Factor Loading Recovery for Smoothed Non-positive Definite Correlation Matrices

Justin D. Kracht[1]

[1] University of Minnesota

Author Note

Correspondence concerning this article should be addressed to Justin D. Kracht, Department of Psychology, University of Minnesota, N218 Elliott Hall 75 East River Road, Minneapolis, MN 55455. E-mail: krach018@umn.edu

Abstract

Researchers commonly use tetrachoric correlation matrices in item factor analysis. Unfortunately, tetrachoric correlation matrices are often non-positive definite (NPD; i.e., having one or more negative eigenvalues). These NPD correlation matrices are problematic because the corresponding population correlation matrices they estimate are definitionally positive semidefinite (PSD; i.e., having strictly non-negative eigenvalues). Therefore, when used in procedures such as factor analysis, NPD tetrachoric correlation matrices may result in poor estimates of factor loadings. Matrix smoothing algorithms attempt to remedy this problem by finding a PSD correlation matrix that is close, in some sense, to a given NPD correlation matrix. However, little research has been done on the effectiveness of matrix smoothing. In the present simulation study, NPD tetrachoric correlation matrices were calculated from simulated binary data sets. Three matrix smoothing algorithms—the Higham (2002), Bentler-Yuan (2011), and Knol-Berger algorithms (1991)—were applied to the NPD tetrachoric correlation matrices. Factor analysis was then conducted on the smoothed and unsmoothed correlation matrices. The results show that smoothed matrices were slightly better estimates of their population counterparts compared to unsmoothed NPD correlation matrices. However, using smoothed compared to unsmoothed NPD correlation matrices for item factor analysis did not meaningfully improve factor loading recovery. Matrix smoothing should therefore be considered only as a tool to facilitate factor analysis of NPD correlation matrices and not as a statistical remedy for the root causes of matrix indefiniteness.

Keywords: matrix smoothing, item factor analysis, factor loading recovery, non-positive definite

Word count: 5881

Factor Loading Recovery for Smoothed Non-positive Definite Correlation Matrices

Tetrachoric correlation matrices (Olsson, 1979) are used to estimate the correlations between the normally-distributed, continuous latent variables assumed to underlie observed binary data. Therefore, tetrachoric correlation matrices are often recommended for use in item factor analysis because the common linear factor model requires the assumption that outcomes are continuous (Wirth & Edwards, 2007). Unfortunately, tetrachoric correlation matrices are frequently *non-positive definite* (NPD), having one or more negative eigenvalues (Bock, Gibbons, & Muraki, 1988; Wothke, 1993). NPD correlation matrices are problematic because proper correlation matrices are, by definition, positive semi-definite (PSD; i.e., having all eigenvalues greater than or equal to 0; Wothke, 1993). Although NPD correlation matrices resemble proper correlation matrices in many ways—they are symmetric, have unit diagonals, and all off-diagonal elements $|r_{ij}| \leq 1$—it is impossible to obtain an NPD matrix of Pearson correlations from complete data. Thus, NPD correlation matrices are improper estimates of their corresponding population correlation matrices in the sense that they are not included in the set of possible population correlation matrices.

Some researchers have suggested that one approach to resolving the problem of NPD tetrachoric correlation matrices is to obtain a PSD correlation matrix that can be reasonably substituted for an NPD tetrachroic correlation matrix (e.g., Devlin, Gnanadesikan, & Kettenring, 1975; Dong, 1985). This approach is often referred to as *matrix smoothing* and many algorithms developed for this purpose, referred to as *matrix smoothing algorithms* (or simply, *smoothing algorithms*), have been proposed in the psychometric literature and elsewhere (Bentler & Yuan, 2011; Devlin et al., 1975; Dong, 1985; Fushiki, 2009; Higham, 2002; Knol & Berger, 1991; Li, Li, & Qi, 2010; Lurie & Goldberg, 1998; Qi & Sun, 2006). However, despite the the frequent occurrence of NPD tetrachoric correlation matrices in psychometric research (Bock et al., 1988, p. 261), the variety of smoothing algorithms available, and suggestions to use matrix smoothing algorithms as a remedy to NPD tetrachoric correlation matrices (Bentler & Yuan, 2011; Knol & Berger, 1991; Wothke, 1993), scant research has been done on the effectiveness of matrix smoothing algorithms in the context of item factor analysis of NPD tetrachoric correlation matrices. In one of the only published comparisons of this kind, Knol and Berger (1991) investigated the effects of using smoothed compared to unsmoothed correlation matrices in factor analysis and found no large differences. However, this comparison was not a main focus of their study and only compared a small number of NPD matrices (10 NPD correlation matrices with 250 subjects and 15 items).

Moreover, few studies have compared the relative performance of matrix smoothing algorithms in the context of factor analysis (Debelak & Tran, 2013, 2016). Debelak and Tran (2013) conducted a simulation study to determine which of three matrix smoothing algorithms — the Higham Alternating Projections algorithm (APA; 2002), Bentler-Yuan algorithm (BY; 2011), and the Knol-Berger (KB; 1991) algorithm — most often recovered the underlying dimensionality when applied to NPD tetrachoric correlation matrices prior to parallel analysis (Horn, 1965). Debelak and Tran simulated binary data using a two-parameter logistic (2PL) item response theory (IRT; Birnbaum, 1968; de Ayala, 2013) model for one- and two-factor models with varying factor correlations, item difficulties, item discriminations, numbers of items, and numbers of subjects. Debelak and Tran then computed tetrachoric correlation matrices for each simulated binary data set. If a tetrachoric correlation matrix was NPD, the three aforementioned smoothing algorithms were applied (resulting in three smoothed correlation matrices in addition to the NPD tetrachoric matrix). Finally, Debelak and Tran conducted parallel analysis using each of these four correlation matrices to obtain estimates of dimensionality. Debelak and Tran concluded that "[the] application of smoothing algorithms generally improved correct identification of dimensionality when the correlation between the latent dimensions was 0.0 or 0.4 in our simulations" (Debelak & Tran, 2013, p. 74). With respect to the relative performance of the Higham, Bentler-Yuan, and Knol-Berger smoothing algorithms in this context, Debelak and Tran concluded that there were "minor differences in the performance of the three smoothing algorithms used in [the] study. In data sets with a clear dimensional structure...the algorithm of Bentler and Yuan (2011) performed best" (Debelak & Tran, 2013, p. 74).

Following on these results, Debelak and Tran (2016) extended their simulation study design to evaluate the relative and absolute effectiveness of matrix smoothing algorithms when applied to NPD polychoric correlation matrices of ordered, categorical (i.e., polytomous) data prior to conducting a parallel analysis. As in their previous study, Debelak and Tran used the accuracy of the parallel analysis dimensionality estimates (i.e., dimensionality recovery) as their evaluation criterion. In addition to extending their design to consider polytomous data, Debelak and Tran (2016) also considered factor models with either one or three major common factors and either zero or forty minor common factors. The minor common factors were meant to represent the effects of model approximation error; that is, the degree of model misfit inherent to mathematical models of natural phenomena in general, and psychological models in particular (MacCallum & Tucker, 1991; MacCallum, Widaman, Preacher, & Hong, 2001; Tucker, Koopman, & Linn, 1969). Debelak and Tran concluded that the analysis of smoothed polychoric correlation matrices generally gave more accurate results than the analysis of NPD polychoric correlation matrices. Moreover, they found that "methods based on the algorithms of Knol and Berger, Higham, and Bentler and Yuan showed a comparable performance with regard to the accuracy to detect the number of underlying major factors, with a slightly better performance of methods based on the Bentler and Yuan algorithm" (Debelak & Tran, 2016, p. 15).

Both Debelak and Tran (2013) and Debelak and Tran (2016) concluded that the Bentler-Yuan (2011) smoothing algorithm led to the most accurate results (in terms of dimesionality recovery) when applied to NPD tetrachoric or polychoric correlation matrices. However, neither study attempted to explain why the Bentler-Yuan algorithm led to better dimensionality recovery relative to the other smoothing methods they investigated. One intriguing possibility is that the smoothed correlation matrices produced by the Bentler-Yuan algorithm were better approximations of population correlation matrix than either the smoothed matrices produced by the Knol-Berger (1991) and Higham algorithms (2002), and also better approximations than the original NPD tetrachoric or polychoric correlation matrices. If this is true, one might also expect that Bentler-Yuan smoothed tetrachoric correlation matrices will also lead to more accurate factor loading estimates compared to the alternatives.

The purpose of the present study was to address two questions related to these hypotheses. First, are smoothed NPD tetrachoric correlation matrices better estimates of their corresponding population correlation matrices than the original NPD tetrachoric correlation matrices and, if so, which smoothing method produces the best estimates? Second, do smoothed NPD tetrachoric correlation matrices lead to better factor loading estimates compared to the unsmoothed tetrachoric matrices when used in exploratory factor analysis and, if so, which smoothing algorithm leads to the best factor loading estimates? To answer these questions, I conducted a simulation study in which I generated 124,346 NPD tetrachoric correlation matrices from a variety of realistic data scenarios. Before describing the simulation design, I first introduce the three matrix smoothing algorithms under investigation, the common factor model, and the three factor analysis algorithms included in this study.

Matrix Smoothing Algorithms

Higham Alternating Projections Algorithm (APA; 2002). The matrix smoothing algorithm proposed by Higham (2002) seeks to find the closest PSD correlation matrix to a given NPD correlation matrix. In this context, closeness is defined as the generalized Euclidean distance (Banerjee & Roy, 2014, p. 492). Higham's algorithm uses a series of alternating projections to locate the PSD correlation matrix ($\mathbf{R}_{\text{APA}}$) closest to a given NPD correlation matrix ($\mathbf{R}_{\text{NPD}}$) of the same order. The algorithm works by first projecting $\mathbf{R}_{\text{NPD}}$ onto the set of symmetric, PSD $p \times p$ matrices, $\mathcal{S}$. The resulting candidate matrix is then projected onto the set of symmetric $p \times p$ matrices with unit diagonals, $\mathcal{U}$. The series of projections repeats until the algorithm converges to a matrix, $\mathbf{R}_{\text{APA}}$, that is PSD, symmetric, and has a unit diagonal, or until the maximum number of iterations is exceeded.

Bentler-Yuan Algorithm (BY; 2011). Bentler and Yuan (2011) proposed a smoothing algorithm based on minimum-trace factor analysis (MTFA; Bentler, 1972; Jamshidian & Bentler, 1998). MTFA seeks to find optimal communality estimates such that unexplained common variance is minimized while constraining the diagonal matrix of unique variances and the observed covariance matrix with the estimated communalities as diagonal elements to be positive semidefinite (PSD). In contrast with the Higham algorithm (2002), the Bentler-Yuan algorithm does not seek to minimize some criterion. Instead, the algorithm uses MTFA to identify Heywood cases (i.e., communality estimates greater than or equal to one and, consequently, negative or zero uniqueness variance estimates; Dillon, Kumar, & Mulani, 1987). The Bentler-Yuan algorithm then rescales the rows and columns of $\mathbf{R}_{\text{NPD}}$ corresponding to these Heywood cases to produce a smoothed, PSD correlation matrix, $\mathbf{R}_{\text{BY}}$. More specifically, the algorithm first conducts an MTFA using $\mathbf{R}_{\text{NPD}}$. Using the results of the MTFA, a diagonal matrix, $\mathbf{H}$ is constructed containing the estimated communalities as diagonal elements. Next, another diagonal matrix, $\mathbf{\Delta}^2$, is constructed with elements δ_i^2 where $\delta_i^2 = 1$ if $h_i < 1$ and $\delta_i^2 = k/h_i$ otherwise (where $k < 1$ is some constant). Finally, the smoothed, PSD correlation matrix $\mathbf{R}_{\text{BY}} = \mathbf{\Delta}\mathbf{R}_0\mathbf{\Delta} + \mathbf{I}$ is obtained, where $\mathbf{R}_0$ is $\mathbf{R}_{\text{NPD}}$ with diagonal elements replaced by zeroes and $\mathbf{I}$ is an identity matrix that ensures that $\mathbf{R}_{\text{BY}}$ has a unit diagonal.

Similar to the Higham algorithm, the Bentler-Yuan algorithm sometimes fails to produce a PSD correlation matrix. This can happen either when (a) the MTFA algorithm fails to converge or (b) when k is too large and does not shrink the targeted elements of the NPD correlation matrix enough for the matrix to become PSD. To help with this non-convergence, I modified the Bentler-Yuan algorithm to adaptively select an appropriate k by initializing at $k = 0.999$ and decreasing k by 0.001 until the algorithm produced a PSD correlation matrix or $k = 0$.[1]

Knol-Berger Algorithm (KB; 1991). In contrast to the Higham (2002) and Bentler-Yuan (2011) smoothing algorithms, the Knol-Berger algorithm is a non-iterative procedure in which the negative eigenvalues of $\mathbf{R}_{\text{NPD}}$ are replaced with some small positive value. The first step in the Knol-Berger algorithm is to compute the eigendecomposition of the $p \times p$ NPD correlation matrix, $\mathbf{R}_{\text{NPD}} = \mathbf{V\Lambda V}'$, where $\mathbf{V}$ is an orthonormal matrix containing the eigenvectors of $\mathbf{R}_{\text{NPD}}$ and $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{R}_{\text{NPD}}$, λ_i, $i \in \{1, \ldots, p\}$, ordered from largest to smallest on the diagonal ($\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_p$, $\lambda_p < 0$). Next, a matrix $\mathbf{\Lambda}_+$ is created by setting all negative elements of $\mathbf{\Lambda}$ equal to some user-specified small, positive constant. Finally, a smoothed, PSD correlation matrix, $\mathbf{R}_{\text{KB}}$, is constructed by replacing $\mathbf{\Lambda}$ with $\mathbf{\Lambda}_+$ in the eigendecomposition of $\mathbf{R}_{\text{NPD}}$ and then scaling to ensure a unit diagonal and all off-diagonal elements $|r_{ij}| \leq 1$:

$$\mathbf{R}_{\text{KB}} = [\text{dg}(\mathbf{V\Lambda}_+\mathbf{V}')]^{-1/2}\mathbf{V\Lambda}_+\mathbf{V}'\text{dg}(\mathbf{V\Lambda}_+\mathbf{V}')]^{-1/2}, \tag{1}$$

where $\text{dg}(\mathbf{V\Lambda}_+\mathbf{V}')$ returns a diagonal matrix containing the diagonal elements of $\mathbf{V\Lambda}_+\mathbf{V}'$ (Magnus & Neudecker, 2019, p. 440).

The Common Factor Model

The linear factor analysis model is used to describe the variance of each observed variable in terms of the contributions of a small number of latent common factors and a

[1] Bentler and Yuan suggest using $k = 0.96$ (Bentler & Yuan, 2011, p. 120) but claim that the precise value of k does not matter a great deal as long as k is marginally less than one.

specific factor unique to that variable (Wirth & Edwards, 2007). In the common factor model, the population correlation matrix, $\mathbf{P}$, can be expressed as:

$$\mathbf{P} = \mathbf{F}\boldsymbol{\Phi}\mathbf{F}' + \boldsymbol{\Theta}^2 \tag{2}$$

where $\mathbf{P}$ is a $p \times p$ population correlation matrix for p observed variables, $\mathbf{F}$ is a $p \times m$ factor loading matrix for m common factors, $\boldsymbol{\Phi}$ is an $m \times m$ matrix of correlations between the m common factors, and $\boldsymbol{\Theta}^2$ is a $p \times p$ diagonal matrix containing the unique variances.

Although the common factor analysis model is often useful, many authors have remarked that it constitutes an oversimplification of the complex processes that generate real, observed data (Cudeck & Henly, 1991; MacCallum & Tucker, 1991; MacCallum et al., 2001). Tucker, Koopman, and Linn (1969) suggested that the lack-of-fit between the common factor model and the complex processes underlying real data could be accounted for by modeling a large number of minor, common factors. The effects of these minor common factors represent model approximation error. The model Tucker et al. (1969) proposed can be written as:

$$\mathbf{P} = \mathbf{F}\boldsymbol{\Phi}\mathbf{F}' + \boldsymbol{\Theta}^2 + \mathbf{W}\mathbf{W}' \tag{3}$$

where $\mathbf{W}$ is a $p \times q$ matrix containing factor loadings for the $q \gg m$ minor factors (Briggs & MacCallum, 2003, p. 32). Given our expectation that the common factor model is not a perfect representation of any real-world data-generating process we might wish to represent, (3) should be preferred to (2) for simulating realistic data (Briggs & MacCallum, 2003; Hong, 1999).

Factor Analysis Algorithms

One purpose of this study was to determine whether the effects of matrix smoothing method on factor loading recovery differ depending on the factor analysis algorithm used.

To that end, three factor analysis methods were used in the current simulation: principal axes (PA), ordinary least-squares (OLS), and maximum-likelihood (ML). These factor analysis methods were chosen because they are some of the most commonly used methods (Fabrigar, Wegener, MacCallum, & Strahan, 1999) and because two of the methods (PA and OLS) work when an NPD correlation matrix is given as input.

Principal Axes Factor Analysis. Principal axes (PA) factor analysis is conceptually similar to principal components analysis (PCA). Whereas PCA seeks to find a low-dimensional approximation of the full observed correlation matrix, PA seeks to find a low-dimensional approximation of the reduced correlation matrix, $\mathbf{R}_*$ (i.e., the observed correlation matrix, $\mathbf{R}$, with communalities on the diagonal). Because the true communalities are unknown, principal axes factor analysis starts by using estimated communalities to form $\mathbf{R}_*$.[2] The eigenvalues of $\mathbf{R}_*$ are then taken to be the updated communality estimates. These updated estimates replace the previous estimates on the diagonal of $\mathbf{R}_*$ and the procedure iterates until the sum of the differences between the communality estimates from the current and previous iterations is less than some small convergence criterion.

Ordinary Least-Squares Factor Analysis. The ordinary least-squares factor analysis method (OLS; also known as "minres"; Comrey, 1962) seeks to minimize the sum of squared differences between the sample correlation matrix, $\mathbf{R}$, and $\hat{\mathbf{P}}$, the correlation matrix implied by the estimated factor model defined in (2). The OLS discrepancy

[2] Many methods of estimating communalities have been proposed, the most common of which are the squared multiple correlation between each variable and the other variables (Dwyer, 1939; Mulaik, 2009, p. 182; Roff, 1936) and the maximum absolute correlation between each variable and the other variables (Mulaik, 2009, p. 175; Thurstone, 1947). However, the particular choice of initial communality estimates has been shown to not have a large effect on the final solution when the convergence criterion is sufficiently stringent (Widaman & Herringer, 1985).

function can then be written as

$$F_{OLS}(\mathbf{R}, \hat{\mathbf{P}}) = \frac{1}{2} \operatorname{tr} \left[(\mathbf{R} - \hat{\mathbf{P}})^2 \right], \tag{4}$$

where tr is the trace operator (Magnus & Neudecker, 2019, p. 11) and $\operatorname{tr}\left[(\mathbf{R} - \hat{\mathbf{P}})^2\right]$ is the trace (sum of the diagonal elements) of the matrix formed by $(\mathbf{R} - \hat{\mathbf{P}})^2$. OLS does not give additional weight to residuals corresponding to large correlations and requires no assumptions about the population distributions of the variables (Briggs & MacCallum, 2003).

Maximum-Likelihood Factor Analysis. The maximum likelihood factor analysis algorithm (ML) is similar to OLS in that it seeks to minimize the discrepancy between $\mathbf{R}$ and $\hat{\mathbf{P}}$. Unlike OLS, however, ML assumes that all variables are multivariate normal in the population. Then, we can write the discrepancy function to be minimized as an alternative form of the multivariate normal log-likelihood function,

$$F_{ML}(\mathbf{R}, \hat{\mathbf{P}}) = \log |\hat{\mathbf{P}}| - \log |\mathbf{R}| + \operatorname{tr}(\mathbf{S}\hat{\mathbf{P}}^{-1}) - p. \tag{5}$$

In addition to the distributional assumptions required by ML factor analysis, the method also assumes that the only source of error in the model is sampling error. Consequently, large correlations (having relatively small standard errors) are fit more closely than small correlations (with relatively large standard errors) under maximum likelihood factor analysis (Briggs & MacCallum, 2003). Also note that when $\mathbf{R}$ is NPD, $|\mathbf{R}|$ is negative and $\log |\mathbf{R}|$ is undefined. Therefore, NPD covariance or correlation matrices cannot be used as input for maximum likelihood factor analysis.

Simulation Procedure

I conducted a simulation study to evaluate four approaches to dealing with NPD tetrachoric correlation matrices (applying matrix smoothing using the Higham [2002], Bentler-Yuan [2011], or Knol-Berger [1991] algorithms, or leaving NPD tetrachoric matrices unsmoothed) in the context of exploratory factor analysis. The simulation study was designed to address two primary questions. First, which smoothing method (Higham, Bentler-Yuan, Knol-Berger, or None) produced (possibly) smoothed correlation matrices ($\mathbf{R}_{\text{Sm}}$) that most closely approximated the corresponding population correlation matrices? Second, which smoothing method produced correlation matrices that led to the best estimates of the population factor loading matrix when used in exploratory factor analyses?

In the first step of the simulation study, I generated random sets of binary data from a variety of orthogonal factor models with varying numbers of major common factors (Factors $\in \{1, 3, 5, 10\}$). Following the procedure of Tucker et al. (1969), I also incorporated the effects of model approximation error into the data by including 150 minor common factors in each population model. In total, these 150 minor common factors accounted for 0%, 10%, or 30% (Error $\in \{0, .1, .3\}$) of the uniqueness variance of the error-free model (i.e., the model with only the major common factors). These conditions were chosen to represent models with perfect, good, or moderate model fit, resembling the conditions used by Briggs and MacCallum (2003). These three levels of model approximation error in the simulation ensured that both ideal (Error $= 0$) and more empirically-plausible levels of model approximation error (Error $\in \{.1, .3\}$) were considered in this study.

In addition to systematically varying the number of major factors and the proportion of variance accounted for by model approximation error, I also varied the number of factor indicators (i.e., items loading on each factor; Items/Factor $\in \{5, 10\}$), and the number of subjects per item (Subjects/Item $\in \{5, 10, 15\}$). The total numbers of items and sample sizes for each factor number condition can be found in Table 1. Each item loaded on only one factor and item factor loadings were uniformly fixed at one of three levels (Loading $\in \{.3, .5, .8\}$). Though "rules-of-thumb" for factor loadings vary, Hair, Black, Babin, and Anderson (2018, p. 151) suggest that "[f]actor loadings in the range of ± 0.30 to ± 0.40 are considered to meet the minimal level for interpretation of structure", and "[l]oadings ± 0.50 or greater are considered practically significant." Moreover, factor loadings of ± 0.8 are considered to be high (MacCallum et al., 2001). Thus, the three factor loadings investigated in this study were chosen to represent low, moderate, and high levels of factor salience.

The combinations of the independent variables specified above resulted in a fully-crossed design with 4 (Factors) $\times$ 3 (Error) $\times$ 2 (Items/Factor) $\times$ 3 (Subjects/Item) $\times$ 3 (Loading) = 216 unique conditions. For each of these conditions, I used the `simFA` function in the R (Version 3.6.2; R Core Team, 2019)[3] *fungible* library (Version 1.95.4.8; Waller, 2019) to generate 1,000 random sets of data in accordance with the factor model corresponding to that condition. To obtain binary responses from continuous observed scores, items were assigned classical item difficulties (d; i.e., the expected proportion of correct responses, Crocker & Algina, 1986) at equal intervals between 0.15 and 0.85. For example, items in a five-item data set were assigned classical item difficulties of .150, .325, .500, .675, and .850. The classical item difficulties were used to obtain threshold values, t, such that $P(X > t) = d$ where $X \sim N(0, 1)$. I then used the thresholds to dichotomize the continuous observed scores and

[3] Additionally, I used the following R packages: *arm* (Version 1.10.1; Gelman & Su, 2018), *broom.mixed* (Version 0.2.4; Bolker & Robinson, 2019), *car* (Version 3.0.7; Fox & Weisberg, 2019), *dplyr* (Version 0.8.5; Wickham et al., 2019), *forcats* (Version 0.5.0; Wickham, 2019a), *ggplot2* (Version 3.3.0; Wickham, 2016), *here* (Version 0.1.11; Müller, 2017), *knitr* (Version 1.28; Xie, 2015), *koRpus* (Version 0.11.5; Michalke, 2018a, 2019), *koRpus.lang.en* (Version 0.1.3; Michalke, 2019), *latex2exp* (Version 0.4.0; Meschiari, 2015), *lattice* (Version 0.20.38; Sarkar, 2008), *lme4* (Version 1.1.23; Bates, Mächler, Bolker, & Walker, 2015), *MASS* (Version 7.3.51.4; Venables & Ripley, 2002), *Matrix* (Version 1.2.18; Bates & Maechler, 2019), *merTools* (Version 0.5.0; Knowles & Frederick, 2019), *papaja* (Version 0.1.0.9942; Aust & Barth, 2018), *patchwork* (Version 1.0.0; Pedersen, 2019), *purrr* (Version 0.3.4; Henry & Wickham, 2019), *questionr* (Version 0.7.0; Barnier, Briatte, & Larmarange, 2018), *readr* (Version 1.3.1; Wickham, Hester, & Francois, 2018), *sfsmisc* (Version 1.1.4; Maechler, 2019), *stringr* (Version 1.4.0; Wickham, 2019b), *sylly* (Version 0.1.5; Michalke, 2018b), *texreg* (Version 1.36.23; Leifeld, 2013), *tibble* (Version 3.0.1; Müller & Wickham, 2019), *tidyr* (Version 1.0.2.9000; Wickham & Henry, 2019), *tidyverse* (Version 1.3.0; Wickham, Averick, et al., 2019), *viridis* (Version 0.5.1; Garnier, 2018), and *wordcountaddin* (Version 0.3.0.9000; Marwick, 2019).

obtain simulated binary response data. If a data set had any homogeneous item response vectors (i.e., had one or more items with zero variance), the data set was discarded and a new sample of data was generated until all items had non-homogeneous response vectors. This procedure was necessary to calculate tetrachoric correlation matrices in the next step of the simulation.

Next, I calculated a tetrachoric correlation matrix for each simulated binary data set. Tetrachoric correlation matrices were calculated using the `tetcor` function in the R *fungible* library (Waller, 2019), which computes maximum likelihood tetrachoric correlation coefficients (Brown & Benedetti, 1977; Olsson, 1979). If a tetrachoric correlation matrix was NPD, the Higham (2002), Bentler-Yuan (2011), and Knol-Berger (1991) matrix smoothing algorithms were applied to the NPD tetrachoric correlation matrix to produce three smoothed, PSD correlation matrices. Matrix smoothing was done using the `smoothAPA`, `smoothBY`, and `smoothKB` implementations of the Higham (2002), Bentler-Yuan (2011), and Knol-Berger (1991) algorithms in *fungible*.

In the third and final step of the simulation procedure, I applied three exploratory factor analysis algorithms (principal axes [PA], ordinary least squares [OLS], and maximum likelihood [ML]) to each of the NPD tetrachoric correlation matrices and the PSD, smoothed correlation matrices. Because ML does not work with NPD correlation or covariance matrices as input, ML was conducted on the Pearson correlation matrix (rather than the NPD tetrachoric correlation matrix) when no smoothing was applied. Each of the factor solutions were then rotated using a quartimin rotation (Carroll, 1957; Jennrich, 2002) and aligned to match the corresponding population factor loading matrix such that the least squares discrepancy between the matrices was minimized. The alignment step ensured that the elements of each estimated factor loading matrix were matched (in order and sign) to the elements of the corresponding population factor loading matrix. These rotation and alignment steps were accomplished using the `faAlign`, `faMain`, and

`GPArotation` functions in the R *fungible* library (Waller, 2019). Code for all aspects of this study including the simulation, analyses, figures, and tables is available at https://github.umn.edu/krach018/masters_thesis.

Results

Recovery of the population correlation matrix

One of the primary reasons for conducting the present simulation study was to determine which of the three investigated smoothing methods — the Higham (2002), Bentler-Yuan (2011), or Knol-Berger (1991) algorithms — resulted in smoothed correlation matrices that were closest to the correlation matrix implied by the major factor model (i.e., the factor model not including the minor factors). In particular, I examined whether smoothed correlation matrices were closer to the model-implied correlation matrix than the unsmoothed, NPD correlation matrix. In this context, the scaled distance between two $p \times p$ correlation matrices $\mathbf{A} = \{a_{ij}\}$ and $\mathbf{B} = \{b_{ij}\}$ was computed as:

$$\mathrm{D_s}(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \frac{(a_{ij} - b_{ij})^2}{p(p-1)/2}}. \tag{6}$$

To understand which of the smoothing algorithms most often produced a smoothed correlation matrix, $\mathbf{R}_{\mathrm{Sm}}$, that was closest to the model-implied correlation matrix, $\mathbf{R}_{\mathrm{Pop}}$, I calculated $\mathrm{D_s}(\mathbf{R}_{\mathrm{Sm}}, \mathbf{R}_{\mathrm{Pop}})$ for each $\mathbf{R}_{\mathrm{Sm}}$ obtained from the 124,346 NPD tetrahoric correlation matrices. Small values of $\mathrm{D_s}(\mathbf{R}_{\mathrm{Sm}}, \mathbf{R}_{\mathrm{Pop}})$ indicated that the smoothed correlation matrix was a good approximation of $\mathbf{R}_{\mathrm{Pop}}$, whereas large values indicated that $\mathbf{R}_{\mathrm{Sm}}$ was a poor approximation of $\mathbf{R}_{\mathrm{Pop}}$. After excluding the three cases where the Higham (2002) algorithm failed to converge, I calculated the mean $\mathrm{D_s}(\mathbf{R}_{\mathrm{Sm}}, \mathbf{R}_{\mathrm{Pop}})$ for each of the remaining 497,384 smoothed matrices. On average, the Bentler-Yuan algorithm produced smoothed correlation matrices that were slightly closer to $\mathbf{R}_{\mathrm{Pop}}$ ($M = 0.112$, $SD = 0.053$) than the smoothed matrices produced by the Knol-Berger ($M = 0.117$, $SD = 0.056$) or

Higham ($M = 0.118$, $SD = 0.056$) algorithms. The mean distance between the NPD correlation matrices, $\mathbf{R}_{\text{NPD}}$, and $\mathbf{R}_{\text{Pop}}$ was larger ($M = 0.121$, $SD = 0.058$) than the mean distances for any of the three smoothing algorithms.

To get a more detailed look at the how the smoothed correlation matrices approximated $\mathbf{R}_{\text{Pop}}$, I fit a linear mixed-effects model regressing $\log \mathrm{D_s}(\mathbf{R}_{\text{Sm}}, \mathbf{R}_{\text{Pop}})$ on number of subjects per item (5, 10, 15), number of items per factor (5, 10), number of factors (1, 3, 5, 10), factor loading (0.3, 0.5, 0.8), model error (0.0, 0.1, 0.3), smoothing algorithm (Higham, Bentler-Yuan, Knol-Berger, or no smoothing), all two-way interactions between these variables, and a random intercept estimated for every unique NPD correlation matrix.[4] The estimated fixed-effect coefficients are in Figure 1. A full summary table for the model appears in Table 2.

Figure 1 shows that only a few variables had non-trivial effects on population matrix recovery. In particular, the three most potent effects were number of factors ($b = -0.313$, $\text{SE} = 0.0004$, $e^{-.313} = 0.731$), number of subjects per item ($b = -0.221$, $\text{SE} = 0.0005$, $e^{-0.221} = 0.802$), and number of items per factor ($b = -0.164$, $\text{SE} = 0.0004$, $e^{-0.164} = 0.849$). These estimated effects were all negative, indicating better recovery of the population correlation matrix for models with larger numbers of major factors, larger numbers of subjects per item, and larger numbers of items per factor, all else being equal. All three smoothing algorithms were associated with small (negative) estimated main effects. In particular, the estimated main effect for the Bentler-Yuan algorithm was smallest ($b = -0.073$, $\text{SE} = 0.0001$, $e^{-0.073} = 0.930$), followed by the estimated main effects for the Knol-Berger ($b = -0.033$, $\text{SE} = 0.0001$, $e^{-0.033} = 0.968$) and Higham ($b = -0.024$, $\text{SE} = 0.0001$, $e^{-0.024} = 0.976$) algorithms. These main effects were offset somewhat by small, positive interaction effects between the smoothing methods and subjects per item,

[4] All numeric predictors were scaled to have a mean of zero and variance of one prior to analysis. Diagnostic plots can be found in Appendix A.

and between the smoothing methods and factor loading (see Table 2 and Figure 1).

To get a better sense of the relative performance of the smoothing algorithms (in terms of population correlation matrix recovery), Figure 2 shows box-plots of $\mathrm{D_s}(\mathbf{R}_{\mathrm{Sm}}, \mathbf{R}_{\mathrm{Pop}})$ for all combinations of smoothing method, factor loading size, number of subjects per item, and number of items per factor. The most apparent feature of Figure 2 was the improvement of population correlation matrix recovery as number of items per factor and number of subjects per item increased. The conditions with loadings fixed at 0.3, and 15 subjects per item (see the cells in the upper right-hand corner of Figure 2) might seem to go against the trend of lower $\mathrm{D_s}(\mathbf{R}_{\mathrm{Sm}}, \mathbf{R}_{\mathrm{Pop}})$ for higher numbers of subjects per item. However, these results are likely an artifact of the lack of NPD correlation matrices for these combinations of conditions (only 50 and 98 NPD tetrachoric correlation matrices for the five items-per-factor and ten items-per-factor conditions, respectively). The other important trend in Figure 2 was that the Bentler-Yuan algorithm performed best relative to the other smoothing methods in conditions with few subjects per item and items per factor. However, this advantage became nearly imperceptible as the numbers of subjects per item and items per factor increased. The interaction between the Bentler-Yuan algorithm and magnitude of factor loading was also evident, such that the Bentler-Yuan algorithm performed worse as factor loadings increased.

Taken as a whole, the results suggest that three variables accounted for the majority of the variation in $\mathrm{D_s}(\mathbf{R}_{\mathrm{Sm}}, \mathbf{R}_{\mathrm{Pop}})$: (a) the number of major factors in the data-generating model, (b) the number of subjects per item, and (c) the number of items per (major) factor. Increases in any of these variables were associated with improved population correlation matrix recovery. Choice of smoothing method was also related to population correlation matrix recovery, to some extent. In particular, smoothed matrices were slightly closer to the population correlation matrix than the unsmoothed tetrachoric correlation matrices. Of the three smoothing algorithms used, the Bentler-Yuan algorithm produced smoothed matrices that were closest to the population correlation matrices. However, differences between smoothing methods were small except in conditions with few subjects per item, few items per factor, and low factor loadings.

Recovery of factor loadings

I next analyzed the results in terms of factor loading recovery. In particular, I sought to determine whether factor analysis of smoothed matrices led to better factor loading estimates than unsmoothed matrices, and if particular smoothing methods led to better factor loading estimates than others. I was also interested in whether the interactions between smoothing methods and the other variables (e.g., number of items per factor, number of subjects per item, factor analysis method, etc.) affected the relative smoothing algorithm performance in terms of factor loading estimation. For the purposes of these analyses, I evaluated factor loading recovery using the root-mean-square error (RMSE) between the estimated and population factor loadings for the major factors. Given a matrix of estimated major factor loadings $\hat{\mathbf{\Lambda}} = \{\hat{\lambda}_{ij}\}_{p \times m}$, and the corresponding matrix of population major factor loadings, $\mathbf{\Lambda} = \{\lambda_{ij}\}_{p \times m}$,

$$\text{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}) = \sqrt{\sum_{i=1}^{p} \sum_{j=1}^{m} \frac{(\lambda_{ij} - \hat{\lambda}_{ij})^2}{pm}}. \tag{7}$$

To determine which smoothing method resulted in the best factor loading estimates, I calculated the $\text{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})$ for each pair of estimated and population factor loading matrices corresponding to the (possibly) smoothed NPD tetrachoric correlation matrices. Relatively small $\text{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})$ values indicated that the estimated factor loading matrices were more similar to their corresponding population factor loading matrices, whereas larger $\text{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})$ values indicated poorly-estimated factor loading matrices. As in the previous section, the four cases where the Higham (2002) algorithm did not converge were not included in my analyses. Furthermore, cases where PA failed to converge were also not included. In total, there were 2,714 cases where the PA algorithm did not converge (convergence rate = 99.5%) and only four cases where the ML algorithm did not converge (convergence rate > 99.9%). For the 1,489,425 cases remaining, factor analysis of the Bentler-Yuan (2011) smoothed matrices resulted in the lowest mean $\text{RMSE}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}})$ ($M = 0.133$, $SD = 0.092$) whereas the smoothed matrices produced by the Higham (2002; $M = 0.135$, $SD = 0.097$) and Knol-Berger (1991; $M = 0.135$, $SD = 0.097$) algorithms led to slightly higher mean $\text{RMSE}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}})$ values. Factor analyzing unsmoothed NPD tetrachoric correlation matrices led to the highest mean $\text{RMSE}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}})$ ($M = 0.137$, $SD = 0.103$).

To obtain estimates of effects, I fit a linear mixed-effects model regressing $\log \text{RMSE}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}})$ on number of subjects per item, number of items per factor, number of factors, factor loading, model error, smoothing algorithm (Higham, Bentler-Yuan, Knol-Berger, or no smoothing), factor analysis method (PA, OLS, or ML), all two-way interactions between these variables, and a random intercept estimated for every unique NPD correlation matrix.[5] The estimated (exponentiated) fixed-effect coefficients are shown in Figure 5. A full summary table including (untransformed) coefficient estimates and standard errors appears in Table 3.

Figure 5 shows that only a few variables had non-negligible effects on factor loading recovery. None of the effects of primary interest to this study—the main effects or two-way interactions involving the smoothing methods—were large enough to hold much practical significance. For instance, the results indicated that not applying smoothing to an NPD tetrachoric correlation matrix prior to factor analysis led to the worst factor loading estimates among the four smoothing methods. However, this effect represented only a minute improvement in $\text{RMSE}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}})$ for smoothed compared to unsmoothed NPD

[5] All numeric predictors were scaled to have a mean of zero and variance of one prior to analysis. Diagnostic plots can be found in Appendix A.

tetrachoric correlation matrices (see Figure 5 and Table 3).

Although none of the primary effects of interest to this study were large, there were some estimated effects that, although ancillary for this study, were large enough to warrant mention. In particular, there were moderate, positive effects for ML ($b = 0.116$, SE = 0.0004, $e^{0.116} = 1.123$) and the interactions between ML and factor loading ($b = 0.204$, SE = 0.0002, $e^{0.204} = 1.226$), ML and subjects per item ($b = 0.107$, SE = 0.0002, $e^{0.107} = 1.113$), and ML and items per factor ($b = 0.082$, SE = 0.0002, $e^{0.082} = 1.085$). There was also a positive estimated effect for model error ($b = 0.104$, SE = 0.0005, $e^{0.104} = 1.11$). These results suggest that both the use of ML factor analysis (compared with PA or OLS) and higher levels of model approximation error led to worse factor loading recovery. Moreover, factor loading estimates for ML were not improved as much by increasing factor loadings, subjects per item, or items per factor (which were associated with negative effects) compared to PA or OLS. To illustrate this interaction, the effects of factor analysis method, factor loading, and number of subject per item are shown in Figure 3, which contains box plots of RMSE($\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}$) for each combination of these variables. This figure shows that ML often led to the lowest RMSE($\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}$) values in conditions with small factor loadings but did not improve as much as OLS or PA as factor loadings increased. Similar effects can be seen for the number of subjects per item; although RMSE($\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}$) values generally decreased as number of subjects per item increased for all factor analysis methods, ML seemed to benefit least.[6] These results should be interpreted carefully, however, because ML factor analysis when no smoothing was applied was (by necessity) conducted on Pearson correlation matrices whereas the other factor analysis algorithms were applied to the NPD tetrachoric correlation matrices in the no smoothing condition.

There were also large, negative effects for the variables that should be expected to

[6] Number of items per factor was not included in Figure 3 because the differences in RMSE($\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}$) across levels of the condition were too small to be clearly seen.

related to improved factor loading recovery, namely, factor loading ($b = -0.438$, SE $=$ 0.0006, $exp^{-0.438} = 0.645$), subjects per item ($b = -0.189$, SE $= 0.0007$, $exp^{-0.189} = 0.828$), and items per factor ($b = -0.175$, SE $= 0.0005$, $exp^{-0.175} = 0.839$). There was also a relatively large and seemingly anomalous negative effect for number of factors ($b = -0.255$, SE $= 0.0005$, $exp^{-0.255} = 0.775$). On its face, this effect seems to suggest that data generated from models with large numbers of major factors led to better factor loading recovery. However, this effect is most likely due to the fact that, whereas number of items per factor and number of subjects per item were fully-crossed with (orthogonal to) number of factors, the total sample size and number of items for each data set were confounded with number of factors. In other words, conditions with larger numbers of factors tended to include more items in total, and therefore also tended to have larger sample sizes despite having the same numbers of indicators (items) per factor and numbers of subjects per indicator. The strong relationship between $\log \mathrm{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})$ and total sample size can be clearly seen in Figure 4, which shows that $\log \mathrm{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})$ decreased as sample size increased. Therefore, it seems reasonable to conclude that the effect of number of factors can be better understood as being related to the total number of items and subjects in a data set. Similarly, the negative interaction between number of factors and ML ($b = -0.090$, SE $= 0.0002$, $exp^{-0.090} = 0.914$) might be interpreted as an interaction between total number of items or subjects and ML. In summary, the results of the simulation study indicated that there was no meaningful advantage of using any smoothing algorithm over any other. Moreover, there was no large advantage (in terms of $\mathrm{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})$) to smoothing NPD tetrachoric correlation matrices prior to conducting exploratory factor analysis.

Discussion

Interpretation of Results

The current study examined how the application of three matrix smoothing algorithms (the Higham [2002], Bentler-Yuan [2011], and Knol-Berger [1991] algorithms) to NPD tetrachoric correlation matrices affected both (a) the recovery of the model-implied population correlation matrix ($\mathbf{R}_{\text{Pop}}$), and (b) the recovery of the population item factor loadings in EFA (compared to leaving the NPD correlation matrices unsmoothed). With respect to recovery of $\mathbf{R}_{\text{Pop}}$, I found that the application of any of the matrix smoothing algorithms included in the present study led to slightly better recovery of the $\mathbf{R}_{\text{Pop}}$ compared to the unsmoothed, NPD tetrachoric correlation matrix. Of the three matrix smoothing algorithms included in this study, the application of the Bentler-Yuan algorithm (2011) produced the best approximations of $\mathbf{R}_{\text{Pop}}$ (on average). In particular, the Bentler-Yuan algorithm led to the best results relative to the other smoothing algorithms in conditions with low factor loadings, few items per factor, and few subjects per item. However, differences between smoothing algorithms (in terms of recovery of $\mathbf{R}_{\text{Pop}}$) were mostly so small as to be of little practical importance. With respect to the recovery of population factor loadings, I found that the particular matrix smoothing algorithm applied to an NPD tetrachoric correlation matrix prior to EFA led to no meaningful differences in factor loading recovery. Moreover, conducting EFA on smoothed, PSD correlation matrices led to only marginally better factor loading recovery compared to conducting EFA on NPD, unsmoothed correlation matrices.

Limitations and Future Directions

As with any simulation study, the present simulation design was not able to cover the full range of realistic data scenarios. For instance, the simulation design included only orthogonal population factor models and did not allow for correlated factors. Future research on this topic should investigate whether more complex correlation structures affect the performance of matrix smoothing algorithms in terms of population correlation matrix recovery and factor loading recovery. Moreover, the present studies only investigated the effects of matrix smoothing on NPD tetrachoric correlation matrices. Further research should be done to investigate the effects of matrix smoothing on NPD polychoric correlation matrices, as well as correlation matrices that are NPD due to other causes. For instance, NPD correlation matrices calculated using pairwise deletion (Wothke, 1993) or composite correlation matrices used in meta-analysis (Furlow & Beretvas, 2005). Little is known about whether the mechanism or "cause" of NPD correlation matrices affects their structure or how these potential differences might interact with the application of matrix smoothing algorithms.

Future research should also investigate ways to side-step the problem of NPD tetrachoric correlation matrices. For instance, Choi, Kim, Chen, and Dannels (2011) found that polychoric correlation matrices estimated using expected a posteriori (EAP) rather than maximum-likelihood estimation led to estimates that were negatively biased but produced comparable (or smaller) RMSE values in terms of recovering the "true" correlations. It seems plausible that the slight shrinkage induced by using EAP as an estimation method would make NPD tetrachoric or polychoric correlation matrices less common. Finally, full-information maximum likelihood (FIML; Bock & Aitkin, 1981) can be used to estimate model parameters directly and doesn't require the estimation of a tetrachoric correlation matrix. Future research should investigate whether the use of FIML (which is computationally intenstive, particularly with large models) offers any benefit, in terms of parameter recovery, when applied to datasets corresponding to NPD tetrachoric correlation matrices.

Conclusion

Despite the lackluster improvement in factor loading recovery when factor analysis was conducted on smoothed rather than NPD tetrachoric correlation matrices, the application of one of the three investigated matrix smoothing algorithms on NPD tetrachoric correlation matrices is still recommended. None of the smoothing algorithms regularly led to worse results (in terms of factor loading recovery) compared to the conditions where the NPD correlation matrix was left unsmoothed. Moreover, all of the smoothing algorithms investigated in this study are computationally inexpensive and are readily available as functions in R packages. For instance, the *fungible* (Waller, 2019), *sfsmisc* (Maechler, 2019), and *Matrix* (Bates & Maechler, 2019) packages all contain implementations of at least one of the three smoothing algorithms discussed in this article. In particular, the Knol-Berger algorithm (1991) is recommended as a smoothing algorithm that is fast, easily implemented in most programming languages, does not have convergence issues, and led to results comparable to the Bentler-Yuan and Higham algorithms.

This recommendation comes with a strong caveat; Namely, that no matrix smoothing algorithm can reasonably be considered a remedy or solution for NPD tetrachoric correlation matrices. Instead, researchers should consider NPD tetrachoric correlation matrices to be symptoms of larger problems (e.g., small sample sizes, bad items. etc.) and be aware that practical solutions such as gathering more data or discarding bad items are likely to lead to better results than the application of matrix smoothing algorithms. In particular, NPD tetrachoric correlation matrices are less likely to occur when sample sizes are large relative to the number of items (see Table 1 in Debelak & Tran, 2013, p. 70), allowing researchers to avoid the question of how to properly deal with an NPD tetrachoric correlation matrix entirely. If collecting more data is not possible, researchers should consider removing problematic items. In short, all three investigated smoothing algorithms are reasonable choices for dealing with NPD tetrachoric correlation matrices prior to factor analysis and seem to offer a modest benefit (in terms of factor loading recovery) compared to leaving the NPD tetrachoric correlation matrix unsmoothed. However, the application of these algorithms should be considered to be little more than a band-aid fix that does not address the underlying issues leading to NPD tetrachoric correlation matrices.

References

Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown.* Retrieved from https://github.com/crsh/papaja

Banerjee, S., & Roy, A. (2014). *Linear algebra and matrix analysis for statistics.* Chapman; Hall/CRC.

Barnier, J., Briatte, F., & Larmarange, J. (2018). *Questionr: Functions to make surveys processing easier*. Retrieved from https://CRAN.R-project.org/package=questionr

Bates, D., & Maechler, M. (2019). *Matrix: Sparse and dense matrix classes and methods.* Retrieved from https://CRAN.R-project.org/package=Matrix

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Bentler, P. (1972). A lower-bound method for the dimension-free measurement of internal consistency. *Social Science Research*, *1*(4), 343–357.

Bentler, P., & Yuan, K.-H. (2011). Positive definiteness via off-diagonal scaling of a symmetric indefinite matrix. *Psychometrika*, *76*(1), 119–123.

Birnbaum, A. L. (1968). Some latent trait models and their use in inferring an examinee's ability. *Statistical Theories of Mental Test Scores.*

Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, *46*(4), 443–459. https://doi.org/10.1007/BF02293801

Bock, R. D., Gibbons, R., & Muraki, E. (1988). Full-information item factor analysis.

Applied Psychological Measurement, *12*(3), 261–280.

Bolker, B., & Robinson, D. (2019). *Broom.mixed: Tidying methods for mixed models.* Retrieved from https://CRAN.R-project.org/package=broom.mixed

Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, *26*(2), 211–243.

Briggs, N. E., & MacCallum, R. C. (2003). Recovery of weak common factors by maximum likelihood and ordinary least squares estimation. *Multivariate Behavioral Research*, *38*(1), 25–56.

Brown, M. B., & Benedetti, J. K. (1977). On the mean and variance of the tetrachoric correlation coefficient. *Psychometrika*, *42*(3), 347–355.

Butler, S. M., & Louis, T. A. (1992). Random effects models with non-parametric priors. *Statistics in Medicine*, *11*(14-15), 1981–2000.

Carroll, J. B. (1957). Biquartimin criterion for rotation to oblique simple structure in factor analysis. *Science*, *126*, 1114–1115. https://doi.org/10.1126/science.126.3283.1114

Choi, J., Kim, S., Chen, J., & Dannels, S. (2011). A comparison of maximum likelihood and Bayesian estimation for polychoric correlation using monte carlo simulation. *Journal of Educational and Behavioral Statistics*, *36*(4), 523–549.

Comrey, A. L. (1962). The minimum residual method of factor analysis. *Psychological Reports*, *11*(1), 15–18.

Crocker, L., & Algina, J. (1986). *Introduction to classical and modern test theory.* ERIC.

Cudeck, R., & Henly, S. J. (1991). Model selection in covariance structures analysis and the “problem” of sample size: A clarification. *Psychological Bulletin*, *109*(3), 512.

de Ayala, R. J. (2013). *The theory and practice of item response theory*. Guilford Publications.

Debelak, R., & Tran, U. S. (2013). Principal component analysis of smoothed tetrachoric correlation matrices as a measure of dimensionality. *Educational and Psychological Measurement*, *73*(1), 63–77.

Debelak, R., & Tran, U. S. (2016). Comparing the effects of different smoothing algorithms on the assessment of dimensionality of ordered categorical items with parallel analysis. *PLOS ONE*, *11*(2), 1–18. https://doi.org/10.1371/journal.pone.0148143

Devlin, S. J., Gnanadesikan, R., & Kettenring, J. R. (1975). Robust estimation and outlier detection with correlation coefficients. *Biometrika*, *62*(3), 531–545.

Dillon, W. R., Kumar, A., & Mulani, N. (1987). Offending estimates in covariance structure analysis: Comments on the causes of and solutions to heywood cases. *Psychological Bulletin*, *101*(1), 126.

Dong, H.-K. (1985). Non-gramian and singular matrices in maximum likelihood factor analysis. *Applied Psychological Measurement*, *9*(4), 363–366.

Dwyer, P. S. (1939). The contribution of an orthogonal multiple factor solution to multiple correlation. *Psychometrika*, *4*(2), 163–171.

Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., & Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods*, *4*(3), 272.

Fox, J., & Weisberg, S. (2019). *An R companion to applied regression* (Third). Thousand Oaks CA: Sage. Retrieved from https://socialsciences.mcmaster.ca/jfox/Books/Companion/

Furlow, C. F., & Beretvas, S. N. (2005). Meta-analytic methods of pooling correlation matrices for structural equation modeling under different patterns of missing data. *Psychological Methods*, *10*(2), 227.

Fushiki, T. (2009). Estimation of positive semidefinite correlation matrices by using convex quadratic semidefinite programming. *Neural Computation*, *21*(7), 2028–2048.

Garnier, S. (2018). *Viridis: Default color maps from 'matplotlib'*. Retrieved from https://CRAN.R-project.org/package=viridis

Gelman, A., & Su, Y.-S. (2018). *Arm: Data analysis using regression and multilevel/hierarchical models*. Retrieved from https://CRAN.R-project.org/package=arm

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2018). *Multivariate data analysis*. Cengage.

Henry, L., & Wickham, H. (2019). *Purrr: Functional programming tools*. Retrieved from https://CRAN.R-project.org/package=purrr

Higham, N. J. (2002). Computing the nearest correlation matrix—a problem from finance. *IMA Journal of Numerical Analysis*, *22*(3), 329–343.

Hong, S. (1999). Generating correlation matrices with model error for simulation studies in factor analysis: A combination of the Tucker-Koopman-Linn model and Wijsman's algorithm. *Behavior Research Methods, Instruments, & Computers*, *31*(4), 727–730.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, *30*(2), 179–185.

Jacqmin-Gadda, H., Sibillot, S., Proust, C., Molina, J.-M., & Thiébaut, R. (2007). Robustness of the linear mixed model to misspecified error distribution.

Computational Statistics & Data Analysis, *51*(10), 5142–5154.

Jamshidian, M., & Bentler, P. M. (1998). A quasi-Newton method for minimum trace factor analysis. *Journal of Statistical Computation and Simulation*, *62*(1-2), 73–89.

Jennrich, R. I. (2002). A simple general method for oblique rotation. *Psychometrika*, *67*(1), 7–19. https://doi.org/10.1007/BF02294706

Knol, D. L., & Berger, M. P. (1991). Empirical comparison between factor analysis and multidimensional item response models. *Multivariate Behavioral Research*, *26*(3), 457–477.

Knowles, J. E., & Frederick, C. (2019). *MerTools: Tools for analyzing mixed effect regression models*. Retrieved from https://CRAN.R-project.org/package=merTools

Koller, M. (2016). robustlmm: An R package for robust estimation of linear mixed-effects models. *Journal of Statistical Software*, *75*(6), 1–24. https://doi.org/10.18637/jss.v075.i06

Leifeld, P. (2013). texreg: Conversion of statistical model output in R to LaTeX and HTML tables. *Journal of Statistical Software*, *55*(8), 1–24. Retrieved from http://www.jstatsoft.org/v55/i08/

Li, Q., Li, D., & Qi, H. (2010). Newton's method for computing the nearest correlation matrix with a simple upper bound. *Journal of Optimization Theory and Applications*, *147*(3), 546–568.

Lurie, P. M., & Goldberg, M. S. (1998). An approximate method for sampling correlated random variables from partially-specified distributions. *Management Science*, *44*(2), 203–218.

MacCallum, R. C., & Tucker, L. R. (1991). Representing sources of error in the common-factor model: Implications for theory and practice. *Psychological Bulletin*, *109*(3), 502.

MacCallum, R. C., Widaman, K. F., Preacher, K. J., & Hong, S. (2001). Sample size in factor analysis: The role of model error. *Multivariate Behav. Res.*, *36*(4), 611–637.

Maechler, M. (2019). *Sfsmisc: Utilities from 'seminar fuer statistik' eth zurich*. Retrieved from https://CRAN.R-project.org/package=sfsmisc

Magnus, J. R., & Neudecker, H. (2019). *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons.

Marwick, B. (2019). *Wordcountaddin: Word counts and readability statistics in r markdown documents*. Retrieved from https://github.com/benmarwick/wordcountaddin

Meschiari, S. (2015). *Latex2exp: Use latexexpressions in plots*. Retrieved from https://CRAN.R-project.org/package=latex2exp

Michalke, M. (2018a). *KoRpus: An r package for text analysis*. Retrieved from https://reaktanz.de/?c=hacking&s=koRpus

Michalke, M. (2018b). *Sylly: Hyphenation and syllable counting for text analysis*. Retrieved from https://reaktanz.de/?c=hacking&s=sylly

Michalke, M. (2019). *KoRpus.lang.en: Language support for 'koRpus' package: English*. Retrieved from https://reaktanz.de/?c=hacking&s=koRpus

Mulaik, S. A. (2009). *Foundations of factor analysis*. Chapman; Hall/CRC.

Müller, K. (2017). *Here: A simpler way to find your files*. Retrieved from https://CRAN.R-project.org/package=here

Müller, K., & Wickham, H. (2019). *Tibble: Simple data frames*. Retrieved from https://CRAN.R-project.org/package=tibble

Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, *44*(4), 443–460.

Pedersen, T. L. (2019). *Patchwork: The composer of plots*. Retrieved from https://CRAN.R-project.org/package=patchwork

Qi, H., & Sun, D. (2006). A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, *28*(2), 360–385.

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Roff, M. (1936). Some properties of the communality in multiple factor theory. *Psychometrika*, *1*(2), 1–6.

Sarkar, D. (2008). *Lattice: Multivariate data visualization with r*. New York: Springer. Retrieved from http://lmdvr.r-forge.r-project.org

Thurstone, L. L. (1947). *Multiple-factor analysis; a development and expansion of the vectors of mind* (pp. xix, 535). University of Chicago Press.

Tucker, L. R., Koopman, R. F., & Linn, R. L. (1969). Evaluation of factor analytic research procedures by means of simulated correlation matrices. *Psychometrika*, *34*(4), 421–459.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (Fourth). New York: Springer. Retrieved from http://www.stats.ox.ac.uk/pub/MASS4

Verbeke, G., & Lesaffre, E. (1997). The effect of misspecifying the random-effects distribution in linear mixed models for longitudinal data. *Computational Statistics & Data Analysis*, *23*(4), 541–556.

Waller, N. G. (2019). *Fungible: Psychometric functions from the Waller lab.*

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from https://ggplot2.tidyverse.org

Wickham, H. (2019a). *Forcats: Tools for working with categorical variables (factors).* Retrieved from https://CRAN.R-project.org/package=forcats

Wickham, H. (2019b). *Stringr: Simple, consistent wrappers for common string operations.* Retrieved from https://CRAN.R-project.org/package=stringr

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L., & Müller, K. (2019). *Dplyr: A grammar of data manipulation*. Retrieved from https://CRAN.R-project.org/package=dplyr

Wickham, H., & Henry, L. (2019). *Tidyr: Tidy messy data.*

Wickham, H., Hester, J., & Francois, R. (2018). *Readr: Read rectangular text data.* Retrieved from https://CRAN.R-project.org/package=readr

Widaman, K. F., & Herringer, L. G. (1985). Iterative least squares estimates of communality: Initial estimate need not affect stabilized value. *Psychometrika*, *50*(4), 469–477.

Wirth, R., & Edwards, M. C. (2007). Item factor analysis: Current approaches and future directions. *Psychological Methods*, *12*(1), 58.

Wothke, W. (1993). Testing structural equation models. In K. Bollen & J. S. Long (Eds.), *Testing structural equation models* (pp. 256–293). A Sage Focus Edition.

Xie, Y. (2015). *Dynamic documents with R and knitr* (2nd ed.). Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://yihui.name/knitr/

Zhang, D., & Davidian, M. (2001). Linear mixed models with flexible distributions of random effects for longitudinal data. *Biometrics*, *57*(3), 795–802.

Table 1

Number of items and subjects resulting from each combination of number of factors (Factors), number of items per factor (Items/Factor), and subjects per item (Subjects/Item).

Factors	Items/Factor	Subjects/Item	Items	Sample Size
1	5	5	5	25
3	5	5	15	75
5	5	5	25	125
10	5	5	50	250
15	5	5	75	375
1	10	5	10	50
3	10	5	30	150
5	10	5	50	250
10	10	5	100	500
15	10	5	150	750
1	5	10	5	50
3	5	10	15	150
5	5	10	25	250
10	5	10	50	500
15	5	10	75	750
1	10	10	10	100
3	10	10	30	300
5	10	10	50	500
10	10	10	100	1000
15	10	10	150	1500

1	5	15	5	75
3	5	15	15	225
5	5	15	25	375
10	5	15	50	750
15	5	15	75	1125
1	10	15	10	150
3	10	15	30	450
5	10	15	50	750
10	10	15	100	1500
15	10	15	150	2250

Table 2

Coefficient estimates and standard errors for the linear mixed effects model using $\log[\mathrm{D_s}(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})]$ *as the dependent variable and estimating a random intercept for each NPD correlation matrix.*

Constant	−2.2162 (0.0004)
Subjects/Item	−0.2207 (0.0005)
Items/Factor	−0.1638 (0.0004)
Factors	−0.3134 (0.0004)
Factor Loading	−0.0354 (0.0005)
Model Error	−0.0038 (0.0004)
Smoothing Method (APA)	−0.0240 (0.0001)
Smoothing Method (BY)	−0.0724 (0.0001)
Smoothing Method (KB)	−0.0328 (0.0001)
Subjects/Item × Items/Factor	−0.0034 (0.0004)
Subjects/Item × Factors	0.0104 (0.0004)
Subjects/Item × Factor Loading	0.0041 (0.0005)
Subjects/Item × Model Error	−0.0004 (0.0004)
Subjects/Item × Smoothing Method (APA)	0.0144 (0.0001)
Subjects/Item × Smoothing Method (BY)	0.0282 (0.0001)
Subjects/Item × Smoothing Method (KB)	0.0204 (0.0001)
Items/Factor × Factors	−0.0199 (0.0004)
Items/Factor × Factor Loading	−0.0028 (0.0004)
Items/Factor × Model Error	0.0013 (0.0004)
Items/Factor × Smoothing Method (APA)	−0.0003 (0.0001)
Items/Factor × Smoothing Method (BY)	0.0128 (0.0001)

Items/Factor × Smoothing Method (KB)	0.0001 (0.0001)
Factors × Factor Loading	0.0229 (0.0004)
Factors × Model Error	0.0007 (0.0004)
Factors × Smoothing Method (APA)	0.0016 (0.0001)
Factors × Smoothing Method (BY)	−0.0046 (0.0001)
Factors × Smoothing Method (KB)	−0.0001 (0.0001)
Factor Loading × Model Error	0.0060 (0.0004)
Factor Loading × Smoothing Method (APA)	−0.0068 (0.0001)
Factor Loading × Smoothing Method (BY)	0.0193 (0.0001)
Factor Loading × Smoothing Method (KB)	−0.0092 (0.0001)
Model Error × Smoothing Method (APA)	−0.0022 (0.0001)
Model Error × Smoothing Method (BY)	0.0011 (0.0001)
Model Error × Smoothing Method (KB)	−0.0032 (0.0001)
AIC	-1808572.2960
BIC	-1808172.0800
Log Likelihood	904322.1480
Num. obs.	497381
Num. groups: id	124346
Var: id (Intercept)	0.0175
Var: Residual	0.0004

Table 3

Coefficient estimates and standard errors for the linear mixed effects model using $\log[RMSE(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})]$ *as the dependent variable and estimating a random intercept for each NPD correlation matrix.*

Constant	−2.2351 (0.0007)
Subjects/Item	−0.1893 (0.0007)
Items/Factor	−0.1749 (0.0005)
Factors	−0.2553 (0.0005)
Factor Loading	−0.4384 (0.0006)
Model Error	0.1042 (0.0005)
Smoothing Method (APA)	−0.0057 (0.0004)
Smoothing Method (BY)	−0.0185 (0.0004)
Smoothing Method (KB)	−0.0112 (0.0004)
Factor Extraction (ML)	0.1099 (0.0004)
Factor Extraction (PA)	0.0053 (0.0004)
Subjects/Item × Items/Factor	0.0165 (0.0006)
Subjects/Item × Factors	0.0215 (0.0006)
Subjects/Item × Factor Loading	−0.0172 (0.0007)
Subjects/Item × Model Error	0.0250 (0.0006)
Subjects/Item × Smoothing Method (APA)	0.0026 (0.0002)
Subjects/Item × Smoothing Method (BY)	0.0026 (0.0002)
Subjects/Item × Smoothing Method (KB)	0.0047 (0.0002)
Subjects/Item × Factor Extraction (ML)	0.1066 (0.0002)
Subjects/Item × Factor Extraction (PA)	−0.0003 (0.0002)
Items/Factor × Factors	−0.0051 (0.0005)

Items/Factor × Factor Loading	−0.0039 (0.0006)
Items/Factor × Model Error	0.0360 (0.0005)
Items/Factor × Smoothing Method (APA)	0.0019 (0.0002)
Items/Factor × Smoothing Method (BY)	0.0120 (0.0002)
Items/Factor × Smoothing Method (KB)	0.0018 (0.0002)
Items/Factor × Factor Extraction (ML)	0.0822 (0.0002)
Items/Factor × Factor Extraction (PA)	−0.0034 (0.0002)
Factors × Factor Loading	−0.0138 (0.0006)
Factors × Model Error	0.0419 (0.0005)
Factors × Smoothing Method (APA)	0.0009 (0.0002)
Factors × Smoothing Method (BY)	0.0001 (0.0002)
Factors × Smoothing Method (KB)	0.0005 (0.0002)
Factors × Factor Extraction (ML)	−0.0900 (0.0002)
Factors × Factor Extraction (PA)	−0.0035 (0.0002)
Factor Loading × Model Error	−0.0466 (0.0006)
Factor Loading × Smoothing Method (APA)	0.0002 (0.0002)
Factor Loading × Smoothing Method (BY)	0.0211 (0.0002)
Factor Loading × Smoothing Method (KB)	−0.0008 (0.0002)
Factor Loading × Factor Extraction (ML)	0.2040 (0.0002)
Factor Loading × Factor Extraction (PA)	−0.0033 (0.0002)
Model Error × Smoothing Method (APA)	0.0001 (0.0002)
Model Error × Smoothing Method (BY)	0.0010 (0.0002)
Model Error × Smoothing Method (KB)	−0.0002 (0.0002)
Model Error × Factor Extraction (ML)	−0.0342 (0.0002)
Model Error × Factor Extraction (PA)	−0.0007 (0.0002)
Smoothing Method (APA) × Factor Extraction (ML)	0.0057 (0.0005)

Smoothing Method (BY) × Factor Extraction (ML)	0.0186 (0.0005)
Smoothing Method (KB) × Factor Extraction (ML)	0.0112 (0.0005)
Smoothing Method (APA) × Factor Extraction (PA)	−0.0023 (0.0005)
Smoothing Method (BY) × Factor Extraction (PA)	−0.0035 (0.0005)
Smoothing Method (KB) × Factor Extraction (PA)	−0.0024 (0.0005)
AIC	-2414463.6688
BIC	-2413804.1182
Log Likelihood	1207285.8344
Num. obs.	1489425
Num. groups: id	124346
Var: id (Intercept)	0.0316
Var: Residual	0.0084

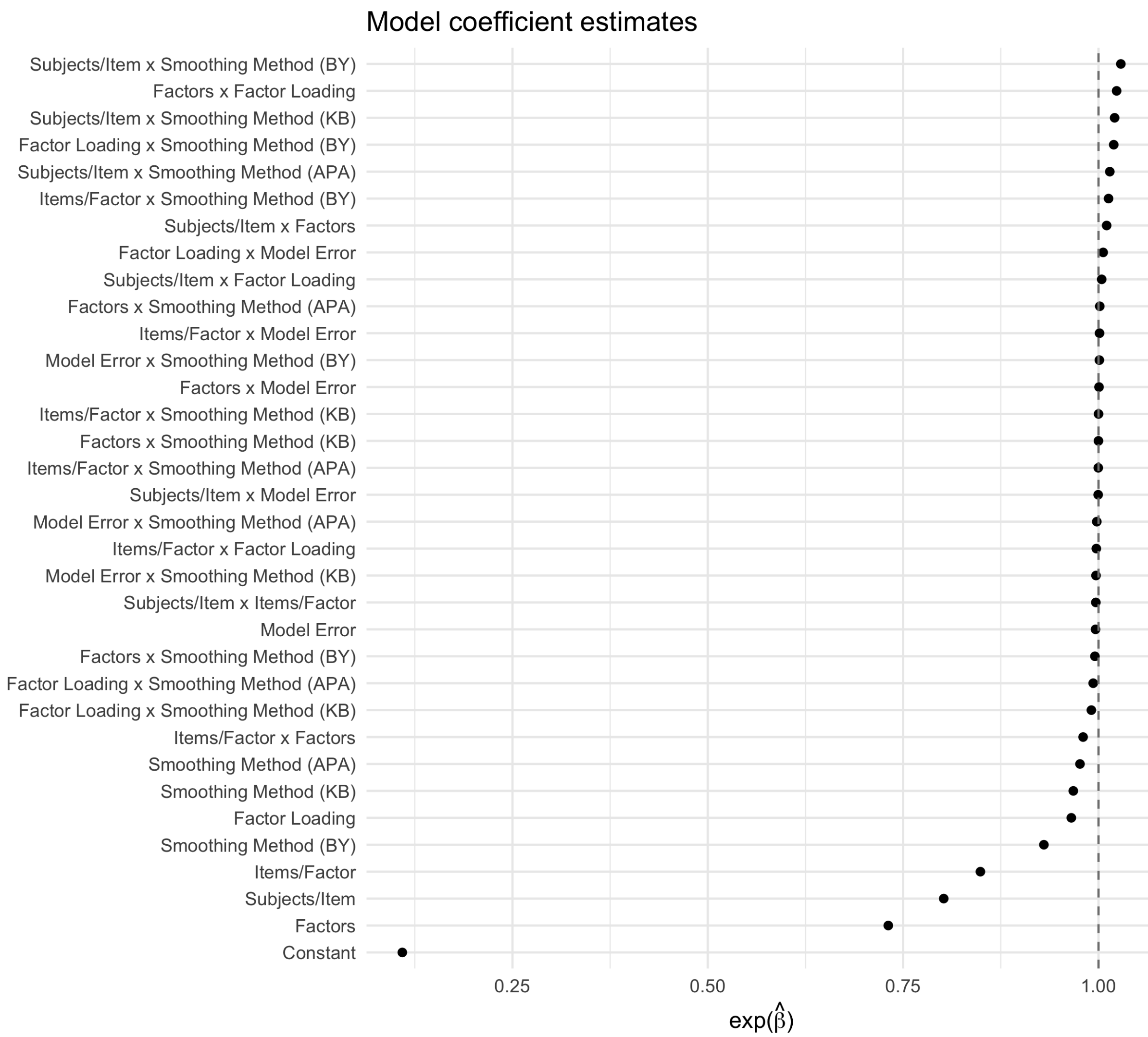

Figure 1. Exponentiated coefficient estimates for the linear mixed effects model using $\log[\mathrm{D_s}(\mathbf{R}_{\mathrm{Sm}}, \mathbf{R}_{\mathrm{Pop}})]$ as the dependent variable and estimating a random intercept for each NPD correlation matrix. The Higham (2002), Bentler-Yuan (2011) and Knol-Berger (1991) algorithms are denoted as APA, BY, and KB, respectively. The effect of the condition where no smoothing was applied is subsumed within the Constant term.

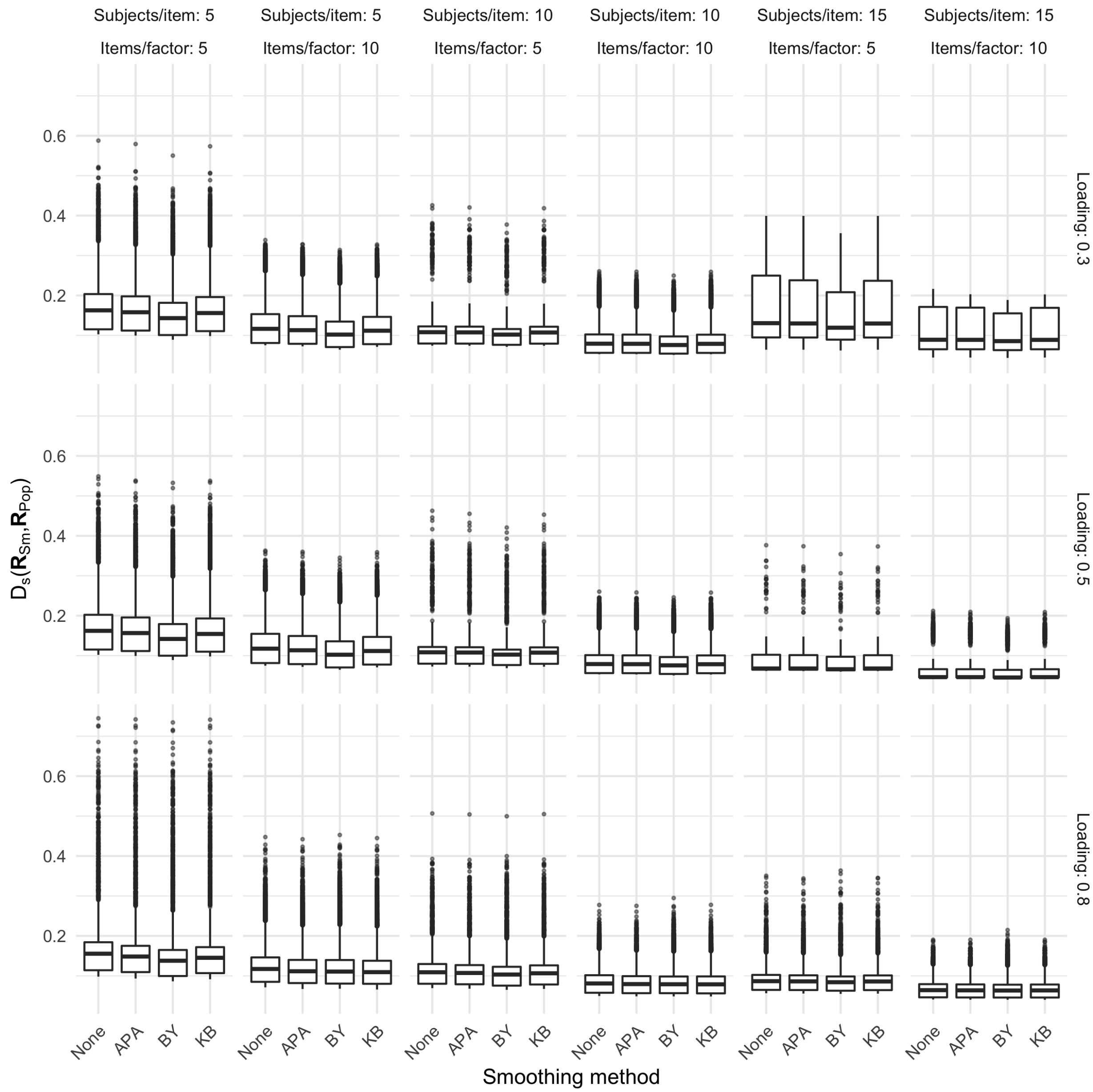

Figure 2. Scaled distance between the smoothed ($\mathbf{R}_{\mathrm{Sm}}$) and model-implied ($\mathbf{R}_{\mathrm{Pop}}$) correlation matrices for the Higham (APA; 2002), Bentler-Yuan (BY; 2011), and Knol-Berger (KB; 1991) smoothing methods and when no smoothing was applied (None).

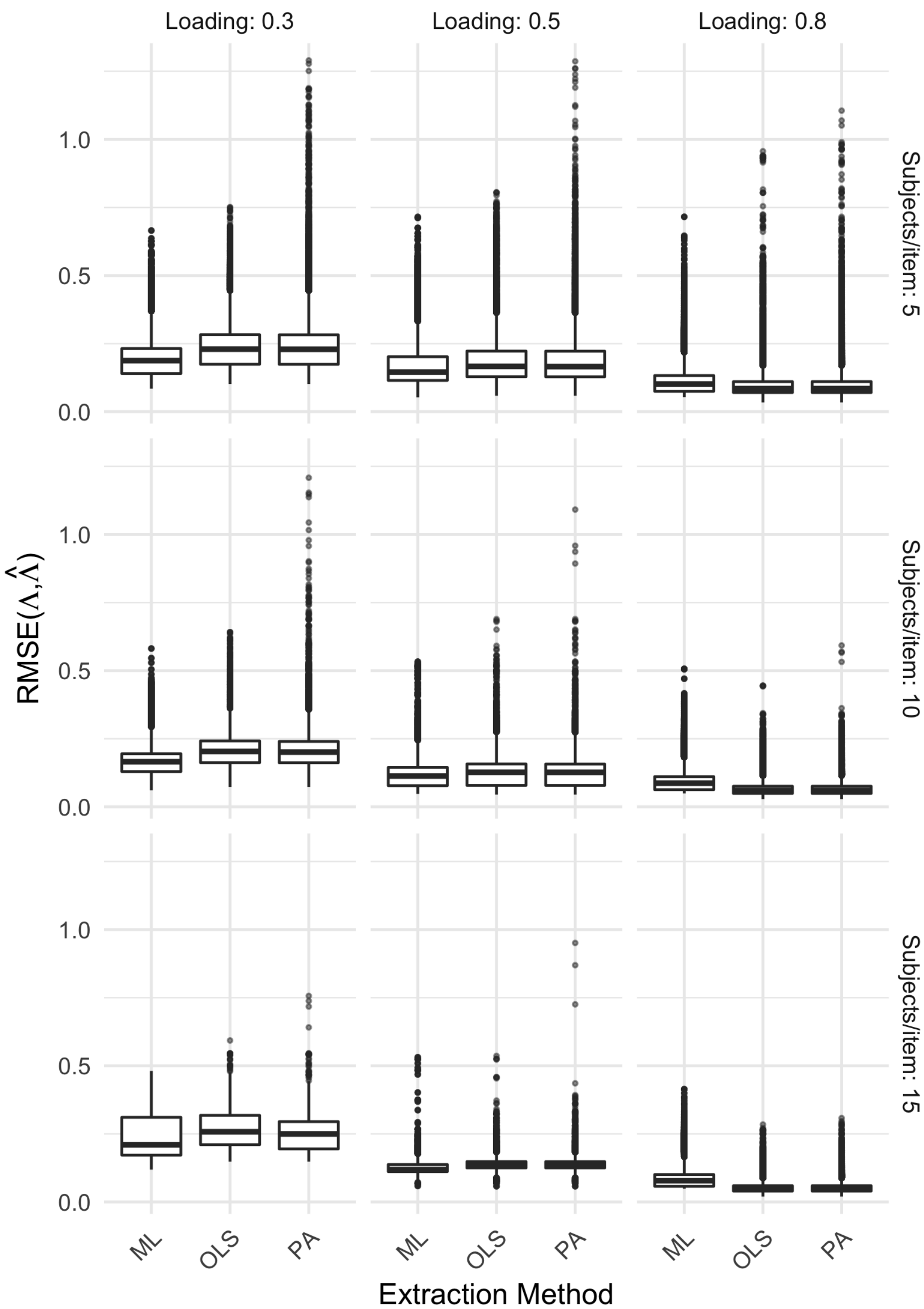

Figure 3. Box plots of RMSE($\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}$) for all combinations of factor analysis method, factor loading, and number of subjects per item. The three factor analysis methods (ordinary least squares, maximum likelihood, and principal axes) are denoted by OLS, ML, and PA, respectively.

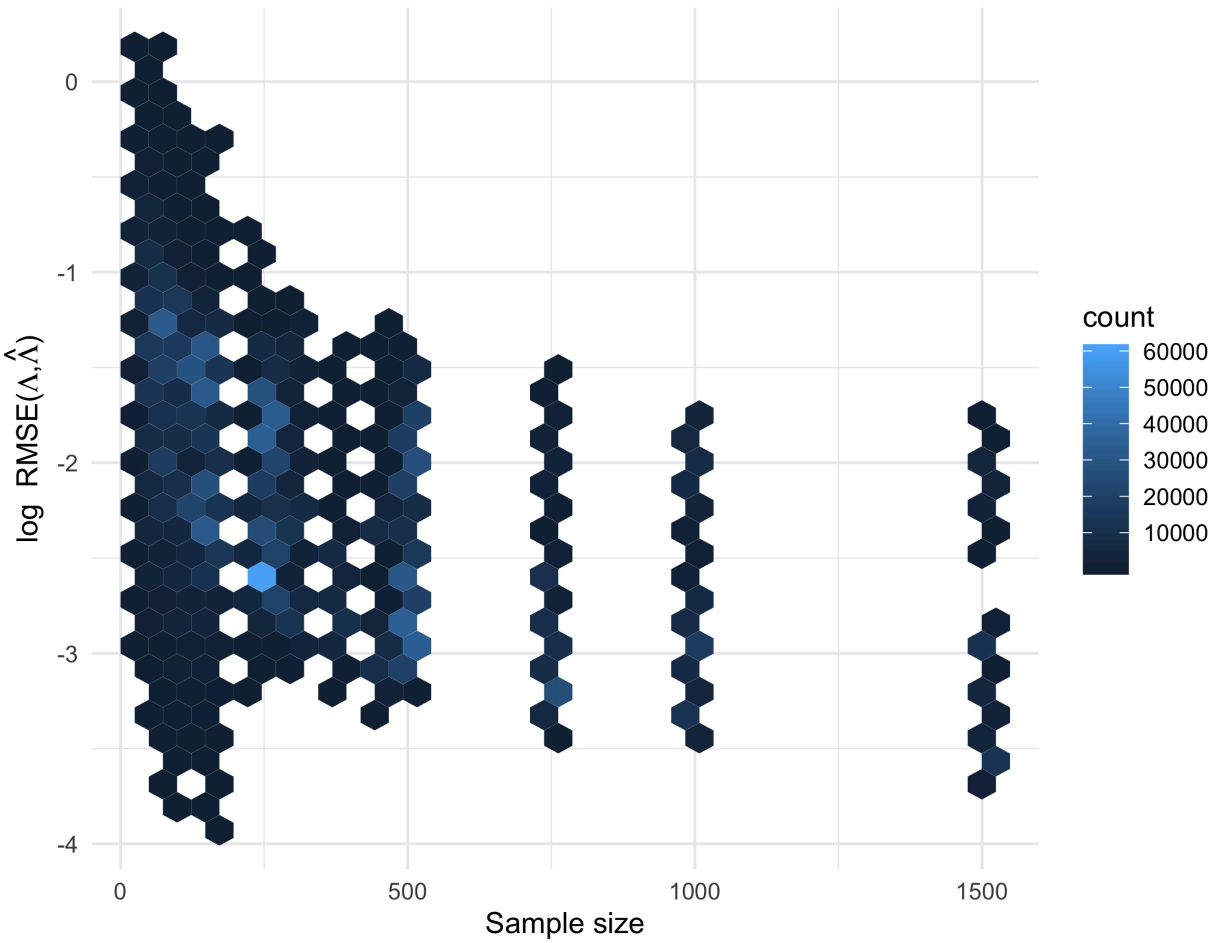

Figure 4. Log root-mean-square error (RMSE) between the true and estimated factor loading matrices as a function of sample size. Due to the large number of data points, hexagonal bins were used to group observations with the density of each hexagon represented by its color.

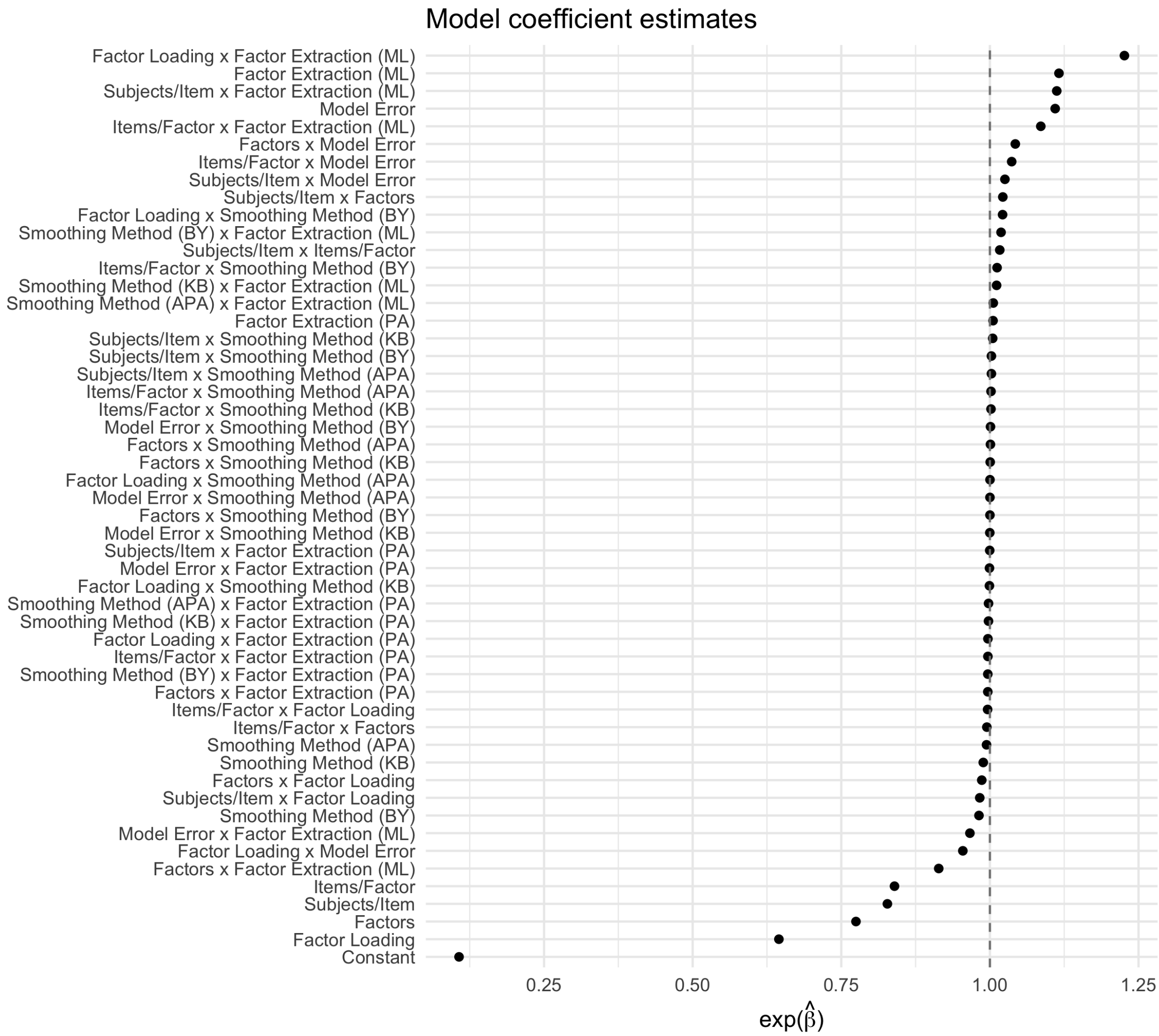

Figure 5. Exponentiated coefficient estimates for the linear mixed effects model using $\log[\text{RMSE}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}})]$ as the dependent variable and estimating a random intercept for each NPD correlation matrix. The Higham (2002), Bentler-Yuan (2011) and Knol-Berger (1991) algorithms are denoted as APA, BY, and KB, respectively. Maximum likelihood factor analysis is denoted by ML and principal axis factor analysis is denoted by PA. The effects of no smoothing and ordinary least squares factor analysis are subsumed within the Constant term.

Appendix

Regression Diagnostics

Model 1: Regression model predicting $\log \mathbf{D}_s(\mathbf{R_{Pop}}, \mathbf{R_{Sm}})$

Model 1 was a linear mixed-effects model predicting the (log) scaled distance between the smoothed and model-implied population correlation matrix and was fit using the R *lme4* package (Version 1.1.23; Bates, Mächler, Bolker, & Walker, 2015). Diagnostic plots showing standardized residuals plotted against fitted values for the model, a quantile-quantile plot for the residuals, and a quantile-quantile plot for the random intercept terms are shown in Figures A1, A2, and A3 respectively. These plots showed that some assumptions of the linear mixed-effects model seemed not to be reasonable for Model 1, even after using a log-transformation on the response variable.

First, Figure A1 showed that the variance of the residuals was not constant over the range of fitted values. In particular, there was little variation near the edges of the range of fitted values and a large amount of variation near the center of the distribution of fitted values. Therefore, the homoscedasticity assumption of the linear mixed-effects model seemed to have been violated. Moreover, Figure A2 showed that the assumption of normally-distributed errors also seemed likely to have been violated. In particular, Figure A2 showed that the distribution of residuals had heavy tails and was positively skewed. Finally, Figure A3 shows that the random effects (random intercepts) were also not normally distributed — the distribution was positively skewed. To address these violations of the model assumptions, I first attempted to fit a robust linear mixed-effects model using `rlmer` function in the R *robustlmm* package (Version 2.3; Koller, 2016). Unfortunately, the data were too large for the `rlmer` function to handle. I also tried a more complex transformation of the dependent variable (using a Box-Cox power transformation; Box & Cox, 1964), but it produced no discernable benefit compared to a log transformation.

The apparent violations of the assumptions of the linear mixed-effects model were concerning. However, inference for the fixed effects in mixed-effects models seems to be somewhat robust to these violations. In particular, Jacqmin-Gadda, Sibillot, Proust, Molina, & Thiébaut (2007) showed that inference for fixed effects was robust for non-Gaussian and heteroscedastic errors. Moreover, Jacqmin-Gadda et al. (2007) cite several studies that indicate that inference for fixed effects are also robust to non-Gaussian random effects (Butler & Louis, 1992; Verbeke & Lesaffre, 1997; Zhang & Davidian, 2001). Finally, the purpose of this analysis was to obtain estimates of the fixed effects of matrix smoothing methods (and the interactions between smoothing methods and the other design factors) on population correlation matrix recovery. Neither *p*-values nor confidence intervals were of primary concern. Therefore, the apparent violation of some model assumptions likely did not affect the main results of the study.

Model 2: Regression model predicting $\log \mathbf{RMSE}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}})$

Model 2 was a linear mixed-effects model predicting $\log \mathrm{RMSE}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}})$ fit using the R *lme4* package (Bates, Mächler, Bolker, & Walker, 2015). As with Model 1, diagnostic plots showing standardized residuals plotted against fitted values for the model, a quantile-quantile plot for the residuals, and a quantile-quantile plot for the random intercept terms are shown in Figures A1, A5, and A6 respectively. These plots indicate many of the same issues in Model 2 as were seen for Model 1. First, Figure A1 shows clear evidence of non-homogenous conditional error variance. The residual variance seemed generally to be larger for larger fitted values. Second, Figure A5 showed that the distribution of residuals for Model 2 was non-normal and similar to the distribution of the residuals from Model 1 (i.e., positively-skewed and having heavy tails). Finally, Figure A6 showed that the estimated random effects were also not normally-distributed (similar to Model 1). The distribution of random intercepts was positively-skewed with heavy tails. Alternative transformations of the dependent variable were tried but did not seem to

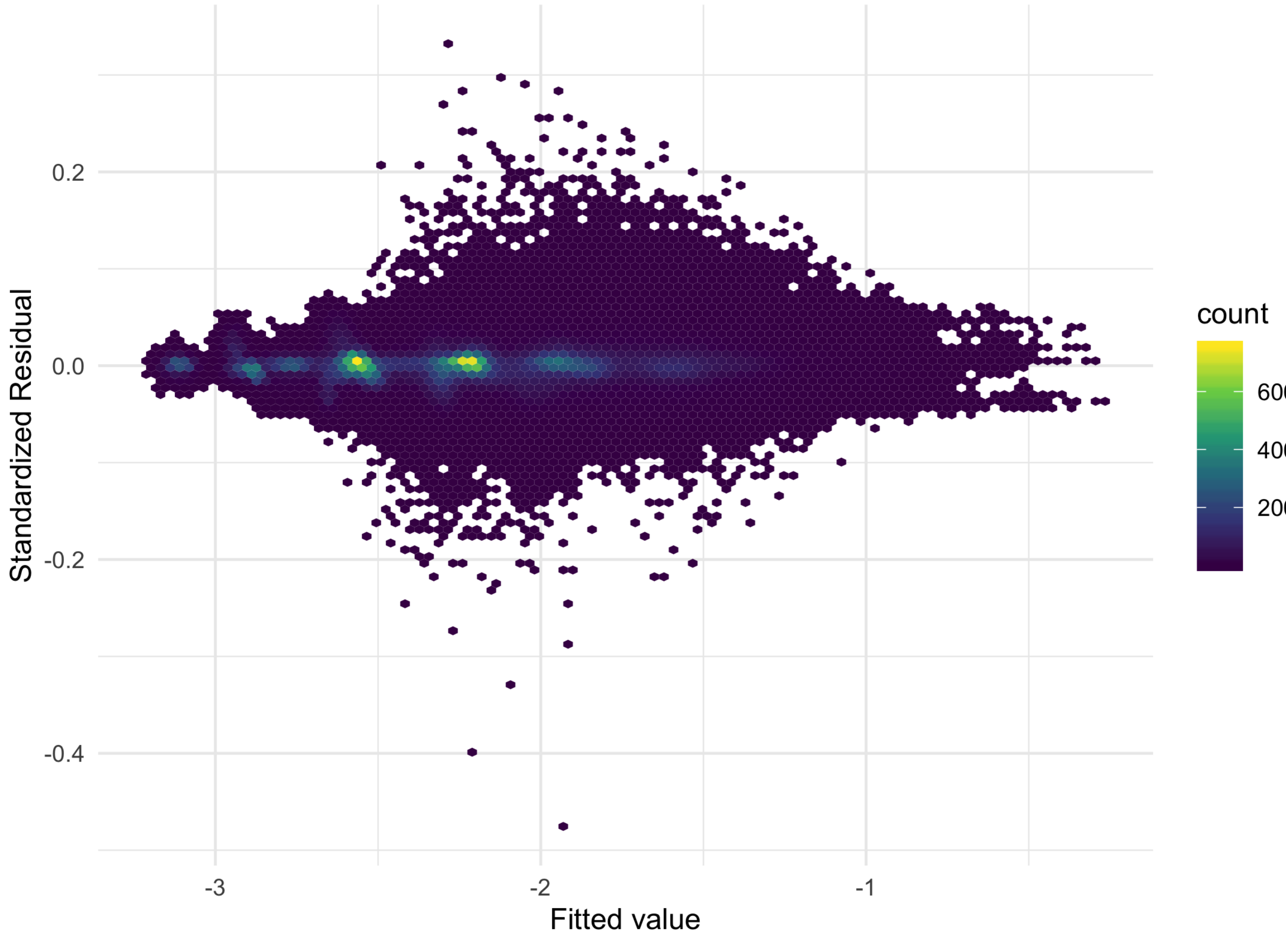

Figure A1. Standardized residuals plotted against fitted values for Model 1.

improve model fit compared to a log transformation. As with Model 1, these violations of the model assumptions are somewhat concerning and indicate that the estimated parameters—the estimated standard errors, in particular—should be treated with some degree of skepticism. However, the main results of the study are unlikely to have been affected greatly by these violations of the model assumptions.

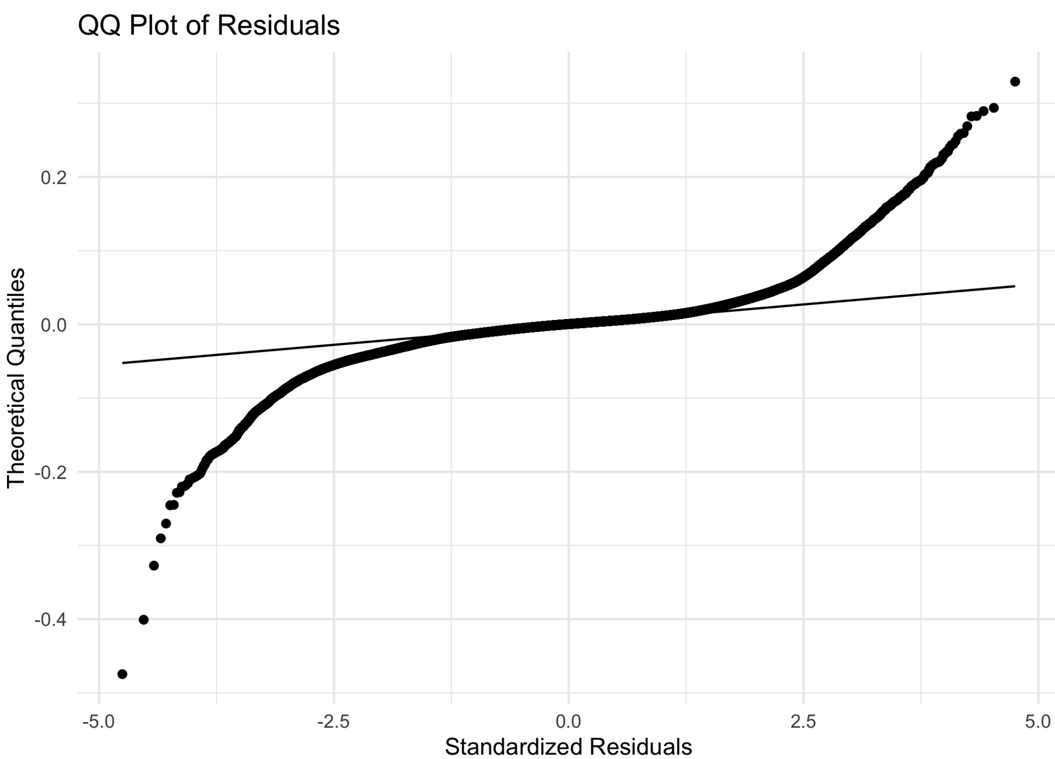

Figure A2. Quantile-quantile plot of residuals for Model 1.

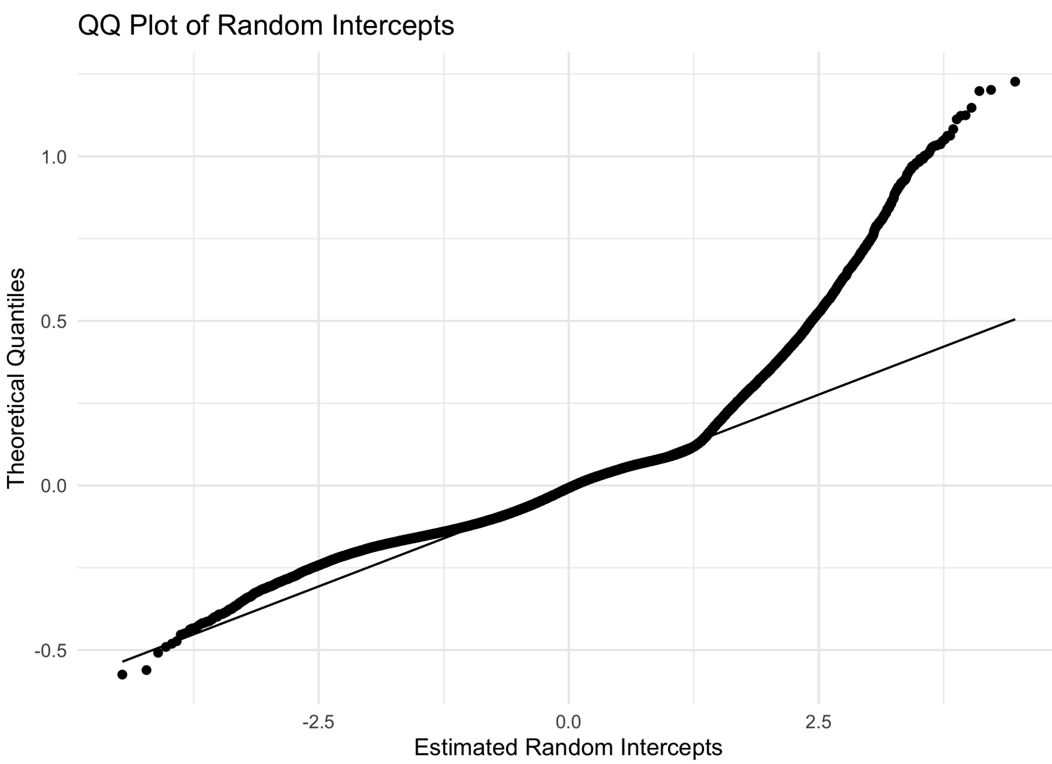

Figure A3. Quantile-quantile plot of random intercept terms for Model 1.

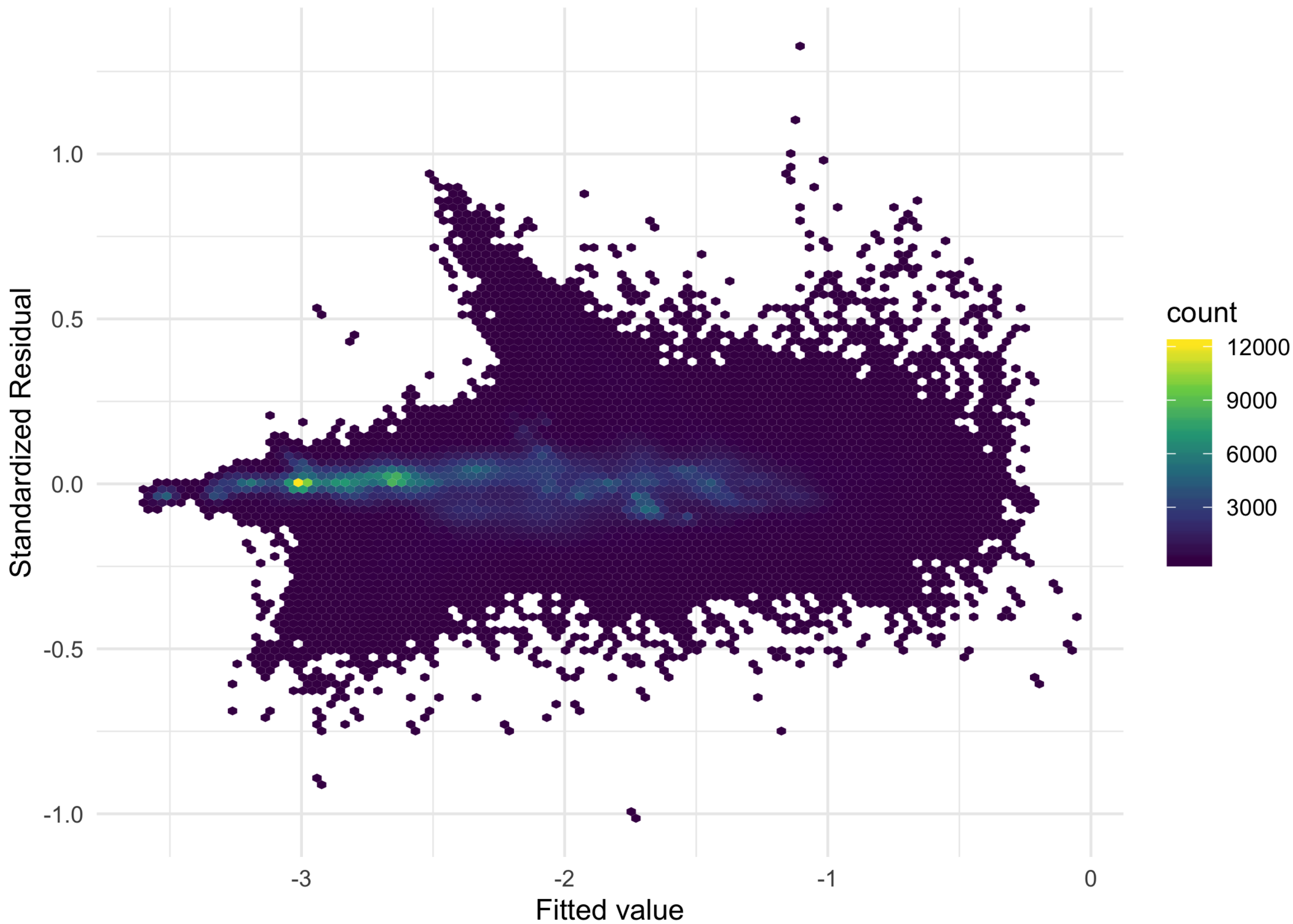

Figure A4. Standardized residuals plotted against fitted values for Model 2.

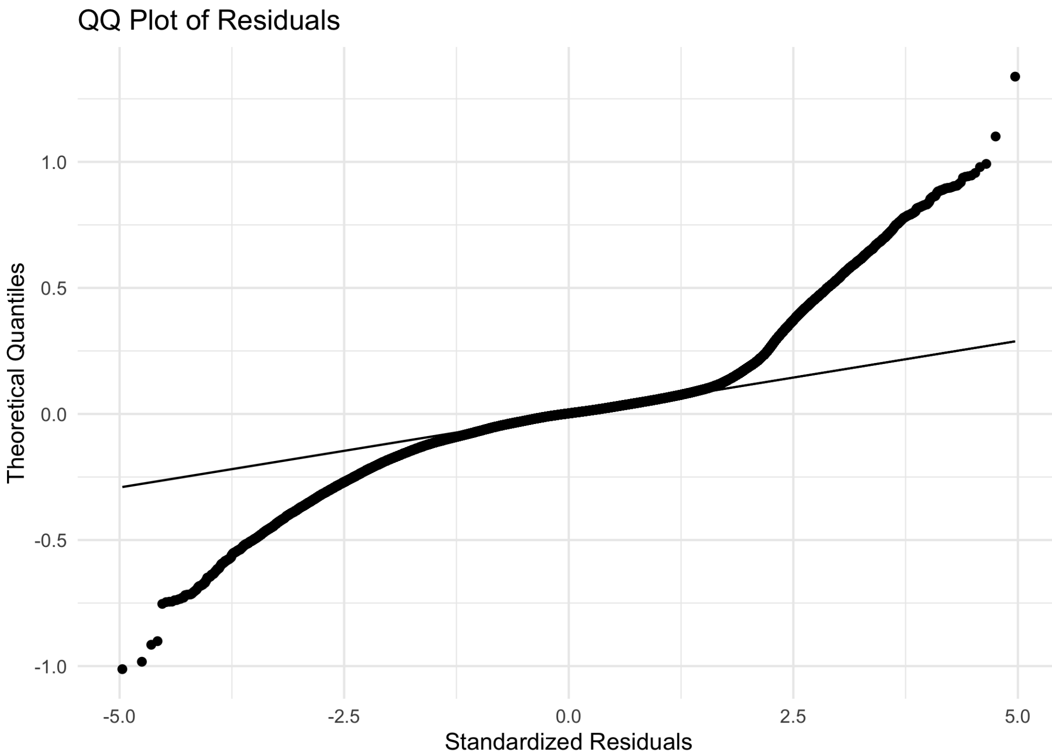

Figure A5. Quantile-quantile plot of residuals for Model 2.

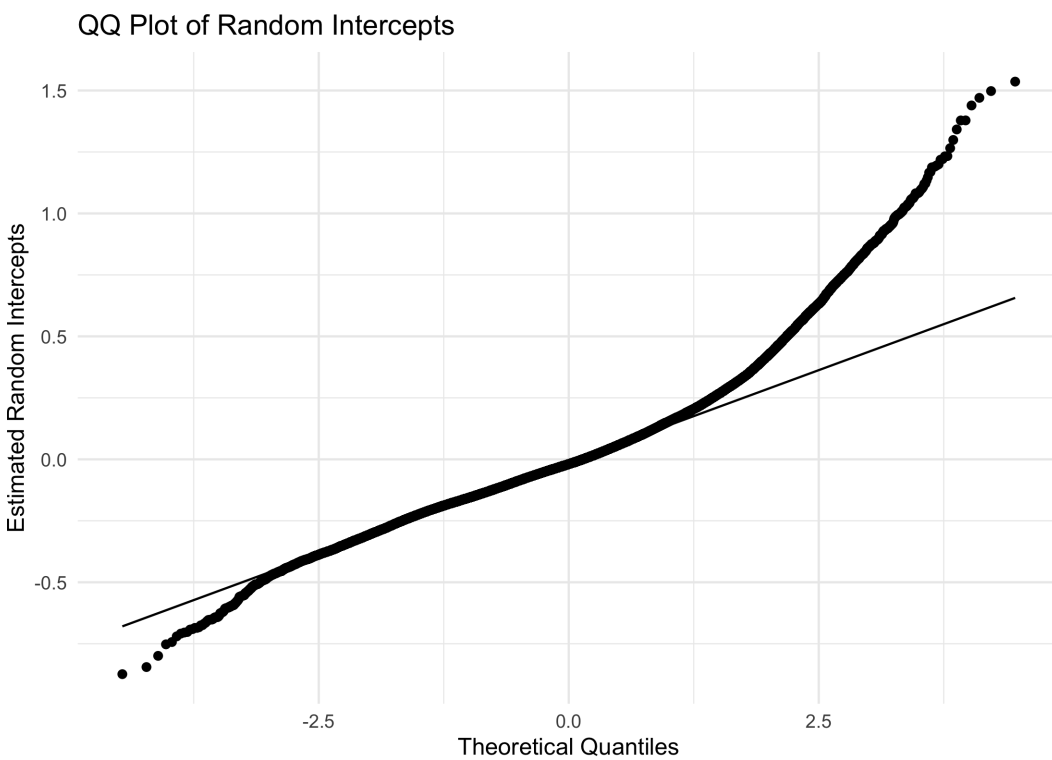

Figure A6. Quantile-quantile plot of random intercept terms for Model 2.