SUSTAINABILITY ANALYSIS IN PYTHON – ASSIGNMENT A

Courses 4413SUSAP / 4413SUAP6Y at Leiden University

Research question

How big are the gaps between the lifestyle carbon footprints and the 1.5°C target, and how much do different consumption categories contribute to the variation in these gaps for your selected country group and target year?

Please indicate your preference for assignment A or B 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/331477/topics/3108569

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).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101003880.

- 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).
- Assignment A only: Use inferential statistics for hypothesis testing (→ Lecture 2).

Data

The input data are based on an analysis from a publication of the Horizon 2020 project EU 1.5° Lifestyles (https://onepointfivelifestyles.eu/):

Cap, S., de Koning, A., Tukker, A., & Scherer, L. (2024). (In)Sufficiency of industrial decarbonization to reduce household carbon footprints to 1.5°C-compatible levels. SUSTAINABLE PRODUCTION AND CONSUMPTION, 45, 216-227. https://doi.org/10.1016/j.spc.2023.12.031

The following input data are provided to you:

- footprints.csv (reporting the lifestyle carbon footprints of all countries and regions available in the input-output tables of EXIOBASE)
- categories.csv (reporting the same lifestyle carbon footprints divided into different consumption categories)

Select a country group (the EU or G20 countries), a target year (2030 or 2050), baseline or projected footprints (year 2015 or target year), and the method of non-parametric correlation (Spearman rank correlation or Kendall's Tau correlation).

Please avoid using the same combination of choices as another student and indicate your choices here (the link will be activated after the groups are settled):
https://leidenuniv1-my.sharepoint.com/:x:/r/personal/schererla_vuw_leidenuniv_nl/Documents/Teaching/Sustainability%20Analysis%20in%20Python/selection_assignment_A_2024.xlsx?d=w27b53d6b693e4353b53fea6f5a27582d&csf=1&web=1&e=0DATzb

A list of the countries within each country group is also provided to you:

- EU.csv
- G20.csv

Tasks

1. Create a virtual environment and run your code there. Before submission, when all packages needed for your complete code have already been 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. Import the provided data into Python with pandas, and indicate near the top of your main script the median target household emissions in your selected target year, which you can look up from Table 2 of the abovementioned publication.

4. Pre-process the data.
    a) Filter the data for the relevant year, depending on the choices of the target year and baseline or projected footprints.
    b) Filter the data for the selected country group.
    c) Reshape the categories data from a long to a wide format so that the emissions of the different categories are in separate columns. This can facilitate later analysis of the data.
    d) Set the region as the index in both data sets and remove the year that became redundant after the filtering.

5. Perform the main analysis.
    a) Calculate the gaps between the lifestyle carbon footprints and the emissions compatible with the 1.5°C target.
    b) Calculate the contribution to variance (CTV) to check how much each consumption category contributes to the variation in the gaps. It is calculated based on a non-parametric correlation coefficient (NPCC) as follows:

$$CTV_i = \frac{NPCC_i^2}{\sum_i NPCC_i^2}$$

    c) Identify the category with the largest contribution to variance. Use this category together with the gaps in the following subtasks.
    d) Make any choices in the following subtasks through conditional statements.
    e) Print the conclusions from the assumption checks and the main test.
    f) Test the normality of both datasets by performing modified Kolmogorov-Smirnov / Lilliefors tests.
    g) Test homoscedasticity by inspecting a scatter plot.
    h) Test the absence of bivariate outliers with the Mahalanobis distance.
    i) Use the above information to choose a parametric (Pearson) or non-parametric correlation coefficient. Calculate the correlation coefficient and its p-value, and decide whether the result is statistically significant.

6. Export the main results to a csv (comma-separated values) file, ensuring it can be read nicely in a spreadsheet.

a) Remove any categories without contributions to variance before the export.
b) Export the contributions to variance as percentages.
c) Indicate the choices made for this analysis (i.e., country group, target year, type of footprints, and correlation method) in the first rows of the exported dataset (which can easily be skipped when importing the data) and separate them by a blank line from the main results.

7. Make plots.
   a) Create a stacked bar plot that distinguishes the parts of the lifestyle carbon footprints below and above the 1.5°C target.
   b) Add a horizontal line to that plot to demarcate the 1.5°C target.
   c) Highlight a country of special interest to you, e.g., by presenting it in bold.
   d) Integrate the project logo (provided as lifestyles_logo.png) into the figure. You can use the Image module of the Python Imaging Library (PIL) for that.
   e) In a separate figure, create a pie chart based on the contributions to variance, also displaying the percentages of these different categories.

   Fulfil the following requirements:
   f) 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 pie chart.
   g) Choose the symbology carefully and change the default colours.
   h) Add a legend where relevant, place it wisely, and choose the order of legend items carefully (e.g., in a stacked bar plot).
   i) Ensure that any text is readable (e.g., axes, legend, annotations), and use super- or subscripts where relevant.
   j) 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.
   k) Set the resolution of the figures to 150 dpi. This resolution is higher than the default and will also increase the figure size when opened with an image viewer.
   l) 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) Create a dropdown menu to choose a country that will subsequently be highlighted in the stacked bar plot.
   b) Use the country selected above as the default choice.
   c) 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 (Task 10).
   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. You will also be asked to report the time reduction in the reflection.

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

Deliverables

- The main Python script named assignment_A_main.py
- The Python script with function definitions named assignment_A_functions.py
- The input data files and the file for your selected country group
- The code documentation as html file (the entire html folder is quite large and usually not needed, just the file, but check which one is the correct html file; some students may have split up the code documentation into multiple html files so that more than one would have to be submitted; if you referred to any other files such as images that you integrated, I might still need the entire html folder)
- The file describing the virtual environment named environment.yaml

When running the script (in the same folder as the data input file; 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 stacked or pie)
- An output data file named contribution.csv

→ Collect all the deliverables (not the folder with the deliverables) in a **zip or tar file** to submit via Brightspace. (Otherwise, there might be trouble uploading the Python scripts to Brightspace.) Deadline: Tuesday, **5 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):

- 10%: Virtual environment, code documentation, and code in general (Tasks 1-2)
- 5%: Import and pre-processing of data (Tasks 3-4)
- 25%: Main analysis (Task 5)
- 5%: Data file content and clarity (Task 6)
- 15%: Plot content and clarity (Task 7)
- 10%: GUI (Task 8)
- 10%: Peer feedback (evaluation by peers, given a rubric)
- 10%: Presentation
- 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 you needed a **retake**, you would get more time for the same assignment but could only reach a 6.0.