Exploring the population growth, energy consumption and GDP development of the World and building a forecasting model for population growth of the World

Ngoc Luu

### **Introduction**

The project will survey the data collected by The World Bank about the population, GDP and Energy Consumption, which we will call ‘features’. These features are collected from each country, however, the data also categorizes these countries into regions, continents, different parts of Asia, etc.

Each feature was downloaded into respectively named csv files: population, energy and gdp. These files were then imported into respectively named panda DataFrame. While importing, the first 4 rows will be omitted, country names will be used as index and missing values will be replaced with NaN. Furthermore, we will also drop columns named: Country Code, Indicator Name, Indicator Code from our dataframes.

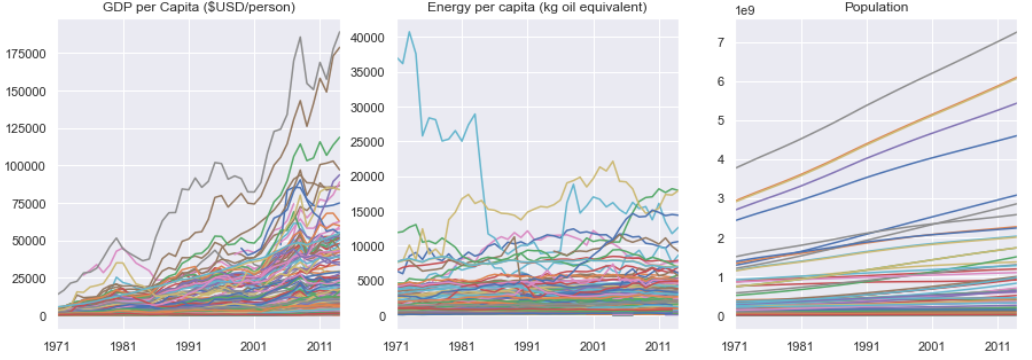
The final dataframe will have 264 rows listing country names, regions, territories, etc. Columns are the years 1960 - 2019 through which the data became available from the World Bank. However, throughout this time period, not all features have the available data. Thus, for this report, we will shorten the period from 1971 - 2014.

The project will perform some observation of the data about the development of the world in this time period. Finally, the project will attempt to build a forecasting model for future development based on its history.

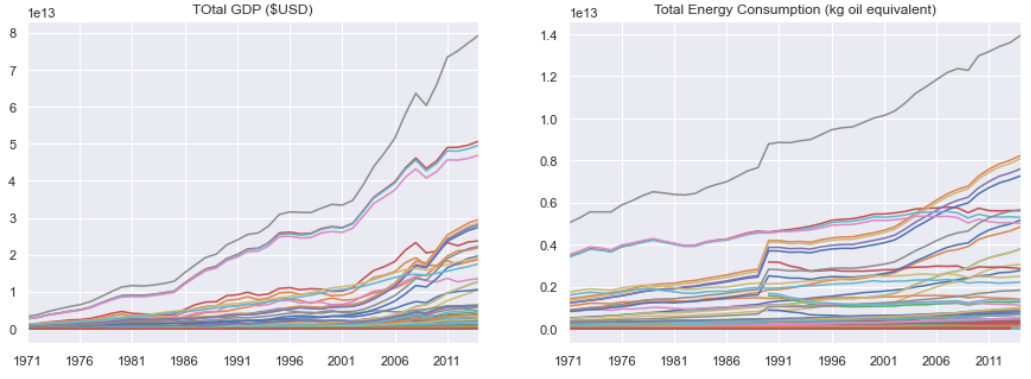
|  |
| --- |
| *#Importing data*  population = pd.read\_csv('population.csv', skiprows=4, index\_col=0, na\_values = ' ')  energy = pd.read\_csv('energy.csv', skiprows=4, index\_col=0, na\_values=' ')  gdp = pd.read\_csv('gdp.csv', skiprows=3, index\_col=0, na\_values = ' ')  *#Cleaning up unneeded columns*  population = population.drop(['Country Code', 'Indicator Name', 'Indicator Code'], axis=1)  energy = energy.drop(['Country Code', 'Indicator Name', 'Indicator Code'], axis=1)  gdp = gdp.drop(['Country Code', 'Indicator Name', 'Indicator Code'], axis=1)  gdp = gdp.drop(['Unnamed: 64'], axis=1)  *#shorten the period to 1971 - 2014*  gdp = gdp.drop(gdp.loc[:, '1960':'1970'].columns, axis =1)  gdp = gdp.drop(gdp.loc[:, '2015':'2019'].columns, axis =1)  population = population.drop(population.loc[:, '1960':'1970'].columns, axis =1)  population = population.drop(population.loc[:, '2015':'2019'].columns, axis =1)  energy = energy.drop(energy.loc[:, '1960':'1970'].columns, axis =1)  energy = energy.drop(energy.loc[:, '2015':'2019'].columns, axis =1) |

### **Data Exploratory**

Let us first visualize our time series data:



Besides population plot, GDP and Energy Consumption were converted to per person. Whole country values are plotted below:

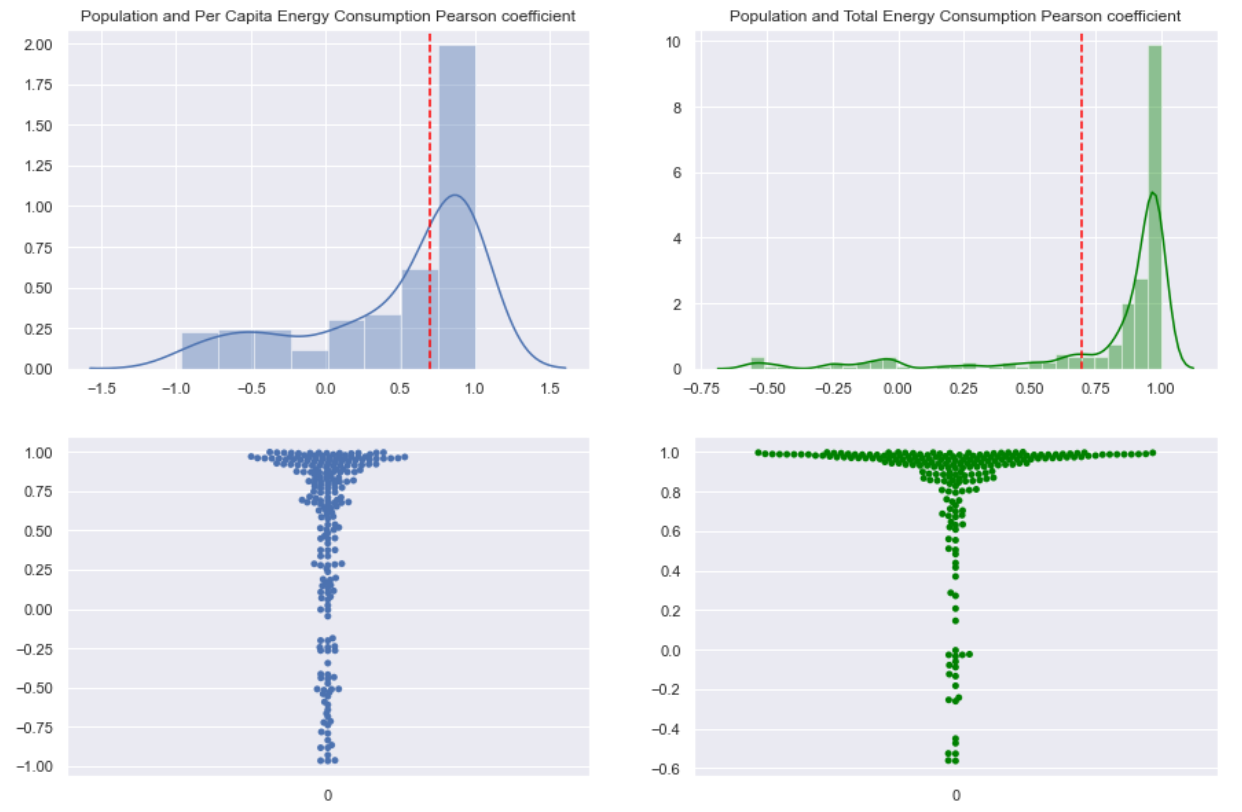


It seems using per person values does not benefit us in doing analysis on the data as it does not give us a clear direction.

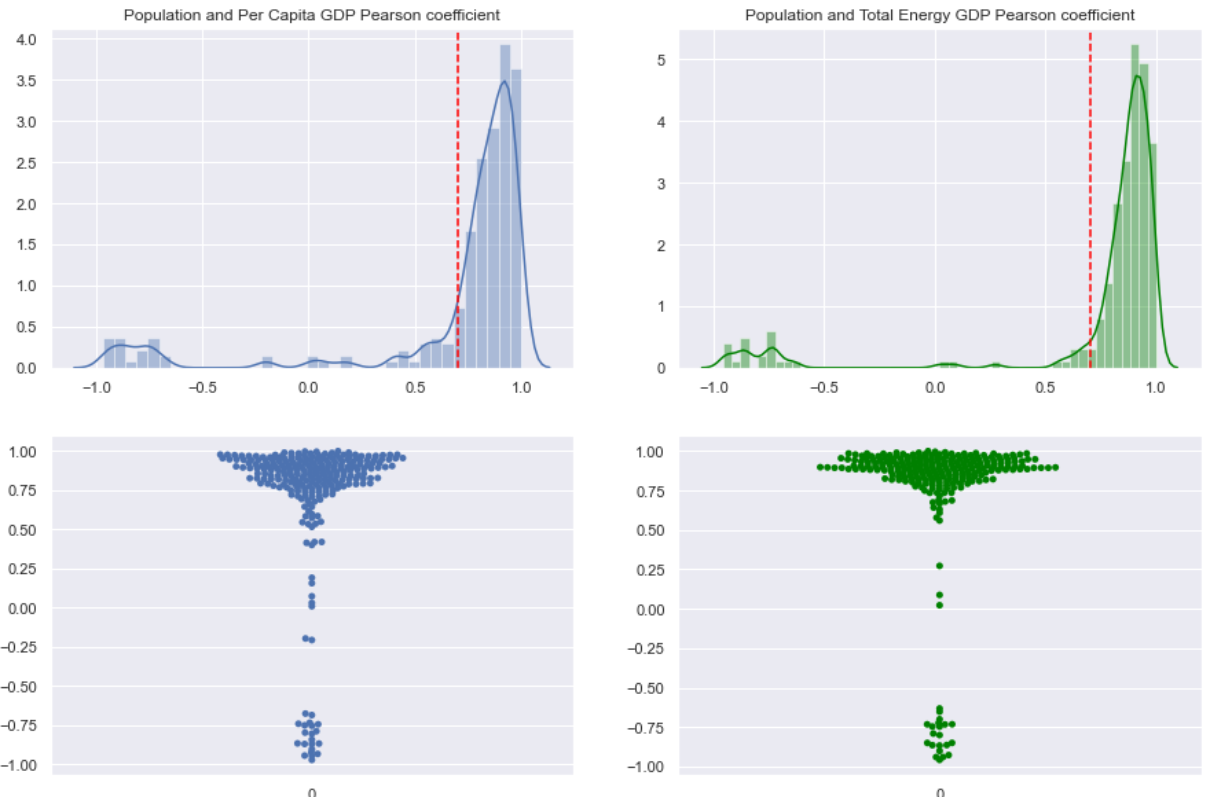
The first observation after plotting these time series data was that overall, countries all have a positive development of their features (population, gdp, energy consumption). However, as observed in the per person plots, it does not always translate to the improvement of their citizen’s quality of life. It may be due to the development of GDP and energy infrastructure could not keep up with the explosion of the population. As a result, a country may have a total increase in GDP and energy consumption but per person actually has consumed lower energy and lower GDP.

A quick check by comparing the percentage of Pearson correlation coefficients that are above the 0.7 (indicating a strong positive correlation) between total values and per capita values. For Energy Consumption, it was only 43% of the rows (out of 264 rows) has a strong positive correlation when performing the correlation using per capita time series, while it was 66% if total energy consumption values were used. For GDP, it was closer to 79% against 85%, but the distribution was different observing the respective swarmplots.

The Pearson CC between Energy Consumption and Population



The Pearson CC between GDP and Population

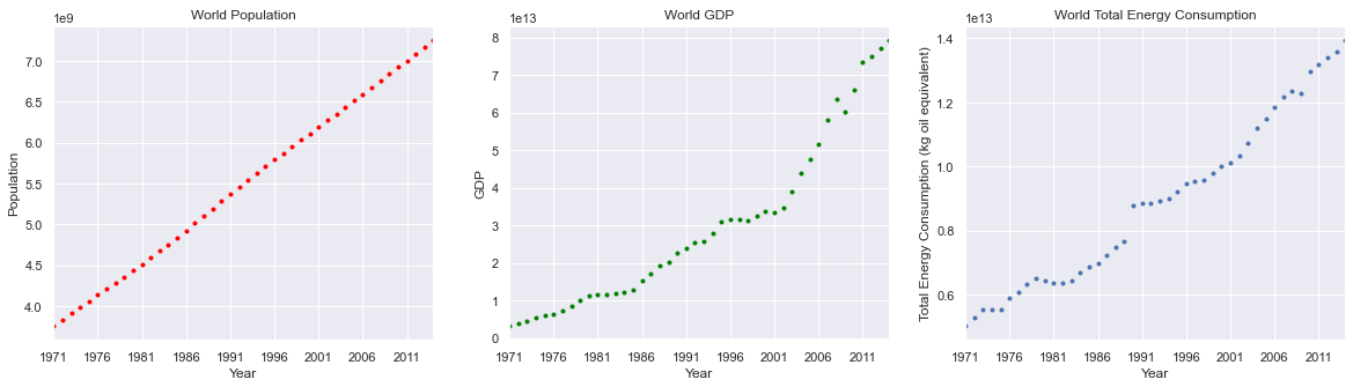


### **Build a forecasting model of population using scikit and Prophet**

Let us focus now only on the data of World. We will extract World data from the list, energy consumption will be converted total value:

|  |
| --- |
| w\_pop = population.loc['World']  w\_gdp = gdp.loc['World']  w\_eng = energy.loc['World']\*w\_pop |

Plotting the World data:



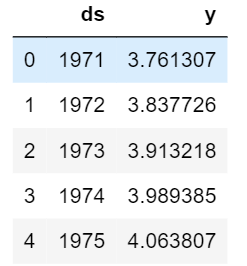
It appeared that population growth is almost linear compared to GDP and energy consumption. Thus, any forecast model built based on population growth may present the least uncertainty.

Python provides a few tools to build forecasting models. Here, we will use scikit-learn and FB prophet. There are pros and cons on either approach. Here, we will use scikit-learn to find the best fitted line through our observation of population growth. Future prediction will be made using the slope and intercept of the best fitted line.

We will create a new dataframe with population growth data to satisfy the requirements to be used with prophet. Prophet requires that date to be stored in ‘ds’ column, while values to be stored in the ‘y’ column, here we will also convert population to billions:

|  |
| --- |
| pop = w\_pop  pop = pop.reset\_index()  pop = pop.rename(columns={'index':'ds', 'World':'y'})  pop['y'] = pop['y']/1e9 |

The result is a new dataframe:



### **Scikit-learn**

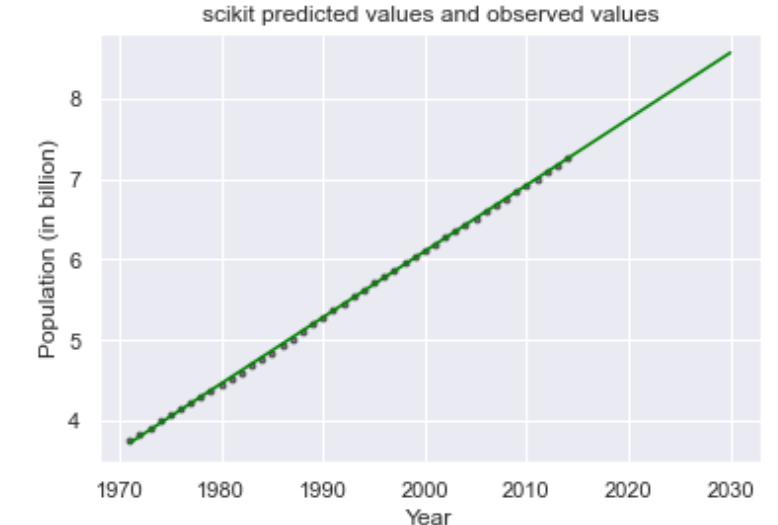
|  |
| --- |
| from sklearn.linear\_model import LinearRegression  *# This creates a LinearRegression object*  X = pop.drop('y', axis=1)  lm = LinearRegression() |

|  |
| --- |
| lm.fit(X, pop.y) |

After fitting our line through the observed data, scikit provides predict() method to obtain the future prediction. Here we will make 16 years into the future (from 2014)

|  |
| --- |
| sk\_predict = pd.DataFrame(lm.predict(np.arange(1971, 2031, 1).reshape(-1,1)))  sk\_predict = sk\_predict.set\_index(np.arange(1971, 2031, 1))  sk\_predict = sk\_predict.rename(columns={0:'sk\_predict'}) |

Plotting our predicted values and compared them with the observed data:

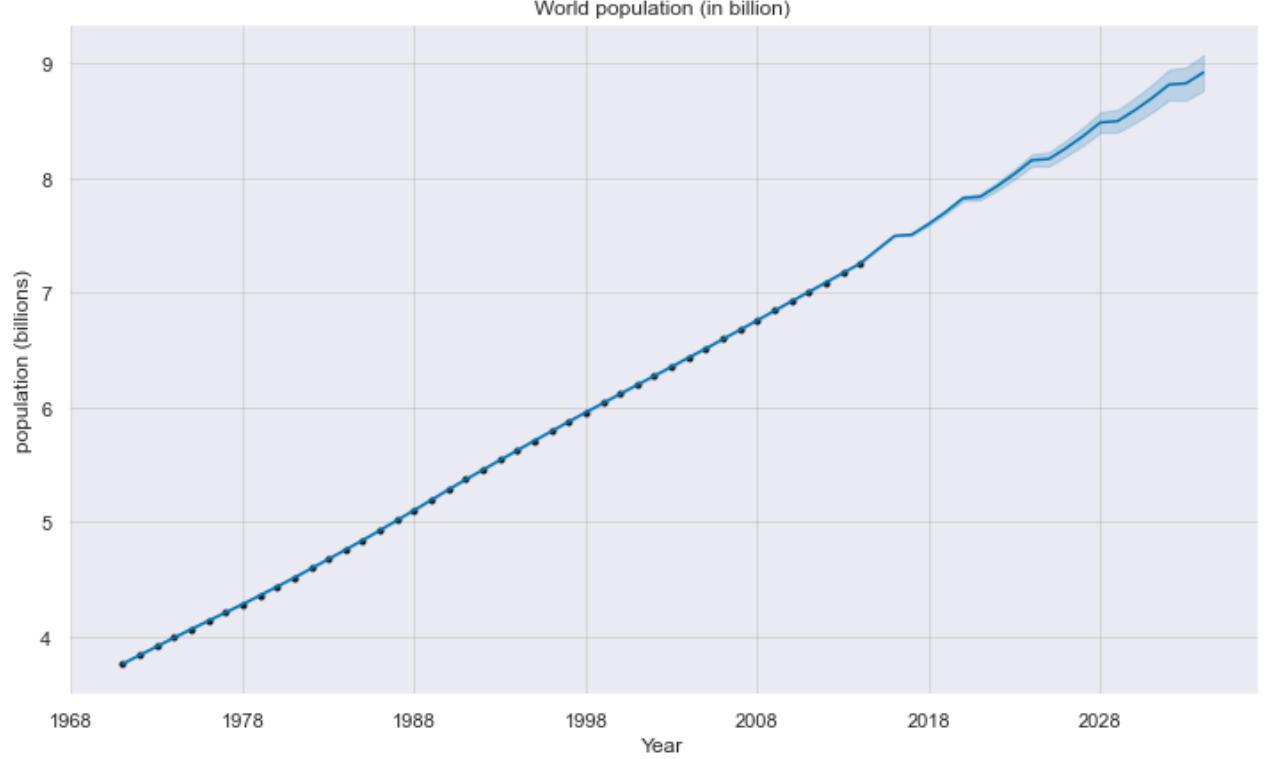


Here, after finding the best fitted line through the observed data, scikit built a new dataframe with all new values calculated based on the slope and intercept of the best fitted line. Thus, in the code above, we had to include the years from 1971 and 16 years into the future (2014 + 16).

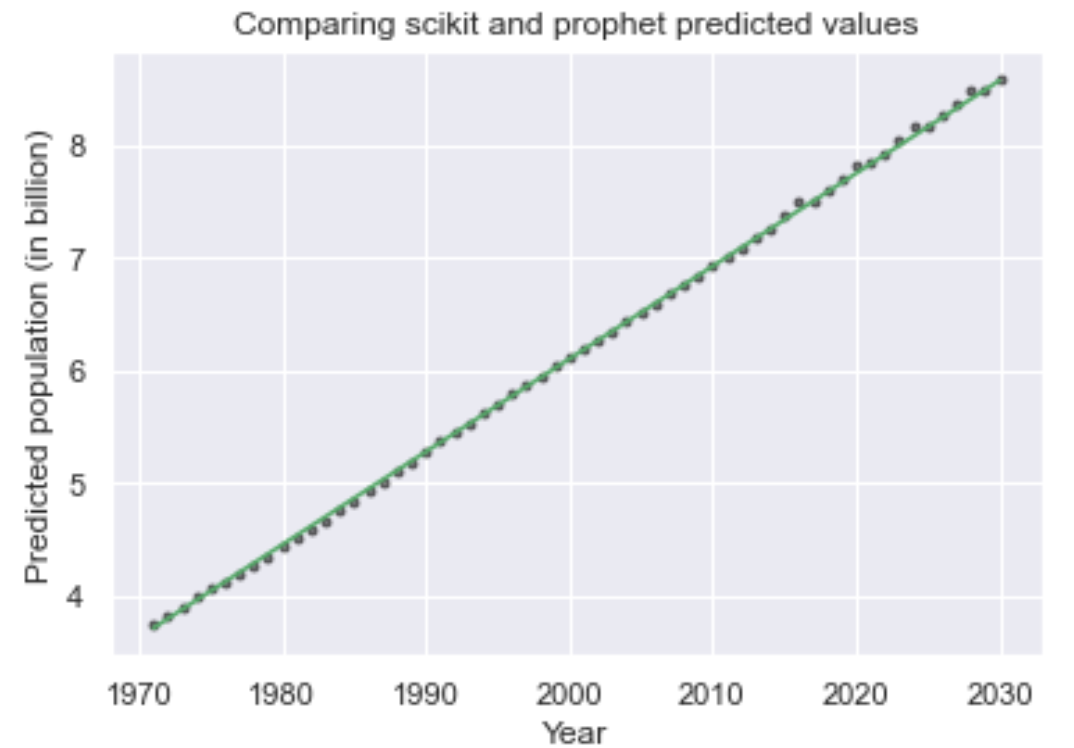
### **FB prophet**

|  |
| --- |
| pop\_prophet = fbprophet.Prophet(changepoint\_prior\_scale=0.15)  pop\_prophet.fit(pop)  pop\_forecast = pop\_prophet.make\_future\_dataframe(periods=17, freq='Y')  pop\_forecast = pop\_prophet.predict(pop\_forecast) |

With prophet, we will also use a linear relationship to build the model.

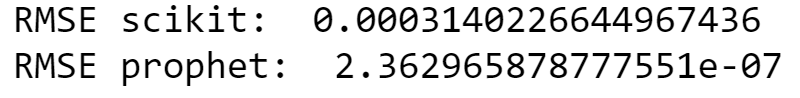


We can compare the two approaches by plotting them on the same graph:



They appear to be very close to each other. We can better gauge their performance by calculating the Root Mean Square Error:

|  |
| --- |
| print('RMSE scikit: ', mean\_squared\_error(w\_pop/1e9, sk\_predict.loc['1971':'2014']))  print('RMSE prophet: ', mean\_squared\_error(w\_pop/1e9, fbprophet\_.loc['1971':'2014'])) |



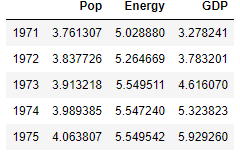
Prophet produces the least RMSE compared to scikit-learn. Thus, if we have to choose, prophet method may be better.

### **Building forecasting model for GDP and energy consumption**

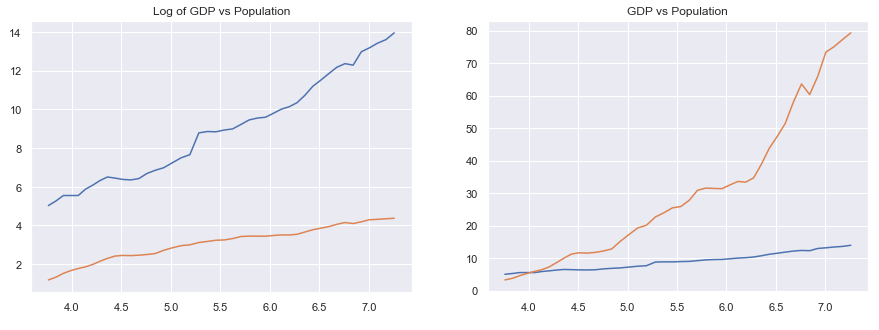
Let consolidate all the World info into a dataframe:

|  |
| --- |
| w = w\_pop/1e9  w = pd.DataFrame(w)  w = w.rename(columns={0:'Pop'})  w['Energy'] = w\_eng/1e12  w['GDP'] = w\_gdp/1e12 |

We will get:



Let us plotting Energy and GDP against population growth:



The plot on the left used the natural log of GDP which produces a relatively more linear than GDP vs population itself. Thus, we will convert the GDP column of our new dataframe to natural log:

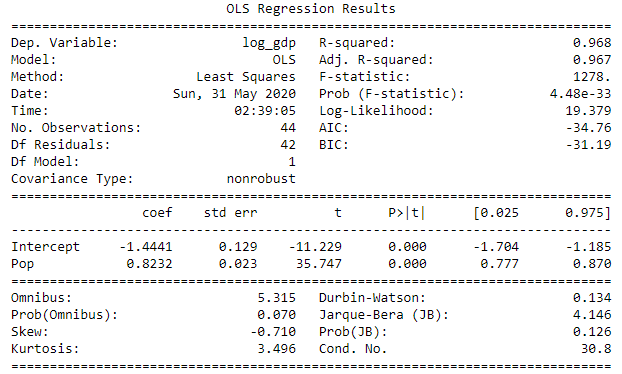
|  |
| --- |
| w['log\_gdp'] = np.log(w.GDP) |

Finding the relationship between GDP and Energy consumption against population:

|  |
| --- |
| *#importing the library*  import statsmodels.api as sm  from statsmodels.formula.api import ols |

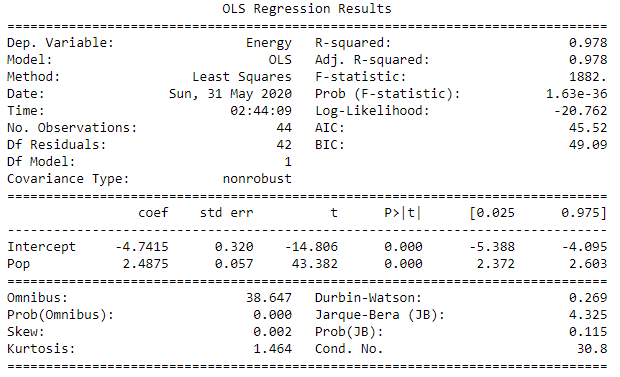
Finding the relationship between log(GDP) against population:

|  |
| --- |
| m\_log\_gdp = ols('log\_gdp ~ Pop', w).fit()  print(m\_log\_gdp.summary()) |



Doing similarly for energy consumption:

|  |
| --- |
| m\_eng = ols('Energy ~ Pop', w).fit()  print(m\_eng.summary()) |



Here we found the slopes and intercepts for both relationships. We will not conduct the calculation here but after finding the relationship between GDP with population growth (and Energy consumption with population) we can use the predicted values for population to find the predicted values for energy consumption and GDP in the future.

### **Conclusion**

The project explored the data collected from the World Bank. The analysis showed that most countries overall have positive trends in population growth, GDP development and total energy consumption. However, it may be due to the population growth that ‘pull’ the a country’s GDP and total energy consumption up, because the plot of per person (GDP and energy consumption) trends are not prevalent, some countries even have a downward trend in personal GDP or energy consumption.

We then built a forecasting model of the World population using two methods: scikit-learn and FB prophet. Although both methods produce predicted values very close to each other, FBprophet performed better evidenced by the lower RMSE compared to scikit.

We then found the relationship between total energy consumption of the World and its population based on the observations, and the same for the natural log of GDP and population. It was not done here but theoretically, we can also calculate the future values for GDP and energy consumption based on the future values of the predicted population.