Lecture 1 exercises

Introduction

In this exercise set, we will use the concepts covered in the first lecture.

To complete this exercise set, you will need the file *lecture_1_materials.zip* which can be downloaded from the course page on Ilias. These exercises require Numpy, SciPy and Matplotlib libraries installed in your computer. If you do not have them installed already, please check documentation web pages of these libraries for installation steps. You may use 'pip' or 'conda' which are useful package management tools for Python.

When needed, please use only these libraries for this exercise set. Solutions with other packages will not be accepted so you will not get any points.

The exercise will be marked as OK if you get 8 / 12 points or more. Points are only awarded for exercises where your code produce the expected result, and where you provide comments describing what the code does.

Be aware that the exercises build on top of each other, and that it may be easier to solve a later exercise if your first try to solve the ones before it.

If you have any questions, send an e-mail to *pinar.goektepe@inf.unibe.ch*. You should describe what you have done so far, and exactly why you believe you are stuck on a specific exercise.

Exercises must be handed in via e-mail to *pinar.goektepe@inf.unibe.ch*. Deliver your submission as a compressed file (zip) containing **one .py or .ipynb file with all the exercises**, as variables may carry over from one exercise to the next.

Deadline: 14:00, March 2

Exercises

Exercise 1

You will find a .mat file named *A*0001.*mat* in the materials for this lecture. This contains ECG data that you need for the exercises. io module of scipy package offers a function *loadmat()* to load .mat file in Python. You can use the following lines of code to load the data from file:

```
import scipy.io # import the io module of scipy

ecg_filename = "A0001.mat" # define filename

ecg_dict = scipy.io.loadmat(ecg_filename) # load file to python dictionary
```

scipy.io.loadmat() function will return a dictionary which is a data structure keeping the data as key-value pairs. In order to access the ECG data from the dictionary

```
ecg_data = ecg_dict['val'] # access data from the dictionary
```

You can use the provided code to load data which will be used in the next steps.

Exercise 1a

Gives 1 points

Load the *A*0001.*mat* file from the materials. Feel free to use the code provided for loading data. This file is part of dataset provided in George B. Moody PhysioNet Challenge. Please visit (https://moody-challenge.physionet.org/2021/) for more details about the dataset.

Find and print the number of leads (channels in ECG) and the number of time points in the data.

Hint : The data are 2 dimensional and have the shape of number of leads x number of time points.

Exercise 1b

Gives 3 point

Plot and visually check data of each lead over time to find a lead whose data visually looks good. Indicate with lead you chose and plot the data of that lead

using Python's matplotlib package.

Hint : Compute time by using sampling frequency of the ECG data (500 Hz) and the time points as following:

times = np.arange(number of time points) / frequency

Exercise 1c

Gives 3 point

Compute the mean and standard deviation over leads. Plot the mean and standard deviation around mean over time.

Exercise 2a

Gives 2 points

Find the peaks in the data of one lead. In this exercise, use the same lead that you have chosen in Exercise 1b.

Hint : Use find_peaks() function from signal module of scipy. Experiment with 'height' parameter to find R-peaks which have the highest amplitude in ECG data.

Exercise 2b

Gives 3 points

Plot the peaks you found in Exercise 2a together with the ECG data of chosen lead.