Scientific Research Publishing

Theoretical Economics Letters, 2024, 14, 543-557
https://www.scirp.org/journal/tel
ISSN Online: 2162-2086
ISSN Print: 2162-2078

Ranked-Choice Voting

Robert A. Agnew

Independent Researcher, Palm Coast, Florida, USA
Email: raagnew1@gmail.com

How to cite this paper: Agnew, R. A. (2024). Ranked-Choice Voting. *Theoretical Economics Letters, 14,* 543-557.
https://doi.org/10.4236/tel.2024.142029

Received: January 26, 2024
Accepted: April 21, 2024
Published: April 24, 2024

Abstract

The mathematical theory of ranked-choice voting is reviewed, with particular focus on Condorcet, Plurality, and Borda Count methods. Maximizing the Borda Count score is shown to be equivalent to minimizing the weighted arithmetic mean across ranks. This leads to alternate Borda methods associated with different weighted means: geometric; harmonic; and root-mean-square. The notion of an ergodic voter population is introduced, together with a statistical confidence interval approach to estimating requisite population size, which is essential to avoid small-population noise. Population size and voting method consistency are further examined via simulation where it is shown that principal voting methods, other than Plurality, converge for a sufficiently large, ergodic voter population. This again highlights pitfalls associated with small voter populations, in terms of both voting method consistency and elicitation of overall voter/social preference.

Keywords

Ranked-Choice Voting, Positional Voting, Social Choice Functions, Condorcet, Borda

1. Introduction

Ranked-choice voting occurs when each member of a voting population ranks k candidates from 1 to k in strict order, 1 best to k worst, where $k \geq 3$. Candidates can be politicians, proposals, or consumer products like automobiles or laundry detergents. The idea here is that a ranked permutation of candidates provides much more information about voter preference than a singleton choice, even though we are often interested in selecting a single winner. The objective of ranked-choice voting is to elicit an aggregate social choice ranking of candidates beyond a simple plurality (most first-place votes) and to designate the aggregate top-ranked winner utilizing all available information about voter preference.

Various voting methods, or social choice functions, have been proposed, going back to Eighteenth Century France with competing methods proposed by Borda and Condorcet. Both of these methods depend on an aggregated, anonymous voter profile which summarizes counts of voters associated with various candidate rank permutations. Borda Count, and other positional scoring functions, depend simply on rank counts for each candidate. The Borda Count score winner, in particular, has a number of commendable attributes, although it is considered suboptimal when compared to the Condorcet winner who beats all pairwise opponents, ignoring others. The Condorcet winner, although generally considered optimal, doesn't always exist, although other robust pairwise methods exist, the most comprehensive and complex being the Kemeny method. In our age of high-powered computation, algorithmic complexity is not really an issue, but understandability of a voting method remains an issue. Hence, Borda Count and other simple scoring methods have continuing allure. Moreover, it is useful to explore where Condorcet and Borda yield the same winner.

Part I of Robinson and Ullman (2017) provides an excellent overview of ranked-choice voting, including all pairwise and positional voting rules and various social choice criteria to evaluate them; they also have a vast array of pedological problems, one of which we highlight later. Saari (2000) and Saari (2023) warn about pitfalls associated with all such voting methods and he champions Borda Count as retaining the most information about voter preference as well as being least impacted by paradoxical behaviors, including strategic manipulative voting, which is further examined in Mata-Perez et al. (2014).

Heilman (2022) deduces, under certain technical conditions, that Borda Count is the stablest voting method in the face of randomly corrupted votes. Conitzer and Sandholm (2005) show that simple scoring rules, like Borda Count, correspond to maximum likelihood estimators for certain voter noise models, unlike some competing pairwise methods. Young and Levenglick (1978) and Young (1995) support Kemeny as the natural embodiment of Condorcet's approach. Crisman (2014) uses representation theory of the symmetry group of the permutahedron to define an algebraic spectrum of voting rules connecting Borda to Kemeny. Other general overviews of ranked-choice voting are provided by Nurmi (2010) and Zhang (2020).

Our focus is twofold. First, we reinterpret the classic Borda Count voting method and derive a family of alternative Borda methods that correspond to various weighted rank means. Our purpose here is not to detract from Borda Count, but rather to reinforce it. Second, we address the issue of voter population composition and size to ensure statistical accuracy of results. Most textbook problems have very small voter populations by design, and they certainly highlight divergences among voting methods. However, these divergences tend to disappear for larger populations that are at once ergodic and sizable.

In the next section, we reinterpret Borda Count as the weighted arithmetic mean across ranks $1,\cdots,k$ with *minimum* score the winner. This opens the door to other weighted means like geometric, harmonic, root-mean-square, and indeed to an infinity of weighted power means. In section 3, we address the issue of voter population composition and size. We demonstrate that size alone does not ensure consistency across voting methods. Ergodicity is also required. Given ergodicity, we indicate how to ensure sufficient voter population size, assuming that is possible. In section 4, we provide an ergodic simulation which demonstrates convergence of Condorcet and Borda variations for sufficiently large population size. In our simulation, only the Plurality method ultimately diverges from the others. Section 5 contains our summary and conclusions and the Appendix contains our R-scripts/functions for Condorcet, Plurality, and alternative Borda methods along with scripts for Problem 2.1 in Robinson & Ullman (2017) and our simulation. Throughout we ignore winner ties as being unlikely in the real world.

2. Borda Variations

We have $k \geq 3$ distinct candidates and a population of voters, each of whom ranks all k candidates in strict order, from 1 best to k worst. Hence, each voter's ranking is a permutation of $1, \cdots, k$ and assuming equal, anonymous voters, we may aggregate results into a voter profile with voter counts in the first column and specific candidate rank permutations in columns $2, \cdots, k+1$. This is a transposed version of the usual textbook profile with candidates labeled by numbers rather than letters.

Table 1 profiles the results of an election with 5 ranks and 11 total votes with only 5 of 120 = 5! possible rank permutations represented. Nevertheless, it is quite instructive. The first row indicates 4 votes associated with a permutation that puts candidate 1 in rank 1, candidate 3 in rank 2, candidate 4 in rank 3, and so on. Candidate 1 is the Plurality winner with 4 first-place (Rank 1) votes. Candidate 3 is the Condorcet winner who beats all other candidates in pairwise, head-to-head competition, i.e., 6:5 against each contender 1, 2, 4, 5. Positional methods like Borda Count (and Plurality) involve a $k \times k$ count matrix $C = \left[c_{ij}\right]$ where c_{ij} is the count of votes for candidate i in rank position j.

Table 1. Candidate rank permutations and votes from Problem 2.1 in Robinson & Ullman (2017).

	Ranks				
Votes	1	2	3	4	5
4	1	3	4	2	5
3	2	5	4	3	1
2	3	2	4	1	5
1	4	2	5	3	1
1	5	4	2	1	3

Borda Count involves weighted candidate scores $s_i = \sum_{j=1}^{k} c_{ij}(k-j)$ and the candidate with maximum score is designated the winner. From Table 1, we compute scores 19, 27, 24, 25, 15 so candidate 2 is the designated Borda Count winner. Calculations for this problem are simple and straightforward but our R-scripts/functions automate the process for this and any other voter profile.

Let n denote the total number of voters and define matrix $P = C/n$. This matrix is doubly stochastic (or doubly Markov, cf Cinlar (1975)) in that its rows and columns all sum to 1. If we replace c_{ij} with p_{ij} in the Borda scoring formula, the winner is the same. If we further replace $k-j$ by j in the scoring formula, the same winner emerges if we switch from maximum to *minimum* score. This revised (and simplified) scoring formula $s_i = \sum_{j=1}^{k} p_{ij}\, j$ is recognizable as the weighted arithmetic mean across ranks $1,\cdots,k$ for weights $p_{i1},\cdots,p_{ik}$. While this seems logical, we have no assurance that the voting population has in mind a linear scale for ranks. Consequently, it seems advisable to compute alternative weighted means to see if they designate the same winner; if so, the Borda case is bolstered.

There is an infinity of weighted power means to consider; we highlight a few in the following Table 2. See Hong et al. (2007) for application of weighted power mean operators in fuzzy information retrieval.

Table 2. Weighted mean variations on the Borda Count scoring formula. Refer to planetmath.org/weightedpowermean and en.wikipedia.org/wiki/Generalized_mean for general relationships among these weighted means.

Weighted Mean	Scoring Formula with Minimum Score Wins
Arithmetic	$s_i = \sum_{j=1}^{k} p_{ij}\, j$ *Same as Borda Count.*
Geometric	$s_i = \prod_{j=1}^{k} j^{p_{ij}} = \exp\left(\sum_{j=1}^{k} p_{ij} \ln(j)\right)$
Harmonic	$s_i = 1/\sum_{j=1}^{k} p_{ij}/j$ *Same as Dowdall/Nauru.*
Root-Mean-Square (RMS)	$s_i = \sqrt{\sum_{j=1}^{k} p_{ij}\, j^2}$
General Power	$s_i = \left(\sum_{j=1}^{k} p_{ij}\, j^a\right)^{1/a}$

Referring to the general power mean formula, Arithmetic corresponds to $a = 1$, Geometric corresponds to $a = 0$ (in the limit), Harmonic corresponds to $a = -1$, and RMS corresponds to $a = 2$. Referring back to the profile in Table 1, and utilizing the PROFILE2.1 function in the Appendix, Geometric and Harmonic yield the same winner as Arithmetic, candidate 2. RMS, on the other hand, yields candidate 4 as winner, yet another divergence. We next turn to the impact of voter population composition and size.

3. Voter Population

Referring back to the voting profile in Table 1, we note that adding N zeros to every element in the votes column changes absolutely nothing. Voter population size doesn't matter if there are large swathes of voters who rank candidates in locked step, by virtue of either mania or intimidation. We need more to ensure that size matters and that it may cure lumps and divergences, as in inferential statistics.

Definition. A voter population is *ergodic* if (1) individual voter rank-permutations are stochastically independent and identically distributed and (2) every voter rank-permutation has positive probability.

This definition is adapted from the usual notion of ergodicity (cf Breiman (1969)) that a statistical average will converge to the true population mean as size increases. Condition 1 ensures large-sample convergence. Condition 2 ensures that convergence encompasses the entire universe of possibilities. We shall see an example later in Section 4.

Once again, let n denote total voter population size and denote P elements as $p_{ij}^{(n)}$. Under ergodicity, we will have independent voter multinomial trials by candidate and we will have $p_{ij}^{(n)} \to \theta_{ij} \in (0,1)$ as $n \to \infty$ by the Law of Large Numbers. We might then ask how large the population should be to ensure a certain level of statistical precision in our estimates. The rows of P are interrelated, so we will focus on any particular row. Goodman (1965) derived *simultaneous* large-sample confidence intervals of the form $\left|p_{ij}^{(n)} - \theta_{ij}\right| \le \sqrt{Q\theta_{ij}\left(1-\theta_{ij}\right)/n}$ with confidence level $1-\alpha$ and where Q is the upper $(\alpha/k)\times 100th$ percentile of the chi-square distribution with one degree of freedom. If, for instance, $\alpha = 0.05$ (to achieve 95% confidence) and $k = 4$, then $Q = 6.238533$.

Once an election is run, we know the $p_{ij}^{(n)}$ so we can solve the quadratic equation $\left(p_{ij}^{(n)} - \theta_{ij}\right)^2 = Q\theta_{ij}\left(1-\theta_{ij}\right)/n$ for upper and lower confidence limits $\theta_{ij}^{+}, \theta_{ij}^{-}$ or we may simply estimate θ_{ij} by $p_{ij}^{(n)}$ and obtain confidence limits $\pm\sqrt{Q\,p_{ij}^{(n)}\left(1-p_{ij}^{(n)}\right)/n}$. Before an election is run, we have no estimates, but we can use conservative limits with specified precision to compute required voter population size. Since $\theta(1-\theta)$ is maximized at $\theta = 1/2$, we can examine a conservative solution to $\delta = \sqrt{\dfrac{Q}{4n}}$ or $n = \dfrac{Q}{4\delta^2}$ where precision δ is stipulated and we solve for population size n. For instance, if $\alpha = 0.05$, $k = 4$, and

$\delta = 0.04$, then $n = 975$ is the required voter population size. Even for a purely ergodic voter population, we may need significant voter population size to ensure a reasonable degree of statistical precision.

We are not asserting that all voter populations are ergodic. In reality, voter populations are probably some mix of hidebound and random rankings. Moreover, we are not asserting that rank-choice voting should be avoided for small voter populations, only that we acknowledge pitfalls and strive for adequate population size (and randomization) when possible. In the next section, we examine these issues further in the context of an ergodic simulation model.

4. Ergodic Simulation Model

Our simulation randomizes independent and identically distributed voters sequentially with a specified starting random number seed. We first randomize a voting bloc (tendency), followed by 4 ranks according to first-place candidate rank probabilities (without replacement) to generate each voter's candidate rank permutation. We then aggregate the voting profile and compute winners for Condorcet, Plurality, and Borda variations.

If we denote the first column of Table 3 as v and the remaining 4×4 square matrix as R, then the $v^{\mathrm{T}} R = (0.30, 0.29, 0.21, 0.20)$ vector represents the *unconditional* first-place candidate rank probabilities for comparison to the first column of the P-matrix simulation results. We designed inputs so that candidates 1 and 2 had close unconditional first-place probabilities while the last two candidates were more "spoilers." Despite all this machinery, the simulation is ergodic with independent multinomial trials summarized in the voting profile, the C-matrix, and the P-matrix.

Table 4 results were generated by the SIMULATION function in the Appendix. For only 10 voters, there is no Condorcet winner and all Borda variations are consistent with the Plurality winner, candidate 1. For 100 voters, the only difference is that Condorcet comes in, again with winning candidate 1. With 1,000 voters and up, however, candidate 2 wins across the board except for Plurality. This shows that true voter preferences were obscured by noise for smaller population sizes.

All 24 = 4! possible rank permutations are represented Table 5, some more than others of course. This lends further credibility to the notion that we are capturing true social preference across voters. Table 6 is the P-matrix that drives the Plurality and Borda winners for a particular random number seed.

We duplicated the simulation for other random number seeds and got different results for smaller voter populations. For random number seed 17, candidate 2 was the winner for all voting methods up through 10,000 votes. Only a tad above 11,000 votes did Plurality revert to candidate 1. For random number seed 23, we got mixed results below 1,000 voters. Even at 1,000, the uniform winner was candidate 1. We had to get up to 1,900 voters and above to duplicate the 1,000 voter pattern from Table 4. For random number seed 29, we had to get up to 23,000 for Borda-Harmonic to switch from candidate 1 to 2. For an ergodic population, random noise declines naturally as size increases, but in close elections like this one, definitive results depend crucially on a large voter population.

Table 3. Simulation inputs specified by the author to reflect a close election.

Voting Bloc Probabilities	Candidate First-Place Rank Probabilities			
	1	2	3	4
0.4	0.6	0.1	0.2	0.1
0.3	0.1	0.6	0.2	0.1
0.2	0.1	0.2	0.1	0.6
0.1	0.1	0.3	0.5	0.1

Table 4. Simulated winning candidates for random number seed 7. Note the transition at 1,000 voters.

Voting Method	Number of Voters			
	10	100	1,000	10,000
Condorcet	None	1	2	2
Plurality	1	1	1	1
Borda-Arithmetic	1	1	2	2
Borda-Geometric	1	1	2	2
Borda-Harmonic	1	1	2	2
Borda-RMS	1	1	2	2

Table 5. Simulated voting profile for 1,000 voters and random number seed 7.

	Ranks			
Votes	1	2	3	4
56	1	2	3	4
39	1	2	4	3
66	1	3	2	4
80	1	3	4	2
26	1	4	2	3
42	1	4	3	2
44	2	1	3	4

Continued

35	2	1	4	3
62	2	3	1	4
62	2	3	4	1
43	2	4	1	3
51	2	4	3	1
35	3	1	2	4
42	3	1	4	2
44	3	2	1	4
39	3	2	4	1
15	3	4	1	2
21	3	4	2	1
28	4	1	2	3
38	4	1	3	2
42	4	2	1	3
44	4	2	3	1
13	4	3	1	2
33	4	3	2	1

Table 6. Simulated P-matrix for 1,000 voters and random number seed 7. Note that the first column is reasonably close to the stipulated candidate first-place unconditional rank probabilities.

	Ranks			
Candidates	1	2	3	4
1	0.309	0.222	0.219	0.250
2	0.297	0.264	0.209	0.230
3	0.196	0.316	0.275	0.213
4	0.198	0.198	0.297	0.307

5. Summary & Conclusion

We have introduced a family of weighted-mean Borda Count variations for comparison to Condorcet and Plurality voting methods. We also introduced the notion of an ergodic voter population and discussed the importance of voter population size in that context, together with a statistical confidence interval approach to estimating requisite size. Via simulation, we demonstrated that ergodic population size is crucial in terms of voting method consistency and elicitation of overall voter/social preference, particularly when an election is likely to be plurality-wise close.

As in normal statistical inference, we should strive, when possible, for voter populations that are sufficiently large and randomized to ensure that election results reflect true voter/social preference. We acknowledge that we can't always achieve these ideals, but we should always be cognizant of inferential pitfalls when they are absent. As we have seen, principal voting methods, other than Plurality, tend to converge for large, ergodic voting populations. There is undoubtedly opportunity for future research on this interplay between voting method and voter population composition and size.

Conflicts of Interest

No potential conflict of interest was reported by the author.

References

Breiman, L. (1969). *Probability and Stochastic Processes with a View toward Applications*. Houghton-Mifflin.

Cinlar, E. (1975). *Introduction to Stochastic Processes*. Prentice-Hall.

Conitzer, V., & Sandholm, T. (2005). Common Voting Rules as Maximum Likelihood Estimators. In F. Bacchus, & T. Jaakkola (Eds.), *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence* (pp. 145-152). AUAI Press. https://doi.org/10.48550/arXiv.1207.1368

Crisman, K. (2014). The Borda Count, the Kemeny Rule, and the Permutahedron. In K. Crisman, & M. A. Jones (Eds.), *The Mathematics of Decisions, Elections, and Games*. American Mathematical Society. https://doi.org/10.1090/conm/624/12475

Goodman, L. A. (1965). On Simultaneous Confidence Intervals for Multinomial Proportions. *Technometrics, 7,* 247-254. https://doi.org/10.2307/1266673

Heilman, S. (2022). Noise Stability of Ranked Choice Voting. https://doi.org/10.48550/arXiv.2209.11183

Hong, W., Chen, S., Wang, L., & Chen, S. (2007). A New Approach for Fuzzy Information Retrieval Based on Weighted Power-Mean Averaging Operators. *Computers & Mathematics with Applications, 53,* 1800-1819. https://doi.org/10.1016/j.camwa.2006.04.033

Mata-Perez, E., Laruelle, A., Martinez, R., & Ponti, G. (2014). Preference Intensity in Positional Voting. *Theoretical Economics Letters, 4,* 727-738. https://doi.org/10.4236/tel.2014.48092

Nurmi, H. (2010). Chap. 6. Voting Theory. In D. Rios Insua, & S. French (Eds.), *e-Democracy. Advances in Group Decision and Negotiation* (Vol. 5, pp. 101-123). Springer. https://doi.org/10.1007/978-90-481-9045-4_7

Robinson, E. A., & Ullman, D. H. (2017). *The Mathematics of Politics* (2nd ed.). CRC Press. https://doi.org/10.1201/9781315366678

Saari, D. G. (2000). *Chaotic Elections! A Mathematician Looks at Voting*. American Mathematical Society.

Saari, D. G. (2023). Selecting a Voting Method: The Case for the Borda Count. *Constitu-*

tional Political Economy, 34, 357-366. https://doi.org/10.1007/s10602-022-09380-y

Young, H. P. (1995). Optimal Voting Rules. *Journal of Economic Perspectives, 9,* 51-64. https://doi.org/10.1257/jep.9.1.51

Young, H. P., & Levenglick, A. (1978). A Consistent Extension of Condorcet's Election Principle. *SIAM Journal on Applied Mathematics, 35,* 285-300. https://doi.org/10.1137/0135023

Zhang, D. (2020). Methods and Rules of Voting and Decision: A Literature Review. *Open Journal of Social Sciences, 8,* 60-72. https://doi.org/10.4236/jss.2020.81005

Appendix. R Scripts/Functions

Our analytical and simulation functions are delineated within R-scripts. Once those scripts are loaded into an R workspace, either manually or via the source function, they can be run repeatedly for different inputs and parameters. Moreover, the entire workspace can be saved for repeated use.

```
CONDORCET <- function(M){
# Script CONDORCET.R
# Produces a Condorcet winner (if available, else 0) from voting profile matrix
M
v <- M[,1] # Vote vector
sv<- sum(v) # Total votes
R <- M[,-1] # Candidate rank permutations
n <- length(v)
k <- dim(R)[2] # Number of candidates and positions
# Tally of pairwise rank orderings
C <- matrix(0,nrow=k,ncol=k)
for (i in 1:n){C <- C + v[i]*outer(order(R[i,]),order(R[i,]),"<")}
# Determine which (if any) candidate wins all pairwise contests
t <- NULL
for (i in 1:k){t <- c(t,min(C[i,i!=(1:k)]))}
# Winner must achieve majority against every rival
ifelse(max(t) <= sv/2,0,which.max(t))}

COUNT <- function(M){
# Script COUNT.R
# Produces candidate by position square count matrix from voting profile matrix
M
v <- M[,1] # Vote vector
R <- M[,-1] # Candidate rank permutations
n <- length(v)
k <- dim(R)[2] # Number of candidates and positions
C <- matrix(0,nrow=k,ncol=k)
for (i in 1:n){C <- C + v[i]*outer(1:k,R[i,],"==")}
C}

PLURALITY <- function(M){
# Script PLURALITY.R
# Produces plurality scores and a winning candidate for voter profile matrix M
# This is the familiar most-first-place votes method
n <- sum(M[,1]) # Total votes
k <- dim(M)[2]-1 # Number of candidates and ranks
P <- COUNT(M)/n # Candidate-by-rank doubly stochastic matrix correspond-
```

```
ing to M
scores<- P%*%c(1,rep(0,(k-1))) # Only rank 1 votes are counted
# Ignoring ties
which.max(scores)}

BORDA.ARITHMETIC <- function(M){
# Script BORDA_ARITHMETIC.R
# Produces Borda Arithmetic scores and a winning candidate for voter profile
matrix M
# This is equivalent to the familiar Borda Count voting method
n <- sum(M[,1]) # Total votes
k <- dim(M)[2]-1 # Number of candidates and ranks
P <- COUNT(M)/n # Candidate-by-rank doubly stochastic matrix correspond-
ing to M
scores<- P%*%(1:k) # Weighted arithmetic mean rank by candidate
# Ignoring ties
which.min(scores)}

BORDA.GEOMETRIC <- function(M){
# Script BORDA_GEOMETRIC.R
# Produces Borda Geometric scores and a winning candidate for voter profile
matrix M
# This is the natural geometric mean companion to Borda Arithmetic
n <- sum(M[,1]) # Total votes
k <- dim(M)[2]-1 # Number of candidates and ranks
P <- COUNT(M)/n # Candidate-by-rank doubly stochastic matrix correspond-
ing to M
scores<- exp(P%*%log(1:k)) # Weighted geometric mean rank by candidate
# Ignoring ties
which.min(scores)}

BORDA.HARMONIC <- function(M){
# Script BORDA_HARMONIC.R
# Produces Borda Harmonic scores and a winning candidate for voter profile
matrix M
# This is the natural harmonic mean companion to Borda Arithmetic
# This is equivalent to the familiar Dowdall/Nauru voting method
n <- sum(M[,1]) # Total votes
k <- dim(M)[2]-1 # Number of candidates and ranks
P <- COUNT(M)/n # Candidate-by-rank doubly stochastic matrix correspond-
ing to M
scores<- 1/(P%*%(1/(1:k))) # Weighted harmonic mean rank by candidate
# Ignoring ties
```

```
which.min(scores)}

BORDA.RMS <- function(M){
# Script BORDA_RMS.R
# Produces Borda Root-Mean-Square scores and a winning candidate for voter
profile matrix M
# This is the natural RMS companion to Borda Arithmetic
n <- sum(M[,1]) # Total votes
k <- dim(M)[2]-1 # Number of candidates and ranks
P <- COUNT(M)/n # Candidate-by-rank doubly stochastic matrix correspond-
ing to M
scores<- sqrt(P%*%(1:k)^2) # Weighted root mean square rank by candidate
# Ignoring ties
which.min(scores)}

PROFILE2.1 <- function(){
# Script PROFILE21.R
# Produces and stores transposed version of voter profile in Robinson & Ullman,
Problem 2.1
# Designates winners for different voting methods
v <- c(4,3,2,1,1)
R <- c(1,3,4,2,5,2,5,4,3,1,3,2,4,1,5,4,2,5,3,1,5,4,2,1,3)
R <- matrix(R,nrow=5,ncol=5)
M <- t(rbind(v,R))
print("Voting Profile")
print(M)
#write.table(M,file="c:/VOTING/PROFILE21.csv",col.names=FALSE,row.names
=FALSE,sep=",")
source("c:/VOTING/CONDORCET.R")
source("c:/VOTING/COUNT.R")
source("c:/VOTING/PLURALITY.R")
source("c:/VOTING/BORDA_RMS.R")
source("c:/VOTING/BORDA_ARITHMETIC.R")
source("c:/VOTING/BORDA_GEOMETRIC.R")
source("c:/VOTING/BORDA_HARMONIC.R")
print("C-Matrix")
print(COUNT(M))
P <- COUNT(M)/11
print("P-Matrix")
print(P)
print(paste0("CONDORCET WINNER = ",CONDORCET(M)))
print(paste0("PLURALITY WINNER = ",PLURALITY(M)))
print(paste0("BORDA-ARITHMETIC WINNER =
",BORDA.ARITHMETIC(M)))
print(paste0("BORDA-GEOMETRIC WINNER = ",BORDA.GEOMETRIC(M)))
print(paste0("BORDA-HARMONIC WINNER = ",BORDA.HARMONIC(M)))
print(paste0("BORDA-RMS WINNER = ",BORDA.RMS(M)))}

SIMULATION <- function(NUM,SEED){
# Script SIMULATION.R
# Produces and stores simulated voter profile
# Designates winners for different voting methods
# NUM = total number of independent voters
# SEED = random number seed for reproducible results
# Simulation for independent voters with probabilistic voting bloc tendencies
# All candidate rank permutations are possible
library(dplyr)
# Randomized voting blocs and independent voter rank permutations within
blocs
BLOC_PROBS <- c(.4,.3,.2,.1) # Voting bloc probabilities
# Candidate first-place rank probabilities by voting bloc
# Used also for subsequent rankings
RANK_PROBS <- c(.6,.1,.2,.1) # Candidate rank probs for Bloc 1
RANK_PROBS <- rbind(RANK_PROBS,c(.1,.6,.2,.1)) # for Bloc 2
RANK_PROBS <- rbind(RANK_PROBS,c(.1,.2,.1,.6)) # for Bloc 3
RANK_PROBS <- rbind(RANK_PROBS,c(.1,.3,.5,.1)) # for Bloc 4
set.seed(SEED) # Set random number seed
# Simulated independent voter rank permutations
M <- NULL
for (i in 1:NUM){
a <- sample.int(4,size=1,replace=FALSE,prob=BLOC_PROBS)
b <- sample.int(4,size=4,replace=FALSE,prob=RANK_PROBS[a,])
M <- rbind(M,b)}
# Aggregation into candidate rank permutation counts
df<- data.frame(M)
A <- data.matrix(df %>% group_by_all() %>% count)
M <- cbind(A[,5],A[,-5])
print("Simulation Voting Profile")
print(M)
#
write.table(M,file="c:/VOTING/SIM_PROFILE.csv",col.names=FALSE,row.nam
es=FALSE,sep=",")
source("c:/VOTING/CONDORCET.R")
source("c:/VOTING/COUNT.R")
source("c:/VOTING/PLURALITY.R")
source("c:/VOTING/BORDA_RMS.R")
source("c:/VOTING/BORDA_ARITHMETIC.R")
source("c:/VOTING/BORDA_GEOMETRIC.R")
source("c:/VOTING/BORDA_HARMONIC.R")
print("Unconditional First-Place Probabilties")
print(BLOC_PROBS%*%RANK_PROBS)
print("C-Matrix")
print(COUNT(M))
P <- COUNT(M)/NUM
print("P-Matrix")
print(P)
#
write.table(P,file="c:/VOTING/SIM_P_MATRIX.csv",col.names=FALSE,row.na
mes=FALSE,sep=",")
print(paste0("CONDORCET WINNER = ",CONDORCET(M)))
print(paste0("PLURALITY WINNER = ",PLURALITY(M)))
print(paste0("BORDA-ARITHMETIC WINNER = ",BORDA.ARITHMETIC(M)))
print(paste0("BORDA-GEOMETRIC WINNER = ",BORDA.GEOMETRIC(M)))
print(paste0("BORDA-HARMONIC WINNER = ",BORDA.HARMONIC(M)))
print(paste0("BORDA-RMS WINNER = ",BORDA.RMS(M)))}
```