Assignment

Sustainable Development Goals – Trade-offs (A) and Progress (B)

Research question

A	B
Are social and environmental Sustainable Development Goals (SDGs) correlated, and is there rather a trade-off or a synergy between the two?	How is the progress towards the Sustainable Development Goals (SDGs) developing over time, and what can we expect in 2030?

Please indicate your preference for one of these two assignments on Brightspace (half of the students will work on one and the other half on the other assignment):

https://brightspace.universiteitleiden.nl/d2l/le/lessons/92576/topics/1474014

Learning goals

Expected knowledge, which will be applied again to gain more practice: Using Python for function definitions, loops, conditions, reading and writing to files, and plotting.

- Work with pandas data objects (→ Lecture 1).
- Write nice code (→ ESSA Lecture 5, slide 9).
- Use Python for developing Python modules, i.e. create a separate script for functions that is sourced in the main script (function: import, script name without extension).
- Document your code according to a standard style (→ Lecture 4).
- Profile your code to find slow sections and potentially optimize them (→ Lecture 4).
- Integrate a GUI element (→ Lecture 5).
- Track your code changes with git (→ Lecture 2).

A	B
• Use inferential statistics for hypothesis testing (→ Lecture 2).	• Run a model, evaluate it (→ ESSA Lecture 4), and validate it (→ Lecture 3).

Data

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

A	B
Select 1 environmental AND 1 social SDG indicator (e.g. 1.1.1; see classification below).	Select 1 SDG indicator (e.g. 1.1.1) with data for at least ten years for many of the countries. The time series can have data gaps

	and skip some years as long as there are still at least ten years available.
Please avoid using the same indicator pair as another student, and indicate your choice here: https://docs.google.com/spreadsheets/d/1ocameGy60LImPKfjMtRJLifKuz8mv-7s/edit?usp=sharing&ouid=117990309998460343136&rtpof=true&sd=true	Please avoid using the same indicator as another student, and indicate your choice here: https://docs.google.com/spreadsheets/d/12gcAO5lSNT2_bT7K0-pMqg1rpkj0RBo0/edit?usp=sharing&ouid=117990309998460343136&rtpof=true&sd=true If two students select the same indicator, please select two different countries later on for task 5a, and indicate your choice again above.

Note that every goal, but not every indicator, is available in the database.

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

Environmental SDGs

Social SDGs

Tasks

1) Import the data into Python with pandas.
   [Note: 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 reported value (column "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) if different bounds (column "[Bounds]") are available (i.e. not lower or upper bounds, encoded as LB or UB).

A	B
b) for the most recent year for which both indicators are available.	b) for one country at a time in a loop. [Note that you can develop the code

[Note that the most recent year of one indicator might not be available for the other indicator.]	initially for just one country to make everything work with a small dataset.]

3) Make sanity checks and print them to the console.

A	B
Split the data into two sets of data – one for each indicator. Both sets of data should be comparable. a) Make sure that the countries follow the same order. I recommend sorting them alphabetically (e.g., using numpy.argsort). Afterwards, make a sanity check to confirm that both country lists are identical. b) Remove value pairs if the value for one or both countries is missing. Afterwards, make a sanity check to confirm that no missing values remain.	a) Count the non-missing values for each country and provide the minimum, maximum, and average of non-missing values.

4) Perform the main analysis.

A	B
a) Test normality of both datasets by performing modified Kolmogorov-Smirnov / Lilliefors tests. b) Test homoscedasticity by inspecting a scatter plot. c) Test the absence of outliers with the Mahalanobis distance. See https://stackoverflow.com/questions/46827580/multivariate-outlier-removal-with-mahalanobis-distance d) Use the information above to choose either a parametric (Pearson) or non-parametric (Spearman) correlation coefficient. Calculate the correlation coefficient and its p-value, and decide if the result is statically significant or not. e) Answer if there is rather a trade-off or a synergy between the two indicators (depending on the indicator pair and a positive or negative correlation).	a) Calibrate a logistic model. You can use the provided code for running and calibrating the logistic model: assignment_model.py. [Note that the model might not result in a good fit in all cases, but you will not be evaluated for the goodness of fit.] b) Calculate the growth rate and the expected indicator value in 2030 for each country. c) Evaluate your models with three criteria: R^2, NRMSE, and PBIAS. Note that R^2 and NRMSE are both not defined for constant data series (i.e. with a standard deviation of zero). This can be an issue during the validation when the test dataset might be drawn from the plateau of the function. If both the observed and the simulated data series are constant, R^2 can be set to 1. If only one of the two data series is

	constant, R^2 can be set to 0. NRMSE can be consistently normalized with the average instead of the standard deviation to avoid the issue. If all observations are zero, set PBIAS and NRMSE to NaN.
	d) Validate your models with the same three criteria (R^2, NRMSE, and PBIAS) and 5-fold cross-validation.

5) Make plots.

A	B
a) Display the countries in a bubble chart, i.e. a scatter plot where the points have different sizes. The sizes shall represent the population of the countries (see this data, total population – both sexes). Moreover, annotations shall indicate the country names or codes for the 10 most populous countries. b) Display the countries in an additional plot within a world map. For spatial data, you can, for example, use geopandas. It also includes links to some spatial data. For our purposes, you can use 'naturalearth_lowres' to get country polygons, using the function datasets.get_path. To link the countries from both databases, you can match the country codes based on this table. The colours of the countries shall indicate the trade-offs between the two SDGs, i.e. if both goals are met, if both goals are not met, or if either one of the two is met and the other not.	a) Select two contrasting countries and display their observed development and simulated trend in one common time series plot. Tip: Run the model again but with a finer time step to get a smooth line for the simulation. b) Display the 30 most populous countries in an additional plot (see this data, total population – both sexes): an ordered dot plot indicating the growth rate with the colour indicating the projected value in 2030.

Fulfil the following requirements:
   c) Add axis titles (but no figure title). Axis titles are important to understand what is shown in your figure.
   d) Set axis limits. For example, if the indicator range starts at zero, the corresponding axis should also start at zero.
   e) Choose the symbology carefully.
   f) Where relevant, add a legend and place it wisely (not needed for bubble chart of group A).

g) Ensure that any text is readable (e.g., axes, legend, annotations).

h) Save the figure without large margins.

i) Export all plots as png files.

6) Export the main results to a text file.

A	B
Export the following to a txt file, ensuring that it can nicely be read in a text editor:	Export the following to a csv file, ensuring that it can nicely be read in a spreadsheet:
a) the two selected SDG indicators and their description, b) the correlation coefficient (including its type), c) the p-value, d) the interpretation about if the correlation is statistically significant at a level of 0.01, 0.05, or 0.1 or not at all (write a function for that), and e) an answer if there is rather a trade-off or a synergy between the two indicators (depending on positive or negative correlations).	a) the selected SDG indicator and its description, b) the countries, c) the growth rates of the indicator values, d) the expected indicator values in 2030, e) the performance values for the evaluation, and f) the performance values for the validation.

7) Optimize your code.

   a) Profile your code and identify the slow sections.

   b) Try to make slow sections more efficient.

8) Create a radio button for user interaction. Bring the pop-up window to the attention of the user, i.e. bring it to the front or make it blinking. Make sure that it is clear to the user what the selection is about. Automatically close the pop-up window after the selection so that the user knows the selection is done.

   Note: Please keep the code without interaction in a comment so that one can easily switch back. This can help you to work on your code and me to review it.

A	B
Use it to choose between the Pearson and Spearman correlation coefficient.	Use it to choose between the 5- and 10-fold cross-validation.

9) Write code that runs smoothly and is clear.

   a) Write nice code.

   b) Write code that free of errors and warnings (especially warning symbols shown on the left of a Python script in Spyder).

   c) Avoid spamming the console (i.e. only print requested results that are not exported).

   d) Write functions into a separate file and call them in the main script.

   e) Document functions in NumPy or Google style. Use sphinx to export the code documentation to html.

10) Use a version-control software, such as git, to keep track of your progress from week to week and/or of certain milestones (e.g. before and after optimization).

Suggested time breakdown

- Workshop 1: Tasks 1
- Workshop 2: Tasks 2 and 3
- Workshop 3: Tasks 4
- Workshop 4: Task 5
- Workshop 5: Task 6
- Workshop 6: Task 7 (+ peer feedback)
- Workshop 7: Task 8 (+ presentation)
- Workshop 8: Final improvements (+ submission + reflection)
- All workshops: Tasks 9 and 10

Deliverables

- The main Python script named assignment_X_main.py (where X indicates group A or B)
- The Python script with function definitions named assignment_X_functions.py
- The input data files with your selected indicator(s) and the population
- The code documentation as html

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

- Two images named xxx.png (where xxx indicates the type of plot as bubble, time_series, map, or dot)
- An output data file named sdg_correlation.txt (group A) or sdg_simulation.csv (group B)

→ 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: Tuesday, **9 November at 18:00**

Assessment

The assignment is the only assessment. There will be no exam. The grading criteria are as follows (grey implies separate submission deadlines):

1) 10%: Pre-processing of data (tasks 1-3)
2) 23%: Main analysis (task 4)
3) 17%: Plot content and clarity (task 5)
4) 5%: Data file content and clarity (task 6)
5) 5%: GUI (task 8)
6) 10%: Code in general (task 9)
7) 10%: Peer feedback (evaluation by peer, given a rubric)
8) 10%: Presentation
9) 10%: Reflection (incl. tasks 7 and 10)

I expect that you can all pass this assignment. If you have difficulties with the assignment, seek help on **Google / Stack Overflow** and attend the **workshops**. If a **retake** should be needed, you will get more time for the same assignment but can at most score a 6.0.