The Stata Journal (2009)
9, Number 3, pp. 329–373

Confirmatory factor analysis using confa

Stanislav Kolenikov
Department of Statistics
University of Missouri
Columbia, MO
kolenikovs@missouri.edu

Abstract. This article describes the `confa` command, which fits confirmatory factor analysis models by maximum likelihood and provides diagnostics for the fitted models. Descriptions of the command and its options are given, and some illustrative examples are provided.

Keywords: st0169, confa, confa postestimation, bollenstine, Bollen–Stine bootstrap, confirmatory factor analysis, factor scores, Satorra–Bentler corrections

1 Confirmatory factor analysis (CFA)

In a wide range of research problems, especially in the social sciences, the researcher may not have access to direct measurements of the variables of interest; for example, intellectual ability is not something that can be measured in centimeters or kilograms. However, people who are more able can work on mental problems faster, make fewer errors, or solve more difficult problems. These differences between individual abilities underlie IQ tests. A more careful analysis might distinguish different dimensions of an intellectual ability, including reasoning on verbal, spatial, logical, and other kinds of problems. As another example, liberal democracy is a characteristic of a society that will not have natural measurement units associated with it (unlike, say, gross domestic product per capita as a measure of economic development). Political scientists would have to rely on expert judgment comparing different societies in terms of how much political freedom citizens may have or how efficient democratic rule is.

In the above problems, researchers will not have accurate measurements of the main variable of interest. Instead, they operate with several proxy variables that share correlation with that (latent) variable but also contain measurement error. A popular tool to analyze problems of this kind is confirmatory factor analysis (CFA). This is a multivariate statistical technique used to assess the researcher's theory, which suggests the number of (latent, or unobserved) factors and their relation to the observed variables, or indicators (Lawley and Maxwell 1971; Bartholomew and Knott 1999; Brown 2006). CFA can be viewed as a subfield of structural equation modeling (SEM) with latent variables (Bollen 1989) when the latent variables are all assumed to be exogenous. The terms "latent variables", "factors", and "latent factors" will be used interchangeably in this article.

The method differs substantially from exploratory factor analysis (EFA). In EFA, the number of factors and their relation to the observed variables is unknown in advance.

The researcher fits several models and compares them using fit criteria, analysis of eigenvalues of certain (functions of) variance–covariance matrices, or substantive considerations. Once the number of factors and the linear subspace of the factors are determined, the researcher tries to find a rotation that would separate variables into groups so that variables within the same group are highly correlated with one another and are said to originate from the same factor. The factors are constructed to be uncorrelated.

In CFA, the model structure must be specified in advance: the number of factors is postulated, as well as relations between those factors and observed variables. The researcher must specify which variables are related to which factor(s). The complete structure of the model is specified in advance. An advantage of this approach is that it permits the usual statistical inference to be performed: the standard errors of the estimated coefficients can be obtained and model tests can be performed.

In Stata, EFA is available via the `factor` estimation command and the associated suite of postestimation commands. See [MV] **factor**.

1.1 The model and identification

Let us denote the unobserved latent factors with ξ_k, $k = 1, \ldots, m$, where m is the number of factors that need to be specified a priori. Let the observed variables be y_j, $j = 1, \ldots, p$. Let index $i = 1, \ldots, n$ enumerate observations. In typical application of CFA, there will be a handful of factors (sometimes just one factor) with several variables per factor. Large psychometric scales may contain as many as several dozen or more than a hundred questions, although most items will be binary rather than continuous.

Linear relations are postulated to hold between the factors and observed variables,

$$y_{ij} = \mu_j + \sum_{k=1}^{m} \lambda_{jk}\xi_{ik} + \delta_{ij}, \quad j = 1, \ldots, p \tag{1}$$

where μ_j is the intercept; λ_{jk} are regression coefficients, or factor loadings; and δ_j are measurement errors, or unique errors. In matrix form, (1) can be written as

$$\mathbf{y}_i = \mu + \Lambda\xi_i + \delta_i \tag{2}$$

where vectors μ, ξ_i, and δ_i denote regression intercepts, latent variables, and measurement errors, respectively, and Λ is the matrix of factor loadings. The measurement errors, δ, are assumed to be independent of the factors, ξ. Let us additionally introduce the (matrices of) parameters

$$\Phi = V(\xi) = E(\xi\xi'), \quad \Theta = V(\delta) = E(\delta\delta')$$

using the usual convention that $E(\xi) = 0$, $E(\delta) = 0$. Then the covariance matrix of the observed variables is

$$V(\mathbf{y}) = E\left\{(\mathbf{y}-\mu)(\mathbf{y}-\mu)'\right\} = E\left\{(\Lambda\xi_i+\delta_i)(\Lambda\xi_i+\delta_i)'\right\} = \Lambda\Phi\Lambda' + \Theta = \Sigma(\theta) \qquad (3)$$

where all parameters are put together into vector θ.

Let us highlight the distinctions between EFA and CFA again using the matrix formulation (3). EFA assumes that matrices Φ and Θ are diagonal, and matrix Λ is freely estimated (and rotated if needed). CFA assumes that matrix Λ has a strong structure with zeroes (or other constraints) in several places, as dictated by researcher's substantive theory. In fact, the most common structure of this matrix is known as the model of factor complexity 1: each variable loads on only one factor. Then Λ has a block structure:

$$\Lambda = \begin{pmatrix} \Lambda_1 & 0 & \dots & 0 \\ 0 & \Lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Lambda_m \end{pmatrix}$$

Other restrictions and corresponding structure of the Λ matrix can be entertained depending on the model.

Before the researcher proceeds to estimation, he or she needs to establish that the model is identified (Bollen 1989). Identification means that no two different sets of parameters can produce the same means and covariance matrix (3).

The minimal set of identification conditions in any latent variable modeling is to set the location and the scale of the latent variables. The former is usually achieved by setting the mean of the latent variable to zero, and that is the convention adopted by `confa`.

There are two common ways to identify the scales of latent factors. One can set the variance of the latent variable ξ_k to 1. Alternatively, one can set one of the loadings λ_{jk} to a fixed number, most commonly 1. Then the latent variable will have the units of that observed variable, which might be useful if the observed variable is meaningful (e.g., the latent variable is wealth, and the observed variable is annual income, in dollars).

A necessary identification condition is that the number of parameters, t, of the model does not exceed the degrees of freedom in the model. In covariance structure modeling (and in CFA, as a special case), this is the number of the nonredundant entries of the covariance matrix (3):

$$\dim\theta = t \le p^* = p(p+1)/2$$

where t is the number of parameters describing the covariance structure. (As long as zero values are assumed for the means of the factors and errors, the mean structure is said to be saturated, and the estimates of μ are the corresponding means, $\widehat{\mu}_j = \overline{y}_j$.) If $t = p^*$, the model is said to be *exactly identified*, and if $t > p^*$, it is said to be *overidentified*. In the latter case, additional degrees of freedom can be used to test for model fit; see below.

There are additional conditions related to identification of the latent structure of the model. Several sufficient identification rules have been developed for CFA. Bollen (1989) lists the following rules:

1. *Three indicator rule*. If the model has factor complexity 1, the covariance matrix of the error terms, $V(\delta) = \Theta$, is diagonal, and each factor has at least three indicators (observed variables associated with that factor), then the CFA model is identified.

2. *Two indicator rule*. If the model has factor complexity 1, the covariance matrix of the error terms, $V(\delta) = \Theta$, is diagonal, there is more than one factor in the model ($m > 1$), each row of Φ has at least one nonzero, off-diagonal element, and each factor has at least two indicators, then the CFA model is identified.

1.2 Estimation, testing, and goodness of fit

One of the most popular methods to estimate the parameters in (1) or (2) is by maximum likelihood (Jöreskog 1969). If assumptions of i.i.d. data and of the multivariate normality of the observed data (equivalent to the assumption of multivariate normality of ξ and δ) are made, then the log likelihood of the data is

$$\begin{aligned}\ln L\{\mathbf{Y}, \Sigma(\theta)\} &= -\sum_{i=1}^{n}\left\{\frac{p}{2}\ln 2\pi + \frac{1}{2}\ln\mid\Sigma(\theta)\mid + \frac{1}{2}(\mathbf{y}_i-\mu)'\Sigma^{-1}(\theta)(\mathbf{y}_i-\mu)\right\}\\ &= -\frac{np}{2}\ln 2\pi - \frac{n}{2}\ln\mid\Sigma(\theta)\mid - \frac{1}{2}\operatorname{tr}\Sigma^{-1}(\theta)S \qquad (4)\end{aligned}$$

where S is the maximum likelihood estimate (MLE) of the (unstructured) covariance matrix of the data. The likelihood (4) can be maximized with respect to the parameters to obtain the MLEs, $\widehat{\theta}$, of the parameters of the model. The asymptotic variance–covariance matrix of the estimates is obtained as the inverse of the observed information matrix, or the negative Hessian matrix, as usual (Gould, Pitblado, and Sribney 2006).

The (quasi-)MLEs retain some desirable properties when the normality assumptions are violated (Anderson and Amemiya 1988; Browne 1987; Satorra 1990). The estimators are still asymptotically normal. Moreover if 1) the model structure is correctly specified and 2) the error terms, δ, are independent of one another and of the factors, ξ, then the inverse information matrix gives consistent estimates of the variances of parameter estimates, except for the variance parameters of nonnormal factors or errors. If those *asymptotic robustness* conditions are violated, the variance–covariance matrix is inconsistently estimated by the observed or expected information matrix.

Alternative methods of variance–covariance matrix estimation have been proposed that ensure inference is asymptotically robust to violations of normality. The most popular estimate is known as Satorra–Bentler "robust" standard errors, after Satorra and Bentler (1994); see section 5. Stata provides another estimator: Huber sandwich standard errors (Huber 1967).

Other point estimation methods in CFA include generalized least squares (Jöreskog and Goldberger 1972) and asymptotically distribution free methods (Browne 1984). They are not currently implemented in `confa`.

Once the MLEs, $\widehat{\theta}$, are obtained, one can form the *implied* covariance matrix $\Sigma(\widehat{\theta})$. The goodness of fit of the model is then the discrepancy between this matrix and the sample covariance matrix S. The substantive researchers can only convincingly claim that their models are compatible with the data if the model fit is satisfactory, and the null hypothesis

$$H_0 : V(\mathbf{y}) = \Sigma(\theta)$$

cannot be rejected.

The discrepancy implied by the maximum likelihood method itself is the likelihood-ratio test statistic

$$T = -2\left[\ln L\left\{\mathbf{Y}, \Sigma(\widehat{\theta})\right\} - \ln L(\mathbf{Y}, S)\right] \xrightarrow{d} \chi^2_q \tag{5}$$

which has asymptotic χ^2 distribution with degrees of freedom equal to the number of overidentifying model conditions $q = p^* - t$.

There are other concepts of fit popular in SEM and CFA literature (Bentler 1990a; Marsh, Balla, and Hau 1996). Absolute measures of fit are addressing the absolute values of the residuals, defined as the entries of the difference matrix $S - \Sigma(\widehat{\theta})$. An example of such measure is the root of mean squared residual (RMSR), given in section 5.1 by (11). Parsimony indices correct the absolute fit by the number of degrees of freedom used to attain that level of fit. An example of such measure is the root mean squared error of approximation (RMSEA), given in section 5.1 by (12). Values of 0.05 or less, or confidence intervals covering this range, are usually considered to indicate a good fit. Comparative fit indices relate the attained fit of the model to the independence model when $\Sigma(\cdot) = \operatorname{diag} S$ with p degrees of freedom. They are intended to work as pseudo-R^2 for structural equation models. Comparative fit indices are close to 0 for models that are believed to fit poorly and close to 1 for the models that are believed to fit well. Some of the indices may take a value greater than 1, and that is usually taken as indication of overfitting. Two such indices are reported by the `confa` postestimation suite: the Tucker–Lewis nonnormed fit index (TLI) and Bentler's comparative fit index (CFI). Values greater than 0.9 are usually associated with good fit. See section 5 for methods and formulas.

When the assumptions of multivariate normality and asymptotic robustness are violated, the (quasi-)likelihood-ratio statistic (5) has a nonstandard distribution based on the sum of weighted χ^2_1 variables. Satorra and Bentler (1994) proposed Satterthwaite-type corrections: T_{sc} given by (18) corrects the scale of the distribution, and T_{adj} given by (19) corrects both the scale and the number of degrees of freedom.

An alternative procedure to correct for the nonstandard distribution of the likelihood-ratio test statistic is by using resampling methods to obtain approximation for the distribution in question. Beran and Srivastava (1985) and Bollen and Stine (1992)

demonstrated how the bootstrap should be performed under the null hypothesis of the correct model structure. Specifically, they proposed to rotate the data according to

$$\mathbf{y}^* = \Sigma^{1/2}(\widehat{\theta})S^{-1/2}\mathbf{y}$$

The new variables $\mathbf{y}^*$ are guaranteed to be compatible with (2) and at the same time retain the multivariate kurtosis properties of the original data. Then a sample of the rotated data, $\mathbf{y}_b^*$, can be taken; the model is fit to that sample; and the test statistic, T_b, is computed; the whole process is repeated for $b = 1, \ldots, B$ sufficiently many times. The bootstrap p-value associated with test statistic T is the fraction of exceedances:

$$p_{BS} = \frac{1}{B}\#(b : T_b > T)$$

Other aspects of fit that practitioners will usually check is that the parameter estimates have expected signs and the proportions of explained variance of the observed variables (squared multiple correlations, also known as indicator *reliability*) are sufficiently high (say, greater than 50%).

1.3 Factor scoring

In many psychological, psychometric, and educational applications, the applied researcher uses the model like (1)–(2) to obtain estimates of the latent traits for individual observations. They are usually referred to as *factor scores*, $\widehat{\xi}$. The model then serves as an intermediate step in obtaining those scores, although goodness of fit is still an important consideration. The procedure of obtaining the predicted values for ξ is usually referred to as *scoring*.

Two common factor scoring methods are implemented through the `predict` postestimation command of the `confa` command. The regression method obtains the estimates (predictions) of the factor scores by minimizing the (generalized) sum of squared deviations of the factors from their true values, which results in factor scores

$$\widehat{\xi}_{ri} = \widehat{\Phi}\widehat{\Lambda}'\Sigma^{-1}(\widehat{\theta})(\mathbf{y}_i - \widehat{\mu}) \tag{6}$$

The hatted matrices are the matrices of the MLEs of the model parameters. Equation (6) can also be justified as an empirical Bayes estimator of $\widehat{\xi}_i$, with the model giving the prior distribution $\xi \sim N(0, \widehat{\Phi})$, and the data from the ith observation used to update that prior, assuming multivariate normality.

Another scoring method, known as the Bartlett method, imposes an additional assumption of unbiasedness and results in factor scores

$$\widehat{\xi}_{Bi} = \left(\widehat{\Lambda}'\widehat{\Theta}\widehat{\Lambda}\right)^{-1}\widehat{\Lambda}'\widehat{\Theta}^{-1}(\mathbf{y}_i - \widehat{\mu}) \tag{7}$$

It is also known as the maximum likelihood method because it provides the maximum likelihood estimates of ξ conditional on the data $\mathbf{y}_i$, with a mild abuse of notation

because the data are used twice, in estimating the parameters and as inputs to the predictions.

The two methods typically give very similar answers with highly correlated results. The factor scores obtained from the Bartlett method are unbiased but have greater variance, while the factor scores obtained from the regression method are shrunk toward zero.

2 Description of confa command

The `confa` command contains estimation and postestimation commands for confirmatory factor analysis. Single-level, single-group estimation is supported.[1] A variety of identification conditions can be imposed, and robust standard errors can be reported. Goodness-of-fit tests can be corrected using the Satorra and Bentler (1994) scaling approach or using the Bollen and Stine (1992) bootstrap. Complex survey designs specified through [SVY] **svyset** are supported.

2.1 Syntax

```
confa factorspec [factorspec ...] [if] [in] [weight] [,
    correlated(corrspec [corrspec ...]) unitvar(factorlist|_all) free
    constraint(numlist) missing usenames vce(vcetype) level(#) svy
    from(ones|2sls|ivreg|smart|ml_init_args) loglevel(#) ml_options]
```

The factor specification, *factorspec*, is

```
(factorname: varlist)
```

The correlated-errors specification, *corrspec*, is

```
[(]varname_k:varname_j[)]
```

The list of factors, *factorlist*, comprises *factorname*s.

The allowed types of weights are `pweight`s, `iweight`s, and `aweight`s.

```
estat fitindices [, aic bic cfi rmsea rmsr tli _all]

estat aic

estat bic
```

1. Estimation of more advanced models in which the latent variables can be regressed on one another, or in which multiple levels of latent or observed variables may be present, or in which mixed responses (continuous, binary, ordinal, and count) may be present is available with the `gllamm` command (Rabe-Hesketh, Skrondal, and Pickles 2002, 2004).

`estat corrrelate [ , level(#) bound ]`

`predict [ type ] newvarlist [ if ] [ in ] [ , regression empiricalbayes ebayes mle bartlett ]`

`bollenstine [ , reps(#) saving(filename) confaoptions(string) bootstrap_options ]`

2.2 Options of confa

Model

`correlated(`*corrspec* `[` *corrspec* ... `])` specifies the correlated measurement errors δ_k and δ_j corresponding to the variables y_k and y_j. Here *corrspec* is of the form

`[ (`]*varname_k*`:`*varname_j*`[ ) ]`

where *varname_k* and *varname_j* are some of the observed variables in the model; that is, they must appear in at least one *factorspec* statement. If there is only one correlation specified, the optional parentheses shown above may be omitted. There should be no space between the colon and *varname_j*.

`unitvar(`*factorlist*`| _all)` specifies the factors (from those named in *factorspec*) that will be identified by setting their variances to 1. The keyword `_all` can be used to specify that all the factors have their variances set to 1 (and hence the matrix Φ can be interpreted as a correlation matrix).

`free` frees up all the parameters in the model (making it underidentified). It is then the user's responsibility to provide identification constraints and adjust the degrees of freedom of the tests. This option is seldom used.

`constraint(`*numlist*`)` can be used to supply additional constraints. There are no checks implemented for redundant or conflicting constraints, so in some rare cases, the degrees of freedom may be incorrect. It might be wise to run the model with the `free` and `iterate(0)` options and then look at the names in the output of `matrix list e(b)` to find out the specific names of the parameters.

`missing` requests full-information maximum-likelihood estimation with missing data. By default, estimation proceeds by listwise deletion.

`usenames` requests that the parameters be labeled with the names of the variables and factors rather than with numeric values (indices of the corresponding matrices). It is a technical detail that does not affect the estimation procedure in any way, but it is helpful when working with several models simultaneously, tabulating the estimation results, and transferring the starting values between models.

Variance estimation

`vce(`*vcetype*`)` specifies different estimators of the variance–covariance matrix. Common estimators (`vce(oim)`, observed information matrix, the default; `vce(robust)`, sandwich information matrix; `vce(cluster` *clustvar*`)`, clustered sandwich estimator with clustering on *clustvar*) are supported, along with their aliases (the `robust` and `cluster(`*clustvar*`)` options). See [R] ***vce_option***.

An additional estimator specific to SEM is the Satorra–Bentler estimator (Satorra and Bentler 1994). It is requested by `vce(sbentler)` or `vce(satorrabentler)`. When this option is specified, additional Satorra–Bentler scaled and adjusted goodness-of-fit statistics are computed and presented in the output. See section 5 for details.

Reporting

`level(#)` changes the confidence level for confidence-interval (CI) reporting.

Other

`svy` instructs `confa` to respect the complex survey design, if one is specified.

`from(ones | 2sls | ivreg | smart |` *ml_init_args*`)` provides the choice of starting values for the maximization procedure. The `ml` command's internal default is to set all parameters to zero, which leads to a noninvertible matrix, Σ, and `ml` has to make many changes to those initial values to find anything feasible. Moreover, this initial search procedure sometimes leads to a domain where the likelihood is nonconcave, and optimization might fail there.

`ones` sets all the parameters to values of one except for covariance parameters (off-diagonal values of the Φ and Θ matrices), which are set to 0.5. This might be a reasonable choice for data with variances of observed variables close to 1 and positive covariances (no inverted scales).

`2sls` or `ivreg` requests that the initial parameters for the freely estimated loadings be set to the two-stage least-squares (2SLS) instrumental-variable estimates of Bollen (1996). This requires the model to be identified by scaling indicators (i.e., setting one of the loadings to 1) and to have at least three indicators for each latent variable. The instruments used are all other indicators of the same factor. No checks for their validity or search for other instruments is performed.

`smart` provides an alternative set of starting values that is often reasonable (e.g., assuming that the reliability of observed variables is 0.5).

Other specification of starting values, *ml_init_args*, should follow the format of `ml init`. Those typically include the list of starting values of the form `from(# # ... #, copy)` or a matrix of starting values `from(`*matname*`, [copy | skip])`. See [R] **ml**.

`loglevel(#)` specifies the details of output about different stages of model setup and estimation, and is likely of interest only to programmers. Higher numbers imply more output.

Additional *ml_options* may be used to control the maximization process. See [R] **maximize** and [R] **ml**. Of these, the `difficult` option, which improves the behavior of the maximizer in relatively flat regions, is likely to be helpful. See its use in the examples below.

2.3 Descriptions and options of estat

The postestimation command `estat fitindices` produces fit indices and supports the following options:

`aic` requests the Akaike information criterion (AIC).

`bic` requests the Schwarz Bayesian information criterion (BIC).

`cfi` requests the CFI (Bentler 1990b).

`rmsea` requests the RMSEA (Browne and Cudeck 1993).

`rmsr` requests the RMSR.

`tli` requests the TLI (Tucker and Lewis 1973).

`_all` requests all the above indices. This is the default behavior if no option is specified.

The computed fit indices are returned as `r()` values.

`estat aic` and `estat bic` compute the Akaike and Schwarz Bayesian information criteria, respectively.

`estat correlate` transforms the covariance parameters into correlations for factor covariances and measurement-error covariances. The delta method standard errors are given; for correlations close to plus or minus 1, the CIs may extend beyond the range of admissible values. Additional options are allowed.

`level(#)` changes the confidence level for CI reporting.

`bound` provides an alternative asymmetrical CI based on Fisher's z transform (Cox 2008) of the correlation coefficient. It guarantees that the end points of the interval are in the $(-1, 1)$ range, provided the estimate itself is in this range.

2.4 Description and options of predict

The postestimation command `predict` can be used to obtain factor scores. The following options are supported:

`regression`, `empiricalbayes`, or `ebayes` requests regression, or empirical Bayes, factor scoring procedure (6).

`mle` or `bartlett` requests Bartlett scoring procedure (7).

2.5 Options of bollenstine

`reps(#)` specifies the number of bootstrap replications. The default is `reps(200)`.

`saving(`*filename*`)` specifies the file where the simulation results (the parameter estimates and the fit statistics) are to be stored. The default is a temporary file that will be deleted as soon as `bollenstine` finishes.

`confaoptions(`*string*`)` allows the transfer of `confa` options to `bollenstine`. If nondefault model (`unitvar` and `correlated`) options were used, one would need to use them with `bollenstine` as well.

If no starting values are specified among `confaoptions`, the achieved estimates `e(b)` will be used as starting values.

In the author's experience, `confa` may fall into nonconvergent regions with some bootstrap samples. It would be then recommended to limit the number of iterations, say with `confaoptions(iter(20) ...)`.

Other *bootstrap_options* (except for the forced `notable`, `noheader`, `nolegend`, and `reject(e(converged) == 0)` options) are allowed and will be transferred to the underlying `bootstrap` command. See [R] **bootstrap**.

3 Example 1: Simple structure CFA with psychometric data

A popular and well-known dataset for confirmatory factor analysis is based on Holzinger and Swineford (1939) data also analyzed by Jöreskog (1969).[2] The dataset contains the measures of performance of 301 children in grades 7 and 8 from two different schools on several psychometric tests. The complete dataset has 26 psychometric variables. The benchmark analyses (Jöreskog 1969; Yuan and Bentler 2007) usually use a smaller subset with 9 or 12 variables, typically linked to three or four factors, respectively. The relevant subset is available as follows:

2. Available at http://www.coe.tamu.edu/~bthompson/datasets.htm.

```
. use hs-cfa
(Holzinger & Swineford (1939))
. describe
Contains data from hs-cfa.dta
  obs:           301                          Holzinger & Swineford (1939)
 vars:            15                          7 Oct 2008 15:14
 size:        24,983 (99.8% of memory free)   (_dta has notes)
```

variable name	storage type	display format	value label	variable label
id	int	%9.0g		Identifier
sex	byte	%8.0g		Gender
ageyr	byte	%9.0g		Age, years
agemo	byte	%9.0g		Age, months
school	byte	%11.0g	school	School
grade	byte	%8.0g		Grade
x1	double	%10.0g		Visual perception test from Spearman vpt, part iii
x2	double	%10.0g		Cubes, simplification of brigham˙s spatial relations test
x3	double	%10.0g		Lozenges from Thorndike--shapes flipped then identify target
x4	double	%10.0g		Paragraph comprehension test
x5	double	%10.0g		Sentence completion test
x6	double	%10.0g		Word meaning test
x7	double	%10.0g		Speeded addition test
x8	double	%10.0g		Speeded counting of dots in shape
x9	double	%10.0g		Speeded discrim straight and curved caps

```
Sorted by:
```

▷ Specification and starting values

We shall factor analyze these data, grouping the variables together in three factors: "visual" factor (`x1`–`x3` variables), "textual" factor (`x4`–`x6` variables), and "math" factor (`x7`–`x9` variables). In matrix terms,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \end{pmatrix} + \begin{pmatrix} \lambda_{11} & 0 & 0 \\ \lambda_{21} & 0 & 0 \\ \lambda_{31} & 0 & 0 \\ 0 & \lambda_{42} & 0 \\ 0 & \lambda_{52} & 0 \\ 0 & \lambda_{62} & 0 \\ 0 & 0 & \lambda_{73} \\ 0 & 0 & \lambda_{83} \\ 0 & 0 & \lambda_{93} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} + \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \\ \delta_7 \\ \delta_8 \\ \delta_9 \end{pmatrix}$$

$$V(\xi) = \Phi, \quad V(\delta) = \mathrm{diag}(\theta_1, \ldots, \theta_9), \quad \mathrm{Cov}(\xi, \delta) = 0$$

A graphical representation using the standard conventions of structural equation modeling path diagrams is given in figure 1. Observed variables are represented as boxes and unobserved variables, as ovals. The directed arrows between objects correspond to the regression links in the model, and stand-alone arrows toward the observed variables are measurement errors (the symbols δ_j are omitted). Two-sided arrows correspond to correlated constructs (factors).

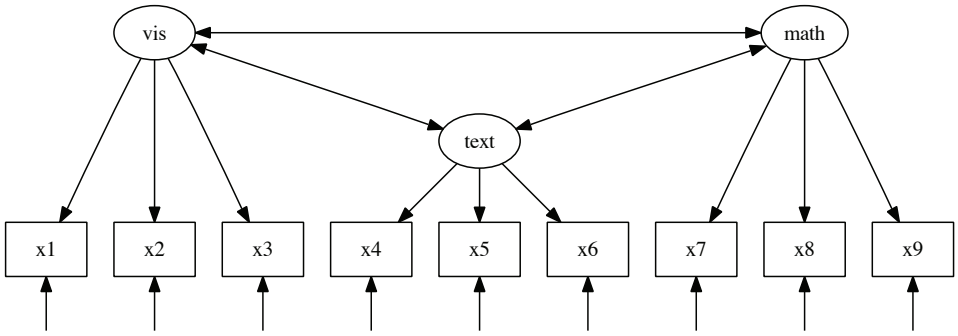

Figure 1. The basic model for Holzinger–Swineford data

As described above, this is a moderate size factor analysis model. A simple initial specification describing the above model is

```
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9)
initial:       log likelihood =  -168453.1
rescale:       log likelihood =  -168453.1
rescale eq:    log likelihood = -4169.0999
could not calculate numerical derivatives
flat or discontinuous region encountered
convergence not achieved
r(430);
```

The default search procedures of `ml` led to a region with flat likelihood, and `ml maximize` was unable to overcome this. As described in the previous section, several options for better starting values are available in `confa`. For the standardized data, the `from(ones)` option will be expected to perform well. If the factors are identified by unit loadings of the first variable (the default), one can use `from(iv)` or its equivalent, `from(2sls)`, to get the initial values of loadings from the Bollen (1996) 2SLS estimation procedure, with factor variances and covariances obtained from the variances of the scaling variables, and error variances obtained by assuming the indicator reliabilities of 0.5. Also, with this normalization by the indicator, the `from(smart)` option provides another set of initial values with initial loadings estimated from the covariances of the variable in question and the scaling variable, with other parameters receiving initial values similarly to the procedure with the `from(iv)` settings. Let us demonstrate those procedures:

```
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(ones)
initial:       log likelihood = -3933.9488
rescale:       log likelihood = -3933.9488
rescale eq:    log likelihood = -3763.1831
Iteration 0:   log likelihood = -3820.0525  (not concave)
Iteration 1:   log likelihood = -3786.3638
Iteration 2:   log likelihood = -3778.5165  (not concave)
Iteration 3:   log likelihood = -3748.4099
Iteration 4:   log likelihood = -3744.5167  (backed up)
Iteration 5:   log likelihood = -3738.5289
Iteration 6:   log likelihood = -3737.8633
Iteration 7:   log likelihood = -3737.7461
Iteration 8:   log likelihood = -3737.7449
Iteration 9:   log likelihood = -3737.7449
  (output omitted)
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(iv)
initial:       log likelihood = -3842.5598
rescale:       log likelihood = -3842.5598
rescale eq:    log likelihood = -3773.2707
Iteration 0:   log likelihood = -3773.2707  (not concave)
Iteration 1:   log likelihood = -3747.5598
Iteration 2:   log likelihood = -3740.8673
Iteration 3:   log likelihood = -3737.8022
Iteration 4:   log likelihood = -3737.7451
Iteration 5:   log likelihood = -3737.7449
  (output omitted)
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(smart)
initial:       log likelihood = -4417.3064
rescale:       log likelihood = -4417.3064
rescale eq:    log likelihood = -4127.3988
Iteration 0:   log likelihood = -4127.3988  (not concave)
Iteration 1:   log likelihood = -3883.7073  (not concave)
Iteration 2:   log likelihood =  -3804.466
Iteration 3:   log likelihood =  -3768.374
Iteration 4:   log likelihood = -3739.6488
Iteration 5:   log likelihood = -3737.7715
Iteration 6:   log likelihood =  -3737.745
Iteration 7:   log likelihood = -3737.7449
  (output omitted)
```

It appears that the 2SLS initial values performed best, and it should not be surprising. The 2SLS estimates are consistent if 1) the model is correctly specified, 2) there are no variables of factor complexity more than 1, and 3) there are no correlated measurement errors. All other starting-value proposals, on the other hand, have some ad-hoc heuristics that produce reasonable, feasible, but far from optimal values. It is not guaranteed, however, that `from(iv)` will always produce the best starting values that would ensure the fastest convergence, especially in misspecified models.

The resulting estimates are identical for all three convergent runs:

log likelihood = -3737.7449 Number of obs = 301

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Means						
x1	4.93577	.0671778	73.47	0.000	4.804104	5.067436
x2	6.08804	.0677543	89.85	0.000	5.955244	6.220836
x3	2.250415	.0650802	34.58	0.000	2.12286	2.37797
x4	3.060908	.066987	45.69	0.000	2.929616	3.1922
x5	4.340532	.0742579	58.45	0.000	4.194989	4.486074
x6	2.185572	.0630445	34.67	0.000	2.062007	2.309137
x7	4.185902	.0626953	66.77	0.000	4.063022	4.308783
x8	5.527076	.0582691	94.85	0.000	5.412871	5.641282
x9	5.374123	.0580698	92.55	0.000	5.260308	5.487938
Loadings						
vis						
x1	1	.	.	.	.	.
x2	.5535013	.1092479	5.07	0.000	.3393794	.7676232
x3	.7293715	.1172686	6.22	0.000	.4995293	.9592138
text						
x4	1	.	.	.	.	.
x5	1.113077	.0649866	17.13	0.000	.9857055	1.240448
x6	.9261464	.0561948	16.48	0.000	.8160066	1.036286
math						
x7	1	.	.	.	.	.
x8	1.179951	.1502869	7.85	0.000	.8853936	1.474507
x9	1.081529	.1951225	5.54	0.000	.6990957	1.463962
Factor cov.						
vis-vis	.8093138	.1497566	5.40	0.000	.5157962	1.102831
text-text	.9794911	.1122102	8.73	0.000	.7595632	1.199419
vis-text	.4082317	.079676	5.12	0.000	.2520696	.5643939
math-math	.3837481	.0920626	4.17	0.000	.2033086	.5641875
text-math	.1734945	.0493133	3.52	0.000	.0768422	.2701468
vis-math	.2622243	.0553834	4.73	0.000	.1536747	.3707738
Var[error]						
x1	.5490568	.11905	4.61	0.000	.315723	.7823905
x2	1.13384	.1042625	10.87	0.000	.9294893	1.338191
x3	.8443248	.0950751	8.88	0.000	.657981	1.030669
x4	.3711736	.047963	7.74	0.000	.2771678	.4651794
x5	.4462552	.0579336	7.70	0.000	.3327075	.559803
x6	.3562031	.0434407	8.20	0.000	.271061	.4413453
x7	.7993921	.0875596	9.13	0.000	.6277784	.9710058
x8	.4876966	.09166	5.32	0.000	.3080462	.667347
x9	.5661322	.0905796	6.25	0.000	.3885995	.7436649

R2	
x1	0.5938
x2	0.1788
x3	0.3366
x4	0.7228
x5	0.7287
x6	0.6999
x7	0.3233
x8	0.5211
x9	0.4408

Goodness of fit test: LR = 85.306 ; Prob[chi2(24) > LR] = 0.0000
Test vs independence: LR = 833.546 ; Prob[chi2(36) > LR] = 0.0000

The reported estimates are as follows: the estimated means of the data (coincide with the sample means for complete data); loadings, λ_{jk}, grouped by the latent variable, in the order in which those factors and variables were specified in the call to `confa`; factor covariances, ϕ_{kl}; and variances of the error terms, δ_j. All parameters are freely estimated, except for loadings used for identification (they have a coefficient estimate equal to 1 and are missing standard errors). This implies that the covariances are not guaranteed to comply with Cauchy inequality and that the error variances are not guaranteed to be nonnegative. Violations of these natural range restrictions are known as Heywood cases and sometimes indicate improper specification of the model.

The next block in the output gives indicator reliabilities defined as a proportion of the variance of the observed variable explained by the model. They can be thought of as R^2's in imaginary regressions of the observed variables on their respective latent factors.

The final set of the displayed statistics is likelihood ratios. The first line is the test against a saturated model (when $\widehat{\Sigma} = S$), and the second line is the test against an independence model (when $\widehat{\Sigma} = \operatorname{diag} S$). The first test shows that the model is not fitting well, which is known in literature, while the second one shows that the current model is still a big improvement when compared with the null model, in which variables are assumed independent.

As a final note on the initial values, the internal logic of `ml search` cannot take into account various parameter boundaries and constraints specific to `confa`. If you see in your output something like

```
. confa (f1: x_1*) (f2: x_2*) (f3: x_3*), from(smart)

initial:       log likelihood = -3332.5231
rescale:       log likelihood = -3290.9289
rescale eq:    log likelihood = -3130.3676
initial values not feasible
```

you have come across such an occurrence. You might want to bypass `ml search` with an additional `search(off)` option.

◁

▷ Standard-error estimation

The results reported above assume multivariate normality and use the inverse observed information matrix as the estimator of the variance–covariance matrix of the coefficient estimates. Other types of estimators are known in SEM, most prominently Satorra and Bentler (1994) variance estimator (16). It can be specified with a nonstandard `vce(sbentler)` option:

```
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(iv)
> vce(sbentler) nolog
log likelihood = -3737.7449                          Number of obs = 301
```

	Satorra-Bentler Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
(output omitted)						
Factor cov.						
vis-vis	.8093134	.1618238	5.00	0.000	.4921447	1.126482
text-text	.9794883	.1187477	8.25	0.000	.746747	1.21223
vis-text	.4082305	.0803487	5.08	0.000	.25075	.565711
math-math	.38375	.0804103	4.77	0.000	.2261487	.5413514
text-math	.1734937	.0551705	3.14	0.002	.0653614	.2816259
vis-math	.2622236	.0543578	4.82	0.000	.1556844	.3687629
Var[error]						
x1	.5490553	.1403178	3.91	0.000	.2740376	.8240731
x2	1.133841	.1007102	11.26	0.000	.9364526	1.331229
x3	.8443246	.0813374	10.38	0.000	.6849062	1.003743
x4	.3711732	.047562	7.80	0.000	.2779533	.4643931
x5	.4462556	.0526208	8.48	0.000	.3431208	.5493905
x6	.3562028	.0447916	7.95	0.000	.2684129	.4439927
x7	.7993899	.0713344	11.21	0.000	.6595771	.9392028
x8	.4876955	.0701502	6.95	0.000	.3502036	.6251874
x9	.5661339	.0629795	8.99	0.000	.4426963	.6895715
(output omitted)						

```
Goodness of fit test: LR = 85.306      ; Prob[chi2(24) > LR] = 0.0000
Test vs independence: LR = 833.546     ; Prob[chi2(36) > LR] = 0.0000

Satorra-Bentler Tsc      = 82.181      ; Prob[chi2(24)   > Tsc ] = 0.0000
Satorra-Bentler Tadj     = 72.915      ; Prob[chi2(21.3) > Tadj] = 0.0000
Yuan-Bentler T2          = 66.468      ; Prob[chi2(24)   > T2  ] = 0.0000
```

The point estimates are the same as before, but the standard errors are different. In models with correctly specified structure, the Satorra–Bentler standard errors are typically larger than the information matrix–based standard errors, although counterexamples can be provided when the distribution of the data has tails lighter than those of the normal distribution. Note also that additional test statistics are reported: T_{sc}, T_{adj}, and T_2. The naïve quasi–maximum-likelihood test statistic reported on the first line of test statistics is no longer valid when the data do not satisfy the asymptotic robustness conditions (see p. 332). These additional tests tend to perform much better. The technical description is given in section 5; see (16) for Satorra–Bentler standard errors and (18)–(20) for the additional test statistics.

As with most of Stata's `ml`-based commands, sandwich standard errors can be obtained with the `robust` option:

(*Continued on next page*)

```
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(iv) robust nolog
log pseudolikelihood = -3737.7449                         Number of obs = 301
```

	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
(*output omitted*)						
Loadings						
vis						
x1	1	.	.	.	.	.
x2	.5535009	.1322981	4.18	0.000	.2942013	.8128005
x3	.7293711	.1413231	5.16	0.000	.452383	1.006359
text						
x4	1	.	.	.	.	.
x5	1.113077	.065795	16.92	0.000	.9841209	1.242033
x6	.9261465	.0614803	15.06	0.000	.8056474	1.046646
math						
x7	1	.	.	.	.	.
x8	1.179948	.1306601	9.03	0.000	.9238593	1.436037
x9	1.081524	.2668148	4.05	0.000	.5585761	1.604471
Factor cov.						
vis-vis	.8093134	.1806965	4.48	0.000	.4551548	1.163472
text-text	.9794883	.121498	8.06	0.000	.7413566	1.21762
vis-text	.4082305	.0994813	4.10	0.000	.2132508	.6032102
math-math	.38375	.1068804	3.59	0.000	.1742683	.5932317
text-math	.1734937	.0563996	3.08	0.002	.0629525	.2840349
vis-math	.2622236	.0601591	4.36	0.000	.1443139	.3801334
Var[error]						
x1	.5490553	.1567305	3.50	0.000	.2418692	.8562415
x2	1.133841	.1120656	10.12	0.000	.9141966	1.353485
x3	.8443246	.1004535	8.41	0.000	.6474394	1.04121
x4	.3711732	.0503657	7.37	0.000	.2724582	.4698882
x5	.4462556	.0567984	7.86	0.000	.3349329	.5575784
x6	.3562028	.0465941	7.64	0.000	.2648801	.4475256
x7	.7993899	.0973832	8.21	0.000	.6085223	.9902576
x8	.4876955	.1197326	4.07	0.000	.2530239	.7223671
x9	.5661339	.1189374	4.76	0.000	.333021	.7992468

```
Goodness of fit test: LR =      .      ; Prob[chi2( .) > LR] =      .
Test vs independence: LR =      .      ; Prob[chi2( .) > LR] =      .
```

Because the `robust` option implies that the assumptions of the model are violated, the likelihood-ratio tests are not computed and indicator reliabilities (squared multiple correlations) are not reported. Similar behavior is shown by other Stata commands, such as `regress, ... robust`, which omits ANOVA table, because this estimator potentially corrects for heteroskedasticity of error terms, and in presence of heteroskedasticity, sums of squared errors are not particularly meaningful. Unlike the Satorra–Bentler variance estimator, the sandwich estimator does not make any assumptions regarding the model structure, and hence is likely to retain consistency under a greater variety of situations compared with the Satorra–Bentler estimator.

◁

▷ Correlated errors

It was argued in substantive literature that one of the reasons the basic CFA model does not fit well for this dataset is because the variables responsible for the speeded

counting (`x7` and `x8`) are measuring similar skills, while the other variable in this factor, `x9`, has a weaker correlation with either of them than they have with one another. Hence, the model where errors of `x7` and `x8` are allowed to correlate might fit better. Here is how this can be implemented.

```
. matrix bb=e(b)
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(bb, skip)
> correlated(x7:x8)
initial:       log likelihood = -3737.7449
rescale:       log likelihood = -3737.7449
rescale eq:    log likelihood = -3737.7449
Iteration 0:   log likelihood = -3737.7449  (not concave)
Iteration 1:   log likelihood = -3732.2812
Iteration 2:   log likelihood = -3730.0893
Iteration 3:   log likelihood = -3723.0064  (not concave)
Iteration 4:   log likelihood = -3722.2265
Iteration 5:   log likelihood = -3721.8698
Iteration 6:   log likelihood = -3721.7297
Iteration 7:   log likelihood = -3721.7283
Iteration 8:   log likelihood = -3721.7283
log likelihood = -3721.7283                     Number of obs = 301
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
(*output omitted*)						
Var[error]						
x1	.5758433	.1034751	5.57	0.000	.3730357	.7786508
x2	1.122499	.1019974	11.01	0.000	.9225877	1.32241
x3	.8321163	.089874	9.26	0.000	.6559664	1.008266
x4	.3722489	.0479869	7.76	0.000	.2781963	.4663014
x5	.4436604	.0580119	7.65	0.000	.3299592	.5573615
x6	.3570578	.0434528	8.22	0.000	.2718919	.4422236
x7	1.036463	.088125	11.76	0.000	.863741	1.209185
x8	.7948157	.0831437	9.56	0.000	.6318571	.9577743
x9	.0875355	.1967033	0.45	0.656	-.2979959	.473067
Cov[error]						
x7-x8	.3527068	.0662993	5.32	0.000	.2227626	.482651
R2						
x1	0.5742					
x2	0.1870					
x3	0.3461					
x4	0.7220					
x5	0.7303					
x6	0.6992					
x7	0.1236					
x8	0.2215					
x9	0.9107					

```
Goodness of fit test: LR = 53.272      ; Prob[chi2(23) > LR] = 0.0003
Test vs independence: LR = 865.579     ; Prob[chi2(36) > LR] = 0.0000
```

Note the use of starting values: the previous parameter estimates are saved and transferred via the `from(..., skip)` option. The `skip` option in parentheses ensures that the values are copied by the names rather than by position in the initial vector. The reported R^2's for variables `x7` and `x8` went down, while the reported R^2 for `x9` went

up and became the largest R^2 in the model. This is not surprising. The `math` factor is primarily based on covariances between the last three variables, and to a lesser extent, on covariances between the last three and the first six variables. The latter component is relatively unchanged between the two models. However, with the covariance between the error terms δ_7 and δ_8 freely estimated, the covariance between `x7` and `x8` no longer contributes to explaining this factor. The burden of identifying this factor shifts to covariances `x7`–`x9` and `x8`–`x9`. The `math` factor now has to contribute less to explaining covariances between `x7` and `x8`, and more to explaining covariance of `x9` with other variables. This produces the observed change in reliabilities.

Is this newly introduced correlation significant? The z statistic is reported to be 5.32, and the likelihood ratio can be formed to be $85.306 - 53.272 = 32.034$, significant when referred to χ^2_1. Virtually identical results can be obtained with the `robust` variance estimator that gives the standard error of 0.0654 and z statistic of 5.39, highly significant at conventional levels.

◁

Let us demonstrate another important procedure for computing significance of the χ^2-difference tests with nonnormal data.

▷ Satorra–Bentler scaled difference test

Nonnormality of the data may cast doubt on the value of both the goodness-of-fit test and the likelihood-ratio tests of nested models. Satorra and Bentler (2001) demonstrated how to obtain a scaled version of the nested models test correcting for multivariate kurtosis. Suppose two models are fit to the data, resulting in the (quasi-)likelihood-ratio test statistics T_0 and T_1; degrees of freedom r_0 and r_1; and scaling factors c_0 and c_1 (18), where index 0 stands for a more restrictive (null) model. Then the test statistic is

$$\overline{T}_d = \frac{(T_0 - T_1)(r_0 - r_1)}{r_0 c_0 - r_1 c_1}$$

to be referred to χ^2 with $r_1 - r_0$ degrees of freedom. It is not guaranteed to be nonnegative in finite samples or with grossly misspecified models.

Here is the sequence of steps to obtain the test statistic $\overline{T}_d$ to test for significance of correlated errors:

```
. qui confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(bb)
> vce(sbentler)
. local T0 = e(lr_u)
. local r0 = e(df_u)
. local c0 = e(SBc)
. qui confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(bb, skip)
> vce(sbentler) correlated(x7:x8)
. local T1 = e(lr_u)
. local r1 = e(df_u)
```

```
. local c1 = e(SBc)
. local DeltaT = (`T0'-`T1')*(`r0'-`r1')/(`r0'*`c0'-`r1'*`c1')
. di as text "Scaled difference Delta = " as res %6.3f `DeltaT' as text "; Prob
> [chi2>" as res %6.3f `DeltaT' as text "] = " as res %6.4f
> chi2tail(`r0'-`r1', `DeltaT')
Scaled difference Delta = 33.484; Prob[chi2>33.484] = 0.0000
```

See the description of returned values in section 5. The test statistic, which has an approximate χ^2 distribution, again confirms that the correlation is significant.

◁

▷ Bollen–Stine bootstrap

Aside from the Satorra–Bentler fit statistics T_{sc} and T_{adj} reported with option `vce(sbentler)`, an alternative way to correct fit statistics for nonnormality is by resampling methods. The bootstrap procedure for covariance matrices was proposed by Beran and Srivastava (1985) and Bollen and Stine (1992). This procedure is implemented via the `bollenstine` command as a part of the `confa` package. See syntax diagrams in section 2.

For a fraction of the bootstrap samples, maximization does not converge (even though the last parameter estimates are used as starting values, by default). Hence, `bollenstine` rejects such samples (via the `reject(e(converged)==0)` option supplied to the underlying `bootstrap`). It is supposed to be used in conjunction with a limit on the number of iterations given by `confaoptions(iter(#) ...)`. In most "good" samples, the convergence is usually achieved in about 5 to 10 iterations. In the output that follows, the limit on the number of iterations is set to 20. There were two samples where the bootstrap did not converge, shown with `x` among the dots produced by the `bootstrap` command. If the number of iterations is set to 5, only 208 out of 500 bootstrap samples produce convergent results.

Note the use of `confaoptions(corr(x7:x8))` to transfer the original model specification to `bollenstine`. Without it, `bollenstine` would be calling the basic model without the correlated errors, thus producing inappropriate results.

(*Continued on next page*)

```
. qui confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(bb, skip)
> correlated(x7:x8)
. set seed 1010101
. bollenstine, reps(500) confaoptions(iter(20) corr(x7:x8))
(running confa on estimation sample)
Bootstrap replications (500)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................   50
..................................................   100
.....................x........................   150
..................................................   200
..................................................   250
..................................................   300
.......................x..........................   350
..................................................   400
..................................................   450
..................................................   500
log likelihood = -3721.7283                     Number of obs = 301
```

	Bollen-Stine Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
(*output omitted*)						
Factor cov.						
vis-vis	.782528	.1362375	5.74	0.000	.5155074	1.049549
text-text	.9784168	.1121731	8.72	0.000	.7585615	1.198272
vis-text	.3995371	.0777012	5.14	0.000	.2472456	.5518285
math-math	.1466786	.0528322	2.78	0.005	.0431294	.2502278
text-math	.1021679	.0360058	2.84	0.005	.0315979	.172738
vis-math	.184376	.0512257	3.60	0.000	.0839754	.2847766
Var[error]						
x1	.5758433	.1034751	5.57	0.000	.3730357	.7786508
x2	1.122499	.1019974	11.01	0.000	.9225877	1.32241
x3	.8321163	.089874	9.26	0.000	.6559664	1.008266
x4	.3722489	.0479869	7.76	0.000	.2781963	.4663014
x5	.4436604	.0580119	7.65	0.000	.3299592	.5573615
x6	.3570578	.0434528	8.22	0.000	.2718919	.4422236
x7	1.036463	.088125	11.76	0.000	.863741	1.209185
x8	.7948157	.0831437	9.56	0.000	.6318571	.9577743
x9	.0875355	.1967033	0.45	0.656	-.2979959	.473067
Cov[error]						
x7-x8	.3527068	.0662993	5.32	0.000	.2227626	.482651
(*output omitted*)						

```
Goodness of fit test: LR = 53.272      ; Prob[chi2(23) > LR] = 0.0003
Test vs independence: LR = 865.579     ; Prob[chi2(36) > LR] = 0.0000
Bollen-Stine simulated Prob[ LR > 53.2722 ] = 0.0020
Based on 498 replications. The bootstrap 90% interval: (13.258,39.852)
```

Standard errors have been replaced by the Bollen–Stine bootstrap ones. In addition to the usual goodness-of-fit tests, the bootstrap p-value and the percentile method CI for the goodness-of-fit test statistic are reported. The computations of the bootstrap p-value, the CI, and the standard errors are based on the converged samples only (498 out of 500). Note how this CI compares with the one implied by the theoretical χ^2_{23} distribution, $[13.091, 35.172]$. The test statistic for the current sample size and multivariate

kurtosis structure appears to be slightly biased upward. The actual test statistic of 53.27 is way outside either interval, and only one out of 498 bootstrap samples produced the test statistics above it.

◁

▷ Postestimation commands: Fit indices and correlations

There are several postestimation commands available in the `confa` command that provide additional estimation and diagnostic results. First, several popular fit indices can be obtained via the `estat fitindices` command:

```
. estat fitindices
  Fit indices
RMSEA  = 0.0662, 90% CI= (0.0430, 0.0897)
RMSR   = 0.0624
TLI    = 0.9429
CFI    = 0.9635
AIC    = 7487.457
BIC    = 7569.013
```

The fit of the model is not that great. RMSEA seems to be barely touching the desirable region (below 0.05), and CFI is rather low although within the range of what are considered good-fitting models (from 0.9 to 1.0).

Second, the covariance parameters can be transformed to correlations by `estat correlate`. The standard errors are computed by the delta method, and the CIs can be computed directly by asymptotic normality, or via Fisher's z transform (Cox 2008) requested by the `bound` option, which produces CIs bound to be within a $(-1, 1)$ interval and shrunk toward zero. If there are any Heywood cases, that is, improper estimates with implied correlations outside a $(-1, 1)$ interval, then z transform is not applicable, and a missing CI will result.

```
. estat corr
Correlation equivalents of covariances
```

	Bollen-Stine Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Factors						
vis-text	.4566098	.0642273	7.11	0.000	.3307266	.5824929
vis-math	.5442157	.0784663	6.94	0.000	.3904246	.6980069
text-math	.2696928	.0684068	3.94	0.000	.1356179	.4037677
Errors						
x7-x8	.3886009	.053664	7.24	0.000	.2834213	.4937804

(*Continued on next page*)

```
. estat corr, bound
Correlation equivalents of covariances
```

	Bollen-Stine Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Factors						
vis-text	.4566098	.0642273	7.11	0.000	.32209	.5730564
vis-math	.5442157	.0784663	6.94	0.000	.3727556	.6797409
text-math	.2696928	.0684068	3.94	0.000	.1311805	.3978771
Errors						
x7-x8	.3886009	.053664	7.24	0.000	.2786917	.4884624

◁

▷ Factor predictions

Factor predictions are obtained by the standard postestimation command `predict`. The feature of this command is that all factors present in the model must be predicted at once, so the *newvarlist* must contain as many new variables as there were factors in the model:

```
. predict fa1-fa3, reg
. predict fb1-fb3, bart
. corr fa1-fb3, cov
(obs=301)
```

	fa1	fa2	fa3	fb1	fb2	fb3
fa1	.573319					
fa2	.386133	.871388				
fa3	.17935	.101985	.135088			
fb1	.785136	.400869	.18499	1.15513		
fb2	.400869	.981677	.102508	.400869	1.10884	
fb3	.184689	.102902	.147167	.18436	.102991	.160725

```
. corr fa1-fb3
(obs=301)
```

	fa1	fa2	fa3	fb1	fb2	fb3
fa1	1.0000					
fa2	0.5463	1.0000				
fa3	0.6445	0.2973	1.0000			
fb1	0.9648	0.3996	0.4683	1.0000		
fb2	0.5028	0.9987	0.2649	0.3542	1.0000	
fb3	0.6084	0.2750	0.9988	0.4279	0.2440	1.0000

The factor covariances within each method resemble the estimated Φ matrix, although the regression (empirical Bayes) method factors are shrunk toward zero (and thus have smaller variances). The factor predictions obtained by the two methods are almost perfectly correlated, which is to be expected because they are measuring the same quantities, albeit on different scales.

◁

Alternative identification

As the last twist that can be applied to these data, let us consider an alternative identification when factor variances are set to 1 and factor loadings are estimated freely.[3]

```
. confa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(ones)
> unitvar(_all) corr(x7:x8)
initial:       log likelihood = -3933.9488
rescale:       log likelihood = -3933.9488
rescale eq:    log likelihood = -3763.1831
Iteration 0:   log likelihood = -3774.4345  (not concave)
  (output omitted)
Iteration 9:   log likelihood = -3721.7283
log likelihood = -3721.7283                     Number of obs = 301
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Means						
x1	4.93577	.0671778	73.47	0.000	4.804104	5.067436
x2	6.08804	.0677543	89.85	0.000	5.955244	6.220836
x3	2.250415	.0650802	34.58	0.000	2.12286	2.37797
x4	3.060908	.066987	45.69	0.000	2.929616	3.1922
x5	4.340532	.0742579	58.45	0.000	4.194989	4.486074
x6	2.185572	.0630445	34.67	0.000	2.062007	2.309137
x7	4.185902	.0626953	66.77	0.000	4.063022	4.308783
x8	5.527076	.0582691	94.85	0.000	5.412871	5.641282
x9	5.374123	.0580698	92.55	0.000	5.260309	5.487938
Loadings						
vis						
x1	.8846049	.0770051	11.49	0.000	.7336778	1.035532
x2	.5092014	.0782212	6.51	0.000	.3558907	.6625121
x3	.6653939	.0739123	9.00	0.000	.5205284	.8102594
text						
x4	.9891496	.0567019	17.44	0.000	.8780159	1.100283
x5	1.102781	.0625864	17.62	0.000	.980114	1.225448
x6	.9161337	.0537635	17.04	0.000	.8107592	1.021508
math						
x7	.3829829	.0689764	5.55	0.000	.2477917	.5181741
x8	.4766196	.0775035	6.15	0.000	.3247156	.6285236
x9	.9630566	.1106833	8.70	0.000	.7461214	1.179992
Factor cov.						
vis-vis	1	.	.	.	.	.
text-text	1	.	.	.	.	.
vis-text	.4566094	.0642274	7.11	0.000	.330726	.5824928
math-math	1	.	.	.	.	.
text-math	.269691	.068409	3.94	0.000	.1356118	.4037702
vis-math	.5442133	.0784713	6.94	0.000	.3904124	.6980142

3. With an additional restriction `if school==2`, the results are accurate within 0.01 to those reported by Yuan and Bentler (2007). The discrepancies are likely to be due to the small differences in the datasets found in different sources on the Internet.

```
Var[error]
          x1    .5758446   .1034751     5.57   0.000      .373037    .7786521
          x2      1.1225   .1019975    11.01   0.000     .9225884    1.322411
          x3    .8321164   .0898742     9.26   0.000     .6559663    1.008267
          x4    .3722483   .0479868     7.76   0.000     .2781958    .4663007
          x5    .4436603   .0580118     7.65   0.000     .3299593    .5573613
          x6     .357058   .0434527     8.22   0.000     .2718922    .4422239
          x7    1.036464   .0881257    11.76   0.000     .8637409    1.209187
          x8     .794817   .0831478     9.56   0.000     .6318503    .9577837
          x9    .0875252   .1967321     0.44   0.656    -.2980627    .4731131
Cov[error]
       x7-x8    .3527083   .0663016     5.32   0.000     .2227595    .4826571
------------------------------------------------------------------------------

R2
          x1       0.5742
          x2       0.1870
          x3       0.3461
          x4       0.7220
          x5       0.7303
          x6       0.6992
          x7       0.1236
          x8       0.2215
          x9       0.9107
------------------------------------------------------------------------------

Goodness of fit test: LR = 53.272      ; Prob[chi2(23) > LR] = 0.0003
Test vs independence: LR = 865.579     ; Prob[chi2(36) > LR] = 0.0000
```

Because scaling of the model is different, the previous estimates might be of limited value, hence the initial values are specified as `from(ones)`. The `ivreg` option is not applicable to this situation. The log-likelihood and goodness-of-fit tests are the same as before: the models are said to be χ^2 identical. The variances and covariances of the error terms are free of the scaling issue and the same as before. Both point estimates of the factor covariances (which are in fact factor correlations with this identification) and their standard errors are very close to the factor correlations and their standard errors reported by `estat correlate` when the model was identified by unit variable loadings (see the section above titled *Postestimation commands: Fit indices and correlations*).

◁

▷ Missing data

By default, `confa` performs listwise deletion of missing data. Any observation that has missing values among the observed variables (or the weight variable if weighted analysis was requested) is dropped from the analysis. Upon excluding such observations, estimation proceeds as if the data were complete.

A more thorough treatment of missing data (full-information maximum-likelihood method for missing data in structural equation modeling) is provided with the `missing` option. When this option is specified, the following modifications are taken:

1. The sample is restricted to the observations identified by the `if` and `in` statements. If the observed variables have missing values, they are still retained.

2. Goodness-of-fit tests and R^2 for observed variables are not computed because they rely on the estimate of the unstructured covariance matrix, which is not available with this method.

3. Factor predictions are not available.

Maximization proceeds by establishing the patterns of missing data and extracting the relevant submatrices of the mean vector, $\mu(\theta)$, and covariance matrix, $\Sigma(\theta)$, for each pattern. A message is printed about the number of missing patterns found; the computation time should be expected to increase linearly with that number because this many submatrices of $\Sigma(\theta)$ should be inverted for each evaluation of the log likelihood.

The naïve listwise deletion analysis is appropriate when the data are missing completely at random (Little and Rubin 2002). The more sophisticated analysis with the `missing` option is technically applicable to more complicated situations when the probability of being missing depends on other observed variables. It can be argued however that in CFA context, the relevant conditioning should be on the exogenous variables ξ and δ, which are unobserved. Typically, in the missing-data situations, listwise deletion will tend to exclude a lot of observations, so specifying the `missing` option is recommended for most uses. Carrying over the starting values from simpler analysis will speed up convergence, as usual. My experience suggests that the likelihoods with missing data tend to have multiple local maximums and thus are more sensitive to starting values.

Let us introduce some missing data in the Holzinger–Swineford example and analyze the resulting dataset.

```
. set seed 123456
. forvalues k=1/9 {
  2.         gen y~k~ = cond(runiform()<0.0~k~, ., x~k~)
  3.         }
(2 missing values generated)
(2 missing values generated)
(8 missing values generated)
(18 missing values generated)
(21 missing values generated)
(14 missing values generated)
(17 missing values generated)
(28 missing values generated)
(33 missing values generated)
```

By default, `confa` will perform listwise deletion:

(*Continued on next page*)

```
. confa (vis: y1 y2 y3) (text: y4 y5 y6) (math: y7 y8 y9), from(bb) nolog
log likelihood = -2349.8705                              Number of obs = 188
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
(*output omitted*)						
Loadings						
vis						
y1	1	.	.	.	.	.
y2	.5961873	.1403271	4.25	0.000	.3211512	.8712234
y3	.7673835	.1403096	5.47	0.000	.4923818	1.042385
text						
y4	1	.	.	.	.	.
y5	1.170694	.0912381	12.83	0.000	.991871	1.349518
y6	.9482258	.0787462	12.04	0.000	.793886	1.102566
math						
y7	1	.	.	.	.	.
y8	1.108808	.1974696	5.62	0.000	.7217751	1.495842
y9	1.101076	.2707746	4.07	0.000	.5703674	1.631784
Factor cov.						
vis-vis	.8740227	.1947933	4.49	0.000	.4922347	1.255811
text-text	.9052388	.1378389	6.57	0.000	.6350794	1.175398
vis-text	.4241773	.1020139	4.16	0.000	.2242338	.6241209
math-math	.369443	.1210115	3.05	0.002	.1322648	.6066213
text-math	.1909222	.0617196	3.09	0.002	.0699539	.3118904
vis-math	.2244777	.068616	3.27	0.001	.0899928	.3589626
Var[error]						
y1	.5456968	.1511219	3.61	0.000	.2495033	.8418903
y2	1.1373	.1376886	8.26	0.000	.8674351	1.407165
y3	.7342031	.114935	6.39	0.000	.5089346	.9594717
y4	.4184883	.063913	6.55	0.000	.2932212	.5437554
y5	.4209509	.0772258	5.45	0.000	.269591	.5723107
y6	.4113066	.0606663	6.78	0.000	.2924029	.5302104
y7	.8200653	.1178993	6.96	0.000	.5889869	1.051144
y8	.5880029	.1172023	5.02	0.000	.3582907	.8177151
y9	.5367541	.1186252	4.52	0.000	.304253	.7692552
(*output omitted*)						

```
Goodness of fit test: LR = 61.405      ; Prob[chi2(24) > LR] = 0.0000
Test vs independence: LR = 503.076     ; Prob[chi2(36) > LR] = 0.0000
```

A more sophisticated analysis is available with the `missing` option:

```
. confa (vis: y1 y2 y3) (text: y4 y5 y6) (math: y7 y8 y9), from(iv) missing
> difficult
Note: 29 patterns of missing data found
initial:       log likelihood = -3579.9111
rescale:       log likelihood = -3579.9111
rescale eq:    log likelihood = -3525.1169
Iteration 0:   log likelihood = -3525.1169
  (output omitted)
Iteration 5:   log likelihood = -3493.7822
log likelihood = -3493.7822                     Number of obs = 301
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Means						
y1	4.868664	.0643479	75.66	0.000	4.742544	4.994784
y2	5.882944	.0659704	89.18	0.000	5.753645	6.012244
y3	2.168227	.0707049	30.67	0.000	2.029648	2.306806
y4	3.076254	.0608798	50.53	0.000	2.956932	3.195577
y5	4.41519	.0704952	62.63	0.000	4.277022	4.553358
y6	2.170605	.0643098	33.75	0.000	2.044561	2.29665
y7	4.165661	.0661282	62.99	0.000	4.036052	4.295269
y8	5.502241	.063663	86.43	0.000	5.377463	5.627018
y9	5.388172	.0603112	89.34	0.000	5.269964	5.50638
Loadings						
vis						
y1	1	.	.	.	.	.
y2	.7196496	.0968129	7.43	0.000	.5298999	.9093994
y3	.9898674	.1114766	8.88	0.000	.7713774	1.208358
text						
y4	1	.	.	.	.	.
y5	1.249689	.0845489	14.78	0.000	1.083977	1.415402
y6	1.08037	.0781354	13.83	0.000	.9272272	1.233512
math						
y7	1	.	.	.	.	.
y8	1.239025	.1565318	7.92	0.000	.9322288	1.545822
y9	1.0219	.1579594	6.47	0.000	.7123056	1.331495
Factor cov.						
vis-vis	.8300679	.1255225	6.61	0.000	.5840484	1.076087
text-text	.6923659	.0896611	7.72	0.000	.5166333	.8680984
vis-text	.2878234	.0663537	4.34	0.000	.1577725	.4178743
math-math	.4502683	.0988643	4.55	0.000	.2564979	.6440387
text-math	.180085	.0462256	3.90	0.000	.0894844	.2706855
vis-math	.261571	.0546761	4.78	0.000	.1544078	.3687341
Var[error]						
y1	.4115598	.0872224	4.72	0.000	.2406071	.5825125
y2	.8734908	.0871599	10.02	0.000	.7026606	1.044321
y3	.6667882	.0965589	6.91	0.000	.4775363	.8560401
y4	.389792	.046189	8.44	0.000	.2992632	.4803209
y5	.3682757	.060919	6.05	0.000	.2488767	.4876747
y6	.4094993	.0531293	7.71	0.000	.3053679	.5136308
y7	.8087322	.0883255	9.16	0.000	.6356174	.9818471
y8	.4544227	.0961848	4.72	0.000	.2659039	.6429415
y9	.5391701	.0829834	6.50	0.000	.3765257	.7018146

```
Goodness of fit test: LR =      .      ; Prob[chi2( .) > LR] =      .
Test vs independence: LR =      .      ; Prob[chi2( .) > LR] =      .
```

In this analysis, both variance–covariance matrices of the coefficient estimates (`vce` or `e(V)`) for the complete-data analysis (with `x*` variables) and missing-data analysis (with `y*` variables and the `missing` option) are smaller than the variance–covariance matrix in the analysis of `y*` variables without the `missing` option. Comparison between the former two is inconclusive.

A word of caution: It appears that this treatment of missing data leads to highly unstable results. Table 1, below, shows the maximization results with different starting values and different maximization techniques. The top value in each cell is the log likelihood at maximum, and the bottom value is the elapsed maximization time. None of the 20 resulting maximums coincided! This behavior was not observed in the complete-data analysis where the same maximum has been consistently found with all starting values and maximization parameters. It is possible that the global maximum of the procedure was not found, and it is unclear which of the local maximums would correspond to consistent estimates.

Table 1. Multiple maximums in missing-data problems

Starting values	`technique(nr)` `difficult`: off	`technique(nr)` `difficult`: on	`technique(dfp)` `difficult`: off	`technique(dfp)` `difficult`: on
Complete analysis	−3454.222 89.05 s	−3487.593 87.75 s	−3504.6316 60.63 s	−3697.2417 67.61 s
Naïve missing	−3532.2684 98.61 s	−3511.787 110.59 s	−3678.0145 62.69 s	−3548.1309 59.08 s
`iv`	−3508.6958 98.38 s	−3563.8789 154.69 s	−3484.9064 98.37 s	−3570.5609 154.69 s
`smart`	−3533.009 131.09 s	−3550.5144 160.49 s	−3601.0655 90.80 s	−3556.5871 234.11 s
`ones`	−3594.406 127.70 s	−3452.5826 157.88 s	−3645.4862 68.67 s	−3569.1392 66.39 s

◁

4 Example 2: Modeling the structure of correlated measurement errors

An interesting class of the CFA models is that of multiple traits and multiple methods (MTMM). In those models, the observed variables are explained by two unrelated sets of factors: traits, or the factors of primary interest, and methods, or auxiliary factors, often modeling relations between measurement errors δ.

Bollen (1993) analyzes two dimensions of liberal democracy, political liberties and democratic rule, using three sources of data[4] (indicators developed by three liberal

4. The complete dataset, codebooks, and data description are available at http://www.icpsr.umich.edu/cocoon/ICPSR/STUDY/02532.xml.

democracy researchers: A. Banks, R. D. Gastil, and L. R. Sussman; see references in Bollen [1993]). Political liberties are measured by freedom of group opposition and party formation, freedom of the broadcast media, and freedom of print media and civil liberties. Democratic rule is measured by effectiveness of the elected legislative body, political rights, competitiveness of nomination process, and chief executive election. The measurement errors are believed to be correlated, with correlations coming from variables that have been produced by the three aforementioned researchers. In MTMM terms, the two substantive dimensions are the traits, and the data sources are the methods. While the general MTMM models may have identification problems (Marsh, Byrne, and Craven 1992; Byrne and Goffin 1993; Grayson and Marsh 1994) due to highly structured covariance matrices, this model does not load every method to every factor and has been shown by Bollen (1993) to be identified. The structure of the model is represented in figure 2. The individual error terms are omitted to reduce the clutter.

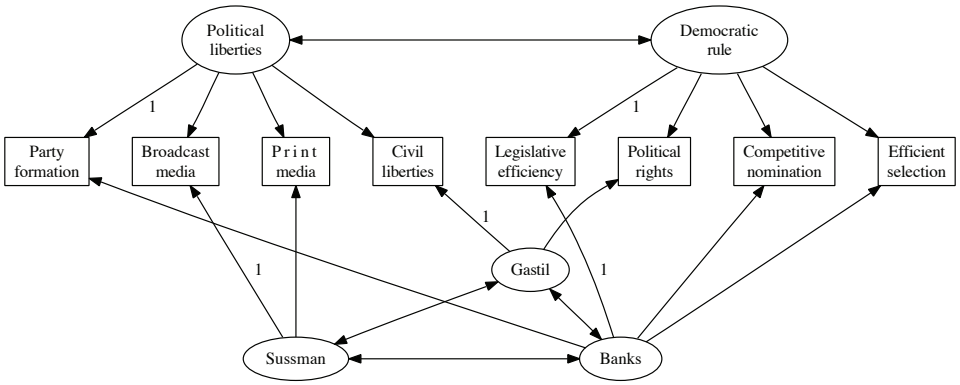

Figure 2. Structure of the MTMM model of Bollen (1993)

▷ Building up a complex CFA model

The default initial values logic with one of `from(iv)`, `from(ones)`, or `from(smart)` does not apply well in this situation, because each variable has a factor complexity of two. The model fails to converge when any of those options is submitted as starting values. Thus we first fit the traits and the methods models separately, using the residuals from the first model as the data for the second model. The estimates are combined to form the starting values for the full model.

(*Continued on next page*)

```
. *traits model
. use libdem80, clear
(Cross-National Indicators of Liberal Democracy, 1950-1990)
. confa (pollib: party broad print civlb) (demrul: leg80 polrt compet effec),
> vce(sbentler) from(smart) difficult usenames
initial:       log likelihood = -3483.2656
rescale:       log likelihood = -3483.2656
rescale eq:    log likelihood =   -3294.09
Iteration 0:   log likelihood =   -3294.09  (not concave)
Iteration 1:   log likelihood = -3232.2538  (not concave)
  (output omitted)
Iteration 14:  log likelihood = -2672.5848
  (output omitted)
. matrix b_t = e(b)
. preserve
. *methods model: obtain the data by replacing the variables with their residuals
. predict f1 f2, bartlett
. foreach x of varlist party80 broad80 print80 civlb80 {
  2.         qui replace `x' = `x' - [lambda_`x'_pollib]_cons*f1
  3. }
. foreach x of varlist leg80 polrt80 compet80 effec80 {
  2.         qui replace `x' = `x' - [lambda_`x'_demrul]_cons*f2
  3. }
. confa (sussman: broad print) (gastil: civlb polrt)
> (banks: leg80 party compet effec), difficult from(smart) usenames iter(20)
initial:       log likelihood = -2072.5146
rescale:       log likelihood = -2072.5146
rescale eq:    log likelihood = -1944.4457
Iteration 0:   log likelihood = -1944.4457  (not concave)
Iteration 1:   log likelihood = -1888.2893  (not concave)
  (output omitted)
Iteration 20:  log likelihood = -1463.6925  (not concave)
convergence not achieved
  (output omitted)
. matrix b_res = e(b)
. restore
```

Next let us fit the full model. First, we define the constraints, specifying that the traits and methods are uncorrelated. Second, we specify the starting values as a combination of the loadings and factor covariances from the two runs. The matrix `b_t` contains the following preliminary estimates: the means of the observed variables, the loadings of the traits (dimensions of political democracy), the covariances of the trait factors, and the residual variances from the first model. The matrix `b_res` contains the following preliminary estimates: the means of the observed variables, the loadings of the methods (sources of data), the covariances of the method factors, and the residual variances from the second model. The matrix `bb2` updates the traits model results with the "new" results from the residual model (the loadings and factor covariances of the methods, and error variances). The range of indices can be identified from output of `matrix list b_t` and `matrix list b_res`. While the parameters are not in the correct order in matrix `bb2`, the combination of `from(..., skip)` and `usenames` ensures that parameters are copied by names rather than by position in the initial values vector.

```
. constraint define 201 [phi_pollib_sussman]_cons = 0
. constraint define 202 [phi_pollib_gastil]_cons  = 0
. constraint define 203 [phi_pollib_banks]_cons   = 0
. constraint define 204 [phi_demrul_sussman]_cons = 0
. constraint define 205 [phi_demrul_gastil]_cons  = 0
. constraint define 206 [phi_demrul_banks]_cons   = 0
. * initial values: combine the previous results
. matrix bb2 = (b_t[1,1..19], b_res[1,9..30] )
. confa (pollib: party broad print civlb) (demrul: leg80 polrt compet effec)
> (sussman: broad print) (gastil: civlb polrt) (banks: leg80 party compet effec),
>  constr(201 202 203 204 205 206) from(bb2) usenames difficult vce(sbentler)
initial:       log likelihood = -2639.5682
rescale:       log likelihood = -2639.5682
rescale eq:    log likelihood = -2592.2313
Iteration 0:   log likelihood = -2595.7894  (not concave)
  (output omitted)
Iteration 10:  log likelihood = -2568.1962
log likelihood = -2568.1962                              Number of obs = 153
```

	Satorra-Bentler					
	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Means						
party80	3.616557	.344394	10.50	0.000	2.941557	4.291557
broad80	3.398693	.3385019	10.04	0.000	2.735241	4.062144
print80	4.575163	.3517822	13.01	0.000	3.885683	5.264644
civlb80	4.422659	.259731	17.03	0.000	3.913596	4.931723
leg80	4.934636	.2885947	17.10	0.000	4.369001	5.500271
polrt80	4.379082	.2918081	15.01	0.000	3.807149	4.951016
compet80	6.24183	.300571	20.77	0.000	5.652722	6.830938
effec80	4.575163	.2921247	15.66	0.000	4.00261	5.147717
Loadings						
pollib						
party80	1	.	.	.	.	.
broad80	.8605268	.0653934	13.16	0.000	.732358	.9886955
print80	.9250379	.0579294	15.97	0.000	.8114983	1.038577
civlb80	.7187934	.043395	16.56	0.000	.6337408	.8038461
demrul						
leg80	1	.	.	.	.	.
polrt80	1.078044	.0659108	16.36	0.000	.9488608	1.207227
compet80	.9393674	.0597369	15.73	0.000	.8222852	1.05645
effec80	.4380042	.0780376	5.61	0.000	.2850532	.5909551
sussman						
broad80	1	.	.	.	.	.
print80	1.191159	.2313778	5.15	0.000	.7376668	1.644651
gastil						
civlb80	1	.	.	.	.	.
polrt80	.6327867	.1780188	3.55	0.000	.2838763	.981697
banks						
party80	-.1835592	.6226701	-0.29	0.768	-1.40397	1.036852
leg80	1	.	.	.	.	.
compet80	2.710965	.7441043	3.64	0.000	1.252547	4.169382
effec80	1.936548	.6181943	3.13	0.002	.7249093	3.148187

```
Factor cov.
pollib-pol~b      16.03   1.382935    11.59   0.000      13.3195     18.7405
demrul-dem~l    10.4853   1.120171     9.36   0.000      8.28981     12.6808
pollib-dem~l    12.85938  1.113003    11.55   0.000     10.67793    15.04082
sussman-su~n    2.568807  1.111159     2.31   0.021     .3909752    4.746638
demrul-sus~n    (omitted)
pollib-sus~n    (omitted)
gastil-gas~l     1.432488  .4740313     3.02   0.003     .5034042    2.361573
sussman-ga~l     1.472053  .6605832     2.23   0.026     .1773339    2.766772
demrul-gas~l    (omitted)
pollib-gas~l    (omitted)
 banks-banks     .6788023  .4804045     1.41   0.158    -.2627733    1.620378
gastil-banks    -.3427659  .2509348    -1.37   0.172    -.8345891    .1490573
sussman-ba~s    -.2801559   .309342    -0.91   0.365    -.8864551    .3261433
demrul-banks    (omitted)
pollib-banks    (omitted)
Var[error]
     party80     2.094032  .8899954     2.35   0.019     .3496733    3.838391
     broad80     3.092162  .4884588     6.33   0.000     2.134801    4.049524
     print80     1.572295  .5140105     3.06   0.002     .5648532    2.579737
     civlb80     .6067974  .1927103     3.15   0.002     .2290921    .9845027
       leg80      1.57879  .2679765     5.89   0.000     1.053566    2.104014
     polrt80       .26886  .3682653     0.73   0.465    -.4529267    .9906467
    compet80    -.4186224  .8945279    -0.47   0.640    -2.171865     1.33462
     effec80     8.499297  1.068135     7.96   0.000     6.405792     10.5928
------------------------------------------------------------------------------
R2
     party80       0.8788
     broad80       0.8181
     print80       0.9108
     civlb80       0.9348
       leg80       0.8705
     polrt80       0.9727
    compet80       1.0239
     effec80       0.3468
------------------------------------------------------------------------------
Goodness of fit test: LR =  9.206      ; Prob[chi2( 8) > LR] = 0.3253
Test vs independence: LR = 1603.033    ; Prob[chi2(28) > LR] = 0.0000

Satorra-Bentler Tsc      =  8.848      ; Prob[chi2( 8)   > Tsc ] = 0.3553
Satorra-Bentler Tadj     =  8.185      ; Prob[chi2( 7.4) > Tadj] = 0.3558
Yuan-Bentler T2          =  8.683      ; Prob[chi2( 8)   > T2  ] = 0.3697
```

The use of the `difficult` option helped to bring down the number of iterations from 43 to 13. Goodness-of-fit measures are identical to those reported in Bollen (1993), so estimation procedures converged to the same maximums as in Bollen (1993).

A mild Heywood case was produced for the `compet80` variable: the reported estimated error variance is negative, and the corresponding R^2 is greater than 1. However, the CI for this parameter covers zero. Thus the interpretation can be offered that the population variance might be a small positive quantity. The error variance of exactly zero is as suspicious as a negative estimate: it means that we have a perfect measure of democratic rule, but we know that it is affected by the measurement error associated with the Banks factor (i.e., this variable came from Banks' dataset). Heywood cases are sometimes indicative of model misspecification. If that is the case, only `vce(robust)` standard errors are asymptotically valid. Here we used `vce(sbentler)` to produce a

range of additional test statistics correcting for multivariate kurtosis expected with this dataset because many variables are ordinal with few categories (3 to 5).

From the substantive perspective, it might be interesting to note that the variance of the Banks factor appears to be insignificant. This means that the variables obtained from Banks and analyzed in the context of the current model are relatively free of the common influences due to idiosyncrasies of that researcher. This cannot be said about the variables coming from the other two researchers, Gastil and Sussman, because they do seem to contain nontrivial amount of common influences. It might be puzzling, however, that the loadings from the Banks factor to its observed `compet80` and `effec80` variables are well identified.

◁

5 Technical notes

5.1 Methods and formulas

`confa` estimates (2) by maximum likelihood. The observed $\mathbf{y}_i$ variables are described by

$$\mathbf{y}_i = \mu + \Lambda\xi_i + \delta_i$$

where

$$\begin{pmatrix} \delta_i \\ \xi_i \end{pmatrix} \sim N\left\{0, \begin{pmatrix} \Phi & 0 \\ 0 & \Theta \end{pmatrix}\right\}$$

Hence,

$$\mathbf{y}_i \sim N(\mu, \Lambda\Phi\Lambda' + \Theta)$$

and the log likelihood for observation i, $\ln L_i = l_i$, is

$$l_i = -\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln\mid\Sigma\mid - \frac{1}{2}(\mathbf{y}_i - \mu)'\Sigma^{-1}(\mathbf{y}_i - \mu) \tag{8}$$

where $\Sigma = \Sigma(\theta) = \Lambda\Phi\Lambda' + \Theta$ is a $p \times p$ matrix, and the parameters θ of the model are the means μ, the free elements of Λ, nonredundant elements of Φ, and the free elements of Θ. The latter are usually the diagonal elements only, but if the `correlated()` option is specified, off-diagonal elements can be estimated, as well. Because the means part of the model is saturated, the number of covariance structure parameters $\dim\theta = t$ must be no greater than the number of the nonredundant moments of the covariance matrix $p^* = p(p+1)/2$.

When some components of $\mathbf{y}_i$ are missing and the `missing` option is specified, the vector of means, μ, and the parametric covariance matrix, Σ, are restricted to the nonmissing components in computation of the likelihood (8).

The conventional standard errors are available as the inverse of the observed information matrix (`vce(oim)` method). Other analytic estimators (`vce(opg)`, `vce(robust)`, and `vce(cluster` *clustvar*`)`) are supported, but resampling estimators need to be specified explicitly via a `bootstrap` or a `jackknife` prefix to the `confa` command; see [R] ***vce_option***, [R] **bootstrap**, and [R] **jackknife**.

The proportions of the observed-variable variance explained by the model, similar to R^2 in regression and variable communality in EFA, are computed and reported. For variable j,

$$R^2_j = \frac{\sigma_{jj}(\theta) - V(\delta_j)}{s^2_j}$$

where s^2_j is the sample variance of y_j.

Two likelihood-ratio tests are computed by default. The first one is a test against a saturated model:

$$H_0 : \Sigma = \Sigma(\theta) \quad \text{versus} \quad H_1 : \Sigma \text{ is unstructured}$$

It has a likelihood-ratio test statistic

$$T_u = -2\left\{ l(\widehat{\theta}) - \left(-\frac{pN}{2}\ln 2\pi - \frac{N}{2}\ln \mid S \mid - \frac{pN}{2}\right)\right\}$$

where subindex u stands for "unstructured". It has an asymptotic χ^2 distribution with the residual degrees of freedom $\mathrm{df}_u = p^* - t$.

The second likelihood-ratio test is the test against an "independence" model:

$$H_0 : \Sigma = \Sigma_0 = \mathrm{diag}(\sigma^2_1, \ldots, \sigma^2_p) \quad \text{versus} \quad H_1 : \Sigma = \Sigma(\theta)$$

It has a likelihood-ratio test statistic

$$T_i = -2\left\{ \left(-\frac{pN}{2}\ln 2\pi - \frac{N}{2}\ln \mid S_0 \mid - \frac{N}{2}\,\mathrm{tr}\, S_0\right) - l(\widehat{\theta})\right\}$$

where $S_0 = \mathrm{diag}(s^2_1, \ldots, s^2_p)$ and subindex i stands for "independent". The test statistic has an asymptotic χ^2 distribution with degrees of freedom $\mathrm{df}_i = t - p$.

The postestimation command `estat fitindices` computes and reports several fit indices that are used to complement the general χ^2 goodness-of-fit test.

CFI (Bentler 1990b) is

$$\mathrm{CFI} = 1 - \frac{\max(T_u - \mathrm{df}_u, 0)}{\max(T_u - \mathrm{df}_u, T_i - \mathrm{df}_i, 0)} \tag{9}$$

TLI (Tucker and Lewis 1973) is

$$\mathrm{TLI} = \left(\frac{T_i}{\mathrm{df}_i} - \frac{T_u}{\mathrm{df}_u}\right) \Bigg/ \left(\frac{T_i}{\mathrm{df}_i} - 1\right) \tag{10}$$

RMSR (Jöreskog and Sörbom 1986) is

$$\mathrm{RMSR} = \left\{ \frac{1}{p^*} \sum_{1 \le i \le j \le p} (s_{ij} - \widehat{\sigma}_{ij})^2 \right\}^{1/2} \tag{11}$$

RMSEA (Steiger 1990; Browne and Cudeck 1993) is

$$\widehat{\epsilon}_a = \sqrt{\max\left\{\frac{T_u}{(N-1)\mathrm{df}_u}, 0\right\}} \tag{12}$$

Let $G(x;\lambda,d)$ be the cumulative distribution function of the noncentral χ^2 with noncentrality parameter λ and d degrees of freedom. If $G(T_u \,|\, 0,d) \geq 0.95$, find $\widehat{\lambda}_L$ as the solution of

$$G(T_u;\widehat{\lambda}_L,\mathrm{df}_u) = 0.95$$

Otherwise, set $\widehat{\lambda}_L = 0$. Likewise, if $G(T_u \,|\, 0,d) \geq 0.05$, find $\widehat{\lambda}_U$ as the solution of

$$G(T_u;\widehat{\lambda}_U,\mathrm{df}_u) = 0.05$$

Otherwise, set $\widehat{\lambda}_U = 0$. Finally, set the 90% CI for RMSEA as

$$\left\{\sqrt{\frac{\widehat{\lambda}_L}{(N-1)\mathrm{df}_u}}, \sqrt{\frac{\widehat{\lambda}_U}{(N-1)\mathrm{df}_u}}\right\}$$

If sandwich standard errors are requested, the data are implicitly assumed not to be independent and identically distributed (or violating the model assumptions otherwise), no test statistics or R^2 is reported, and no fit indices are produced by `estat fitindices`.

An additional variance estimator (Satorra and Bentler 1994) is available with the `vce(sbentler)` nonstandard option. Let $\mathbf{s} = \operatorname{vech} S$, $\sigma = \operatorname{vech}\Sigma$, where vech is vectorization operator suppressing redundant elements (Magnus and Neudecker 1999), and dependence of Σ and σ on θ is implied. Suppose the model has a correct structural specification but an incorrect distributional specification. That is, the number of factors and their relations to observed variables are the true ones, but the distribution of the data is not multivariate normal. Then, under some regularity conditions, the sample moments are asymptotically normal:

$$\sqrt{N}(\mathbf{s}-\sigma) \to N(0,\Gamma)$$

The simplest estimator of Γ is based on the fourth-order moments of data,

$$\widehat{\Gamma}_N = \frac{1}{N-1}\sum_i (\mathbf{b}_i - \overline{\mathbf{b}})(\mathbf{b}_i - \overline{\mathbf{b}})' \tag{13}$$

where $\mathbf{b}_i = (y_i - \overline{y})(y_i - \overline{y})'$. Introduce the *normal theory weight matrix*,

$$V_N = \frac{1}{2}D'(\Sigma\otimes\Sigma)D \tag{14}$$

where D is the duplication matrix (Magnus and Neudecker 1999), and the Jacobian matrix,

$$\widehat{\Delta} = \left.\frac{\partial\sigma}{\partial\theta}\right|_{\theta=\widehat{\theta}} \tag{15}$$

Then the Satorra–Bentler variance estimator is

$$\widehat{\text{acov}}(\widehat{\theta}) = (N-1)^{-1}\big(\widehat{\Delta}'V_N\widehat{\Delta}\big)^{-1}\widehat{\Delta}'V_N\Gamma_N V_N\widehat{\Delta}\big(\widehat{\Delta}'V_N\widehat{\Delta}\big)^{-1} \tag{16}$$

When the observed variables come from a nonnormal distribution, the (quasi-)likelihood-ratio test statistic becomes a mixture of χ^2

$$T_u \xrightarrow{d} \sum_{j=1}^{\text{df}_u} \alpha_j X_j, \quad X_j \sim \text{i.i.d. } \chi^2_1$$

and α_j are eigenvalues of the matrix $U\Gamma$ with

$$U = V - V\Delta(\Delta'V\Delta)^{-1}\Delta'V \tag{17}$$

Satorra and Bentler (1994) proposed to use the scaled statistic

$$T_{\text{sc}} = \frac{T}{\widehat{c}}, \quad \widehat{c} = \frac{1}{\text{df}_u}\,\text{tr}(\widehat{U}\widehat{\Gamma}_N) \tag{18}$$

which has an approximate $\chi^2_{\text{df}_u}$ distribution, where $\widehat{U}$ is U evaluated at θ, and the adjusted statistic

$$T_{\text{adj}} = \frac{\widehat{d}}{\widehat{c}}T, \quad \widehat{d} = \frac{\Big\{\text{tr}(\widehat{U}\widehat{\Gamma}_N)\Big\}^2}{\text{tr}\Big\{(\widehat{U}\widehat{\Omega}_N)^2\Big\}} \tag{19}$$

which has an approximate $\chi^2_{\widehat{d}}$ distribution, where the degrees of freedom $\widehat{d}$ might be a noninteger number.

Another correction to the T statistic proposed by Yuan and Bentler (1997) is

$$T_2 = T/(1 + T/N) \tag{20}$$

which has an approximate χ^2 distribution with df_u degrees of freedom.

5.2 Implementation details

The `confa` package consists of the following ado-files: `confa` (the main estimation engine), `confa_estat` (postestimation commands), `confa_lfm` (likelihood evaluator), `confa_p` (prediction), and `bollenstine` (Bollen–Stine bootstrap). The Mata functions for `confa` are available in the `lconfa.mlib` library. The likelihood maximization is implemented through the `ml lf` mechanism (observation-by-observation likelihoods with numerical derivatives). There are approximately 43 KB of ado-code (about 1,400 lines) and 13 KB of Mata code (about 450 lines).

The ado-code uses the `listutil` package by N. J. Cox. Its presence is checked, and if the package is not found, an attempt is made to install it from the Statistical Software Components archive.

The memory requirements of `confa` are likely to be mild. To compute the sandwich standard errors (with the `robust` or `cluster` options or with `svy` settings), `confa` will generate # parameters scores, which would require at least 4 × (# parameters) × (# observations) bytes of memory. Even for sizeable models with, say, 20 variables (and thus about 50 or so parameters) and 10,000 observations, this is 2 MB.

5.3 Parameter names and saved results

The nomenclature of the parameter names is as follows.

By default, the parameters are labeled with numeric indices. The observed variables and factors are numbered in the order of their appearance in *factorspec* statements. The estimated means of the observed variables are referred to as `[mean_`j`]_cons`, with $j = 1, \ldots, p$ indexing the observed variables. The factor loadings are `[lambda_`j`_`k`]_cons`. The factor variances and covariances are `[phi_`k`_`l`]_cons`, $1 \leq k \leq l \leq m$. The error variances are `[theta_`j`]_cons`, and error covariances, if specified, are `[theta_`j`_`h`]_cons`.

If the `usenames` option is specified, all the variable and factor indices are replaced with their names in the dataset and factor specifications.

Thus, for instance, the model

```
. confa (f: x1 x2 x3 x4)
```

will have the `lambda_1_1`, `lambda_2_1`, `lambda_3_1`, `lambda_4_1`, `phi_1_1`, `theta_1`, `theta_2`, `theta_3`, and `theta_4` parameters with default settings; and the `lambda_x1_f`, `lambda_x2_f`, `lambda_x3_f`, `lambda_x4_f`, `phi_f_f`, `theta_x1`, `theta_x2`, `theta_x3`, and `theta_x4` parameters when the `usenames` option is specified. Specifying the `usenames` option will make the low-level output (such as `matrix list e(b)`) produce very long and sparse listings. On the other hand, it is extremely handy when comparing models using the `estimates table` command or when transferring starting values between commands, as shown in one of the examples above.

The saved results include the standard outcomes from `ml`, such as `e(N)` and `e(ll)`. Additional saved results are as follows:

(*Continued on next page*)

Scalars

`e(pstar)`	total degrees of freedom
`e(df_m)`	model degrees of freedom
`e(df_u)`	residual degrees of freedom
`e(ll_0)`	log likelihood of the unrestricted model, $\widehat{\Sigma} = S$
`e(ll)`	log likelihood at the maximum
`e(ll_indep)`	log likelihood of "independence" model
`e(lr_u)`	likelihood-ratio statistic against unrestricted model; same as `e(chi2)`
`e(p_u)`	p-value against unrestricted model; same as `e(p)`
`e(lr_indep)`	likelihood ratio against "independence" model
`e(df_indep)`	model degrees of freedom of "independence" model
`e(p_indep)`	p-value against "independence" model

Macros

`e(factors)`	list of factors
`e(observed)`	list of observed variables
`e(factor`k`)`	unabbreviated factor statements, $k = 1, \ldots, m$
`e(correlated)`	unabbreviated correlated errors statements
`e(unitvar)`	the list of factors identified by unit variances
`e(missing)`	indicates that `missing` option was specified

Matrices

`e(S)`	sample covariance	`e(Sigma)`	implied covariance
`e(Lambda)`	estimated loadings, $\widehat{\Lambda}$	`e(Theta)`	estimated error variances, $\widehat{\Theta}$
`e(Phi)`	estimated factor covariances, $\widehat{\Phi}$	`e(CONFA_Struc)`	model structure description

Additional saved results posted when the `vce(sbentler)` option is used are the following:

Scalars

`e(SBc)`	scaling correction $\widehat{c}$ in (18)	`e(Tsc)`	scaled statistic, T_{sc}, in (18)
`e(SBd)`	scaling correction $\widehat{d}$ in (19)	`e(p_Tsc)`	p-value associated with T_{sc}
`e(T2)`	T_2 statistic in (20)	`e(Tadj)`	adjusted statistic, T_{adj}, in (19)
`e(p_T2)`	p-value associated with T_2	`e(p_Tsc)`	p-value associated with T_{adj}

Matrices

`e(SBU)`	matrix U in (17)	`e(SBDelta)`	matrix $\widehat{\Delta}$ in (15)
`e(SBV)`	matrix V in (14)	`e(SBGamma)`	matrix $\widehat{\Gamma}_n$ in (13)

Additional saved results posted by `bollenstine` are the following:

Scalars

`e(B_BS)`	number of replications	`e(T_BS_05)`	5th bootstrap percentile
`e(p_u_BS)`	bootstrap p-value	`e(T_BS_95)`	95th bootstrap percentile

Values returned by `estat fit` are the following:

Scalars

`r(AIC)`	AIC	`r(RMSEA)`	root mean squared error of approximation (12)
`r(BIC)`	BIC		
`r(CFI)`	CFI (9)	`r(RMSEA05)`	5% lower limit for RMSEA
`r(TLI)`	TLI (10)	`r(RMSEA95)`	95% upper limit for RMSEA
`r(RMSR)`	root mean squared residual (11)		

5.4 Computational complexity

A small simulation was conducted to establish the computational complexity of `confa`, i.e., the approximate functional dependence of computational time on the number of observations, size, and structure of the model. Sample size varied from 100 to 1,000, the number of factors varied from 1 to 5, and the number of indicators per factor varied from 2 to 6.

Table 2. Computational complexity simulation results

	(1)	(2)	(3)	(4)	(5)	(6)
# observations	0.680	0.680	0.680	0.680	0.680	0.680
# factors	2.283		2.469	0.341		
# observed variables		2.368		2.128		1.245
# indicators per factor			2.128			
# parameters					2.207	1.059
AIC	984.48	−226.93	−415.16	−415.16	−201.37	−382.49
BIC	996.51	−214.89	−399.12	−399.12	−189.34	−366.45
R^2	0.7541	0.9874	0.9921	0.9921	0.9866	0.9914

The results are summarized in table 2. The entries are coefficients in the regression, where the dependent variable is the log of elapsed time and explanatory variables are the logs of the quantities in the first column. The dependence on the sample size is of the order $O(n^{0.68})$ (the sample size is orthogonal to the size and model structure, in the sense of ANOVA factor orthogonality). The dependence on the model complexity is of the order $O(k^{2.4})$, where model complexity k can be understood as the number of parameters t, the number of observed variables p, or the number of factors m.

Those dependencies are within expectations. The only dependence on the sample size is due to the summation of the likelihood terms, and sublinear growth indicates good memory management and speed optimization of array arithmetics by Stata. The growth rate of computational time in model complexity between quadratic and cubic is indicative of the matrix manipulation complexity, because the algorithms of $k \times k$ matrix inversion achieve complexity between $O(k^3)$ for simple algorithms down to approximately $O(k^{2.4})$ for the fastest ones. The matrix inversion operations involved are inversion of $p \times p$ matrix $\Sigma(\theta)$ and inversion of $t \times t$ Hessian matrix in the Newton–Raphson optimization method.

5.5 Verification and certification

Verification (Gould 2001) of `confa` estimation results was conducted using some published results (Yuan and Bentler 2007; Bollen 1993) as well as other software packages for Holzinger–Swineford data. `confa` reproduced the point estimates and standard errors reported by Mplus 3.1 (Muthén and Muthén 2004). However, both sets of results disagreed in the third decimal place with the published results of Yuan and Bentler

(2007). Both Mplus output and Yuan and Bentler (2007) were given to three decimal places. `confa` agreed with `gllamm` (running with adaptive quadrature and 12 integration points per factor) to at least two decimal places in point estimates, OIM standard errors, and robust standard errors (see [R] ***vce_option***) for all parameters except the error variances $V[\delta]$. The discrepancies in the latter are likely due to a different implementation of the error variance parameters in `gllamm` via a nonlinear transformation.

5.6 Distribution

The package is maintained and updated by the author, Stanislav Kolenikov. To check for the most recent update, in Stata type

```
. net from http://web.missouri.edu/~kolenikovs/stata/
```

The version of the package at the time of publication is 2.0. Please send comments and bug reports to the email address given on the title page.

6 Acknowledgments

Support from NSF grant SES-0617193 with funds from the Social Security Administration is gratefully acknowledged. I would also like to thank Ken Higbee for helpful comments on improving the code stability, Sophia Rabe-Hesketh and Jeff Pitblado for helpful discussions on numeric issues, Ke-Hai Yuan for providing the verification data, and an anonymous referee for thorough suggestions that led to improvements in missing-data analysis, as well as clearer exposition throughout the paper. All remaining errors, omissions, and bugs are my responsibility.

7 References

Anderson, T. W., and Y. Amemiya. 1988. The asymptotic normal distribution of estimators in factor analysis under general conditions. *Annals of Statistics* 16: 759–771.

Bartholomew, D. J., and M. Knott. 1999. *Kendall's Library of Statistics 7: Latent Variable Models and Factor Analysis*. 2nd ed. London: Arnold.

Bentler, P. M. 1990a. Comparative fit indexes in structural models. *Psychological Bulletin* 107: 238–246.

———. 1990b. Fit indexes, Lagrange multipliers, constraint changes and incomplete data in structural models. *Multivariate Behavioral Research* 25: 163–172.

Beran, R., and M. S. Srivastava. 1985. Bootstrap tests and confidence regions for functions of a covariance matrix. *Annals of Statistics* 13: 95–115.

Bollen, K. 1993. Liberal democracy: Validity and method factors in cross-national measures. *American Journal of Political Science* 37: 1207–1230.

Bollen, K., and R. Stine. 1992. Bootstrapping goodness-of-fit measures in structural equation models. *Sociological Methods and Research* 21: 205–229.

Bollen, K. A. 1989. *Structural Equations with Latent Variables*. New York: Wiley.

———. 1996. An alternative two stage least squares (2SLS) estimator for latent variable equations. *Psychometrika* 61: 109–121.

Brown, T. A. 2006. *Confirmatory Factor Analysis for Applied Research*. New York: Guilford Press.

Browne, M. W. 1984. Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology* 37: 62–83.

———. 1987. Robustness of statistical inference in factor analysis and related models. *Biometrika* 74: 375–384.

Browne, M. W., and R. Cudeck. 1993. Alternative ways of assessing model fit. In *Testing Structural Equation Models*, ed. K. A. Bollen and J. S. Long, 136–162. Newbury Park, CA: Sage.

Byrne, B. M., and R. D. Goffin. 1993. Modeling MTMM data from additive and multiplicative covariance structures: An audit of construct validity concordance. *Multivariate Behavioral Research* 28: 67–96.

Cox, N. J. 2008. Speaking Stata: Correlation with confidence, or Fisher's z revisited. *Stata Journal* 8: 413–439.

Gould, W. 2001. Statistical software certification. *Stata Journal* 1: 29–50.

Gould, W., J. Pitblado, and W. Sribney. 2006. *Maximum Likelihood Estimation with Stata*. 3rd ed. College Station, TX: Stata Press.

Grayson, D., and H. Marsh. 1994. Identification with deficient rank loading matrices in confirmatory factor analysis: Multitrait–multimethod models. *Psychometrika* 59: 121–134.

Holzinger, K. J., and F. Swineford. 1939. A study in factor analysis: The stability of a bifactor solution. Technical Report 48, Supplementary Educational Monographs, University of Chicago.

Huber, P. J. 1967. The behavior of maximum likelihood estimates under nonstandard conditions. In Vol. 1 of *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 221–233. Berkeley: University of California Press.

Jöreskog, K. G. 1969. A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika* 34: 183–202.

Jöreskog, K. G., and A. S. Goldberger. 1972. Factor analysis by generalized least squares. *Psychometrika* 37: 243–260.

Jöreskog, K. G., and D. Sörbom. 1986. *Lisrel* VI*: Analysis of linear structural relationships by the method of maximum likelihood*. Mooresville, IN: Scientific Software.

Lawley, D. N., and A. E. Maxwell. 1971. *Factor Analysis as a Statistical Method*. 2nd ed. London: Butterworths.

Little, R. J. A., and D. B. Rubin. 2002. *Statistical Analysis with Missing Data*. 2nd ed. Hoboken, NJ: Wiley.

Magnus, J. R., and H. Neudecker. 1999. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Rev. ed. New York: Wiley.

Marsh, H. W., J. R. Balla, and K.-T. Hau. 1996. An evaluation of incremental fit indices: A clarification of mathematical and empirical properties. In *Advanced Structural Equation Modeling: Issues and Techniques*, ed. G. A. Marcoulides and R. E. Schumacker, 315–353. Mahwah, NJ: Erlbaum.

Marsh, H. W., B. M. Byrne, and R. Craven. 1992. Overcoming problems in confirmatory factor analyses of MTMM data: The correlated uniqueness model and factorial invariance. *Multivariate Behavioral Research* 27: 489–507.

Muthén, L. K., and B. O. Muthén. 2004. *Mplus: Statistical Analysis with Latent Variables. User's Guide*. Los Angeles, CA, 3rd ed.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata Journal* 2: 1–21.

———. 2004. Generalized multilevel structural equation modeling. *Psychometrika* 69: 167–190.

Satorra, A. 1990. Robustness issues in structural equation modeling: A review of recent developments. *Quality and Quantity* 24: 367–386.

Satorra, A., and P. M. Bentler. 1994. Corrections to test statistics and standard errors in covariance structure analysis. In *Latent Variables Analysis*, ed. A. von Eye and C. C. Clogg, 399–419. Thousand Oaks, CA: Sage.

———. 2001. A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika* 66: 507–514.

Steiger, J. H. 1990. Structural model evaluation and modification: An interval estimation approach. *Multivariate Behavioral Research* 25: 173–180.

Tucker, L. R., and C. Lewis. 1973. A reliability coefficient for maximum likelihood factor analysis. *Psychometrika* 38: 1–10.

Yuan, K.-H., and P. M. Bentler. 1997. Mean and covariance structure analysis: Theoretical and practical improvements. *Journal of the American Statistical Association* 92: 767–774.

———. 2007. Structural equation modeling. In *Handbook of Statistics 26: Psychometrics*, ed. C. R. Rao and S. Sinharay, 297–358. Oxford: Elsevier.

About the author

Stanislav Kolenikov is an assistant professor in the Department of Statistics at the University of Missouri in Columbia, Missouri. His research interests include statistical methods in social sciences, with focus on structural equation models, microeconometrics, and analysis of complex survey data.