

¹ Factor Loading Recovery for Smoothed Tetrachoric Correlation Matrices

² Justin D. Kracht¹

³ ¹ University of Minnesota

Abstract

Researchers commonly use tetrachoric correlation matrices in item factor analysis. Unfortunately, tetrachoric correlation matrices are often indefinite (i.e., having one or more negative eigenvalues). These indefinite 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, indefinite 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 indefinite correlation matrix. However, little research has been done on the effectiveness of matrix smoothing for recovering the population correlation matrix, or for recovering factor loadings when smoothed matrices were used in exploratory factor analysis. In the present simulation study, indefinite 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 indefinite 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 indefinite correlation matrices. However, using smoothed compared to unsmoothed indefinite 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 indefinite correlation matrices and not as a statistical remedy for the root causes of matrix indefiniteness.

27 *Keywords:* matrix smoothing, item factor analysis, factor loading recovery, indefinite
28 Word count: 8,292

29 Factor Loading Recovery for Smoothed Tetrachoric Correlation Matrices

30 Tetrachoric correlation matrices (Olsson, 1979) are used to estimate the correlations

31 between the normally-distributed, continuous latent variables often assumed to underlie

32 observed binary data. Therefore, tetrachoric correlation matrices are often recommended

33 for use in item factor analysis (i.e., factor analyses with binary or polytomous data)

34 because the common linear factor model requires the assumption that outcomes are

35 continuous (Wirth & Edwards, 2007). Unfortunately, tetrachoric correlation matrices are

36 frequently indefinite, having one or more negative eigenvalues (Bock, Gibbons, & Muraki,

37 1988; Wothke, 1993). Indefinite tetrachoric correlation matrices are most likely to occur

38 when computed from data sets with many items, relatively small sample sizes, and extreme

39 item loadings and thresholds (Lorenzo-Seva & Ferrando, 2020). These Indefinite

40 tetrachoric correlation matrices are problematic because proper correlation matrices are, by

41 definition, positive semi-definite (PSD; i.e., having all eigenvalues greater than or equal to

42 zero; Wothke, 1993). Although indefinite correlation matrices resemble proper correlation

43 matrices in many ways—they are symmetric, have unit diagonals, and all off-diagonal

44 elements less than or equal to one in absolute value—it is impossible to obtain an indefinite

45 matrix of Pearson correlations from complete data. Thus, indefinite correlation matrices

46 are improper estimates of their corresponding population correlation matrices in the sense

47 that they are not included in the set of possible population correlation matrices.

48 Some researchers have suggested resolving the problem of indefinite tetrachoric

49 correlation matrices by obtaining a PSD correlation matrix that can be reasonably

50 substituted for an indefinite tetrachoric correlation matrix (e.g., Devlin, Gnanadesikan, &

51 Kettenring, 1975; Dong, 1985). This approach is often referred to as matrix smoothing,

52 and many algorithms developed for this purpose (referred to as matrix smoothing

53 algorithms, or simply smoothing algorithms) have been proposed in the psychometric

54 literature and elsewhere (Bentler & Yuan, 2011; Devlin et al., 1975; Dong, 1985; Fushiki,

55 2009; Higham, 2002; Knol & Berger, 1991; Li, Li, & Qi, 2010; Lurie & Goldberg, 1998; Qi
56 & Sun, 2006). However, despite the frequent occurrence of indefinite tetrachoric
57 correlation matrices in psychometric research (Bock et al., 1988, p. 261), the variety of
58 smoothing algorithms available, and suggestions to use matrix smoothing algorithms as a
59 remedy to indefinite tetrachoric correlation matrices (Bentler & Yuan, 2011; Knol &
60 Berger, 1991; Wothke, 1993), scant research has been done on the effectiveness of matrix
61 smoothing algorithms in the context of item factor analysis of indefinite tetrachoric
62 correlation matrices (Lorenzo-Seva & Ferrando, 2020). In one of the only published
63 comparisons of this kind, Knol and Berger (1991) investigated the effects of using
64 smoothed compared to unsmoothed correlation matrices in factor analysis and found no
65 large differences in factor loading recovery. However, this comparison was not a main focus
66 of their study and only compared a small number of indefinite matrices (10 indefinite
67 correlation matrices with 250 subjects and 15 items).

68 Additionally, few studies have compared the *relative* performance of matrix
69 smoothing algorithms in the context of factor analysis (Debelak & Tran, 2013, 2016).
70 Debelak and Tran (2013) conducted a simulation study to determine which of three matrix
71 smoothing algorithms—the Higham alternating-projections algorithm (APA; 2002), the
72 Bentler-Yuan algorithm (BY; 2011), and the Knol-Berger (KB; 1991) algorithm—most
73 often recovered the underlying dimensionality when applied to indefinite tetrachoric
74 correlation matrices prior to parallel analysis (Horn, 1965). Debelak and Tran simulated
75 binary data using a two-parameter logistic (2PL) item response theory (IRT; Birnbaum,
76 1968; de Ayala, 2013) model for one- and two-factor models with varying factor
77 correlations, item difficulties, item discriminations, numbers of items, and numbers of
78 subjects. Debelak and Tran then computed tetrachoric correlation matrices for each
79 simulated binary data set. If a tetrachoric correlation matrix was indefinite, the three
80 aforementioned smoothing algorithms were applied (resulting in three smoothed correlation
81 matrices in addition to the indefinite tetrachoric matrix). Finally, Debelak and Tran

82 conducted parallel analysis using each of the four correlation matrices to obtain estimates
83 of dimensionality. Debelak and Tran concluded that “[the] application of smoothing
84 algorithms generally improved correct identification of dimensionality when the correlation
85 between the latent dimensions was 0.0 or 0.4 in our simulations” (Debelak & Tran, 2013, p.
86 74). With respect to the relative performance of the Higham, Bentler-Yuan, and
87 Knol-Berger smoothing algorithms in this context, Debelak and Tran concluded that there
88 were “minor differences in the performance of the three smoothing algorithms used in [the]
89 study. In data sets with a clear dimensional structure... the algorithm of Bentler and Yuan
90 (2011) performed best” (Debelak & Tran, 2013, p. 74).

91 Following on these results, Debelak and Tran (2016) extended their previous
92 simulation design to evaluate the relative and absolute effectiveness of matrix smoothing
93 algorithms when applied to indefinite polychoric correlation matrices of ordered,
94 categorical (i.e., polytomous) data prior to conducting a parallel analysis. As in their
95 previous study, Debelak and Tran used the accuracy of the parallel analysis dimensionality
96 estimates as their evaluation criterion. In addition to extending their design to consider
97 polytomous data, Debelak and Tran (2016) also considered factor models with either one
98 or three major common factors and either zero or forty minor common factors. The minor
99 common factors represented the effects of model approximation error; that is, the degree of
100 model misfit inherent to mathematical models of natural phenomena in general, and
101 psychological models in particular (MacCallum & Tucker, 1991; MacCallum, Widaman,
102 Preacher, & Hong, 2001; Tucker, Koopman, & Linn, 1969). Debelak and Tran concluded
103 that the analysis of smoothed polychoric correlation matrices generally led to more
104 accurate results than the analysis of indefinite polychoric correlation matrices. Moreover,
105 they found that “methods based on the algorithms of Knol and Berger, Higham, and
106 Bentler and Yuan showed a comparable performance with regard to the accuracy to detect
107 the number of underlying major factors, with a slightly better performance of methods
108 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 dimensionality recovery) when applied to indefinite 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 the population correlation matrices than either the smoothed matrices produced by the Knol-Berger (1991) and Higham algorithms (2002) or the original indefinite 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 indefinite tetrachoric correlation matrices better estimates of their corresponding population correlation matrices than the original indefinite tetrachoric correlation matrices and, if so, which smoothing method produces the best estimates? Second, do smoothed indefinite 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 indefinite tetrachoric correlation matrices from a variety of realistic data scenarios. Before describing the simulation design, I first introduce tetrachoric correlations, the three matrix smoothing algorithms under investigation, the common factor model, and the three factor analysis algorithms included in this study.

¹³³ **Tetrachoric Correlations**

¹³⁴ A tetrachoric correlation is an estimate of the linear association between two
¹³⁵ continuous, normally-distributed latent variables, y_1^* and y_2^* , obtained using dichotomous,
¹³⁶ observed manifestations of those variables, y_1 and y_2 . The variables y_1^* and y_2^* are assumed
¹³⁷ to follow a bivariate normal distribution,

$$\begin{pmatrix} y_1^* \\ y_2^* \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix} \right],$$

¹³⁸ where r is the true correlation between y_1^* and y_2^* that is estimated by the tetrachoric
¹³⁹ correlation, \hat{r} . To compute the tetrachoric correlation, a 2×2 contingency table is first
¹⁴⁰ created using y_1 and y_2 as described in Brown and Benedetti (1977). If any of the cell
¹⁴¹ frequencies in the contingency table are zero, those elements are replaced with 0.5 and the
¹⁴² other elements adjusted to leave the marginal sums unchanged (Brown & Benedetti, 1977).
¹⁴³ The proportions of correct responses for y_1 and y_2 are represented by the marginals p_1 and
¹⁴⁴ p_2 . The standard normal deviate thresholds, τ_1 and τ_2 , used to dichotomize y_1^* and y_2^* are
¹⁴⁵ then estimated using $1 - \Phi(\hat{\tau}_1) = p_1$ and $1 - \Phi(\hat{\tau}_2) = p_2$, and solving for $\hat{\tau}_1$ and $\hat{\tau}_2$. Here,
¹⁴⁶ $\Phi(z)$ denotes the standard normal cumulative distribution function (Divgi, 1979). Because
¹⁴⁷ y_1^* and y_2^* are assumed to follow a bivariate normal distribution with correlation r , the joint
¹⁴⁸ probability of $(y_1 > \hat{\tau}_1, y_2 > \hat{\tau}_2)$ can be written as:

$$L(\hat{\tau}_1, \hat{\tau}_2, r) = \frac{1}{2\pi\sqrt{1-r^2}} \int_{\hat{\tau}_2}^{\infty} \int_{\hat{\tau}_1}^{\infty} \exp \left(-\frac{y_1^{*2} + y_2^{*2} - 2ry_1^*y_2^*}{2(1-r^2)} \right) dy_1^* dy_2^*. \quad (1)$$

¹⁴⁹ An estimate of r can then be obtained by setting Equation (1) equal to p_{11} (the
¹⁵⁰ observed proportion of correct responses for both y_1 and y_2) and solving for r using an
¹⁵¹ iterative procedure. In particular, the Newton-Raphson method can be used to obtain
¹⁵² successive approximations of r given an initial estimate, \hat{r}_0 :

$$\hat{r}_{i+1} = \hat{r}_i - \frac{L(\hat{\tau}_1, \hat{\tau}_2, \hat{r}_i) - p_{11}}{L'(\hat{\tau}_1, \hat{\tau}_2, \hat{r}_i)}, \quad (2)$$

153 where $L'(\hat{\tau}_1, \hat{\tau}_2, \hat{r}_i)$ is the first derivative of $L(\hat{\tau}_1, \hat{\tau}_2, \hat{r}_i)$ (Divgi, 1979). Iteration continues
 154 until convergence is achieved (when $\hat{r}_{i+1} - \hat{r}_i < \delta$ for some small value of δ) or until some
 155 maximum number of iterations occur. For p dichotomous variables, the $p \times p$ symmetric
 156 matrix \mathbf{R}_{Tet} is called the tetrachoric correlation matrix. The \mathbf{R}_{Tet} matrix has a unit
 157 diagonal and has off-diagonal elements consisting of pairwise tetrachoric correlation
 158 coefficients \hat{r}_{jk} , $j, k \in \{1, \dots, p\}$. Just as the tetrachoric correlation \hat{r}_{jk} estimates r_{jk} , the
 159 tetrachoric correlation matrix \mathbf{R}_{Tet} estimates the $p \times p$ population correlation matrix,
 160 \mathbf{R}_{Pop} , which is symmetric with off-diagonal elements r_{jk} , and a unit diagonal.

161 Matrix Smoothing Algorithms

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

173 Specifically, Higham's algorithm (2002) consists of alternating projection functions,
 174 P_U , the projection onto \mathcal{U} , and P_S , the projection onto \mathcal{S} . For some symmetric $\mathbf{A} \in \mathbb{R}^{p \times p}$

¹⁷⁵ with elements a_{ij} ,

$$P_U(\mathbf{A}) = (p_{ij}), p_{ij} = \begin{cases} a_{ij}, & i \neq j \\ 1, & i = j. \end{cases} \quad (3)$$

¹⁷⁶ Stated simply, $P_U(\mathbf{A})$ replaces all elements of the diagonal of \mathbf{A} with ones. The projection
¹⁷⁷ onto \mathcal{S} is less straightforward. Higham (2002) outlines the steps as follows. For some
¹⁷⁸ symmetric, indefinite matrix $\mathbf{A} \in \mathbb{R}^{p \times p}$, let $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^T$ be the eigendecomposition of \mathbf{A} ,
¹⁷⁹ where \mathbf{V} is the orthonormal matrix of eigenvectors and $\Lambda = \text{diag}(\lambda_i)$ is a diagonal matrix
¹⁸⁰ with the eigenvalues of \mathbf{A} , $\lambda_i, i \in \{1, \dots, p\}$, ordered from largest to smallest on the
¹⁸¹ diagonal ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p, \lambda_p < 0$). Also let $\Lambda_+ = \text{diag}(\max(\lambda_i, 0))$, where the diag
¹⁸² operator takes a vector and returns a diagonal matrix with the input vector on the
¹⁸³ diagonal. Then the projection of \mathbf{A} onto \mathcal{S} can be written as

$$P_S(\mathbf{A}) = \mathbf{V}\Lambda_+\mathbf{V}^T. \quad (4)$$

¹⁸⁴ Starting with $\mathbf{A} = \mathbf{R}_-$, \mathbf{R}_{APA} can be obtained by repeatedly applying the operation
¹⁸⁵ $\mathbf{A} \leftarrow P_U(P_S(\mathbf{A}))$ until convergence occurs or until some maximum number of iterations is
¹⁸⁶ reached (Higham, 2002, p. 337).¹

¹⁸⁷ **Bentler-Yuan Algorithm (BY; 2011).** The Bentler-Yuan (2011) smoothing
¹⁸⁸ algorithm is based on minimum-trace factor analysis (MTFA; Bentler, 1972; Jamshidian &
¹⁸⁹ Bentler, 1998). MTFA seeks to find optimal communality estimates such that unexplained
¹⁹⁰ common variance is minimized. This minimization is subject to two constraints. First, the
¹⁹¹ diagonal matrix of unique variances is constrained to be positive semidefinite (PSD).
¹⁹² Second, the matrix formed by replacing the diagonal elements of the observed covariance
¹⁹³ matrix with the estimated communalities is also constrained to be PSD. In contrast with
¹⁹⁴ the Higham algorithm (2002), the Bentler-Yuan algorithm does not seek to minimize some

¹ This is a somewhat simplified explanation of Higham's algorithm. The full algorithm includes a correction to each projection (see Higham, 2002, Algorithm 3.3, p. 337).

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}_- corresponding to these Heywood cases to produce a smoothed, PSD correlation matrix, \mathbf{R}_{BY} . More specifically, the algorithm first conducts an MTFA using \mathbf{R}_- . Using the results of the MTFA, a diagonal matrix, \mathbf{H} is constructed containing the estimated communalities as diagonal elements. Next, another diagonal matrix, Δ^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 $\mathbf{R}_{BY} = \Delta \mathbf{R}_0 \Delta + \mathbf{I}$ is obtained, where \mathbf{R}_0 is \mathbf{R}_- with diagonal elements replaced by zeroes and \mathbf{I} is an identity matrix that ensures that \mathbf{R}_{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 indefinite correlation matrix enough for the matrix to become PSD. To help with this non-convergence, I used the modified Bentler-Yuan algorithm implementation provided by the `smoothBY()` function in the R *fungible* package (Waller, 2019) to adaptively select an appropriate k . The k parameter was initialized at $k = 0.999$ and decreased by 0.001 until the algorithm produced a PSD correlation matrix or $k = 0$.²

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}_- are replaced with some small positive value. The first step in the Knol-Berger algorithm is to compute the eigendecomposition of the $p \times p$ indefinite correlation matrix, as defined in the previous section. Next, a matrix Λ_+ is created by setting all negative elements of Λ equal to some user-specified small,

² Bentler and Yuan suggest using $k = 0.96$ (Bentler & Yuan, 2011, p. 120) but suggest that the precise value of k does not matter a great deal as long as k is marginally less than one.

²²⁰ positive constant. Finally, a smoothed, PSD correlation matrix, \mathbf{R}_{KB} , is constructed by
²²¹ replacing Λ with Λ_+ in the eigendecomposition of \mathbf{R}_- and then scaling to ensure a unit
²²² diagonal and that the absolute value of all off-diagonal elements is less than or equal to one:

$$\mathbf{R}_{KB} = [\text{dg}(\mathbf{V}\Lambda_+\mathbf{V}')]^{-1/2}\mathbf{V}\Lambda_+\mathbf{V}'[\text{dg}(\mathbf{V}\Lambda_+\mathbf{V}')]^{-1/2}, \quad (5)$$

²²³ where the dg operator is defined such that $\text{dg}(\mathbf{A})$ returns a diagonal matrix
²²⁴ containing the diagonal elements of \mathbf{A} (Magnus & Neudecker, 2019, p. 6).

²²⁵ The Common Factor Model

²²⁶ The linear factor analysis model is used to describe the variance of each observed
²²⁷ variable in terms of the contributions of a small number of latent common factors and a
²²⁸ specific factor unique to that variable (Wirth & Edwards, 2007). In the common factor
²²⁹ model, the population correlation matrix, \mathbf{P} , can be expressed as:

$$\mathbf{P} = \mathbf{F}\Phi\mathbf{F}' + \boldsymbol{\Theta}^2, \quad (6)$$

²³⁰ where \mathbf{P} is a $p \times p$ population correlation matrix for p observed variables, \mathbf{F} is a $p \times m$
²³¹ factor loading matrix for m common factors, Φ is an $m \times m$ matrix of correlations between
²³² the m common factors, and $\boldsymbol{\Theta}^2$ is a $p \times p$ diagonal matrix containing the unique variances.

²³³ Although the common factor analysis model represented in Equation (6) is often
²³⁴ useful, many authors have remarked that it constitutes an oversimplification of the
²³⁵ complex processes that generate real, observed data (Cudeck & Henly, 1991; MacCallum &
²³⁶ Tucker, 1991; MacCallum et al., 2001). Tucker et al. (1969) suggested that the lack-of-fit
²³⁷ between the common factor model and the complex processes underlying real data could be
²³⁸ represented by modeling a large number of minor common factors of small effect. The

²³⁹ model Tucker et al. (1969) proposed can be written as:

$$\mathbf{P} = \mathbf{F}\Phi\mathbf{F}' + \boldsymbol{\Theta}^2 + \mathbf{W}\mathbf{W}', \quad (7)$$

²⁴⁰ where \mathbf{W} is a $p \times q$ matrix containing factor loadings for the $q \gg m$ minor factors (Briggs
²⁴¹ & MacCallum, 2003, p. 32). Given our expectation that the common factor model is not a
²⁴² perfect representation of any real-world data-generating process we might wish to
²⁴³ represent, Equation (7) is arguably preferable to Equation (6) for simulating realistic data
²⁴⁴ (Briggs & MacCallum, 2003; Hong, 1999).

²⁴⁵ Factor Extraction Methods

²⁴⁶ Various factor extraction methods have been proposed for estimating item factor
²⁴⁷ loadings, factor correlations, and unique item variances. One purpose of this study was to
²⁴⁸ determine whether the effects of matrix smoothing method on factor loading recovery differ
²⁴⁹ depending on which factor extraction method is used. To that end, three of the most
²⁵⁰ commonly-used factor extraction methods (Fabrigar, Wegener, MacCallum, & Strahan,
²⁵¹ 1999) were used in the present simulation study: principal axis (PA), ordinary least-squares
²⁵² (OLS), and maximum-likelihood (ML).

²⁵³ **Principal Axis Factor Analysis.** Principal axis (PA) factor analysis is
²⁵⁴ conceptually similar to principal components analysis (PCA). Whereas PCA seeks to find a
²⁵⁵ low-dimensional approximation of the full observed correlation matrix, PA seeks to find a
²⁵⁶ low-dimensional approximation of the reduced correlation matrix, \mathbf{R}_* (i.e., the observed
²⁵⁷ correlation matrix, \mathbf{R} , with communalities on the diagonal). Because the true
²⁵⁸ communalities are unknown, principal axis factor analysis starts by using estimated
²⁵⁹ communalities to form \mathbf{R}_* .³ The eigenvalues of \mathbf{R}_* are then taken to be the updated

³ 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

260 communality estimates. These updated estimates replace the previous estimates on the
 261 diagonal of \mathbf{R}_* and the procedure iterates until the sum of the differences between the
 262 communality estimates from the current and previous iterations is less than some small
 263 convergence criterion.

264 **Ordinary Least-Squares Factor Analysis.** The ordinary least-squares factor
 265 analysis method (OLS; also known as “minres”; Comrey, 1962) seeks to minimize the sum
 266 of squared differences between the sample correlation matrix, \mathbf{R} , and $\hat{\mathbf{P}} = \hat{\mathbf{F}}\hat{\Phi}\hat{\mathbf{F}}' + \hat{\Theta}^2$, the
 267 correlation matrix implied by the estimated factor model corresponding to Equation (6).
 268 The OLS discrepancy function can then be written as

$$F_{OLS}(\mathbf{R}, \hat{\mathbf{P}}) = \frac{1}{2} \text{tr} [(\mathbf{R} - \hat{\mathbf{P}})^2], \quad (8)$$

269 where tr is the trace operator (Magnus & Neudecker, 2019, p. 11) and $\text{tr} [(\mathbf{R} - \hat{\mathbf{P}})^2]$ is the
 270 trace (sum of the diagonal elements) of the matrix formed by $(\mathbf{R} - \hat{\mathbf{P}})^2$. OLS does not give
 271 additional weight to residuals corresponding to large correlations and requires no
 272 assumptions about the population distributions of the variables (Briggs & MacCallum,
 273 2003).

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

$$F_{ML}(\mathbf{R}, \hat{\mathbf{P}}) = \log |\hat{\mathbf{P}}| - \log |\mathbf{R}| + \text{tr}(\mathbf{S}\hat{\mathbf{P}}^{-1}) - p. \quad (9)$$

(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 & Herring, 1985).

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

286 **Simulation Procedure**

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

297 In the first step of the simulation study, I generated random sets of binary data from
298 a variety of orthogonal factor models with varying numbers of major common factors
299 (Factors $\in \{1, 3, 5, 10\}$). Using the method of Tucker et al. (1969), I also incorporated the
300 effects of model approximation error into the data by including 150 minor common factors
301 in each population model. In total, these 150 minor common factors accounted for 0%,
302 10%, or 30% ($v_E \in \{0, .1, .3\}$) of the uniqueness variance of the error-free model (i.e., the
303 model with only the major common factors). These conditions were chosen to represent
304 models with perfect, good, or moderate model fit, resembling the conditions used by Briggs

305 and MacCallum (2003). These three levels of model error variance ensured that both ideal
 306 ($v_E = 0$) and more empirically-plausible levels of model error variance ($v_E \in \{.1, .3\}$) were
 307 considered in this study.

308 In addition to systematically varying the number of major factors and the proportion
 309 of uniqueness variance accounted for by model approximation error, I also varied the
 310 number of factor indicators (i.e., items loading on each factor; Items/Factor $\in \{5, 10\}$), and
 311 the number of subjects per item (Subjects/Item $\in \{5, 10, 15\}$). The total numbers of items
 312 (p) and sample sizes (N) for each factor number condition can be found in Table 1. Each
 313 item loaded on only one factor and item factor loadings were uniformly fixed at one of
 314 three levels (Loading $\in \{.3, .5, .8\}$). Though “rules-of-thumb” for factor loadings vary, Hair,
 315 Black, Babin, and Anderson (2018, p. 151) suggest that “[f]actor loadings in the range of
 316 ± 0.30 to ± 0.40 are considered to meet the minimal level for interpretation of structure”,
 317 and “[l]oadings ± 0.50 or greater are considered practically significant.” Moreover, factor
 318 loadings of ± 0.8 are considered to be high (MacCallum et al., 2001). Thus, the three factor
 319 loadings investigated in this study were chosen to represent low, moderate, and high levels
 320 of factor salience.

321 The combinations of the independent variables specified above resulted in a
 322 fully-crossed design with 4 (Factors) \times 3 (Model Error, v_E) \times 2 (Items/Factor) \times 3
 323 (Subjects/Item) \times 3 (Loading) = 216 unique conditions. For each of these conditions, the
 324 **simFA()** function in the R (Version 3.6.2; R Core Team, 2019)⁴ *fungible* package (Version

⁴ 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), *lattice2exp* (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,

³²⁵ 1.95.4.8; Waller, 2019) was used to generate 1,000 random sets of binary data.

³²⁶ **Data Generation**

³²⁷ Each data set in the simulation was generated as follows. First, a model-implied
³²⁸ population correlation matrix, \mathbf{R}_{Pop} , was generated using

$$\mathbf{R}_{\text{Pop}} = \mathbf{F}\Phi\mathbf{F}' + \Theta^2 + \mathbf{W}\mathbf{W}'. \quad (10)$$

³²⁹ Here, \mathbf{F} denotes a $p \times m$ matrix of major factor loadings with simple structure such that
³³⁰ each factor had exactly p/m salient loadings (fixed at the value indicated by the level of
³³¹ Loading) and all other loadings fixed at zero. Because only orthogonal models were
³³² considered in this study, the factor correlation matrix Φ was an $m \times m$ identity matrix.

³³³ The $p \times q$ matrix of minor common factor loadings, \mathbf{W} , was constructed in multiple
³³⁴ steps. First, a $p \times q$ provisional matrix, \mathbf{W}^* , was generated such that the i th column of \mathbf{W}^*
³³⁵ consisted of p independent samples from $\mathcal{N}(0, (1 - \epsilon)^{2(i-1)})$ where $\epsilon \in [0, 1]$ was a
³³⁶ user-specified constant. The value of ϵ determined how the minor common factor (error)
³³⁷ variance was distributed. Values of ϵ close to zero resulted in the error variance being
³³⁸ spread relatively equally among the minor common factors. Values of ϵ close to one
³³⁹ resulted in error variance primarily being distributed to the first minor factor, with the
³⁴⁰ remaining variance distributed to the other minor factors in a decreasing geometric
³⁴¹ sequence. To ensure that the minor common factors accounted for the specified proportion
³⁴² of uniqueness variance (denoted as v_E), \mathbf{W}^* was scaled to create \mathbf{W} . This scaling was done
³⁴³ in several steps. First, a diagonal matrix $\Theta_{p \times p}^*$ was created such that

$$\Theta^* = \mathbf{I}_p - \text{dg}(\mathbf{F}\mathbf{F}'), \quad (11)$$

2019), *tidyverse* (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).

³⁴⁴ where $\text{dg}(\mathbf{FF}')$ is to be read as the diagonal matrix formed from the diagonal entries in \mathbf{FF}'
³⁴⁵ and \mathbf{I}_p denotes a $p \times p$ identity matrix. Then the matrix \mathbf{W} was formed using

$$\mathbf{W} = (\text{dg}(\mathbf{W}^*\mathbf{W}'')^{-1}\boldsymbol{\Theta}^*v_E)^{1/2}\mathbf{W}^*. \quad (12)$$

³⁴⁶ This process ensured that the q minor common factors accounted for the specified
³⁴⁷ proportion of the variance not accounted for by the major common factors. The \mathbf{W} matrix
³⁴⁸ was then used to create the diagonal matrix of unique variances,
³⁴⁹ $\boldsymbol{\Theta}^2 = \mathbf{I}_p - \text{dg}(\mathbf{FF}' + \mathbf{WW}')$. The \mathbf{F} , $\boldsymbol{\Theta}^2$, and \mathbf{W} matrices were then used to construct
³⁵⁰ population correlation matrix, \mathbf{R}_{Pop} , as shown in Equation (10).

³⁵¹ Having specified the elements of the population common factor model, the next step
³⁵² in the data-generation procedure was to draw a sample correlation matrix, \mathbf{R} , (for a given
³⁵³ sample size, N) from \mathbf{R}_{Pop} using the method of Kshirsagar (1959; see also Browne, 1968).
³⁵⁴ The sample correlation matrix was then used to generate a matrix of continuous data,
³⁵⁵ $\mathbf{X}_{N \times p} = (X_1, \dots, X_N)'$, where $X \sim \mathcal{N}_p(\mathbf{0}_p, \mathbf{R})$. To obtain binary responses from the
³⁵⁶ continuous data, items were assigned classical item difficulties (d ; i.e., the expected
³⁵⁷ proportion of correct responses, Crocker & Algina, 1986) at equal intervals between 0.15
³⁵⁸ and 0.85. For example, items in a five-item data set were assigned classical item difficulties
³⁵⁹ of .150, .325, .500, .675, and .850. The classical item difficulties were used to obtain
³⁶⁰ threshold values, t , such that $1 - \Phi(t) = d$. Using these thresholds, the continuous data
³⁶¹ were converted to binary data. If a data set had any homogeneous item response vectors
³⁶² (i.e., had one or more items with zero variance), the data set was discarded and a new
³⁶³ sample of data was generated until all items had non-homogeneous response vectors.
³⁶⁴ Homogeneous response vectors were not allowed because such response vectors can lead to
³⁶⁵ poorly-estimated tetrachoric correlations (Brown & Benedetti, 1977).

³⁶⁶ Next, a tetrachoric correlation matrix was computed for each simulated binary data
³⁶⁷ set. Tetrachoric correlation matrices were calculated using the `tetcor()` function in the R

368 *fungible* package (Waller, 2019), which computes maximum likelihood tetrachoric
369 correlation coefficients (Brown & Benedetti, 1977; Olsson, 1979). If a tetrachoric correlation
370 matrix was indefinite, the Higham (2002), Bentler-Yuan (2011), and Knol-Berger (1991)
371 matrix smoothing algorithms were applied to the indefinite tetrachoric correlation matrix
372 to produce three smoothed, PSD correlation matrices. Matrix smoothing was done using
373 the `smoothAPA()`, `smoothBY()`, and `smoothKB()` implementations of the Higham (2002),
374 Bentler-Yuan (2011), and Knol-Berger (1991) algorithms in the *fungible* package.

375 In the final step of the simulation procedure, three exploratory factor analysis
376 algorithms (principal axis [PA], ordinary least squares [OLS], and maximum likelihood
377 [ML]) were applied to each of the indefinite tetrachoric correlation matrices and the PSD,
378 smoothed correlation matrices. Because ML does not work with indefinite correlation or
379 covariance matrices as input, ML was conducted on the Pearson correlation matrix (rather
380 than the indefinite tetrachoric correlation matrix) when no smoothing was applied. Each of
381 the factor solutions were rotated using a quartimin rotation (Carroll, 1957; Jennrich, 2002)
382 and aligned to match the corresponding population factor loading matrix such that the
383 least squares discrepancy between the matrices was minimized. The alignment step
384 ensured that the elements of each estimated factor loading matrix were matched (in order
385 and sign) to the elements of the corresponding population factor loading matrix. These
386 rotation and alignment steps were accomplished using the `faMain()` and `faAlign()`
387 functions in the R *fungible* package (Waller, 2019). Code for all aspects of this study is
388 available at https://github.umn.edu/krach018/masters_thesis.

389

Results

390 **Recovery of R_{Pop}**

391 One of the primary reasons for conducting the present simulation study was to
392 determine which of the three investigated smoothing methods—the Higham (2002),

³⁹³ Bentler-Yuan (2011), or Knol-Berger (1991) algorithms—resulted in smoothed correlation
³⁹⁴ matrices that were closest to the correlation matrix implied by the major factor model (i.e.,
³⁹⁵ the factor model not including the minor factors). In particular, I examined whether
³⁹⁶ smoothed correlation matrices were closer to the model-implied correlation matrix than the
³⁹⁷ unsmoothed, indefinite correlation matrix. In this context, the scaled distance between two
³⁹⁸ $p \times p$ correlation matrices $\mathbf{A} = \{a_{ij}\}$ and $\mathbf{B} = \{b_{ij}\}$ was computed as:

$$D_s(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{p-1} \sum_{j=i+1}^p \frac{(a_{ij} - b_{ij})^2}{p(p-1)/2}}. \quad (13)$$

³⁹⁹ To understand which of the smoothing algorithms most often produced a smoothed
⁴⁰⁰ correlation matrix, \mathbf{R}_{Sm} , that was closest to the model-implied correlation matrix, \mathbf{R}_{Pop} , I
⁴⁰¹ calculated $D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})$ for each \mathbf{R}_{Sm} obtained from the 124,346 indefinite tetrachoric
⁴⁰² correlation matrices.⁵ Small values of $D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})$ indicated that the smoothed
⁴⁰³ correlation matrix was a good approximation of \mathbf{R}_{Pop} , whereas large values indicated that
⁴⁰⁴ \mathbf{R}_{Sm} was a poor approximation of \mathbf{R}_{Pop} . After excluding three cases where the Higham
⁴⁰⁵ (2002) algorithm failed to converge, I fit a linear mixed-effects model (Model 1A) regressing
⁴⁰⁶ $\log D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})$ on all of the simulation design variables and their two-way interactions.
⁴⁰⁷ Additionally, a random intercept was estimated for every unique indefinite correlation
⁴⁰⁸ matrix to account for the correlation between observations corresponding the same
⁴⁰⁹ indefinite correlation matrix.⁶ The estimated fixed-effect coefficients are shown in Figure 1.
⁴¹⁰ A full summary table for the model is contained in Table 2. Figure 1 and Table 2 also
⁴¹¹ summarize the results of a second model (Model 1B) that included second-degree
⁴¹² polynomial terms for number of factors, factor loading, and subjects per item in addition
⁴¹³ to the terms included in Model 1A. The results in Table 2 indicated that Model 1B should

⁵ A table reporting the percent of tetrachoric correlation matrices in each condition that were indefinite can be found in Appendix B.

⁶ All numeric predictors were scaled to have a mean of zero and variance of one prior to analysis. Diagnostic plots are shown in Appendix A.

⁴¹⁴ be preferred based on the AIC (Akaike, 1973) and BIC (e.g., Hastie, Tibshirani, &
⁴¹⁵ Friedman, 2009) criteria. Therefore, coefficient estimates and estimated marginal means
⁴¹⁶ reported in this section were obtained using Model 1B.

⁴¹⁷ The design variables that most influenced $D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})$ values can be seen in Figure
⁴¹⁸ 1, which shows coefficient estimates with 99% confidence intervals, ordered by size. Note
⁴¹⁹ that exponentiated coefficients less than 1.01 and greater than 0.99 were omitted from the
⁴²⁰ figure to conserve space. Figure 1 shows that only a few variables had non-trivial effects on
⁴²¹ population matrix recovery. In particular, the three largest effects were for number of
⁴²² factors ($\hat{b} = -0.52$, $SE = 0.00$, $e^{-0.52} = 0.59$), number of items per factor ($\hat{b} = -0.26$,
⁴²³ $SE = 0.00$, $e^{-0.26} = 0.77$), and number of subjects per item ($\hat{b} = -0.25$, $SE = 0.00$,
⁴²⁴ $e^{-0.25} = 0.78$). These estimated effects were all negative, indicating better recovery of the
⁴²⁵ population correlation matrix for models with larger numbers of major factors, larger
⁴²⁶ numbers of items per factor, and larger numbers of subjects per item. The effects of
⁴²⁷ number of factors and number of subjects per item were somewhat offset, however, by large
⁴²⁸ (positive) estimated effects for the squared number of factors ($\hat{b} = 0.19$, $SE = 0.00$,
⁴²⁹ $e^{0.19} = 1.21$) and squared number of subjects per item ($\hat{b} = 0.08$, $SE = 0.00$, $e^{0.08} = 1.08$)
⁴³⁰ terms. The effects of all of the independent variables can be more easily understood by
⁴³¹ looking at Figure 2, which shows estimated marginal mean $D_s(\mathbf{R}_{Pop}, \mathbf{R}_{Sm})$ values (and 99%
⁴³² confidence intervals) at each level of number of factors, number of subjects per item,
⁴³³ number of items per factor, factor loading, and smoothing method.⁷

⁴³⁴ The effects most relevant to the research question were the effects of the smoothing
⁴³⁵ methods and their interactions with other variables. These effects were all relatively small,
⁴³⁶ but can still be seen in Figure 2. For instance, the Bentler-Yuan algorithm (2011) had the
⁴³⁷ largest (negative) main effect ($\hat{b} = -0.06$, $SE = 0.00$, $e^{-0.06} = 0.94$), closely followed by the
⁴³⁸ Knol-Berger (1991; $\hat{b} = -0.01$, $SE = 0.00$, $e^{-0.01} = 0.99$) and Higham (2002; $\hat{b} = -0.01$,

⁷ Estimated marginal means were used to summarize results because the data were unbalanced (due to only using indefinite tetrachoric correlation matrices in the analyses). Additional tables and figures showing results from the raw data can be found in Appendix B.

439 $SE = 0.00$, $e^{-0.01} = 0.99$) algorithms. These results suggest that all three algorithms
440 generally led to smoothed correlation matrices that were closer to their population
441 counterparts than were the unsmoothed, indefinite correlation matrices. However, the
442 differences among the smoothing algorithms were largest for conditions with small numbers
443 of subjects per item, small numbers of items per factor, and low factor loadings, as shown
444 in Figure 2. Indeed, the results show that the application of matrix smoothing was most
445 beneficial in conditions where \mathbf{R}_{Pop} was poorly estimated, regardless of which smoothing
446 algorithm was used (or whether matrix smoothing was applied at all). In conditions where
447 \mathbf{R}_{Pop} tended to be recovered better overall, there were at best only small differences
448 between the four smoothing methods.

449

450 Recovery of Factor Loadings

451 I next analyzed the simulation results in terms of factor loading recovery. In
452 particular, I was interested in whether factor analysis of smoothed indefinite correlation
453 matrices led to better factor loading estimates compared to when factor analysis was
454 conducted on the indefinite correlation matrices directly. I was also interested in whether
455 particular smoothing methods led to better factor loading estimates than others and
456 whether the interactions between smoothing methods and the other variables (e.g., number
457 of items per factor, number of subjects per item, factor analysis method, etc.) affected
458 factor loading estimation. For the purposes of these analyses, I evaluated factor loading
459 recovery using the root-mean-square error (RMSE) between the estimated and population
460 factor loadings for the major factors. Given a matrix of estimated major factor loadings
461 $\hat{\mathbf{F}} = \{\hat{f}_{ij}\}_{p \times m}$, and the corresponding matrix of population major factor loadings,

⁴⁶² $\mathbf{F} = \{f_{ij}\}_{p \times m}$,

$$\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}}) = \sqrt{\sum_{i=1}^p \sum_{j=1}^m \frac{(f_{ij} - \hat{f}_{ij})^2}{pm}}. \quad (14)$$

⁴⁶³ To determine which smoothing method resulted in the best factor loading estimates, I
⁴⁶⁴ calculated the $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ for each pair of estimated and population factor loading
⁴⁶⁵ matrices corresponding to the (possibly) smoothed indefinite tetrachoric correlation
⁴⁶⁶ matrices. Relatively small $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values indicated that the estimated factor loading
⁴⁶⁷ matrices were more similar to their corresponding population factor loading matrices,
⁴⁶⁸ whereas larger $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values indicated poorly-estimated factor loading matrices. As
⁴⁶⁹ in the previous section, the four cases where the Higham (2002) algorithm did not converge
⁴⁷⁰ were not included in my analyses. Furthermore, cases where PA failed to converge were
⁴⁷¹ also not included. In total, there were 2,714 cases where the PA algorithm did not converge
⁴⁷² (convergence rate = 99.5%) and only four cases where the ML algorithm did not converge
⁴⁷³ (convergence rate > 99.9%).

⁴⁷⁴ Using the converged data, I fit a mixed-effects model (Model 2A) regressing
⁴⁷⁵ $\log \text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ on number of subjects per item, number of items per factor, number of
⁴⁷⁶ factors, factor loading, model error, smoothing algorithm, factor analysis method (PA,
⁴⁷⁷ OLS, or ML), all two-way interactions between these variables, and a random intercept
⁴⁷⁸ estimated for every unique indefinite correlation matrix.⁸ I also fit a second mixed-effects
⁴⁷⁹ model (Model 2B) with additional second-degree polynomial terms for number of subjects
⁴⁸⁰ per item, number of factors, and factor loading. The results for both models are
⁴⁸¹ summarized in Table 3 and indicated that Model 2B should be preferred to Model 2A
⁴⁸² based on the AIC and BIC criteria. Therefore, coefficient estimates and estimated marginal
⁴⁸³ means reported in this section were obtained using Model 2B.

⁴⁸⁴ To show the variables that most affected $\log \text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$, ordered, exponentiated

⁸ 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.

485 coefficient estimates with 99% confidence intervals for Model 2B are shown in Figure 3
486 (note that exponentiated coefficients less than 1.01 and greater than 0.99 were omitted to
487 conserve space). Figure 3 shows that factor loading, items per factor, number of factors,
488 model error, and subjects per item had relatively large effects on $\log \text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$.
489 Additionally, many of the polynomial terms and interactions between these variables also
490 had relatively large estimated effects (see Figure 3 and Table 3). To better understand the
491 effects of each of these design variables on $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$, Figure 4 shows estimated marginal
492 means conditioned on number of factors, model error, number of subjects per item, number
493 of items per factor, smoothing method, factor extraction method, and factor loading. As in
494 the previous section, estimated marginal means were used instead of raw means because
495 each condition of the design had a different number of observations due to only using
496 results for indefinite tetrachoric correlation matrices.

497 Concerning the primary question of interest in this section, the coefficient estimates
498 from Model 2B (and the marginal means shown in Figure 4) indicated that choice of
499 smoothing method generally had little impact on $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$. Of the effects involving
500 smoothing methods, only the effects involving the Bentler-Yuan algorithm were
501 non-negligible. Therefore, only the application of the Bentler-Yuan algorithm to indefinite
502 tetrachoric correlation matrices seemed to be related to any improvement (or indeed,
503 difference) in $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values when used for factor analysis. Moreover, even the
504 estimated main effect associated with the Bentler-Yuan algorithm (2011) was quite modest
505 ($\hat{b} = -0.03$, $SE = 0.00$, $e^{-0.03} = 0.97$) and was offset by the positive estimated interaction
506 effects with factor loading ($\hat{b} = 0.02$, $SE = 0.00$, $e^{0.02} = 1.02$) and items per factor
507 ($\hat{b} = 0.02$, $SE = 0.00$, $e^{0.02} = 1.02$). These effects are evident in Figure 4, which shows that
508 the Bentler-Yuan algorithm (2011) led to slightly lower $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values for conditions
509 with low factor loadings and few items per factor, but led to nearly identical results to the
510 alternative methods as factor loading magnitude and number of items per factor increased.

511 Although choice of smoothing method did not have a large influence on factor loading

recovery, the other design variables did have an influence on RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values. The effects of these other variables (including interactions and polynomial terms) are best understood using the marginal means shown in Figure 4. Considering first the effect of factor loading, Figure 4 shows that RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values tended to decrease as factor loadings increased ($\hat{b} = -0.57, SE = 0.00, e^{-0.57} = 0.57$). Interestingly, there was also a relatively large interaction between factor loading and ML factor extraction ($\hat{b} = 0.22, SE = 0.00, e^{0.22} = 1.25$) such that ML seemed to benefit less than PA or OLS from higher factor loadings.

After factor loading, the number of items per factor had the largest estimated effect on RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values ($\hat{b} = -0.37, SE = 0.00, e^{-0.37} = 0.69$). As can be seen in Figure 4, increasing the number of items per factor tended to improve factor loading recovery (i.e., led to lower RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values). There was a similar (albeit smaller) effect for the number of subjects per item, such that increasing the number of subjects per item tended to lead to smaller RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values ($\hat{b} = -0.18, SE = 0.00, e^{-0.18} = 0.84$). However, both the effects of number of items per factor and number of subjects per item were affected by their associated polynomial terms and interactions. For instance, the effect of increasing the number of items per factor was largest when factor loadings were relatively small, as indicated by the interaction between the number of items per factor and squared factor loadings ($\hat{b} = 0.14, SE = 0.00, e^{0.14} = 1.15$). The effect of the number of items per factor was also influenced by model error (as will be discussed shortly). Concerning the effect of the number of subjects per item, there was a large estimated effect for squared number of subjects per item such that the beneficial effect of increasing the number of subjects dissipated as the number of subjects per item increased ($\hat{b} = 0.16, SE = 0.00, e^{0.16} = 1.17$).

Concerning the effect of number of factors on RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values, Figure 4 shows that RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values tended to be lower for conditions with more factors compared to those with fewer factors ($\hat{b} = -0.34, SE = 0.00, e^{-0.34} = 0.71$). Moreover, the decrease in RMSE($\mathbf{F}, \hat{\mathbf{F}}$) seemed to be nonlinear, such that the effect of increasing the number of

539 factors was largest when there were relatively few factors, as indicated by the quadratic
540 term for number of factors ($\hat{b} = 0.09$, $SE = 0.00$, $e^{0.09} = 1.09$). On its face, these effects
541 seem to suggest that models with large numbers of major factors led to better factor
542 loading recovery than those with fewer factors. However, another explanation for this effect
543 involves the total numbers of subjects and items. Whereas number of items per factor and
544 number of subjects per item were fully-crossed with number of factors, the total sample
545 size and total number of items for each data set were confounded with number of factors.
546 In other words, conditions with larger numbers of factors tended to include more total
547 subjects and items. The strong relationship between $\log \text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ and sample size can
548 be clearly seen in Figure 5, which shows that $\log \text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ decreased as sample size
549 increased. Therefore, it seems reasonable that the effect of number of factors might be
550 better understood as being related to the total number of items and subjects in a data set.
551 Similarly, the negative interaction between number of factors and ML ($\hat{b} = -0.20$,
552 $SE = 0.00$, $e^{-0.20} = 0.82$) could be interpreted instead as an interaction between total
553 number of items or subjects and ML.

554 Moving next to model error, Model 2B indicated that increasing the proportion of
555 uniqueness variance accounted for by the minor factors (v_E) was associated with worse
556 factor loading recovery ($\hat{b} = 0.16$, $SE = 0.00$, $e^{0.16} = 1.17$). Additionally, the detrimental
557 effect of model error on factor loading recovery seemed to worsen as the number of factors
558 and number of items per factor increased (see Figure 4 and Table 3). On the other hand,
559 model error seemed to have less of an impact on $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ as factor loadings increased,
560 as can also be seen in Figure 4. A potential explanation for this effect is that model error
561 accounted for less of the total variance in conditions with high factor loadings because the
562 levels of model error were defined as proportions of the uniqueness variance. I.e., conditions
563 with high factor loadings had small uniqueness variances and correspondingly small model
564 error variances.

565 Another notable effect involving model approximation error was the interaction

566 between model error and ML factor extraction ($\hat{b} = -0.03$, $SE = 0.00$, $e^{-0.03} = 0.97$). This
567 result indicated that, of the three factor extraction methods, ML was less affected by
568 model error than OLS or PA. Moreover, the main effect of ML indicated that it led to
569 lower overall RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values than OLS or PA when all other variables were held
570 constant ($\hat{b} = -0.05$, $SE = 0.00$, $e^{-0.05} = 0.95$). However, the previously-discussed
571 interactions between ML and factor loading, number of items per factor, and number of
572 subjects per item indicated that ML led to better results than PA or OLS only when the
573 numbers of subjects per item and items per factor were small and factor loadings were low.
574 In conditions with higher numbers of subjects and items, OLS and PA led to better (and
575 highly similar) results in terms of RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values.

576

577

Discussion

578 The current study examined how the application of three matrix smoothing
579 algorithms (the Higham [2002], Bentler-Yuan [2011], and Knol-Berger [1991] algorithms) to
580 indefinite tetrachoric correlation matrices affected both (a) the recovery of the
581 model-implied population correlation matrix (\mathbf{R}_{Pop}), and (b) the recovery of the population
582 item factor loadings in EFA (compared to leaving the indefinite correlation matrices
583 unsmoothed). With respect to recovery of \mathbf{R}_{Pop} , I found that that three variables were
584 most related to $D_s(\mathbf{R}_{\text{Sm}}, \mathbf{R}_{\text{Pop}})$: (a) the number of major factors in the data-generating
585 model, (b) the number of subjects per item, and (c) the number of items per (major)
586 factor. Increases in any of these variables were associated with improved population
587 correlation matrix recovery. I also found that choice of smoothing method was somewhat
588 related to population correlation matrix recovery. Specifically, the application of any of the
589 three investigated matrix smoothing algorithms led to smoothed matrices were slightly

closer to the population correlation matrix (\mathbf{R}_{Pop}) than the unsmoothed, indefinite tetrachoric correlation matrices. The results indicated that although the three smoothing algorithms led to very similar $D_s(\mathbf{R}_{\text{Sm}}, \mathbf{R}_{\text{Pop}})$ values in most conditions, the Bentler-Yuan algorithm (2011) led to slightly lower $D_s(\mathbf{R}_{\text{Sm}}, \mathbf{R}_{\text{Pop}})$ values in conditions with few subjects per item, few items per factor, and low factor loadings.

Concerning factor loading recovery, the simulation study results indicated that choice of smoothing algorithm—or, in fact, whether smoothing was applied at all—was not an important determinant of factor loading recovery when EFA was applied to smoothed or unsmoothed indefinite tetrachoric correlation matrices. Similar to the previous analyses, the Bentler-Yuan algorithm (2011) led to slightly better results (i.e., lower $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values) than the alternative smoothing methods when factor loadings were low and there were few items per factor. Moreover, the Bentler-Yuan algorithm led to slightly better results when paired with maximum likelihood factor extraction (ML) compared to when the ordinary least squares (OLS) or principal axis (PA) extraction methods were used. However, the differences between the four smoothing methods (in terms of $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values) were never large enough to be of practical importance. Although smoothing method choice was not found to be important for determining factor loading recovery, many of the other design variables were found to be important. In particular, $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values were smallest for conditions with high factor loadings, many items per factor, and with little or no model approximation error. Moreover, the results indicated that the OLS and PA factor extraction methods led to highly similar results under all conditions. ML factor extraction method led to better results than OLS and PA in conditions with low factor loadings, few items per factor, and few subjects per item. The results also indicated that factor loading recovery for ML was less affected by model approximation error than were OLS or PA.

The results of this simulation study concerning both population correlation matrix recovery ($D_s(\mathbf{R}_{\text{Sm}}, \mathbf{R}_{\text{Pop}})$) and population factor loading recovery ($\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$) can be put in the context of previous research. First, the current results provided additional evidence

that the application of matrix smoothing algorithms to indefinite tetrachoric correlation matrices led to, at most, only a small effect on factor loading estimates in subsequent factor analyses. This result lends additional support to the conclusion of Knol and Berger (1991) that the effect of applying matrix smoothing to indefinite tetrachoric correlation matrices prior to conducting factor analysis was negligible.

To the extent that there were small differences among the smoothing methods in terms of $D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})$ and $RMSE(\mathbf{F}, \hat{\mathbf{F}})$, the Bentler-Yuan algorithm (2011) tended to lead to slightly better results than the alternative algorithms. Although I am not aware of any previous comparisons of relative smoothing algorithm performance in terms of population correlation matrix or factor loading recovery, Debelak and Tran (2013) and Debelak and Tran (2016) both found that the Bentler-Yuan algorithm led to somewhat better results than the Higham (2002) or Knol-Berger (1991) algorithms used with indefinite polychoric correlation matrices in the context of parallel analysis. These results, combined with the results from the present study, suggest that the Bentler-Yuan algorithm (2011) should be the default choice for smoothing indefinite tetrachoric or polychoric correlation matrices prior to conducting parallel analysis or factor analysis.

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. Moreover, the present study only included factor models with equal numbers of salient items per factor and fixed, uniform factor loadings. It might be the case that these loading matrices were overly-simplified and not representative of real data. Future research on this topic should investigate whether more complex factor loading and correlation structures affect the performance of matrix smoothing algorithms in terms of population correlation matrix recovery and factor loading recovery. Additionally, the present study only investigated the

643 effects of matrix smoothing on indefinite tetrachoric correlation matrices. Further research
644 should be done to investigate the effects of matrix smoothing on indefinite polychoric
645 correlation matrices, as well as correlation matrices that are indefinite due to other causes
646 such as indefinite correlation matrices calculated using pairwise deletion (Wothke, 1993) or
647 composite correlation matrices used in meta-analysis (Furlow & Beretvas, 2005). Little is
648 known about whether the mechanism or “cause” of indefinite correlation matrices affects
649 their structure, or how these potential differences might interact with the application of
650 matrix smoothing algorithms.

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

664 Conclusion

665 Despite the lackluster improvement in factor loading recovery when factor analysis
666 was conducted on smoothed rather than indefinite tetrachoric correlation matrices, the
667 application of one of the three investigated matrix smoothing algorithms on indefinite
668 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 indefinite 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 Bentler-Yuan algorithm (2011) often led to results that were at least as good (and sometimes slightly better) than the alternative smoothing algorithms and therefore seems a default choice of smoothing algorithm. Where the Bentler-Yuan algorithm is not available, the Knol-Berger algorithm (1991) is an alternative that is fast, easily implemented in most programming languages, does not have convergence issues, and generally led to results comparable to the Bentler-Yuan algorithm.

These recommendations come with a strong caveat. Namely, no matrix smoothing algorithm can reasonably be considered a remedy or solution for indefinite tetrachoric correlation matrices. Instead, researchers should consider indefinite 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, indefinite 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 indefinite 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 indefinite tetrachoric correlation matrices prior to factor analysis and seem to offer a modest benefit (in terms of factor loading recovery) compared to leaving the indefinite tetrachoric correlation matrix unsmoothed. However, the application of these algorithms should be considered to be little

- 696 more than a band-aid fix that does not address the underlying issues leading to indefinite
697 tetrachoric correlation matrices nor to a marked improvement in factor loading recovery.

References

- 698 Akaike, H. (1973). *Information theory and an extension of the maximum likelihood principle*. (B. N. Petrov & F. Caski, Eds.) (pp. 267–281). Budapest: Akademiai Kiado.
- 702 Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown*.
703 Retrieved from <https://github.com/crsh/papaja>
- 704 Banerjee, S., & Roy, A. (2014). *Linear algebra and matrix analysis for statistics*.
705 Chapman; Hall/CRC. <https://doi.org/10.1201/b17040>
- 706 Barnier, J., Briatte, F., & Larmarange, J. (2018). *Questionr: Functions to make surveys*
707 *processing easier*. Retrieved from <https://CRAN.R-project.org/package=questionr>
- 708 Bates, D., & Maechler, M. (2019). *Matrix: Sparse and dense matrix classes and methods*.
709 Retrieved from <https://CRAN.R-project.org/package=Matrix>
- 710 Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects
711 models using lme4. *Journal of Statistical Software*, 67(1), 1–48.
712 <https://doi.org/10.18637/jss.v067.i01>
- 713 Bentler, P. (1972). A lower-bound method for the dimension-free measurement of internal
714 consistency. *Social Science Research*, 1(4), 343–357.
715 [https://doi.org/10.1016/0049-089X\(72\)90082-8](https://doi.org/10.1016/0049-089X(72)90082-8)
- 716 Bentler, P., & Yuan, K.-H. (2011). Positive definiteness via off-diagonal scaling of a
717 symmetric indefinite matrix. *Psychometrika*, 76(1), 119–123.
718 <https://doi.org/10.1007/s11336-010-9191-3>
- 719 Birnbaum, A. L. (1968). Some latent trait models and their use in inferring an examinee's
720 ability. *Statistical Theories of Mental Test Scores*.
- 721 Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item
722 parameters: Application of an em algorithm. *Psychometrika*, 46(4), 443–459.

- 723 <https://doi.org/10.1007/BF02293801>
- 724 Bock, R. D., Gibbons, R., & Muraki, E. (1988). Full-information item factor analysis.
725 *Applied Psychological Measurement*, 12(3), 261–280.
- 726 <https://doi.org/10.1177/014662168801200305>
- 727 Bolker, B., & Robinson, D. (2019). *Broom.mixed: Tidying methods for mixed models*.
728 Retrieved from <https://CRAN.R-project.org/package=broom.mixed>
- 729 Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal
730 Statistical Society: Series B (Methodological)*, 26(2), 211–243.
731 <https://doi.org/10.1111/j.2517-6161.1964.tb00553.x>
- 732 Briggs, N. E., & MacCallum, R. C. (2003). Recovery of weak common factors by maximum
733 likelihood and ordinary least squares estimation. *Multivariate Behavioral Research*,
734 38(1), 25–56. https://doi.org/10.1207/S15327906MBR3801_2
- 735 Brown, M. B., & Benedetti, J. K. (1977). On the mean and variance of the tetrachoric
736 correlation coefficient. *Psychometrika*, 42(3), 347–355.
737 <https://doi.org/10.1007/BF02293655>
- 738 Browne, M. W. (1968). A comparison of factor analytic techniques. *Psychometrika*, 33(3),
739 267–334. <https://doi.org/10.1007/BF02289327>
- 740 Butler, S. M., & Louis, T. A. (1992). Random effects models with non-parametric priors.
741 *Statistics in Medicine*, 11(14-15), 1981–2000.
742 <https://doi.org/10.1002/sim.4780111416>
- 743 Carroll, J. B. (1957). Biquartimin criterion for rotation to oblique simple structure in factor
744 analysis. *Science*, 126, 1114–1115. <https://doi.org/10.1126/science.126.3283.1114>
- 745 Choi, J., Kim, S., Chen, J., & Dannels, S. (2011). A comparison of maximum likelihood
746 and Bayesian estimation for polychoric correlation using monte carlo simulation.
747 *Journal of Educational and Behavioral Statistics*, 36(4), 523–549.

- 748 <https://doi.org/10.3102/1076998610381398>
- 749 Comrey, A. L. (1962). The minimum residual method of factor analysis. *Psychological Reports*, 11(1), 15–18. <https://doi.org/10.2466/pr0.1962.11.1.15>
- 750
- 751 Crocker, L., & Algina, J. (1986). *Introduction to classical and modern test theory*. Orlando,
- 752 FL: Wadsworth Publishing.
- 753 Cudeck, R., & Henly, S. J. (1991). Model selection in covariance structures analysis and
- 754 the “problem” of sample size: A clarification. *Psychological Bulletin*, 109(3), 512.
- 755 <https://doi.org/10.1037/0033-2909.109.3.512>
- 756 de Ayala, R. J. (2013). *The theory and practice of item response theory*. Guilford
- 757 Publications.
- 758 Debelak, R., & Tran, U. S. (2013). Principal component analysis of smoothed tetrachoric
- 759 correlation matrices as a measure of dimensionality. *Educational and Psychological*
- 760 *Measurement*, 73(1), 63–77. <https://doi.org/10.1177/0013164412457366>
- 761 Debelak, R., & Tran, U. S. (2016). Comparing the effects of different smoothing algorithms
- 762 on the assessment of dimensionality of ordered categorical items with parallel
- 763 analysis. *PLOS ONE*, 11(2), 1–18. <https://doi.org/10.1371/journal.pone.0148143>
- 764 Devlin, S. J., Gnanadesikan, R., & Kettenring, J. R. (1975). Robust estimation and outlier
- 765 detection with correlation coefficients. *Biometrika*, 62(3), 531–545.
- 766 <https://doi.org/10.1093/biomet/62.3.531>
- 767 Dillon, W. R., Kumar, A., & Mulani, N. (1987). Offending estimates in covariance
- 768 structure analysis: Comments on the causes of and solutions to Heywood cases.
- 769 *Psychological Bulletin*, 101(1), 126. <https://doi.org/10.1037/0033-2909.101.1.126>
- 770 Divgi, D. R. (1979). Calculation of the tetrachoric correlation coefficient. *Psychometrika*,
- 771 44(2), 169–172. <https://doi.org/10.1007/BF02293968>
- 772 Dong, H.-K. (1985). Non-Gramian and singular matrices in maximum likelihood factor

- 773 analysis. *Applied Psychological Measurement*, 9(4), 363–366.
- 774 <https://doi.org/10.1177/014662168500900404>
- 775 Dwyer, P. S. (1939). The contribution of an orthogonal multiple factor solution to multiple
776 correlation. *Psychometrika*, 4(2), 163–171. <https://doi.org/10.1007/BF02288494>
- 777 Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., & Strahan, E. J. (1999). Evaluating
778 the use of exploratory factor analysis in psychological research. *Psychological
779 Methods*, 4(3), 272. <https://doi.org/10.1037/1082-989X.4.3.272>
- 780 Fox, J., & Weisberg, S. (2019). *An R companion to applied regression* (Third). Thousand
781 Oaks CA: Sage. Retrieved from
782 <https://socialsciences.mcmaster.ca/jfox/Books/Companion/>
- 783 Furlow, C. F., & Beretvas, S. N. (2005). Meta-analytic methods of pooling correlation
784 matrices for structural equation modeling under different patterns of missing data.
785 *Psychological Methods*, 10(2), 227. <https://doi.org/10.1037/1082-989X.10.2.227>
- 786 Fushiki, T. (2009). Estimation of positive semidefinite correlation matrices by using convex
787 quadratic semidefinite programming. *Neural Computation*, 21(7), 2028–2048.
788 <https://doi.org/10.1162/neco.2009.04-08-765>
- 789 Garnier, S. (2018). *Viridis: Default color maps from 'matplotlib'*. Retrieved from
790 <https://CRAN.R-project.org/package=viridis>
- 791 Gelman, A., & Su, Y.-S. (2018). *Arm: Data analysis using regression and
792 multilevel/hierarchical models*. Retrieved from
793 <https://CRAN.R-project.org/package=arm>
- 794 Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2018). *Multivariate data
795 analysis*. Cengage.
- 796 Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning:
797 Data mining, inference, and prediction* (Second Edition). New York, NY: Springer

- 798 New York.
- 799 Henry, L., & Wickham, H. (2019). *Purrr: Functional programming tools*. Retrieved from
800 <https://CRAN.R-project.org/package=purrr>
- 801 Higham, N. J. (2002). Computing the nearest correlation matrix—a problem from finance.
802 *IMA Journal of Numerical Analysis*, 22(3), 329–343.
803 <https://doi.org/10.1093/imanum/22.3.329>
- 804 Hong, S. (1999). Generating correlation matrices with model error for simulation studies in
805 factor analysis: A combination of the Tucker-Koopman-Linn model and Wijsman's
806 algorithm. *Behavior Research Methods, Instruments, & Computers*, 31(4), 727–730.
807 <https://doi.org/10.3758/BF03200754>
- 808 Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis.
809 *Psychometrika*, 30(2), 179–185. <https://doi.org/10.1007/BF02289447>
- 810 Jacqmin-Gadda, H., Sibillot, S., Proust, C., Molina, J.-M., & Thiébaut, R. (2007).
811 Robustness of the linear mixed model to misspecified error distribution.
812 *Computational Statistics & Data Analysis*, 51(10), 5142–5154.
813 <https://doi.org/10.1016/j.csda.2006.05.021>
- 814 Jamshidian, M., & Bentler, P. M. (1998). A quasi-Newton method for minimum trace
815 factor analysis. *Journal of Statistical Computation and Simulation*, 62(1-2), 73–89.
816 <https://doi.org/10.1080/00949659808811925>
- 817 Jennrich, R. I. (2002). A simple general method for oblique rotation. *Psychometrika*,
818 67(1), 7–19. <https://doi.org/10.1007/BF02294706>
- 819 Knol, D. L., & Berger, M. P. (1991). Empirical comparison between factor analysis and
820 multidimensional item response models. *Multivariate Behavioral Research*, 26(3),
821 457–477. https://doi.org/10.1207/s15327906mbr2603_5
- 822 Knowles, J. E., & Frederick, C. (2019). *MerTools: Tools for analyzing mixed effect*

- 823 *regression models*. Retrieved from <https://CRAN.R-project.org/package=merTools>
- 824 Koller, M. (2016). robustlmm: An R package for robust estimation of linear mixed-effects
825 models. *Journal of Statistical Software*, 75(6), 1–24.
826 <https://doi.org/10.18637/jss.v075.i06>
- 827 Kshirsagar, A. M. (1959). Bartlett decomposition and Wishart distribution. *The Annals of
828 Mathematical Statistics*, 30(1), 239–241. <https://doi.org/10.1214/aoms/1177706379>
- 829 Leifeld, P. (2013). texreg: Conversion of statistical model output in R to LaTeX and
830 HTML tables. *Journal of Statistical Software*, 55(8), 1–24.
831 <https://doi.org/10.18637/jss.v055.i08>
- 832 Li, Q., Li, D., & Qi, H. (2010). Newton's method for computing the nearest correlation
833 matrix with a simple upper bound. *Journal of Optimization Theory and
834 Applications*, 147(3), 546–568. <https://doi.org/10.1007/s10957-010-9738-6>
- 835 Lorenzo-Seva, U., & Ferrando, P. J. (2020). Not positive definite correlation matrices in
836 exploratory item factor analysis: Causes, consequences and a proposed solution.
837 *Structural Equation Modeling: A Multidisciplinary Journal*, 1–10.
838 <https://doi.org/10.1080/10705511.2020.1735393>
- 839 Lurie, P. M., & Goldberg, M. S. (1998). An approximate method for sampling correlated
840 random variables from partially-specified distributions. *Management Science*, 44(2),
841 203–218. <https://doi.org/10.1287/mnsc.44.2.203>
- 842 MacCallum, R. C., & Tucker, L. R. (1991). Representing sources of error in the
843 common-factor model: Implications for theory and practice. *Psychological Bulletin*,
844 109(3), 502. <https://doi.org/10.1037/0033-2909.109.3.502>
- 845 MacCallum, R. C., Widaman, K. F., Preacher, K. J., & Hong, S. (2001). Sample size in
846 factor analysis: The role of model error. *Multivariate Behav. Res.*, 36(4), 611–637.
847 https://doi.org/10.1207/S15327906MBR3604_06

- 848 Maechler, M. (2019). *Sfsmisc: Utilities from 'seminar fuer statistik' eth zurich*. Retrieved
849 from <https://CRAN.R-project.org/package=sfsmisc>
- 850 Magnus, J. R., & Neudecker, H. (2019). *Matrix differential calculus with applications in*
851 *statistics and econometrics*. John Wiley & Sons.
852 <https://doi.org/10.1002/9781119541219>
- 853 Marwick, B. (2019). *Wordcountaddin: Word counts and readability statistics in r markdown*
854 *documents*. Retrieved from <https://github.com/benmarwick/wordcountaddin>
- 855 Meschiari, S. (2015). *Latex2exp: Use latexexpressions in plots*. Retrieved from
856 <https://CRAN.R-project.org/package=latex2exp>
- 857 Michalke, M. (2018a). *KoRpus: An r package for text analysis*. Retrieved from
858 <https://reaktanz.de/?c=hacking&s=koRpus>
- 859 Michalke, M. (2018b). *Sylly: Hyphenation and syllable counting for text analysis*.
860 Retrieved from <https://reaktanz.de/?c=hacking&s=sylly>
- 861 Michalke, M. (2019). *KoRpus.lang.en: Language support for 'koRpus' package: English*.
862 Retrieved from <https://reaktanz.de/?c=hacking&s=koRpus>
- 863 Mulaik, S. A. (2009). *Foundations of factor analysis*. Chapman; Hall/CRC.
864 <https://doi.org/10.1201/b15851>
- 865 Müller, K. (2017). *Here: A simpler way to find your files*. Retrieved from
866 <https://CRAN.R-project.org/package=here>
- 867 Müller, K., & Wickham, H. (2019). *Tibble: Simple data frames*. Retrieved from
868 <https://CRAN.R-project.org/package=tibble>
- 869 Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient.
870 *Psychometrika*, 44(4), 443–460. <https://doi.org/10.1007/BF02296207>
- 871 Pedersen, T. L. (2019). *Patchwork: The composer of plots*. Retrieved from
872 <https://CRAN.R-project.org/package=patchwork>

- 873 Qi, H., & Sun, D. (2006). A quadratically convergent Newton method for computing the
874 nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*,
875 28(2), 360–385. <https://doi.org/10.1137/050624509>
- 876 R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna,
877 Austria: R Foundation for Statistical Computing. Retrieved from
878 <https://www.R-project.org/>
- 879 Roff, M. (1936). Some properties of the communality in multiple factor theory.
880 *Psychometrika*, 1(2), 1–6. <https://doi.org/10.1007/BF02287999>
- 881 Sarkar, D. (2008). *Lattice: Multivariate data visualization with R*. New York: Springer.
882 <https://doi.org/10.1007/978-0-387-75969-2>
- 883 Thurstone, L. L. (1947). *Multiple-factor analysis; a development and expansion of the*
884 *vectors of mind* (pp. xix, 535). University of Chicago Press.
- 885 Tucker, L. R., Koopman, R. F., & Linn, R. L. (1969). Evaluation of factor analytic
886 research procedures by means of simulated correlation matrices. *Psychometrika*,
887 34(4), 421–459. <https://doi.org/10.1007/BF02290601>
- 888 Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (Fourth). New
889 York: Springer. <https://doi.org/10.1007/978-0-387-21706-2>
- 890 Verbeke, G., & Lesaffre, E. (1997). The effect of misspecifying the random-effects
891 distribution in linear mixed models for longitudinal data. *Computational Statistics*
892 & Data Analysis, 23(4), 541–556. [https://doi.org/10.1016/S0167-9473\(96\)00047-3](https://doi.org/10.1016/S0167-9473(96)00047-3)
- 893 Waller, N. G. (2019). *Fungible: Psychometric functions from the Waller lab*.
- 894 Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New
895 York. <https://doi.org/10.1007/978-3-319-24277-4>
- 896 Wickham, H. (2019a). *Forcats: Tools for working with categorical variables (factors)*.
897 Retrieved from <https://CRAN.R-project.org/package=forcats>

- 898 Wickham, H. (2019b). *Stringr: Simple, consistent wrappers for common string operations.*
899 Retrieved from <https://CRAN.R-project.org/package=stringr>
- 900 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ...
901 Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*,
902 4(43), 1686. <https://doi.org/10.21105/joss.01686>
- 903 Wickham, H., François, R., Henry, L., & Müller, K. (2019). *Dplyr: A grammar of data
904 manipulation.* Retrieved from <https://CRAN.R-project.org/package=dplyr>
- 905 Wickham, H., & Henry, L. (2019). *Tidyr: Tidy messy data.* Retrieved from
906 <https://CRAN.R-project.org/package=tidyr>
- 907 Wickham, H., Hester, J., & Francois, R. (2018). *Readr: Read rectangular text data.*
908 Retrieved from <https://CRAN.R-project.org/package=readr>
- 909 Widaman, K. F., & Herring, L. G. (1985). Iterative least squares estimates of
910 communality: Initial estimate need not affect stabilized value. *Psychometrika*,
911 50(4), 469–477. <https://doi.org/10.1007/BF02296264>
- 912 Wirth, R., & Edwards, M. C. (2007). Item factor analysis: Current approaches and future
913 directions. *Psychological Methods*, 12(1), 58.
914 <https://doi.org/10.1037/1082-989X.12.1.58>
- 915 Wotheke, W. (1993). Testing structural equation models. In K. A. Bollen & J. S. Long
916 (Eds.), *Testing structural equation models* (pp. 256–293). A Sage Focus Edition.
- 917 Xie, Y. (2015). *Dynamic documents with R and knitr* (2nd ed.). Boca Raton, Florida:
918 Chapman; Hall/CRC. <https://doi.org/10.1201/b15166>
- 919 Zhang, D., & Davidian, M. (2001). Linear mixed models with flexible distributions of
920 random effects for longitudinal data. *Biometrics*, 57(3), 795–802.
921 <https://doi.org/10.1111/j.0006-341X.2001.00795.x>

Table 1

Number of items (p) and subjects (N) 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	p	N
1	5	5	5	25
3	5	5	15	75
5	5	5	25	125
10	5	5	50	250
1	10	5	10	50
3	10	5	30	150
5	10	5	50	250
10	10	5	100	500
1	5	10	5	50
3	5	10	15	150
5	5	10	25	250
10	5	10	50	500
1	10	10	10	100
3	10	10	30	300
5	10	10	50	500
10	10	10	100	1,000
1	5	15	5	75
3	5	15	15	225
5	5	15	25	375
10	5	15	50	750
1	10	15	10	150
3	10	15	30	450
5	10	15	50	750
10	10	15	100	1,500

Table 2

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

	Linear Model	Polynomial Model
Constant	-2.209 (0.001)	-2.339 (0.001)
Subjects/Item	-0.300 (0.001)	-0.249 (0.001)
Items/Factor	-0.229 (0.001)	-0.262 (0.001)
Factors	-0.371 (0.001)	-0.521 (0.001)
Factor Loading	-0.048 (0.001)	-0.048 (0.001)
Model Error	-0.008 (0.001)	-0.010 (0.001)
Smoothing Method (APA)	-0.015 (0.000)	-0.009 (0.000)
Smoothing Method (BY)	-0.067 (0.000)	-0.058 (0.000)
Smoothing Method (KB)	-0.020 (0.000)	-0.013 (0.000)
Subjects/Item ²		0.078 (0.002)
Factors ²		0.189 (0.001)
Model Error ²		-0.004 (0.002)
Subjects/Item × Items/Factor	-0.006 (0.001)	-0.005 (0.001)
Subjects/Item × Factors	0.016 (0.001)	0.005 (0.001)
Subjects/Item × Factor Loading	0.007 (0.001)	-0.027 (0.001)
Subjects/Item × Model Error	-0.001 (0.001)	-0.004 (0.001)
Subjects/Item × Smoothing Method (APA)	0.019 (0.000)	0.017 (0.000)
Subjects/Item × Smoothing Method (BY)	0.038 (0.000)	0.032 (0.000)
Subjects/Item × Smoothing Method (KB)	0.027 (0.000)	0.025 (0.000)
Subjects/Item × Factors ²		-0.006 (0.001)
Subjects/Item × Model Error ²		-0.000 (0.001)
Items/Factor × Factors	-0.034 (0.001)	0.009 (0.001)
Items/Factor × Factor Loading	-0.005 (0.001)	0.001 (0.001)
Items/Factor × Model Error	0.002 (0.001)	0.001 (0.001)
Items/Factor × Smoothing Method (APA)	-0.000 (0.000)	-0.000 (0.000)
Items/Factor × Smoothing Method (BY)	0.018 (0.000)	0.016 (0.000)
Items/Factor × Smoothing Method (KB)	0.000 (0.000)	-0.000 (0.000)
Items/Factor × Subjects/Item ²		0.003 (0.001)
Items/Factor × Factors ²		-0.003 (0.001)
Items/Factor × Model Error ²		-0.000 (0.001)
Factors × Factor Loading	0.033 (0.001)	0.077 (0.001)
Factors × Model Error	0.001 (0.001)	-0.002 (0.001)
Factors × Smoothing Method (APA)	0.002 (0.000)	0.003 (0.000)
Factors × Smoothing Method (BY)	-0.005 (0.000)	-0.014 (0.000)
Factors × Smoothing Method (KB)	-0.000 (0.000)	-0.002 (0.000)
Factors × Subjects/Item ²		0.002 (0.001)
Factors × Model Error ²		-0.001 (0.001)

	Linear Model	Polynomial Model
Factor Loading \times Model Error	0.009 (0.001)	0.008 (0.001)
Factor Loading \times Smoothing Method (APA)	-0.008 (0.000)	-0.008 (0.000)
Factor Loading \times Smoothing Method (BY)	0.024 (0.000)	0.024 (0.000)
Factor Loading \times Smoothing Method (KB)	-0.011 (0.000)	-0.011 (0.000)
Factor Loading \times Subjects/Item ²		0.002 (0.001)
Factor Loading \times Factors ²		-0.049 (0.001)
Factor Loading \times Model Error ²		0.003 (0.001)
Model Error \times Smoothing Method (APA)	-0.003 (0.000)	-0.003 (0.000)
Model Error \times Smoothing Method (BY)	0.001 (0.000)	0.000 (0.000)
Model Error \times Smoothing Method (KB)	-0.004 (0.000)	-0.004 (0.000)
Model Error \times Subjects/Item ²		0.001 (0.001)
Model Error \times Factors ²		0.002 (0.001)
Smoothing Method (APA) \times Subjects/Item ²		-0.009 (0.000)
Smoothing Method (BY) \times Subjects/Item ²		-0.028 (0.000)
Smoothing Method (KB) \times Subjects/Item ²		-0.013 (0.000)
Smoothing Method (APA) \times Factors ²		-0.002 (0.000)
Smoothing Method (BY) \times Factors ²		0.012 (0.000)
Smoothing Method (KB) \times Factors ²		0.002 (0.000)
Smoothing Method (APA) \times Model Error ²		-0.001 (0.000)
Smoothing Method (BY) \times Model Error ²		0.001 (0.000)
Smoothing Method (KB) \times Model Error ²		-0.001 (0.000)
Subjects/Item ² \times Factors ²		0.000 (0.001)
Subjects/Item ² \times Model Error ²		0.002 (0.002)
Factors ² \times Model Error ²		0.001 (0.001)
AIC	-1808591.524	-1938579.038
BIC	-1808191.308	-1937878.660
Log Likelihood	904331.762	969352.519
Num. obs.	497381	497381
Num. groups: id	124346	124346
Var: id (Intercept)	0.017	0.008
Var: Residual	0.000	0.000

Table 3

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

	Linear Model	Polynomial Model
Constant	-2.235 (0.001)	-2.229 (0.003)
Subjects/Item	-0.189 (0.001)	-0.183 (0.003)
Items/Factor	-0.175 (0.001)	-0.369 (0.002)
Factors	-0.255 (0.001)	-0.344 (0.002)
Factor Loading	-0.438 (0.001)	-0.574 (0.002)
Model Error	0.104 (0.001)	0.229 (0.002)
Smoothing Method (APA)	-0.006 (0.000)	-0.004 (0.001)
Smoothing Method (BY)	-0.019 (0.000)	-0.033 (0.001)
Smoothing Method (KB)	-0.011 (0.000)	-0.008 (0.001)
Extraction Method (ML)	0.110 (0.000)	-0.049 (0.001)
Extraction Method (PA)	0.005 (0.000)	0.004 (0.001)
Subjects/Item ²		0.157 (0.003)
Factor Loading ²		0.006 (0.003)
Factors ²		0.095 (0.002)
Model Error ²		0.099 (0.003)
Subjects/Item × Items/Factor	0.016 (0.001)	0.024 (0.001)
Subjects/Item × Factors	0.021 (0.001)	0.033 (0.001)
Subjects/Item × Factor Loading	-0.017 (0.001)	-0.076 (0.003)
Subjects/Item × Model Error	0.025 (0.001)	0.019 (0.001)
Subjects/Item × Smoothing Method (APA)	0.003 (0.000)	0.003 (0.000)
Subjects/Item × Smoothing Method (BY)	0.003 (0.000)	0.001 (0.000)
Subjects/Item × Smoothing Method (KB)	0.005 (0.000)	0.006 (0.000)
Subjects/Item × Extraction Method (ML)	0.107 (0.000)	0.146 (0.000)
Subjects/Item × Extraction Method (PA)	-0.000 (0.000)	-0.000 (0.000)
Subjects/Item × Factor Loading ²		-0.004 (0.004)
Subjects/Item × Factors ²		-0.020 (0.001)
Subjects/Item × Model Error ²		0.012 (0.002)
Items/Factor × Factors	-0.005 (0.001)	0.041 (0.001)
Items/Factor × Factor Loading	-0.004 (0.001)	-0.052 (0.001)
Items/Factor × Model Error	0.036 (0.001)	0.054 (0.001)
Items/Factor × Smoothing Method (APA)	0.002 (0.000)	0.003 (0.000)
Items/Factor × Smoothing Method (BY)	0.012 (0.000)	0.017 (0.000)
Items/Factor × Smoothing Method (KB)	0.002 (0.000)	0.002 (0.000)
Items/Factor × Extraction Method (ML)	0.082 (0.000)	0.102 (0.000)
Items/Factor × Extraction Method (PA)	-0.003 (0.000)	-0.005 (0.000)
Items/Factor × Subjects/Item ²		-0.005 (0.001)
Items/Factor × Factor Loading ²		0.144 (0.001)

	Linear Model	Polynomial Model
Items/Factor \times Factors ²		-0.012 (0.001)
Items/Factor \times Model Error ²		0.021 (0.001)
Factors \times Factor Loading	-0.014 (0.001)	-0.014 (0.001)
Factors \times Model Error	0.042 (0.001)	0.068 (0.001)
Factors \times Smoothing Method (APA)	0.001 (0.000)	0.001 (0.000)
Factors \times Smoothing Method (BY)	0.000 (0.000)	-0.003 (0.000)
Factors \times Smoothing Method (KB)	0.001 (0.000)	-0.000 (0.000)
Factors \times Extraction Method (ML)	-0.090 (0.000)	-0.203 (0.000)
Factors \times Extraction Method (PA)	-0.004 (0.000)	-0.006 (0.000)
Factors \times Subjects/Item ²		-0.007 (0.002)
Factors \times Factor Loading ²		-0.073 (0.002)
Factors \times Model Error ²		0.026 (0.002)
Factor Loading \times Model Error	-0.047 (0.001)	-0.023 (0.001)
Factor Loading \times Smoothing Method (APA)	0.000 (0.000)	0.000 (0.000)
Factor Loading \times Smoothing Method (BY)	0.021 (0.000)	0.022 (0.000)
Factor Loading \times Smoothing Method (KB)	-0.001 (0.000)	-0.001 (0.000)
Factor Loading \times Extraction Method (ML)	0.204 (0.000)	0.217 (0.000)
Factor Loading \times Extraction Method (PA)	-0.003 (0.000)	-0.004 (0.000)
Factor Loading \times Subjects/Item ²		0.004 (0.003)
Factor Loading \times Factors ²		0.018 (0.001)
Factor Loading \times Model Error ²		0.003 (0.002)
Model Error \times Smoothing Method (APA)	0.000 (0.000)	0.000 (0.000)
Model Error \times Smoothing Method (BY)	0.001 (0.000)	0.001 (0.000)
Model Error \times Smoothing Method (KB)	-0.000 (0.000)	-0.000 (0.000)
Model Error \times Extraction Method (ML)	-0.034 (0.000)	-0.035 (0.000)
Model Error \times Extraction Method (PA)	-0.001 (0.000)	-0.001 (0.000)
Model Error \times Subjects/Item ²		-0.034 (0.001)
Model Error \times Factor Loading ²		-0.117 (0.001)
Model Error \times Factors ²		-0.022 (0.001)
Smoothing Method (APA) \times Extraction Method (ML)	0.006 (0.001)	0.006 (0.000)
Smoothing Method (BY) \times Extraction Method (ML)	0.019 (0.001)	0.019 (0.000)
Smoothing Method (KB) \times Extraction Method (ML)	0.011 (0.001)	0.011 (0.000)
Smoothing Method (APA) \times Extraction Method (PA)	-0.002 (0.001)	-0.002 (0.000)
Smoothing Method (BY) \times Extraction Method (PA)	-0.004 (0.001)	-0.003 (0.000)
Smoothing Method (KB) \times Extraction Method (PA)	-0.002 (0.001)	-0.002 (0.000)
Smoothing Method (APA) \times Subjects/Item ²		-0.002 (0.000)
Smoothing Method (BY) \times Subjects/Item ²		-0.011 (0.000)
Smoothing Method (KB) \times Subjects/Item ²		-0.004 (0.000)
Smoothing Method (APA) \times Factor Loading ²		0.001 (0.000)
Smoothing Method (BY) \times Factor Loading ²		0.016 (0.000)
Smoothing Method (KB) \times Factor Loading ²		0.001 (0.000)
Smoothing Method (APA) \times Factors ²		0.000 (0.000)
Smoothing Method (BY) \times Factors ²		0.005 (0.000)
Smoothing Method (KB) \times Factors ²		0.001 (0.000)

	Linear Model	Polynomial Model
Smoothing Method (APA) \times Model Error ²		-0.000 (0.000)
Smoothing Method (BY) \times Model Error ²		-0.001 (0.000)
Smoothing Method (KB) \times Model Error ²		-0.000 (0.000)
Extraction Method (ML) \times Subjects/Item ²		0.011 (0.000)
Extraction Method (PA) \times Subjects/Item ²		0.001 (0.000)
Extraction Method (ML) \times Factor Loading ²		0.105 (0.000)
Extraction Method (PA) \times Factor Loading ²		0.001 (0.000)
Extraction Method (ML) \times Factors ²		0.139 (0.000)
Extraction Method (PA) \times Factors ²		0.004 (0.000)
Extraction Method (ML) \times Model Error ²		-0.016 (0.000)
Extraction Method (PA) \times Model Error ²		-0.000 (0.000)
Subjects/Item ² \times Factor Loading ²		-0.087 (0.004)
Subjects/Item ² \times Factors ²		0.010 (0.002)
Subjects/Item ² \times Model Error ²		-0.003 (0.002)
Factor Loading ² \times Factors ²		0.040 (0.002)
Factor Loading ² \times Model Error ²		-0.052 (0.002)
Factors ² \times Model Error ²		-0.027 (0.002)
AIC	-2414463.669	-2801683.227
BIC	-2413804.118	-2800461.837
Log Likelihood	1207285.834	1400941.614
Num. obs.	1489425	1489425
Num. groups: id	124346	124346
Var: id (Intercept)	0.032	0.017
Var: Residual	0.008	0.007

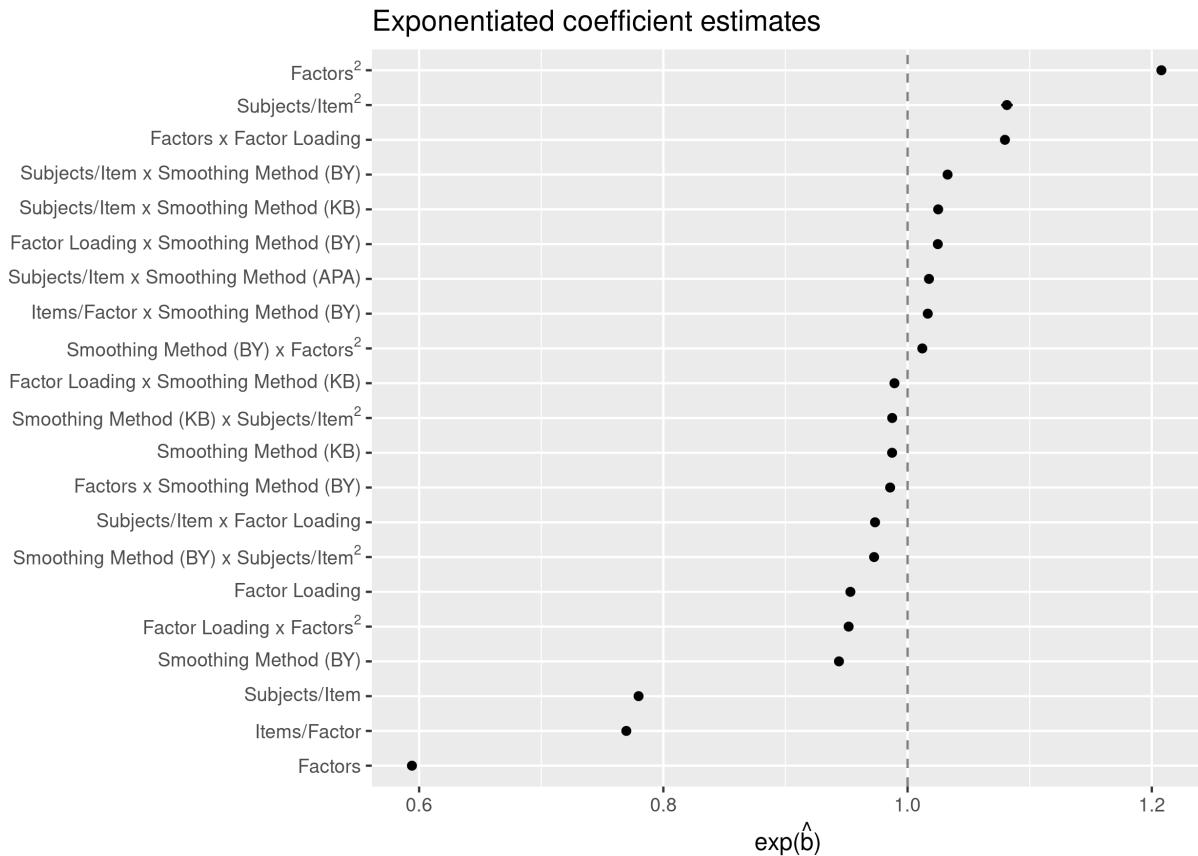


Figure 1. Exponentiated coefficient estimates for the mixed effects model using $\log[D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})]$ as the dependent variable (Model 1B). APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991). The effect of the condition where no smoothing was applied is subsumed within the Constant term.

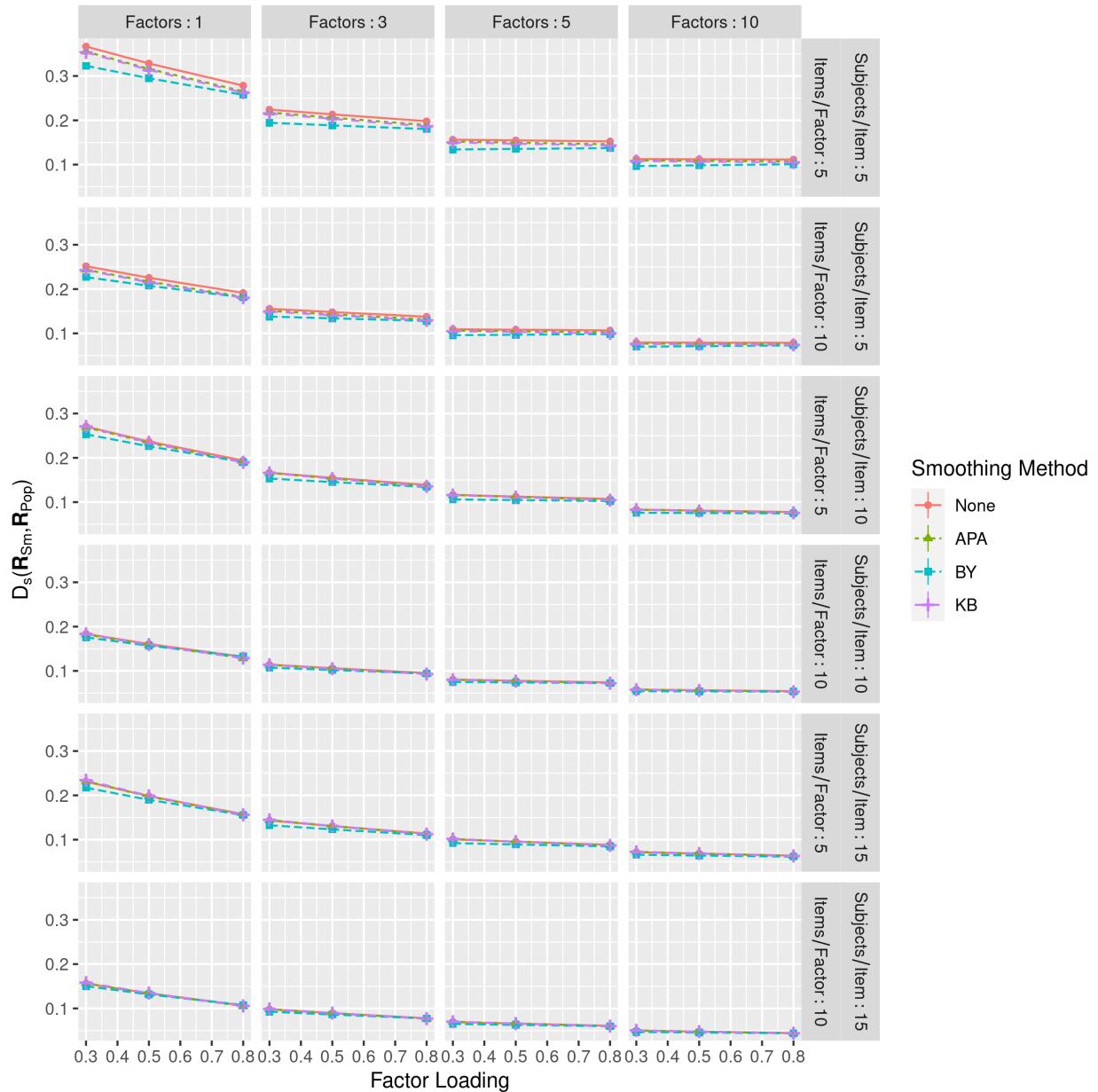


Figure 2. Scaled distance between the smoothed (\mathbf{R}_{Sm}) and model-implied (\mathbf{R}_{Pop}) correlation matrices. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing.

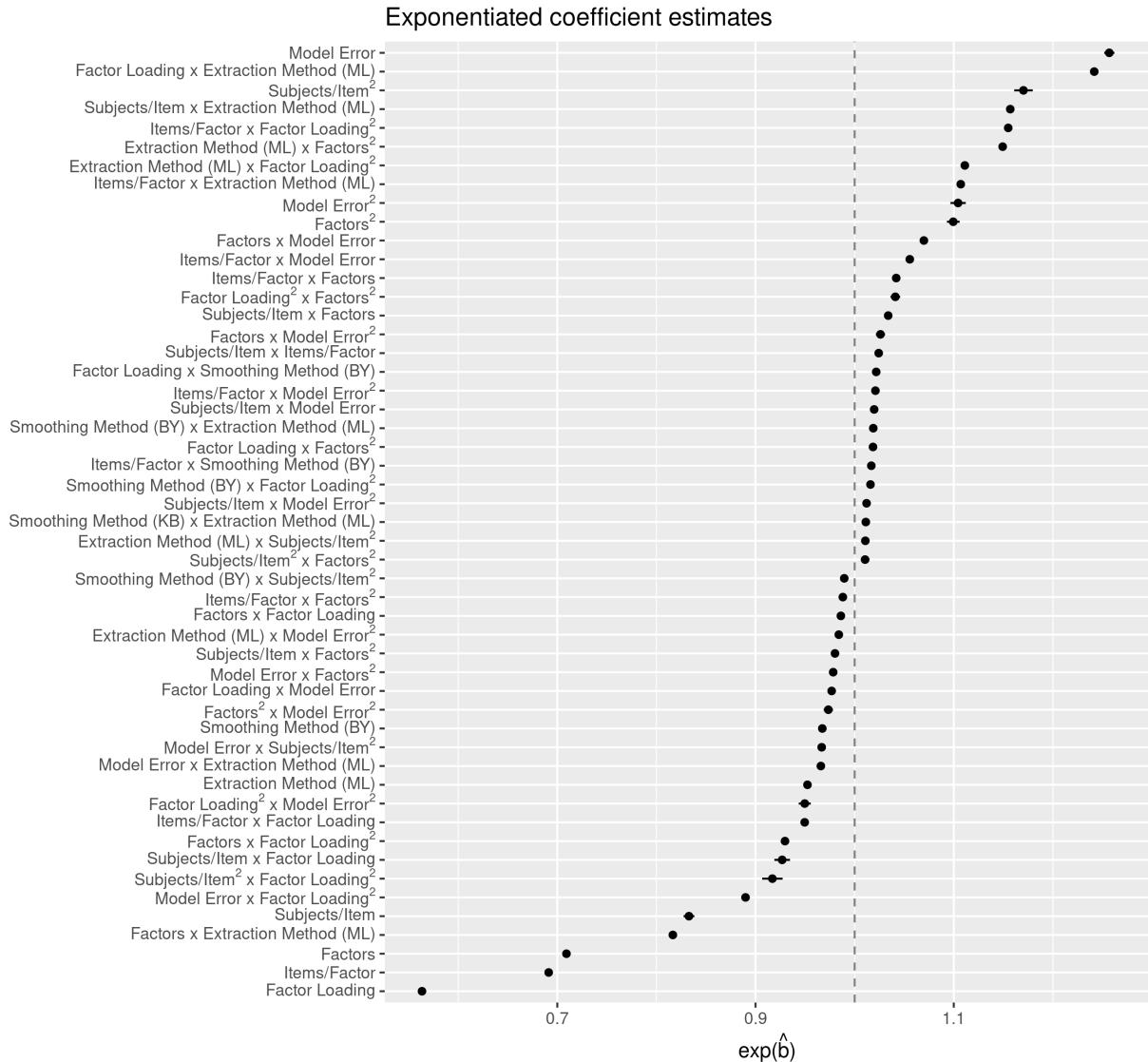


Figure 3. Exponentiated coefficient estimates for the mixed effects model using $\log[\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})]$ as the dependent variable (Model 2B). APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); ML = Maximum likelihood; PA = Principal axis. The effects of no smoothing and ordinary least squares factor analysis are subsumed within the Constant term.

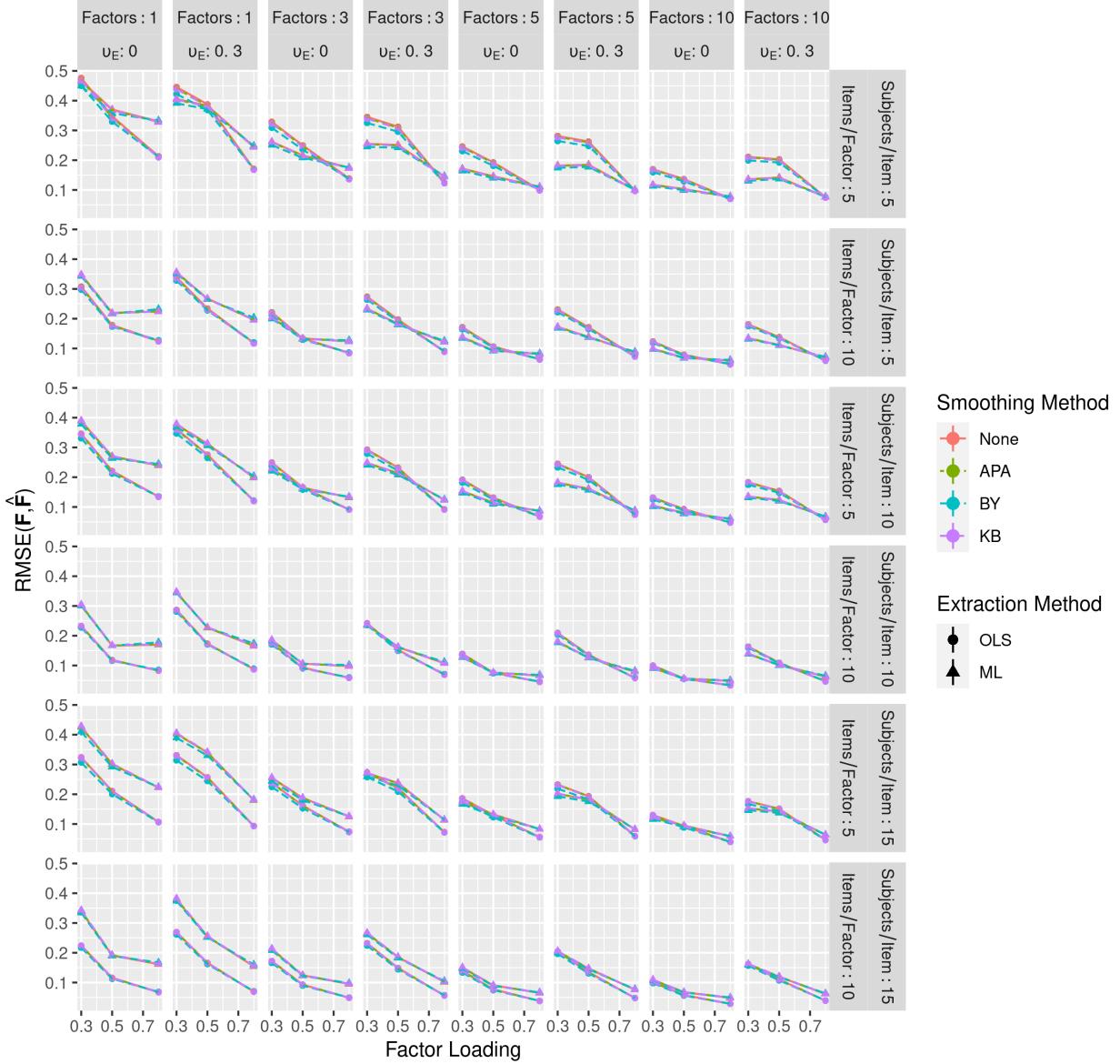


Figure 4. Estimated marginal mean $\text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ values and 99% confidence intervals. To conserve space, the intermediate values of model error and subjects per item have been omitted. The principal axis factor extraction method was also omitted because it led to nearly identical results compared to ordinary least squares. OLS = ordinary least squares; ML = maximum likelihood; APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991).

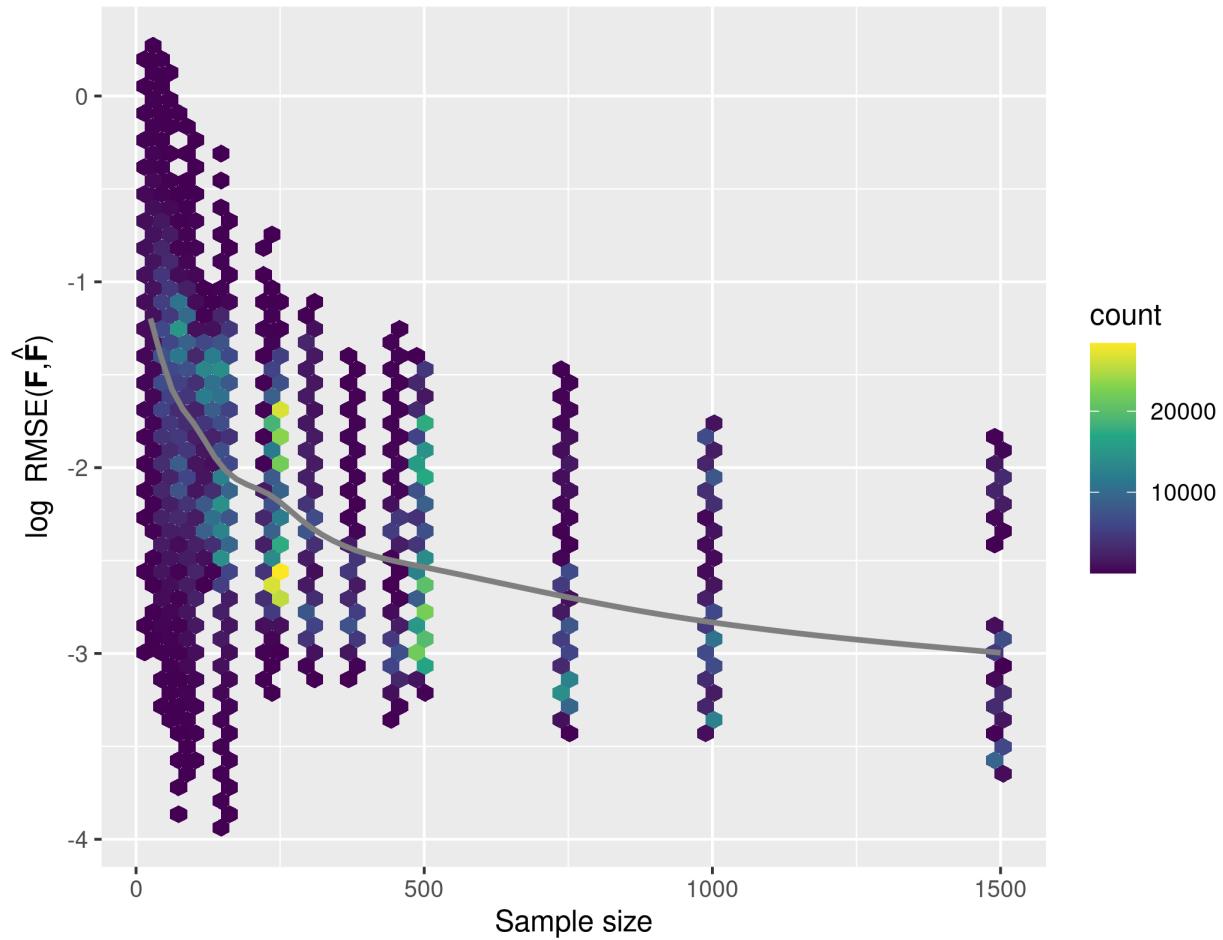


Figure 5. Log root-mean-square error (RMSE) between the true and estimated factor loading matrices as a function of sample size.

Appendix A

Regression Diagnostics

922 Models 1A and 1B: Regression models predicting $\log D_s(R_{Sm}, R_{Pop})$

923 Models 1A and 1B were a linear mixed-effects models predicting the (log) scaled
 924 distance between the smoothed and model-implied population correlation matrix and was
 925 fit using the R *lme4* package (Version 1.1.23; Bates, Mächler, Bolker, & Walker, 2015).
 926 Model 1A was a linear model fit using all simulation variables and their interactions. In
 927 Model 1B, second-degree polynomial terms were added for number of factors, number of
 928 subjects per item, factor loading, and model error. Diagnostic plots showing standardized
 929 residuals plotted against fitted values for both models, quantile-quantile (QQ) plots of the
 930 residuals, and QQ plots for the random intercept terms are shown in Figures A3, A1, and
 931 A2 respectively. These plots show that some assumptions of the linear mixed-effects model
 932 seem to have been violated for Models 1A and 1B, even after applying a log-transformation
 933 to the response variable.

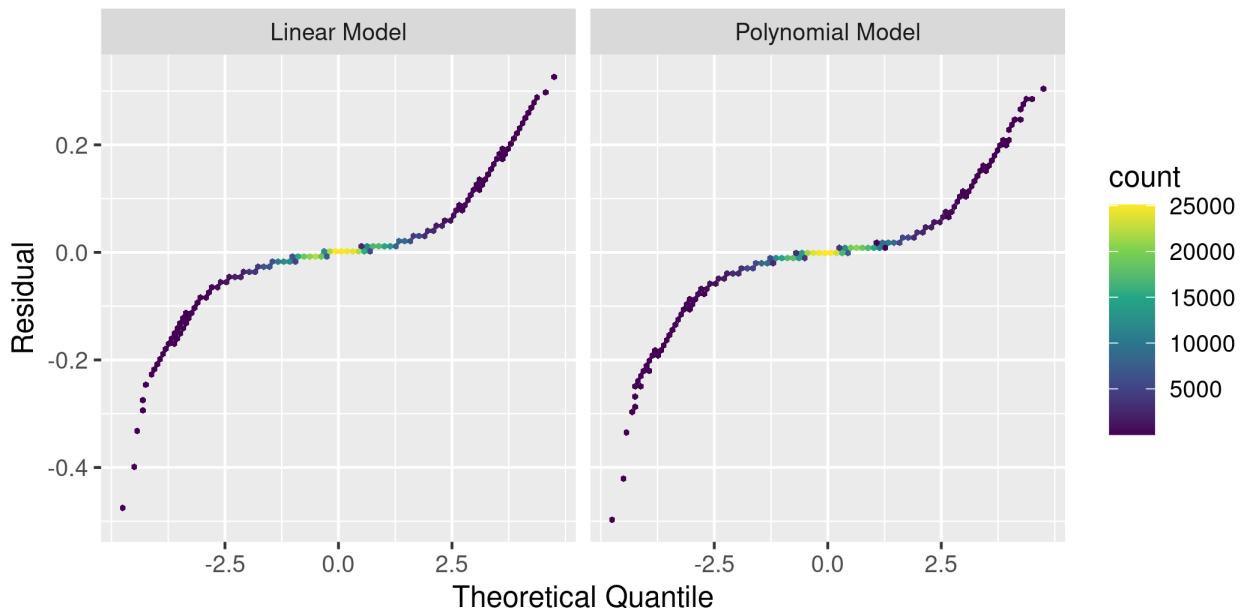


Figure A1. Quantile-quantile plot of residuals for Models 1A and 1B.

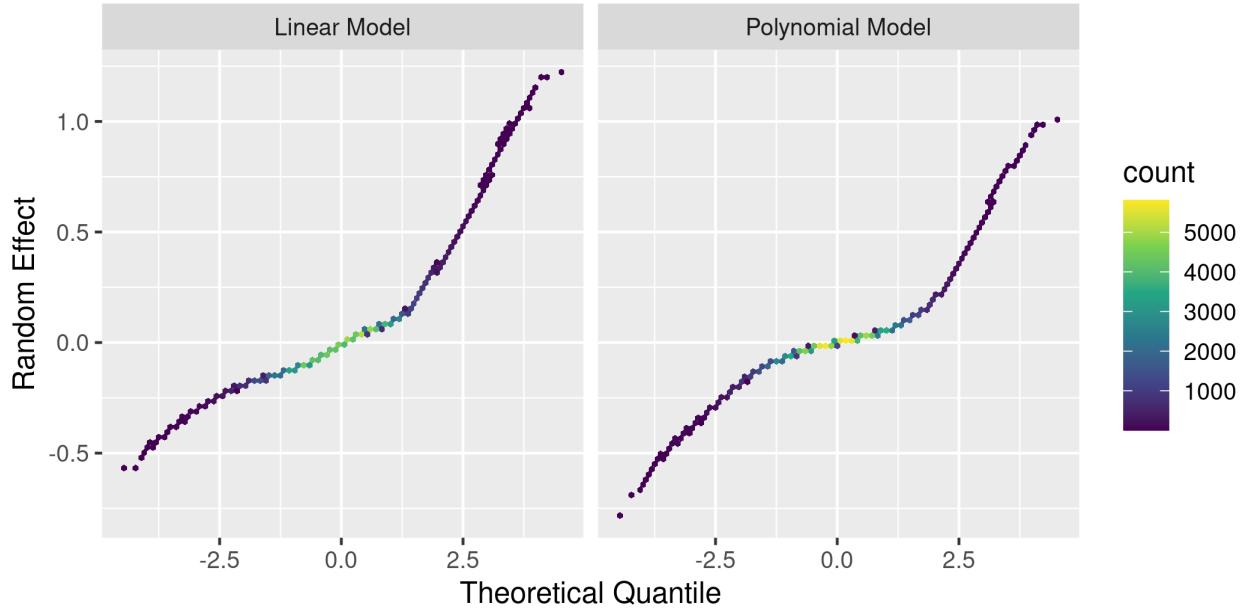


Figure A2. Quantile-quantile plot of random intercept terms for Models 1A and 1B.

934 Figure A3 shows that the variance of the residuals was not constant over the range of
 935 fitted values for both the linear and polynomial models. In particular, for both models
 936 there was little variation near the edges of the range of fitted values and a large amount of
 937 variation near the center of the distribution of fitted values. Therefore, the
 938 homoscedasticity assumption seemed to have been violated. Moreover, Figure A1 shows
 939 that the assumption of normally-distributed errors was also likely violated. In particular,
 940 Figure A1 shows that the distributions of residuals (for both models) had heavy tails and
 941 had a slight positive skew (Model 1A: kurtosis = 16.25, skew = 0.60; Model 1B: kurtosis =
 942 18.61, skew = 0.23). Finally, Figure A2 shows that the random effects (random intercepts)
 943 were not normally-distributed for either the linear or polynomial model (Model 1A: kurtosis
 944 = 5.52, skew = 1.52; Model 1B: kurtosis = 10.33, skew = 0.59). To address these violations
 945 of the model assumptions, I first attempted to fit a robust mixed-effects model using
 946 `rlmer()` function in the R *robustlmm* package (Version 2.3; Koller, 2016). Unfortunately,
 947 the data set was too large for the `rlmer()` function to handle. I also tried a more complex
 948 transformation of the dependent variable (using a Box-Cox power transformation; Box &

⁹⁴⁹ Cox, 1964), but it produced no discernible benefit compared to a log transformation.

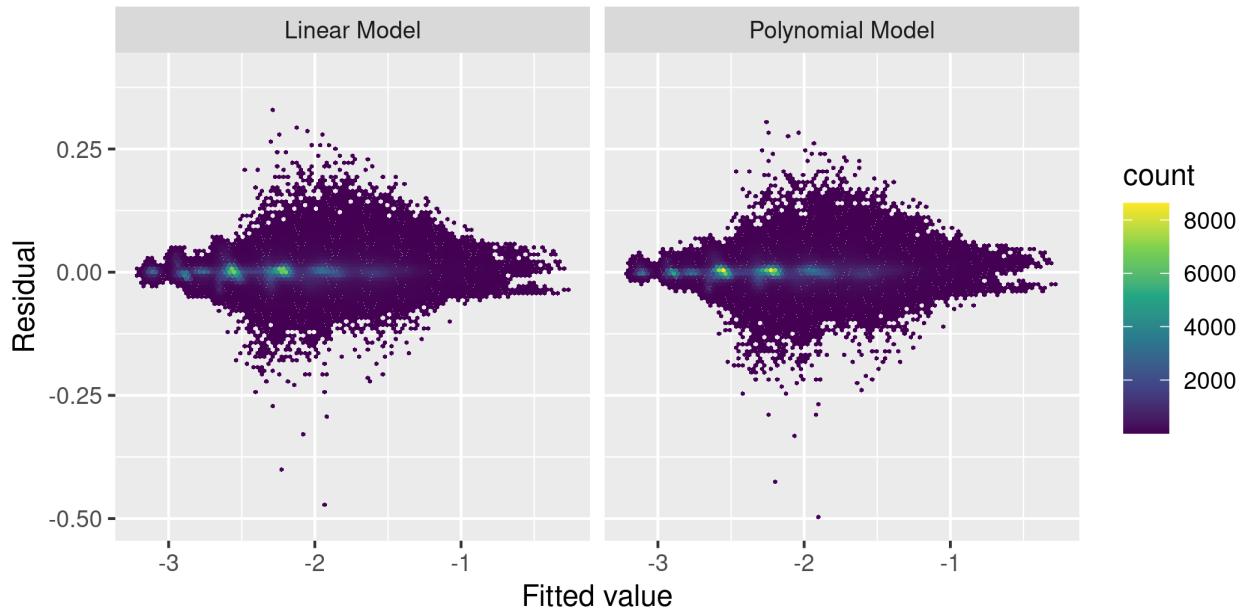


Figure A3. Residuals plotted against fitted values for Models 1A and 1B.

⁹⁵⁰ The apparent violations of the assumptions of the mixed-effects model were
⁹⁵¹ concerning. However, inference for the fixed effects in mixed-effects models seems to be
⁹⁵² somewhat robust to these violations. In particular, Jacqmin-Gadda, Sibillot, Proust,
⁹⁵³ Molina, & Thiébaut (2007) showed that inference for fixed effects is robust for
⁹⁵⁴ non-Gaussian and heteroscedastic errors. Moreover, Jacqmin-Gadda et al. (2007) cited
⁹⁵⁵ several studies indicating that inference for fixed effects is also robust to non-Gaussian
⁹⁵⁶ random effects (Butler & Louis, 1992; Verbeke & Lesaffre, 1997; Zhang & Davidian, 2001).
⁹⁵⁷ Finally, the purpose of the present analysis was to obtain estimates of the fixed effects of
⁹⁵⁸ matrix smoothing methods (and the interactions between smoothing methods and the
⁹⁵⁹ other design factors) on population correlation matrix recovery. Neither *p*-values nor
⁹⁶⁰ confidence intervals were of primary concern. Therefore, the apparent violation of some
⁹⁶¹ model assumptions likely did not affect the main results of this study.

962 **Models 2A and 2B: Regression models predicting $\log \text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$**

963 Models 2A and 2B were mixed-effects models predicting $\log \text{RMSE}(\mathbf{F}, \hat{\mathbf{F}})$ and fit
 964 using the R *lme4* package (Bates, Mächler, Bolker, & Walker, 2015). Model 2A was a
 965 linear model fit using all simulation variables and their interactions. In Model 2B,
 966 second-degree polynomial terms were added for number of factors, number of subjects per
 967 item, factor loading, and model error. As with Models 1A and 1B, diagnostic plots showing
 968 standardized residuals plotted against fitted values for both models, QQ plots for the
 969 residuals, and QQ plots for the random intercept terms are shown in Figures A3, A5, and
 970 A6 respectively.

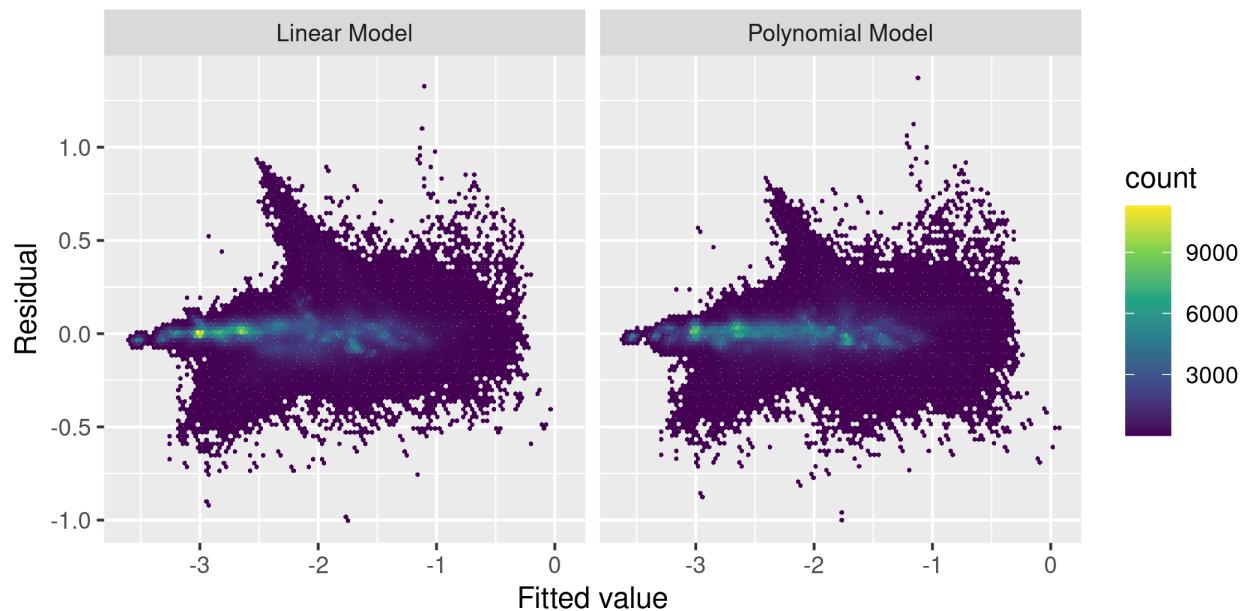


Figure A4. Residuals plotted against fitted values for Models 2A and 2B.

971 These plots indicate many of the same issues in Models 2A and 2B as were seen for
 972 Models 1A and 1B. First, Figure A3 shows clear evidence of non-homogeneous conditional
 973 error variance for both the linear and polynomial models. Specifically, the residual variance
 974 seemed generally to be larger for larger fitted values. Second, Figure A5 shows that the
 975 distribution of residuals for both models was non-normal and similar to the distributions of

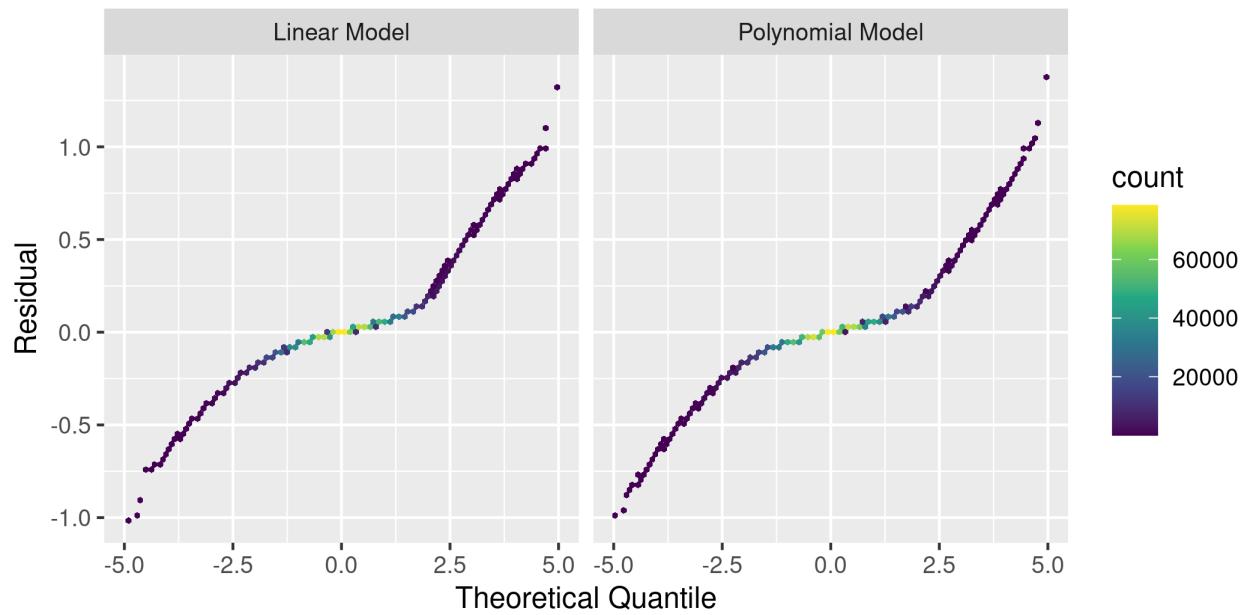


Figure A5. Quantile-quantile plot of residuals for Models 2A and 2B.

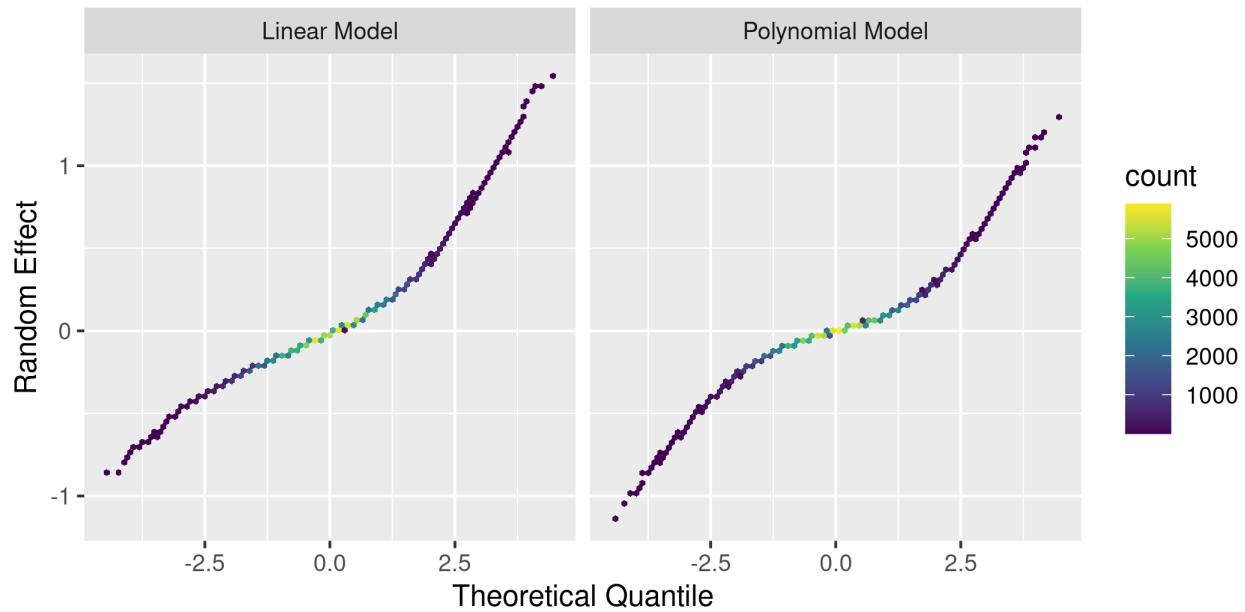


Figure A6. Quantile-quantile plot of random intercept terms for Models 2A and 2B.

the residuals from Model 1A and 1B (i.e., positively-skewed and having heavy tails).
Finally, Figure A6 shows that the estimated random effects were likewise not normally-distributed. The distribution of random intercepts was positively-skewed with heavy tails. Alternative transformations of the dependent variable were tried but did not seem to 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.

Appendix B
Supplemental Tables and Figures

985 Indefinite Matrix Frequency

986 The percent of indefinite tetrachoric correlation matrices differed from condition to
987 condition. Table B1 reports the percent of indefinite matrices for each of the 216 conditions
988 of the study design. One of the more obvious trends in this table is that conditions with
989 more (major) factors tended to produce more indefinite tetrachoric correlation matrices.
990 Based on the results reported by Debelak and Tran (2013; 2016), who found that indefinite
991 tetrachoric and polychoric correlation matrices were much more common for data sets with
992 many items, this is likely due to the correlation of factor number with total number of
993 items. (See Lorenzo-Seva and Ferrando, 2020, for further discussion of the relationship
994 between the number of items and matrix indefiniteness.) Moreover, the results in Table B1
995 indicate that indefinite matrices were more common for conditions with more items per
996 factor, fewer subjects per item, and higher factor loadings. All of these trends corroborate
997 the similar results by Debelak and Tran (2013; 2016) and their conclusions about which
998 variables most affected the frequency of indefinite tetrachoric or polychoric correlation
999 matrices.

Table B1

Percent of indefinite tetrachoric correlation matrices by Number of Subjects Per Item (N/p), Number of Items per Factor (p/m), Factor Loading, Model Error (v_E), and Number of Factors.

N/p	p/m	Loading	v_E	Factors			
				1	3	5	10
5	5	0.3	0.0	10.5	96.6	100.0	100.0
5	5	0.3	0.1	10.6	97.3	100.0	100.0
5	5	0.3	0.3	13.5	99.3	100.0	100.0
5	5	0.5	0.0	15.6	98.9	100.0	100.0
5	5	0.5	0.1	14.4	99.0	100.0	100.0
5	5	0.5	0.3	15.6	100.0	100.0	100.0
5	5	0.8	0.0	13.5	100.0	100.0	100.0
5	5	0.8	0.1	13.4	100.0	100.0	100.0
5	5	0.8	0.3	13.9	100.0	100.0	100.0
5	10	0.3	0.0	78.0	100.0	100.0	100.0
5	10	0.3	0.1	79.1	100.0	100.0	100.0
5	10	0.3	0.3	85.5	100.0	100.0	100.0
5	10	0.5	0.0	88.1	100.0	100.0	100.0
5	10	0.5	0.1	89.3	100.0	100.0	100.0
5	10	0.5	0.3	94.1	100.0	100.0	100.0
5	10	0.8	0.0	98.9	100.0	100.0	100.0
5	10	0.8	0.1	99.3	100.0	100.0	100.0
5	10	0.8	0.3	99.6	100.0	100.0	100.0
10	5	0.3	0.0	2.5	7.9	9.4	5.8
10	5	0.3	0.1	2.0	10.5	12.3	18.1
10	5	0.3	0.3	3.3	26.8	54.6	93.1
10	5	0.5	0.0	3.4	21.3	29.0	49.1
10	5	0.5	0.1	3.5	26.0	38.1	70.8
10	5	0.5	0.3	4.4	48.8	82.5	99.7
10	5	0.8	0.0	13.1	98.4	100.0	100.0
10	5	0.8	0.1	11.4	98.2	100.0	100.0
10	5	0.8	0.3	12.5	99.3	100.0	100.0
10	10	0.3	0.0	8.7	8.0	7.9	5.1
10	10	0.3	0.1	11.0	14.0	19.5	38.2
10	10	0.3	0.3	21.2	70.2	94.4	100.0
10	10	0.5	0.0	23.8	39.5	63.3	94.8
10	10	0.5	0.1	24.2	56.3	83.7	99.9
10	10	0.5	0.3	39.8	94.4	100.0	100.0
10	10	0.8	0.0	84.2	100.0	100.0	100.0
10	10	0.8	0.1	83.3	100.0	100.0	100.0

10	10	0.8	0.3	89.9	100.0	100.0	100.0
15	5	0.3	0.0	0.4	0.0	0.0	0.0
15	5	0.3	0.1	0.2	0.1	0.0	0.0
15	5	0.3	0.3	0.8	1.4	1.2	0.9
15	5	0.5	0.0	0.8	0.8	0.0	0.0
15	5	0.5	0.1	0.5	1.1	0.7	0.2
15	5	0.5	0.3	1.0	4.9	7.9	18.7
15	5	0.8	0.0	9.4	65.9	87.4	100.0
15	5	0.8	0.1	9.3	69.1	92.1	100.0
15	5	0.8	0.3	9.3	85.4	99.2	100.0
15	10	0.3	0.0	0.3	0.0	0.0	0.0
15	10	0.3	0.1	0.5	0.0	0.0	0.0
15	10	0.3	0.3	3.7	2.2	1.1	2.0
15	10	0.5	0.0	2.8	0.3	0.0	0.0
15	10	0.5	0.1	3.6	0.5	0.0	0.1
15	10	0.5	0.3	7.1	14.5	29.8	78.0
15	10	0.8	0.0	51.0	96.4	100.0	100.0
15	10	0.8	0.1	51.8	99.2	100.0	100.0
15	10	0.8	0.3	65.9	100.0	100.0	100.0

1000 Observed $D_s(R_{Sm}, R_{Pop})$ Values

1001 In addition to the estimated marginal means shown in the main text, the following
1002 figures (Figures B1–B4) show box-plots of $D_s(R_{Sm}, R_{Pop})$ for each condition in the
1003 simulation design. These box-plots match well with the estimated marginal means shown in
1004 the main text. However, notice that some conditions in these figures are missing box-plots
1005 (e.g., three factors, 15 subjects per item, 10 items per factor, $v_E = 0$, and Loading = 0.3)
1006 because no indefinite tetrachoric correlation matrices were produced for those conditions.

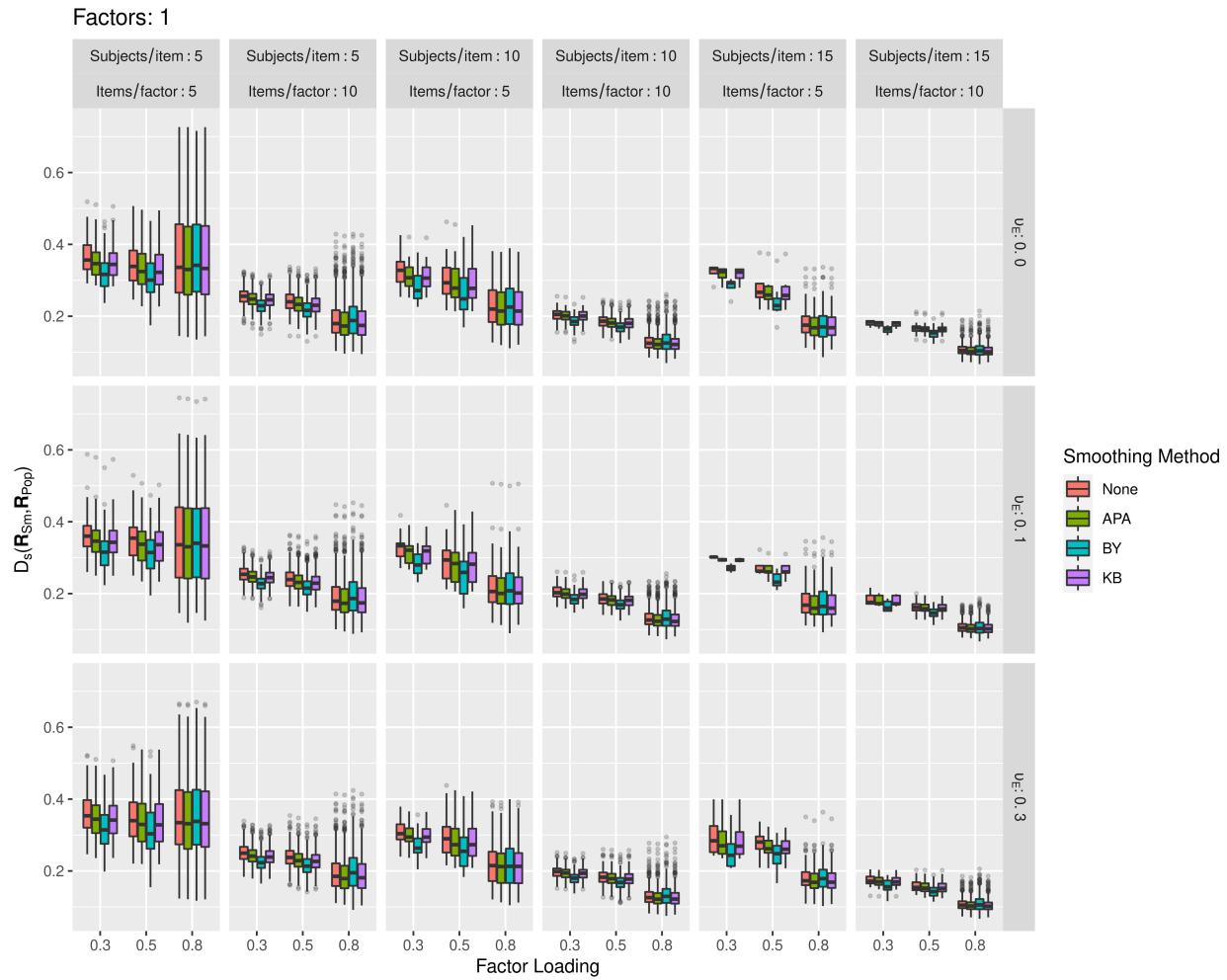


Figure B1. $D_s(R_{Sm}, R_{Pop})$ values for one-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.

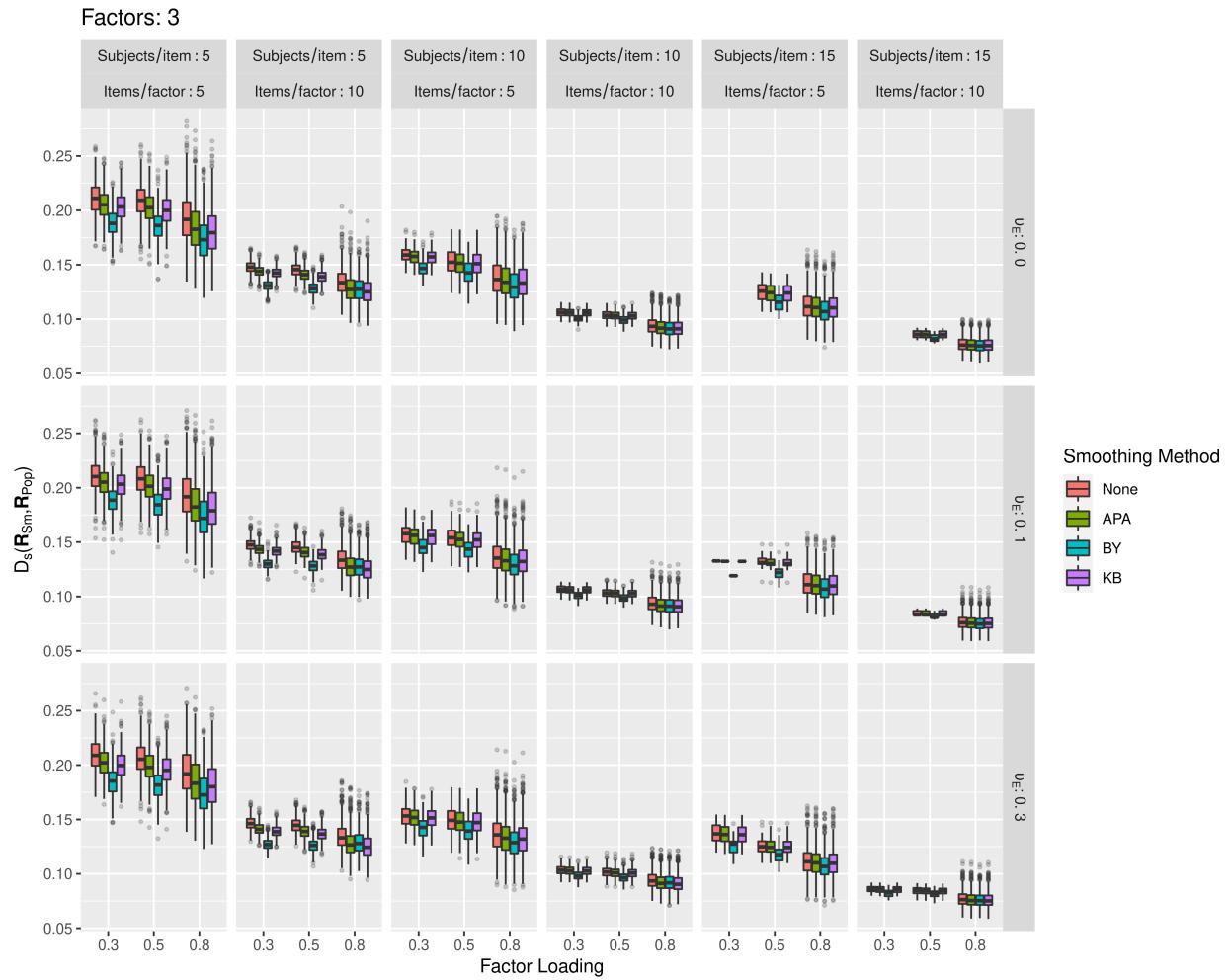


Figure B2. $D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})$ values for three-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.

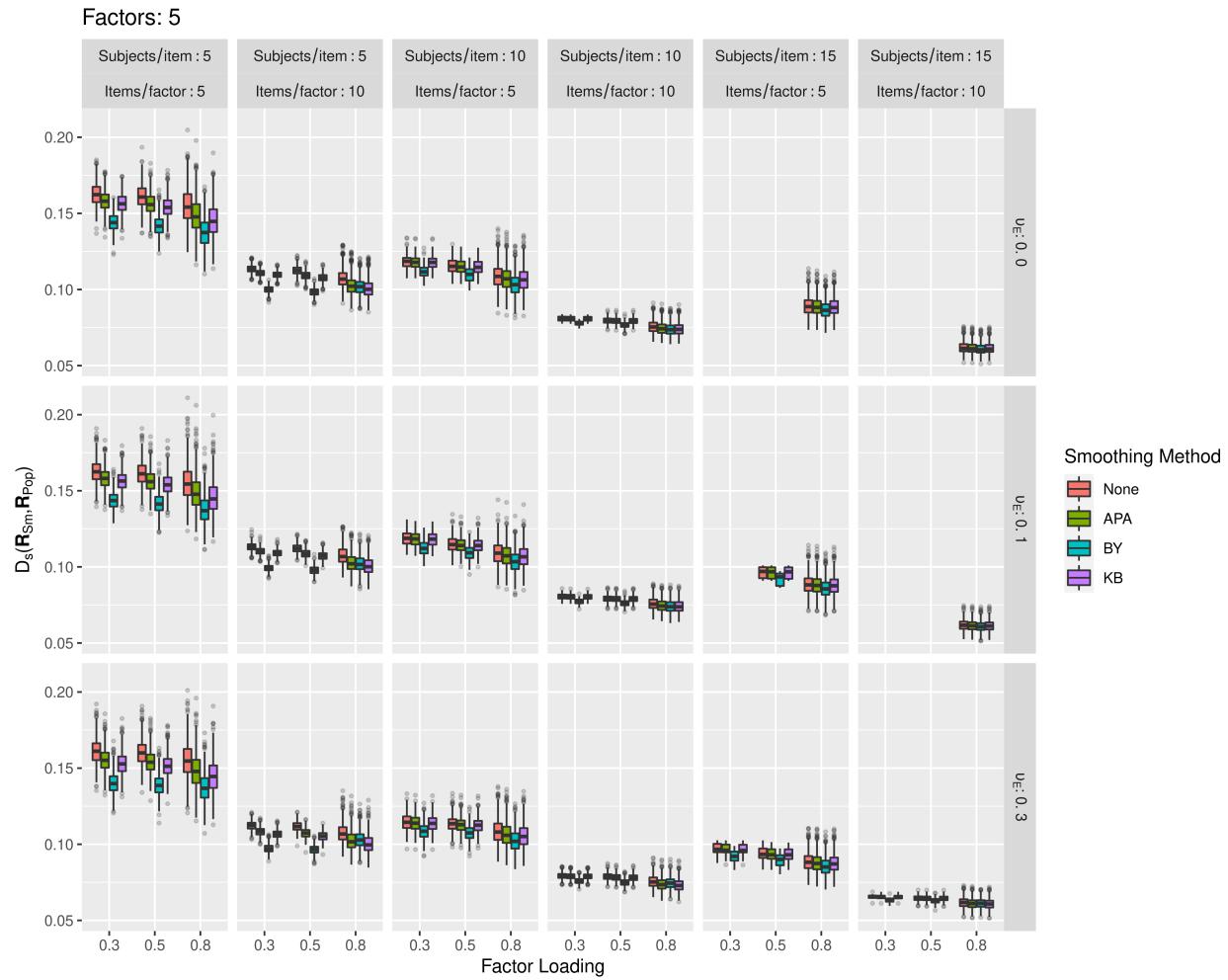


Figure B3. D_s(R_{Sm}, R_{Pop}) values for five-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.

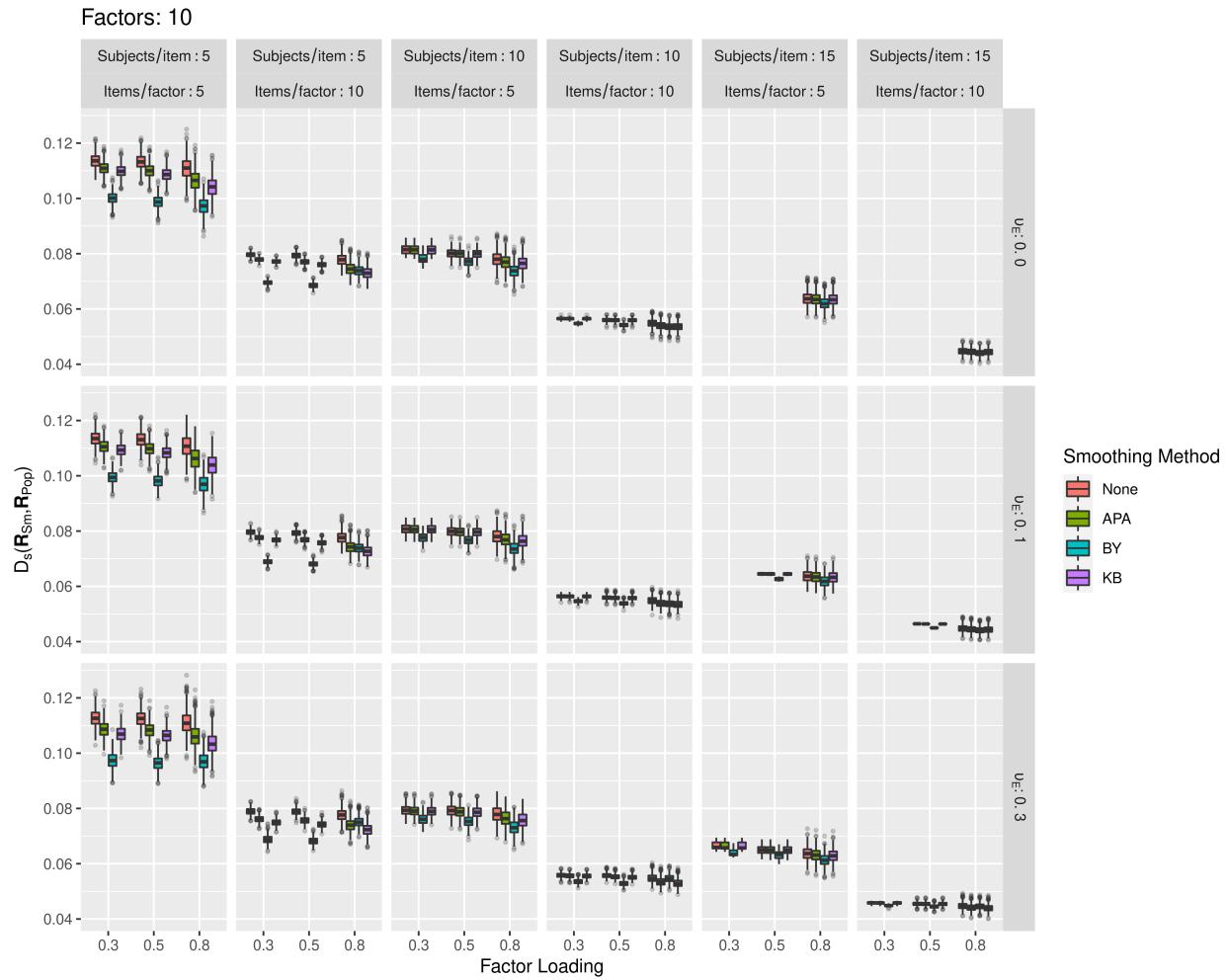


Figure B4. $D_s(\mathbf{R}_{Sm}, \mathbf{R}_{Pop})$ values for ten-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.

1007 **Observed RMSE(\mathbf{F} , $\hat{\mathbf{F}}$) Values**

1008 Figures B5–B8 in this section show box-plots of RMSE(\mathbf{F} , $\hat{\mathbf{F}}$) for each condition in
1009 the study design. Similar to the figures in the previous section, these box-plots for the
1010 most part agree well with the estimated marginal means presented in the main text, but
1011 are missing data for conditions with no indefinite tetrachoric correlation matrices.

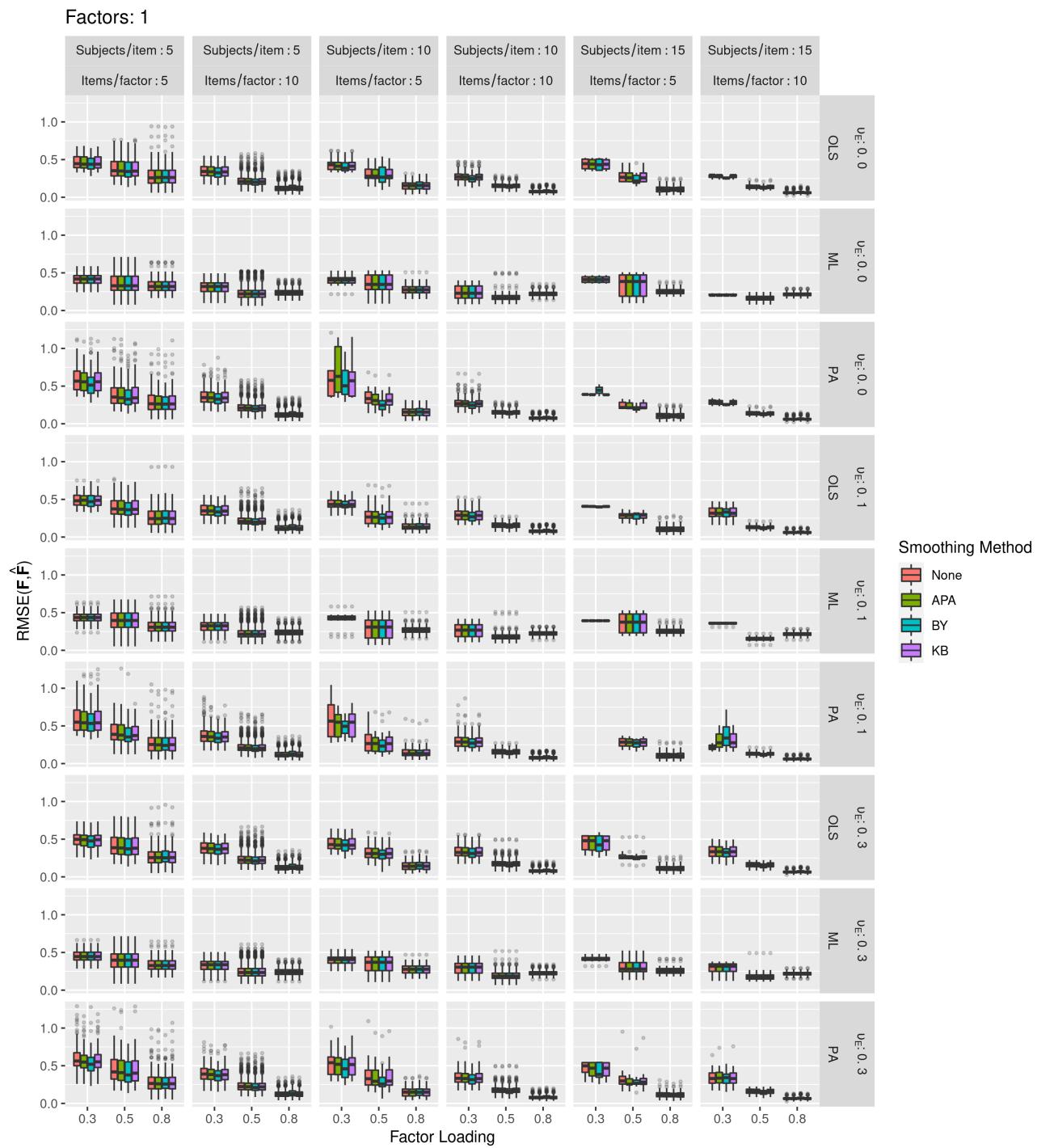


Figure B5. RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values for one-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; OLS = Ordinary least squares; ML = Maximum likelihood; PA = Principal axis; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.

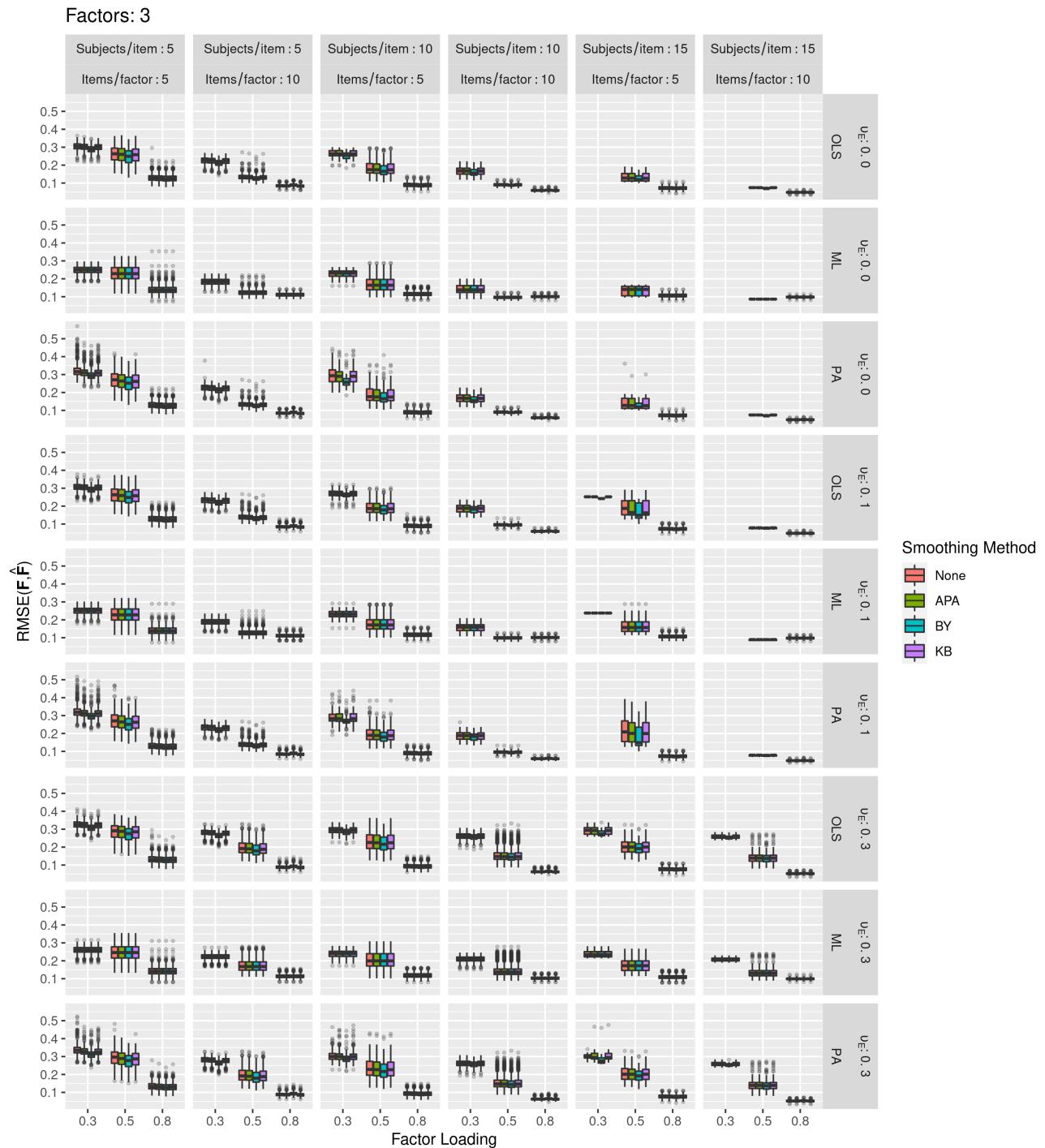


Figure B6. RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values for three-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; OLS = Ordinary least squares; ML = Maximum likelihood; PA = Principal axis; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.

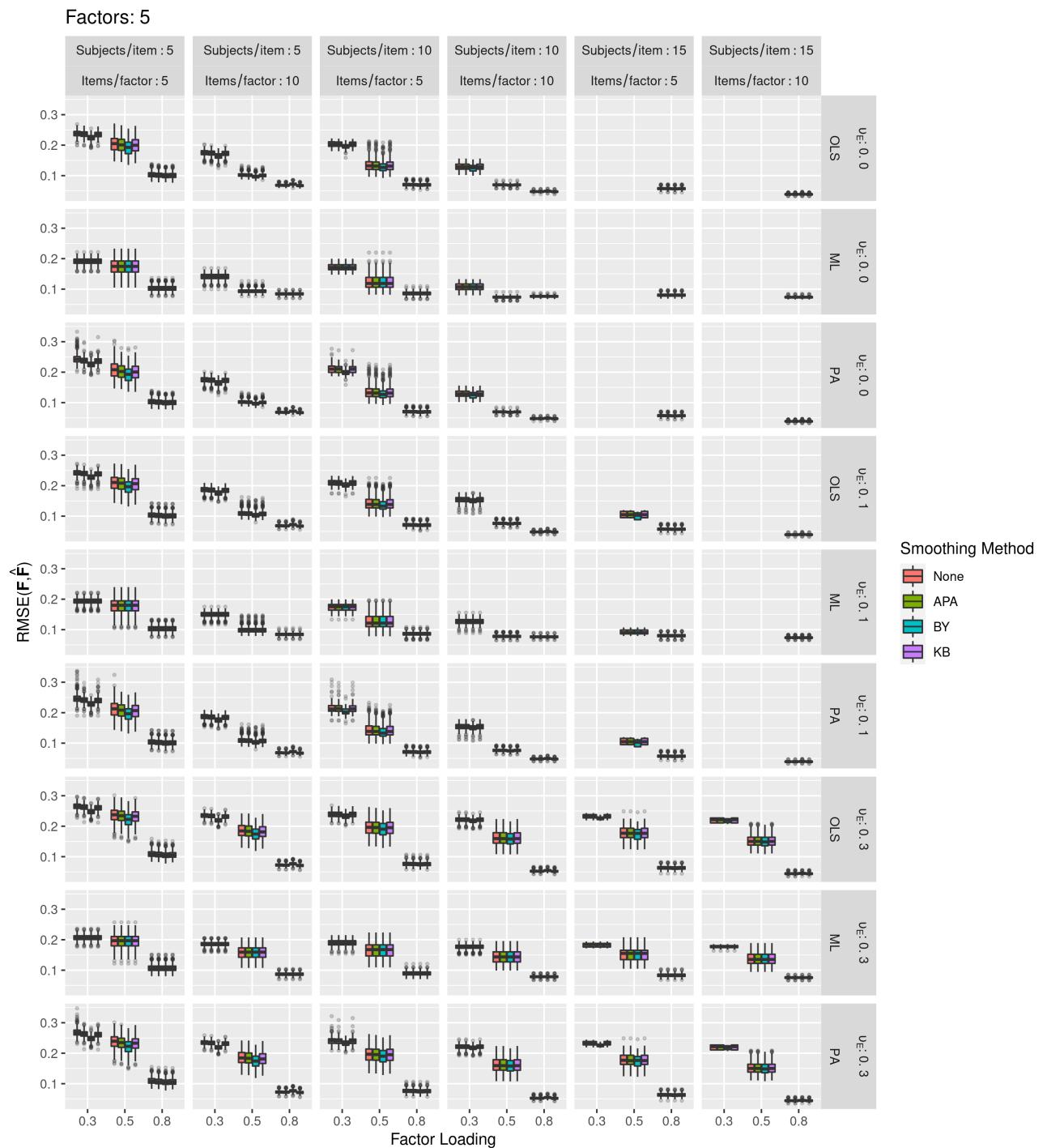


Figure B7. RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values for five-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; OLS = Ordinary least squares; ML = Maximum likelihood; PA = Principal axis; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.

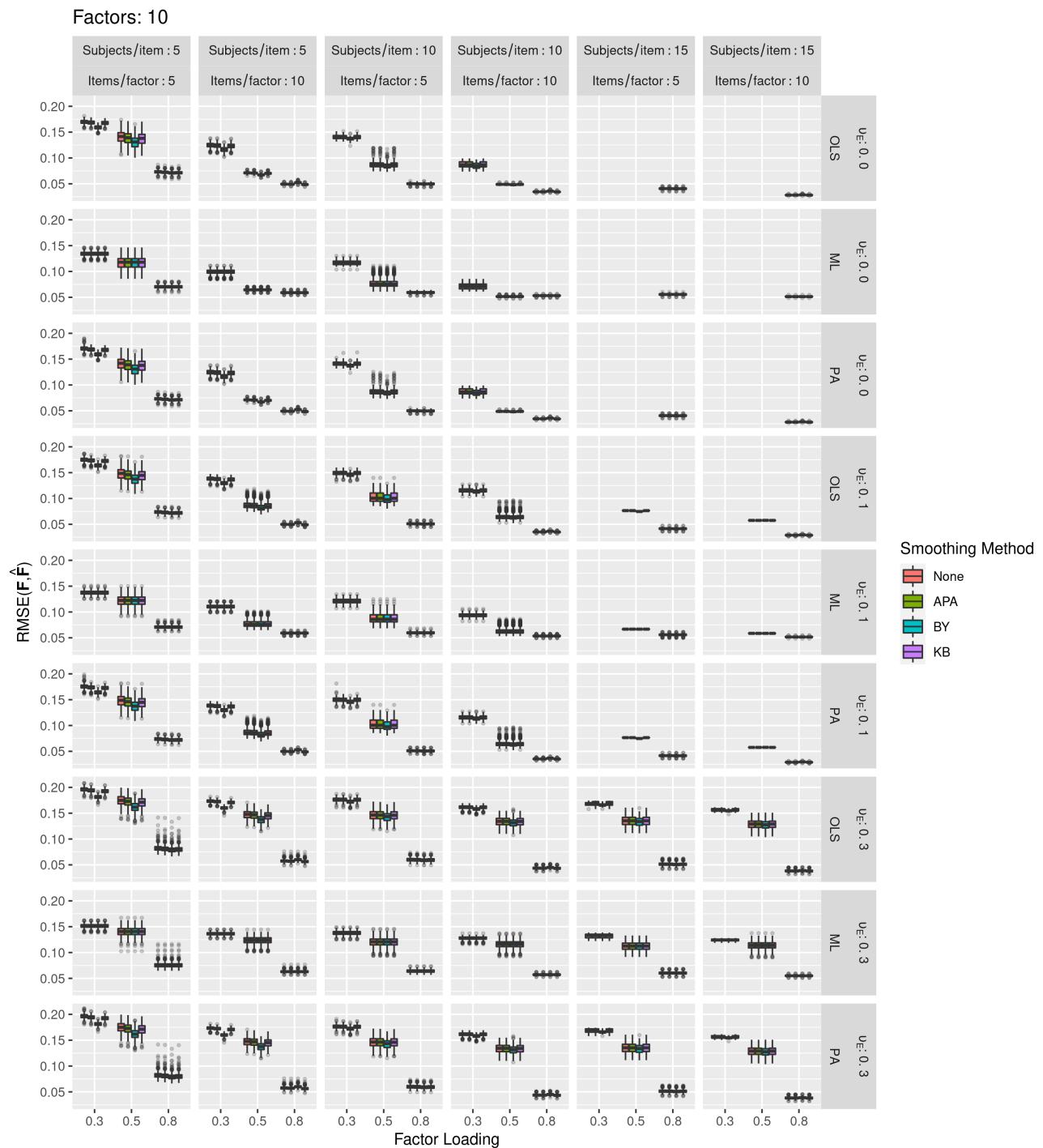


Figure B8. RMSE($\mathbf{F}, \hat{\mathbf{F}}$) values for ten-factor models. APA = Higham (2002); BY = Bentler-Yuan (2011); KB = Knol-Berger (1991); None = no smoothing; OLS = Ordinary least squares; ML = Maximum likelihood; PA = Principal axis; v_E = Proportion of uniqueness variance redistributed to the minor common factors representing model approximation error.