Module 3 Challenge

New Attempt

Due Wednesday by 11:59pm **Points** 100 **Submitting** a text entry box or a website url

Background

Congratulations! You've helped Seth and Tom submit the election audit results to the election commission. But wait! The election commission has requested some additional data to complete the audit:

- The voter turnout for each county
- The percentage of votes from each county out of the total count
- The county with the highest turnout

Working from this module's `election_results.csv` file, use `for` loops and conditional statements with membership and logical operators to find the requested results. Then, print the results to the command line and save them to your `election_results.txt` file.

Finally, you'll provide a written analysis of the election audit for the election commission, including the new results and a clearly written overview of your methods. As with all written analyses, this will help your audience understand what you did and what they might be able to do with the data you presented.

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 (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/v2/module_3/PyPoll_Challenge_starter_code.py)

Deliverable 1: Election Results Printed to the Command Line (50 points)

Deliverable 1 Instructions

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

REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 3.2.2:** Run a Python file in the command line or VS Code.
- **Lesson 3.2.4:** Perform Calculations.
- **Lesson 3.2.5:** Create and add to a list.
- **Lesson 3.2.7:** Create and add keys and values to a dictionary.
- **Lesson 3.2.8:** Use decision statements to check a condition.
- **Lesson 3.2.9:** Apply membership and logical operators to decision statements.
- **Lesson 3.2.10:** Use repetition statements to iterate through a list or dictionary.
- **Lesson 3.2.11:** Write print statements using f-strings.

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 1 Requirements

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

Candidate Results

- Total Votes in the election are printed to the terminal. **(5 pt)**
- Each candidate's total votes and percentage of votes are printed to the terminal. **(5 pt)**
- The winner of the election, winning vote count, and winning percentage of votes are printed to the terminal. **(5 pt)**

County Results

- Each county and its total vote count are printed to the terminal. **(15 pt)**
- Each county and its percentage of the total votes are printed to the terminal. **(10 pt)**
- The county with the largest number of voters is printed to the terminal. **(10 pt)**

Deliverable 2: Election Results Saved to a Text File (30 points)

Deliverable 2 Instructions

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.

REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 3.2.2:** Run a Python file in command line or VS Code.
- **Lesson 3.2.10:** Write data to a file.

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

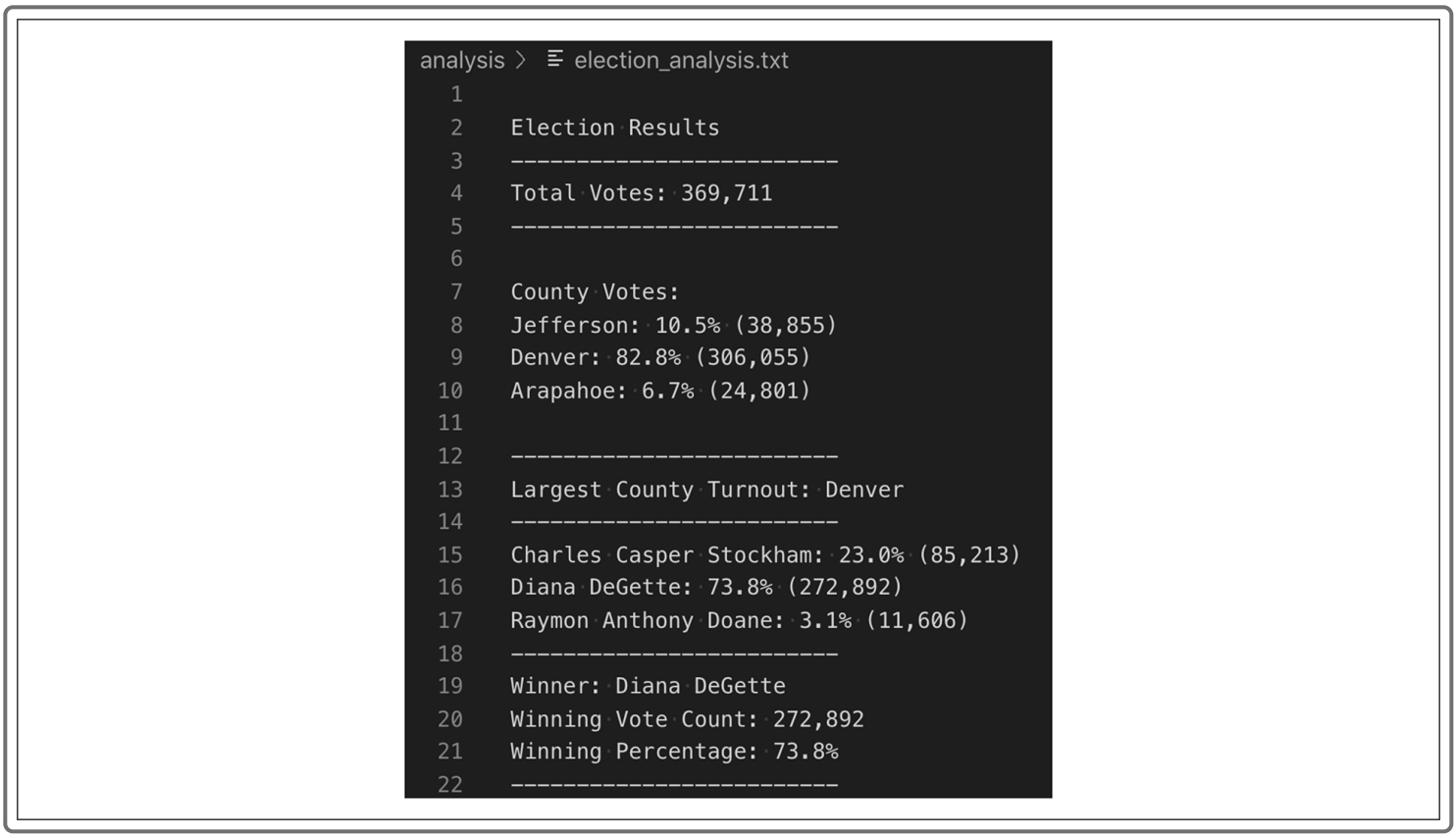
Step 6e:

- 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:



analysis > election_analysis.txt

```
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 Requirements

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

Candidate Results

- Total Votes in the election are saved in the `election_results.txt` file. **(2 pt)**
- Each candidate's total votes and percentage of votes are saved in the `election_results.txt` file. **(3 pt)**
- The winner of the election, winning vote count, and winning percentage of votes are saved in the `election_results.txt` file. **(5 pt)**

County Results

- Each county and its total vote count are saved in the `election_results.txt` file. **(10 pt)**
- Each county and its percentage of the total votes are saved in the `election_results.txt` file. **(5 pt)**
- The county with the largest number of voters is saved in the `election_results.txt` file. **(5 pt)**

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

Deliverable 3 Instructions

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.

Deliverable 3 Requirements

Structure, Organization, and Formatting (6 points)

The written analysis has the following structure, organization, and formatting:

- There is a title, and there are multiple sections. **(2 pt)**
- Each section has a heading. **(2 pt)**
- Links to images are working, and code is formatted and displayed correctly. **(2 pt)**

Analysis (14 points)

The written analysis has the following:

- Overview of Election Audit
    - The purpose of this election analysis audit is well defined. **(3 pt)**
- Election Audit Results
    - There is a bulleted list where each election outcome is addressed. **(7 pt)**
- Election Audit Summary

- There is a statement to the election commission that explores how this script can be used for any election, with two examples for modifying the script. **(4 pt)**

Submission

Once you're ready to submit, make sure to check your work against the rubric to ensure you are meeting the requirements for this Challenge one final time. It's easy to overlook items when you're in the zone!

As a reminder, the deliverables for this Challenge are as follows:

- 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)

Upload the following to your Election_Analysis GitHub repository:

1. The `PyPoll_Challenge.py` file
2. The analysis folder with the `election_results.txt` file
3. The Resources folder with the `election_results.csv` file

To submit your challenge assignment for grading in Bootcamp Spot, click Start Assignment, click the Website URL tab, then provide the URL of your Election_Analysis GitHub repository, and then click Submit. Comments are disabled for graded submissions in BootCampSpot. If you have questions about your feedback, please notify your instructional staff or the Student Success Manager. If you would like to resubmit your work for an improved grade, you can use the **Re-Submit Assignment** button to upload new links. You may resubmit up to 3 times for a total of 4 submissions.

IMPORTANT

Once you receive feedback on your Challenge, make any suggested updates or adjustments to your work. Then, add this week's Challenge to your professional portfolio.

NOTE

You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Next, and move on to the next Module.