

Obtaining representative nominal groups

MATTHEW R. KELLEY

Lake Forest College, Lake Forest, Illinois

AND

DANIEL B. WRIGHT

Florida International University, Miami, Florida

Many researchers studying the effectiveness of working in groups have compared group performance with the scores of individuals combined into nominal groups. Traditionally, methods for forming nominal groups have been shown to be poor, and more recent procedures (Wright, 2007) are difficult to use for complex designs and are inflexible. A new procedure is introduced and tested in which thousands of possible combinations of nominal groups are sampled. Sample characteristics, such as the mean, variance, and distribution, of all these sets are calculated, and the set that is most representative of all of these sets is returned. The user can choose among different ways of conceptualizing the meaning of *most representative*, but on the basis of simulations and the fact that most subsequent statistical procedures are based on the mean and variance, we argue that finding the set with the mean and variance most similar to the means of the representative statistics for all of the sets is the preferred approach. The algorithm is implemented in a stand-alone C++ executable program and as an R function. Both of these allow anyone to use the procedures freely.

The goal of many social psychology researchers has been to test whether groups perform better or worse than collections of individuals. In our area of research—memory in groups—the general finding (e.g., Basden, Basden, Bryner, & Thomas, 1997; Weldon & Bellinger, 1997; Weldon, Blair, & Huebsch, 2000; Wright & Klumpp, 2004; but see also Andersson & Rönnerberg, 1996; Meade, Nokes, & Morrow, 2009; Takahashi & Saito, 2004; Thorley & Dewhurst, 2007, 2009, who found situations in which individuals did not outperform groups) is that a set of people tend to perform worse if they work together (collaborative group) than if each individual works on his or her own and then individual outputs are combined (nominal group). Historically, to calculate nominal group performance, researchers compared the performance of groups with that of individuals from a control group arbitrarily assigned to nominal groups. Wright (2007) showed that there was much variation in the resulting statistics among the different possible sets of nominal groups and, therefore, that how the particular set was chosen could greatly affect the results. The arbitrary nature of the traditional procedure adds needless sampling error to these studies and should be avoided.

Conceptually, the best solution is to try all possible combinations of the individuals in the control group and to base the nominal group statistics on some central tendency of these values. However, the number of combinations is prohibitively high for the typical sample sizes used in psychology research. For example, if you had 40 individuals, so 20 nominal pairs, there are approximately 2×10^{34} possible

sets, which is far too many for modern computers. Instead of calculating the values of all possible sets, Wright (2007) presented a function written in the software R (and S-Plus) that calculated statistics (mean, variance, and adjusted variance) on the basis of all pairs in the control group. Unfortunately, there are several limitations to this function. First, it only produced the values for the mean and variances and required users to calculate by hand subsequent inferential statistics, which can be very cumbersome for complex designs. Second, he found that the variance estimate from the statistics for all pairs was biased, so he ran a limited series of simulations and produced a rough adjustment method. Because these simulations were based on very particular circumstances (with small *ns*), a more general procedure would be valued. Finally, researchers unfamiliar with R and its syntax might have difficulty adapting and using the function. Although R is an excellent package, it has a different interface from many other packages and therefore may be daunting for non-R users. The approach taken here allows a more general method and one that researchers, we believe, will find more useful.

Our solution is to first calculate statistics from a large number (10,000) of randomly chosen sets of nominal groups. We then search for the single set that is most representative of this large set. This set is output to the users so that they can use the values from these pairs in any manner that they wish. We wrote both a stand-alone executable program (written in C++) and an R function (which is available as part of the R package `mrt` on the Comprehensive R Archive Network).

M. R. Kelley, kelley@lakeforest.edu



The Nominal Group Finder: Executable Program

The Nominal Group Finder can be implemented on any PC, provided that two files are downloaded and installed. First, since the program was written in C++, a user must install the free Microsoft Visual C++ 2008 Redistributable Package (x86) from the Microsoft Web site to run the program ([www.microsoft.com/downloads/details.aspx?FamilyID = 9b2da534-3e03-4391-8a4d-074b9f2bc1bf](http://www.microsoft.com/downloads/details.aspx?FamilyID=9b2da534-3e03-4391-8a4d-074b9f2bc1bf); accessed July 8, 2009). Next, the user needs to download a zipped folder from either of the authors' Web sites (<http://campus.lakeforest.edu/kelley>; www.fiu.edu/~dwright/nomg), which contains the executable program, sample files, and an instruction guide. When the program is opened, it asks for three parameters: the name of the data file (e.g., data-file.txt), the number of control group participants, and the number of recall test items (see Figure 1 for a screenshot of the program screen and subsequent data files).

As input, the program requires a tab-delimited text file in which each row corresponds to a different participant

and each column corresponds to a different item on the recall test. The number of participants in the file should be a multiple of two, to ensure that the complete sample can be divided into pairs, and can be up to 100; the number of items also must not exceed 100, but this value can be even or odd. In the file, correct responses are coded as 1s, and incorrect responses are coded as 0s; each value should be separated by a tab. If the original data are in a spreadsheet (e.g., Microsoft Excel), the user can copy the rows/columns, paste them into a Notepad file, and save the file, or use the "Save As . . ." option available in many spreadsheets and statistics packages.

After the user has entered the first three parameters, the program then forms 10,000 sets of random nominal group pairings (e.g., Set 1 pairings, Participants 1 and 28, 2 and 13, . . .; Set 2 pairings, Participants 1 and 12, 2 and 4, . . .).¹ Specifically, each set is formed by randomizing the order of the participants, using a random shuffle algorithm, and the pairs are assigned in this order. For each set, the program calculates the dot product of the two participants' scores. Prior to dot product calculation,

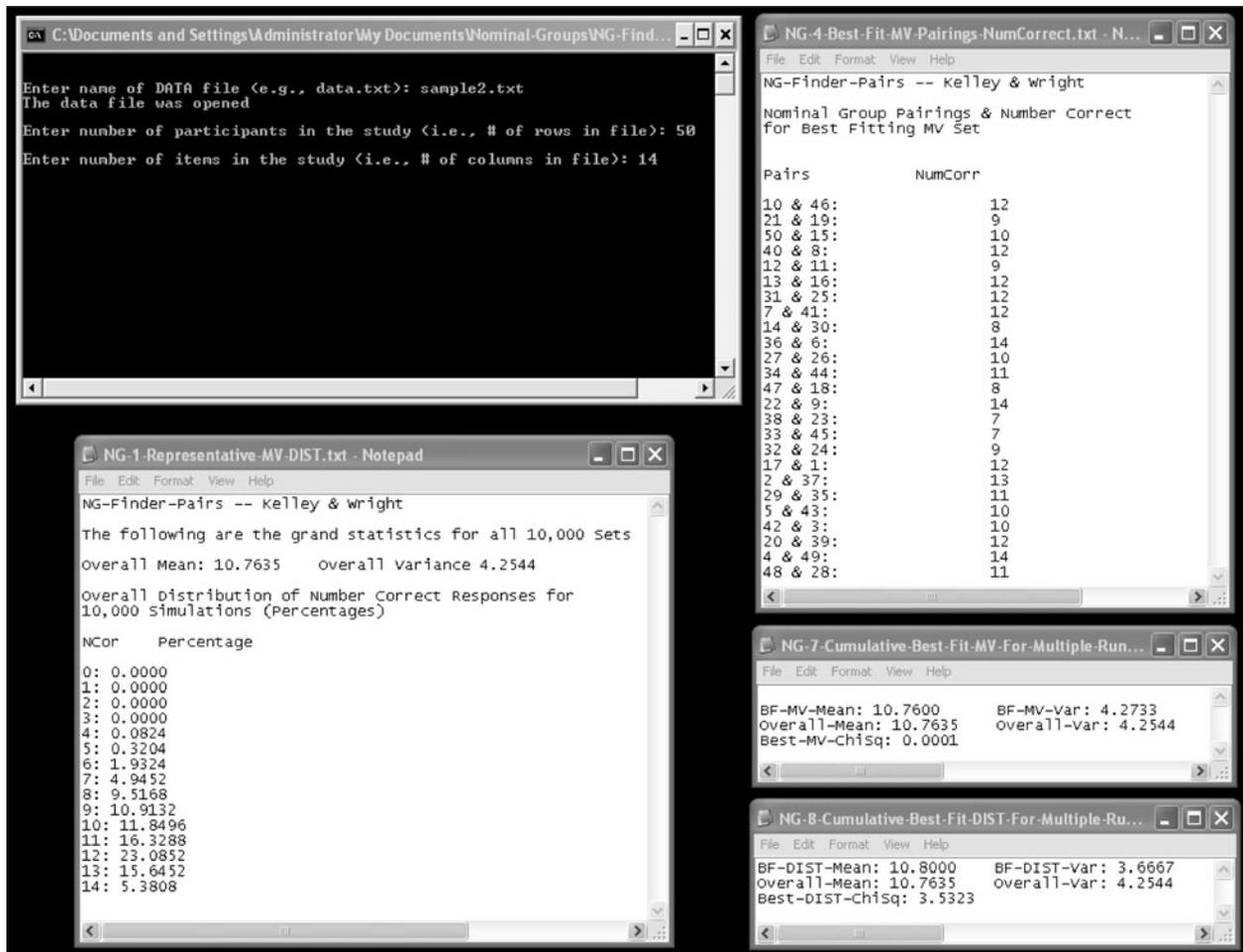


Figure 1. Screenshot of the Nominal Group Finder Program and four output files, which displays the representative statistics (NG-1), the best-fitting MV-Fit set pairings and number correct (NG-4), and the summary statistics for the MV-Fit (NG-7) and DIST-Fit (NG-8) approaches.

Table 1
Stability of the Representative Means and Variances Across Simulations

Simulation Condition	$A_M - I_M$		$A_{Var} - I_{Var}$	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
20 participants, 5 items	0.0012	0.0008	0.0032	0.0026
20 participants, 20 items	0.0036	0.0019	0.0275	0.0161
20 participants, 100 items	0.0083	0.0068	0.0856	0.0664
100 participants, 5 items	0.0008	0.0006	0.0012	0.0009
100 participants, 20 items	0.0016	0.0014	0.0114	0.0094
100 participants, 100 items	0.0031	0.0030	0.0526	0.0393

Note— A_M , the overall means from all 10 simulations; I_M , the mean from each individual simulation; A_{Var} , the overall variance from all 10 simulations; I_{Var} , the variance from each individual simulation.

the program automatically reverses the coding of correct (0s) and incorrect (1s) responses, such that the dotproduct is the number of items that *neither* participant correctly answered. After the dot product for each pair in the set is computed, the values are converted to number correct (i.e., the number of items that *at least 1* participant correctly answered), and the summary statistics—mean, variance, and the distribution of number correct—are stored in an array. Once the statistics have been calculated for all 10,000 sets, the representative statistics are calculated by determining the overall mean, variance, and distribution of number correct.

The Nominal Group Finder program then searches for the set whose statistics most closely resemble the representative statistics, using two approaches: mean and variance (MV-Fit) and distribution (DIST-Fit). MV-Fit is the preferred approach and finds the set whose mean and variance most closely resemble the representative mean and variance. The best-fitting set is defined as the one with the smallest value for

$$\frac{(\text{mean} - \text{overall mean})^2}{\text{overall mean}} + \frac{(\text{variance} - \text{overall variance})^2}{\text{overall variance}}$$

Smaller values mean smaller differences between the observed and expected values; the program finds the smallest of the 10,000 sets. This means that the chosen set will have a mean and variance very near those found from the bootstrap samples.

The DIST-Fit approach uses the Pearson χ^2 goodness-of-fit procedure to compare the distribution of each set with the overall distribution of all of the sets. The program finds the set with the most similar distribution, as is determined by the minimum χ^2 value. The Finder program saves to files the nominal group pairings for the best MV-Fit and DIST-Fit, the number correct for each of these pairs, and the statistics for both methods for the best-fitting set (see Figure 1). The output files are located in the same folder as the Finder program. They do not open automatically, and subsequent runs of the program will append the data to the existing files.

Ten simulations were run for each of six combinations of numbers of participants and numbers of items; specifically, two numbers of participants (20, 100) were combined factorially with three numbers of items (5, 20, and 100). Table 1 shows that the representative mean and variance (i.e., average of 10,000 random sets) are stable across simulations, which provides support for the assertion that these are indeed representative values. Table 2 displays the best-fitting statistics for each simulation condition and fit approach: the mean Pearson χ^2 test statistic, the average difference between the observed and expected means, and average difference between the observed and expected

Table 2
Simulation Statistics for Each Simulation Condition

Simulation Condition	Pearson χ^2		$O_M - E_M$		$O_{Var} - E_{Var}$		$O_{Skew} - E_{Skew}$	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
20 participants, 5 items								
MV	0.0024	0.0004	0.0159	0.0018	0.0411	0.0036	0.4650	0.0000
DIST	5.1554	0.0546	0.1158	0.0019	0.3256	0.0038	0.3470	0.0000
20 participants, 20 items								
MV	0.0002	0.0001	0.0411	0.0174	0.0221	0.0126	0.1539	0.1450
DIST	14.3452	0.1217	0.1317	0.0043	1.9365	0.0331	0.1754	0.1328
20 participants, 100 items								
MV	0.0001	0.0000	0.0428	0.0466	0.0420	0.0243	0.0943	0.0401
DIST	97.5746	2.3322	0.8456	0.5865	29.7855	5.3036	0.0236	0.0037
100 participants, 5 items								
MV	0.0001	0.0000	0.0104	0.0010	0.0093	0.0018	0.2366	0.0962
DIST	0.3560	0.0085	0.0091	0.0006	0.0134	0.0023	0.2101	0.1330
100 participants, 20 items								
MV	0.0000	0.0000	0.0074	0.0036	0.0077	0.0051	0.0840	0.0992
DIST	4.0761	0.3033	0.1385	0.0530	0.7862	0.4185	0.1299	0.0791
100 participants, 100 items								
MV	0.0000	0.0000	0.0124	0.0096	0.0150	0.0093	0.1114	0.0393
DIST	21.7677	1.2883	0.3175	0.2459	8.0298	3.2987	0.0729	0.0464

Note—MV, mean variance fit; DIST, distribution fit; O_M , observed mean; E_M , expected mean; O_{Var} , observed variance; E_{Var} , expected variance; O_{Skew} , observed skewness; E_{Skew} , expected skewness.

variances. Overall, the MV-Fit approach provided sets that closely fit the representative statistics to within a few hundredths of an item. Although the DIST-Fit approach yielded sets with reasonable fits to the representative mean (within a few tenths of an item), the variances of these sets often differed substantially from the representative variance.

Of course, one could argue that these are not fair comparisons; after all, the MV-Fit was designed to fit the representative mean and variance, and one would expect a closer fit when the mean and variance are the criteria. On the other hand, the DIST-Fit approach was designed to characterize the distribution and might provide a better fit to a statistic that reflects other distribution properties, such as skewness. To assess this possibility, overall skewness was calculated for the 10,000 sets in each simulation, as well as for the best-fitting set with each approach (see the final column of Table 2). Although the DIST-Fit approach often provided a marginally closer fit, both approaches tended to characterize the overall skewness statistic reasonably well. Generally, then, we advocate the use of the MV-Fit approach, because it yields a nominal group set that closely approximates the representative mean and variance and provides a reasonable estimate of skewness. Furthermore, because the most common subsequent statistical tests (e.g., ANOVAs) rely on means and variances, it is most important to get these values right. The Finder program, however, always reports the results for both approaches.

One potential limitation of this version of the Nominal Group Finder program is that it is limited to a group size of 2. To remedy this, we have created two additional exe-

cutable programs to accommodate group sizes of 3 and 4, as well. The MV-Fit approaches in these versions of the Finder program provide best-fitting sets with similar characteristics to those reported in Table 2. These are available in the same folder as the program for nominal pairs.

The R Function `ngfinder`

R is a freeware statistical package based on the commercial package S/S-Plus. R and much information about R can be downloaded from www.r-project.org (see Chambers, 2009, for a brief historical overview). According to Ritz and Streibig (2008), it has become the lingua franca of graphical display and statistical analysis. An introduction to R for psychologists is available at www.sagepub.com/upm-data/26922_Wright_Chapter_One.pdf (Wright & London, 2009, Chapter 1).

The basic algorithm described above was written as an R function called `ngfinder`. Because of the large number of built-in statistical functions, the object-oriented aspects of R, and that the fact people using this function will probably be doing their further analyses within R, there are a couple of differences.

The function is part of the R package `mrt` (short for *modern regression techniques*; Wright & London, 2009). From within R, the user can install and activate the R package `mrt` either using the *Packages* tab in the tool bar or with the following commands:

```
install.packages("mrt")
library(mrt)
```

The screenshot shows the R GUI with the following code and output in the console:

```
> set.seed(32)
> dmat <- rbinom(120,1,.4)
> dim(dmat) <- c(12,10)
> ngfinder(dmat,number=3,replic=5000)
$forgroups
 [1] 7 6 11 10 3 5 1 4 12 9 8 2

$values
 [1] 8 9 8 7

> ngfinder(dmat,number=3,moments=3)
$forgroups
 [1] 8 2 3 1 9 7 11 6 5 12 10 4

$values
 [1] 7 7 10 7
```

Callouts in the image provide the following explanations:

- Creates data (0s and 1s) with 12 rows and 10 columns (pointing to the `dmat` creation lines).
- Runs `ngfinder` for groups of 3 with 5,000 samples. Optimality is defined by how similar the set's distribution is to the overall distribution. The first optimal group is {7,6,11}. They were correct on 8 items. (pointing to the first `ngfinder` call and its output).
- Optimality is defined by how similar the set is to the mean, variance, and skewness of all of the sets (i.e., the first three moments). (pointing to the second `ngfinder` call and its output).

Figure 2. Using the `ngfinder` in R. The screen shows data being created (the first three lines) and then the function being used twice.

The user will be prompted to choose a mirror site from which to download the function. The user should choose one near his or her location. This also downloads an online help file that can be accessed from within R with the commands `help(ngfinder)` or `?ngfinder`, and examples that can be accessed with `example(ngfinder)`.

In R, the user writes at a command prompt the function name with parentheses, within which any arguments are enclosed. Some examples are shown in Figure 2 with comments in textboxes. For example, if the user has a data matrix (called `data1`), with 1 meaning that the item is recalled, then the command `ngfinder(data1)` will create 1,000 sets of possible nominal pairs and will return the values and the numbers for the set that is the most representative according to the difference in distance from the overall distribution. Within `ngfinder`, the R function `sample` is used without replacement to randomize the order of the participants, and then sets of the appropriate size (see the `number` option below) are taken in this order. To calculate the number of correct items, the R function `any` is used (computationally slower than finding the dot product, but a more general method). The function `ngfinder` only requires that the data matrix be entered, but other arguments can be entered that alter how the function works. These, with their default values, are

<code>number = 2</code>	If you have groups of 3, 4, etc., include this option. If the number of participants is not divisible by this number, each set sampled will randomly exclude the remainder.
<code>right = 1</code>	If the data matrix has a different value for correct responses, use this option.
<code>replic = 1000</code>	This is how many sample sets should be drawn. Running 10,000, as in the C++ function, takes a few minutes, so the default is 1,000, which is ample for most purposes. This R function is slower than the C++ function, because of the way the dot product is calculated.
<code>moments = NULL</code>	Any positive integer will tell the computer to match on those moments, so <code>moments = 2</code> is essentially the MV-Fit method from above, <code>moments = 3</code> would include skewness, and so forth. This function standardizes the values for all moments and then finds the set with the overall lowest sum of values for these. This was done since squaring the residuals can pro-

duce particularly large values for higher moments.

`distmeth = "ks"` The default method for comparing distributions uses the statistic from the Kolmogorov-Smirnov test. The option `distmeth = "pearson"` uses Pearson's method as described for the DIST-fit method for the C++ program.

The object-oriented aspect means that the pairs and values can be used directly in other functions. For example, if the number of items recalled by actual groups is a variable called `actual`, the command `t.test(actual, ngfinder(data1)$values)` runs a *t* test between these variables without having to calculate the nominal groups separately. The function `ngfinder(data1)$forgroups` produces the participant numbers for the groupings.

Summary

The Nominal Group Finder program and R function `ngfinder` provide two simple and effective solutions to the problem of obtaining representative nominal groups in studies of collaborative performance. They find representative nominal group statistics from a large number of random sets of nominal groups and return the particular set that most closely resembles the representative statistics. The program and function are free, run on any PC (the R function works also on Unix and Mac machines), accommodate the sizes of data sets commonly seen in the collaborative literature (i.e., up to 100 participants, 1–100 items), and require no more than a few seconds before writing the desired output to a file. Moreover, the program and function provide output that can easily be used in any statistical software program, thus eliminating the need to calculate statistics by hand, as was suggested in Wright (2007).

AUTHOR NOTE

Correspondence concerning this article should be addressed to M. R. Kelley, Department of Psychology, Lake Forest College, Lake Forest, IL 60045 (e-mail: kelly@lakeforest.edu).

REFERENCES

- ANDERSSON, J., & RÖNNBERG, J. (1996). Collaboration and memory: Effects of dyadic retrieval on different memory tasks. *Applied Cognitive Psychology*, *10*, 171-181. doi:10.1002/(SICI)1099-0720(199604)10:2<171::AID-ACP385>3.0.CO;2-D
- BASDEN, B. H., BASDEN, D. R., BRYNER, S., & THOMAS, R. L., III (1997). A comparison of group and individual remembering: Does collaboration disrupt retrieval strategies? *Journal of Experimental Psychology: Learning, Memory, & Cognition*, *23*, 1176-1189. doi:10.1037/0278-7393.23.5.1176
- CHAMBERS, J. M. (2009). Facets of R. *R Journal*, *1*, 5-8.
- MEADE, M. L., NOKES, T. J., & MORROW, D. G. (2009). Expertise promotes facilitation on a collaborative memory task. *Memory*, *17*, 39-48. doi:10.1080/09658210802524240
- RITZ, C., & STREIBIG, J. C. (2008). *Nonlinear regression with R*. New York: Springer.

- TAKAHASHI, M., & SAITO, S. (2004). Does test delay eliminate collaborative inhibition? *Memory*, **12**, 722-731. doi:10.1080/09658210344000521
- THORLEY, C., & DEWHURST, S. A. (2007). Collaborative false recall in the DRM procedure: Effects of group size and group pressure. *European Journal of Cognitive Psychology*, **19**, 867-881. doi:10.1080/09541440600872068
- THORLEY, C., & DEWHURST, S. A. (2009). False and veridical collaborative recognition. *Memory*, **17**, 17-25. doi:10.1080/09658210802484817
- WELDON, M. S., & BELLINGER, K. D. (1997). Collective memory: Collaborative and individual processes in remembering. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **23**, 1160-1175. doi:10.1037/0278-7393.23.5.1160
- WELDON, M. S., BLAIR, C., & HUEBSCH, P. D. (2000). Group remembering: Does social loafing underlie collaborative inhibition? *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **26**, 1568-1577. doi:10.1037/0278-7393.26.6.1568
- WRIGHT, D. B. (2007). Calculating nominal group statistics in collaboration studies. *Behavior Research Methods*, **39**, 460-470.
- WRIGHT, D. B., & KLUMPP, A. (2004). Collaborative inhibition is due to the product, not the process, of recalling in groups. *Psychonomic Bulletin & Review*, **11**, 1080-1083.
- WRIGHT, D. B., & LONDON, K. (2009). *Modern regression techniques using R: A practical guide for students and researchers*. London: Sage.

NOTE

1. Simulations compared the stability of the mean and variance estimates following 1,000 and 10,000 sets of random nominal group pairings. Although 1,000 sets provided stable estimates for small to moderate numbers of participants and items, 10,000 sets yielded the most stable estimates for the full range of allowable parameters.

(Manuscript received July 10, 2009;
revision accepted for publication August 24, 2009.)