Aashrays Homework 1 - Exploratory Data Analysis in IPython

September 16, 2014

0.1 Aashray's Data Science Home Work 1 : Exploratory Data Analysis in IPython.

0.1.1 Aim

The data set we will analyze concerns statistics about the nations of the world, and is available at http://www.cs.stonybrook.edu/~skiena/591/hw1/country-data.csv. We have assembled a table of information about each country, with approximately 20 fields including: name, countrycode, type of government, longitude and latitude of capital city, population, life expectancy, GDP, area, literacy rate, and more. You are to explore this data and uncover interesting observations about the success and fate of nations. You are to return all your results in a single, well-documented IPython notebook documenting your methods and the exact sequence of operations you needed to produce the resulting tables and figures.

0.1.2 Task -1

Produce five informative plots revealing aspects of this data. These must include
At least one data map
At least one scatter plot
At least one histogram or bar chart
For each plot, write a paragraph in your notebook showing interesting stuff the visualization reveals.
Import numpy

```
In [412]: import numpy as np
          from __future__ import print_function
```

Best practice to import matplotlib

```
In [413]: %matplotlib inline
          import matplotlib.pyplot as plt
```

The country-data.csv file given for the assignment contains some stings (Example : countrty name) which have a comma in them. Example of one such occurance is on the the 13th line where the name of the country is "Bahamas, The". Numpy's genfromtxt fuction does not handle such cases. Now I have two options : - Edit the CSV file and replace the comma in the country name - Use pandas, pandas has a parameter in its read_csv function to specify a quotestring, that will take care of this.

I think many times in data science we will data such cases where the data is incorrect or does not fit well, better to handle in my program rather than depend on fixing the file every time I find such things. **Hence, I will use pandas.**

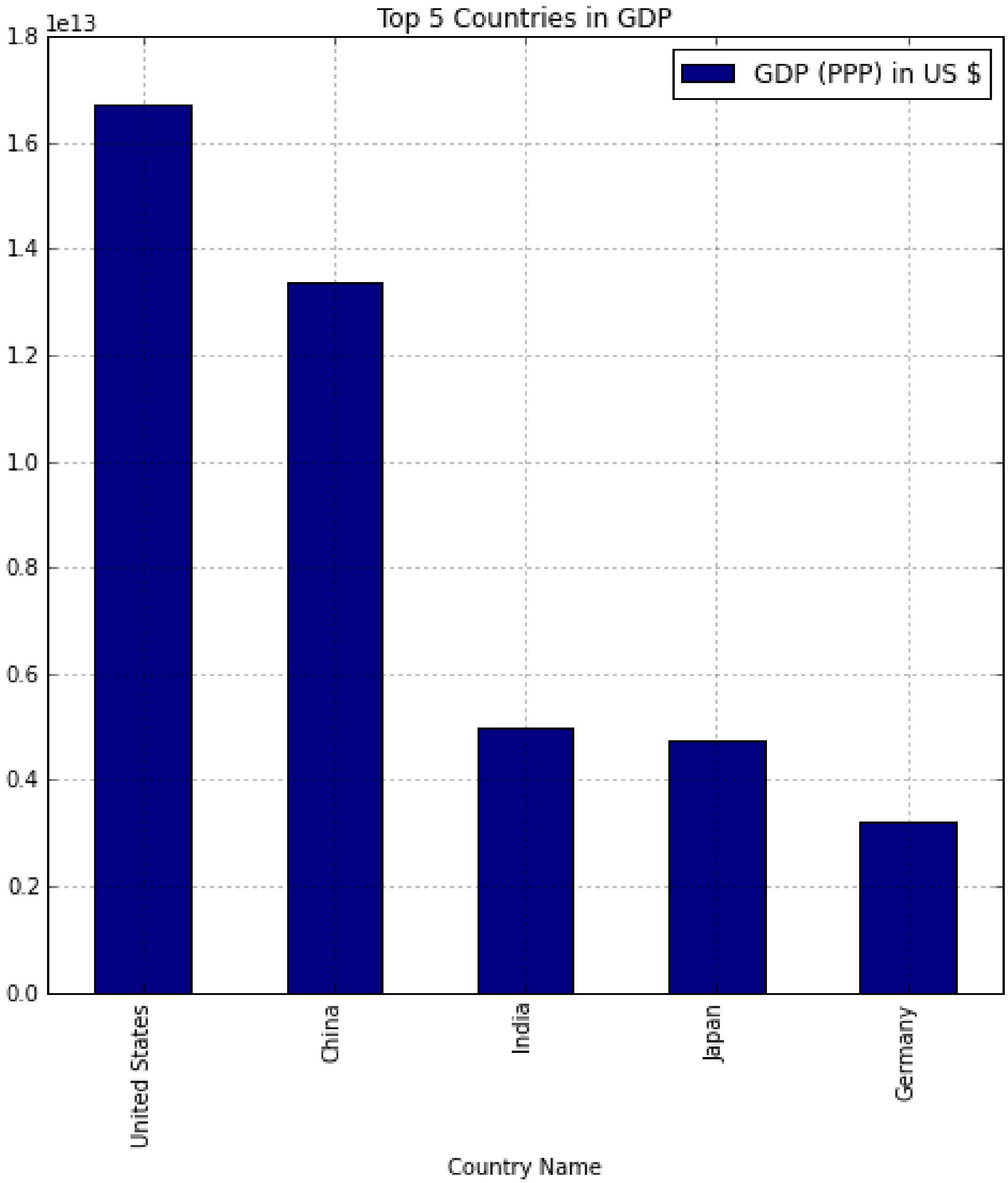
```
In [414]: import pandas as pd
          from pandas import Series, DataFrame
```

```
In [415]: # Task - 1 : part 1 - A histogram
          df=pd.read_csv('country-data.csv',quotechar='"',skipinitialspace=True)
          df = df.set_index('Country Name');
```

Alright ! So I have got my data in a pandas DataFrame now.
HISTOGRAM

In [416]:
```
fig = plt.figure(num=None, figsize=(8, 6), dpi=80, facecolor='w', edgecolor='b')
result= df.sort(['GDP (PPP) in US $'], ascending=False)
result = result['GDP (PPP) in US $']
result.head(5).plot(kind='bar',figsize=(8,8),legend=True,title="Top 5 Countries in GDP",colorm
```

Out[416]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11c22e610>
```

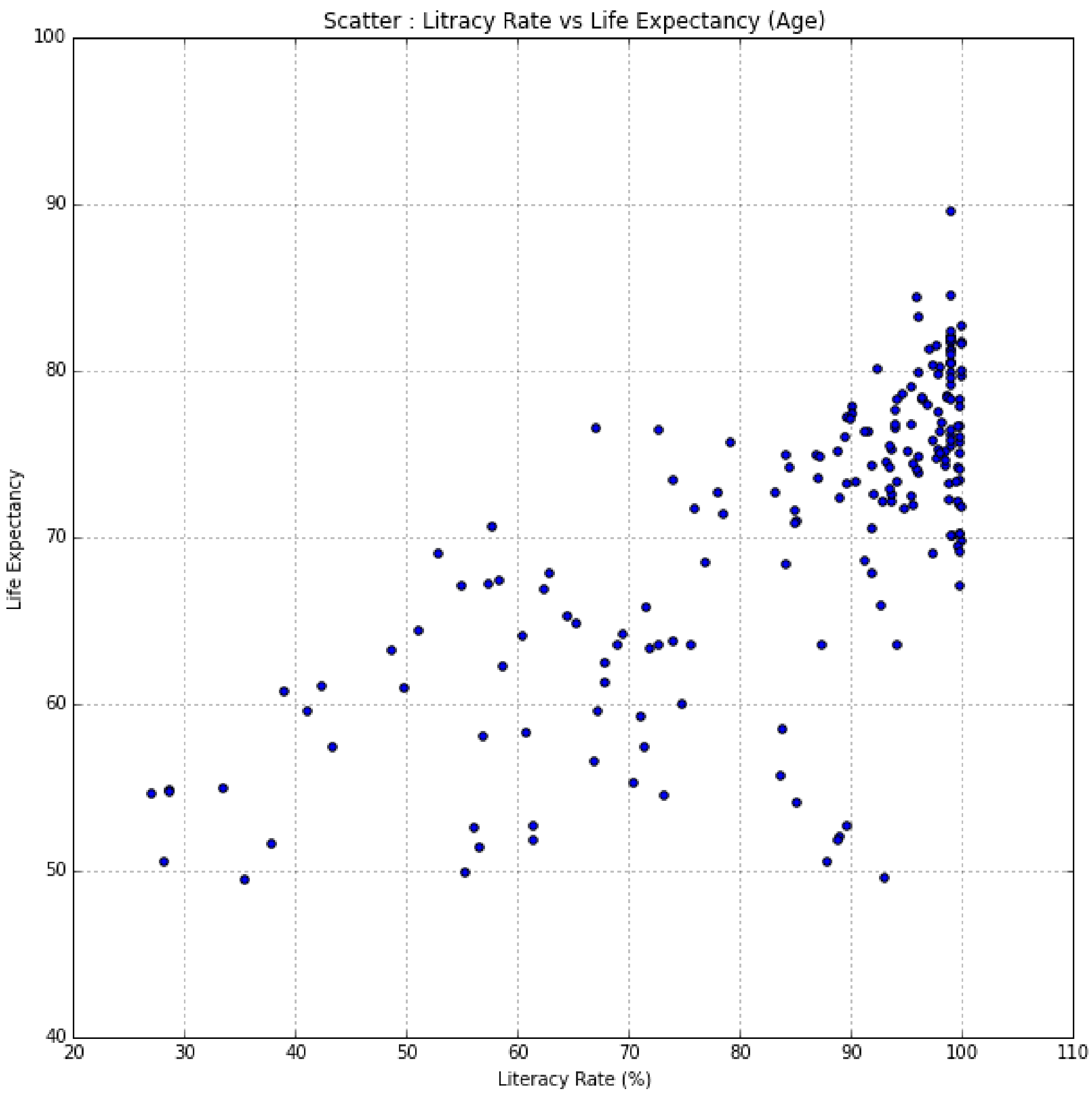


This is a simple histogram, more complex plots follow. This plot reveals that the top 5 countries by GDP (PPP) are United states, China, India, Japan and Germany. It show significantly higher the United states is compared to the rest. The Difference between the second and Third positions is very significant as well (China and India).

SCATTER PLOT

```
In [417]: df_scatter = df.convert_objects(convert_numeric=True)
          df_scatter.plot(kind='scatter',figsize=(10,10),x='Literacy Rate (%)',y='Life Expectancy',leger
```

```
Out[417]: <matplotlib.axes._subplots.AxesSubplot at 0x11c2ed390>
```

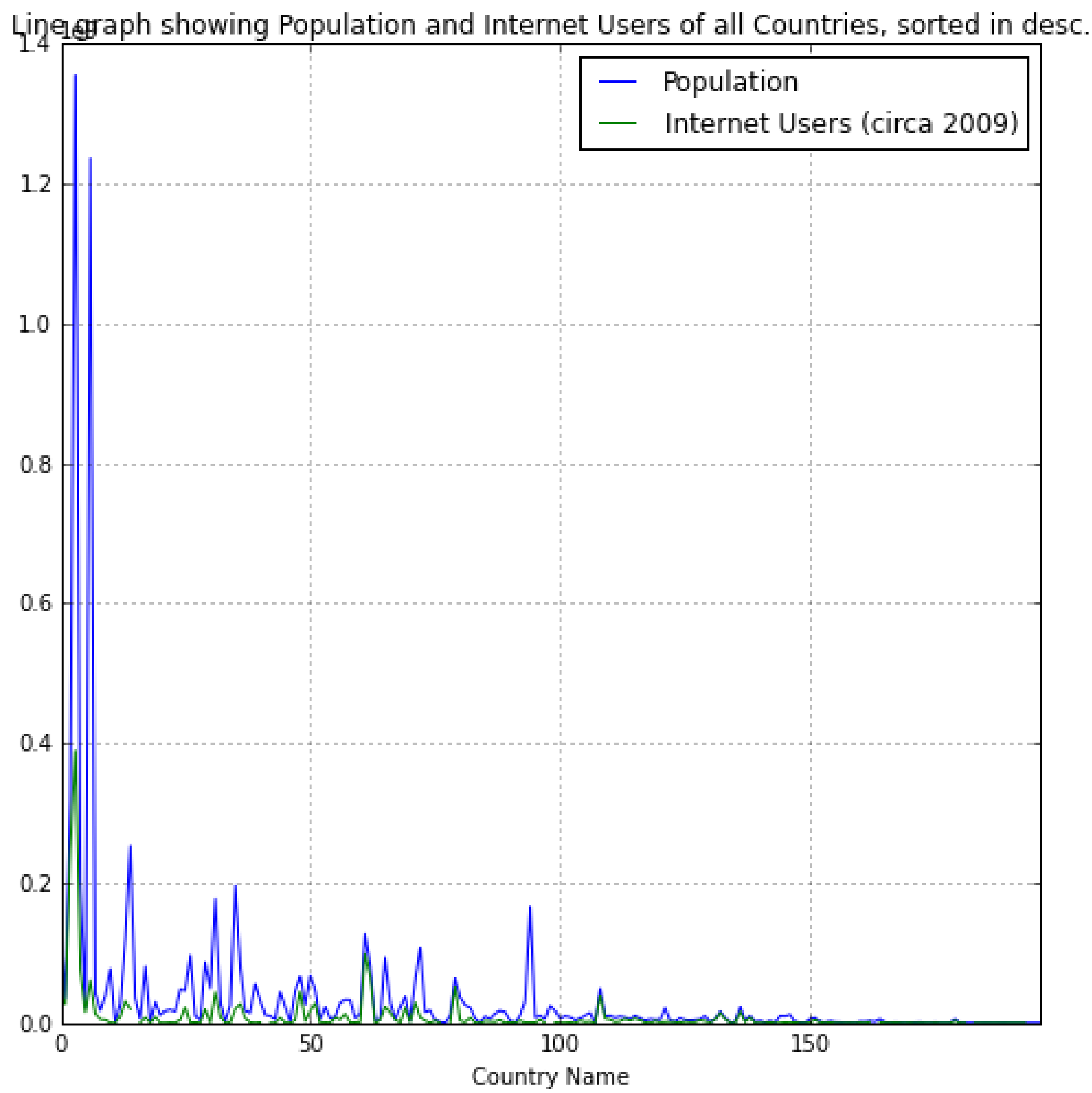


This is a scatter plot of Litrarcy Rate in percentage on the x-axis and Life Expectancy on the y-axis. This graph is very interesting. Apart from a few outliers, this graph basically shows and increased Life Expectancy when the Literacy is high and visa- versa. Right at the top is Monaco, which is not a mojpr country and can hence be considered at outlier. Countries with almost 99-100% Literacy rates have above 70 Life Expectancies, some even reach above 80.

LINE GRAPH

```
In [418]: result2= df.sort(['Area (sq km)'], ascending=False)
          result2= result2.convert_objects(convert_numeric=True)
          result2 = result2[['Population','Internet Users (circa 2009)']]
          #result2 = result2.div(result2.sum(axis=0), axis=1)
          result2.plot(use_index=False,kind='line',figsize=(8,8),legend=True,title="Line graph showing P
```

```
Out[418]: <matplotlib.axes._subplots.AxesSubplot at 0x112623990>
```



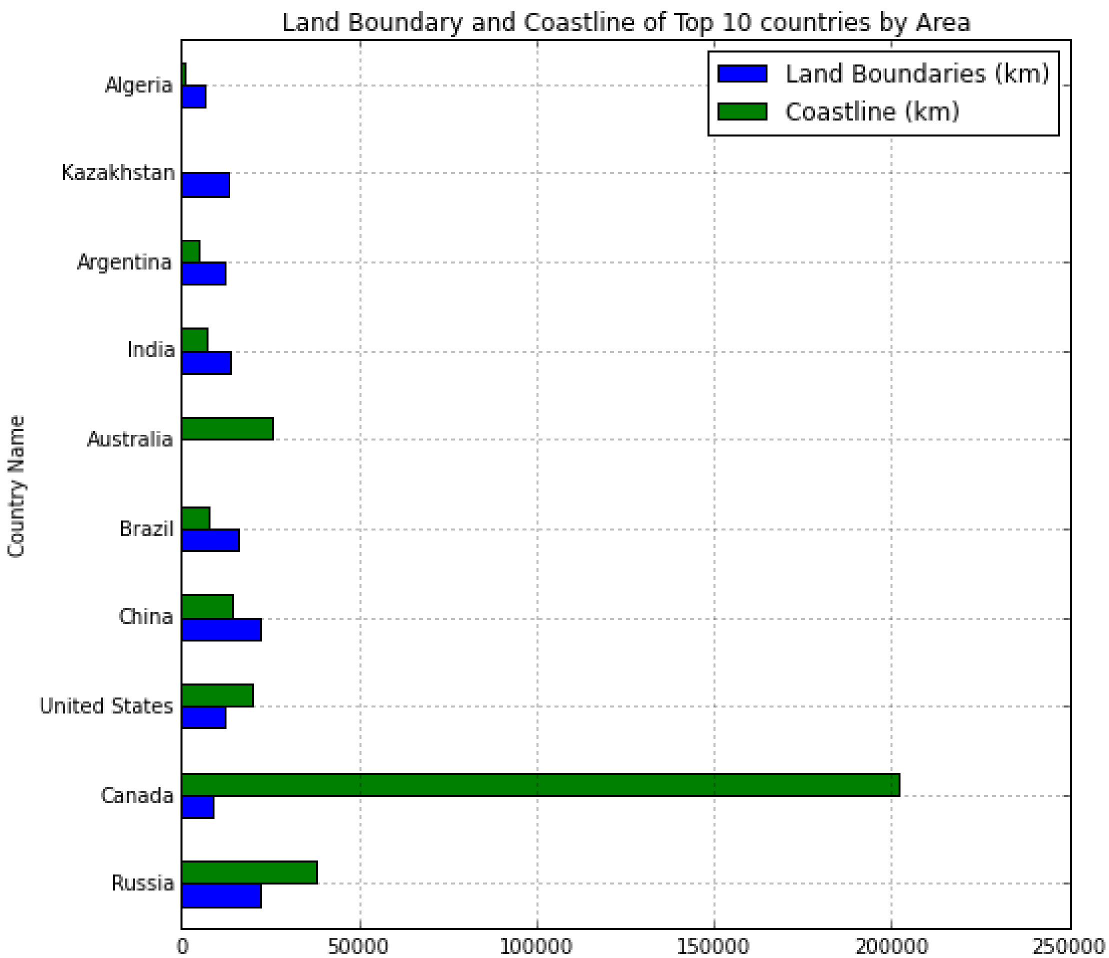
This simple graph reveals the direct correspondance between Population and Internet users. Larger population implies great percentage of it using the internet. There are not too many suprises where the number of internet users is not directly propotional to the population, which is natural. There are few countries however (many the more developed ones) where the number of internet users is closer to the total population than other countries.

HORIZONTAL HISTOGRAM WITH TWO BARS

```
In [419]: result2= df.sort(['Area (sq km)'], ascending=False)
          result2= result2.convert_objects(convert_numeric=True)
```

```
result2 = result2[['Land Boundaries (km)','Coastline (km)']]
#result2 = result2.div(result2.sum(axis=0), axis=1)
result2.head(10).plot(use_index=True,kind='barh',figsize=(8,8),legend=True,title="Land Bounda
```

Out[419]: <matplotlib.axes._subplots.AxesSubplot at 0x11cd86e10>



This data reveals many interesting things. I have taken the 10 largest (by area) countries and compared their Landoundaries and costline. Canada has then second most area, but its Coastline is significantly larger than any other country compared.

Australia on the other hand has absolutely no coastline ! It is the 6 largest country by area. It has the third largest coastline in this graph.

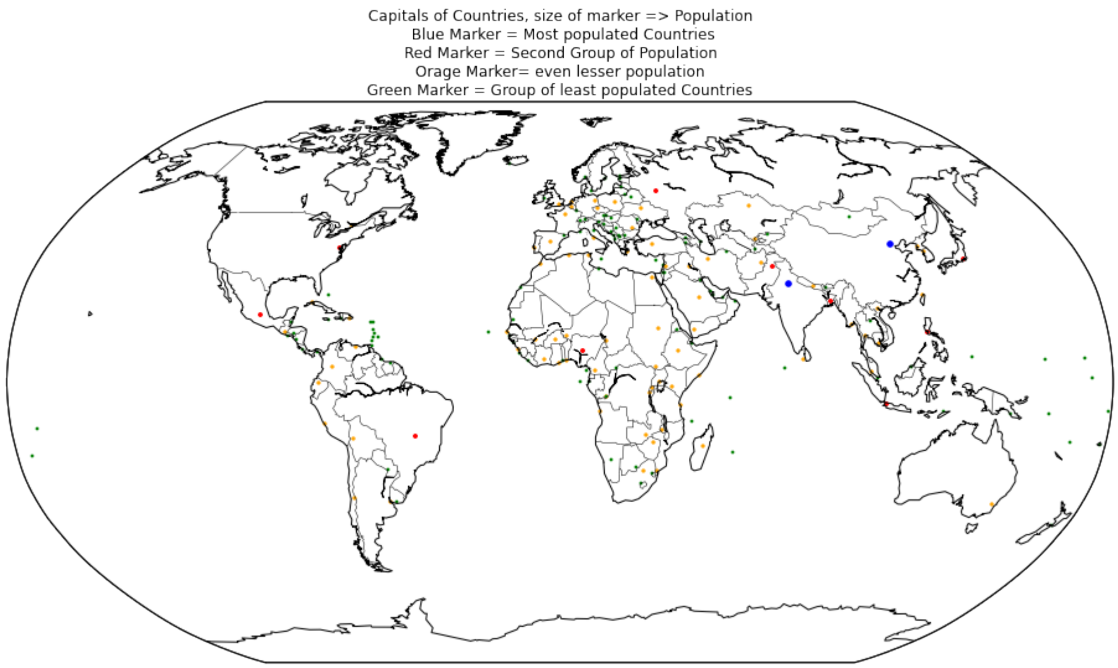
Kazakhastan has no coastline.

DATA MAP

In [420]:
```
from mpl_toolkits.basemap import Basemap
```

In [421]:
```
plt.figure(figsize=(16,16))
map = m = Basemap(projection='robin',lon_0=0,resolution='c')
map.drawcoastlines()
map.drawcountries()
map.drawmapboundary()
```

```
dfmap = df[['Latitude of Capital','Longitude of Capital']]
s = dfmap['Latitude of Capital'].str.split(' ').apply(Series, 1).stack()
s.index = s.index.droplevel(-1) # to line up with df's index
s.name = 'Lat - Split' # needs a name to join
s2 = dfmap['Longitude of Capital'].str.split(' ').apply(Series, 1).stack()
s2.index = s2.index.droplevel(-1) # to line up with df's index
s2.name = 'Long - Split' # needs a name to join
signlati = 1
signlongi = 1
for i in dfmap.index: # i is country name
    signlati=1
    signlongi=1
    try:
        if(s[i][2]=='N'):
            lat = s[i][0]+"."+s[i][1]
        else:
            lat = s[i][0]+"."+s[i][1]
            signlati = -1
        if(s2[i][2]=='E'):
            longi = s2[i][0]+"."+s2[i][1]
        else:
            longi = s2[i][0]+"."+s2[i][1]
            signlongi = -1
        longi = float(longi)
        lat = float(lat)
        xpt,ypt = m(longi*signlongi,lat*signlati)
    except KeyError:
        continue
    if((len(str(df['Population'][i])))>9):
        mycolor='blue'
        mysize=20
    elif((len(str(df['Population'][i])))>8):
         mycolor='red'
         mysize = 8
    elif((len(str(df['Population'][i])))>7):
         mycolor='orange'
         mysize = 5
    else:
        mycolor ='green'
        mysize = 3
    m.scatter(xpt,ypt,mysize,marker='o',color=mycolor)
plt.title('Capitals of Countries, size of marker => Population\n Blue Marker = Most populated
plt.show()
```



0.1.3 Task - 2

Q : Do a pairwise correlation on all pairs of variables to identify which pairs correlate the strongest and which the weakest. Do a permutation test to determine the p-value of each observed correlation to test which are significant (what fraction of permutations produce at least this high a correlation).

```
In [422]: corr = df.corr() # non-numeric columns are automatically excluded from the correlation calcul
          print ("Correlations between variables in data set:\n")
          print (corr)
```

```
Correlations between variables in data set:

                      Population  Life Expectancy  GDP (PPP) in US $  \
Population              1.000000         0.014249           0.697280
Life Expectancy         0.014249         1.000000           0.175071
GDP (PPP) in US $       0.697280         0.175071           1.000000
Area (sq km)            0.453228         0.033022           0.592445
Land Boundaries (km)    0.575146        -0.219494           0.500468
Coastline (km)          0.120474         0.162933           0.204414

                      Area (sq km)  Land Boundaries (km)  Coastline (km)
Population                0.453228              0.575146        0.120474
Life Expectancy           0.033022             -0.219494        0.162933
GDP (PPP) in US $         0.592445              0.500468        0.204414
Area (sq km)              1.000000              0.749098        0.521336
Land Boundaries (km)      0.749098              1.000000        0.195977
Coastline (km)            0.521336              0.195977        1.000000
```

This is interesting. It shows high correlation between Area and Land Booundaries ˜0.749. Which seems natural. Low (read negative) correlation between Area and Life Expectancy. Low between Life expectancy and Land Boundary. High Correlation (~0.697) between Population and GDP (PPP). Also moderately high between GDP (PPP) and Area.

```
In [423]: N = np.sum(corr)
          t = corr*np.sqrt((N-2)/(1-corr**2))
          import scipy.stats #Cumulative density function.
          p = 1-scipy.stats.t.cdf(abs(t),N-2)  # one-tailed
          print ("P-Values:")
          print(p)
```

```
P-Values:
[[ 0.                 nan  0.23146971  0.31915513  0.33107804  0.48907075]
 [ 0.49591925         nan  0.43770889  0.48713514  0.4396873   0.48516473]
 [ 0.27571969         nan  0.          0.25889931  0.35603467  0.48130024]
 [ 0.3645799          nan  0.27799017  0.          0.26353935  0.44835005]
 [ 0.32290429         nan  0.31578053  0.18448448  0.          0.4820908 ]
 [ 0.4654032          nan  0.42715344  0.29016949  0.44625792  0.        ]]
```

There are 7 columns. I've added a new column for GDP PER CAPITA which is nee for the next question. The p-values shows are shown for each correlation between these columns.

0.1.4 Task -3

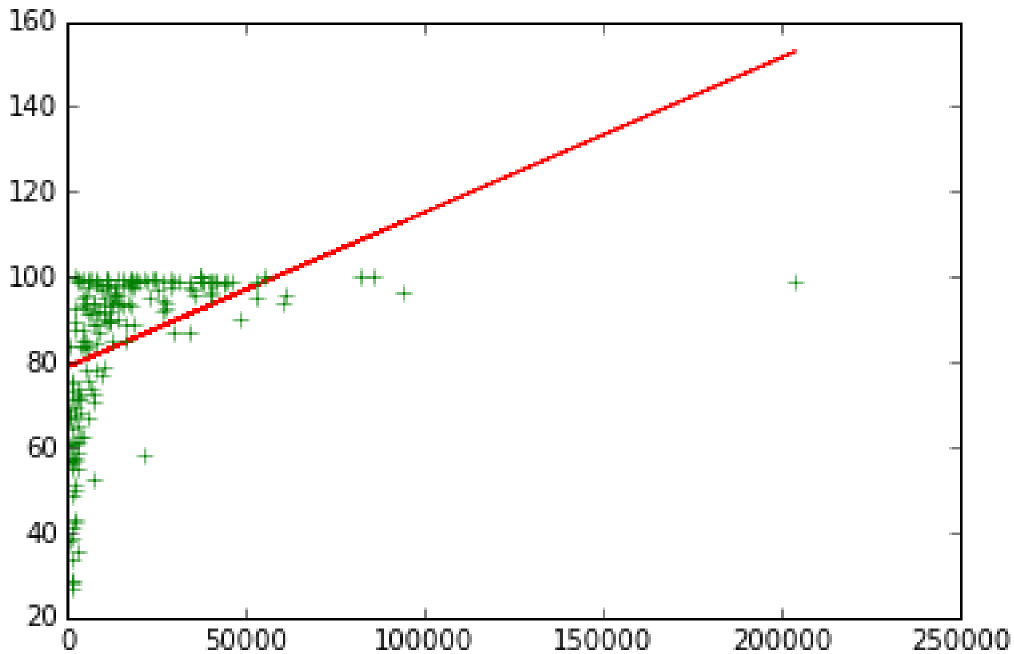
Q : Set up a simple linear regression model to predict the average income (GDP per capita) as a function of one or more of the other variables. Which countries are most above the forecast? Which are most below? Can you explain why?

I am going to set up a simple linear regression model to predict the average income (GDP per capita) as a function of Literacy rate .

```
In [424]: #linear regresion


          from scipy import stats
          from pylab import plot,show
          #First Calculate the GDP PPP (Purchasing Power Parity) per capita
          gdbpercapita = df['GDP (PPP) in US $']/df['Population']
          df['GDP PPP PER CAPITA'] = Series(gdbpercapita, index=df.index);
          y = gdbpercapita
          x = df['Literacy Rate (%)']
          dfgdb_lit = pd.concat([y, x], axis=1)
          #print (dfgdb_lit)
          dfgdb_lit = dfgdb_lit[dfgdb_lit[1] != 'unknown'] # remove rows that have literacy rate as 'un
```

```
In [425]: y = dfgdb_lit[1].astype(float) # litrarcy rate
          x = dfgdb_lit[0] # GDP
          #print (df.sort(['Literacy Rate (%)'], ascending=False).head(5)) #also sort by GDP PPP PER CA
          slope, intercept, r_value, p_value, std_err = stats.linregress(x,y)
          line = slope*x+intercept
          plot(x,line,'r-',x,y,'g+') # o - circle marker, g - green, + - plus sign marker
          show()
```



Linear regresion - GDP PER CAPITA as function of Literacy rate.

The country Monaco is move above the forcast. Reason is that it has the highest GDP and a high Literracy rate, but it is an outlier because it is a very small country and has a very very small population and almost all of the population is literate.

Liechtenstein is also very high in GDP PER CAPITA at 85760.994828 GDP. But it fits close to the line.

USA fits this model the best, almost falling on the red line.

Azerbaijan has a literary rate of 99.8 but GDP PER CAPITA of only 10602.702192. So it is very below the forecast.

0.1.5 Task - 4

Set up a scoring/ranking function to measure general social welfare. Which countries do best by your measure? Which do worst? Write a few paragraphs to describe your measure and evaluate how good/bad you think the results are.

Ans : I am defining social welfare as a function of GDP per capita and Life expectancy.

```
In [426]: df_welfare = df[['GDP PPP PER CAPITA', 'Life Expectancy']]
          df_welfare_rank = df_welfare.rank()
          df_welfare_rank['Welfare Rank'] = (df_welfare_rank['GDP PPP PER CAPITA']+ df_welfare_rank['Li
          df_welfare_rank = df_welfare_rank.rank(ascending=False);
          df_welfare_rank = df_welfare_rank.sort('Welfare Rank')
          df_welfare_rank['Welfare Rank']
```

```
Out[426]: Country Name
          Monaco                              1.0
          Singapore                           2.0
          Liechtenstein                       3.0
          Switzerland                         4.0
          Australia                           5.0
          Norway                              6.0
```

San Marino	7.0
Canada	8.0
Sweden	9.0
Japan	10.0
Luxembourg	11.0
Netherlands	13.0
Andorra	13.0
Iceland	13.0
Austria	15.0
Qatar	16.0
United States	17.0
Germany	18.0
Ireland	19.0
France	20.0
Italy	21.5
Belgium	21.5
Israel	23.5
United Kingdom	23.5
Taiwan	25.0
Kuwait	26.0
Spain	27.0
New Zealand	28.0
Denmark	29.0
Finland	30.0
United Arab Emirates	31.0
Brunei	32.0
Korea, South	33.0
Malta	34.0
Greece	35.0
Bahrain	36.0
Czech Republic	37.0
Slovenia	38.5
Portugal	38.5
Chile	40.0
Cyprus	41.0
Slovakia	42.0
Panama	43.0
Poland	44.0
Argentina	45.0
Oman	46.0
Saudi Arabia	47.0
Uruguay	48.0
Lithuania	49.5
Croatia	49.5
Costa Rica	51.0
Hungary	52.0
Saint Lucia	53.0
Barbados	54.0
Antigua and Barbuda	55.0
Saint Kitts and Nevis	56.0
Dominica	57.0
Seychelles	58.0
Cuba	59.0
Estonia	61.0

Lebanon	61.0
Mexico	61.0
Mauritius	63.0
Albania	64.5
Bahamas, The	64.5
Dominican Republic	66.5
Greenland	66.5
Libya	68.5
Malaysia	68.5
Ecuador	70.5
Latvia	70.5
Bulgaria	72.0
Venezuela	73.0
Romania	75.5
Macedonia	75.5
Saint Vincent and the Grenadines	75.5
Trinidad and Tobago	75.5
Colombia	78.0
Serbia	80.0
Bosnia and Herzegovina	80.0
Tunisia	80.0
Paraguay	82.0
Turkey	83.0
Montenegro	84.0
Algeria	85.5
Grenada	85.5
Tonga	87.0
China	88.0
Belarus	89.0
Maldives	90.0
Sri Lanka	91.5
Brazil	91.5
Thailand	93.0
Morocco	94.5
Russia	94.5
Peru	96.0
Timor-Leste	97.0
Palau	98.0
Georgia	99.0
Equatorial Guinea	100.0
El Salvador	101.0
Suriname	102.5
Jamaica	102.5
Kazakhstan	104.0
Iran	105.0
Armenia	106.5
Azerbaijan	106.5
Egypt	108.0
Jordan	110.0
Micronesia, Federated States of	110.0
Turkmenistan	110.0
Marshall Islands	112.0
Samoa	113.5
Solomon Islands	113.5

Iraq	115.0
Kosovo	116.0
Belize	117.0
Nicaragua	118.0
Vanuatu	119.0
Guyana	120.0
Uzbekistan	121.0
Indonesia	122.5
Ukraine	122.5
Guatemala	124.0
Fiji	125.0
Vietnam	126.0
Bhutan	127.0
Philippines	128.0
Gabon	129.0
Botswana	130.0
Honduras	131.0
Mongolia	132.0
Cabo Verde	133.0
Syria	134.0
Bolivia	135.0
Nauru	136.0
Kiribati	137.0
Moldova	138.0
India	139.0
South Africa	140.0
Kyrgyzstan	141.0
Bangladesh	142.0
Tuvalu	143.0
Papua New Guinea	145.0
Pakistan	145.0
Ghana	145.0
Angola	147.0
Namibia	148.0
Tajikistan	149.0
Korea, North	150.0
Laos	151.0
Cambodia	152.0
Yemen	153.0
Burma	154.0
Djibouti	155.0
Congo, Republic of the	156.0
Sao Tome and Principe	157.0
Sudan	158.0
Gambia, The	159.5
Nepal	159.5
Mauritania	161.0
Kenya	162.0
Senegal	163.0
Swaziland	164.0
Western Sahara	165.0
Benin	166.0
Cameroon	167.0
Madagascar	168.5

```
Tanzania                               168.5
Haiti                                  170.5
Nigeria                                170.5
Togo                                   172.5
Cote d'Ivoire                          172.5
Comoros                                174.0
Eritrea                                175.0
Lesotho                                176.0
Sierra Leone                           177.5
Ethiopia                               177.5
Rwanda                                 179.0
Chad                                   180.0
Guinea                                 181.0
Burkina Faso                           182.0
Malawi                                 183.0
Uganda                                 184.0
Zambia                                 185.0
South Sudan                            186.0
Mali                                   187.0
Burundi                                188.5
Liberia                                188.5
Afghanistan                            190.0
Niger                                  191.0
Mozambique                             192.0
Congo, Democratic Republic of the      193.5
Zimbabwe                               193.5
Guinea-Bissau                          195.0
Central African Republic               196.5
Somalia                                196.5
Name: Welfare Rank, Length: 197, dtype: float64
```

Countries like Singapore, Switzerland, Australia, Sweden and Canada do well in my ranking function.

Countries like Congo, Afganistan, Zimbabwe and Somalia do the worst in my ranking function.

I think my ranking function in general is pretty decent in terms of correctness. Surely countries like Switzerland, Australia, Sweden, Canada, USA and Singapore should be at the top of such a function. And they are rightly so.

Surely countries like Congo, Afganistan, Zimbabwe and Somalia should be at the bottom and they are rightly so.

There are however a few improvement that can be made with more data and/or time. For example, Monaco tops my list, but it is too small a country and hence one may not be happy there given it's limited options and people to interect with. Other than a few such cases and am pretty convinced with the ranking.

My Ranking Measure : takes into account the GDP (PPP) PER CAPITA of each country and the Life Expectancy of each country. I played around with some other fields as well, like Literacy Rate and Health Expenditure/GDP, but I got most convincing results (personally speaking) with just these two fields.

0.1.6 Task - 5

5) Set up a meaningful distance function to measure how similar/difference pairs of countries are. Produce a table showing the nearest and farthest neighbor to each nation on earth?
Write a short analysis of this table describing: (a) What kinds of similarities does your measure get right? (b) What are the most interesting/surprising pairs to fall out of this analysis? and (c) Where does it goof up?

My distance function takes into account the Literacy Rate, GDP, Life Expectancy, Population and Area into account to calculate the similarity/difference between two countries.

In [427]:
```
from scipy.spatial import distance
df_for_dist = df[['Literacy Rate (%)','GDP (PPP) in US $','Life Expectancy','Population','Area
df_for_dist = df_for_dist.convert_objects(convert_numeric=True)
pairwise_dists = distance.squareform(distance.pdist(df_for_dist))
pairwise_dists = distance.cdist(df_for_dist,df_for_dist)
```

In [428]:
```
count = df['Country Code'].count()
df_copy = df.reset_index()
df_copy = df_copy[['Country Name']]
def get_min_max(x):                       # My Distance Function
    pairwise_dists[x][x]=99999999999;
    min1 = np.argmin(pairwise_dists[x])
    pairwise_dists[x][x]=-1;
    max1 = np.argmax(pairwise_dists[x])
    return df_copy['Country Name'].iloc[min1],df_copy['Country Name'].iloc[max1]
df_copy['Nearest Neighbour'] = df_copy['Country Name']
df_copy['Farthest Neighbour'] = df_copy['Country Name']
for i in range (0, count+1):
    df_copy['Nearest Neighbour'].iloc[i],df_copy['Farthest Neighbour'].iloc[i] = get_min_max(i
pd.set_option('display.max_rows', len(df))
df_copy
```

Out[428]:

	Country Name	Nearest Neighbour \
0	Afghanistan	Mozambique
1	Albania	Macedonia
2	Algeria	Saudi Arabia
3	Andorra	Liechtenstein
4	Angola	Sudan
5	Antigua and Barbuda	Tonga
6	Argentina	Canada
7	Armenia	Estonia
8	Australia	Canada
9	Austria	Greece
10	Azerbaijan	Belarus
11	Bahamas, The	Fiji
12	Bahrain	Brunei
13	Bangladesh	Iraq
14	Barbados	Antigua and Barbuda
15	Belarus	Azerbaijan
16	Belgium	Netherlands
17	Belize	Cabo Verde
18	Benin	Guinea
19	Bhutan	Gambia, The
20	Bolivia	Kenya
21	Bosnia and Herzegovina	Croatia
22	Botswana	Namibia
23	Brazil	Indonesia
24	Brunei	Bahrain
25	Bulgaria	Serbia
26	Burkina Faso	Senegal
27	Burundi	Rwanda
28	Cabo Verde	Vanuatu
29	Cambodia	Ghana
30	Cameroon	Cote d'Ivoire
31	Canada	Australia

32	Central African Republic	Somalia
33	Chad	Mali
34	Chile	Sweden
35	China	Mexico
36	Colombia	Venezuela
37	Comoros	Sao Tome and Principe
38	Congo, Democratic Republic of the	Tanzania
39	Congo, Republic of the	Laos
40	Costa Rica	Panama
41	Cote d'Ivoire	Cameroon
42	Croatia	Bosnia and Herzegovina
43	Cuba	Czech Republic
44	Cyprus	Brunei
45	Czech Republic	Belgium
46	Denmark	Ireland
47	Djibouti	Comoros
48	Dominica	Saint Lucia
49	Dominican Republic	Sri Lanka
50	Ecuador	Tunisia
51	Egypt	Iran
52	El Salvador	Jamaica
53	Equatorial Guinea	Kosovo
54	Eritrea	Togo
55	Estonia	Latvia
56	Ethiopia	Tanzania
57	Fiji	Bahamas, The
58	Finland	Norway
59	France	Germany
60	Gabon	Botswana
61	Gambia, The	Djibouti
62	Georgia	Lithuania
63	Germany	France
64	Ghana	Syria
65	Greece	Portugal
66	Greenland	Mauritania
67	Grenada	Saint Vincent and the Grenadines
68	Guatemala	Syria
69	Guinea	Benin
70	Guinea-Bissau	Djibouti
71	Guyana	Suriname
72	Haiti	Togo
73	Honduras	Nicaragua
74	Hungary	Bulgaria
75	Iceland	Cyprus
76	India	Pakistan
77	Indonesia	Brazil
78	Iran	Egypt
79	Iraq	Syria
80	Ireland	Denmark
81	Israel	Singapore
82	Italy	United Kingdom
83	Jamaica	Mauritius
84	Japan	Italy
85	Jordan	Serbia

86	Kazakhstan	Ukraine
87	Kenya	Burma
88	Kiribati	Tuvalu
89	Korea, North	Nepal
90	Korea, South	United Kingdom
91	Kosovo	Fiji
92	Kuwait	Qatar
93	Kyrgyzstan	Tajikistan
94	Laos	Congo, Republic of the
95	Latvia	Estonia
96	Lebanon	Dominican Republic
97	Lesotho	Swaziland
98	Liberia	Eritrea
99	Libya	Paraguay
100	Liechtenstein	Andorra
101	Lithuania	Slovakia
102	Luxembourg	Andorra
103	Macedonia	Albania
104	Madagascar	Yemen
105	Malawi	Cambodia
106	Malaysia	Thailand
107	Maldives	Saint Vincent and the Grenadines
108	Mali	Niger
109	Malta	Saint Lucia
110	Marshall Islands	Palau
111	Mauritania	Central African Republic
112	Mauritius	Jamaica
113	Mexico	China
114	Micronesia, Federated States of	Palau
115	Moldova	Armenia
116	Monaco	San Marino
117	Mongolia	Turkmenistan
118	Montenegro	Maldives
119	Morocco	Algeria
120	Mozambique	Zambia
121	Burma	Kenya
122	Namibia	Botswana
123	Nauru	Tuvalu
124	Nepal	Korea, North
125	Netherlands	Belgium
126	New Zealand	Ireland
127	Nicaragua	Honduras
128	Niger	Mali
129	Nigeria	Ethiopia
130	Norway	Finland
131	Oman	Paraguay
132	Pakistan	India
133	Palau	Marshall Islands
134	Panama	Costa Rica
135	Papua New Guinea	Laos
136	Paraguay	Libya
137	Peru	Saudi Arabia
138	Philippines	Vietnam
139	Poland	United Kingdom

140	Portugal	Greece
141	Qatar	Kuwait
142	Romania	Malaysia
143	Russia	Ukraine
144	Rwanda	Burundi
145	Saint Kitts and Nevis	Saint Vincent and the Grenadines
146	Saint Lucia	Dominica
147	Saint Vincent and the Grenadines	Grenada
148	Samoa	Grenada
149	San Marino	Monaco
150	Sao Tome and Principe	Comoros
151	Saudi Arabia	Peru
152	Senegal	Burkina Faso
153	Serbia	Bulgaria
154	Seychelles	Palau
155	Sierra Leone	Togo
156	Singapore	Israel
157	Slovakia	Denmark
158	Slovenia	Lithuania
159	Solomon Islands	Vanuatu
160	Somalia	South Sudan
161	South Africa	Burma
162	South Sudan	Mali
163	Spain	France
164	Sri Lanka	Dominican Republic
165	Sudan	Tanzania
166	Suriname	Guyana
167	Swaziland	Lesotho
168	Sweden	Chile
169	Switzerland	Belgium
170	Syria	Ghana
171	Taiwan	Netherlands
172	Tajikistan	Kyrgyzstan
173	Tanzania	Sudan
174	Thailand	Turkey
175	Timor-Leste	Bhutan
176	Togo	Eritrea
177	Tonga	Antigua and Barbuda
178	Trinidad and Tobago	Montenegro
179	Tunisia	Ecuador
180	Turkey	Thailand
181	Turkmenistan	Belarus
182	Tuvalu	Nauru
183	Uganda	Cameroon
184	Ukraine	Uzbekistan
185	United Arab Emirates	Dominican Republic
186	United Kingdom	Italy
187	United States	Canada
188	Uruguay	Croatia
189	Uzbekistan	Ukraine
190	Vanuatu	Cabo Verde
191	Venezuela	Colombia
192	Vietnam	Philippines
193	Western Sahara	Kyrgyzstan

194	Yemen	Cameroon
195	Zambia	Mozambique
196	Zimbabwe	Malawi

	Farthest Neighbour
0	Monaco
1	India
2	Nauru
3	Russia
4	Monaco
5	India
6	Tuvalu
7	India
8	Sao Tome and Principe
9	Guinea-Bissau
10	Liechtenstein
11	United States
12	India
13	Nauru
14	India
15	Liechtenstein
16	Greenland
17	United States
18	United States
19	United States
20	Monaco
21	India
22	Monaco
23	Nauru
24	India
25	Liechtenstein
26	Monaco
27	United States
28	United States
29	Monaco
30	Monaco
31	Sao Tome and Principe
32	Japan
33	Monaco
34	Tuvalu
35	Nauru
36	Nauru
37	United States
38	Monaco
39	Japan
40	Nigeria
41	Monaco
42	Nigeria
43	Greenland
44	India
45	Greenland
46	Niger
47	United States
48	India

49	Tuvalu
50	Tuvalu
51	Nauru
52	Russia
53	United States
54	United States
55	India
56	Monaco
57	United States
58	Tuvalu
59	Sao Tome and Principe
60	Japan
61	United States
62	India
63	Sao Tome and Principe
64	Monaco
65	Guinea-Bissau
66	Japan
67	India
68	Nauru
69	Monaco
70	United States
71	United States
72	United States
73	United States
74	Greenland
75	Nigeria
76	Nauru
77	Nauru
78	Nauru
79	Nauru
80	Niger
81	Greenland
82	Guinea-Bissau
83	United States
84	Sao Tome and Principe
85	Liechtenstein
86	Liechtenstein
87	Monaco
88	India
89	Nauru
90	Greenland
91	United States
92	Congo, Democratic Republic of the
93	Liechtenstein
94	United States
95	India
96	Russia
97	United States
98	United States
99	Tuvalu
100	Russia
101	Nigeria
102	Russia

103	India
104	Monaco
105	Monaco
106	Tuvalu
107	India
108	Monaco
109	Russia
110	United States
111	Japan
112	Russia
113	Nauru
114	United States
115	India
116	India
117	Liechtenstein
118	India
119	Tuvalu
120	Monaco
121	Liechtenstein
122	Japan
123	India
124	Monaco
125	Greenland
126	Guinea-Bissau
127	United States
128	Monaco
129	Monaco
130	Tuvalu
131	Tuvalu
132	Nauru
133	United States
134	Nigeria
135	Monaco
136	Tuvalu
137	Nauru
138	Liechtenstein
139	Liechtenstein
140	Tuvalu
141	Niger
142	Liechtenstein
143	Liechtenstein
144	United States
145	India
146	Russia
147	India
148	India
149	India
150	United States
151	Nauru
152	Monaco
153	Greenland
154	United States
155	United States
156	Niger

157	Greenland
158	Nigeria
159	Russia
160	Monaco
161	Liechtenstein
162	Monaco
163	Tuvalu
164	Tuvalu
165	Monaco
166	United States
167	United States
168	Guinea-Bissau
169	Greenland
170	Monaco
171	Greenland
172	Liechtenstein
173	Monaco
174	Nauru
175	United States
176	United States
177	India
178	India
179	Tuvalu
180	Nauru
181	Liechtenstein
182	India
183	Monaco
184	Liechtenstein
185	Tuvalu
186	Sao Tome and Principe
187	Tuvalu
188	Nigeria
189	Liechtenstein
190	United States
191	Nauru
192	Nauru
193	India
194	Monaco
195	Monaco
196	Monaco

What kinds of similarities does your measure get right?

Among most developed countries my similarity feels right ! Like for USA I've got the nearest neightbour as Canada and Farthest as Tuvalu, which seems right ! Even among most under-developed countries my distance function feels right, for example, for Congo I've got nearest neighbour as Tanzania and farthest as Monaco (wich has Very high literacy rate and GDP)

What are the most interesting/surprising pairs to fall out of this analysis?

One of them is that the nearest neighbour to Mexico is China ! This a new to me, I would have not though of them being similar. But it turns out that they have similar most measures like Life Expectancy, Literacy rate, Mexico is 9th/10th most populated country, China is most. It does not seem too incorrect that these two are similar. Only the Area of both countries seems very differnt but I guess the other similarities outway this difference. And relitively speaking, in a world of 197 countries, mexico is the 14th largest, China being third. So yes, naturely that difference is not much.

Where does it goof up?

One place I can thing of as a slight Goof up is Maldives having its farthest Neighbour as India ! I think both a the size and population of both countries casued this. In that sense it is correct, but logically I would say they a moderately fat apart, but not the farthest ! In a distance function where the weightage to these two factors is lesser this would certainly not be the case. Also ot would be good if I could take care of outlier like Monaco that is very small but has very high GDP and Literacy rate and hence skews the results a bit. But I have added Area and Population so that it's effect it not that bad.

1 ˜THE END˜