PCGEN: A FORTRAN IV program to generate paired-comparison stimuli

GORDON A. ALLEN
Miami University, Oxford, Ohio 45056

and

LAWRENCE M. BALDWIN
Brown/Butler Family Therapy Training Program,
Butler Hospital, Providence, Rhode Island 02906

When using the method of paired comparisons, generating the presentation order of stimuli poses several problems for investigators. First, every stimulus item should appear in each position of a pair equally often. Second, the number of pairs that intervene between successive presentations of the same stimulus item should be equated, so as not to introduce any undue consistency or bias on the part of the subject. A third problem arises when the number of stimulus items is large. The number of pairs generated by the method is $N(N - 1)/2$, where N is the number of stimulus items. For example, when N is 15, the number of pairs is 105. As the number of stimulus items becomes large (e.g., 25), fatigue and boredom become major contributors to the subject's later choices.

Several algorithms have been employed to generate the pairs. Simple randomization is probably the most frequently used procedure, although it does not explicitly address any of the problems. An algorithm developed by Ross (1934) addresses the first two problems but not the third. PCGEN is a program that implements an algorithm that addresses all three. In addition, the program can generate stimulus materials.

The algorithm divides the set of stimulus pairs into blocks such that each stimulus item appears twice within each block, with each stimulus item appearing in both positions. The presentation order of the pairs within a block is randomized subject to the constraint that at least one pair appear between pairs containing the same stimulus item.

Table 1 illustrates pairs allocated to blocks. The triangular matrix represents the set of stimulus pairs making up one complete replication. The row items represent the left stimulus of a pair, and columns represent the item on the right. Entries in the matrix are the block numbers of the stimulus pair. Thus the pair C-A is in Block 2, and the pair C-B is in Block 1.

Two algorithm features must be considered when

Table 1
Example of PC Pairs Using Five Items

Left Stimulus	Right Stimulus			
	A	B	C	D
B	1			
C	2	1		
D	2	2	1	
E	1	2	2	1

selecting stimulus sets. First, with an odd number of stimuli, there are $(N - 1)/2$ blocks, each containing N pairs of stimuli. With an even number of stimuli, there are $N/2$ blocks; one of these contains only $N/2$ pairs of stimuli, and the rest contain N pairs. Table 1 illustrates a simple case with an odd number of stimuli; a similar table is easily constructed for the even case. The unequal block sizes for the even case is sometimes undesirable, such as when subjects are being given incomplete designs.

The other characteristic of the algorithm that might lead to problems is the fact that stimulus pairs within a block are equidistant from each other with respect to position on the input list. If the stimuli are ordered (e.g., by weight) and their natural order corresponds to their order of input to the program, then the rank difference between item pairs in a block will be restricted and in fact contain only two levels. This is seen in the example in Table 1, in which four of five stimulus pairs in Block 1 contain adjacent letters in the alphabet. This problem can be easily avoided by randomizing the stimulus item order.

Input and Options. The user must supply certain job parameters; in addition, there are several options. The first job parameter is the number of stimulus items, preferably an odd number. The second parameter is the number of replications that are to be generated. Third is the number of blocks within a replication that are to be generated without a pause or break in the output. This capability is useful when the computer output is given directly to the subject. The implementation of the pause is either a blank computer card or a skip to the top of the next page.

The stimulus pairs generated are of two types: stimulus identification numbers (the numbers 1 through N are created within the program) and the stimulus pairs, which may consist of the labels of the stimuli or the stimuli themselves if they are verbal. The user enters the labels, which may be up to 30 characters long. If labels are being used, a further option is to specify whether the labels are to be shuffled after each complete replication has been produced. If this option is not specified, the similar blocks of each replication will contain the same stimulus pairs, but the presentation order will differ. Another option is that stimulus ID numbers may be

produced beside the stimulus pair; this option usually makes a computer analysis easier. A title, printed at the top of each subject's page, may be specified.

Output. Depending upon the option specified, the output consists of a list of pairs of stimulus ID numbers or pairs of stimulus labels. The output medium may be a printed listing, a deck of punched cards (which needs to be interpreted), or a data file.

Language and Computer. Two versions are currently available. Both versions are written in FORTRAN IV. Both versions require a random number generator (RANDU). One version is interactive and is written for a PDP-11/10 operating under RT-11. The usual output is a printed listing, but it can be a data file. The other version is written for an IBM 370 using batch processing. The output is either a printed list or a deck of punched cards.

Limitations. The current versions are limited to 30 stimulus items (435 stimulus pairs per replication). The maximum label length is 30 characters. Both of these values can be increased easily.

Availability. A listing of the program is available at no cost. A copy of the program on cards or a floppy disk is available at cost from Gordon Allen, Psychology Department, Miami University, Oxford, Ohio 45056.

REFERENCE

Ross, R. T. Optimum orders for the presentation of pairs in the method of paired comparison. *Journal of Educational Psychology*, 1934, 25, 375-382.