**Results when you execute Python scripts**

**pyBank**

# First we'll import the os module

# This will allow us to create file paths across operating systems

import os

# Module for reading CSV files

import csv

from statistics import mean

profit=0

months = []

list\_avg=[]

list\_profit=[]

list\_loss=[]

monthly\_changes =[]

losses = 0

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

# Method 2: Improved Reading using CSV module

with open(csvpath, newline='') as csvfile:

    # CSV reader specifies delimiter and variable that holds contents

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

    # Read the header row first (skip this step if there is no header)

    csv\_header = next(csvreader)

    for row in csvreader:

        months.append(row[0])

        list\_profit.append(int(row[1]))

    print("\nFinancial Analysis")

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

#The total number of months included in the dataset

    print("Total Month: %d" % len(months))

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

    print("Total: $%d" % sum(list\_profit))

 #The changes in "Profit/Losses" over the entire period, and then the average of those changes

    for i in range(len(list\_profit)-1):

        monthly\_changes.append(list\_profit[i+1]-list\_profit[i])

    print("Average Change: $%.2f" % mean(monthly\_changes))

#The greatest increase in profits (date and amount) over the entire period

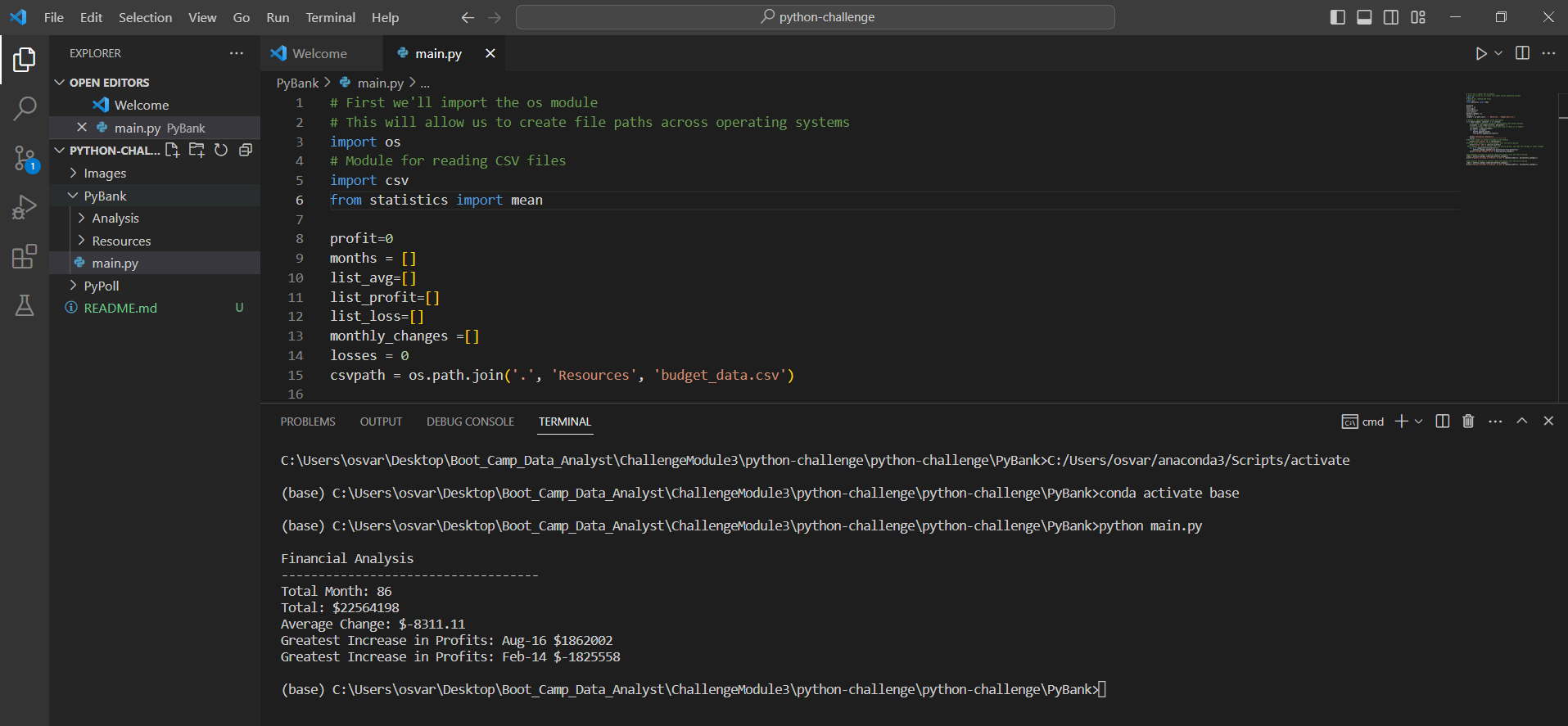
index = monthly\_changes.index(max(monthly\_changes))

print("Greatest Increase in Profits: %s $%s" % (months[index+1], max(monthly\_changes)))

#The greatest decrease in profits (date and amount) over the entire period

index = monthly\_changes.index(min(monthly\_changes))

print("Greatest Increase in Profits: %s $%s" % (months[index+1], min(monthly\_changes)))



**PYPoll**

# First we'll import the os module

# This will allow us to create file paths across operating systems

import os

# Module for reading CSV files

import csv

candidates = []

candidateVote=[]

total\_votes=[]

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

# Method 2: Improved Reading using CSV module

with open(csvpath, newline='') as csvfile:

    # CSV reader specifies delimiter and variable that holds contents

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

    # Read the header row first (skip this step if there is no header)

    csv\_header = next(csvreader)

    for row in csvreader:

       total\_votes.append(int(row[0]))

       candidates.append(row[2])

    uniqueCandidates =[]

    #extract unique candidates

    for i in range(0, len(candidates)-1) :

        if candidates[i] not in uniqueCandidates :

            uniqueCandidates.append(candidates[i])

    print("\nElection Results")

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

    #The total number of votes cast

    print("Total Votos: %d" % len(total\_votes))

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

    #From my unique candidate list count how many times apear to get Votes

    for i in range(0, len(uniqueCandidates)) :

        tmpVotes = 0

        prcnt = 0.0

        for j in range(0, len(candidates)) :

            if uniqueCandidates[i] == candidates[j] :

    #The total number of votes each candidate won

                tmpVotes +=1

    #The percentage of votes each candidate won

                prcnt = (tmpVotes /len(total\_votes)\*100)

        candidateVote.append((uniqueCandidates[i],tmpVotes, prcnt))

        print("%s: %.3f %% (%d) Votes" % (uniqueCandidates[i], prcnt, tmpVotes))

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

    #The winner of the election based on popular vote

    for i in range(0, len(candidateVote)-1) :

        winner = ""

        current =candidateVote[i]

        next =candidateVote[i+1]

        if int(current[1]) > int(next[1]) :

            winner = current[0]

        else :

            winner = next[0]

    print("Winner: %s" % winner)

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

