

VoteKit: A Python package for computational social choice research

Christopher Donnay¹, Moon Duchin², Jack Gibson³, Zach Glaser³, Andrew Hong⁴, Malavika Mukundan⁵, and Jennifer Wang⁶

¹ The Ohio State University ² Cornell University ³ MGGG Redistricting Lab ⁴ Stanford University ⁵ Boston University ⁶ Brown University

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Open Journals](#) ↗

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

The scholarly study of elections, known as *social choice theory*, centers on the provable properties of voting rules. Practical work in democracy reform focuses on designing or selecting systems of election to produce electoral outcomes that promote legitimacy and broad-based representation. For instance, the dominant electoral system in the United States is a one-person-one-vote/winner-take-all system, sometimes known as PSMD (plurality in single member districts); today, there is considerable reform momentum in favor of ranked choice voting because it is thought to mitigate the effects of vote-splitting and to strengthen prospects for minority representation, among other claimed properties.¹ Across the world, systems of election—and prospects for system change—vary substantially. From both a scholarly and a practical perspective, many questions arise about comparing the properties and tendencies of diverse systems of election in a rigorous manner.

VoteKit ([MGGG Redistricting Lab, 2023b](#)) is a Python package designed to facilitate just that kind of analysis, bringing together multiple types of functionality. Users can:

1. Create synthetic *preference profiles* (collections of ballots) with a choice of generative models and behavioral parameters;
2. Read in real-world *cast vote records* (CVRs) as observed examples of preference profiles; clean and process ballots, including by deduplication and handling of undervotes and overvotes;
3. Run a variety of *voting rules* to ingest preference profiles and output winner sets and rankings; and
4. Produce a wide range of *summary statistics* and *data visualizations* to compare and analyze profiles and election outcomes.

A tutorial that includes step-by-step example code for all of the aforementioned topics can be found in the VoteKit documentation ([MGGG Redistricting Lab, 2024](#)).

Statement of need

There has been a significant gap in the literature of computational social choice—and in the landscape of software—between the theory and the practice of democracy. On the software side,

¹Recent ranked-choice voting reforms include the adoption of instant runoff voting (IRV) in Maine, Alaska, New York City, and single transferable vote (STV) in Portland, Oregon. Advocacy groups claiming various pro-democratic properties of ranked choice include [Campaign Legal Center](#), [FairVote](#), and many others.

researchers have built a multitude of different packages for generating and analyzing elections.² Most packages, to our knowledge, handle just one part of the research arc; for instance, PrefSampling (Boehmer et al., 2024) generates profiles but does not conduct elections, while VoteLib (Jan Šimbera, 2020) only conducts elections. Others, like PrefLibTools (Nicholas Mattei and Simon Rey, 2022) and PrefVoting (Eric Pacuit and Wesley H. Holliday, 2022), provide support for generating profiles and conducting single-winner elections. Packages with multi-winner capability, like abcvoting (Lackner et al., 2023) or Apportionment (Martin Lackner, 2022), do not support ranked voting. Note that single transferable voting (STV), a voting system actually used for political election in six countries, is curiously absent. VoteKit is built to provide an end-to-end pipeline that supports ranked, scored, and approval profiles as well as single- and multi-winner elections, with an emphasis on practical applicability.

Area of need: Generative models

For one concrete example of a literature and software gap, consider the construction of *generative models*. This term is often associated with large language models as paradigms of artificial intelligence; here, what is being generated is realistic voting rather than realistic language. In this setting, a generative model of voting is a probability distribution on the set of all possible ballots that can be cast in a given election style; profiles can be sampled from a generative model to produce simulated or synthetic elections. Having sources of rich, varied, and realistic data is essential to an empirically grounded research program to probe the properties of voting rules. Good generative models are also essential to advise reformers deciding between options in a new locality, as they enable generation of synthetic profiles keyed to the scale, demographics, and election specs of that specific place.

VoteKit implements many of the models typically used in computational social choice, as well as newer mathematical models that give users the ability to generate profiles that are designed to comport with real-world ranking behavior and particularly to generate polarized elections. Two leading choices are based on classic statistical ranking mechanisms, called the Plackett–Luce (PL) and Bradley–Terry (BT) models; another model called the Cambridge Sampler (CS) draws from historical ranking data in Cambridge, MA city council elections. These models have flexible parameters, allowing users to vary voting bloc proportions, candidate strength within slates, and polarization between blocs. These parameters can be specified or randomly sampled.

Area of need: Comparison and communication

In the realm of democracy reform, groups of stakeholders often ask researchers to provide modeling studies to decide on what shift to make in electoral systems, as the project list below makes clear. VoteKit implements voting rules that stakeholders often seek to compare, with parameters designed to be tailored by the user to the specific locality under study. Available voting rules include:

- **Ranking-based (ordinal).** Plurality/SNTV, STV and IRV, (generalized) Borda, Alaska³, Top-Two, Dominating sets/Smith method, Condo-Borda⁴, Sequential RCV.

²See for instance the extensive array of open-source tools on the Computational Social Choice (COMSOC) community page (Ulle Endriss and Simon Rey, 2024) including the widely used collection of ranked data called PrefLib (Ulle Endriss and Simon Rey, 2024). See also the materials provided by FairVote, including their DataVerse and GitHub (FairVote, 2024). The ArXiv preprint (Boehmer et al., 2024) provides an impressively comprehensive list of numerical experiments on elections. The PRAGMA Project (<https://perma.cc/2P6V-8ZER>) echoes our statement of need, noting that the current literature and software falls short in practical applicability and that the understanding of real and synthetic data is “very limited.”

³Our model of the Alaska method is an SNTV/STV hybrid that uses single non-transferable vote to choose a set of finalists, then runs STV on the same preference profile to fill the seats. Alaska’s elections run this with four finalists and one seat; the top-two system runs this with two finalists and one seat.

⁴This system orders candidates within dominating sets by Borda score. Note that this is distinct from Black’s method (Black, 1986), which uses Borda score as a backup system in case the smallest dominating set is not a singleton.

- 74 ▪ **Score-based (cardinal).** Range voting, Cumulative, Limited.
- 75 ▪ **Approval-based (set).** Approval voting, Bloc plurality.

76 This list does not include every method that has attracted theoretical investigation; rather, it
77 is oriented to methods used or considered for political representation, such as the final-four
78 system in Alaska or the sequential RCV in Utah local elections. See generally (Amorós et
79 al., 2016; Emerson, 2013; McCune et al., 2023; Reynolds et al., 2008; Tideman, 1995) for
80 references. In addition, VoteKit is flexible enough to allow users to write custom voting rules.

81 Reform advocates also need to describe voting mechanisms and their likely outcomes effectively
82 to members of their communities. The end-to-end pipeline provided by VoteKit allows
83 advocates to toggle different system settings and compare expected outcomes. For example,
84 Figure 1 is reprinted from a report on reform proposals for the chambers of the Washington
85 state legislature, with Systems 0-3 as paired bicameral systems and Systems 4-5 as unicameral
86 solutions. Using the codebase that formed the foundation of VoteKit, researchers compared
87 the expected outcomes for minority representation under these six systems.

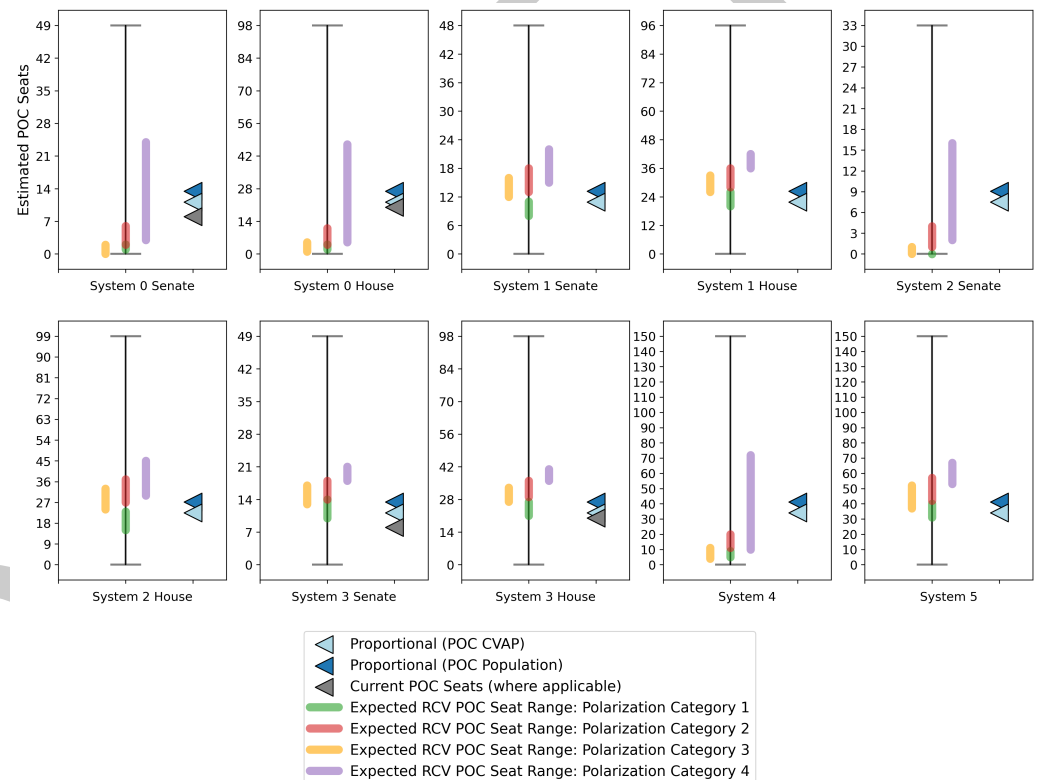


Figure 1: A comparison of a variety of electoral systems and their effect on minority representation in a case study of reform proposals for the Washington state legislature (MGGG Redistricting Lab, 2021). It can be seen that certain systems, like System 0, are projected to be less successful for minority representation, while other systems, like System 1, tend to secure proportional representation for minorities.

88 Area of need: Resources for research

89 Previous research works such as (Elkind et al., 2017) have compared properties of earlier
90 generative models; VoteKit has functionality to fully replicate this work and facilitates robust
91 comparisons across a more comprehensive and up-to-date list of alternatives. It also offers
92 new analytical tools that will support research on elections. Some examples are shown in
93 Figure 2. At left is a *ballot graph*, where nodes are ballots weighted by their frequency in the
94 profile; a recent research paper shows that ballot graphs can be metrized to realize classical

95 statistical ranking distances, like Kendall tau and the Spearman footrule (Duchin & Tapp,
96 2024). VoteKit also implements a class of election distances, as surveyed in (Boehmer et al.,
97 2022). Choices for measuring the difference between two profiles on the same set of candidates
98 include L^p distance and Wasserstein (earth-mover) distance. At right is a multidimensional
99 scaling (MDS) plot of a different set of data, showing mutual L^1 differences between generated
100 profiles across various selections of model (shown in colors) and candidate strength parameters
101 (shown with symbols), enabling comparisons in the style of (Szufa et al., 2020).

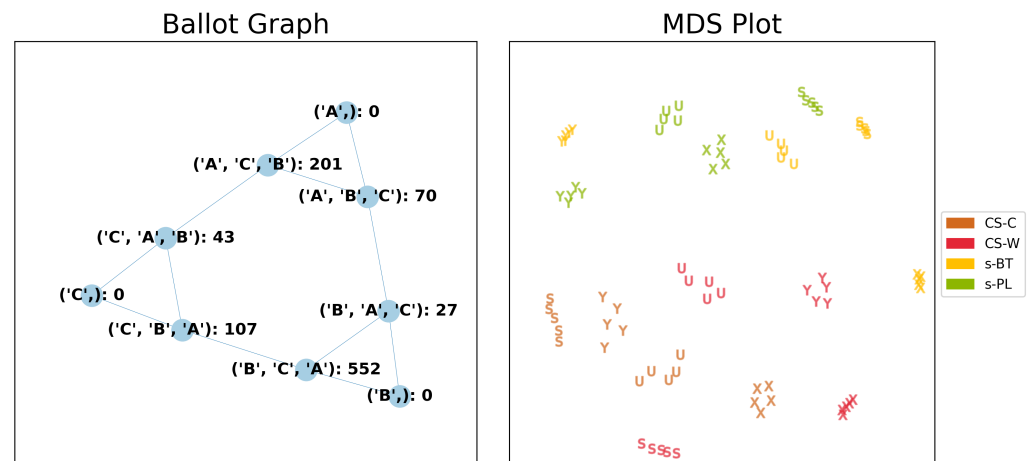


Figure 2: At left, the ballot graph for a 3-candidate election. There is one node per possible ballot, an edge goes between two ballots if they either differ by one candidate at the end of the ballot, or by swapping two adjacent candidates, and the weights show the number of instances of that ballot in the profile. We note that ballots of length 2 are identified with ballots of length 3. At right, a multidimensional scaling (MDS) plot for 80 synthetic profiles containing 1000 ballots made with various generative models and candidate strength parameters for two slates of 3 candidates each. The MDS plot is a low-distortion planar embedding of those 80 profiles and their pairwise differences. This plot shows that two of the generative models, s-BT and s-PL, are more similar to each other than to either CS model, and that different candidate strength settings cause the models to stratify within themselves.

102 Finally, VoteKit interacts seamlessly with a wide range of actual vote data, such as thousands
103 of political elections collected by FairVote and a cleaned repository of over 1000 Scottish STV
104 local government elections (FairVote, 2024; MGGG Redistricting Lab, 2023a). Previously, the
105 use of real data in election research was often extremely limited; for instance, a recent survey
106 reports that the single most popular “real-life” dataset has been a survey of 5000 respondents’
107 sushi preferences (Boehmer et al., 2024).

108 Projects

109 A significant number of white papers and scholarly articles have used VoteKit (and its
110 predecessor codebase) in recent years. These include the following.

- 111 ■ A large number of case studies in ranked-choice modeling, such as studies for the city
112 councils of Chicago, IL (MGGG Redistricting Lab, 2019b) and Lowell, MA (MGGG
113 Redistricting Lab, 2019a). the state legislatures of Oregon and
- 114 ■ A study modeling the impact of proposed legislation called the Fair Representation Act,
115 which would convert U.S. Congressional elections to the single transferable vote system
116 (MGGG Redistricting Lab, 2022);
- 117 ■ A peer-reviewed article for an election law audience on the impact of STV elections on
118 minority representation (Benadè et al., 2021);

- 119 ▪ A peer-reviewed article for a CS/econ audience that probes whether STV delivers
120 proportional representation (Benadè et al., 2024); and
- 121 ▪ A peer-reviewed article for an CS/operations research audience on optimizing to “learn”
122 blocs and slates in real-world elections (Duchin & Tapp, 2024).

123 Acknowledgements

124 This work was initiated in a research cluster in Summer 2023, funded by the Democracy Fund
125 and graciously hosted at the Faculty of Computing and Data Sciences at Boston University
126 and the Tisch College of Civic Life at Tufts University. Major contributors to the initiation of
127 the project include Brenda Macias, Emarie De La Nuez, Greg Kehne, Jordan Phan, Rory Erlich,
128 James Turk, and David McCune. Earlier code contributions were made by Chanel Richardson,
129 Anthony Pizzimenti, Gabe Schoenbach, Dylan Phelan, Thomas Weighill, Dara Gold, and
130 Amy Becker; recent code contributors include Peter Rock, Kevin Quinn, and Divij Sinha.
131 The authors also thank Deb Otis, Jeanne Clelland, and Michael Parsons for helpful feedback.
132 FairVote’s data repository in Dataverse (https://dataverse.harvard.edu/dataverse/rcv_cvrs)
133 and RCV Cruncher code on GitHub (https://github.com/fairvotereform/rcv_cruncher/) are
134 open-source efforts that provided inspiration for the current project.

135 References

- 136 Amorós, P., Puy, M. S., & Martínez, R. (2016). Closed primaries versus top-two primaries.
137 *Public Choice*, 167, 21–35. <https://doi.org/10.1007/s11127-016-0328-5>
- 138 Benadè, G., Buck, R., Duchin, M., Gold, D., & Weighill, T. (2021). *Ranked choice voting and*
139 *proportional representation*. <https://doi.org/10.2139/ssrn.3778021>
- 140 Benadè, G., Donnay, C., Duchin, M., & Weighill, T. (2024). Proportionality for ranked voting,
141 in theory and practice. In *Preprint*. <https://mggg.org/PRVTP>.
- 142 Black, D. (1986). *The theory of committees and elections*. Springer Netherlands. <https://doi.org/10.1007/978-94-009-4225-7>
- 144 Boehmer, N., Faliszewski, P., Janeczko, Ł., Kaczmarczyk, A., Lisowski, G., Pierczyński, G.,
145 Rey, S., Stolicki, D., Szufa, S., & Wąs, T. (2024). Guide to numerical experiments on
146 elections in computational social choice. In K. Larson (Ed.), *Proceedings of the thirty-*
147 *third international joint conference on artificial intelligence, IJCAI-24* (pp. 7962–7970).
148 International Joint Conferences on Artificial Intelligence Organization. <https://doi.org/10.24963/ijcai.2024/881>
- 150 Boehmer, N., Faliszewski, P., Niedermeier, R., Szufa, S., & Wąs, T. (2022). Understanding
151 distance measures among elections. In L. D. Raedt (Ed.), *Proceedings of the thirty-*
152 *first international joint conference on artificial intelligence, IJCAI-22* (pp. 102–108).
153 International Joint Conferences on Artificial Intelligence Organization. <https://doi.org/10.24963/ijcai.2022/15>
- 155 Duchin, & Tapp. (2024). Learning blocs and slates from observed elections. *Preprint*.
- 156 Elkind, E., Faliszewski, P., Laslier, J.-F., Skowron, P., Slinko, A., & Talmon, N. (2017).
157 What do multiwinner voting rules do? An experiment over the two-dimensional euclidean
158 domain. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1). <https://doi.org/10.1609/aaai.v31i1.10612>
- 160 Emerson, P. (2013). The original borda count and partial voting. *Social Choice and Welfare*,
161 40, 353–358. <https://doi.org/10.1007/s00355-011-0603-9>
- 162 Eric Pacuit and Wesley H. Holliday. (2022). *Pref_voting*. <https://github.com/voting-tools/>

- 163 [pref_voting](#)
- 164 FairVote. (2024). *RCV cruncher*. https://github.com/fairvotereform/rcv_cruncher/
- 165 Jan Šimbera. (2020). *Votelib*. <https://github.com/simberaj/votelib>
- 166 Lackner, M., Regner, P., & Krenn, B. (2023). Abcvoting: A Python package for approval-
167 based multi-winner voting rules. *Journal of Open Source Software*, 8(81), 4880. <https://doi.org/10.21105/joss.04880>
- 168
- 169 Martin Lackner. (2022). *Apportionment*. <https://github.com/martinlackner/apportionmentb>
- 170 McCune, D., Martin, E., Latina, G., & Simms, K. (2023). *A comparison of sequential ranked-*
171 *choice voting and single transferable vote*. <https://doi.org/10.1007/s42001-024-00249-8>
- 172 MGGG Redistricting Lab. (2019a). Findings on the city of lowell's election systems. In *White*
173 *Paper*. <https://mggg.org/Lowell-Detailed-Report>.
- 174 MGGG Redistricting Lab. (2019b). Study of reform proposals for chicago city council. In
175 *White Paper*. <https://mggg.org/publications/Chicago.pdf>.
- 176 MGGG Redistricting Lab. (2021). Analysis of election systems for washington state. In *White*
177 *Paper*. <https://mggg.org/Washington>.
- 178 MGGG Redistricting Lab. (2022). Modeling the Fair Representation Act. In *White Paper*.
179 <https://mggg.org/FRA-Report>.
- 180 MGGG Redistricting Lab. (2023a). *Scottish STV election repo*. <https://github.com/mggg/scot-elex>
- 181
- 182 MGGG Redistricting Lab. (2023b). *VoteKit*. <https://github.com/mggg/VoteKit>
- 183 MGGG Redistricting Lab. (2024). *VoteKit documentation*. <https://votekit.readthedocs.io/en/latest/>
- 184
- 185 Nicholas Mattei and Simon Rey. (2022). *PrefLib-tools*. <https://github.com/PrefLib/preflibtools>
- 186
- 187 Reynolds, A., Reilly, B., & Ellis, A. (2008). *Electoral system design: The new international*
188 *IDEA handbook*. International Institute for Democracy; Electoral Assistance.
- 189 Szufa, S., Faliszewski, P., Skowron, P., Slinko, A., & Talmon, N. (2020). [Drawing a map of elec-](#)
190 [tions in the space of statistical cultures](#). *Proceedings of the 19th International Conference*
191 *on Autonomous Agents and MultiAgent Systems*, 1341–1349. ISBN: 9781450375184
- 192 Tideman, N. (1995). The single transferable vote. *Journal of Economic Perspectives*, 9(1),
193 27–38. <https://doi.org/10.1257/jep.9.1.27>
- 194 Ulle Endriss and Simon Rey. (2024). *COMSOC community site*. <https://comsoc-community.org/tools>
- 195