Assignment

Sustainability Indicators – Trade-offs (A) and Distribution (B)

Research question

A	B
Are social and environmental sustainability indicators correlated, and is there rather a trade-off or a synergy between the two?	How well do countries perform regarding sustainability, and does the overall distribution change if more countries are considered through imputation?

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/239815/topics/2523286

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).
- Integrate a GUI element (→ Lecture 4).
- Document your code according to a standard style (→ Lecture 5).
- Profile your code to find slow sections and potentially optimize them (→ Lecture 5).
- Track your code changes with git (→ Lecture 2).
- Work in a virtual environment (→ Lecture 1).

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

Data

Data about sustainability-related indicators for different countries and years can be found at the World Bank here: https://data.worldbank.org/indicator?tab=all

A	B
Select a country, 3 indicators related to social	Select a primary and a secondary

sustainability, as well as 3 related to environmental sustainability. Please avoid using the same country and set of indicators as another student (among these 7 choices, at least 3 should differ) and indicate your choices here (the link will be activated after the groups are settled): https://docs.google.com/spreadsheets/d/16TonbG-S8_yiHX2l0tBtg7W_ragMGq2U/edit?usp=drive_link&ouid=117990309998460343136&rtpof=true&sd=true	sustainability-related indicator. Ideally, the primary indicator has several missing country values, while the secondary indicator has much fewer data gaps and can be used to support filling data gaps of the primary indicator. Please avoid using the same indicator pair as another student (at least one should differ) and indicate your choice here (the link will be activated after the groups are settled): https://docs.google.com/spreadsheets/d/114b52JqGv3rqYySZYmdKCmD3Tb1Nnswd/edit?usp=drive_link&ouid=117990309998460343136&rtpof=true&sd=true

Tasks

1) Create a virtual environment and run your code there. Before submission, when all packages needed for your complete code are already installed, export the environment to a yaml file.
2) 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. print requested results that are not exported but not content like large data objects that are difficult to look at in the console and hide more interesting output).
   d) Write functions into a separate file and call them in the main script.
   e) Document at least one of the functions in NumPy or Google style. Use sphinx to export the code documentation to html.
3) Download the data as a csv file and import it into Python with pandas OR use the package wbgapi to load the data directly from the web into Python as pandas objects (e.g., with the data sub-package).
   Note: For using the wbgapi package, you can find examples of how to use it at https://pypi.org/project/wbgapi/ and https://blogs.worldbank.org/opendata/introducing-wbgapi-new-python-package-accessing-world-bank-data.
4) Pre-process the data.

A	B
a) Indicate one of the indicators as your indicator of most interest by assigning its name to a variable near the top of your script.	a) Filter the data for a recent year. Note: You can choose the year based on information on the website for the indicator and do not need to write code to

b) Filter the data for your country of choice. c) Check if each of the chosen indicators has at least 3 non-missing values and print a statement about it. d) Check if each indicator pair has at least 3 overlapping years with non-missing values and print a statement about it. Save the counts in a variable, as you will reuse them in task 5b. e) Remove all leading years with only missing values to reduce the data size. In other words, take a subset of your data that starts in the earliest year, in which the data is not missing for all 6 indicators. Note: If c or d do not apply, choose a different indicator or country to ensure you have sufficient data for your analysis.	identify the most recent year. b) Filter the data for all countries, i.e., exclude regions. Note: You can find information on countries vs. regions in Metadata_Country_API_xxx.csv OR in the feature economy. c) Check the number of missing values in the primary and secondary indicators and print the results. d) Determine the number of countries for which the data gaps in the primary indicator can be filled based on information available for the secondary indicator and print a statement about it. e) Create a multi-index based on the two country-related attributes: the ISO3 codes and the names. (It makes some later analyses a bit easier by not having to exclude this column several times.)

5) Perform the main analysis.

A	B
a) Make any choices in the following subtasks through conditional statements. b) Print the conclusions from the assumption checks below, the main test used, and its conclusion to the console. c) Calculate a correlation matrix among all 6 indicators, using the Spearman rank correlation coefficient. d) Select a second indicator that forms a pair with the indicator selected in task 4a by considering only those indicators with at least 10 paired data points (or 5 if it is too limiting) and selecting among these the one with the strongest (highest absolute) correlation. Use this indicator pair in the following steps of the main analysis. e) Test normality of both datasets by performing modified Kolmogorov-Smirnov / Lilliefors tests. f) Test homoscedasticity by inspecting a	a) Fill missing values in the primary indicator based on univariate imputation with the average. b) Fill missing values in the primary indicator based on the secondary indicator and a multivariate imputation with a small number of nearest neighbours. Weigh each nearest neighbour uniformly. c) Fill missing values in the primary indicator based on the secondary indicator and a multivariate imputation with a small number of nearest neighbours. Weigh each nearest neighbour by distance. d) Validate the three different types of imputation with three criteria (R^2, NRMSE, and PBIAS) and 5-fold cross-validation. e) Take the average of the performance metrics from the 5 rounds of cross-validation so that you have just one set of

scatter plot. g) Test the absence of outliers with the Mahalanobis distance. h) Use the above information to choose a parametric (Pearson) or non-parametric (Spearman) correlation coefficient. Calculate the correlation coefficient and its p-value, and decide whether the result is statistically significant. i) 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).	performance metrics per type of imputation. f) Collect the results from subtask e in a common data frame and print the results.

6) Export the main results to a csv file, ensuring that it can be read nicely in a spreadsheet.

A	B
a) Export the correlation matrix, including both the indicator code and name as row and column names (i.e., use multi-indices in the data frame). b) Remove the names of the series with the row and column names before the export. (This avoids blank cells between the headers and the data and thereby improves the formatting of the exported file.)	a) The original primary indicator (not the secondary indicator) b) The three sets of imputed data on the primary indicator c) The country-related information from the multi-index d) All including intuitive but short column names (e.g., “economy” from the World Bank dataset is not intuitive for ISO3 country codes)

7) Make plots.

A	B
a) Display the lower triangle of the correlation matrix in a heat map while also displaying the correlation coefficients in the cells. (A full correlation matrix would be repetitive, as one triangle mirrors the other.) b) Highlight the selected indicator, e.g., by presenting it in bold. c) Distinguish the social and environmental indicators, e.g., by grouping them and drawing a separating line between the groups. d) In a separate figure, create a scatter plot for your selected indicator pair, in which	a) Plot a histogram of the original primary indicator. b) Divide the original and the three imputed primary indicators each into three segments. You can choose the two thresholds for this division based on the data distribution shown in subtask a and what seems meaningful to you to distinguish low, moderate, and high sustainability. c) Make a stacked bar chart of these segmentations that shows the proportions from 0 to 1.

the colours of the points reflect the year.	

Fulfil the following requirements:

c) Add axis titles where appropriate (but no figure title). Axis titles can be important to understand what is shown in your figure.
   Note: You can omit the axis titles in the heat map (group A), as well as on the axis where you indicate the four indicator sets for the stacked bar chart (group B).
d) Choose the symbology carefully, and change the default colours.
e) Add a legend or colour bar where relevant and place it wisely.
f) Ensure that any text is readable (e.g., axes, legend, annotations).
g) Save the figure without large margins.
   Avoid also unnecessary white space because of excessively long axis titles or labels. You could, for example, write axis titles across two lines if necessary.
h) Set the resolution of the figures to 150 dpi.
i) Export all figures as png files.

8) Create a GUI element for user interaction. Bring the pop-up window to the user's attention, i.e. bring it to the front (or activate it to make the icon blink in the taskbar). Make sure that it is clear to the user what the selection is about.
   Add a Boolean variable near the top of the main script where the interactivity can be activated or disabled. This can help you to work on your code and me to review it.
   Note: Use object-oriented, event-driven programming like with PyQt or Tkinter but not easygui.

A	B
a) Create a toggle button to indicate if the assumptions for the Pearson correlation coefficient were met or not. You can, for example, use a push button as a basis that changes the colour and text upon clicking on the button. Alternatively, you could work with the qtwidgets library that includes toggle buttons and add labels to the left and right sides of it to make the meaning of the toggle button very clear. b) Add an OK button that closes the pop-up window. c) Based on the indication about the assumptions, choose automatically between the Pearson and Spearman correlation coefficients.	a) Create two sliders to choose two thresholds that will subsequently be used for the segmentation in task 7b. b) Use the thresholds selected above as the default choices. c) Display the currently selected values within the GUI element. d) Display a warning message if the thresholds are not chosen in the intended order. e) Add an OK button that closes the pop-up window.

9) Use version-control software, such as git, to keep track of certain milestones, e.g., before and after optimization.

Note: Keep in mind that you will show code comparisons from git in your reflection. If you did not manage to optimize your code, show any other changes.

10) Optimize your code.
    Note: Disable the interactivity while doing so. Keep in mind that you will write about the optimization in your reflection and use git to show code sections before and after the optimization.
    a) Profile your code and identify the slow sections. Profiling is more informative if you have defined several own functions. If it does not seem very helpful in your case, use your own judgment to identify slow sections.
    b) Try to make slow sections more efficient. Even if the code already runs quickly, there is likely still some potential to make it even faster.
    c) Measure the running time before and after the optimization to show the improvement in efficiency.

Suggested time breakdown

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

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 for your selected indicator if you do not use the wbgapi package
- The code documentation as html file (not the entire folder, just the file, but check which one is the correct html file)
- The file describing the virtual environment named environment.yaml

When running the script (in the same folder as the data input file where relevant; do not use any hardcoded absolute file paths), the following additional files should be automatically created and also submitted:

- Images named xxx.png (where xxx indicates the type of plot as heat_map, scatter_plot, or stacked_bars)
- An output data file named correlation.csv (group A) or data_imputed.csv (group B)

→ Collect all the deliverables (not the folder with the deliverables) in a **zip file** to submit via Brightspace. (Otherwise, there might be trouble uploading the Python scripts to Brightspace.)

Deadline: Tuesday, **7 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%: Virtual environment and code in general (tasks 1-2)
2) 5%: Import and pre-processing of data (tasks 3-4)
3) 25%: Main analysis (task 5)
4) 5%: Data file content and clarity (task 6)
5) 15%: Plot content and clarity (task 7)
6) 10%: GUI (task 8)
7) 10%: Peer feedback (evaluation by peers, given a rubric)
8) 10%: Presentation
9) 10%: Reflection (incl. tasks 9 and 10)

I expect that you can all pass this assignment. If you struggle with the assignment, seek help on **Google / Stack Overflow** and attend the **workshops**. If a **retake** should be needed, you would get more time for the same assignment but could only reach a 6.0.