Unit 3 | Assignment - Py Me Up, Charlie

Background

Well... you've made it!

It's time to put away the Excel sheet and join the big leagues. Welcome to the world of programming with Python. In this homework assignment, you'll be using the concepts you've learned to complete **2** Python Challenges, PyBank and PyPoll. Both of these challenges encompasses a real-world situation where your newfound Python scripting skills can come in handy. These challenges are far from easy so expect some hard work ahead!

Before You Begin

1. Create a new GitHub repo called `python-challenge`. Then, clone it to your computer.

2. Inside your local git repository, create a directory for both of the Python Challenges. Use folder names corresponding to the challenges: **PyBank** and**PyPoll**.

3. Inside of each folder that you just created, add a new file called `main.py`. This will be the main script to run for each analysis.

4. Push the above changes to GitHub.

PyBank

- In this challenge, you are tasked with creating a Python script for analyzing the financial records of your company. You will give a set of financial data called budget_data.csv. The dataset is composed of two columns: `Date` and `Profit/Losses`. (Thankfully, your company has rather lax standards for accounting so the records are simple.)

- Your task is to create a Python script that analyzes the records to calculate each of the following:

  - The total number of months included in the dataset
  - The total net amount of "Profit/Losses" over the entire period
  - The average change in "Profit/Losses" between months over the entire period
  - The greatest increase in profits (date and amount) over the entire period
  - The greatest decrease in losses (date and amount) over the entire period

- As an example, your analysis should look similar to the one below:

```
Financial Analysis
----------------------------
Total Months: 86
Total: $38382578
Average  Change: $-2315.12
Greatest Increase in Profits: Feb-2012 ($1926159)
Greatest Decrease in Profits: Sep-2013 ($-2196167)
```

- In addition, your final script should both print the analysis to the terminal and export a text file with the results.

PyPoll

- In this challenge, you are tasked with helping a small, rural town modernize its vote-counting process. (Up until now, Uncle Cleetus had been trustfully tallying them one-by-one, but unfortunately, his concentration isn't what it used to be.)

- You will be give a set of poll data called election_data.csv. The dataset is composed of three columns: `Voter ID`, `County`, and `Candidate`. Your task is to create a Python script that analyzes the votes and calculates each of the following:

  - The total number of votes cast
  - A complete list of candidates who received votes
  - The percentage of votes each candidate won
  - The total number of votes each candidate won
  - The winner of the election based on popular vote.

- As an example, your analysis should look similar to the one below:

```
Election Results
-------------------------
Total Votes: 3521001
-------------------------
Khan: 63.000% (2218231)
Correy: 20.000% (704200)
Li: 14.000% (492940)
O'Tooley: 3.000% (105630)
-------------------------
Winner: Khan
-------------------------
```

- In addition, your final script should both print the analysis to the terminal and export a text file with the results.

Hints and Considerations

- Consider what we've learned so far. To date, we've learned how to import modules like `csv`; to read and write files in various formats; to store contents in variables, lists, and dictionaries; to iterate through basic data structures; and to debug along the way. Using what we've learned, try to break down you tasks into discrete mini-objectives. This will be a *much*better course of action than attempting to Google Search for a miracle.

- As you will discover, for some of these challenges, the datasets are quite large. This was done purposefully, as it showcases one of the limits of Excel-based analysis. While our first instinct, as data analysts, is often to head straight into Excel, creating scripts in Python can provide us with more robust options for handling "big data".

- Your scripts should work for each dataset provided. Run your script for each dataset separately to make sure that the code works for different data.

- Feel encouraged to work in groups, but don't shortchange yourself by copying someone else's work. You get what you put in, and the art of programming is extremely unforgiving to moochers. Dig your heels in, burn the night oil, and learn this while you can! These are skills that will pay dividends in your future career.

- Start early, and reach out for help often! Challenge yourself to identify *specific*questions for your instructors and TAs. Don't resign yourself to simply saying, "I'm totally lost." Come prepared to show your effort and thought patterns, we'll be happy to help along the way.

- Always commit your work and back it up with GitHub pushes. You don't want to lose hours of your work because you didn't push it to GitHub every half hour or so.

    - **Commit often**.

Copyright