CSCA20 - Lab 8

Putting it all together

Learning Objectives

Now that we've covered all the core and advanced components, it's time to practice putting everything together in prepration for your projects. This lab has a lot to do, but you can focus on the parts you want to practice/demonstrate, so you have a good deal of flexibility.

Prelab

To make the most of your time in the lab, spend some time beforehand making some simple plots in matplotlib. Play around and make sure you're comfortable with looking up examples on the website and putting them into practice.

Demonstration & Evaluation

In this lab, you will have the opportunity to demonstrate competency and mastery in all components of the course. You can add or subtract components as necessary to get practice in any of the key areas, or to focus on the areas you want to practice ahead of your project.

The Scenario

The Universally Tallying Shopping Cart (UTSC) is an application that tracks purchases by users, and compares those purchases against that user's budget, with nice graphs to show who is buying what products, when they are buying them, and who is exceeding their weekly budget. Your job is to build this app.

Data Files

You have been provided with 3 example files:

- a budget file that consists of names of individuals and their weekly budget
- a price file that consists of items for sale and the cost of each item[1]
- a purchases file that shows who used the cart on each day, and which items they bought

Demonstrating Competency

In order to demonstrate competency in all areas, you will need to build an app that generates a menu, allows the user to choose files, produces appropriate data structures, and produces at least 2 simple graphs (e.g., total spent by day). Also, your code must make appropriate use of functions to break up tasks into smaller sections and avoid repeated code.
If you only wish to demonstrate competency in the core components of the course, you may do so by creating a simple menu that asks a user for the name of the purchases file, the name of a person, day of the week and a food item, and returns the number of copies of that item that the person bought on that day. (e.g., "On Thursday, Gertrude bought 3 apples"). You must also create and use at least one function in completing your task (we recommend a function that takes a line of text and a food, and returns the number of times that food appears on that line)

Mastery

This app will allow you to demonstrate mastery in all components:

- User I/O - have a robust and error correcting menu
- Variables, Loops, Selection, Files - allow the user to update budgets and prices without the need to re-read the purchases file
- Functions - provide well documented and re-usable functions that can be useful beyond the scope of this project (e.g., instead of building to create these specific dictionaries, build a function that can create generic dictionaries that fit certain criteria, or functions that convert strings prices to floats, etc)
- Sets/Dictionaries & External tools - Provide robust and interesting graphs to the users. (e.g., showing amount budgeted vs spent for each user, or number of purchases of each item across days of the week)

[1] As a 'behind the scenes' peek... I used random to build these files, so a lot of the prices won't make sense, but they're just examples

Hints

Here are a few hints that might help you with this assignment:

- This lab is purposely a bit more open ended than previous labs. We aren't going to tell you exactly what your graphs should look like or how you should store your data, but appropriately using dictionaries and functions will be necessary to complete the lab efficiently (and to demonstrate the relevant competencies)

- The files provided are just examples, your code should work with different names/items/ranges of prices/etc. You may assume that the days of the week always appear in order, but otherwise your code should work with any sensible data in the files

- Producing graphs is finicky. Matplotlib isn't the easiest tool to use, but on the plus side, odds are that any graph you want to make, someone else has already made and provided code on how they made it. So try to find examples you can tinker with to learn how it works rather than trying to learn every little detail of the tool.