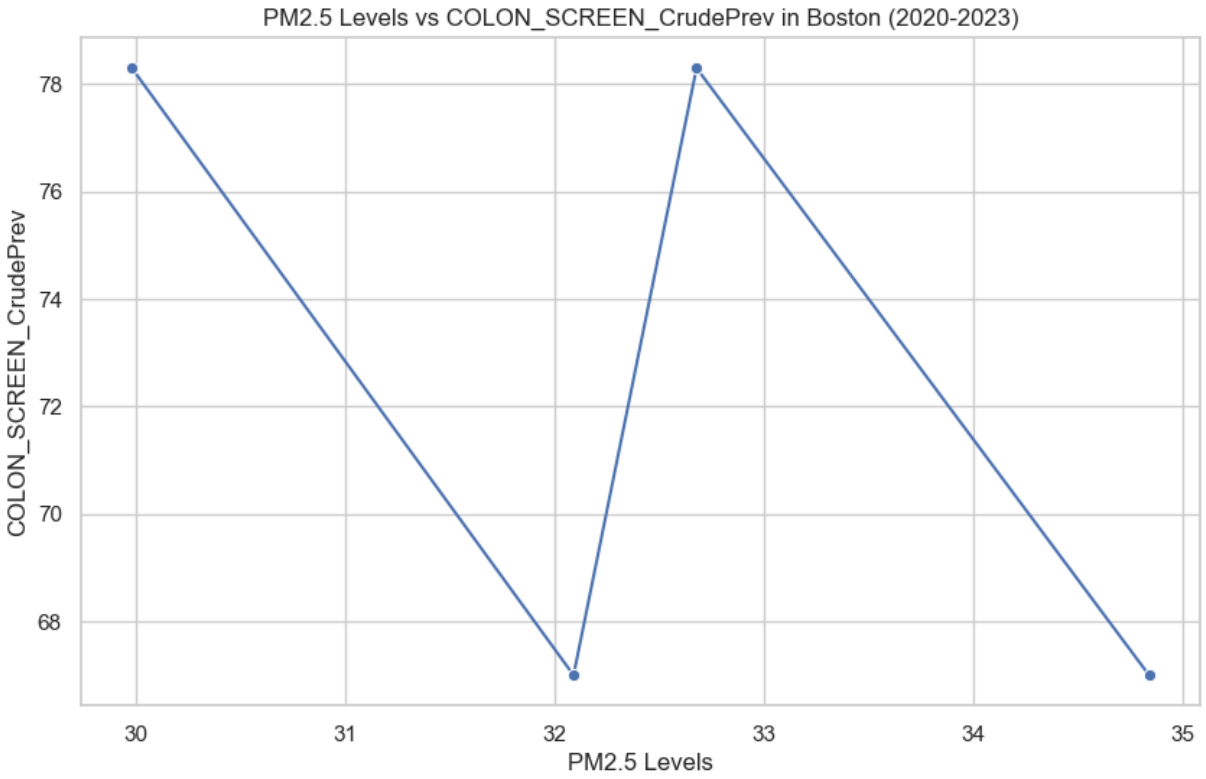

PM2.5 Levels vs CASTHMA_CrudePrev in Boston (2020-2023)
CASTHMA_CrudePrev
11.6
11.5
11.4
11.3
11.2
11.1
30
31
32
33
34
35
PM2.5 Levels

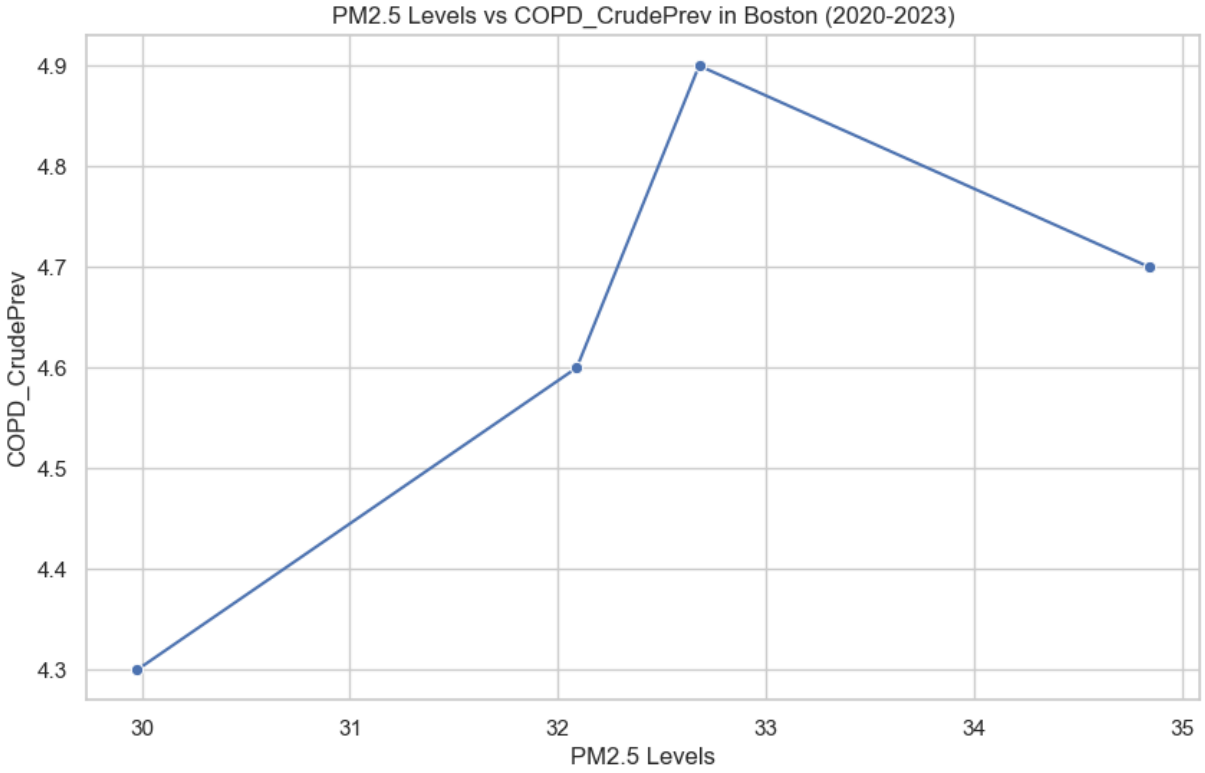

PM2.5 Levels vs CERVICAL_CrudePrev in Boston (2020-2023)
CERVICAL_CrudePrev
82.6
82.4
82.2
82.0
81.8
81.6
81.4
81.2
30
31
32
33
34
35
PM2.5 Levels

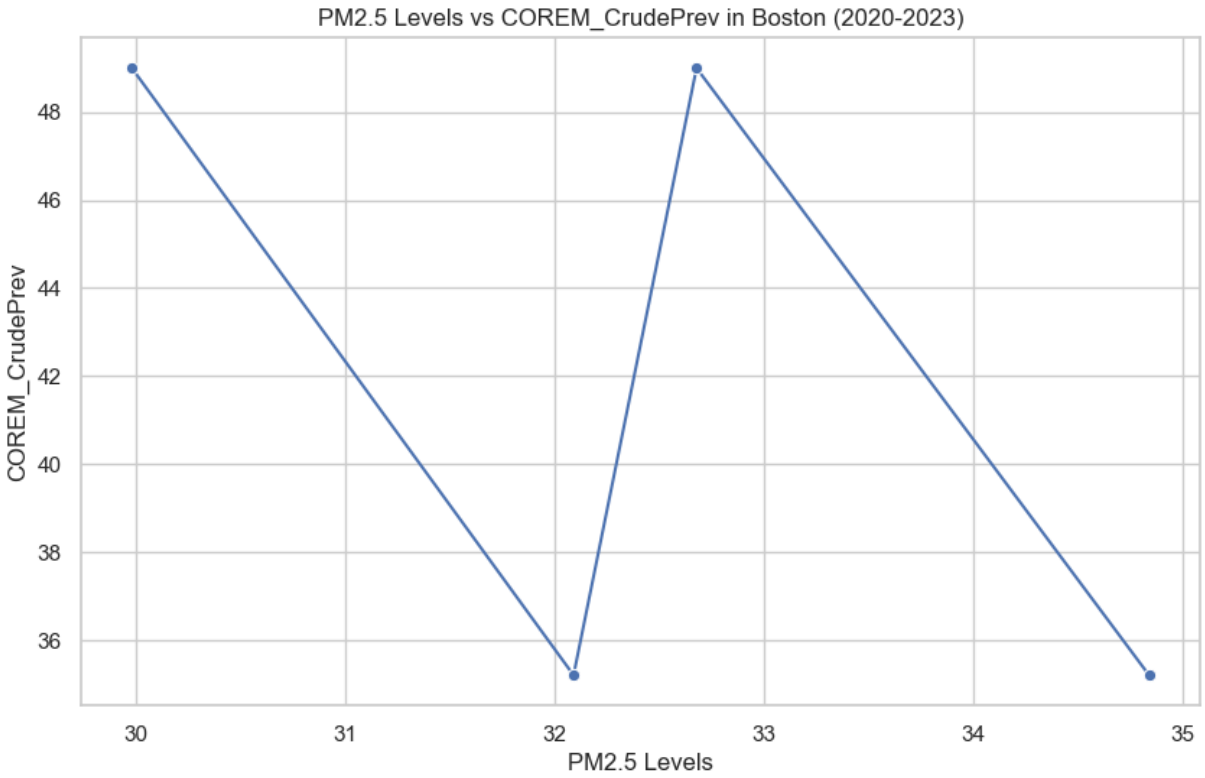

PM2.5 Levels vs CHD_CrudePrev in Boston (2020-2023)
CHD_CrudePrev
4.5
4.4
4.3
4.2
4.1
4.0
30
31
32
33
34
35
PM2.5 Levels

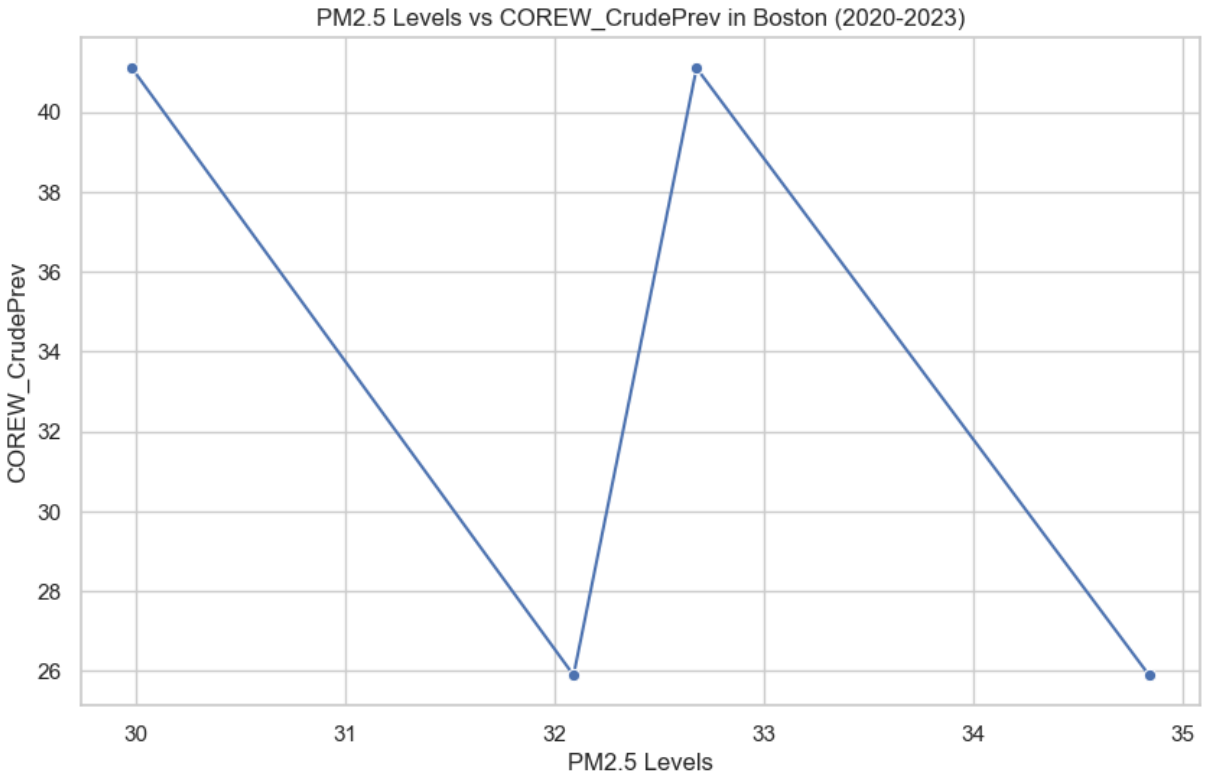

PM2.5 Levels vs CHECKUP_CrudePrev in Boston (2020-2023)
CHECKUP_CrudePrev
77
76
75
74
73
72
71
30
31
32
33
34
35
PM2.5 Levels

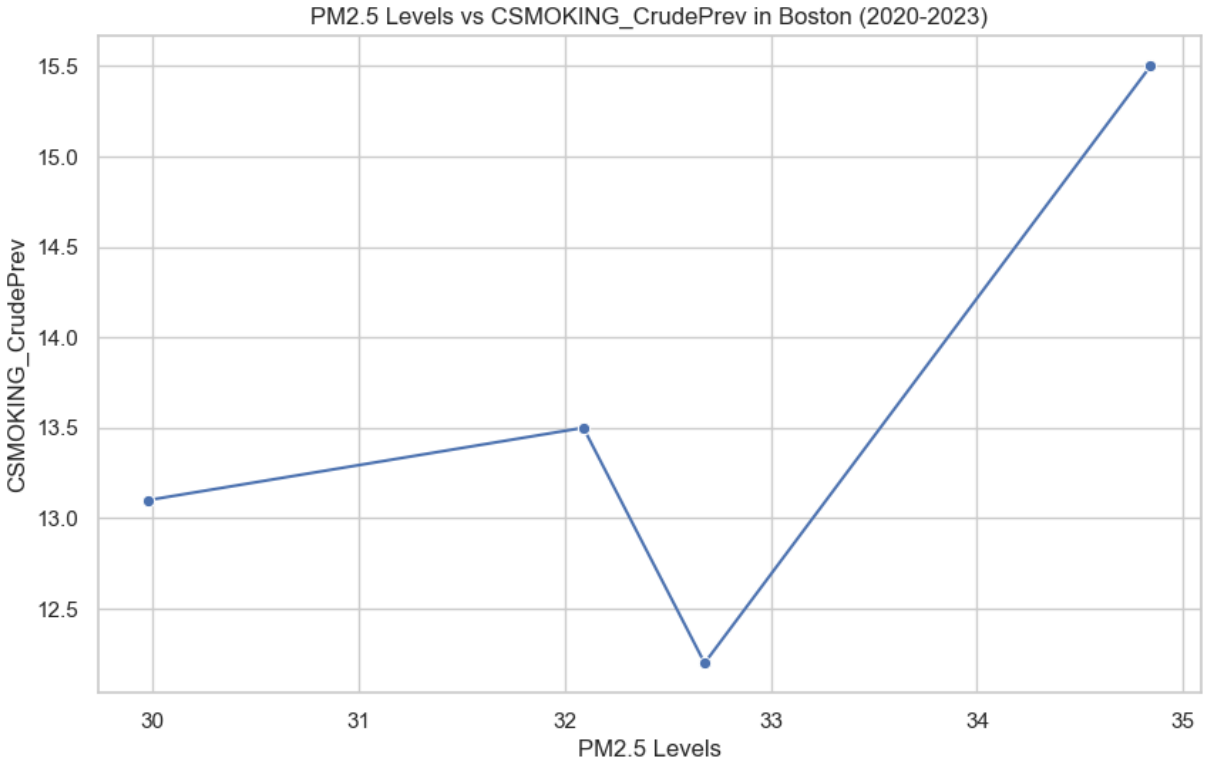

PM2.5 Levels vs CHOLSCREEN_CrudePrev in Boston (2020-2023)
CHOLSCREEN_CrudePrev
85.0
84.5
84.0
83.5
83.0
82.5
82.0
30
31
32
33
34
35
PM2.5 Levels

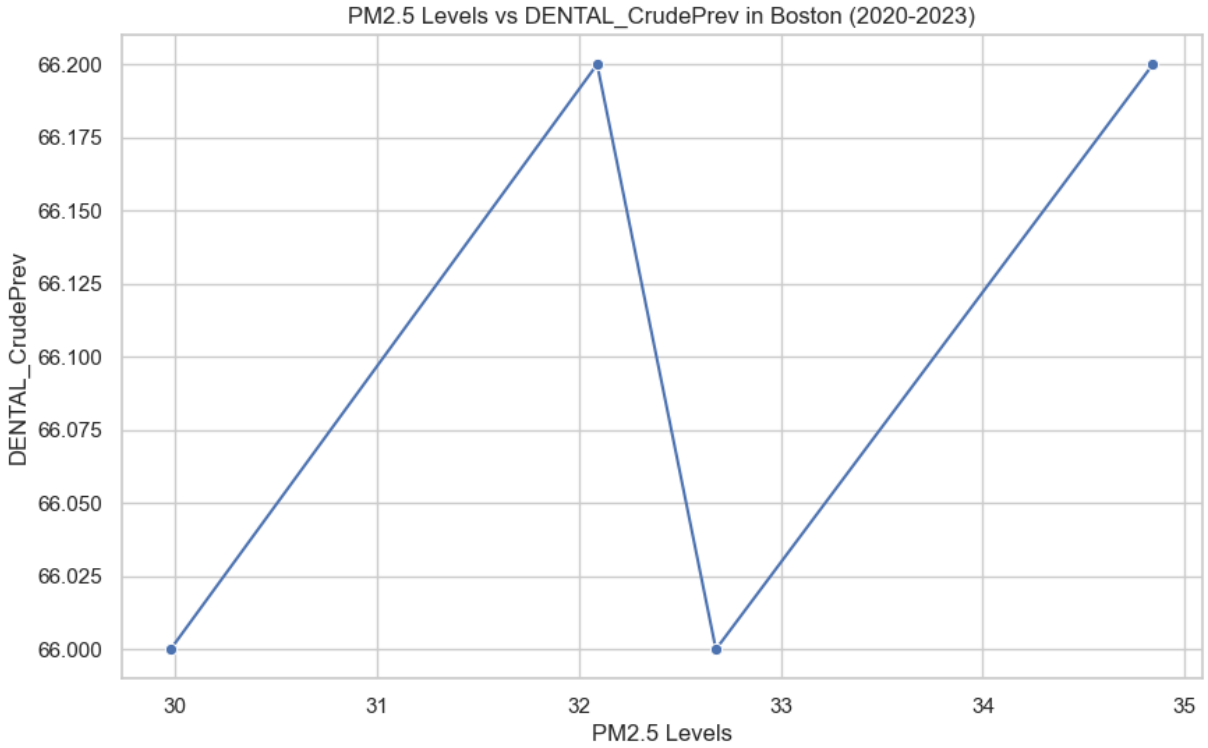

PM2.5 Levels vs COLON_SCREEN_CrudePrev in Boston (2020-2023)
COLON_SCREEN_CrudePrev
78
76
74
72
70
68
30
31
32
33
34
35
PM2.5 Levels

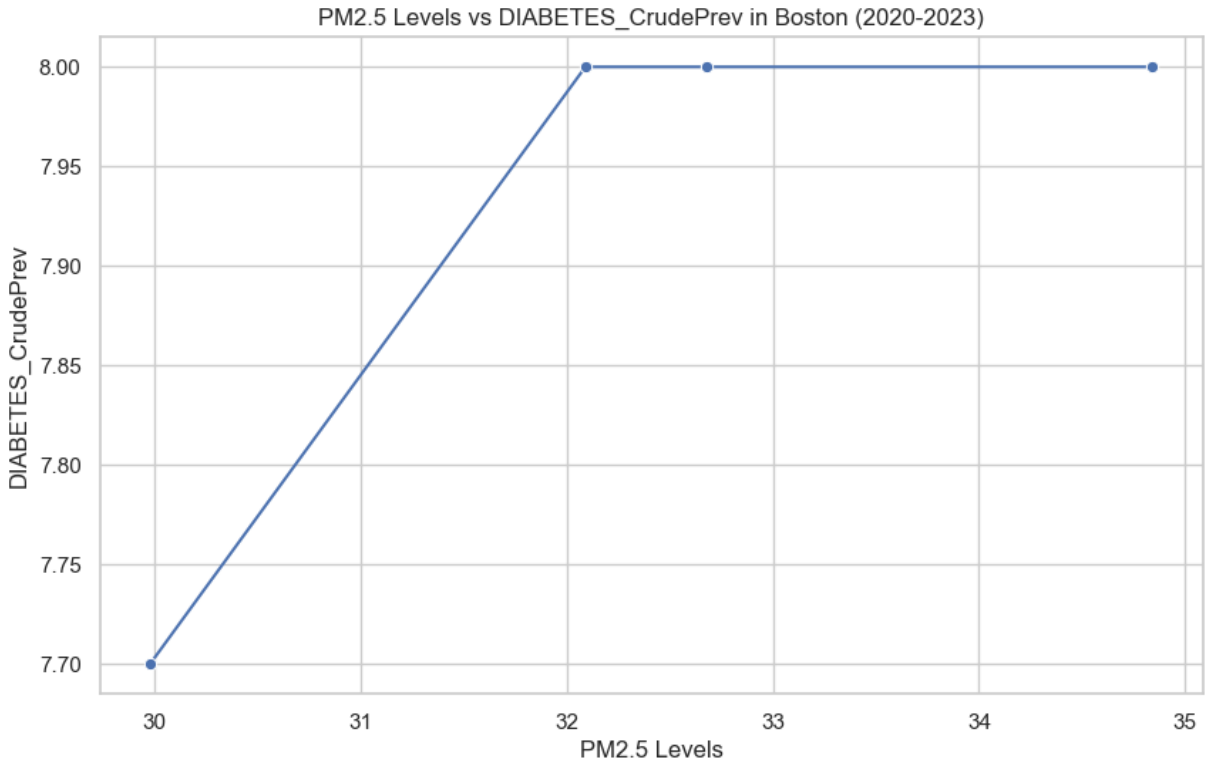

PM2.5 Levels vs COPD_CrudePrev in Boston (2020-2023)
COPD_CrudePrev
4.9
4.8
4.7
4.6
4.5
4.4
4.3
30
31
32
33
34
35
PM2.5 Levels

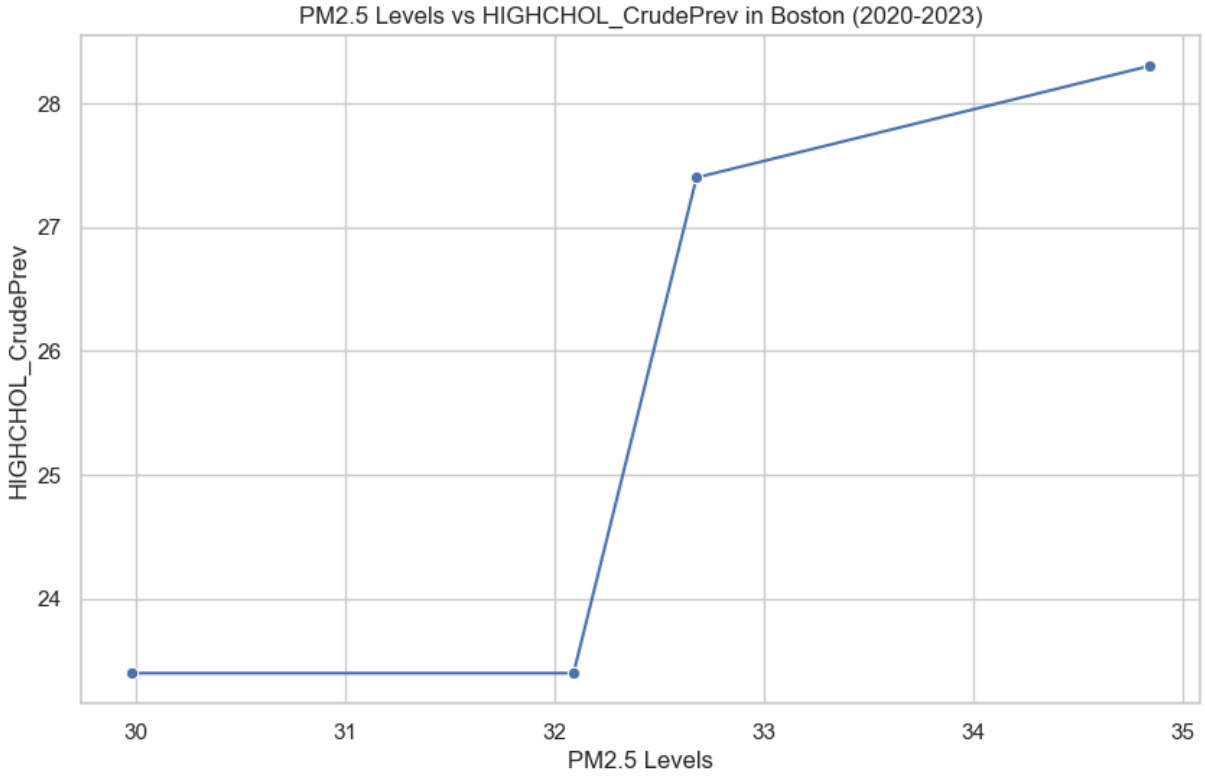

PM2.5 Levels vs COREM_CrudePrev in Boston (2020-2023)
COREM_CrudePrev
48
46
44
42
40
38
36
30
31
32
33
34
35
PM2.5 Levels

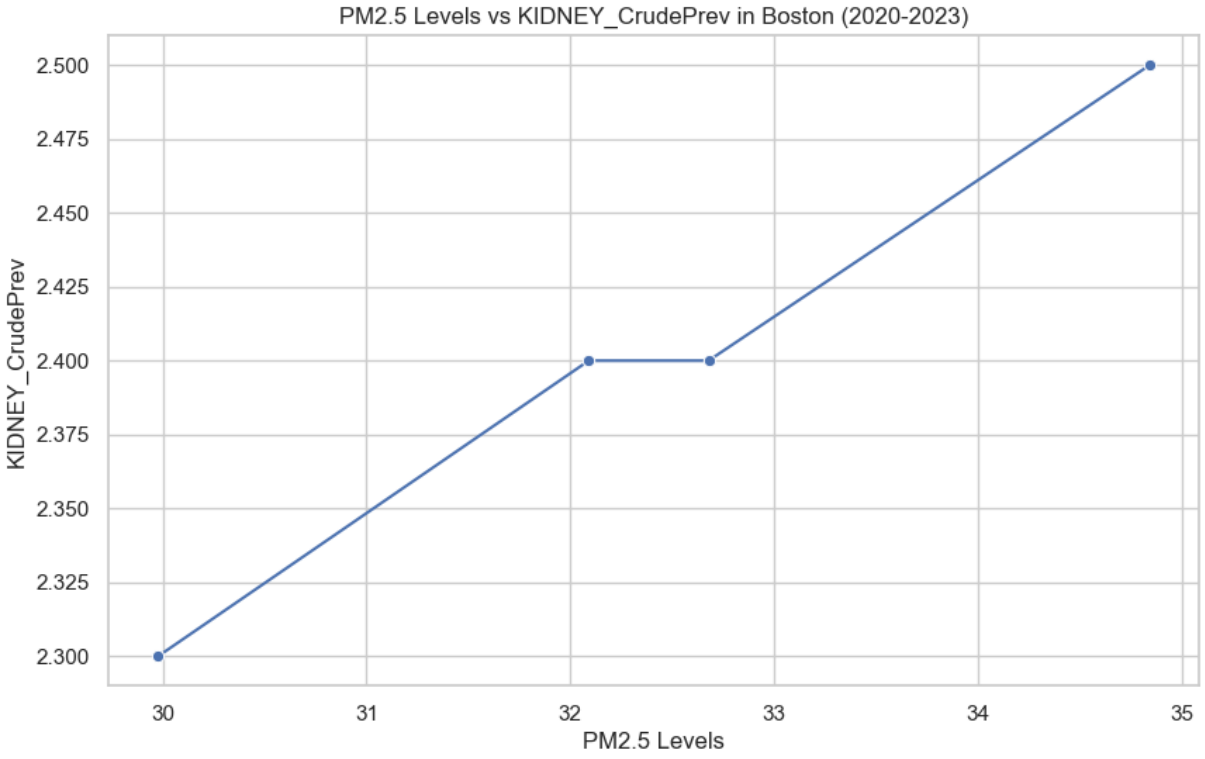

PM2.5 Levels vs COREW_CrudePrev in Boston (2020-2023)
COREW_CrudePrev
40
38
36
34
32
30
28
26
30
31
32
33
34
35
PM2.5 Levels

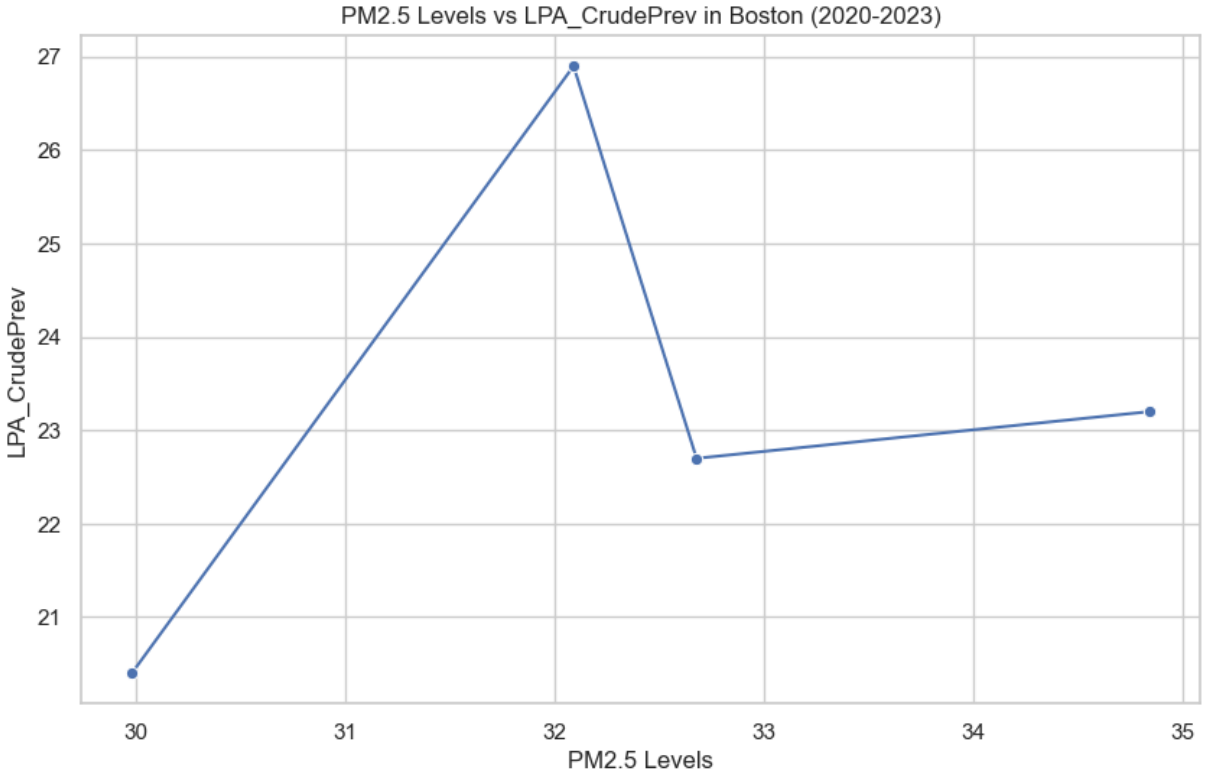

PM2.5 Levels vs CSMOKING_CrudePrev in Boston (2020-2023)
CSMOKING_CrudePrev
15.5
15.0
14.5
14.0
13.5
13.0
12.5
30
31
32
33
34
35
PM2.5 Levels

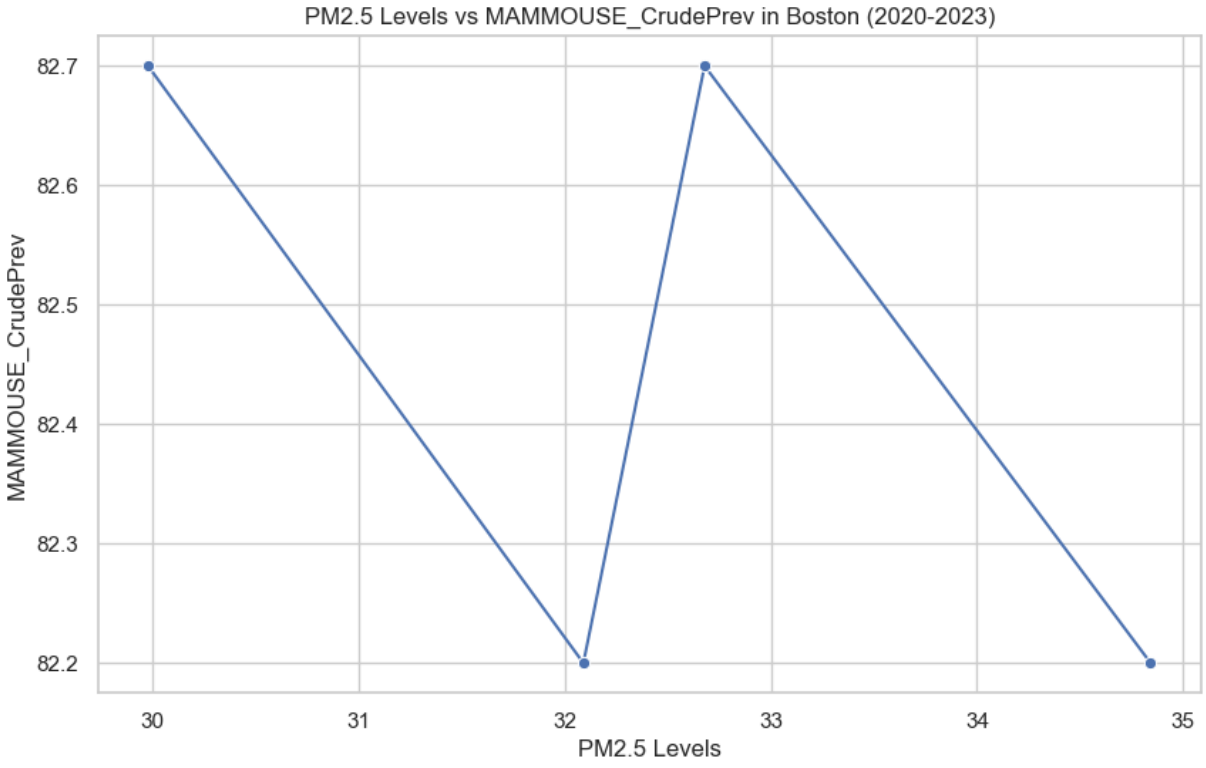

PM2.5 Levels vs DENTAL_CrudePrev in Boston (2020-2023)
DENTAL_CrudePrev
66.200
66.175
66.150
66.125
66.100
66.075
66.050
66.025
66.000
30
31
32
33
34
35
PM2.5 Levels

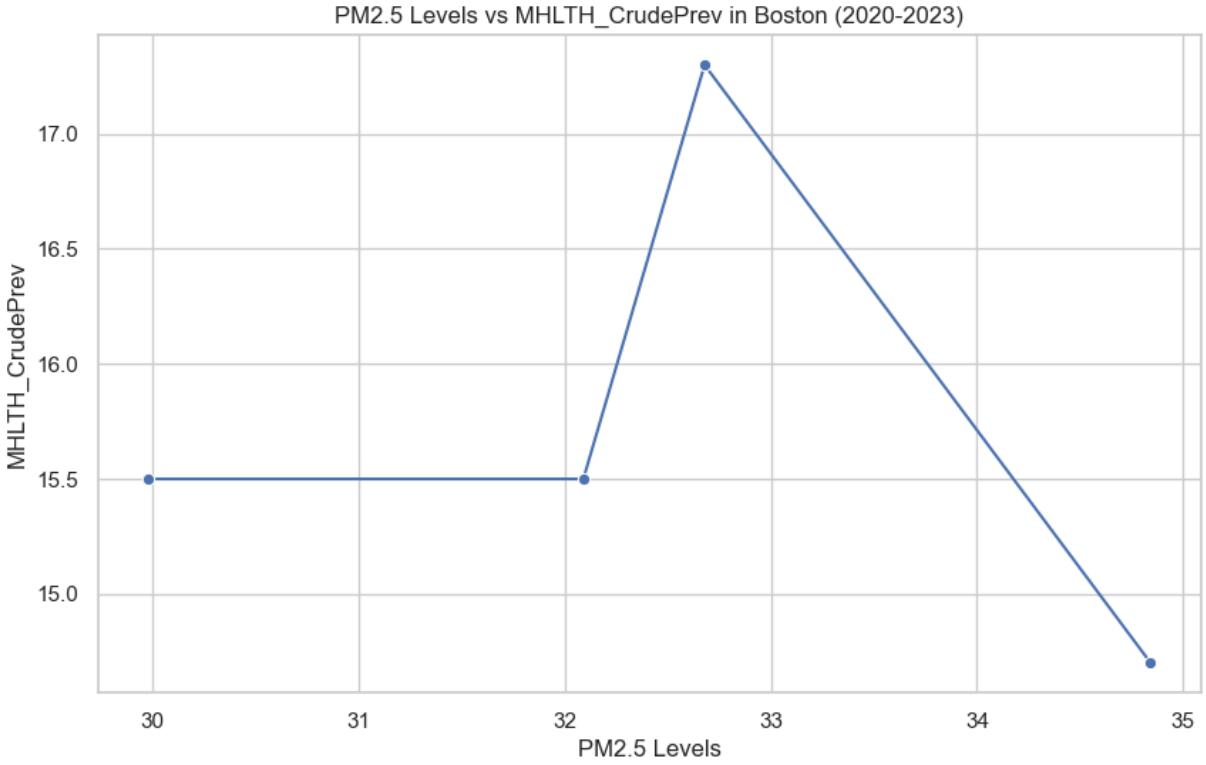

PM2.5 Levels vs DIABETES_CrudePrev in Boston (2020-2023)
DIABETES_CrudePrev
8.00
7.95
7.90
7.85
7.80
7.75
7.70
30
31
32
33
34
35
PM2.5 Levels

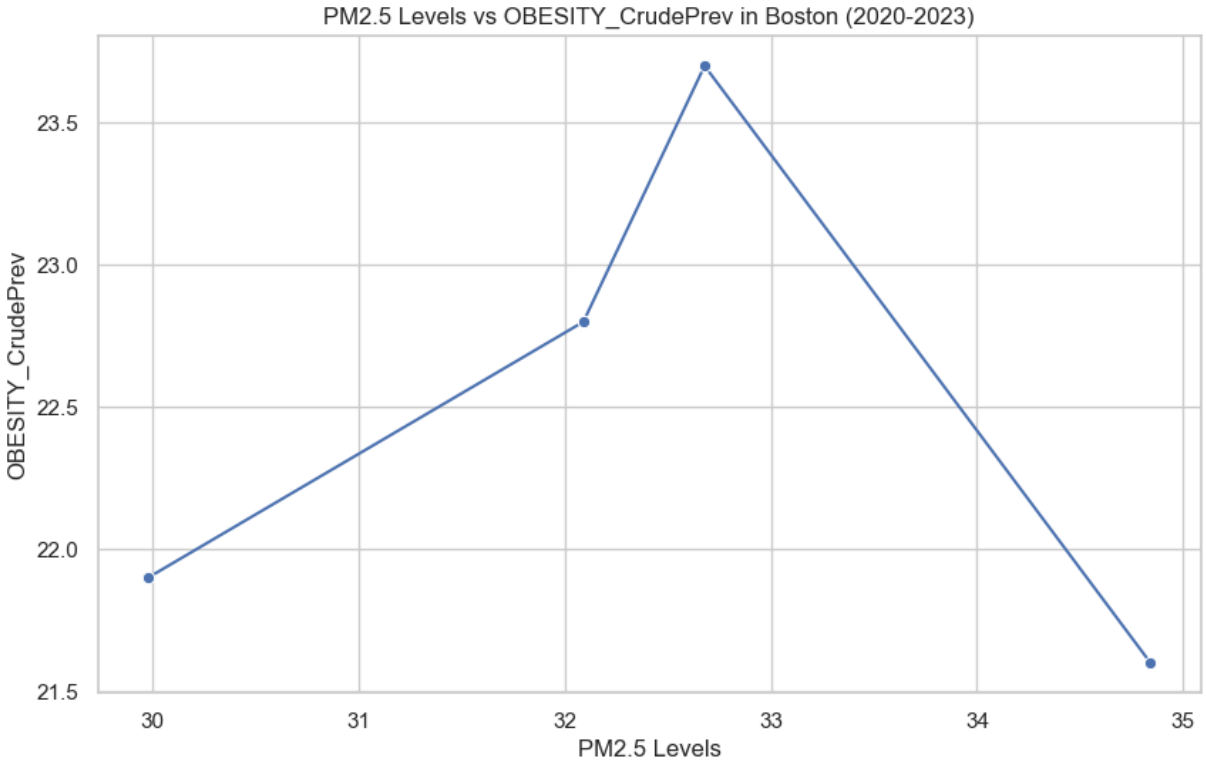

PM2.5 Levels vs HIGHCHOL_CrudePrev in Boston (2020-2023)
HIGHCHOL_CrudePrev
28
27
26
25
24
30
31
32
33
34
35
PM2.5 Levels

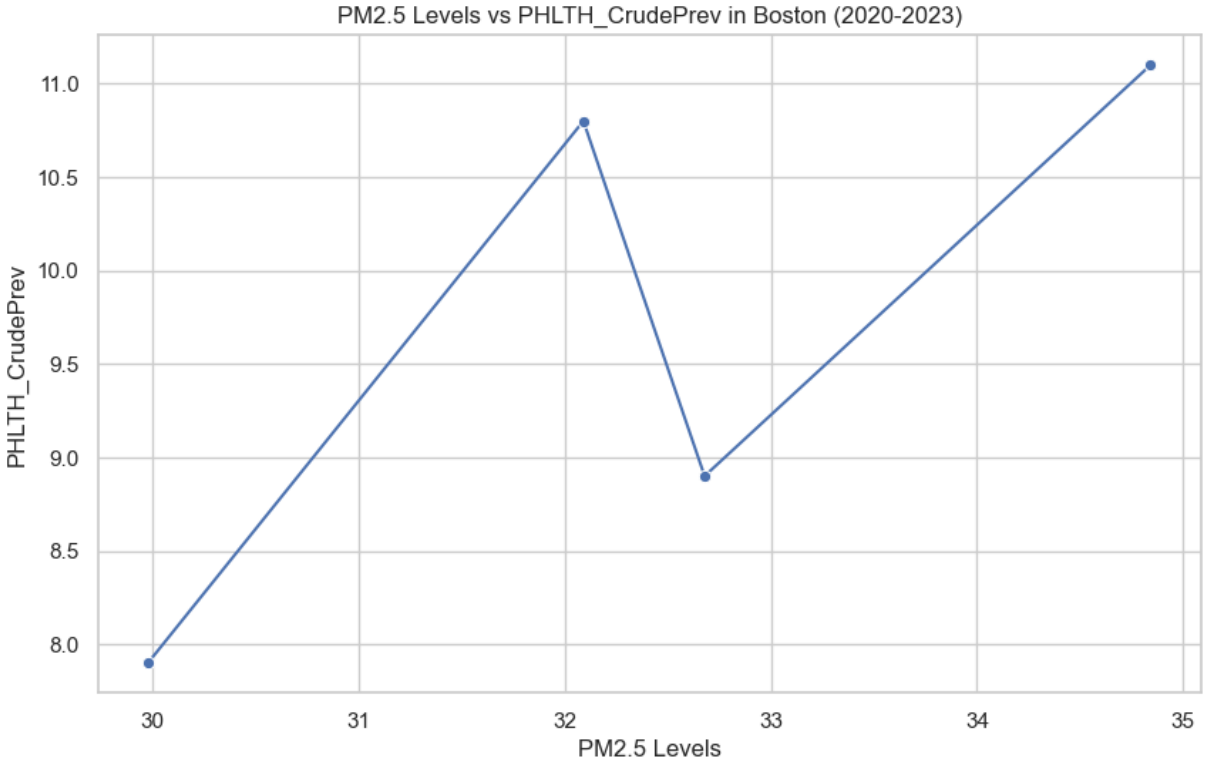

PM2.5 Levels vs KIDNEY_CrudePrev in Boston (2020-2023)
KIDNEY_CrudePrev
2.500
2.475
2.450
2.425
2.400
2.375
2.350
2.325
2.300
30
31
32
33
34
35
PM2.5 Levels

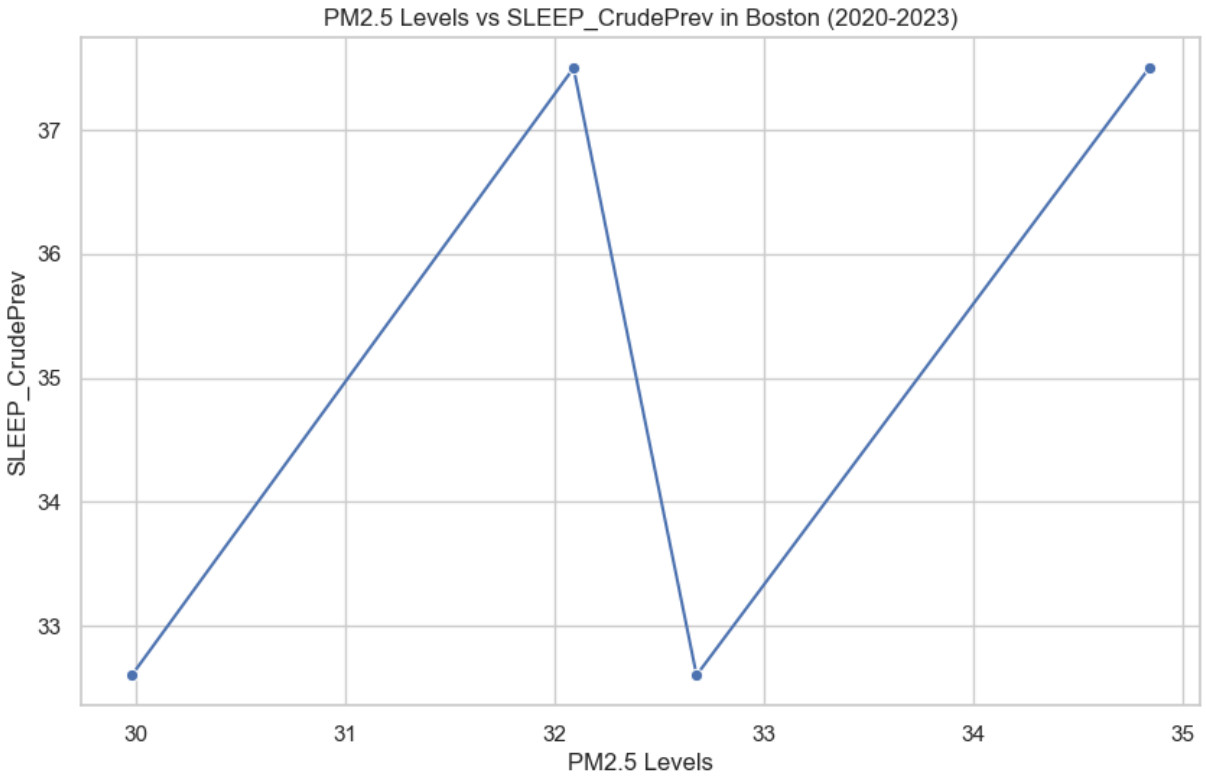

PM2.5 Levels vs LPA_CrudePrev in Boston (2020-2023)
LPA_CrudePrev
27
26
25
24
23
22
21
30
31
32
33
34
35
PM2.5 Levels

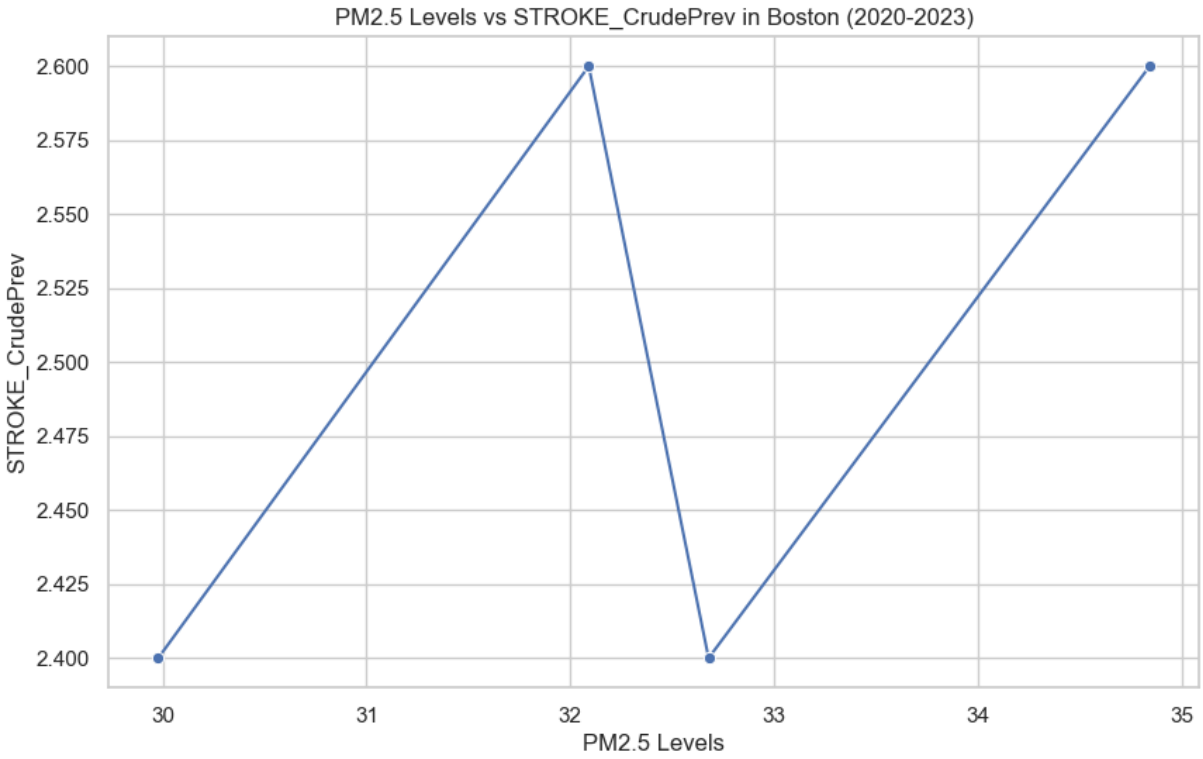

PM2.5 Levels vs MAMMOUSE_CrudePrev in Boston (2020-2023)
MAMMOUSE_CrudePrev
82.7
82.6
82.5
82.4
82.3
82.2
30
31
32
33
34
35
PM2.5 Levels

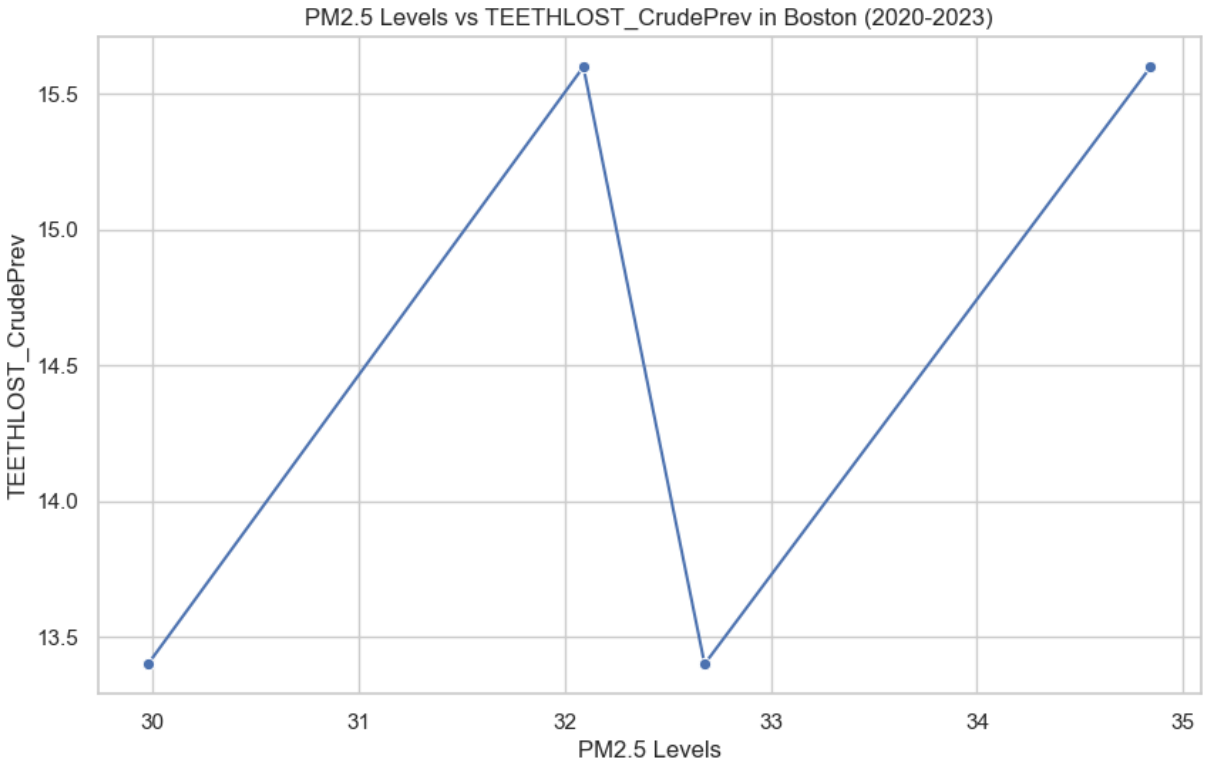

PM2.5 Levels vs MHLTH_CrudePrev in Boston (2020-2023)
MHLTH_CrudePrev
17.0
16.5
16.0
15.5
15.0
30
31
32
33
34
35
PM2.5 Levels


PM2.5 Levels vs OBESITY_CrudePrev in Boston (2020-2023)
OBESITY_CrudePrev
23.5
23.0
22.5
22.0
21.5
30
31
32
33
34
35
PM2.5 Levels


PM2.5 Levels vs PHLTH_CrudePrev in Boston (2020-2023)
PHLTH_CrudePrev
11.0
10.5
10.0
9.5
9.0
8.5
8.0
30
31
32
33
34
35
PM2.5 Levels


PM2.5 Levels vs SLEEP_CrudePrev in Boston (2020-2023)
SLEEP_CrudePrev
37
36
35
34
33
30
31
32
33
34
35
PM2.5 Levels





In [19]:
```
non_disease_columns = {'StateAbbr', 'StateDesc', 'PlaceName', 'PlaceFIPS', '
disease_columns = set(boston_df.columns) - non_disease_columns

melted_data = pd.melt(boston_df, id_vars=['Year', 'pm25'], value_vars=diseas
                      var_name='Disease', value_name='CrudePrev')

plt.figure(figsize=(20,10))
sns.lineplot(x='Year', y='CrudePrev', hue='Disease', data=melted_data)
plt.title('Yearly Changes in Crude Rates for Boston (2020-2023)')
```

```
plt.xlabel('Year')
plt.ylabel('Crude Prevalence Rate')
plt.legend(title='Disease', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.show()
```