Pandas

August 11, 2020

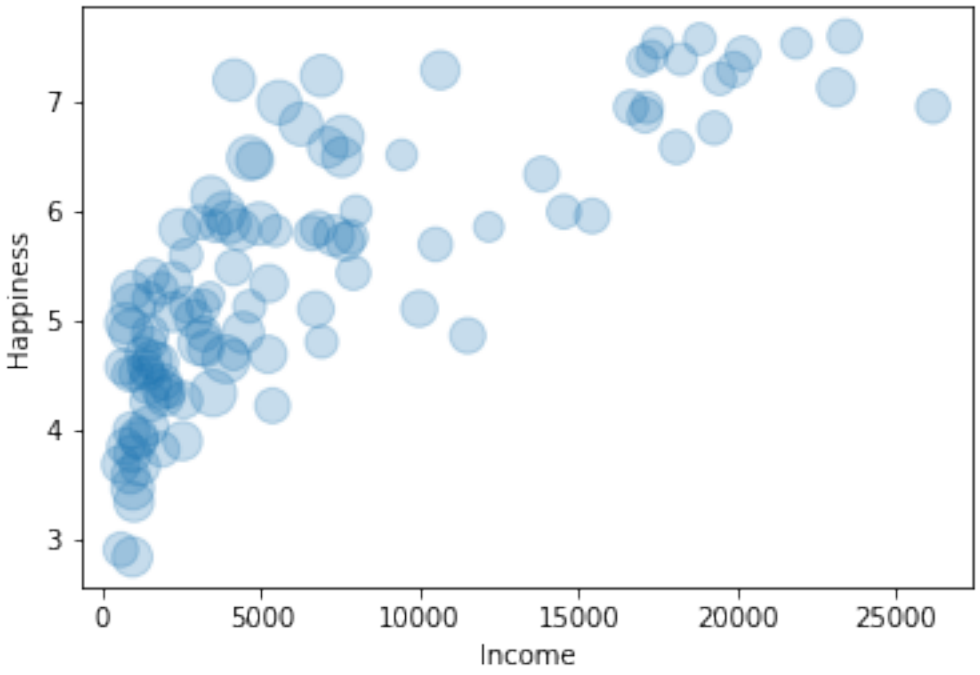
In [9]:
```
#pandas allows us to read in csv files and visualize them
import pandas as pd

data = pd.read_csv('happyscore_income.csv')

happy = data['happyScore']
income = data['avg_income']

print(happy, income)  #notice that the happiness increases as the income increases
```

```
0      4.350
1      4.033
2      6.574
3      7.200
4      7.284
5      5.212
6      4.694
7      6.937
8      3.587
9      4.218
10     2.905
11     3.340
12     5.890
13     6.983
14     4.332
15     5.813
16     7.427
17     7.587
18     6.670
19     4.252
20     5.140
21     6.477
22     7.226
23     5.689
24     6.505
25     6.750
26     4.369
27     7.527
```

```
28     4.885
29     5.975
       ...
81     5.073
82     5.194
83     5.791
84     5.102
85     5.878
86     5.124
87     5.123
88     5.716
89     3.465
90     7.364
91     5.848
92     5.995
93     4.507
94     3.904
95     6.130
96     3.667
97     2.839
98     6.455
99     4.786
100    4.739
101    5.332
102    3.781
103    4.681
104    3.931
105    7.119
106    6.485
107    5.360
108    4.642
109    5.129
110    4.610
Name: happyScore, Length: 111, dtype: float64 0       2096.760000
1       1448.880000
2       7101.120000
3      19457.040000
4      19917.000000
5       3381.600000
6       1265.340000
7      17168.505000
8        870.840000
9       5354.820000
10       572.880000
11       989.040000
12      3985.710000
13      5567.235000
14      3484.680000
```

```
15      5453.933333
16     20190.780000
17     23400.040000
18      7557.990000
19      1490.520000
20      2673.642857
21      4618.062857
22      6901.466667
23     10493.955000
24      9430.905000
25     19285.960000
26      1875.240000
27     17496.510000
28      4430.760000
29      3835.653333
           ...
81      2224.464000
82      1463.856000
83      6582.465882
84      9982.875000
85      4938.520000
86      3174.150000
87      4629.908571
88      7647.195000
89       946.520000
90     17032.755000
91     12174.765000
92      7986.396923
93       850.080000
94      1135.080000
95      3410.893333
96      1177.680000
97       936.360000
98      4792.500000
99      1497.030000
100     3251.280000
101     5242.666667
102      941.400000
103     4129.680000
104     1126.480000
105    23127.000000
106     7544.400000
107     2231.400000
108     3889.320000
109      956.760000
110     1768.560000
Name: avg_income, Length: 111, dtype: float64
```

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('happyscore_income.csv')

happy = data['happyScore']
income = data['avg_income']
inequality = data['income_inequality']

plt.xlabel("Income")
plt.ylabel("Happiness")
plt.scatter(income, happy, s=inequality*5, alpha=0.25)
plt.show()
```



In [18]:
```
#sorting data
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('happyscore_income.csv')
data.sort_values('avg_income', inplace=True)
#inplace=True is needed to actually sort the table, else it'll return the sorted valu
data
```

Out[18]:

	country	adjusted_satisfaction	avg_satisfaction	\
10	Burundi	25.0	2.9	

65	Madagascar	33.0	3.7
58	Liberia	37.0	4.4
72	Mozambique	34.0	3.8
73	Niger	34.0	3.8
93	Sierra Leone	27.0	3.5
8	Burkina Faso	37.0	4.4
66	Mali	32.0	4.7
59	Lesotho	40.0	5.2
74	Nigeria	43.0	5.3
97	Togo	22.0	2.6
102	Tanzania	19.0	2.5
89	Rwanda	40.0	4.9
109	Zambia	37.0	5.0
11	Benin	20.0	3.0
32	Ethiopia	33.0	4.2
104	Uganda	39.0	4.5
94	Senegal	41.0	4.5
96	Chad	40.0	5.4
6	Bangladesh	43.0	5.3
46	India	45.0	5.5
78	Nepal	48.0	5.3
1	Angola	26.0	4.3
82	Pakistan	45.0	6.0
19	Cameroon	36.0	3.9
56	Laos	55.0	6.2
51	Kenya	30.0	3.7
99	Tajikistan	46.0	5.1
43	Indonesia	50.0	6.1
38	Ghana	41.0	5.4
..	...	...	...
88	Russia	43.0	5.6
41	Croatia	47.0	6.0
30	Estonia	50.0	6.2
92	Slovakia	51.0	6.3
24	Czech Republic	54.0	6.6
84	Portugal	47.0	5.8
23	Cyprus	58.0	7.1
45	Israel	61.0	7.3
39	Greece	54.0	6.5
91	Slovenia	57.0	7.0
31	Spain	60.0	7.0
50	Japan	54.0	6.4
49	Italy	57.0	6.6
44	Ireland	64.0	7.5
90	Sweden	67.0	7.8
36	United Kingdom	60.0	7.1
7	Belgium	63.0	7.2
33	Finland	70.0	7.9

27	Denmark	74.0	8.4
34	France	52.0	6.4
76	Netherlands	69.0	7.6
48	Iceland	71.0	8.1
25	Germany	61.0	7.2
3	Austria	59.0	7.2
4	Australia	65.0	7.6
16	Canada	69.0	8.0
77	Norway	70.0	8.0
105	United States	62.0	7.3
17	Switzerland	70.0	8.0
61	Luxembourg	66.0	7.7

	std_satisfaction	avg_income	median_income	income_inequality	\
10	1.96	572.880000	436.920000	33.360000	
65	1.86	574.200000	415.480000	40.720000	
58	2.02	653.040000	528.720000	36.480000	
72	1.76	714.720000	488.520000	45.580000	
73	1.75	718.400000	535.560000	37.726667	
93	2.36	850.080000	669.360000	33.990000	
8	2.02	870.840000	630.240000	39.760000	
66	2.90	903.300000	695.340000	35.985000	
59	2.49	908.640000	532.920000	54.180000	
74	2.19	910.320000	649.200000	42.970000	
97	2.08	936.360000	636.000000	44.115000	
102	2.26	941.400000	693.300000	39.030000	
89	2.15	946.520000	549.440000	51.273333	
109	2.61	956.760000	510.060000	55.120000	
11	2.70	989.040000	657.000000	43.440000	
32	2.25	1050.720000	857.160000	33.170000	
104	1.89	1126.480000	780.160000	42.716667	
94	1.62	1135.080000	850.020000	39.755000	
96	2.61	1177.680000	876.600000	43.320000	
6	2.10	1265.340000	994.140000	32.665000	
46	2.13	1357.848000	1042.200000	34.918000	
78	1.65	1428.120000	1155.000000	32.840000	
1	3.19	1448.880000	1044.240000	42.720000	
82	2.72	1463.856000	1174.368000	31.196000	
19	1.69	1490.520000	1030.080000	42.820000	
56	1.72	1491.720000	1118.940000	37.265000	
51	2.14	1492.680000	949.080000	48.510000	
99	1.68	1497.030000	1260.690000	30.972500	
43	2.14	1541.747368	1161.066667	34.908421	
38	2.45	1577.040000	1148.280000	42.770000	
..	...	...	...	...	
88	2.43	7647.195000	5520.975000	41.240000	
41	2.48	7828.080000	6622.114286	32.008571	
30	2.19	7906.725000	6540.135000	32.527500	

92	2.21	7986.396923	7180.301538	27.056154
24	2.13	9430.905000	8363.370000	26.413750
84	2.16	9982.875000	7800.645000	36.630000
23	2.07	10493.955000	8624.295000	31.840000
45	2.09	10645.240000	8234.680000	41.940000
39	2.07	11507.565000	9776.475000	35.001250
91	2.14	12174.765000	11071.995000	24.678750
31	1.85	13842.990000	11782.395000	34.625000
50	2.04	14542.800000	12541.080000	32.110000
49	1.81	15437.595000	13163.070000	34.126250
44	1.85	16657.770000	13823.160000	32.418750
90	1.72	17032.755000	15166.605000	26.950000
36	1.98	17099.550000	14172.735000	34.432500
7	1.72	17168.505000	15166.455000	28.745000
33	1.53	17310.195000	14962.560000	27.723750
27	1.53	17496.510000	15630.885000	28.155000
34	2.15	18096.788571	14971.251429	32.255714
76	1.38	18234.435000	15880.545000	29.271250
48	1.64	18828.345000	16179.315000	28.780000
25	1.99	19285.960000	16291.260000	31.541667
3	2.11	19457.040000	16879.620000	30.296250
4	1.80	19917.000000	15846.060000	35.285000
16	1.71	20190.780000	16829.100000	33.790000
77	1.62	21877.710000	19477.620000	27.307500
105	1.92	23127.000000	17925.360000	41.090000
17	1.62	23400.040000	19442.920000	32.930000
61	1.76	26182.275000	22240.230000	31.950000

	region	happyScore	GDP	country.1
10	'Sub-Saharan Africa'	2.905	0.01530	Burundi
65	'Sub-Saharan Africa'	3.681	0.20824	Madagascar
58	'Sub-Saharan Africa'	4.571	0.07120	Liberia
72	'Sub-Saharan Africa'	4.971	0.08308	Mozambique
73	'Sub-Saharan Africa'	3.845	0.06940	Niger
93	'Sub-Saharan Africa'	4.507	0.33024	Sierra Leone
8	'Sub-Saharan Africa'	3.587	0.25812	Burkina Faso
66	'Sub-Saharan Africa'	3.995	0.26074	Mali
59	'Sub-Saharan Africa'	4.898	0.37545	Lesotho
74	'Sub-Saharan Africa'	5.268	0.65435	Nigeria
97	'Sub-Saharan Africa'	2.839	0.20868	Togo
102	'Sub-Saharan Africa'	3.781	0.28520	Tanzania
89	'Sub-Saharan Africa'	3.465	0.22208	Rwanda
109	'Sub-Saharan Africa'	5.129	0.47038	Zambia
11	'Sub-Saharan Africa'	3.340	0.28665	Benin
32	'Sub-Saharan Africa'	4.512	0.19073	Ethiopia
104	'Sub-Saharan Africa'	3.931	0.21102	Uganda
94	'Sub-Saharan Africa'	3.904	0.36498	Senegal
96	'Sub-Saharan Africa'	3.667	0.34193	Chad

```
6                      'Southern Asia'       4.694  0.39753      Bangladesh
46                     'Southern Asia'       4.565  0.64499           India
78                     'Southern Asia'       4.514  0.35997           Nepal
1                  'Sub-Saharan Africa'      4.033  0.75778          Angola
82                     'Southern Asia'       5.194  0.59543        Pakistan
19                 'Sub-Saharan Africa'      4.252  0.42250        Cameroon
56                  'Southeastern Asia'      4.876  0.59066            Laos
51                 'Sub-Saharan Africa'      4.419  0.36471           Kenya
99         'Central and Eastern Europe'      4.786  0.39047      Tajikistan
43                  'Southeastern Asia'      5.399  0.82827       Indonesia
38                 'Sub-Saharan Africa'      4.633  0.54558           Ghana
..                                   ...        ...      ...             ...
88         'Central and Eastern Europe'      5.716  1.13764          Russia
41         'Central and Eastern Europe'      5.759  1.08254         Croatia
30         'Central and Eastern Europe'      5.429  1.15174         Estonia
92         'Central and Eastern Europe'      5.995  1.16891        Slovakia
24         'Central and Eastern Europe'      6.505  1.17898  Czech Republic
84                     'Western Europe'      5.102  1.15991        Portugal
23                     'Western Europe'      5.689  1.20813          Cyprus
45   'Middle East and Northern Africa'       7.278  1.22857          Israel
39                     'Western Europe'      4.857  1.15406          Greece
91         'Central and Eastern Europe'      5.848  1.18498        Slovenia
31                     'Western Europe'      6.329  1.23011           Spain
50                      'Eastern Asia'       5.987  1.27074           Japan
49                     'Western Europe'      5.948  1.25114           Italy
44                     'Western Europe'      6.940  1.33596         Ireland
90                     'Western Europe'      7.364  1.33171          Sweden
36                     'Western Europe'      6.867  1.26637  United Kingdom
7                      'Western Europe'      6.937  1.30782         Belgium
33                     'Western Europe'      7.406  1.29025         Finland
27                     'Western Europe'      7.527  1.32548         Denmark
34                     'Western Europe'      6.575  1.27778          France
76                     'Western Europe'      7.378  1.32944     Netherlands
48                     'Western Europe'      7.561  1.30232         Iceland
25                     'Western Europe'      6.750  1.32792         Germany
3                      'Western Europe'      7.200  1.33723         Austria
4           'Australia and New Zealand'      7.284  1.33358       Australia
16                      'North America'      7.427  1.32629          Canada
77                     'Western Europe'      7.522  1.45900          Norway
105                     'North America'      7.119  1.39451   United States
17                     'Western Europe'      7.587  1.39651     Switzerland
61                     'Western Europe'      6.946  1.56391      Luxembourg

[111 rows x 11 columns]
```

In [22]:
```
#filtering data
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('happyscore_income.csv')
data.sort_values('avg_income', inplace=True)

#countries with avg income greater than 15k only
richest = data[data['avg_income'] > 15000]
richest  #the entire table

#richest.iloc[0] #first in the database
```

Out[22]:

	country	adjusted_satisfaction	avg_satisfaction	std_satisfaction	avg_income	median_income	income_inequality
49	Italy	57.0	6.6	1.81	15437.595000	13163.070000	34.126250
44	Ireland	64.0	7.5	1.85	16657.770000	13823.160000	32.418750
90	Sweden	67.0	7.8	1.72	17032.755000	15166.605000	26.950000
36	United Kingdom	60.0	7.1	1.98	17099.550000	14172.735000	34.432500
7	Belgium	63.0	7.2	1.72	17168.505000	15166.455000	28.745000
33	Finland	70.0	7.9	1.53	17310.195000	14962.560000	27.723750
27	Denmark	74.0	8.4	1.53	17496.510000	15630.885000	28.155000
34	France	52.0	6.4	2.15	18096.788571	14971.251429	32.255714
76	Netherlands	69.0	7.6	1.38	18234.435000	15880.545000	29.271250
48	Iceland	71.0	8.1	1.64	18828.345000	16179.315000	28.780000
25	Germany	61.0	7.2	1.99	19285.960000	16291.260000	31.541667
3	Austria	59.0	7.2	2.11	19457.040000	16879.620000	30.296250
4	Australia	65.0	7.6	1.80	19917.000000	15846.060000	35.285000
16	Canada	69.0	8.0	1.71	20190.780000	16829.100000	33.790000
77	Norway	70.0	8.0	1.62	21877.710000	19477.620000	27.307500
105	United States	62.0	7.3	1.92	23127.000000	17925.360000	41.090000
17	Switzerland	70.0	8.0	1.62	23400.040000	19442.920000	32.930000
61	Luxembourg	66.0	7.7				

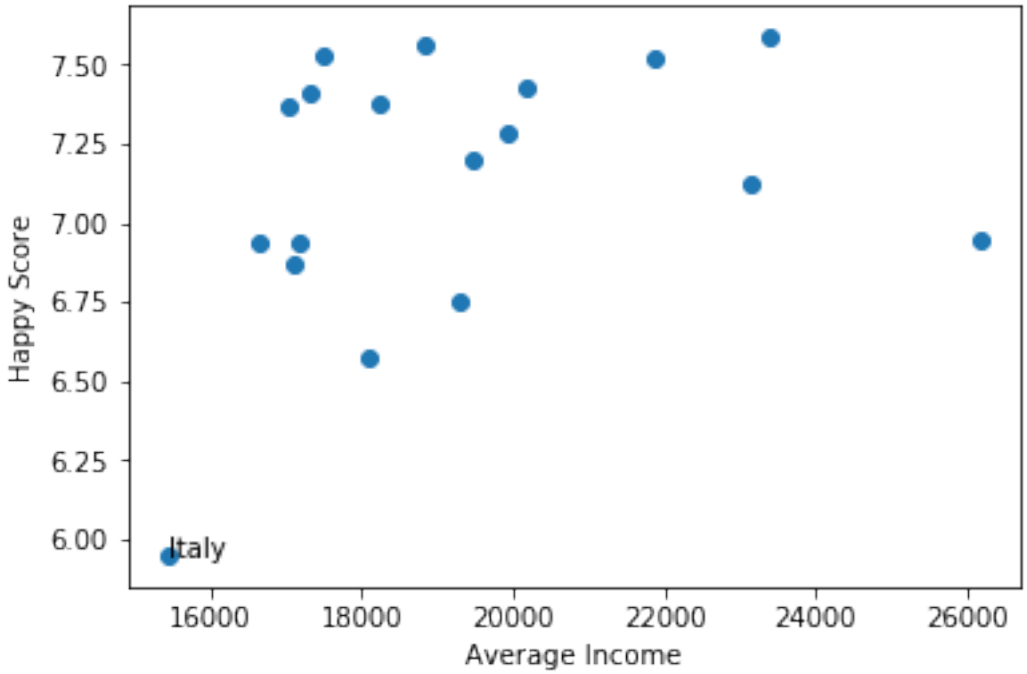
```
61                1.76  26182.275000   22240.230000          31.950000

                            region  happyScore      GDP       country.1
49               'Western Europe'       5.948  1.25114           Italy
44               'Western Europe'       6.940  1.33596         Ireland
90               'Western Europe'       7.364  1.33171          Sweden
36               'Western Europe'       6.867  1.26637  United Kingdom
7                'Western Europe'       6.937  1.30782         Belgium
33               'Western Europe'       7.406  1.29025         Finland
27               'Western Europe'       7.527  1.32548         Denmark
34               'Western Europe'       6.575  1.27778          France
76               'Western Europe'       7.378  1.32944     Netherlands
48               'Western Europe'       7.561  1.30232         Iceland
25               'Western Europe'       6.750  1.32792         Germany
3                'Western Europe'       7.200  1.33723         Austria
4    'Australia and New Zealand'       7.284  1.33358       Australia
16                'North America'       7.427  1.32629          Canada
77               'Western Europe'       7.522  1.45900          Norway
105               'North America'       7.119  1.39451   United States
17               'Western Europe'       7.587  1.39651     Switzerland
61               'Western Europe'       6.946  1.56391      Luxembourg
```

In [23]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('happyscore_income.csv')
data.sort_values('avg_income', inplace=True)

richest = data[data['avg_income'] > 15000]

#calculate average of the avg_incomes of the richest countries
np.mean(richest['avg_income'])
```

Out[23]:
```
19266.680753968256
```

In [30]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('happyscore_income.csv')
data.sort_values('avg_income', inplace=True)

richest = data[data['avg_income'] > 15000]
plt.xlabel('Average Income')
plt.ylabel('Happy Score')
plt.scatter(richest['avg_income'], richest['happyScore'])
```

```
#labels the point
plt.text(richest.iloc[0]['avg_income'], richest.iloc[0]['happyScore'], richest.iloc[0]

#plt.text(richest.iloc[-1]['avg_income'], richest.iloc[-1]['happyScore'], richest.iloc

plt.show()
```



In [36]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('happyscore_income.csv')
data.sort_values('avg_income', inplace=True)

richest = data[data['avg_income'] > 15000]
plt.xlabel('Average Income')
plt.ylabel('Happy Score')
plt.scatter(richest['avg_income'], richest['happyScore'])

#prints avg income of each country, from our richest sorted array
#each item in richest.iterrows() is a tuple, k refers to the index and row being the
# for k, row in richest.iterrows():
#     print(row['avg_income'])

for k, row in richest.iterrows():
```

```
        #x coord, y coord, text
        plt.text(row['avg_income'], row['happyScore'], row['country'])

plt.show()
```