

SIT220/731 2023.T1: Task 1P

Introduction to Python and Jupyter Notebooks

Last updated: 2023-03-07

Contents

1	Introduction	1
2	Task	1
3	Additional Tasks for Postgraduate (SIT731) Students (*)	2
4	Artefacts	3
5	Intended Learning Outcomes	3

1 Introduction

This task is related to Module 1 (See the *Learning Resources* on the unit site).

Ideally, you should complete this task by the end of **Week 1**.

It is due on **Week 3 (Friday)**. Start tackling this task as early as possible. If we find your first solution incomplete or otherwise incorrect, you will still be able to amend it based on the generous feedback we will give you (allow 3–5 working days). In case of any problems/questions, do not hesitate to attend our on-campus/online classes or use the Discussion Board on the unit site.

Submitting after the aforementioned due date might incur a late penalty. The **cut-off date is Week 4 (Friday)**. There will be **no extensions** (this is a Week 1 task, after all...) and no solutions will be accepted thereafter. At that time, if your submission is not 100% complete, it will be marked as FAIL, without the possibility of correcting and resubmitting. This task is part of the **hurdle requirements** in this unit. Not submitting the correct version on time results in failing the unit.

A good data engineer must have fine time management skills. To ensure a fair environment for all, we are always very strict about deadlines. Luckily, most students know how to do the right thing. If you are one of them, you are the best; thank you.

2 Task

Create a single Jupyter/IPython notebook (see the *Artefacts* section below for all the requirements), where you perform what follows.

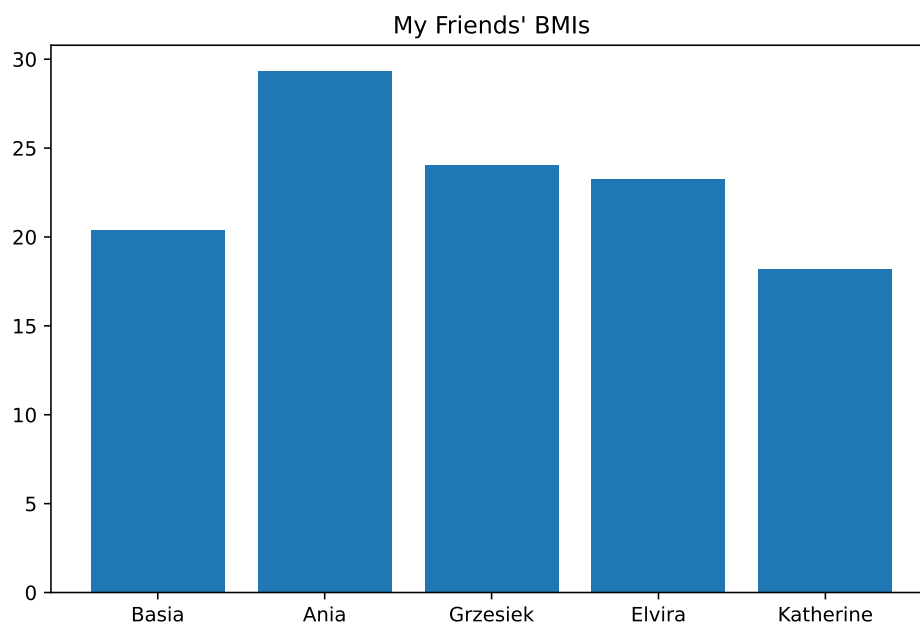
1. Input three Python lists of identical lengths (at least five items each) giving basic data on your friends/family, for example:

```
names = ["Basia", "Ania", "Grzesiek", "Elvira", "Katherine"]
heights = [ 173, 161, 180, 158, 177 ] # in centimetres
weights = [ 61, 76, 78, 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. Create a Python list `bmis` such that `bmis[i]` gives the [body mass index](#) of the *i*-th person.
3. Draw a barplot like:



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

4. Based on the BMI categories as defined by the WHO (underweight if below 18.5, normal range up to 25.0, etc.), write some code that outputs a series of strings like "{name} has BMI of {bmi} which is {bmi_category}.":

```
## Basia      has BMI of 20.38 which is normal.
## Ania       has BMI of 29.32 which is overweight.
## Grzesiek   has BMI of 24.07 which is normal.
## Elvira     has BMI of 23.23 which is normal.
## Katherine  has BMI of 18.19 which is underweight.
```

Make sure the formatting is neat and tidy.

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

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

3 Additional Tasks for Postgraduate (SIT731) Students (*)

Postgraduate students, apart from the above tasks, are additionally **required** to solve/address/discuss what follows.

1. For each person, calculate also the *BMI prime* measure as defined in https://en.wikipedia.org/wiki/Body_mass_index.
2. Draw a separate barplot of the BMI prime indices of the persons in our database.
3. Modify the textual reporting scheme so that the output strings are like "{name} has BMI of {bmi} which is {bmi_category}". The BMI prime index is {bmi_prime}.".
4. In your own words, discuss the possible advantages and limitations of this measure.

4 Artefacts

The solution to the task must be included in a single Jupyter/IPython notebook (an .ipynb file) running against a Python 3 kernel.

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).

Imagine it is a report 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**, **email address**, and whether you are an **undergraduate (SIT220)** or **postgraduate (SIT731)** student.

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.).

Finally, submit **the version of the Jupyter/IPython notebook converted to a PDF file** (e.g., via *File* → *Export Notebook As* → *PDF* or convert to HTML and from that to PDF with your web browser; any method will do).

You do not need to submit the .ipynb file via OnTrack, but you must store it for further reference – a marking tutor might ask for it later, e.g., at the end of the trimester.

Checklist:

1. Header, introduction, conclusion (Markdown chunks).
2. Text divided into sections, comments and discussion 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 (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.)

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

5 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