SAPY Assignment 2025-2026

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

Research question

A: Are social and environmental Sustainable Development Goals (SDGs) correlated, and is there a trade-off or a synergy between the two?	B: How is the progress towards the Sustainable Development Goals (SDGs) developing over time, and what can we expect in 2035?

Please indicate your preference for one of these two assignments. The class will need to divide itself evenly over both assignment options: Assignment Registration.xlsx

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 6).
- Profile your code to find slow sections and potentially optimize them (→ Lecture 8).
- Track your code changes with git (→ Lecture 2).
- Work in a virtual environment (→ Lecture 1).

A: Use inferential statistics for hypothesis testing (→ Lecture 3).	B: Run a model, evaluate it (→ ESSA Lecture 4), and validate it (→ Lecture 5).

Data

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

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.

A	B
Select 1 environmental AND 1 social SDG indicator (e.g. 1.1.1; see classification below). Please avoid using the same indicator pair as another student.	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 as another student. If two students select the same indicator, they should pick two different countries for task 6a.

Environmental SDGs

6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND

Social SDGs

1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES

Tasks

1) Create a virtual environment for your code. Before submission, when all packages needed for your complete code are installed, export the environment as a yaml file.
2) 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 other values 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).]
3) 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. Note that the most recent year of one indicator might not be available for the other indicator.	b) for one country at a time in a loop. Note that you can develop the code initially for just one country to make everything work with a small dataset.

4) 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). Apply 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. Confirm that no missing values remain with a sanity check.	a) Count the non-missing values for each country and provide the minimum, maximum, and average of non-missing values.

5) 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 statistically 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_B_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 2035 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.

6) 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 2035.

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.
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.

7) Export the main results to a text file.

A	B
Export the following to a txt file, ensuring that it can be nicely read in a text editor: a) the two selected SDG indicators and their description, b) the correlation coefficient (including its type), c) the p-value, d) the interpretation of whether 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).	Export the following to a csv file, ensuring that it can be nicely read in a spreadsheet: 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 2035, e) the performance values for the evaluation, and f) the performance values for the validation.

8) Optimize your code.
   a) Profile your code and identify the slow sections.
   b) Without AI, try to make slow sections more efficient.
   c) If desired, use AI to figure out options to optimize your code. Include your prompt, and compare the results with 8b. Are there any surprises? If you use any of these ideas, include an AI disclosure with your final submission (see Task 11).
9) Write code that runs smoothly and is clear.
   a) Write nice code.
   b) Write code that is 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 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).
11) If applicable: If you used AI for brainstorming, improving your code, troubleshooting, code improvements, etc.:
    a) Include an AI disclaimer that states how AI was used in your final submission.
    b) Clearly comment on which line(s)/segments were written or improved using AI.

Suggested time breakdown

- Week 1: Tasks 1-2
- Week 2: Tasks 3 and 4
- Week 3: Tasks 5
- Week 4: Task 6
- Week 5: Task 7 (+ peer feedback)
- Week 6: Task 8 (+ presentation)
- Week 7: Task 9
- Week 8: Final improvements (+ submission + reflection)
- All workshops: Tasks 9, 10, and 11

Deliverables for the final code submission

- 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
- The file describing the virtual environment named environment.yaml

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: Thursday, **6 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) 12%: Virtual environment and pre-processing of data (tasks 2-4)
2) 26%: Main analysis (task 5)
3) 17%: Plot content and clarity (task 6)
4) 5%: Data file content and clarity (task 7)
5) 10%: Code in general (task 1, 8, and 9)
6) 10%: Peer feedback (evaluation by peer, given a rubric)
7) 10%: Presentation
8) 10%: Reflection (incl. tasks 7, 10, and 11)

Overview of deliverables and due dates

		Deliverables due at 18:00 local time
Thursday	9-Oct	Preliminary code
Wednesday	15-Oct	Peer-feedback
Friday	17-Oct	Evaluation of peer feedback
Tuesday	27-Oct	Presentation slides
Thursday	6-Nov	Final code
Tuesday	11-Nov	Written reflection

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 you can at most score a 6.0.