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

Load csv into IDE

```
In [20]: ballot_cols = ["1ST CHOICE MAYOR MINNEAPOLIS","2ND CHOICE MAYOR MINNEAPOLIS","3RD CHOICE MAYOR MINNEAPOLIS"]
```

```
In [38]: df = mntest[ballot_cols]
```

```
In [39]: df
```

Out[39]:

	1ST CHOICE MAYOR MINNEAPOLIS	2ND CHOICE MAYOR MINNEAPOLIS	3RD CHOICE MAYOR MINNEAPOLIS
0	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"
1	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"
2	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"
3	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"
4	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"
...	...	...	...
80096	UWI	UWI	UWI
80097	UWI	UWI	UWI
80098	UWI	UWI	UWI
80099	UWI	UWI	UWI
80100	UWI	UWI	UWI

80101 rows × 3 columns

```
In [40]: df=df.groupby(df.columns.tolist(),as_index=False).size()
```

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

Out[47]:

	size	1ST CHOICE MAYOR MINNEAPOLIS	2ND CHOICE MAYOR MINNEAPOLIS	3RD CHOICE MAYOR MINNEAPOLIS
0	61	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"
1	1	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	DAN COHEN
2	1	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	JAMES EVERETT
3	1	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	MARK V ANDERSON
4	1	ABDUL M RAHAMAN "THE ROCK"	ABDUL M RAHAMAN "THE ROCK"	TROY BENJEGERDES
...	...	...	...	...
7079	1	undervote	undervote	JEFFREY ALAN WAGNER
7080	20	undervote	undervote	MARK ANDREW
7081	1	undervote	undervote	STEPHANIE WOODRUFF
7082	1	undervote	undervote	overvote
7083	639	undervote	undervote	undervote

7084 rows × 4 columns

```
In [48]: output_file = "MN-Test.csv"
         df.to_csv(output_file,index=False)
```

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 csv

```
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>
```