Graph

March 7, 2019

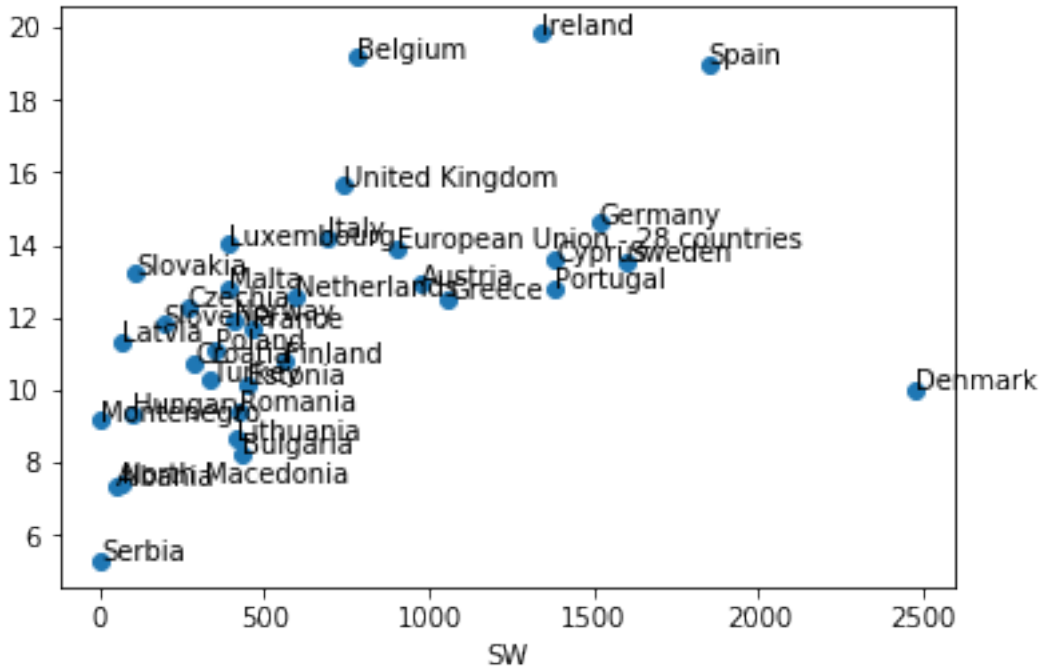
In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

renewable = pd.ExcelFile("renewable2.xls")

# read the info
wind_prod = pd.read_excel(renewable, "Data5", skiprows=10, index_col=None)
solar_prod_therm = pd.read_excel(renewable, "Data7", skiprows=10, index_col=None)
solar_prod_photo = pd.read_excel(renewable, "Data9", skiprows=10, index_col=None)
elec_consump = pd.read_excel("energy2.xls", skiprows=12, index_col=None)

#population table
pop = pd.read_excel("pop2.xls", skiprows=10, index_col=None)
pop = pop.drop(columns=["2009", "2010", "2011", "2012", "2013", "2014", "2015", "2017"
pop = pop.dropna()
pop = pop[pop['2016'] != ":"]

# house keeping
df1 = wind_prod.drop(columns=["2004", "2005", "2006", "2007", "2008", "2009", "2010",
df2 = solar_prod_therm.drop(columns=["2004", "2005", "2006", "2007", "2008", "2009", "
df3 = solar_prod_photo.drop(columns=["2004", "2005", "2006", "2007", "2008", "2009", "
df4 = elec_consump.drop(columns=['2013S2', '2014S1', '2014S2', '2015S1', '2015S2', '201
df4 = df4.rename(columns = {"2016S2": "2016"})
df4 = df4.dropna()
df4 = df4.drop(df4.index[[43]])

#conversion 1 euro = 1.0567 dollar in 2016
df4['2016'] = df4['2016'].apply(lambda x: x*1.0567*100)

#summing up green
df1['2016'] = df1["2016"] + df2["2016"] + df3["2016"]
df1 = df1.dropna()
df1 = df1.drop(df1.index[[]])

#merging tables
df5 = pd.merge(df4, df1, on='GEO/TIME')
```

```
df5 = df5.rename(columns = {"2016_y": "SW"})
df5 = df5.rename(columns = {"2016_x": "Electric Consumption"})
df5 = pd.merge(df5, pop, on='GEO/TIME')



#modifying thousand BOE ton -> kwh
df5['SW'] = df5["SW"]*11630000/df5['2016']
df5['GEO/TIME'] = df5['GEO/TIME'].replace({'Germany (until 1990 former territory of the

df5
```

Out[1]:

	GEO/TIME	Electric Consumption	SW	2016
0	European Union - 28 countries	13.885038	898.227	510181874
1	Belgium	19.179105	777.108	11311117
2	Bulgaria	8.252827	429.351	7153784
3	Czechia	12.278854	269.762	10553843
4	Denmark	10.017516	2470.99	5707251
5	Germany	14.603594	1515.01	82175684
6	Estonia	10.144320	451.61	1315944
7	Ireland	19.844826	1336.16	4726286
8	Greece	12.521895	1057.55	10783748
9	Spain	18.978332	1849	46440099
10	France	11.687102	461.18	66638391
11	Croatia	10.757206	289.732	4190669
12	Italy	14.180914	694.324	60665551
13	Cyprus	13.578595	1383.29	848319
14	Latvia	11.348958	64.9735	1968957
15	Lithuania	8.643806	416.312	2888558
16	Luxembourg	14.022409	389.517	576249
17	Hungary	9.362362	103.281	9830485
18	Malta	12.817771	392.474	450415
19	Netherlands	12.564163	591.805	16979120
20	Austria	12.912874	974.728	8700471
21	Poland	11.127051	350.855	37967209
22	Portugal	12.796637	1380.24	10341330
23	Romania	9.436331	425.871	19760585
24	Slovenia	11.803339	193.816	2064188
25	Slovakia	13.198183	111.236	5426252
26	Finland	10.778340	565.678	5487308
27	Sweden	13.536327	1599.11	9851017
28	United Kingdom	15.628593	740.04	65379044
29	Norway	11.940710	405.989	5210721
30	Montenegro	9.151022	3.73824	622218
31	North Macedonia	7.407467	64.0098	2071278
32	Albania	7.344065	51.7681	2875592
33	Serbia	5.283500	5.25919	7076372
34	Turkey	10.249990	332.456	78741053

In [2]:
```
ax = df5.set_index('SW')['Electric Consumption'].plot(style='o')
def label_point(x, y, val, ax):
    a = pd.concat({'x': x, 'y': y, 'val': val}, axis=1)
    for i, point in a.iterrows():
        ax.text(point['x'], point['y'], str(point['val']))

label_point(df5["SW"], df5["Electric Consumption"], df5["GEO/TIME"], ax)
```



In [156]:

In [ ]: