Foundations of Urban Science Assignment 4

November 25, 2014

By Sriniketh Vijayaraghavan - sv1272

1 Overview

The objective of this assignment is to analyze and try to determine, if there is such a thing as the optimal size of a city. There are various factors which determine this. It includes but is not limited to, the population density, the transportation to and from different regions around the area and carbon emissions.

In this assignment, we look only at the carbon emissions produced by a city to try and determine its optimal size. We also try to validate the veracity of this argument through plots and visualizations.

When we think of carbon emissions, we immediately think about the factories and industries which pollute the air and also dump toxic waste into the water supply. We generally do not associate carbon emissions through subtle usage such as the centralized heater in a home or the fuel and emissions from a car. This leads to a skewed way of thinking which allows us to believe easily that cities in general are filled with pollutants and are a very unhealthy place to live.

While this may be true to some extent, if we look closely we can see that big cities such as New York and Boston force people to live in smaller spaces and use public transport. This would, with some logical sense lead to the idea that cities are in fact more green than sprawl.

We can put our theories to test by using the **Vulcan Data** as found in *http://vulcan.project.asu.edu/*. We get the per capita carbon emissions from the ASU vulcan project. This is the focal point of our analysis. We also gather the county information from Social Explorer in order to be able to get the population and area of each county. We can use these two datasets together to form our analysis.

2 Literature Review

Our goal is to estimate the optimal size of the city in terms of carbon emissions. In [1], we can see that the authors argue that carbon emissions sometimes leads to a super linear relation with population density while sometimes it becomes sublinear. In this paper, the authors assume that most urban indicators can be determined in terms of the following ubiquitous scaling law:

$$Y(t) = Y_0(t)N(t)^{\beta}$$

where $Y(t)$ and $N(t)$ are the urban indicator and the population size of the city at time t respectively. In our case, $Y(t)$ is the carbon emissions in the city. $Y_0(t)$ is a normalization constant and β is less than 1 which tries to provide a sublinear relation between the population of the city to the urban indicator.

So, the authors of [1] are optimistic about the sublinear relation which urban indicators tend to have on the population of a city. These authors also talk about London as a dragon-king, or a node which highly influences the nature of the regression line by being a far reaching outlier.

These outliers tend to distort the graphs to skew the results in one direction or the other. But, at the same time they cannot be ignored since they might also give rise to an ideal scenario which has emerged due to a combination of factors. The authors also conclude that no single factor can be used to classify whether the size of the city would lead to lesser emissions. They call cities complex beings which are hard to model.

In [2] however, the name of the paper itself tries to disregard the sublinear model proposed in [1]. By arguing that large cities are less green they are promoting the suburban sprawl lifestyle. These authors use the City Clustering Algorithm (CCA) to determine the size of the city and then use the above equation to compute the value of β. They find out that the value of β is greater than 1 and so forms a super linear relationship with the population of the city. By doing this, they argue that large cities produce much larger emissions than the corresponding smaller cities.

The authors also argue that the analysis using the MSA gave a much different result due to overestimation of the MSA areas. So, they argue naively that the MSAs are overestimated in area and the total carbon emissions in larger cities are much higher than the corresponding smaller cities.

In the next section, I will discuss my methodology of using the collections of MSAs and their carbon emissions to visually plot the regression line and determine the accuracy of the plot. I believe that the carbon emissions produced per person in a larger city should be much smaller than its corresponding smaller city. This is because, in a larger city, there is a disproportionate difference in the number of people living to the area of the MSA. This can be normalized and found out by selecting the population density of an MSA instead of just the population.

One more change is that we first take the total carbon emissions and plot it against the population density. Then, we plot the population density against the carbon emission per person and try to notice the difference between the plots. We can determine the accuracy of this model by determining the value of the R^2 value. We can also determine the optimal size of the MSA by finding the global minima of the carbon emissions and determine which city is closest to it.

3 Data Cleaning

We start the data cleaning process by first putting the data into excel. First, we select the county population data and store it in **.xls** format. Then we proceed to get the vulcan data in the same way. We clean out the empty rows and select only those attributes which we need.

Then, we have 2 excel files. One containing the county populations and its land area and the other one containing the CO2 emissions. We also have to aggrigate the counties by MSAs so that we can come closer to analyzing the carbon emission of an entire city rather than just single counties.

We then assign each county its MSA by mapping it to another file which contains the MSA list and the counties within it. Once this is done, we get a combined csv file which contains the carbon emissions, the land area and the population alongside the MSA name that each county belongs to.

We then save it as a csv file and then move into python for the data analysis.

We use the pandas dataframe to store the clean data and start by importing the initial necessary libraries as shown below.

```
In [14]: import pandas as pd
         import matplotlib.pyplot as plt
         plt.style.use('bmh')
         print plt.style.available
         %matplotlib inline
```

```
[u'grayscale', u'bmh', u'dark_background', u'ggplot', u'fivethirtyeight']
```

The above code shows us what we need to import in order to run this simulation. The entire project is being demonstrated through the use of the IPython notebook. We select the plotting style to be 'bmh' amongst the other that are available such as:

- grayscale
- dark_background
- ggplot
- fivethirtyeight

These styles are shown above through the use of print plt.style.available
The libraries that are used are:

1. Pandas for putting the entire cleaned data into a dataframe for manipulation and plotting.
2. Matplotlib.pyplot for showing the plot and the legend.

Now that we have the files, we first import the clean county data with populations & land and print out the head of the data.

```
In [90]: df = pd.read_csv('countypop.csv')
         df.head()
```

Out[90]:

	STATE	COUNTY NAME	POP 2000	FIPS	MSA/PMSA	\
0	TX	Taylor County	126555	48441	40	
1	GA	Dougherty County	96065	13095	120	
2	GA	Lee County	24757	13177	120	
3	NY	Albany County	294565	36001	160	
4	NY	Montgomery County	49708	36057	160	

	MSA	TOTAL CO2	CO2 PER CAP
0	Abilene, TX MSA	0.382555	3.022838
1	Albany, GA MSA	0.806702	8.397459
2	Albany, GA MSA	0.042486	1.716133
3	Albany--Schenectady--Troy, NY MSA	1.248946	4.239967
4	Albany--Schenectady--Troy, NY MSA	0.185540	3.732602

The above is the cleaned data. But, we go one step further in excel and we aggregate all the counties under an MSA and add up the total CO2 emissions as well as the population & total area. Once this is done, we again read the final data (named A4Data.csv) to perform the analysis.

```
In [91]: df = pd.read_csv('A4Data.csv')
         df.head()
```

Out[91]:

	MSA NAME	TOTAL POP IN MSA	TOTAL CO2 IN MSA	TOTAL LAND IN MSA	POP DENSITY	CO2/CAPITA
0	Abilene, TX MSA	126555	0.382555	915.6259	138.216929	0.000003
1	Albany, GA MSA	120822	0.849188	685.3690	176.287518	0.000007
2	Albany--Schenectady--Troy, NY MSA	875583	3.002046	3222.1940	271.735035	0.000003
3	Albuquerque, NM MSA	712738	2.141536	5943.0000	119.928992	0.000003
4	Alexandria, LA MSA	126337	1.561100	1322.5410	95.525961	0.000012

After this step, we add another column in the data frame called **'CO2 density'** which is the CO2/CAPITA column which is multiplied by a factor of 10^6 for analysis.

One more thing we needed to do to clean the data is to remove the MSAs with less than 100,000 as the population. This way we do not skew the results by a large amount. This is done in the step below.

```
In [92]: df = df[(df['TOTAL POP IN MSA'] > 100000)]
```

```
In [93]: df['CO2 density'] = df['TOTAL CO2 IN MSA']/df['TOTAL POP IN MSA']*1000000
         df1 = df[['POP DENSITY', 'TOTAL CO2 IN MSA', 'CO2/CAPITA', 'CO2 density', 'TOTAL POP IN MSA']]
         print df1.head()
```

```
POP DENSITY  TOTAL CO2 IN MSA  CO2/CAPITA  CO2 density  TOTAL POP IN MSA
0   138.216929          0.382555    0.000003     3.022838            126555
1   176.287518          0.849188    0.000007     7.028424            120822
2   271.735035          3.002046    0.000003     3.428626            875583
3   119.928992          2.141536    0.000003     3.004661            712738
4    95.525961          1.561100    0.000012    12.356635            126337
```

4 Data Analysis

In this section, we try to perform analysis on the data which we have cleaned & parsed to obtain the exact information we need. We start with the numpy library which helps us work with polynomials to perform the regression analysis. In this analysis, we perform a linear, quadratic and cubic regression in order to estimate the best fit for the graph. We start off by determining the equation for Population Density with the Total CO2 produced in an MSA. The linear fit model has an equation as follows:

$$y = mx + c$$

```
In [94]: import numpy as np

         z = np. polyfit(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'], 1)
         f = np.poly1d(z)

         x_new = np.linspace(min(np.log(df1['POP DENSITY'])), max(np.log(df1['POP DENSITY'])), 100)
         y_new = f(x_new)
```

The coefficients of the equation m and c are:

```
In [95]: f.c
```

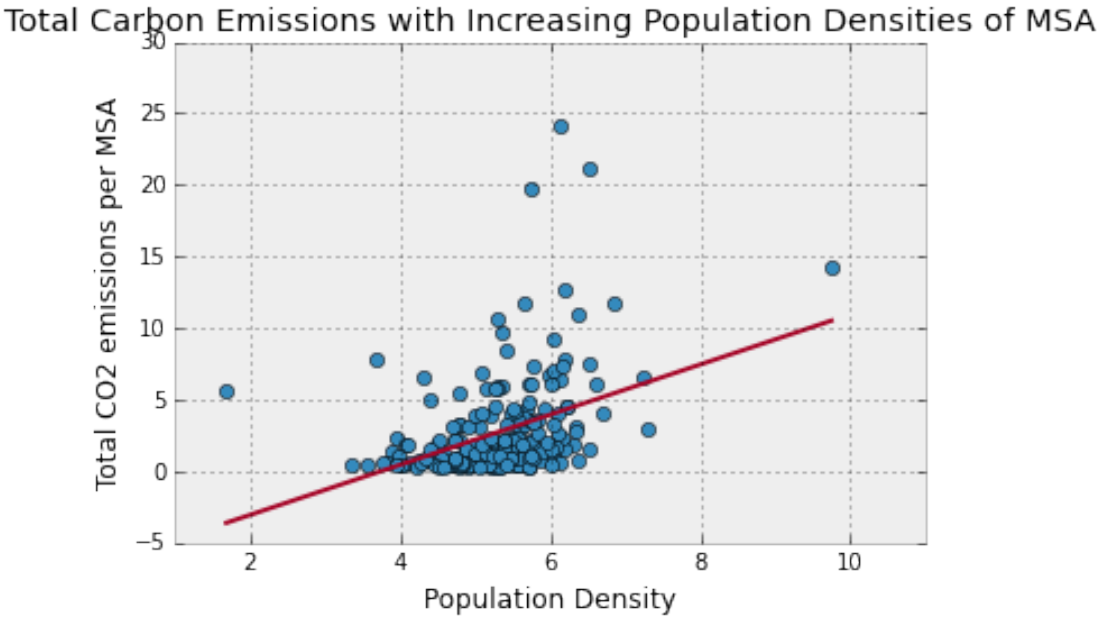
```
Out[95]: array([ 1.75258282, -6.55531642])
```

we get

$$\text{m} = 1.75258282$$
$$\text{c} = -6.55531642$$

From this plot we get a regression line like this:

```
In [96]: plt.scatter(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'])

         plt.plot(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'], 'o', x_new,y_new)
         plt.title('Total Carbon Emissions with Increasing Population Densities of MSA')
         plt.xlabel('Population Density')
         plt.ylabel('Total CO2 emissions per MSA')
         plt.show()
```

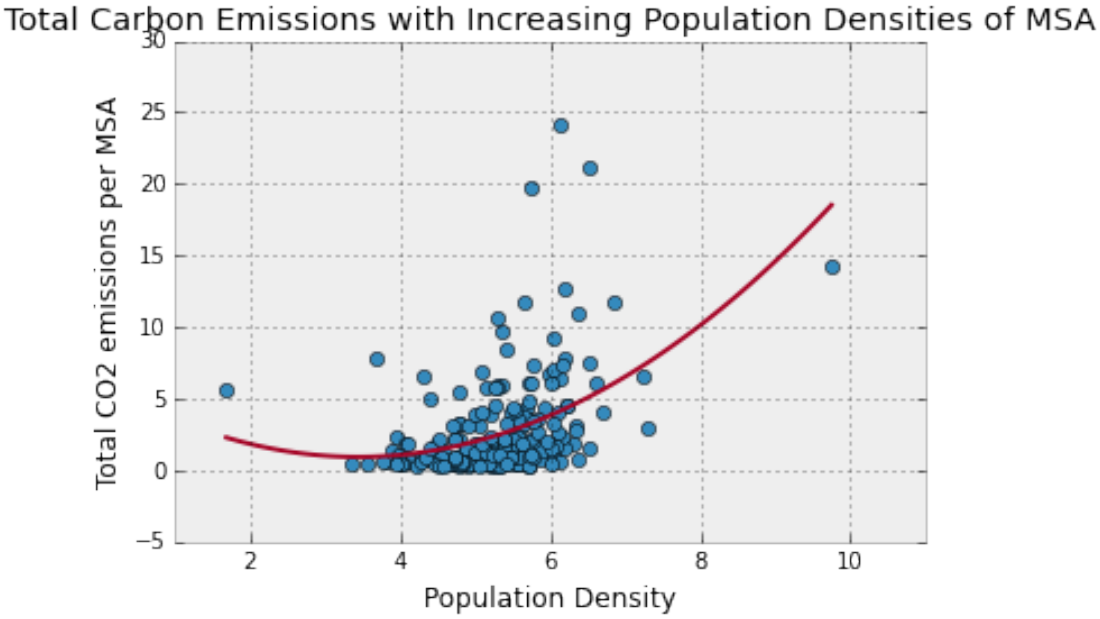


Now, we try to perform the same analysis while taking the regression lines to be quadratic and cubic having equations like

$$y = ax^2 + bx + c$$
$$y = ax^3 + bx^2 + cx + d$$

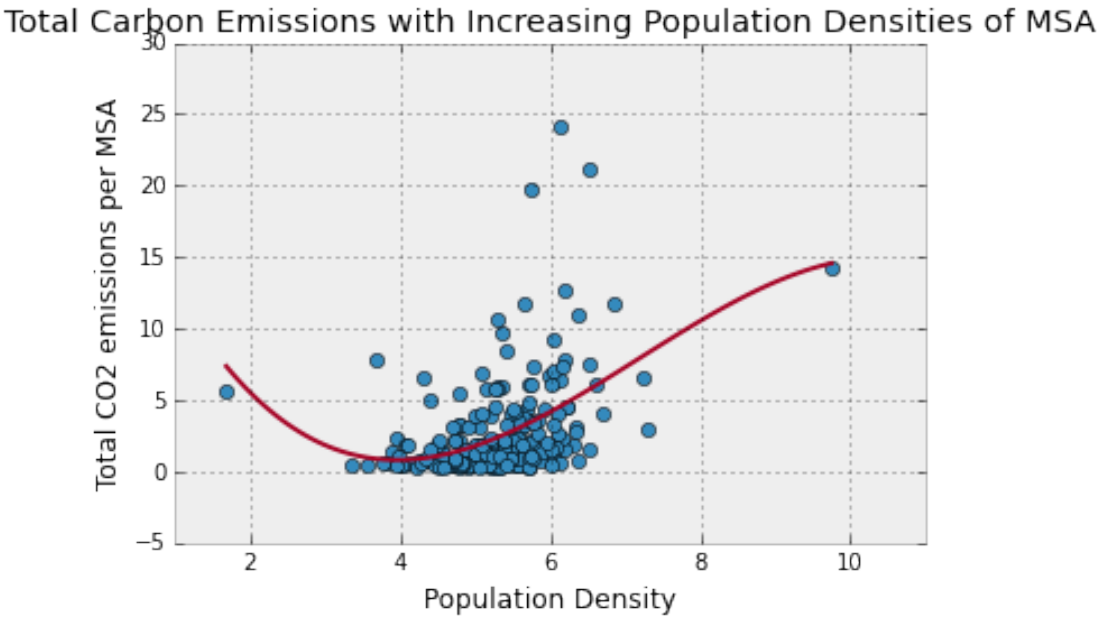
```
In [97]: z = np. polyfit(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'], 2)
         f = np.poly1d(z)

         x_new = np.linspace(min(np.log(df1['POP DENSITY'])), max(np.log(df1['POP DENSITY'])), 100)
         y_new = f(x_new)
         f.c
```

```
Out[97]: array([ 0.442323  , -3.04266411,  6.17124503])

In [98]: plt.scatter(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'])

         plt.plot(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'], 'o', x_new,y_new)
         plt.title('Total Carbon Emissions with Increasing Population Densities of MSA')
         plt.xlabel('Population Density')
         plt.ylabel('Total CO2 emissions per MSA')
         plt.show()
```



The coefficients for the quadratic regression are shown above and the ones for the cubic relations are calculated below.

```
In [99]: z = np. polyfit(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'], 3)
         f = np.poly1d(z)

         x_new = np.linspace(min(np.log(df1['POP DENSITY'])), max(np.log(df1['POP DENSITY'])), 100)
         y_new = f(x_new)
         f.c

Out[99]: array([ -0.10633172,   2.2902314 , -13.14103277,  23.54288621])

In [100]: plt.scatter(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'])

          plt.plot(np.log(df1['POP DENSITY']), df1['TOTAL CO2 IN MSA'], 'o', x_new,y_new)
          plt.title('Total Carbon Emissions with Increasing Population Densities of MSA')
          plt.xlabel('Population Density')
          plt.ylabel('Total CO2 emissions per MSA')
          plt.show()
```



As we can see, the cubic regression line seems to be an overfit while the quadratic regression line seems to curve up around the middle of the graph. These 3 plots in general showcase that the total carbon emissions do in fact increase with the increase in the population density of an MSA.

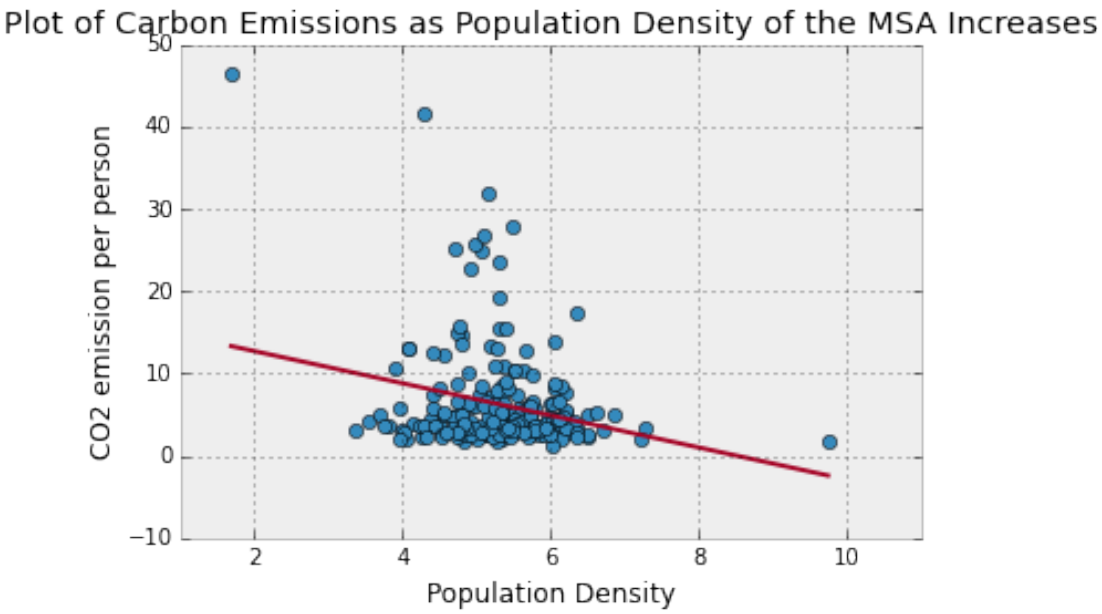
There is one more case which is interesting to check. If we try to plot the CO2 emission per person in an MSA with respect to increasing population density, we get some more interesting results. These can be seen as follows:

```
In [101]: z = np. polyfit(np.log(df1['POP DENSITY']), df1['CO2 density'], 1)
          f = np.poly1d(z)

          x_new = np.linspace(min(np.log(df1['POP DENSITY'])), max(np.log(df1['POP DENSITY'])), 100)
          y_new = f(x_new)
          f.c

Out[101]: array([ -1.94991736,  16.61684719])
```

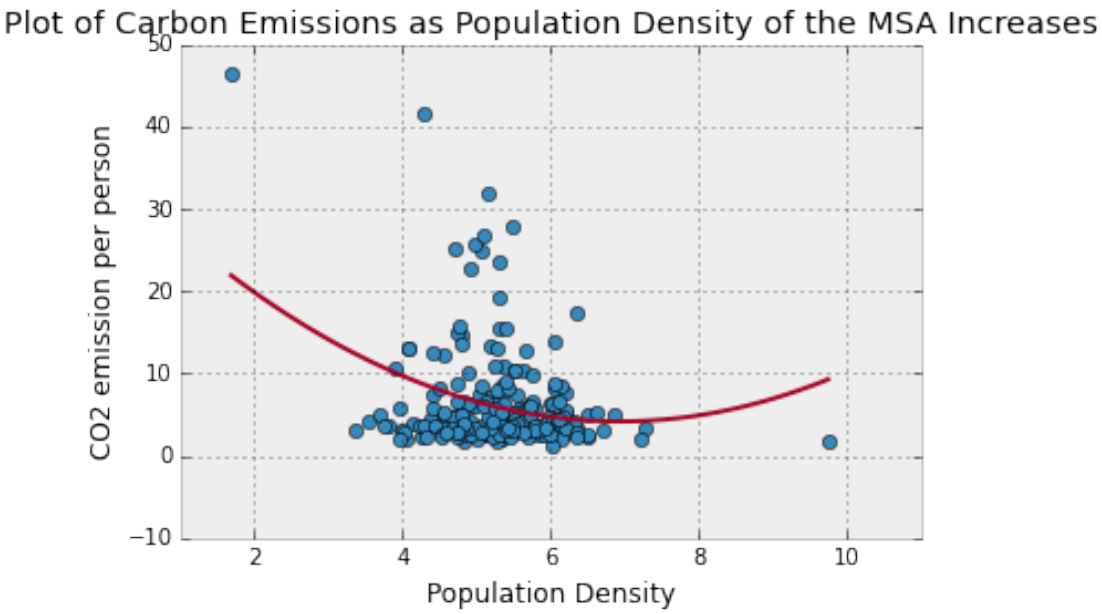
As before, we get the slope and intercept of the linear regression plot.

$$m = -1.94991736$$
$$c = 16.61684719$$

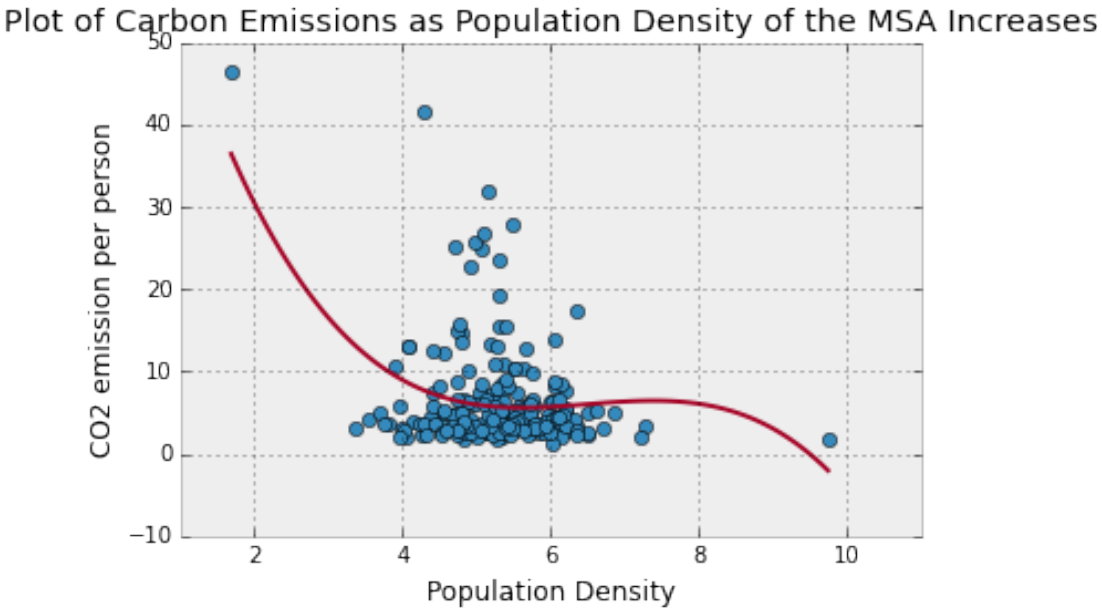
```
In [102]: plt.scatter(np.log(df1['POP DENSITY']), df1['CO2 density'])

          plt.plot(np.log(df1['POP DENSITY']), df1['CO2 density'], 'o', x_new,y_new)
          plt.title('Plot of Carbon Emissions as Population Density of the MSA Increases')
          plt.xlabel('Population Density')
          plt.ylabel('CO2 emission per person')
          plt.show()
```




```
In [107]: z = np. polyfit(np.log(df1['POP DENSITY']), df1['CO2 density'], 2)
          f = np.poly1d(z)

          x_new = np.linspace(min(np.log(df1['POP DENSITY'])), max(np.log(df1['POP DENSITY'])), 100)
          y_new = f(x_new)
          f.c

Out[107]: array([  0.64479142,  -8.94013506,  35.16885015])

In [104]: plt.scatter(np.log(df1['POP DENSITY']), df1['CO2 density'])

          plt.plot(np.log(df1['POP DENSITY']), df1['CO2 density'], 'o', x_new,y_new)
          plt.title('Plot of Carbon Emissions as Population Density of the MSA Increases')
          plt.xlabel('Population Density')
          plt.ylabel('CO2 emission per person')
          plt.show()
```




```
In [105]: z = np. polyfit(np.log(df1['POP DENSITY']), df1['CO2 density'], 3)
          f = np.poly1d(z)

          x_new = np.linspace(min(np.log(df1['POP DENSITY'])), max(np.log(df1['POP DENSITY'])), 100)
          y_new = f(x_new)
          f.c

Out[105]: array([ -0.3038396 ,   5.92513253, -37.79591122,  84.8077776 ])

In [106]: plt.scatter(np.log(df1['POP DENSITY']), df1['CO2 density'])

          plt.plot(np.log(df1['POP DENSITY']), df1['CO2 density'], 'o', x_new,y_new)
          plt.title('Plot of Carbon Emissions as Population Density of the MSA Increases')
          plt.xlabel('Population Density')
          plt.ylabel('CO2 emission per person')
          plt.show()
```



Now, from these plots, we can say that the quadratic regression line was the most informative. This is because it has one point of inflection, i.e. it has one global/local maxima or minima. So, we notice that for the carbon emissions per person in MSA, we get a global minima at about the 5-7 mark on the log population density plot. By looking at the MSAs that are close to that figure, we can try to estimate the most ideal size of a city.

```
In [85]: low, high = 6.5, 7
         df2 = df[(np.log(df['POP DENSITY']) > low) & (np.log(df['POP DENSITY'])< high)]
         df2['MSA NAME']
```

```
Out[85]: 13                                Atlanta, GA MSA
         32                Buffalo--Niagara Falls, NY MSA
         65                                El Paso, TX MSA
         194                Salt Lake City--Ogden, UT MSA
         197                              San Diego, CA MSA
         223    Tampa--St. Petersburg--Clearwater, FL MSA
         Name: MSA NAME, dtype: object
```

We can see from the above that based on the quadratic curve, the above 6 MSAs should be of the ideal size and population density for the lowest carbon emission. We can also see that the data is being skewed by the fact that there are large outliers like New York City which show unbelievably good results for carbon emissions but the rest of the cities are not like that. So, we also calculate the R^2 value for the fit to understand where we stand. The higher the value of the R^2, the better the prediction of the model. So, for R^2 between and 1, we get

```
In [108]: import statsmodels.api as sm
          import statsmodels.formula.api as smf
          temp = df1['POP DENSITY']
          temp1 = df1['CO2 density']
```

```
results = smf.ols('temp1 ~ np.log(temp)', data=df1).fit()

# Inspect the results
print results.summary()
```

```
OLS Regression Results
==============================================================================
Dep. Variable:                  temp1   R-squared:                       0.058
Model:                            OLS   Adj. R-squared:                  0.054
Method:                 Least Squares   F-statistic:                     13.76
Date:                Tue, 25 Nov 2014   Prob (F-statistic):           0.000262
Time:                        08:47:48   Log-Likelihood:                -720.22
No. Observations:                 224   AIC:                             1444.
Df Residuals:                     222   BIC:                             1451.
Df Model:                           1
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      16.6168      2.809      5.916      0.000        11.082    22.152
np.log(temp)   -1.9499      0.526     -3.710      0.000        -2.986    -0.914
==============================================================================
Omnibus:                      158.090   Durbin-Watson:                   1.865
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1229.679
Skew:                           2.836   Prob(JB):                    9.52e-268
Kurtosis:                      12.979   Cond. No.                         38.4
==============================================================================
```

We see that the R^2 value is 0.058, which is extremely less. This means that the predictive power of our model is very poor and not enough features have been considered for us to make an educated guess about the size of the city and how carbon emisions can decide their boundaries.

The last thing we are left to discuss is the cities which produce the highest carbon emissions. This can be seen as follows as we rank the top 5 cities by carbon emissions.

```
In [121]: df2 = df.sort(columns = ['CO2 density'], ascending = False, inplace = False)
          df2['MSA NAME'].head()
```

```
Out[121]: 75             Flagstaff, AZ--UT MSA
          19     Beaumont--Port Arthur, TX MSA
          125             Lake Charles, LA MSA
          154                   Monroe, LA MSA
          239             Wheeling, WV--OH MSA
          Name: MSA NAME, dtype: object
```

5 Conclusion

Thus, we can see that since the value of R^2 is very less, the predictive capability of the model is not too high. We cannot depend on the results of this fit since we have only one feature that has been used to model a system as complex as a city. But, it can give us indications in general.

Some of these indications include the fact that the carbon emission per person in cities with higher population densities were in fact much lesser than those that were not as we can see from the downward slope of the regression lines. We can also see that cities like New York, though are outliers in the analysis. It becomes a dragon-king in the data and skews the regression line. It also helps us understand that cities can in fact reduce the carbon footprint by sharing ammenities such as public transport and heat with other people.

These were some of the key takeaways from this analysis.

References

[1] Arcaute, E., Hatna, E., Ferguson, P., Youn, H., Johansson, A., & Batty, M. (2014). Constructing cities, deconstructing scaling laws. The Royal Society, 12(102).

[2] Oliveira, E., Andrade Jr., J., & Makse, H. (2014). Large cities are less green. Scientific Reports, 4235.