CentraleSupélec

Master DSBI
Decision Modeling
Practical works

Voting rules in python

Let us consider an election with n voters ($0 \leq n \leq 100$) and m candidates ($0 \leq m \leq 5$). We assume that :

- The preferences of each voter are given as a linear order on the set of candidates
- All the preferences (of the n voters) **are contained in an Excel file or a csv file.**

This work aims at computing in python language the four voting rules introduced in Chapter 3. You can use the examples of this chapter to test your functionalities, especially the following example where $m = 4$ candidates $\{a, b, c, d\}$ and $n = 27$:

5 voters :	$a \succ b \succ c \succ d$
4 voters :	$a \succ c \succ b \succ d$
2 voters :	$d \succ b \succ a \succ c$
6 voters :	$d \succ b \succ c \succ a$
8 voters :	$c \succ b \succ a \succ d$
2 voters :	$d \succ c \succ b \succ a$

1. Compute a function `MajorityRule` returning the result of a simple majority rule voting, between two candidates.
2. Compute a function `Plurality` returning the result of a plurality voting.
3. Compute a function `PluralityRunoff` returning the result of a plurality Runoff voting (plurality with two rounds).
4. Compute a function `CondorcetVoting` returning the result of the application of the Condorcet principle (the existence of the Condorcet winner).
5. Compute a function `BordaVoting` returning the result of the application of the Borda principle.