French, R. M. & Perruchet, P. (2009). Generating constrained randomized sequences: Item frequency matters. *Behavior Research Methods.* (in press).

Generating constrained randomized sequences: Item frequency matters

Robert M. French and Pierre Perruchet
LEAD-CNRS UMR 5022, University of Burgundy, Dijon, France
{robert.french, pierre.perruchet}@u-bourgogne.fr

Abstract

All experimental psychologists understand the importance of randomizing lists of items. However, randomization is generally constrained and these constraints, in particular, not allowing immediately repeated items, which are designed to eliminate particular biases, frequently engender others. We describe a simple Monte Carlo randomization technique that solves a number of these problems. However, in many experimental settings, we are concerned not only with the number and distribution of items, but also with the number and distribution of transitions between items. The above algorithm provides no control over this. We, therefore, introduce a simple technique using transition tables for generating correctly randomized sequences. We present an analytic method of producing item-pair frequency tables and item-pair transitional probability tables when immediate repetitions are not allowed. We illustrate these difficulties – and how to overcome them – with reference to a classic paper on infant word segmentation. Finally, we make available an Excel file that allows users to generate transition tables with up to ten different item types, and to generate appropriately distributed randomized sequences of any length without immediately repeated elements. This file is freely available at: http://leadserv.u-bourgogne.fr/IMG/xls/TransitionMatrix.xls

Introduction

All experimental psychologists understand the importance of randomizing lists of items. Randomization is, arguably, the most widely used and most effective means of eliminating order biases. However, there are virtually always constraints on the items to be randomized and the problem, too often overlooked, forgotten or ignored, is that these constraints, designed to eliminate particular biases, frequently engender others.

Consider a simple problem, one that most experimental psychologists have faced at one time or another and that some face every time they design an experiment – namely, randomizing a list of items of different frequencies without immediate repetitions. Creating such a list is generally considered to be relatively straightforward. It turns out, however, that to do this correctly is considerably harder than it would seem. This paper will point out a very serious bias introduced by a standard, widely used list randomization algorithm, will show that constrained randomization can engender other problems that are unrelated to the specific randomization algorithm chosen and will introduce a series of techniques that allow these problems be avoided. In this paper we will undertake an analysis of list randomization under the simplest, arguably most universal, and seemingly innocuous, of all constraints — namely, the prohibition of immediately repeated identical elements.

Removing immediately repeated items from randomized lists

There are many reasons why immediately repeated identical elements must be removed from sets of familiarization items and sets of test items. Removing repeated items is almost universally practiced among experimental psychologists, except, of course in studies aimed at

investigating the specific effect of immediate repetitions (e.g., in repetition priming studies, Jacoby, 1983, or in studies on massed practice, Seabrook, Brown , & Solity, 2004). For instance, in word segmentation studies using a continuous speech stream (e.g., Saffran, Newport, & Aslin,1996), the immediate repetition of artificial words is universally prohibited in the familiarization language. Likewise, there is no repetition in studies using serial reaction times tasks (SRT, e.g., Nissen & Bullemer, 1987). In a large range of domains, immediate repetitions of items during the test phase is also avoided to prevent the appearance of sequential effects.

A serious problem with a standard list randomization algorithm

We begin by examining a very widely used list randomization algorithm to randomize a list of items in such a way that there are no immediately repeated items. For the sake of illustration, we will start with a set W of 45 As, 45 Bs, 90 Cs, and 90 Ds and draw from this set. We create the randomized item list, S, by randomly drawing items from W and adding them to the end of S. If a newly chosen item from W is the same as the previously chosen item added to S, the new item is returned to W and another item is drawn from W. This is continued until W is empty. This is a modification of an algorithm described in Brysbaert (1991, Algorithm 10). This algorithm, when immediately repeated items are allowed, is perfectly appropriate for generating correctly randomized sequences. However, when it is modified as described above in order to eliminate repeated items – as it very often is – very serious problems can result.

It turns out that this commonly used randomization algorithm will produces a dramatic bias in S when the item frequencies in W are different. Since there are twice as many Cs and Ds in W as As and Bs, we expect the ratio of As (or Bs) to Cs (or Ds) to be 0.5 throughout the list. However, the requirement of no immediately repeated items causes this algorithm to produce a list in which the ratio of A's (or B's) to C's (or D's) is significantly greater than 0.5 (around 0.6) in the first fifth of the list and considerably less than 0.5 (approximately 0.3) over the final fifth of the list (Figure 1).

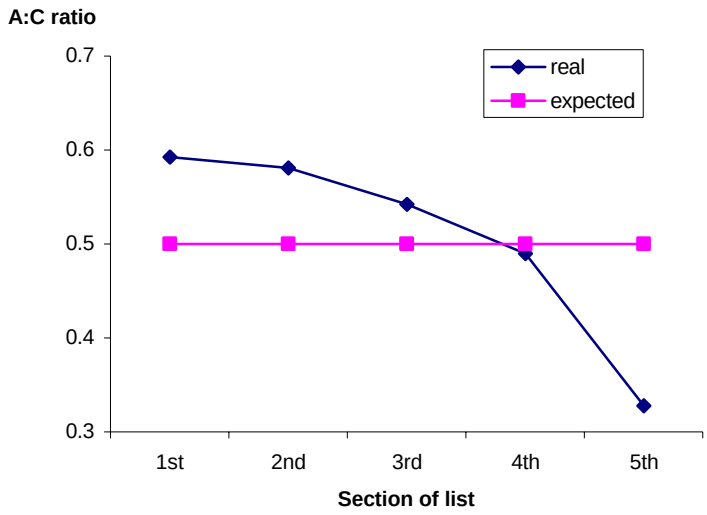

Figure 1. The distribution of the ratio of less-frequent to more-frequent items in a list produced by a standard list randomization algorithm. Differing item frequencies in the items to be randomized result in significant differences with expected item frequencies in different sections of the final list.

Depending on the experiment being run, this extreme imbalance across the list could result in frequency effects being confounded with primacy or recency effects. For instance,

an investigator could be led to minimize the impact of item frequency on memory simply because less frequent items tend to occur more often than expected at the beginning of the study list, hence benefiting from a greater primacy effect than more frequent items.

The problem is caused by returning repeated items to W when certain items are more frequent than others. In our example, there are twice as many C's as A's in W. Consequently, there will be at least *four* times as many CC repeats as AA repeats when creating S. To see this, assume that we had simply randomized W without worrying about repeated items. In this case, the probability of an AA pair (requiring an A to be returned to W) would be 1/36 because the $p(A) = 1/6$; the probability of a CC pair (requiring a C to be returned to W) would be 1/9 because $p(C) = 1/3$. In other words, we will have to return *four times* as many Cs to W as As, even though Cs occur only twice as often as As in the original list, W. This causes the beginning of the randomized list, S, to be disproportionately overloaded with A's and B's and the end of the list to be disproportionately overloaded with C's and D's.

An efficient "distributed" randomization algorithm

The item imbalance problem associated with this algorithm can be eliminated by using a simple "distributed" randomization algorithm. We begin by putting all of the items from W that will make up the randomized list into S. We randomly shuffle these items and then remove the repeated element of all immediately repeated pairs of items. These repeats are put in a list R. This ensures that S, a list of length *n*, now contains no immediately repeated items. We then create an index list, I, consisting of the numbers from 1 to *n* in random order. We pick an element, *r*, from R and run down the index list, I, attempting to find an index at which to insert *r* into S that will satisfy the order constraints of the desired sequence. Once *r* is inserted into S, S will have length *n+1*. We now create a new randomized index list consisting of the numbers from 1 to *n+1*, and pick a new item from R to be inserted in S, etc. If ever a given *r* cannot be inserted into S, we return it to R and pick another *r*, and so on until all of the items of R have been inserted into S. This algorithm is fast (for example, on a PC running Windows XP with 1.2 GHz processor, a Matlab implementation of the algorithm takes less than one second to create 100 randomized lists of 270 items) and eliminates the item distribution imbalances created by the standard randomization algorithm discussed in the previous section.

However, in many experimental settings, we are concerned not only with the number and distribution of items, but also with the number and distribution of transitions between items. The present algorithm provides no control over this. The number of various item-pairs can vary significantly from sequence to sequence when using this algorithm. To solve this particular problem, as well as a number of others that we discuss below, we now introduce a simple and efficient technique for generating correctly randomized sequences.

Transition-frequency and transition-probability tables

Unfortunately, imbalances in item frequencies across a list are not the only problems that arise when randomizing lists on which there are even simple order constraints. In order to develop a method of revealing these problems and, ultimately, creating correctly randomized lists, in general, we need to introduce the notion of transition tables. Instead of focusing on the frequencies of the items in a list, a transition table allows us to keep track of the frequencies of the transitions between items in a list.

The simplest use of transition tables is as an accounting tool. If we have a particular sequence, say, *ACABCBABCABACBCBAC*, we can tally the number of immediately adjacent items (we will call these item-pairs or transitions) and record these tallies in a table. We could draw up the following item-pair frequency table (Table 1a) for the sequence above:

		Second item		
		A	**B**	**C**
First item	**A**	0	3	3
	B	3	0	3
	C	3	3	0

Table 1a. Item-pair frequency table

Item-pair frequency tables can easily be converted to conditional probability transition tables by dividing all of the elements of each row by the total number of elements in that row (Table 1b). Each cell in this case will contain the probability that, given the first item, the second item will follow it. If we start with an A, one-half of the time it will be followed by a B, one-half of the time by a C, and never by an A. This is expressed as $p(B|A) = 0.5$, $p(C|A) = 0.5$, $p(A|A) = 0$.

		Second item		
		A	**B**	**C**
First item	**|A**	0	0.5	0.5
	|B	0.5	0	0.5
	|C	0.5	0.5	0

Table 1b. Conditional probability transition table

Finally, we can also derive a table of joint probabilities from Table 1a. This table (Table 1c) is obtained by dividing the individual numbers of transition by the total number of all transitions. There are a total of 16 transitions and we thus obtain:

		Second item		
		A	**B**	**C**
First item	**A**	0	.167	.167
	B	.167	0	.167
	C	.167	.167	0

Table 1c. Joint probabilities of item pairs

In other words, these tables provide a detailed description of a particular sequence. But transition tables are far more than an accounting tool for characterizing a specific sequence.

Creating transition tables based on sequence constraints

Now, instead of starting with a specific sequence and counting the item-pair frequencies to create an item-pair frequency table, as was done above, we start with the properties that we would like our sequences to have, determine the item-pair frequency table (or conditional probabilities table) for all sequences with those properties, and then use these tables to generate the randomized sequences we need.

Remillard and Clark (1999) and Remillard (2008) discuss the use of transition tables to generate correctly randomized sequences and present an efficient algorithm for doing so. We also present a simple algorithm for sequence generation at the end of the present article that starts with a transition table. But a major problem remains – namely, how to create the transition tables that are used as input to these algorithms. As we will see, this is not a

particularly simple problem. One of the main goals of this paper is to show how to correctly derive these transition tables starting with the desired item frequencies.

For example, assume we wish to create sequences with 12 As, 12 Bs and 12 Cs, uniformly distributed with no immediately repeated items. We create the conditional probability transition table by reasoning that if we have just drawn an A, then in order to avoid drawing another A, we restrict ourselves to drawing Bs and Cs, of which there are 12 and 12, respectively. Therefore, the conditional probability of drawing either a B or a C after having just drawn an A, is 0.5. Similarly, if we draw a B, the conditional probability of drawing an A is 12/(12+12) = 0.5 or a C is 12/24 = 0.5. Finally, if we've drawn a C, then the probability of drawing an A is 12/24 = 0.5 and of drawing a C is 12/24 = 0.5. This gives us the following conditional probability transition table and item-pair frequency table (Tables 2a and 2b) for *any* randomized sequence with 12 As, 12 Bs, 12 Cs and no immediately repeated items.

These tables are perfectly correct and can be used to generate the desired sequences.

		Second item		
		A	**B**	**C**
First item	**|A**	0	0.5	0.5
	|B	0.5	0	0.5
	|C	0.5	0.5	0

		Second item		
		A	**B**	**C**
First item	**A**	0	6	6
	B	6	0	6
	C	6	6	0

Tables 2a and 2b. Conditional probabilities and item-pair frequency table derived from initial constraints

Now, consider creating a similar item-frequency table for sequences with 6 As, 12 Bs, and 12 Cs and no immediately repeated items. Exactly as before, we reason that if we have just drawn an A, then in order to avoid drawing another A, we restrict ourselves to drawing Bs and Cs, of which there are 12 and 12, respectively. Therefore, the conditional probability of drawing either a B or a C after having just drawn an A, is 0.5. Similarly, if we draw a B, the conditional probability of drawing an A is 6/(6+12) = 0.33 or a C is 12/18 = 0.67. Finally, if we draw a C, then the probability of drawing an A is 6/18 = 0.33 and of drawing a C is 12/18 = 0.67. This gives us the following conditional probability transition table and item-pair frequency tables (Tables 3a and 3b) for any randomized sequence with 6 As, 12 Bs, 12 Cs and no immediately repeated items.

		Second item		
		A	**B**	**C**
First item	**|A**	0	0.5	0.5
	|B	0.33	0	0.67
	|C	0.33	0.67	0

		Second item		
		A	**B**	**C**
First item	**A**	0	3	3
	B	4	0	8
	C	4	8	0

Tables 3a and 3b. Incorrect conditional probabilities and item-pair frequency tables for 6 As, 12 Bs, and 12 Cs.

The problem is that Tables 3a and 3b are *false* and the logic used to generate them – seemingly identical to the logic that was used to create the correct Tables 2a and 2b – is incorrect! The correct tables are Tables 4a and 4b.

		Second item		
		A	**B**	**C**
First item	**|A**	0	0.5	0.5
	|B	0.25	0	0.75
	|C	0.25	0.75	0

		Second item		
		A	**B**	**C**
First item	**A**	0	3	3
	B	3	0	9
	C	3	9	0

Tables 4a and 4b. Correct conditional probabilities and item-pair frequency table for 6 As, 12 Bs, and 12 Cs. (The values in the shaded cells differ from those in Tables 3a and 3b.)

In other words, if we wish to generate a correctly randomized sequence of 6 As, 12 Bs and 12 Cs with no immediately repeated items, we must use Tables 4a or 4b, whose derivation is not obvious, and not Tables 3a or 3b! In other words, creating sequences in which there are no immediate item repetitions and where item frequencies differ requires tools like those developed in this paper to transform our desired sequence properties into correct item-pair frequency tables from which we can generate the correctly randomized sequences.

Checking and generating randomized sequences

First, we will show some general properties of item-pair frequency tables for correctly randomized sequences without immediately repeated elements. We will then develop a simple, general method of using initial item frequencies to produce the correct item-pair

frequency table that can be used to generate correctly randomized sequences with no immediately repeated items.

General properties of randomized sequences without immediate item repeats

We will start with an item-pair frequency table (Table 5) corresponding to correctly randomized sequences in which there are four item types and no immediately repeated items. We will derive the general properties of such a table. The frequencies of each item type are N_1, N_2, N_3, and N_4. We will then show how this table can be used to analyze a real problem. We will then generalize this technique to tables with any number of item types.

		Second item				
		1	**2**	**3**	**4**	sub-totals
First item	**1**	0	n_{12}	n_{13}	n_{14}	N_1
	2	n_{21}	0	n_{23}	n_{24}	N_2
	3	n_{31}	n_{32}	0	n_{34}	N_3
	4	n_{41}	n_{42}	n_{43}	0	N_4

Table 5. A general item-pair frequency table for randomized sequences with 4 item types

The requirement of a random distribution of items will ensure that $\forall_{i,j}\ n_{ij} = n_{ji}$ (This is because there is no a priori reason that, if all items are randomly distributed across the list, that for a given item-type more items of another item-type will preferentially precede or follow it.) Under these conditions the following relations hold for the number of item-pairs in each cell of the item-pair frequency matrix:

$$\frac{N_1 + N_2}{2} - n_{12} = \frac{N_3 + N_4}{2} - n_{34}$$

$$\frac{N_1 + N_3}{2} - n_{13} = \frac{N_2 + N_4}{2} - n_{24}$$

etc…

Analyzing a real example

A classic experiment on infant segmentation of continuous speech (Aslin, Saffran and Newport, 1998) relies on the relation between item frequencies and between-item transition frequencies in the sequence of syllables constituting the familiarization sequence. This experiment has not only been frequently cited but its design has be used by other researchers on several occasions (Graf Estes, Evans, Alibali, & Saffran, 2007). The experimental design called for a set of 45 As, 45 Bs, 90 Cs and 90 Cs and allowed no immediate repeats. For reasons that need not concern us, the number of As had to be equal to the number of CD-pairs.

But the above equations reveal a surprising fact, one that has, moreover, escaped notice for the last ten years – namely, that it is impossible to construct such a list unless we accept that it contains *no* AB or BA pairs!

That there can be no AB or BA transitions follows immediately from the fact that $\frac{N_1 + N_2}{2} - n_{12} = \frac{N_3 + N_4}{2} - n_{34}$. This implies that: $\frac{45+45}{2} - n_{12} = \frac{90+90}{2} - n_{34}$ From this we conclude that $n_{12} = n_{34} - 45$. In other words, there must be 45 fewer AB-transitions than CD-transitions. But one of the constraints of the design is that the number of CD-pairs is equal to the number of As (i.e., 45). Thus, $n_{34} = 45$. From this it follows immediately and necessarily that $n_{12} = 0$. In other words, we can fulfill the constraints of the Aslin et al. design *only* if there are *no* AB or BA pairs in the list. Obviously, the complete absence of these transitions could potentially have had a significant effect on their results.

Generalizing the equations

The above relations can be generalized to any item-pair frequency table with any number of item types. In general, we have:

$$\forall k \quad n_{kk} = 0 \quad and \quad \forall_{i,j} \; n_{ij} = n_{ji} \qquad (1)$$

$$\forall i, j \quad \frac{N_i + N_j}{2} - n_{ij} = \frac{\sum_{\forall k \neq i,j} N_k}{2} - \sum_{\forall u,v \neq i,j} n_{uv} \qquad (2)$$

Constructing the right transition table

The correct item-pair frequency table (or, equivalently, the conditional probability transition table) corresponding to the constraints of a given problem can be notoriously hard to construct and, as the example in Tables 3a and 3b shows, requires tools more sophisticated than the "obvious" reasoning that led to the construction of these incorrect tables.

Let us return, again for the purposes of illustration, to the paper by Aslin et al. (1998), in which the key randomized sequence was supposed to contain 45 As, 45 Bs, 90 Cs and 90 Ds with no immediately repeated items. Relying on the same (erroneous) logic that produced Tables 3a and 3b, Aslin et al. derived the conditional probabilities that were at the heart of their study. These are the seemingly "obvious" conditional probabilities shown in Table 6a. So, for example, to avoid immediate repeats, an A can only be succeeded by a B, C, or D. Since there are 45 Bs, 90 Cs and 90 Ds, this would seem to imply that the probability of drawing a B after an A is $p(B \mid A) = \frac{45}{45+90+90} = \frac{1}{5} = 0.2$. In a similar manner, we can calculate all of the other conditional probabilities, allowing us to create Table 6a.

		Second item			
		A	**B**	**C**	**D**
First item	**A**	0	0.2	0.4	0.4
	B	0.2	0	0.4	0.4
	C	0.25	0.25	0	0.5
	D	0.25	0.25	0.5	0

Table 6a. The "obvious" conditional probability transition table for Aslin et al. (1998)

Crucially for the design of Aslin et al.'s experiments, p(D|C) needs to be 0.5, which would mean that the number of CD-pairs would, on average, be equal to the number of As[1].

However, Table 6a, like Table 3a, is wrong. When it is used to generate a sequence of 270 items, there are, on average, considerably too many As and Bs (52 compared to the 45 desired) than there should be and too few Cs and Ds (83 instead of the desired 90). This gives an overall A:CD-pair ratio of 1.3, almost a third higher than the desired ratio of unity.

And, in fact, no other randomization algorithm gets it right, either. With 45 As and Bs and 90 Cs and Ds, the standard randomization algorithm discussed at the beginning of this article not only produces an overall A:CD-pair ratio that is too low (0.87), but this ratio varies radically across the list, being 1.22 over the first fifth of the list and steadily decreasing to 0.45 in the final fifth. The distributed algorithm introduced earlier in this paper produces an A:CD-pair ratio that is constant across the list, but is also too low overall (about 0.88).

[1] There are as many As as Bs and as many Cs as Ds. Consequently, when we refer to the ratio of the number of As to the number of CD-pairs, designated by A:CD-pairs, we are also referring to all other ratios of the number of low-frequency items to the number of pairs of items made up of high-frequency items. (i.e., A:DC-pairs, B:CD-pairs, and B:DC-pairs).

So, is it, in fact, possible to find a conditional probability transition table with some number of As, Bs, Cs and Ds that does, indeed, satisfy the other constraints? The answer is yes, and is given in Tables 6b and 6c. How these correct conditional probabilities tables are created is, however, not obvious. In the next section we present a general method for the construction of transition tables for generating uniformly randomized lists with no repeated elements.

Second item

First item	A	B	C	D
A	0	0.135	0.432	0.432
B	0.135	0	0.432	0.432
C	0.231	0.231	0	0.538
D	0.231	0.231	0.538	0

Second item

First item	A	B	C	D
A	0	6	20	21
B	6	0	21	20
C	20	21	0	47
D	21	20	47	0

Table 6b and 6c. The correct conditional probability transition table and item-pair frequency table for a randomized sequence with no immediately repeated items. Notice that we had to modify the number of items of each type Now there are 47 As, 47 Bs, 88 Cs and 88 Ds.

Direct computation of transition tables.

Starting with the desired item frequencies, how can the item-pair frequency table be created that will generate correctly randomized sequences with no repeated elements for any number of items and item-types?

Assume we have N_1, N_2, N_3,…, items of each type, and a total of N_{total} items. We begin by calculating the table, *M*, of raw expected item-pair frequencies for each cell of the table, including for repeated elements. We remove the diagonal elements and put them in a separate vector, *d*. The values on the diagonal of *M* are set to 0.

$$n_{IJ} = \frac{N_I N_J}{N_{total} - 1} = \text{expected number of transitions in cell (I, J) of } M, \ \forall x \ \ n_{xx} = 0$$

$$d_k = \frac{N_K (N_K - 1)}{N_{total} - 1} = \text{expected number of immediate repeats of k}^{\text{th}} \text{ item type}$$

The final item-pair frequency values (i.e., n_{IJ}^{new}) used to build the item-pair frequency table are given by the following equation:

$$n_{IJ}^{new} = n_{IJ}\left(1 - R\right) + n_I R_J + n_J R_I \qquad (3)$$

where

$$R = \sum_i R_i \quad where \ \ R_K = \frac{d_K}{s_K}$$

$$s_K = n - 2n_K \quad where \ \ n = \sum_{\forall i \neq j} n_{ij} \ \ and \ \ n_K = \sum_{\forall j} n_{Kj}$$

Equation (3) was used to generate Tables 4b and 6c. (See below for its implementation in an Excel spreadsheet.)

Generating correctly randomized sequences

Once we have the correct item-pair frequency table, generating a correctly randomized sequence is straightforward. We can either use the program from Remillard (2008), which provides a very efficient means of generating sequences once the correct item-pair frequency table (or alternatively the correct transitional probabilities table) is provided as input or we can generate the sequence by the simple algorithm given here. Our technique relies on treating the *item-pairs* in the item-pair frequency table as the elements of the sequence. We begin by randomly drawing an item-pair, based on its overall probability of occurrence across the item-pair frequency table and begin the sequence to be generated with this item pair. We decrement the number for that particular item-pair in the item-pair frequency table. The second element of that item-pair tells what the first element of the next item-pair must be, i.e., tells us from what row of the item-pair frequency table to pick the second item-pair. We then pick the next item-pair based on probabilities of occurrence of the item-pairs in that row, add its second element to the list, decrement the number of that item-pair in the table, and go to the row of the table corresponding to the second item in the item-pair. We continue in this manner until the item-pair frequency table is empty.

Consider the frequency table 1c. To generate a list from this table, we randomly pick an item-pair from the table, based on the frequencies of each item-pair. Item-pair BC, having a probability of 9/30 = .3 gets picked. We begin our sequence, *S*, with this pair, so *S* = BC, and we decrement the number of BC item-pairs by one. This gives us Table 7a.

		Second item		
		A	B	C
First item	**A**	0	3	3
	B	3	0	8
	C	3	9	0

Table 7a. Item-pair frequencies from Table 1c after one BC item-pair has been removed.

We then go to row C. We have 3/12 = 0.25 chance of drawing a CA-pair and 9/12 = 0.75 chance of drawing a CB-pair. Say, we draw a CB-pair. We add the second item in this pair to *S*, so *S* = BCB and we decrement the number of CB item-pairs in the table, giving Table 7b.

		Second item		
		A	**B**	**C**
First item	**A**	0	3	3
	B	3	0	8
	C	3	8	0

Table 7b. Item-pair frequencies from Table 7a after one CB item-pair has been removed.

We return to row B. There is a 3/11 chance of drawing an BA-pair and 8/11 chance of drawing a BC-pair. We draw a BA-pair, which means we add an A to *S*, giving *S* = BCBA. We decrement by one the number of BA-pairs in the table and go to row A, where we have a 0.5 chance of drawing an AB-pair and a 0.5 chance of drawing an AC-pair, etc.

Simple Excel tools for generating correctly randomized sequences

We have developed a set of simple tools, implemented in Excel, that allow experimentalists to generate all of the item-frequency and conditional probability tables mentioned in this article. These tools can be downloaded at the following site: http://leadserv.u-bourgogne.fr/IMG/xls/TransitionMatrix.xls

The user is only required to enter the desired number of items of each type (up to 10 item-types) in the "Transition Matrix" worksheet. This worksheet generates the exact item-pair frequency table and transitional probability table. The latter table can be used as input to Remillard (2008)'s program to generate sequences. This first worksheet should run on any version of Excel on either a Mac or a PC. Two additional worksheets are also provided, which rely on macros and hence may be more restricted in use (e.g., the Excel macros unfortunately do not work for Excel 2008 for the Mac; an earlier version must be used on the Mac). Pressing Ctrl-r in the "Rounding" worksheet generates the appropriate integer-valued item-pair frequency table corresponding to the exact item-pair frequency table produced in the "Transition Matrix" worksheet. This table can also serve as input to Remillard's program to generate sequences. Pressing Ctrl-t in the "Sequence Generation" worksheet then generates randomized sequences, the number of which is set by the user, corresponding to the exact item-pair frequency table produced in the "Rounding" worksheet. Note that for large item-pair frequency tables, it is advisable to use the algorithm developed by Remillard (2008). Indeed, the Excel-based algorithm implemented in the "Sequence Generation" worksheet uses no advanced back-tracking techniques and, for large tables, can be slow.

Conclusion

Blais (2008) recently pointed out biases associated with randomization without replacement. These problems apply to all randomized lists but they are particularly acute for short lists of items and, as such, are not of serious concern for the points raised in this paper. Brysbaert (1991) and Castallan (1992) have also discussed various problems with randomizing lists but the problems they discuss are related, in general, to computer implementations of randomization algorithms.

In the present article we have shown that the use of standard randomization algorithms can lead to significant biases in the final randomized list. Particular care is called for when randomizing lists where:

- initial item frequencies are not equal
- repeated items, especially immediately repeated item, are not allowed.

These are very frequently encountered situations for experimentalists.

One might reasonably wonder why some of these list randomization problems have gone largely unnoticed in the past. We believe that the answer lies in the fact that for most experimentalists, list randomization is considered obvious and, as a result, they pay little attention to precisely how it is done. For this reason most articles include little – and, more often, no – information on how item sequences were randomized. This is a practice that needs to change. We hope that the simple tools provided in this paper and by our Excel files will contribute to this change and will help researchers produce correctly randomized lists of items for their studies.

Acknowledgments

This work was supported in part by European Commission grant FP6-NEST-029088 to the first author.

References

Aslin, R.N., Saffran, J.R. & Newport, E.L. (1998). Computation of conditional probability statistics by 8-month-old infants", *Psychological Science*, 9(4). 321-324. http://dx.doi.org/10.1111/1467-9280.00063

Blais, C. (2008). Random without replacement is not random: Caveat emptor. *Behavior Research Methods, 40* (4), 961-968 http://dx.doi.org/10.3758/BRM.40.4.961

Brysbaert, M. (1991). Algorithms for randomness in the behavioral sciences: A tutorial. *Behavior Research Methods, Instruments, and Computers*, 23(1). *45-60.*

Castellan, N. J. (1992). Shuffling arrays: Appearances may be deceiving. *Behavior Research Methods, Instruments, & Computers,* 24, 72-77.

Graf Estes, K., Evans, J.L., Alibali, M.W., & Saffran, J.R. (2007). Can Infants Map Meaning to Newly Segmented Words? Statistical Segmentation and Word Learning. *Cognitive Science, 18(3)*, 254-260. http://dx.doi.org/10.1111/j.1467-9280.2007.01885.x

Jacoby, L. L. (1983). Perceptual enhancement: Persistent effects of an experience. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *9*(1), 21–38. http://dx.doi.org/10.1037/0278-7393.9.1.21

Nissen, M.J., Bullemer, P.T. (1987) Attentional requirements for learning: Evidence from performance measures. *Cognitive Psychology, 19,* 1-32. http://dx.doi.org/10.1016/0010-0285%2887%2990002-8

Remillard, G. (2008). A program for generating randomized simple and context-sensitive sequences. *Behavior Research Methods, 40(2)*, 484-492.

Remillard, G., & Clark, J. M. (1999). Generating fixed-length sequences satisfying any given *n*th-order transition probability matrix. *Behavior Research Methods, Instruments, & Computers*, 31, 235-243.

Saffran, J.R., Newport, E.L., & Aslin, R.N. (1996). Word segmentation: The role of distributional cues. *Journal of Memory and Language, 35,* 606-621. http://dx.doi.org/10.1006/jmla.1996.0032

Seabrook, R., Brown, G.D.A., Solity, J.E. (2005). Distributed and massed practice: from laboratory to classroom. *Applied Cognitive Psychology* , 19(1), 107-122. http://dx.doi.org/10.1002/acp.1066

Appendix

The motivation for the Equation (3) describing explicitly how to build a correct item-pair frequency table is as follows. We assume that sequences wrap around. Starting with the desired item frequencies, we derive an initial item-pair frequency table of the expected numbers of each transition that includes the frequencies of transitions consisting of immediately repeated item-pairs (e.g., AA, BB, CC, and DD). We then remove and randomly redistribute all immediately repeated items elsewhere in the list in such a way that no new repeats are created. The insertion of a repeated item elsewhere in the list will, of course, split the item-pair into which the new item is inserted (i.e, there will be one less item-pair of this type), thereby creating two new item-pairs. For example, a B inserted into an AC pair will decrease the number of AC pairs by 1 and increase the number of AB and BC pairs by 1 each. By keeping track of the expected numbers of split transitions and additionally created transitions, we arrive at the appropriate item-pair frequency table.

We begin by filling in the "raw" item-pair frequency table, *M*. This table will include the frequencies of the repeated elements. The probability of an item-pair XY, where X and Y are different is: $p(n_{IJ}) = \frac{N_I}{N_{total}} \frac{N_J}{N_{total}-1}$. The probability of an item-pair, XX, is $p(n_{KK}) = \frac{N_K}{N_{total}} \frac{N_K - 1}{N_{total}-1}$. We multiply $p(n_{IJ})$ and $p(n_{KK})$ by N_{total} to arrive at the expected number of items in each cell of the initial item-pair frequency table. This gives: $n_{IJ} = \frac{N_I N_J}{N_{total}-1}$ when I ≠ J and $n_{JJ} = \frac{N_J (N_J - 1)}{N_{total}-1}$ when I = J. We put the n_{JJ} values (i.e., the numbers of each letter that will have to be "redistributed" elsewhere) into a vector, *d*, and then zero the diagonal of *M*.

We will continue the explanation with a matrix with 4 item-types. The generalization to *m* item-types is straightforward. We have:

	1	**2**	**3**	**4**
1	0	n_{12}	n_{13}	n_{14}
2	n_{12}	0	n_{23}	n_{24}
3	n_{13}	n_{23}	0	n_{34}
4	n_{14}	n_{24}	n_{34}	0

d **=**

d_1	d_2	d_3	d_4

We will focus on one transition, **23.** Only 1's and 4's can be inserted into a 23-transition without creating a **22** or **33** double. We wish to see how many 1's would be inserted into **23.**

	1	**2**	**3**	**4**
1	0	n_{12}	n_{13}	n_{14}
2	n_{21}	0	n_{23}	n_{24}
3	n_{31}	n_{32}	0	n_{34}
4	n_{41}	n_{42}	n_{43}	0

1's can only be inserted into transitions in the shaded area. The total number of items in this area is $n - 2n_1$ where n_1 is the number of 1's (Note: $n_K = \sum_{\forall i} n_{iK} = \sum_{\forall j} n_{Kj}$) and *n* is the sum of

all n_{ij} making up *M*. Thus $\frac{n_{23}}{n-2n_1}d_1$ **1**'s will be inserted into **23**. Similarly, $\frac{n_{23}}{n-2n_4}d_4$ **4**'s will be inserted into **23**. Altogether, the number of **23** transitions will decrease by:

$\sum_{\forall k\neq 2,3}\frac{n_{23}}{n-2n_k}d_k$ For simplicity, we let $s_k = n-2n_k$. So, the total number of **23** transitions decreases by $\sum_{\forall k\neq 2,3}\frac{d_k}{s_k}n_{23}$

We now count the number of additional number of **23** transitions created by inserting repeated items throughout *M*. A moment's reflection will show that the only item insertions that can add to the number of **23** transitions are: **2** inserted into transitions ending in **3**, and **3** inserted into transitions beginning with **2**. These are the transitions shown below.

	1	**2**	**3**	**4**
1	0	n_{12}	n_{13}	n_{14}
2	n_{21}	0	n_{23}	n_{24}
3	n_{31}	n_{32}	0	n_{34}
4	n_{41}	n_{42}	n_{43}	0

By the same logic as above, the number of **3**'s inserted into **21** will be $\frac{n_{21}}{n-2n_3}d_3 = n_{21}\frac{d_3}{s_3}$ and into **24** will be $\frac{n_{24}}{n-2n_3}d_3 = \frac{d_3}{s_3}n_{24}$. Regrouping these terms we have $(n_{21}+n_{24})\frac{d_3}{s_3}$. But $n_{21}+n_{24} = n_2 - n_{23}$. In other words, the insertion of **3**'s will create $(n_2-n_{23})\frac{d_3}{s_3}$ new **23**'s. A similar calculation shows that the insertion of **2**'s will create $(n_3-n_{23})\frac{d_2}{s_2}$ new **23**'s.

To calculate the new value of n_{23}, we add together all of these terms:

$$n_{23}^{new} = n_{23} - \sum_{\forall k\neq 2,3}\frac{d_k}{s_k}n_{23} + (n_2-n_{23})\frac{d_3}{s_3} + (n_3-n_{23})\frac{d_2}{s_2}$$

This simplifies to:

$$n_{23}^{new} = n_{23} - n_{23}\sum_{\forall k}\frac{d_k}{s_k} + n_2\frac{d_3}{s_3} + n_3\frac{d_2}{s_2}$$

If we let:

$$R_K = \frac{d_K}{s_K} \quad and \quad R = \sum_i R_i$$

Then the above equation simplifies to: $n_{23}^{new} = n_{23}(1-R) + n_2R_3 + n_3R_2$

Without loss of generality, we have:

$\forall_{I\neq J}\ \ n_{IJ}^{new} = n_{IJ}(1-R) + n_IR_J + n_JR_I$

$\forall_{I=J}\ \ n_{IJ}^{new} = 0$