An R package to compute commonality coefficients in the multiple regression case: An introduction to the package and a practical example

Kim Nimon
Southern Methodist University, Dallas, Texas

Mitzi Lewis
University of North Texas, Denton, Texas

Richard Kane
University of North Florida, Jacksonville, Florida

AND

R. Michael Haynes
University of North Texas, Denton, Texas

Multiple regression is a widely used technique for data analysis in social and behavioral research. The complexity of interpreting such results increases when correlated predictor variables are involved. Commonality analysis provides a method of determining the variance accounted for by respective predictor variables and is especially useful in the presence of correlated predictors. However, computing commonality coefficients is laborious. To make commonality analysis accessible to more researchers, a program was developed to automate the calculation of unique and common elements in commonality analysis, using the statistical package R. The program is described, and a heuristic example using data from the Holzinger and Swineford (1939) study, readily available in the MBESS R package, is presented.

Multiple regression is a widely used technique for data analysis in social and behavioral research (Fox, 1991; Huberty, 1989). It is a method for determining the amount of variance of two or more predictor variables on a criterion variable. These predictor variables are often correlated, increasing the complexity of interpreting results (Pedhazur, 1997; Zientek & Thompson, 2006).

Stepwise regression is often used in educational and psychological research to evaluate the order of importance of variables and to select useful subsets of variables (Huberty, 1989; Thompson, 1995). Pedhazur (1997) suggested that stepwise regression methods provide researchers with a methodology with which to determine a predictor's individual meaningfulness as it is introduced into the regression model. However, stepwise regression can lead to serious Type I errors (Thompson, 1995), and the selection/entry order into the model can "drastically" misrepresent a variable's usefulness (Kerlinger, 1986, p. 543).

Commonality analysis provides an effective alternative for determining the variance accounted for by respec-tive predictor variables (Onwuegbuzie & Daniel, 2003; Rowell, 1996). Also called element analysis, commonality analysis was developed in the 1960s as a method of partitioning variance (R^2) into unique and nonunique parts (Mayeske et al., 1969; Mood, 1969, 1971; Newton & Spurrell, 1967). This has important implications, because theory advancement and research findings' usefulness

depend not only on establishing that a relationship exists among predictors and the criterion, but also upon determining the extent to which those independent variables, singly and in all combinations, share variance with the dependent variable. *Only then can we fully know the relative importance of independent variables with regard to the dependent variable in question* [italics added]. (Seibold & McPhee, 1979, p. 355)

However, commonality analysis can be a laborious process. The present article provides an overview of commonality analysis and introduces an R program for easily calculating commonality coefficients.[1]

K. Nimon, kim.nimon@gmail.com

Table 1
Unique and Commonality Formulas
for Three Predictor Variables

$U_{(i)}$	$=$	$R^2_{y.ijk} - R^2_{y.jk}$
$U_{(j)}$	$=$	$R^2_{y.ijk} - R^2_{y.ik}$
$U_{(k)}$	$=$	$R^2_{y.ijk} - R^2_{y.ij}$
$C_{(ij)}$	$=$	$-R^2_{y.ik} + R^2_{y.jk} - R^2_{y.k} - R^2_{y.ijk}$
$C_{(ik)}$	$=$	$-R^2_{y.ij} + R^2_{y.jk} - R^2_{y.j} - R^2_{y.ijk}$
$C_{(jk)}$	$=$	$-R^2_{y.ij} + R^2_{y.ik} - R^2_{y.i} - R^2_{y.ijk}$
$C_{(ijk)}$	$=$	$R^2_{y.i} + R^2_{y.j} + R^2_{y.k} - R^2_{y.ij} - R^2_{y.ik} - R^2_{y.jk} + R^2_{y.ijk}$

Calculation of Commonality Coefficients

The unique contribution (U) of a predictor variable is the proportion of variance of the dependent variable that is attributed to it when it is entered last in a regression analysis. In other words, the unique contribution is the squared semi-partial correlation between the predictor variable of interest and the dependent variable, after partialling out all the other predictor variables (Pedhazur, 1997). For example, in the regression case with two predictor variables, i and j,

$$U(i) = R^2_{y.ij} - R^2_{y.j}, \qquad (1)$$

$$U(j) = R^2_{y.ij} - R^2_{y.i}, \qquad (2)$$

and

$$C(ij) = R^2_{y.ij} - U(i) - U(j) \qquad (3)$$

allow for the computation of the unique contribution of variable $i [U(i)]$, the unique contribution of variable $j [U(j)]$, and the commonality of variables i and $j [C(ij)]$. Substituting the right side of the first two equations for $U(i)$ and $U(j)$ in the right side of the third equation results in

$$C(ij) = R^2_{y.ij} - (R^2_{y.ij} - R^2_{y.j}) - (R^2_{y.ij} - R^2_{y.i})$$
$$= R^2_{y.j} + R^2_{y.i} - R^2_{y.ij}. \qquad (4)$$

The number of equations required for a commonality analysis is $2^k - 1$ components, where k is the number of predictor variables in the regression analysis. Therefore, the complexity of commonality analysis increases exponentially with the number of variables entered into the model. For example, in conducting a commonality analysis with four independent variables, 15 unique and combinations of variance accounted for are generated. With five

or six independent variables, the number increases to 31 and 63, respectively.

Some researchers have suggested factor or cluster analysis as a method of collapsing myriad variables into fewer, more manageable groups (Mood, 1969; Seibold & McPhee, 1979; Wisler, 1972, as cited by Rowell, 1991). However, Rowell (1991) also notes that this action defeats the purpose of commonality analysis, in that the ability to identify the most useful *individual* variable is lost.

Tables 1 and 2 list the equations required for three and four predictor variable commonality analyses.

These computations for commonality analysis are not included in any of the commonly available statistical software packages (Onwuegbuzie & Daniel, 2003).[2] As was illustrated above, the computation of unique and non-unique variance is cumbersome, requiring that these series of formulas be written and applied to output from multiple computer-assisted statistical analyses through either (1) manual calculations or (2) assistance of a spreadsheet program (still requiring that the formulas and statistical analyses output be manually entered into the spreadsheet program). To simplify this process and make commonality analysis accessible to more researchers, a program was developed to automate the calculation of unique and common elements in commonality analysis.

Program Description

In order to facilitate data analysis and accessibility, the statistical package R was used. R is a free statistical programming language and environment for the Unix, Windows, and Mac families of operating systems (Hornik, 2007). R is gaining popularity in the behavioral, educational, and social sciences, as evidenced in part by the recent introduction of the *Methods for Behavioral, Educational, and Social Sciences* (MBESS) R package (Kelley, 2006). Instructions for downloading and installing R, as well as other R documentation and resources, are available on the R-Project Internet homepage (R Development Core Team, 2007).

The commonality coefficient program is an R package based on Mood's (1969) procedure for computing commonality analysis formulas for any number (k) of

Table 2
Unique and Commonality Formulas for Four Predictor Variables

$U_{(i)}$	$=$	$R^2_{y.ijkl} - R^2_{y.jkl}$
$U_{(j)}$	$=$	$R^2_{y.ijkl} - R^2_{y.ikl}$
$U_{(k)}$	$=$	$R^2_{y.ijkl} - R^2_{y.ijl}$
$U_{(l)}$	$=$	$R^2_{y.ijkl} - R^2_{y.ijk}$
$C_{(ij)}$	$=$	$-R^2_{y.kl} + R^2_{y.ik} + R^2_{y.jkl} - R^2_{y.ijkl}$
$C_{(ik)}$	$=$	$-R^2_{y.jl} + R^2_{y.ijl} + R^2_{y.jkl} - R^2_{y.ijkl}$
$C_{(il)}$	$=$	$-R^2_{y.jk} + R^2_{y.ijk} + R^2_{y.jkl} - R^2_{y.ijkl}$
$C_{(jk)}$	$=$	$-R^2_{y.il} + R^2_{y.ijl} + R^2_{y.ikl} - R^2_{y.ijkl}$
$C_{(jl)}$	$=$	$-R^2_{y.ik} + R^2_{y.ijk} + R^2_{y.ikl} - R^2_{y.ijkl}$
$C_{(kl)}$	$=$	$-R^2_{y.ij} + R^2_{y.ijk} + R^2_{y.ijl} - R^2_{y.ijkl}$
$C_{(ijk)}$	$=$	$-R^2_{y.l} + R^2_{y.il} + R^2_{y.jl} + R^2_{y.kl} - R^2_{y.ijl} - R^2_{y.ikl} - R^2_{y.jkl} + R^2_{y.ijkl}$
$C_{(ijl)}$	$=$	$-R^2_{y.k} + R^2_{y.ik} + R^2_{y.jk} + R^2_{y.kl} - R^2_{y.ijk} - R^2_{y.ikl} - R^2_{y.jkl} + R^2_{y.ijkl}$
$C_{(ikl)}$	$=$	$-R^2_{y.j} + R^2_{y.ij} + R^2_{y.jk} + R^2_{y.jl} - R^2_{y.ijk} - R^2_{y.ijl} - R^2_{y.jkl} + R^2_{y.ijkl}$
$C_{(jkl)}$	$=$	$-R^2_{y.i} + R^2_{y.ij} + R^2_{y.ik} + R^2_{y.il} - R^2_{y.ijk} - R^2_{y.ijl} - R^2_{y.ikl} + R^2_{y.ijkl}$
$C_{(ijkl)}$	$=$	$R^2_{y.i} + R^2_{y.j} + R^2_{y.k} + R^2_{y.l} - R^2_{y.ij} - R^2_{y.ik} - R^2_{y.il} - R^2_{y.jk} - R^2_{y.jl} - R^2_{y.kl} + R^2_{y.ijk} + R^2_{y.ijl} + R^2_{y.ikl} + R^2_{y.jkl} + R^2_{y.ijkl}$

predictor variables. In Mood's (1969) procedure, $(1 - x)$ was used to represent variables in the common variance subset, and (x) was used to represent variables not in the common variance subset. By negating the product of the variables in the subset and the variables not in the subset, deleting the -1 resulting from the expansion of the product, and replacing x with R^2, Mood (1969) noted that the formula for computing any commonality coefficient can be derived. For example, Formula 5 represents the variance common to the subset of Variables 1 and 3 out of four independent variables:

$$-(1 - x_1)(1 - x_3)x_2x_4 =$$
$$-R_{1234}^2 + R_{124}^2 + R_{234}^2 - R_{24}^2. \quad (5)$$

The commonality coefficient program begins by creating a bit matrix containing a column for each commonality coefficient and a row for each independent variable. The number of independent variables determines the number of commonality coefficients $(2^k - 1)$. Each column contains the binary representation of the coefficient ID (1 to $2^k - 1$). The commonality coefficient ID also represents the associated common variance subset independent variable IDs. For example, the variance common to the subset of Variables 1 and 3 out of four independent variables is associated with commonality coefficient 5.

Each column in the bit matrix is analyzed to conduct all possible regressions $(2^k - 1)$ for the number of independent variables. A one indicates that the independent variable is to be included in the regression formula. A zero indicates that the independent variable is to be excluded from the regression formula. Thus, if a column contains a one in Rows 1 and 2, along with zeros in all the other rows, the dependent variable would be regressed by Independent Variables 1 and 2, yielding $R_{y.x_1x_2}^2$. The resulting R^2 values are stored in a commonality matrix, indexable by the associated commonality coefficient ID.

To determine the R^2 values to be used in computing a commonality coefficient C_n, the algorithm accesses the bit matrix at Column n (i.e., for C_1, access Column 1). Each entry in the column represents the contribution for an independent variable, where Row m represents the independent variable ID. A one indicates that the independent variable is in the common variance subset and is processed as $(1 - x_m)$. A zero indicates that the independent variable is not in the common variance subset and is processed as (x_m).

For each C_n, the index of R^2 values is seeded with either $(0, -m)$ or (m) on the basis of whether the first independent variable is or is not in the common variance subset. The list is then manipulated on the basis of the status of the remaining independent variables. Independent variables not in the common variance subset cause the list to be processed by a sequential *arithmetic or* of the absolute values of the entries on the list with the entry (m) and an *exclusive or* of their signs. Independent variables in the common variance subset cause the list to be concatenated with the results of sequential *arithmetic or* of the absolute values of the entries on the list with the entry $(-m)$ and an *exclusive or* of their signs.

The resultant lists are then processed to calculate the commonality coefficients. For each item on the list, the R^2 value is retrieved from the commonality matrix. All of the retrieved values are summed to produce the commonality coefficient. Each R^2 value retrieved is added to the sum if the list entry is positive, or it is subtracted from the sum if the list entry item is negative.

The function outputs a list of two tables. The first table contains the list of commonality coefficients, as well as the percentage of variance associated with each effect. The second table provides a total of the unique and common effects for each independent variable.

Conducting a Commonality Analysis: A Practical Example

For illustrative purposes, data from the Holzinger and Swineford (1939) study are used to contextualize the discussion. The Holzinger and Swineford study consisted of 26 tests administered to 301 students from Paster School and Grant-White School. These tests measured the students' spatial, verbal, mental speed, memory, and mathematical ability. These data were selected because of their logical utility for demonstrating the techniques discussed in this article and because the reader would also have the opportunity to generate the analysis.

Data from four tests in the Holzinger and Swineford (1939) study were utilized for the present analysis; these four tests and the rest of the complete data set are readily available in the MBESS R package. The simplest way to get MBESS is to use the "install package(s)" facility. Once the package is installed, the commands listed in Table 3 will load the data set into the data editor and will attach the data set into the R search path so that variables can be directly accessed by simply giving their names.

Replicating Oxford and Daniel's (2001) initial regression analysis, data from a paragraph comprehension test (`paragrap`) was regressed on four verbal tests: (1) general information (`general`), (2) sentence comprehension (`sentence`), (3) word classification (`wordc`), and (4) word meaning (`wordm`) to determine the extent to which verbal ability predicts paragraph comprehension (Table 4 lists commands to accomplish this regression). Performance on the four selected verbal tests explains 61.14% of the performance on the paragraph comprehension test.

Next, the commonality coefficient package was utilized to perform a commonality analysis in order to answer the following questions. (1) What percentage of the explained variance in paragraph comprehension is associated with unique effects (i.e., general information, sentence comprehension, word classification, and word meaning)? (2) What percentage of explained variance in paragraph comprehension is associated with first-order

Table 3
Commands to Load Data Set

Command 1: `library(MBESS)`
Command 2: `data(HS.data)`
Command 3: `attach(HS.data)`

Table 4
Commands to Run Regression Analysis

Command 1:
```
regr <- lm(paragrap ~ general + sentence + wordc + wordm)
```
Command 2:
```
summary(regr)
```

common effects (i.e., general information and sentence comprehension, general information and word classification, sentence comprehension and word classification, general information and word meaning, sentence comprehension and word meaning, and word classification and word meaning)? (3) What percentage of explained variance in paragraph comprehension is associated with second-order common effects (i.e., general information, sentence comprehension, and word classification; general information, sentence comprehension, and word meaning; general information, word classification, and word meaning; and sentence comprehension, word classification, and word meaning)? (4) What percentage of explained variance in paragraph comprehension is associated with the third-order common effect (i.e., general information, sentence comprehension, word classification, and word meaning)?

To perform the commonality analysis, functions in the commonality coefficient package were run with the commands listed in Table 5. The resulting output is presented in Table 6. The first part of the table of the output presents the partitioning of the regression effect. Note that the total of the commonality coefficients is equal to the R^2 from the regression analysis (i.e., .6114). The individual entries in the table can be used to determine how much variance is explained by unique effects, first-order common effects, second-order common effects, and so forth, as well as which coefficients contribute most to the regression effect. In this example, the third-order effect (.2637) contributed 43.14% of the explained variance in paragraph comprehension (.6114). The second-order effect, involving the `general`, `sentence`, and `wordm` variables, provided 16.28% of the regression effect. The first-order effect, involving the `sentence` and `wordm` variables, provided 7.69% of the explained variance in paragraph comprehension. Finally, the `sentence` and `wordm` variables uniquely provided 8.79% and 5.54%, respectively, of the explained variance in paragraph comprehension. In total, these five effects account for over 81.44% of the explained variance in paragraph comprehension (note that this is 81.44% of the effect, not of the total variance in the dependent variable). These results indicate that a large amount of the regression effect was explained by

common effects. Notable exceptions were the `sentence` and `wordm` variables.

The second part of the table in the sample output (see Table 6) provides another view of the commonality effects. The unique effect for each predictor is tabularized, as well as the total of all common effects for which the predictor is involved. The last column is a sum of the unique and common totals. These totals can be compared alongside the B (unstandardized) weights resulting from the multiple regression to add another layer of consideration when determining which variables are important to a regression equation (see Table 7). In this example, the regression coefficients for the `general` and `wordc` variables are statistically insignificant. Although their unique contributions to the regression effect (.0039 and .0029) are consistent with these findings, combining these contributions with their total involvement in common effects presents a different picture. In total, the `general` and `wordc` variables were involved with 70.85% and 55.41% of the explained variance (R^2) in passage comprehension. Therefore, although these variables provided little unique contribution to the regression effect, they did share a significant amount of variance with the regression effect.

Note that the percentage of variance explained by each variable generated from the commonality analysis can also be computed as a squared structure coefficient from a regression analysis, with identical results. A structure coefficient is the bivariate correlation between predictor scores and the predicted values ($\hat{y}$) resulting from a regression equation (Courville & Thompson, 2001; Thompson, 2006). A squared structure coefficient, therefore, represents the amount of variance that a predictor shares with the regression effect. For a given predictor (x), the squared structure coefficient can be computed using the formula

$$r_s^2 = \frac{r_{x.y}^2}{R^2} \tag{6}$$

where $r_{x.y}^2$ equals the square of the bivariate correlation between x and y, which can also be derived by summing x's unique and common effects. The benefit of employing commonality analysis in conjunction with the analysis of squared structure coefficients is that the researcher

Table 5
Commands to Run `commonalityCoefficients`

Command 1:
```
CCData=commonalityCoefficients(HS.data, "paragrap",
list("general", "sentence", "wordc", "wordm"), "F")
```
Command 2:
```
print (CCData)
```

Table 6
Output Example

```
$CC:
                                                    Coefficient   % Total
Unique to general                                        0.0039      0.65
Unique to sentence                                       0.0537      8.79
Unique to wordc                                          0.0029      0.48
Unique to wordm                                          0.0339      5.54
Common to general, and sentence                          0.0127      2.07
Common to general, and wordc                             0.0022      0.37
Common to sentence, and wordc                            0.0186      3.04
Common to general, and wordm                             0.0221      3.61
Common to sentence, and wordm                            0.0470      7.69
Common to wordc, and wordm                               0.0012      0.19
Common to general, sentence, and wordc                   0.0210      3.44
Common to general, sentence, and wordm                   0.0995     16.28
Common to general, wordc, and wordm                      0.0077      1.26
Common to sentence, wordc, and wordm                     0.0212      3.47
Common to general, sentence, wordc, and wordm            0.2637     43.14
Total                                                    0.6114    100.00

$CCTotalbyVar:
              Unique    Common    Total
general       0.0039    0.4290    0.4329
sentence      0.0537    0.4838    0.5375
wordc         0.0029    0.3357    0.3386
wordm         0.0339    0.4624    0.4963
```

can determine how much variance each variable uniquely contributes and how much each shares, if any, with every other variable in the regression.

Conclusions and Future Developments

It appears that commonality analysis is an analysis that few researchers are using. Not only is there the possibility that researchers do not understand the value of conducting a commonality analysis, the dearth of a program for computing the tedious calculations involved with a large set of predictors most certainly provides an obstacle.

By conducting a commonality analysis, researchers can clearly see the components of a regression effect, as well as examine how much variance a variable contributes uniquely or in common with other variables. In the heuristic example provided, the commonality analysis data showed that the majority of the regression effect was explained by a small subset of unique and common effects. It further showed that each of the predictors shared a significant amount of variance with the regression effect.

The software package presented provides researchers with a straightforward vehicle with which to compute com-

monality coefficients in the multiple regression context. The R functions that make up the commonality coefficient package appear in the Appendix and can be obtained at no cost by contacting the corresponding author. It is the intention of the authors to continue development on this package. Further improvements could include updating the package to accommodate other multivariate analysis (e.g., canonical correlation) and converting the package so that it can be utilized with the Statistical Package for the Social Sciences (SPSS).

AUTHOR NOTE

Correspondence concerning this article should be addressed to K. Nimon, 18352 Dallas Parkway, #136-407, Dallas, TX 75287 (e-mail: kim.nimon@gmail.com).

REFERENCES

AMADO, A. J. (2003). Partitioning predicted variance into constituent parts: A primer on regression commonality analysis. *Research in the Schools*, **10**, 91-97.

COURVILLE, T., & THOMPSON, B. (2001). Use of structure coefficients in published multiple regression articles: β is not enough. *Educational & Psychological Measurement*, **61**, 229-248.

FOX, J. (1991). *Regression diagnostics: An introduction* (Sage University Paper Series on Quantitative Applications in the Social Sciences, Series 07-079). Newbury Park, CA: Sage.

Table 7
Regression Results for Heuristic Data Predicting Paragraph Comprehension (y)

Predictor (x)	R	R^2	R^2_{adj}	B	Sig. of B	Unique	Common	Total	% of R^2
Model 1	.782	.611	.606						
Constant				.071	.910				
General				.030	.084	.0039	.4290	.4329	70.85
Sentence				.263	<.001	.0537	.4838	.5375	87.97
Wordc				.047	.136	.0029	.3357	.3386	55.41
Wordm				.137	<.001	.0339	.4624	.4963	81.23

Note—Sig., significance; Unique, x's unique effect; Common, Σ x's common effects; Total = Unique + Common; % of R^2 = Total/R^2.

HOLZINGER, K. J., & SWINEFORD, F. (1939). *A study in factor analysis: The stability of a bi-factor solution* (Supplementary Monographs No. 48, pp. 81-91). Chicago: University of Chicago, Department of Education.

HORNIK, K. (2007). *The R FAQ*. Retrieved April 14, 2007, from cran.r-project.org/doc/FAQ/.

HUBERTY, C. J. (1989). Problems with stepwise methods—better alternatives. In B. Thompson (Ed.), *Advances in social science methodology* (Vol. 1, pp. 43-70). Greenwich, CT: JAI Press.

KELLEY, K. (2006). Methods for the behavioral, educational, and social sciences (MBESS) [Computer software and manual]. Available at www.cran.r-project.org/.

KERLINGER, F. N. (1986). *Foundations of behavioral research* (3rd ed.). New York: Holt, Rinehart & Winston.

MAYESKE, G. W., COHEN, W. M., WISLER, C. E., OKADA, T., BEATON, A. E., PROSHEK, J. M., ET AL. (1969). *A study of our nation's schools.* Washington, DC: U.S. Department of Health, Education, and Welfare, Office of Education.

MOOD, A. M. (1969). Macro-analysis of the American educational system. *Operations Research*, **17**, 770-784.

MOOD, A. M. (1971). Partitioning variance in multiple regression analyses as a tool for developing learning models. *American Educational Research Journal*, **8**, 191-202.

MORRIS, J. D. (1976). A computer program to accomplish commonality analysis. *Educational & Psychological Measurement*, **36**, 721-723.

NEWTON, R. G., & SPURRELL, D. J. (1967). A development of multiple regression for the analysis of routine data. *Applied Statistics*, **16**, 51-64.

ONWUEGBUZIE, A. J., & DANIEL, L. G. (2003, February 19). Typology of analytical and interpretational errors in quantitative and qualitative educational research. *Current Issues in Education* [Online], **6**(2). Available at cie.ed.asu.edu/volume6/number2/.

OXFORD, R. M., & DANIEL, L. G. (2001). Basic cross-validation: Using the "holdout" method to assess the generalizability of results. *Research in the Schools*, **8**, 83-89.

PEDHAZUR, E. J. (1997). *Multiple regression in behavioral research: Explanation and prediction* (3rd ed.). Fort Worth, TX: Harcourt Brace.

R DEVELOPMENT CORE TEAM (2007). R-project Internet home page, R Foundation for Statistical Computing. Available at www.r-project.org/.

ROWELL, R. K. (1991, January). *Partitioning predicted variance into constituent parts: How to conduct commonality analysis.* Paper presented at the annual meeting of the Southwest Educational Research Association, San Antonio, TX. (ERIC Document Reproduction Service No. ED328589)

ROWELL, R. K. (1996). Partitioning predicted variance into constituent parts: How to conduct regression commonality analysis. *Advances in Social Science Methodology*, **4**, 33-43.

SAS INSTITUTE INC. (1999). *SAS/STAT user's guide, Version 8*. Cary, NC: Author.

SEIBOLD, D. R., & MCPHEE, R. D. (1979). Commonality analysis: A method for decomposing explained variance in multiple regression analysis. *Human Communication Research*, **5**, 355-363.

THOMPSON, B. (1995). Stepwise regression and stepwise discriminant analysis need not apply here: A guidelines editorial. *Educational & Psychological Measurement*, **55**, 525-534.

THOMPSON, B. (2006). *Foundations of behavioral statistics: An insight-based approach*. New York: Guilford.

ZIENTEK, L. R., & THOMPSON, B. (2006). Commonality analysis: Partitioning variance to facilitate better understanding of data. *Journal of Early Intervention*, **28**, 299-307.

NOTES

1. A complete discussion of commonality analysis is beyond the scope of this article; readers are referred to accessible treatments of the topic by Amado (2003), Mood (1971), Pedhazur (1997), Rowell (1996), Seibold and McPhee (1979), and Zientek and Thompson (2006).

2. SAS software (SAS Institute Inc., 1999) does have the PROC RSQUARE statement that will calculate R^2 values for all possible combinations of independent variables in the model, one of the steps in completing a commonality analysis. Also, a FORTRAN IV computer program to accomplish commonality analysis was introduced by Morris in 1976. However, this program is now obsolete, since it requires input job control cards.

APPENDIX
Commonality Coefficient Package for R

```
#########################################################################
commonalityCoefficients <-function (dataMatrix, dv, ivlist, diag=F){
#########################################################################

##DESCRIPTION
##Returns a list of two tables.
##The first table (CC) contains the list of commonality coeffcents and % of variance for each effect.
##The second table (CCTotalByVar) totals the unique and common effects for each independent variable.


##REQUIRED ARGUMENTS
##dataMatrix      dataset containing the dependent and independent variables
##dv              dependent variable named in the dataset
##iv              list of independent variables named in the dataset


##OPTIONAL ARGUMENTS
##diag            diagnostic flag - default to FALSE


##PSEUDO CODE
## Determine the number of independent variables (n).
## Generate an ID for each independent variable to 2^(n-1).
## For example, the ID of the 1st independent variable is 2^0 =1.
## Determine the number of commonality coefficients (2^n-1).
## Generate a bitmap matrix containing the bit representation of each commonality coefficient.
## Use the bitmap matrix to compute the R2 value for each combination of independent variables.
## Store the R2 value based on an index that is computed by ORing the IDs of the related IV.
## Use the bitmap matrix to generate the list of R2 values needed for each commonality coefficient.
## Use the list of R2 values to compute each commonality coefficient.
```

APPENDIX (Continued)

```
## Calculate the % explained variance for each commonality coefficient.
## Use the bitmap matrix to generate row headings for the first output table.
## Use the bitmap matrix to total the commonality coefficient effects by variable.
## Return the list of two tables.

## Determine the number of independent variables.
ivlist <- unlist(ivlist)
nvar=length(ivlist)

## Generate an ID for each independent variable to 2^(n-1).
ivID <- matrix(nrow=nvar,ncol=1)
for (i in 0: nvar-1){
        ivID[i+1]=2^i
}
if (diag=="T") print (ivID)
## Determine the number of commonality coefficients.
numcc=2**nvar-1

## Generate a matrix containing the bit representation of each commonality coefficient
effectBitMap<-matrix(0, nvar, numcc)
for (i in 1:numcc){
        effectBitMap<-setBits(i, effectBitMap)
}
if (diag=="T") print (effectBitMap)

## Use the bitmap matrix to compute the R2 value for each combination of independent variables
## Store the R2 value based on an index that is computing by ORing the IDs of the related IVs.
commonalityMatrix <- matrix(nrow=numcc,ncol=3)
for (i in 1: numcc){
        formula=paste(dv,"~", sep="")
        for (j in 1: nvar){
                bit = effectBitMap[j,i]
                if (bit == 1){
                        formula=paste(formula,paste("+",ivlist[[j]], sep=""), sep="")
                }
        }
        commonalityMatrix[i,2]<-summary(lm(formula,dataMatrix))$r.squared
}
if (diag=="T") print (commonalityMatrix)

## Use the bitmap matrix to generate the list of R2 values needed.
commonalityList<-vector("list", numcc)
for (i in 1: numcc){
        bit = effectBitMap[1,i]
        if (bit == 1) ilist <-c(0,-ivID[1])
                else ilist<-ivID[1]
        for (j in 2: nvar){
                bit = effectBitMap[j,i]
                if (bit == 1){
                        alist<-ilist
                        blist<-genList(ilist,-ivID[j])
                        ilist<-c(alist,blist)
                }
                else ilist<-genList(ilist,ivID[j])
        }
        ilist<-ilist*-1
        commonalityList[[i]]<-ilist
}
if (diag=="T") print (commonalityList)

## Use the list of R2 values to compute each commonality coefficient.
for (i in 1: numcc){
        r2list <- unlist(commonalityList[i])
        numlist = length(r2list)
```

APPENDIX (Continued)

```
        ccsum=0
        for (j in 1:numlist){
                indexs = r2list[[j]]
                indexu = abs (indexs)
                if (indexu !=0) {
                        ccvalue = commonalityMatrix[indexu,2]
                        if (indexs < 0)ccvalue = ccvalue*-1
                        ccsum=ccsum+ccvalue
                }
        }
        commonalityMatrix[i,3]=ccsum
}
if (diag=="T") print (commonalityMatrix)

## Calculate the % explained variance for each commonality coefficient.
orderList<-vector("list", numcc)
index=0
for (i in 1:nvar){
        for (j in 1:numcc){
                nbits=sum(effectBitMap[,j])
                if (nbits == i){
                        index=index+1
                        commonalityMatrix[index,1]<-j
                }
        }
}
if (diag == "T") print (commonalityMatrix)

## Prepare first output table.
outputCommonalityMatrix <- matrix(nrow=numcc+1,ncol=2)
totalRSquare <- sum(commonalityMatrix[,3])
for (i in 1:numcc){
        outputCommonalityMatrix[i,1]<-round(commonalityMatrix[commonalityMatrix[i,1],3], digit=4)
        outputCommonalityMatrix[i,2]<-
round((commonalityMatrix[commonalityMatrix[i,1],3]/totalRSquare)*100, digit=2)
}
outputCommonalityMatrix[numcc+1,1]<-round(totalRSquare,digit=4)
outputCommonalityMatrix[numcc+1,2]<-round(100,digit=4)

## Use the bitmap matrix to generate row headings for the first output table.
rowNames=NULL
for (i in 1: numcc){
        ii=commonalityMatrix[i,1]
        nbits=sum(effectBitMap[,ii])
        cbits=0
        if (nbits==1) rowName="Unique to "
                else rowName = "Common to "
        for (j in 1:nvar){
                if (effectBitMap[j,ii]==1){
                        if (nbits==1)rowName=paste(rowName,ivlist[[j]],sep= "")
                        else {
                                cbits=cbits+1
                                if (cbits==nbits){
                                        rowName=paste(rowName,"and ", sep="")
                                        rowName=paste(rowName,ivlist[[j]],sep="")
                                }
                                else{
                                        rowName=paste(rowName,ivlist[[j]],sep= "")
                                        rowName=paste(rowName,",", sep="")
                                }
                        }
                }
        }
        rowNames=c(rowNames,rowName)
```

APPENDIX (Continued)

```
}
rowNames=c(rowNames,"Total")
rowNames<-format.default(rowNames,justify="left")
colNames<-format.default(c ("Coefficient", " % Total"), justify="right")
dimnames(outputCommonalityMatrix)<-list(rowNames,colNames)
if (diag =="T") print (outputCommonalityMatrix)


## Use the bitmap matrix to total the commonality coefficient effects by variable.
outputCCbyVar<-matrix(nrow=nvar,ncol=3)
for (i in 1:nvar){
        outputCCbyVar[i,1]=outputCommonalityMatrix[i,1]
        outputCCbyVar[i,3]=round(sum(effectBitMap[i,]*commonalityMatrix[,3]), digit=4)
        outputCCbyVar[i,2]=outputCCbyVar[i,3]-outputCCbyVar[i,1]
}
dimnames(outputCCbyVar)<-list(ivlist,c("Unique", "Common", "Total"))


## Return the list of two output tables.
outputList<-list(CC=outputCommonalityMatrix, CCTotalbyVar=outputCCbyVar)
return (outputList)
}


#######################################################################
setBits<-function(col, effectBitMap) {
#######################################################################


##DESCRIPTION
##Creates the binary representation of n and stores it in the nth column of the matrix


##REQUIRED ARGUMENTS
##col              Column of matrix to represent in binary image
##effectBitMap     Matrix of mean combinations in binary form


##Initialize variables
row<-1
val<-col
##Create the binary representation of col and store it in its associated column
##One is stored in col 1; Two is stored in col 2; etc.
##While (val >= 1)
## If the LSB of val is 1; increment the appropriate entry in combo matrix
## Shift the LSB of val to the right
        while (val!=0){
        if (odd(val)) {
                effectBitMap[row,col]=1
        }
        val<-as.integer(val/2)
        row<-row+1
}


##Return matrix
return(effectBitMap)
}


#######################################################################
odd<-function(val) {
#######################################################################
##DESCRIPTION
##Returns true if value is odd; false if true


##REQUIRED ARGUMENTS
##val                                 Value to check


##Returns true if val is odd
        if (((as.integer(val/2))*2)!=val) return(TRUE)
        return (FALSE)
```

APPENDIX (Continued)

```
}

########################################################################
genList<-function(ivlist, value){
########################################################################

numlist = length(ivlist)
newlist<-ivlist
newlist=0
for (i in 1:numlist){
        newlist[i]=abs(ivlist[i])+abs(value)
        if(((ivlist[i]<0) && (value >= 0))|| ((ivlist[i]>=0) && (value <0)))newlist[i]=newlist[i]*-1
}
return(newlist)
}
```