**PythonChallenge - PyPoll**

**PyPoll Assignment**

## 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](/ku-coding-boot-camp/KU-OVE-DATA-PT-01-2020-U-C/blob/master/02-Homework/03-Python/Instructions/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:
  + 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

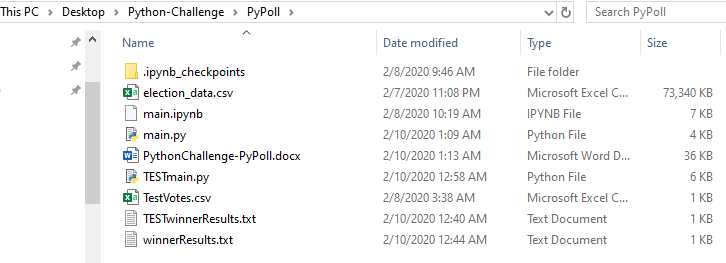
-------------------------

* 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**.

**Overview:**



**Python Code:**

C:/Users/VEA/Desktop/Python-Challenge/PyPoll/main.py

# ---------------------------------------------------------------

# -----    Python Challenge Homework Assignment #2        -------

# -----  02-10-2020                    Thomas A. VanEyck  -------

# ---------------------------------------------------------------

# -----handle dependencies

# ---------------------------------------------------------------

import os

import csv

# -----determine incoming file details

voter\_csv = os.path.join("election\_data.csv")

# -----create candidate votes Lists

VotesName = []

VotesCount = []

# -----------------------------------------------------------------

# -----   Write a function that assigns votes by candidate

# -----------------------------------------------------------------

def assignVotes(voteDetail,voteCands,votesCast,voteName,voteCount):

    # initialize

    voterId = str(voteDetail[0])

    voterCounty = str(voteDetail[1])

    Candidate = str(voteDetail[2])

    # --------------------------------

    cand = ""

    candIdx = 0

    voteCand = 0

    # -----Store Vote in Lists

    NewCandidate = True

    for cand in voteName:

        if cand == Candidate:

            NewCandidate = False

            break

        else:

            candIdx += 1

            NewCandidate = True

    # ----- Determine insert or update

    votesCast += 1

    if NewCandidate:

        voteName.insert(candIdx,Candidate)

        voteCount.insert(candIdx,1)

        voteCands += 1

    else:

        voteCand = voteCount[candIdx]

        voteCand += 1

        voteCount[candIdx] = voteCand

    return (voteCands, votesCast, voteName, voteCount)

# ---------------------------------------------------------------

# ------------------ Program Main Routine start  ----------------

# ---------------------------------------------------------------

totCands = 0

totVotes = 0

# -----Open and read csv

with open(voter\_csv, newline="") as csvfile:

    csvvoter = csv.reader(csvfile, delimiter=",")

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

    csv\_header = next(csvfile)

    print(f"Header: {csv\_header}")

    # -----Read through each row of data after the header

    for row in csvvoter:

        # -----determine where to accumulate with function assignVotes

        totCands,totVotes,VotesName,VotesCount = assignVotes(row,totCands,totVotes,VotesName,VotesCount)

# ---------------------------------------------------------------

# ---------------  Produce Report output to Terminal

# ---------------------------------------------------------------

WinCtr = 0

WinNm =""

WinPrct = 0

print ("")

Line1 = "-------------------------------------"

Line2 = "<<<<<<<<< Election Results >>>>>>>>>>"

Line3 = "  Total Votes Cast: " + str(totVotes)

print (Line1)

print (Line2)

print (Line1)

print (Line3)

print (Line1)

for voteIdx, voteNm in enumerate(VotesName):

    WinPrct = (VotesCount[voteIdx]/totVotes)

    Line4 = (str(voteNm) + ":     {:.3%}".format(WinPrct) + "      (" + str(VotesCount[voteIdx]) + ")")

    print (Line4)

    if VotesCount[voteIdx] > WinCtr:

        WinCtr = VotesCount[voteIdx]

        WinNm = voteNm

print (Line1)

Line5 = "  Winner: " + WinNm + " with " + str(WinCtr) + " votes"

print (Line5)

print (Line1)

# ---------------------------------------------------------------

# --------------------- write output file

# ---------------------------------------------------------------

WinPrct = 0

with open("winnerResults.txt","w") as writer\_file:

    writer\_file.write (Line1 + "\n")

    writer\_file.write (Line2 + "\n")

    writer\_file.write (Line1 + "\n")

    writer\_file.write (Line3 + "\n")

    writer\_file.write (Line1 + "\n")

    for voteIdx, voteNm in enumerate(VotesName):

        WinPrct = (VotesCount[voteIdx]/totVotes)

        Line4 = (str(voteNm) + ":     {:.3%}".format(WinPrct) + "      (" + str(VotesCount[voteIdx]) + ")")

        writer\_file.write (Line4 + "\n")

    writer\_file.write (Line1 + "\n")

    writer\_file.write (Line5 + "\n")

    writer\_file.write (Line1 + "\n")

**Terminal Output:**

