# 3.5.2 Get the Candidates in the Election

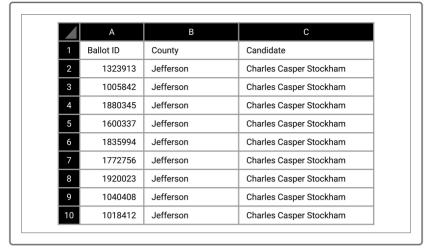
**Great** job getting the total votes in the election! Next, you will retrieve the names of the individual candidates in the election. This is important, because later you will need to determine how many votes each candidate received and their percentage of the total vote. So go grab some coffee (or your beverage of choice) and get ready to write some code with Tom.

Remember that when we inspected the data in election\_results.csv, using the first 10 rows and the last 10 rows, there were at two candidate names: Charles Casper Stockham and Raymon Anthony Doane.

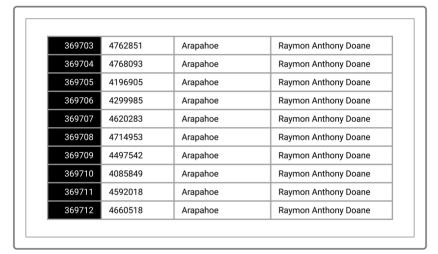
## **REWIND**

Here's what we found when we looked at the first 10 and last 10 rows of election\_results.csv:

# First 10 Rows



#### Last 10 Rows



You may have scrolled through the CSV file using Excel or VS Code and found three candidate's names. However, this method isn't very efficient, as it takes a long time to scroll through 369,712 rows. With Python, we can iterate through the rows in the CSV file and get the candidates from the "Candidate" column, and then add their names to a list.

## **REWIND**

Remember, when we iterated through the rows of <a href="mailto:election\_results.csv">election\_results.csv</a>, the last 10 rows that were printed to the

terminal had the structure of Python list.

```
'Arapahoe',
            'Raymon Anthony Doane'
'Arapahoe'
            'Raymon Anthony Doane
'Arapahoe',
            'Raymon Anthony Doane'
'Arapahoe',
            'Raymon Anthony Doane'
'Arapahoe',
            'Raymon Anthony Doane'
Arapahoe'
            'Raymon Anthony Doane
Arapahoe'
            'Raymon Anthony Doane
'Arapahoe',
            'Raymon Anthony Doane
'Arapahoe',
            'Raymon Anthony Doane'
'Arapahoe'.
            'Raymon Anthony Doane
```

To get the candidate from each list when we iterate through the row, we can use indexing on the for loop variable, row. The Candidate column is the third column that has the second index, so we would use, row[2] to reference the Candidate column.

Let's test this to make sure. Follow these steps:

- 1. Declare a new list, <a href="mailto:candidate\_options">candidate\_options</a> = [] by adding it before the <a href="with">with</a> <a href="mailto:open()">open()</a> statement in our script.
- 2. Add the following code to get the candidate's name from the row within the for loop.

```
# Print the candidate name from each row
     candidate_name = row[2]
```

3. Add the candidate\_name to the candidate\_options list using the append() method.

## **REWIND**

To add an item to a list, use the [append()] method.

4. Add a print statement that is flush with the left margin to print out the candidate\_options list.

Your file should look like this:

```
# Add our dependencies.
import csv
import os
# Assign a variable to load a file from a path.
file_to_load = os.path.join("Resources", "election_results.csv")
# Assign a variable to save the file to a path.
file_to_save = os.path.join("analysis", "election_analysis.txt")
# Initialize a total vote counter.
total votes = 0
# Candidate Options
candidate options = []
# Open the election results and read the file.
with open(file to load) as election data:
   file_reader = csv.reader(election_data)
    # Read the header row.
   headers = next(file reader)
   # Print each row in the CSV file.
    for row in file_reader:
        # Add to the total vote count.
        total_votes += 1
        # Print the candidate name from each row.
        candidate name = row[2]
        # Add the candidate name to the candidate list.
        candidate options.append(candidate name)
# Print the candidate list.
print(candidate options)
```

When we run this file, we will see all the elements, or candidates' names, from each row in the candidate\_options list:



However, we do not want every candidate from each row. Instead, we need to get only the unique candidate names.

To get the unique names in the <code>candidate\_options</code> list, we can use an <code>if</code> statement with the <code>not in</code> membership operator to check if the candidate has been added to the list. If the candidate's name has been added to the list, then the next time the candidate's name is found in a row using the <code>for</code> loop, it will not be added to the list.

# **SHOW HINT**

Inside the for loop, we will need to check if the candidate has been added to the candidate\_options list. Therefore, add the following code to PyPoll.py inside the for loop.

```
# If the candidate does not match any existing candidate...
    if candidate_name not in candidate_options:
        # Add it to the list of candidates.
        candidate_options.append(candidate_name)
```

The code in the for loop should look like this:

```
# Print each row in the CSV file.
    for row in file_reader:
        # Add to the total vote count.
        total_votes += 1

        # Print the candidate name from each row.
        candidate_name = row[2]

# If the candidate does not match any existing candidate...
        if candidate_name not in candidate_options:
```

Run the file in the VS Code terminal.



The output will be a list of the candidates in the election.

```
['Charles Casper Stockham', 'Diana DeGette', 'Raymon Anthony Doane']
```

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.