In [1]:

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

In [2]:

```
#Parsing the data from the csv file into a table
dataFrame = pd.DataFrame(pd.read_csv("/Users/gaurwik/Documents/Science_Fair_2023/calenvi

#Printing the table
numeric_only = dataFrame._get_numeric_data()
numeric_only
```

Out[2]:

	Census Tract	Total Population	ZIP	Longitude	Latitude	CES 3.0 Score	CES 3.0 Percentile	Ozone	Ozone Pctl	PM2.5
0	6019001100	3174	93706	-119.781696	36.709695	94.09	100.00	0.065	98.18	15.400000
1	6071001600	6133	91761	-117.618013	34.057780	90.68	99.99	0.062	91.10	13.310000
2	6019000200	3167	93706	-119.805504	36.735491	85.97	99.97	0.062	91.10	15.400000
3	6077000801	6692	95203	-121.314524	37.940517	82.49	99.96	0.046	53.02	12.540000
4	6019001500	2206	93725	-119.717843	36.681600	82.03	99.95	0.065	98.18	15.400000
...	...	...	...	...	...	...	...	...	...	...
8030	6009000504	942	95223	-120.211151	38.405130	NaN	NaN	0.055	77.87	4.645934
8031	6065940100	166	92239	-114.475335	34.000183	NaN	NaN	0.044	40.49	9.945784
8032	6053011502	1710	93923	-121.735102	36.301079	NaN	NaN	0.035	16.94	3.991772
8033	6083980100	11	57	-120.048221	33.948186	NaN	NaN	0.040	25.87	9.536303
8034	6111980000	56	61	-119.503588	33.255655	NaN	NaN	0.042	31.84	NaN

8035 rows × 53 columns

In [3]:

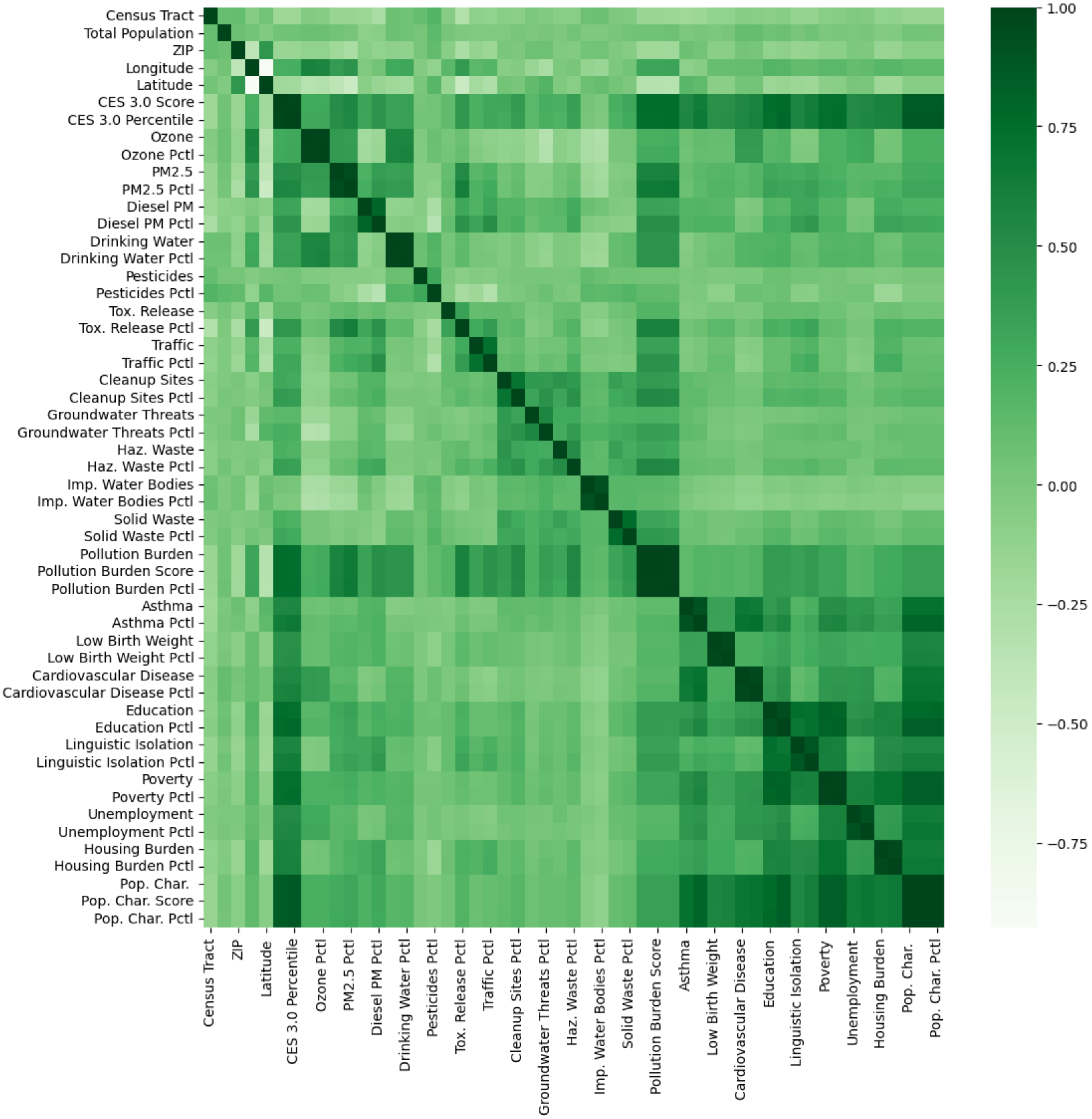
```
#Creating the correlation matrix from the data table
corr = dataFrame.corr()
```

In [4]:

```
#plotting the correlation matrix with with colors
f,ax=plt.subplots(figsize=(12,12))
sns.heatmap(corr, cmap="Greens",annot=False)
```

Out[4]:

```
<AxesSubplot:>
```



In [12]:

```
#Utility method used for removing the pairs with correlated metrics that aren't importan
def isExcluded(row):
    excludedWords = {"Score", "Pctl", "CES 3.0", "Pop. Char.", "Longitude", "Latitude",
    for i in excludedWords:
        if (i in row[0][0] or i in row[0][1]):
            return False

    return row[0][0] != row[0][1]
```

In [13]:

```
#removing duplicates and sorting the pairs in decreasing order by correlation score
c1 = corr.abs().unstack()
c1 = c1.drop_duplicates()
c1 = c1.sort_values(ascending=False)

#Creating a new data table of correlated pairs sorted by score
oldDf = pd.DataFrame(c1)
df = pd.Series
first = True
for row in oldDf.iterrows():
    if (isExcluded(row)):
        x = pd.Series([row[0][0], row[0][1], row[1]])
        if (first):
            df = x
            first = False
        else:
            df = pd.concat([df, x], axis = 1, ignore_index=True)

df = pd.DataFrame(df)
df = df.transpose()
display(df)
```

	0	1	2
0	Education	Poverty	0 0.819775 Name: (Education, Poverty), dtyp...
1	Education	Linguistic Isolation	0 0.736859 Name: (Education, Linguistic Iso...
2	Asthma	Cardiovascular Disease	0 0.663415 Name: (Asthma, Cardiovascular Di...
3	Linguistic Isolation	Poverty	0 0.616986 Name: (Linguistic Isolation, Pov...
4	Poverty	Unemployment	0 0.597475 Name: (Poverty, Unemployment), d...
...	...	...	...
226	Total Population	Asthma	0 0.002285 Name: (Total Population, Asthma)...
227	Traffic	Solid Waste	0 0.001538 Name: (Traffic, Solid Waste), dt...
228	Total Population	Low Birth Weight	0 0.001339 Name: (Total Population, Low Bir...
229	Total Population	Poverty	0 0.001189 Name: (Total Population, Poverty...
230	Pesticides	Haz. Waste	0 0.00038 Name: (Pesticides, Haz. Waste), d...

231 rows × 3 columns

In [14]:

```
#all the different metrics
print(set(df[0]))
```

```
{'Asthma', 'Haz. Waste', 'Traffic', 'Drinking Water', 'Low Birth Weight', 'Diesel PM',
'Groundwater Threats', 'Total Population', 'Poverty', 'PM2.5', 'Ozone', 'Linguistic Isol
ation', 'ZIP', 'Education', 'Cardiovascular Disease', 'Imp. Water Bodies', 'Census Trac
t', 'Cleanup Sites', 'Solid Waste', 'Pesticides', 'Tox. Release'}
```

In [15]:

```
#10 most correlated data
print(df[0:10:1])
```

```
                        0                       1  \
0               Education                 Poverty
1               Education    Linguistic Isolation
2                  Asthma  Cardiovascular Disease
3    Linguistic Isolation                 Poverty
4                 Poverty            Unemployment
5                   Ozone          Drinking Water
6                  Asthma                 Poverty
7           Cleanup Sites              Haz. Waste
8               Education            Unemployment
9  Cardiovascular Disease            Unemployment

                                                   2
0  0    0.819775
Name: (Education, Poverty), dtyp...
1  0    0.736859
Name: (Education, Linguistic Iso...
2  0    0.663415
Name: (Asthma, Cardiovascular Di...
3  0    0.616986
Name: (Linguistic Isolation, Pov...
4  0    0.597475
Name: (Poverty, Unemployment), d...
5  0    0.556655
Name: (Ozone, Drinking Water), d...
6  0    0.487867
Name: (Asthma, Poverty), dtype: ...
7  0    0.452287
Name: (Cleanup Sites, Haz. Waste...
8  0    0.449526
Name: (Education, Unemployment),...
9  0    0.419873
Name: (Cardiovascular Disease, U...
```

In [16]:

```
#10 least correlated data
print(df[-10::1])
```

```
                     0                    1  \
221  Imp. Water Bodies         Unemployment   
222                ZIP         Unemployment   
223         Pesticides               Asthma   
224       Census Tract  Groundwater Threats   
225                ZIP        Cleanup Sites   
226   Total Population               Asthma   
227            Traffic          Solid Waste   
228   Total Population     Low Birth Weight   
229   Total Population              Poverty   
230         Pesticides           Haz. Waste   

                                                     2  
221  0     0.005826
Name: (Imp. Water Bodies, Unempl...  
222  0     0.004765
Name: (ZIP, Unemployment), dtype...  
223  0     0.004315
Name: (Pesticides, Asthma), dtyp...  
224  0     0.003269
Name: (Census Tract, Groundwater...  
225  0     0.002367
Name: (ZIP, Cleanup Sites), dtyp...  
226  0     0.002285
Name: (Total Population, Asthma)...  
227  0     0.001538
Name: (Traffic, Solid Waste), dt...  
228  0     0.001339
Name: (Total Population, Low Bir...  
229  0     0.001189
Name: (Total Population, Poverty...  
230  0     0.00038
Name: (Pesticides, Haz. Waste), d...  
```

In [17]:

```
#Parsing the income metrics for each zipcode into a data table
incomeDf = pd.DataFrame(pd.read_csv("/Users/gaurwik/Documents/Science_Fair_2023/Personal
incomeDf
```

Out[17]:

	Taxable Year	Zip Code	State	City	County	Returns	CA AGI	Total Tax Liability	CountyLatitude	Coun
0	2020	92137	CA	San Diego	San Diego	188	38663083	3084980	32.789640	
1	2020	94557	CA	Hayward	Alameda	107	5104485	159000	37.720226	

2	2020	93005	CA	Ventura	Ventura	227	16117556	899344	34.277091
3	2020	93227	CA	Goshen	Tulare	354	23665658	1764599	36.282543
4	2020	93523	CA	Edwards	Kern	693	30550251	907583	35.376768
...	...	...	...	...	...	...	...	...	...
68583	1998	95009	CA	CampBell	Santa Clara	402	18980130	881246	37.234238
68584	1995	92375	CA	Redlands	San Bernardino	521	32684197	1996312	34.522586
68585	1995	95812	CA	Sacramento	Sacramento	553	18338511	724751	38.192378
68586	1997	91786	CA	Upland	San Bernardino	17749	591798095	17299972	34.522586
68587	1995	94568	CA	Dublin	Alameda	9773	466759300	17409923	37.720226

68588 rows × 13 columns

In [6]:

```
#Combining the zipcode metrics for data with zipcode metrics for pollution and health.
dataFrame = dataFrame.join(incomeDf, how='left', lsuffix='_left', rsuffix='_right')
dataFrame

#Saving the complete data table as a csv file
dataFrame.to_csv("calenviroscreen_results_june_2018_and_Personal_Income_Tax_Statistics_B
dataFrame.to_csv("/Users/gaurwik/Documents/Science_Fair_2023/calenviroscreen_results_jun
```

Out[6]:

	Census Tract	Total Population	California County	ZIP	Nearby City \n(to help approximate location only)	Longitude	Latitude	CES 3.0 Score	CES 3.0 Percentile	\n
0	6019001100	3174	Fresno	93706	Fresno	-119.781696	36.709695	94.09	100.00	
1	6071001600	6133	San Bernardino	91761	Ontario	-117.618013	34.057780	90.68	99.99	
2	6019000200	3167	Fresno	93706	Fresno	-119.805504	36.735491	85.97	99.97	
3	6077000801	6692	San Joaquin	95203	Stockton	-121.314524	37.940517	82.49	99.96	
4	6019001500	2206	Fresno	93725	Fresno	-119.717843	36.681600	82.03	99.95	

...	...	...	...	...	...	...	...	...	...
8030	6009000504	942	Calaveras	95223	Arnold	-120.211151	38.405130	NaN	NaN
8031	6065940100	166	Riverside	92239	Desert Center	-114.475335	34.000183	NaN	NaN
8032	6053011502	1710	Monterey	93923	Carmel	-121.735102	36.301079	NaN	NaN
8033	6083980100	11	Santa Barbara	57	Channel Islands	-120.048221	33.948186	NaN	NaN
8034	6111980000	56	Ventura	61	Channel Is Air Guard Station	-119.503588	33.255655	NaN	NaN

8035 rows × 70 columns