**Python Challenge HW3**

# Python Challenge: PyBank and PyPoll

## Background

In this Python challenge, we will be analyzing financial records and election data using Python scripting. The goal is to apply your Python programming skills to real-world situations and solve the given tasks.

## Objective

The objective of this challenge is to create Python scripts that can analyze financial records (PyBank) and election data (PyPoll). For PyBank, calculate financial metrics such as total months, total profit/loss, average change, and greatest increase/decrease in profits. For PyPoll, calculate the total number of votes, the percentage of votes for each candidate, and determine the winner based on the popular vote.

## Dataset/File

- PyBank: The financial dataset is provided in a CSV file called budget\_data.csv. It contains two columns: "Date" and "Profit/Losses".

Date,Profit/Losses

Jan-10,1088983

Feb-10,-354534

Mar-10,276622

Apr-10,-728133

May-10,852993

- PyPoll: The election dataset is provided in a CSV file called election\_data.csv. It contains three columns: "Voter ID", "County", and "Candidate".

Ballot ID,County,Candidate

1323913,Jefferson,Charles Casper Stockham

1005842,Jefferson,Charles Casper Stockham

1880345,Jefferson,Charles Casper Stockham

1600337,Jefferson,Charles Casper Stockham

1835994,Jefferson,Charles Casper Stockham

## Dependencies

- Python 3.11

- CSV module

- OS module

- VS code (I used VS code to write/run python script)

## Installation Instructions

1. Clone the repository to local machine.

2. Navigate to the Python challenge directory.

3. Ensure that you have Python installed on your machine.

4. Make sure the file path is defined properly in the script

## Tasks to Run the Code

1. Run the main.py script for PyBank and PyPoll separately using the command `python main.py` or from VS code

## Usage

- Ensure that the CSV files (budget\_data.csv and election\_data.csv) are in the correct path, format and contain the required data.

- Run the main.py scripts to analyze the data and generate the results.

# Define the file path

#budget\_csv = os.path.join("..", 'Resources', "budget.csv")

#budget\_csv = os.path.join('/Users/pandari/Desktop/KP/edXData/UofM-VIRT-DATA-PT-03-2024-U-LOLC/python-challenge/PyBank/Resources/budget\_data.csv')

budget\_csv = os.path.join('python-challenge', 'PyBank', 'Resources', 'budget\_data.csv')

# Export results to a text file

#output\_file = os.path.join("..", 'analysis', "financial\_analysis.txt")

output\_file = os.path.join('python-challenge/PyBank/analysis/financial\_analysis.txt')

# Define the file path for election data

#election\_csv = os.path.join("..", 'Resources', "election\_data.csv")

#election\_csv = os.path.join('/Users/pandari/Desktop/KP/edXData/UofM-VIRT-DATA-PT-03-2024-U-LOLC/python-challenge/PyPoll/Resources/election\_data.csv')

election\_csv = os.path.join('python-challenge', 'PyPoll', 'Resources', 'election\_data.csv')

# Export results to a text file

#output\_file = os.path.join("..", 'analysis', "election\_results.txt")

output\_file = os.path.join('python-challenge/PyPoll/analysis/election\_results.txt')

## Deliverables

- For PyBank, the script will print the financial analysis results to the terminal and export them to a text file called financial\_analysis.txt in the analysis folder.

(base) pandari@Krishnas-MacBook-Pro UofM-VIRT-DATA-PT-03-2024-U-LOLC % /usr/local/bin/python3 /Users/pandari/Desktop

/KP/edXData/UofM-VIRT-DATA-PT-03-2024-U-LOLC/python-challenge/PyBank/main.py

Financial Analysis

-------------------------

Total Months: 86

Total: $22564198

Average Change: $-8311.11

Greatest Increase in Profits: Aug-16 ($1862002)

Greatest Decrease in Profits: Feb-14 ($-1825558)

Results exported to python-challenge/PyBank/analysis/financial\_analysis.txt

(base) pandari@Krishnas-MacBook-Pro UofM-VIRT-DATA-PT-03-2024-U-LOLC %

File output:

Financial Analysis

-------------------------

Total Months: 86

Total: $22564198

Average Change: $-8311.11

Greatest Increase in Profits: Aug-16 ($1862002)

Greatest Decrease in Profits: Feb-14 ($-1825558)

- For PyPoll, the script will print the election results to the terminal and export them to a text file called election\_results.txt in the analysis folder.

(base) pandari@Krishnas-MacBook-Pro UofM-VIRT-DATA-PT-03-2024-U-LOLC % /usr/local/bin/python3 /Users/pandari/Desktop

/KP/edXData/UofM-VIRT-DATA-PT-03-2024-U-LOLC/python-challenge/PyPoll/main.py

Election Results

-------------------------

Total Votes: 369711

-------------------------

Charles Casper Stockham: 23.05% (85213)

Diana DeGette: 73.81% (272892)

Raymon Anthony Doane: 3.14% (11606)

-------------------------

Winner: Diana DeGette

-------------------------

Results exported to python-challenge/PyPoll/analysis/election\_results.txt

(base) pandari@Krishnas-MacBook-Pro UofM-VIRT-DATA-PT-03-2024-U-LOLC %

File output:

Election Results

-------------------------

Total Votes: 369711

-------------------------

Charles Casper Stockham: 23.05% (85213)

Diana DeGette: 73.81% (272892)

Raymon Anthony Doane: 3.14% (11606)

-------------------------

Winner: Diana DeGette

-------------------------

## Code and Solution

- PyBank: The main.py script reads the budget\_data.csv file, calculates the required financial metrics, and prints/export the results.

import os

import csv

# Define the file path

#budget\_csv = os.path.join("..", 'Resources', "budget.csv")

#budget\_csv = os.path.join('/Users/pandari/Desktop/KP/edXData/UofM-VIRT-DATA-PT-03-2024-U-LOLC/python-challenge/PyBank/Resources/budget\_data.csv')

budget\_csv = os.path.join('python-challenge', 'PyBank', 'Resources', 'budget\_data.csv')

# Initialize variables

total\_months = 0

total\_profit\_loss = 0

previous\_profit\_loss = 0

profit\_loss\_changes = []

months = []

# Read the CSV file

with open(budget\_csv, 'r') as csvfile:

csvreader = csv.reader(csvfile, delimiter=',')

next(csvreader) # Skip header row

for row in csvreader:

# Extract data

date = row[0]

profit\_loss = int(row[1])

# Calculate total months and total profit/loss

total\_months += 1

total\_profit\_loss += profit\_loss

# Calculate profit/loss change

if total\_months > 1:

change = profit\_loss - previous\_profit\_loss

profit\_loss\_changes.append(change)

months.append(date)

previous\_profit\_loss = profit\_loss

# Calculate average change

average\_change = sum(profit\_loss\_changes) / (total\_months - 1)

# Find greatest increase and decrease in profits

greatest\_increase = max(profit\_loss\_changes)

greatest\_decrease = min(profit\_loss\_changes)

increase\_month = months[profit\_loss\_changes.index(greatest\_increase)]

decrease\_month = months[profit\_loss\_changes.index(greatest\_decrease)]

# Print results to terminal

print("Financial Analysis")

print("-------------------------")

print(f"Total Months: {total\_months}")

print(f"Total: ${total\_profit\_loss}")

print(f"Average Change: ${average\_change:.2f}")

print(f"Greatest Increase in Profits: {increase\_month} (${greatest\_increase})")

print(f"Greatest Decrease in Profits: {decrease\_month} (${greatest\_decrease})")

# Export results to a text file

#output\_file = os.path.join("..", 'analysis', "financial\_analysis.txt")

output\_file = os.path.join('python-challenge/PyBank/analysis/financial\_analysis.txt')

with open(output\_file, 'w') as txtfile:

txtfile.write("Financial Analysis\n")

txtfile.write("-------------------------\n")

txtfile.write(f"Total Months: {total\_months}\n")

txtfile.write(f"Total: ${total\_profit\_loss}\n")

txtfile.write(f"Average Change: ${average\_change:.2f}\n")

txtfile.write(f"Greatest Increase in Profits: {increase\_month} (${greatest\_increase})\n")

txtfile.write(f"Greatest Decrease in Profits: {decrease\_month} (${greatest\_decrease})\n")

print(f"Results exported to {output\_file}")

- PyPoll: The main.py script reads the election\_data.csv file, calculates the total votes, percentage of votes for each candidate, determines the winner, and prints/export the results.

import os

import csv

# Define the file path for election data

#election\_csv = os.path.join("..", 'Resources', "election\_data.csv")

#election\_csv = os.path.join('/Users/pandari/Desktop/KP/edXData/UofM-VIRT-DATA-PT-03-2024-U-LOLC/python-challenge/PyPoll/Resources/election\_data.csv')

election\_csv = os.path.join('python-challenge', 'PyPoll', 'Resources', 'election\_data.csv')

# Initialize variables

total\_votes = 0

candidate\_votes = {} # Dictionary to store candidate votes

# Read the CSV file

with open(election\_csv, 'r') as csvfile:

csvreader = csv.reader(csvfile)

next(csvreader) # Skip header row

for row in csvreader:

# Extract data

voter\_id, county, candidate = row

# Update total votes

total\_votes += 1

# Update candidate votes

if candidate in candidate\_votes:

candidate\_votes[candidate] += 1

else:

candidate\_votes[candidate] = 1

# Calculate the percentage of votes for each candidate

candidate\_percentages = {}

for candidate, votes in candidate\_votes.items():

candidate\_percentages[candidate] = (votes / total\_votes) \* 100

# Determine the winner

winner = max(candidate\_votes, key=candidate\_votes.get)

# Print results

print("Election Results")

print("-------------------------")

print(f"Total Votes: {total\_votes}")

print("-------------------------")

for candidate, votes in candidate\_votes.items():

print(f"{candidate}: {candidate\_percentages[candidate]:.2f}% ({votes})")

print("-------------------------")

print(f"Winner: {winner}")

print("-------------------------")

# Export results to a text file

#output\_file = os.path.join("..", 'analysis', "election\_results.txt")

output\_file = os.path.join('python-challenge/PyPoll/analysis/election\_results.txt')

with open(output\_file, 'w') as txtfile:

txtfile.write("Election Results\n")

txtfile.write("-------------------------\n")

txtfile.write(f"Total Votes: {total\_votes}\n")

txtfile.write("-------------------------\n")

for candidate, votes in candidate\_votes.items():

txtfile.write(f"{candidate}: {candidate\_percentages[candidate]:.2f}% ({votes})\n")

txtfile.write("-------------------------\n")

txtfile.write(f"Winner: {winner}\n")

txtfile.write("-------------------------\n")

print(f"Results exported to {output\_file}")

## Contributing

If you want to contribute to this project, please fork the repository, make your changes, and submit a pull request. Contributions are welcome!

## Credits

- The dataset used in this project is provided by edXData Bootcamp.

- Leveraged Google/ChatGPT, Copoilet as/where needed to develop/validate/troubleshoot code/data/functions.

## Conclusion

This Python challenge provides a practical exercise to apply Python programming skills to analyze real-world data. By completing this challenge, you will enhance your understanding of Python scripting and data analysis techniques.