Python Homework

# Python Homework - 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 the \*\*two\*\* Python Challenges, PyBank and PyPoll. Both of these challenges encompass 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 repository for this project called `python-challenge`.
   1. \*\*Do not add this homework to an existing repository\*\*.
2. Clone the new repository to your computer.
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\*\*.
4. Inside of each folder that you just created, add the following:
   1. A new file called `main.py`. This will be the main script to run for each analysis.
   2. A "Resources" folder that contains the CSV files you used. Make sure your script has the correct path to the CSV file.
   3. An "analysis" folder that contains your text file that has the results from your analysis.
5. Push the above changes to GitHub or GitLab.

## PyBank

![Revenue](Images/revenue-per-lead.png) 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.)

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

1. The total number of months included in the dataset
2. The net total amount of "Profit/Losses" over the entire period
3. The average of the changes in "Profit/Losses" over the entire period
4. The greatest increase in profits (date and amount) over the entire period
5. The greatest decrease in losses (date and amount) over the entire period
6. As an example, your analysis should look similar to the one below:

```text

Financial Analysis

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

Total Months: 86

Total: $38382578

Average Change: $-2315.12

Greatest Increase in Profits: Feb-2012 ($1,926,159)

Greatest Decrease in Profits: Sep-2013 ($-2196167)

```

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

## PyPoll

![Vote Counting](Images/Vote\_counting.png)

In this challenge, you are tasked with helping a small, rural town modernize its vote counting process.

You will be given 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:

1. The total number of votes cast
2. A complete list of candidates who received votes
3. The percentage of votes each candidate won
4. The total number of votes each candidate won
5. The winner of the election based on popular vote.
6. As an example, your analysis should look similar to the one below:

```text

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 your tasks into discrete mini-objectives. This will be a \_much\_ better course of action than spending all your time looking for a solution on Stack Overflow.
* 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".
* Write one script for each dataset provided. Run your script separately to make sure that the code works for its respective dataset.
* 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." If you need help, reach out because we're happy to help. But, come prepared and show us what you have done and your thought process.
* 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.
* Ensure your repository has regular commits (i.e. 20+ commits) and a thorough README.md file

Setting up the repository …

***How do I use Git and GitHub?***

[This handy video](https://www.youtube.com/watch?v=seICQOd2qsY) can walk you through the below steps:

1. Create a new repository
2. Copy the link
3. In terminal, cd into the directory where you want your repo to live
4. Clone the repo using git clone <repo name>
5. cd into the repo
6. Add a file
7. Stage the changes in your current working directory using git add .
8. Commit the changes using git commit -m "some message"
9. Push the changes using git push origin master
10. #Rodgers PyBank
11. #Import the operating sytem
12. import os
13. #Import the csv module
14. import csv
15. # Path to collect file from resources folder
16. csvpath = os.path.join('Resources', 'budget\_data.csv')
17. # Open file for interpretation and calculations
18. with open(csvpath) as csvfile:
19. csvreader = csv.reader(csvfile, delimiter=',')
20. #print(csvreader)
21. csv\_header = next(csvreader)
22. count = 0
23. netprofit = 0
24. bestmonth = 0
25. worstmonth= 0
26. bestname = 'Kent'
27. worstname = 'Matthew'
28. month2month = 0
29. priormonth = 0
30. bestprior = 0
32. #Loop through rows to compare and calculate
33. for row in csvreader:
34. #print(row)
35. count = count + 1
36. monthlyprofit = int(row[1])
37. netprofit = netprofit + monthlyprofit
38. if count == 1:
39. month2month = 0
40. bestmonth = 0
41. worstmonth = 0
42. priormonth = int(row[1])
43. bestprior = int(row[1])
45. if count >1:
46. month2month = month2month + int(row[1]) - priormonth
47. priormonth = int(row[1])
48. if count >1:
49. if int(row[1])- bestprior > bestmonth:
50. bestmonth = int(row[1]) - priormonth
51. bestname = str(row[0])
53. if int(row[1]) < worstmonth:
54. worstmonth = int(row[1])
55. worstname = str(row[0])
56. #Output section
57. print('  ')
58. print('  ')
59. print('~~~~~~~~~~~~~~~~~~~~~~~~~~~~')
60. print('Financial Analysis')
61. print('----------------------------')
62. print('Total Months: ' + str(count))
63. print('Total Profit for Period: $' + str(netprofit))
64. print(f'Average Monthly Change: $' + str(month2month/(count-1)))
65. print(f'Greatest Increase in Profits: $' + str(bestmonth) + ' in ' + bestname)
66. print(f'Greatest Decrease in Profits: $' + str(worstmonth) + ' in ' + worstname)
67. print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~kmr")
68. print('  ')
69. print('  ')

# Specify the file to write to

output\_path = os.path.join("..", "output", "new.csv")

# Open the file using "write" mode. Specify the variable to hold the contents

with open(output\_path, 'w') as csvfile:

    # Initialize csv.writer

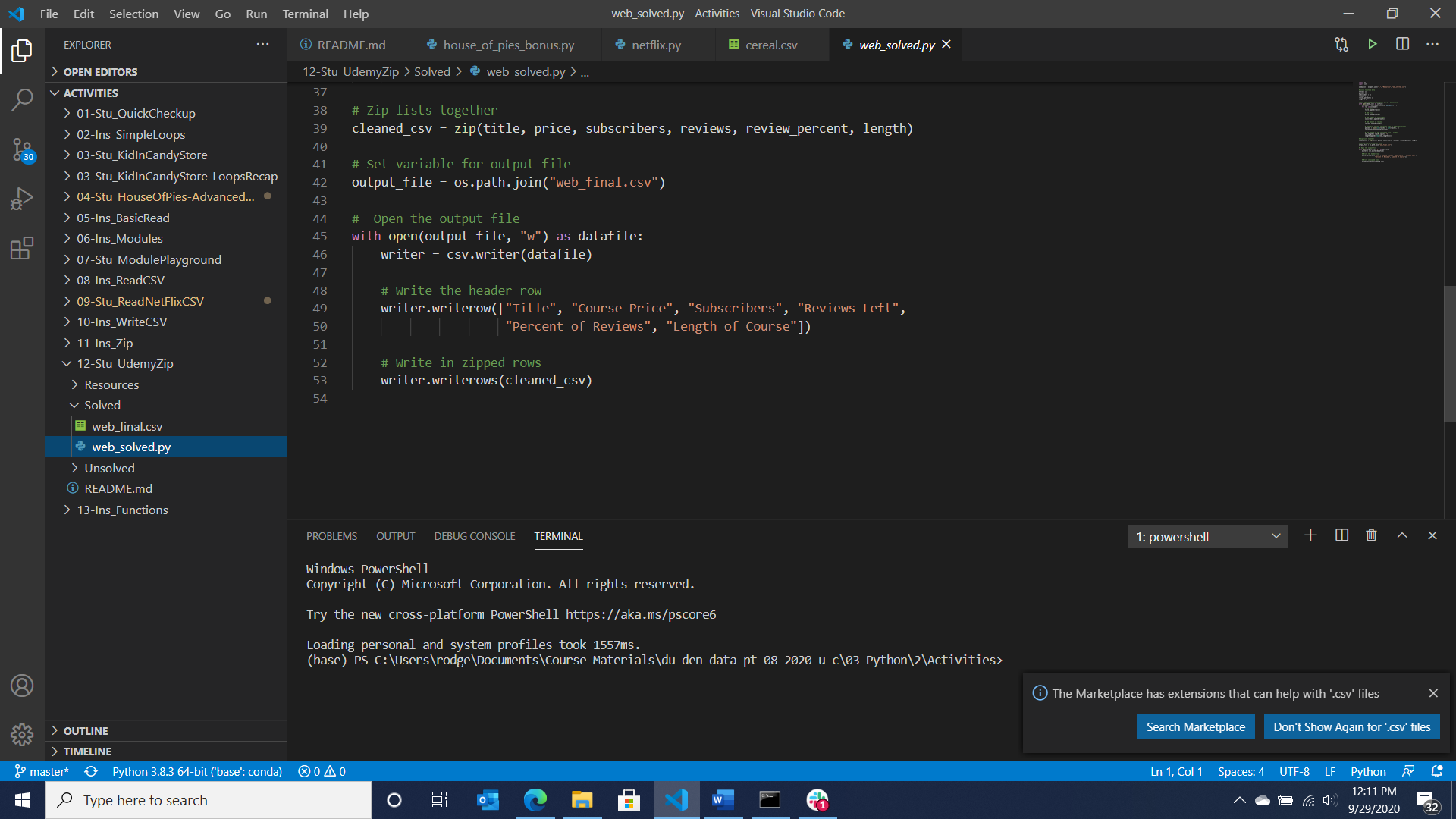
    csvwriter = csv.writer(csvfile, delimiter=',')

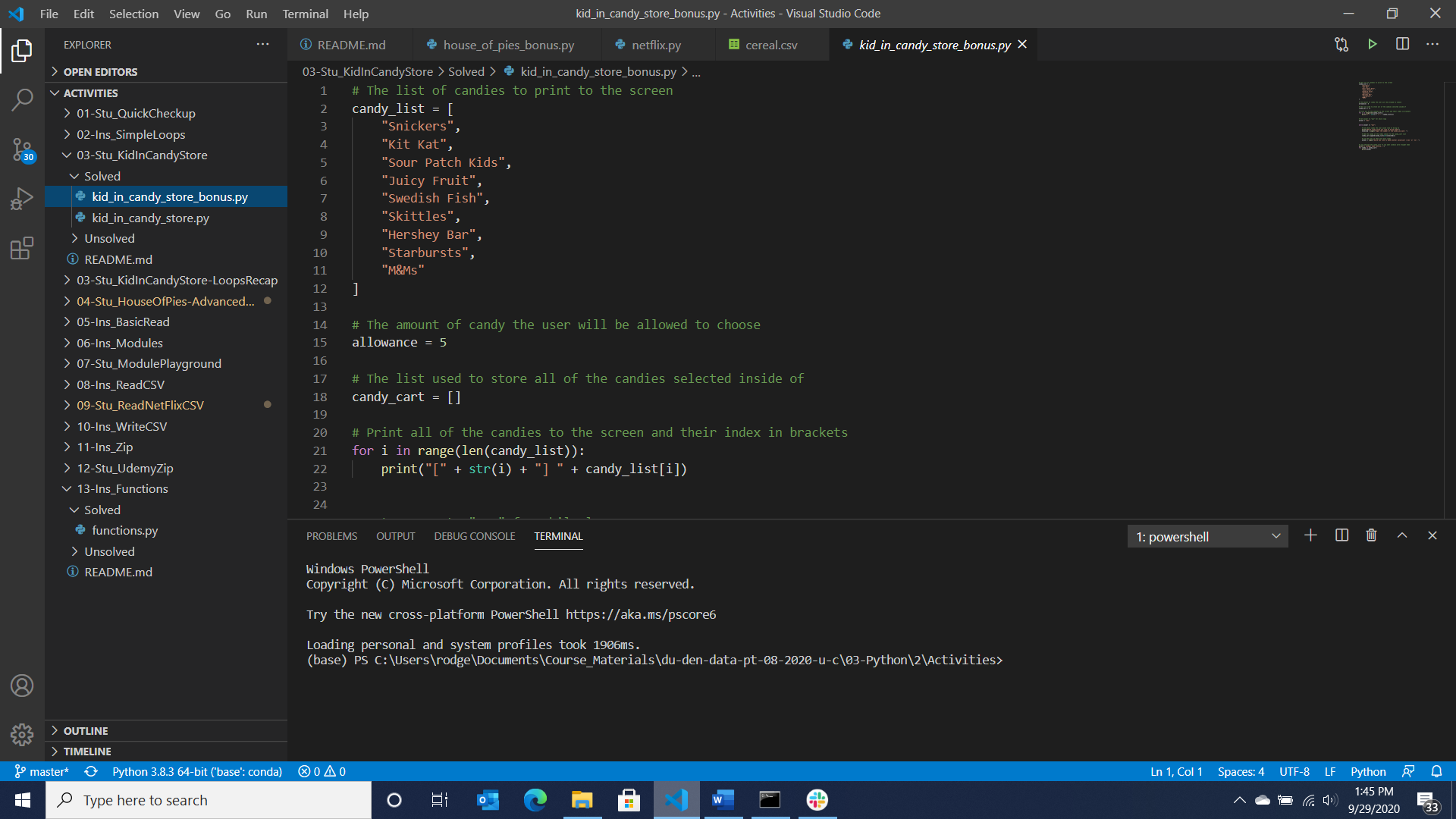
    # Write the first row (column headers)

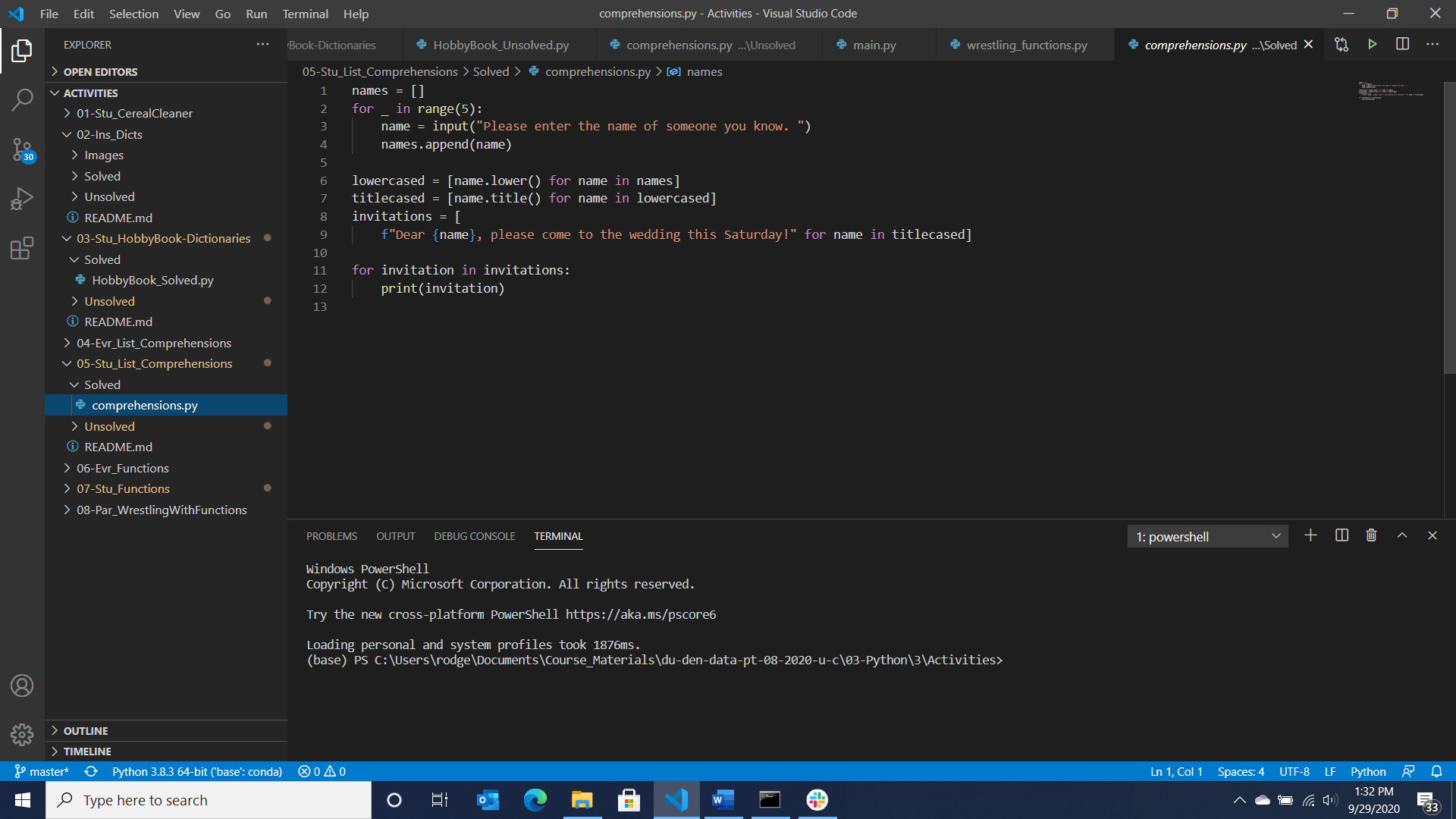
    csvwriter.writerow(['First Name', 'Last Name', 'SSN'])

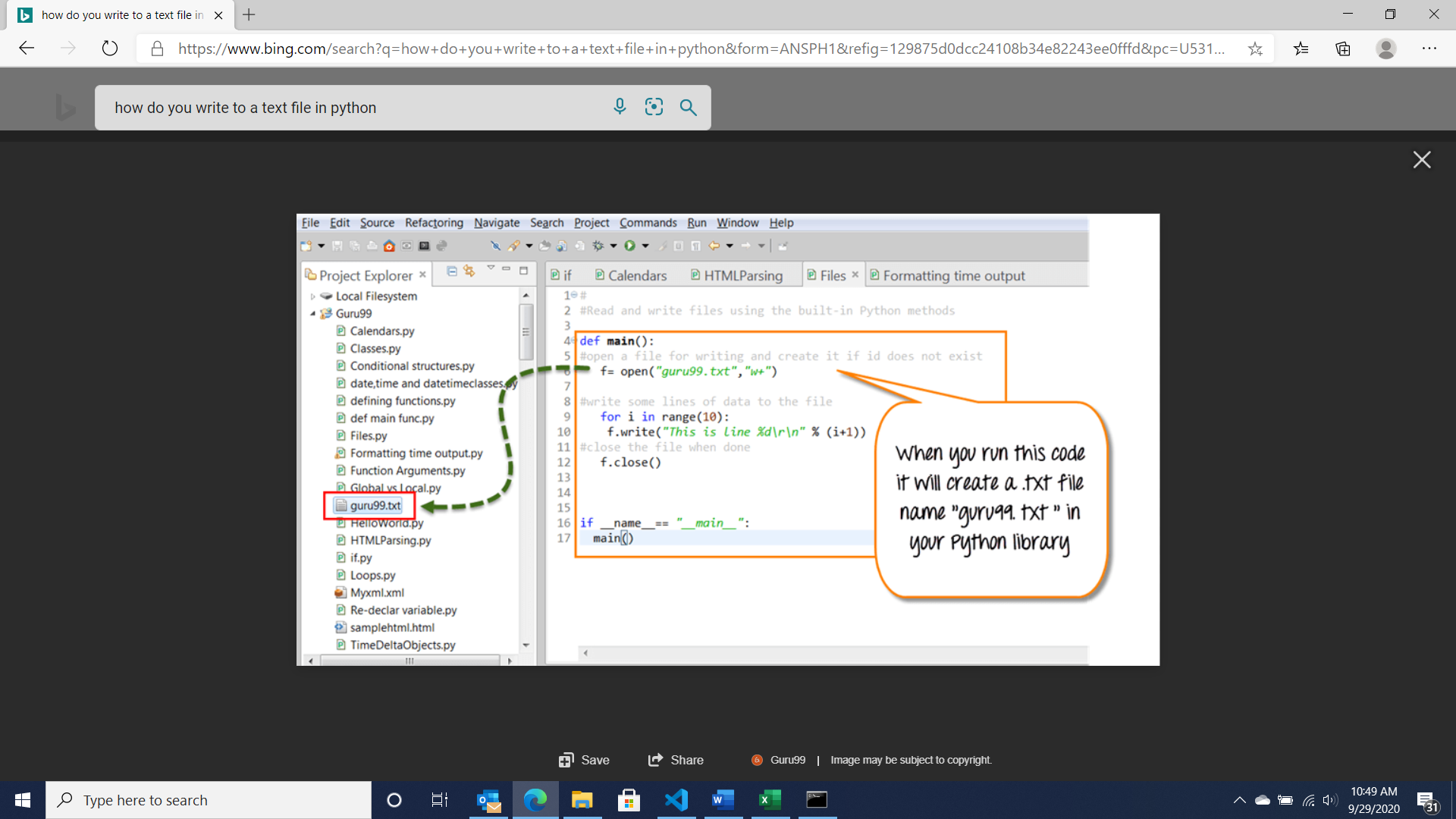
    # Write the second row

    csvwriter.writerow(['Caleb', 'Frost', '505-80-2901'])

x







Wed PM work to date with git issues …

#Rodgers PyPoll

import os

import csv

csvpath = os.path.join('Resources', 'election\_data.csv')

with open(csvpath) as csvfile:

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

    csvheader = next(csvreader)

    # Stuff that needs to be set outside the for loop

    votescast = 0

    khancounter = 0

    correycounter = 0

    licounter = 0

    otooleycounter = 0

    for row in csvreader:

        # First Goal ... Total Number of Votes Cast

        votescast = votescast + 1

        otooleycounter = votescast - khancounter - correycounter - licounter

        # Second Goal ... Complete list of candidates with votes

        if str(row[2])== 'Khan':

            khancounter = khancounter + 1

        if str(row[2])== 'Correy':

            correycounter = correycounter + 1

        if str(row[2])== 'Li':

            licounter = licounter + 1

    # Vote percentage calculations

    khanpercent = khancounter / votescast

    correypercent = correycounter / votescast

    lipercent = licounter / votescast

    tooleypercent = otooleycounter / votescast

    # Winner confirmation logic

    winnervotes = 0

    winnername = 'Kent'

    if khancounter > winnervotes:

        winnervotes = khancounter

        winnername = 'Khan'

    if correycounter > winnervotes:

        winnervotes = correycounter

        winnername = 'Correy'

    if licounter > winnervotes:

        winnervotes = licounter

        winnername = "Li"

    if otooleycounter > winnervotes:

        winnervotes = otooleycounter

        winnername = 'OTooley'

    # Really should have logic in here to deal with a tie

    print('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~')

    print(' ')

    print('Election Results')

    print('------------------------------')

    print('Total Votes: ' +str(votescast))

    print('------------------------------')

    print('Candidate Results as follows')

    print(f'Khan: ' + str(round((khanpercent\*100),1)) + '%   ' + str(khancounter) + '   Votes')

    print(f'Correy: ' + str(round((correypercent\*100),1)) + '%   ' + str(correycounter)+ '   Votes')

    print(f'Li: ' + str(round((lipercent\*100),1)) +'%        ' + str(licounter)+ '   Votes')

    print(f'OTooley: ' + str(round((tooleypercent\*100),1)) + '%   ' + str(otooleycounter) + '   Votes')

    print('------------------------------')

    print('Winner Is: ' + str(winnername) + '  with  ' + str(winnervotes) + '   Votes')

    print(' ')

    print('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~kmr')

#Rodgers PyBank September 2020

#Import the operating sytem

import os

#Import the csv module

import csv

# Path to collect file from resources folder

csvpath = os.path.join('Resources', 'budget\_data.csv')

# Open file for interpretation and calculations

with open(csvpath) as csvfile:

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

    #print(csvreader)

    csv\_header = next(csvreader)

    count = 0

    netprofit = 0

    # Counters for average monthly gain or loss

    month2month = 0

    priormonth = 0

    # Counters for greatest/worst gain/loss

    greatestname = 'Winner'

    worstname = 'Losser'

    #Loop through rows to compare and calculate

    for row in csvreader:

        count = count + 1

        monthlyprofit = int(row[1])

        netprofit = netprofit + monthlyprofit

        if count == 1:

            # Average monthly change counters

            month2month = 0

            priormonth = int(row[1])

            # Greatest monthly change counters

            greatestprior = int(row[1])

            greatestgain = 0

            greatestaward = 0

            # Worst monthly change counters

            worstprior = int(row[1])

            worstgain = 0

            worstaward = 0

        if count >1:

            month2month = month2month + int(row[1]) - priormonth

            priormonth = int(row[1])

        if count >1:

            greatestgain = int(row[1]) - greatestprior

            greatestprior = int(row[1])

            if greatestgain > greatestaward:

                greatestaward = greatestgain

                greatestname = str(row[0])

        if count >1:

            worstgain = int(row[1]) - worstprior

            worstprior = int(row[1])

            if worstgain < worstaward:

                worstaward = worstgain

                worstname = str(row[0])

    #Report-out section

    print('  ')

    print('  ')

    print('~~~~~~~~~~~~~~~~~~~~~~~~~~~~')

    print('Financial Analysis')

    print('----------------------------')

    print('Total Months: ' + str(count))

    print('Total Profit for Period: $' + str(netprofit))

    print(f'Average Monthly Change: $' + str(month2month/(count-1)))

    print(f'Greatest Increase in Profits: $' + str(greatestaward) + ' in ' + greatestname)

    print(f'Greatest Decrease in Profits: $' + str(worstaward) + ' in ' + worstname)

    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~kmr")

    print('  ')

    print('  ')

    # Lines to push the text information out to a text file

    # Specify what it is we want to output

    output=('~~~~~~~~~~~~~~~~~~~~~~~~~~~~'

    f'Financial Analysis')

    '----------------------------'

    'Total Months: ' + str(count)

    'Total Profit for Period: $' + str(netprofit)

    f'Average Monthly Change: $' + str(month2month/(count-1))

    f'Greatest Increase in Profits: $' + str(greatestaward) + ' in ' + greatestname

    f'Greatest Decrease in Profits: $' + str(worstaward) + ' in ' + worstname

    "~~~~~~~~~~~~~~~~~~~~~~~~~~~~kmr")

    print(output)

    # Specify the file to write to

    output\_path = os.path.join('Analysis', 'PyBankKMR.txt')

    # Open the output file to be written to as a text file; then write

    with open(output\_path, "w") as txt\_file:

        txt\_file.write(output\_path)

x