What You're Creating

This new assignment consists of two technical analysis deliverables and a written report to deliver your results. You will submit the following:

- Deliverable 1: The Election Results Printed to the Command Line
- Deliverable 2: The Election Results Saved to a Text File
- Deliverable 3: A written Analysis of the Election Audit (README.md)

Files

Use the following link to download the challenge starter code, which includes the Module 3 PyPoll solution.

Download challenge starter code

Links to an external site.

Instructions

Deliverable 1: Election Results Printed to the Command Line (60 points)

Using repetition statements, conditional statements with logical operators, and print statements, print out the candidate and county election results to the command line.

1. Download the `PyPoll_Challenge_starter_code.py` file and rename it `PyPoll_Challenge.py`.
2. Use the step-by-step instructions below to add code where indicated by the numbered comments in the starter code file.

Step 1:

- Initialize a county list, like the `candidate_options` list, that will hold the names of the counties.
- Initialize a dictionary, like the `candidate_votes` dictionary, that will hold the county as the key and the votes cast for each county as the values.

Step 2:

- Initialize an empty string, like `winning_candidate`, that will hold the county name for the county with the largest turnout.
- Initialize a variable, like the `winning_count` variable, that will hold the number of votes of the county that had the largest turnout.

Step 3:

- While reading the election results from each row inside the `for` loop, write a script that gets the county name from each row.

Step 4a:

- Write a decision statement with a logical operator to check if the county name acquired in Step 3 is in the county list you created in Step 1.

Step 4b:

- If the county is not in the list created in Step 1, add it to the list of county names like you did when adding a candidate to the `candidate_options` list.

Step 4c:

- Write a script that initializes the county vote to zero, like you did when you began to track the vote counts for the candidates.

Step 5:

- Write a script that adds a vote to the county's vote count as you are looping through all the rows, like you did for the candidate's vote count.

Step 6a:

- Write a repetition statement to get the county from the county dictionary that was created in Step 1.

Step 6b:

- Initialize a variable to hold the county's votes as they are retrieved from the county votes dictionary.

Step 6c:

- Write a script that calculates the county's votes as a percentage of the total votes.

Step 6d:

- Write a print statement that prints the current county, its percentage of the total votes, and its total votes to the command line.

Step 6e: ***This step will be completed in Deliverable 2.***

Step 6f:

- Write a decision statement that determines the county with the largest vote count and then adds that county and its vote count to the variables created in Step 2.

Step 7:

- Write a print statement that prints out the county with the largest turnout.

After you run your solution to Deliverable 1, confirm that the output to the command line matches the following image:

```
Election Results
-------------------------
Total Votes: 369,711
-------------------------

County Votes:
Jefferson: 10.5% (38,855)
Denver: 82.8% (306,055)
Arapahoe: 6.7% (24,801)

-------------------------
Largest County Turnout: Denver
-------------------------

Charles Casper Stockham: 23.0% (85,213)

Diana DeGette: 73.8% (272,892)

Raymon Anthony Doane: 3.1% (11,606)

-------------------------
Winner: Diana DeGette
Winning Vote Count: 272,892
Winning Percentage: 73.8%
-------------------------
```

Deliverable 2: Election Results Saved to a Text File (20 points)

Using your knowledge of writing data to a text file, write the winning candidate results and the county election results to the `election_results.txt` file.

Use the step-by-step instructions below to add code where indicated by the numbered comments in the starter code file.

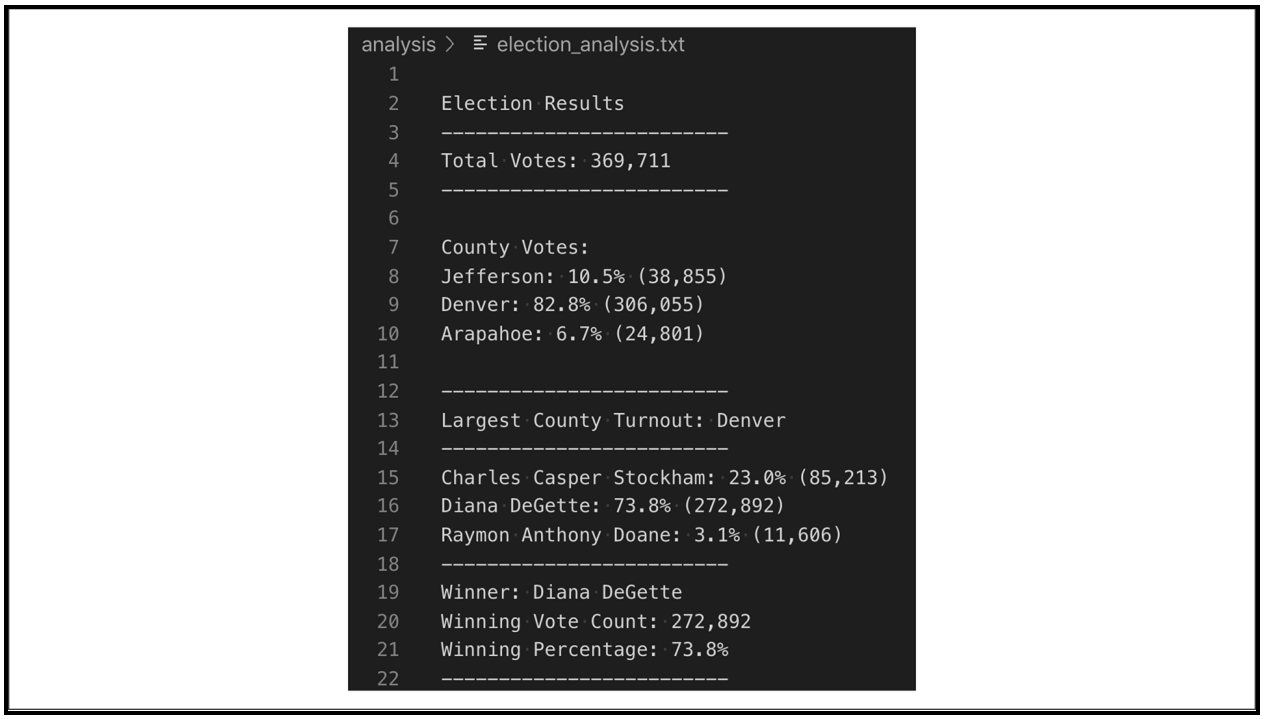
Step 6e:

- Write a script that saves each county, the county's total votes, and the county's percentage of total votes to the `election_results.txt` file.

Step 8:

- Write a script that saves the county with the largest turnout to the `election_results.txt` file.

After you run your solution to Deliverable 2, confirm that your `election_results.txt` file matches the following image:



Deliverable 3: Written Analysis of the Election Audit (20 points)

Use your repository README to write your analysis of Deliverables 1 and 2. The analysis should contain the following:

1. **Overview of Election Audit:** Explain the purpose of this election audit analysis.
2. **Election-Audit Results:** Using a bulleted list, address the following election outcomes. Use images or examples of your code as support where necessary.
    - How many votes were cast in this congressional election?
    - Provide a breakdown of the number of votes and the percentage of total votes for each county in the precinct.
    - Which county had the largest number of votes?

    - Provide a breakdown of the number of votes and the percentage of the total votes each candidate received.
    - Which candidate won the election, what was their vote count, and what was their percentage of the total votes?
3. **Election-Audit Summary:** In a summary statement, provide a business proposal to the election commission on how this script can be used—with some modifications—for any election. Give at least two examples of how this script can be modified to be used for other elections.

Requirements

Deliverable 1: Election Results Printed to the Command Line (60 points)

You will earn a perfect score for Deliverable 1 by completing all requirements below:

County data is extracted from the CSV file with code that performs the following (14 points)

- A list is initialized to store the county names and a dictionary is initialized to store the county votes. (Step 1) **(2 pt)**
- Two variables are initialized to store the county with the largest turnout and the largest vote count. (Step 2) **(2 pt)**
- The county name is extracted from each row and saved as a variable. (Step 3) **(2 pt)**
- An if statement checks if the county does not already exist in the county list. (Step 4a) **(2 pt)**

- County names are added to the county list only when the county does not exist in the county list. (Step 4b) **(2 pt)**
- County names and a zero vote count are added to the country dictionary only when the county does not exist in the county list. (Step 4c) **(2 pt)**
- County vote counts are updated in the county dictionary. (Step 5) **(2 pt)**

County results are printed to the terminal with code that performs the following (26 points)

- A for loop is created to loop through the county dictionary. (Step 6a) **(3 pt)**
- The county vote count is extracted from the dictionary and stored in a variable. (Step 6b) **(3 pt)**
- A calculation is performed to determine the percentage of votes in each county. (Step 6c). **(4 pt)**
- Each county and its total vote count are printed to the terminal. (Step 6d) **(3 pt)**
- Each county and its percentage of the total votes are printed to the terminal. (Step 6d) **(3 pt)**
- An if statement determines the winning county and stores its vote count. (Step 6f) **(5 pt)**
- The county with the largest number of voters is printed to the terminal. (Step 7) **(5 pt)**

County results are accurate (15 points)

- Each county and its total vote count printed to the terminal match the correct results. **(5 pt)**
- Each county and its percentage of the total votes printed to the terminal match the correct results. **(5 pt)**
- The county with the largest number of voters printed to the terminal is correct. (Denver) **(5 pt)**

Candidate results are printed to the terminal (5 points)

- Total Votes in the election are printed to the terminal. **(1 pt)**
- Each candidate's total votes and percentage of votes are printed to the terminal. **(2 pt)**
- The winner of the election, winning vote count, and winning percentage of votes are printed to the terminal. **(2 pt)**

  NOTE

  The starter code provided already prints these results to the terminal.

Deliverable 2: Election Results Saved to a Text File (20 points)

You will earn a perfect score for Deliverable 2 by completing all requirements below:

County Results saved to a Text File (15 points)

- Each county and its total vote count are saved in the `election_results.txt` file. (Step 6e) **(5 pt)**
- Each county and its percentage of the total votes are saved in the `election_results.txt` file. (Step 6e) **(5 pt)**

- The county with the largest number of voters is saved in the `election_results.txt` file. (Step 8) **(5 pt)**

Candidate Results saved to a Text File (5 points)

- Total Votes in the election are saved in the `election_results.txt` file. **(1 pt)**
- Each candidate's total votes and percentage of votes are saved in the `election_results.txt` file. **(2 pt)**
- The winner of the election, winning vote count, and winning percentage of votes are saved in the `election_results.txt` file. **(2 pt)**

  NOTE

  The starter code provided already saves these results to the text file.