

Confirmatory factor analysis using `cfa`

Stanislav Kolenikov
University of Missouri
Columbia, MO/USA
kolenikovs@missouri.edu

Abstract. This article describes `cfa` package that fits confirmatory factor analysis models by maximum likelihood and provides diagnostics for the fitted models. Description of the commands and their options is given, and some illustrative examples are provided.

Keywords: `st0001`, Bollen-Stine bootstrap, confirmatory factor analysis, CFA, factor scores, Satorra-Bentler corrections

1 Confirmatory factor analysis

In a wide range of research problems, especially in social sciences, the researcher may not have access to direct measurements of the variables of interest. Intellectual ability is not something that can be measured in centimeters or kilograms, however more able people would be able to work on mental problems faster, make fewer errors, and/or solve more difficult problems. This is the concept that underlies 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 GDP per capita as a measure of economic development). Political scientists would have to rely on expert judgement 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 a number of proxy variables that share correlation with that (latent) variable, but also contain measurement error. A popular tool to analyze those kind of problems is confirmatory factor analysis (CFA). This is a multivariate statistical technique used to assess the researcher's theory that 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). It 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" or "latent factors" will be used interchangeably in this article.

The method differs quite 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 a number of models and compares them using fit cri-

teria, 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). In other words, 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, exploratory factor analysis is available via `factor` estimation command and associated suite of post-estimation commands. See [R] `factor`.

1.1 The model and identification

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

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

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

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, the model (1) can be written as:

$$\mathbf{y}_i = \boldsymbol{\mu} + \boldsymbol{\Lambda} \boldsymbol{\xi}_i + \boldsymbol{\delta}_i \quad (2)$$

where vectors $\boldsymbol{\mu}$, $\boldsymbol{\xi}_i$ and $\boldsymbol{\delta}_i$ denote regression intercepts, latent variables/factors and measurement errors, respectively, and $\boldsymbol{\Lambda}$ is the matrix of factor loadings. The measurement errors $\boldsymbol{\delta}$ are assumed to be independent of the factors $\boldsymbol{\xi}$. Let us additionally introduce the following (matrices of) parameters:

$$\boldsymbol{\Phi} = \mathbb{V}[\boldsymbol{\xi}] = \mathbb{E}[\boldsymbol{\xi} \boldsymbol{\xi}'], \quad \boldsymbol{\Theta} = \mathbb{V}[\boldsymbol{\delta}] = \mathbb{E}[\boldsymbol{\delta} \boldsymbol{\delta}'] \quad (3)$$

employing the usual convention that $\mathbb{E}[\boldsymbol{\xi}] = \mathbf{0}$, $\mathbb{E}[\boldsymbol{\delta}] = \mathbf{0}$. Then the covariance matrix of the observed variables is

$$\mathbb{V}[\mathbf{y}] = \mathbb{E}[(\mathbf{y} - \boldsymbol{\mu})(\mathbf{y} - \boldsymbol{\mu})'] = \mathbb{E}[(\boldsymbol{\Lambda} \boldsymbol{\xi}_i + \boldsymbol{\delta}_i)(\boldsymbol{\Lambda} \boldsymbol{\xi}_i + \boldsymbol{\delta}_i)'] = \boldsymbol{\Lambda} \boldsymbol{\Phi} \boldsymbol{\Lambda}' + \boldsymbol{\Theta} = \boldsymbol{\Sigma}(\boldsymbol{\theta}) \quad (4)$$

where all parameters are put together into vector θ .

Let us highlight the distinctions between EFA and CFA again using the matrix formulation (4). 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 a lot of 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 only loads on 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, they need 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 (4).

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's the convention adopted by `cfa`.

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. In that case, the latent variable will have the units of that observed variable, which might be useful if the latter is meaningful (e.g., the latent variable is wealth, and the measured 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 non-redundant entries of the covariance matrix (4):

$$\dim \theta = t \leq p^* = p(p+1)/2 \quad (5)$$

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, $\hat{\mu}_j = \bar{y}_j$). If $t = p^*$, the model is (said to be) *exactly identified*, and if $t > p^*$, *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. A number of sufficient identification rules has been developed for confirmatory factor analysis. Bollen (1989) lists the following ones:

1. *three indicator rule*: if the model has factor complexity 1, the covariance matrix of

the error terms $\mathbb{V}[\delta] = \Theta$ 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 $\mathbb{V}[\delta] = \Theta$ is diagonal, there are more than one factor in the model ($m > 1$), each row of Φ has at least one non-zero 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 the CFA model (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

$$\begin{aligned} \ln L(\mathbf{Y}, \Sigma(\theta)) &= - \sum_{i=1}^n \left[\frac{p}{2} \ln 2\pi + \frac{1}{2} \ln |\Sigma(\theta)| + \frac{1}{2} (\mathbf{y}_i - \mu)' \Sigma^{-1}(\theta) (\mathbf{y}_i - \mu) \right] \\ &= - \frac{np}{2} \ln 2\pi - \frac{n}{2} \ln |\Sigma(\theta)| - \frac{1}{2} \text{tr} \Sigma^{-1}(\theta) S \end{aligned} \quad (6)$$

where S is the maximum likelihood estimate of the (unstructured) covariance matrix of the data. The likelihood (6) can be maximized with respect to the parameters to obtain the MLEs $\hat{\theta}$ 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 et al. 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 (i) the model structure is correctly specified and (ii) 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 non-normal 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 automatically 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 `cfa`.

Once the maximum likelihood estimates $\hat{\theta}$ are obtained, one can form the *implied* covariance matrix $\Sigma(\hat{\theta})$. 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

$$H_0 : \mathbb{V}[\mathbf{y}] = \Sigma(\theta) \quad (7)$$

cannot be rejected.

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

$$T = -2[\ln L(\mathbf{Y}, \Sigma(\hat{\theta})) - \ln L(\mathbf{Y}, S)] \xrightarrow{d} \chi_q^2 \quad (8)$$

that 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 et al. (1996). Absolute measures of fit are addressing the absolute values of the residuals, defined as the entries of the difference matrix $S - \Sigma(\hat{\theta})$. An example of such measure is the root of mean squared residual, RMSR, given by (18). 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 RMSEA, the root mean squared error of approximation (19). Values of 0.05 or less, or confidence intervals covering those ranges, are usually considered to indicate a good fit. Comparative fit indices relate the attained fit of the model to the independence model when $\Sigma(\cdot) = \text{diag } S$ 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 value greater than 1, and that is usually taken as indication of overfitting. Two such indices are reported by `cfa` post-estimation suite, Tucker-Lewis non-normed 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 (8) has a non-standard distribution based on the sum of weighted χ_1^2 variables. Satorra and Bentler (1994) proposed Satterthwaite-type corrections: T_{sc} statistic corrects the scale of the distribution, and T_{adj} corrects both the scale and the number of degrees of freedom.

An alternative procedure to correct for the non-standard distribution of the LRT 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 to obtain the data set with variables

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

guaranteed to be compatible with the model (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 it, and the test statistic T_b computed, with the whole process repeated for $b = 1, \dots, B$ sufficiently many times. The bootstrap p -value associated with test statistic T is the fraction of exceedances:

$$p_{BS} = \frac{1}{B} \#\{b : T_b > T\} \quad (9)$$

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* $\hat{\xi}$. 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*, and the resulting predictions, as the *factor scores*.

Two common factor scoring methods are implemented through `predict` post-estimation command of `cfa` package. 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

$$\hat{\xi}_{ri} = \hat{\Phi} \hat{\Lambda}' \Sigma^{-1}(\hat{\theta})(\mathbf{y}_i - \hat{\mu}) \quad (10)$$

The hatted matrices are the matrices of the maximum likelihood estimates of the model parameters. Equation (10) can also be justified as an empirical Bayes estimator of $\hat{\xi}_i$, with the model giving the prior distribution $\xi \sim N(0, \hat{\Phi})$, and the data from i -th 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

$$\hat{\xi}_{Bi} = (\hat{\Lambda}' \hat{\Theta} \hat{\Lambda})^{-1} \hat{\Lambda}' \hat{\Theta}^{-1}(\mathbf{y}_i - \hat{\mu}) \quad (11)$$

It is also known as the maximum likelihood method, as it provides the maximum likelihood estimates of ξ conditional on the data \mathbf{y}_i , with a mild abuse of notation since 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 Bartlett method are unbiased, but have greater variance, while the factor scores obtained from regression method are shrunk towards zero.

2 Description of `cfa` package

The package `cfa` contains estimation and post-estimation commands for confirmatory factor analysis. Single level single group estimation is supported¹. A variety of identification condition can be imposed, and robust standard errors can be reported. Goodness of fit tests can be corrected using Satorra and Bentler (1994) scaling approach or using Bollen and Stine (1992) bootstrap. Complex survey designs specified through [SVY] `svyset` are supported.

2.1 Syntax

```
cfa factorspec [factorspec ...] [if] [in] [weight] [, correlated(corrspec
  [corrspec ...]) unitvar(factorlist) free missing from(fromspec)
  constraint(numlist) vce(vcetype) usenames level(#) ml_maximize_options ]
```

The factor specification `factorspec` is

```
(factorname:varlist)
```

The correlated errors specification `corrspec` is

```
[ ( )varname1:varname2( ) ]
```

The initial values specification `fromspec` is one of:

```
smart | ones | ivreg | 2sls | ml_init_args
```

The list of factors `factorlist` is comprised of `factornames`.

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

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

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

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

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

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

2.2 Options of cfa

`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 options `free iter(0)` and then look at the names in the output of `matrix list e(b)` to find out the specific names of the parameters.

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

$$[(] \text{ varname}_k : \text{varname}_j [)]$$

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. There should be no space between the colon and *varname_j*.

`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. Seldom used.

`from(ones | 2s1s | iv | smart | ml_init_args)` provides the choice of starting values for the maximization procedure. The `ml`'s internal default is to set all parameters to zero, which leads to a non-invertible matrix Σ , and `ml` has to make a lot of changes to those initial values to find anything feasible. Moreover, this initial search procedure sometimes leads to a domain where the likelihood is non-concave, and optimization might fail there.

Option `ones` sets all the parameters to the values of one except for covariance parameters (off-diagonal values of the Φ and Θ matrices) that 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).

Options `2s1s` or `iv` request the initial parameters for the freely estimated loadings to be set to the 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 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 are performed.

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

Other specification of starting values 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`.

`level(#)` allows to change the confidence level for CI reporting.

`missing` requests FIML estimation with missing data.

unitvar(*factorlist*) 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.)

useenames specifies an alternative labeling of coefficients.

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

An additional estimator specific to SEM is 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.

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

2.3 Options of estat

Post-estimation command **estat** `fitindices` produces fit indices and supports the following options:

aic requests AIC, Akaike information criteria.

bic requests BIC, Schwarz Bayesian information criteria.

cfi requests CFI, comparative fit index (Bentler 1990b).

rmsea requests RMSEA, root mean squared error of approximation (Browne and Cudeck 1993).

srmr requests RMSR, root mean squared residual

tli requests TLI, Tucker-Lewis index (Tucker and Lewis 1973).

`_all` prints all of the above indices. This is the default behavior if no options are specified.

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

Post-estimation command **estat** `correlate` transforms the covariance parameters (off-diagonal values of Φ and Θ) to correlations and supports the following options:

bound provides alternative asymmetrical confidence intervals based on Fisher's *z*-transform (Cox 2008)

`level(#)` allows to change the confidence level for CI reporting.

2.4 Options of predict

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

`regression`, `empiricalbayes` or `ebayes` request regression, or empirical Bayes, factor scoring procedure (10).

`bartlett` or `mle` request Bartlett scoring procedure (11).

2.5 Options of bollenstine

`cfaoptions(string)` allows transfer of `cfa` options to `bollenstine`. If non-default model options (`unitvar` and `correlated`) were used, one would need to utilize them with `bollenstine` as well.

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

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

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

`saving(filename)` allows to save the bootstrap results.

Other bootstrap options (except for the forced `notable noheader nolegend 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 data set for confirmatory factor analysis is based on Holzinger and Swineford (1939) data also analyzed by Jöreskog (1969)². The data set contains the measures of performance of 301 children in grades 7 and 8 from two different schools on several psychometric tests. The complete data set 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 follows³:

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

3. The data set appears to be in public domain, so a version of it can be placed on Stata Journal website. Some of the variables have been rescaled to have variance between 1 and 2 to improve

```

. use http://web.missouri.edu/~kolenikovs/stata/hs-cfa.dta
(Holzinger & Swineford (1939))

. describe

Contains data from http://web.missouri.edu/~kolenikovs/stata/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 this data grouping the variables together in three factors: “visual” factor (variables x1–x3), “textual” factor (variables x4–x6) and “math” factor (variables x7–x9). In matrix terms,

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

$$\mathbb{V}[\xi] = \Phi, \quad \mathbb{V}[\delta] = \text{diag}(\theta_1, \dots, \theta_9), \quad \text{COV}[\xi, \delta] = 0$$

convergence and ensure comparability with published results.

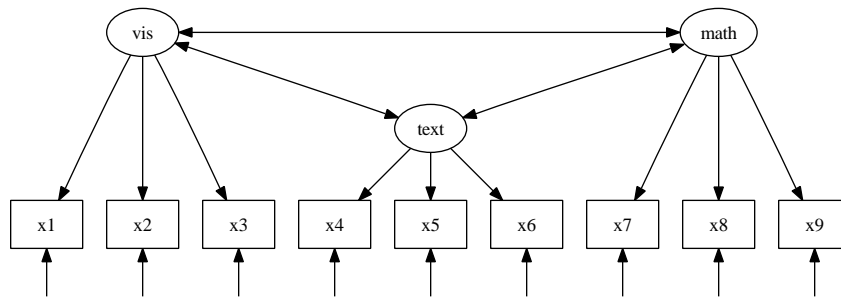


Figure 1: The basic model for Holzinger-Swineford data.

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

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

```
. cfa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9)
initial:      log likelihood = -17210154
rescale:      log likelihood = -17210154
rescale eq:   log likelihood = -4187.6584
could not calculate numerical derivatives
flat or discontinuous region encountered
```

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 `cfa`. For the standardized data, option `from(ones)` 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(2s1s)` to get the initial values of loadings from 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, option `from(smooth)` 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 `from(iv)` settings. Let us demonstrate those procedures:

(Continued on next page)

```

. cfa (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 = -3786.3639
Iteration 2:  log likelihood = -3778.5185 (not concave)
Iteration 3:  log likelihood = -3748.4058
Iteration 4:  log likelihood = -3744.4812 (backed up)
Iteration 5:  log likelihood = -3738.3336
Iteration 6:  log likelihood = -3737.8055
Iteration 7:  log likelihood = -3737.7453
Iteration 8:  log likelihood = -3737.7449
(output omitted)
. cfa (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 = -3747.5599
Iteration 2:  log likelihood = -3740.8656
Iteration 3:  log likelihood = -3737.8022
Iteration 4:  log likelihood = -3737.7451
Iteration 5:  log likelihood = -3737.7449
(output omitted)
. cfa (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 = -3883.7069 (not concave)
Iteration 2:  log likelihood = -3804.4653
Iteration 3:  log likelihood = -3768.3843
Iteration 4:  log likelihood = -3739.6495
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 (i) the model is correctly specified, (ii) there are no variables of factor complexity more than 1, and (iii) there are no correlated measurement errors. All other starting value proposals, on the other hand, have some ad-hoc heuristics that produces 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:

(Continued on next page)

Confirmatory factor analysis

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	.062695	66.77	0.000	4.063022	4.308782
x8	5.527076	.0582688	94.85	0.000	5.412872	5.641281
x9	5.374123	.0580694	92.55	0.000	5.260309	5.487937
Loadings						
vis						
x1	1
x2	.5535004	.1092473	5.07	0.000	.3393797	.767621
x3	.7293706	.1172677	6.22	0.000	.4995301	.9592111
text						
x4	1
x5	1.113076	.0649865	17.13	0.000	.9857052	1.240448
x6	.9261463	.0561947	16.48	0.000	.8160067	1.036286
math						
x7	1
x8	1.179963	.150285	7.85	0.000	.8854101	1.474516
x9	1.081522	.19511	5.54	0.000	.6991138	1.463931
Factor cov.						
vis-vis	.8093149	.1497557	5.40	0.000	.5157991	1.102831
text-text	.9794918	.1122102	8.73	0.000	.7595638	1.19942
vis-text	.4082318	.0796757	5.12	0.000	.2520703	.5643932
math-math	.3837355	.0920521	4.17	0.000	.2033168	.5641543
text-math	.1734924	.0493121	3.52	0.000	.0768425	.2701424
vis-math	.262222	.0553823	4.73	0.000	.1536746	.3707694
Var[error]						
x1	.549055	.1190493	4.61	0.000	.3157225	.7823874
x2	1.133841	.1042624	10.87	0.000	.9294901	1.338191
x3	.8443251	.0950748	8.88	0.000	.6579819	1.030668
x4	.3711732	.047963	7.74	0.000	.2771676	.4651789
x5	.4462556	.0579336	7.70	0.000	.3327079	.5598033
x6	.3562027	.0434406	8.20	0.000	.2710607	.4413447
x7	.7993925	.0875572	9.13	0.000	.6277836	.9710013
x8	.4876912	.0916591	5.32	0.000	.3080427	.6673397
x9	.566136	.0905773	6.25	0.000	.3886079	.7436642
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	0.4407					

Goodness of fit test: LR = 85.306 ; Prob[chi2(24) > LR] = 0.0000
 Test vs independence: LR = 833.546 ; Prob[chi2(12) > 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 `cfa`; factor covariances ϕ_{kl} ; and variances of the error terms δ_j . All parameters are freely estimated, except for loadings used for identification (have coefficient estimate equal to 1 and missing standard errors). This implies the covariances are not guaranteed to comply with Cauchy inequality, and the error variances are not guaranteed to be non-negative. 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 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 are likelihood ratios. The first line is the test against a saturated model (when $\hat{\Sigma} = S$), and the second line is the test against an independence model (when $\hat{\Sigma} = \text{diag } S$). 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 to the null model in which variables are assumed independent.

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

```
. cfa (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 one of such occurrences. 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 variance-covariance matrix of the coefficient estimates. Other types of estimators are known in SEM, most prominently Satorra and Bentler (1994) variance estimator (24). It can be specified with a non-standard `vce(sbentler)` option:

(Continued on next page)

```
. cfa (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]
<i>(output omitted)</i>					
Factor cov.					
vis-vis	.8093164	.1618238	5.00	0.000	.4921476 1.126485
text-text	.9794923	.1187478	8.25	0.000	.746751 1.212234
vis-text	.4082327	.0803488	5.08	0.000	.2507521 .5657134
math-math	.3837481	.0804101	4.77	0.000	.2261472 .5413489
text-math	.1734949	.0551705	3.14	0.002	.0653627 .281627
vis-math	.2622249	.0543577	4.82	0.000	.1556857 .3687641
Var[error]					
x1	.5490544	.1403178	3.91	0.000	.2740366 .8240721
x2	1.133841	.1007102	11.26	0.000	.9364527 1.331229
x3	.8443249	.0813373	10.38	0.000	.6849068 1.003743
x4	.3711733	.0475621	7.80	0.000	.2779534 .4643932
x5	.4462556	.0526208	8.48	0.000	.3431208 .5493904
x6	.3562027	.0447916	7.95	0.000	.2684128 .4439926
x7	.7993922	.0713343	11.21	0.000	.6595796 .9392048
x8	.487698	.0701501	6.95	0.000	.3502064 .6251896
x9	.5661313	.0629796	8.99	0.000	.4426936 .689569
<i>(output omitted)</i>					
Goodness of fit test: LR = 85.306 ; Prob[chi2(24) > LR] = 0.0000					
Test vs independence: LR = 833.546 ; Prob[chi2(12) > 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. 4). These additional tests tend to perform much better. The technical description is given in Section 5; see equation (24) for Satorra-Bentler standard errors, and equations (26)–(28) for the additional test statistics.

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

```
. cfa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(iv) robust nolog
```

(Continued on next page)

Log likelihood = -3737.7449 Number of obs = 301

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
<i>(output omitted)</i>						
Loadings						
vis						
x1	1
x2	.5534998	.1092474	5.07	0.000	.3393788	.7676208
x3	.7293701	.1172679	6.22	0.000	.4995292	.959211
text						
x4	1
x5	1.113076	.0649865	17.13	0.000	.9857052	1.240448
x6	.9261465	.0561948	16.48	0.000	.8160068	1.036286
math						
x7	1
x8	1.17995	.1502875	7.85	0.000	.8853917	1.474508
x9	1.08153	.1951236	5.54	0.000	.6990951	1.463965
Factor cov.						
vis-vis	.8093164	.1497563	5.40	0.000	.5157995	1.102833
text-text	.9794923	.1122103	8.73	0.000	.7595641	1.199421
vis-text	.4082327	.079676	5.12	0.000	.2520707	.5643948
math-math	.3837481	.0920634	4.17	0.000	.2033071	.5641891
text-math	.1734949	.0493135	3.52	0.000	.0768421	.2701476
vis-math	.2622249	.0553836	4.73	0.000	.1536751	.3707748
Var[error]						
x1	.5490544	.1190496	4.61	0.000	.3157214	.7823874
x2	1.133841	.1042625	10.87	0.000	.9294904	1.338192
x3	.8443249	.0950749	8.88	0.000	.6579816	1.030668
x4	.3711733	.047963	7.74	0.000	.2771676	.465179
x5	.4462556	.0579335	7.70	0.000	.332708	.5598032
x6	.3562027	.0434406	8.20	0.000	.2710606	.4413448
x7	.7993922	.0875598	9.13	0.000	.6277782	.9710062
x8	.487698	.09166	5.32	0.000	.3080477	.6673483
x9	.5661313	.0905796	6.25	0.000	.3885984	.7436641

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

Note that since the assumptions of the model are assumed to be 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` that omits ANOVA table, since this estimator potentially corrects for heteroskedasticity of error terms, and in presence of heteroskedasticity, sums of squared errors are not particularly meaningful. Unlike Satorra-Bentler 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 to Satorra-Bentler estimator.

Correlated errors

It was argued in substantive literature that one of the reasons the basic CFA model does not fit well 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 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)
. cfa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(bb, skip)
> corr( x7:x8 ) robust
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 = -3732.2811
Iteration 2:  log likelihood = -3730.0615
Iteration 3:  log likelihood = -3722.9903 (not concave)
Iteration 4:  log likelihood = -3722.2229
Iteration 5:  log likelihood = -3721.852
Iteration 6:  log likelihood = -3721.7299
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	.575844	.1034752	5.57	0.000	.3730364	.7786517
x2	1.122499	.1019974	11.01	0.000	.9225874	1.32241
x3	.8321161	.0898741	9.26	0.000	.6559661	1.008266
x4	.3722486	.0479868	7.76	0.000	.2781961	.466301
x5	.4436605	.0580119	7.65	0.000	.3299593	.5573616
x6	.3570578	.0434528	8.22	0.000	.2718919	.4422237
x7	1.036463	.0881249	11.76	0.000	.8637413	1.209185
x8	.7948166	.083143	9.56	0.000	.6318592	.9577739
x9	.0875321	.1966993	0.45	0.656	-.2979914	.4730555
Cov[error]						
x7-x8	.3527072	.066299	5.32	0.000	.2227636	.4826509
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(13) > LR] = 0.0000
```

Note the use of starting values: the previous parameter estimates are saved and transferred via `from(..., skip)` option. The option `skip` 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 that of 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 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 factor `math` 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_1^2 . Virtually identical results can be obtained with `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 non-normal data.

Satorra-Bentler scaled difference test

Non-normality 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-)LRT test statistics T_0 and T_1 , degrees of freedom r_0 and r_1 , and scaling factors c_0 and c_1 where index 0 stands for a more restrictive (null) model. Then the test statistic is:

$$\bar{T}_d = \frac{(T_0 - T_1)(r_0 - r_1)}{r_0 c_0 - r_1 c_1} \quad (12)$$

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

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

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

(Continued on next page)

```
. 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 referred to χ_1^2 distribution again confirms that the correlation is significant.

Bollen-Stine bootstrap

Besides the Satorra-Bentler fit statistics T_{sc} and T_{adj} reported with `vce(sbentler)` option, an alternative way to correct fit statistics for non-normality 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 `bollenstine` command as a part of `cfa` suite. 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 `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 `cfaoptions(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 4 samples where the bootstrap did not converge, shown with `x` among the dots produced by the `bootstrap`. If the number of iterations is set to 5, only 208 out of 500 bootstrap samples produce convergent results.

Note the use of `cfaoptions(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.

```
. qui cfa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(bb, skip) corr(x7:x8)
. set seed 1010101
. bollenstine , reps(500) cfaoptions( iter(20) corr( x7:x8 ) )
(running cfa on estimation sample)
Bootstrap replications (500)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 50
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 100
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 150
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 200
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 250
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 300
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 350
.....|.....|.....|.....|.....|.....|.....|.....|.....|.....| 400
```

(Continued on next page)

```

..... 450
..... 500
Log likelihood = -3721.7283          Number of obs = 301

```

	Bollen-Stine		z	P> z	[95% Conf. Interval]	
	Coef.	Std. Err.				
<i>(output omitted)</i>						
Factor cov.						
vis-vis	.7825264	.1362378	5.74	0.000	.5155053	1.049548
text-text	.9784181	.1121732	8.72	0.000	.7585626	1.198274
vis-text	.3995373	.0777012	5.14	0.000	.2472456	.5518289
math-math	.1466814	.0528321	2.78	0.005	.0431323	.2502305
text-math	.1021689	.0360053	2.84	0.005	.0315999	.1727379
vis-math	.1843774	.0512246	3.60	0.000	.083979	.2847757
Var[error]						
x1	.5758435	.1034752	5.57	0.000	.3730357	.7786512
x2	1.122499	.1019974	11.01	0.000	.9225874	1.32241
x3	.8321179	.0898743	9.26	0.000	.6559676	1.008268
x4	.3722481	.0479868	7.76	0.000	.2781956	.4663006
x5	.4436611	.0580119	7.65	0.000	.3299598	.5573623
x6	.3570581	.0434528	8.22	0.000	.2718922	.4422239
x7	1.036463	.088125	11.76	0.000	.8637415	1.209185
x8	.7948159	.0831431	9.56	0.000	.6318584	.9577735
x9	.0875349	.1966994	0.45	0.656	-.2979888	.4730586
Cov[error]						
x7-x8	.352707	.0662991	5.32	0.000	.2227632	.4826508
<i>(output omitted)</i>						

```

Goodness of fit test: LR = 53.272      ; Prob[chi2(23) > LR] = 0.0003
Test vs independence: LR = 865.579    ; Prob[chi2(13) > LR] = 0.0000
Bollen-Stine simulated Prob[ LR > 53.2722 ] = 0.0020
Based on 496 simulations. The bootstrap 90% interval: (13.258,39.852)

```

Standard errors have been replaced by the Bollen-Stine bootstrap ones. Additionally to the usual goodness of fit tests, the bootstrap p -value and the percentile method confidence interval for the goodness of fit test statistic are reported. The computations of the bootstrap p -value, the confidence interval, and the standard errors are based on the converged samples only (496 out of 500). Note how this confidence interval compares to 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 upwards. The actual test statistic of 53.27 is way outside either interval, and only one out of 496 bootstrap samples produced the test statistics above it.

Post-estimation commands: fit indices and correlations

There are several post-estimation commands available in `cfa` package that provide additional estimation and diagnostic results. First, several popular fit indices can be obtained via `estat fit` command:

(Continued on next page)

```
. estat fit
  Fit indices
RMSEA = 0.0662, 90% CI= (0.0430, 0.0897)
RMSR  = 0.0624
TLI    = 1.0000
CFI    = 0.9375
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's 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 confidence intervals can be computed directly by asymptotic normality, or via Fisher's z -transform (Cox 2008) requested by `bound` option that produces confidence intervals bounded to be within $(-1,1)$ interval, and shrunk towards zero. If there are any *Heywood cases*, that is, improper estimates with implied correlations outside $[-1, 1]$ interval, then z -transform is not applicable, and missing CI will result.

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

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
Factors						
vis-text	.4566102	.0642273	7.11	0.000	.330727	.5824934
vis-math	.544215	.0784652	6.94	0.000	.390426	.6980041
text-math	.2696926	.0684064	3.94	0.000	.1356186	.4037666
Errors						
x7-x8	.3886009	.0536639	7.24	0.000	.2834215	.4937802

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

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
Factors						
vis-text	.4566102	.0642273	7.11	0.000	.3220904	.5730568
vis-math	.544215	.0784652	6.94	0.000	.3727575	.6797386
text-math	.2696926	.0684064	3.94	0.000	.1311812	.397876
Errors						
x7-x8	.3886009	.0536639	7.24	0.000	.2786919	.4884623

Factor predictions

Factor predictions are obtained by the standard post-estimation 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	.573318					
fa2	.386133	.871388				
fa3	.17935	.101986	.13509			
fb1	.785135	.400868	.184991	1.15513		
fb2	.400868	.981677	.102509	.400869	1.10884	
fb3	.184689	.102902	.147169	.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 towards zero (and thus have smaller variances). The factor predictions obtained by the two methods are almost perfectly correlated, which is to be expected since they are measuring the same quantities, albeit on different scales.

Alternative identification

As the last twist that can be applied to this data, let us consider an alternative identification when factor variances are set to 1, and factor loading are estimated freely⁴.

```
. cfa (vis: x1 x2 x3) (text: x4 x5 x6) (math: x7 x8 x9), from(ones)
> unitvar(_all) corr(x7:x8)
initial:      log likelihood = -3933.9488
(output omitted)
Iteration 9:  log likelihood = -3721.7283
```

(Continued on next page)

4. 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 data sets found in different sources on the Internet.

Confirmatory factor analysis

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	.077005	11.49	0.000	.7336778	1.035532
x2	.5092014	.0782211	6.51	0.000	.3558908	.6625121
x3	.6653938	.0739123	9.00	0.000	.5205284	.8102593
text						
x4	.9891495	.0567019	17.44	0.000	.8780158	1.100283
x5	1.102781	.0625864	17.62	0.000	.9801141	1.225448
x6	.9161337	.0537635	17.04	0.000	.8107591	1.021508
math						
x7	.3829824	.0689758	5.55	0.000	.2477924	.5181724
x8	.4766186	.0775012	6.15	0.000	.324719	.6285183
x9	.9630581	.1106754	8.70	0.000	.7461382	1.179978
Factor cov.						
vis-vis	1
text-text	1
vis-text	.4566094	.0642274	7.11	0.000	.3307259	.5824928
math-math	1
text-math	.2696905	.0684081	3.94	0.000	.1356132	.4037678
vis-math	.5442123	.0784691	6.94	0.000	.3904157	.698009
Var[error]						
x1	.5758446	.1034751	5.57	0.000	.3730371	.7786521
x2	1.1225	.1019975	11.01	0.000	.9225882	1.322411
x3	.8321165	.0898742	9.26	0.000	.6559664	1.008267
x4	.3722483	.0479868	7.76	0.000	.2781959	.4663007
x5	.44366	.0580117	7.65	0.000	.3299591	.5573609
x6	.3570581	.0434527	8.22	0.000	.2718923	.4422239
x7	1.036464	.0881253	11.76	0.000	.8637414	1.209186
x8	.7948173	.0831455	9.56	0.000	.6318551	.9577795
x9	.087522	.1967188	0.44	0.656	-.2980397	.4730838
Cov[error]						
x7-x8	.352708	.0663003	5.32	0.000	.2227618	.4826543
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					

(Continued on next page)


```

Goodness of fit test: LR = 53.272      ; Prob[chi2(23) > LR] = 0.0003
Test vs independence: LR = 865.579    ; Prob[chi2(13) > LR] = 0.0000

```

Since scaling of the model is different, the previous estimates might be of limited value, hence the initial values are specified as `from(ones)`. The option `ivreg` is not applicable to this situation, either. 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 p. 22).

Missing data

By default, `cfa` 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 (FIML method for missing data in structural equation modeling) is provided with `missing` option. When this option is specified, the following modifications are taken:

1. The sample is restricted to the observations identified by the `[if] [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, since they rely on the estimate of the unstructured covariance matrix that 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 since this many sub-matrices 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 `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 `missing` option is recommended

for most uses. Carrying over the starting values from simpler analysis will speed up convergence, as usual. The author's experience suggests that the likelihoods with missing data tend to have multiple local maxima, and thus are more sensitive to starting values.

Let us introduce some missing data in Holzinger-Swineford example, and analyze the resulting data set.

```
. set seed 123456
.
. forvalues k=1/9 {
2.
.   gen y`k` = cond( uniform()<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, `cfa` will perform listwise deletion:

```
. cfa (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]	
<i>(output omitted)</i>						
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	.0787463	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	.8740226	.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	.3694431	.1210115	3.05	0.002	.1322648	.6066213
text-math	.1909221	.0617196	3.09	0.002	.0699539	.3118904
vis-math	.2244777	.068616	3.27	0.001	.0899927	.3589626

(Continued on next page)

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	.9594716	
y4	.4184883	.063913	6.55	0.000	.2932212	.5437554	
y5	.4209509	.0772258	5.45	0.000	.269591	.5723108	
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	.5367542	.1186252	4.52	0.000	.3042531	.7692552	

(output omitted)

Goodness of fit test: LR = 61.405 ; Prob[chi2(24) > LR] = 0.0000
 Test vs independence: LR = 503.076 ; Prob[chi2(12) > LR] = 0.0000

A more sophisticated analysis is available with missing option:

```
. cfa (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 = -3567.2372
rescale:      log likelihood = -3567.2372
rescale eq:   log likelihood = -3488.2637
Iteration 0:  log likelihood = -3488.2637 (not concave)
```

(output omitted)

```
Iteration 5:  log likelihood = -3445.5049
```

```
log likelihood = -3445.5049
```

Number of obs = 301

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
Means						
y1	4.770861	.0627675	76.01	0.000	4.647839	4.893883
y2	6.067073	.0603871	100.47	0.000	5.948716	6.185429
y3	2.027213	.068062	29.78	0.000	1.893814	2.160612
y4	3.013495	.0616438	48.89	0.000	2.892675	3.134315
y5	4.409553	.0723191	60.97	0.000	4.267811	4.551296
y6	2.113805	.057752	36.60	0.000	2.000613	2.226997
y7	3.993575	.0669016	59.69	0.000	3.862451	4.1247
y8	5.362717	.0593293	90.39	0.000	5.246433	5.479
y9	5.24863	.0600502	87.40	0.000	5.130934	5.366326
Loadings						
vis						
y1	1
y2	.3187452	.0793454	4.02	0.000	.1632311	.4742594
y3	.7523965	.1001313	7.51	0.000	.5561428	.9486502
text						
y4	1
y5	1.244166	.0833969	14.92	0.000	1.080711	1.407621
y6	.9289148	.068114	13.64	0.000	.7954139	1.062416
math						
y7	1
y8	1.02023	.1210663	8.43	0.000	.7829446	1.257516
y9	.9108521	.139011	6.55	0.000	.6383956	1.183309

(Continued on next page)

Factor cov.						
vis-vis	.9622819	.146954	6.55	0.000	.6742573	1.250306
text-text	.7251942	.0926325	7.83	0.000	.543638	.9067505
vis-text	.3908656	.065003	6.01	0.000	.263462	.5182692
math-math	.5236953	.1075686	4.87	0.000	.3128647	.7345258
text-math	.2572699	.0523718	4.91	0.000	.1546231	.3599166
vis-math	.3168506	.0603864	5.25	0.000	.1984955	.4352057
Var[error]						
y1	.2189937	.1139041	1.92	0.055	-.0042542	.4422415
y2	.9931048	.0844253	11.76	0.000	.8276342	1.158575
y3	.8245169	.0916871	8.99	0.000	.6448136	1.00422
y4	.384733	.047143	8.16	0.000	.2923345	.4771315
y5	.4022962	.0625891	6.43	0.000	.2796238	.5249685
y6	.3553242	.0430136	8.26	0.000	.271019	.4396294
y7	.7668271	.0908251	8.44	0.000	.5888131	.9448411
y8	.4471766	.0773266	5.78	0.000	.2956192	.598734
y9	.5629666	.0795455	7.08	0.000	.4070602	.718873

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 `missing` option) are smaller than the variance-covariance matrix. 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, elapsed maximization time. None of the 20 resulting maxima 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 quite possible that the global maximum of the procedure was not found, and it is unclear which of the local maxima would correspond to consistent estimates.

Table 1: Multiple maxima in missing data problems.

Starting values	technique(nr)		technique(dfp)	
	difficult: off	difficult: on	difficult: off	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

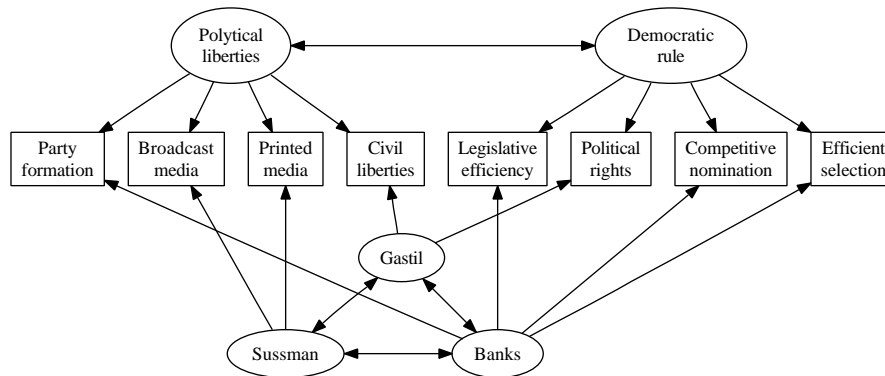


Figure 2: Structure of MTMM model of Bollen (1993).

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 (indicators developed by three liberal democracy researchers, A. Banks, R. D. Gastil and L. R. Sussman, see references in Bollen (1993)⁵). Political liberties are measured by freedom of group opposition and party formation, freedom of the broadcast media, 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 the fact that variables 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 et al. 1992; Byrne and Goffin 1993; Grayson and Marsh 1994) due to highly structured covariance matrices, this particular model does not load every method to every factor, and has been shown by Bollen (1993) to be identified. The structure of the model represented on Fig. 2. The individual error terms are omitted to reduce the clutter.

5. The complete data set, codebooks and data description is available at <http://www.icpsr.umich.edu/cocoon/ICPSR/STUDY/02532.xml>.

Building up a complex CFA model

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

```
. * base model
. cfa (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 = -3232.2597 (not concave)
(output omitted)
Iteration 15: log likelihood = -2672.5848
Iteration 16: log likelihood = -2672.5848
(output omitted)
. mat b_t = e(b)
. preserve
. * replace the variables by their residuals
. predict f1 f2, bartlett
. qui foreach x of varlist party80 broad80 print80 civlb80 {
2.   replace `x' = `x' - [lambda_`x'_pollib]_cons*f1
3. }
. qui foreach x of varlist leg80 polrt80 compet80 effec80 {
2.   replace `x' = `x' - [lambda_`x'_demrul]_cons*f2
3. }
.
. cfa (sussman: broad print) (gastil: civlb polrt) (banks: leg80 party compet
> effec) , difficult from(smart) usenames iter(20)

initial:      log likelihood = -2072.5109
rescale:      log likelihood = -2072.5109
rescale eq:   log likelihood = -1944.4419
Iteration 0:  log likelihood = -1944.4419 (not concave)
Iteration 1:  log likelihood = -1888.308 (not concave)
(output omitted)
Iteration 19: log likelihood = -1445.3238 (not concave)
Iteration 20: log likelihood = -1444.0299 (not concave)
convergence not achieved
(output omitted)
. mat b_res = e(b)
.
. restore
```

Next, let us fit the full model. First, we define the constraints specifying that the traits and methods are independent. 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 particular 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 right order in the 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.

```
. constr def 201 [phi_pollib_sussman]_cons = 0
. constr def 202 [phi_pollib_gastil]_cons = 0
. constr def 203 [phi_pollib_banks]_cons = 0
. constr def 204 [phi_demrul_sussman]_cons = 0
. constr def 205 [phi_demrul_gastil]_cons = 0
. constr def 206 [phi_demrul_banks]_cons = 0
. * initial values: combine the previous results
. mat bb2 = (b_t[1,1..19], b_res[1,9..30] )
. cfa (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)
Iteration 0: log likelihood = -2669.0535 (not concave)
(output omitted)
Iteration 13: log likelihood = -2568.1962
Log likelihood = -2568.1962                                Number of obs = 153
```

	Satorra-Bentler					[95% Conf. Interval]
	Coef.	Std. Err.	z	P> z		
Means						
party80	3.616557	.3443859	10.50	0.000	2.941573	4.291541
broad80	3.398693	.3384966	10.04	0.000	2.735252	4.062134
print80	4.575163	.3517763	13.01	0.000	3.885694	5.264632
civlb80	4.422659	.2597262	17.03	0.000	3.913605	4.931713
leg80	4.934636	.2885878	17.10	0.000	4.369014	5.500258
polrt80	4.379082	.2918017	15.01	0.000	3.807161	4.951003
compet80	6.24183	.3005655	20.77	0.000	5.652732	6.830927
effec80	4.575163	.2921235	15.66	0.000	4.002612	5.147715
Loadings						
pollib						
party80	1
broad80	.8605317	.0653975	13.16	0.000	.7323548	.9887085
print80	.9250436	.0579332	15.97	0.000	.8114966	1.038591
civlb80	.7187981	.0433978	16.56	0.000	.6337399	.8038562
demrul						
leg80	1
polrt80	1.078052	.0659157	16.36	0.000	.9488599	1.207245
compet80	.9393669	.0597397	15.72	0.000	.8222792	1.056454
effec80	.4380023	.0780422	5.61	0.000	.2850425	.5909621

(Continued on next page)

Confirmatory factor analysis

sussman							
broad80	1
print80	1.191157	.2313826	5.15	0.000	.7376553	1.644658	
gastil							
civlb80	1
polrt80	.6327745	.1780275	3.55	0.000	.2838471	.9817019	
banks							
party80	-.183531	.6226467	-0.29	0.768	-1.403896	1.036834	
leg80	1
compet80	2.71092	.7440816	3.64	0.000	1.252546	4.169293	
effec80	1.936524	.6181741	3.13	0.002	.7249251	3.148123	
Factor cov.							
pollib-pol-b	16.0291	1.382937	11.59	0.000	13.3186	18.73961	
demrul-dem-l	10.48467	1.120173	9.36	0.000	8.289173	12.68017	
pollib-dem-l	12.85863	1.113011	11.55	0.000	10.67717	15.04009	
sussman-su-n	2.568781	1.111172	2.31	0.021	.3909238	4.746638	
demrul-sus-n	0
pollib-sus-n	0
gastil-gas-l	1.432466	.4740364	3.02	0.003	.5033713	2.36156	
sussman-ga-l	1.472027	.6605919	2.23	0.026	.1772902	2.766763	
demrul-gas-l	0
pollib-gas-l	0
banks-banks	.6788297	.4804152	1.41	0.158	-.2627668	1.620426	
gastil-banks	-.342767	.2509392	-1.37	0.172	-.8345989	.1490649	
sussman-ba-s	-.2801573	.3093466	-0.91	0.365	-.8864654	.3261508	
demrul-banks	0
pollib-banks	0
Var[error]							
party80	2.094077	.8899958	2.35	0.019	.3497171	3.838436	
broad80	3.092158	.4884583	6.33	0.000	2.134798	4.049519	
print80	1.572301	.5140154	3.06	0.002	.5648489	2.579752	
civlb80	.6067903	.1927146	3.15	0.002	.2290766	.984504	
leg80	1.578789	.2679755	5.89	0.000	1.053567	2.104012	
polrt80	.2688616	.3682683	0.73	0.465	-.4529311	.9906542	
compet80	-.4185984	.8945167	-0.47	0.640	-2.171819	1.334622	
effec80	8.499291	1.068135	7.96	0.000	6.405785	10.5928	
R2							
party80	0.8787						
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(20) > 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 option `difficult` 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 maxima as in Bollen (1993).

A mild *Heywood case* was produced for `compet80` variable: the reported estimated error variance is negative, and the corresponding R^2 is greater than 1. However the confidence interval 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 Banks factor (i.e., the fact that this variable came from Banks' data set). 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 data set since many variables are ordinal with a small number of categories (3 to 5).

From the substantive perspective, it might be interesting to note that the variance of "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 other two researchers, Gastil and Sussman, as they do seem to contain non-trivial amount of common influences. It might be puzzling however that the loadings from "Banks" factor to its observed variables `compet80` and `effec80` are well identified.

5 Technical notes

5.1 Methods and formulas

`cfa` estimates the model (2) by maximum likelihood. The observed variables \mathbf{y}_i 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

$$l_i = -\frac{p}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} (\mathbf{y}_i - \mu)' \Sigma^{-1} (\mathbf{y}_i - \mu) \quad (13)$$

where $\Sigma = \Sigma(\theta) = \Lambda\Phi\Lambda' + \Theta$ is $p \times p$ matrix, and the parameters θ of the model are the means μ , the free elements of Λ , non-redundant elements of Φ , and the free elements of Θ . The latter are usually the diagonal elements only, but if option `correlated()` is specified, off-diagonal elements can be estimated, as well. Since 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 non-redundant moments of the covariance matrix $p^* = p(p+1)/2$.

When some components of \mathbf{y}_i vector are missing, and option `missing` is specified, the vector of means μ and the parametric covariance matrix Σ are restricted to the non-missing components in computation of the likelihood (13).

The conventional standard errors are available as the inverse of the observed information matrix (`vce(oim)` method). Other analytic estimators (`vce(opg)`, `vce(robust)`, `vce(cluster ...)`) are supported, but resampling estimators need to be specified explicitly via `bootstrap ... : cfa ...` or `jackknife ... : cfa ...`. See [R] `vce_option`, [R] `bootstrap`, [R] `jackknife`.

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

$$R_j^2 = \frac{s_j^2 - \sigma_{jj}(\hat{\theta})}{s_j^2}$$

where s_j^2 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:

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

It has a likelihood ratio test statistic

$$T_u = -2 \left[l(\hat{\theta}) - \left(-\frac{pN}{2} \ln 2\pi - \frac{N}{2} \ln |S| - \frac{pN}{2} \right) \right] \quad (14)$$

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

The second likelihood ratio test is the test against an “independence” model

$$H_0 : \Sigma = \Sigma_0 = \text{diag}(\sigma_1^2, \dots, \sigma_p^2) \quad \text{vs.} \quad H_1 : \Sigma = \Sigma(\theta)$$

It has a likelihood ratio test statistic

$$T_i = -2 \left[\left(-\frac{pN}{2} \ln 2\pi - \frac{N}{2} \ln |S_0| - \frac{N}{2} \text{tr} S_0 \right) - l(\hat{\theta}) \right] \quad (15)$$

where $S_0 = \text{diag}(s_1^2, \dots, s_p^2)$, and subindex i stands for “independent”. The test statistic has an asymptotic χ^2 distribution with degrees of freedom $\text{df}_i = t - p$.

Post-estimation command `estat fitindices` computes and reports a number of fit indices that are used to complement the general χ^2 goodness of fit test.

Comparative fit index (Bentler 1990b) is

$$\text{CFI} = 1 - \frac{\max(T_u, 0)}{\max(T_u - \text{df}_u, T_i - \text{df}_i, 0)} \quad (16)$$

Tucker-Lewis non-normed fit index (Tucker and Lewis 1973) is

$$\text{TLI} = \left(\frac{T_i}{\text{df}_i} - \frac{T_u}{\text{df}_u} \right) / \left(\frac{T_i}{\text{df}_i} - 1 \right) \quad (17)$$

Root mean square residual (Jöreskog and Sörbom 1986) is

$$\text{RMSR} = \left[\frac{1}{p^*} \sum_{1 \leq i \leq j \leq p} (s_{ij} - \hat{\sigma}_{ij})^2 \right]^{1/2} \quad (18)$$

Root mean squared error of approximation (Steiger 1990; Browne and Cudeck 1993) is

$$\hat{\epsilon}_a = \sqrt{\max \left[\frac{T_u}{(N-1)\text{df}_u}, 0 \right]} \quad (19)$$

Let $G(x; \lambda, d)$ be the cdf of the non-central χ^2 with non-centrality parameter λ and d degrees of freedom. If $G(T_u|0, d) \geq 0.95$, find $\hat{\lambda}_L$ as solution of

$$G(T_u; \hat{\lambda}_L, \text{df}_u) = 0.95.$$

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

$$G(T_u; \hat{\lambda}_U, \text{df}_u) = 0.05.$$

Otherwise, set $\hat{\lambda}_U = 0$. Finally, set 90% confidence interval for RMSEA as

$$\left(\sqrt{\frac{\hat{\lambda}_L}{(N-1)\text{df}_u}}, \sqrt{\frac{\hat{\lambda}_U}{(N-1)\text{df}_u}} \right) \quad (20)$$

If sandwich standard errors are requested, the data are implicitly assumed non-i.i.d. (or violating the model assumptions otherwise), no test statistics or R^2 are reported, and no fit indices are produced by `estat fitindices`.

An additional variance estimator (Satorra and Bentler 1994) is available with `vce(sbentler)` non-standard option. Let $\mathbf{s} = \text{vech } S$, $\sigma = \text{vech } \Sigma$, 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:

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

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

$$\hat{\Gamma}_N = \frac{1}{N-1} \sum_i (\mathbf{b}_i - \bar{\mathbf{b}})(\mathbf{b}_i - \bar{\mathbf{b}})' \quad (21)$$

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

$$V_N = \frac{1}{2}D'(\Sigma \otimes \Sigma)D \quad (22)$$

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

$$\hat{\Delta} = \left. \frac{\partial \sigma}{\partial \theta} \right|_{\theta=\hat{\theta}} \quad (23)$$

Then Satorra-Bentler estimator is

$$\widehat{\text{acov}}(\hat{\theta}) = (N - 1)^{-1}(\hat{\Delta}'V_N\hat{\Delta})^{-1}\hat{\Delta}'V_N\Gamma_NV_N\hat{\Delta}(\hat{\Delta}'V_N\hat{\Delta})^{-1} \quad (24)$$

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

$$T_u \xrightarrow{d} \sum_{j=1}^{\text{df}_u} \alpha_j X_j, \quad X_j \sim \text{i.i.d.} \chi_1^2$$

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

$$U = V - V\Delta(\Delta'V\Delta)^{-1}\Delta'V \quad (25)$$

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

$$T_{sc} = \frac{T}{\hat{c}}, \quad \hat{c} = \frac{1}{\text{df}_u} \text{tr}[\hat{U}\hat{\Gamma}_N] \quad (26)$$

referred to $\chi_{\text{df}_u}^2$, where \hat{U} is U evaluated at θ , and adjusted statistic

$$T_{adj} = \frac{\hat{d}}{\hat{c}}T, \quad \hat{d} = \frac{(\text{tr}[\hat{U}\hat{\Gamma}_N])^2}{\text{tr}[(\hat{U}\hat{\Omega}_N)^2]} \quad (27)$$

referred to $\chi_{\hat{d}}^2$ where the degrees of freedom \hat{d} might be a non-integer number.

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

$$T_2 = T/(1 + T/N) \quad (28)$$

referred to χ^2 with df_u degrees of freedom.

5.2 Implementation details

The confirmatory factor analysis `cfa` package consists of the following ado-files: `cfa` (the main estimation engine), `cfa_estat` (post-estimation commands), `cfa_lfm` (likelihood evaluator), `cfa_p` (prediction), and `bollenstine` (Bollen-Stine bootstrap). The Mata functions for `cfa` are available in `lcfa.mlib` library. The likelihood maximization is

implemented through `ml lf` mechanism (observation-by-observation likelihoods with numerical derivatives). There are approximately 43k of ado code (about 1400 lines) and 13k of Mata code (about 450 lines).

The ado code uses `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 SSC.

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

5.3 Parameter names and returned values

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 $j = 1, \dots, p$ 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 `usenames` option is specified, all the variable and factor indices are replaced with their names in the data set and factor specifications.

Thus, for instance, the model

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

will have parameters `lambda_1_1`, `lambda_2_1`, `lambda_3_1`, `lambda_4_1`, `phi_1_1`, `theta_1`, `theta_2`, `theta_3`, `theta_4` with default settings, and parameters `lambda_x1_f`, `lambda_x2_f`, `lambda_x3_f`, `lambda_x4_f`, `phi_f_f`, `theta_x1`, `theta_x2`, `theta_x3`, `theta_x4` when option `usenames` is specified. Specifying this 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 in comparing models using `estimates table` command, or transferring starting values between commands, as shown in one of the examples above.

The returned values include the standard outcomes from `ml`, such as `e(N)`, `e(11)`, etc. Additional returned values are as follows.

(Continued on next page)

Scalars

<code>e(pstar)</code>	Total degrees of freedom
<code>e(df_m)</code>	Model degrees of freedom
<code>e(df_u)</code>	Residual degrees of freedom
<code>e(ll_0)</code>	Log-likelihood of the unrestricted model $\hat{\Sigma} = S$
<code>e(ll)</code>	Log-likelihood at the maximum
<code>e(ll_indep)</code>	Log-likelihood of “independence” model
<code>e(lr_u)</code>	Likelihood ratio statistic against unrestricted model; same as <code>e(chi2)</code>
<code>e(p_u)</code>	p -value against unrestricted model; same as <code>e(p)</code>
<code>e(lr_indep)</code>	Likelihood ratio against “independence” model
<code>e(df_indep)</code>	Model degrees of freedom of “independence” model
<code>e(p_indep)</code>	p -value against “independence” model

Macros

<code>e(factors)</code>	List of factors
<code>e(observed)</code>	List of observed variables
<code>e(factork)</code>	Unabbreviated factor statements, $k = 1, \dots, m$
<code>e(correlated)</code>	Unabbreviated correlated errors statements
<code>e(unitvar)</code>	The list of factors identified by unit variances
<code>e(missing)</code>	Indicates that <code>missing</code> option was specified

Matrices

<code>e(S)</code>	Sample covariance	<code>e(Sigma)</code>	Implied covariance
<code>e(Lambda)</code>	Estimated loadings $\hat{\Lambda}$	<code>e(Theta)</code>	Estimated error variances $\hat{\Theta}$
<code>e(Phi)</code>	Estimated factor covariances $\hat{\Phi}$	<code>e(CFA_Struc)</code>	Model structure description

Additional returned values posted when `vce(sbentler)` option is used are:

Scalars

<code>e(SBc)</code>	scaling correction \hat{c} in (26)	<code>e(Tsc)</code>	scaled statistic T_{sc} in (26)
<code>e(SBd)</code>	scaling correction \hat{d} in (27)	<code>e(p_Tsc)</code>	p -value associated with T_{sc}
<code>e(T2)</code>	T_2 statistic in (28)	<code>e(Tadj)</code>	adjusted statistic T_{adj} in (27)
<code>e(p_T2)</code>	p -value associated with T_2	<code>e(p_Tsc)</code>	p -value associated with T_{adj}

Matrices

<code>e(SBU)</code>	matrix U in (25)	<code>e(SBDelta)</code>	matrix $\hat{\Delta}$ in (23)
<code>e(SBV)</code>	matrix V in (22)	<code>e(SBGamma)</code>	matrix $\hat{\Gamma}_n$ in (21)

Additional returned values posted by `bollenstine` are:

Scalars

<code>e(B_BS)</code>	number of replications	<code>e(T_BS_05)</code>	5th bootstrap percentile
<code>e(p_u_BS)</code>	bootstrap p -value	<code>e(T_BS_95)</code>	95th bootstrap percentile

Values returned by `estat fit` are:

Scalars

<code>r(AIC)</code>	AIC	<code>r(BIC)</code>	BIC
<code>r(CFI)</code>	Comparative fit index CFI (16)	<code>r(RMSEA)</code>	Root mean squared error of approximation (19)
<code>r(TLI)</code>	Tucker-Lewis incremental fit index (17)	<code>r(RMSEA05)</code>	5% lower limit for RMSEA
<code>r(RMSR)</code>	Root mean squared residual (18)	<code>r(RMSEA95)</code>	95% upper limit for RMSEA

5.4 Computational complexity

A small simulation was conducted to establish the computational complexity of `cfa`, 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 1000, the number of factors, from 1 to 5, and the number of indicators per factor, 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 with dependent variable equal to log of elapsed time, and explanatory variables, the logs of the quantities in the first column. The dependence on the sample size is of the order $O(n^{0.68})$ (note that the sample size is orthogonal to the size and 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, as 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 Newton-Raphson optimization method. Multidimensional nature of the optimization problem is likely to be contributing to the increase in computational time, although the increase in time due to extra iterations needed for convergence vs. increase due to matrix processing times was not studied in this simulation.

5.5 Distribution

The module is maintained and updated by the author, Stanislav Kolenikov. To check for the recent updates, point your Stata to

```
. 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.

Acknowledgements

Support from NSF grant SES-0617193 with funds from Social Security Administration is gratefully acknowledged. The author would also like to thank Ken Higbee for helpful comments on improving the code stability, and 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 the author's responsibility.

6 References

- Anderson, T. W., and Y. Amemiya. 1988. The Asymptotic Normal Distribution of Estimators in Factor Analysis under General Conditions. *Annals of Statistics* 16(2): 759–771.
- Bartholomew, D. J., and M. Knott. 1999. *Latent Variable Models and Factor Analysis*, vol. 7 of *Kendall's Library of Statistics*. 2nd ed. London: Arnold Publishers.
- Bentler, P. M. 1990a. Comparative fit indexes in structural models. *Psychological Bulletin* 107(2): 238–246.
- . 1990b. Fit Indexes, Lagrange Multipliers, Constraint Changes and Incomplete Data in Structural Models. *Multivariate Behavioral Research* 25(2): 163–172.
- Beran, R., and M. S. Srivastava. 1985. Bootstrap Tests and Confidence Regions for Functions of a Covariance Matrix. *The Annals of Statistics* 13(1): 95–115.
- Bollen, K. 1993. Liberal Democracy: Validity and Method Factors in Cross-National Measures. *American Journal of Political Science* 37(4): 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 Models. *Psychometrika* 61(1): 109–121.
- Brown, T. A. 2006. *Confirmatory Factor Analysis for Applied Research (Methodology In The Social Sciences)*. The Guilford Press.
- Browne, M. W. 1984. Asymptotically distribution-free methods for the analysis of the 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, chap. 6, 136–162. Newbury Park, CA, USA: SAGE Publishers.
- 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(1): 67–96.
- Cox, N. J. 2008. Speaking Stata: Correlation with confidence, or Fisher’s z revisited. *Stata Journal* 8(3): 413–439.
- Gould, W., J. Pittblado, 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(1): 121–134. <http://dx.doi.org/10.1007/BF02294271>.
- Holzinger, K. J., and F. Swineford. 1939. A study in factor analysis: The stability of a bi-factor solution. Technical Report 48, University of Chicago, Chicago, IL.
- Huber, P. 1967. The behavior of the maximum likelihood estimates under nonstandard conditions. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 221–233. Berkeley: University of California Press.
- Jöreskog, K. 1969. A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika* 34(2): 183–202.
- Jöreskog, K., and A. Goldberger. 1972. Factor analysis by generalized least squares. *Psychometrika* 37(3): 243–260.
- Jöreskog, K., and D. Sörbom. 1986. *LISREL VI: Analysis of Linear Structural Relationships by Maximum Likelihood and Least Squares Methods*. Scientific Software, Inc, Mooresville, IN.
- Lawley, D. N., and A. E. Maxwell. 1971. *Factor Analysis as a Statistical Method*. Butterworths Mathematical Texts, Butterworths.
- Little, R. J. A., and D. B. Rubin. 2002. *Statistical Analysis with Missing Data*. 2nd ed. Wiley Series in Probability and Statistics, New York: Wiley.
- Magnus, J. R., and H. Neudecker. 1999. *Matrix calculus with applications in statistics and econometrics*. 2nd ed. John Wiley & Sons.
- 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 Techniques*, ed. G. Marcoulides and R. Schumaker, 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(4): 489–507.
- Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata Journal* 2(1): 1–21.
- . 2004. Generalized multilevel structural equation modeling. *Psychometrika* 69(2): 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. Bentler. 2001. A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika* 66(4): 507–514.
- 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. Thousands Oaks, CA: Sage.
- Steiger, J. H. 1990. Structural Model Evaluation and Modification: An Interval Estimation Approach. *Multivariate Behavioral Research* 25(2): 173–180.
- Tucker, L., and C. Lewis. 1973. A reliability coefficient for maximum likelihood factor analysis. *Psychometrika* 38(1): 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(438): 767–774.
- Yuan, K.-H., and P. M. Bentler. 2007. Structural Equation Modeling. In *Handbook of Statistics: Psychometrics*, ed. C. Rao and S. Sinharay, vol. 26 of *Handbook of Statistics*, chap. 10. Elsevier.

About the author

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