# HCDE310: Interactive Systems Design & Technology

Human Centered Design & Engineering, University of Washington

# Homework 4: Functions and Output

**Due** Wednesday, October 22, 7:00a

## What to hand in via Canvas

- hw4-exercises.py program (all your code from part 1)
- hw4-feedprocessor.py program (all your code from part 2)
- the contributors.csv file generated in part 2
- an image of your plot from part 3

## Part 0: Download the starter code

You'll need the homework files. Get them by typing the following lines in terminal:

```
cd ~
./getHW.sh 4
```

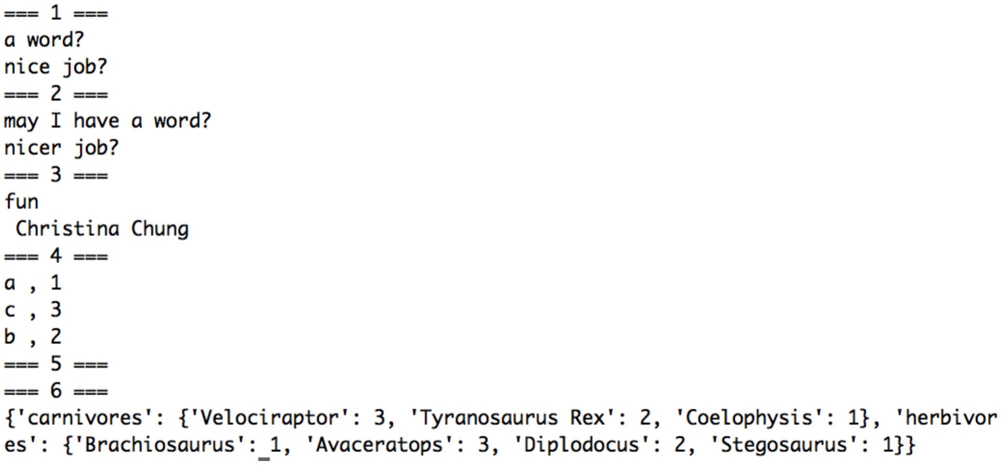
Don't forget that if you already have Eclipse open, you may need to right click on the homeworks project and "refresh" it, to make the new folder 4 appear.

### Part 1: Exercises

Open up your starter code and use it as a guide as you answer the exercise questions, which will give you some practice with defining and invoking functions.

Note: part 6 involves nested dictionaries. Similar exercises have tripped some students up in previous years. I recommend starting early and planning to talk through it with someone during class if you find yourself a bit stuck.

Your output should look like:

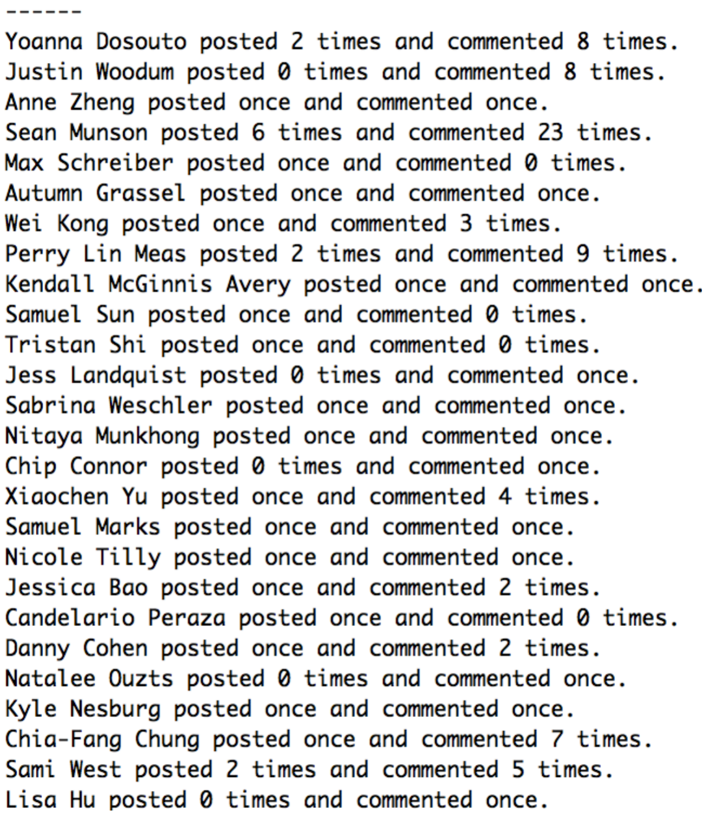


```
=== 1 ===
a word?
nice job?
=== 2 ===
may I have a word?
nicer job?
=== 3 ===
fun
 Christina Chung
=== 4 ===
a , 1
c , 3
b , 2
=== 5 ===
=== 6 ===
{'carnivores': {'Velociraptor': 3, 'Tyranosaurus Rex': 2, 'Coelophysis': 1}, 'herbivor
es': {'Brachiosaurus': 1, 'Avaceratops': 3, 'Diplodocus': 2, 'Stegosaurus': 1}}
```

## Part 2: Generalizing with functions

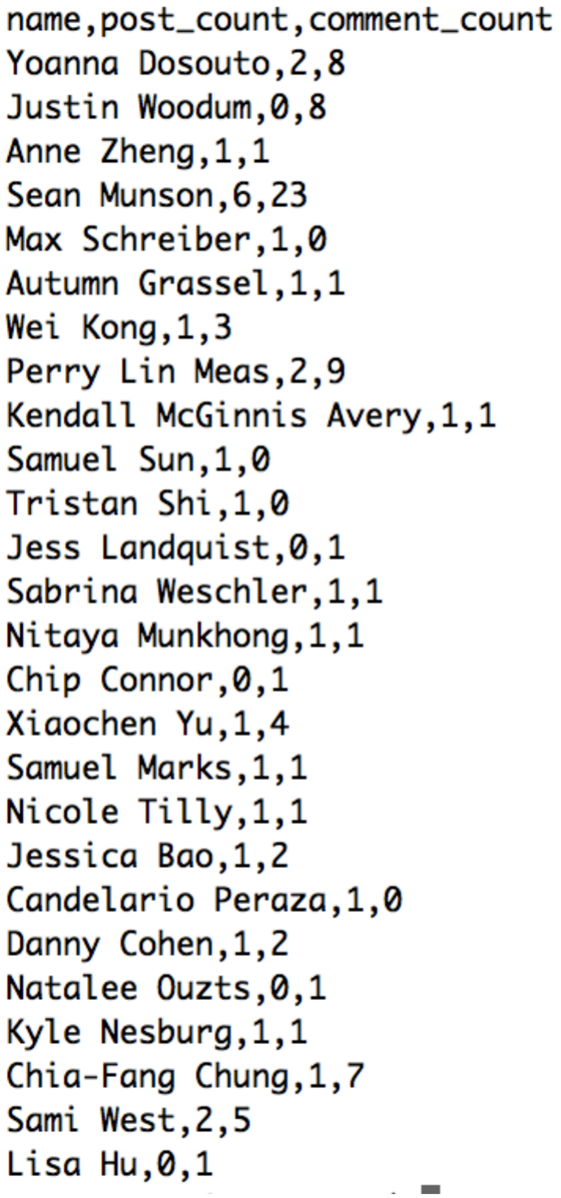
Your next task is to extend the functionality from last week's homework assignment using functions: count comments and posts separately per person, and output a CSV file. Follow the instructions in the code.

Your output should look like (order may vary)::



```
------
Yoanna Dosouto posted 2 times and commented 8 times.
Justin Woodum posted 0 times and commented 8 times.
Anne Zheng posted once and commented once.
Sean Munson posted 6 times and commented 23 times.
Max Schreiber posted once and commented 0 times.
Autumn Grassel posted once and commented once.
Wei Kong posted once and commented 3 times.
Perry Lin Meas posted 2 times and commented 9 times.
Kendall McGinnis Avery posted once and commented once.
Samuel Sun posted once and commented 0 times.
Tristan Shi posted once and commented 0 times.
Jess Landquist posted 0 times and commented once.
Sabrina Weschler posted once and commented once.
Nitaya Munkhong posted once and commented once.
Chip Connor posted 0 times and commented once.
Xiaochen Yu posted once and commented 4 times.
Samuel Marks posted once and commented once.
Nicole Tilly posted once and commented once.
Jessica Bao posted once and commented 2 times.
Candelario Peraza posted once and commented 0 times.
Danny Cohen posted once and commented 2 times.
Natalee Ouzts posted 0 times and commented once.
Kyle Nesburg posted once and commented once.
Chia-Fang Chung posted once and commented 7 times.
Sami West posted 2 times and commented 5 times.
Lisa Hu posted 0 times and commented once.
```

and your file contents should be (order may vary):



```
name,post_count,comment_count
Yoanna Dosouto,2,8
Justin Woodum,0,8
Anne Zheng,1,1
Sean Munson,6,23
Max Schreiber,1,0
Autumn Grassel,1,1
Wei Kong,1,3
Perry Lin Meas,2,9
Kendall McGinnis Avery,1,1
Samuel Sun,1,0
Tristan Shi,1,0
Jess Landquist,0,1
Sabrina Weschler,1,1
Nitaya Munkhong,1,1
Chip Connor,0,1
Xiaochen Yu,1,4
Samuel Marks,1,1
Nicole Tilly,1,1
Jessica Bao,1,2
Candelario Peraza,1,0
Danny Cohen,1,2
Natalee Ouzts,0,1
Kyle Nesburg,1,1
Chia-Fang Chung,1,7
Sami West,2,5
Lisa Hu,0,1
```

## Part 3: Graphing post and comment frequency

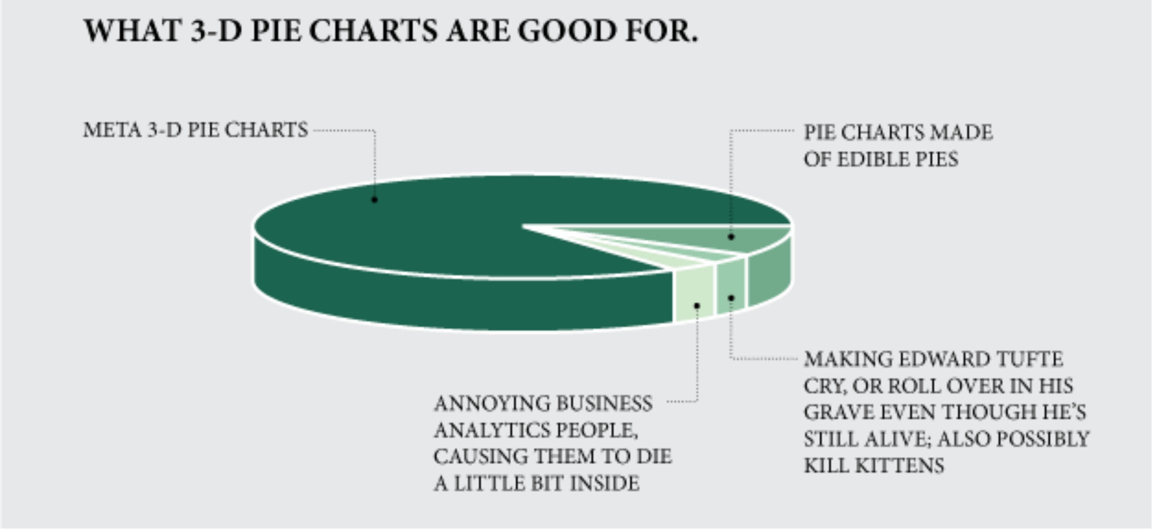
This next part guides you through making a graph of how many times each user posts to the Facebook group. If you do not have a Google account already, go to drive.google.com to register.

1) Go to drive.google.com in your web browser, log in, and click the 'Upload...' icon in the upper left hand corner, next to the Create button.
2) Click 'Files…' and select the 'contributors.csv' file located in the same directory as hw4.py. Then, click 'Start upload'
3) If prompted, select to have the file converted to the corresponding Google Docs format.

4) When the upload is successful, you'll see contributors.csv listed.
5) Click on contributors.csv. It should open up a spreadsheet of your data in a new window or tab.
6) Click "Open (Open with Google Sheets)" from the top menu bar if the file is not opened in a spreadsheet.
7) Click Insert -> Chart from the menu bar, and select a chart style that you think best communicates the data (see note below). When you've finished, click 'Insert'.
8) Edit the Chart title and axis titles as appropriate.
9) Click on the upper right hand corner (on the down arrow) of your newly inserted chart. Click 'Save Image'. The image will be downloaded to your computer.

No matter which format you chose, this isn't a particularly useful graph. For simple counts, tables are *often* (but not always) better: they have precise data (and can have more data), trust you to do the math, and take up less space.

Also, now is probably a good time to tell you that I have strong opinions about when pie charts are useful (almost never, but for a more balanced perspective, read http://eagereyes.org/techniques/pie-charts) and when 3D charts are useful (never).



source: http://flowingdata.com/2012/06/15/what-3-d-pie-charts-are-good-for/

This is not an infoviz class (that's HCDE 411), so I won't belabor the point, but pie charts tend to take up a lot of space and make exact comparison difficult for similarly sized pieces. 3D charts tend to distort the data: for example, in a 3D pie chart, the perspective effect will make the portion in the front appear larger – over-representing its weight – and the portions in the back appear smaller than they are – underrepresenting their weight.

## Just for fun: Plots in D3.

In this exercise, you can play a bit with D3, a rather powerful and beautiful visualization toolkit for the web, written in JavaScript.[1] The documentation is at http://d3js.org/. In the first weekend I tried it, I built a tool to graph my inbox (http://www.smunson.com/emailfull.php) and a tool to map my flights (http://www.smunson.com/thereandback/).

D3 includes some CSV parsing methods, so, now that we have CSV files, we can start displaying things. There's one catch: your browser's security settings will prevent JavaScript from loading the CSV files from your hard drive (details on this are at: https://github.com/mrdoob/three.js/wiki/How-to-run-things-locally). To enable that, we need to start a local, simple web server by following these steps:

1) Navigate to your homework directory by typing the following in terminal:

```
cd ~/Homeworks/hw4/
```

2) Start a simple web server by typing:

```
python -m SimpleHTTPServer
```

3) Now, leaving the terminal window open, you should be able to open Chrome (within Mint!) and navigate to http://localhost:8000/jff.html to open a plot based on contributors.csv.

4) You can edit the JavaScript in the HTML files to manipulate the graphs. This is an undirected exercise, but the code is fairly well commented (I hope!) and you can consult the D3 documentation if you want to get more daring and change the plot type or add other features (e.g., dynamically re-sort the graph by posts, comments, or contributor name; add a legend).

5) When you are done and want to close the web server, you can type Ctrl+C in the terminal window to stop Python.

[1] One of the creators, Jeff Heer, is now a professor in CSE here.