

Election Audit with python

1.Introduction

Voting is a process by which individuals express their opinions or preferences in an election or other decision-making process. Python is a powerful programming language that can be used to audit voting systems by analyzing and interpreting large amounts of data. Python's extensive libraries and frameworks, such as NumPy and pandas, make it well-suited for data analysis and visualization. Additionally, Python's simplicity and readability make it an accessible language for individuals and organizations who may not have a background in computer science. With Python, it is possible to create scripts and programs to perform audits on voting systems, such as checking for voter fraud or discrepancies in vote counts. Python can be used to perform an audit of voting results by analyzing data stored in a CSV (Comma Separated Values) file. One approach would be to use the csv library and OS library. which provides powerful tools for working with data in a tabular format. In this analysis read a CSV file of voting results and perform some analysis and write the results in to text file. This analysis generated the results of each country and county and largest turn over and each candidate votes counts and percentage and finally winner of the pole.

2. Method and Codes

First two libraries of csv and OS was imported.

```
# Add our dependencies.
import csv
import os
```

Then OS library used to define input files and output files path.

```
# Add a variable to load a file from a path.
file_to_load = os.path.join("Resources", "election_results.csv")
# Add a variable to save the file to a path.
file_to_save = os.path.join("analysis", "election_results.txt")
```

Initial variables was initiated for get total vote counter, Candidate Options , candidate votes, county list, county votes dictionary, winning candidate, vote count , percentage, largest county and county voter turnout.

```
# 1: Create a county list and county votes dictionary.

country_List = []
county_Votes = {}

# Track the winning candidate, vote count and percentage
winning_candidate = ""
winning_count = 0
winning_percentage = 0

# 2: Track the largest county and county voter turnout.

win_country_Name = ""
win_county_votes = 0
```

Then vote results data file opened and each column used to save into different lists.

```
with open(file_to_load) as election_data:
    reader = csv.reader(election_data)

    # Read the header
    header = next(reader)

    # For each row in the CSV file.
    for row in reader:

        # Add to the total vote count
        total_votes = total_votes + 1

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

        # 3: Extract the county name from each row.
        country_Name = row[1]
```

Then using this data set votes were counted.

```
if candidate_name not in candidate_options:

    # Add the candidate name to the candidate list.
    candidate_options.append(candidate_name)

    # And begin tracking that candidate's voter count.
    candidate_votes[candidate_name] = 0

# Add a vote to that candidate's count
candidate_votes[candidate_name] += 1

# 4a: Write an if statement that checks that the
# county does not match any existing county in the county list.
if country_Name not in country_List:

    # 4b: Add the existing county to the list of counties.
    country_List.append(country_Name)

    # 4c: Begin tracking the county's vote count.
    county_Votes[country_Name] = 0

# 5: Add a vote to that county's vote count.
county_Votes[country_Name] += 1
```

Then using counted results different analysis and results were generated.

```
# Save the results to our text file.
with open(file_to_save, "w") as txt_file:

    # Print the final vote count (to terminal)
    election_results = (
        f"\nElection Results\n"
        f"-----\n"
        f"Total Votes: {total_votes:,}\n"
        f"-----\n\n"
        f"County Votes:\n")
    print(election_results, end="")
```

```

txt_file.write(election_results)

# 6a: Write a for loop to get the county from the county dictionary.
for i in county_Votes:
    # 6b: Retrieve the county vote count.
    country_Vcount = county_Votes[i]

    # 6c: Calculate the percentage of votes for the county.
    per_Count = float(country_Vcount)/float(total_votes)*100

    # 6d: Print the county results to the terminal.
    country_Result = (f"{i}: {per_Count:.1f}% ({country_Vcount:,})")

    print(country_Result )
    # 6e: Save the county votes to a text file.
    txt_file.write(country_Result)
    txt_file.write("\n")

    # 6f: Write an if statement to determine the winning county and
get its vote count.
    if country_Vcount > win_county_votes:
        win_country_Name = i
        win_county_votes = country_Vcount

# 7: Print the county with the largest turnout to the terminal.
largest_turnout = (f"\n"
    f"-----\n"
    f"Largest County Turnout: {win_country_Name}\n"
    f"-----\n")

print(largest_turnout)

# 8: Save the county with the largest turnout to a text file.
txt_file.write(largest_turnout)

# Save the final candidate vote count to the text file.
for candidate_name in candidate_votes:

    # Retrieve vote count and percentage
    votes = candidate_votes.get(candidate_name)
    vote_percentage = float(votes) / float(total_votes) * 100
    candidate_results = (
        f"{candidate_name}: {vote_percentage:.1f}% ({votes:,})\n")

    # Print each candidate's voter count and percentage to the
    # terminal.
    print(candidate_results)
    # Save the candidate results to our text file.
    txt_file.write(candidate_results)

    # Determine winning vote count, winning percentage, and candidate.
    if (votes > winning_count) and (vote_percentage >
winning_percentage):
        winning_count = votes
        winning_candidate = candidate_name
        winning_percentage = vote_percentage

```

```

# Print the winning candidate (to terminal)
winning_candidate_summary = (
    f"-----\n"
    f"Winner: {winning_candidate}\n"
    f"Winning Vote Count: {winning_count:,}\n"
    f"Winning Percentage: {winning_percentage:.1f}%\n"
    f"-----\n")
print(winning_candidate_summary)

# Save the winning candidate's name to the text file
txt_file.write(winning_candidate_summary)

```

Finally result and analysis write into the text file.

3.Results analysis

- There was total 369,711 votes in the dataset
- Arapahoe got the least count of votes and it's 24,801 and as a percentage it's a 6.7% of total.
- Jefferson got the 38,855 votes as a percentage it's a 10.5% total and it's the second most in votes.
- Finally, Denver has the most votes country and it's having 306,055 and it was 82.8% of total. Also, it was the largest county turnout.
- Among candidates Raymon Anthony Doane got the least votes and it was 11,606 and as a percentage it is 3.1% from total.
- Charles Casper Stockham got the second most votes and it is 85,213 and as a percentage it is a 23% of total votes.
- Diana DeGette was the most votes gainer and it was 272,892 and it is 73.8% from total votes.
- From these results Diana DeGette selected as a winner.

4.Conclusion

As a conclusion this can be use for the audit the larger dataset of votes in the country and this can be modified with using NumPy or pandas packages for the larger datasets and those packages are really speed that will increase the efficient of the program. Also this can modified into machine learn method to identify possible votes pattern and that can help to investigate possible frauds in voting results.