Homework #2: Describing Patterns in Accelerometer Data (100 points)

Can you identify a person's activity based on 3D accelerometer data? In this assignment, your job is to use Python to *process*, *visualize*, and *explore* the provided accelerometer dataset; your main task is to **manually** *identify* and *describe* general patterns that link *accelerometer* data to <u>at least</u> **four** different *activities*.

To clarify, you have <u>**two**</u> deliverables:

1. Python code to process and visualize the provided accelerometer data.

2. A *plain-English* document (with figures) that describes possible link(s) between *human-recognizable* **patterns** in the accelerometer data and <u>**4**</u> (or more) **activities**.

The **provided dataset** contains uncalibrated (raw) data from 3-axis accelerometers mounted on the chests of **15** *participants* during **7** different *activities*.

*   There are 15 different *csv* files (1.csv, 2.csv, …, 15.csv), one for each participant
    *   See **README.txt** for specific *csv* file structure and types of **activities**

EXAMPLE

The following figure shows normalized x, y, z accelerometer data (blue, green, red) for participant #14. There are several trends worth noting; for instance, how would *you* describe the similarities and differences in accelerometer data for each activity?
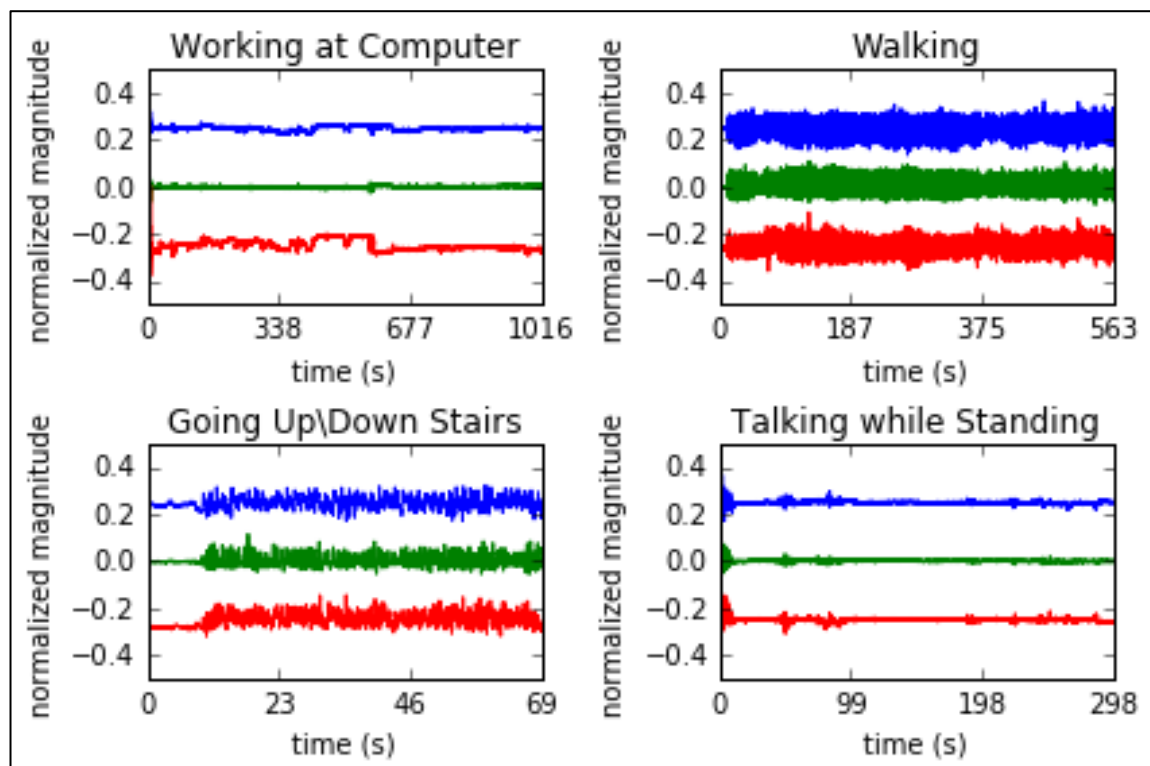
<u>HINTS</u>

- simple way to normalize vector **x** to ±1.0
  - ```
    x /= max( abs(x) )
    ```

- shift vector x to have **mean** *0.0*
  - ```
    x -= mean(x)
    ```

- ```
  import numpy as np
  ```
  - ```
    np.array( … )
    ```
  - ndarray *slice*
    - ```
      x = data[:,2]
      ```
  - ndarray *indexing*
    - ```
      i = (data[:,4]==1)
      ```

- ```
  import matplotlib as plt
  ```
  - ```
    plt.subplot( … )
    ```
  - ```
    plt.plot( … )
    ```
  - ```
    plt.xlabel( … )
    ```
  - ```
    plt.ylabel( … )
    ```
  - ```
    plt.title( … )
    ```
  - ```
    plt.axis( … )
    ```
  - ```
    plt.xticks( … )
    ```
  - ```
    plt.savefig( … )
    ```

- sypder app within the Anaconda Python package is very helpful

- Work incrementally!!