amse
aix marseille
school of economics

Working Papers / Documents de travail

Allocation of Ordered Exclusive Choices

Marc Sangnier

WP 2013 - Nr 27

Aix*Marseille université · cnrs · L'ECOLE DES HAUTES ETUDES EN SCIENCES SOCIALES · Inserm · IRD Institut de recherche pour le développement · Centrale Marseille

Allocation of ordered exclusive choices

Marc Sangnier*

April 2013

Abstract

This note describes the Stata command `alloch` which helps to allocate exclusive choices among individuals who have ordered preferences over available alternatives.
KEYWORDS: `alloch`, random allocation, choice criterion

1 Introduction

Scarcity makes exclusive choices a frequent feature of the real world as everyone cannot get their preferred outcome when someone's choice prevents an object to be allocated to someone else. Such situations arise when allocating pupils to schools or when deciding where friends will sit around a table. Similarly, teachers often face situations where they need to allocate students to different tasks, e.g. presentations or book reviews. In this note, I present a simple command–`alloch`–that handles situations where each choice can only be attributed to one individual. This command has initially been designed to allocate students to classroom presentations but can be used to allocate workers to individual tasks or employees to desktops located in different places. The command performs this task in flexible ways by (i) taking care of individual preferences clashes and (ii) allowing different criteria to look for the "best" allocation.

The `alloch` command enables the user to quickly allocate individuals to choices they expressed. It simply uses a data set that contains ordered choices made by individuals. Its basic principle is to randomly draw a first individual and to give him his preferred choice. Then, a second individual is drawn and receives his first choice if it is still available, otherwise he receives his second

*Aix-Marseille Univ. (Aix-Marseille School of Economics), CNRS & EHESS.

choice. And so on, and so forth. A property of the resulting allocation of choices among individuals is that it is renegotiation proof: once individuals are made aware of the resulting allocation, no one has any interest to exchange with another one.

As it relies on random draws, many different allocations may be obtained using this process and the user may be tempted to look for the "best" one. This requires a criterion to discriminate between feasible allocations. Should we prefer to make happy Mr. or Ms. Smith? The `alloch` command proposes several criteria to choose among allocations. To achieve this, the command first creates a given number of different allocations, and then chooses the most efficient one according to the chosen criterion.[1]

2 Required structure of the data, syntax and options

The data used by the `alloch` command must be structured as follows. Each line must contain the ordered choices of a separate individual over a list of alternatives. Each individual must be uniquely identified by a list of variables. The chosen objects must be identified by consecutive identification numbers ranging from 1 to N, where N is the total number of alternatives available to individuals. The most preferred choice of each individual must be contained by a variable that ends with 1 (for example, *pref_1*). The second preferred choice must be contained by a variable that ends with 2 and starts with the same name as the previous one (for example, *pref_2*).

In practice, the data used by the command can be constructed in two steps. First, provide individuals with a list of numbered alternatives and ask them to order alternatives using alternatives' identification number. Note that the number of available alternatives must be at least as large as number of individuals.[2] Then, append these ordered preferences in a single data set where each line correspond to the preferences of a single individual and where each individual has a unique identifier.[3]

The command is designed to deal with choices badly filled by individuals. Individuals may rank the same alternative in two different places (for example,

[1] In such a framework, algorithms to determine the "best" allocation could be used *ex ante*. However, such processes start to be very demanding once the numbers of choices and individuals becomes large. This is why this part of the `alloch` command rather relies on a series of random draws.

[2] Otherwise, it would be impossible to allocate each individual a different choice.

[3] Online forms or spreadsheets are very convenient to achieve these preliminary steps.

rank the same object as first and fourth preferred choices). They may also leave a blank position (that is, not provide an object for the second choice, but fill the first and the third ones). Finally, they may rank only an insufficient number of objects (for example, rank only two objects out of a list of four while objects will be allocated among three individuals). By default, the `alloch` command does not treat such individuals in any particular way. That is, their are allocated their preferred choice if it is still available when the individual is drawn, otherwise, they receive a random choice once the process is over for all individuals who did not make any error when expressing their preferences. Under the `sanction` option (see below), such individuals are removed from the pool at the beginning of the process and randomly allocated to one of the remaining objects at the very end of the process, even if their preferred choice is still available.

The `alloch` command has been developed under Stata version 11.2. Its syntax is following:

`alloch` [`using` *filename*], `choice(`*var*`) alter(`#`) ident(`*varlist*`)` [...]

The command uses the active data set if `using` *filename* is not specified. Required and optional options are described below.

`choice(`*var*`)` is required. *var* must be the common part of names of variables that contains the ordered choices. For example, if the most preferred choice is entered in variable *pref_1* and the second preferred choice is entered in variable *pref_2*, then *var* must be set to *pref_*. Note that choices variables must be numerated consecutively.

`alter(`#`)` is required. # must be set to the total number of different alternatives available to individuals. For example, if individuals had to choose among 30 different items, then # must be set to 30, whatever the number of choices that have been expressed. This option is necessary to deal with individuals for which the set of expressed choices is incomplete or badly filled.[4]

`ident(`*varlist*`)` is required. *varlist* must be set to the variable (or variables' list) that uniquely identify observations.

`iter(`#`)` is optional. This option specifies the number of iterations. By default, # equals 100.

`linear` and `quadratic` are optional. These options specifies the criterion used to choose among several distributions. The default criterion is a Rawlsian one where the chosen distribution is the one that minimizes the number of individuals with low-ranked choices.[5] `linear` minimizes the sum of ranks.

[4] See above for mode details about the required data structure.

[5] See [Rawls, 1974]. This criterion is also known as "maxmin". Technically, this criterion

`quadratic` minimizes the sum of squared ranks. `linear` and `quadratic` may not be used simultaneously.

`unique` is optional. This option specifies to run only one random allocation and to make it the final one. `unique` offsets `linear`, `quadratic` and `iter(#)`.

`sanction` is optional. This option specifies to exclude from iterations all individuals for which at least one choice is missing or who put the same choice in different positions. These individuals will not be used when choosing among distributions and will be randomly allocated to a choice when the process is over. By default, such individuals are kept in the process and allocated to one of their choices if possible.

`detail` is optional. This option specifies to return precise information about individuals with missing or redundant choices.

`plot` is optional. This option produces a simple histogram of the chosen allocation of choices.

`alloch` preserves the used data set and returns a data set that contains observations' unique identifier and three variables: *outcome* displays the identification number of the allocated choice, *draw* displays the position at which the individual has been drawn, and *choice_rank* presents the rank of the choice allocated to the individual.

3 Example

Table 1 presents the general structure of data used by the `alloch` command. Here, 44 individuals identified by their name and their first name have been asked to choose among 50 different choices. To ensure consistency, the planner required them to express 47 choices.[6] These data are original choices expressed by students over presentations to be allocated.[7] Let us type:

```
use alloch_example, clear
alloch, alter(50) ident(first_name name) choice(pref_) iter(10)
```

Since option `iter(#)` is set to 10, one would obtain the "best" out of ten allocations from the Rawlsian point of view as `linear` and `quadratic` options are not specified. That is, the resulting allocation is such than the nine other

compares any pair of allocations and selects the one that allows the worst-off individuals to receive lower-ranked choices. This command implement a lexicographic Rawlsian approach as it tries to improve the outcome of the second worst-off individual if it is not possible to improve the outcome of the worst-off individual, and so on.

[6]Note that choices ranked from 45 to 47 are useless. Such a situation may however occur when the planner over-estimated the number of respondents when gathering preferences.

[7]Students' names have been replaced by names of characters from The Simpsons.

allocation do not provide the worst-off individual with lower-ranked choices.[8] Resulting data are presented in table 2. This table means that O. Simpson was the first to be drawn and received his first choice, i.e. object number 26. On the opposite, R. Wolfcastle was the last to be drawn and received his ninth choice. The Stata output is as follows:

```
. alloch, alter(50) ident(first_name name) choice(pref_) iter(10)
Progress (10)
..........
------------------------------------------------------------------------------
Random allocation of ordered exclusive choices
------------------------------------------------------------------------------
Warning: Observations with the same choice in different positions.
Observations #: 12, 43
Warning: Observations with missing choices.
Observations #: 26, 30
------------------------------------------------------------------------------
Number of iterations: 10
Choice criterion: Rawls
------------------------------------------------------------------------------
Number of choices: 47
Number of alternatives: 50
Number of observations: 44
Number of post-allocated: 0
------------------------------------------------------------------------------
```

The first block of information tells the user that individuals 12, 26, 30, and 43 did not submit enough choices or put the same choice in more than one position. The second block of information recalls the criterion used and the number of iterations required by the user. Finally, the last output block displays basic information about the process. The initial data set contain the 47 ordered choices (from a list of objects numbered from 1 to 50) expressed by 44 individuals. In the resulting allocation, none of the four individuals with badly filled preferences have been randomly assigned to choices.[9] By adding options `plot`, the command would have produced figure 1. This figure means that the resulting allocation managed to provide 29 individuals with their most preferred choice, 11 with their second preferred choice, etc.

Assume that the planner wants to impose sanctions on the individuals who made errors in the expression of their preferences. Let us type:

```
use alloch_example, clear
alloch, alter(50) ident(first_name name) choice(pref_) iter(10) sanction
detail
```

[8] See the description of `linear` and `quadratic` options for additional insight about the Rawlsian criterion.

[9] If one of them had been drawn at a point where the list of still available objects had not contained one of his expressed choices, then the *number of post-allocated* individuals would be different from zero.

The Stata output is as follows:

```
. alloch, alter(50) ident(first_name name) choice(pref_) iter(10) sanction deta
> il
Progress (10)
..........
------------------------------------------------------------------------------
Random allocation of ordered exclusive choices
------------------------------------------------------------------------------
Warning: Observations with the same choice in different positions.
Leonard Lenny (12), pref_15=pref_18
Van Houten Milhouse (43), pref_5=pref_6, pref_1=pref_9
Warning: Observations with missing choices.
Wiggum Ralph (26), pref_9
Simpson Homer (30), pref_47, pref_46, pref_45, pref_44, pref_43, pref_42, pref_
> 41, pref_40, pref_39, pref_38, pref_37, pref_36, pref_35, pref_34, pref_33, p
> ref_32, pref_31, pref_30, pref_29, pref_28, pref_27, pref_26, pref_25, pref_2
> 4, pref_23,
------------------------------------------------------------------------------
Number of iterations: 10
Choice criterion: Rawls
------------------------------------------------------------------------------
Number of choices: 47
Number of alternatives: 50
Number of observations: 44
Number of sanctions: 4
------------------------------------------------------------------------------
```

By adding the `detail` option, the user is informed about the identity and the precise errors of the four individuals who made errors. Here, L. Leonard ranked the same object in positions 15 and 18. M. Van Houten also performed similar errors. R. Wiggum forgot to fill his ninth choice and H. Simpson simply stop ranking objects after his 22^{nd} preferred choice. By specifying the `sanction` option, the user requests to remove these four individuals from the process and to allocate them random choices after all others individuals have been drawn.

References

[Rawls, 1974] Rawls, J. (1974). Some reasons for the maximin criterion. *The American Economic Review*, 64(2):pp. 141–146.

Table 1: Structure of data used by the command.

first_name	name	pref_1	pref_2	pref_3	...	pref_45	pref_46	pref_47
Kent	Brockman	37	19	38		47	2	39
Bart	Simpson	41	26	47		24	28	45
Homer	Simpson	33	19	31				
⋮	⋮							
Milhouse	Van Houten	4	9	15		36	20	32
Rainier	Wolfcastle	48	17	25		45	12	19

Table 2: Structure of data produced by the command.

first_name	name	outcome	draw	choice_rank
Orville	Simpson	26	1	1
Gros	Tony	29	2	1
⋮	⋮			
Cyrus	Simpson	10	43	1
Rainier	Wolfcastle	28	44	9

Figure 1: Distribution of allocated choices.