An Exploration of Likely Correlations of CO2 Emissions

Amanda MacGregor

Abstract

The question asked was, which indicators most reliably correlate to a country's CO_2 emission level? After initial data cleaning and narrowing the scope to a recent 15 year window, an initial kmeans exploration of the World Development Indicators dataset provided direcrtion to dig into agricultural activity of those countries using the Global Food & Agriculture Statistics. By transposing, merging and creating a bubble chart of both datasets, we can conclude that CO_2 emissions cannot be traced to any one factor such as population.

Motivation

As countries continue to develop, reliance on potentially poor agricultural practices, manufacturing of certain goods, population, and other factors historically have driven CO_2 emissions higher. By exploring what factors most strongly correlate with higher CO_2 levels, we can know where effort can have the most impact to climate change.

Dataset(s)

World Development Indicators - Kaggle.com

- This dataset provides a multitude of economic factors such as population, CO_2 emissions, and income by country.

Global Food & Agriculture Statistics - Kaggle.com

- This dataset has a large variety of data involving agricultural activity by country, such as if the food produced is for food or feed, the type of food, etc.

Data Preparation and Cleaning

Both datasets are quite large, and required in depth review to understand and apply a variety of filters to extract the needed data. This was the biggest time investment - it took much trial and error.

In addition to filters and dropping n/a's, I also had to figure out how to pivot and transpose the data from long to wide format, and appropriately merge and chart both data sets for the preliminary kmeans review, and later for the final bubble chart.

Research Question(s)

Is population the strongest predictor of CO_2 emission for a given country, since the number of people inherently increases the need for food, energy, and other production?

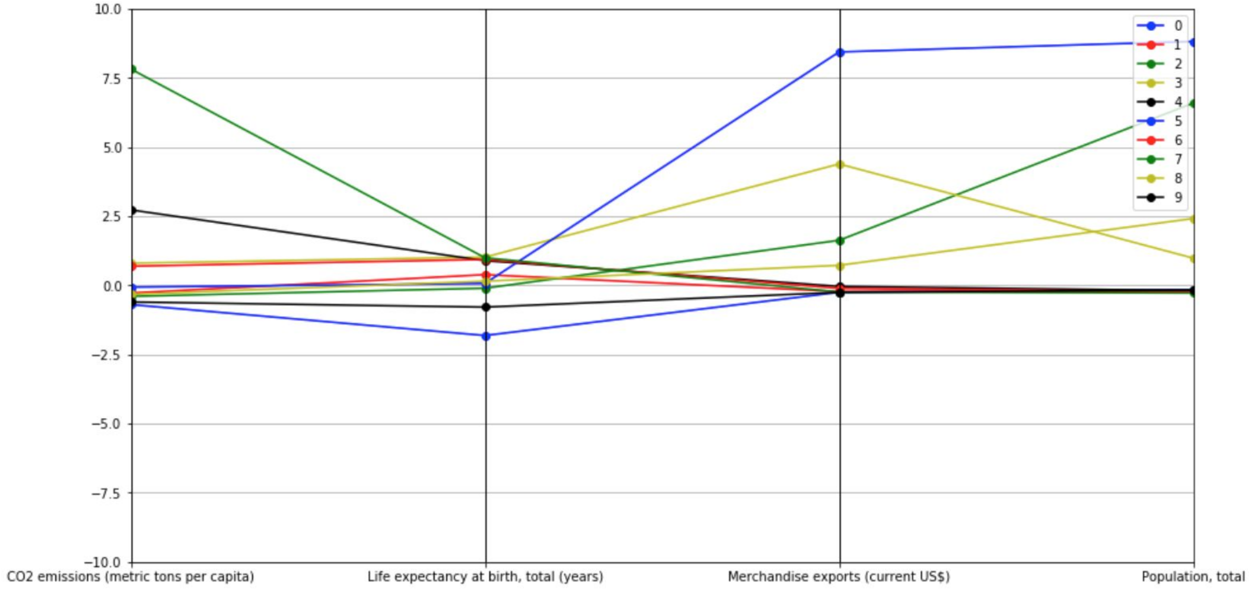
Methods

I imported, cleaned, and filtered the data to prep for a kmeans evaluation to determine if population was a primary driver of C02 levels and if not, where to look next. By clustering different World Indicators, I hoped to achieve some guidance on where to take my analysis next.

This also took pivoting/transposing and merging two different datasets for an inclusive chart to visualize findings.

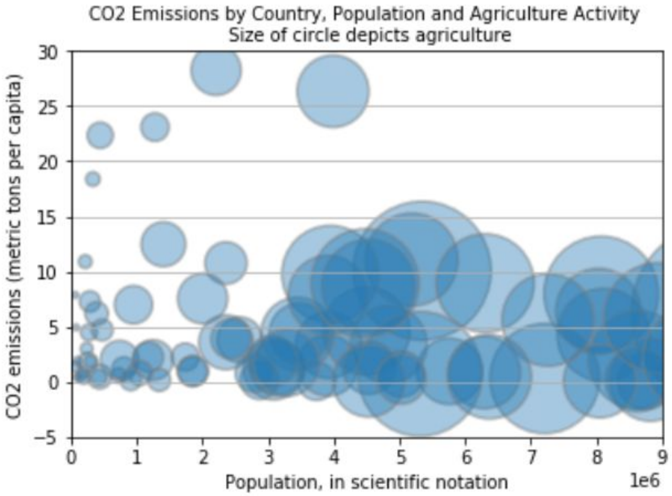
Findings

Once I chose some Indicators from the World Development Indicators dataset, I preformed a kmeans clustering on them to see some potential relationships:



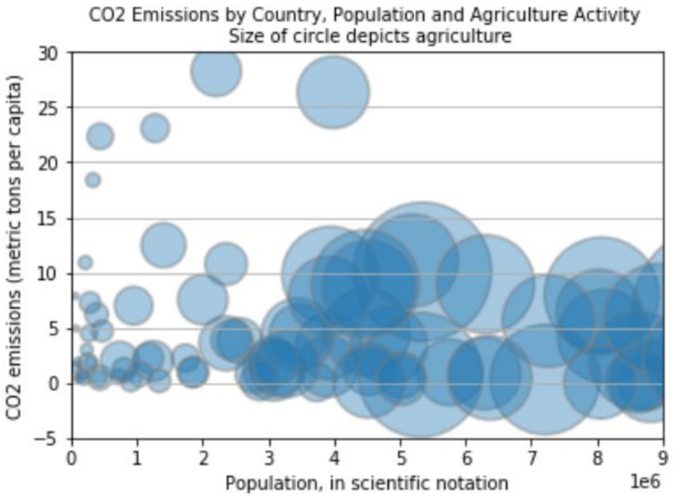
Findings

The kmeans clustering indicated that my assumption that population of countries would be the primary driver of CO_2 levels was not fully accurate. I then decided to review another assumption I had that CO_2 would have a strong relationship to agricultural activity. This result was not as conclusive as I had hoped and will take further action to establish more concrete correlation, but this is a good jumping off point.



Findings

As we would expect, higher population seems to trend with more food production - but neither inherently appear to correlate to CO_2 emissions. The bubble chart depicts some countries producing the highest amount of CO_2 emissions, with a fraction of the population and growing less food than more populated countries.



Limitations

Limitations in this data include a lack of granularity when it comes to countries - I chose to display a few simple metrics in order to get a grasp of broad causes of CO_2 emissions - but there are many, and they should be brought into scope.

I did exclude some NA data during the data-cleaning phase of this process.

Conclusions

Population and CO_2 emissions for countries did not result in the strong correlation I assumed, nor did the next step I took regarding agricultural activity by country.

The logical next step in this conversation would be to pull even more variables into the picture, and look at several trends at once instead of just one or two of what I thought were the more ‘obvious’ ones. Reviewing and adding to the initial kmeans clustering would help with this.

Acknowledgements

Kaggle.com supplied the datasets

Presentation proofed by roommate.

In [182]:
```python
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from itertools import cycle, islice
from pandas.plotting import parallel_coordinates
```

In [10]:
```python
indicators = pd.read_csv('./world-development-indicators/Indicators.csv'
)
```

In [217]:
```python
indicators.head()
```

Out[217]:

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
0	Arab World	ARB	Adolescent fertility rate (births per 1,000 wo...	SP.ADO.TFRT	1960	1.335609e+02
1	Arab World	ARB	Age dependency ratio (% of working-age populat...	SP.POP.DPND	1960	8.779760e+01
2	Arab World	ARB	Age dependency ratio, old (% of working-age po...	SP.POP.DPND.OL	1960	6.634579e+00
3	Arab World	ARB	Age dependency ratio, young (% of working-age ...	SP.POP.DPND.YG	1960	8.102333e+01
4	Arab World	ARB	Arms exports (SIPRI trend indicator values)	MS.MIL.XPRT.KD	1960	3.000000e+06

In [212]:
```
filteredC02 = indicators['IndicatorName'] == 'CO2 emissions (metric tons
per capita)'
filteredPop = indicators['IndicatorName'] == 'Population, total'
filteredLife = indicators['IndicatorName'] == 'Life expectancy at birth,
total (years)'
filteredExp = indicators['IndicatorName'] == 'Merchandise exports (curre
nt US$)'
filteredYear = indicators[(indicators.Year >= 1990)  & (indicators.Year
<= 2015)]
filteredYear.head(10)
```

Out[212]:

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
1880402	Arab World	ARB	Access to electricity (% of population)	EG.ELC.ACCS.ZS	1990	7.544751e+01
1880403	Arab World	ARB	Access to electricity, rural (% of rural popul...	EG.ELC.ACCS.RU.ZS	1990	5.868064e+01
1880404	Arab World	ARB	Access to electricity, urban (% of urban popul...	EG.ELC.ACCS.UR.ZS	1990	9.184235e+01
1880405	Arab World	ARB	Access to non-solid fuel (% of population)	EG.NSF.ACCS.ZS	1990	7.422351e+01
1880406	Arab World	ARB	Adjusted net enrolment rate, primary, both sex...	SE.PRM.TENR	1990	7.210472e+01
1880407	Arab World	ARB	Adjusted net enrolment rate, primary, female (%)	SE.PRM.TENR.FE	1990	6.530609e+01
1880408	Arab World	ARB	Adjusted net enrolment rate, primary, male (%)	SE.PRM.TENR.MA	1990	7.864871e+01
1880409	Arab World	ARB	Adjusted net national income (current US$)	NY.ADJ.NNTY.CD	1990	3.516367e+11
1880410	Arab World	ARB	Adjusted net national income per capita (curre...	NY.ADJ.NNTY.PC.CD	1990	1.584457e+03
1880411	Arab World	ARB	Adjusted net savings, excluding particulate em...	NY.ADJ.SVNX.GN.ZS	1990	2.432691e+00

In [194]:
```
dfC02 = indicators[filteredC02]
dfPop = indicators[filteredPop]
dfLife = indicators[filteredLife]
dfExp = indicators[filteredExp]
dfExp.head()
```

Out[194]:

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
23	Arab World	ARB	Merchandise exports (current US$)	TX.VAL.MRCH.CD.WT	1960	4.645919e+09
104	Caribbean small states	CSS	Merchandise exports (current US$)	TX.VAL.MRCH.CD.WT	1960	6.237171e+08
283	East Asia & Pacific (all income levels)	EAS	Merchandise exports (current US$)	TX.VAL.MRCH.CD.WT	1960	1.531711e+10
403	East Asia & Pacific (developing only)	EAP	Merchandise exports (current US$)	TX.VAL.MRCH.CD.WT	1960	6.249855e+09
536	Euro area	EMU	Merchandise exports (current US$)	TX.VAL.MRCH.CD.WT	1960	3.479041e+10

In [54]:
```
indicators.dftypes
```

Out[54]:
```
CountryName       object
CountryCode       object
IndicatorName     object
IndicatorCode     object
Year               int64
Value            float64
dtype: object
```

In [460]:
```
joined1 = pd.merge(dfC02, dfPop, how = 'outer')
joined2 = pd.merge (joined1, dfLife, how = 'outer')
joined3 = pd.merge(joined2, dfExp, how = 'outer')
joined4 = pd.merge(joined3, dfPop, how = 'outer')
final = joined4[(joined4.Year >= 1990)  & (joined4.Year <= 2011)]
final.head()
```

Out[460]:

	CountryName	CountryCode	IndicatorName	IndicatorCode	Year	Value
5670	Arab World	ARB	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	1990	3.203907
5671	Caribbean small states	CSS	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	1990	5.367886
5672	Central Europe and the Baltics	CEB	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	1990	8.847908
5673	East Asia & Pacific (all income levels)	EAS	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	1990	2.600991
5674	East Asia & Pacific (developing only)	EAP	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	1990	1.803359

In [142]:
```
final.transpose()
```

Out[142]:

	5670	5671	5672	5673	
CountryName	Arab World	Caribbean small states	Central Europe and the Baltics	East Asia & Pacific (all income levels)	East (developir
CountryCode	ARB	CSS	CEB	EAS	
IndicatorName	CO2 emissions (metric tons per capita)	CO2 emissions (metric tons per capita)	CO2 emissions (metric tons per capita)	CO2 emissions (metric tons per capita)	CO2 em (metric t
IndicatorCode	EN.ATM.CO2E.PC	EN.ATM.CO2E.PC	EN.ATM.CO2E.PC	EN.ATM.CO2E.PC	EN.ATM.C(
Year	1990	1990	1990	1990	
Value	3.20391	5.36789	8.84791	2.60099	1

6 rows × 20372 columns

In [462]:
```
summary = final.groupby(['CountryName', 'IndicatorName'], as_index = False).agg({'Value':'mean'})
summary.transpose()
summary.head(10)
```

Out[462]:

	CountryName	IndicatorName	Value
0	Afghanistan	CO2 emissions (metric tons per capita)	1.158157e-01
1	Afghanistan	Life expectancy at birth, total (years)	5.514472e+01
2	Afghanistan	Merchandise exports (current US$)	2.672001e+08
3	Afghanistan	Population, total	2.058324e+07
4	Albania	CO2 emissions (metric tons per capita)	1.152550e+00
5	Albania	Life expectancy at birth, total (years)	7.438587e+01
6	Albania	Merchandise exports (current US$)	5.549526e+08
7	Albania	Population, total	3.086820e+06
8	Algeria	CO2 emissions (metric tons per capita)	3.080816e+00
9	Algeria	Life expectancy at birth, total (years)	7.036701e+01

In [465]:
```
test = final.pivot_table(index = 'CountryName', columns = 'IndicatorName', values = 'Value')
test.head()
```

Out[465]:

IndicatorName	CO2 emissions (metric tons per capita)	Life expectancy at birth, total (years)	Merchandise exports (current US$)	Population, total
CountryName				
Afghanistan	0.115816	55.144717	2.672001e+08	2.058324e+07
Albania	1.152550	74.385871	5.549526e+08	3.086820e+06
Algeria	3.080816	70.367010	2.936674e+10	3.133294e+07
American Samoa	NaN	NaN	3.634331e+08	5.521114e+04
Andorra	6.926680	NaN	NaN	7.085382e+04

In [348]:
```
test2 = test.dropna()
```

In [349]:
```
X = StandardScaler().fit_transform(test2)
```

In [200]:
```
kmeans = KMeans(n_clusters=10)
model = kmeans.fit(X)
print("model\n", model)
centers = model.cluster_centers_
```

```
model
 KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=10, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

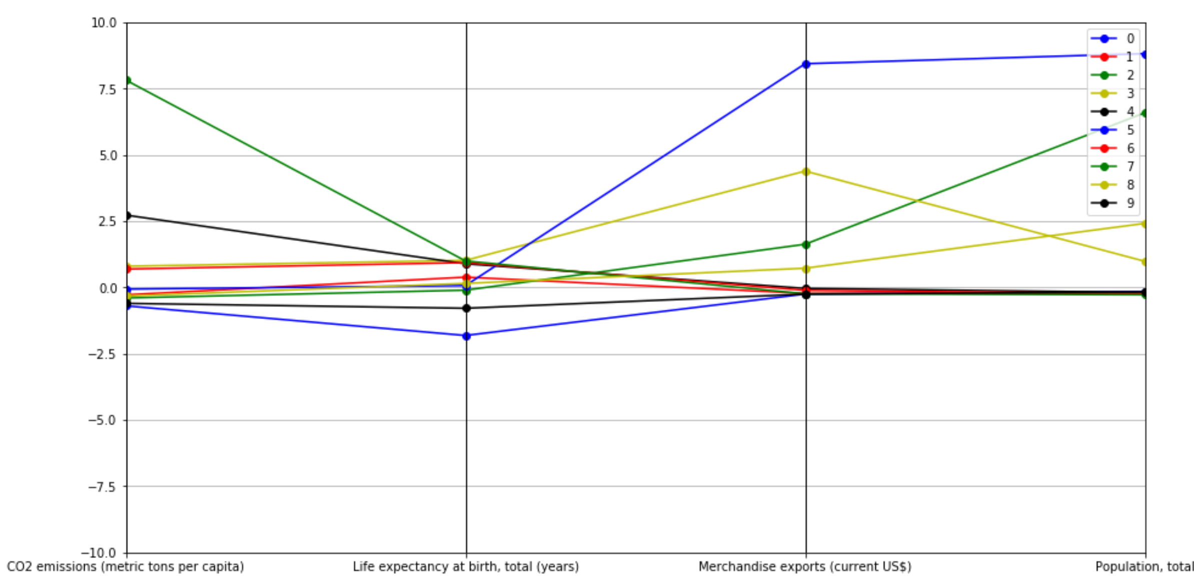
In [204]:
```
features = ['CO2 emissions (metric tons per capita)', 'Life expectancy a
t birth, total (years)', 'Merchandise exports (current US$)', 'Populatio
n, total']
```

In [201]:
```
def pd_centers(featuresUsed, centers):
        colNames = list(featuresUsed)
        colNames.append('prediction')

        Z = [np.append(A, index) for index, A in enumerate(centers)]

        P = pd.DataFrame(Z, columns=colNames)
        P['prediction'] = P['prediction'].astype(int)
        return P
```

In [207]:
```
def parallel_plot(data):
        my_colors = list(islice(cycle(['b', 'r', 'g', 'y', 'k']), None,
len(data)))
        plt.figure(figsize=(15,8)).gca().axes.set_ylim([-10,+10])
        parallel_coordinates(data, 'prediction', color = my_colors, mark
er='o')
```

In [289]:
```
P = pd_centers(features, centers)
```

In [209]:
```
parallel_plot(P)
```



In [264]:
```
agri = pd.read_csv('./FAO.csv', encoding = "ISO-8859-1")
```

In [288]:
```
agri.head()
```
Out[288]:

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitu
0	AFG	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.
1	AFG	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.
2	AFG	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.
3	AFG	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.
4	AFG	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.

5 rows × 64 columns

In [266]:
```
agri['10Year']=agri['Y2003'] + agri['Y2004'] + agri['Y2005'] + agri['Y20
06'] + agri['Y2007'] + agri['Y2008'] + agri['Y2009'] + agri['Y2010'] + a
gri['Y2011'] + agri['Y2012'] + agri['Y2013']
agri.head()
```
Out[266]:

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitu
0	AFG	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.
1	AFG	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.
2	AFG	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.
3	AFG	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.
4	AFG	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.

5 rows × 64 columns

In [322]:
```
summary2 =agri.groupby(['Area', 'Element'], as_index = False).agg({'10Year':'mean'})
summary2.head()
```

Out[322]:

	Area	Element	10Year
0	Afghanistan	Feed	1373.800000
1	Afghanistan	Food	2766.917808
2	Albania	Feed	495.720000
3	Albania	Food	695.285714
4	Algeria	Feed	2758.772727

In [357]:
```
agri_final = summary2.pivot_table(index = 'Area', columns = 'Element', values = '10Year')
agri_final.reset_index(inplace=True)
agri_final.rename(columns={'Area': 'CountryName'}, inplace=True)
agri_final.head()
```

Out[357]:

Element	CountryName	Feed	Food
0	Afghanistan	1373.800000	2766.917808
1	Albania	495.720000	695.285714
2	Algeria	2758.772727	5487.352941
3	Angola	8212.266667	2590.872340
4	Antigua and Barbuda	0.307692	11.711538

In [496]:
```
indFinal = test2.drop(['Life expectancy at birth, total (years)', 'Merchandise exports (current US$)'], axis=1)
indFinal.reset_index(inplace=True)
indFinal.head()
```

Out[496]:

IndicatorName	CountryName	CO2 emissions (metric tons per capita)	Population, total
0	Afghanistan	0.115816	2.058324e+07
1	Albania	1.152550	3.086820e+06
2	Algeria	3.080816	3.133294e+07
3	Angola	0.820365	1.583110e+07
4	Antigua and Barbuda	4.937855	7.622632e+04

In [408]:

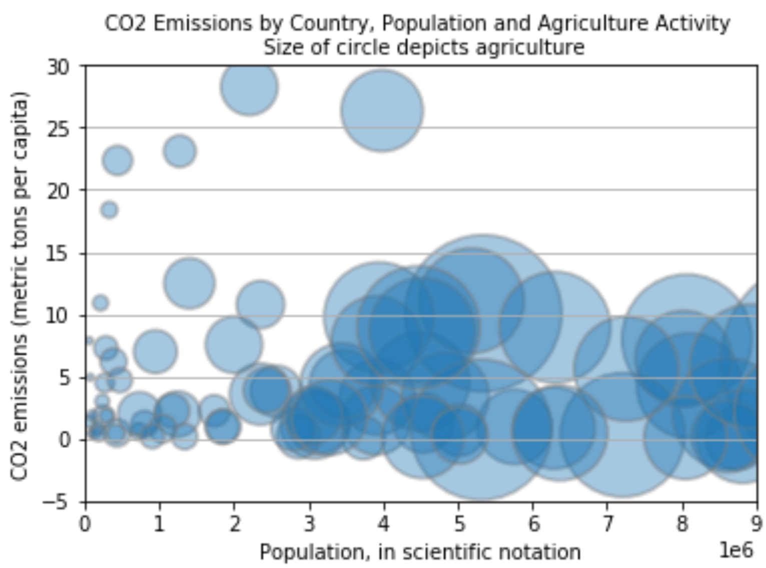
```
agri_finalset = pd.merge(agri_final, indFinal, how = 'left')
agri_finalset.head()
plot_data = agri_finalset.assign(TotalFood = agri_finalset.Feed + agri_f
inalset.Food)
plot_data2 = plot_data.assign(PercentFeed = plot_data.Feed / plot_data.T
otalFood * 100)
plot_data2.head()
```

Out[408]:

	CountryName	Feed	Food	CO2 emissions (metric tons per capita)	Population, total	TotalFood	PercentFeed
0	Afghanistan	1373.800000	2766.917808	0.115816	2.058324e+07	4140.717808	33.177822
1	Albania	495.720000	695.285714	1.152550	3.086820e+06	1191.005714	41.621967
2	Algeria	2758.772727	5487.352941	3.080816	3.133294e+07	8246.125668	33.455381
3	Angola	8212.266667	2590.872340	0.820365	1.583110e+07	10803.139007	76.017412
4	Antigua and Barbuda	0.307692	11.711538	4.937855	7.622632e+04	12.019231	2.560000

In [495]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

fig, axis = plt.subplots()
axis.yaxis.grid(True)
axis.set_title('CO2 Emissions by Country, Population and Agriculture Act
ivity \n Size of circle depicts agriculture',fontsize=10)
axis.set_xlabel('Population, in scientific notation',fontsize=10)
axis.set_ylabel('CO2 emissions (metric tons per capita)',fontsize=10)
y = plot_data2['CO2 emissions (metric tons per capita)']
z = plot_data2['TotalFood']
x = plot_data2['Population, total']

plt.scatter(x, y, s= z, cmap = 'Blues', alpha = 0.4, edgecolors = "grey"
, linewidth = 2)
plt.xlim(0,9e+6)
plt.ylim(-5, 30)
plt.ticklabel_format(style = 'sci', axis = 'x', scilimits = (0,0))
plt.show()
```



References

https://www.kaggle.com/worldbank/world-development-indicators

https://www.kaggle.com/unitednations/global-food-agriculture-statistics

Heavily referenced workbooks from this course, as well as StackOverflow for random syntax reviews.