Technical Report 1394

Interpreting Johnson's Relative Weights Analysis: Insights from New Formulae and Computational Tools

Garett N. Howardson
U.S. Army Research Institute

June 2021

United States Army Research Institute for the Behavioral and Social Sciences

U.S. Army Research Institute
for the Behavioral and Social Sciences

Department of the Army,
Deputy Chief of Staff, G1

Authorized and approved:

MICHELLE L. ZBYLUT, Ph.D.
Director

Technical review by

Rachel Amey, U.S. Army Research Institute

DISPOSITION

This Technical Report has been submitted
to the Defense Information Technical Center (DTIC).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
June 2021	Final	May 2018 – January 2021

4. TITLE AND SUBTITLE

Interpreting Johnson's Relative Weights Analysis: Insights from New Formulae and Computational Tools

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHOR(S)
Howardson, Garett N.

5d. PROJECT NUMBER
AA4

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
591

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U. S. Army Research Institute
for the Behavioral & Social Sciences
6000 6TH Street (Bldg. 1464 / Mail Stop 5610)
Fort Belvoir, VA 22060-5610

8. PERFORMING ORGANIZATION REPORT NUMBER

Technical Report 1394

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Institute
for the Behavioral & Social Sciences
6000 6TH Street (Bldg. 1464 / Mail Stop 5610)
Fort Belvoir, VA 22060-5610

10. SPONSOR/MONITOR'S ACRONYM(S)
ARI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Technical Report 1394

12. DISTRIBUTION/AVAILABILITY STATEMENT:
Distribution Statement A: Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES
ARI Research POC: Dr. Garett N. Howardson, Predictive Analytics and Modeling Research Unit

14. ABSTRACT

Relative weights and dominance analysis offer two promising relative importance methods for multiple regression. Whereas dominance analysis offers more statistically interpretable solutions, calculating such solutions is computationally burdensome. Conversely, although relative weights are computationally simpler, interpreting said weights is more difficult. This paper presents a new relative weights derivation and formula that is both computationally efficient and clarifies relative weights' statistical interpretation. Specifically, the derivation contained herein suggests two new relative weights interpretations drawing from (1) principal components analysis (PCA) and (2) generalized least squares (GLS) regression. This new method is then translated into the R language while also explicitly clarifying the relationship between mathematical formulae and computer code. The new method contains two key advantages over existing computational implementations in that the new method (1) avoids matrix inversion entirely and (2) is better suited for modern missing data methods (e.g., full information maximum likelihood). The inferential and computational benefits of this new method notwithstanding, caution is ultimately warranted in that the new method makes clear certain relative weights matrix choices are, in current practice, arbitrary.

15. SUBJECT TERMS
Variable Importance, Multiple Regression, Statistical Interpretation, Statistical Computing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Dr. Gregory A. Ruark
a. REPORT	b. ABSTRACT	c. THIS PAGE	Unlimited	35	19b. TELEPHONE NUMBER
Unclassified	Unclassified	Unclassified	Unclassified		703-545-2441

Technical Report 1394

Interpreting Johnson's Relative Weights Analysis: Insights from New Formulae and Computational Tools

Garett N. Howardson
U.S. Army Research Institute

Foundational Science Research Unit
Gregory A. Ruark, Chief

June 2021

EXECUTIVE SUMMARY

Research Requirement:

The U.S. Army currently uses a whole-person approach to personnel testing and assessment wherein multiple sources of information are combined to form an overall individual assessment. As the volume and variety of individual information continue to expand, it becomes all the more important to determine which pieces of information are relevant to the whole-person approach and which pieces of information are superfluous. This process of separating the relevant information from the irrelevant information is known as *relative importance* analysis. Unfortunately, currently used relative importance methods (e.g., relative weights analysis, dominance analysis) cannot easily incorporate the variety and volume of contemporary Soldier data and suffer from several computational and inferential challenges.

Procedure:

The purpose of this research program is to mathematically and computationally demonstrate that extant relative importance analysis methods such as *relative weights analysis* and *dominance analysis* are special cases of a more general statistical learning theory of relative importance. This specific Technical Report re-derives Johnson's (2000) method of relative weights analysis using different matrix algebra methods (e.g., Hadamard product) that translate into expository computer code.

Findings:

This Technical Report demonstrates that Johnson's (2000) relative weights analysis is actually a form of principal components analysis (PCA) and is also related to the generalized least squares (GLS) multiple regression estimator. Further, the specific formulae and computational algorithms provided here clarify the interpretation of Johnson's (2000) relative weights and offer computationally more flexible and efficient means of calculating said relative weights. Finally, the interpretational clarity offered here also suggests that relative weights analysis should be used cautiously unless researchers' explicit analytic goals are to perform PCA.

Utilization and Dissemination of Findings:

The findings offered here expand the scope and rigor of current relative importance methods to include more contemporary statistical and computational methods from machine learning and artificial intelligence. As is common with many such methods, however, the interpretational challenges of using such methods should be weighed carefully against any gains in computational efficiency, especially when using these methods to derive meaningful conclusions about which pieces of information are truly important for evaluating the whole Soldier.

The insights gained from this Technical Report's derivations are also implemented algorithmically and presented in both vectorized and iterative R code. Both the vectorized and

iterative versions are presented in three ways: (1) Tables 2 and 3 map non-executable R code to the algorithm steps presented in Table 1 and to the equations presented in the text, (2) Tables A1 and A2 present executable R code with comments describing the relevant variables and functions in greater detail, and (3) Tables A3 and A4 present compact, executable R code with no comments.

CONTENTS

Page

Page

List of Tables

List of Figures

INTERPRETING JOHNSON'S RELATIVE WEIGHTS ANALYSIS: INSIGHTS FROM NEW FORMULAE AND COMPUTATIONAL TOOLS

Introduction

Organizational researchers increasingly rely on standardized effect sizes to make decisions. Effect sizes are more intuitive (Cumming, 2014; Cumming & Finch, 2005), conceptually simpler than standard statistical results (Brooks, Dalal, & Nolan, 2014), and are independent of sample-specific features (e.g., sample size) allowing for more meaningful comparisons across studies (Kelly & Preacher, 2012). Such characteristics make effect sizes particularly attractive to applied researchers in that communicating complex scientific findings to lay decision makers is greatly simplified (Brooks et al., 2014).

Relative importance multiple regression methods offer predictor-specific effect sizes accounting for between-predictor correlations (Azen & Budescu, 2003; Braun & Oswald, 2011; Budescu, 1993; J. Johnson, 2000; J. Johnson & LeBreton, 2004; Nimon & Oswald, 2013). Two of the most popular such relative importance methods are *dominance analysis* (Azen & Budescu, 2003) and *relative weights* (J. Johnson, 2000). Although dominance analysis weights are statistically more interpretable (e.g., Grömping, 2007; Thomas, Zumbo, Kwan, & Schweitzer, 2014), relative weights are computationally more tractable (J. Johnson, 2000). The purpose of this research is to address this computational tractability / theoretical interpretability tradeoff to improve relative weights' statistical interpretability.

To do so, I first derive a new set of relative weights equations based on the original formulation in J. Johnson (2000). The derivation presented herein is novel in that I use different matrix operations and matrix identities making clearer relative weights' statistical interpretation. These new matrix operations and identities are then translated into summation algebra, which is likely more familiar to applied researchers than the original matrix representations.

Challenging is that that the original relative weights matrix properties allow for two algebraically equivalent formulae (see Equations 16 and 17 below). Using the path tracing rules of structural equation modeling, however, I demonstrate that only one of these representations (Equation 17) is valid and warrants interpretation. I then offer two interpretations of the final formula: one related to principal components analysis (PCA) and the other related to generalized least squares (GLS) regression. Step-by-step instructions are also offered for computationally implementing the matrix and the summation algebra versions in the R statistical programming environment (R Core Team, 2018).

The arguments that follow require an introductory knowledge of matrix algebra. For readers unfamiliar with matrix algebra, there are a number of introductory texts designed specifically for social science researchers. One short and quite accessible such work is Namboodiri (1984). Additionally, the "matlib" package for R (https://github.com/friendly/matlib) offers a comprehensive introduction to learning matrix algebra with R. While a full exposition of such matrix operations is beyond the scope of the current paper, certain critical operations are described briefly below before continuing to the core derivations. Throughout the remainder of

this paper, matrices are indicated by bold, upper case symbols such as $\mathbf{X}$, $\mathbf{V}$, $\mathbf{D}$, $\mathbf{U}$, or $\mathbf{\Lambda}$; vectors are indicated by bold, lower case symbols such as $\mathbf{y}$, $\boldsymbol{\beta}$, $\mathbf{1}$, $\mathbf{0}$, or $\boldsymbol{\varepsilon}^2$; and scalars or single numbers are represented by un-bolded symbols such as N, p, 1, or σ.

Overview of Relevant Matrix Operations

Let any matrix $\mathbf{M}$ superscripted with T (e.g., $\mathbf{M}^T$) indicate the transposition operation whereby the matrix rows and columns are switched. Then, let any matrix $\mathbf{M}$ superscripted with an asterisk (e.g., $\mathbf{M}^*$) indicate conjugate transposition operation. With respect to real numbers, the distinction between simple (e.g., $\mathbf{M}^T$) and conjugate (e.g., $\mathbf{M}^*$) transposition disappears. When the matrix $\mathbf{M}$ contains complex numbers, however, the two operations diverge in that the conjugate transpose must also negate the imaginary components of its complex numbers. Further, when the columns of $\mathbf{M}$ are standardized and orthogonal to each other, $\mathbf{M}$ becomes of a special kind known as a *unitary* matrix.

Most notable is the property that a unitary matrix's conjugate transpose is identical to its inverse (Arndt, 2011; Golub & Van Loan, 2013). Let the inverse of any matrix $\mathbf{M}$ be denoted $\mathbf{M}^{-1}$ and defined as any matrix such that when pre- or post-multiplied by the original matrix produces an identity matrix with unit main diagonal and zeroes elsewhere. In other words, $\mathbf{M}^{-1}$ is an inverse of $\mathbf{M}$ if and only if $\mathbf{M}\mathbf{M}^{-1} = \mathbf{M}^{-1}\mathbf{M} = I$. If the matrix $\mathbf{M}$ is unitary, then it also holds that $\mathbf{M}^{-1} = \mathbf{M}^*$ and, consequently, that $\mathbf{M}\mathbf{M}^* = \mathbf{M}^*\mathbf{M} = I$.

Note that the inverse or transpose of a matrix product is the product of each matrix's inverse with their order reversed. In other words, $(\mathbf{QR})^{-1} = \mathbf{R}^{-1}\mathbf{Q}^{-1}$ and $(\mathbf{QR})^T = \mathbf{R}^T\mathbf{Q}^T$ and this extends to any number of matrices used in the product such as $(\mathbf{QRP})^{-1} = \mathbf{P}^{-1}\mathbf{R}^{-1}\mathbf{Q}^{-1}$ and $(\mathbf{QRP})^T = \mathbf{P}^T\mathbf{R}^T\mathbf{Q}^T$. Note also that the inverse of an inverse returns the original matrix, as with:

$$\begin{aligned}[(\mathbf{QR})^{-1}]^{-1} &= [\mathbf{R}^{-1}\mathbf{Q}^{-1}]^{-1} \\ &= (\mathbf{Q}^{-1})^{-1}(\mathbf{R}^{-1})^{-1} \\ &= \mathbf{QR}\end{aligned}$$

Likewise, the transpose of a transpose gives the original matrix as with:

$$\begin{aligned}[(\mathbf{QR})^T]^T &= [\mathbf{R}^T\mathbf{Q}^T]^T \\ &= (\mathbf{Q}^T)^T(\mathbf{R}^T)^T \\ &= \mathbf{QR}.\end{aligned}$$

Since the transpose and inverse are one and the same for unitary matrices, any combination of inversion and transposition of the original matrix returns the original matrix. If $\mathbf{Q}$ and $\mathbf{R}$ are unitary matrices, in other words, the following hold true:

$$
\begin{aligned}
[(\mathbf{QR})^*]^{-1} &= [\mathbf{R}^*\mathbf{Q}^*]^{-1} \\
&= (\mathbf{Q}^*)^{-1}(\mathbf{R}^*)^{-1} \\
&= (\mathbf{Q}^{-1})^{-1}(\mathbf{R}^{-1})^{-1} \\
&= (\mathbf{Q}^*)^*(\mathbf{R}^*)^* \\
&= \mathbf{QR}.
\end{aligned}
$$

With these matrix operations in mind, the subsequent sections offer a detailed technical account of relative weights. While other such accounts certainly exist (e.g., J. Johnson, 2000), the method below employs slightly different identities and matrix operations, both of which ultimately lead to novel insights about relative weights' interpretation. Before describing in detail this new method, however, I offer an overview of the ordinary least squares (OLS) multiple linear regression problem.

Brief Overview of Ordinary Least Squares Multiple Regression

The general OLS multiple regression problem can be expressed as minimizing the objective function $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ where $\hat{\mathbf{y}}$ is the predicted outcome score and the double bar notation $\|\cdot\|_2^2$ is shorthand for the squared two norm or the squared Euclidean norm. Except for a multiplicative constant of $\frac{1}{N-1}$, $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ is equivalent to the residual variance, the error variance, or the mean squared prediction error given as $\sigma_e^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{y} - \hat{\mathbf{y}})^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{y} - \mathbf{X\beta})^2$. The minimum of this objective function occurs when $\mathbf{X}^T\hat{\mathbf{y}} = \mathbf{X}^T\mathbf{y}$ or, equivalently, when $\mathbf{X}^T\mathbf{X\beta} = \mathbf{X}^T\mathbf{y}$. Re-arranging the above and solving for β then gives the multiple regression coefficient solution of $\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

As shown in the original formulation (J. Johnson, 2000, see Equations 3 and 4 on p. 5 and p. 7, respectively) and several additional accounts thereof (e.g., Tonidandel et al., 2009, see Equations 7 and 8 on p. 398 and p. 390, respectively), the relative weights procedure calculates two sets of parameters (i.e., $\boldsymbol{\beta}_{RWA}$ and $\boldsymbol{\Lambda}$) using a similar OLS procedure. To do so, relative weights introduce a new set of variables $\mathbf{Z}$, which then are used as predictors in two OLS regressions. These relative weights procedures are described in much greater and formal detail in the following sections. To help situate such details, I first offer a brief conceptual overview of the relative weights problem.

Overview of Relative Weights Analysis

Relative weights analysis begins much as with the standard regression problem but emphasizes a new challenge: multicollinearity (e.g., Belsley, Kuh, & Welsch, 1980). In particular, the OLS solution noted above produces regression estimates that are completely independent of each other, holding constant all other variables when calculating each regression coefficient. Stated somewhat differently, the OLS regression coefficients produce estimates of the *unique* contribution of each predictor to the outcome after removing the influence of the remaining predictors, or while holding constant the remaining predictors. Often, however, variables in multiple regression are correlated with each other indicating that the standard regression

coefficient estimates omit the shared or common sources of explained variability in the outcome (LeBreton, Ployhart, & Ladd, 2004; Nimon, Henson, & Gates, 2010; Nimon, Lewis, Kane, & Haynes, 2008; Schoen, DeSimone, & James, 2010; Tonidandel, LeBreton, & J. Johnson, 2009).

Relative weights analysis strives to recover this shared or common explained outcome variability using a matrix decomposition method known as the *singular value decomposition*. In essence, relative weights analysis decomposes the set of observed predictors into a completely orthogonal or unrelated set of variables that, despite being uncorrelated with each other, are maximally correlated with one of the original predictors (J. Johnson, 2000). One might conceptualize this new set of orthogonal variables as a set of p latent variables with each observed predictor variable loading onto all p latent variables; the observed predictor loading highest onto each latent variable then serves as said latent variable's identifying predictor. This is not unlike identifying a common factor in confirmatory factor analysis by constraining one item loading to unity. In an ideal scenario, each predictor loads strongly onto only one latent predictor and loads weakly onto the remaining latent predictors, much as with rotating an exploratory common factor solution to simple structure (Mulaik, 2010).

The unique relationships between the predictors and outcome are then given by the relationships between these new, orthogonal latent variables and the outcome. The shared relationship between each predictor and the outcome through the remaining predictors is captured by the latent variables' loadings onto the set of observed predictors. Squaring and then multiplying the unique paths from the latent variables to the outcome by the set of paths from the latent variables to the predictors then gives each predictor's total importance in terms of both uniquely and commonly explained variability. An attractive characteristic of relative weights analysis is that if the predictors and outcome data are standardized to have unit variance, the total relative weights will sum to the model's squared multiple correlation. For more detailed insight into the relative weights analysis method, I offer below an in-depth technical description using the matrix algebra operations described above. In particular, the fact that a unitary matrix is its own inverse, or that $\mathbf{M}\mathbf{M}^{*} = \mathbf{M}\mathbf{M}^{-1} = \mathbf{M}^{*}\mathbf{M} = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$, will be used to simplify equations throughout.

The Relative Weights Method Technical Details

Let $\mathbf{X}$ represent an $N \times p$ multiple regression design matrix in standard score form where N is the sample size and p is the number predictors. Relative weights analysis begins by decomposing $\mathbf{X}$ into a set of new matrices possessing some useful properties, as shown below:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{*}. \quad (1)$$

The matrix $\mathbf{U}$ corresponds to the eigenvectors[1] of the matrix $\mathbf{X}\mathbf{X}^{T}$; the matrix $\mathbf{V}$ corresponds to the eigenvectors of $\mathbf{X}^{T}\mathbf{X}$; and $\mathbf{D}$ is a square matrix with the $\mathbf{X}^{T}\mathbf{X}$ and $\mathbf{X}\mathbf{X}^{T}$ singular values on its main diagonal and zeroes elsewhere (singular values are the square roots of eigenvalues).

[1] For the purposes of this Technical Report, an eigenvector can be conceptualized as a set of values that preserve the information and direction contained in an original set of correlations. If, for example, $\mathbf{X}_1$ and $\mathbf{X}_2$ are correlated $-.30$, the first element of an eigenvector (i.e., $\mathbf{v}_{12}$) encodes the direction of this correlations. As such, the matrix $\mathbf{V}$ then encodes the information about predictor correlations contained in $\mathbf{X}^{T}\mathbf{X}$, as does $\mathbf{U}$ encode the within-person correlations contained in $\mathbf{X}\mathbf{X}^{T}$.

Note the asterisk superscripting the matrix $\mathbf{V}$ indicating that $\mathbf{V}$ is a unitary matrix, the conjugate transpose of which is given by $\mathbf{V}^*$. The decomposition shown in Equation 1 is known as the singular value decomposition wherein both the $\mathbf{U}$ and $\mathbf{V}$ matrices are unitary (Golub & Van Loan, 2013; Horn & C. Johnson, 2013; Watkins, 2010). Relative weights analysis then proceeds using the matrices $\mathbf{U}$ and $\mathbf{V}$ to create a new set of auxiliary latent variables denoted as $\mathbf{Z}$. More specifically, this new set of latent predictor variables is given as (e.g., J. Johnson, 2000; Tonidandel et al., 2009):

$$\mathbf{Z} = \mathbf{U}\mathbf{V}^*. \tag{2}$$

It should be noted that Equation 2 contains several implicit assumptions about the specific singular value decomposition implementation. Most notably, Equation 2 assumes that the product term $\mathbf{U}\mathbf{V}^*$ exists, which is only possible when the matrices $\mathbf{U}$ and $\mathbf{V}^*$ have compatible dimensions. In the original relative weights method, it is stated that the matrices $\mathbf{U}$ and $\mathbf{V}$ correspond, respectively, to the eigenvectors of the matrices $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ (e.g., J. Johnson, 2000; Tonidandel et al. 2009). Since $\mathbf{X}\mathbf{X}^T$ has dimensions $N \times N$, so too does the matrix of eigenvectors for $\mathbf{X}\mathbf{X}^T$ (Arndt, 2011; Golub & Van Loan, 2013; Horn & C. Johnson, 2013; Watkins, 2010) requiring that the matrix $\mathbf{U}$ has dimensions $N \times N$. Further, since $\mathbf{V}$ is the matrix of eigenvectors for $\mathbf{X}^T\mathbf{X}$, both are of dimension $p \times p$. Thus, the matrices $\mathbf{U}$ and $\mathbf{V}^*$, as originally defined in the relative weights method, are not compatible in that an $N \times N$ matrix $\mathbf{U}$ cannot be post-multiplied by the $p \times p$ matrix $\mathbf{V}^*$.

For the matrix $\mathbf{Z}$ to exist, Equation 2 must only use a subset of the full $\mathbf{U}$ matrix. The procedure must select only p of the N columns in $\mathbf{U}$ (hereinafter $\mathbf{U}$ refers to a sample of p columns from the full $N \times N$ matrix $\mathbf{U}$). Selecting only p of the $\mathbf{U}$ columns has implications for Equation 1 in that $\mathbf{D}$ must then be a square $p \times p$ matrix with the $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ singular values on its main diagonal and zeroes elsewhere (i.e., $\mathbf{D}$ must be a diagonal matrix; Hastie, Tibshirani, & Friedman, 2009). If $\mathbf{D}$ is diagonal, then so too is its inverse $\mathbf{D}^{-1}$. The diagonal elements of $\mathbf{D}^{-1}$ are then given by the reciprocal of $\mathbf{D}$'s non-zero elements (Golub & Van Loan, 2013; Horn & C. Johnson, 2013; Watkins, 2010). Consider, for instance, the $1 \times p$ vector $[\sigma_1, \ldots, \sigma_p]$ collecting the p singular values from $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ (i.e., the main diagonal of $\mathbf{D}$); this vector's reciprocal is $[1/\sigma_1, \ldots, 1/\sigma_p]$ thereby giving the main diagonal elements of $\mathbf{D}^{-1}$.

When the above is true and $\mathbf{D}^{-1}$ exists, the relationship between Equations 1 and 2 is formalized by post-multiplying both sides of Equation 1 by $\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^*$ and simplifying using the unitary matrix properties noted above. Given $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^*$, that is, $\mathbf{X}\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^* = \mathbf{U}\mathbf{D}\mathbf{V}^*\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^* = \mathbf{U}\mathbf{V}^*$. Pre-multiplying both sides of Equation 1 by $\mathbf{U}\mathbf{D}^{-1}\mathbf{U}^*$ and simplifying gives an alternate procedure for constructing $\mathbf{Z}$ given as $\mathbf{U}\mathbf{D}^{-1}\mathbf{U}^*\mathbf{X} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^*\mathbf{U}\mathbf{D}\mathbf{V}^* = \mathbf{U}\mathbf{V}^* = \mathbf{Z}$. In other words, the matrix $\mathbf{Z}$ is a linearly transformed version of the original predictor matrix $\mathbf{X}$ (J. Johnson, 2000; Thomas et al., 2014), which is then scaled by the inverse singular values and transformed by either the $\mathbf{U}$ or $\mathbf{V}$ matrix, as formalized below in Equation 3:

$$\mathbf{X}\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^* = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^*\mathbf{X} = \mathbf{U}\mathbf{V}^* = \mathbf{Z}. \tag{3}$$

After creating $\mathbf{Z}$, the relative weights analysis procedure continues with the standard OLS regression procedure noted above. In relative weights terms, two new quantities $\boldsymbol{\beta}_{RWA}$ and $\boldsymbol{\Lambda}$ are estimated by substituting $\mathbf{Z}$ for $\mathbf{X}$ to calculate $\boldsymbol{\beta}_{RWA}$ and also substituting $\mathbf{X}$ for $\mathbf{y}$ to calculate $\boldsymbol{\Lambda}$ (J. Johnson, 2000; Tonidandel et al., 2009). More formally, $\boldsymbol{\beta}_{RWA}$ and $\boldsymbol{\Lambda}$ are given, respectively, as:

$$\boldsymbol{\beta}_{RWA} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y} \tag{4}$$

and

$$\boldsymbol{\Lambda} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X}. \tag{5}$$

The final step in the procedure involves combining the Equation 4 and 5 results to arrive at a single quantity representing each predictor's relative weight, which is sometimes presented as (see Tonidandel et al., 2009, Equation 9 on p. 390):

$$\boldsymbol{\varepsilon}^2 = \boldsymbol{\Lambda}^2\boldsymbol{\beta}_{RWA}^2. \tag{6}$$

Before continuing it should be noted that the matrix operations implied by Equation 6 deviate from the relative weights' matrix operations as described in other relative weights publications (J. Johnson, 2000; Tonidandel & LeBreton, 2015). Whereas Equation 6 implies the squared matrix operation proper, the relative weights computation suggests the element-wise square of each element in $\boldsymbol{\Lambda}$ and $\boldsymbol{\beta}_{RWA}$. Before describing the detailed relative weights derivations, I briefly discuss below the difference between the matrix square proper and element-wise matrix square operations, an understanding of which is important for interpreting the new formula presented later in the report.

Matrix Squares Proper Versus Element-Wise Matrix Squares

In strict matrix terminology (Arndt, 2011; Golub & Van Loan, 2013), the square of a matrix, or a matrix raised to the power of two, is the product of a matrix and itself. In terms of Equation 6, $\boldsymbol{\Lambda}^2$ and $\boldsymbol{\beta}_{RWA}^2$ would then be defined as $\boldsymbol{\Lambda}\boldsymbol{\Lambda}$ and $\boldsymbol{\beta}_{RWA}\boldsymbol{\beta}_{RWA}$. In the relative weights literature, the quantities described and calculated in Equation 6 are not the matrix squares proper. Rather, the quantities are the squares of each matrix element. These quantities are also known as the element-wise matrix product or the Hadamard product operator, which is often denoted by the $\circ$ symbol (Golub & Van Loan, 2013). Using the Hadamard product, the relative weights matrices described as $\boldsymbol{\Lambda}^2$ and $\boldsymbol{\beta}_{RWA}^2$ become $\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda}$ and $\boldsymbol{\beta}_{RWA} \circ \boldsymbol{\beta}_{RWA}$, respectively. Thus, Equation 6 should be re-expressed as

$$\boldsymbol{\varepsilon}^2 = (\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda})(\boldsymbol{\beta}_{RWA} \circ \boldsymbol{\beta}_{RWA}). \tag{7}$$

Although subtle, the distinction between Equations 6 and 7 has non-trivial implications in that the Hadamard or element-wise products in Equation 7 are computationally simpler than the full matrix product operations implied by Equation 6. Moreover, this computational burden can be further decreased noting that the relative weights terms $\boldsymbol{\Lambda}$ and $\boldsymbol{\beta}_{RWA}$ can, and as will be shown below, be calculated without inverting the $\mathbf{Z}^T\mathbf{Z}$ matrix. In other words, I show below that the $(\mathbf{Z}^T\mathbf{Z})^{-1}$ term in Equations 4 and 5 may be eliminated without loss of generality. It should be noted that the original relative weights paper demonstrated precisely the same noting that $\mathbf{Z}^T\mathbf{Z} =$

$\mathbf{VU}^*\mathbf{UV}^*$ and that, by the unitary matrix properties (see above), $\mathbf{VU}^*\mathbf{UV}^* = \mathbf{VV}^* = \mathbf{I}$ requiring that $\mathbf{Z}^T\mathbf{Z} = (\mathbf{Z}^T\mathbf{Z})^{-1} = \mathbf{I}$ (see J. Johnson, 2000). Although the derivation below ultimately reaches the same result, I use instead the Equation 3 identity that $\mathbf{Z} = \mathbf{XVD}^{-1}\mathbf{V}^*$. Using the Equation 3 identity demonstrates a relationship between the ordinary least squares estimator shown above and the relative weights quantities $\boldsymbol{\beta}_{RWA}$ and $\boldsymbol{\Lambda}$.

Simplifying the Relative Weights Equations

Beginning with $\boldsymbol{\beta}_{RWA} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y}$ (Equation 4), substituting for each instance of $\mathbf{Z}$ the term $\mathbf{XVD}^{-1}\mathbf{V}^*$(see Equation 3) and simplifying according to the matrix operations and unitary matrix properties, the result is:

$$\begin{aligned}\boldsymbol{\beta}_{RWA} &= [(\mathbf{XVD}^{-1}\mathbf{V}^*)^*\mathbf{XVD}^{-1}\mathbf{V}^*]^{-1}(\mathbf{VD}^{-1}\mathbf{V}^*)^*\mathbf{X}^T\mathbf{y} \\ &= (\mathbf{VD}^{-1}\mathbf{V}^*\mathbf{X}^T\mathbf{XVD}^{-1}\mathbf{V}^*)^{-1}\mathbf{VD}^{-1}\mathbf{V}^*\mathbf{X}^T\mathbf{y} \\ &= \mathbf{VDV}^*(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{VDV}^*\mathbf{VD}^{-1}\mathbf{V}^*\mathbf{X}^T\mathbf{y} \\ &= \mathbf{VDV}^*(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{VDV}^*\boldsymbol{\beta}.\end{aligned} \tag{8}$$

Equation 8 demonstrates that $\boldsymbol{\beta}_{RWA}$ is the OLS estimator for $\boldsymbol{\beta}$ pre-multiplied by the term $\mathbf{VDV}^*$, which will be elucidated below. To do so, I repeat the steps in Equation 8 for $\boldsymbol{\Lambda}$ (see Equation 5) and simplify to arrive at:

$$\begin{aligned}\boldsymbol{\Lambda} &= (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X} \\ &= (\mathbf{VD}^{-1}\mathbf{V}^*\mathbf{X}^T\mathbf{XVD}^{-1}\mathbf{V}^*)^{-1}\mathbf{VD}^{-1}\mathbf{V}^*\mathbf{X}^T\mathbf{X} \\ &= \mathbf{VDV}^*(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{VDV}^*\mathbf{VD}^{-1}\mathbf{V}^*\mathbf{X}^T\mathbf{X} \\ &= \mathbf{VDV}^*(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X} \\ &= \mathbf{VDV}^*.\end{aligned} \tag{9}$$

The Equation 9 identity $\boldsymbol{\Lambda} = \mathbf{VDV}^*$ can then be substituted into Equation 8 to give $\boldsymbol{\beta}_{RWA} = \boldsymbol{\Lambda}\boldsymbol{\beta}$, which will become instrumental in subsequent sections. Note for now, however, that $\mathbf{Z}^T\mathbf{Z} = \mathbf{VU}^*\mathbf{UV}^*$, and that substituting this identity into Equation 9 gives an alternative definition of the lambda matrix as (see also J. Johnson, 2000):

$$\begin{aligned}\boldsymbol{\Lambda} &= (\mathbf{VU}^*\mathbf{UV}^*)^{-1}\mathbf{Z}^T\mathbf{X} \\ &= \mathbf{VU}^*\mathbf{UV}^*\mathbf{Z}^T\mathbf{X} \\ &= \mathbf{VU}^*\mathbf{UV}^*\mathbf{VU}^*\mathbf{X} \\ &= \mathbf{VU}^*\mathbf{X} \\ &= \mathbf{Z}^T\mathbf{X}.\end{aligned}$$

Next, consider that the OLS estimator method can be expressed in singular value decomposition terms by substituting for each instance of $\mathbf{X}$ in the OLS equations its singular value decomposition given in Equation 1 above. Doing so, the OLS regression coefficients' solution can be derived as (see Hastie et al., 2009):

$$\begin{aligned}\boldsymbol{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= ((\mathbf{UDV}^*)^T\mathbf{UDV}^*)^{-1}(\mathbf{UDV}^*)^T\mathbf{y} \\ &= (\mathbf{VDU}^*\mathbf{UDV}^*)^{-1}\mathbf{VDU}^*\mathbf{y} \\ &= \mathbf{VD}^{-1}\mathbf{U}^*\mathbf{UD}^{-1}\mathbf{V}^*\mathbf{VDU}^*\mathbf{y} \\ &= \mathbf{VD}^{-1}\mathbf{U}^*\mathbf{y}.\end{aligned} \tag{10}$$

Substituting Equation 10 back into Equation 8 and simplifying gives $\boldsymbol{\beta}_{RWA} = \boldsymbol{\Lambda}\boldsymbol{\beta} = \mathbf{VDV}^*\mathbf{VD}^{-1}\mathbf{U}^*\mathbf{y} = \mathbf{VU}^*\mathbf{y} = \mathbf{Z}^T\mathbf{y}$ (see J. Jonhson, 2000 for an alternate derivation). Thus, Equations 4 and 5 above can be simplified and expressed as:

$$\boldsymbol{\beta}_{RWA} = \mathbf{Z}^T\mathbf{y} \tag{11}$$

and

$$\boldsymbol{\Lambda} = \mathbf{Z}^T\mathbf{X}. \tag{12}$$

Then, the relative weights formula shown above in Equation 7 becomes:

$$\boldsymbol{\varepsilon}^2 = (\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{X})(\mathbf{Z}^T\mathbf{y} \circ \mathbf{Z}^T\mathbf{y}). \tag{13}$$

Equation 13 may be expressed in summation terms to calculate the i^{th} relative weight as:

$$\varepsilon_i^2 = \sum_{j=1}^{p} (\mathbf{Z}^T\mathbf{X})_{ij}^2 \left(\mathbf{Z}_j^T\mathbf{y}\right)^2 \tag{14}$$

where, as above, p is the number of predictors, $(\mathbf{Z}^T\mathbf{X})_{ij}$ is the ij^{th} element of $\mathbf{Z}^T\mathbf{X}$ and $\mathbf{Z}_j$ is the j^{th} column of $\mathbf{Z}$. Noting further that since both $(\mathbf{Z}^T\mathbf{X})_{ij}$ and $\mathbf{Z}_j^T\mathbf{y}$ are single or scalar values, the summation term may be simplified; the product of two squared scalar terms is simply the square of the scalar product itself. Then, Equation 14 becomes:

$$\varepsilon_i^2 = \sum_{j=1}^{p} \left[(\mathbf{Z}^T\mathbf{X})_{ij}\mathbf{Z}_j^T\mathbf{y}\right]^2 \tag{15}$$

As noted in the original relative weights analysis method (J. Johnson, 2000), the matrix $\boldsymbol{\Lambda}$ is symmetric requiring $\boldsymbol{\Lambda}^T = \boldsymbol{\Lambda}$. Since $\boldsymbol{\Lambda} = \mathbf{Z}^T\mathbf{X}$ (see Equation 12 above), then $(\mathbf{Z}^T\mathbf{X})^T = \mathbf{X}^T\mathbf{Z}$ must also be true. Clearly, then, $\mathbf{X}^T\mathbf{Z} = \mathbf{Z}^T\mathbf{X}$ must hold (see also J. Johnson, 2000 and Thomas et al. 2014). In the summation terms of Equation 15, the $\mathbf{X}^T\mathbf{Z} = \mathbf{Z}^T\mathbf{X}$ identity indicates that the *i* and *j*

subscripts of $(\mathbf{Z}^T\mathbf{X})_{ij}$ can be factored through in either order giving the equivalent algebraic identities:

$$\varepsilon_i^2 = \sum_{j=1}^{p}\left(\mathbf{Z}_i^T\mathbf{X}_j\mathbf{Z}_j^T\mathbf{y}\right)^2 = \sum_{j=1}^{p}\left(\mathbf{X}_j^T\mathbf{Z}_i\mathbf{Z}_j^T\mathbf{y}\right)^2 = \sum_{j=1}^{p}\left[\left(\sum_{k=1}^{n} z_{ki}x_{kj}\right)\left(\sum_{k=1}^{n} z_{kj}y_k\right)\right]^2 \quad (16)$$

and

$$\varepsilon_i^2 = \sum_{j=1}^{p}\left(\mathbf{Z}_j^T\mathbf{X}_i\mathbf{Z}_j^T\mathbf{y}\right)^2 = \sum_{j=1}^{p}\left(\mathbf{X}_i^T\mathbf{Z}_j\mathbf{Z}_j^T\mathbf{y}\right)^2 = \sum_{j=1}^{p}\left[\left(\sum_{k=1}^{n} x_{ki}z_{kj}\right)\left(\sum_{k=1}^{n} z_{kj}y_k\right)\right]^2. \quad (17)$$

Note that since $\mathbf{Z}_j^T\mathbf{X}_i$ and $\mathbf{Z}_i^T\mathbf{X}_j$ are scalars and that the transpose of the scalar is the scalar itself, $\mathbf{Z}_j^T\mathbf{X}_i = \left(\mathbf{Z}_j^T\mathbf{X}_i\right)^T = \mathbf{X}_i^T\mathbf{Z}_j$ and $\mathbf{Z}_i^T\mathbf{X}_j = \left(\mathbf{Z}_i^T\mathbf{X}_j\right)^T = \mathbf{X}_j^T\mathbf{Z}_i$ must be true. Consequently, Equations 16 and 17 give algebraically equivalent versions of Equation 15. Equation 17 states the i^{th} relative weight fixes the i^{th} column of $\mathbf{X}$ or that the i^{th} relative weight, much as with OLS regression, is calculated holding constant all remaining $p-1$ observed predictors (see Thomas et al., 2014 for a similar argument). Alternatively, Equation 16 states the i^{th} relative weight fixes the i^{th} column of $\mathbf{Z}$ while allowing $\mathbf{X}$ to vary (i.e., $\mathbf{X}$ is subscripted by j). Equation 16 associating the i^{th} relative weight with the unobserved $\mathbf{Z}$ is acceptable to the extent that interpreting latent regression parameters in structural equation modeling is acceptable (e.g., Mulaik, 2010). Unfortunately, however, Equation 16 also introduces an interpretational paradox: although $\mathbf{Z}$ is considered fixed for the $(\mathbf{Z}^T\mathbf{X})_{ij} = \mathbf{Z}_i^T\mathbf{X}_j$ term, the same $\mathbf{Z}$ is allowed to vary across the latent predictors in the $\mathbf{Z}_j^T\mathbf{y}$ term. Equation 16, that is, allows the latent $\mathbf{Z}$ variables to be both fixed and varying for a given relative weight, which provides an interpretational challenge (e.g., Thomas et al., 2014).

Equation 16's limitations are further emphasized when reframing the relative weights procedure as a graphical model, as is commonly done in the extant literature (e.g., Tonidandel et al., 2009). Figures 1-3 depict such models wherein Figures 1 and 2 correspond to Equation 16 and Figure 3 corresponds to Equation 17. Each panel in Figure 1 represents one iteration j for a hypothetical set of p predictors. As noted above, Figure 1 depicts that $\mathbf{Z}$ is simultaneously fixed with respect to $\mathbf{X}$ while also varying with respect to $\mathbf{y}$.

In graphical model terms, Equation 16 implies invalid path tracing rules in that one cannot trace from $\mathbf{y}$ to $\mathbf{X}$ through $\mathbf{Z}$; doing so would require the additional constraints depicted in Figure 2, where each dashed line represents a path constrained to unity. Alternatively, the process implied by Equation 17 and illustrated in Figure 3 depicts valid path traces from $\mathbf{y}$ to $\mathbf{X}$ through $\mathbf{Z}$ without additional constraints. Thus, only Equation 17 is interpretable. In the sections that follow, I offer two such interpretations related to principal components analysis and generalized least squares regression.

Interpreting Relative Weights via the New Formulae

Relative Weights and Principal Components

Recall from above the standard multiple regression problem of minimizing $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$, which may be equivalently expressed as $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$. Both the observed predictor scores $\mathbf{X}$ and the outcome vector $\mathbf{y}$ are assumed known, measured, or observed. On the other hand, the vector $\boldsymbol{\beta}$ is considered unknown and must be estimated from the data itself. The single best estimate of this quantity, as shown above, is $\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. In more statistical parlance, while the vector $\boldsymbol{\beta}$ is a random variable, the set of predictor scores $\mathbf{X}$ is a fixed, known quantity.

Likewise, least squares regression-based factor analytic methods (e.g., principal components) assume $\mathbf{X}$ is fixed, and that the unknown quantity is the set of weights or loadings through which said scores combine linearly. Fixing each observed predictor when calculating the i^{th} relative weight (see Equation 17 and Figure 3) suggests a relationship between the relative weights procedure and component-based regression methods (e.g., principal components regression). Indeed, Equation 3, as described further below, depicts the relationship precisely.

Recall that in order for the Equation 2 identity $\mathbf{Z} = \mathbf{U}\mathbf{V}^*$ to be valid, Equation 3 must also hold, which states $\mathbf{Z} = \mathbf{X}\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^*$ or equivalently that $\mathbf{Z} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^*\mathbf{X}$. In terms of either the singular value decomposition or the eigenvalue decomposition, the quantity $\mathbf{X}\mathbf{V}$ is the set of p unstandardized principal components (Hastie et al., 2009, p. 66). Then, $\mathbf{X}\mathbf{V}\mathbf{D}^{-1}$ is the standardized set of all p principal components in that the unstandardized $\mathbf{X}\mathbf{V}$ becomes standardized by the set of p singular values (i.e., $\mathbf{D}^{-1}$).

In terms of the original singular value decomposition shown in Equation 1, and using the same reasoning from which Equation 3 was derived, the standardized principal components are arrayed in $\mathbf{U}$ in that if $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^*$, then $\mathbf{X}\mathbf{V}\mathbf{D}^{-1} = \mathbf{U}$ (see also Hastie et al., 2009, p. 64-67). Thus, the relative weights definition $\mathbf{Z} = \mathbf{X}\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^* = \mathbf{U}\mathbf{V}^*$ implies $\mathbf{Z}$ is the set of p standardized principal component scores (i.e., $\mathbf{X}\mathbf{V}\mathbf{D}^{-1} = \mathbf{U}$) transformed by $\mathbf{V}^*$. Further, since $\mathbf{V}^*$ is a rotation matrix (Golub & Van Loan, 2013; Horn & C. Johnson, 2013), the transformation $\mathbf{V}^*$ performs on $\mathbf{X}\mathbf{V}\mathbf{D}^{-1} = \mathbf{U}$ is rotating the latent standardized principal components (e.g., Mulaik, 2010; see also, Thomas et al., 2014, p. 332-333).

If $\mathbf{Z}$ is the rotated set of p standardized principal components, $\mathbf{Z}^T\mathbf{X}$ becomes the principal components structure coefficients matrix, which is a transformed factor or component loading matrix (Mulaik, 2010). Then, as shown in Equations 14 and 17, $(\mathbf{Z}^T\mathbf{X})_{ij}^2 = \left(\mathbf{Z}_j^T\mathbf{X}_i\right)^2 = \left(\mathbf{X}_i^T\mathbf{Z}_j\right)^2$ represents the squared loading mapping each latent Z onto the set observed predictors $\mathbf{X}$. Further, squaring each $\mathbf{X}^T\mathbf{Z}$ term, summing the resulting columns, and dividing each quantity by one less than the number of cases produces the $1 \times p$ unit vector (J. Johnson, 2000; Thomas et al. 2014). More formally, $\mathbf{1}^T(\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{X})(N-1)^{-1} = \mathbf{1}^T$ where $\mathbf{1} = \left[1, 1, 1, \ldots, 1_p\right]^T$ is the $1 \times p$ unit vector.

Since $\mathbf{X}^T\mathbf{Z}$ is a component-loading matrix, squared and summed $\mathbf{X}^T\mathbf{Z}$ columns equaling unity is analogous to squared and summed principal component loadings equaling unity. In factor

analytic terms, summing the squared principal component loadings gives the observed variables' communalities (Mulaik, 2010). Whereas extracting fewer than p principal components will give less than unit communalities, extracting the same number of principal components as observed variables necessarily requires unit communalities. In other words, p principal components must explain p observed variables perfectly (Mulaik, 2010). Thus, calculating $\mathbf{Z} = \mathbf{XVD}^{-1}\mathbf{V}^{*} = \mathbf{UV}^{*}$ is analogous to performing a principal components analysis, extracting p components from the p observed predictors, rotating said p components by $\mathbf{V}^{*}$, and calculating the principal component scores for said p rotated principal components.

Taken together, the results above suggest the following interpretations for $\boldsymbol{\beta}_{RWA} = \mathbf{Z}^{T}\mathbf{y}$ and $\boldsymbol{\Lambda} = \mathbf{Z}^{T}\mathbf{X} = \mathbf{X}^{T}\mathbf{Z}$: the vector $\boldsymbol{\beta}_{RWA}$ is proportional to the vector of correlations between the p rotated principal components of $\mathbf{X}$ and the outcome $\mathbf{y}$; the matrix $\boldsymbol{\Lambda}$ is proportional to the correlation matrix between the p rotated principal components of $\mathbf{X}$ and the original $\mathbf{X}$ variables. Thus, the vector of relative weights $\boldsymbol{\varepsilon}^{2}$ is a squared combination of the p principal components' correlations with the original predictor matrix $\mathbf{X}$ and the original outcome matrix $\mathbf{y}$.

Relative Weights and Generalized Least Squares

The method above also demonstrates a novel relationship between the relative weights' parameters and the OLS regression parameters. Namely, combining Equations 8 and 9 demonstrates that $\boldsymbol{\beta}_{RWA} = \boldsymbol{\Lambda}\boldsymbol{\beta}$, or that the relative weights parameter vector $\boldsymbol{\beta}_{RWA}$ is a linear combination of the OLS regression parameter vector $\boldsymbol{\beta}$. Such a relationship becomes intuitive when recalling that $\mathbf{Z}$ is a linear combination of the original $\mathbf{X}$. Noting, further, that the method of generalized least squares (GLS) is a linear transformation of OLS regression then suggests a relationship between relative weights and GLS. Indeed, as shown below, this is precisely the case.

Recall the original OLS problem of minimizing $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ by finding an appropriate parameter vector $\boldsymbol{\beta}$. The objective is to solve for $\boldsymbol{\beta}$ in the set of linear equations $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$. Let $\mathbf{W}$ be an $N \times N$ weight matrix constituting a linear transformation of $\mathbf{X}$ and $\mathbf{y}$. Pre-multiplying both sides of the equation by this weight matrix gives $\mathbf{WX}\boldsymbol{\beta} = \mathbf{Wy}$. Pre-multiplying both sides of this equation by $\mathbf{X}^{T}$ then gives $\mathbf{X}^{T}\mathbf{WX}\boldsymbol{\beta} = \mathbf{X}^{T}\mathbf{Wy}$, which is a weighted version of the original OLS equation $\mathbf{X}^{T}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^{T}\mathbf{y}$. The solution to this weighted regression problem is $\boldsymbol{\beta}_{GLS} = (\mathbf{X}^{T}\mathbf{WX})^{-1}\mathbf{X}^{T}\mathbf{Wy}$, the vector of GLS regression coefficients (i.e., $\boldsymbol{\beta}_{GLS}$ is a linear transformation of $\boldsymbol{\beta}$). Next, let the GLS predicted value be $\hat{\mathbf{y}}_{GLS} = \mathbf{WX}\boldsymbol{\beta}_{GLS}$ where, as above, $\mathbf{W}$ is a weight matrix linearly transforming $\mathbf{X}$.

The correlation between the original predictors in $\mathbf{X}$ and the GLS predicted values would then be $\mathbf{X}^{T}\hat{\mathbf{y}}_{GLS} = \mathbf{X}^{T}\mathbf{WX}\boldsymbol{\beta}_{GLS}$. Substituting the GLS estimator noted above and simplifying then gives $\mathbf{X}^{T}\hat{\mathbf{y}}_{GLS} = \mathbf{X}^{T}\mathbf{WX}(\mathbf{X}^{T}\mathbf{WX})^{-1}\mathbf{X}^{T}\mathbf{Wy} = \mathbf{X}^{T}\mathbf{Wy}$. In other words, the correlation between the original $\mathbf{X}$ and the GLS predicted values $\hat{\mathbf{y}}_{GLS}$ is a weighted correlation between the original predictors and outcome. In relative weights terms, let $\mathbf{W}_j = \mathbf{Z}_j\mathbf{Z}_j^{T}$. Then Equation 17 becomes $\varepsilon_i^2 = \sum_{j=1}^{p}\left(\mathbf{X}_i^{T}\mathbf{W}_j\mathbf{y}\right)^2$, which is related to the GLS identity $\mathbf{X}^{T}\hat{\mathbf{y}}_{GLS} = \mathbf{X}^{T}\mathbf{Wy}$. Relative weights appear to be a squared-then-summed set of weighted correlations between the original predictors and outcome; alternatively, relative weights seemingly estimate the squared correlation between

the original $\mathbf{X}$ and the $\hat{\mathbf{y}}_{GLS}$ for a particular set of weight matrices constructed from $\mathbf{Z}$. To make the GLS and principal components interpretations above more tractable, it is helpful to translate Equations 13 and 17 computationally, which is accomplished below using the R statistical programming environment (R Core Team, 2018).

Implementing the New Formulae Computationally

Tables 1-3 provide the means through which the new formulae may be implemented computationally. Table 1 describes the new relative weights method in natural language. Tables 2 and 3 provide R (R Core Team, 2018) vectorized and iterative code for implementing Equation 13 and Equation 17, respectively.

Although conceptually similar to the relative weights matrix estimation method shown above in Equation 13, the vectorized Table 2 code simplifies calculation even further by noting that $\varepsilon^2 = (\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{X})(\mathbf{Z}^T\mathbf{y} \circ \mathbf{Z}^T\mathbf{y})$ may equivalently be written as $\boldsymbol{\varepsilon}^2 = [\mathbf{1}^T(\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T)]^T = [\mathbf{1}^T(\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T \circ \mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T)]^T$ where, as above, $\mathbf{1} = \left[1, 1, 1, \ldots, 1_p\right]^T$. The term $\mathbf{Z}^T\mathbf{y}\mathbf{1}^T$ serves to create a $p \times p$ matrix where each column is a replicate of the $\mathbf{Z}^T\mathbf{y} = \boldsymbol{\beta}_{RWA}$ vector, which is indicated in the Table 2 R code on line (18) by zty ← array(zt %*% yd, dim = c(p, p)). Once created, the $p \times p$ matrix, zty, can be elementwise multiplied by ztx, which is indicated in the R code by ztx * zty (see line (20) in Table 2). Finally, note that in R the square operator (denoted as Q**2 or Q^2) applied to a matrix indicates that each element of the matrix is squared. In other words, Q**2 is equivalent to Q * Q and $\mathbf{Q} \circ \mathbf{Q}$ rather than Q %*% Q and $\mathbf{QQ}$. Thus, (ztx * zty)**2 is equivalent to ztx * zty * ztx * zty, which corresponds to the matrix representation of $\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T \circ \mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T$ noted above. Finally, the R function colSums() iterates through each column of a matrix and sums all of the values of said column, which serves the same purpose as the vector $\mathbf{1}^T$ when pre-multiplying a given matrix. Thus, the Table 2 R code colSums((ztx * zty)**2) shown on line (20) is equivalent to the matrix representation $\boldsymbol{\varepsilon}^2 = [\mathbf{1}^T(\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T)]^T = [\mathbf{1}^T(\mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T \circ \mathbf{Z}^T\mathbf{X} \circ \mathbf{Z}^T\mathbf{y}\mathbf{1}^T)]^T$**.**

The iterative R code version presented in Table 3 will produce the same results as the vectorized version having the advantage of more explicitly stating the steps involved in Equation 17. Note that the iterated version in Table 3 is identical to the vectorized version in Table 2 up to line (15) where the Z matrix is created. The iterative version continues on line (16) by creating a $p \times 1$ vector named rws in which the calculated relative weights will eventually be stored. The first for() loop on line (18) with subscripts i iterates p times because p relative weights will ultimately be calculated. The second for() loop on line (20) with subscripts j corresponds to the outermost summation term in Equation 17. The third and final for() loop on line (23) with subscripts k is an iterative substitute for the $\mathbf{X}_i^T\mathbf{Z}_j\mathbf{Z}_j^T\mathbf{y}$ matrix operations, which is captured by the $\left(\sum_{k=1}^{n} x_{ki}z_{kj}\right)\left(\sum_{k=1}^{n} z_{kj}y_k\right)$ terms in Equation 17. More specifically, summing across all N ks in ls ← ls + xd[k,i] * z[k,j] on line (24) is equivalent to $\mathbf{X}_i^T\mathbf{Z}_j = \left(\sum_{k=1}^{n} x_{ki}z_{kj}\right)$. Likewise, summing across all N ks in bs ← bs + z[k,j] * yd[k] on line (25) is equivalent to $\mathbf{Z}_j^T\mathbf{y} = \left(\sum_{k=1}^{n} z_{kj}y_k\right)$.

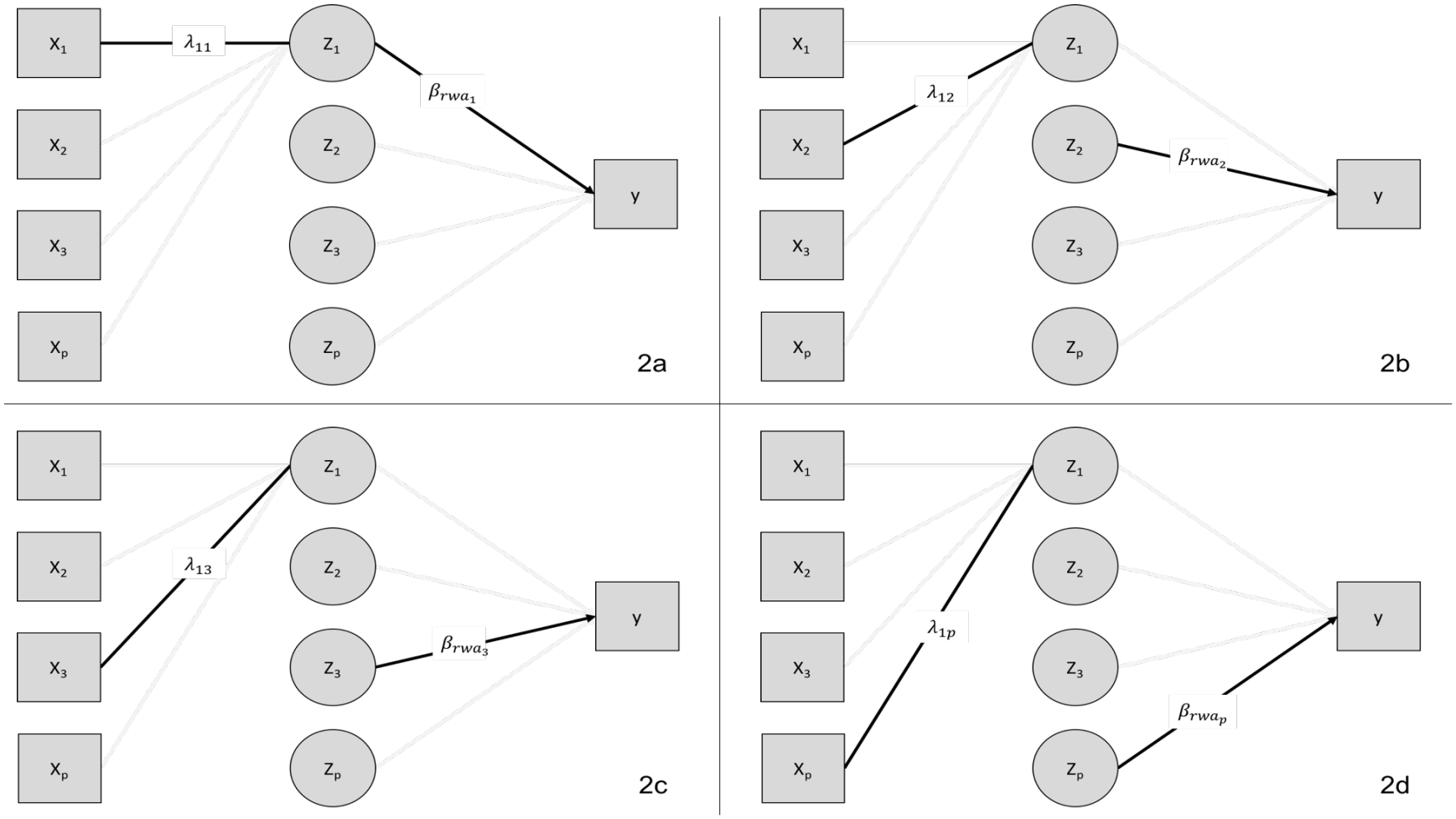

Figure 1. Depiction of actual algorithmic steps from Equation 16. In line with Equation 16, the first lambda subscript refers to the i^{th} variable (i.e., $\mathbf{Z}_i$) associated with the i^{th} relative weight. The second subscript refers to the varying j remaining predictors. All beta subscripts from $\mathbf{Z}$ to the outcome indicate the j^{th} element of $\boldsymbol{\beta}_{RWA} = \mathbf{Z}^T\mathbf{y}$. The paths from latent $\mathbf{Z}$ to observed $\mathbf{X}$ are purposefully undirected given the direction of said paths is an ongoing point of discussion (e.g., Thomas et al., 2014).

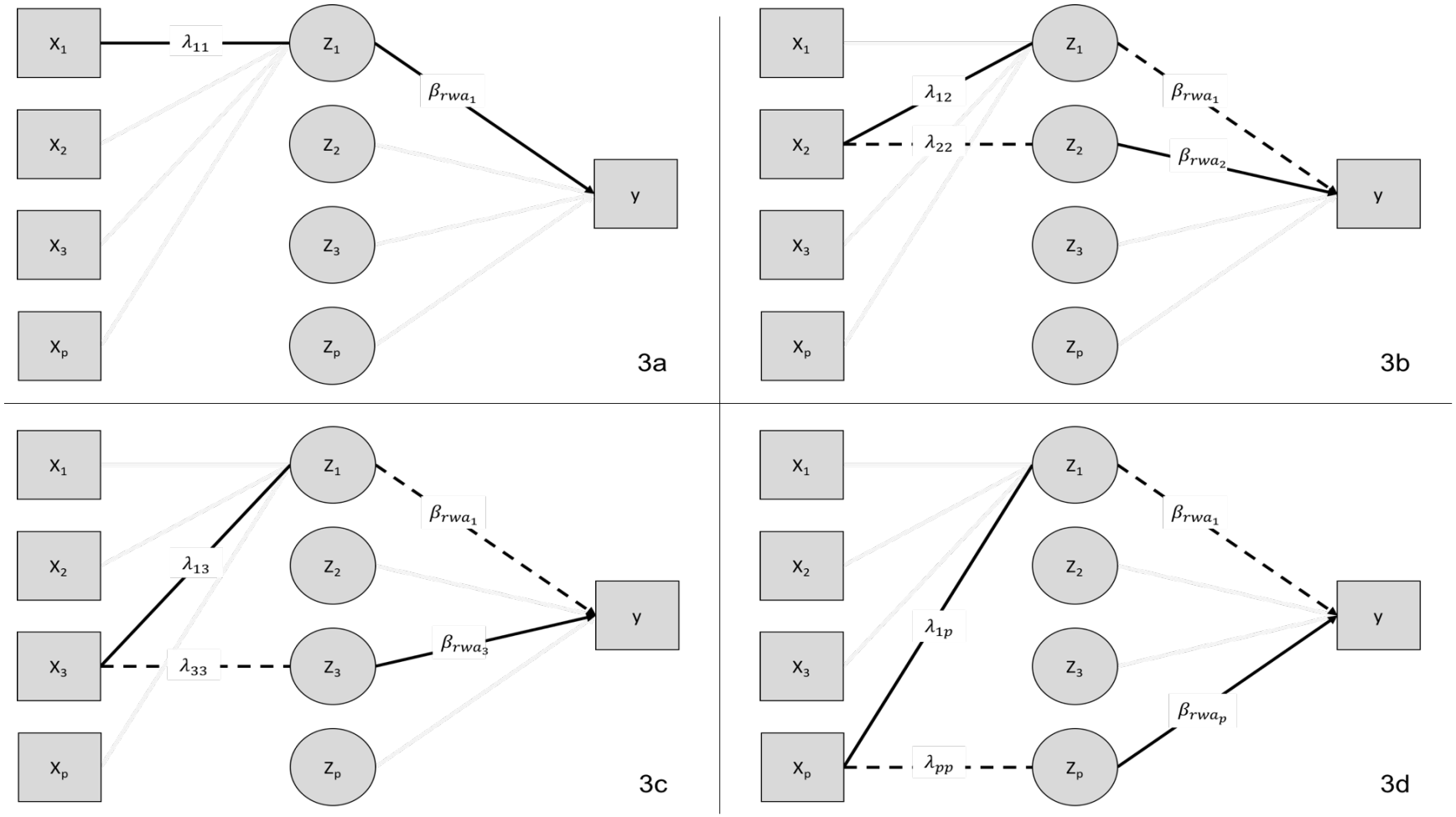

Figure 2. Depiction of implied algorithmic steps from Equation 16. In line with Equation 16, the first lambda subscript refers to the i^{th} variable (i.e., $\mathbf{Z}_i$) associated with the i^{th} relative weight. The second subscript refers to the varying j remaining predictors. All beta subscripts from $\mathbf{Z}$ to the outcome indicate the j^{th} element of $\boldsymbol{\beta}_{RWA} = \mathbf{Z}^T\mathbf{y}$. Black dashed lines indicate paths implicitly constrained to 1.0 for path tracing rules to be valid. The paths from latent $\mathbf{Z}$ to observed $\mathbf{X}$ are purposefully undirected given the direction of said paths is an ongoing point of discussion (e.g., Thomas et al., 2014).

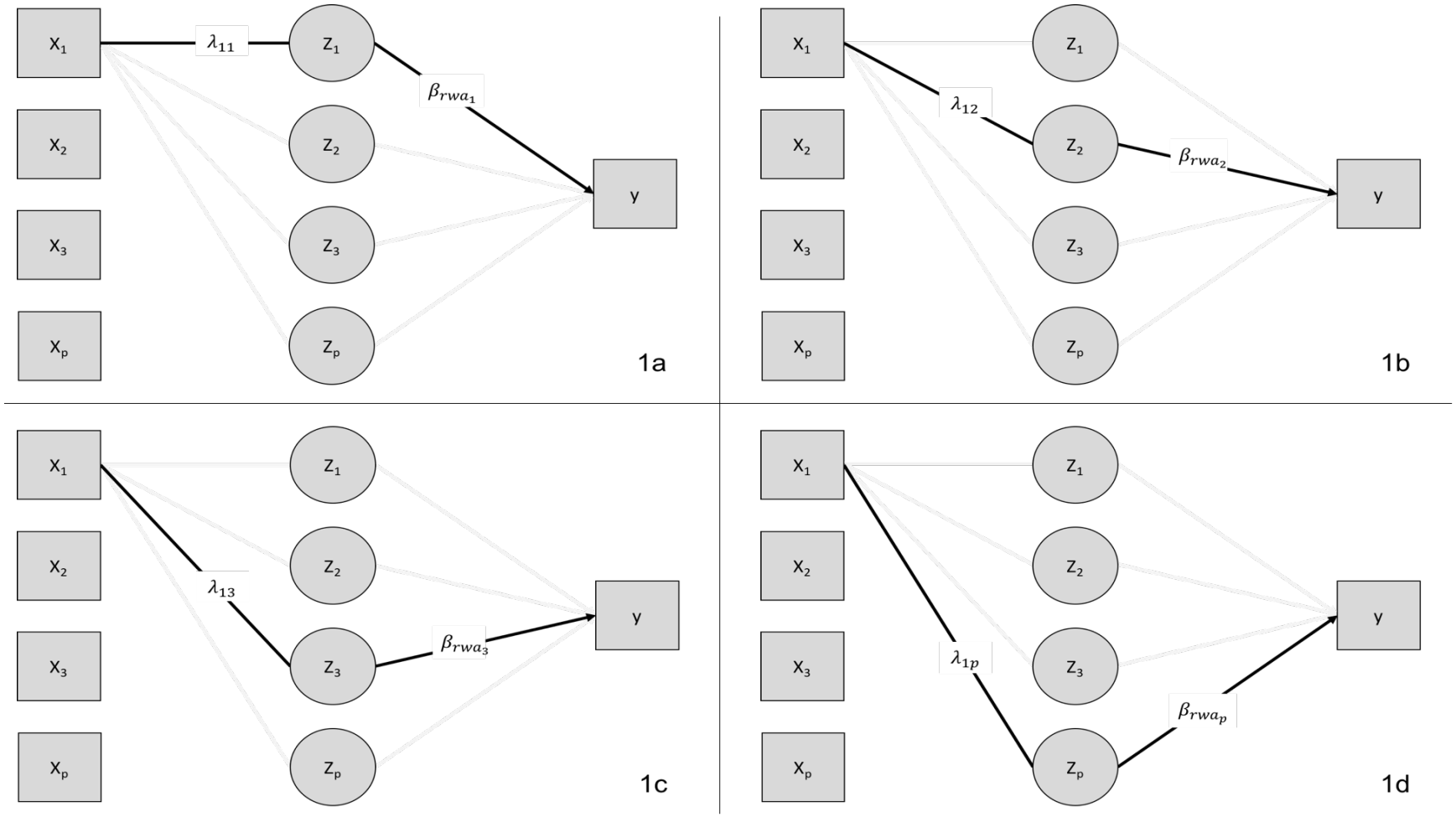

Figure 3. Depiction of algorithmic steps specified in Equation 17. In line with Equation 17, the first lambda subscript refers to the i^{th} variable (i.e., $\mathbf{X}_i$) associated with the i^{th} relative weight, and the second subscript refers to the varying j remaining predictors. All beta subscripts from $\mathbf{Z}$ to the outcome indicate the j^{th} element of $\boldsymbol{\beta}_{RWA} = \mathbf{Z}^T\mathbf{y}$. The paths from latent $\mathbf{Z}$ to observed $\mathbf{X}$ are purposefully undirected given the direction of said paths is an ongoing point of discussion (e.g., Thomas et al., 2014)

Table 1. Relative weights analysis algorithm

1. Standardize all data to have zero mean and unit norm / variance.
2. Separate data into design matrix of predictor scores **X** and the vector of outcome scores **y**.
3. Store the number of predictors (i.e., number of **X** columns) as a new variable p, and store the degrees of freedom (i.e., one less the number of rows in **X**) as a new variable c.
4. Perform and store a *singular value decomposition* (SVD) on **X** extracting only p left singular vectors (i.e., extracting only p eigenvectors of $\mathbf{X}\mathbf{X}^T$).
5. Store the SVD left singular vectors as new matrix **U**, and store the SVD right singular vectors as new matrix **V**.
6. Calculate and store new matrix $\mathbf{Z}^T = (\mathbf{U}\mathbf{V}^T)^T = \mathbf{V}\mathbf{U}^T$.
7. Calculate and store new matrix $\mathbf{\Lambda} = \mathbf{Z}^T\mathbf{X}$ and new vector $\boldsymbol{\beta}_{RWA} = \mathbf{Z}^T\mathbf{y}$.
8. For all $j \in \{1, 2, \ldots, p\}$ columns of $= \mathbf{Z}^T\mathbf{X}$, calculate the i^{th} raw relative weight as the squared product between the ij^{th} element of $\mathbf{\Lambda}$ and the j^{th} element of $\boldsymbol{\beta}_{RWA}$ summing across all j (see Equation 17).
9. Calculate the i^{th} standardized relative weight ε_i by dividing the i^{th} raw relative weight by the squared degrees of freedom c^2.

Table 2. Vectorized relative weights analysis code for R

```
(1)   rwa_vec <- function(dat, xs, ys){
(2)
(3)       d <- scale(dat)            # Step 1 Table 1
(4)       xd <- d[,xs]               # Step 2 Table 1
(5)       yd <- d[,ys]               # Step 2 Table 1
(6)
(7)       p <- ncol(xd)              # Step 3 Table 1
(8)       n <- nrow(dat)             # Step 3 Table 1
(9)       c <- n - 1L                # Step 3 Table 1
(10)
(11)      svdx <- svd(xd, nu = p)    # Step 4 Table 1
(12)      u <- svdx$u                # Step 5 Table 1
(13)      v <- svdx$v                # Step 5 Table 1
(14)
(15)      zt <- v %*% t(u)           # Step 6 Table 1
(16)
(17)      ztx <- zt %*% xd                        # Step 7 Table 1
(18)      zty <- array(zt %*% yd, dim = c(p, p))  # Step 7 Table 1
(19)
(20)      rw <- colSums((ztx * zty)**2) / c**2    # Steps 8-9 Table 1
(21)      res <- matrix(rw, p, 1, T, list(xs, "rw"))
(22)
(23)      return(res)
(24)  }
```

An additional advantage of the iterative code version shown in Table 3 is that such code can be implemented across a wide range of computing languages (e.g., Fortran or C++). The method proposed herein offers additional advantages beyond the improved statistical interpretations offered above. Most notably, the method proposed above avoids matrix inversion and is better suited for modern missing data methods such as full information maximum likelihood, both of which are described further below.

Table 3. Iterative relative weights analysis code for R

```
(1)   rwa_iter <- function(dat, xs, ys){
(2)
(3)       d <- scale(dat)            # Step 1 Table 1
(4)       xd <- d[,xs]               # Step 2 Table 1
(5)       yd <- d[,ys]               # Step 2 Table 1
(6)
(7)       p <- ncol(xd)              # Step 3 Table 1
(8)       n <- nrow(dat)             # Step 3 Table 1
(9)       c <- n - 1L                # Step 3 Table 1
(10)
(11)      svdx <- svd(xd, nu = p)    # Step 4 Table 1
(12)      u <- svdx$u                # Step 5 Table 1
(13)      v <- svdx$v                # Step 5 Table 1
(14)
(15)      zt <- v %*% t(u)           # Step 6 Table 1
(16)      rws = matrix(0, p, 1, T, list(xs, "rw"))
(17)
(18)      for(i in 1:p) {                # Step 9 Table 1 begins
(19)        rw <- 0
(20)        for(j in 1:p){               # Step 8 Table 1 begins
(21)            ls <- 0
(22)            bs <- 0
(23)            for(k in 1:n){
(24)                ls <- ls + xd[k,i] * z[k,j]    # Analogous to X_i^T Z_j
(25)                bs <- bs + z[k,j] * yd[k]      # Analogous to  Z_j^T y
(26)            }
(27)            rw <- rw + (bs * ls)**2
(28)         }                           # Step 8 Table 1 ends
(29)         rws[i,] <- rw / c**2        # Step 9 Table 1 ends
(30)      }
(31)      return(rws)
(32)  }
```

Additional Advantages of the New Formulae

Avoiding Matrix Inversion

Although defined in terms of the singular value decomposition, the original relative weights method notes that such decomposition could, at the time, be computationally intensive, especially with large sample sizes and many predictors (see Appendix in J. Jonhson, 2000). As such, relative weights are, in practice, calculated using an eigenvalue decomposition of the predictor correlation matrix. More specifically, extant relative weights methods begin by decomposing $\mathbf{X}^T\mathbf{X}$ into the corresponding matrices of eigenvectors and eigenvalues. In the notation above, this eigenvalue decomposition is given by $\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^*$. As above, $\mathbf{V}$ is the matrix of eigenvectors corresponding to $\mathbf{X}^T\mathbf{X}$ and $\mathbf{D}^2 = \mathbf{D}\mathbf{D}$ where $\mathbf{D}$ is the square diagonal matrix of singular values.

After performing an eigenvalue decomposition, relative weights' computational routines take the square root of the diagonal elements of $\mathbf{D}^2$ to get $\mathbf{D}$; the $\mathbf{\Lambda}$ matrix is then constructed using Equation 9 as above (i.e., $\mathbf{\Lambda} = \mathbf{V}\mathbf{D}\mathbf{V}^*$). Consequently, this method requires all elements of $\mathbf{D}^2$ to be non-negative, which is equivalent to stating that $\mathbf{X}^T\mathbf{X}$ must be non-negative definite (e.g., Arndt, 2011; Golub & Van Loan, 2013; Horn & C. Johnson, 2013; Watkins, 2010). When $\mathbf{X}^T\mathbf{X}$ violates non-negative definitiveness, at least one element of $\mathbf{D}^2$ is negative. Thus calculating the square roots introduces complex numbers that extant relative weights procedures do not easily accommodate.

Assuming for now that $\mathbf{X}^T\mathbf{X}$ is non-negative definite and that $\mathbf{\Lambda} = \mathbf{V}\mathbf{D}\mathbf{V}^*$ can be calculated from the eigenvalue decomposition of $\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^*$, the extant relative weights routines continue by calculating $\boldsymbol{\beta}_{RWA}$. To do so, the $\mathbf{\Lambda}$ matrix is inverted and pre-multiplied by the vector of correlations between original predictors and the outcome (see also Appendix in J. Johnson, 2000). That is, $\boldsymbol{\beta}_{RWA} = \mathbf{\Lambda}^{-1}\mathbf{X}^T\mathbf{y}$. Using the matrix operations noted above, inverting $\mathbf{\Lambda}$ requires inverting $\mathbf{V}\mathbf{D}\mathbf{V}^*$, which is given as $(\mathbf{V}\mathbf{D}\mathbf{V}^*)^{-1} = (\mathbf{V}^*)^{-1}\mathbf{D}^{-1}\mathbf{V}^{-1}$. Recall that $\mathbf{V}$ is unitary, $\mathbf{V}^* = \mathbf{V}^{-1}$ and so this can be simplified as $(\mathbf{V}^*)^*\mathbf{D}^{-1}\mathbf{V}^* = \mathbf{V}\mathbf{D}^{-1}\mathbf{V}^*$.

Since $\mathbf{D}$ is diagonal, its inverse is simply the reciprocal of its diagonal elements. In other words, and as shown above, if the diagonal of $\mathbf{D}$ is $[\sigma_1, \ldots, \sigma_p]$, $\mathbf{D}^{-1}$ is a diagonal matrix with its nonzero elements equal to $[1/\sigma_1, \ldots, 1/\sigma_p]$. Clearly, a necessary condition for calculating said reciprocals is that all singular values be greater than zero lest the process attempt to divide by zero. Therefore, all diagonal elements of $\mathbf{D}$ must be strictly non-negative, which only occurs when the elements of $\mathbf{D}^2$ are strictly positive. Stated otherwise, the extant relative weights methods require $\mathbf{X}^T\mathbf{X}$ to be positive definite and will fail when this condition is not met. Note that Equation 17 above does not require matrix inversion and, thus, can be used even when $\mathbf{X}^T\mathbf{X}$ is non positive definite.[2]

[2] Whether relative weights should be used at all in these instances is worth discussing, but such discussions are beyond the scope of the current paper.

Potential for Handling Missing Data in the Predictors

The second advantage of the proposed method above is the potential for using missing data methods. Extant relative weights routines use the raw data to calculate the predictor correlation matrix, which is followed by an eigenvalue decomposition and the steps noted above. In calculating such correlations, researchers must decide how to handle missing data in the predictors, which is most often accomplished using listwise or pairwise deletion. The method above offers additional missing data procedures beyond listwise and pairwise deletion. One such approach might be multiple imputation where several **X** datasets are created with randomly imputed values for missing data. Relative weights could then be calculated for each of these datasets using Equations 3 and 17 and the results pooled into a final estimate.

An alternative, and perhaps more promising, approach is to employ full information maximum likelihood via the expectation maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). Using continuous integral calculus and correlations rather than the matrix terminology, Equation 17 can be re-written as $\varepsilon_i^2 = \int \left(r_{x_i y} \mid \mathbf{Z}\right)^2 d\mathbf{Z}$. In other words, Equation 17 indicates that the i^{th} relative weight is an expectation of the squared correlation between the i^{th} predictor and the outcome **y** conditional upon the j^{th} **Z** with **Z** ultimately marginalized out. Recall that the set of **Z** variables are not actually observed but rather are a latent, linear combinations of the original **X** variables. As such, Equation 17 and its continuous integral counterpart indicate that the objective of relative weights is to estimate the expectation of an observed correlation conditional upon a set of unobserved, latent variables, which can be calculated even in the presence of missing data in **X** (e.g., Dempster et al., 1977). To our knowledge, however, current relative weights estimation methods cannot easily incorporate such missing data methods.

Topics Warranting Caution and Additional Research

Although the new method proposed herein describes more precisely the relative weights analysis procedure, the exact meaning of relative weights remains somewhat ambiguous (Thomas et al., 2014). Other than summing to the squared multiple correlation, little is known about the meaning and properties of the relative weights themselves. Conversely, dominance analysis is a relatively well-defined method and produce easily interpretable weights (Grömping, 2007); dominance weights are the average incremental variability explained by a specific predictor across all possible regression sub-models (Azen & Budescu, 2003). Although such precise interpretation currently evades relative weights, the new method proposed herein does offer – through the concept of principal components and GLS – promising insights for guiding future such research. Such promise must nevertheless be couched in caution, as described further below.

Thomas et al. (2014) recently noted that the geometric rotations inherent to the singular value decomposition pose several interpretational difficulties for relative weights. The formulae described herein demonstrate that **Z** is the set of standardized principal components scores rotated by the matrix $\mathbf{V}^*$. Challenging, however, is that the matrix $\mathbf{V}^*$ simply arrays the set of eigenvectors corresponding to $\mathbf{X}^T\mathbf{X}$ and is, in general, not unique. Consider the general definition of an eigenvector **v** for a $p \times p$ matrix **A** as any $p \times 1$ vector satisfying the identity $\mathbf{Av} = \sigma_i^2\mathbf{v}$ where σ_i^2 is the squared singular value of the i^{th} variable or column in **A**. This identity can also be expressed as $(\mathbf{A} - \sigma_i^2\mathbf{I})\mathbf{v} = \mathbf{0}$ where **I** is the $p \times p$ identity matrix. In terms of the notation

above, the eigenvector matrix $\mathbf{V}$ is any set of vectors satisfying the identity $\mathbf{X}^T\mathbf{X}\mathbf{V} = \mathbf{D}^2\mathbf{V}$ or, equivalently, $(\mathbf{X}^T\mathbf{X} - \mathbf{D}^2)\mathbf{V} = \mathbf{0}$. Any matrix $\mathbf{V}$ multiplied by a scalar value α also satisfies this identity in that $(\mathbf{X}^T\mathbf{X} - \mathbf{D}^2)\mathbf{V}\alpha = \mathbf{0}$ can be expanded to $\alpha\mathbf{X}^T\mathbf{X}\mathbf{V} - \alpha\mathbf{D}^2\mathbf{V} = \mathbf{0}$ and re-arranging then gives $\alpha\mathbf{X}^T\mathbf{X}\mathbf{V} = \alpha\mathbf{D}^2\mathbf{V}$, which, after dividing both sides by the scalar α is clearly the original identity $\mathbf{X}^T\mathbf{X}\mathbf{V} = \mathbf{D}^2\mathbf{V}$. Thus, choosing $\mathbf{V}^*$ instead of any other $\alpha\mathbf{V}^*$ as the matrix by which to rotate the standardized principal components $\mathbf{U}$ is largely arbitrary.

It is worth noting that one argument for choosing $\mathbf{V}$ as a rotation matrix is that the resulting relative weights sum to the model squared multiple correlation (e.g., J. Johnson, 2000). As Thomas et al. (2014) note, however, any matrix whose elementwise-squared columns sum to unity will similarly decompose the squared multiple correlation (p. 333). More formally, the model squared multiple correlation will be additively decomposed substituting for $\mathbf{V}$ any matrix $\mathbf{A}$ that satisfies the identity $\mathbf{1}^T(\mathbf{A} \circ \mathbf{A}) = \mathbf{1}^T$ where, as above, $\mathbf{1} = \left[1, 1, 1, \ldots, 1_p\right]^T$. If $\mathbf{V}$ satisfies this identity, then clearly so does $-\mathbf{V}$ (i.e., the elementwise square removes the sign). Further, $-\mathbf{V}$ satisfies the requirements of an eigenvector of $\mathbf{X}^T\mathbf{X}$ by setting the eigenvector multiplier above to negative unity, or $\alpha = -1$. Thus, even if researchers wish to additively decompose the squared multiple correlation, the choice between $\mathbf{V}$ or $-\mathbf{V}$ as a principal component rotation matrix is arbitrary.

The general problem of non-unique rotation matrices is far from novel (e.g., Schönemann, 1966; Schönemann & Steiger, 1978). In fact, it is quite well known that *rotational indeterminacy* is addressed by specifying a desired pattern of item-factor loadings against which an empirical solution is compared (e.g., Mulaik, 2010). In other words, rotational indeterminacy is addressed by choosing a particular objective function against which the estimated solutions will be compared. Researchers may currently choose among several, well-researched factor rotation objective functions, such as oblimin, quartimin, varimax, and promax (see, e.g., Mulaik, 2010). Currently, however, the relative weights procedure offers no such guidance for choosing an appropriate rotation matrix. Future relative weights research would benefit from comparing several such rotational matrix objective functions, particularly if relying on the principal components-based relative weights interpretation presented above.

Another important caveat to the principal components interpretation presented above is that the p components are extracted from the p predictor scale scores where, presumably, a set of observed items have been aggregated to a single score. In such instances, the relative weights procedure implies a hierarchical or second order principal components model where the first order model is a unit weighted composite. The relative weights procedure then performs a principal components analysis on these p unit weighted composite variables extracting p principal components that must perfectly explain the first order weighted composites. Thus, the full measurement model implied by the relative weights procedure is a second order model. Future research should explore the implications of assuming such a measurement structure when performing the relative weights procedure (see J. Johnson, 2004 for one such example).

Conclusion

Relative importance methods offer attractive means of interpreting multiple regression results within an effect size context (LeBreton, Hargis, Griepentrog, Oswald, & Ployhart, 2007). Arguably, the two most popular such methods are relative weights and dominance analysis. Although the statistical properties of dominance weights are relatively better known (Grömping, 2007), the method can prove computationally intensive requiring $2^p - 1$ different multiple regressions. Although relative weights were designed to be computationally simpler, the specific matrix decompositions implemented obfuscate the method's interpretation (Thomas et al., 2014). Thus, one must often choose between computational simplicity (i.e., relative weights) and theoretical interpretability (i.e., dominance weights). Equation 17 and the new relative weights method proposed herein, combined with the transparency of the R code provided, however, offer novel insights increasing relative weights' theoretical interpretability while remaining computationally efficient.

References

Arndt, J. (2011). *Matters computational*. Springer-Verlag.

Azen, R., & Budescu, D. V. (2003). The dominance analysis approach for comparing predictors in multiple regression. *Psychological Methods, 8*(2), 129-148. doi: 10.1037/1082-989X.8.2.129

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics: Identifying influential data and sources of collinearity.* John Wiley & Sons.

Braun, M. T., & Oswald, F. L. (2011). Exploratory regression analysis: A tool for selecting models and determining predictor importance. *Behavior Research Methods, 43*(2), 331-339. doi: 10.3758/s13428-010-0046-8

Brooks, M. E., Dalal, D. K., & Nolan, K. P. (2014). Are common language effect sizes easier to understand than traditional effect sizes? *Journal of Applied Psychology, 99*(2), 332-340. doi: 10.1037/a0034745

Budescu, D. V. (1993). Dominance analysis: A new approach to the problem of relative importance of predictors in multiple regression. *Psychological Bulletin, 114*(3), 542-551. doi: 10.1037/0033-2909.114.3.542

Cumming, G. (2014). The new statistics: Why and how. *Psychological Science, 25*(1), 7-29. doi: 10.1177/0956797613504966

Cumming, G., & Finch, S. (2005). Inference by eye: Confidence intervals and how to read pictures of data. *American Psychologist, 60*(2), 170-180. 10.1037/0003-066X.60.2.170

Dempster, A. P, Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1), 1-38. doi: 10.1111/j.2517-6161.1977.tb01600.x

Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations*. The Johns Hopkins University Press.

Grömping, U. (2007). Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician, 61*(2), 139-147. doi: 10.1198/000313007X188252

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer Science & Business Media.

Horn, R. A., & Johnson, C. R. (2013). *Matrix analysis.* Cambridge University Press.

Johnson, J. W. (2000). A heuristic method for estimating the relative weight of predictor variables in multiple regression. *Multivariate Behavioral Research, 35*(1), 1-19. doi: 10.1207/S15327906MBR3501_1

Johnson, J. W. (2004). Factors affecting relative weights: The influence of sampling and measurement error. *Organizational Research Methods, 7*(3), 283-299. doi: 10.1177/1094428104266018

Johnson, J. W., & LeBreton, J. M. (2004). History and use of relative importance indices in organizational research. *Organizational Research Methods, 7*(3), 238-257. doi/10.1177/1094428104266510

Kelly, K., & Preacher, K. J. (2012). On effect size. *Psychological Methods, 17*(2), 137-152. doi: 10.1037/a0028086

LeBreton, J. M., Hargis, M. B., Griepentrog, B., Oswald, F. L., & Ployhart, R. (2007). A multidimensional approach for evaluating variables in organizational research and practice. *Personnel Psychology, 60*(2), 475-498. doi: 10.1111/j.1744-6570.2007.00080.x

LeBreton, J. M., Ployhart, R. E., & Ladd, R. T. (2004). A Monte Carlo comparison of relative importance methodologies. *Organizational Research Methods, 7*(3), 258-282. doi: 10.1177/1094428104266017

Mulaik, S. A. (2010). *Foundations of factor analysis* (2nd ed.). Chapman & Hall/CRC.

Namboodiri, K. (1984). *Matrix algebra: An introduction.* Sage Publications.

Nimon, K., Henson, R. K., & Gates, M. S. (2010). Revisiting interpretation of canonical correlation analysis: A tutorial and demonstration of canonical commonality analysis. *Multivariate Behavioral Research, 45*(4), 702-724. doi: 10.1080/00273171.2010.498293

Nimon, K., Lewis, M., Kane, R., & Haynes, R. M. (2008). An R package to compute commonality coefficients in the multiple regression case: An introduction to the package and a practical example. *Behavior Research Methods, 40*(2), 457-466. doi: 10.3758/BRM.40.2.457

Nimon, K., & Oswald, F. L. (2013). Understanding the results of multiple linear regression: Beyond standardized regression coefficients. *Organizational Research Methods, 16*(4), 650-674. doi: 10.1177/1094428113493929

R Core Team (2018). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Schoen, J. L., DeSimone, J. A., & James, L. R. (2010). Exploring joint variance between independent variable and a criterion: Meaning, effect, and size. *Organizational Research Methods, 14*(4), 674-695. doi: 10.1177/1094428110381787

Schönemann, P. H. (1966). A generalized solution of the orthogonal Procrustes problem. *Psychometrika, 31*(1), 1-10. doi: 10.1007/BF02289451

Thomas, D. R., Zumbo, B. D., Kwan, E., & Schweitzer, L. (2014). On Johnson's (2000) relative weights method for assessing variable importance: A reanalysis. *Multivariate Behavioral Research, 49*(4), 329-338. doi: 10.1080/00273171.2014.905766

Tonidandel, S., & LeBreton, J. M. (2015). RWA web: A free, comprehensive, web-based, and user-friendly tool for relative weights analysis. *Journal of Business and Psychology, 30*(2), 207-216. doi: 10.1007/s10869-014-9351-z

Tonidandel, S., LeBreton, J. M., & Johnson, J. W. (2009). Determining the statistical significance of relative weights. *Psychological Methods, 14*(4), 387-399. doi: 10.1037/a0017735.

Watkins, D. S. (2010). *Fundamentals of matrix computations*. John Wiley & Sons.

Wright, S. (1921). Correlation and causation. *Journal of Agricultural Research, 20*(7), 557-585.

Wright, S. (1934). The method of path coefficients. *The Annals of Mathematical Statistics, 5*(3), 161-215.

Appendix

Table A1. Commented executable vectorized relative weights analysis code for R

```
rwa_vec <- function(dat, xs, ys){   # dat = data;
                                    # xs = predictor names character vector;
                                    # ys = outcome name in quotes;

   d <- scale(dat)      # Step 1: Standardize all data
   xd <- d[,xs]         # Step 2: Collect predictors into the design matrix
   yd <- d[,ys]         # Step 2: Collect outcome into vector

   p <- ncol(xd)        # Step 3: Store number of predictors
   n <- nrow(dat)       # Step 3: Store sample size
   c <- n - 1L          # Step 3: Store degrees of freedom

   svdx <- svd(xd, nu = p)    # Step 4: Perform singular value decomposition
   u <- svdx$u                # Step 5: Store left singular vectors
   v <- svdx$v                # Step 5: Store right singular vectors
   zt <- v %*% t(u)           # Step 6: Store transpose of transformed predictors

   ztx <- zt %*% xd                          # Step 7: Calculate lambda matrix
   zty <- array(zt %*% yd, dim = c(p, p))    # Step 7: Calculate Beta vector
   rw <- colSums((ztx * zty)**2) / c**2      # Steps 8-9: Calculate epsilon vector

   res <- matrix(rw, nrow = p,               # Column vector for results
                     ncol = 1,
                     byrow = T,
                     dimnames = list(xs, "rw"))

   return(res)
}
```

Table A2. Commented executable iterative relative weights analysis code for R

```
rwa_iter <- function(dat, xs, ys){  # dat = data;
                                    # xs = predictor names character vector;
                                    # ys = outcome name in quotes;

   d <- scale(dat)      # Step 1: Standardize all data
   xd <- d[,xs]         # Step 2: Collect predictors into the design matrix
   yd <- d[,ys]         # Step 2: Collect outcome into vector

   p <- ncol(xd)        # Step 3: Store number of predictors
   n <- nrow(dat)       # Step 3: Store sample size
   c <- n - 1L          # Step 3: Store degrees of freedom

   svdx <- svd(xd, nu = p)    # Step 4: Perform singular value decomposition
   u <- svdx$u                # Step 5: Store left singular vectors
   v <- svdx$v                # Step 5: Store right singular vectors
   zt <- v %*% t(u)           # Step 6: Store transpose of transformed predictors

   rws = matrix(0, nrow = p,  # Column vector to store results
                ncol = 1,
                dimnames = list(xs, "rw"))

   for(i in 1:p) {  # Loop through p times, once for each observed predictor
      rw <- 0           # Stores sum of relative weights across j Zs
      for(j in 1:p){       # Loop once for each transformed predictor in Z
           ls <- 0         # Stores sum of lambdas across k people
           bs <- 0         # Stores sum of betas across k people
           for(k in 1:n){
                ls <- ls + xd[k,i] * z[k,j]   # Same as (Xi)'Zj for each ij
                bs <- bs + z[k,j] * yd[k]     # Same as Z'y for each j
           }
           rw <- rw + (bs * ls)**2  # Mimics matrix products from Steps 7-8
      }
      rws[i,] <- rw / c**2          # Step 9 Table 1
   }
   return(rws)
}
```

Table A3. Simple executable vectorized relative weights analysis code for R

```
rwa_vec <- function(dat, xs, ys){
    d <- scale(dat); xd <- d[,xs]; yd <- d[,ys]
    p <- ncol(xd); n <- nrow(dat); c <- n - 1L
    svdx <- svd(xd, nu = p); u <- svdx$u; v <- svdx$v
    zt <- v %*% t(u); ztx <- zt %*% xd
    zty <- array(zt %*% yd, dim = c(p, p))
    rw <- colSums((ztx * zty)**2) / c**2
    res <- matrix(rw, p, 1, T, list(xs, "rw"))
    return(res)}
```

Table A4. Simple executable iterative relative weights analysis code for R

```
rwa_iter <- function(dat, xs, ys){
   d <- scale(dat); xd <- d[,xs]; yd <- d[,ys]
   p <- ncol(xd); n <- nrow(dat); c <- n - 1L
   svdx <- svd(xd, nu = p); u <- svdx$u; v <- svdx$v; z <- u %*% t(v)
   rws = matrix(0, p, 1, T, list(xs, "rw"))
   for(i in 1:p) {rw <- 0; for(j in 1:p){ls <- 0; bs <- 0
   for(k in 1:n){ls <- ls + xd[k,i] * z[k,j]; bs <- bs + z[k,j] * yd[k]};
   rw <- rw + (bs * ls)**2}; rws[i,] <- rw / c**2}
   return(rws)}
```