SIG731 2023: Task 1P

Introduction to Python and Jupyter Notebooks

Last updated: 2023-11-24

Contents

1 Task

Create a single Jupyter/IPython notebook (see the *Artefacts* section below for all the requirements – read the whole task specification first!), where you perform what follows.

Do not use **numpy** nor **pandas**. This is an exercise on base Python.

1. Input three Python lists of identical lengths (at least five items each) giving basic data on your friends/family, for example:

```
names   = ["Barbara", "Anne", "Greg", "Maggie", "Nynke"]
heights = [173,       161,    180,    158,      177 ]  # in centimetres
weights = [61,        79,     83,     58,       57  ]  # in kilograms
```

   It is assumed that names[i] gives the name of the i-th person whose height in centimetres is heights[i] and weight in kilograms is weights[i].

   You should enter different data, not exactly the values above.

2. Using a *for* loop, create (programmatically) a Python list bmis such that bmis[i] gives the body mass index of the i-th person.

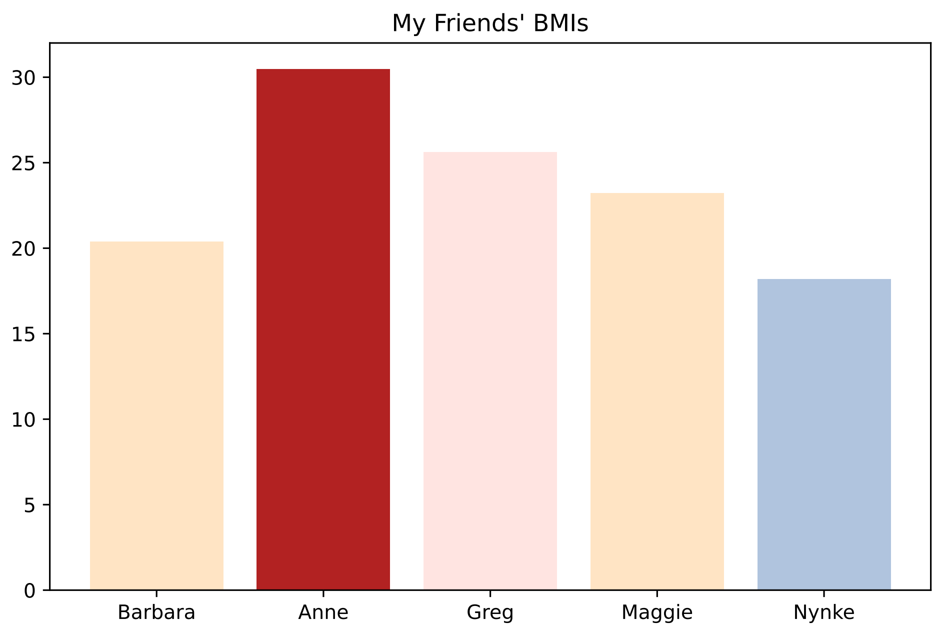
   Then, for each person, calculate *also* the *New BMI (exponent of 2.5)* measure as defined in https://en.wikipedia.org/wiki/Body_mass_index.

3. Based on the BMI categories as defined by the WHO (underweight if below 18.5, normal range below 25.0, overweight below 30.0, obese otherwise), use a *for* loop to print a series of strings like "{name} has BMI of {bmi} which is {bmi_category}. The new BMI index is {new_bmi}.":

```
## Barbara  has BMI of 20.38 which is normal. The new BMI index is 20.14.
## Anne     has BMI of 30.48 which is obese. The new BMI index is 31.23.
## Greg     has BMI of 25.62 which is overweight. The new BMI index is 24.82.
## Maggie   has BMI of 23.23 which is normal. The new BMI index is 24.03.
## Nynke    has BMI of 18.19 which is underweight. The new BMI index is 17.78.
```

Make sure the formatting is neat and tidy.

4. Draw a bar plot like below, where the bar colours depend on the BMI categories.



Hint: you can find the relevant code example in *Module 1* on our unit site.

5. The Wikipedia article on BMI correctly identifies both indexes as very simple measures. In your own words, discuss what are the benefits and limitations of the BMI and the new BMI from both the medical and societal perspective, including its possible misuses (write at least three text paragraphs).

Enter `names`, `heights`, and `weights` only once at the beginning of the notebook. Note that your code must work correctly if someone decides to modify these lists: for example, add another person to the database.

Do not hardcode the values of the `bmis` – they must be computed programmatically.

Minimise the printing of unnecessary objects.

2 Artefacts

Make sure that your notebook has a **readable structure**; in particular, that it is divided into sections. Use rich Markdown formatting (text in dedicated Markdown chunks – not just Python comments).

Do not include the questions/tasks from the task specification. Your notebook should read nicely and smoothly – like a report from data analysis that you designed yourself. Make the flow read natural (e.g., *First, let us load the data on... Then, let us determine... etc.*). Imagine it is a piece of work that you would like to show to your manager or clients — you certainly want to make a good impression. Check your spelling and grammar. Also, use formal language.

At the start of the notebook, you need to provide: the **title** of the report (e.g., *Task 42: How Much I Love This Unit*), your **name**, **student number**, and **email address**.

Then, add 1–2 introductory paragraphs (an introduction/abstract – what the task is about).

Before each nontrivial code chunk, briefly **explain** what its purpose is. After each code chunk, **summarise and discuss the obtained results** (in a few sentences).

Conclude the report with 1–2 paragraphs (summary/discussion/possible extensions of the analysis etc.).

Checklist:

1. Header, introduction, conclusion (Markdown chunks).
2. Text divided into sections, all major code chunks commented and discussed in your own words (Markdown chunks).
3. Every subtask addressed/solved. In particular, all reference results that are part of the task specification have been reproduced (plots, computed aggregates, etc.).
4. The report is readable and neat. In particular:
    - all code lines are visible in their entirety (they are not too long),
    - code chunks use consecutive numbering (select *Kernel - Restart and Run All* from the Jupyter menu),
    - rich Markdown formatting is used (`# Section Title`, `* bullet list`, `1. enumerated list`, `| table |`, `*italic*`, etc.),
    - the printing of unnecessary/intermediate objects is minimised (focus on reporting the results specifically requested in the task specification).

Submissions which do not *fully* (100%) conform to the task specification *on* the cut-off date will be marked as FAIL.

Good luck!

3 Intended Learning Outcomes

ULO	Is Related?
ULO1 (Data Processing/Wrangling)	YES
ULO2 (Data Discovery/Extraction)	
ULO3 (Requirement Analysis/Data Sources)	
ULO4 (Exploratory Data Analysis)	
ULO5 (Data Privacy and Ethics)	YES