**Introduction to Programming – Fall 2017**

**Assignment 4**

**Notes:**

1. In order for it to be considered for grading, the program you submit **\*\*must\*\*** run and produce output. For example, we should be able to go to the shell/Terminal or PyCharm and execute your program to see the results. Partial grading (in case of partially correct answers) will only be done if this condition is met.
2. **Due date:** 10 Nov 2017, by 11:59 pm.

This final assignment is meant to be worked out as a group. Each group will have 2-3 members, and will submit 1 collective response in the form of a git repository on github that will contain all the sourcecode for the group. While you have 2 weeks to complete the assignment, **we strongly urge you to begin early and ask questions if you’re stuck, as this will be a totally new workflow for most of you.**

Contents of the assignment are in Assignment4.zip archive, which contains the following files:

1. an excel spreadsheet (06222016 Staph Array Data.xlsx) which contains the data to be analyzed
2. a text file (output-layout.txt) that lists all the columns which need to be plotted
3. a folder containing output plots for you to use as reference for what we expect (06222016-Staph-Array-Data.dir/)
4. a ‘.gitignore file’, which contains filename(patterns) to be ignored by git. Currently this has a pattern to ignore the .idea folder which is generated by Pycharm as well as .xml files which are also generated by pycharm. These are internal project management files that PyCharm uses, and .gitignore file makes sure they are not version controlled.

In order to begin, each group should do the following (on **one** machine):

1. Create a new Pycharm project within the same environment / interpreter that we created in class
2. Copy the contents of assignment4.zip into the project directory
3. Create a new GitHub repository (VCS->import into version control->share project on GitHub)
4. Give your repository a name and click share
5. Click Ok
6. Go to github.com and invite the other group members as collaborators to this project
7. The other group members should now create a Pycharm project by checking out from version control (The link to the github repository will be available on the github project site created in step 3 above)
8. Remember to change the interpreter to the one we created in class (settings->Project:’project\_name’->Project Interpreter-> select the correct one from the drop down-python version 3.6.1)

**Commenting your code is very important.** It will allow other members in your group to understand what you were trying to accomplish with your code. Use a # symbol to add comments, any characters after the # will be treated as a comment

Eg:

values = [1, 2, 3, 4]

new = []

for i in values: # iterate over all the values and multiply them by 2

new.append(i\*2)

----------------------------------------------------------------------------------------------------

#takes two values and returns the product

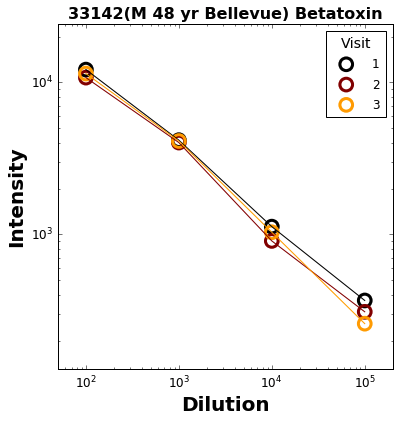
def multiply(num\_a, num\_b):

return num\_a \* num\_b

You are **required to comment** your code for this assignment.

The excel workbook you have been provided with consists of 11 worksheets named Plate1-11. This is a real-world dataset, coming from a real lab. As a result, formatting is messy, and requirements not completely specified (ask us for clarifications if you get stuck!!). You’ll need to do some data cleaning before starting to analyze and plot.

The first row of each sheet can be ignored. The first column, **Sample ID**, contains the **patient id**, **replicate/visit number** and **dilution (23234 V1 100).** Each row corresponds to a unique (replicate, visit, dilution) triplet. For ex., in Plate-1 corresponds to Visit 1 of the patient with id 23234, and dilution of 100. Note that these three values are not always consistent in the spreadsheet. The goal is to generate the plots listed in the output-layout file. Each patient id will result in a set of plots (one for each column listed in output-layout.txt), showing **dilution vs. <column intensity> (** bothlog normalized**).** In the plots,visits / replicates will be represented by line plots. For ex, the figure shown below plots the dilution vs. intensity of Betatoxin for 3 replicates of patient 33142 found on Plate 1.



------------------------------------------------------------------------------------------------------------------------------

Assignment4.zip files:

* .gitignore - This file tells git what file types or folders to ignore (.xml, .idea folder)
* Excel spreadsheet
* Layout File
* Output folder
  + index.html
  + ./Plate <1-11>
    - .png files for all plots for all patients in the plate
  + Plate <1-11> - <patient\_id>.html

-------------------------------------------------------------------------------------------------------------------------------

**Steps:**

1. Set up a git repository on GitHub. You only need one repository per group

* If you have questions, send a link to your github repository with your latest code pushed.
* Also check out “issue tracking” on GitHub. it is a very useful feature to keep track of outstanding issues, bugs, features to implement etc.

**Note:** You are expected to commit your changes at least a few times during your project at different stages of development.

1. Read the excel workbook (**Hint:** you may use pandas.read\_excel() method. In order to use it, you may have to install another dependency “xlrd”, which can be done by using PyCharm’s built in package manager (setting->Project:<project name>->Project Interpreter) or by running the cmd: pip install xlrd from the terminal. Don’t forget to activate your environment before running pip install.)
2. Parse the first column (**Sample ID**) into 3 separate columns. eg: ‘**23234 V1 100**’
   1. PatientID - **23234**
   2. Replicate/visit - **V1**
   3. Dilution- **100**

The data in this excel spreadsheet cannot be processed without first parsing column 1 properly. It is possible (and likely) that code you write for one Plate will break for others due to minor formatting differences. Your script should handle those differences.

**Tip:** Regular Expressions are a method of finding patterns in text. You will need to import the ‘**re**’ library to use them in your program. Taking the time to figure out how they work will simplify this step. While we haven’t covered this topic in class, the material we have covered should give you enough background to learn on your own (see resources below). Email Mark if you have any questions.

**Resources:**

Chapter 11 in “Python for Informatics” book

<https://docs.python.org/3/library/re.html>

<https://automatetheboringstuff.com/chapter7/>

* 1. Some formatting issues to watch out for:
     1. there are cases where the PatientID will be multiple words (eg:Plate 7)
     2. there are cases where the replicate/visit is not given (eg:Plate 3, row 42)

1. Plotting
   1. Line plots - line per replicate/visit
   2. Intensity of column listed in output-layoutfile ( both log normalized) VS Dilution. Note that Dilution is parsed from column 1 in step 5 above. For ex. one of the plots in Plate-1 will correspond to Patient 23234, for column PSMalpha2. The two lines in the plot will correspond to the following list of (x, y) coordinates:
      1. V1: (100, 959), (1000, 551), (10000, 169), (100000, 67)
      2. V2: (100, 1083), (1000, 609), (10000, 242), (100000, 79)

Since this assignment results in hundreds of plots being generated, you should think about whether you want these to be tracked by your git repository or not. If you do not, add the file extension pattern (ex. \*.png or \*.pdf) in the .gitignore file to ensure they will not be tracked.

1. Write a function to generate a properly formatted tab-delimited file per plate (11 files in total) that expands composite values like “Sample ID” into individual columns (PatientID, Visit, Dilution), so that future processing may be done easily. Also, fill in the empty cells in Hospital, Age and Gender columns with implied values.
2. You are expected to document your code. So add comments explaining what your code does.

There are many ways to write a script which will produce the correct output. You will be graded on whether all the plots are generated correctly, if your formatted output files are properly generated, how well your code is documented and your usage of git (ranked in that order)

The final submission should be a python script. Feel free to use jupyter notebooks to develop your solution but the contents of the notebook should be compiled into .py file(s) for your final submission.