User and Maintenance Guide

Steven Bower, Nate Dunlap, Matthew Luczaj, and Reese Wells

Introduction

This software is intended to help you visualize presidential election results. It's made to coincide with the votes casted in Ohio for the 2016 presidential race, but you're not limited to that, and we hope you'll find it scales well to larger amounts of data. You can load the data from a directory of CSV files, and the program will automatically import all the data for you and detect any errors that may be present. It lets you select a variety of different views of the data and show them as either a pie chart, a bar chart, a textual summary, or any combination of these. We hope you'll find this program to be a friendly but powerful way of looking through election results. Good luck, and we hope you have as much fun using this program as we did making it!

Installation Guide

Before you use this program, you'll need a directory containing at least one election results file. An election results file looks like this:

```
Butler,Locust Grove,78,0,275
Butler,Green,58,0,89
Butler,Liberty,75,0,268
Carroll,BrattonW2,58,1,340
Carroll,Locust Grove,78,0,275
Carroll,Green,58,0,89
Carroll,Liberty,75,0,268
Clark,BrattonW92,58,1,340
Clark,Locust Grove,78,0,275
Clark,Green,58,0,89
Clark,Liberty,75,0,268
Delaware,Bratton,58,1,340
Delaware,Locust Grove,78,0,275
Delaware,Green,58,0,89
Delaware,Liberty,75,0,268
Logan,Perry,98,0,75
```

Each line corresponds to one precinct within a county. It contains (1) the county name; (2) the precinct code; (3) the number of republican votes; (4) the number of democratic votes; and (5) the number of independent votes in that precinct.

You may also include a registered voters file in the directory. A registered voters file looks like this:

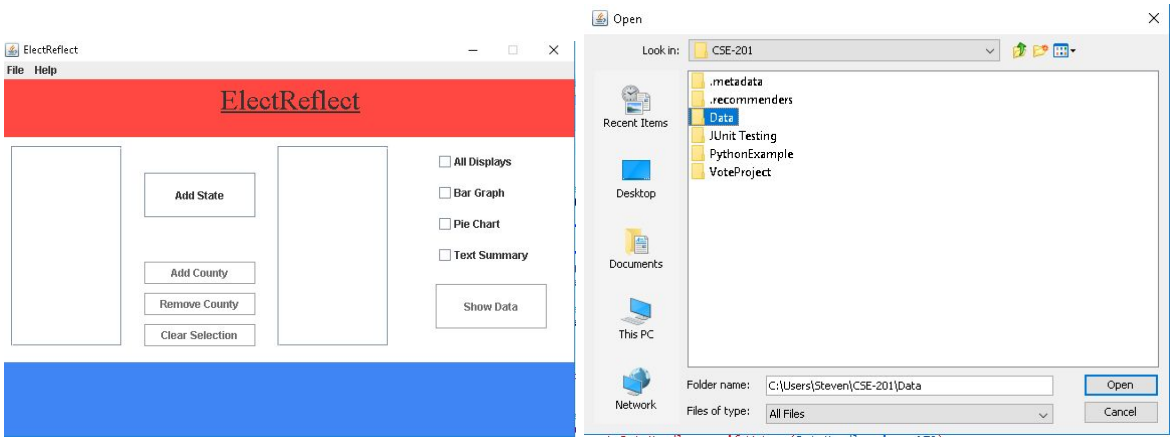
```
Allen,67697
Ashland,35181
Ashtabula,60131
Athens,43266
Auglaize,31421
Belmont,46731
Brown,28038
Butler,244275
Carroll,17868
Champaign,25521
Clark,87871
Clermont,134623
Clinton,26015
Columbiana,64991
Coshocton,22740
Crawford,27880
Cuyahoga,876013
```

Each line corresponds to one county. It contains (1) the name of the county; and (2) the number of registered voters in that county.

The output of the program will go in a file titled something like: console_log_2016-12-04.txt in the same directory as the JAR file for the program. It contains a complete record of any errors found in the data.

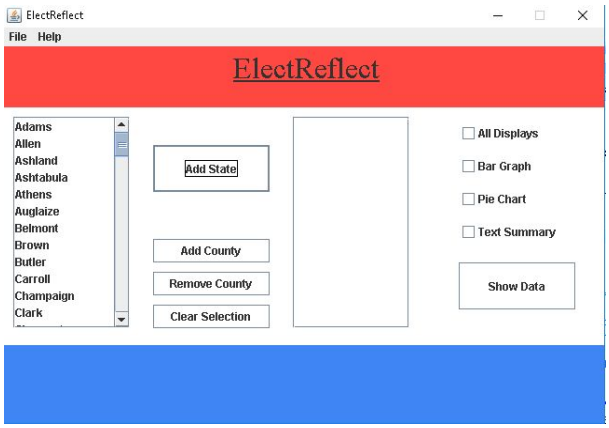
User's Guide

When you first open the program, it will display a window like this:



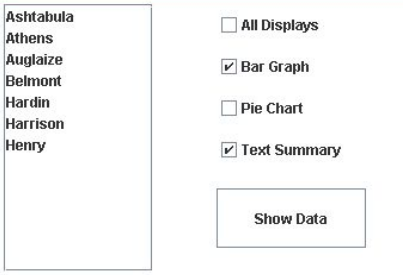
The first thing you'll want to do is import some data into the program. Click the Add State button, then in the file chooser, highlight the directory where you have your election results and click Choose. Do the same thing with the directory where you have your registered voter data.

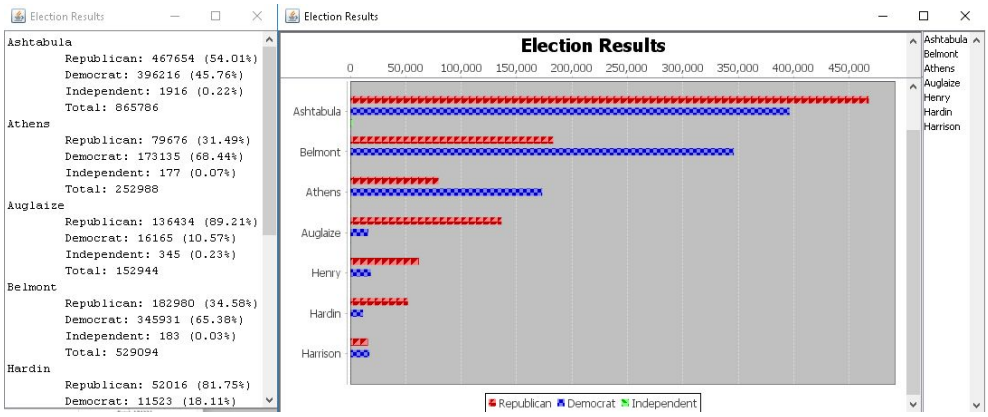
Now you should have a window looking something like this:



You can select any combination of counties, and then press the Add County button to copy them over into the other list. You can adjust the selection you made by selecting an additional set of counties and pressing Add County again, or by selecting counties from the right-hand list and pressing Remove County.

You can select any combination of visualizations, or choose All Displays, which is the same as choosing all three. Then press Show Data. It will open a window for each visualization, and you can rearrange them on the screen to your preference.





From there, you can explore the data. The textual display is a copyable text area. With the graphs you can drag rightward to zoom in or leftward to zoom out. You can right click on the graphs to save them as a PNG. The list to the right of the bar graph is a quick way of scrolling to a specific county: just click on the name of the county and the graph will jump there and zoom

that county's data to fit the screen. You can hover over the bars of the bar chart and the slices of the pie chart to see the actual numbers they represent.

When you're done, choose File > Exit from the menu of the main window, or save it first if you would like, choose File > Save, and then select a location and name for your file.

We hope you had a good time exploring our program and we hope to see you again soon.

Gallery Maintenance

Main-Page Layout
With the current version of ElectReflect, users cannot change how the interface looks.
Files that come with ElectReflect
In order for ElectReflect to offer all of its abilities, it must come with this wonderful user guide. To access the user guide, click on the help tab in the menu bar; from there, select user guide.
Image Requirements
The current version of ElectReflect has no image requirements to work properly.

Software Maintenance

If you ever want to use different data files, just add them to a new directory and use that directory instead when you're first opening the program and choosing Add State.

You might come across several sorts of errors in the log file. One common one looks like this:

    duplicate region found at removeDuplicates, duplicate found at Adams: BrattonW2.

This means that the program found multiple entries for that precinct in your data. To address this, you want to look through your data files and figure out which entry for that precinct you want to keep.

Another looks like this:

    data format error at checkFormat, Allen, Allen, Allen, Allen.

This means that there's a line in your data that looks like "Allen, Allen, Allen, Allen" and contains errors. To address this, you can either correct the line if you can make sense of it, or just delete it. This error can be safely ignored if you're not expecting the file to contain meaningful data. For example, a registered voter data file will report a data format error if you happen to have it in the same directory as your election results files, but the program will read it in successfully as a registered voter data file later on.

Console Log File:
There are 9 possible errors that are logged in the console log:

1. Extraction Error
    a. If the file finds a line that it cannot automatically fix, it throws an extraction error.
    b. This error is handled automatically, the offending line is removed and logged.
2. File Not Found Error

    a. If the program encounters another folder in the selected folder, it throws a file not found error.
    b. This error is handled automatically, the offending folder will not be used
3. Data Format Error
    a. If the program encounters data in a file that does not match the required format of a registered voter file or a voter data file and it cannot fix the data, it throws a data format error.
    b. This error is handled automatically, the offending line is removed and logged.
4. Duplicate Error
    a. If the program encounters two districts and counties of the same name it throws a duplicate error.
    b. This error is handled automatically, the offending district is removed and logged
5. Too Many Votes Error
    a. If the program finds a county with too many recorded votes, it will throw a too many votes error.
    b. This error is handled automatically, the offending county is removed and logged.
6. Negative Votes Error
    a. If the program finds a county or district with negative votes, it throws a negative votes error.
    b. This error is handled automatically, the offending district is removed and logged.
7. Missing Value Error
    a. If the program finds a missing value in the data it will throw a missing value error.
    b. This error is handled automatically, the offending line is removed and the error is logged.
8. Extra Symbol Error
    a. If the program finds an extra symbol in a number it will attempt to remove it, it will throw an extra symbol error.
    b. This error is handled automatically, the offending number is edited and the extra symbol is removed.
9. Error Not Recognized Error
    a. If the program throws an error that is not recognized by the system, it throws an error not recognized error.
    b. This error is not handled automatically, and the program will have to be restarted

Console Log Format:

1. Begin Logging
    a. A message that shows the successful creation of an error log file
2. Prompting for directory
    a. Shows that the user is being prompted to select the directory with the data
3. Getting file from given path
    a. Shows where the program is currently searching for a file
4. The file found has data: {true} or {false}
    a. Shows whether the selected file has any data in it.

5. Skipping file
    a. Shows whether the program has decided to skip a file due to a lack of data
6. Error Logs
    a. Any of the above mentioned errors are printed
7. Creating State Object
    a. All the errors have been resolved and the data is being loaded into the GUI

Debugging Tips

If the user ever accidentally imports bad data or simply wants to choose different data, exit the program and reopen it.

If the user selects a directory that contains other directories, the data handler will not go into the subdirectories to look for voter data.

Known Issues and Vulnerabilities

One known issues is that it is possible to zoom in too far on the bar graph and pie chart, to where they no longer show meaningful data. To fix this on the bar graph, select another county on the jump list to the right (or deselect and reselect the current one). This should reset the zoom to a reasonable level.
To fix a similar problem on the pie chart, drag counterclockwise on the chart (the opposite of the way you dragged to zoom in).

Another known issue is that if users' have a file in the selected directory with the same format as the voter files, ElectReflect may consider parts of these files to be votes.