

Can we make a change on climate change?

EPA1333 Final Assignment

October-November 2017

Introduction

Regardless of the many debates over the impact of humans on climate change, global warming is an observable fact. According to [NASA](#), highest 16 global temperatures have been recorded since 2001, the arctic ice shrank to a minimum in 2012, and carbon dioxide levels in the air are higher than ever!

Data analysis can help us better understand how the climate has changed, why, to make predictions for the future years, and to evaluate measures on how to suppress these changes. In this assignment you receive climate change data from [World Bank](#), and are expected to perform an original and non-trivial analysis using Python.



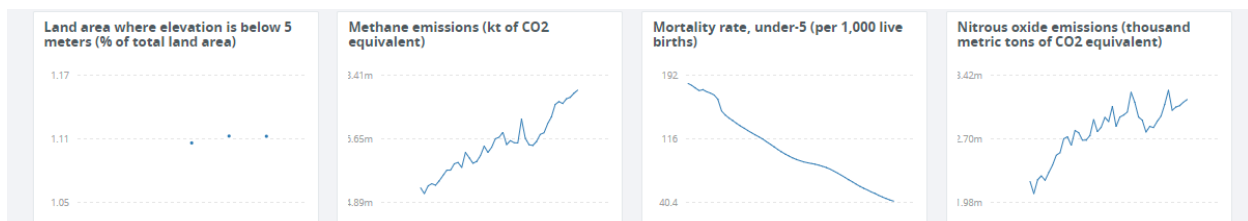
Photo: jocrebbin/Getty Images

Data

Climate change indicators

The climate change data offered by World Bank contains a large set of indicators, such as for example CO2 emissions, population growth or renewable energy output. Values of these indicators are available per country and per year. You can download the data in *.csv format from

<https://data.worldbank.org/topic/climate-change>.



Example data and content types.

Climate data API

The Climate Data API provides programmatic access to most of the climate data used on the World Bank's [Climate Change Knowledge Portal](#). You can access this data directly from Python using [requests](#). In addition to what was already downloadable as csv data, with this API you are able to access temperature, precipitation and basin level data. Read about it in more detail here:

<https://datahelpdesk.worldbank.org/knowledgebase/articles/902061-climate-data-api>

Below is an example of how to access yearly temperature historical data per country from Python. You can select a country using its ISO alpha3 code: <https://unstats.un.org/unsd/methodology/m49/>.

```
In [1]: import requests

In [2]: r = requests.get('http://climatedataapi.worldbank.org/climateweb/rest/v1/country/cru/tas/year/ROU')

In [4]: rou = r.json()

In [6]: type(rou)
Out[6]: list

In [8]: len(rou)
Out[8]: 112

In [10]: rou[-10:]
Out[10]: [{'data': 9.459227, 'year': 2003},
          {'data': 9.777441, 'year': 2004},
          {'data': 9.302619, 'year': 2005},
          {'data': 9.660953, 'year': 2006},
          {'data': 10.971429, 'year': 2007},
          {'data': 10.774226, 'year': 2008},
          {'data': 10.781309, 'year': 2009},
          {'data': 10.492812, 'year': 2010},
          {'data': 10.075417, 'year': 2011},
          {'data': 10.799427, 'year': 2012}]
```

As with most data sources, you might find that you data contains missing values. Please handle them appropriately, for example by using interpolation.

Other resources

You are encouraged to find more data sources that will make your analysis more meaningful. Please make sure that you document everything carefully. Only use freely available datasets.

Assignment

- Create a Jupyter Notebook that contains your explanations and analyses.
- Start the notebook with a clear description of the type of analysis you are going to perform.
- The *conceptual* contents of the Notebook should be roughly similar to a normal written report of 10-20 pages.
- Some (minimum) properties of the Notebook and your analyses on which we will grade:
 - *Required:* Combine different data from multiple sources in your analyses.
 - *Required:* Use multiple types of visualizations of your results.
 - *Required:* Make sure your Notebook **does not** generate errors!

- *Required:* You should use Python to answer your research questions. Your code should read, clean and format, process and visualize the data. There should be at least some non-trivial processing involved.
- Whenever possible, make your Notebook read the data directly from the web. This way, your notebook will always use the most up-to-date data available. If not, document carefully where the data was collected what to do when you want to use more up-to-date data.
- Demonstrate your skills in Python by using typical Python constructs and using the appropriate data structures (lists, dictionaries, tuples, arrays, dataframes, series, recursion, etc.)
- Write *clear, understandable* code:
 - Document your code! Put comments when necessary.
 - Use sensible variable names.
 - Break up your code into parts. Use (fruitful) functions.
- Try to think of reusability of your code. How easy would it be to use your same code if we want to try to do a small variations of your analysis? Can we easily adapt/play around with your code?
- How difficult were your analyses?
- Make your Notebook self-explanatory. So, it should contain text (with references) as well as your actual analysis code and results.
- If you want to use other libraries for your analyses or visualizations, feel free to do so. However, only use *free available and well-known libraries*. The ones that come standard with Anaconda are fine. If you want to use something else, that you first need to install, check with us first.
- Create a zip archive and upload it on Brightspace.

Example

Here are some example questions you might ask to start the analysis. Please note that **this is just an example** and that you are expected to come up with your own questions and analyses.

- The EU has the following goal in the Paris agreement: "At least a 40% domestic reduction in greenhouse gases by 2030 compared to 1990 levels." How is the EU doing at the moment? If they don't change policy (i.e extrapolation of current trends) where will they end up? [\[source\]](#)
- Can you classify in good/neutral/bad countries?
- What is the trend per continent with respect to gas emissions? How do countries within the same continent behave? Is it fair to make statements over whole continents?
- Where do you find the highest increase in temperatures compared to 1960?
- Suppose that each country has a % growth or reduction of CO₂, where do we end up?
- What are the countries with most gas emissions? How does that change when you normalize by their size / population?
- What is the country with the best trend in renewable energy over the last 10 years?

Note : If uncertain about you analysis content, please contact us using EPA1333 Final Assignment in the subject of your email.

Deadline: Sun Nov 5, 23:59