"Simple Data Visualization"

Practical Assignment 3

Informatics 1 for Biomedical Engineering

Graz University of Technology

---

1. Tasks

1.1. Functions (1pt)

Your program should contain a proper main function.Further, use functions at any point where you think it makes sense.

1.2. Command Line Parameters (2pt)

Your program should accept a number of command line parameters. Again, use the `argparse`[1] package.

Implement the following parameters:

`-i / --in` Path to the input file. Default: `ass_2.p`

`-o / --out` Path to your outputs. Default: `.`

Naturally, you have to verify if the input file is present on the system. Else, your program should The same applies to the `--out` Parameter. Note that this should be a folder. Check this and if it is not the case again terminate the program with `exit()`

Use the default values for the parameters should the user not utilize them. You do not need to check if you have read/write access to the paths defined by the user!

1.3. Combine separated Records (8pt)

Use the pickle from Assignment 2 to load the data structure and search for separated records. These are identified through letters at the end oftheir names (e.g.: `drive17a` and `drive17b`. Combine all these separated sub-records to one combined record entry. You need to check if all parameters match (such as samples per frame). Should the parts not be compatible, then simple do not merge the pieces. Note, that some values need to be updated in the datastructure. Missing data (an empty list in the data field) should result in your program not merging the records.

Merge records ONLY, if there is an `a`-record (e.g.: `drive17a`) present and there is no missing piece. For example `a,b,c,d` can be combined, but not `a,c,d`, since the record `b` is missing

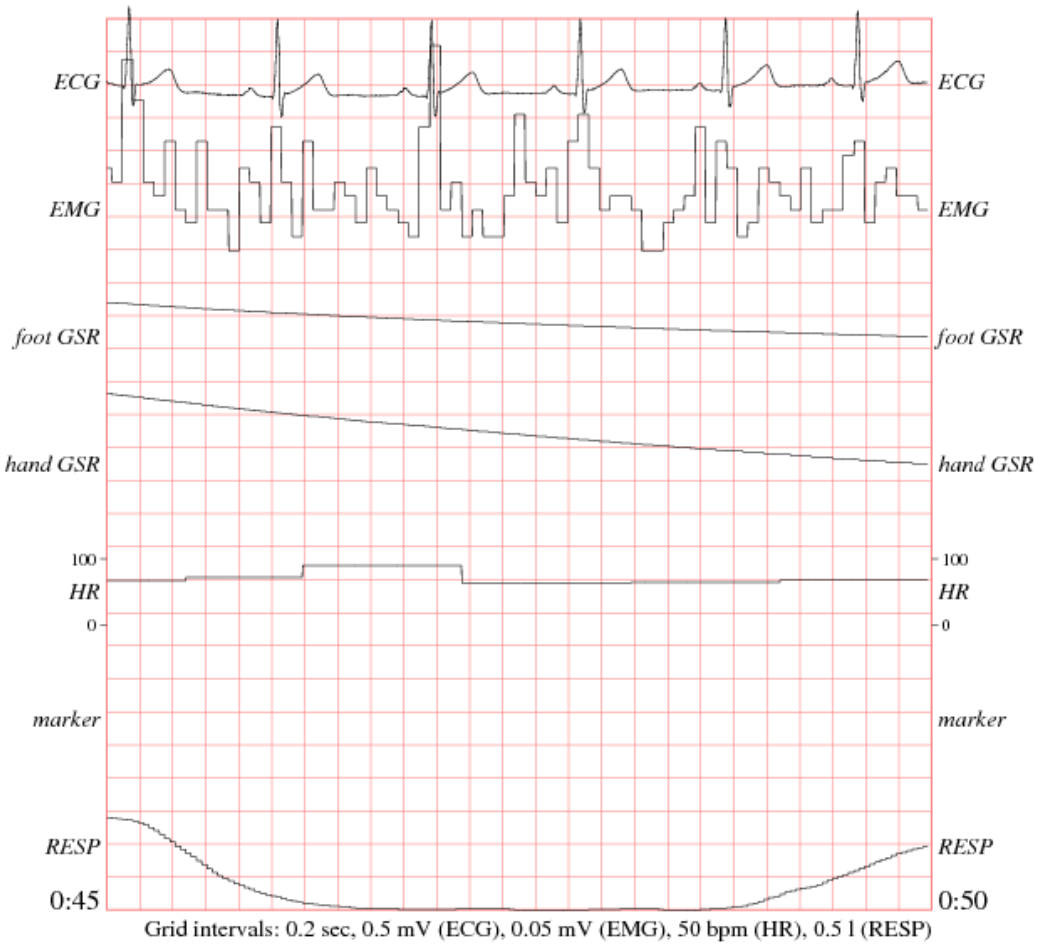
[1] https://docs.python.org/3.6/library/argparse.html

After you have successfully merged a record, remove all the sub-pieces from the data structure.

Finally, store the data structure under the name `ass_3.p` in the folder definied by the parameter `--out`.

1.4. Visualizing (14pt)

Use `matplotlib`[2] to visualize all records. Try to imitate the following style.



Grid intervals: 0.2 sec, 0.5 mV (ECG), 0.05 mV (EMG), 50 bpm (HR), 0.5 l (RESP)

You can ignore the sensor-specific information (Grid intervals: 0.2sec...) at the bottom of the example image.

Generate two plots for each record. First, visualise the entire drive and second, only show the timeframe from second 45 to 50. Save each plot both as `.png` and as `.pdf` (in total 4 plots per record). The names should be defined as follows: the name of the record + `_short` for the short timeframe and + `_full` for the visualisation of the full drive. Again, save them into the folder defined by `--out`.

Example: drive01 should result in the following files:

[2] http://matplotlib.org/

- drive01_full.png
- drive01_short.png
- drive01_full.pdf
- drive01_short.pdf

Only plot records containing at least one sensor (with data) and only plot those sensors where you have data loaded in your data structure. The use of the *seaborn*[3] is allowed.

The step structure for the line plots can be achieved using the `pyplot.step()`[4] function. Further, you can use `time.strftime()`[5] to convert indices to timestamps. Keep the signal frequence in mind here!.

Hint: When you plot the shorter timeframes, adapt the y_limites in such a way, that the data uses the as much space as possible on your figure.

1.5. BONUS: Visualize using bokeh(2pt)*

Create similar plots as you did in the last task. bit this time use the *bokeh*[6] package. Save the data in the `.html` format. Again, you should create 2 files per record named in a similar fashion. One `_full` and one `_short`. Save the plots in the directory defined by `--out`.

1.6. BONUS: Bokeh Hovertool(3pt)*

Use the bokeh hovertool[7] to create an overlay. There you should show the values for the sensors and the exact timestamp for the given mouse position. Use the `datasource` objects.

2. Packages

The following Python packages are allowed or required in this assigmnent.

- argparse, bokeh, collections, math, matplotlib, numpy, os, pickle, scipy, seaborn, struct, sys, time

3. Restrictions

- Do not add any bloat to your submission
- Use built-in functions where possible
- Do NOT load extra packages (no imports)

[3] http://seaborn.pydata.org/
[4] http://matplotlib.org/api/_as_gen/matplotlib.pyplot.step.html
[5] https://docs.python.org/3.6/library/time.html#time.strftime
[6] https://bokeh.pydata.org/en/latest/
[7] https://bokeh.pydata.org/en/latest/docs/reference/models/tools.html

4. File Headers

All Python source files have to contain a comment header with the following information:

- Author: – Your name
- MatNr: – Your matriculate number
- Description: – Purpose of the file
- Comments: – Any comments as to why if you deviate from the task or restrictions

Please copy and paste the example and change its contents to your data. (Depending on your PDF-viewer you may have to fix the whitespaces)!

Example Header:

```
##############################################################################
# Author:      Patrick Kasper
# MatNr:       0730294
# Description: The main file. Assignment only has 1 file...
# Comments:    This is the example comment. I just made it a bit
#              longer so it spans across multiple lines.
##############################################################################
```

5. Coding Standard

For this lecture and the practicals we use PEP 8 [8] as a coding standard guideline. Please note that whilst we do not strictly enforce all these rules we will not tolerate messy or unreadable code. Please concentrate in particular on the following aspects:

Language Programming is done in English. This is to ensure someone else at the other end of the world can read your code. Please make sure all sources you submit are thus written in English. This covers both variable names and comments!

Spaces, not tabs. Indent your files with 4 spaces instead of tabs. PEP 8 forces this in Python 3 (whilst allowing more freedom in Python 2). Most editors have an option to indent with 4 spaces when you press the tab button!

Descriptive names. Use descriptive names for variables where possible! Whilst a simple i can be enough for a simple single loop code can become very messy really fast. If a variable has no purpose at all, use a single underscore (_) for its name.

Max 72 characters. Do not have lines longer than 72 characters. Whilst PEP 8 allows 79 in some cases we ask you to use the lower cap of 72 characters per line. Please note that this includes indentation.

[8] https://www.python.org/dev/peps/pep-0008/

6. Automated Tests

Your file will be executed with the following command:

```
>python assignment_3.py <command line params>
```

Please make sure your submission follows all the restrictions defined in this document.

7. Submission

7.1. Deadline

12. January 2018 at 23:59:59

Any submission handed in too late will be ignored with the obvious exception of emergencies. In case the submission system is globally unreachable at the deadline it will be pushed back for 24 hours. If for whatever reason you are unable to submit your work, contact your tutor *PRIOR* to the deadline.

7.2. Uploading your submission

Assignment submissions in this course will always be archives. You are allowed to use .zip or .tar.gz formats.
In addition to your Python source files, please also pack a *readme.txt*. The file is mandatory but its contents are optional. In this file please write down how much time you spent on the assignment and where you ran into issues. Your feedback allows for immediate adjustments to the lecture and tutor sessions.
Upload your work to the Palme website. Before you do please double-check the following aspects:

- File and folder structure (see below)
- Comment header in every source file
- Coding Standard

7.3. Your Submission File

```
assignment_3.zip (or assignment_3.tar.gz)
    assignment_3.py
    readme.txt
```

Please do **NOT** submit the dataset or the input pickle. These paths will automatically be linked to your submission during tested!