CSC110 Final Project: The Relationship between Emission and Non-Emission Power Plants on Carbon Emissions

Raazia Hashim, Shilin Zhang, Kenneth Miura

Monday, December 14, 2020

1 Introduction

The United Nations has recognized climate change to be an imminent threat to humankind. In response, the Paris Agreement, signed in 2015 by a total of 197 countries around the world aims to limit the global temperature rise below 2 degree celsius, while aiming to limit the increase under 1.5 degree celsius (United Nations). Naturally, a large focus of these efforts is on the global energy sector, as 25% of 2010 global greenhouse gas emissions were a result of electricity and heat production.(EPA, 2020) With international cooperation and commitment to systematic change, electricity generation from renewable sources, like wind, solar, hydro and other renewable energy sources has increased significantly since then, as seen in the over 7% increase worldwide seen in 2018 (IEA, 2019). According to the IEA, the increased use of renewables in 2018 had a large impact on CO_2 emissions, avoiding 215 megatons of emissions, the vast majority caused by the transition to renewables in the power sector (IEA, 2019). The carbon footprint of renewable energy sources like solar, wind and nuclear are much lower than coal or gas, even after accounting for the emissions caused during the manufacturing and construction process. (Evans, 2017) Research shows that often hidden emissions in building wind turbines, solar panels and nuclear plants are very low, especially when they are compared to the emissions from burning fossil fuels (Evans, 2017). Additionally, nuclear power plants produce no greenhouse gas emissions in their operation and over the course of its life-cycle, nuclear produces about the same amount of carbon emissions per unit of electricity as wind, and one-third of the emissions per unit of electricity compared to solar (world nuclear org). However despite this, the increase in renewable energy sources has not been fast enough to keep pace with the rapid increase in the demand for electricity globally, which has led to an increase in generation from fossil fuel power plants. (IEA, 2019) Therefore, renewables have failed to displace fossil fuels in the energy sector and carbon dioxide emissions related to energy continue to rise, reaching 33.1 billion tonnes in 2018, and have increased by more than 40% since the year 2000. (world nuclear org) In 2018, nonrenewable energy sources like coal, natural gas, and petroleum account for about 63% of total U.S. electricity generation, but accounted for 99% of U.S. CO_2 emissions related to their energy sector (U.S. Energy Information Administration, 2020).
Our goal is to use current powerplant and carbon emission data to predict how the number of emission and non-emission powerplants in a country impact their carbon emissions. The research question we will be exploring in our project is,**"How does the number of Emissions and Non-Emissions powerplants per capita in a country predict their Carbon Emissions per capita?"** For our purposes, emission powerplants will be defined as powerplants that use non-renewable energy sources, with the exception of nuclear energy and non-emission powerplants as powerplants that run on renewable energy sources, with the addition of nuclear energy. Additionally, we will be exploring how the number of nuclear powerplants per capita predicts carbon emissions, to see if a major global shift towards nuclear energy would accomplish the goal of limiting carbon emissions and subsequently the rise in global temperatures.

2 Dataset Description

We are using three datasets in our project. One dataset is called the carbon emission dataset that contains carbon emissions for 207 countries in the world for the last 60 to 70 years, the second dataset is called the global powerplant dataset that contains information about approximately 30000 power plants around the world and the last one is called countries of the world that contains information about 227 countries around the world such as population and area. All the information from the third dataset is in 2017. The carbon emission dataset is obtained from a website organization called "Our World in Data" as a csv file. This website organization is trusted by lots of media such as the BBC and their data are also used in teaching by lots of universities including Harvard and Stanford. This

dataset contains 38 variables that include country name, year of the data, total carbon emission in the given year and the amount of carbon emission by burning coal, oil and gas. Another point to mention here is that the unit of carbon emission in this dataset is measured in million tonnes. We will be using the carbon emission variables and the country name variables in our project. The power plant dataset is obtained from a website organization called "World Resources Institute" as a csv file. There are approximately 29910 power plants from 164 countries around the world. We can observe the country the power plant is located, the type of fuel it uses, name of the power plant and much more variable from the dataset. The variables that we use in our project are the primary fuel variable, the country name variable, the name of the power plant, the longitude and the latitude of power plants. The third data set is from a website called Kaggle as a csv file and it contains population, area, GDP and much more variables about 227 countries around the world. These data are updated in 2017. We are using the country name and the population variables from this data set.

3 Computational Overview

3.1 Data processing

All data processing is done in `data_processing.py`. First, we need to find out what countries to appear in all three data sets. So I create three functions called `read_powerplant_data`, `read_co2_data` and `read_pop_data` that read all the information we need from each data set. The `read_powerplant_data` read in the country, name of the powerplant, type of the powerplant and the longitude and latitude of the powerplant and return it as a dictionary. A thing that we notice is that in the power plant data set, the country name is the United States of America but the country name is the United States in the population data set and carbon emission data set so every time the country name is United States of America, we will replace it with the United States.

The `read_co2_data` reads the country name and carbon emission of the country if it has available data in 2017 (since the population dataset only includes data up to 2017) and returns it as a dictionary. The `read_pop_data` read the country name and the population of the country and return it as a dictionary. A problem here is that there is an extra space after each country name so I have to take off the last character of the country using string operations. In order for a country to be considered in our project, it needs to exist in all three dictionaries so we create a `common_country` function that finds out all the countries that are in all three dictionaries and return it as a set. There are five outputs that we need each correspond to one function in the data processing file.

The first function is `read_powerplant_file` that returns the country name and the corresponding number of emission power plants per capita and the number of non-emission power plants per capita. We consider Oil, Gas, Petcoke, Coal, Storage, Cogeneration as emission power plants and the others are non-emission power plants. There is a 'other' type in the power plant data set and we are ignoring these power plants since we don't know what type they are. Then, I will use the `read_powerplant_data` function to get the country name and the type of power plant from the power plant data set. Then, we count the number of emission power plants and non-emission power plants for each country. Then, I will pick out the country that is in all three data sets and have available carbon emission in 2017. I did it by comparing if it is in the output of `common_country`. If the country is in the set, then it is considered in our project. Then, I will read the population data set and pick out values for the country in a set that is produced by `common_country`. Since I know that the country name is ordered in alphabetical order in the data set, then the list after picking out countries will still be ordered and since they are all countries in the set and we know that every data set have the country, so the order of the two lists, that is population and powerplant, must be the same and the country at any index for the two list refer to the same country. We also know that the length of these two lists must be the same as the length of the set from `common_country`. This means that the number of emission power plants per capita and the number of non-emission power plants per capita for a country can be found by dividing the number of emission and non-emission power plants at the index of that country by the population at the same index. Then, we stored the number of emission power plants per capita and the number of non-emission power plants per capita as a tuple and returned the list of two lists. First list contains the country name and the second list contains the number of emission power plants per capita and the number of non-emission power plants per capita as a tuple.

The second information we need to find is the carbon emission per capita for the countries from the output of `read_powerplant_file` or for the countries from the output of `common_country`. First, we can use `common_country`, `read_pop_data` and `read_co2_data` to obtain the country we need and the data from population and carbon emission data sets. Then, we pick out all the country that is in the set from `common_country` and calculate the carbon emission per capita by getting the carbon emission from the dictionary produced by `read_co2_data` using the country name

and divide it by the population of the country using the country name as the key for the dictionary. Then, we will return the country name and the carbon emission per capita as a list of two lists. First list is the country names and the second list is the carbon emission per capita.

The third piece of information we need is the number of nuclear power plants per capita using `read_nuclear_powerplant` function. First we need to read the power plant data set using the `read_powerplant_data` and get all the countries in all three data sets using `common_country`. Then, I will first find out the number of nuclear power plants for each country and ignore the countries with no nuclear power plants. Then, I will check if the country is in the set from `common_country` to see if it is valid in our project. After this, I will use the `read_pop_data` function to read the population data set and then use the country name after picking from the set of `common_country` to find the population for corresponding countries and divide the number of nuclear power plants by the population of the country to get the number of nuclear power plants per capita for the country as a list of two lists.

The fourth information we need is the carbon emission per capita for the countries from `read_powerplant_data` and we create a function called `read_nuclear_powerplant_co2` to obtain the information. First, I get all the countries by calling the `read_nuclear_powerplant` and use the `read_pop_data` and `read_co2_data` to get the information from the carbon emission data set and the population data set. Then, I divide the carbon emission by the population for each country in the output of the `read_nuclear_powerplant` function and I can obtain the carbon emission and population by searching through the dictionaries produced by `read_co2_data` and `read_pop_data` using the country name from `read_nuclear_powerplant` as the key.

The last information is the country, name, longitude and latitude for every nuclear power plant that is in the country from the `read_nuclear_powerplant` so we create a new function called `nuclear_longtitude_latitude` to achieve this. First, we use the `read_powerplant_data` to read the power plant data set and use the `read_nuclear_powerplant` to get the countries that we are looking for. Then, I will pick out power plants that are in one of the countries from `read_nuclear_powerplant` and is a nuclear power plant and store the country, name of the power plants and the longitude and latitude of the power plants into a list of four lists. The first list is the country name, the second list is the power plant name and the third is the longitude of the powerplant and the last one is the latitude of the power plant.

3.2 Regression

We implemented linear regression in `ols_linear_regression` inside `regression.py` using the ordinary least squares algorithm, based on the formula for the vector of coefficients for the line of best fit from Pillow (2018). For an input matrix X and output matrix Y, the formula from Pillow (2018) for the vector $\vec{w}$ that minimizes the squared error of the predictions is:

$$\vec{w} = (X^T X)^{-1}(X^T Y)$$

In order to find an offset with the regression, we added a column of 1s to the input matrix X, based on Bremer (2012). We added the column of 1s to the right of the input matrix X so that the final element of $\vec{w}$ would be the intercept, which matches the order for outputs of the sklearn `LinearRegression.fit` method.

In order to test our implementation, we wrote `similar_to_sklearn` in `regression.py`. `similar_to_sklearn` splits the data into a train-test split, at a ratio of 8:2, and then learned coefficients and intercept using both the `sklearn LinearRegression().fit` and `ols_linear_regression` methods on the train dataset. Then, we computed the average error on the test dataset for both the sklearn coefficients, and our coefficients, and returned whether the average error for our coefficients is close to or less than the average error for the sklearn coefficients. We emulated the train test split in Varoquaux (2020).

We wrote specific test cases for each regression we wanted to do for the final product, such as `test_nuclear_regression`. These all access the relevant data from `data_processing.py` and convert them into `numpy` arrays, and use the `similar_to_sklearn` to verify that the accuracy is similar.

The computations in `regression.py` were implemented in `numpy` because it's suited for the tasks of doing matrix operations (for our implementation of linear regression) and doing matrix multiplication to make predictions based on the regression coefficients and offset.

3.3 Visualizations

The first step to visualizing our data was to use the data from the data processing step, that was in the format of lists of lists and convert that into Data Frames using the python library `pandas`. For this we implemented 3

functions, `power_plant_df`, `nuclear_emissions_df` and `nuclear_locations_df`. All of these use functions from `data_processing.py` to read the data, and return `pandas.DataFrame` objects, that can easily be used as inputs to graphs made using the `plotly` python library.
The first type of graphs we implemented were 2D scatter plots with linear regressions, using `plotly.express` and `plotly.graph_objects`. We implemented `emissions_power_plants_plot` with emission power plants per capita as the independent variable, `non_emissions_power_plants_plot` with non-emission power plants per capita as the independent variable and `nuclear_emissions_plot` with nuclear power plants per capita as the independent variable, all to predict carbon emissions per capita. All of the above functions have `our_slope` and `our_intercept` as input parameters which come from our regression computations (as seen in `main.py`) to plot the linear relationship between the two variables.
Another type of visualization used is a 3D scatter plot, using the same libraries as before, this time using the `scatter_3d` method to plot emission power plants per capita and non-emission power plants per capita as our independent variables to predict carbon emissions per capita. We then added a regression surface for our prediction with the help of the `numpy` python library. The input parameters are again calculated in `regression.py`.
Next, we wanted to be able to visualize the locations of nuclear power plants around the world, and how those countries compare in terms of carbon emissions per capita. We implemented `nuclear_position_map`, which uses the `scatter_mapbox` method from `plotly.express` and uses the latitudes and longitudes of nuclear power plants from our data to plot them on a world map.

4 Instructions for Obtaining Datasets

All three of our data sets can be downloaded from three different websites (first three link in the reference), but we have uploaded all of them as .csv files to our github repo. Note that we still included the websites where we obtained the data in the references section. All three data set csv files should be stored in the same folder as all the python files (`data_processing.py`, `main.py`, `regression.py`, `visualizations.py`)

Dataset links and instructions to download:

- **Link:** `https://github.com/BraisedShortRib/CSC110-Final-Project/blob/main/countries%20of%20the%20world.csv`

  Instructions to download: Right click the "raw" button, and choose "Save link as...", and then save it in the same folder as all the python files

  NOTE: It may not be "Save link as...", depending on what browser or operating system you are using. In that case, choose the equivalent option for your browser/OS.

- **Link:** `https://github.com/BraisedShortRib/CSC110-Final-Project/blob/main/owid-co2-data.csv`

  Instructions to download: Right click the "download" button, and choose "Save link as...", and then save it in the same folder as all the python files.

  NOTE: It may not be "Save link as...", depending on what browser or operating system you are using. In that case, choose the equivalent option for your browser/OS.

- **Link:** `https://github.com/BraisedShortRib/CSC110-Final-Project/blob/main/global_power_plant_database.csv`

  Instructions to download: Right click the "download" button, and choose "Save link as...", and then save it in the same folder as all the python files

  NOTE: It may not be "Save link as...", depending on what browser or operating system you are using. In that case, choose the equivalent option for your browser/OS.

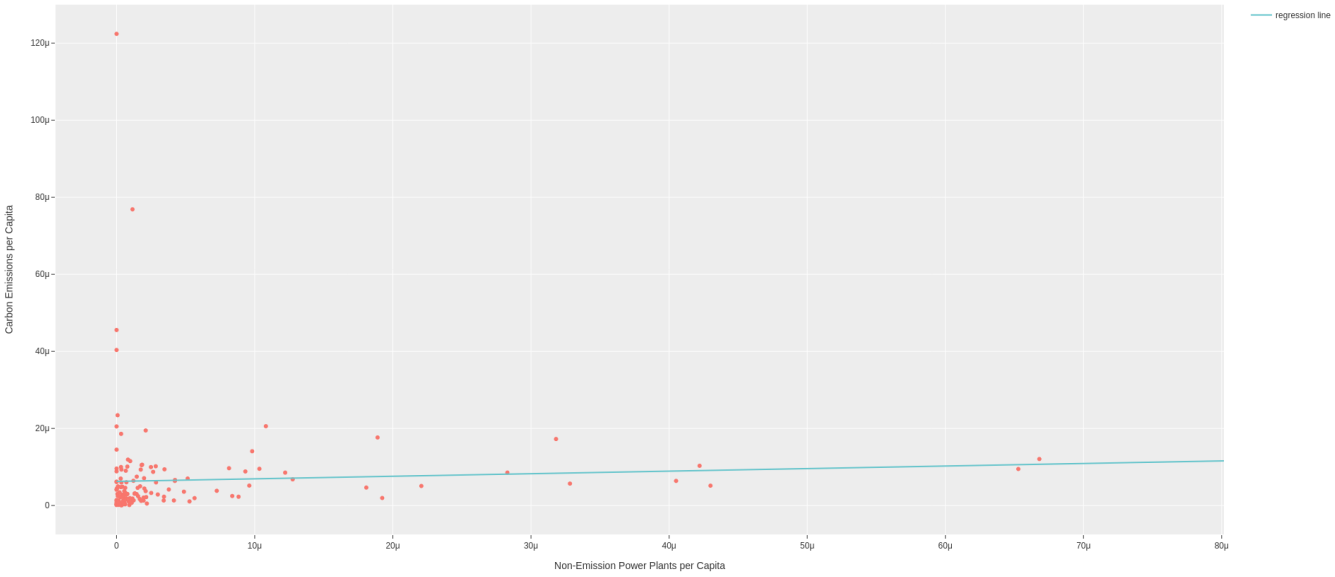
5 Instructions for Running the Project

NOTE: Make sure the `mapbox_token` file is in the same folder as all the python files

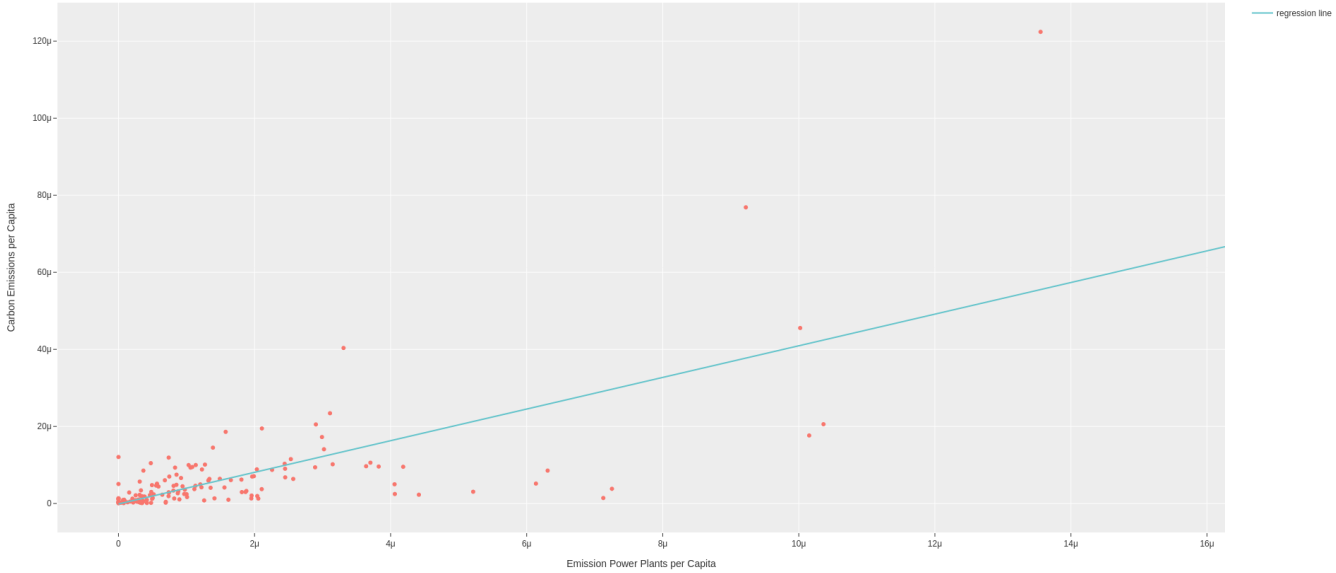
The results of running `main.py` will either appear in your browser, if you already have it open, or open an instance of your default browser and appear in it. If all the graphs do not load, please rerun `main.py`.

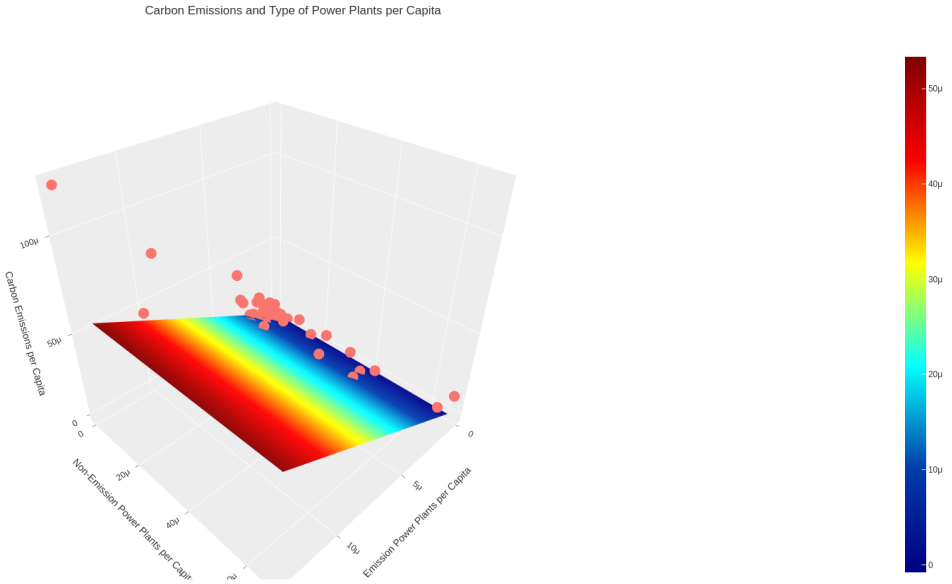
By default, running `main.py` will produce two 2D graphs, and a 3D graph which each show the datapoints and a regression. They are:

Carbon Emissions and Non-Emission Power Plants per Capita

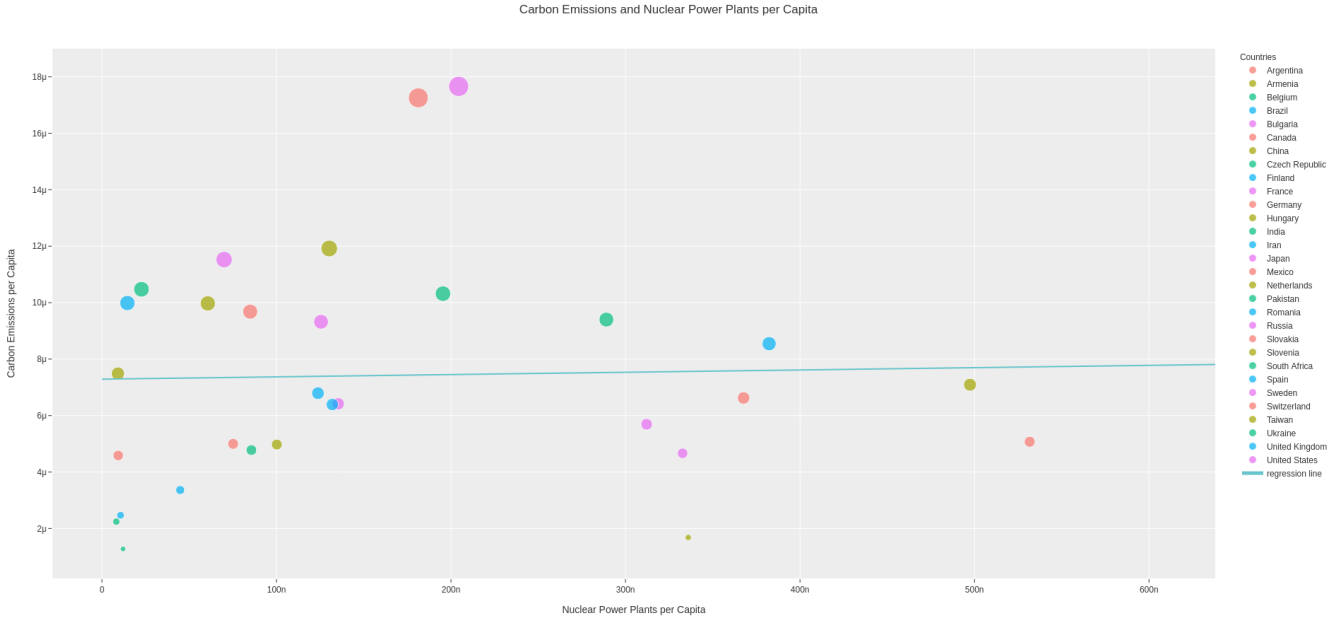


Carbon Emissions and Emission Power Plants per Capita

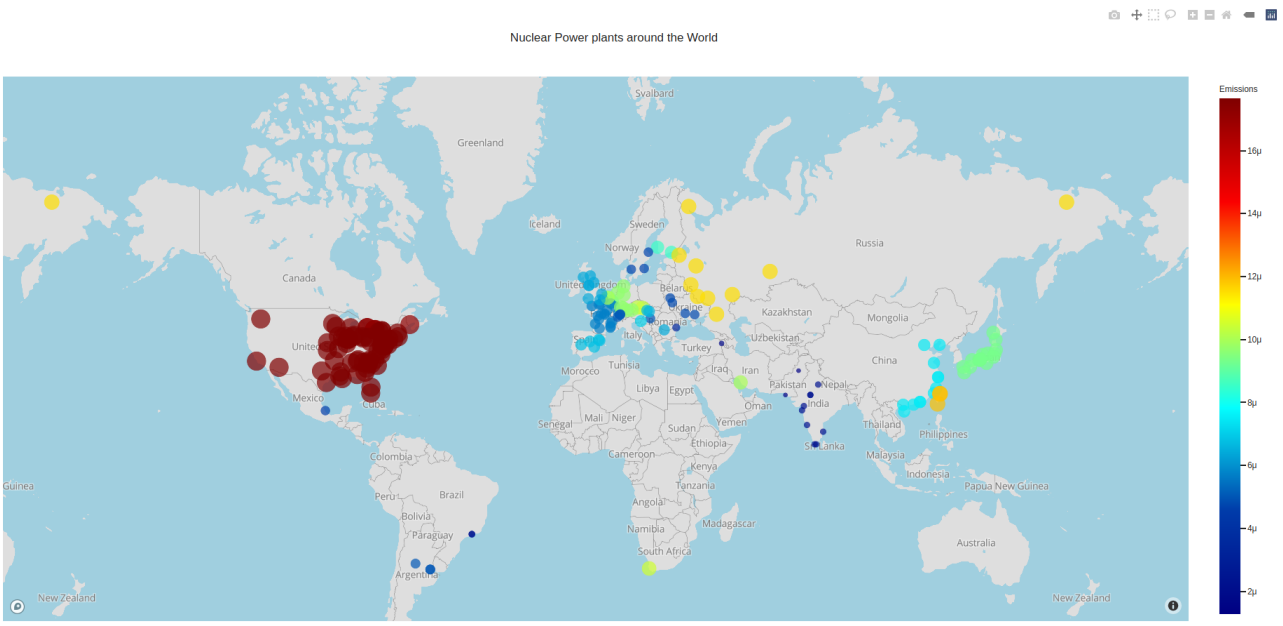




If you uncomment the lines below `UNCOMMENT BELOW TO VIEW NUCLEAR EMISSIONS REGRESSION PLOT AND NUCLEAR PLANTS POSITION MAP`, then you will also see another graph and a map of Nuclear Powerplants in the world:



Each dot represents a country. The color of each dot corresponds to the country while the size corresponds to Carbon Emissions per capita.



Each dot represents a nuclear power plant. The color and size correspond to the country's Carbon Emissions per capita.

The 2D graphs can be interacted with using the labelled buttons on the top right. Individual points can also be hovered over, in order to show their X and Y values.

The 3D graph can be interacted with by clicking and dragging to change the angle that the graph is seen at. You can zoom in or out by scrolling.

The world map can be navigated by clicking and dragging to move, and individual points can be moused over to show the emissions per capita of the country the powerplant is in, as well as the latitude, longitude, and name of the powerplant.

6 Changes to Project Plan

We originally planned to do a regression with two independent variables: the proportion of non-emission powerplants to all power plants in a country and the proportion of emission powerplants to all powerplants in a country for a regression to predict carbon emissions. However, we switched to using Non-Emission powerplants per capita in a country and Emission powerplants per capita in a country as the independent variables and carbon emissions per capita in a country after realizing that the proportion of non-emission powerplants to all power plants in a country is dependent on the proportion pf emission powerplants to all powerplants in a country.

We also changed methods from a gradient-descent based linear regression to an ordinary least-squares linear regression.

We also added the regression with independent variable: Nuclear Powerplants per capita in a country and dependent variable: carbon emissions per capita in a country, as well as the map of nuclear powerplants around the world.

7 Discussion

Our goal with the computations and the visualization was to answer our initial research question, which was, "How does the number of Emissions and Non-Emissions power plants per capita in a country predict their Carbon Emissions per capita?" Our question can be answered through the multiple linear regression and the scatter plots that we implemented. When comparing the first two 2D scatter plots, the ones that use emission power plants or non-emission power plants per capita to predict carbon emissions, we see that the plot with non-emission power plants has a much smaller slope in its regression line. This tells us that carbon emissions increase much faster if a the number of emission power plants per capita in a country increase. And in this case, we can confidently that a plan to replace all power plants that use non-renewable energy sources with ones that use renewable energy souces (including nuclear) is a solid step in limiting global temperature rise. The 3D scatter plot and regression surface, that predicts

carbon emissions using the two independent variables, also supports this claim. With the 3D regression, we found that carbon emissions are zero (or more realistically, very low) when emission power plants per capita are zero, and that non-emissions power plants seems to have very little impact.

Moving on to our exploration of nuclear power plants around the world. In our computations for this project, we calculated a linear relationship between the two variables and ended up with a very small slope value. However, we believe a better predictor of the relationship between these two variables would be non-linear, more specifically an inverse quadratic or square root relationship (deduced by looking at the points plotted on the scatter plot). This is a limitation to our computations, in the future, we could extend upon our project by implementing non-linear regressions as well. So with the scatter plot and the non-linear relationship we see, we can say that the more nuclear power plants a country has per carpita, the less their carbon emissions will change by increasing them. And that the number of nuclear power plants in a country does not have a significant impact on their carbon emissions.

We encountered some issues with our original method for linear regression, based on gradient descent. Because the range of values for our independent values were so small, the gradient for the coefficients were much smaller than the intercept. This meant that our linear regression implementation would essentially leave the initial value for the coefficients unchanged, and only change the intercept. This led us to switch to an ordinary least-squares linear regression implementation, which resulted in more accurate results for our computations.

However, the computations we performed and the data sets we used for them is limited in what we can actually conclude. From our background research, we know that only about 27% of green house gas emissions in the United States in 2018 and 25% of carbon emissions globally in 2010 were a result of the production of electricity and heat (EPA, 2020). This tells us that carbon emissions per capita is not a direct function of power plants, rather there are many other factors, like the country's industries, their agriculture, transportation, commercial activities and residential use that correspond to the increase in carbon emissions that we are seeing today. This is an excellent area for further research, by finding data sets that give information on these other sectors that contribute to global green house gas emissions and looking at the entire whole picture at once, to see what specific changes need to be made so we can limit the rise in global temperatures. Another extension of the work we did for this project, that is maybe more within our reach with the data set we have available to us now is to isolate the other type of power plants that we have, similar to how we looked at the nuclear power plants around the world. This may help us find a power plant type that if in general a country has more of per capita, their carbon emission is lower.

In conclusion, to answer our initial research question, an increase in emission power plants causes a much faster increase in carbon emissions than an increase in non-emissions power plants.

References

[1] Ritchie, H., Roser, M. (2017, May 11). CO and Greenhouse Gas Emissions. Retrieved December 10, 2020, from https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions

[2] Global Power Plant Database - Data: World Resources Institute. (n.d.). Retrieved December 10, 2020, from https://datasets.wri.org/dataset/globalpowerplantdatabase

[3] Lasso, F. (2018, April 26). Countries of the World. Retrieved December 10, 2020, from https://www.kaggle.com/fernandol/countries-of-the-world

[4] Bremer, M. (2012). Multiple linear regression. Retrieved December 13, 2020, from http://mezeylab.cb.bscb.cornell.edu/labmembers/documents/supplement%205%20-%20multiple%20regression.pdf

[5] EPA. (2020, September 10). Global Greenhouse Gas Emissions Data. Retrieved November 06, 2020, from https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data

[6] Evans, S. (2017, December 08). Solar, wind and nuclear have 'amazingly low' carbon footprints, study finds. Retrieved November 05, 2020, from https://www.carbonbrief.org/solar-wind-nuclear-amazingly-low-carbon-footprints

[7] Scatter Plots on Mapbox. (n.d.). Retrieved December 13, 2020, from https://plotly.com/python/scattermapbox/

[8] Scatter Plots. (n.d.). Retrieved December 13, 2020, from https://plotly.com/python/line-and-scatter/

[9] 3D Scatter Plots. (n.d.). Retrieved December 13, 2020, from https://plotly.com/python/3d-scatter-plots/

[10] Sklearn.linear_model.LinearRegression. (2020). Retrieved December 13, 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[11] How can nuclear combat climate change? (n.d.). Retrieved November 05, 2020, from https://www.world-nuclear.org/nuclear-essentials/how-can-nuclear-combat-climate-change.aspx

[12] IEA. (2019, March). Emissions – Global Energy CO2 Status Report 2019 – Analysis. Retrieved November 05, 2020, from https://www.iea.org/reports/global-energy-co2-status-report-2019/emissions

[13] ML Regression. (n.d.). Retrieved December 13, 2020, from https://plotly.com/python/ml-regression/

[14] Pillow, J. (2018). Statistical Modeling and Analysis of Neural Data. Retrieved December 13, 2020, from http://pillowlab.princeton.edu/teaching/statneuro2018/slides/notes03b_LeastSquaresRegression.pdf

[15] United Nations. (n.d.). What is the Paris Agreement? Retrieved November 06, 2020, from https://unfccc.int/process-and-meetings/the-paris-agreement/what-is-the-paris-agreement

[16] U.S. Energy Information Administration (EIA). (2020, February 20). How much carbon dioxide is produced per kilowatthour of U.S. electricity generation? Retrieved November 05, 2020, from https://www.eia.gov/tools/faqs/faq.php?id=74

[17] Varoquaux, G. (2020). Sparsity Example: Fitting only features 1 and 2. Retrieved December 13, 2020, from https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols_3d.html