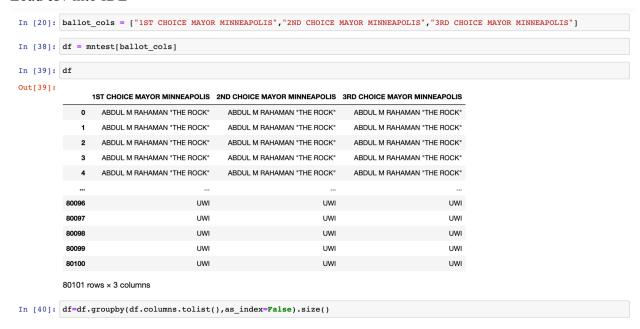
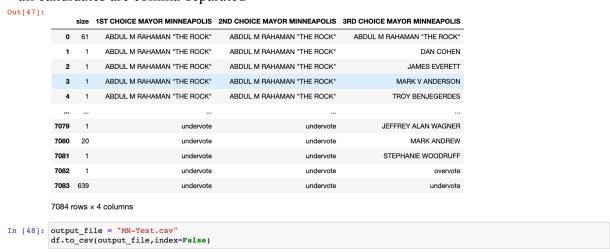
Remove undervotes, overvotes, and write-ins from CSV, along with any other candidates to remove

Load csv into IDE



Use above code to process df into following format: input file should have one line for each "type" of ballot

- first entry is number of ballots of this type
- successive entries are ranked candidates (no non-votes or repeats)
- all candidates are comma-separated



Run following code in IDE to find the losers of each round and the elected candidate(s)

```
In [49]: mntestd = make_dict_from_csv('../MN-Test.csv')
mntestd_losers = evaluate_election(mntestd)
```

Run following code in IDE to display the same information in pedigree form and write the results to a new csy

```
In [50]: get_pedigrees(mntestd,mntestd_losers,makecsv=True,csvfn='MN-TEST-csv-d3')
```

Run following code in terminal to transfer the dictionary information to a JSON python convert-csv-json.py MN-TEST-csv-d3.csv mn_test_data > elec-mn_test.js

In elec-structs.js:

- Create an electionname_text var. This should list the name of the election and any relevant information.
- Create an electionname_order var. The list should contain lists where each list gives the candidate, Lost/Elected, the number of votes they had when eliminated or elected, and the round in which they were eliminated or elected. This information can come from the evaluate_election or get_pedigrees
- In elec dict, add:
 - A key that is the name of the election
 - A value that is a list of:
 - The data
 - A dictionary with the number of winners, the description ("electionname text"), and the electionname order

In accumulation-chart.html, add the following code to import the elec-name.js to the html site.

<script src="NAME.js"></script>