

Sample – a FORTRAN program for taking series of random samples from populations

Version 1.0

Werner Ulrich

Nicolaus Copernicus University in Toruń

Department of Animal Ecology

Gagarina 9, 87-100 Toruń; Poland

e-mail: ulrichw @ uni.torun.pl

Latest update: 24.11.2004

1. Introduction

Sample is a small program that takes series of random samples from a given population. In ecological research this is sometimes necessary for constructing null models or for jackknifing and bootstrapping.

The idea behind *Sample* is simple. You have a data set or a whole matrix with several data sets. The elements of this data sets (the items) occur at different proportions (probabilities, numbers, frequencies). The problem is to take random samples from these data sets that match the different occurrences of elements. For instance, if you have a community of 5 species (items) with 100, 50, 30, 20, and 10 individuals (occurrences) and take 20 individuals at random you expect that your sample will have a similar frequency distribution than the population. *Sample* takes such random samples by picking single individuals from the populations until a predefined sample size is reached.

2. Data structure

Sample needs unformatted ASCII files (text files) as shown in the table below. The first lines must always be a comment lines that starts with an asterisk (*). The data sets need not to be sorted. *Sample* throws out zeros.

However, to read the ASCII file properly columns must be sorted according to the number of (items) elements, starting with the item richest data set.

Sample handles either absolute values or frequencies. If the maximum value and the minimum value of the input data are in the range of 0 to 1 the program assumes frequencies and divides all

* Data matrix

102	54	84.32	17.705	49
954	302	149.39	273.713	16
342	1	260.38	0.631	114
2	5	0.90	1.367	1
4	8	3.24	2.903	2
6	47	3.87	43.081	2
56	20	8.29	3.105	5
20	28	8.85		
17	40			
584				
350				

data through the minimum value. These corrected values (sorted from largest to smallest) are also the output values.

3. Program run

The program asks first what do to, sampling until a predefined number of items or a predefined number of occurrences is reached. For instance, if the first data set of the previous table is to be sampled the program could sample occurrences until the sample contains three of the eleven items of this data set. It could also sample 50 of the 2437 occurrences irrespective of the number of items in

* Sample sizes	the sample.
0.102	Then the program asks whether
0.954	to sample with replacement (it
0.342	est the original number of occa-
0.2	sions remains constant) or with-
0.4	out. In the latter case the data set
0.6	is reduced by the sampled occa-
0.56	sions.
0.2	<i>Sample</i> needs a second input file
0.17	that defines how many items or
0.584	
0.35	

occurrences have to be sampled. This file must also be an unformatted ASCII file with a comment line at the beginning. The example beside shows such a file that defines a sample protocol where successively samples will contain 10.2%, 95.4%, 34.2% and so on items or occasions. The values of the definition file contains therefore proportions (values between 0 and 1).

The program samples successively from all data set of the matrix.

4. The Output file

Sample produces a single output file, *Sample.txt*. This file is shown on the next page. A community of 31 species and 628 occasions had been sampled with replacement until 0.102 (=4),

0.05 (=2), 0.142 (=5),... species had been found (the next integer of $0.102 \cdot 31 = 3.16$ is 4). The first column gives the original data set.

5. System requirements

Sample is written in FORTRAN 95 and runs under Windows 9.x, and XP. Computation abilities are only limited by the computer's memory.

6. Citing *Sample*

Sample is freeware but nevertheless if you use *Sample* in scientific work you should cite *Sample* as follows:

Ulrich W. 2003 - *Sample* – a FORTRAN program for taking series of random samples - www.uni.torun.pl/~ulrichw

8. Acknowledgements

The development of this program was supported by a grant of the Polish Science Committee (KBN, 3 F04F 03422).

