

Graph

March 13, 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", "Data7", 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", "2011", "2012", "2013", "2014", "2015", "2017"])
df2 = solar_prod_therm.drop(columns=["2004", "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2017"])
df3 = solar_prod_photo.drop(columns=["2004", "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2017"])
df4 = elec_consump.drop(columns=['2013S2', '2014S1', '2014S2', '2015S1', '2015S2', '2016S1', '2016S2'])
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 th
```

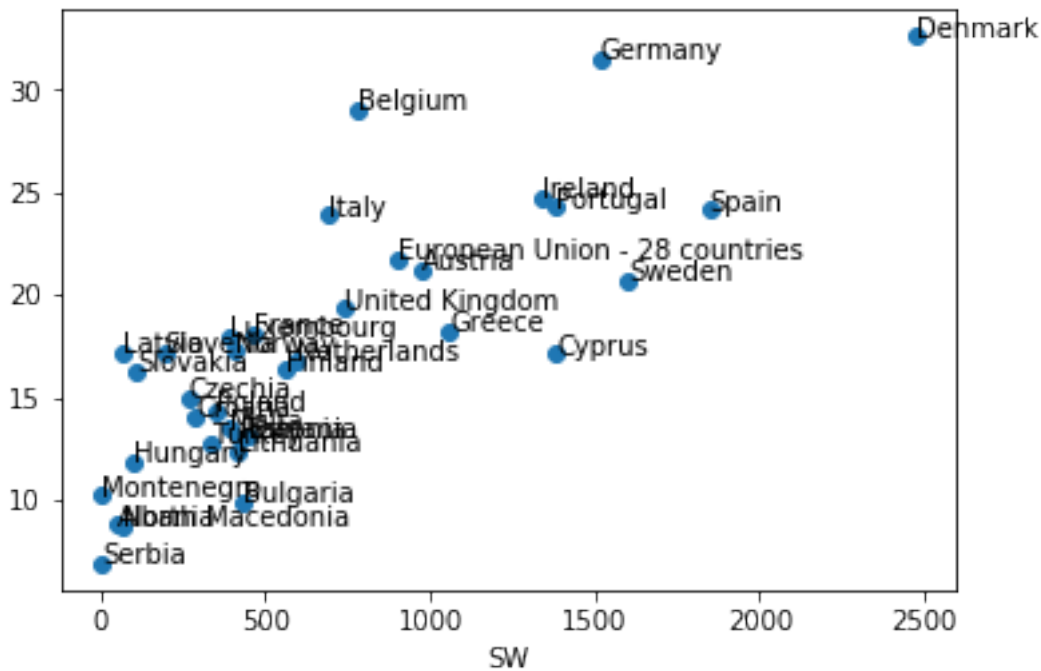
```
df5
```

```
Out[1]:
```

	GEO/TIME	Electric Consumption	SW	2016
0	European Union - 28 countries	21.694051	898.227	510181874
1	Belgium	29.006415	777.108	11311117
2	Bulgaria	9.911846	429.351	7153784
3	Czechia	15.015707	269.762	10553843
4	Denmark	32.588628	2470.99	5707251
5	Germany	31.457959	1515.01	82175684
6	Estonia	13.081946	451.61	1315944
7	Ireland	24.705646	1336.16	4726286
8	Greece	18.206941	1057.55	10783748
9	Spain	24.135028	1849	46440099
10	France	18.080137	461.18	66638391
11	Croatia	14.064677	289.732	4190669
12	Italy	23.891987	694.324	60665551
13	Cyprus	17.129107	1383.29	848319
14	Latvia	17.160808	64.9735	1968957
15	Lithuania	12.373957	416.312	2888558
16	Luxembourg	17.942766	389.517	576249
17	Hungary	11.887875	103.281	9830485
18	Malta	13.462358	392.474	450415
19	Netherlands	16.822664	591.805	16979120
20	Austria	21.239670	974.728	8700471
21	Poland	14.286584	350.855	37967209
22	Portugal	24.282966	1380.24	10341330
23	Romania	13.029111	425.871	19760585
24	Slovenia	17.213643	193.816	2064188
25	Slovakia	16.241479	111.236	5426252
26	Finland	16.326015	565.678	5487308
27	Sweden	20.732454	1599.11	9851017
28	United Kingdom	19.348177	740.04	65379044
29	Norway	17.234777	405.989	5210721
30	Montenegro	10.249990	3.73824	622218
31	North Macedonia	8.749476	64.0098	2071278
32	Albania	8.823445	51.7681	2875592
33	Serbia	6.910818	5.25919	7076372
34	Turkey	12.733235	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 [3]: #df5.to_excel("output.xlsx")
```

```
In [4]: ppp_adj = pd.read_excel("ppp.xls", skiprows=9, index_col=None)
```

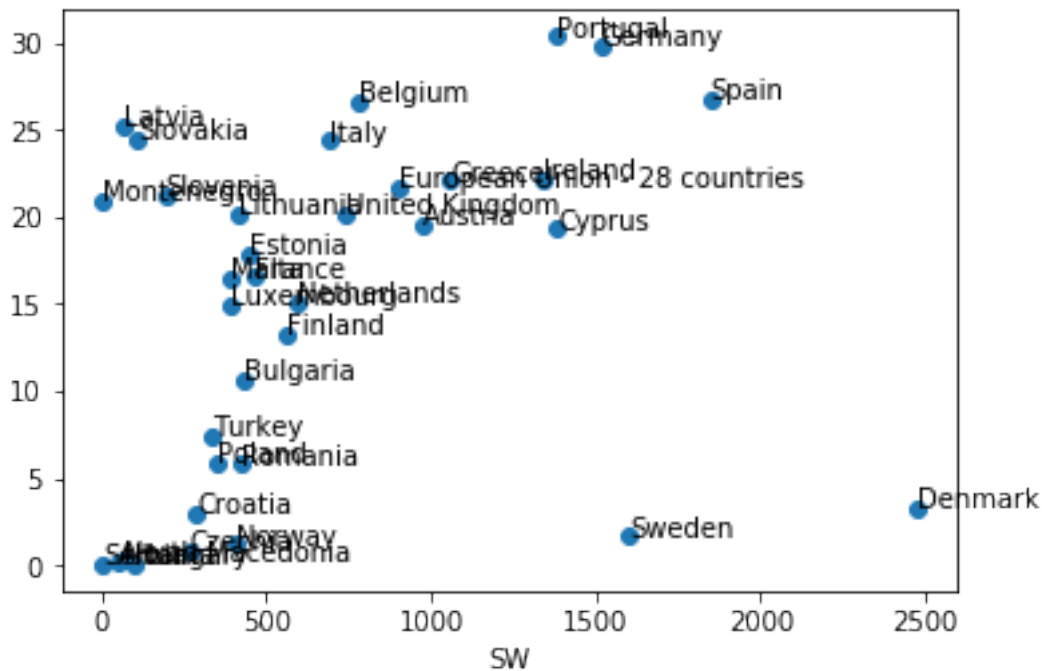
```
ppp_adj['GEO/TIME'] = ppp_adj['GEO/TIME'].replace({'Germany (until 1990 former territor'
ppp_adj = ppp_adj.filter(items=['GEO/TIME', '2016'])
ppp_adj = ppp_adj.dropna()
```

```
df6 = pd.merge(df5, ppp_adj, on='GEO/TIME')
#df6 = df6.drop([17])
```

```
#print(df6)
```

```
df6['Electric Consumption'] = df6['Electric Consumption'] / df6['2016_y'] #2016 y is t
df6 = df6.filter(items=['GEO/TIME', 'Electric Consumption', 'SW'])
```

```
ax2 = df6.set_index('SW')['Electric Consumption'].plot(style='o')
label_point(df6["SW"], df6["Electric Consumption"], df6["GEO/TIME"], ax2)
```



In []:

In []: