Eric Heitman

Budgeroo: A Friendly Budgeting Tool

Final Project Assignment: Reflection

5/5/20

Budgeroo: A Friendly Budgeting Tool

Progress Summary:

From its initial planning stages, Budgeroo accomplished all of its safe goals and one of its stretch goals. Initially, the project's overall goal was to provide a user with a simple python program which would allow them to import their expenses from a .csv file, visualize their expenses via a pie chart with labels for each categories and percentages, identify their most frequent expenses and also create a budget for them if they did not have one. All of these goals were accomplished through the use of libraries such as Pandas and Matplotlib, which allowed for data framing, organization, and visualization. Additionally, I achieved one of my stretch goals of incorporating a GUI for the user to access all of these functions via buttons and entry boxes.

Development began on the project within the Pycharms IDE, a development tool for easy and efficient code debugging. After creating a new project folder and python file, I began to first work on the function to read in the .csv file. With a .csv file I had previously created with placeholder expense values in Google sheets, I leveraged the python library, Pandas, to read in this .csv file and format it into a dataframe.

With all of the data formatted, I could now proceed onto my first task of completing the first objective of the project: to create a pie chart which analyzes the percentage for each expense against their total of expenses for the month. To do this, I used the Matplotlib library and leveraged the pie() function to create a pie chart. I had to assign the proper labels for each expense, so I pulled this information from the dataframe I had created, along with displaying the total sum of expenses in each category in a legend. This step required me to ensure that similar

categories were not only summed once. For example, in the dataset I used the expense for "coffee" appeared 4 times. I had to ensure that the expense for "coffee" was not just summed from one of the entries, but all of the entries on the .csv file. This step proved to be a quick success.

Next, I began to work on the expense frequency analyzer. Using prior knowledge from using Matplotlib, I analyzed the data frame in a separate function call to find the count of the "Product" column in dataframe where the expenses resided. Once found, I created a simple line graph which included the expense name (x axis) and the frequency of purchase (y axis). After this was plotted, I included a statement in the supertitle of the chart displaying the most frequent purchase by leveraging the mode() function.

Once these two functions were designed, I began to work on the budget creation tool. I created a new window for the tool with two entry boxes underneath two simple questions the user could answer about their yearly salary and lifestyle. Once the submit button was pressed, the salary value was divided by 12 and assigned to a variable where it would be multiplied based on predetermined percentages tied to the lifestyle choice they typed. Once this was done, a pie chart appeared displaying their total salary in the supertitle, a pie chart containing labels of common expenses and the percentage they should be spending on each, and a legend displaying the maximum they can spend in each category monthly.

Detours:

Getting the ball rolling in a few of the major areas as far as establishing a familiarity with each libraries' respective functions and syntax was the first major hurdle I had to cross. By referencing official documentation available online for Pandas, Tkinter, and Matplotlib, I was

able to more or less find all of the answers to the questions I could not solve on my own in the realm of properly calling functions.

Once I had fundamental knowledge of how the functions I needed for this project worked, I ran into issues regarding the styling of the GUI and Matplotlib pie charts. The legends on the pie charts were not displaying correctly, and I had to find a solution to display them as to not interfere with the pie chart itself. Eventually I found simply assigning it a static place on the screen allowed to display properly in most cases. For the GUI, specifically not being able to successfully incorporate a background image on the main GUI window. Instead of focusing on this, I opted to set the background color to a static color as to not takeaway from other major objectives of the task. This was further justified by the GUI being a stretch goal in my initial proposal of the project.

Button functionality also stumped me on the budget creation tool. I initially did not write the program to include a GUI, but once I did, I had to reorganize my code into callable functions so the buttons would know what to produce when pressed. This proved to be quite the endeavor, and a result, produced more errors than I had anticipated. Most notably, I mistakenly was calling the function in the line of code where I place the button on the screen instead of where I create it. Without realizing all I had to do was move this line to the line above it, I went down the rabbithole and began researching things I did not need to, wasting about a day in the process.

Lessons Learned and Reflection

Some important lessons I learnt over the course of developing Budgeroo include paying attention to syntax, organizing code into functions first if it will incorporate any type of events,

and being more detailed with how I approach my searches online if I am struggling with a roadblock.

At the end of developing Budgeroo, I found that I still had a lot of ideas where I could take this project in the future. The initial stretch goal of incorporating a live search is still something I wish to pursue, which would allow a user to populate a .csv file with purchases made from Amazon.com, as well as improving the GUI's design, incorporating a file import feature into the GUI window. Another important goal would also include the ability to analyze other .csv files that do not have the template in which I provided the program with to create a dataframe.