Assignment 1

Trade-offs and synergies between SDGs

Research question

Are social and environmental Sustainable Development Goals (SDGs) correlated, and is there rather a trade-off or a synergy between the two?

Learning goals

- Use Python for function definitions, loops, conditions (→ Lecture 2.2), plotting, and reading and writing to files (→ Lecture 2.3).
- Handle errors (→ Lecture 3.3) and use the function help and an online search engine to find solutions to programming issues, and understand and incorporate them.
- Write nice code (→ Lecture 2.2, slide 28).
- Use inferential statistics for hypothesis testing (→ Lecture 4.2).

Data

Data about the SDG indicators for different countries and years can be downloaded here: https://unstats.un.org/sdgs/indicators/database/

Select 1 environmental AND 1 social SDG indicator (e.g. 1.1.1; see classification below). Note that every goal, but not every indicator, is available in the database.

Please avoid using the same indicator pair as another student, and indicate your choice here: https://drive.google.com/open?id=1rWN7oCJ6mODzrLDoU2UG1uuZkP4hEh7N.

Download the two SDG indicators for all countries (not continents or other regions) and all years.

Environmental SDGs

Social SDGs

Tasks

1) Import the data into Python.
   Note three issues:
   a) The column separator is sometimes also used within the quotes of a column. So, the quotes must additionally be considered when splitting columns.

   b) There are blank lines at the end of the file. Just let your code run until it fails, look at the text line in Python to see how it encodes the blank line, and then tell the programme to break the loop (if you're using a loop) if the text line starts like the blank line.
   c) Open the original data file outside of Python (e.g., in Excel) to check how missing values are encoded, and if there are other values that might not be recognized as numeric. If the reporting value is below the data resolution (e.g., < x), you can set it to this minimum value (i.e. x).
2) Filter the data:
   a) for midpoints (MP) (i.e. exclude LB or UB if such bounds are available),
      [Note that you could do this already during the import.]
   b) for the most recent year for which both indicators are available.
      [Note that the most recent year of one indicator might not be available for the other indicator.]
3) Split the data into two sets of data – one for each indicator, and make and print a sanity check to ensure that both sets of data are comparable.
   Note two issues:
   a) Not all indicators might be available for all countries. Make sure that the countries follow the same order. I recommend sorting them alphabetically (e.g., using numpy.argsort). Afterwards, make the sanity check to confirm that both country lists are identical.
   b) Remove value pairs if the value for one or both countries is missing. Otherwise, it might cause trouble in a subsequent task.
4) Display the two indicators in a scatter plot and export it to a png file.
   Fulfil the following requirements:
   a) Add axis titles (but no figure title). Axis titles are important to understand what is shown in your figure.
   b) Set axis limits. For example, if the indicator range starts at zero, the corresponding axis should also start at zero.
   c) Set the aspect ratio to 1. This makes it easier to compare both indicators.
   d) Change the default settings of at least either the marker symbol or the colour.
   e) Save the figure without large margins.
   f) Do not show the figure in the console.
5) Calculate the linear and monotonic correlation coefficients and their p-values.
6) Print statements to the console which:
   a) separate the results from the earlier sanity check with a blank line
   b) list the two indicators you used, including their description,
   c) give the correlation coefficients,
   d) give their p-values,
   e) give interpretations about if the correlations are statistically significant at a level of 0.01, 0.05, or 0.1 or not at all (write a function for that),
   f) answer if there is rather a trade-off or a synergy between the two indicators (depending on positive or negative correlations).

7) Write code that is nice (see learning goals above) and free of errors and warnings (especially warning symbols shown on the left of a Python script in Spyder), avoid spamming the console (i.e. only print requested results that are not exported), and do not integrate user interaction.

Suggested time breakdown

- Workshop 1: Tasks 1 and 2
- Workshop 2: Tasks 2 and 3
- Workshop 3: Tasks 3 and 4
- Workshop 4: Tasks 5 and 6
- Workshop 5: buffer and review
- All workshops: Task 7

Deliverables

- A Python script named assignment1_*x.x.x_y.y.y*.py (where x.x.x and y.y.y indicate the numbers referring to your selected SDG indicators)
- The input data file with your selected indicators

When running the script in the same folder as the data input file, no user interaction should be requested. One additional file should be automatically created and also submitted:

- An image named sdg_scatter_*x.x.x_y.y.y*.png

→ Collect all the deliverables in a **zip file** to submit it via Brightspace. (Otherwise, there might be trouble uploading the Python script to Brightspace.)

Deadline: Wednesday, **4 December, 18:00**

Assessment

The assignment is worth 1/5 of the final grade, and the grading criteria are:

1) 30%: Pre-processing of data (import, filter, split)
2) 25%: Correlation analysis, including interpretation
3) 25%: Plot content and clarity
4) 20%: Code in general (nice, error- and warning-free, console spam-free)

There will be **no retake** within the same year. So, please take this assignment seriously. There is no reason to fail. If you have difficulties with the assignment, seek help on **Google / Stack Overflow** and attend the **workshops**.