Student Name: Mano Velu

Batch: Nov-12-2019 to May-09-2020

Project: Python-Challenge

Due Date: 12-14-2019

**Table of Contents**

[1. Python-Challenge ReadMe 3](#_Toc27051983)

[2. Input files 6](#_Toc27051984)

[3. Design criteria and Assumptions 6](#_Toc27051985)

[Design for PyBank: 6](#_Toc27051986)

[Design for PyPoll: 7](#_Toc27051987)

[4. Metrics Definition 7](#_Toc27051988)

[PyBank 7](#_Toc27051989)

[PyPoll: 7](#_Toc27051990)

[5. Pseudo Code 8](#_Toc27051991)

[PyBank: 8](#_Toc27051992)

[PyPoll: 8](#_Toc27051993)

[6. Python code 8](#_Toc27051994)

[7. Code Validation & Testing 8](#_Toc27051995)

[8. Conclusion and Screenshots 9](#_Toc27051996)

[PyBank: 9](#_Toc27051997)

[PyPoll: 9](#_Toc27051998)

# Python-Challenge ReadMe

1. **# Python Homework - Py Me Up, Charlie**
2. **## Background**
3. Well... you've made it!
4. 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.
5. 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!
6. **### Before You Begin**
7. 1. Create a new repository for this project called `python-challenge`. **\*\*Do not add this homework to an existing repository\*\***.
8. 2. Clone the new repository to your computer.
9. 3. Inside your local git repository, create a directory for both of the  Python Challenges. Use folder names corresponding to the challenges: **\*\*PyBank\*\*** and  **\*\*PyPoll\*\***.
10. 4. 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.
11. 5. Push the above changes to GitHub or GitLab.
12. **## PyBank**
13. ![Revenue](Images/revenue-per-lead.png)
14. \* 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](PyBank/Resources/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.)
15. \* Your task is to create a Python script that analyzes the records to calculate each of the following:
16. \* The total number of months included in the dataset
17. \* The net total amount of "Profit/Losses" over the entire period
18. \* The average of the changes in "Profit/Losses" over the entire period
19. \* The greatest increase in profits (date and amount) over the entire period
20. \* The greatest decrease in losses (date and amount) over the entire period
21. \* As an example, your analysis should look similar to the one below:
22. ```text
23. Financial Analysis
24. ----------------------------
25. Total Months: 86
26. Total: $38382578
27. Average  Change: $-2315.12
28. Greatest Increase in Profits: Feb-2012 ($1926159)
29. Greatest Decrease in Profits: Sep-2013 ($-2196167)
30. ```
31. \* In addition, your final script should both print the analysis to the terminal and export a text file with the results.
32. **## PyPoll**
33. ![Vote-Counting](Images/Vote\_counting.png)
34. \* 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.)
35. \* You will be give a set of poll data called [election\_data.csv](PyPoll/Resources/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:
36. \* The total number of votes cast
37. \* A complete list of candidates who received votes
38. \* The percentage of votes each candidate won
39. \* The total number of votes each candidate won
40. \* The winner of the election based on popular vote.
41. \* As an example, your analysis should look similar to the one below:
42. ```text
43. Election Results
44. -------------------------
45. Total Votes: 3521001
46. -------------------------
47. Khan: 63.000% (2218231)
48. Correy: 20.000% (704200)
49. Li: 14.000% (492940)
50. O'Tooley: 3.000% (105630)
51. -------------------------
52. Winner: Khan
53. -------------------------
54. ```
55. \* In addition, your final script should both print the analysis to the terminal and export a text file with the results.
56. **## Hints and Considerations**
57. \* 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.
58. \* 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".
59. \* 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.
60. \* 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.
61. \* 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.
62. \* 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.
63. \* **\*\*Commit often\*\***.
64. **## Copyright**
65. Trilogy Education Services © 2019. All Rights Reserved.

# 2. Input files

(i) budget\_data: Contains sample financial data Profit/Loss for a certain company

(ii) election\_data: Contains the voter Id, Candidate name and county name

# 3. Design criteria and Assumptions

## Design for PyBank:

The solution is aimed at creating a process with the following criteria.

1. The code is structured to open the input file only once and the data is processed in sequence
2. The data is processed record by record
3. All the metrics – Total months, Sum of profit/loss and average of profit/loss and Greatest increase and decrease month-over-month is all calculated in single run
4. Design will work for any volume of data
5. None of the parameters have been hard coded
6. Results are written to a file and rendered on the screen

## Design for PyPoll:

The solution is aimed at creating a process with the following criteria.

1. The code is structured to open the input file once to find distinct candidates and once again for to process the data in sequence
2. The data is processed record by record
3. The following metrics are calculated: Total votes cast, votes secured by candidate, % votes secured and determine the winning candidate
4. Design will work for any volume of data
5. None of the parameters have been hard coded
6. Results are written to a file and rendered on the screen
7. While writing to a file append command is used to write and append mode is used
8. County information has not been used anywhere in the code since the results are determined based on popular votes

# 4. Metrics Definition

The definition of each of the metrics is calculated is defined as follows:

PyBank:

1. Total months: It is assumed no month is prepeared. Hence number of rows without header is the total number of months data is available
2. Total: The Profit/Loss is summed for all the months
3. Average change: Sum of (Difference in Profit/Loss between current month and previous month) over Total number of intervals, which is total month - 1
4. Greatest increase: Highest positive difference in Profit/Loss between current month and previous month
5. Greatest decrease: Highest negative difference in Profit/Loss between current month and previous month

## PyPoll:

1. Total votes: Total number of records without the header. It is assumed no duplicate voter Id exits.
2. Votes by candidates: Count of votes by Candidates
3. % Votes: Votes secured by candidate over Total votes expressed in %
4. Winner: Candidate with highest votes count

# 5. Pseudo Code

## PyBank:

1. Create path, create file handle and open the file to read input data
2. Read the header row
3. Read the first row and store the first month data
4. Start iteration using file “csvreader”
   1. Read row
   2. Increment month count
   3. Cumulate the profit/loss
   4. Find the difference between previous mon and current month
   5. Check if it is the highest thus far if yes store as highest
   6. Check if it is the lowest thus far, if yes store as lowest
5. Create path, create file handle and open the file to write output data

## PyPoll:

1. Create path, create file handle and open the file to read input data
2. Read the header row
3. Start read run 1: to get distinct candidates name and total votes
4. Determine the number of candidates
5. Start read run 2: check who won the vote and increment
6. Go to the next record till end of file (EOF)
7. Print results to screen
8. Create path, create file handle and open the file to write output results

# 6. Python code

The following files are attached in zipped format:

1. PyBank\_homework.py
2. PyPoll\_homework.py

# 7. Code Validation & Testing

PyBank: Has only 86 rows. Opened in Excel and data were validated.

PyPoll: The data was partially extracted to an excel and a sample csv file was created to test the code.

# 8. Conclusion and Screenshots

## PyBank:

The following metrics were calculated

\* The total number of months included in the dataset

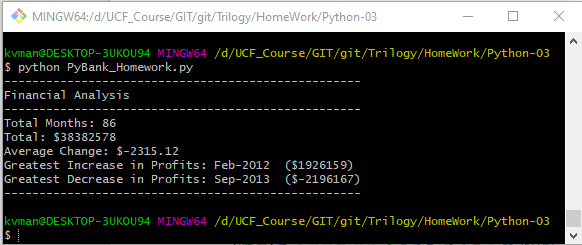
\* The net total amount of "Profit/Losses" over the entire period

\* The average of the changes in "Profit/Losses" 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

The final analysis looks as below: Results file created is attached.





## PyPoll:

The following metrics were calculated

\* \* 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.

The final analysis looks as below: Results file created is attached.

