# The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the web page

http://www.stata-journal.com

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index®
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch®)



The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata and Mata are registered trademarks of StataCorp LP.

**Subscriptions** are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

**Subscription rates**

The listed subscription rates include both a printed and an electronic copy unless otherwise mentioned.

Subscriptions mailed to U.S. and Canadian addresses:

| | |
|---|---|
| 3-year subscription | $195 |
| 2-year subscription | $135 |
| 1-year subscription | $ 69 |
| 1-year student subscription | $ 42 |
| 1-year university library subscription | $ 89 |
| 1-year institutional subscription | $195 |

Subscriptions mailed to other countries:

| | |
|---|---|
| 3-year subscription | $285 |
| 2-year subscription | $195 |
| 1-year subscription | $ 99 |
| 3-year subscription (electronic only) | $185 |
| 1-year student subscription | $ 69 |
| 1-year university library subscription | $119 |
| 1-year institutional subscription | $225 |

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

Volume 9 Number 3 2009

# The Stata Journal

**Articles and Columns** 329

Confirmatory factor analysis using confa . . . . . S. Kolenikov 329
Graphical representation of multivariate data using Chernoff faces . . . . . R. Raciborski 374
Improved degrees of freedom for multivariate significance tests obtained from multiply imputed, small-sample data . . . . . Y. V. Marchenko and J. P. Reiter 388
Implementing weak-instrument robust tests for a general class of instrumental-variables models . . . . . K. Finlay and L. M. Magnusson 398
A seasonal unit-root test with Stata . . . . . D. Depalo 422
Robust regression in Stata . . . . . V. Verardi and C. Croux 439
Nonparametric testing of distributions—the Epps–Singleton two-sample test using the empirical characteristic function . . . . . S. J. Goerg and J. Kaiser 454
Multiple imputation of missing values: Further update of ice, with an emphasis on categorical variables . . . . . P. Royston 466
Speaking Stata: Creating and varying box plots . . . . . N. J. Cox 478

**Notes and Comments** 497

Stata tip 77: (Re)using macros in multiple do-files . . . . . J. Herrin 497
Stata tip 78: Going gray gracefully: Highlighting subsets and downplaying substrates . . . . . N. J. Cox 499
Stata tip 79: Optional arguments to options . . . . . N. J. Cox 504

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

# Confirmatory factor analysis using confa

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

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

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

## 1 Confirmatory factor analysis (CFA)

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

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

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

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

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

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

## 1.1 The model and identification

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

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

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

where $\mu_j$ is the intercept; $\lambda_{jk}$ are regression coefficients, or factor loadings; and $\delta_j$ are measurement errors, or unique errors. In matrix form, (1) can be written as

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

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

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

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

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

where all parameters are put together into vector $\theta$.

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

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

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

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

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

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

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

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

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

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

## 1.2 Estimation, testing, and goodness of fit

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

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

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

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

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

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

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

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

cannot be rejected.

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

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

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

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

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

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

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

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

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

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

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

## 1.3 Factor scoring

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

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

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

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

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

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

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

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

# 2 Description of confa command

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

## 2.1 Syntax

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

The factor specification, *factorspec*, is

(*factorname*`:` *varlist*)

The correlated-errors specification, *corrspec*, is

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

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

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

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

estat aic

estat bic
```

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

`estat` `corr`elate [ , `level(`#`)` `bound` ]

`predict` [ *type* ] *newvarlist* [ *if* ] [ *in* ] [ , `reg`ression `empi`ricalbayes `eb`ayes `mle` `bart`lett ]

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

## 2.2 Options of confa

### Model

`correlated(`*corrspec* [ *corrspec* ... ]`)` specifies the correlated measurement errors $\delta_k$ and $\delta_j$ corresponding to the variables $y_k$ and $y_j$. Here *corrspec* is of the form

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

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

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

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

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

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

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

### Variance estimation

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

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

### Reporting

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

### Other

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

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

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

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

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

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

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

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

## 2.3 Descriptions and options of estat

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

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

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

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

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

`rmsr` requests the RMSR.

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

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

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

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

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

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

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

## 2.4 Description and options of predict

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

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

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

### 2.5 Options of bollenstine

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

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

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

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

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

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

## 3 Example 1: Simple structure CFA with psychometric data

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

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

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

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

```
Sorted by:
```

## ▷ Specification and starting values

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

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

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

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

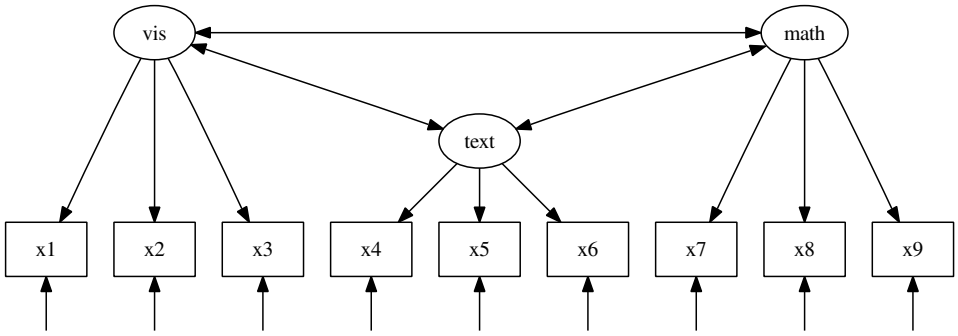



Figure 1. The basic model for Holzinger–Swineford data

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

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

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

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

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

The resulting estimates are identical for all three convergent runs:

log likelihood = -3737.7449 Number of obs = 301

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

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

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

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

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

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

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

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

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

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

◁

### ▷ Standard-error estimation

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

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

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

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

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

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

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

(*Continued on next page*)

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

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

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

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

◁

### ▷ Correlated errors

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

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

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

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

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

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

up and became the largest $R^2$ in the model. This is not surprising. The `math` factor is primarily based on covariances between the last three variables, and to a lesser extent, on covariances between the last three and the first six variables. The latter component is relatively unchanged between the two models. However, with the covariance between the error terms $\delta_7$ and $\delta_8$ freely estimated, the covariance between `x7` and `x8` no longer contributes to explaining this factor. The burden of identifying this factor shifts to covariances `x7`–`x9` and `x8`–`x9`. The `math` factor now has to contribute less to explaining covariances between `x7` and `x8`, and more to explaining covariance of `x9` with other variables. This produces the observed change in reliabilities.

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

◁

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

### ▷ Satorra–Bentler scaled difference test

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

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

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

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

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

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

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

◁

### ▷ Bollen–Stine bootstrap

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

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

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

(*Continued on next page*)

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

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

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

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

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

◁

### ▷ Postestimation commands: Fit indices and correlations

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

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

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

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

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

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

(*Continued on next page*)

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

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

◁

## ▷ Factor predictions

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

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

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

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

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

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

◁

### Alternative identification

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

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

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

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

```
Var[error]
          x1 |   .5758446   .1034751     5.57   0.000      .373037    .7786521
          x2 |     1.1225   .1019975    11.01   0.000     .9225884    1.322411
          x3 |   .8321164   .0898742     9.26   0.000     .6559663    1.008267
          x4 |   .3722483   .0479868     7.76   0.000     .2781958    .4663007
          x5 |   .4436603   .0580118     7.65   0.000     .3299593    .5573613
          x6 |    .357058   .0434527     8.22   0.000     .2718922    .4422239
          x7 |  1.036464   .0881257    11.76   0.000     .8637409    1.209187
          x8 |   .794817   .0831478     9.56   0.000     .6318503    .9577837
          x9 |   .0875252   .1967321     0.44   0.656    -.2980627    .4731131
Cov[error]
       x7-x8 |   .3527083   .0663016     5.32   0.000     .2227595    .4826571
-------------+------------------------------------------------------------------
R2
          x1 |      0.5742
          x2 |      0.1870
          x3 |      0.3461
          x4 |      0.7220
          x5 |      0.7303
          x6 |      0.6992
          x7 |      0.1236
          x8 |      0.2215
          x9 |      0.9107
------------------------------------------------------------------------------

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

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

◁

## ▷ Missing data

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

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

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

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

3. Factor predictions are not available.

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

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

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

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

By default, `confa` will perform listwise deletion:

(*Continued on next page*)

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

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

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

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

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

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

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

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

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

Table 1. Multiple maximums in missing-data problems

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

◁

# 4 Example 2: Modeling the structure of correlated measurement errors

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

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

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

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

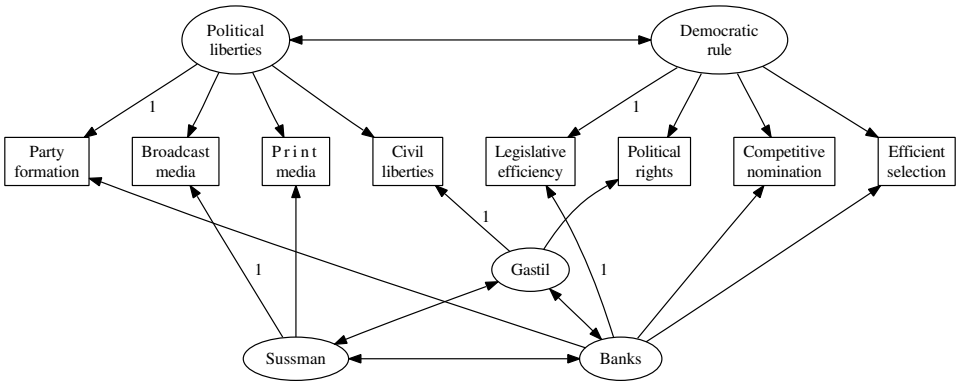



Figure 2. Structure of the MTMM model of Bollen (1993)

## ▷ Building up a complex CFA model

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

(*Continued on next page*)

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

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

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

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

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

Goodness of fit test: LR =  9.206      ; Prob[chi2( 8) > LR] = 0.3253
Test vs independence: LR = 1603.033    ; Prob[chi2(28) > LR] = 0.0000

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

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

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

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

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

◁

# 5 Technical notes

## 5.1 Methods and formulas

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

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

where

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

Hence,

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

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

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

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

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

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

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

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

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

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

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

It has a likelihood-ratio test statistic

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

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

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

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

It has a likelihood-ratio test statistic

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

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

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

CFI (Bentler 1990b) is

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

TLI (Tucker and Lewis 1973) is

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

RMSR (Jöreskog and Sörbom 1986) is

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

RMSEA (Steiger 1990; Browne and Cudeck 1993) is

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

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

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

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

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

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

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

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

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

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

The simplest estimator of $\Gamma$ is based on the fourth-order moments of data,

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

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

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

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

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

Then the Satorra–Bentler variance estimator is

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

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

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

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

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

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

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

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

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

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

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

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

which has an approximate $\chi^2$ distribution with $\text{df}_u$ degrees of freedom.

## 5.2 Implementation details

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

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

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

## 5.3 Parameter names and saved results

The nomenclature of the parameter names is as follows.

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

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

Thus, for instance, the model

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

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

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

(*Continued on next page*)

Scalars

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

Macros

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

Matrices

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

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

Scalars

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

Matrices

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

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

Scalars

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

Values returned by `estat fit` are the following:

Scalars

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

## 5.4 Computational complexity

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

Table 2. Computational complexity simulation results

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

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

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

## 5.5 Verification and certification

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

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

### 5.6 Distribution

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

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

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

## 6 Acknowledgments

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

## 7 References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**About the author**

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

The Stata Journal (2009)
**9**, Number 3, pp. 374–387

# Graphical representation of multivariate data using Chernoff faces

Rafal Raciborski[1]
Department of Political Science
Emory University
Atlanta, GA
rafal.raciborski@emory.edu

**Abstract.** Chernoff (1971, Technical Report 71, Department of Statistics, Stanford University; 1973, *Journal of the American Statistical Association* 68: 361–368) proposed the use of cartoon-like faces to represent points in $k$ dimensions. This article describes a Stata implementation of a face-generating algorithm using the method proposed by Flury (1980, Technical Report 3, Institute of Mathematical Statistics and Actuarial Science, Bern University), Schüpbach (1987, Technical Report 25, Institute of Mathematical Statistics and Acturial Science, Bern University), and Friendly (1991, http://www.math.yorku.ca/SCS/sasmac/faces.html). I present examples of applying Chernoff faces to data clustering and outlier detection.

**Keywords:** gr0038, chernoff, Chernoff faces, graphs

## 1 Introduction

Chernoff (1971, 1973) proposed a method of representing multivariate data as cartoon faces. The main use of face graphs was to enhance "the user's ability to detect and comprehend important phenomena" and to serve "as a mnemonic device for remembering major conclusions". Other advantages of faces include the ease of monitoring the sensitivity of variables to each other, fast identification of key differentiating dimensions, and the detection of longitudinal trends. Recent applications of Chernoff faces include tracking changes in laboratory data (Lott and Durbridge 1990), consumer perception of brand image (Golden and Sirdesai 1992), classification of forest tree clones (Camussi, Raddi, and Raddi 1992), portrayal of service quality data (Nel, Pitt, and Webb 1994), attitudes toward environmental protection policies (Apaiwongse 1995), and classification of drinking water samples (Astel et al. 2006).

However, the method of constructing faces as proposed by Chernoff suffers from certain disadvantages. First, to make all the faces of equal size, the width and length of each face need to be normalized, which almost obliterates the effect of the variables assigned to those two features (Chernoff 1971, 19). Second, extreme values of certain parameters compress the range of other parameters, which results in "artificial dependencies not originating from the data being represented" (Flury and Riedwyl 1981, 757). The

1. I would like to thank Taha Kass-Hout and an anonymous referee for helpful suggestions.

algorithm described in this article was developed by Flury (1980), Schüpbach (1987), and Friendly (1991), and avoids the above mentioned pitfalls.

The article proceeds as follows. Section 2 describes the `chernoff` command and its options. Section 3 presents examples of data clustering and outlier detection. Section 4 offers general advice on drawing and interpreting Chernoff faces. Section 5 concludes.

# 2 The chernoff command

## 2.1 Syntax

`chernoff` [ `,` `isize(`*exp*`)` `iangle(`*exp*`)` `ihor(`*exp*`)` `ivert(`*exp*`)` `psize(`*exp*`)`
`ppos(`*exp*`)` `bcurv(`*exp*`)` `bdens(`*exp*`)` `bhor(`*exp*`)` `bvert(`*exp*`)` `fline(`*exp*`)`
`hupper(`*exp*`)` `hlower(`*exp*`)` `hdark(`*exp*`)` `hslant(`*exp*`)` `nose(`*exp*`)` `msize(`*exp*`)`
`mcurv(`*exp*`)` `gmin(`#`)` `gmax(`#`)` [ `lhalf` | `rhalf` ] `hspace(`#`)` `ititle(`*varname*`)`
`inote(`*varname*`)` `ilabel(`*varname*`)` `lsize(`*textsizestyle*`)` `xlabel(`#`)` `ylabel(`#`)`
`placement(`*clockdirstyle*`)` `justification(`*jstyle*`)` `iscale(`*varname*`)`
`imargin(`*marginstyle*`)` `iregion(`*marginstyle*`)` `xface(`#`)` `yface(`#`)` `show`
`saveall` `rescale(`#`)` `legend(`{`2`|`3`} [ `nolabel` ]`)` `order(`*varlist*`)` `rows(`#`)`
`cols(`#`)` `colfirst` `xcombined(`#`)` `ycombined(`#`)` `nocombine` `title(`*tinfo*`)`
`subtitle(`*tinfo*`)` `note(`*tinfo*`)` `nodraw` `saving(`*filename*`, replace)` `timer` ]

## 2.2 Options

### Face feature

`isize(`*exp*`)` through `mcurv(`*exp*`)` represent face features and are described in table 1, below. *exp* is specified as

*varname* | `.` | `_null_` [ `,` [ # | `.` ] [ # | `.` ] ]

*varname* is linearly rescaled to a $(0, 1)$ interval before being plotted. If a given face feature is not specified or is specified as missing, it assumes the default value of 0.50. Missing values in *varname* are also assigned the value of 0.50. `_null_` prevents a "logical" set of features from being drawn. For example, specifying any of the eye features as `_null_` will prevent the eyes from being plotted.

The optional part represents a theoretical minimum and maximum for *varname*. For example, if the variable `gpa` in your data ranges from 1.7 to 3.8 but you want to plot it relative to the possible range $(1.0, 4.0)$, you can do so by specifying *facefeature*`(gpa,1 4)`. If you want to specify min only, type *facefeature*`(gpa,1 .)` or *facefeature*`(gpa,1)`. If you want to specify max only, type *facefeature*`(gpa,. 4)`.

Table 1. Explanation of face features

| option | face feature | _null_ group |
|---|---|---|
| `isize(`*exp*`)` | eye size | _null_ |
| `iangle(`*exp*`)` | eye angle | |
| `ihor(`*exp*`)` | eye horizontal position | |
| `ivert(`*exp*`)` | eye vertical position | |
| `psize(`*exp*`)` | pupil size | _null_ |
| `ppos(`*exp*`)` | pupil position | |
| `bcurv(`*exp*`)` | brow curvature | _null_ |
| `bdens(`*exp*`)` | brow density | |
| `bhor(`*exp*`)` | brow horizontal position | |
| `bvert(`*exp*`)` | brow vertical position | |
| `fline(`*exp*`)` | face line | _null_ |
| `hupper(`*exp*`)` | hair upper line | _null_ |
| `hlower(`*exp*`)` | hair lower line | _null_ |
| `hdark(`*exp*`)` | hair darkness | _null_ |
| `hslant(`*exp*`)` | hair shading slant | |
| `nose(`*exp*`)` | nose line | _null_ |
| `msize(`*exp*`)` | mouth size | _null_ |
| `mcurv(`*exp*`)` | mouth curvature | |

### Individual face graph

`gmin(`#`)` and `gmax(`#`)` stand for a global minimum and maximum. These options override the "local" minimums and maximums specified within face features. See section 3.2 for more details.

`lhalf` | `rhalf` tells Stata to draw only the left side or the right side of the face, respectively. Only one option may be specified.

`hspace(`#`)` controls the column spacing between half-face graphs. The range of `hspace()` is $(0.50, 1)$, with the default being `hspace(0.75)`. Specifying a lower value will bring the faces closer, and specifying a higher value will spread them apart.

`ititle(`*varname*`)` specifies titles for individual face graphs. The variable may be string or numeric.

`inote(`*varname*`)` specifies notes for individual face graphs. The variable may be string or numeric.

`ilabel(`*varname*`)` tells Stata to label faces with the values of *varname*. The variable may be string or numeric.

`lsize(`*textsizestyle*`)` specifies the size of the label. The default is `lsize(large)`. See [G] ***textsizestyle*** for a list of available styles.

`xlabel(#)` and `ylabel(#)` denote the $(x, y)$ coordinates of the face label. The default position is $(10, -10)$ for full faces, $(-8, -10)$ for left faces, and $(8, -10)$ for right faces.

`placement(`*clockdirstyle*`)` specifies the position of the face label relative to (`xlabel()`, `ylabel()`) coordinates. Possible values are 0, 1, 2, ..., 12. The default value is `placement(9)`, or `placement(3)` if `lhalf` is specified.

`justification(`*jstyle*`)` specifies text justification of the face label. Possible values are `left`, `right`, and `center`. The default value is `justification(left)`, or `justification(right)` if `lhalf` is specified.

`iscale(`*varname*`)` specifies a multiplier that affects the size of `ititle()`, `inote()`, and `ilabel()`. A *varname* is required, which allows for different scales for different faces.

`imargin(`*marginstyle*`)` specifies the margins between the plot area and the outside area of a face graph. This option is equivalent to `graphregion(margin(`*marginstyle*`))`; see [G] ***marginstyle***. The default margin is `imargin(zero)`.

`iregion(`*marginstyle*`)` specifies the axes offset from the contents of the plot. This option is equivalent to `plotregion(margin(`*marginstyle*`))`; see [G] ***region_options***. The default margin is `iregion(medsmall)`.

`xface(#)` and `yface(#)` denote the size of the individual face graph. The default size is `xface(5.00)` and `yface(6.00)`.

`show` tells Stata to draw individual face graphs. The default is to draw only the final combined graph.

`saveall` tells Stata to save all individual face graphs. This is useful if the user later wants to combine individual face graphs manually. Graphs are saved in the current directory as `FACE1.gph`, `FACE2.gph`, .... Also, a blank graph, `FACE0.gph`, is saved.

`rescale(#)` restricts the range of all variables to an interval narrower than $(0, 1)$; use this option only in the rare cases when some features of the face intersect, producing an effect that is not aesthetically pleasing. For example, specifying `rescale(0.95)` compresses all the variables to the range $(0.05, 0.95)$. It is not recommended to use values less than 0.90 because it introduces an artificial reduction in the variation of the data.

### Combined graph

`legend(`{`2`|`3`} [`nolabel`]`)` generates a legend based on variable labels or, if labels are missing, on variable names. The legend is displayed at the bottom of the combined graph. You may specify `legend(2)` or `legend(3)` to display the legend in two or three columns, respectively. If you do not want variable labels to be included in the legend, add the `nolabel` option. The `legend()` option overrides the `note()` option.

`order(`*varlist*`)` tells Stata to draw faces sorted by *varlist*.

`rows(`#`)` and `cols(`#`)` denote the number of rows and columns in a combined face graph. If both options are specified, `cols()` takes precedence.

`colfirst` tells Stata to display face graphs down columns.

`xcombined(`#`)` and `ycombined(`#`)` specify the size of the combined graph.

`nocombine` tells Stata not to construct the final combined graph, which is the most time-consuming part of the `chernoff` command. This option is most useful in conjunction with `saveall` or if the user wants to take a "first peek" as to whether the individual face graphs look "right".

`title(`*tinfo*`)`, `subtitle(`*tinfo*`)`, `note(`*tinfo*`)`, `nodraw`, and `saving(`*filename*`, replace)` are standard Stata graph options; see [G] ***twoway_options***.

### Utility

`timer` reports the amount of time the command spent constructing individual face graphs (timer 1) and, once the command concludes, the time spent on constructing the combined graph (timer 2). The first reported time should give you a rough idea of how long it will take the command to complete—in testing, stage 2 took on average about 20 percent longer than the time spent on stage 1.

Figure 1 demonstrates a variety of Chernoff faces using the randomly generated variables `x1`–`x18`. Labels have been assigned randomly to half the variables. The code for this and the next examples can be found in the ancillary files and in the online appendix.[1]

1. The online appendix is located at http://www.roofoos.net.

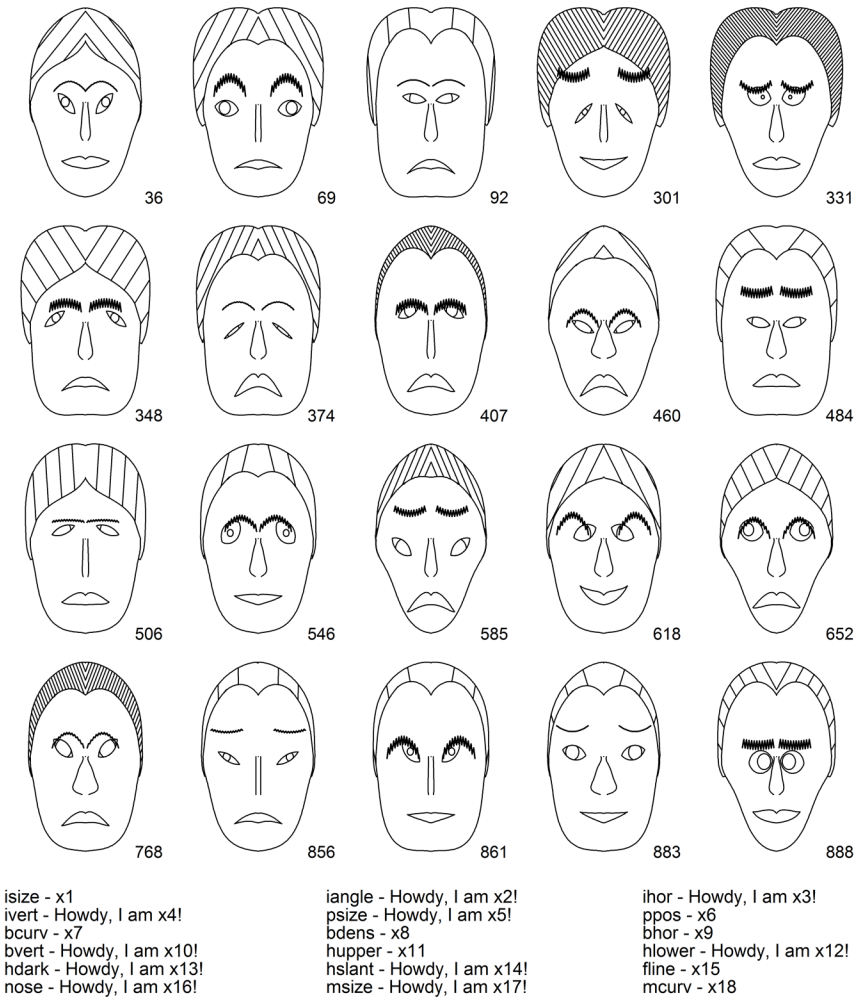



Figure 1. Random faces with a legend

# 3 Applications

## 3.1 Classification/clustering

To illustrate the use of Chernoff faces for the purposes of classification, I replicate the example of public utility companies from Johnson and Wichern (2007). The variables are `covr` (fixed-charge coverage ratio, income/debt), `rtrn` (rate of return on capital), `cost` (cost per kW capacity in place), `load` (annual load factor), `grow` (peak kWh

demand growth from 1974 to 1975), `sale` (sales, kWh use per year), `nuke` (percent nuclear), and `fuel` (total fuel costs, cents per kWh).

The reader should keep in mind that the purpose of this exercise is to demonstrate how to create sophisticated face graphs in Stata rather than argue about the appropriateness of assignment to a particular group. Table 2 presents the data in a tabular format. It is not clear whether there is any clustering in the data.

Table 2. Public utility data, 1975

| id | company | state | covr | rtrn | cost | load | grow | sale | nuke | fuel |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Arizona Public | AZ | 1.06 | 9.2 | 151 | 54.4 | 1.6 | 9077 | 0 | 0.63 |
| 2 | Boston Edison | MA | 0.89 | 10.3 | 202 | 57.9 | 2.2 | 5088 | 25.3 | 1.56 |
| 3 | C. Louisiana El. | LA | 1.43 | 15.4 | 113 | 53 | 3.4 | 9212 | 0 | 1.06 |
| 4 | Comm. Edison | IL | 1.02 | 11.2 | 168 | 56 | 0.3 | 6423 | 34.3 | 0.70 |
| 5 | Con. Edison | NY | 1.49 | 8.8 | 192 | 51.2 | 1 | 3300 | 15.6 | 2.04 |
| 6 | Florida Power | FL | 1.32 | 13.5 | 111 | 60 | −2.2 | 11127 | 22.5 | 1.24 |
| 7 | Hawaiian El. | HI | 1.22 | 12.2 | 175 | 67.6 | 2.2 | 7642 | 0 | 1.65 |
| 8 | Idaho Power | ID | 1.1 | 9.2 | 245 | 57 | 3.3 | 13082 | 0 | 0.31 |
| 9 | Kentucky Utils | KY | 1.34 | 13 | 168 | 60.4 | 7.2 | 8406 | 0 | 0.86 |
| 10 | Madison Gas | WI | 1.12 | 12.4 | 197 | 53 | 2.7 | 6455 | 39.2 | 0.62 |
| 11 | Nevada Power | NV | 0.75 | 7.5 | 173 | 51.5 | 6.5 | 17441 | 0 | 0.77 |
| 12 | New England El. | NE | 1.13 | 10.9 | 178 | 62 | 3.7 | 6154 | 0 | 1.90 |
| 13 | Northern States | MN | 1.15 | 12.7 | 199 | 53.7 | 6.4 | 7179 | 50.2 | 0.53 |
| 14 | Oklahoma Gas | OK | 1.09 | 12 | 96 | 49.8 | 1.4 | 9673 | 0 | 0.59 |
| 15 | Pacific Gas | CA | 0.96 | 7.6 | 164 | 62.2 | −0.1 | 6468 | 0.9 | 1.40 |
| 16 | Puget Sound | WA | 1.16 | 9.9 | 252 | 56 | 9.2 | 15991 | 0 | 0.62 |
| 17 | San Diego Gas | CA | 0.76 | 6.4 | 136 | 61.9 | 9 | 5714 | 8.3 | 1.92 |
| 18 | The Southern Co. | AZ | 1.05 | 12.6 | 150 | 56.7 | 2.7 | 10140 | 0 | 1.11 |
| 19 | Texas Utils | TX | 1.16 | 11.7 | 104 | 54 | −2.1 | 13507 | 0 | 0.64 |
| 20 | Wisconsin El. | WI | 1.2 | 11.8 | 148 | 59.9 | 3.5 | 7287 | 41.1 | 0.70 |
| 21 | United Illum. | CT | 1.04 | 8.6 | 204 | 61 | 3.5 | 6650 | 0 | 2.12 |
| 22 | Virginia El. | VA | 1.07 | 9.3 | 174 | 54.3 | 5.9 | 10093 | 26.6 | 1.31 |

Figure 2 presents the companies divided into clusters as suggested by Johnson and Wichern (2007). I performed the following assignments:

```
... bdens(cost) fline(nuke) hdark(covr) iangle(load) isize(grow) ///
            mcurv(sale) hslant(rtrn) nose(fuel)
```

As can be seen, companies group largely according to geographical location. On a practical side, I first called `chernoff` with the `nocombine` and `saveall` options and made cluster name graphs by hand. I then put the graphs in their appropriate places by using `graph combine`. The empty spaces in the graph were obtained with the multiple use of `FACE0.gph`.

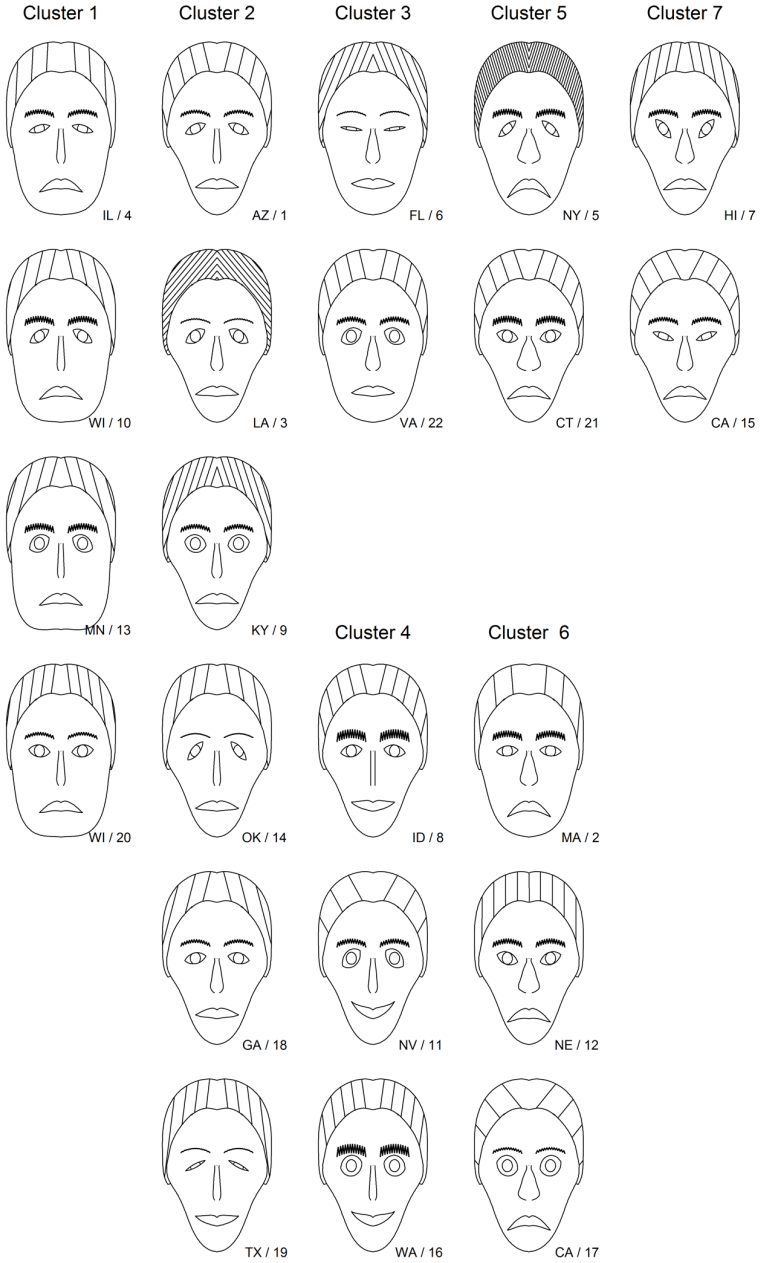



Figure 2. Chernoff faces for 22 public utilities

## 3.2 Outlier detection

An outlier can have a profound effect on the appearance of face features. Consider a variable with values $0, 1, 2, \ldots, 10$. If I assign this variable to all the 18 face features, the face with the values 0 will look "saddest" and the face with the values 10 will look "happiest".

Now consider adding one more observation with the value of 100 and redrawing the faces. Because internally the values are mapped to a $(0, 1)$ range using the formula

$$x_i^* = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

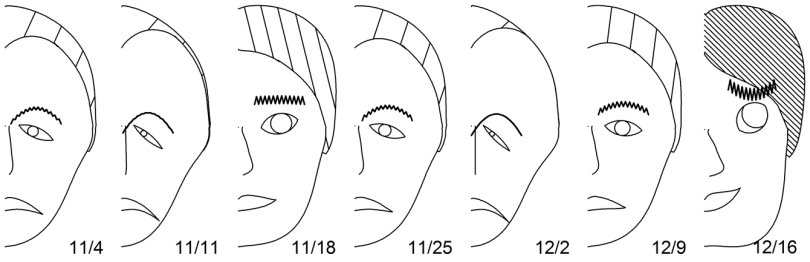
the old mapping $10 \to 1$ now becomes $10 \to 0.10$, and all the original faces will get "compressed" and look "sad" compared with the new arrival. I illustrate this phenomenon on biosurveillance data where I have data points arriving at fixed time intervals. Every time period, the user is presented with faces corresponding to the last seven time periods. If the latest data point appears extreme relative to the past data points, it may signify the beginning of a disease outbreak. Table 3 shows weekly flu data representing the frequency of Google Internet search queries containing the word *flu*.[2] In theory, the frequency of search queries ranges from 0 to 100, and because a public health perspective finds lower values to be more desirable, I define and plot `myflu = 100 - flu`.

Table 3. Flu data, 2004

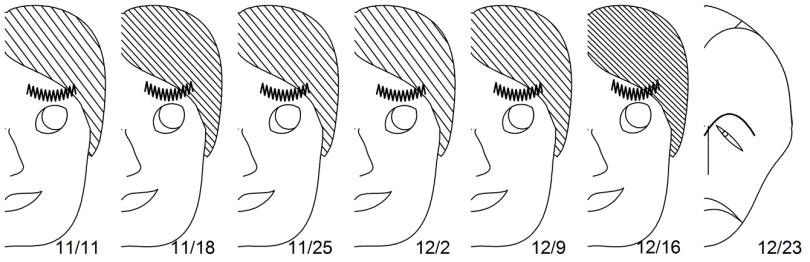
| `t` | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | $t_8$ |
|---|---|---|---|---|---|---|---|---|
| `date` | 11/4 | 11/11 | 11/18 | 11/25 | 12/2 | 12/9 | 12/16 | 12/23 |
| `flu` | 19.38 | 20.99 | 15.71 | 18.80 | 21.42 | 17.73 | 12.53 | 97.17 |
| `myflu` | 80.62 | 79.01 | 84.29 | 81.20 | 78.58 | 82.27 | 87.47 | 2.83 |

Figure 3a shows faces for $t_1 - t_7$ drawn without considering the theoretical extrema. All the values of `myflu` are relatively high compared with the possible minimum of 0, yet I get an impression that most of the time I should be worried about a flu outbreak. Now consider what happens when `myflu` at $t_8$ is added. Because the new arrival represents a true departure from the previous pattern, it completely changes the look of the faces (figure 3b). The researcher will certainly be baffled by the resulting picture.

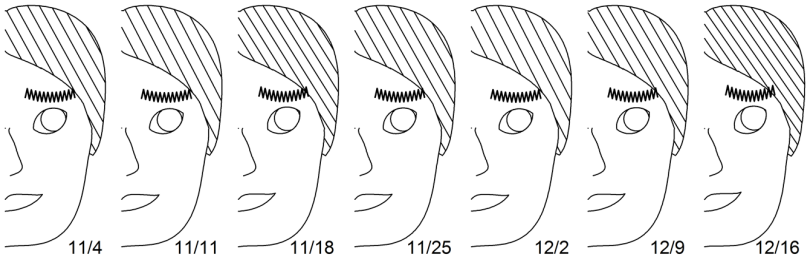
2. Data courtesy of Taha Kass-Hout and Google Insights for Search.



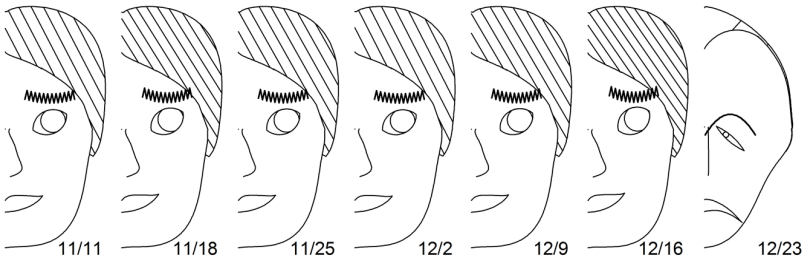

(a) Time period $t_1$–$t_7$ without considering global extrema




(b) Time period $t_2$–$t_8$ without considering global extrema




(c) Time period $t_1$–$t_7$ with adjustment for global extrema




(d) Time period $t_2$–$t_8$ with adjustment for global extrema

Figure 3. Flu data

Now let's revisit the data but this time include a theoretical min and max. This can be done two ways. First, you can specify

```
. chernoff, isize(myflu, 0 100) iangle(myflu, 0 100) ... mcurv(myflu, 0 100)
```

which gets the job done but is tedious. This is where the `gmin()` and `gmax()` options come in handy. If all your variables share the same theoretical min or max, you can specify a global min or max for all of them by typing

```
. chernoff, isize(myflu) iangle(myflu) ... mcurve(myflu) gmin(0) gmax(100)
```

Figure 3c presents the same data as figure 3a, but I account for extremes. Now all the faces are correctly depicted as "happy". Also, the arrival of an outlier does not alter face representation for the previous time periods (figure 3d).

The flu example illustrates the importance of including a theoretical minimum and maximum for variables, especially for continually updated or "live" data. This will not always be possible because some phenomena do not have a strictly defined min or max, for example, growth rates. In those cases, the researcher should be aware that an arrival of an outlier can significantly alter the face landscape.

# 4 Further advice

Over the years, Chernoff faces received their fair share of criticism. However, most pitfalls can be avoided by carefully thinking about the problem at hand and by knowing your data. In this section, I offer several suggestions with respect to the use of Chernoff graphs and the `chernoff` command.

## 4.1 Inverted scales

Be aware of what your variables represent. If you have a depression score where higher values indicate higher depression, do not assign it to the mouth feature because higher values will result in a smile. The usual solution is to reverse the values before the assignment, similar to what I did in the flu example. Good candidates for inverted scales include employee turnover, occurrence of earthquakes, dropout rate, number of bankruptcies, and measures of corruption. Be aware that some popular indices are coded counterintuitively—for example, the Freedom House index of democracy uses 1 to denote free countries and 7 to denote not free countries.[3]

## 4.2 No if or in

The sensitivity of face features with respect to outliers is the reason that I have not implemented the `if` or `in` option. You could be tempted to code

```
. chernoff if foreign==1 ...
```

3. The Freedom House web site is http://www.freedomhouse.org.

but does it mean that you want face features to be calculated on the foreign car subsample or do you want calculations to be performed on the whole sample but graph only faces representing foreign cars? To avoid the confusion, `chernoff` does not honor qualifiers. If you wish to plot faces for a particular subsample, you can always drop the unneeded observations, although you should probably specify a min and max from the entire sample.

## 4.3 Limits

The main challenge in programming Chernoff faces is that each individual face requires many data points to be plotted. For example, a SAS implementation by Friendly (1991) uses "approximately 800 annotate observations for each face". In my implementation, each face requires 51 temporary variables, but those are recycled for each face; thus the user is not limited by the number of variables Stata can hold. The only restriction is that the number of sersets is capped at 1,999—because each face uses two sersets, this limits the size of the combined graph to 999 faces; see [P] **serset** for more details. Hair shading is stored as a separate serset, thus specifying `hdark(_null_)` or `hslant(_null_)` will allow for 1,999 faces to be plotted simultaneously.

## 4.4 Number of observations

Several face features are implemented as fifth-degree polynomials. Upper and lower hair line, stored in temporary variables, each needs 121 data points to be plotted. When the `chernoff` command starts, it checks whether there is a sufficient number of observations, and if not, it issues `set obs 121`. The number of observations is restored to the original value when the command finishes. However, if you press the *Break* key in the middle of the execution, the command will not be able to restore the original number of observations. For this reason, at the start the command reports the number of observations being processed. If you interrupted the command and had less than 121 observations to begin with, make sure to check for redundant observations.

## 4.5 Size of a combined graph

The `chernoff` command is smart enough to calculate the correct dimensions of the combined graph whether you choose to draw full or half faces and whether you specify the number of rows or columns or not. Because Stata does not allow the $x$ size or $y$ size of the graph to exceed 20, the command will rescale both dimensions proportionately if they do. You may need to adjust the size of the combined graph when you add a title and subtitle, change the position of the label, etc.

# 5 Conclusion

In 1994, Nel, Pitt, and Webb noted that the difficulty with the implementation of Chernoff faces is that "some companies may not have access to the facilities and skills with which to generate faces" (p. 253). They also expressed hope that "recent developments in the field of personal computers and user-friendly software may eventually alleviate this to some extent". The solution is long overdue.

The `chernoff` command, introduced in this article, is straightforward to use and does not require any special statistical skills. To my knowledge, this is a second implementation of Chernoff faces in a mainstream statistical software, with the first one being due to Friendly (1991). I hope that my command will prove useful in many diverse fields that rely on visualization of highly dimensional data to detect patterns, clusters, outliers, and temporal trends.

# 6 References

Apaiwongse, T. S. 1995. Facial display of environmental policy uncertainty. *Journal of Business and Psychology* 10: 65–74.

Astel, A., K. Astel, M. Biziuk, and J. Namieśnik. 2006. Classification of drinking water samples using the Chernoff's faces visualization approach. *Polish Journal of Environmental Studies* 15: 691–697.

Camussi, A., S. Raddi, and P. Raddi. 1992. Visual identification of forest-tree clones by using Chernoff's faces. *Taxon* 41: 451–458.

Chernoff, H. 1971. The use of faces to represent points in $n$-dimensional space graphically. Technical Report 71, Department of Statistics, Stanford University.

———. 1973. The use of faces to represent points in $k$-dimensional space graphically. *Journal of the American Statistical Association* 68: 361–368.

Flury, B. 1980. Construction of the asymmetrical face to represent multivariate data graphically. Technical Report 3, Institute of Mathematical Statistics and Actuarial Science, Bern University.

Flury, B., and H. Riedwyl. 1981. Graphical representation of multivariate data by means of asymmetrical faces. *Journal of the American Statistical Association* 76: 757–765.

Friendly, M. 1991. Faces: Faces display of multivariate data. Department of Psychology, York University. http://www.math.yorku.ca/SCS/sasmac/faces.html.

Golden, L. L., and M. Sirdesai. 1992. Chernoff faces: A useful technique for comparative image analysis and representation. *Advances in Consumer Research* 19: 123–128.

Johnson, R. A., and D. W. Wichern. 2007. *Applied Multivariate Statistical Analysis*. 6th ed. Upper Saddle River, NJ: Prentice Hall.

Lott, J. A., and T. C. Durbridge. 1990. Use of Chernoff faces to follow trends in laboratory data. *Journal of Clinical Laboratory Analysis* 4: 59–63.

Nel, D., L. Pitt, and T. Webb. 1994. Using Chernoff faces to portray service quality data. *Journal of Marketing Management* 10: 247–255.

Schüpbach, M. 1987. ASYMFACE asymmetrical faces in TurboPascal. Technical Report 25, Institute of Mathematical Statistics and Actuarial Science, Bern University.

**About the author**

Rafal Raciborski is a graduate student in the Department of Political Science at Emory University. His interests include political economy, applied econometrics, and statistical computing.

The Stata Journal (2009)
**9**, Number 3, pp. 388–397

# Improved degrees of freedom for multivariate significance tests obtained from multiply imputed, small-sample data

Yulia V. Marchenko
StataCorp
College Station, TX
ymarchenko@stata.com

Jerome P. Reiter
Department of Statistical Science
Duke University
Durham, NC
jerry@stat.duke.edu

**Abstract.** We propose improvements to existing degrees of freedom used for significance testing of multivariate hypotheses in small samples when missing data are handled using multiple imputation. The improvements are for 1) tests based on unrestricted fractions of missing information and 2) tests based on equal fractions of missing information with $M(p-1) \leq 4$, where $M$ is the number of imputations and $p$ is the number of tested parameters. Using the `mi` command available as of Stata 11, we demonstrate via simulation that using these adjustments can result in a more sensible degrees of freedom (and hence closer-to-nominal rejection rates) than existing degrees of freedom.

**Keywords:** st0170, multiple imputation, degrees of freedom, sample, missing, testing, multivariate

## 1 Introduction

Multiple imputation developed by Rubin (1987) is a popular approach for handling missing data. The basic idea is for the data collector or imputer to simulate values for the missing data repeatedly by sampling from predictive distributions of the missing values. The data analyst, who may be the same person as the imputer or may be a secondary data user, performs the desired analysis on each completed dataset and combines the results using simple formulas (Rubin 1987, 76–77). As of Stata 11, the `mi` command provides methods for generating multiple imputations and implements the formulas for combining results (StataCorp 2009). Users also can perform multiple imputation by using `ice` and `mim` (Royston 2004, 2005a, 2005b, 2007; Carlin, Galati, and Royston 2008; and Royston, Carlin, and White 2009). For reviews of multiple imputation, see Schafer (1997), Little and Rubin (2002), and Reiter and Raghunathan (2007).

Often analysts seek to test multivariate hypotheses, for example, if several regression coefficients are equal to zero. Rubin (1987) suggests two approaches to doing so with multiply imputed data. The first approach, which is the most widely used method, presumes that the fractions of missing information (FMI) are equal across the parameters of interest. A reference $F$ distribution for this method was derived by Li, Raghunathan, and Rubin (1991). The second approach does not presume equal

FMIs; however, it may not give well calibrated $p$-values unless the number of imputed datasets is large (Li, Raghunathan, and Rubin 1991).

The derivations of these test statistics and their reference distributions presume infinite sample size. However, Reiter (2007) demonstrates that, for the equal FMI test, the infinite sample-size assumption can result in nonsensical procedures. For example, in modest samples, the computed degrees of freedom for the reference distributions can exceed the number of cases in the dataset, which should not be possible. A related phenomenon is illustrated by Barnard and Rubin (1999), who derive small-sample degrees of freedom for univariate inferences.

Reiter (2007) goes on to develop small-sample degrees of freedom for the equal FMI test that results in better performance than the infinite sample degrees of freedom of Li, Raghunathan, and Rubin (1991). However, Reiter's (2007) degrees of freedom requires $M(p-1) > 4$, where $M$ is the number of imputations and $p$ is the number of tested parameters. While this case may not be a concern in practice because analysts can set $M$ to be large, it nonetheless must be accounted for when designing software to implement multiple imputation. For multivariate tests based on unrestricted FMIs, we are not aware of any published research on small-sample adjustments to the degrees of freedom.

Motivated by the development of `mi`, we propose to fill these gaps in the literature. Specifically, we present small-sample degrees of freedom for the unrestricted FMI test and for the equal FMI test with $M(p-1) \leq 4$. We demonstrate with simulation results that using the adjusted degrees of freedom can result in more sensible reference distributions (and hence closer-to-nominal rejection rates) than using degrees of freedom based on infinite sample sizes.

## 2 Significance tests with multiple imputation

We first review the unrestricted and equal FMI tests. Let $\mathbf{q}$ be the $p \times 1$ vector of parameters of interest, such as $p$ regression coefficients. In each completed dataset $i$, where $i = 1, \ldots, M$, let $\widehat{\mathbf{q}}_i$ be the completed-data estimate of $\mathbf{q}$, and let $\widehat{\mathbf{U}}_i$ be its associated completed-data variance estimate. The analyst combines each $\widehat{\mathbf{q}}_i$ and $\widehat{\mathbf{U}}_i$ using

$$\begin{aligned} \overline{\mathbf{q}} &= \frac{1}{M}\sum_{i=1}^{M}\widehat{\mathbf{q}}_i \\ \mathbf{T} &= \overline{\mathbf{U}} + (1+\frac{1}{M})\mathbf{B} \end{aligned}$$

Here $\overline{\mathbf{U}} = \sum_{i=1}^{M}\widehat{\mathbf{U}}_i/M$ is the within-imputation variance–covariance matrix, and $\mathbf{B} = \sum_{i=1}^{M}(\mathbf{q}_i - \overline{\mathbf{q}})(\mathbf{q}_i - \overline{\mathbf{q}})'/(M-1)$ is the between-imputation variance–covariance matrix. The analyst can use $\overline{\mathbf{q}}$ as a point estimate of $\mathbf{q}$ and $\mathbf{T}$ as an estimate of the variance of $\overline{\mathbf{q}}$.

We now suppose that the analyst seeks to test the null hypothesis, $H_0$: $\mathbf{q} = \mathbf{q}_0$. The *unrestricted FMI test* proposed by Rubin (1987) is

$$(\mathbf{q}_0 - \overline{\mathbf{q}})\mathbf{T}^{-1}(\mathbf{q}_0 - \overline{\mathbf{q}})'/p \sim F_{p,\nu} \tag{1}$$

where

$$\begin{aligned} \nu &= (M-1)(1 + 1/r_{\text{ave}})^2 \\ r_{\text{ave}} &= (1 + 1/M)\text{tr}(\mathbf{B}\overline{\mathbf{U}}^{-1})/p \end{aligned}$$

Here $r_{\text{ave}}$ is the average relative variance increase due to missing data.

Note that even under the assumption of infinite sample size, multivariate testing uses an $F$ reference distribution rather than a chi-squared distribution. This is because of the fact that the variance $\mathbf{T}$ in the test statistic (1) involves estimates of the within- and between-imputations variances based on the finite number of $M$ imputations. Therefore, the denominator degrees-of-freedom parameter $\nu$ in (1) represents the amount of independent information used to estimate the variance after accounting for a finite number of imputations. In standard multiple-imputation contexts, this amount of information theoretically cannot exceed the number of cases in the dataset, which sometimes happens with the approximations of Rubin (1987) and Li, Raghunathan, and Rubin (1991).

The *equal FMI test* originally suggested by Rubin (1987) is

$$(1 + r_{\text{ave}})^{-1}(\mathbf{q}_0 - \overline{\mathbf{q}})\overline{\mathbf{U}}^{-1}(\mathbf{q}_0 - \overline{\mathbf{q}})'/p \sim F_{p,(p+1)\nu/2} \tag{2}$$

Reiter (2007) uses the same test statistic as (2) with an alternative denominator degrees of freedom for the $F$ distribution appropriate when $M(p-1) > 4$.

The key distinction between the two test statistics is the variance inside the quadratic form. The unrestricted FMI test uses $\mathbf{T}$, whereas the equal FMI test uses $(1 + r_{\text{ave}})\overline{\mathbf{U}}$. This difference arises because of the equal FMI condition. To see this, define $\mathbf{B}_\infty = \lim \mathbf{B}$ as $M \to \infty$, and define $\mathbf{T}_\infty = \overline{\mathbf{U}} + (1 + 1/M)\mathbf{B}_\infty$; we could obtain these values if we had an infinite number of datasets to estimate $\mathbf{B}$ and $\mathbf{T}$. Under equal FMIs, $\overline{\mathbf{U}} = \rho\mathbf{B}_\infty$ for some constant $\rho$, and thus $\mathbf{T}_\infty = (1 + \rho)\overline{\mathbf{U}}$. The relative variance increase, $r_{\text{ave}}$, in (2) is an estimate of $\rho$.

At first glance, the unrestricted test would seem to be always preferable because it is derived under more general conditions. However, Rubin (1987) shows that the unrestricted test can perform poorly when $M$ is small relative to $p$ because $\mathbf{B}$ can be unreliable. Essentially, using $\mathbf{B}$ to estimate $\mathbf{B}_\infty$ from the $M$ datasets is akin to estimating a $p \times p$ covariance matrix with only $M$ observations, which can be problematic when $M < p$. Using the equal FMI test mitigates these difficulties because the analyst estimates only one parameter, $\rho$, rather than $p^2 + p(p-1)/2$ parameters. Li, Raghunathan, and Rubin (1991) demonstrate that testing procedures based on the assumption of equal FMIs perform well as long as the fractions do not vary substantially.

## 3 Small-sample degrees-of-freedom adjustments

We now consider adjustments for the denominator degrees of freedom in the reference distributions in (1) and, for cases with $M(p-1) \leq 4$, in (2) to reflect small samples.

For the unrestricted FMI test, we propose to use the small-sample degrees of freedom of Barnard and Rubin (1999) in place of $\nu$ in (1). That is, we use

$$\nu_{\text{br}} = \left(\nu_{\star}^{-1} + \widehat{\nu}_{\text{obs}}^{-1}\right)^{-1}$$

where

$$\begin{aligned} \nu_{\star} &= (M-1)\gamma_{\text{ave}}^{-2} \\ \widehat{\nu}_{\text{obs}} &= (1-\gamma_{\text{ave}})\nu_{\text{com}}(\nu_{\text{com}}+1)/(\nu_{\text{com}}+3) \\ \gamma_{\text{ave}} &= (1+1/M)\text{tr}(\mathbf{B}\mathbf{T}^{-1})/p \end{aligned}$$

Here $\nu_{\text{com}}$ is the degrees of freedom if the data were complete, and $\gamma_{\text{ave}}$ is the approximate average FMI. The quantity $\nu_{\text{br}}$ has several features that led Barnard and Rubin (1999) to recommend its general use, regardless of the sample size. First, $\nu_{\text{br}} \leq \nu_{\text{com}}$, whereas $\nu$ can exceed $\nu_{\text{com}}$. This property of $\nu_{\text{br}}$ is desirable because the presence of missing data should reduce the degrees of freedom rather than increase it. Second, $\nu_{\text{br}} < \nu$ with approximate equality when the sample size is large, so using $\nu_{\text{br}}$ instead of $\nu$ is slightly conservative in large samples. Third, $\nu_{\text{br}}$ is always between $\nu_{\text{com}}$ and $\nu$, making it a compromise degrees of freedom.

Barnard and Rubin (1999) illustrate the effectiveness of this degrees of freedom for univariate inferences. To our knowledge, $\nu_{\text{br}}$ is rarely, if ever, used for multivariate inferences. However, Barnard and Rubin (1999) note that the steps in the derivation of $\nu_{\text{br}}$ for multivariate $\mathbf{q}$ follow immediately under equal FMIs. Hence, by using $\nu_{\text{br}}$ for the degrees of freedom in the unrestricted test, we lean on the equal FMI assumption to avoid unrealistic degrees of freedom, but we do allow the variance in the quadratic form to be estimated without the restriction.

For the equal FMI test, we suggest a refinement to the degrees of freedom of Reiter (2007) for cases when $M(p-1) \leq 4$. Here we again propose to substitute $\nu_{\text{br}}$ for $\nu$ in (2). This is similar in spirit to the suggestion of Li, Raghunathan, and Rubin (1991), who use (2) for cases when $M(p-1) \leq 4$ for their large-sample tests. The primary difference is that we use a degrees of freedom, $\nu_{\text{br}}$, that has more desirable properties in small samples.

## 4 Simulation studies of properties of adjustments

The proposed adjusted degrees of freedom are ad hoc in nature. As noted by Rubin (1987), there is little way around such constructions, because we are approximating complicated Bayesian integrals with simple distributions. Thus it is imperative to evaluate the operating characteristics of tests based on these procedures by using simulation studies.

In all simulations studies, we generate an outcome, $Y$, and covariates, ($X_1$, $X_2$, ..., $X_p$), where $p$ depends on the simulation scenario, for 50 observations. The covariates are sampled from a multivariate normal distribution with means equal to zero, variances equal to one, and all pairwise correlations equal to 0.5. The outcome is sampled from a normal distribution with mean equal to zero and variance equal to one independently of covariates. The `simulate` command is used to generate the data. We investigate the empirical significance levels of the procedures when testing if all coefficients in the regression of $Y$ on $(X_1, X_2, \ldots, X_p)$ are equal to zero; that is, we test $H_0$: $\beta_1 = \beta_2 = \cdots = \beta_p = 0$. The empirical significance levels are based on 10,000 replications.

We consider four simulation scenarios in which we vary FMIs; these are described in sections 4.1 and 4.2. Multiple imputations are performed using `mi impute mvn`, which implements multivariate normal imputation. The estimation step is performed using `mi estimate`. The results of the equal FMI test are obtained from the default settings of `mi test`. The results of the unrestricted FMI test are obtained by specifying the `ufmitest` option with `mi test`. The results from the corresponding large-sample tests are obtained by specifying the `nosmall` option with `mi test`.

## 4.1 Small-sample adjustment for the unrestricted FMI test

Scenarios 1, 2, and 3 use $p = 4$ covariates, and scenario 4 uses $p = 5$ covariates. In scenario 1, we randomly delete 10% of the 50 observations, which corresponds to approximately equal fractions of information missing due to nonresponse. Scenario 2 is similar to scenario 1 but with 30% of the observations deleted. In scenario 3, we introduce variation among the FMIs by randomly deleting 10% of the data from $X_2$, 20% of the data from $X_4$, and 35% of the data from $X_3$; here $X_1$ and $Y$ are complete. Scenario 4 represents a relatively large deviation from equal FMI with increased missingness: 10% of the data are deleted from $X_4$, 30% of the data are deleted from $X_2$, 50% of the data are deleted from $X_1$ and $X_3$, and $X_5$ and $Y$ are complete. We use $M = 20$ multiple imputations.

Table 1 displays key results from the 10,000 replications. Across all scenarios, the small-sample degrees of freedom, $\nu_{\text{br}}$, is more sensible than the large-sample degrees of freedom in (2), $\nu$, which always greatly exceeds the sample size of 50. The unrestricted FMI test using $\nu_{\text{br}}$ provides close-to-nominal significance levels and is somewhat conservative. In contrast, the unrestricted FMI test using $\nu$ is anticonservative; its empirical significance levels always exceed the corresponding nominal significance levels. The difference between the empirical and nominal levels is always smaller for the test based on $\nu_{\text{br}}$. Thus we recommend $\nu_{\text{br}}$ over $\nu$ for the unrestricted FMI test.

Table 1. Simulated significance levels for the unrestricted FMI test of all coefficients equal to zero. $\overline{\nu}$ denotes the denominator degrees of freedom averaged over replications.

| Scenario | DF | $\overline{\nu}$ | $\alpha = 0.1$ | $\alpha = 0.05$ | $\alpha = 0.01$ |
|---|---|---|---|---|---|
| 10% missing, equal FMI (1) | small | 36.22 | 0.0941 | 0.0455 | 0.0081 |
| | large | 1180.76 | 0.1164 | 0.0636 | 0.0168 |
| 30% missing, equal FMI (2) | small | 22.57 | 0.0819 | 0.0361 | 0.0069 |
| | large | 130.58 | 0.1103 | 0.0587 | 0.0162 |
| max 35% missing, unequal FMI (3) | small | 33.04 | 0.0965 | 0.0477 | 0.0082 |
| | large | 660.37 | 0.1209 | 0.0674 | 0.0185 |
| max 50% missing, unequal FMI (4) | small | 23.57 | 0.0892 | 0.0452 | 0.0089 |
| | large | 151.24 | 0.1224 | 0.0725 | 0.023 |

## 4.2 Small-sample adjustment for the equal FMI test

To evaluate the performance of the testing procedure under the equal FMI assumption for the case when $M(p-1) \leq 4$, we consider four simulation scenarios similar to those used for the unrestricted test. We use $p = 2$ covariates, $(X_1, X_2)$, and $M = 3$ imputations so that $M(p-1) = 3$. In scenario 1, we randomly delete 10% of all observations. In scenario 2, we randomly delete 30% of all observations. In scenario 3, we randomly delete 10% of the data from $X_1$ and 35% of the data from $X_2$. In scenario 4, we randomly delete 30% of the data from $X_1$ and 50% of the data from $X_2$.

Table 2 displays the key results from the 10,000 replications. In all cases, $\nu_{\text{br}}$ is less than the sample size of 50, whereas the degrees of freedom in (2) far exceeds 50. For the scenarios with modest FMIs (scenarios 1 and 3), the test based on $\nu_{\text{br}}$ generally has closer-to-nominal empirical significance levels than the test based on the degrees of freedom in (2). However, the picture is less clear with large FMIs (scenarios 2 and 4): the levels for the test based on $\nu_{\text{br}}$ are closer to nominal when $\alpha = 0.01$ but not when $\alpha \in (0.05, 0.10)$. For the scenarios with equal FMIs, the test based on $\nu_{\text{br}}$ is conservative, whereas the test based on the degrees of freedom in (2) can be anticonservative. For both degrees of freedom, the tests in scenarios 3 and 4 are reasonably well calibrated despite the unequal FMI, although the levels for the test based on $\nu_{\text{br}}$ can exceed the nominal $\alpha$ in this case.

(*Continued on next page*)

Table 2. Simulated significance levels for the equal FMI test of two coefficients equal to zero. $\overline{\nu}$ denotes the denominator degrees of freedom averaged over replications.

| Scenario | DF | $\overline{\nu}$ | $\alpha = 0.1$ | $\alpha = 0.05$ | $\alpha = 0.01$ |
|---|---|---|---|---|---|
| 10% missing, equal FMI (1) | small | 40.09 | 0.0933 | 0.0485 | 0.0099 |
| | large | 2025.32 | 0.1017 | 0.0558 | 0.0123 |
| 30% missing, equal FMI (2) | small | 15.65 | 0.0795 | 0.0406 | 0.0096 |
| | large | 97.74 | 0.0886 | 0.0480 | 0.0127 |
| max 35% missing, unequal FMI (3) | small | 31.08 | 0.1023 | 0.0536 | 0.0132 |
| | large | 888.59 | 0.1111 | 0.0625 | 0.0170 |
| max 50% missing, unequal FMI (4) | small | 17.22 | 0.0872 | 0.0435 | 0.011 |
| | large | 154.50 | 0.0967 | 0.0509 | 0.0147 |

Taking these results as a whole, we recommend using the adjusted degrees of freedom when $M(p-1) \leq 4$. The test based on $\nu_{\text{br}}$ tends to be conservative when the assumption of equal FMI is true or nearly true, which is when these tests perform best. Of course, data analysts need not force themselves into choosing between these two degrees of freedom. They can increase $M$ sufficiently so that $M(p-1) > 4$ and use the degrees of freedom developed by Reiter (2007) for the equal FMI test, which has been shown to perform well with approximately equal FMIs. For small sample sizes, using a large $M$ should not be a computational burden and can greatly improve analyses.

## 5 Illustration of testing with multiple imputation in Stata

As an example of testing multivariate hypotheses, we use multiply imputed data on house resale prices, `mhouses1993s30.dta`, from example 2 in the Stata manual for the `mi estimate` command, [MI] **mi estimate**. The original data are provided by the Albuquerque Board of Realtors and distributed by the Data and Story Library (http://lib.stat.cmu.edu/DASL/Stories/homeprice.html).

We are interested in the effect of house characteristics like square footage, age of house, and amount of taxes paid on house prices, which we estimate with a linear regression. The data contain missing values on age and taxes. `mhouses1993s30.dta` contains $M = 30$ imputations created using `mi impute mvn`, which invokes multivariate normal imputations. The imputation strategies are described in detail in example 3 of the Stata manual entry [MI] **mi impute mvn**.

Below we present the results of the regression on the multiply imputed data. These results are obtained by using `mi estimate`. We specify the `vartable` option to display the estimated FMIs. The test statistic and $p$-value for the test of all coefficients equaling zero are displayed in the regression output header. By default, this test is based on the

equal FMI test with the degrees of freedom of Reiter (2007). Based on this test, there is significant evidence to reject the null hypothesis that all coefficients equal zero.

```
. use http://www.stata-press.com/data/r11/mhouses1993s30
(Albuquerque Home Prices Feb15-Apr30, 1993)
. mi estimate, vartable: regress price sqft age nfeatures ne custom corner tax
Multiple-imputation estimates                     Imputations     =         30
Variance information
```

| | Imputation variance | | | | | Relative |
|---|---|---|---|---|---|---|
| | Within | Between | Total | RVI | FMI | efficiency |
| sqft | .004442 | .003623 | .008186 | .842713 | .464984 | .984737 |
| age | .277762 | .896309 | 1.20395 | 3.33446 | .778164 | .974717 |
| nfeatures | 157.333 | 26.7139 | 184.937 | .175452 | .150568 | .995006 |
| ne | 1104.74 | 114.734 | 1223.29 | .107319 | .097502 | .99676 |
| custom | 1783.12 | 85.8858 | 1871.87 | .049772 | .04756 | .998417 |
| corner | 1548.13 | 93.6976 | 1644.95 | .06254 | .059084 | .998034 |
| tax | .012421 | .00814 | .020832 | .677183 | .410355 | .986506 |
| _cons | 3834.84 | 257.487 | 4100.91 | .069382 | .065152 | .997833 |

```
Linear regression                                 Number of obs   =        117
                                                  Average RVI     =     0.5415
                                                  Complete DF     =        109
DF adjustment:   Small sample                     DF:     min     =      16.42
                                                          avg     =      72.83
                                                          max     =     101.18
Model F test:       Equal FMI                     F(   7,   96.3) =      45.63
Within VCE type:          OLS                     Prob > F        =     0.0000
```

| price | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| sqft | .2900879 | .0904748 | 3.21 | 0.003 | .1073624 | .4728134 |
| age | -.7524605 | 1.097246 | -0.69 | 0.502 | -3.073675 | 1.568754 |
| nfeatures | 4.361055 | 13.59917 | 0.32 | 0.749 | -22.67719 | 31.3993 |
| ne | 5.495913 | 34.97562 | 0.16 | 0.875 | -63.95148 | 74.94331 |
| custom | 132.3453 | 43.26507 | 3.06 | 0.003 | 46.52087 | 218.1697 |
| corner | -66.95606 | 40.55801 | -1.65 | 0.102 | -147.4264 | 13.51429 |
| tax | .5516444 | .1443319 | 3.82 | 0.000 | .2612817 | .842007 |
| _cons | 130.3491 | 64.03837 | 2.04 | 0.044 | 3.277868 | 257.4203 |

We can use `mi test` to test hypotheses about subsets of coefficients. Suppose that we seek to test the null hypothesis that the coefficients for `age`, `nfeatures` (number of certain features), and `ne` (whether the city is located in the northeast, largest residential, area) all equal zero. By default, `mi test` performs the equal FMI test, as illustrated below.

```
. mi test age nfeatures ne
note: assuming equal fractions of missing information
 ( 1)  age = 0
 ( 2)  nfeatures = 0
 ( 3)  ne = 0
       F(  3,  70.4) =    0.39
            Prob > F =    0.7639
```

However, from the output of `mi estimate`, the assumption of equal FMIs for `age`, `nfeatures`, and `ne` does not seem plausible: the estimated FMIs range from 0.10 for `ne` to 0.78 for `age`. We therefore perform the unrestricted FMI test with the `ufmitest` option, as follows.

```
. mi test age nfeatures ne, ufmitest
 ( 1)  age = 0
 ( 2)  nfeatures = 0
 ( 3)  ne = 0
       F(  3,  41.8) =    0.28
            Prob > F =    0.8376
```

The unrestricted FMI test results in a larger $p$-value than the equal FMI test. However, both tests indicate that these three variables are not strong predictors of house resale prices, at least according to the model we fit here.

# 6 Conclusion

We proposed improvements to the existing degrees of freedom for multivariate tests for multiply imputed data. In particular, we proposed a small-sample adjustment to the degrees of freedom of the unrestricted FMI test, and we refined the small-sample adjustment for the equal FMI test when $M(p-1) \leq 4$. Empirical evaluations of these adjustments, while admittedly limited in scope as all such evaluations must be, demonstrated that using tests based on the proposed small-sample adjustments can improve performance over using tests based on the large-sample analogues. Simulations also showed that the proposed testing procedures become more conservative as FMIs increase or start varying substantially. The deviations from nominal significance result because the adjustments are, as noted previously, unavoidably ad hoc in nature. For example, the derivation of the proposed degrees of freedom presumes that FMIs are approximately equal even though this assumption is not used in the test statistic. Additionally, estimates of the within-imputation and between-imputations variance components can be unreliable for small sample sizes and modest numbers of imputations.

We also considered using the denominator degrees of freedom suggested by Reiter (2007) for the unrestricted FMI test. This led to a slightly more conservative test than the one using the degrees of freedom from Barnard and Rubin (1999).

Other simulations not shown here suggested that the small-sample unrestricted FMI test performs better than the small-sample equal FMI test when the FMIs vary noticeably, and that the small-sample equal FMI test performs better when the FMIs are approximately equal. Further research is needed to compare the properties of these two tests in a wide range of plausible scenarios.

# 7 References

Barnard, J., and D. B. Rubin. 1999. Small-sample degrees of freedom with multiple imputation. *Biometrika* 86: 948–955.

Carlin, J. B., J. C. Galati, and P. Royston. 2008. A new framework for managing and analyzing multiply imputed data in Stata. *Stata Journal* 8: 49–67.

Li, K.-H., T. E. Raghunathan, and D. B. Rubin. 1991. Large-sample significance levels from multiply imputed data using moment-based statistics and an $F$ reference distribution. *Journal of the American Statistical Association* 86: 1065–1073.

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

Reiter, J. P. 2007. Small-sample degrees of freedom for multi-component significance tests with multiple imputation for missing data. *Biometrika* 94: 502–508.

Reiter, J. P., and T. E. Raghunathan. 2007. The multiple adaptations of multiple imputation. *Journal of the American Statistical Association* 102: 1462–1471.

Royston, P. 2004. Multiple imputation of missing values. *Stata Journal* 4: 227–241.

———. 2005a. Multiple imputation of missing values: Update. *Stata Journal* 5: 188–201.

———. 2005b. Multiple imputation of missing values: Update of ice. *Stata Journal* 5: 527–536.

———. 2007. Multiple imputation of missing values: Further update of ice, with an emphasis on interval censoring. *Stata Journal* 7: 445–464.

Royston, P., J. B. Carlin, and I. R. White. 2009. Multiple imputation of missing values: New features for mim. *Stata Journal* 9: 252–264.

Rubin, D. B. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.

Schafer, J. L. 1997. *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: Chapman & Hall/CRC.

StataCorp. 2009. *Stata 11 Multiple-Imputation Reference Manual*. College Station, TX: Stata Press.

#### About the authors

Yulia V. Marchenko is a senior statistician at StataCorp.

Jerome P. Reiter is an associate professor in the Department of Statistical Science at Duke University in Durham, North Carolina. His research interests include methodology for missing data, methodology for protecting confidentiality in public use data, methodology for causal inference, and applied statistical modeling.

The Stata Journal (2009)
**9**, Number 3, pp. 398–421

# Implementing weak-instrument robust tests for a general class of instrumental-variables models

Keith Finlay
Tulane University
New Orleans, LA
kfinlay@tulane.edu

Leandro M. Magnusson
Tulane University
New Orleans, LA
lmagnuss@tulane.edu

**Abstract.** We present a minimum distance approach for conducting hypothesis testing in the presence of potentially weak instruments. Under this approach, we propose size-correct tests for limited dependent variable models with endogenous explanatory variables such as endogenous tobit and probit models. Additionally, we extend weak-instrument tests for the linear instrumental-variables model by allowing for variance–covariance estimation that is robust to arbitrary heteroskedasticity or intracluster dependence. We invert these tests to construct confidence intervals on the coefficient of the endogenous variable. We also provide a postestimation command for Stata, called `rivtest`, for computing the tests and estimating confidence intervals.

**Keywords:** st0171, rivtest, ivregress, ivprobit, ivtobit, condivreg, ivreg2, weak instruments, endogenous tobit, endogenous probit, two-stage least squares, hypothesis testing, confidence intervals

## 1 Introduction

In this article, we present an indirect method for performing hypothesis testing based on the classical minimum distance approach. This method allows us to develop two extensions to the current set of weak-instrument robust tests that are available for linear instrumental-variables (IV) models. The first extension allows one to perform size-correct inference for a class of limited dependent variable (LDV) models that includes the endogenous tobit and probit models. The second extension allows size-correct inference with the linear IV model when dealing with covariance matrices with arbitrary heteroskedasticity or intracluster dependence.

There exists vast literature dealing with inference in the linear IV model when instruments are weak (see Stock, Wright, and Yogo [2002] for a review). When instruments are weak, point estimators are biased and Wald tests are unreliable. There are several tests available for linear IV models that have the correct size even when instruments are weak. These include the Anderson–Rubin (AR) statistic (Anderson and Rubin 1949), the Kleibergen–Moreira Lagrange multiplier (LM) test (Moreira 2003; Kleibergen 2007), the overidentification ($J$) test, and the conditional likelihood-ratio (CLR) test.

Concern about weak identification is not isolated to linear IV models. Identification issues also arise in the popular class of LDV models with endogenous explanatory variables. The endogenous tobit and endogenous probit models are two examples of these models (the `ivtobit` and `ivprobit` commands in Stata).

Extending the weak-instrument robust tests from the linear IV case to the LDV models is not straightforward. In the LDV models, the untested (nuisance) parameters are not separable from the structural parameters. As such, the orthogonal transformation that projects nuisance parameters out from the tests in the linear IV is not valid in the LDV case.

Fortunately, for this particular class of LDV models, the structural model also has a reduced-form representation. Consequently, inference on the structural parameter can be conducted indirectly by testing the restrictions on the reduced-form coefficients imposed by the underlying relationship between the structural and reduced-form parameters. Magnusson (2008a) describes this method of conducting inference under weak identification as the minimum distance approach. Our proposed tests for the endogenous variable coefficient have the correct size regardless of whether the identification condition holds.

Working with the reduced-form models also allows us to relax the homoskedastic assumption used in other implementations of the tests (e.g., the `condivreg` command of Moreira and Poi [2003] and Mikusheva and Poi [2006]). This is possible because the asymptotic behavior of our tests is derived from the reduced-form parameters estimator. In the linear IV model, this property allows us to use the heteroskedastic-robust variance–covariance matrix estimate as the reduced-form parameters covariance matrix. The same method allows us to deal with covariance matrices with cluster dependence. Some of these tests are asymptotically equivalent to those proposed by Chernozhukov and Hansen (2008), who also use a reduced-form approach.

Once we compute the statistical tests, we derive confidence intervals by inverting them. This guarantees that our confidence intervals have the correct coverage probability despite the instruments' strength or weakness. For the linear IV model under homoskedasticity, the existence of a closed-form solution for confidence intervals has been shown by Dufour (2003) for the AR test and by Mikusheva (2005) for the LM and CLR tests. However, their methods do not extend to nonlinear models or models with nonspherical residuals, so we use a grid search for estimating confidence intervals for the other models.

Because our tests are not model specific, we propose just one postestimation command for Stata, called `rivtest`. The command tests the simple composite hypothesis $H_0 : \beta = \beta_0$ against the alternative $H_a : \beta \neq \beta_0$ using five statistics: AR, LM, $J$, the combination of LM and $J$, and CLR. The command will also compute the confidence intervals based on these statistics. `rivtest` can be used after running `ivregress`, `ivreg2`, `ivprobit`, or `ivtobit` in Stata with one endogenous variable.

In the next section, we present a brief description of our tests. Then we present a general algorithm for implementing them. Next we discuss the command syntax of our postestimation command, `rivtest`, and provide examples of its use. Finally, we show results from Monte Carlo simulations we performed using the `rivtest` command.

# 2 Weak-instrument robust tests in LDV models: A minimum distance approach

## 2.1 Setup

We start by considering a class of models that includes both typical two-stage least-squares models and LDV models. Suppose there exists a model that satisfies the following structural form representation:

$$\begin{cases} y_i^* = x_i\beta + w_i\gamma + u_i \\ x_i = z_i\pi_z + w_i\pi_w + v_i \end{cases} \qquad \text{for} \quad i = 1, \ldots, n \tag{1}$$

where $y_i^*$ is a latent endogenous variable and $x_i$ is a continuously observed endogenous explanatory variable; $w_i$ and $z_i$ are, respectively, vectors of included and excluded instruments with dimensions $1 \times k_w$ and $1 \times k_z$; and the residuals $u_i$ and $v_i$ are independently distributed. Rather than observing $y_i^*$, we observe

$$y_i = f\left(y_i^*\right)$$

where $f$ is a known function. This representation is compatible with the class of LDV models in this study. For the endogenous tobit model, let $d_{lb}$ and $d_{ub}$ be, respectively, the lower and the upper bound. So, we have $y_i = d_{lb}$ if $y_i^* \leq d_{lb}$; $y_i = y_i^*$ if $d_{lb} < y_i^* < d_{ub}$; and $y_i = d_{ub}$ if $y_i^* \geq d_{ub}$. For the endogenous probit, we have $y_i = 0$ if $y_i^* \leq 0$ and $y_i = 1$ if $y_i^* > 0$. In particular, when $y_i = y_i^*$ we have the well-known linear IV model.

Our goal is to test $H_0 \colon \beta = \beta_0$ against $H_a \colon \beta \neq \beta_0$. However, whereas the coefficient $\gamma$ can be concentrated out of the linear IV model, this is not possible under a more general specification, so the available tests are inappropriate.

An unrestricted reduced-form model derived from (1) is

$$\begin{cases} y_i^* = z_i\delta_z + w_i\delta_w + \epsilon_i \\ x_i = z_i\pi_z + w_i\pi_w + v_i \end{cases} \tag{2}$$

where $\epsilon_i = v_i\beta + u_i$. The restrictions imposed by the structural model over the reduced-form parameters give us the following relation:

$$\delta_z = \pi_z\beta \tag{3}$$

We use (3) to develop our tests on the structural parameter, $\beta$, based on the unrestricted model (2). In this representation, the global identification of $\beta$ requires that $\| \pi_z \| \neq 0$. So as $\pi_z$ approaches zero, the instruments become weaker.

For now, let's assume that $\delta_z$ and $\pi_z$ are consistently estimated by $\widehat{\delta}_z$ and $\widehat{\pi}_z$. Let's also assume that $\Lambda$, the asymptotic variance–covariance of $\sqrt{n}\left[(\widehat{\delta}_z - \delta_z)', \quad (\widehat{\pi}_z - \pi_z)'\right]'$, is also consistently estimated by

$$\widehat{\Lambda} = \begin{bmatrix} \widehat{\Lambda}_{\delta_z\delta_z} & \widehat{\Lambda}_{\delta_z\pi_z} \\ \widehat{\Lambda}_{\pi_z\delta_z} & \widehat{\Lambda}_{\pi_z\pi_z} \end{bmatrix}$$

Let's introduce two more statistics:

$$\begin{aligned}\widehat{\Psi}_\beta &= \widehat{\Lambda}_{\delta_z\delta_z} - \beta\widehat{\Lambda}_{\delta_z\pi_z} - \beta\widehat{\Lambda}_{\pi_z\delta_z} + (\beta)^2\widehat{\Lambda}_{\pi_z\pi_z} \\ \widehat{\overline{\pi}}_\beta &= \widehat{\pi}_z - \left(\widehat{\Lambda}_{\pi_z\delta_z} - \beta\widehat{\Lambda}_{\pi_z\pi_z}\right)\widehat{\Psi}_\beta^{-1}\left(\widehat{\delta}_z - \widehat{\pi}_z\beta\right)\end{aligned}$$

The first statistic is an estimate of the asymptotic covariance matrix of $\sqrt{n}(\widehat{\delta}_z - \widehat{\pi}_z\beta)$. The second statistic is an estimate of $\pi_z$, whose properties are discussed in Magnusson (2008a).

## 2.2 Weak-instrument robust tests

Under $H_0\colon \beta = \beta_0$, our version of the AR test is

$$\begin{aligned}\text{AR}_{\text{MD}}(\beta_0) &= n\left(\widehat{\delta}_z - \widehat{\pi}_z\beta_0\right)'\widehat{\Psi}_{\beta_0}^{-1}\left(\widehat{\delta}_z - \widehat{\pi}_z\beta_0\right) \\ &\xrightarrow{d} \chi^2(k_z)\end{aligned}$$

where the value inside the parentheses indicates the chi-squared distribution degrees of freedom. Then we reject $H_0$ at significance level $\alpha$ if $\text{AR}_{\text{MD}}(\beta_0)$ is greater than the $1-\alpha$ percentile of the $\chi^2(k_z)$ distribution.

The $\text{AR}_{\text{MD}}$ statistic simultaneously tests the value of the structural parameter and the overidentification restriction. We can make an orthogonal decomposition of the $\text{AR}_{\text{MD}}$ test into two statistics, namely, the $\text{LM}_{\text{MD}}$ and $J_{\text{MD}}$ tests. Under the null hypothesis, the $\text{LM}_{\text{MD}}$ statistic tests the value of the structural parameter given that the overidentification condition holds, while the $J_{\text{MD}}$ statistic tests the overidentification restriction given the value of $\beta_0$. They are

$$\text{LM}_{\text{MD}}(\beta_0) = n\left\{\widehat{\Psi}_{\beta_0}^{-\frac{1}{2}}\left(\widehat{\delta}_z - \widehat{\pi}_z\beta_0\right)\right\}'\widehat{P}_{\beta_0}\left\{\widehat{\Psi}_{\beta_0}^{-\frac{1}{2}}\left(\widehat{\delta}_z - \widehat{\pi}_z\beta_0\right)\right\} \tag{4}$$

$$J_{\text{MD}}(\beta_0) = n\left\{\widehat{\Psi}_{\beta_0}^{-\frac{1}{2}}\left(\widehat{\delta}_z - \widehat{\pi}_z\beta_0\right)\right\}'\widehat{M}_{\beta_0}\left\{\widehat{\Psi}_{\beta_0}^{-\frac{1}{2}}\left(\widehat{\delta}_z - \widehat{\pi}_z\beta_0\right)\right\} \tag{5}$$

where

$$\begin{aligned}\widehat{P}_{\beta_0} &= \frac{\left(\widehat{\Psi}_{\beta_0}^{-\frac{1}{2}}\widehat{\overline{\pi}}_{\beta_0}\right)\left(\widehat{\Psi}_{\beta_0}^{-\frac{1}{2}}\widehat{\overline{\pi}}_{\beta_0}\right)'}{\left(\widehat{\overline{\pi}}'_{\beta_0}\widehat{\Psi}_{\beta_0}^{-1}\widehat{\overline{\pi}}_{\beta_0}\right)} \\ \widehat{M}_{\beta_0} &= I_{k_z} - \widehat{P}_{\beta_0}\end{aligned}$$

and $I_{k_z}$ is a $k_z \times k_z$ identity matrix. Assuming that some regularity conditions hold under the null hypothesis, we have

$$\begin{aligned}\text{LM}_{\text{MD}}(\beta_0) &\xrightarrow{d} \chi^2(1) \\ J_{\text{MD}}(\beta_0) &\xrightarrow{d} \chi^2(k_z - 1)\end{aligned}$$

independent of whether the instruments are weak (see Magnusson [2008a] for more details). From (4) and (5), we have

$$\text{AR}_{\text{MD}} = \text{LM}_{\text{MD}} + J_{\text{MD}}$$

It is well-known that the $\text{LM}_{\text{MD}}$ test suffers from a spurious decline of power at some regions of the parameter space. In those regions, the $J_{\text{MD}}$ test approximates the $\text{AR}_{\text{MD}}$ test, which always has discriminatory power. We combine the $\text{LM}_{\text{MD}}$ and $J_{\text{MD}}$ tests to rule out regions where the $\text{LM}_{\text{MD}}$ test behaves spuriously. For example, testing $H_0 : \beta = \beta_0$ at the 5% significance level could be performed by testing the null at the 4% significance level with the $\text{LM}_{\text{MD}}$ test and at the 1% significance level with the $J_{\text{MD}}$ test. We reject the null if either $K_{\text{MD}}$ or $J_{\text{MD}}$ is rejected. We call this combination test the LM-$\text{J}_{\text{MD}}$ test.

The minimum distance version of Moreira's (2003) conditional likelihood-ratio test is

$$\begin{aligned}\text{CLR}_{\text{MD}}(\beta_0) =& \frac{1}{2}\Big[\text{AR}_{\text{MD}}(\beta_0) - \text{rk}(\beta_0) + \\ & \sqrt{\{\text{AR}_{\text{MD}}(\beta_0) + \text{rk}(\beta_0)\}^2 - 4J_{\text{MD}}(\beta_0)\text{rk}(\beta_0)}\Big]\end{aligned}$$

where

$$\begin{aligned}\text{rk}(\beta_0) =& \; n\left(\widehat{\overline{\pi}}'_{\beta_0}\widehat{\Xi}^{-1}_{\beta_0}\widehat{\overline{\pi}}_{\beta_0}\right) \\ \widehat{\Xi}_{\beta_0} =& \widehat{\Lambda}_{\pi_z\pi_z} - \left(\widehat{\Lambda}_{\pi_z\delta_z} - \beta_0\widehat{\Lambda}_{\pi_z\pi_z}\right)\widehat{\Psi}^{-1}_{\beta_0}\left(\widehat{\Lambda}_{\delta_z\pi_z} - \beta_0\widehat{\Lambda}_{\pi_z\pi_z}\right)\end{aligned}$$

The asymptotic distribution of the $\text{CLR}_{\text{MD}}$ is not pivotal and depends on $\text{rk}(\beta_0)$. The critical values of this test are calculated by simulating independent values of $\chi^2(1)$ and $\chi^2(k_z-1)$ for a given value of $\text{rk}(\beta_0)$. This approach is not satisfactory because accuracy demands many simulations, which can be computationally intensive. For linear IV models under homoskedasticity, Andrews, Moreira, and Stock (2007) provide a formula for computing the $p$-value function of the CLR test (which is embedded in the `condivreg` command). Although this is not the correct $p$-value function when homoskedasticity is violated, our simulations indicate that it provides a good approximation.

Two Stata packages currently provide some functionality to perform these tests. For the linear IV case under homoskedastic residuals, the `condivreg` command in Stata provides a set of weak-instrument robust tests (Moreira and Poi 2003; Mikusheva and Poi 2006). Our command, `rivtest`, complements `condivreg` by offering weak-instrument robust tests for a larger class of models. For nonhomoskedastic residuals, Baum, Schaffer, and Stillman (2007) provide the AR test in the `ivreg2` package. The degrees of freedom of the AR test depends on the number of instruments and not on the number of endogenous variables, so its power decreases as one increases the number of instruments. We complement this package by offering a set of tests that are valid even with many instruments.

## 2.3 Confidence intervals

Confidence intervals for the proposed tests are derived by inverting the statistical tests. By definition, confidence intervals derived from the $\text{AR}_{\text{MD}}$, $\text{LM}_{\text{MD}}$, $\text{LM}_{\text{MD}}$–$J_{\text{MD}}$, and $\text{CLR}_{\text{MD}}$ tests are, respectively,

$$\begin{aligned}
C^{\text{AR}_{\text{MD}}}_{(1-\tau)} &= \left\{\beta_0 : \text{AR}_{\text{MD}}(\beta_0) < \chi^2_{1-\tau,k_z}\right\} \\
C^{\text{LM}_{\text{MD}}}_{(1-\tau)} &= \left\{\beta_0 : \text{LM}_{\text{MD}}(\beta_0) < \chi^2_{1-\tau,1}\right\} \\
C^{\text{LM}_{\text{MD}}-J_{\text{MD}}}_{(1-\tau)} &= \left\{\beta_0 : \left[\text{LM}_{\text{MD}}(\beta_0) < \chi^2_{1-w_1\tau,1}\right] \cap \right. \\
&\qquad \left. \left[J_{\text{MD}}(\beta_0) < \chi^2_{1-w_2\tau,k_z-1}\right]\right\} \\
C^{\text{CLR}_{\text{MD}}}_{(1-\tau)} &= \left\{\beta_0 : \text{CLR}_{\text{MD}}(\beta_0) < c\left[\text{rk}(\beta_0)\right]\right\}
\end{aligned}$$

where $\tau$ denotes the significance level, $w_1 + w_2 = 1$, and $c\{\text{rk}(\beta_0)\}$ is the 95th percentile of the distribution of the $\text{CLR}_{\text{MD}}$ tests conditional on the value of $\text{rk}(\beta_0)$.

The weak instrument robust confidence intervals are not necessarily convex or symmetric as is the usual Wald-type confidence interval, which includes points two standard deviations from the estimated coefficient. For example, they can be a union of disjoint intervals or the real line when the instruments are completely irrelevant. The $\text{AR}_{\text{MD}}$ confidence interval can be empty. This occurs when the overidentifying restriction is rejected for any value of $\beta$. However, the $\text{LM}_{\text{MD}}$ and $\text{CLR}_{\text{MD}}$ confidence intervals are never empty because the continuous updating minimum distance estimate always belongs to them.[1]

Dufour (2003) and Mikusheva (2005) provide closed-form solutions for obtaining confidence intervals in the homoskedastic linear IV model. In particular, Mikusheva, by solving quadratic inequalities, proposes a numerically simple algorithm for estimating confidence intervals derived from the $\text{LM}_{\text{MD}}$ and $\text{CLR}_{\text{MD}}$ tests. However, their methods are not generalized to either nonspherical residuals or models with LDV. We employ their solutions for the homoskedastic linear IV model. In the other models, we use the grid search method for generating the confidence intervals by testing points in the parameter space. Points $\overline{\beta}$ for which $H_0 : \beta = \overline{\beta}$ is not rejected belong in the confidence interval. The user has the option to choose the interval and the number of points in the grid search. For the LM-$\text{J}_{\text{MD}}$ test, the user can select the weight, $w_1$, given to the $\text{LM}_{\text{MD}}$. The default option is $w_1 = 0.8$.

1. The continuous updating minimum distance estimate is the value that minimizes the $\text{AR}_{\text{MD}}$ test. It is not numerically equal to the generalized method of moments continuous updating estimate of Hansen, Heaton, and Yaron (1996).

# 3 Generic algorithm for implementing minimum distance weak-instrument robust tests

The implementation of our weak-instrument robust tests takes advantage of several built-in functions of Stata. We separate our implementation into two cases: one in which residuals are homoskedastic and another in which residuals have either arbitrary heteroskedasticity or intracluster dependence.

## 3.1 Homoskedastic residuals

Under a homoskedastic assumption, we use the fact that $u_i = v_i\alpha + \varepsilon_i$, where $\alpha = \sigma_{vu}^{-1}\sigma_{vv}$. This condition is suitable, for example, if residuals are jointly normally distributed. Moreover, the assumption allows the computation of the tests by using built-in functions available in Stata (Magnusson 2008b). The reduced-form (2) becomes

$$\begin{cases} y_i^* = z_i\delta_z + w_i\delta_w + v_i\delta_v + \varepsilon_i \\ x_i = z_i\pi_z + w_i\pi_w + v_i \end{cases} \tag{6}$$

In the above representation, $\varepsilon_i$ and $v_i$ are independent by construction. The test algorithm has the following steps:

1. Estimate $\pi_z$ and $\Lambda_{\pi_z\pi_z}$ by ordinary least squares (OLS). Denote the estimated values as $\widehat{\pi}_z$ and $\widehat{\Lambda}_{\pi_z\pi_z}$. Also compute the OLS estimated residuals:

   $$\widehat{v}_i = x_i - z_i\widehat{\pi}_z - w_i\widehat{\pi}_w$$

2. Estimate $\delta_z$ and $\delta_w$ by using the following equation:

   $$y_i^* = z_i\delta_z + w_i\delta_w + \widehat{v}_i\delta_v + \widetilde{\varepsilon}_i$$

   where $\widetilde{\varepsilon}_i = \varepsilon_i - (\widehat{v}_i - v_i)\delta_v$. Denote the estimated values of $\delta_z$, $\delta_w$, and $\delta_v$ as $\widehat{\delta}_z$, $\widehat{\delta}_w$, and $\widehat{\delta}_v$, respectively. For the endogenous probit model, our algorithm fixes $\sigma_{\varepsilon\varepsilon} = 1$ for normalization, which is a different normalization than the default option in Stata ($\sigma_{uu} = 1$) but the same as the Newey two-step estimator (see [R] **ivprobit**).

3. Save $\widehat{\Gamma}_{\delta_z\delta_z}$, the output of the variance–covariance matrix estimate of $\widehat{\delta}_z$. This is not the "correct" variance–covariance of $\widehat{\delta}_z$ because we are not adjusting for the presence of $\widehat{v}_i$.

Using the same notation as in the body of the text, we have

$$\begin{aligned} \widehat{\Psi}_\beta &= \widehat{\Gamma}_{\delta_z\delta_z} + (\widehat{\delta}_v - \beta)^2\widehat{\Lambda}_{\pi_z\pi_z} \\ \widehat{\overline{\pi}}_\beta &= \widehat{\pi}_z - (\widehat{\delta}_v - \beta)^2\widehat{\Psi}_\beta^{-1}\widehat{\Lambda}_{\pi_z\pi_z} \\ \widehat{\Xi}_{\beta_0} &= \widehat{\Lambda}_{\pi_z\pi_z} - (\widehat{\delta}_v - \beta)^2\widehat{\Lambda}_{\pi_z\pi_z}\widehat{\Psi}_\beta^{-1}\widehat{\Lambda}_{\pi_z\pi_z} \end{aligned}$$

### 3.2 Heteroskedastic/clustered residuals

For heteroskedasticity or cluster dependence in the distribution of errors, we consider just the linear model. Baum, Schaffer, and Stillman (2007) provide an option using a generalized method of moments approach for autocorrelation- and heteroskedasticity-robust AR tests in the `ivreg2` command. We extend this functionality for the $\text{LM}_{\text{MD}}$, $\text{LM-J}_{\text{MD}}$, and $\text{CLR}_{\text{MD}}$ tests.

The implementation is similar to the homoskedastic case. The reduced-form model is

$$\begin{cases} y_i = z_i\delta_z + w_i\delta_w + e_i \\ x_i = z_i\pi_z + w_i\pi_w + v_i \end{cases}$$

We estimate the $\delta_z$, $\pi_z$, $\Lambda_{\delta_z\delta_z}$, and $\Lambda_{\pi_z\pi_z}$ by running two separate regressions with the appropriate robust or cluster options. The covariance term $\Lambda_{\pi_z\delta_z}$ has the general sandwich formula

$$\widehat{\Lambda}_{\pi_z\delta_z} = A\ B\ A'$$

where $A = (Z^{\perp\prime}Z^{\perp})^{-1}$ is a $k_z \times k_z$ matrix, $Z^{\perp} = M_W Z$, and $M_W = I_n - W(W'W)^{-1}W'$, the matrix that projects $Z$ to the orthogonal space spanned by $W$. Let's denote $\widehat{v}$ and $\widehat{e}$ as the vectors of OLS residuals. The $B$ matrix is given by:

$$\sum_j z_j^{\perp\prime}\widehat{v}_j\widehat{e}_j' z_j^{\perp}$$

For robust standard errors, $z_j^{\perp}$ is a $k_z \times 1$ vector, and $\widehat{v}_j$ and $\widehat{u}_j$ are scalars. For clustered standard errors, $z_j^{\perp}$ is a $k_z \times n_j$ matrix, and $\widehat{v}_j$ and $\widehat{u}_j$ are $n_j \times 1$ vectors, where $n_j$ is the number of observations in cluster $j$.

The tests obtained here and by Chernozhukov and Hansen (2008) are closely related. They work with the following regression model:

$$y_i - Y_i\beta = Z_i\gamma + u_i \tag{7}$$

A simple $t$ test, $\widehat{\gamma}/s_{\widehat{\gamma}}$, is the same as testing $H_0 : \beta = \beta_0$, where $\widehat{\gamma}$ is the OLS estimator derived from (7) replacing $\beta$ with $\beta_0$. Our AR test and the AR test of Chernozhukov and Hansen (2008) are identical. Our LM test, however, is only asymptotically equivalent to theirs; they are slightly different in small samples.[2]

## 4 The rivtest command

The software package accompanying this article contains a Stata command, `rivtest`, to implement the tests discussed above after using the `ivregress`, `ivreg2`, `ivprobit`, or `ivtobit` command.

2. A proof is available upon request.

## 4.1 Command description

For `ivregress` and `ivreg2`, `rivtest` supports limited-information maximum likelihood and two-stage least-squares models (the `liml` and `2sls` options of `ivregress`, respectively), as well as `vce(robust)` and `vce(cluster` *clustvar*`)` options for variance–covariance estimation. For `ivprobit` and `ivtobit`, `rivtest` supports all variance–covariance estimation options except the `vce(robust)` and `vce(cluster` *clustvar*`)` options. Weights are allowed as long as they are supported by the appropriate IV command.

`rivtest` calculates the minimum distance version of the AR test statistic. When the IV model contains more than one instrumental variable, `rivtest` also conducts the minimum distance versions of the CLR test, the LM test, the $J$ overidentification test, and a combination of the LM multiplier and overidentification tests (LM-J). As a reference, `rivtest` also presents the Wald test.

The AR test is a joint test of the structural parameter and the overidentification restrictions. The AR statistic can be decomposed into the LM statistic, which tests only the structural parameter, and the $J$ statistic, which tests only the overidentification restrictions. (This $J$ statistic, evaluated at the null hypotheses, is different from the Hansen $J$ statistic, which is evaluated at the parameter estimate.) The LM test loses power in some regions of the parameter space when the likelihood function has a local extrema or inflection. In the linear IV model with homoskedasticity, the CLR statistic combines the LM statistic and the $J$ statistic in the most efficient way, thereby testing both the structural parameter and the overidentification restrictions simultaneously. The LM-J combination test is another approach for testing the hypotheses simultaneously. It is more efficient than the AR test and allows different weights to be put on the parameter and overidentification hypotheses. The CLR test is the most powerful test for the linear model under homoskedasticity (within a class of invariant similar tests), but this result has not been proven yet for other IV-type estimators, so we present all test results.

`rivtest` can also estimate confidence intervals based on the AR, CLR, LM, and LM-J tests. With `ivregress` there is a closed-form solution for these confidence intervals only when homoskedasticity is assumed. More generally, `rivtest` estimates confidence intervals through test inversion over a grid. The default grid is twice the size of the confidence interval based on the Wald test. As a reference, `rivtest` also presents the Wald confidence interval.

## 4.2 Syntax

The following is the command syntax for `rivtest`:

```
rivtest [ , null(#) lmwt(#) small ci grid(numlist) points(#)
   gridmult(#) usegrid retmat level(#) ]
```

## 4.3 Options

The options for `rivtest` relate to testing and confidence-interval estimation.

### Testing options

`null(#)` specifies the null hypothesis for the coefficient on the endogenous variable in the IV model. The default is `null(0)`.

`lmwt(#)` is the weight put on the LM test statistic in the LM-J test. The default is `lmwt(0.8)`.

`small` specifies that small-sample adjustments be made when test statistics are calculated. The default is given by whatever small-sample adjustment option was chosen in the IV command.

### Confidence-interval options

`ci` requests that confidence intervals be estimated. By default, these are not estimated because grid-based test inversion can be time intensive.

`grid(`*numlist*`)` specifies the grid points over which to calculate the confidence sets. The default grid is centered around the point estimate with a width equal to twice the Wald confidence interval. That is, if $\widehat{\beta}$ is the estimated coefficient on the endogenous variable, $\widehat{\sigma}_\beta$ is its estimated standard error, and $1-\alpha$ is the confidence level, then the default endpoints of the interval over which confidence sets will be calculated are $\widehat{\beta} \pm 2z_{\alpha/2}\widehat{\sigma}_\beta$. With weak instruments, this is often too small of a grid to estimate the confidence intervals. `grid(`*numlist*`)` may not be used with the other two grid options: `points(#)` and `gridmult(#)`. If one of the other options is used, only input from `grid(`*numlist*`)` will be used to construct the grid.

`points(#)` specifies the number of equally spaced values over which to calculate the confidence sets. The default is `points(100)`. Increasing the number of grid points will increase the time required to estimate the confidence intervals, but a greater number of grid points will improve precision.

`gridmult(#)` is another way of specifying a grid to calculate confidence sets. This option specifies that the grid be # times the size of the Wald confidence interval. The default is `gridmult(2)`.

`usegrid` forces grid-based test inversion for confidence-interval estimation under the homoskedastic linear IV model. The default is to use the analytic solution. Under the other models, grid-based estimation is the only method.

`retmat` returns a matrix of test results over the confidence-interval search grid. This matrix can be large if the number of grid points is large, but it can be useful for graphing confidence sets.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`. Because the LM-J test has no $p$-value function, we report whether the test is rejected. Changing `level(#)` also changes the level of significance used to determine this result: [100-`level(#)`]%.

### 4.4 Saved results

`rivtest` saves the following in `r()`:

Scalars

| | |
|---|---|
| `r(null)` | null hypothesis |
| `r(clr_p)` | CLR test $p$-value |
| `r(clr_stat)` | CLR test statistic |
| `r(ar_p)` | AR test $p$-value |
| `r(ar_chi2)` | AR test statistic |
| `r(lm_p)` | LM test $p$-value |
| `r(lm_chi2)` | LM test statistic |
| `r(j_p)` | $J$ test $p$-value |
| `r(j_chi2)` | $J$ test statistic |
| `r(lmj_r)` | LM-J test rejection indicator |
| `r(rk)` | rk statistic |
| `r(wald_p)` | Wald test $p$-value |
| `r(wald_chi2)` | Wald test statistic |
| `r(points)` | number of points in grid used to estimate confidence sets |

Macros

| | |
|---|---|
| `r(clr_cset)` | confidence set based on CLR test |
| `r(ar_cset)` | confidence set based on AR test |
| `r(lm_cset)` | confidence set based on LM test |
| `r(lmj_cset)` | confidence set based on LM-J test |
| `r(wald_cset)` | confidence set based on Wald test |
| `r(inexog)` | list of instruments included in the second-stage equation |
| `r(exexog)` | list of instruments excluded from the second-stage equation |
| `r(endo)` | endogenous variable |
| `r(grid)` | range of grid used to estimate confidence sets |

Matrices

| | |
|---|---|
| `r(citable)` | table with test statistics, $p$-values, and rejection indicators for every grid point over which hypothesis was tested |

## 5 Examples: Married female labor market participation

We demonstrate the use of the `rivtest` command in a set of applications with the data from Mroz (1987), available from the Stata web site at http://www.stata.com/data/jwooldridge/eacsap/mroz.dta. These examples are related to married female labor supply and illustrate the differences between robust and non-robust inference when instruments are potentially weak.

### 5.1 Example 1: Two-stage least squares with unknown heteroskedasticity

In this example, we fit a two-stage least-squares model with Stata's `ivregress` command using the robust variance–covariance estimation option to account for arbitrary

heteroskedasticity. We regress working hours (`hours`), on log wages (`lwage`), other household income in logs (`nwifeinc`), years of education (`educ`), number of children less than 6 years old (`kidslt6`), and the number of children at least 6 years old (`kidsge6`). As instruments for the wage, we use labor market experience (`exper`) and its square (`expersq`), and father's and mother's years of education (`fatheduc` and `motheduc`). We consider the subsample of women who are participating in the labor market and have strictly positive wages.

```
. use http://www.stata.com/data/jwooldridge/eacsap/mroz.dta
. ivregress 2sls hours nwifeinc educ age kidslt6 kidsge6 (lwage = exper expersq
> fatheduc motheduc) if inlf==1 , first vce(robust)
First-stage regressions
```

```
                                        Number of obs   =        428
                                        F(   9,    418) =      10.78
                                        Prob > F        =     0.0000
                                        R-squared       =     0.1710
                                        Adj R-squared   =     0.1532
                                        Root MSE        =     0.6655
```

| lwage | Coef. | Robust Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| nwifeinc | .0057445 | .0027375 | 2.10 | 0.036 | .0003636 | .0111255 |
| educ | .1127654 | .0154679 | 7.29 | 0.000 | .0823609 | .1431699 |
| age | -.0053092 | .0063134 | -0.84 | 0.401 | -.0177191 | .0071007 |
| kidslt6 | -.066367 | .103709 | -0.64 | 0.523 | -.2702231 | .137489 |
| kidsge6 | -.0192837 | .0292029 | -0.66 | 0.509 | -.0766866 | .0381191 |
| exper | .0404503 | .0151505 | 2.67 | 0.008 | .0106697 | .0702309 |
| expersq | -.0007512 | .0004056 | -1.85 | 0.065 | -.0015485 | .000046 |
| fatheduc | -.0061784 | .0106541 | -0.58 | 0.562 | -.0271208 | .0147639 |
| motheduc | -.016405 | .0119691 | -1.37 | 0.171 | -.039932 | .0071221 |
| _cons | -.2273025 | .3343392 | -0.68 | 0.497 | -.8844983 | .4298933 |

```
Instrumental variables (2SLS) regression            Number of obs =     428
                                                    Wald chi2(6)  =   18.22
                                                    Prob > chi2   =  0.0057
                                                    R-squared     =       .
                                                    Root MSE      =  1143.2
```

| hours | Coef. | Robust Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| lwage | 1265.326 | 473.6747 | 2.67 | 0.008 | 336.9408 | 2193.711 |
| nwifeinc | -8.353995 | 4.57849 | -1.82 | 0.068 | -17.32767 | .6196797 |
| educ | -148.2865 | 54.38669 | -2.73 | 0.006 | -254.8824 | -41.69053 |
| age | -10.23769 | 9.299097 | -1.10 | 0.271 | -28.46358 | 7.98821 |
| kidslt6 | -234.3907 | 181.9979 | -1.29 | 0.198 | -591.1001 | 122.3187 |
| kidsge6 | -59.62672 | 49.24854 | -1.21 | 0.226 | -156.1521 | 36.89865 |
| _cons | 2375.395 | 535.4835 | 4.44 | 0.000 | 1325.867 | 3424.923 |

```
Instrumented:  lwage
Instruments:   nwifeinc educ age kidslt6 kidsge6 exper
               expersq fatheduc motheduc
```

```
. rivtest, ci grid(-1000(10)8000)
Estimating confidence sets over grid points
----+--- 1 ----+--- 2 ----+--- 3 ----+--- 4 ----+--- 5
..................................................   50
..................................................  100
..................................................  150
..................................................  200
..................................................  250
..................................................  300
..................................................  350
..................................................  400
..................................................  450
..................................................  500
..................................................  550
..................................................  600
..................................................  650
..................................................  700
..................................................  750
..................................................  800
..................................................  850
..................................................  900
.
Weak instrument robust tests and confidence sets for linear IV with robust VCE
H0: beta[hours:lwage] = 0
```

| Test | Statistic | p-value | 95% Confidence Set |
|---|---|---|---|
| CLR | stat(.) = 27.27 | Prob > stat = 0.0000 | [ 810, 5330] |
| AR | chi2(4) = 32.61 | Prob > chi2 = 0.0000 | [ 770, 6930] |
| LM | chi2(1) = 21.22 | Prob > chi2 = 0.0000 | |
| | [ -830, -670] U [ 790, 5460] | | |
| J | chi2(3) = 11.39 | Prob > chi2 = 0.0098 | |
| LM-J | H0 rejected at 5% level | | [ 760, 5940] |
| Wald | chi2(1) = 7.14 | Prob > chi2 = 0.0076 | [ 336.941, 2193.71] |

```
Note: Wald test not robust to weak instruments. Confidence sets estimated for
901 points in [-1000,8000].
```

The confidence intervals derived from weak-instrument robust tests are wider than the Wald confidence interval, indicating that instruments are not strong and that point estimates are biased. The negative values of the LM confidence set are discarded in the LM-J confidence interval, indicating the spurious behavior of the LM test in that part of the parameter space. The above result suggests a positive effect of wages on the labor supply, but `rivtest` is unable to predict the magnitude of the effect.

## 5.2 Example 2: Endogenous probit

Next we fit a model of labor force participation for the married women in the sample. The binary variable `inlf` equals one if the woman is in the labor market and zero otherwise. The endogenous explanatory variable is nonwife household income, which is instrumented by husband's hours of work (`hushrs`), father's education, mother's education, and the county-level unemployment rate (`unem`). As exogenous variables,

we include education, years of labor market experience, experience squared, number of children less than 6 years old, number of children at least 6 years old, and a dummy for whether the individual lives in a metropolitan area (`city`).

```
. ivprobit inlf educ exper expersq kidslt6 kidsge6 city (nwifeinc = hushrs
> fatheduc motheduc unem), twostep first
Checking reduced-form model...
First-stage regression
```

| Source | SS | df | MS | | |
|---|---|---|---|---|---|
| Model | 18057.3855 | 10 | 1805.73855 | | |
| Residual | 83739.7301 | 742 | 112.856779 | | |
| Total | 101797.116 | 752 | 135.368505 | | |

| | | | | | |
|---|---|---|---|---|---|
| Number of obs | = | 753 | | | |
| F( 10, 742) | = | 16.00 | | | |
| Prob > F | = | 0.0000 | | | |
| R-squared | = | 0.1774 | | | |
| Adj R-squared | = | 0.1663 | | | |
| Root MSE | = | 10.623 | | | |

| nwifeinc | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| hushrs | .0029782 | .0006719 | 4.43 | 0.000 | .0016591 | .0042972 |
| fatheduc | .1760206 | .1385697 | 1.27 | 0.204 | -.0960147 | .4480558 |
| motheduc | -.1395621 | .1458037 | -0.96 | 0.339 | -.425799 | .1466749 |
| unem | .1652976 | .1283373 | 1.29 | 0.198 | -.0866498 | .417245 |
| educ | 1.218966 | .2011015 | 6.06 | 0.000 | .8241703 | 1.613762 |
| exper | -.3562876 | .1406571 | -2.53 | 0.012 | -.632421 | -.0801543 |
| expersq | .0031554 | .0045229 | 0.70 | 0.486 | -.0057239 | .0120346 |
| kidslt6 | -.3788863 | .7624489 | -0.50 | 0.619 | -1.8757 | 1.117928 |
| kidsge6 | -.1729039 | .3105805 | -0.56 | 0.578 | -.782625 | .4368172 |
| city | 4.949449 | .8419922 | 5.88 | 0.000 | 3.296478 | 6.602419 |
| _cons | -2.916913 | 2.883583 | -1.01 | 0.312 | -8.577865 | 2.744039 |

Two-step probit with endogenous regressors

| | | |
|---|---|---|
| Number of obs | = | 753 |
| Wald chi2(7) | = | 136.69 |
| Prob > chi2 | = | 0.0000 |

| | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| nwifeinc | -.0631912 | .0292417 | -2.16 | 0.031 | -.1205038 | -.0058785 |
| educ | .2148807 | .0473224 | 4.54 | 0.000 | .1221304 | .307631 |
| exper | .1067194 | .0225831 | 4.73 | 0.000 | .0624574 | .1509813 |
| expersq | -.0022201 | .0006423 | -3.46 | 0.001 | -.003479 | -.0009611 |
| kidslt6 | -.5794973 | .1113274 | -5.21 | 0.000 | -.797695 | -.3612996 |
| kidsge6 | .1284411 | .0429235 | 2.99 | 0.003 | .0443126 | .2125696 |
| city | .1421479 | .1805589 | 0.79 | 0.431 | -.2117411 | .4960368 |
| _cons | -2.038166 | .3551659 | -5.74 | 0.000 | -2.734279 | -1.342054 |

```
Instrumented:  nwifeinc
Instruments:   educ exper expersq kidslt6 kidsge6 city
               hushrs fatheduc motheduc unem
```

Wald test of exogeneity: chi2(1) = 3.05 Prob > chi2 = 0.0808

(*Continued on next page*)

```
. rivtest, ci grid(-.2(.001).6)
Estimating confidence sets over grid points
----+--- 1 ----+--- 2 ----+--- 3 ----+--- 4 ----+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
..................................................   250
..................................................   300
..................................................   350
..................................................   400
..................................................   450
..................................................   500
..................................................   550
..................................................   600
..................................................   650
..................................................   700
..................................................   750
..................................................   800
.
Weak instrument robust tests and confidence sets for IV probit
H0: beta[inlf:nwifeinc] = 0
```

| Test | Statistic | p-value | 95% Confidence Set |
|---|---|---|---|
| CLR | stat(.) = 5.82 | Prob > stat = 0.0249 | [ -.172, -.01] |
| AR | chi2(4) = 9.50 | Prob > chi2 = 0.0498 | [ -.197, -.001] |
| LM | chi2(1) = 4.75 | Prob > chi2 = 0.0293 | |
| | [ -.177, -.008] U [ .17, .534] | | |
| J | chi2(3) = 4.75 | Prob > chi2 = 0.1913 | |
| LM-J | H0 rejected at 5% level | | [ -.186, -.005] |
| Wald | chi2(1) = 4.67 | Prob > chi2 = 0.0307 | [-.120504,-.005879] |

```
Note: Wald test not robust to weak instruments. Confidence sets estimated for
801 points in [-.2,.6].
```

In the endogenous probit model, the `rivtest` command uses the normalization of Newey's minimum chi-squared estimator, $\sigma_\varepsilon = 1$ in (6), which is different from the default normalization used in maximum likelihood estimation, $\sigma_u = 1$ in (1) (see [R] **ivprobit** for further explanation). Therefore, the confidence intervals produced by `rivtest` and the maximum likelihood version of `ivprobit` are not comparable.

In this example, although one instrument, husband's hours of work, has a first-stage $t$ statistic greater than 4, the confidence intervals produced from the weak-instrument tests are significantly larger than the nonrobust Wald confidence interval; for example, the LM-J confidence interval is 50% larger than the Wald confidence interval. Thus the presence of only one strong instrument in the first stage among other weaker ones does not imply that classical inference is correct.

## 5.3 Example 3: Endogenous tobit

In the following example, we fit an endogenous tobit model with Stata's `ivtobit` command. We regress hours of work, including the many observations in which the woman does not supply labor, on the same regressors as in the previous example.

```
. ivtobit hours educ exper expersq kidslt6 kidsge6 city (nwifeinc = hushrs
> fatheduc motheduc unem), ll(0) first nolog
Tobit model with endogenous regressors           Number of obs   =        753
                                                 Wald chi2(7)    =     173.12
Log likelihood = -6686.3386                      Prob > chi2     =     0.0000
```

| | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| hours | | | | | | |
| nwifeinc | -71.02316 | 33.59912 | -2.11 | 0.035 | -136.8762 | -5.170087 |
| educ | 183.002 | 51.47165 | 3.56 | 0.000 | 82.11939 | 283.8846 |
| exper | 121.0376 | 23.65995 | 5.12 | 0.000 | 74.66493 | 167.4102 |
| expersq | -2.478807 | .623252 | -3.98 | 0.000 | -3.700358 | -1.257255 |
| kidslt6 | -639.99 | 116.5606 | -5.49 | 0.000 | -868.4446 | -411.5353 |
| kidsge6 | 74.23684 | 41.79029 | 1.78 | 0.076 | -7.670611 | 156.1443 |
| city | 187.9859 | 194.1849 | 0.97 | 0.333 | -192.6095 | 568.5814 |
| _cons | -1436.843 | 351.9196 | -4.08 | 0.000 | -2126.593 | -747.0931 |
| nwifeinc | | | | | | |
| educ | 1.284978 | .198927 | 6.46 | 0.000 | .8950883 | 1.674868 |
| exper | -.368858 | .1399175 | -2.64 | 0.008 | -.6430913 | -.0946248 |
| expersq | .0033886 | .0044934 | 0.75 | 0.451 | -.0054183 | .0121955 |
| kidslt6 | -.3558916 | .75725 | -0.47 | 0.638 | -1.840074 | 1.128291 |
| kidsge6 | -.1665826 | .308437 | -0.54 | 0.589 | -.771108 | .4379429 |
| city | 4.833468 | .8349314 | 5.79 | 0.000 | 3.197033 | 6.469904 |
| hushrs | .0027375 | .0007263 | 3.77 | 0.000 | .001314 | .0041611 |
| fatheduc | .1481241 | .1277639 | 1.16 | 0.246 | -.1022886 | .3985368 |
| motheduc | -.2084148 | .1309959 | -1.59 | 0.112 | -.465162 | .0483325 |
| unem | .2506685 | .1163957 | 2.15 | 0.031 | .022537 | .4787999 |
| _cons | -2.883293 | 2.871029 | -1.00 | 0.315 | -8.510407 | 2.743821 |
| /alpha | 57.91175 | 34.02567 | 1.70 | 0.089 | -8.777325 | 124.6008 |
| /lns | 7.062261 | .0372561 | 189.56 | 0.000 | 6.989241 | 7.135282 |
| /lnv | 2.356454 | .02581 | 91.30 | 0.000 | 2.305867 | 2.40704 |
| s | 1167.081 | 43.48089 | | | 1084.897 | 1255.491 |
| v | 10.55346 | .2723849 | | | 10.03287 | 11.10106 |

```
Instrumented:  nwifeinc
Instruments:   educ exper expersq kidslt6 kidsge6 city
               hushrs fatheduc motheduc unem
Wald test of exogeneity (/alpha = 0): chi2(1) =     2.90  Prob > chi2 = 0.0888
  Obs. summary:        325  left-censored observations at hours<=0
                       428     uncensored observations
                         0 right-censored observations
```

```
. rivtest, ci points(500) gridmult(14)
Estimating confidence sets over grid points
----+--- 1 ----+--- 2 ----+--- 3 ----+--- 4 ----+--- 5
..................................................   50
..................................................  100
..................................................  150
..................................................  200
..................................................  250
..................................................  300
..................................................  350
..................................................  400
..................................................  450
..................................................  500
Weak instrument robust tests and confidence sets for IV Tobit
H0: beta[hours:nwifeinc] = 0
```

| Test | Statistic | p-value | 95% Confidence Set |
|---|---|---|---|
| CLR | stat(.) = 5.35 | Prob > stat = 0.0315 | [-176.335, -10.053] |
| AR | chi2(4) = 11.53 | Prob > chi2 = 0.0212 | [-154.164,-17.4433] |
| LM | chi2(1) = 3.73 | Prob > chi2 = 0.0535 | |
| | [-202.201, 1.03251] U [ 122.973, 813.968] | | |
| J | chi2(3) = 7.81 | Prob > chi2 = 0.0502 | |
| LM-J | H0 not rejected at 5% level | | [-216.982, 4.72767] |
| Wald | chi2(1) = 4.47 | Prob > chi2 = 0.0345 | [-136.876,-5.17009] |

```
Note: Wald test not robust to weak instruments. Confidence sets estimated for
500 points in [-992.966, 850.92].
```

After the `rivtest` command, we have requested two `rivtest` options related to confidence estimation: `points(500)` and `gridmult(14)`, which specify that confidence set estimation should be performed on a grid of 500 points over a width of 14 times the Wald confidence interval (centered around the IV point estimate).[3]

Here we obtain similar results to the ones in the endogenous probit example. While the estimated confidence sets are generally consistent with a negative effect of nonwife income on labor supply, the estimated confidence sets from the weak-instrument tests are wider than the Wald confidence interval.

# 6 Monte Carlo simulations

To show the performance of the tests, we perform Monte Carlo simulations of the `rivtest` command with linear IV, IV probit, and IV tobit. We show simulations from small ($N = 200$) samples, but results were qualitatively similar with larger samples. We performed simulations with both weak ($\pi_z = 0.1$) and nonweak ($\pi_z = 1$) instruments. The coefficient $\beta$ is 0.5 and the excluded instruments are drawn from independent standard normal distributions and are the same for all simulations. Finally, we experimented

3. Calculation of the test statistics is almost instantaneous, but grid-based confidence-interval estimation takes time (increasing linearly with the number of grid points). In the IV tobit example, the command required about 2 seconds for 100 grid points and 8 seconds for 500 points.

with three levels of correlation between the error terms in the two equations ($\rho$): 0.1 for low levels of simultaneity, 0.5 for moderate simultaneity, and 0.8 for a high degree of simultaneity. For each Monte Carlo experiment, we generated 5,000 simulations and computed the rejection probability under the true null hypothesis. All simulations were performed in Stata with the built-in regression commands with our `rivtest` command.[4]

Table 1 shows the results of Monte Carlo simulations for the linear IV model under homoskedasticity and arbitrary heteroskedasticity. Panel A shows the results when the errors are homoskedastic. Here we see that the Wald test does not have the correct size when the instrument is weak for all different degrees of simultaneity. For example, with a highly correlated interequation error ($\rho = 0.8$), the Wald test incorrectly rejected the true parameter in 44.94% of the simulations.

Panel B shows the results when the errors are arbitrarily heteroskedastic.[5] The performance of the Wald test with weak instruments ($\pi = 0.1$) is similar to the previous case: it overrejects the null hypothesis when the errors in the two equations are moderately or highly correlated ($\rho = 0.5$ or $\rho = 0.8$), and underrejects the null hypothesis when the simultaneity is low ($\rho = 0.1$). For the case of strong instruments ($\pi = 1$), the tests have similar nominal sizes.

4. The Monte Carlo simulations include five instruments excluded from the second-stage equation, but only one of the instruments has a nonzero coefficient in the first stage. In the tables, we refer to this coefficient as $\pi$. Also, two control variables entered the model, including a vector of ones. The error terms were drawn from a bivariate standard normal distribution with correlation coefficient $\rho$.
5. We generated this heteroskedasticity by multiplying homoskedastic errors by an independently drawn uniform random variable between zero and two—separately for each equation error.

Table 1. Size (in percent) for testing $H_0 : \beta = 0.5$ at the 5% significance level in the linear IV model under homoskedasticity and arbitrary heteroskedasticity

| *Models* Simulation parameters | | | Test size Rejection rate for tests (percent) | | | | | |
|---|---|---|---|---|---|---|---|---|
| *A.* 2SLS *with homoskedasticity* | | | | | | | | |
| $N$ | $\pi$ | $\rho$ | CLR | AR | LM | $J$ | LM-J | Wald |
| 200 | 0.1 | 0.8 | 5.34 | 5.40 | 5.34 | 5.30 | 5.62 | 44.94 |
| | | | (0.32) | (0.32) | (0.32) | (0.32) | (0.33) | (0.70) |
| 200 | 0.1 | 0.5 | 5.22 | 5.08 | 5.42 | 5.48 | 5.38 | 13.28 |
| | | | (0.31) | (0.31) | (0.32) | (0.32) | (0.32) | (0.48) |
| 200 | 0.1 | 0.1 | 5.84 | 5.52 | 6.00 | 5.02 | 5.56 | 0.90 |
| | | | (0.33) | (0.32) | (0.34) | (0.31) | (0.32) | (0.13) |
| 200 | 1 | 0.8 | 5.06 | 5.38 | 5.08 | 5.40 | 5.28 | 5.68 |
| | | | (0.31) | (0.32) | (0.31) | (0.32) | (0.32) | (0.33) |
| 200 | 1 | 0.5 | 4.64 | 5.34 | 4.68 | 5.36 | 4.94 | 4.96 |
| | | | (0.30) | (0.32) | (0.30) | (0.32) | (0.31) | (0.31) |
| 200 | 1 | 0.1 | 5.32 | 5.52 | 5.34 | 5.10 | 5.46 | 5.10 |
| | | | (0.32) | (0.32) | (0.32) | (0.31) | (0.32) | (0.31) |
| *B.* 2SLS *with arbitrary heteroskedasticity* | | | | | | | | |
| $N$ | $\pi$ | $\rho$ | CLR | AR | LM | $J$ | LM-J | Wald |
| 200 | 0.1 | 0.8 | 6.34 | 6.68 | 6.08 | 6.42 | 6.16 | 36.66 |
| | | | (0.34) | (0.35) | (0.34) | (0.35) | (0.34) | (0.68) |
| 200 | 0.1 | 0.5 | 6.60 | 6.72 | 6.18 | 6.58 | 6.22 | 11.60 |
| | | | (0.35) | (0.35) | (0.34) | (0.35) | (0.34) | (0.45) |
| 200 | 0.1 | 0.1 | 6.80 | 6.46 | 6.30 | 6.44 | 6.56 | 0.84 |
| | | | (0.36) | (0.35) | (0.34) | (0.35) | (0.35) | (0.13) |
| 200 | 1 | 0.8 | 6.26 | 6.84 | 6.22 | 5.92 | 6.76 | 6.20 |
| | | | (0.34) | (0.36) | (0.34) | (0.33) | (0.36) | (0.34) |
| 200 | 1 | 0.5 | 5.70 | 6.46 | 5.72 | 6.36 | 6.42 | 5.38 |
| | | | (0.33) | (0.35) | (0.33) | (0.35) | (0.35) | (0.32) |
| 200 | 1 | 0.1 | 6.06 | 6.32 | 6.02 | 6.28 | 6.12 | 5.08 |
| | | | (0.34) | (0.34) | (0.34) | (0.34) | (0.34) | (0.31) |

Note: Simulation standard errors are in parentheses.

In table 2, we present the result from some Monte Carlo simulations for the linear IV model when the errors have intracluster dependence.[6] We experimented with different combinations of overall sample sizes ($N$), number of clusters ($G$), and resulting cluster sizes ($M_g$). In general, asymptotics related to cluster–robust variance–covariance estimation apply only to the case when the cluster sample sizes are small and the num-

6. Within clusters, errors were drawn from a multivariate normal distribution with a nondiagonal covariance matrix. The off-diagonal blocks are multiplied by the cross-equation correlation coefficient. Across clusters, the errors are independent.

ber of clusters goes to infinity. In our simulations, we find that this is true for the weak-instrument robust tests as well.

Table 2. Size (in percent) for testing $H_0 : \beta = 0.5$ at the 5% significance level in the linear IV model with intracluster-dependent errors

| *Models* Simulation parameters | | | | | Test size Rejection rate for tests (percent) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | $G$ | $M_g$ | $\pi$ | $\rho$ | CLR | AR | LM | $J$ | LM-J | Wald |
| 400 | 100 | 4 | 0.1 | 0.8 | 6.44 | 6.52 | 5.96 | 6.20 | 6.28 | 1.12 |
| | | | | | (0.35) | (0.35) | (0.33) | (0.34) | (0.34) | (0.15) |
| 400 | 100 | 4 | 0.1 | 0.5 | 6.88 | 7.08 | 6.34 | 6.20 | 6.66 | 1.10 |
| | | | | | (0.36) | (0.36) | (0.34) | (0.34) | (0.35) | (0.14) |
| 400 | 100 | 4 | 0.1 | 0.1 | 6.82 | 7.20 | 6.42 | 6.46 | 6.54 | 0.98 |
| | | | | | (0.36) | (0.37) | (0.35) | (0.35) | (0.35) | (0.14) |
| 400 | 100 | 4 | 1 | 0.8 | 6.30 | 7.16 | 6.22 | 6.76 | 6.76 | 4.78 |
| | | | | | (0.34) | (0.36) | (0.34) | (0.36) | (0.36) | (0.30) |
| 400 | 100 | 4 | 1 | 0.5 | 5.98 | 7.16 | 5.96 | 6.84 | 6.22 | 4.86 |
| | | | | | (0.34) | (0.36) | (0.33) | (0.36) | (0.34) | (0.30) |
| 400 | 100 | 4 | 1 | 0.1 | 6.26 | 7.08 | 6.22 | 6.48 | 6.90 | 4.94 |
| | | | | | (0.34) | (0.36) | (0.34) | (0.35) | (0.36) | (0.31) |
| 500 | 50 | 10 | 0.1 | 0.8 | 8.46 | 8.74 | 7.68 | 7.18 | 8.64 | 1.50 |
| | | | | | (0.39) | (0.40) | (0.38) | (0.37) | (0.40) | (0.17) |
| 500 | 50 | 10 | 0.1 | 0.5 | 7.88 | 8.04 | 7.26 | 6.94 | 7.94 | 1.12 |
| | | | | | (0.38) | (0.38) | (0.37) | (0.36) | (0.38) | (0.15) |
| 500 | 50 | 10 | 0.1 | 0.1 | 8.56 | 8.72 | 7.66 | 7.46 | 8.62 | 1.38 |
| | | | | | (0.40) | (0.40) | (0.38) | (0.37) | (0.40) | (0.14) |
| 500 | 50 | 10 | 1 | 0.8 | 6.90 | 8.50 | 6.92 | 7.90 | 7.74 | 4.62 |
| | | | | | (0.36) | (0.39) | (0.36) | (0.38) | (0.38) | (0.30) |
| 500 | 50 | 10 | 1 | 0.5 | 6.98 | 8.40 | 6.98 | 7.38 | 7.98 | 5.05 |
| | | | | | (0.36) | (0.39) | (0.36) | (0.37) | (0.38) | (0.31) |
| 500 | 50 | 10 | 1 | 0.1 | 7.82 | 8.94 | 7.86 | 7.66 | 8.70 | 4.98 |
| | | | | | (0.38) | (0.40) | (0.38) | (0.38) | (0.40) | (0.31) |

Note: Simulation standard errors are in parentheses.

In the first six simulations, with 400 observations split into 100 clusters, the weak-instrument robust tests slightly overreject the null hypothesis, having a nominal size between 5% and 8%. This holds with weak or nonweak instruments. The Wald test, however, has a less predictable pattern; it consistently underrejects when instruments are weak but has the correct size when instruments are not weak. In the second six simulations, with 500 observations split into 50 clusters (an example consistent with many applications that use cross-sectional data from U.S. states), the weak-instrument robust tests also overreject, but their performance is still closer to the correct size than the Wald tests when instruments are weak.

We also conducted simulations with larger and smaller numbers of clusters and different numbers of observations within cluster. We found that the number of clusters is the most important element in determining the rejection probability of the tests. The overrejection decreases as the number of clusters increases.[7] We recommend bootstrapping the test to find appropriate critical values when the number of clusters is small (less than 50). A discussion of techniques that work well in the single equation linear model can be found in Cameron, Gelbach, and Miller (2008).

In table 3, we present the results from Monte Carlo simulations for the endogenous probit and tobit models (panels A and B, respectively). To avoid having to rescale the maximum likelihood test in the endogenous probit model, we let the population parameter, $\beta$, equal zero.[8]

7. Results are available upon request.
8. When $\beta = 0$, we have $\beta/\sigma_u = \beta/\sigma_\varepsilon = 0$ for positive values of $\sigma_u$ and $\sigma_\varepsilon$.

Table 3. Size (in percent) for testing $H_0 : \beta = 0$ at the 5% significance level in the endogenous probit model and $H_0 : \beta = 0.5$ at the 5% significance level in the endogenous tobit model

| *Models* Simulation parameters | | | Test size Rejection rate for tests (percent) | | | | | |
|---|---|---|---|---|---|---|---|---|
| A. *IV probit* ($\beta = 0$) | | | | | | | | |
| $N$ | $\pi$ | $\rho$ | CLR | AR | LM | $J$ | LM-J | Wald |
| 200 | 0.1 | 0.8 | 3.58 | 3.52 | 4.59 | 4.07 | 4.01 | 32.95 |
| | | | (0.26) | (0.26) | (0.30) | (0.28) | (0.28) | (0.67) |
| 200 | 0.1 | 0.5 | 3.99 | 3.93 | 5.03 | 4.49 | 4.77 | 41.94 |
| | | | (0.28) | (0.28) | (0.31) | (0.29) | (0.30) | (0.70) |
| 200 | 0.1 | 0.1 | 4.90 | 4.70 | 5.24 | 4.68 | 4.90 | 45.17 |
| | | | (0.31) | (0.30) | (0.32) | (0.30) | (0.31) | (0.70) |
| 200 | 1 | 0.8 | 3.94 | 3.88 | 3.96 | 4.72 | 3.82 | 5.12 |
| | | | (0.28) | (0.27) | (0.28) | (0.30) | (0.27) | (0.31) |
| 200 | 1 | 0.5 | 4.68 | 4.88 | 4.66 | 4.90 | 4.38 | 5.68 |
| | | | (0.30) | (0.30) | (0.30) | (0.31) | (0.29) | (0.33) |
| 200 | 1 | 0.1 | 5.24 | 5.10 | 5.26 | 5.32 | 5.16 | 6.18 |
| | | | (0.32) | (0.31) | (0.32) | (0.32) | (0.31) | (0.34) |
| B. *IV tobit* | | | | | | | | |
| $N$ | $\pi$ | $\rho$ | CLR | AR | LM | $J$ | LM-J | Wald |
| 200 | 0.1 | 0.8 | 5.18 | 5.38 | 5.24 | 5.16 | 5.06 | 18.10 |
| | | | (0.31) | (0.32) | (0.32) | (0.31) | (0.31) | (0.54) |
| 200 | 0.1 | 0.5 | 5.34 | 5.50 | 5.16 | 5.44 | 5.24 | 7.20 |
| | | | (0.32) | (0.32) | (0.31) | (0.32) | (0.32) | (0.37) |
| 200 | 0.1 | 0.1 | 6.28 | 5.86 | 6.02 | 5.36 | 6.10 | 0.74 |
| | | | (0.34) | (0.33) | (0.34) | (0.32) | (0.34) | (0.12) |
| 200 | 1 | 0.8 | 5.12 | 5.22 | 5.10 | 5.40 | 5.22 | 5.14 |
| | | | (0.31) | (0.31) | (0.31) | (0.32) | (0.31) | (0.31) |
| 200 | 1 | 0.5 | 5.30 | 5.66 | 5.24 | 5.26 | 5.44 | 5.20 |
| | | | (0.32) | (0.33) | (0.32) | (0.32) | (0.32) | (0.31) |
| 200 | 1 | 0.1 | 5.16 | 5.84 | 5.26 | 5.72 | 5.26 | 5.04 |
| | | | (0.31) | (0.33) | (0.32) | (0.33) | (0.32) | (0.31) |

Note: Simulation standard errors are in parentheses.

(*Continued on next page*)

With any value of the simultaneity parameter, we find that the Wald test performs poorly when the instruments are weak ($\pi = 0.1$) in both the endogenous probit and tobit models. Surprisingly, the rejection probability for the Wald test in the endogenous probit model with weak instruments is above 30% independent of the degree of simultaneity, which contrasts with patterns observed in the linear IV and endogenous tobit models.[9] Regardless of the strength or weakness of the instruments, our tests are estimated to have rejection rates between 3.5% and 6.3%, close to the correct size of 5%.

# 7 Acknowlegments

We thank Mark Schaffer, Alan Barreca, Tom Palmer, and an anonymous referee for helpful comments, and the Tulane Research Enhancement Fund and the Committee on Research Summer Fellowship for funding.

# 8 References

Anderson, T. W., and H. Rubin. 1949. Estimation of the parameters of a single equation in a complete system of stochastic equations. *Annals of Mathematical Statistics* 20: 46–63.

Andrews, D. W. K., M. J. Moreira, and J. H. Stock. 2007. Performance of conditional Wald tests in IV regression with weak instruments. *Journal of Econometrics* 139: 116–132.

Baum, C. F., M. E. Schaffer, and S. Stillman. 2007. Enhanced routines for instrumental variables/generalized method of moments estimation and testing. *Stata Journal* 7: 465–506.

Cameron, A. C., J. B. Gelbach, and D. L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics* 90: 414–427.

Chernozhukov, V., and C. Hansen. 2008. The reduced form: A simple approach to inference with weak instruments. *Economics Letters* 100: 68–71.

Dufour, J.-M. 2003. Identification, weak instruments and statistical inference in econometrics. Working Paper 2003s-49, Scientific Series, CIRANO. http://www.cirano.qc.ca/pdf/publication/2003s-49.pdf.

9. With weak instruments, the endogenous probit models have a difficult time converging. For example, for the weak-instrument ($\delta = 0.1$) simulations shown in table 3, 32 simulations did not converge under high simultaneity ($\rho = 0.8$), 10 simulations did not converge under moderate simultaneity ($\rho = 0.5$), and 1 simulation did not converge under low simultaneity ($\rho = 0.1$). In our experiments, the rejection rates for the weak-instrument robust tests were not affected by the proportion of simulations that did not converge. The Wald test, however, will reject more frequently under non-convergence. In table 3, we exclude test results for simulations that did not achieve convergence in the IV model, so the Wald rejection rates for IV probit with weak instruments should be considered lower bounds.

Hansen, L. P., J. Heaton, and A. Yaron. 1996. Finite-sample properties of some alternative GMM estimators. *Journal of Business and Economic Statistics* 14: 262–280.

Kleibergen, F. 2007. Generalizing weak instrument robust IV statistics towards multiple parameters, unrestricted covariance matrices and identification statistics. *Journal of Econometrics* 139: 181–216.

Magnusson, L. M. 2008a. Inference in limited dependent variable models robust to weak identification. Working Paper 0801, Department of Economics, Tulane University. http://ideas.repec.org/p/tul/wpaper/0801.html.

———. 2008b. Tests in censored models when the structural parameters are not identified. Working Paper 0802, Department of Economics, Tulane University. http://ideas.repec.org/p/tul/wpaper/0802.html.

Mikusheva, A. 2005. Robust confidence sets in the presence of weak instruments. Working Paper No. 07-27, Department of Economics, Massachusetts Institute of Technology. http://ssrn.com/abstract=1021366.

Mikusheva, A., and B. P. Poi. 2006. Tests and confidence sets with correct size when instruments are potentially weak. *Stata Journal* 6: 335–347.

Moreira, M. J. 2003. A conditional likelihood ratio test for structural models. *Econometrica* 71: 1027–1048.

Moreira, M. J., and B. P. Poi. 2003. Implementing tests with correct size in the simultaneous equations model. *Stata Journal* 3: 57–70.

Mroz, T. A. 1987. The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions. *Econometrica* 55: 765–799.

Stock, J. H., J. H. Wright, and M. Yogo. 2002. A survey of weak instruments and weak identification in generalized method of moments. *Journal of Business and Economic Statistics* 20: 518–529.

**About the authors**

Keith Finlay and Leandro M. Magnusson are assistant professors in the Department of Economics at Tulane University.

The Stata Journal (2009)
**9**, Number 3, pp. 422–438

# A seasonal unit-root test with Stata

Domenico Depalo
Bank of Italy
Roma, Italy
domenico.depalo@bancaditalia.it

**Abstract.** Many economic time series exhibit important systematic fluctuations within the year, i.e., seasonality. In contrast to usual practice, I argue that using original data should always be considered, although the process is more complicated than that of using seasonally adjusted data. Motivations to use unadjusted data come from the information contained in their peaks and troughs and from economic theory. One major complication is the possible unit root at seasonal frequencies. In this article, I tackle the issue of implementing a test to identify the source of seasonality. In particular, I follow Hylleberg et al. (1990, *Journal of Econometrics* 44: 215–238) for quarterly data.

**Keywords:** st0172, sroot, unit roots, seasonality

## 1 Introduction

Many economic time series exhibit important systematic fluctuations within the year, i.e., seasonality. Although applied econometricians have long used seasonally adjusted data, there exists increasing consensus that this practice is suboptimal for at least two reasons. First, peaks and troughs convey information that is lost during the adjustment; second, seasonally adjusted data often conflicts with the economic theory. Consider the rational expectation hypothesis or the permanent income hypothesis. Seasonal adjustment, for example, by the widely known CENSUS-X11, invalidates the theory by construction, because it is a two-sided filter, which thus violates the key orthogonality condition between the data at time $t$ and the available information at the same time. To avoid these flaws, one can use the original data and either control for a set of seasonal dummies or redefine the error term to incorporate the seasonal fluctuations. The first solution is weak because "data adjusted by the seasonal dummy technique will [...] tend to reject the model if it contains fundamental nonlinearities" (Miron 1986, 1260). The second solution is wrong because the error terms would be predictable to some extent, thus invalidating the rational expectation hypothesis (Osborn 1988). These simple facts have two important consequences: using seasonally adjusted data can have serious consequences on our results and the treatment of seasonality requires a serious systematic approach.

In this article, I am particularly interested in seasonality and unit roots at seasonal frequencies. I first review some basic theory about unit roots at seasonal frequencies (section 2); then I describe the new `sroot` command, which performs a formal test for unit roots in quarterly data (section 3); and then I give some advice for the applied researcher based on some Monte Carlo simulations (section 4). In section 5, I use the `sroot` command to detect seasonal unit roots in the original series of consumption in the UK for the years 1955–2006.

## 2 Unit roots at various frequencies

The spectrum of a seasonal series has distinct peaks at seasonal frequencies $\omega_s = 2\pi j/s$, where $j = 1, \ldots, s/2$ and $s$ is the number of periods within a year. In particular, we deal with $s = 4$ because it is the most common case.

While there is consensus on the importance of seasonality, there is little agreement on its treatment. Indeed, there are several ways to handle seasonality, each implicitly making different assumptions about the process, namely, as if it is

- a purely *deterministic* seasonal process,
- a *stationary* seasonal process, or
- an *integrated* seasonal process.

In applied work, the general (incorrect) belief is that the three methodologies are equivalent. In fact, they imply a very different data-generating process, as discussed below.

In a purely deterministic seasonal process, the reference model for the conditional mean of the dependent variable, $y$, can be written as

$$y = \mathbf{x}\,\beta + \sum_{i=1}^{3} \delta_i\, D_i$$

where $y$ is a vector of dimension $n$; $\mathbf{x}$ is an $n \times k$ matrix with the first column containing only ones; $\beta$ is a vector of length $k$; and each $\delta_i$ is the coefficient attached to the vector $D_i$, a dummy vector equal to 1 only in season $i$. This notation will be employed throughout the article.

A stationary seasonal process can be written as an autoregressive model,

$$\phi(L)y_t = \epsilon_t \tag{1}$$

with all the roots of $\phi(L)$ outside the unit circle (but some come in complex pairs). If $s = 4$, then a stationary seasonal process is $y_t = \rho L^4 y_t + \epsilon_t$, where $L$ is the lag operator and $L^4 y_t = y_{t-4}$. If some of the roots lie on the unit circle, the process is an integrated seasonal process.

Continuing with $s = 4$, a seasonally integrated series can be further decomposed into

$$\begin{aligned}(1 - L^4)\, y_t &= \epsilon_t \\ &= (1 - L)(1 + L)(1 + L^2) y_t\end{aligned} \tag{2}$$

which shows that in seasonal processes, the roots of modulus 1 can be four and not only one, as for the classical case. Also, two of the roots will be complex. Properties of each root are very similar to those at zero frequency; in particular, shocks have a permanent effect on the seasonal pattern, and their variances increase linearly with time, but shocks are asymptotically uncorrelated with unit-root processes of other frequencies. To see this more formally, consider the process in (2) as a stochastic difference process (details are in Hylleberg et al. [1990]), whose homogeneous solutions are

$$\begin{aligned}s_{1,t} &= \textstyle\sum_{j=0}^{t-1} \epsilon_{t-j} && \text{for zero-frequency root;} \\ s_{2,t} &= \textstyle\sum_{j=0}^{t-1} (-1)^j \epsilon_{t-j} && \text{for two-cycle-per-year root;} \\ s_{3,t} &= \textstyle\sum_{j=0}^{\text{int}\{(t-1)/2\}} (-1) \Delta\epsilon_{t-2j} && \text{for one-cycle-per-year root}\end{aligned} \tag{3}$$

By expanding each single component of (3), we can show that the variance of each frequency increases linearly with time [specifically, $V(s_{1,t}) = V(s_{2,t}) = V(s_{3,t}) = t\sigma^2$]. Using the same technique, we can show that covariances are zero for complete years of data when the series are excited by the same $\epsilon_t$ and, thus, that the series are uncorrelated; for example,

$$\begin{aligned}\text{cov}(s_1, s_2) &= \overbrace{(\epsilon_t + \epsilon_{t-1} + \epsilon_{t-2} + \epsilon_{t-3} + \ldots)}^{s_{1,t}} \overbrace{(\epsilon_t - \epsilon_{t-1} + \epsilon_{t-2} - \epsilon_{t-3} + \ldots)}^{s_{2,t}} \\ &= \sigma^2 - \sigma^2 + \sigma^2 - \sigma^2 + \ldots \\ &= 0\end{aligned}$$

Differently from what has been suggested by many practitioners, I argue that using not seasonally adjusted (NSA) data should always be considered. At least as a robustness check, one should perform all the analysis with both seasonally adjusted (SA) and NSA data. However, we showed that NSA data have more involved processes than SA data, particularly because of unit roots at seasonal frequencies.

In what follows, I analyze a formal test to study the presence of seasonal unit roots on a statistical basis, focusing on Hylleberg et al. (1990).

A general expression for seasonal processes combines the three seasonal processes and is compactly represented by

$$d(L)a(L)(y_t - \mu_t) = \epsilon_t$$

where the roots of $a(L) = 0$ lie outside the unit circle, the roots of $d(L) = 0$ lie on the unit circle, and $\mu_t = x\,\beta + \sum_{i=1}^{3} \delta_i\, D_i$. It follows that stationary components of $y$ are in $a(L)$, while deterministic seasonality is in $\mu_t$ when there are no seasonal unit roots in $d(L)$. The test by Hylleberg et al. (1990) studies this model and detects seasonal unit roots at different seasonal frequencies, as well as at zero frequency.

The methodology strongly relies on a Lagrangian polynomial expansion for $\phi(L)$ in (1). Applying this representation for quarterly data, Hylleberg et al. (1990) study

$$\phi(L)y_{4,t} = \pi_1\, y_{1,t-1} + \pi_2\, y_{2,t-1} + \pi_3\, y_{3,t-2} + \pi_4\, y_{3,t-1} + \epsilon_t \tag{4}$$

where

$$\begin{aligned} y_{1,t} &= (1 + L + L^2 + L^3)\, y_t \\ y_{2,t} &= -(1 - L + L^2 - L^3)\, y_t \\ y_{3,t} &= (1 - L^2)\, y_t \\ y_{4,t} &= (1 - L^4)\, y_t \end{aligned}$$

and $\pi_i$'s are coefficients for seasonal roots, which we test to establish the nature of seasonality. In particular, at root $1 - L$ the test is on coefficient $\pi_1 = 0$, at seasonal root $1 + L$ the test is on coefficient $\pi_2 = 0$, and finally, at seasonal roots $1 + L^2$ the test is joint on coefficients $\pi_3 = \pi_4 = 0$. For a unit root in a given frequency, the associated coefficient $\pi_i$ is zero. If $\pi_2$ and either $\pi_3$ or $\pi_4$ are different from zero, there is no seasonal unit root. Similarly, if $\pi_1$ is also different from zero, the series has no unit roots at all. The natural alternative for these tests is stationarity, $\pi_1 < 0$ and $\pi_2 < 0$, respectively, for $\pi_1$ and $\pi_2$, or that $\pi_3$ and $\pi_4$ are not jointly equal to zero. To consider all the possible cases in the three seasonal processes, (4) can be augmented in various directions, such as lagged values of $y_4$ or deterministic components, and consistently estimated by ordinary least squares.

Although I focus on quarterly data, this identical setup can be readily generalized to other cases frequently encountered in practice, like biannual data or monthly data (see Franses and Hobijn [1997]).

The asymptotic distribution of the estimator of the coefficients in (4) is nonstandard. Because the method is analogous to that of Dickey and Fuller (1979), the distribution theory for these tests can be extracted from Dickey and Fuller (1979) and Fuller (1976) for $\pi_1$ and $\pi_2$, and from Dickey, Hasza, and Fuller (1984) for $\pi_3$, if $\pi_4$ is assumed to be zero. The tests are asymptotically invariant with respect to nuisance parameters. According to Hylleberg et al. (1990, 224), the finite-sample results are well approximated by the asymptotic theory, and the tests have reasonable power against each of the specific alternatives. The intercept and trend in the model affect only the distribution of $\pi_1$, whereas seasonal dummies affect only the distributions of $\pi_2$, $\pi_3$, and $\pi_4$.

I would like to conclude this section with a natural extension of the seasonal unit root, i.e., a seasonal cointegration and seasonal vector error-correction model. There are several methods for testing and estimation of cointegration at seasonal frequencies (Lee [1995], Johansen and Schaumburg [1999], and Cubadda [2001], among others), and each deserves a specific treatment, which we leave for future extensions. However, a simpler approach goes back to Engle and Granger (1987) and is adapted to the seasonal case by Engle et al. (1993). This simpler approach is a two-step estimator that simply requires estimating the linear combination(s) of levels on data transformed to account for seasonality and, for seasonal vector error-correction model estimation, relies on the speed of convergence in the first step.

# 3 The sroot command

The increasing variety of time-series methods in Stata has increased the number of time-series users with Stata. Thanks to the simplicity of data management, the proposed command makes extensive use of Stata's routines for lag operators and the `regress` command. The syntax of `sroot` is

`sroot` *varname* [ *if* ] [ *in* ] [ `,` `noconstant` `trend` `season(`*varlist*`)` `regress` `lags(`#`)` `generate(`*string*`)` `residuals(`*string*`)` ]

## 3.1 Options

`noconstant` suppresses the constant term (intercept) in the model and indicates that the process under the null hypothesis is a random walk without drift. `noconstant` may not be used with the `trend` or `season(`*varlist*`)` option.

`trend` specifies that a trend term be included in the associated regression and that the process under the null hypothesis is a random walk, perhaps with drift. This option may not be used with the `noconstant` option.

`season(`*varlist*`)` indicates that the process under the null hypothesis is a random walk augmented for seasonal dummies. It is possible that *varlist* contains only one word (in which case the command builds the dummies) or that *varlist* contains the full set of dummies (in which case the command drops the last quarter because of multicollinearity). This option may not be used with the `noconstant` option.

`regress` specifies that the associated regression table appear in the output. By default, the regression table is not produced.

`lags(`#`)` specifies the number of lagged difference terms to include in the covariate list.

`generate(`*string*`)` generates a set of variables adjusted for seasonal filtering.

`residuals(`*string*`)` generates a variable containing the residual terms.

# 4 Some practical issues

In this section, I give some advice for applied research. I first explore distinctive features of `sroot` with respect to an existing similar command (section 4.1), and then I give some practical guidelines useful in empirical applications (section 4.2).

## 4.1 Why a new command?

The `hegy4` command in Stata performs the Hylleberg et al. (1990).[1] The two commands (`hegy4` and `sroot`) have key distinctions that I briefly explore in this section. I conclude

1. I thank C. Baum for bringing to my attention a very similar routine (Baum and Sperling 2001).

that `hegy4` and `sroot` are similar; thus suggestions in section 4.2 will be valid for both procedures.

First of all, the default in `hegy4` is to run a sequential test for the proper number of lags. A simple (unreported) simulation reveals that in some circumstances, it could be inappropriate. I designed the simulation for 48, 100, and 200 observations (12, 25, and 50 years), and for parameters $\beta_i \Delta_4 y_{t-i} = \{-0.8, -0.4, -0.2, -0.02, 0, 0.02, 0.2, 0.4, 0.8\}$ with $i = 1, 2$, along with their combinations (notice that coefficients are exactly equal to zero sometimes). The performance of a sequential test increases with sample size and with the absolute values of coefficients. However, two remarks are needed: First, as either $\beta_1$ or $\beta_2$ approaches but is different from zero, the practice is questionable and only in a bunch of cases, all with 200 observations, is the lag selection 100% correct. With 48 observations, the operational tool performs poorly. Second, undoubtable advantages are when $\beta_1$ or $\beta_2$ is indeed zero. However, `hegy4` offers the option "`notest` [that] may be specified to suppress the lag length test and utilize the lags specified in the option in generating the test statistic" (Baum and Sperling 2001).

Even though the Hylleberg et al. (1990) results are unaffected by nuisance parameters, according to the experiments in the next section, in cases of uncertainty about the correct number of lags, specifying the `notest` option seems a more convincing approach.

The second, most important, difference is in the `generate()` option. Engle et al. (1993) show that it is possible to study seasonal cointegration starting from transformed variables. The interested reader is referred to that article for further details, but for what matters here, we can build stationary combinations from transformed nonstationary variables in levels and study a vector autoregression augmented for these components in a seasonal error-correction model. I view this as a key distinction to push efforts toward the new `sroot` command.

A very minor difference is the `regress` option for `sroot`.

Of course, the tests, ceteris paribus, give the very same numbers, as shown with the following example:

```
. sroot x_nsa, lag(1) trend season(quarter)
HEGY test for SEASONAL unit roots                 Number of obs   =       203
                 Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value            Value
------------------------------------------------------------------------------
 Z(t) - Fr 0      0.180            -4.050            -3.490           -3.180
 Z(t) - Fr 1/2    2.522            -3.520            -2.910           -2.600
 Z(t) - L.Ann.    0.437            -4.040            -3.410           -3.100
 Z(t) - Annual    0.286            -2.650            -1.920           -1.480
 Joint Annual     0.133             8.960             6.570            5.560
 All SEAS. fr.    2.170                 .             5.890            5.100
 All freq.        1.644                 .             6.380            5.610
```

(*Continued on next page*)

```
. hegy4 x_nsa, lag(1) det(strend)

HEGY Quarterly seasonal unit root test for x_nsa

Number of observations  : 203
Deterministic variables : Seasonal dummies + constant + trend
Lags tested: 1
Augmented by lags : 1

          Stat     5% critical   10% critical
------------------------------------------------------------------------------
t[Pi1]    0.180      -3.490        -3.180
t[Pi2]    2.522      -2.910        -2.600
t[Pi3]    0.437      -3.410        -3.100
t[Pi4]    0.286      -1.920        -1.480

F[3-4]    0.133       6.570         5.560
F[2-4]    2.170       5.890         5.100
F[1-4]    1.644       6.380         5.610
```

## 4.2 Some practical guidelines

The most troublesome practical issues with this seasonal unit-root test are related to the deterministic terms and to the appropriate number of lags. In particular, with respect to the deterministic terms, the important question is whether they should be included in the model specification; with respect to the lags, the important question is how many lags should be considered. I try to answer these questions in this section by using Monte Carlo experiments based on 5,000 repetitions and designs specified below.

Nevertheless, I strongly suggest that the researcher verify, case by case, that residuals have desired properties, through the `residuals()` option. For example, Baum and Sperling (2001) suggest the regression of the (generated) residuals on four lags and the original regressors under the rationale that if all the information has already been considered, the null hypothesis that all coefficients are jointly equal to zero should not be rejected; other useful checks could be performed on specific moments of the distribution of the residuals, like the third moment (skewness) and the fourth moment (kurtosis).

Finally, it should be clear that here I adopt an empirical approach; theoretical consequences can be found in Ghysels, Lee, and Noh (1994).

### More deterministic terms is better than fewer deterministic terms

I first examine the importance of deterministic terms in the model specification. According to the common wisdom, I will conclude that in the empirical applications, in case of uncertainty, it is safer to include deterministic terms even when they are not in the true data-generating process (DGP), rather than vice versa, neglecting deterministic terms that, in fact, are in the true DGP.

I use the following experiment:

$$y_t = \rho\, y_{t-4} + \sum_{i=1}^{4} \delta_i\, D_i + \gamma t + u_t \qquad t = 1, \ldots, T$$

where $\rho$ determines the (non)stationarity of the model; $D_i$ is for seasonal dummies, associated to parameters $\delta_i = (-0.1, 0.05, -0.05, 0.1)$, which correspond to a $-10\%$ annualized drop, followed by a 5% increase, a 5% decrease, and a 10% increase, respectively; $\gamma$ is set equal to 0.05; and $u_t$ is white noise. The correct lag length in the DGP is zero, but we carried on the test for different lags, from zero to four. Additional lags will be indicative of the consequences of controlling for more lags than are needed in the presence of deterministic components. This experiment, except for the time trend, is used in Ghysels, Lee, and Noh (1994).

In table 1, I report the share of rejection of the null hypothesis of the unit root. The table has two main parts: on the left-hand side, the true DGP does in fact contain unit roots at all frequencies ($\rho = 1$), and the right-hand side is stationary ($\rho = 0.85$; see Ghysels, Lee, and Noh [1994] for further details on this specific value). Each side is further differentiated: in one case, the model is misspecified because we neglect the presence of seasonal dummies, and in the other case, it is correctly specified because all the deterministic components are controlled for. For easier readability, we report only the share of stationary roots at zero frequency and jointly at all the frequencies. When unit roots are in the true DGP, the entry should be zero, whereas when the model is stationary, the entry should be one. I discuss these measures.

When the data contain unit roots at all frequencies, neglecting seasonal dummies has serious consequences on the share of rejection. Because we control, by default of the command, for the intercept, the consequences on frequency 0 will be attenuated (as expected from section 2). Nevertheless, when we consider the whole set of frequencies, the conclusion will be seriously biased. As suggested in Ghysels, Lee, and Noh (1994), the reason is that test statistics under this misspecification are functions of the unknown seasonal dummy coefficients.

When we consider stationary data, two main conclusions can be drawn from the table. First, when the data are stationary but we neglect the deterministic terms in the model specification, our conclusions about unit roots at zero frequency are biased toward the nonrejection of the null hypothesis, which (however) is false (technically, the power of the test against this misspecification is low). Second, once deterministic terms are considered, a correct specification of lag length is less important, as shown by the comparison of the first panel (labeled "Lag: 0") with respect to lower panels. This specific aspect is elaborated upon in the next subsection.

Aside from the correct specification, the number of observations plays a critical role. A performance with fewer than 100 observations is unsatisfactory, whereas a performance with 200 or more observations is good.

Table 1. Consequences of neglecting deterministic components

| Obser. | Unit root | | | | Stationary | | | |
|---|---|---|---|---|---|---|---|---|
| | Misspec. | | Correct | | Misspec. | | Correct | |
| | Fr 0 | All | Fr 0 | All | Fr 0 | All | Fr 0 | All |
| | | | | Lag: 0 | | | | |
| 48 | 0.000 | 0.701 | 0.001 | 0.019 | 0.000 | 0.070 | 0.018 | 0.085 |
| 100 | 0.000 | 1.000 | 0.001 | 0.007 | 0.000 | 0.388 | 0.064 | 0.175 |
| 200 | 0.000 | 1.000 | 0.000 | 0.006 | 0.000 | 0.846 | 0.340 | 0.700 |
| 300 | 0.000 | 1.000 | 0.000 | 0.006 | 0.000 | 0.983 | 0.684 | 0.974 |
| | | | | Lag: 1 | | | | |
| 48 | 0.000 | 0.426 | 0.002 | 0.019 | 0.000 | 0.058 | 0.017 | 0.080 |
| 100 | 0.000 | 1.000 | 0.001 | 0.010 | 0.000 | 0.311 | 0.063 | 0.169 |
| 200 | 0.000 | 1.000 | 0.000 | 0.007 | 0.000 | 0.784 | 0.321 | 0.676 |
| 300 | 0.000 | 1.000 | 0.000 | 0.007 | 0.000 | 0.973 | 0.657 | 0.967 |
| | | | | Lag: 2 | | | | |
| 48 | 0.000 | 0.284 | 0.002 | 0.024 | 0.000 | 0.057 | 0.015 | 0.083 |
| 100 | 0.000 | 0.994 | 0.002 | 0.011 | 0.000 | 0.279 | 0.059 | 0.159 |
| 200 | 0.000 | 1.000 | 0.001 | 0.006 | 0.000 | 0.731 | 0.292 | 0.650 |
| 300 | 0.000 | 1.000 | 0.000 | 0.006 | 0.000 | 0.961 | 0.619 | 0.961 |
| | | | | Lag: 3 | | | | |
| 48 | 0.000 | 0.183 | 0.004 | 0.025 | 0.000 | 0.049 | 0.012 | 0.074 |
| 100 | 0.000 | 0.891 | 0.002 | 0.009 | 0.000 | 0.235 | 0.061 | 0.157 |
| 200 | 0.000 | 0.990 | 0.000 | 0.008 | 0.000 | 0.679 | 0.272 | 0.614 |
| 300 | 0.000 | 0.985 | 0.000 | 0.006 | 0.000 | 0.946 | 0.574 | 0.956 |
| | | | | Lag: 4 | | | | |
| 48 | 0.000 | 0.149 | 0.002 | 0.019 | 0.000 | 0.045 | 0.015 | 0.057 |
| 100 | 0.000 | 0.660 | 0.002 | 0.008 | 0.000 | 0.218 | 0.055 | 0.127 |
| 200 | 0.000 | 0.772 | 0.000 | 0.007 | 0.000 | 0.640 | 0.235 | 0.553 |
| 300 | 0.000 | 0.553 | 0.000 | 0.006 | 0.000 | 0.922 | 0.536 | 0.931 |

Of course, one can wonder what the consequences are of controlling for undue deterministic terms. The same experiment from above, without deterministic terms, supports the view that the impact of undue deterministic components in the model specification is rather limited. Indeed, the shares of rejection of nonstationary root are close between the misspecified model that controls for deterministic terms and the correctly specified models (table 2).

Table 2. Consequences of imposing deterministic components

| Obser. | Unit root | | | | Stationary | | | |
|---|---|---|---|---|---|---|---|---|
| | Misspec. | | Correct | | Misspec. | | Correct | |
| | Fr 0 | All | Fr 0 | All | Fr 0 | All | Fr 0 | All |
| | | | | Lag: 0 | | | | |
| 48 | 0.022 | 0.060 | 0.035 | 0.048 | 0.030 | 0.106 | 0.051 | 0.130 |
| 100 | 0.031 | 0.048 | 0.051 | 0.056 | 0.052 | 0.157 | 0.088 | 0.362 |
| 200 | 0.040 | 0.046 | 0.051 | 0.055 | 0.107 | 0.491 | 0.211 | 0.907 |
| 300 | 0.041 | 0.045 | 0.052 | 0.052 | 0.202 | 0.865 | 0.410 | 0.999 |
| | | | | Lag: 1 | | | | |
| 48 | 0.015 | 0.061 | 0.032 | 0.046 | 0.015 | 0.090 | 0.047 | 0.117 |
| 100 | 0.027 | 0.047 | 0.051 | 0.052 | 0.042 | 0.147 | 0.083 | 0.351 |
| 200 | 0.036 | 0.045 | 0.053 | 0.054 | 0.097 | 0.480 | 0.200 | 0.901 |
| 300 | 0.040 | 0.044 | 0.050 | 0.052 | 0.187 | 0.861 | 0.392 | 0.998 |
| | | | | Lag: 2 | | | | |
| 48 | 0.018 | 0.060 | 0.035 | 0.043 | 0.016 | 0.091 | 0.044 | 0.115 |
| 100 | 0.033 | 0.048 | 0.049 | 0.051 | 0.041 | 0.146 | 0.084 | 0.350 |
| 200 | 0.035 | 0.046 | 0.051 | 0.050 | 0.096 | 0.479 | 0.193 | 0.900 |
| 300 | 0.039 | 0.044 | 0.050 | 0.053 | 0.177 | 0.861 | 0.383 | 0.998 |
| | | | | Lag: 3 | | | | |
| 48 | 0.015 | 0.060 | 0.038 | 0.050 | 0.010 | 0.076 | 0.043 | 0.107 |
| 100 | 0.027 | 0.047 | 0.049 | 0.048 | 0.031 | 0.132 | 0.081 | 0.334 |
| 200 | 0.036 | 0.048 | 0.050 | 0.050 | 0.087 | 0.465 | 0.181 | 0.901 |
| 300 | 0.036 | 0.043 | 0.050 | 0.053 | 0.158 | 0.853 | 0.356 | 0.998 |
| | | | | Lag: 4 | | | | |
| 48 | 0.016 | 0.048 | 0.032 | 0.045 | 0.020 | 0.067 | 0.036 | 0.080 |
| 100 | 0.030 | 0.042 | 0.045 | 0.046 | 0.043 | 0.114 | 0.073 | 0.272 |
| 200 | 0.037 | 0.044 | 0.050 | 0.048 | 0.096 | 0.385 | 0.176 | 0.826 |
| 300 | 0.038 | 0.044 | 0.049 | 0.050 | 0.172 | 0.766 | 0.338 | 0.994 |

For these reasons, I strongly suggest controlling for deterministic terms when performing a seasonal unit-root test.

### More lags is better than fewer lags

The second important issue is related to the appropriate specification of the number of lags. I will conclude that it is a less important decision than that about deterministic terms, but in case of uncertainty about the true DGP, it could be safer to control for more lags than for fewer lags.

This conclusion is based on the following experiment:

$$y_t = \rho\, y_{t-4} + \sum_{j=1}^{2} \alpha_j \Delta y_{t-j} + u_t \qquad t = 1, \ldots, T$$

for various combinations of $\alpha_1$ and $\alpha_2$. The correct lag length would be two, but we carried on the test for different lags, from zero to four. In table 3, we select only lag zero (i.e., fewer lags than needed), lag two (i.e., correctly specified), and lag four (i.e., more lags than needed).

The parameter $\rho$ determines the (non)stationarity of the model. In table 3, on the left, the true DGP is nonstationary ($\rho = 1$), while on the right-hand side, it is stationary ($\rho = 0.85$).

The main message from table 3 is that a correct lag specification is less important than a correct specification of deterministic terms. As expected, the best performances are achieved when the lag is correctly specified, overall with 300 observations. However, under lag misspecification, controlling for more lags than are needed could be safer than controlling for fewer. Indeed, based on experiments in Ghysels, Lee, and Noh (1994, 425), adding lags beyond what is necessary could be understood as an attempt to control for possible moving-average components whose "bias shrinks as additional lags of the autoregressive terms are included in the model". Finally, from table 3, the trade-off in the number of lags is clear, because adding lagged values reduces the *power* of the test, while the *size* suffers if too few parameters are included (Engle et al. 1993).

Although the evidence is not clear-cut, based on theoretical considerations, the practical guidance for the applied researcher in case of uncertainty is that it is safer to control for more lags than are needed.

Table 3. Consequences of neglecting lags; the model is stationary

| | | | Unit root | | Stationary | |
|---|---|---|---|---|---|---|
| Observ. | $\alpha_{t-1}$ | $\alpha_{t-2}$ | Fr 0 | All | Fr 0 | All |
| | | | Lag: 0 | | | |
| 100 | 0.000 | 0.000 | 0.063 | 0.066 | 0.106 | 0.405 |
| 100 | 0.020 | 0.000 | 0.063 | 0.066 | 0.106 | 0.405 |
| 100 | 0.020 | 0.020 | 0.055 | 0.059 | 0.085 | 0.391 |
| 100 | 0.400 | 0.000 | 0.063 | 0.066 | 0.106 | 0.405 |
| 100 | 0.400 | 0.020 | 0.055 | 0.059 | 0.085 | 0.391 |
| 100 | 0.400 | 0.400 | 0.086 | 0.403 | 0.000 | 0.885 |
| 300 | 0.000 | 0.000 | 0.058 | 0.059 | 0.422 | 0.999 |
| 300 | 0.020 | 0.000 | 0.058 | 0.059 | 0.422 | 0.999 |
| 300 | 0.020 | 0.020 | 0.050 | 0.051 | 0.373 | 0.999 |
| 300 | 0.400 | 0.000 | 0.058 | 0.059 | 0.422 | 0.999 |
| 300 | 0.400 | 0.020 | 0.050 | 0.051 | 0.373 | 0.999 |
| 300 | 0.400 | 0.400 | 0.100 | 0.487 | 0.000 | 1.000 |
| | | | Lag: 2 | | | |
| 100 | 0.000 | 0.000 | 0.052 | 0.051 | 0.086 | 0.346 |
| 100 | 0.020 | 0.000 | 0.052 | 0.051 | 0.086 | 0.346 |
| 100 | 0.020 | 0.020 | 0.052 | 0.052 | 0.089 | 0.346 |
| 100 | 0.400 | 0.000 | 0.052 | 0.051 | 0.086 | 0.346 |
| 100 | 0.400 | 0.020 | 0.052 | 0.052 | 0.089 | 0.346 |
| 100 | 0.400 | 0.400 | 0.054 | 0.056 | 0.658 | 0.759 |
| 300 | 0.000 | 0.000 | 0.050 | 0.051 | 0.374 | 0.998 |
| 300 | 0.020 | 0.000 | 0.050 | 0.051 | 0.374 | 0.998 |
| 300 | 0.020 | 0.020 | 0.050 | 0.051 | 0.401 | 0.999 |
| 300 | 0.400 | 0.000 | 0.050 | 0.051 | 0.374 | 0.998 |
| 300 | 0.400 | 0.020 | 0.050 | 0.051 | 0.401 | 0.999 |
| 300 | 0.400 | 0.400 | 0.056 | 0.052 | 1.000 | 1.000 |
| | | | Lag: 4 | | | |
| 100 | 0.000 | 0.000 | 0.046 | 0.051 | 0.078 | 0.267 |
| 100 | 0.020 | 0.000 | 0.046 | 0.051 | 0.078 | 0.267 |
| 100 | 0.020 | 0.020 | 0.046 | 0.050 | 0.078 | 0.264 |
| 100 | 0.400 | 0.000 | 0.046 | 0.051 | 0.078 | 0.267 |
| 100 | 0.400 | 0.020 | 0.046 | 0.050 | 0.078 | 0.264 |
| 100 | 0.400 | 0.400 | 0.054 | 0.055 | 0.507 | 0.584 |
| 300 | 0.000 | 0.000 | 0.048 | 0.051 | 0.331 | 0.993 |
| 300 | 0.020 | 0.000 | 0.048 | 0.051 | 0.331 | 0.993 |
| 300 | 0.020 | 0.020 | 0.048 | 0.051 | 0.357 | 0.994 |
| 300 | 0.400 | 0.000 | 0.048 | 0.051 | 0.331 | 0.993 |
| 300 | 0.400 | 0.020 | 0.048 | 0.051 | 0.357 | 0.994 |
| 300 | 0.400 | 0.400 | 0.050 | 0.052 | 1.000 | 1.000 |

# 5 Example

In this section, I use the `sroot` command to test for the presence of the unit root at seasonal frequency for the series of consumption in the UK. The data are from the National Institute of Statistics for the years 1955–2006 on a quarterly basis. I first test for the presence of a unit root for NSA data:

```
. sroot x_nsa,lag(4) trend season(quarter) regress gen(pi1 pi2 pi3)
HEGY test for SEASONAL unit roots                 Number of obs   =       200
                 Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t) - Fr 0       0.330           -4.050            -3.490            -3.180
 Z(t) - Fr 1/2     2.739           -3.520            -2.910            -2.600
 Z(t) - L.Ann.     1.014           -4.040            -3.410            -3.100
 Z(t) - Annual    -0.023           -2.650            -1.920            -1.480
 Joint Annual      0.514            8.960             6.570             5.560
 All SEAS. fr.     2.842                .             5.890             5.100
 All freq.         2.166                .             6.380             5.610
------------------------------------------------------------------------------
------------------------------------------------------------------------------
x_nsa            |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------+------------------------------------------------------------
       x_nsa     |
     Freq.0      |   .0001847   .0005594     0.33   0.742    -.0009189    .0012882
    Freq.1/2     |   .0461743   .0168582     2.74   0.007     .0129175    .0794311
    L.Annual     |   .0154886   .0152743     1.01   0.312    -.0146435    .0456206
      Annual     |  -.0003434   .0152415    -0.02   0.982    -.0304109    .0297241
         LD.     |   .7711865   .0777132     9.92   0.000     .6178793    .9244938
        L2D.     |   .0894235   .0952005     0.94   0.349    -.0983814    .2772285
        L3D.     |   .1412389   .0953463     1.48   0.140    -.0468536    .3293314
        L4D.     |  -.2178033   .0753951    -2.89   0.004    -.3665376   -.069069
      _trend     |   .0117578   .0031245     3.76   0.000      .005594    .0179216
          Q1     |  -.1432412   .1811381    -0.79   0.430    -.5005779    .2140954
          Q2     |  -.0581299   .1837385    -0.32   0.752    -.4205965    .3043368
          Q3     |   .1162713   .1815115     0.64   0.523    -.2418021    .4743447
       _cons     |  -.3112752   .1896996    -1.64   0.103    -.6855015    .0629511
------------------------------------------------------------------------------
```

Because I specified `regress`, the result has two main pieces, i.e., the test in the upper panel and the regression table in the lower panel. Let's start from the lower panel for clarity. It is helpful to have a look at regression results because there are four important components. The first four regressors are crucial for the test statistics. The second component is the set of lagged values, which are included in an attempt to remove serial correlation in $\epsilon_{it}$. Third are the deterministic components, namely, a `_trend` and a set of seasonal dummies. The set of seasonal dummies automatically drops the last quarter because of multicollinearity. The user may either specify the *varname* for quarter or specify directly for the complete set of dummies. Fourth, there is the constant term.

We are mainly interested in the upper panel, which is intrinsically tied to the lower panel. In particular, the $t$ statistics of the variables `Freq.0`–`Annual` from the lower panel will be the very same numbers that we find in test statistics in the upper panel. However, because the distribution is nonstandard, we also report the critical values at

some sensible confidence level, namely, 1, 5, and 10%. Further, for frequency zero and frequency 1/2, we can rely on the significance level for single coefficients, whereas for `L.Ann.` and `Annual`, we should test that their coefficients are jointly equal to zero, as we do in the line `Joint Annual`, along with their own critical values. The last two lines test the joint hypotheses that all *seasonal* coefficients are zero, i.e., the presence of seasonal unit roots, and that all relevant unit root coefficients are zero, i.e., full set of unit roots at all frequencies. In these cases, critical values are available only for 5 and 10% confidence levels.

In what follows, I interpret these numbers.

According to the $t$ statistics from `Freq.0`, we do not reject that $\pi_1$ is different from zero at a conventional confidence level. Equivalently, we cannot reject that the time series has a unit root at frequency zero. According to section 2, test statistics and critical values for this frequency could have been obtained from those already tabulated from the Dickey–Fuller test, and most importantly, the decision is based on the same rule.

For frequency $\pi/2$, we do not reject the presence of a (seasonal) unit root at, say, the 95% confidence level. This is because the alternative hypothesis concerning $\pi_2$ in (4) is stationarity, or $\pi_2 < 0$; thus values of the $t$ statistic smaller than the critical values at the preferred confidence level reject the null hypothesis of unit root. Vice versa, values of the $t$ statistic larger than the critical values at the preferred confidence level do not reject the null hypothesis of unit root. Here the $t$ statistic is 2.739 against a critical value of $-2.910$ at a 5% confidence level, and thus we cannot reject the presence of a (seasonal) unit root.

In (4), we have the annual frequency and its lag, and in principle they can return contrasting results. However, from section 2 we know that results depend on the joint test on coefficients. Being an $F$-type statistic, we reject the null hypothesis in cases where the test statistic is larger than the critical value. For the example at hand, we cannot reject the unit root at the annual frequency based on the line `Joint Annual`.

The test for unit roots at all seasonal frequencies and the test for unit roots at all frequencies are also $F$-type; thus the decision is based on the same rule of the annual frequency. From the line `All SEAS. fr.`, we do not reject the joint significance of seasonal unit roots, and from the line `All freq.`, the joint significance of the full set of unit roots, at seasonal and nonseasonal frequencies.

The evidence indicates that UK consumption has a unit root at frequency zero, as could be inferred from the classical Dickey–Fuller test. The new `sroot` command indicates that there are two more roots, one at frequency 1/2 (or biannual) and the other at annual frequency. `hegy4` returns the same qualitative conclusions. In general, `hegy4` and `sroot` test statistics need not be equal because `hegy4` uses an automatic lag selection method unless `notest` is specified. In the case at hand, the sequential tests on lags of the dependent variable select only lags 1 and 4:

```
. hegy4 x_nsa, lag(1 2 3 4) det(strend)

HEGY Quarterly seasonal unit root test for x_nsa

Number of observations  : 200
Deterministic variables : Seasonal dummies + constant + trend
Lags tested: 1 2 3 4
Augmented by lags : 1 4

          Stat     5% critical   10% critical
------------------------------------------------------------------------------
t[Pi1]    0.354      -3.490        -3.180
t[Pi2]    2.812      -2.910        -2.600
t[Pi3]    0.589      -3.410        -3.100
t[Pi4]    0.414      -1.920        -1.480

F[3-4]    0.254       6.570         5.560
F[2-4]    2.731       5.890         5.100
F[1-4]    2.084       6.380         5.610
```

For comparison purposes, we repeat the Hylleberg et al. (1990) test for SA data. As expected, they have only one unit root, found at frequency zero:

```
. sroot x_sa, lag(4) trend season(quarter)
HEGY test for SEASONAL unit roots                 Number of obs   =       200
                 Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value            Value
------------------------------------------------------------------------------
 Z(t) - Fr 0       0.290           -4.050            -3.490           -3.180
 Z(t) - Fr 1/2    -4.784           -3.520            -2.910           -2.600
 Z(t) - L.Ann.    -5.310           -4.040            -3.410           -3.100
 Z(t) - Annual    -4.676           -2.650            -1.920           -1.480
 Joint Annual     28.976            8.960             6.570            5.560
 All SEAS. fr.    32.859                .             5.890            5.100
 All freq.        24.734                .             6.380            5.610
```

The actual existence of seasonal unit roots in the series of consumption sheds more light on the potentially dramatic impact that a suboptimal econometric technique can have on a test of an economic theory. In this sense, the results from `sroot` are important per se. However, we can go a step further.

In particular, the `generate()` option is crucial to consider one possible extension of the unit root at seasonal frequencies, namely, cointegration at seasonal frequencies. Indeed, the option stores three different variables obtained from the transformation employed in the procedure. We just plot the transformed series in figure 1 as they are generated by `sroot` (i.e., with no editing adjustment). Although not pursued here, notice that the option allows the replication of the procedure by Engle et al. (1993) to fit a seasonal vector error-correction model.

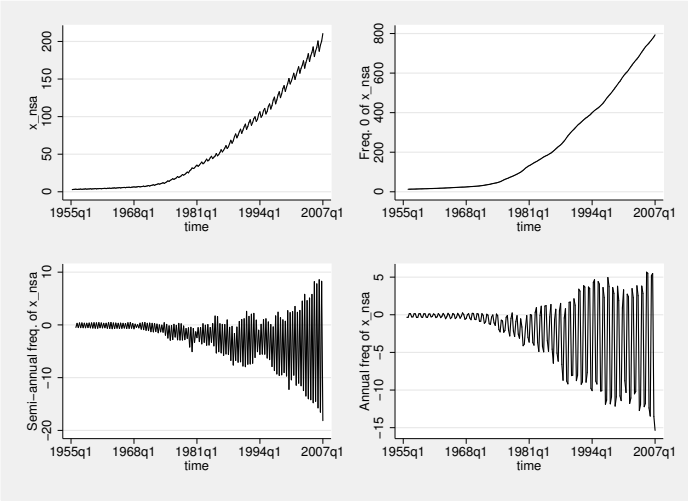



Figure 1. Series of consumption and its transformations

# 6 Conclusion

In this article, I presented the new `sroot` command, which implements a test to detect unit roots at frequencies other than zero, in quarterly data. The motivation for the new command is that many time series may have seasonal unit roots. Although the usual practice is to work with seasonally adjusted data, I view this as a weak solution because fluctuations do contain information and because adjustments can be responsible for rejection of economic theories even though the underlying model is correct. I argue that one should always consider using seasonally unadjusted data, which can be characterized by seasonal unit roots. It is important to go beyond the classical test at frequency zero, as I propose with `sroot`, paying much attention to the model specification. Finally, a promising extension is cointegration at seasonal frequencies that can be studied by exploiting the `generate()` option in `sroot`, even though more efficient methods are available in the literature.

# 7 Acknowledgments

I would like to thank Christopher Baum for stimulating suggestions and comments and Lucia Corno for comments on earlier drafts of the article. I also thank an anonymous referee and the editor for their suggestions. The article was written while I was completing my PhD at the Universitá degli Studi di Roma "Tor Vergata", where I benefited from the stimulating environment. All errors are mine.

# 8 References

Baum, C. F., and R. Sperling. 2001. HEGY4: Stata module to compute Hylleberg et al. seasonal unit root test. Statistical Software Components, Boston College Department of Economics. http://ideas.repec.org/c/boc/bocode/s416502.html.

Cubadda, G. 2001. Complex reduced rank models for seasonally cointegrated time series. *Oxford Bulletin of Economics and Statistics* 63: 497–511.

Dickey, D. A., and W. A. Fuller. 1979. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association* 74: 427–431.

Dickey, D. A., D. P. Hasza, and W. A. Fuller. 1984. Testing for unit roots in seasonal time series. *Journal of the American Statistical Association* 79: 355–367.

Engle, R. F., and C. W. J. Granger. 1987. Co-integration and error correction: Representation, estimation, and testing. *Econometrica* 55: 251–276.

Engle, R. F., C. W. J. Granger, S. Hylleberg, and H. S. Lee. 1993. Seasonal cointegration: The Japanese consumption function. *Journal of Econometrics* 55: 275–298.

Franses, P. H., and B. Hobijn. 1997. Critical values for unit root tests in seasonal time series. *Journal of Applied Statistics* 24: 25–48.

Fuller, W. A. 1976. *Introduction to Statistical Time Series*. New York: Wiley.

Ghysels, E., H. S. Lee, and J. Noh. 1994. Testing for unit roots in seasonal time series: Some theoretical extensions and a Monte Carlo investigation. *Journal of Econometrics* 62: 415–442.

Hylleberg, S., R. F. Engle, C. W. J. Granger, and B. S. Yoo. 1990. Seasonal integration and cointegration. *Journal of Econometrics* 44: 215–238.

Johansen, S., and E. Schaumburg. 1999. Likelihood analysis of seasonal cointegration. *Journal of Econometrics* 88: 301–339.

Lee, L.-F. 1995. Semiparametric maximum likelihood estimation of polychotomous and sequential choice models. *Journal of Econometrics* 65: 381–428.

Miron, J. A. 1986. Seasonal fluctuations and the life cycle—permanent income model of consumption. *Journal of Political Economy* 94: 1258–1279.

Osborn, D. R. 1988. Seasonality and habit persistence in a life cycle model of consumption. *Journal of Applied Econometrics* 3: 255–266.

**About the author**

Domenico Depalo received his PhD in econometrics and empirical economics from the Universitá degli Studi di Roma "Tor Vergata". After attending a postdoctoral program at the Sapienza–Universitá di Roma, he is currently a junior researcher at the Bank of Italy. The views expressed in the article are those of the author and do not involve the responsibility of the bank.

The Stata Journal (2009)
**9**, Number 3, pp. 439–453

# Robust regression in Stata

Vincenzo Verardi[1]
University of Namur (CRED)
and Université Libre de Bruxelles (ECARES and CKE)
Rempart de la Vierge 8, B-5000
Namur, Belgium
vverardi@fundp.ac.be

Christophe Croux
K. U. Leuven, Faculty of Business and Economics
Naamsestraat 69, B-3000
Leuven, Belgium
christophe.croux@econ.kuleuven.be

**Abstract.** In regression analysis, the presence of outliers in the dataset can strongly distort the classical least-squares estimator and lead to unreliable results. To deal with this, several robust-to-outliers methods have been proposed in the statistical literature. In Stata, some of these methods are available through the `rreg` and `qreg` commands. Unfortunately, these methods resist only some specific types of outliers and turn out to be ineffective under alternative scenarios. In this article, we present more effective robust estimators that we implemented in Stata. We also present a graphical tool that recognizes the type of detected outliers.

**Keywords:** st0173, mmregress, sregress, msregress, mregress, mcd, S-estimators, MM-estimators, outliers, robustness

## 1 Introduction

The objective of linear regression analysis is to study how a dependent variable is linearly related to a set of regressors. In matrix notation, the linear regression model is given by

$$y = X\theta + \varepsilon$$

where, for a sample of size $n$, $y$ is the $n \times 1$ vector containing the values for the dependent variable, $X$ is the $n \times p$ matrix containing the values for the $p$ regressors, and $\varepsilon$ is the $n \times 1$ vector containing the error terms. The $p \times 1$ vector $\theta$ contains the unknown regression parameters and needs to be estimated. On the basis of the estimated parameter $\widehat{\theta}$, it is then possible to fit the dependent variable by $\widehat{y} = X\widehat{\theta}$ and compute the residuals $r_i = y_i - \widehat{y}_i$ for $i = 1 \leq i \leq n$. Although $\theta$ can be estimated in several ways, the underlying idea is always to try to get as close as possible to the true value by reducing the magnitude of the residuals, as measured by an aggregate prediction error. For the

1. Vincenzo Verardi is an associated researcher of the FNRS and gratefully acknowledges their financial support.

well-known ordinary least squares (OLS), this aggregate prediction error is defined as the sum of squared residuals. The vector of parameters estimated by OLS is then

$$\widehat{\theta}_{\text{OLS}} = \arg\min_{\theta} \sum_{i=1}^{n} r_i^2(\theta)$$

with $r_i(\theta) = y_i - \theta_0 - \theta_1 X_{i1} - \cdots - \theta_p X_{ip}$ for $1 \leq i \leq n$. This estimation can be performed in Stata by using the `regress` command. A drawback of OLS is that by considering squared residuals, it tends to award an excessive importance to observations with very large residuals and, consequently, distort parameters' estimation in case of the existence of outliers.

The scope of this article is, first, to describe regression estimators that are robust with respect to outliers and, second, to propose Stata commands to implement them in practice. The structure of the article is the following: in the next section, we briefly present the types of outliers that can be found in regression analysis and introduce the basics of robust regression. We recommend using estimators with a high breakdown point, which are known to be resistant to outliers of different types. In section 3, we describe them and provide a sketch of the Stata code we implemented to estimate them in practice. In section 4, we give an example using the well-known Stata `auto.dta` dataset. In section 5, we provide some simulation results to illustrate how the estimators with a high breakdown point outperform the robust estimators available in Stata. Finally, in section 6, we conclude.

## 2 Outliers and robust regression estimators

In regression analysis, three types of outliers influence the OLS estimator. Rousseeuw and Leroy (2003) define them as *vertical outliers*, *bad leverage points*, and *good leverage points*. To illustrate this terminology, consider a simple linear regression as shown in figure 1 (the generalization to higher dimensions is straightforward). Vertical outliers are those observations that have outlying values for the corresponding error term (the $y$ dimension) but are not outlying in the space of explanatory variables (the $x$ dimension). Their presence affects the OLS estimation and, in particular, the estimated intercept. Good leverage points are observations that are outlying in the space of explanatory variables but that are located close to the regression line. Their presence does not affect the OLS estimation, but it affects statistical inference because they do deflate the estimated standard errors. Finally, bad leverage points are observations that are both outlying in the space of explanatory variables and located far from the true regression line. Their presence significantly affects the OLS estimation of both the intercept and the slope.

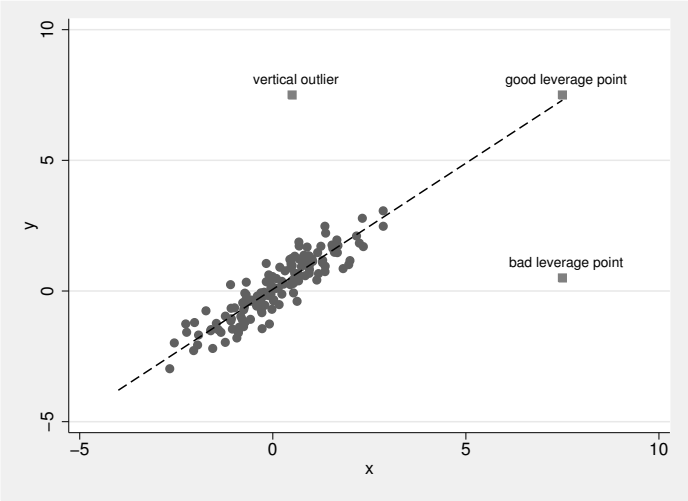



Figure 1. Outliers in regression analysis

Edgeworth (1887) realized that because of the squaring of the residuals, OLS becomes extremely vulnerable to the presence of outliers. To cope with this, he proposed a method consisting of minimizing the sum of the absolute values of the residuals rather than the sum of their squares. More precisely, his method defines the $L_1$, or *median regression*, estimator as

$$\widehat{\theta}_{L_1} = \arg\min_{\theta} \sum_{i=1}^{n} \mid r_i(\theta) \mid \tag{1}$$

The median regression estimator is available with Stata's offical `qreg` command. This estimator protects against vertical outliers but not against bad leverage points. It has an efficiency of only 64% at a Gaussian error distribution (see Huber [1981]).

Huber (1964) generalized median regression to a wider class of estimators, called M-estimators, by considering functions other than the absolute value in (1). This allows an increase in Gaussian efficiency while keeping robustness with respect to vertical outliers. An M-estimator is defined as

$$\widehat{\theta}_{M} = \arg\min_{\theta} \sum_{i=1}^{n} \rho\left\{\frac{r_i(\theta)}{\sigma}\right\} \tag{2}$$

where $\rho(\cdot)$ is a loss function, which is even, nondecreasing for positive values and less increasing than the square function. To guarantee scale equivariance (i.e., independence with respect to the measurement units of the dependent variable), residuals are standardized by a measure of dispersion $\sigma$. M-estimators are called monotone if $\rho(\cdot)$ is convex over the entire domain and redescending if $\rho(\cdot)$ is bounded.

The practical implementation of M-estimators uses an iteratively reweighted OLS algorithm. To simplify, suppose that $\sigma$ is known, and define weights as $\omega_i = \rho(r_i/\sigma)/r_i^2$. Then (2) can be rewritten as

$$\widehat{\theta}_M = \arg\min_{\theta} \sum_{i=1}^{n} \omega_i r_i^2(\theta)$$

which is a weighted OLS estimator. The weights $\omega_i$ are, however, a function of $\theta$ and are thus unknown. Using an initial estimate $\widetilde{\theta}$ for $\theta$, the weights can be computed and serve as the start of an iteratively reweighted OLS algorithm. Unfortunately, the latter is guaranteed to converge to the global minimum of (2) only for monotone M-estimators, which are not robust with respect to bad leverage points.

In Stata, the `rreg` command computes a highly efficient M-estimator. The loss function used is the Tukey biweight function defined as

$$\rho(u) = \begin{cases} 1 - \left\{1 - \left(\frac{u}{k}\right)^2\right\}^3 & \text{if } |\,u\,| \le k \\ 1 & \text{if } |\,u\,| > k \end{cases} \tag{3}$$

where $k = 4.685$. The starting value of the iterative algorithm $\widetilde{\theta}$ is taken to be a monotone M-estimator with a Huber $\rho(\cdot)$ function:

$$\rho(u) = \begin{cases} \frac{1}{2}(u)^2 & \text{if } |\,u\,| \le c \\ c\,|\,u\,| - \frac{1}{2}c^2 & \text{if } |\,u\,| > c \end{cases}$$

where $c = 1.345$. Moreover, to give protection against bad leverage points, observations associated with Cook distances larger than 1 receive a weight of zero. A command (`mregress`) to compute a standard monotone M-estimator with a Huber $\rho(\cdot)$ function is described in section 6.

Unfortunately, the `rreg` command does not have the expected robustness properties for two main reasons. First, Cook distances only manage to identify isolated outliers and are inappropriate when clusters of outliers exist, where one outlier can mask the presence of another (see Rousseeuw and van Zomeren [1990]). It can therefore not be guaranteed to have identification of all leverage points. Second, the initial values for the iteratively reweighted OLS algorithm are monotone M-estimators that are not robust to bad leverage points and that may lead the algorithm to converge to a local instead of a global minimum.

## 3 Estimators with a high breakdown point

Full robustness can be achieved by tackling the regression problem from a different perspective. The OLS estimator is based on the minimization of the variance of the residuals. Hence, because the variance is highly sensitive to outliers, OLS is largely influenced as well. For this reason, Rousseeuw and Yohai (1984) propose to minimize

a measure of dispersion of the residuals that is less sensitive to extreme values than the variance.[2] They call this class of estimators the S-estimators. The intuition behind the method is simple. For OLS, the objective is to minimize the variance, $\widehat{\sigma}^2$, of the residuals. The latter can be rewritten as $1/n \sum_{i=1}^{n} \left(r_i/\widehat{\sigma}\right)^2 = 1$. As stated previously, the square value can be damaging because it gives a huge importance to large residuals. Thus, to increase robustness, the square function could be replaced by another loss function, $\rho$, that awards less importance to large residuals.[3] The estimation problem would now consist of finding the smallest robust scale of the residuals. This robust dispersion, denoted by $\widehat{\sigma}^S$, satisfies

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left\{\frac{r_i(\theta)}{\widehat{\sigma}^S}\right\} = b \tag{4}$$

where $b = E\{\rho(Z)\}$ with $Z \sim N(0,1)$. The value of $\theta$ that minimizes $\widehat{\sigma}^S$ is then called an S-estimator. More formally, an S-estimator is defined as

$$\widehat{\theta}^S = \arg\min_{\theta} \ \widehat{\sigma}^S \left\{r_1(\theta), \ldots, r_n(\theta)\right\} \tag{5}$$

where $\widehat{\sigma}^S$ is the robust estimator of scale as defined in (4).

The choice of $\rho(\cdot)$ is crucial to have good robustness properties and a high Gaussian efficiency. The Tukey biweight function defined in (3), with $k = 1.547$, is a common choice. This S-estimator resists contamination of up to 50% of outliers; it is said to have a breakdown point of 50%. Unfortunately, this S-estimator has a Gaussian efficiency of only 28.7%. If $k = 5.182$, the Gaussian efficiency rises to 96.6%, but the breakdown point drops to 10%. To cope with this, Yohai (1987) introduced MM-estimators that combine a high breakdown point and a high efficiency. These estimators are redescending M-estimators as defined in (2), but with the scale fixed at $\widehat{\sigma}^S$. So an MM-estimator is defined as

$$\widehat{\theta}^{MM} = \arg\min_{\theta} \sum_{i=1}^{n} \rho\left\{\frac{r_i(\theta)}{\widehat{\sigma}^S}\right\} \tag{6}$$

The preliminary S-estimator guarantees a high breakdown point, and the final MM-estimate guarantees a high Gaussian efficiency. It is common to use a Tukey biweight $\rho(\cdot)$ function for both the preliminary S-estimator and the final MM-estimator. The tuning constant $k$ can be set to 1.547 for the S-estimator to guarantee a 50% breakdown point, and it can be set to 4.685 for the second-step MM-estimator in (6) to guarantee a 95% efficiency of the final estimator.

2. The least trimmed squares estimator and the least median squares estimator, introduced by Rousseeuw (1984) rely on the same logic. We programmed these two estimators in Stata and made them available through the `ltsregress` and `lmsregress` commands. `ltsregress` and `lmsregress` are available from the authors upon request.
3. As before, $\rho(\cdot)$ is a function that is even, nondecreasing for positive values, less increasing than the square with a unique minimum at zero.

For computing the MM-estimator, the iteratively reweighted OLS algorithm can be used, taking $\widehat{\theta}^S$ as its initial value. Once the initial S-estimate is computed, $\widehat{\theta}^{MM}$ comes at almost no additional computational cost. We programmed an S- and an MM-estimator in Stata (with Tukey biweight loss function) using the algorithm of Salibian-Barrera and Yohai (2006). Explicit formulas for the estimators are not available, and it is necessary to call on numerical optimization to compute them. We present, in the next section, a sketch of the algorithm we implemented in Stata. The commands to compute S- and MM-estimators (called `sregress` and `mmregress`, respectively) are described in section 6.

## 3.1 S-estimator and MM-estimator algorithms

The algorithm implemented in Stata for computing the S-estimator starts by randomly picking $N$ subsets of $p$ observations (defined as $p$-subset), where $p$ is the number of regression parameters to estimate. For each $p$-subset, the equation of the hyperplane that fits all points perfectly is obtained, yielding a trial solution of (5). This trial value is more reliable if all $p$ points are regular observations, such that the $p$-subset does not contain outliers. The number $N$ of subsamples to generate is chosen to guarantee that at least one $p$-subset without outliers is selected with high probability. As shown in Salibian-Barrera and Yohai (2006), this can be achieved by taking

$$N = \left\lceil \frac{\log(1 - P_{\text{clean}})}{\log\{1 - (1 - \alpha)^p\}} \right\rceil \tag{7}$$

where $\alpha$ is the (maximal) expected proportion of outliers, $p$ is the number of parameters to estimate, and $P_{\text{clean}}$ is the desired probability to have at least one $p$-subset without outliers among the $N$ subsamples.[4]

For each of the $p$-subsets, a hyperplane that perfectly fits the $p$-subset is computed. Then, for all $n$ observations in the sample, residuals with respect to this hyperplane are computed, and a scale estimate, $\widehat{\sigma}^S$, is computed from them as in (4). In this way, scale estimates are obtained for each $p$-subset, and an approximation for the final scale estimate, $\widehat{\sigma}^S$, is then given by the trial value that leads to the smallest scale over all $p$-subsets. This approximation can be improved further by carrying some refinement steps that bring the approximation even closer to the solution of (5).

This algorithm is implemented in Stata and can be called either directly using the `sregress` command or indirectly using the `mmregress` command and invoking the `initial` option. Once the S-estimator is obtained, the MM-estimator directly follows by applying the iteratively reweighted OLS algorithm up to convergence. We provide a Stata command for MM-estimators through the `mmregress` command. As far as inference is concerned, standard errors robust to heteroskedasticity (and asymmetric errors) are computed according to the formulas available in the literature (see, e.g., Croux, Dhaene, and Hoorelbeke [2008]).

4. The default values we use in the implementation of the algorithm are $\alpha = 0.2$ and $P_{\text{clean}} = 0.99$.

The need of calling on subsampling algorithms becomes the Achilles' heel of the algorithm when several dummy variables are present. Indeed, as stated by Maronna and Yohai (2000), subsampling algorithms can easily lead to collinear subsamples if various dummies are among the regressors. To cope with this, Maronna and Yohai (2000) introduce the MS-estimator that alternates an S-estimator (for continuous variables) and an M-estimator (for dummy ones) till convergence. This estimator is out of the scope of this article, and we thus do not elaborate on it here. We nevertheless briefly describe the Stata command implemented to compute it in practice (`msregress`). This estimator can be particularly helpful in the fixed-effects panel-data models, as suggested by Bramati and Croux (2007).

### 3.2 Outlier detection

In addition to reducing the importance of outliers on the estimator, robust statistics are also intended to identify atypical individuals. Once identified, they could be analyzed separately from the bulk of the data. To do so, it is important to recognize their type. This can be easily achieved by calling on the graphical tool proposed by Rousseeuw and van Zomeren (1990). This graphical tool is constructed by plotting, on the vertical axis, the robust standardized residuals, defined as $r_i/\widehat{\sigma}^S$, with $r_i \equiv r_i(\widehat{\theta}^S)$, to give an idea of outlyingness with respect to the fitted regression plane. On the horizontal axis, a measure of the (multivariate) outlyingness of the explanatory variables is plotted. The latter is measured by Mahalanobis distance defined as $d_i = \sqrt{(X_i - \mu)\Sigma^{-1}(X_i - \mu)'}$, where $\mu$ is the multivariate location vector, $\Sigma$ is the covariance matrix of the explanatory variables, and $X_i$ is the $i$th row vector of matrix $X$, for $1 \leq i \leq n$. Obviously, both $\mu$ and $\Sigma$ should be estimated robustly if we want these distances to resist the presence of outliers. Several methods have been proposed to robustly estimate the Mahalanobis distances. In Stata, the `hadimvo` command is available, but more robust estimates for the covariance matrix (such as the minimum covariance determinant estimator) are also available. We briefly describe the command (`mcd`) to compute the minimum covariance determinant in section 6.

It is possible to set the limits outside which individuals can be considered as outliers. For the $y$ dimension, we set them to $-2.25$ and $+2.25$. These represent the values of the standard normal that separate the 2.5% remotest area of the distribution from the central mass. For the $x$ dimension, we set the limit to $\sqrt{\chi^2_{p,0.975}}$, motivated by the fact that the squared Mahalanobis distance is $\chi^2_p$ distributed under normality.

## 4 Example

To illustrate the usefulness of the robust methods, we present an example based on the well-known Stata `auto.dta` dataset. More specifically, we regress the price of cars on the following set of characteristics: the mileage (mpg), the headroom (in.), the trunk space (cu. ft.), the length (in.), the weight (lbs.), the turn circle (ft.), the displacement (cu. in.), the gear ratio, four dummies identifying the categorical variable repair record

in 1978, and a foreign dummy identifying whether the car was built in the United States. We first identify outliers. For this purpose, we call on the graphical tool described in section 3.2. The resulting plot is pictured in figure 2. This can be easily replicated by typing the following Stata commands (which are described more precisely in section 6).

```
. use http://www.stata-press.com/data/r11/auto
(1978 Automobile Data)
. xi: mmregress price mpg headroom trunk length weight turn displacement
> gear_ratio foreign i.rep78, outlier graph label(make)
```

(*output omitted*)

Several features emerge. First, the Cadillac Seville is a bad leverage point. Indeed, it is an outlier in the horizontal as well as in the vertical dimension. This means that its characteristics are pretty different from those of the bulk of the data and its price is much higher than it should be according to the fitted model. The Volkswagen Diesel and the Plymouth Arrow are large good leverage points because they are outlying in the horizontal dimension but not in the vertical one. This means that their characteristics are rather different from the other cars but their prices are in accordance with what the model predicts. Finally, the Cadillac Eldorado, the Lincoln Versailles, the Lincoln Mark V, the Volvo 260, and some others are standard in their characteristics but are more expensive than the model would suggest. They correspond to vertical outliers.

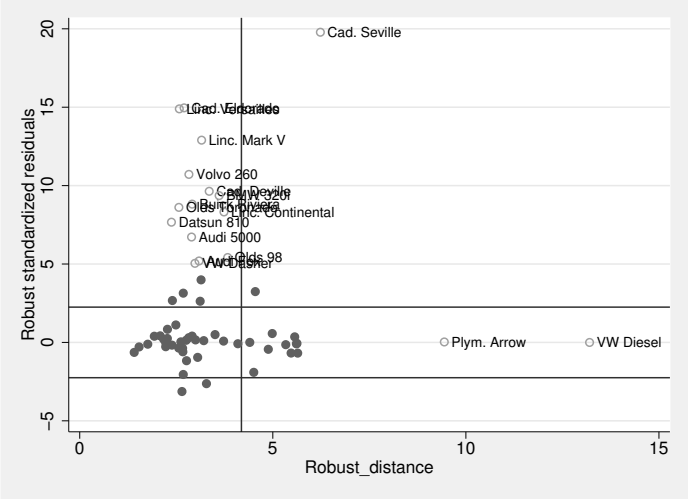



Figure 2. Diagnostic plot of standardized robust residuals versus robust Mahalanobis distances for the `auto.dta` dataset

Are these outlying observations sufficient to distort classical estimations? Because several vertical outliers are present as well as a severe bad leverage point, there is a serious risk that the OLS estimator becomes strongly attracted by the outliers. To illustrate this, we compare the results obtained by using the recommended estimator

with a high breakdown point, `mmregress`, with those obtained by using OLS (`regress`), Huber's monotonic M-estimator (`rreg`), and median regression (`qreg`). MM-estimators with 70% and with 95% efficiency (for normal errors) are considered. The commands (used in a do-file) to estimate these models are

```
. webuse auto, clear
. local exogenous="mpg headroom trunk length weight turn displacement
> gear_ratio foreign i.rep78"
. xi: regress price `exogenous'
. xi: qreg price `exogenous'
. xi: rreg price `exogenous'
. xi: mmregress `exogenous', eff(0.7)
. xi: mmregress `exogenous', eff(0.95)
```

The differences are, as expected, important. We present the regression output in table 1.

(*Continued on next page*)

Table 1: Pricing of autos

Auto dataset. Dependent variable: Price in US$

| | `regress` | `qreg` | `rreg` | MM(0.70) | MM(0.95) |
|---|---|---|---|---|---|
| Mileage | −43.95 | −44.45 | −68.91 | −44.88 | −46.74 |
| | (0.52) | (0.55) | (0.92) | (−1.67) | (1.56) |
| Headroom | −689.40* | −624.19* | −739.30** | −311.96** | −440.06*** |
| | (1.72) | (1.71) | (2.09) | (2.52) | (4.10) |
| Trunk space | 74.29 | 37.50 | 114.53 | 186.60*** | 128.98*** |
| | (0.74) | (0.40) | (1.29) | (7.10) | (3.53) |
| Length | −80.66* | −48.78 | −27.50 | −33.74** | 0.03 |
| | (1.86) | (1.17) | (0.72) | (2.57) | (0.00) |
| Weight | 4.67*** | 2.89** | 2.59* | 1.03*** | 0.37 |
| | (3.19) | (2.10) | (1.99) | (5.29) | (0.62) |
| Turn circle | −143.71 | 30.22 | −104.26 | 10.51 | −23.79 |
| | (1.11) | (0.30) | (0.91) | (0.48) | (0.69) |
| Displacement | 12.71 | 9.79 | 11.34 | 2.31 | 2.51 |
| | (1.45) | (1.27) | (1.46) | (0.98) | (0.58) |
| Gear ratio | 115.08 | 92.28 | 917.19 | 492.467 | 370.20 |
| | (0.09) | (0.08) | (0.82) | (0.89) | (0.99) |
| Foreign | 3064.52*** | 2496.04** | 2326.91** | −91.66 | 763.91* |
| | (2.89) | (2.38) | (2.48) | (0.19) | (1.89) |
| `rep78==2` | 1353.80 | −355.92 | 465.98 | 5.99 | 31.45 |
| | (0.79) | (0.27) | (0.31) | (0.02) | (0.11) |
| `rep78==3` | 955.44 | 19.24 | 488.23 | −720.50*** | −286.70 |
| | (0.59) | (0.02) | (0.34) | (2.76) | (1.17) |
| `rep78==4` | 976.63 | 241.79 | 813.11 | −275.89 | 390.71 |
| | (0.59) | (0.18) | (0.55) | (1.04) | (1.49) |
| `rep78==5` | 1758.00 | 1325.18 | 1514.13 | 606.77* | 359.01 |
| | (0.97) | (0.91) | (0.95) | (1.70) | (0.86) |
| Constant | 9969.75 | 4083.51 | 2960.68 | 5352.18*** | 3495.97 |
| | (1.40) | (0.60) | (0.47) | (3.10) | (1.43) |

Absolute value of $t$ statistics is in parentheses.
Significant at ***1%, **5%, and *10%.

Let's compare the results. First, headroom, trunk space, and length seem to be unimportant in explaining prices (at a 5% level) when looking at the OLS, median, and M-estimators (i.e., `regress`, `qreg`, and `rreg`). However, when the influence of outliers (and especially of the bad leverage point) is taken into account (i.e., `MM(0.7)` column), they turn out to be significantly different to zero. If we consider a more efficient estimator (i.e., `MM(0.95)` column), length again becomes insignificant. The weight variable is flagged as significant by most specifications (though the size of the effect is very different). The turn, displacement, and gear ratio variables turn out to be insignificant in all specifications. The foreign dummy is insignificant when using only the most robust estimators.

# 5 Simulations

Several recent articles have proven the theoretical properties of the estimators described in the previous sections. In this article, we will compare the performances of the Stata codes we implemented with the previously available robust commands and OLS. To do so, we run some simulations according to the following setup. We start by creating a dataset (of size $n = 1{,}000$) by randomly generating five independent explanatory continuous variables (labeled $X_1, \ldots, X_5$) and an error term ($e$) from six independent univariate normal distributions with mean zero and unit variance. A $y$ variable is then generated according to the formula $y_i = \beta_0 + \sum_{j=1}^{5} \beta_j X_{ij} + e_i$, where $\beta_0 = 0$ and $\beta_j = 1$ for $j = 1, \ldots, 5$. This dataset is called the clean dataset. We then contaminate the data by randomly replacing 10% of the $X_1$ observations without modifying $y$. These contaminated points are generated from a normal distribution with mean 5 and standard deviation 0.1 and are bad leverage points. We call this the contaminated dataset. We then repeat this procedure 1,000 times, and each time we estimate the parameters using OLS, $L_1$, M-estimators, S-estimators, and MM-estimators (with a 95% and a 70% efficiency). On the basis of all the estimated parameters, we measure the bias (i.e., the average of the estimated parameters minus the true value) and the mean squared error (MSE) (i.e., the variance of the estimated parameters plus the square of the bias). The results are presented in table 2. We do not present the results associated with the clean sample because all estimation methods lead to comparable and very low biases.

Table 2: Simulated bias and MSE (sample size $n = 1{,}000$, 10% of outliers)

| Estimation method | | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\beta_4$ | $\beta_5$ | $\beta_0$ |
|---|---|---|---|---|---|---|---|
| OLS | Bias | 0.7149 | 0.0015 | 0.0010 | 0.0002 | 0.0016 | −0.1440 |
| `reg` | MSE | 0.5118 | 0.0017 | 0.0018 | 0.0019 | 0.0018 | 0.0223 |
| $L_1$ | Bias | 0.6369 | 0.0006 | 0.0013 | 0.0004 | 0.0011 | −0.1281 |
| `qreg` | MSE | 0.4071 | 0.0026 | 0.0024 | 0.0027 | 0.0027 | 0.0188 |
| M | Bias | 0.6725 | 0.0012 | 0.0010 | 0.0005 | 0.00167 | −0.1353 |
| `rreg` | MSE | 0.4532 | 0.0018 | 0.0018 | 0.0019 | 0.0019 | 0.0200 |
| MM(0.95) | Bias | 0.6547 | 0.0011 | 0.0009 | 0.0010 | 0.00167 | −0.1318 |
| `mmregress` | MSE | 0.4298 | 0.0018 | 0.0018 | 0.0020 | 0.0020 | 0.0190 |
| MM(0.7) | Bias | 0.0867 | 0.0012 | 0.0028 | −0.0008 | −0.0010 | −0.0164 |
| `mmregress` | MSE | 0.0236 | 0.0015 | 0.0015 | 0.0015 | 0.0014 | 0.0024 |

The results of the simulations clearly show that for this contamination setup, the least biased estimator among those we considered is the MM-estimator with an efficiency of 70%. Its bias and MSE are 0.087 and 0.024, respectively, for $\beta_1$ and −0.016 and 0.002 for $\beta_0$. As a comparison, the bias and MSE of OLS are 0.715 and 0.512 for $\beta_1$ and −0.144 and 0.022 for $\beta_0$. For the other coefficients, the performances of all estimators are comparable. It is important to stress that if we set the efficiency of MM to 95%, its performance in terms of bias worsens too much and would thus not be desirable. The $L_1$ and M-estimators (computed respectively with the `qreg` and `rreg` commands) behave rather poorly and have a bias and an MSE comparable to that of OLS.

# 6 The implemented commands

The `mmregress` command computes the MM-estimators with a high breakdown point, described in section 3, and their standard errors. The general syntax for the command is

`mmregress` *depvar indepvars* [ *if* ] [ *in* ] [ `, noconstant eff(#)`
`dummies(`*dummies*`) outlier graph label(`*varname*`) replic(#) initial` ]

The optional parameter `eff()` fixes the efficiency of the MM-estimator. It can take any value between 0.287 and 1; the higher its value, the more efficient the MM-estimator. While the breakdown point of the MM-estimator is always 50%, its bias increases with its efficiency. Therefore, to have a good compromise between robustness and efficiency of the MM-estimator, we take `eff(0.7)` as a default. The `dummies()` option specifies which variables are dichotomous. If `dummies()` is declared, the initial estimator will be the MS-estimator rather than the S-estimator. Not declaring this option when dummy variables are present may cause the algorithm for computing the S-estimator to fail (see section 3.1).

The `noconstant` option specifies that no constant term has to be considered in the regression. The `outlier` option provides robust standardized residuals and robust Mahalanobis distances. These can be used to construct a diagnostic plot, as discussed in section 3.2. The `graph` option calls on this graphical tool for outlier identification. The `label()` option specifies the variable that will label the outlier. This option only works jointly with the `graph` option. If `label()` is not declared, the label will be the observation number.

The `replic()` option fixes the number of $p$-subsets to consider in the initial steps of the algorithm. The user can use (7) to change the value of $N$ in accordance to the desired level of $P_{\text{clean}}$ or $\alpha$. The default value for $N$ corresponds to $P_{\text{clean}} = 0.99$ and $\alpha = 0.2$. Finally, the `initial` option will return as output the initial S-estimator, or the MS-estimator if the `dummies()` option is invoked, instead of the final MM-estimator.

The general syntax for the command to compute the S-estimator is

`sregress` *depvar indepvars* [ *if* ] [ *in* ] [ `, noconstant outlier graph`
`replic(#)` ]

The optional parameters available are a subset of those available in `mmregress`; their use is therefore the same as described above. If `sregress` is called exclusively defining a dependent variable, the code will return an M-estimator of scale (sometimes called an S-estimator of scale) and an S-estimator of location of that variable.

The general syntax for the command to compute the MS-estimator is

`msregress` *depvar indepvars* [ *if* ] [ *in* ], `dummies(`*dummies*`)` [ `noconstant outlier graph replic(`#`)` ]

Here again the use of options is comparable to `mmregress`. The `dummies()` option is compulsory and is used to declare which variables among the explanatory are dichotomous.

The general syntax for the command to compute the Huber M-estimator is

`mregress` *depvar indepvars* [ *if* ] [ *in* ] [ , `noconstant tune(`#`) level(`#`)` ]

The `noconstant` option removes the constant, while the `tune()` option changes the tuning parameter as in Stata's `rreg` command. `mregress` is only a slight modification of the `rreg` command.

The general syntax for the minimum covariance determinant command is

`mcd` *varlist* [ *if* ] [ *in* ] [ , `e(`#`) proba(`#`) trim(`#`) outlier bestsample raw setseed(`#`)` ]

The `e()` and `proba()` options are used to modify $\alpha$ and $P_{\text{clean}}$, respectively, in (7); `trim()` sets the percentage of trimming desired; `outlier` calls for robust Mahalanobis distances and flags outliers; `bestsample` identifies the observations that have been used for calculating the robust covariance matrix; `raw` returns the raw robust covariance matrix estimated classically, but on the sample cleaned of identified outliers; and `setseed()` sets the seed. The algorithm for computing the minimum covariance determinant is described in Rousseeuw and van Driessen (1999).

# 7 Conclusion

The strong impact of outliers on the OLS regression estimator has been known for a long time. Consequently, much literature has been developed to find robust estimators that cope with the "atypical" observations and have a high breakdown point. At the same time, the statistical efficiency of the robust estimators needs to remain sufficiently high. In recent years, it seems that a consensus has emerged to recommend the MM-estimators as the best-suited estimation method, because they combine a high resistance to outliers and high efficiency for regression models with normal errors.

On the other hand, robust methods were not often used by applied researchers, mainly because their practical implementation remained quite cumbersome. Over the last decade, efficient and relatively fast algorithms for computing robust estimators, including MM-estimators, were developed. Nowadays, the use of robust statistical methods has become much more widespread in the applied sciences, like engineering

and chemistry. By providing the Stata code, we also make robust regression methods available for the econometrics research community.

In this article, we summarized the properties of the best-known robust estimation procedures and provided Stata commands to implement them. We created the `mmregress` command (based on a set of commands that can be run separately if needed); furthermore, we showed how this estimator outperforms all "robust" estimators available in Stata by means of a modest simulation study. We hope that this article will contribute to the development of further robust methods in Stata. In particular, development of robust procedures for panel-data and time-series models would be of major interest for applied economic research. The time-series setting will give rise to new problems; for example, selecting random $p$-subsets will not be appropriate because they break the temporal structure of the data.

# 8 References

Bramati, M. C., and C. Croux. 2007. Robust estimators for the fixed effects panel data model. *Econometrics Journal* 10: 521–540.

Croux, C., G. Dhaene, and D. Hoorelbeke. 2008. Robust standard errors for robust estimators. Unpublished manuscript. http://www.econ.kuleuven.be/ew/academic/econmetr/members/Dhaene/papers/rsejan2004.pdf.

Edgeworth, F. Y. 1887. On observations relating to several quantities. *Hermathena* 6: 279–285.

Huber, P. J. 1964. Robust estimation of a location parameter. *Annals of Mathematical Statistics* 35: 73–101.

———. 1981. *Robust Statistics*. New York: Wiley.

Maronna, R. A., and V. J. Yohai. 2000. Robust regression with both continuous and categorical predictors. *Journal of Statistical Planning and Inference* 89: 197–214.

Rousseeuw, P. J. 1984. Least median of squares regression. *Journal of the American Statistical Association* 79: 871–880.

Rousseeuw, P. J., and A. M. Leroy. 2003. *Robust Regression and Outlier Detection*. New York: Wiley.

Rousseeuw, P. J., and K. van Driessen. 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41: 212–223.

Rousseeuw, P. J., and B. C. van Zomeren. 1990. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association* 85: 633–639.

Rousseeuw, P. J., and V. J. Yohai. 1984. Robust regression by means of S-estimators. In *Robust and Nonlinear Time Series Analysis*, ed. J. Franke, W. Härdle, and R. D. Martin, 256–276. New York: Springer.

Salibian-Barrera, M., and V. J. Yohai. 2006. A fast algorithm for S-regression estimates. *Journal of Computational and Graphical Statistics* 15: 414–427.

Yohai, V. J. 1987. High breakdown-point and high efficiency robust estimates for regression. *Annals of Statistics* 15: 642–656.

**About the authors**

Vincenzo Verardi is an associate researcher of the FNRS (Belgian National Science Foundation) and a professor of economics and econometrics at the University of Namur and at the Université Libre de Bruxelles (Belgium). His research interests are applied econometrics, development economics, political economics, and public finance.

Christophe Croux is a professor of statistics and econometrics at the Katholieke Universiteit Leuven (Belgium). His research interests are robust statistics, multivariate data analysis, computational statistics, applied time-series analysis, and predictive marketing modeling. His work has been published in *Biometrika*, *Journal of the American Statistical Association*, *Journal of Multivariate Analysis*, *Journal of Marketing Research*, and *The Review of Economics and Statistics*, among other publications.

The Stata Journal (2009)
**9**, Number 3, pp. 454–465

# Nonparametric testing of distributions—the Epps–Singleton two-sample test using the empirical characteristic function

Sebastian J. Goerg
Max Planck Institute for Research on Collective Goods
Kurt-Schumacher-Straße 10
53113 Bonn, Germany
goerg@coll.mpg.de

Johannes Kaiser
Deutsche Bundesbank[1]
Wilhelm-Epstein-Straße 14
60431 Frankfurt, Germany
johannes.kaiser@bundesbank.de

**Abstract.** In statistics, two-sample tests are used to determine whether two samples have been drawn from the same population. An example of such a test is the widely used Kolmogorov–Smirnov two-sample test. There are other distribution-free tests that might be applied in similar occasions. In this article, we describe a two-sample omnibus test introduced by Epps and Singleton, which usually has a greater power than the Kolmogorov–Smirnov test although it is distribution free. The superiority of the Epps–Singleton characteristic function test is illustrated in two examples. We compare the two tests and supplement this contribution with a Stata implementation of the omnibus test.

**Keywords:** st0174, escftest, nonparametric tests, Kolomogorov–Smirnov, Epps–Singleton, two-sample case

## 1 Introduction

In many empirical scientific fields, statistical tests are used to determine whether two samples have been drawn from the same population. The commonly used procedure is to test the data in question against the null hypothesis, $H_0$, that the underlying distributions of the two samples are equal. The Kolmogorov–Smirnov two-sample (KS) test, the Wilcoxon–Mann–Whitney rank-sum (MW) test, and the Epps–Singleton (ES) test are examples of this approach. Implementations of the KS and MW tests are included in Stata. In this article, we introduce a Stata implementation of the ES test. The KS and ES tests are able to detect differences in distributions—be it by location, scale, or family. The MW test detects only locational shifts. The reason for this is its directional

1. Any opinions expressed in this paper are those of the authors and do not reflect the opinion of the Bundesbank.

alternative hypothesis, $H_1$, which states that the underlying distribution of one sample is stochastically larger than the underlying distribution of the other sample.

It has been shown by Epps and Singleton (1986) that the ES test is usually more powerful than the KS test. There exists one more advantage of the ES test over the KS test: An assumption of the KS test is that the data are drawn from a continuous distribution. Contrary to that, both continuous and discrete data may be used for the ES test. This also holds true for the MW test.

In the following, the rationale of the ES test is described. We next explain the syntax of the Stata implementation. Then we apply the tests to two examples and compare the results. Finally, we close with some short conclusions.

## 2 The ES test

In this section, we give a brief outline of the ES test and concentrate on the important relations and functions. Hereby, we limit our remarks to a description of the procedure and leave out details on proofs and derivations. The interested reader can find these details in the original paper by Epps and Singleton (1986).

The $p$-value of the ES test gives the probability of falsely rejecting $H_0$ that both samples have been drawn from the same population. It tests for dissimilarities by comparing the empirical characteristic functions, $\phi_1(t)$ and $\phi_2(t)$, of the two samples instead of the observed distributions, $F_1$ and $F_2$.

The empirical characteristic function is the Fourier transform of the observed distribution function. The characteristic function of a distribution can be used to conveniently derive its moments and thus contains more information than just one measure, like the mean, the median, or the variance. However, this also holds true for the probability density. Additionally, the use of the probability density is more intuitive than the use of the characteristic function. Epps (1993) describes the geometrical representation of the characteristic function as the center of mass of a distribution wrapped around the unit circle in the complex plane. These caveats raise doubts on the necessity of applying them: Why should one use the empirical characteristic function for statistical tests?

One advantage of the characteristic function is that it can be used as a representation of distributions whose probability densities cannot be specified. One example is the family of alpha-stable distributions introduced by Paul Lévy, where only three distributions (Gaussian, Cauchy, and Lévy) in closed form for densities are known. Typical applications for distributions whose forms are not closed are models with returns from stock markets (Epps 1993; Borak, Härdle, and Weron 2005).

Another advantage, and more relevant here, is that the characteristic function is completely defined for discrete and continuous data, while the distribution function is completely defined only for continuous data. For discrete data, the distribution function is defined only in certain points.

An important prerequisite for the application of the test is that all observations are independent, both within and across samples. The null hypothesis of the test states

$$H_0 : \phi_1(t) = \phi_2(t), \text{with} - \infty < t < \infty$$

The characteristic function is defined as $\phi_k(t) = \int_{-\infty}^{\infty} e^{itx} dF_{n_k}(x)$, where $t$ is a real number and $i = \sqrt{-1}$. For a sample, $k$, with a size of $n_k$, with $X_{km}$ denoting the $m$th observation in sample $k$, and a distribution function $F_{n_k}(x)$, the empirical characteristic function is defined as

$$\phi_{n_k}(t) = \int_{-\infty}^{\infty} e^{itx} dF_{n_k}(x) = n_k^{-1} \sum_{m=1}^{n_k} e^{itX_{km}}$$

To make use of the characteristic function for the ES test, a set of parameters $t_1, t_2, \ldots, t_J$ has to be chosen. For the sake of applicability, these parameters need to be calibrated to provide the test with a sufficient power against a broad class of alternatives. Epps and Singleton (1986) did simulations with nine different families of distributions[2] in 30 samples altogether. They found that with $t_1 = 0.4$ and $t_2 = 0.8$ ($J = 2$), the test performed optimally, conditional on their sample of 30 comparisons. In the following, we will briefly summarize the proceedings as described by Epps and Singleton (1986). For a more exhaustive description of the calibration, refer to their work.

The $t_j$ need to be standardized with an estimate of scale $\widehat{\sigma}$—Epps and Singleton (1986) claim that a sufficiently good scale measure for $\widehat{\sigma}$ is the semi-interquartile range. As a consequence, the test is carried out with $\widetilde{t_j} = t_j/\widehat{\sigma}, j = 1, 2$.

For each $X_{km}$, a $4 \times 1$ vector $g(X_{km})$ is created:

$$g(X_{km}) = (\cos\, t_1 X_{km}, \sin\, t_1 X_{km}, \cos\, t_2 X_{km}, \sin\, t_2 X_{km})'$$

Let $g_k$ contain the real and imaginary parts of the characteristic function of the sample for both $t_1$ and $t_2$:

$$g_k = n_k^{-1} \sum_{m=1}^{n_k} g(X_{km})$$

Let $G_2 = g_1 - g_2$ be the difference between both vectors. If $H_0$ was true, $\sqrt{n_1 + n_2} G_2$ would be distributed asymptotically as multivariate $N(\vec{0}, \Omega)$. Epps and Singleton derive an estimator for the covariance matrix $\Omega$. Let $\nu_k = n_k/(n_1 + n_2)$ be the share of sample $k$ in the combined sample and

$$\widehat{S}_k = \frac{n_k - 1}{n_k} \text{cov}\{g(X_{km})\}$$

be the sample covariance matrix of sample $k$. A sufficient estimator for $\Omega$ would then be

$$\widehat{\Omega} = \frac{1}{\nu_1} \widehat{S}_1 + \frac{1}{\nu_2} \widehat{S}_2$$

2. They chose normal, uniform, Cauchy, Laplace, symmetric stable, gamma, Poisson, binomial, and negative binomial distributions.

The test statistic of the ES test is defined as $W_2 = (n_1 + n_2) \cdot G_2' \cdot \widehat{\Omega}^+ \cdot G_2$ with $\widehat{\Omega}^+$ being the generalized inverse of $\widehat{\Omega}$. $W_2$ is distributed asymptotically as chi-squared with $r$ degrees of freedom, where $r$ denotes the rank of $\widehat{\Omega}^+$. This is how the $p$-level of the test can be computed. Roughly spoken, $W_2$ is a measure for the statistical distance between the empirical characteristic functions of both samples standardized by the variance–covariance matrices, with the characteristic functions being descriptors for the distributions underlying the two samples in question.

If the sample size of both observations is small, Epps and Singleton suggest to use a small-sample correction factor, $\widehat{C}(n_1, n_2)$. They conducted simulations and concluded that $W_2$ can be excessive for small $n_k$. Hence, if each one of the two samples includes less than 25 observations, a factor of

$$\widehat{C}(n_1, n_2) = \left\{1 + (n_1 + n_2)^{-0.45} + 10.1\left(n_1^{-1.7} + n_2^{-1.7}\right)\right\}^{-1}$$

should be applied on the test statistic $W_2$. The idea behind $\widehat{C}$ was to find a transformation $T(W_2; n_1, n_2) = C(n_1, n_2) \cdot W_2$ fulfilling $\sup P\{T(W_2; n_1, n_2) \geq \chi^2_\alpha\} \leq \alpha$, with $\chi^2_\alpha$ being the $1 - \alpha$ percentile of the $\chi^2$ distribution with four degrees of freedom. Epps and Singleton estimated the highest value of $C(n_1, n_2)$ in 1,000-trial simulations with different $\alpha$'s and sample sizes. The parameters of the correction factor $\widehat{C}$ were estimated to minimize the error $C(n_1, n_2) - \widehat{C}(n_1, n_2)$.

Epps and Singleton compared their test with the Anderson–Darling, the Cramér–von Mises, and the KS tests by means of computational simulations and came to the following conclusions:

- If discrete data are used, apply the ES test.
- If continuous data are used, the KS test usually has a lower power than the ES test.
- Sometimes, the Anderson–Darling and the Cramér–von Mises tests can have a higher power than the ES test.

# 3 The escftest command

## 3.1 Description

We include with this article a Stata implementation of the ES test in the program `escftest`. After installation, the new commands `escftest` and `help escftest` are available. In the algorithm described above, both matrix and vector operations are used. We used a Mata function in the code to accomplish these calculations. The reader should be aware that Mata was introduced to the Stata software package in version 9, so the command will refuse to work in versions earlier than 9.

## 3.2 Syntax

The syntax of the command to execute the ES characteristic function test is

`escftest` *varname* [ *if* ] [ *in* ], `group(`*groupvar*`)` [ `t1(`#`) t2(`#`)` ]

*varname* specifies the variable to test.

## 3.3 Options

`group(`*groupvar*`)` is required. It specifies the grouping variable. There must be exactly two different groups in the specified sample.

`t1(`#`)` specifies the parameter $t_1$ as defined by Epps and Singleton (1986). In this paper, details on this parameter are given in section 2. If omitted, the default is `t1(0.4)`. It should not be necessary to specify `t1()`.

`t2(`#`)` specifies the parameter $t_2$ as defined by Epps and Singleton (1986). In this paper, details on this parameter are given in section 2. If omitted, the default is `t2(0.8)`. It should not be necessary to specify `t2()`.

## 3.4 Saved results

Normally, it should not be necessary to modify `t1()` or `t2()`. These parameters should be modified only if one wants to calibrate the test for a specific task. `escftest` saves some of the results of the performed test in `r()`:

Scalars

| | |
|---|---|
| `r(crit_val_1)` | the critical value for the test statistic $W_2$ at a significance level of 0.01 |
| `r(crit_val_5)` | the critical value for the test statistic $W_2$ at a significance level of 0.05 |
| `r(crit_val_10)` | the critical value for the test statistic $W_2$ at a significance level of 0.1 |
| `r(p_val)` | the $p$-value associated with the actual test statistic $W_2$ |
| `r(correction)` | the small-sample correction factor, $C$ (if applied) |
| `r(t1)` | the value used for $t_1$ in the empirical characteristic function |
| `r(t2)` | the value used for $t_2$ in the empirical characteristic function |

Macros

| | |
|---|---|
| `r(group1)` | value of the grouping variable for the first group |
| `r(group2)` | value of the grouping variable for the second group |

# 4 Some applications

In this section, we compute two examples with the tests mentioned above. The first application refers to the numerical example from Epps and Singleton (1986); the data are taken from a study by Delse and Feather (1968). In this study, the ability of two groups to control salivation is compared; one group receives a biofeedback stimulus and the other group does not. The second example is taken from the field of experimental economics and applies an intercultural methodology introduced by Goerg and Walkowitz (2008) on Chinese and Germans.

First, we take a glance at the example described by Delse and Feather (1968). They investigate the effect of letting subjects hear a salivation signal and try to control their salivation. For the study, 20 subjects were equally distributed in two groups. Each subject was told to try to increase his salivation rate when observing a light signal on the left side and to decrease it when observing a light signal on the right side. In the experiment, one of the two groups received a biofeedback stimulus in terms of a tone (1,000 cycles per second, 0.2 seconds) for each saliva drop collected by a special apparatus. The other group did not receive such feedback. The data collected are shown in the table in figure 1. Each observation represents the difference between the mean number of saliva drops over 13 increase signals and the mean number of drops over 13 decrease signals. The data are taken from Hollander and Wolfe (1999, 180).[3] The quantile–quantile plot in figure 1 already reveals that the data of the two groups are not identically distributed.

| Biofeedback group | Control group |
|---|---|
| -1.15 | -0.35 |
| -0.15 | 2.55 |
| 2.48 | 1.73 |
| 3.25 | 0.73 |
| 3.71 | 0.35 |
| 4.29 | 2.69 |
| 5.00 | 0.46 |
| 7.74 | -0.94 |
| 8.38 | -0.37 |
| 8.60 | 12.07 |

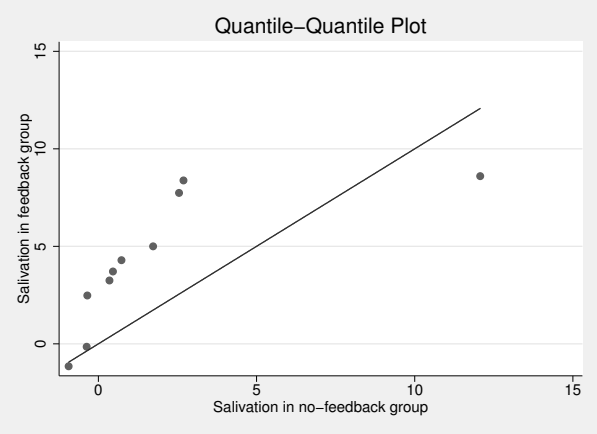



Figure 1. Values and quantile–quantile plot of the study by (Delse and Feather 1968)

Before taking a look at a comparison of the results of the ES test with the results of the KS test, we would like to mention that the numerical example from section 5 of Epps and Singleton (1986) contains an error that is either a simple typing error or a programming error: On page 202, the scale measure $\widehat{\sigma}$ for standardizing $t_j$ is stated to be 1.95. This is not correct. If one calculates $\widehat{\sigma}$ by hand, it becomes clear that this value should be 2.05. Christian Rojas,[4] who did some research on the ES test, arrived at the same conclusion. Nevertheless, the result of the numerical example is correct.

The variable `salivationDF` gives the participant's mean change rate of salivation from the Delse and Feather study. The variable `groupDF` defines the two subject groups in the study: group one with the biofeedback stimulus and group two without it. Both groups consist of 10 participants. Let's take a look at the test results:

3. Epps and Singleton take the data from an earlier edition of the same book.
4. See http://www.umass.edu/resec/faculty/rojas/index.shtml.

```
. escftest salivationDF, group(groupDF)

Epps-Singleton Two-Sample Empirical Characteristic Function test

Sample sizes: groupDF = 1           10
              groupDF = 2           10
              total                 20
t1                                0.400
t2                                0.800

Critical value for W2 at 10%      7.779
                         5%       9.488
                         1%      13.277
Test statistic W2                15.141

Ho: distributions are identical
P-value                         0.00442

Note: a small sample correction factor of C(10,10) = 0.60140 has been applied
to W2.
```

The ES test gives the necessary values of the test statistic $W_2$ for significance at 10%, 5%, and 1%. In this example, the test statistic $W_2 = 15.141$ totals to a value much higher than the necessary 13.277 for significance at the 1% level: the $p$-level is at 0.44%. A small-sample correction factor is applied because both observations are smaller than 25.

```
. ksmirnov salivationDF, by(groupDF) exact

Two-sample Kolmogorov-Smirnov test for equality of distribution functions

 Smaller group       D       P-value      Exact
 ------------------------------------------------
 1:                  0.1000    0.905
 2:                 -0.6000    0.027
 Combined K-S:       0.6000    0.055      0.035
```

Because of the small sample size, we apply `ksmirnov, exact`. The KS test gives the $p$-value for the one-sided comparison, once with a smaller group 1 and once with a smaller group 2. The combined value gives the exact $p$-value for the two-sided comparison. $H_0$ is rejected at a level of 5.5%. This is a much weaker significance level than the one for the ES test.

The second example is from the field of experimental economics.[5] A popular research question in this field is the comparison of economic behavior across different populations and decision conditions. Typical characteristics of data obtained by economic experiments are relatively small sample sizes and often the discreteness of attributes. The last point forbids the application of the KS test. Thus the question of whether behavior between subject groups differs and by what means is normally determined by the MW test. In contrast to the ES test, the MW test has a directional alternative hypothesis, $H_1$, which is that one sample is stochastically larger than the other. On one hand, if significant results are obtained by the MW test, they include more information than results from the ES test. On the other hand, if no sample is stochastically larger, the MW test finds no differences. The following example, where the KS test is not applicable,

5. In contrast to experiments in psychology, participants in experiments by economists receive a payoff that is determined by the decisions made in the experiment. This is done to ensure monetary incentives, which economists are interested in.

illustrates this limit of the MW test and the advantage of the ES test. The features of data gathered by economic experiments, described above, make the ES test a valuable tool for this research area where it is casually applied (for example, Henrich [2000], Eckel and Grossman [1998], and Hoffman, McCabe, and Smith [1996]).

In the experiment by Goerg and Walkowitz (2008), the cooperative behavior of participants from different countries is compared. Participants received an initial endowment of 10 Talers.[6] Two matched participants had to decide simultaneously and anonymously whether to send a part of their initial endowments to the matched player. The transfer amount had to be an integer between 0 and 10. This transferred amount reached the matched player doubled. The total payoff for the participant was his or her initial endowment minus the amount sent to the other player plus the doubled amount sent from the other player.

A participant who tries to maximize his own payoff would transfer nothing and hope that the matched player would send something to him. A player who wants to maximize the collective payoff would send everything and expect the matched player to transfer everything, too. Thus transferring nothing is understood as no cooperation, transferring something is understood as gradual cooperation, and transferring everything is understood as full cooperation. The method is introduced in more detail in Goerg and Walkowitz (2008), where it is applied on participants from Israel and Palestine.

The new and yet unpublished data that is discussed here contains the choices of 20 participants in China and 20 participants in Germany. The variable `cooperation` contains the transferred amount between 0 and 10, and the variable `country` defines the two groups.

6. A fictional currency used in the experiment, with a fixed exchange rate to Euros.

| Cooperation in Germany | Cooperation in China |
|---|---|
| 0 | 10 |
| 1 | 4 |
| 2 | 0 |
| 2 | 5 |
| 2 | 10 |
| 2 | 10 |
| 3 | 0 |
| 3 | 5 |
| 3 | 6 |
| 3 | 7 |
| 4 | 10 |
| 5 | 3 |
| 5 | 1 |
| 5 | 7 |
| 5 | 0 |
| 6 | 8 |
| 10 | 1 |
| 10 | 5 |
| 10 | 8 |
| 10 | 10 |

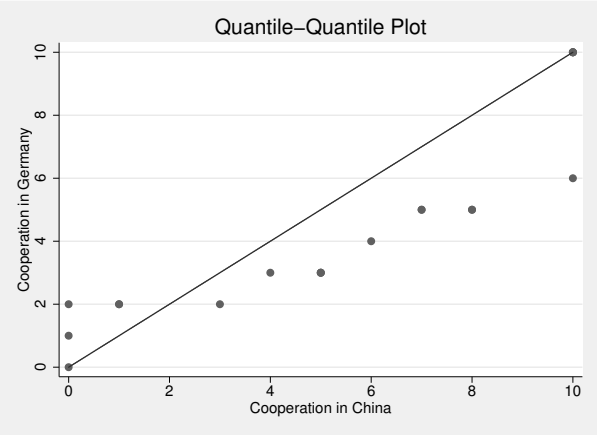



Figure 2. Cooperation in China and Germany

The quantile–quantile plot in figure 2 reveals differences between the two samples. Recall that the participants could choose only integer numbers as transfer amounts. The discreteness of the observed attribute rules out the application of the KS test to the data. We will search for quantitative support of this qualitative result by applying the MW test and the ES test. Let's start with the MW test:

```
. ranksum cooperation, by(country)

Two-sample Wilcoxon rank-sum (Mann-Whitney) test

     country |       obs    rank sum    expected
-------------+---------------------------------
           C |        20       441.5         410
           G |        20       378.5         410
-------------+---------------------------------
    combined |        40         820         820

unadjusted variance     1366.67
adjustment for ties      -28.72
                        ---------

adjusted variance       1337.95

Ho: cooper~n(country==C) = cooper~n(country==G)
            z =   0.861
    Prob > |z| =   0.3891
```

The two-sided rank-sum test reveals no significant difference ($p = 0.3891$) between the behavior in the two countries. The differences revealed by the quantile–quantile plot are of a kind that the MW test is not capable of showing. In contrast to this, the ES test detects more types of deviations than does the MW test. Thus the ES test leads to a different result:

```
. escftest cooperation, group(country)

Epps-Singleton Two-Sample Empirical Characteristic Function test

Sample sizes: country = C           20
              country = G           20
              total                 40
t1                                0.400
t2                                0.800

Critical value for W2 at 10%      7.779
                         5%       9.488
                         1%      13.277
Test statistic W2                 8.900

Ho: distributions are identical
P-value                        0.06364

Note: a small sample correction factor of C(20,20) = 0.76092 has been applied
to W2.
```

The ES test finds a significant difference between the distributions of behavior in the two countries, with a $p$-value of 0.0636. Obviously, the distribution of cooperative behavior in the two populations (participants in Germany and participants in China) differs. In both countries, the experimental conditions were kept identical regarding stakes, incentives, and distributions of demographic attributes among the participants. Thus the observed differences are most likely implied by the different cultural backgrounds.

The rank-sum test could not detect differences between participants from the two countries. This example impressively demonstrates the importance of the ES test for situations where discrete data are investigated, and these situations frequently occur in the field of experimental economics. While the MW test captures only central tendencies, the ES test can capture distributional characteristics.

# 5 Conclusions

In this article, we briefly described a powerful alternative to the Kolmogorov–Smirnov two-sample test and a complement to the Wilcoxon–Mann–Whitney rank-sum test, namely, the Epps–Singleton characteristic function test. We explained the use of the Stata implementation and applied the tests on two examples. The first example compared the $p$-levels of the KS test with those of the ES test and showed that the $p$-level of the ES test is far better. The second example showed a situation where the KS test cannot be applied and the MW test does not lead to significant results.

We provide the community with a Stata implementation of the ES test and hope that it might be of use. There is still room for future work; neither the Cramér–von Mises nor the Anderson–Darling two-sample test has been introduced to Stata so far (the Anderson–Darling goodness-of-fit test has already been adopted to Stata by Royston [1996]).

## 6 Acknowledgments

We thank the editor and the anonymous referee for speed improvements in the supplemented program code and for very valuable comments and suggestions. Furthermore, we thank Christian Rojas for discussions with us and for providing a Matlab implementation of the ES test.

## 7 References

Borak, S., W. Härdle, and R. Weron. 2005. Stable distributions. SFB 649 Discussion Paper 2005-008, Center for Applied Statistics and Economics, Humboldt-Universität zu Berlin. http://141.20.100.9/papers/pdf/SFB649DP2005-008.pdf.

Delse, F. C., and B. W. Feather. 1968. The effect of augmented sensory feedback on the control of salivation. *Psychophysiology* 5: 15–21.

Eckel, C. C., and P. J. Grossman. 1998. Are women less selfish than men?: Evidence from dictator experiments. *Economic Journal* 108: 726–735.

Epps, T. W. 1993. Characteristic functions and their empirical counterparts: Geometrical interpretations and applications to statistical inference. *American Statistician* 47: 33–38.

Epps, T. W., and K. J. Singleton. 1986. An omnibus test for the two-sample problem using the empirical characteristic function. *Journal of Statistical Computation and Simulation* 26: 177–203.

Goerg, S. J., and G. Walkowitz. 2008. Presentation effects in cross-cultural experiments: An experimental framework for comparisons. Discussion Paper No. 4/2008, Bonn Econ Discussion Papers, Bonn Graduate School of Economics, University of Bonn. http://econpapers.repec.org/paper/bonbonedp/bgse4_5f2008.htm.

Henrich, J. 2000. Does culture matter in economic behavior? Ultimatum game bargaining among the Machiguenga of the Peruvian Amazon. *American Economic Review* 90: 973–979.

Hoffman, E., K. A. McCabe, and V. L. Smith. 1996. On expectations and the monetary stakes in ultimatum games. *International Journal of Game Theory* 25: 289–301.

Hollander, M., and D. A. Wolfe. 1999. *Nonparametric Statistical Methods*. 2nd ed. New York: Wiley.

Royston, P. 1996. sg47: A plot and a test for the $\chi^2$ distribution. *Stata Technical Bulletin* 29: 26–27. Reprinted in *Stata Technical Bulletin Reprints*, vol. 5, pp. 142–144. College Station, TX: Stata Press.

**About the authors**

Sebastian Goerg holds a master's degree in economics. His main areas of research include experimental economics, algorithmic learning rules, and intercultural behavioral economics.

Johannes Kaiser holds a master's degree in business and engineering from the University of Karlsruhe and obtained a doctorate in economics from the University of Bonn. His research interests include behavioral and experimental economics, behavioral finance, and nonparametric statistics.

The Stata Journal (2009)
**9**, Number 3, pp. 466–477

# Multiple imputation of missing values: Further update of ice, with an emphasis on categorical variables

Patrick Royston
Hub for Trials Methodology Research
MRC Clinical Trials Unit and University College London
London, UK
pr@ctu.mrc.ac.uk

**Abstract.** Multiple imputation of missing data continues to be a topic of considerable interest and importance to applied researchers. In this article, the `ice` package for multiple imputation by chained equations (also known as fully conditional specification) is further updated. Special attention is paid to categorical variables. The relationship between `ice` and the new multiple-imputation system in Stata 11 is clarified.

**Keywords:** st0067_4, multiple imputation, chained equations, categorical variables, negative binomial distribution, ice, uvis, mi

## 1 Introduction

Royston (2004) introduced `mvis`, an implementation for Stata of multiple imputation by chained equations (MICE), a method of multiple multivariate imputation of missing values under missing-at-random (MAR) assumptions. The name of the main command was changed to `ice` (imputation by chained equations) in Royston (2005a). Two updates of `ice` have followed (Royston 2005b, 2007). This article presents a further update; the focus is on categorical variables. The main features are a considerable simplification of how imputation models for categorical variables are specified and revision of the `conditional()` option.

The `ice` system comprises three ado-files: `ice`, `ice_`, and `uvis`. Previous components `micombine`, `mijoin`, `misplit`, and `ice_reformat` are out of date and have been removed. `micombine` has been superseded by a more comprehensive command, `mim` (Carlin, Galati, and Royston 2008), itself recently updated with new features (Royston, Carlin, and White 2009).

Before describing the new features, it is important to clarify the impact of the Stata 11 multiple-imputation system on `ice`.

## 2 ice, uvis, and the Stata 11 multiple-imputation system

The `ice` program was written for Stata 9 and above to perform imputation via chained equations (van Buuren, Boshuizen, and Knook 1999). On 27 July 2009, Stata 11 was released, bearing a major new feature: the `mi` system for multiple imputation and esti-

mation of models with multiply imputed data. The system comprises a new database architecture for imputed datasets; utilities for manipulating, checking, and validating such datasets; a sequence of commands for doing imputation, `mi impute`; and a command for combining estimation results using Rubin's rules, `mi estimate`; see the *Stata 11 Multiple-Imputation Reference Manual* (StataCorp 2009) for details. Many (but not all) of the univariate imputation models available in `uvis` are replicated in new commands of the form `mi impute` *XXX*, where *XXX* is a keyword, such as `regress` for linear regression. Multivariate imputation in Stata 11 can be performed using `mi impute monotone` when the missingness pattern is monotone and using `mi impute mvn` when the missingness pattern is arbitrary. `mi impute monotone` implements a noniterative imputation method based on a sequence of independent univariate conditional specifications. It is similar to the implementation of option `monotone` of `ice`. `mi impute mvn` performs multivariate imputation assuming that the data have a multivariate normal distribution. It implements the NORM method of Schafer (1997)—an iterative Markov chain Monte Carlo method (data augmentation) based on multivariate normality. `ice` implements an alternative iterative multivariate imputation method based on a sequence of univariate full conditional specifications, also known as imputation via chained equations. Thus `ice` is not replicated in Stata 11 and is still needed for performing multiple imputation by chained equations for data with arbitrary patterns of missingness.

The `mi import ice` and `mi export ice` commands in Stata 11 make it easy to transport data between the existing `ice` data format and the official `mi` data format introduced in Stata 11. An intermediate step in integrating `ice` more completely into Stata 11 is a program called `mi ice`. The files can be installed from my web page (`net from http://www.homepages.ucl.ac.uk/~ucakjpr/stata/`) under the heading `mi_ice`. The next step is the development of a full-featured new command for Stata 11, likely to be called `mi impute ice`. Because in many people's eyes the flexibility of fully conditional specification embedded in the MICE algorithm offers several advantages over the multivariate normal approach, I expect `ice` and its sequels to continue to be used and useful in Stata 11.

# 3 Syntax

`ice` *mainvarlist* [ *if* ] [ *in* ] [ *weight* ] [ `,` `boot`[ `(`*varlist*`)` ] `by(`*varlist*`)` `cc(`*varlist*`)`
`clear` `cmd(`*cmdlist*`)` `conditional(`*condlist*`)` `cycles(`#`)` `dropmissing` `dryrun`
`eq(`*eqlist*`)` `eqdrop(`*eqdroplist*`)` `genmiss(`*string*`)` `id(`*newvar*`)` `interval(`*intlist*`)`
`m(`#`)` `match`[ `(`*varlist*`)` ] `monotone` `noconstant` `nopp` `noshoweq` `noverbose`
`nowarning` `on(`*varlist*`)` `orderasis` `passive(`*passivelist*`)` `persist`
`restrict(`[ *varname* ] [ *if* ]`)` `saving(`*filename* [ `, replace` ]`)` `seed(`#`)`
`substitute(`*sublist*`)` `trace(`*trace_filename*`)` ]

where a typical element of *mainvarlist* is [ `i.` | `m.` | `o.` ]*varname*.

`uvis` *regression_cmd* {*yvar* | *llvar ulvar*} *xvarlist* [ *if* ] [ *in* ] [ *weight* ],
`gen(`*newvar*`)` [`boot match by(`*varlist*`) noconstant nopp noverbose replace`
`restrict(`[ *varname* ] [ *if* ]`) seed(`#`)` ]

where *regression_cmd* may be `intreg`, `logistic`, `logit`, `mlogit`, `ologit`, `nbreg`, or `regress`. All weight types supported by *regression_cmd* are allowed. *llvar* and *ulvar* are required with `uvis intreg`. Variables imputed using `nbreg` must be nonnegative but are not restricted to integer values.

# 4 What is new?

The principal changes to `ice` (version 1.7.3) and `uvis` (version 1.5.5) compared with the December 2007 release (Royston 2007) (versions 1.4.4 and 1.2.7, respectively) are as follows:

1. `ice` and `uvis` now require Stata 10.1. (Strictly speaking, only the new negative binomial regression option requires Stata 10.1; all other features work under Stata 9.2 or higher.)
2. `ice` understands an abbreviated syntax for multilevel categorical variables to be imputed using `ologit` or `mlogit`. This important feature is described in detail below.
3. Negative binomial regression is available via the `cmd(`*varlist*`: nbreg)` option in `ice`, or via `uvis nbreg`, to impute count or count-like data. Typically, such data a) are integer-valued, b) are nonnegative, and c) have variance exceeding the mean. Noninteger variables are allowed, although their missing values are imputed as integers. Negative values are disallowed.
4. The syntax and operation of the `conditional()` option of `ice` have changed substantially (as described below).
5. A `by()` option has been added to `ice` and `uvis` to support imputation in independent subsets of the data (as described below).
6. A `restrict()` option has been added to `ice` and `uvis` to allow one to fit models on a specified subsample but impute missing data for the entire estimation sample (as described below).
7. A `clear` option has been added to `ice` to allow the imputed data to reside in memory without (yet) having been (manually) saved to a file using Stata's `save` command.
8. An `eqdrop()` option has been added to `ice` to delete variables from prediction equations.
9. A `persist` option has been added to `ice` to ignore errors from `uvis` when imputing a "difficult" variable.

# 5 Options for ice and uvis

Only new or changed options are described.

## 5.1 Options for ice

`clear` clears the original data from memory and loads the imputed dataset. Unless the `saving()` option is also specified, the data in memory are not permanently saved; this must then be done manually using the `save` or the `saveold` command.

`conditional(`*condlist*`)` invokes conditional imputation. Each item of *condlist* has the form *varlist*`:` `if` *condition*. Items are separated by a backslash (\). Members of *varlist* are imputed only for the subset of observations for which `if` *condition* is true (i.e., *condition* evaluates to a nonzero quantity). Observations on all members of *varlist* for which `if` *condition* is false (i.e., *condition* evaluates to zero) are left unchanged. *condition* must be a Stata expression constructed so that `if` *condition* is meaningful and valid for the current dataset. Note that variables appearing in *condition* may be members of *mainvarlist* or merely variables in the dataset. This is the only situation in which variables that do not appear in *mainvarlist* may be used in an `ice` command. Examples of its use are given in the help file.

`eqdrop(`*eqdroplist*`)` deletes variables from prediction equations. The syntax of *eqdroplist* is *varname1*`:` *varlist1* [ `,` *varname2*`:` *varlist2* ... ], where each *varname#* (or *varlist#*) is a member (or subset) of *mainvarlist*. One can only remove predictors from equations for variables with missing values (although trying to remove predictors from nonexistent equations is not a fatal error; an information message is issued). Variable names prefixed by `i.` are allowed, provided that the names were prefixed by `i.`, `m.`, or `o.` in *mainvarlist*. They are translated to the corresponding dummy variables created by `xi:`.

`noverbose` suppresses the display of the imputation number (as #) and cycle number within imputations (as .), which show the progress of the imputations.

`persist` causes `ice` to ignore errors raised by `uvis` when trying to impute a "difficult" variable or impute with a model that is difficult to fit to the data at hand. Trying to impute a difficult variable by using the `ologit` or `mlogit` command is the most common cause of failure. By default, `ice` stops with an error message. With `persist`, `ice` continues to the next variable to be imputed, not updating the variable that raised an error. Often, by chance, the same variable is successfully updated in a subsequent cycle, and no damage is done to the imputation process.

If the error for a given variable appears in every cycle, you should consider changing the prediction equation for that variable, because its imputed values are unlikely to be appropriate.

`restrict(`[ *varname* ] [ *if* ]`)` specifies that imputation models be computed using the subsample identified by *varname* and *if*. The subsample is defined by the observations for which *varname*`!=0` that also meet the *if* conditions. Typically, *varname*`=1`

defines the subsample and *varname*=0 indicates observations not belonging to the subsample. For observations whose subsample status is uncertain, *varname* should be set to a missing value; such observations are dropped from the subsample. By default, `ice` fits imputation models and imputes missing values using the sample of observations identified in the [ *if* ] and [ *in* ] expressions. The `restrict()` option identifies a subset of this sample to be used for model estimation. Imputation is restricted to the sample identified in the [ *if* ] and [ *in* ] expressions. Thus predictions and their associated imputations are made "out of sample" with respect to the subsample defined by `restrict()`. Examples of its use are given in the help file.

### 5.2 Options for uvis

`noverbose` suppresses nonerror messages while `uvis` is running.

`restrict(`[ *varname* ] [ *if* ]`)`; see the equivalent option for `ice`.

## 6 Simplified syntax for imputing multilevel categorical variables

### 6.1 Handling multilevel categorical variables

Experience and common sense suggest that correctly handling a multilevel categorical variable, say, $x$, in `ice` presents problems for users, for three reasons: 1) the user must decide on and specify an imputation model for predicting $x$, choosing between `mlogit` for unordered variables (the default) and `ologit` for ordered variables; 2) dummy variables corresponding to the levels of $x$ need to be "passively" imputed (i.e., reconstructed from the imputed values of $x$) following imputation of $x$; and 3) the dummy variables need to substitute for $x$ in equations for other variables in which $x$ is a predictor. `ice` comprehensively handles these three aspects through the `cmd()`, `passive()`, and `substitute()` options, respectively. Consider the following example, which uses data from a case–control study in leprosy.

Between 1980–1984, a population of about 112,000 people living in Northern Malawi were screened for leprosy (Fine et al. 1986). New cases of leprosy in initially uninfected people were identified during a follow-up period of five years (Pönnighaus et al. 1992). For illustrative purposes, we use data from a substudy in which controls without leprosy at baseline were selected at random from the screened population. The aim is to assess the effect of BCG vaccination (`bcg`) on the incidence of leprosy. Covariates are `age`, `sex`, `house`, and `school`.

```
. use lep
(1:4 unmatched leprosy and bcg)
. tabulate school, generate(s)
```

| Schooling | Freq. | Percent | Cum. |
|---|---|---|---|
| none | 282 | 22.19 | 22.19 |
| 1-5yr primary | 606 | 47.68 | 69.87 |
| 6-8yr primary | 350 | 27.54 | 97.40 |
| secondary/tertiary | 33 | 2.60 | 100.00 |
| Total | 1,271 | 100.00 | |

```
. tabulate house, generate(h)
```

| Housing | Freq. | Percent | Cum. |
|---|---|---|---|
| burnt brick | 240 | 19.25 | 19.25 |
| sun-dried bricks or pounded mud | 295 | 23.66 | 42.90 |
| wattle and daub | 679 | 54.45 | 97.35 |
| temporary shelter | 33 | 2.65 | 100.00 |
| Total | 1,247 | 100.00 | |

```
. ice d age sex bcg school s2 s3 s4 house h2 h3 h4, clear m(5)
> substitute(school:s2 s3 s4, house:h2 h3 h4) passive(s2:school==1 \
> s3:school==2 \ s4:school==3 \ h2:house==1 \ h3:house==2 \ h4:house==3)
> cmd(school:ologit, house:mlogit)
```

| #missing values | Freq. | Percent | Cum. |
|---|---|---|---|
| 0 | 1,186 | 86.57 | 86.57 |
| 4 | 146 | 10.66 | 97.23 |
| 8 | 38 | 2.77 | 100.00 |
| Total | 1,370 | 100.00 | |

| Variable | Command | Prediction equation |
|---|---|---|
| d | | [No missing data in estimation sample] |
| age | | [No missing data in estimation sample] |
| sex | | [No missing data in estimation sample] |
| bcg | | [No missing data in estimation sample] |
| school | ologit | d age sex bcg h2 h3 h4 |
| s2 | | [Passively imputed from school==1] |
| s3 | | [Passively imputed from school==2] |
| s4 | | [Passively imputed from school==3] |
| house | mlogit | d age sex bcg s2 s3 s4 |
| h2 | | [Passively imputed from house==1] |
| h3 | | [Passively imputed from house==2] |
| h4 | | [Passively imputed from house==3] |

```
Imputing .....1.....2.....3.....4.....5
[note: imputed dataset now loaded in memory]
Warning: imputed dataset has not (yet) been saved to a file
```

The variables `school` and `house` are categorical, each having four levels. `school` is the number of years of schooling and is ordinal, whereas `house` is the type of dwelling and is unordered. Hence `school` may be modeled by using `ologit`, and `house` may be modeled by using `mlogit`.

The specification of the `ice` command involves creating the dummy variables `s2`, `s3`, and `s4` for `school` and `h2`, `h3`, and `h4` for `house`, and using the `passive()` and `substitute()` options to manage the two sets of three dummy variables. The setup takes some effort and looks quite complicated, even in a simple example such as the present one. With more such variables (and `ice` specifications can have many), the complexity and potential for making errors increase considerably.

## 6.2 The m. and o. prefixes

Now consider the following, which is an identical setup but uses a new, more streamlined syntax:

```
. use lep, clear
(1:4 unmatched leprosy and bcg)
. ice d age sex bcg o.school m.house, clear m(5)
=> xi: ice d age sex bcg school i.school house i.house, cmd(house:mlogit,
> school:ologit) substitute(school:i.school, house:i.house) clear m(5)
i.school          _Ischool_0-3        (naturally coded; _Ischool_0 omitted)
i.house           _Ihouse_0-3         (naturally coded; _Ihouse_0 omitted)
```

| #missing values | Freq. | Percent | Cum. |
|---|---|---|---|
| 0 | 1,186 | 86.57 | 86.57 |
| 1 | 146 | 10.66 | 97.23 |
| 2 | 38 | 2.77 | 100.00 |
| Total | 1,370 | 100.00 | |

| Variable | Command | Prediction equation |
|---|---|---|
| d | | [No missing data in estimation sample] |
| age | | [No missing data in estimation sample] |
| sex | | [No missing data in estimation sample] |
| bcg | | [No missing data in estimation sample] |
| school | ologit | d age sex bcg _Ihouse_1 _Ihouse_2 _Ihouse_3 |
| _Ischool_1 | | [Passively imputed from (school==1)] |
| _Ischool_2 | | [Passively imputed from (school==2)] |
| _Ischool_3 | | [Passively imputed from (school==3)] |
| house | mlogit | d age sex bcg _Ischool_1 _Ischool_2 _Ischool_3 |
| _Ihouse_1 | | [Passively imputed from (house==1)] |
| _Ihouse_2 | | [Passively imputed from (house==2)] |
| _Ihouse_3 | | [Passively imputed from (house==3)] |

```
Imputing .........1.........2.........3.........4.........5
[note: imputed dataset now loaded in memory]
Warning: imputed dataset has not (yet) been saved to a file
```

`ice` accepts `o.school` and `m.house` as a sufficient specification of how to model these two variables. The program does the necessary work of setting up the required `passive()` and `substitute()` options, creating dummy variables and defining sensible prediction equations. `ice` invokes Stata's `xi:` command to produce the variables labeled in standard fashion, `_Ischool_1`, etc. The "expanded" `ice` command is displayed before the tables of missing values and prediction equations are presented.

Now suppose that we wanted the prediction equation for `school` to be `d age _Ihouse_1 _Ihouse_2 _Ihouse_3` rather than the default equation, which also includes `sex bcg`. To specify this, we use the `i.` prefix within the `eq()` option for `school` to signify the `_I*` variables for `house`:

```
. ice d age sex bcg o.school m.house, eq(school: d age i.house) dryrun
=> xi: ice d age sex bcg school i.school house i.house, cmd(house:mlogit,
> school:ologit) substitute(school:i.school, house:i.house) eq(school: d age
> i.house) dryrun
```

(*output omitted*)

| Variable | Command | Prediction equation |
|---|---|---|
| d | | [No missing data in estimation sample] |
| age | | [No missing data in estimation sample] |
| sex | | [No missing data in estimation sample] |
| bcg | | [No missing data in estimation sample] |
| school | ologit | d age _Ihouse_1 _Ihouse_2 _Ihouse_3 |
| _Ischool_1 | | [Passively imputed from (school==1)] |
| _Ischool_2 | | [Passively imputed from (school==2)] |
| _Ischool_3 | | [Passively imputed from (school==3)] |
| house | mlogit | d age sex bcg _Ischool_1 _Ischool_2 _Ischool_3 |
| _Ihouse_1 | | [Passively imputed from (house==1)] |
| _Ihouse_2 | | [Passively imputed from (house==2)] |
| _Ihouse_3 | | [Passively imputed from (house==3)] |

```
End of dry run. No imputations were done, no files were created.
```

Finally, an additional advantage of the `m.` and `o.` syntax is that the number of missing values is counted correctly. In the original syntax, missing values are multiply counted due to inclusion of the dummy variables with the "parent" variables in *mainvarlist*.

## 6.3 The i. prefix

The other new feature is the `i.` prefix, which simply applies `xi:` to the variable in question. (Note: Do not confuse the `i.` prefix in `ice` with the same syntax in Stata 11, which indicates a factor variable. At this point, `ice` does not support factor variables; it may do so in the future.) It is important to emphasize that for the `i.` prefix to work correctly with a multilevel categorical variable, the latter must have *no missing data in the estimation sample*, that is, it must be complete except when all other variables in *mainvarlist* have missing values. This condition is checked by `ice` and an error message is issued if it is violated. The `i.` prefix automatically handles the resulting dummy variables in the correct manner, that is, they become predictors for other variables but are not themselves imputed (because they are assumed to have no missing data). If a categorical variable does have missing data, either the `m.` or the `o.` prefix must be used to invoke the machinery needed to create a valid set of imputation models.

To clarify further, consider the following example with the leprosy data:

```
. use lep, clear
(1:4 unmatched leprosy and bcg)
. ice age sex bcg i.school, dryrun
=> xi: ice age sex bcg i.school, dryrun
i.school          _Ischool_0-3        (naturally coded; _Ischool_0 omitted)
```

| #missing values | Freq. | Percent | Cum. |
|---|---|---|---|
| 0 | 1,271 | 92.77 | 92.77 |
| 1 | 99 | 7.23 | 100.00 |
| Total | 1,370 | 100.00 | |

```
99 missing values of variable school found in the estimation sample
variables with an i. prefix must be complete in the estimation sample
you can use an m. or o. prefix to impute incomplete variables of this type
1 specification error(s) found
r(198);
```

Using `i.school` invokes `xi:` as expected. However, because we have not specified what is to happen to the dummy variables `_Ischool_1`, `_Ischool_2`, and `_Ischool_3`, and because they each have 99 missing values, we find that each of them is independently predicted from the other two dummy variables and `age`, `sex`, and `bcg`. This is clearly incorrect. The correct solution is either `ice age sex bcg m.school` or `ice age sex bcg o.school`, depending on how `school` is to be modeled.

# 7 Conditional imputation—the conditional() option

Consider an (artificial) dataset including the variables `age`, `female`, and `pregnant`, where `age` is continuous, approximately normally distributed and has missing values; `female` is binary (1 for females, 0 for males) and is complete; and `pregnant` is binary (1 for pregnant, 0 for not pregnant) and also has missing values. Such a dataset is supplied in `pregnant.dta`. Suppose that the probability of being pregnant is related to age. Because males cannot be pregnant, we do not wish to impute pregnancy in males; we should therefore impute missing values of `pregnant` using `age` in females only:

```
. use pregnant, clear
(Artificial dataset on pregnancy and age)
. ice age pregnant female, conditional(pregnant: if female==1) dryrun
```

| #missing values | Freq. | Percent | Cum. |
|---|---|---|---|
| 0 | 388 | 77.60 | 77.60 |
| 1 | 110 | 22.00 | 99.60 |
| 2 | 2 | 0.40 | 100.00 |
| Total | 500 | 100.00 | |

| Variable | Command | Prediction equation |
|---|---|---|
| age | regress | pregnant female |
| pregnant | logit | age if female==1 |
| female | | [No missing data in estimation sample] |

```
End of dry run. No imputations were done, no files were created.
```

The prediction equation for `age` is `pregnant female`, whereas the equation for `pregnant` is just `age` of the females. It is important here that the values of `pregnant` are correctly defined as 0 for males, even though imputation of `pregnant` is not performed for `female==0`. If `pregnant` were set to, say, −1 for males, then `ice` would see three distinct values of `pregnant`. It would try to impute `pregnant` using `mlogit` rather than `logit`, which would cause an error.

More than one conditional imputation can occur in the same run. Suppose, for example, that the dataset included a variable with missing values called `fertile`, giving the result of a female fertility test and coded 1 = fertile, 0 = infertile. In `ice`, one might specify this case as

```
ice age pregnant female fertile, ///
 conditional(pregnant: if female==1 & fertile==1 \ fertile: if female==1) dryrun
```

to reflect that only fertile females can become pregnant and only females have a fertility test. If males had also had a fertility test, the phrase `fertile: if female==1` would have been omitted.

# 8 Out-of-sample imputation—the restrict() option

The `restrict()` option is designed for situations in which one wishes to fit imputation equations to a subset of the observations but make imputations across the entire dataset. An example is a "validation study" in which a primary dataset is used to determine and estimate a multivariable model of some kind and a secondary dataset is available to test the accuracy of the model. Both the primary and the secondary datasets may have missing values. The combined (primary + secondary) dataset would typically include a variable, say, `primary`, coded as 1 for the primary and 0 for the secondary dataset. A schematic `ice` run to impute missing values out of sample in the secondary dataset is

```
ice mainvarlist, restrict(primary) <other_stuff> ...
```

The imputation models would be estimated on the subset `primary != 0`, and multiple imputation of the subset consisting of all nonmissing values of `primary` would be done. See the `ice` help file for further comments on this option.

## 9 Imputation in independent subsets—the by() option

Using `by(`*varlist*`)` performs multiple imputation separately for all combinations of the variables in *varlist*. Observations with missing values for any members of *varlist* are excluded.

An application of `by()` is in randomized trials, where interactions as yet unidentified between treatment and patient characteristics (covariates) may be present. If imputation is not done separately for each treatment group, estimates of interactions with treatment in the analysis model are biased toward zero. The `by()` approach may be useful more generally for coping with interactions with a categorical variable.

Common sense must be applied to the `by()` option. For example, if `by(`*varlist*`)` subdivides the dataset too finely, the imputation models may become unstable and imprecise, compromising the quality of the imputations.

## 10 Conclusion

Development of `ice` continues as new features are requested by users or considered by the author to be worthwhile. Closer integration with Stata 11 `mi` will follow in due course. The syntax for categorical variables should prove particularly helpful for users and for teachers of multiple imputation in Stata.

## 11 Acknowledgments

Ian White suggested and partly coded the prefix syntax for categorical variables and the negative binomial regression command for `ice` and `uvis`. I am grateful to Dr Paul Fine for permission to use the leprosy data and to Yulia Marchenko for remarks on the Stata 11 `mi` system.

## 12 References

Carlin, J. B., J. C. Galati, and P. Royston. 2008. A new framework for managing and analyzing multiply imputed data in Stata. *Stata Journal* 8: 49–67.

Fine, P. E. M., J. M. Pönnighaus, N. Maine, J. A. Clarkson, and L. Bliss. 1986. Protective efficacy of BCG against leprosy in northern Malawi. *Lancet* ii: 499–502.

Pönnighaus, J. M., P. E. M. Fine, J. A. C. Sterne, R. J. Wilson, E. Msosa, P. J. K. Gruer, P. A. Jenkins, S. B. Lucas, G. Liomba, and L. Bliss. 1992. Efficacy of BCG vaccine against leprosy and tuberculosis in Northern Malawi. *Lancet* 339: 636–639.

Royston, P. 2004. Multiple imputation of missing values. *Stata Journal* 4: 227–241.

———. 2005a. Multiple imputation of missing values: Update. *Stata Journal* 5: 188–201.

———. 2005b. Multiple imputation of missing values: Update of ice. *Stata Journal* 5: 527–536.

———. 2007. Multiple imputation of missing values: Further update of ice, with an emphasis on interval censoring. *Stata Journal* 7: 445–464.

Royston, P., J. B. Carlin, and I. R. White. 2009. Multiple imputation of missing values: New features for mim. *Stata Journal* 9: 252–264.

Schafer, J. L. 1997. *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: Chapman & Hall/CRC.

StataCorp. 2009. *Stata 11 Multiple-Imputation Reference Manual*. College Station, TX: Stata Press.

van Buuren, S., H. C. Boshuizen, and D. L. Knook. 1999. Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine* 18: 681–694.

**About the author**

Patrick Royston is a medical statistician with 30 years of experience who has a strong interest in biostatistical methods and in statistical computing and algorithms. He now works in cancer clinical trials and related research issues. Currently, he is focusing on problems of model building and validation with survival data, including prognostic factor studies; on parametric modeling of survival data; on multiple imputation of missing values; and on novel clinical trial designs.

The Stata Journal (2009)
**9**, Number 3, pp. 478–496

# Speaking Stata: Creating and varying box plots

Nicholas J. Cox
Department of Geography
Durham University
Durham City, UK
n.j.cox@durham.ac.uk

**Abstract.** Box plots have been a standard statistical graph since John W. Tukey and his colleagues and students publicized them energetically in the 1970s. In Stata, `graph box` and `graph hbox` are commands available to draw box plots, but sometimes neither is sufficiently flexible for drawing some variations on standard box plot designs. This column explains how to use `egen` to calculate the statistical ingredients needed for box plots and `twoway` to re-create the plots themselves. That then allows variations such as adding means, connecting medians, or showing all data points beyond certain quantiles.

**Keywords:** gr0039, box plots, dispersion diagrams, distributions, egen, graphics, percentile, quantile, range bars, twoway

## 1 Box plots

### 1.1 Origins

Box plots were so named by John W. Tukey and were publicized energetically within statistics by him, his colleagues, and his students from the 1970s on (e.g., Tukey [1972, 1977]; Velleman and Hoaglin [1981]; and Hoaglin, Mosteller, and Tukey [1983]). Box plots spread beyond statistics into several quantitative sciences through their own literature (e.g., Kleiner and Graedel [1980] and Cox and Jones [1981]). The publicity was so successful that the box plot is now widely regarded as a standard statistical graph. It appears in most introductory statistical texts; indeed, the exceptions to this rule (e.g., Freedman, Pisani, and Purves [2007]) are more striking than the examples. Further, the box plot is often assumed not to need explanation beyond such texts.

Box plots had several under-appreciated precursors under different names, including range bars (Spear 1952, 1969) and dispersion diagrams in geography and climatology (e.g., Crowe [1933] and Monkhouse and Wilkinson [1971]). Despite this earlier history, my guess is that box plots would not now be nearly so popular without Tukey's reinvention and propaganda.

### 1.2 Purpose

Stata users wishing to see box plots can call upon `graph box` or `graph hbox`. The manual entry [G] **graph box** explains several ways of tuning that command. Mitchell (2008) gives many examples of possible results and the code to get them. This column

focuses on showing what to do whenever you want some variation on the standard design that cannot be met with `graph box` or `graph hbox`. To show that, we must understand how to re-create box plots using `graph twoway`. It is very much a case of *reculer pour mieux sauter*.

## 1.3 Structure

Let us first remind ourselves of the structure of a box plot by using the life expectancy data shipped with Stata. We will compare life expectancy in 1998 for three groups of countries: in Europe and Central Asia, North America, and South America (figure 1). We use `graph box`. Here and subsequently we will spell out a preference for horizontal axis labels.

```
. sysuse lifeexp
. label var lexp "Life expectancy (years)"
. graph box lexp, over(region) yla(, ang(h))
```

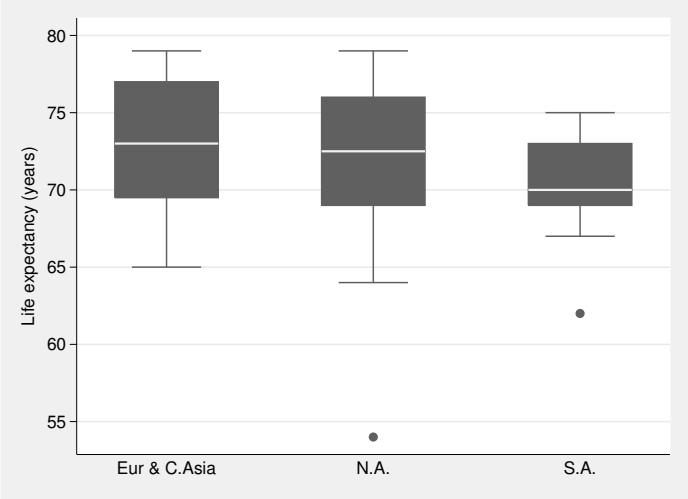



Figure 1. Box plots of life expectancy in 1998 for various countries in three regions

The main ingredient of a box plot is the eponymous box, used to indicate the lower and upper quartiles of the variable or group being plotted against a magnitude scale. The median is represented by a line subdividing the box, or, alternatively, by a point symbol. The length of the box thus represents the interquartile range (IQR). Tukey used a variety of alternative terms for both the quartiles (hinges, fourths, etc.) and their difference, the range or spread between them, but most such terms were adopted only locally or briefly and have long since faded away. It seems simpler now to revert to the classical terms of quartiles and IQR. Whatever the terminology, recall that numerous slightly different calculation rules exist for quartiles and quantiles or percentiles

generally (Frigge, Hoaglin, and Iglewicz 1989; Hyndman and Fan 1996). The different rules explain some of the differences in box plots from different software, but otherwise are not of great interest. Stata's rule is set out in [R] **summarize**. Among other details, note that any practical rule must extend to data with weights assigned.

Box plots differ in what else may be shown outside the box. `graph box` and `graph hbox` by default follow what is perhaps the most common recipe (Tukey 1977):

1. Lines, often called whiskers, are drawn to span all data points within 1.5 IQR of the nearer quartile. That is, one whisker extends to include all data points within 1.5 IQR of the upper quartile and stops at the largest such value, while the other whisker extends to include all data within 1.5 IQR of the lower quartile and stops at the smallest such value. Tukey called the outer limits of the whiskers *adjacent values*. The whiskers also explain his alternative term, *box-and-whiskers plots*. Note that either whisker could be of zero length. In practice, that will occur only with very small datasets or heavily tied data.

2. Any data points beyond the whiskers are shown individually and often labeled informatively.

De Veaux, Velleman, and Bock (2008, 81) record Tukey's laconic reply when asked the reason for 1.5: 1 would be too small and 2 would be too large. Evidently, the choice of multiplier gives an informal but objective rule for outlier identification. Any choice is a compromise between revealing too much (flagging data points that are of neither statistical nor scientific concern) and revealing too little (missing data points that require thought or action). Dümbgen and Riedwyl (2007) recently discovered a clever way of justifying 1.5, but experience that it often works quite well is a more compelling basis for the rule.

This kind of box plot, and indeed most other kinds, thus conveys information about level (median); spread (interquartile range and range are both represented directly); symmetry or asymmetry about the median both within and beyond the central half of the data; and, on its own definition, possible outliers. It is thus a fairly information-rich graphical reduction of key quantiles (or of order statistics, if you prefer).

That may be the most common recipe, but many others have been entertained. McGill, Tukey, and Larsen (1978) suggested two refinements: varying the width of boxes to indicate group sizes and notching boxes to indicate approximate confidence intervals. Harris (1999, 57) even reported that some box plots are based at least in part on mean and standard deviation. It is natural to hope that different conventions are all explained clearly for the benefit of readers, but unfortunately, that is often not the case. For example, several authors in the collection edited by Chen, Härdle, and Unwin (2008) use differing varieties of box plots, but the differences are typically unexplained.

However, many variations encountered appear to be essentially cosmetic. In particular, box plots may be horizontal as well as vertical. There can be a small struggle between the convention of showing response or outcome variables increasing vertically

and the desire that text labels explaining variables or groups can be spelled out fully and legibly. Whatever the reasoning, Stata users can reach for `graph hbox` if they prefer horizontal alignment. As a matter of careful and conscious design, the change between typing `box` and `hbox` is the only change that need be made. Contrary to mathematical custom, the $y$ axis of box plots in Stata is considered to be whichever axis the response is plotted against. (`graph bar` and `graph hbar` are related in exactly the same way.)

## 1.4 Utility

Box plots can be very useful, particularly for comparison, especially if the number of variables or groups is nearer 20 or 200 rather than 2. But if you have just a few variables or groups, you have enough space for the greater detail of (say) histograms, dot plots, density traces, or quantile or distribution plots. And because they are reductions of the data, box plots may be uninformative about key details. They tend to perform poorly whenever data are highly skewed—which in many fields is overwhelmingly usual. Naturally, one simple answer to skewness is to transform data. If box plots of a variable are highly asymmetric, then roots or logs or reciprocals are likely to improve matters considerably.

There are deeper problems yet. What is so special about quartiles, in particular? Medians have a clear statistical role as defining midpoints on distribution functions, and they are natural and resistant summaries for (approximately) symmetric distributions. Quartiles take the median idea one step further by being medians of each half of the distribution, but beyond that, their role is much less evident. Simplicity of definition and familiarity from early teaching do not add up to a statistically natural role. In any case, if half the data lie inside the boxes, then half too lie outside the boxes, yet that half—often statistically or scientifically the more important half—is represented in a mostly generalized way within box plots.

So, other quantiles besides quartiles may well be as or more worthy of display. That argument leads ultimately to displaying all quantiles, a tactic discussed in other issues of the *Stata Journal* (Cox 2005, 2007).

With a nod of gratitude for an example given by Wainer (1990, 345), figure 2 points out one further weakness of box plots.

(*Continued on next page*)

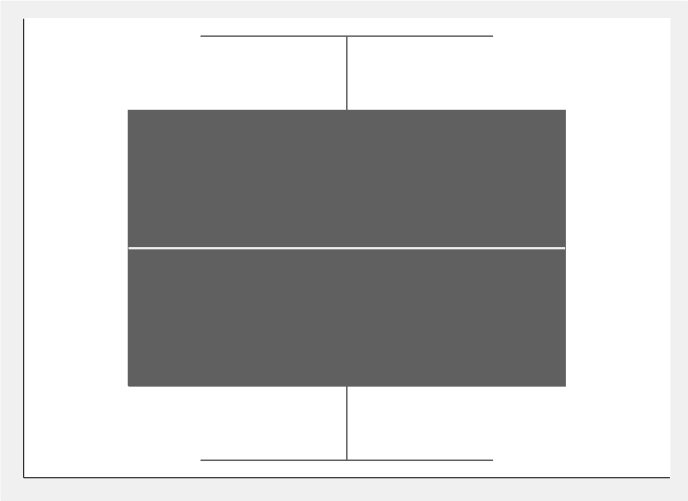


Figure 2. An innocent-looking box plot with a surprise wrapped inside

Asked what can be inferred about the distribution from this plot, even very experienced data analysts typically mutter something about a short-tailed symmetric unimodal distribution. But the box plot clearly implies that the average density in the tails is much greater than that in the middle, so the best inference should be something like a U-shaped distribution. My guess is that although respondents are all familiar with the main idea of box plots, they are being misled by the subdued representation of the tails. Guessing apart, no detailed histogram, density trace, or quantile plot would be guilty of such ambiguity. More generally, box plots inevitably gloss over bimodality or multimodality or granularity of distribution.

To reveal the small surprise, figure 2 is based on a set of quantiles from a beta distribution:

```
. generate y = invibeta(0.6, 0.6, (_n - 0.5) / _N)
```

With these parameter values, the distribution is indeed U-shaped, as the histogram in figure 3 shows more clearly.

```
. histogram y, width(0.1) start(0) horizontal yla(, ang(h))
```

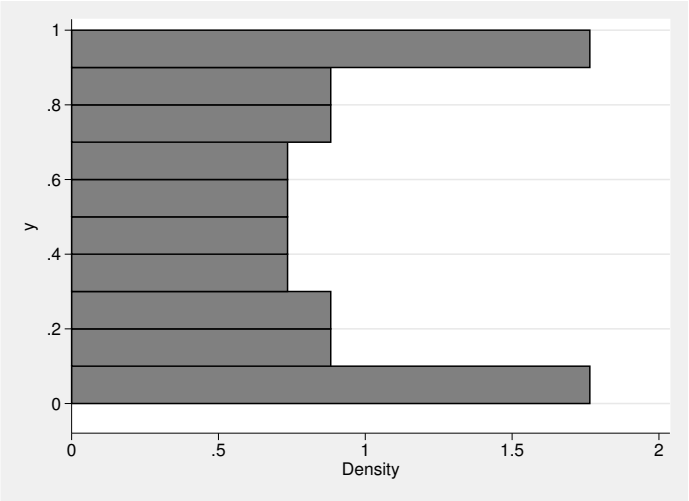



Figure 3. The distribution underlying the innocent-looking box plot: a U-shaped beta distribution

# 2 Using twoway to create box plots

## 2.1 Ingredients

To re-create a box plot from scratch given some data, we need to calculate the basic summary statistics. Here the `egen` command is your friend, particularly because its `by()` option allows recording of results for two or more groups. The `by()` option is undocumented in favor of doing things with `by` *varlist*`:`, but it is supported for those `egen` functions of concern to us here. Either way, using `by:` as prefix is exactly equivalent. See the online help or [D] **egen** for more details on that command. A tutorial discussing `egen` is available in Cox (2002).

The median and quartiles are easiest:

```
. egen median = median(lexp), by(region)
. egen upq = pctile(lexp), p(75) by(region)
. egen loq = pctile(lexp), p(25) by(region)
```

We could now get the IQR by subtraction, `upq - loq`, which would be more efficient, but we will mention that it has its own `egen` function.

```
. egen iqr = iqr(lexp), by(region)
```

In fact, we do not strictly need the IQR, as will become clear shortly, but if you like box plots, you might as well know ways of getting the IQR easily into a variable.

The upper and lower limits of the whiskers require a little more thought. Here is one way to get them. The upper limit is the largest value not greater than `upq + 1.5 * iqr`. That can be calculated in one line:

```
. egen upper = max(min(lexp, upq + 1.5 * iqr)), by(region)
```

That one line could bear some deconstruction, however. The outer `max()` is an `egen` function, as the context implies. The inner `min()` is emphatically not another `egen` function, as might be guessed: it is just the standard Stata function `min()`. Why is that allowed here? Because the syntax of `egen` allows here an arbitrary expression, indicated in the syntax diagram by *exp*. Often that expression is just one variable name, but it could be more complicated. Here the entire expression is `min(lexp, upq + 1.5 * iqr)`. The expression could have been `min(lexp, upq + 1.5 * (upq - loq))`, showing that the IQR variable is indeed redundant.

As before, the `by(region)` option ensures that maximums for the expression supplied are calculated separately for each region.

For lower limits of whiskers, we can use the same tactic, except for swapping minimum and maximum:

```
. egen lower = min(max(lexp, loq - 1.5 * iqr)), by(region)
```

We now have in hand all the ingredients we need. But one basic point needs emphasis. By construction, the values for the median, quartiles, and upper and lower limits of the whiskers are repeated for each distinct value of `region`. If instead of comparing groups we were comparing variables, then values would be repeated for each observation. Unless we do nothing further, the graphical consequence will be repeated plotting of the same information, which could be time-consuming and which leads to unnecessarily bloated graph files. It would be important to do something about that in any program with pretensions to efficiency, but for our purposes, we will set this detail aside, beyond noting that `collapse` and `egen, tag()` offer some solutions.

## 2.2 Assembly

Let us jump immediately to a tolerable mock-up of a box plot and then talk through all the details. Let me also stress that even Stata graph experts never write down code just like this, unless they happen to have solved the problem a few minutes earlier and have excellent memory. To get here requires much experiment and consultation of the help. Figure 4 shows the result.

```
. twoway rbar med upq region, pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rbar med loq region, pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rspike upq upper region, pstyle(p1) ||
> rspike loq lower region, pstyle(p1) ||
> rcap upper upper region, pstyle(p1) msize(*2) ||
> rcap lower lower region, pstyle(p1) msize(*2) ||
> scatter lexp region if !inrange(lexp, lower, upper), ms(Oh) mla(country)
> legend(off)
> xla(1 "" "Europe and" "Central Asia" "" 2 "North America" 3 "South America",
> noticks) yla(, ang(h)) ytitle(Life expectancy (years)) xtitle("")
```

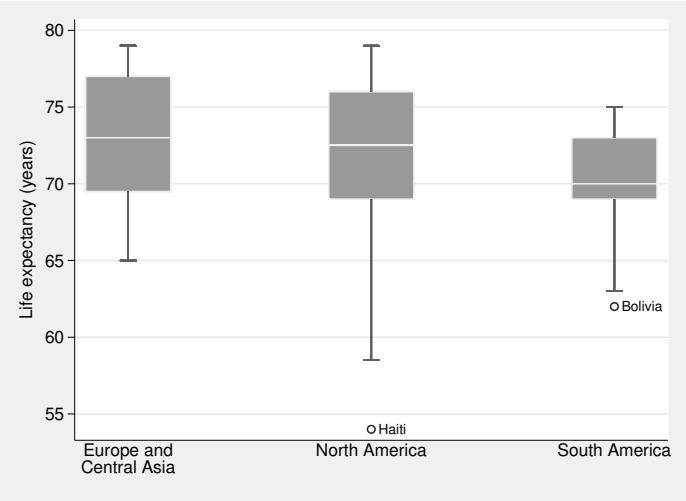



Figure 4. Box plots of life expectancy in 1998 for various countries in three regions, but constructed entirely using `twoway`

Now the commentary:

1. The details may look scary in total, but note first the strategy, which is divide and conquer. Different parts of `twoway` are enlisted to draw different parts of the graph. Similarly, divide and conquer is the strategy to understand the code. There is clearly no need to try to reproduce all the details produced by `graph box` if you prefer something different.

2. `region` is a numeric variable, so we can plot against it. Its values are 1, 2, and 3, and value labels are attached, so it is already in good condition for graphics. If you had a variable that was not in good condition, say, because it was a string variable or a numeric variable that needed tidying up, then creating a new variable with `egen, group()` with its `label` option is the best way to proceed. `encode` is an alternative for string variables.

3. `pstyle(p1)` is a simple trick to enforce general consistency of style. You can then depart from whatever results in your preferred directions.

4. The boxes are drawn with `twoway rbar`, one from the median to the upper quartile and one from the median to the lower quartile. A light outline color, `blcolor(gs15)`, is sufficient to indicate where the medians are. I chose as a matter of personal taste a lighter color for the bar fill than the default in the `sj` scheme. Light color for fill and dark color for outline are equally acceptable statistically, and perhaps preferable aesthetically. `barwidth(0.35)` reflects my personal taste: I regard the default boxes of `graph box` as a little fat. If the values of the categorical variable did not differ by 1, a quite different bar width would be needed.

5. The whiskers are drawn with `twoway rspike`, one from the lower quartile to the lower end of the whiskers, and one from the upper quartile to their upper end.

6. The whiskers are capped using `twoway rcap`. Note that there is no typo in `rcap upper upper region` or `rcap lower lower region`. The code was not `rcap upq upper region` or `rcap loq lower region` to ensure that no caps are visible interfering with the box. The marker size is twice default, but even so is much less than the default of `graph box`, to say nothing of what can be obtained using its `capsize()` option.

7. Clearly, the caps could be omitted if so desired, simply by omitting the calls to `twoway rcap`. Why does the standard box plot design include them? It seems to be an admission of weakness, namely, that the whiskers might be overlooked if the graph did not emphasize where they end.

8. Data points beyond the whiskers are shown using `scatter`. Hollow circles given by `ms(Oh)` are a personal choice as suitably prominent yet tolerating overlap well (think of the overlapping rings of the Olympic symbol). Note the simple logic: points within the range of the boxes and whiskers are `inrange(lexp, lower, upper)` and so points beyond them are the logical complement, obtained by negation, `!`. See Cox (2006) for more on `inrange()` if so desired. Putting this into words as "not in range" is a simple way of underlining what is being done.

9. Such data points are labeled using marker labels, `mla(country)`. In this case, defaults work fine. In other cases, we might want to tune marker label size or other properties, as later examples will make clear.

10. All the different `twoway` calls produce a complicated `legend`, which we just suppress. So many different variables are being portrayed, from `twoway`'s point of view, that we have to add our own $y$-axis title.

11. In this particular case, the value labels attached to `region` are over-abbreviated, so we step in and provide our own. I agree with the designer of `graph box` that axis ticks serve no useful purpose when distinct categories are being shown. The default `xtitle()` would be the variable name `region`, which also is dispensable here. (In other contexts, I routinely suppress variable names indicating date or year when axis labels such as 1990 or 2000 make abundantly clear what is being shown.)

## 2.3 Horizontal

Clearly, we need to know how to produce horizontal box plots too. Here is a first stab, with the result in figure 5:

```
. twoway rbar med upq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rbar med loq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rspike upq upper region, horiz pstyle(p1) ||
> rspike loq lower region, horiz pstyle(p1) ||
> rcap upper upper region, horiz pstyle(p1) msize(*2) ||
> rcap lower lower region, horiz pstyle(p1) msize(*2) ||
> scatter region lexp if !inrange(lexp, lower, upper), mla(country) legend(off)
> yla(1 "" "Europe and" "Central Asia" "" 2 "North America" 3 "South America",
> ang(h) noticks) xtitle(Life expectancy (years)) ytitle("")
```

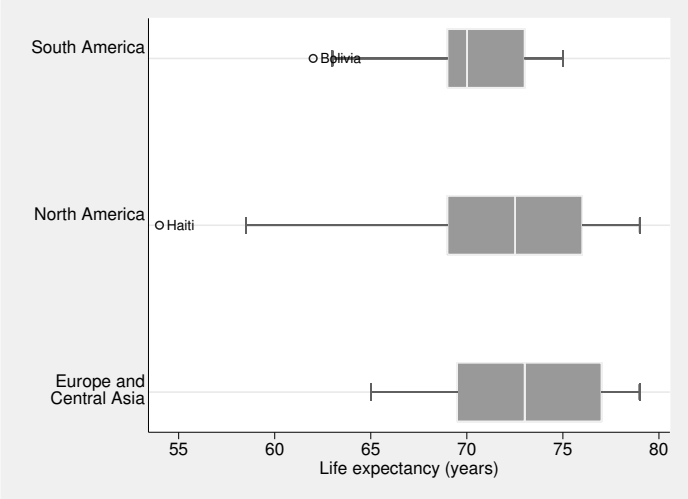



Figure 5. Horizontal box plots of life expectancy in 1998 for various countries in three regions, but constructed entirely using `twoway`

The necessary changes are to add the `horizontal` option to calls to `rbar`, `rspike`, and `rcap` and to swap $y$ and $x$ within the call to `scatter` (variables are swapped, `x` options become `y` options, and vice versa).

The result is most of the way to where we want to be. The marker labels would be better lifted clear of the whiskers. The $x$ axis also needs to be lengthened a little to give enough space for the text `Haiti`. A little experiment shows that the extra options `xsc(r(53,.))`, `mlabpos(12)`, and `mlabgap(1.5)` give those improvements; see figure 6.

(*Continued on next page*)

```
. twoway rbar med upq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rbar med loq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rspike upq upper region, horiz pstyle(p1) ||
> rspike loq lower region, horiz pstyle(p1) ||
> rcap upper upper region, horiz pstyle(p1) msize(*2) ||
> rcap lower lower region, horiz pstyle(p1) msize(*2) ||
> scatter region lexp if !inrange(lexp, lower, upper), mla(country)
> mlabpos(12) mlabgap(1.5)  xsc(r(53, .)) legend(off)
> yla(1 "" "Europe and" "Central Asia" "" 2 "North America" 3 "South America",
> ang(h) noticks) xtitle(Life expectancy (years)) ytitle("") ;
```

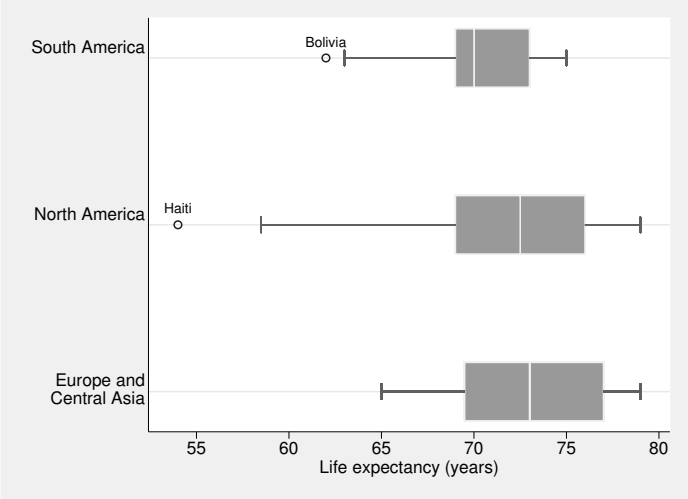



Figure 6. Horizontal box plots of life expectancy in 1998 for various countries in three regions, with improved positioning of marker labels for outliers

# 3 Moving beyond standard designs

Provided that you are broadly familiar with how `twoway` works, you should now have a sense that a small new world is open before you, in which you can add to, subtract from, or otherwise vary box plot designs exactly as you wish. If you do this repeatedly, you will want to encapsulate code for favored designs in a do-file or program. Explaining that further would take us beyond the main story, but both the User's Guide and Baum (2009) are excellent sources of advice and examples.

## 3.1 Adding means

One common request, on Statalist and elsewhere, is to add means to box plots. For this, you need an extra variable containing means. `egen` is again convenient:

```
. egen mean = mean(lexp), by(region)
```

We need to add a `scatter` call to the code above:

```
scatter region mean, ms(Dh) msize(*2) ||
```

A simple but crucial detail is plotting the means after, and therefore on top of, the boxes. Usually, although not inevitably, means will lie between the quartiles, and so their symbols would disappear under the boxes otherwise. Figure 7 shows the result.

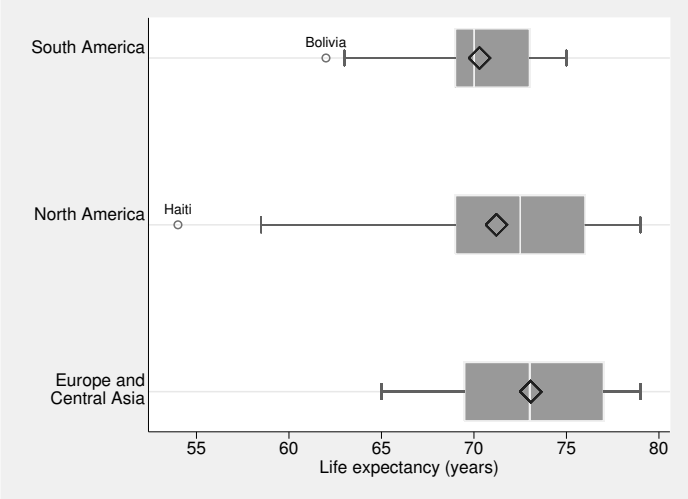



Figure 7. Horizontal box plots of life expectancy in 1998 for various countries in three regions; diamond symbols indicate means

## 3.2 Connecting medians

Another common request is connecting medians. One context for this could be that the box plots indicate variation within time periods. The connected medians thus would emphasize variation between time periods. This request is met as previously, by adding another `twoway` call such as

```
line median timevar, lw(*2)
```

or

```
line timevar median, lw(*2)
```

depending on whether plots are vertical or horizontal. Emphasis is added if and as desired, here by doubling line width. As before, plot connecting lines after, and so on top of, boxes.

## 3.3 Unequal spacing

Nothing in the `twoway` route to box plots commits you to equal spacing of box plots. Unequal spacing is perfectly possible: you just specify the positions of the box plots. Binning of responses or residuals in unequal intervals of a covariate is one large class of possible examples.

## 3.4 Variable width

If there were a desire for boxes of variable width, that could be met by repeated calls to `twoway rbar` with differing `barwidth()` options. `barwidth()` requires a single number as argument, and does not accept a numeric variable indicating width.

## 3.5 Percentile-based whiskers

Let us now imagine a different design in which whiskers are drawn out to 10% and 90% points. Cleveland (1985) showed such box plots. They have three advantages over the standard design. First, the definition of whiskers is of the same kind as the definition of boxes. Second, almost always, we see some detail in the tails. The exceptions when there is heavy tying in one or the other tail are also discernible. Third, to a very good approximation, drawing such box plots commutes with any monotonic transformation so that, for example, the box plot of a logged variable is the log of the box plot of the variable on the original scale. Some minor inaccuracy may arise in practice because quantiles may be calculated as the average of two order statistics: see the FAQ at http://www.stata.com/support/faqs/graphics/boxandlog.html for more on this thorny little detail.

Evidently, the choice of 10% and 90% is in no sense compulsory: other values may suit some purposes better.

We will also ensure that all points outside the whiskers are labeled. Because we are in complete control, we will go back to vertical, reverse box coloring and drop those whisker caps that we do not much like.

We know how to get further percentiles:

```
. egen p10 = pctile(lexp), p(10) by(region)
. egen p90 = pctile(lexp), p(90) by(region)
```

There are no new tricks needed for the graph, or so we might think; see figure 8.

```
. twoway rbar med upq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rbar med loq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rspike upq p90 region, pstyle(p1) ||
> rspike loq p10 region, pstyle(p1) ||
> scatter lexp region if !inrange(lexp, p10, p90), ms(Oh) mla(country)
> mlabgap(1.5) legend(off)
> xla(1 `" "Europe and" "Central Asia" "' 2 "North America" 3 "South America",
> noticks) yla(, ang(h)) ytitle(Life expectancy (years)) xtitle("")
```

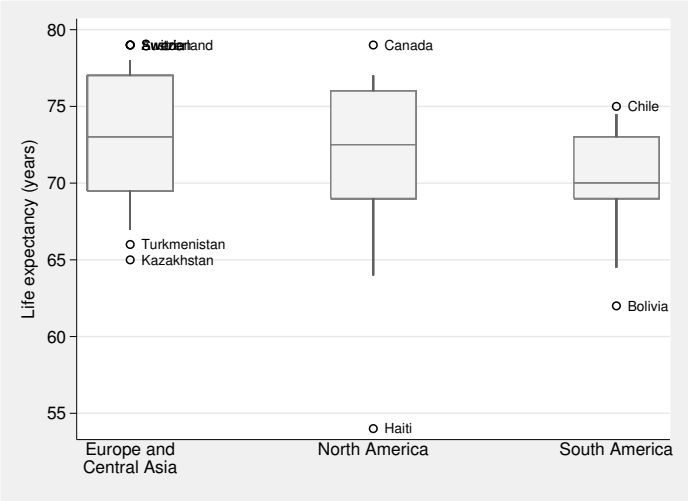



Figure 8. Box plots of life expectancy in 1998 for various countries in three regions; whiskers extend to 10% and 90% points of the distribution

A little mess of marker labels turns out to arise because Austria, Sweden, and Switzerland tie at 79 years. Some experimenting indicates that we can just rotate two of those labels away from the default position. Figure 9 is the improved graph.

```
. gen pos = cond(country == "Austria", 1, cond(country == "Sweden", 4, 3))
. twoway rbar med upq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rbar med loq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rspike upq p90 region, pstyle(p1) ||
> rspike loq p10 region, pstyle(p1) ||
> scatter lexp region if !inrange(lexp, p10, p90), ms(Oh) mla(country)
> mlabgap(1.5) legend(off) mlabvpos(pos)
> xla(1 "" "Europe and" "Central Asia" "" 2 "North America" 3 "South America",
> noticks) yla(, ang(h)) ytitle(Life expectancy (years)) xtitle("")
```

(*Continued on next page*)

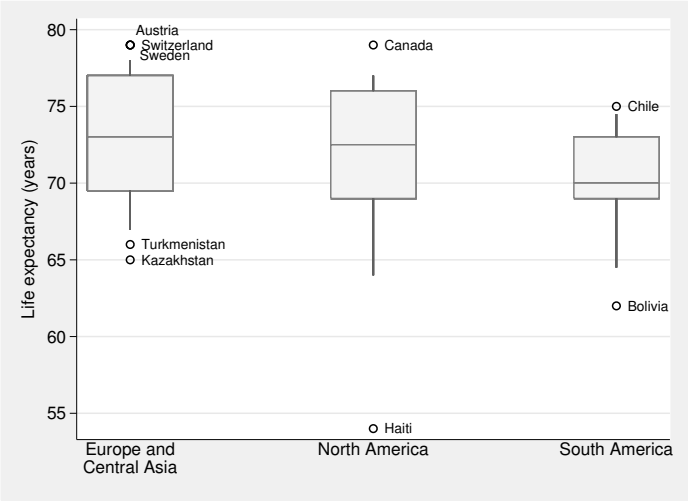



Figure 9. Box plots of life expectancy in 1998 for various countries in three regions; whiskers extend to 10% and 90% points of each distribution; marker labels for Austria and Sweden have been moved to avoid overlap

## 3.6 Other data structures

So far, we have considered only the case of one response variable, subdivided by groups of a categorical variable. Box plots are often needed for other data structures. We need to see that they are also within reach given a little technique.

Another dataset shipped with Stata contains temperature data for 956 cities in the United States, including variables `tempjan` and `tempjuly` indicating mean monthly temperatures for January and July. The cities are classified coarsely by `region` and more finely by `division`. We will produce box plots of the Cleveland (1985) kind for the two temperature responses `tempjan` and `tempjuly`, subdivided by division and month.

We could just superimpose box plots for `tempjan` and `tempjuly`, but a `reshape` of the data makes matters easier thereafter. `reshape` requires a unique identifier, so we put the observation number into a new variable to act as a pacifier. The identifier will play no role thereafter in our graphics. See the online help or [D] **reshape** if you need more discussion.

```
. sysuse citytemp, clear
. gen id = _n
. reshape long temp, i(id) string j(month)
```

The new string variable `month` takes on two values, `jan` and `july`. The summary statistics come from `egen`. The extra twist that we need to distinguish both `division` and `month` is easily satisfied:

```
. egen median = median(temp), by(division month)
. egen loq = pctile(temp), p(25)  by(division month)
. egen upq = pctile(temp), p(75)  by(division month)
. egen p10 = pctile(temp), p(10)  by(division month)
. egen p90 = pctile(temp), p(90)  by(division month)
```

`division` is an integer variable with values from 1 to 9 and value labels attached. To show box plots for January and July side by side, we just need a position variable in which months are offset. Cui (2007) gives further discussion of this simple trick. We still want to use the value labels of `division`, so we assign them to the new variable.

```
. gen division2 = division + cond(month == "jan", -0.2, 0.2)
. label val division2 division
```

The code is now very similar to previous examples. Figure 10 gives the result.

```
. twoway rbar median upq division2, bfc(gs15) blc(gs8) barw(0.35) ||
> rbar median loq division2, bfc(gs15) blc(gs8) barw(0.35) ||
> rspike loq p10 division2 ||
> rspike upq p90 division2 ||
> scatter temp division2 if !inrange(temp, p10, p90), ms(o) legend(off)
> xaxis(1 2) xla(1/9, valuelabel noticks grid axis(1))
> xla(1/9, valuelabel noticks axis(2)) xtitle("", axis(1)) xtitle("", axis(2))
> yaxis(1 2) yla(14(18)86, ang(h) axis(2))
> yla(14 "-10" 32 "0" 50 "10" 68 "20" 86 "30", ang(h) axis(1))
> ytitle(mean temperature ({c 176}F), axis(2))
> ytitle(mean temperature ({c 176}C), axis(1))
> ysc(titlegap(0) axis(1)) ysc(titlegap(0) axis(2))
> plotregion(lstyle(p1))
```

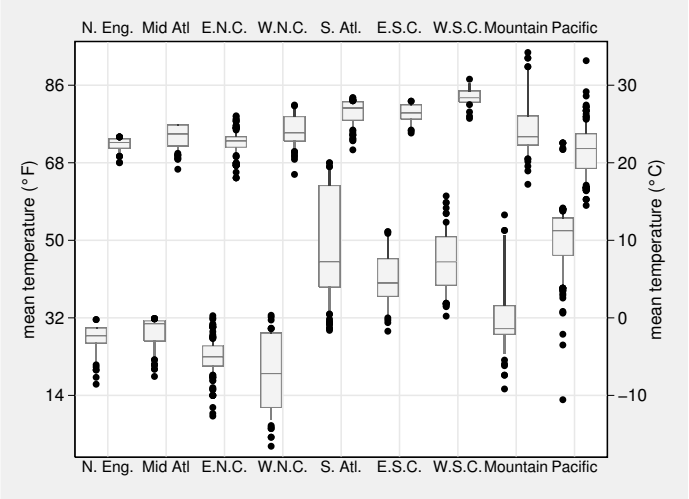



Figure 10. Box plots of mean temperatures in January (left plot in each group) and July (right plot in each group) for various places in divisions of the United States; whiskers extend to 10% and 90% points of each distribution

Here we know that each division will take up less space than in previous graphs, so we bump up the marker symbol to `ms(o)` so that they remain visible.

We also add a few `twoway` flourishes. Horizontal axis labels at the top as well as the bottom of the graph ease look-up of division labels. 0°F is of less importance than 32°F as a reference temperature. We also align equivalent temperatures on Fahrenheit and Celsius scales. Note the trick to get the degree symbol (Cox 2004).

Incidentally, when we look at the box plot to learn something about the data, we see that the upper tail of Mid-Atlantic July temperatures is curiously truncated. Inspection of the data shows that 20 places are all given a mean July temperature of 76.8°F. This is the highest temperature observed for that division but is also the 90% point, because those 20 are more than 10% of the places in the division. Thus no places are plotted as having a higher temperature than the 90% point. Hence the graph is correct in terms of the data, but the graph has also told us something new about the data, which is as it should be. People more familiar with how the U.S. Census reports temperature data may be able to throw more light on this little mystery.

### 3.7 Convenience and efficiency

The stress in this column has been on getting results conveniently using standard commands. It is nevertheless proper to repeat a note of caution sounded earlier. The commands used here are not the most efficient way to get box plots, nor will the graph files produced be as lean as they could be. For small or moderate datasets, you would have to strain to notice that, but otherwise you might be bitten. Industrial-strength alternatives to these commands would need to work at lower levels to optimize speed and storage, by replacing calls to `egen` with direct calls to `summarize` and by ensuring that the information defining box plot ingredients is not duplicated unnecessarily.

## 4 Conclusion

`graph box` and `graph hbox` are very useful commands, but they only do what they claim to do. This column has shown an alternative way to create, and then to vary, box plots, using `egen` for calculations and `twoway` for graphics. Once the problem is broken down into components, it can be solved without any programming. That then gives researchers scope for whatever variants of box plots are likely to prove interesting and useful. Cleveland's 1985 variant seems especially worthy of further consideration.

## 5 Errata to previous column

Marcel Zwahlen helpfully pointed out an inconsistency between an equation on p. 308 and the corresponding Mata code on p. 309 within the previous column (Cox 2009). The equation was incorrect.

$$K = \frac{N^2 - \sum_{i=1}^{I} n_i^2}{\sum_{i=1}^{I} n_i - Nq_i}$$

should have been

$$K = \frac{N^2 - \sum_{i=1}^{I} n_i^2}{\sum_{i=1}^{I} (n_i - Nq_i)^2}$$

# 6 References

Baum, C. F. 2009. *An Introduction to Stata Programming*. College Station, TX: Stata Press.

Chen, C., W. Härdle, and A. Unwin, ed. 2008. *Handbook of Data Visualization*. Berlin: Springer.

Cleveland, W. S. 1985. *The Elements of Graphing Data*. Monterey, CA: Wadsworth.

Cox, N. J. 2002. Speaking Stata: On getting functions to do the work. *Stata Journal* 2: 411–427.

———. 2004. Stata tip 6: Inserting awkward characters in the plot. *Stata Journal* 4: 95–96.

———. 2005. Speaking Stata: The protean quantile plot. *Stata Journal* 5: 442–460.

———. 2006. Stata tip 39: In a list or out? In a range or out? *Stata Journal* 6: 593–595.

———. 2007. Stata tip 47: Quantile–quantile plots without programming. *Stata Journal* 7: 275–279.

———. 2009. Speaking Stata: I. J. Good and quasi-Bayes smoothing of categorical frequencies. *Stata Journal* 9: 306–314.

Cox, N. J., and K. Jones. 1981. Exploratory data analysis. In *Quantitative Geography: A British View*, ed. N. Wrigley and R. J. Bennett, 135–143. London: Routledge and Kegan Paul.

Crowe, P. R. 1933. The analysis of rainfall probability: A graphical method and its application to European data. *Scottish Geographical Magazine* 49: 73–91.

Cui, J. 2007. Stata tip 42: The overlay problem: Offset for clarity. *Stata Journal* 7: 141–142.

De Veaux, R. D., P. F. Velleman, and D. E. Bock. 2008. *Stats: Data and Models*. 2nd ed. Boston, MA: Addison–Wesley.

Dümbgen, L., and H. Riedwyl. 2007. On fences and asymmetry in box-and-whiskers plots. *American Statistician* 61: 356–359.

Freedman, D., R. Pisani, and R. Purves. 2007. *Statistics*. 4th ed. New York: W. W. Norton.

Frigge, M., D. C. Hoaglin, and B. Iglewicz. 1989. Some implementations of the boxplot. *American Statistician* 43: 50–54.

Harris, R. L. 1999. *Information Graphics: A Comprehensive Illustrated Reference*. New York: Oxford University Press.

Hoaglin, D. C., F. Mosteller, and J. W. Tukey, ed. 1983. *Understanding Robust and Exploratory Data Analysis*. New York: Wiley.

Hyndman, R. J., and Y. Fan. 1996. Sample quantiles in statistical packages. *American Statistician* 50: 361–365.

Kleiner, B., and T. E. Graedel. 1980. Exploratory data analysis in the geophysical sciences. *Reviews of Geophysics* 18: 699–717.

McGill, R., J. W. Tukey, and W. A. Larsen. 1978. Variations of box plots. *American Statistician* 32: 12–16.

Mitchell, M. 2008. *A Visual Guide to Stata Graphics*. 2nd ed. College Station, TX: Stata Press.

Monkhouse, F. J., and H. R. Wilkinson. 1971. *Maps and Diagrams*. London: Methuen.

Spear, M. E. 1952. *Charting Statistics*. New York: McGraw–Hill.

———. 1969. *Practical Charting Techniques*. New York: McGraw–Hill.

Tukey, J. W. 1972. Some graphic and semigraphic displays. In *Statistical Papers in Honor of George W. Snedecor*, ed. T. A. Bancroft and S. A. Brown, 293–316. Ames, IA: Iowa State University Press.

———. 1977. *Exploratory Data Analysis*. Reading, MA: Addison–Wesley.

Velleman, P. F., and D. C. Hoaglin. 1981. *Applications, Basics, and Computing of Exploratory Data Analysis*. Boston, MA: Duxbury.

Wainer, H. 1990. Graphical visions from William Playfair to John Tukey. *Statistical Science* 5: 340–346.

#### About the author

Nicholas Cox is a statistically minded geographer at Durham University. He contributes talks, postings, FAQs, and programs to the Stata user community. He has also coauthored 15 commands in official Stata. He wrote several inserts in the *Stata Technical Bulletin* and is an editor of the *Stata Journal*.

The Stata Journal (2009)
**9**, Number 3, pp. 497–498

# Stata tip 77: (Re)using macros in multiple do-files

Jeph Herrin
Yale School of Medicine
Yale University
New Haven, CT
jeph.herrin@yale.edu

Local and global macros provide an extremely useful way to define (for example) groups of values or variable names that can be (re)used in data management and analysis. For example, you may want to fit many different models using a given subset, or several different subsets, of independent variables. So you might define

```
. local indvars1 "var1 var2"
. local indvars2 "var1 var2 var3 var4"
```

Thereafter, subsequent models can be easily specified with reference to the same sets of variables:

```
. logit y `indvars1'
. regress y `indvars1'
. logit y `indvars2'
. regress y `indvars2'
```

However, users often like to define macros that can be referenced within several do-files. One way to have macros persist across do-files is to use globals, but globals should generally be reserved for macros that truly are wanted to be available in all contexts. An alternative is to borrow from other programming environments the use of "header" files, which contain preamble code that can be included at the top (or head) of each file of code to make common definitions.

For this purpose, Stata offers the `include` command, which is similar to `run` except that all local macro definitions are retained. See the manual entry [P] **include** for complete details. In the above example, we could have a file called `locals.do`:

begin `locals.do`

```
local indvars1 "var1 var2"
local indvars2 "var1 var2 var3 var4"
```

end `locals.do`

Then, in any file in which we would like to use these local macros, we can simply type

```
include locals.do
```

and thereafter refer to `indvars1`, `indvars2`, etc. If we subsequently want to modify the independent variables in our lists, we need only edit the `locals.do` file. The changes are automatically carried into other do-files that `include` that `locals.do` file.

This very useful feature can also be used to define directory paths and other programming data that we want to be available in local macros to all our do-files. For instance, a useful definition is

```
local today=string(date("`c(current_date)'","DMY"),"%tdCCYY.NN.DD")
```

which creates a string containing the current date in a lexically ordered format. Including this in a header file, and thence in all do-files, gives a standard way of adding a sortable date suffix to all saved files. You may `include` more than one file, as well as nesting `include`s, so that each project can `include` a `project.do` file that not only defines project-specific macros but also `include`s a `master.do`, which defines more general macros.

In summary, the use of `include` to call a file containing macro definitions allows one instance of those local (or global) macros to be made easily available to multiple do-files.

The Stata Journal (2009)
**9**, Number 3, pp. 499–503

# Stata tip 78: Going gray gracefully: Highlighting subsets and downplaying substrates

Nicholas J. Cox
Department of Geography
Durham University
Durham, UK
n.j.cox@durham.ac.uk

In graphics, as in life, going gray is often forced upon us, yet it is also occasionally a deliberate choice. Journals may enforce publication of your graphs in black and white whenever full-blown color is prohibited, or else prohibitively expensive. Even when allowed, color may prove problematic for various and quite different reasons, ranging from physiology and psychology to sociology and aesthetics. For example, many people are red–green color-blind, while the spectral or rainbow sequence from red to violet is not in fact perceived as a monotonic scale. Wilkinson (2005) and Ware (2008) give good introductions to the use of color in visualization. Fortner and Meyer (1997) give a more detailed discussion. Brewer (2005) gives many specific suggestions on color schemes, which are as appropriate for statistical graphics as they are for cartography.

Choosing differing shades of gray is also worth consideration for positive reasons. Expressing qualitative contrasts just with gray can be both effective and attractive. A previous Stata tip (Cox 2005) showed how distinct values on an ordered scale could be shown separately on scatterplots by markers of different gray-scale colors. This tip expands the theme with further examples. Naturally, black and white themselves qualify as extreme gray shades and may work very well. Here I will emphasize the use of intermediate shades. Let me underline that the examples here use the `sj` scheme. See [G] **schemes intro** for more information.

Consider the highlighting of subsets. You may want to show the distribution of a subset with the distribution of the complete set as context. Unwin, Theus, and Hofmann (2006); Chen, Härdle, and Unwin (2008); and Myatt and Johnson (2009) include several examples of this device for various kinds of graphs.

On a histogram with a frequency scale, this can be done by laying down the distribution of the complete set first and plotting the distribution of the subset on top. Display of the subset can never occlude the display of the complete set, because at most all the observations in any bin belong to the subset.

For example, after reading in a dataset from the U.S. National Longitudinal Study of Young Women in 1988,

```
. sysuse nlsw88
```

we may look at the wage distribution for college graduates compared with the complete set. Figure 1 shows such a histogram.

```
. twoway histogram wage, freq width(1) bcolor(gs14) blw(*.4) blcolor(black)
> || histogram wage if collgrad, yla(, ang(h)) xtitle(hourly wage (USD))
> ytitle(frequency) freq width(1) bcolor(gs6) blw(*.4) blcolor(black) legend(off)
```

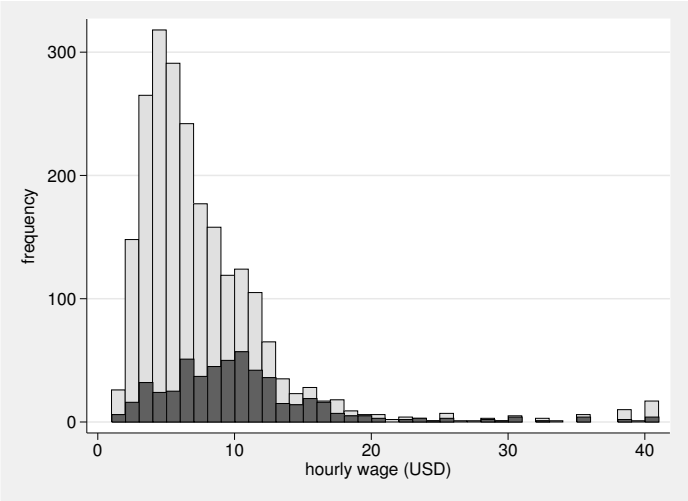



Figure 1. Wage distribution. College graduates are highlighted.

See [G] **graph twoway histogram** if you desire more detail on the command. The important detail here is spelling out that you want the same binning for comparability, as specified by `width()` and—if necessary—`start()`. Otherwise, the choices are matters of taste. For a written report, the legend is arguably dispensable, because what is being highlighted can be explained in the caption that you write within your word processor or text editor, as in this tip. For a talk, the need to make a graph self-explanatory might indicate otherwise.

The same distinction between complete set as backdrop and subset as highlight can be used in other plots. We will look at a scatterplot of wage against educational grade completed. Grade is discrete, but wage is not. We will do our own jittering of grade (only) by adding uniform noise beforehand, if only because that ensures consistency between graphs, and we plot wage on a logarithmic scale. Figure 2 shows a scatterplot with college graduates highlighted once again.

```
. generate grade2 = grade + .5 * (runiform() - .5)
. label var grade2 "`: var label grade'"
. scatter wage grade2, ms(Oh) mc(gs10) ysc(log)
> || scatter wage grade2 if collgrad, legend(off) ms(O) mc(gs2) ysc(log)
> ytitle(hourly wage (USD)) xla(0 4/18) yla(40 20 10 5 2, ang(h))
```

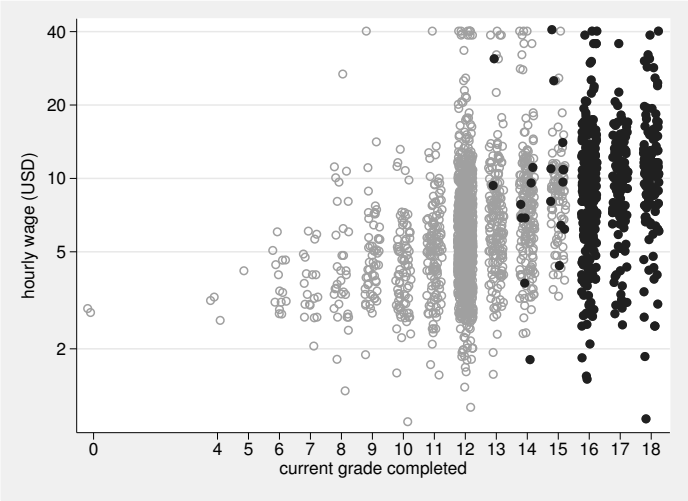



Figure 2. Wage and grade. College graduates are highlighted. Grade is jittered to give a better impression of variation.

Note that the `': '` construct inserts a variable label on the fly; see [P] **macro** for more details. Hollow and filled marker symbols, such as `Oh` and `O`, are helpfully complementary.

Sometimes we want to highlight an exploratory smooth or a fitted model prediction and correspondingly downplay the substrate of the data. With this dataset, noneconomists can join economists in being unsurprised at the great variability of wage within grade. All are likely to be much more interested in the average relationship. Nevertheless, suppressing the data on a graph would often be excessive, if not dishonest. Let us first smooth on a logarithmic scale using restricted cubic splines, experimenting only with default choices; [R] **mkspline** includes details and references. Note that the smooth is calculated for the unjittered grades.

```
. gen lnwage = ln(wage)
. mkspline spline = grade, cubic
. regress lnwage spline?
. predict smooth
```

Figure 3 shows a scatterplot with overlaid smooth. We have to do a little work to get $y$-axis labels in dollars. See Cox (2008) for further discussion. The logic, however, is easy: we just need to spell out which axis labels we want and where to put them. With this scheme, Stata automatically makes the scatterplot lighter than black, but it seems that we can fairly go further.

```
. scatter lnwage grade2, || mspline smooth grade, xla(0 4/18)
> legend(off) ytitle(hourly wage (USD))
> yla(`= ln(40)' "40" `=ln(20)' "20" `=ln(10)' "10" `=ln(5)' "5" `=ln(2)' "2",
> ang(h))
```

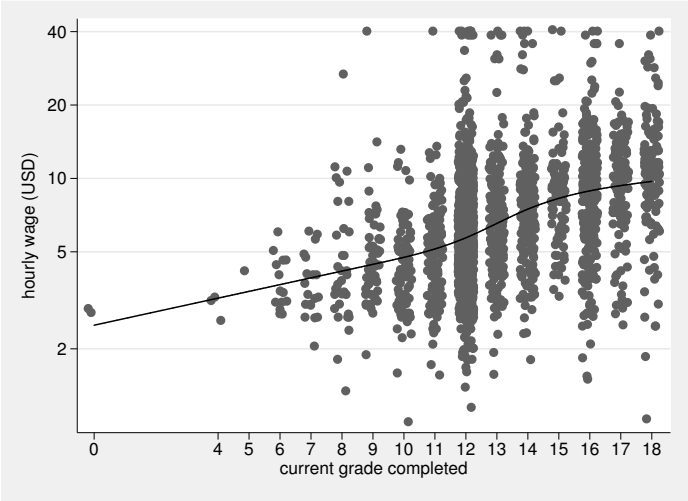



Figure 3. Restricted cubic spline smooth of wage versus grade on a logarithmic scale. Grade is jittered to give a better impression of variation in data.

Figure 4 shows the data points using a lighter gray and increases the width of the smooth. The result is likely to be closer to the researcher's message.

```
. scatter lnwage grade2, mcolor(gs10) || mspline smooth grade, lw(*3) lp(solid)
> xla(0 4/18) legend(off) ytitle(hourly wage (USD))
> yla(`= ln(40)` "40" `=ln(20)` "20" `=ln(10)` "10" `=ln(5)` "5" `=ln(2)` "2",
> ang(h))
```

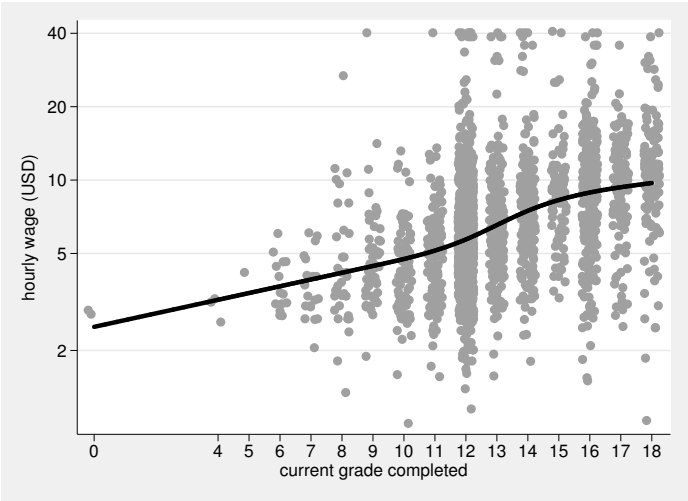



Figure 4. Restricted cubic spline smooth of wage on a logarithmic scale. Grade is jittered to give a better impression of variation in data. Note how the data are downplayed and the smooth highlighted compared with figure 3.

The ideas here can be taken in various further directions. Gray scale can be good for showing the scaffolding of the graph (axes, grids, and so forth) in a subdued but still discernible manner. Highlighting subsets can be extended to show three or more subsets. For example, to show a frequency-based histogram of three subsets, lay down the total distribution of all, followed by the total of two, followed by that of one, so that occlusion produces the desired effect. Alternatively, use `graph bar` or `graph hbar` to produce stacked or subdivided bars, introducing as much or as little histogram style as desired.

## References

Brewer, C. A. 2005. *Designing Better Maps: A Guide for GIS Users*. Redlands, CA: ESRI Press.

Chen, C., W. Härdle, and A. Unwin, ed. 2008. *Handbook of Data Visualization*. Berlin: Springer.

Cox, N. J. 2005. Stata tip 27: Classifying data points on scatter plots. *Stata Journal* 5: 604–606.

———. 2008. Stata tip 59: Plotting on any transformed scale. *Stata Journal* 8: 142–145.

Fortner, B., and T. E. Meyer. 1997. *Number by Colors: A Guide to Using Color to Understand Technical Data*. New York: Springer.

Myatt, G. J., and W. P. Johnson. 2009. *Making Sense of Data II: A Practical Guide to Data Visualization, Advanced Data Mining Methods, and Applications*. Hoboken, NJ: Wiley.

Unwin, A., M. Theus, and H. Hofmann. 2006. *Graphics of Large Datasets: Visualizing a Million*. New York: Springer.

Ware, C. 2008. *Visual Thinking for Design*. Burlington, MA: Morgan Kaufmann.

Wilkinson, L. 2005. *The Grammar of Graphics*. 2nd ed. New York: Springer.

The Stata Journal (2009)
**9**, Number 3, p. 504

# Stata tip 79: Optional arguments to options

Nicholas J. Cox
Department of Geography
Durham University
Durham, UK
n.j.cox@durham.ac.uk

Programmers occasionally would like an option for a program to come in two flavors: a simple or default option, with no arguments, and a more complicated but more flexible alternative, with arguments. For example, the simple option might call up a graph with programmer-chosen defaults, while the complicated option might pass graph options to the graph command in question, signaling variations from those defaults.

With a simple trick, you can implement two options that appear to the user to be this single option that is either simple or complicated. Following age-old programmer jargon, let us imagine an option that can be `foobar` or `foobar(`*arguments*`)`.

Step 1: Declare to `syntax` that there are two options, say, `foobar` and `FOOBAR2()`, and the latter is precisely, say, `FOOBAR2(string)`. The outburst of uppercase letters indicates to `syntax` that the latter can be abbreviated `foobar()`. You can also indicate names that can be abbreviated more, say, `FOObar` and `FOObar2(string)`.

Step 2: Process input within your program. For example,

```
if "`foobar'`foobar2'" != ""
```

is a test of whether either option has been called. If `foobar2()` has been called, then the local macro `foobar2` will be defined and can be treated further. If the argument to `foobar2()` might itself contain quotation marks, then compound double quotes, `‘" "’`, are in order.

Step 3: In documentation for the user, you need not mention the two options but may merely declare that the syntax is that an argument is optional, e.g., `foobar`[ `(`*string*`)` ].

If curious users find out by looking at the code that the option with arguments is really `foobar2()`, no harm is done. They would be partway to working out, independently of this tip, how the optional options are coded. Browsing code and borrowing tricks that you want to use yourself remains one of the best ways to grow as a programmer.

# Announcements

**Seminars on Stata**

| | |
|---|---|
| Date: | Thursday, October 21, 2009 |
| Venue: | Hotel Monaco |
| | Washington, D.C. |
| Speakers: | Roberto G. Gutierrez |
| | *Director of Statistics, StataCorp* |
| | Bill Rising |
| | *Director of Educational Services, StataCorp* |
| Logistics organizers: | StataCorp |
| More information: | http://www.stata.com/meeting/dcsem09/ |

**Canadian Stata Users Group meeting: Program announced**

| | |
|---|---|
| Date: | Thursday, October 22, 2009 |
| Venue: | Pantages Hotel |
| | Toronto, Ontario |
| | Canada |
| Scientific organizers: | Leslie-Anne Keown |
| | *Statistics Canada* |
| | Louis Gagnon |
| | *Queens University* |
| Logistics organizer: | StataCorp |
| More information: | http://www.stata.com/meeting/canada09/ |

**Australian and New Zealand Stata Users Group meeting: Announcement**

| | |
|---|---|
| Date: | Thursday, November 5, 2009 |
| Venue: | Darlington Centre<br>University of Sydney<br>Australia |
| Scientific organizers: | Demetris Christodoulou<br>*University of Sydney, Faculty of Economics and Business*<br>Vasilis Sarafidis<br>*University of Sydney, Faculty of Economics and Business* |
| Logistics organizer: | Survey Design, distributor of Stata<br>in Australia and New Zealand |
| More information: | http://www.stata.com/meeting/australia09/ |

**Italian Stata Users Group meeting: Announcement**

| | |
|---|---|
| Date: | Thursday, November 19, and Friday, November 20, 2009 |
| Venue: | L'Hotel Anglo American<br>Florence, Italy |
| Scientific organizers: | Una-Louis Bell<br>*TStat S.r.l.*<br>Rino Bellocco<br>*Karolinska Institutet*<br>Giovanni Capelli<br>*Universitá degli Studi di Cassino*<br>Maurizio Pisati<br>*Universitá degli Studi di Milano Bicocca*<br>Marcello Pagano<br>*Harvard School of Public Health* |
| Logistics organizer: | TStat S.r.l., distributor of Stata<br>in Italy |
| More information: | http://www.stata.com/meeting/italy09/ |

**German Stata Users Group meeting: Proceedings available**

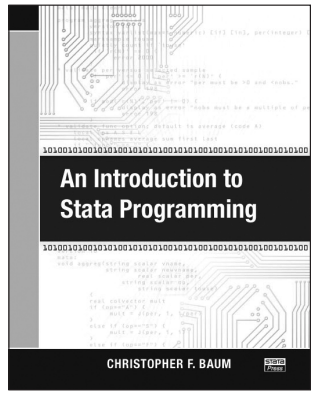
| | |
|---|---|
| Web site: | http://www.stata.com/meeting/germany09/ |

**DC09 Stata Conference: Proceedings available**

| | |
|---|---|
| Web site: | http://www.stata.com/meeting/dcconf09/ |

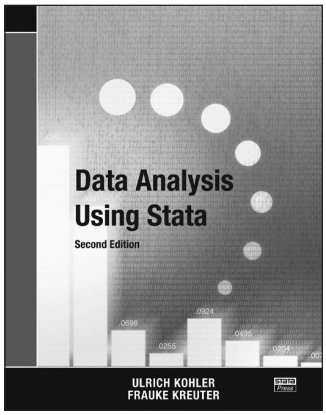
# Recently released titles from Stata Press

## An Introduction to Stata Programming




Christopher F. Baum's *An Introduction to Stata Programming* steps the reader through the three levels of Stata programming: do-files, ado-files, and programming with Mata. Baum's programming examples cover an array of topics, illustrate some of Stata's built-in tools (such as the resampling techniques of bootstrapping and jackknifing), and offer solutions to tricky data-management questions. He even shows how users can write their own custom estimation commands by using both Stata's built-in numerical maximum-likelihood estimation routine, `ml`, and its built-in nonlinear least-squares routines, `nl` and `nlsur`. If you are interested in programming in Stata, this book is for you. Find out more at **www.stata-press.com/books/isp.html**.

Christopher F. Baum • 362 pages • ISBN 978-1-59718-045-0 • $54

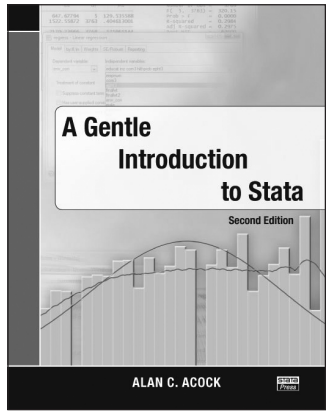



## Data Analysis Using Stata, Second Edition

*Data Analysis Using Stata, Second Edition* comprehensively introduces Stata and will be useful to those who are just learning statistics and Stata, as well as to users of other statistical packages who are making the switch to Stata. The authors take a hands-on approach, leading you step by step through actual Stata sessions to answer practical questions commonly asked by social scientists. Find out more at **www.stata-press.com/books/daus2.html**.

Ulrich Kohler and Frauke Kreuter • 388 pages •
ISBN 978-1-59718-046-7 • $52

## A Gentle Introduction to Stata, Second Edition




*A Gentle Introduction to Stata, Second Edition* is aimed at new Stata users who want to become proficient in Stata. Acock assumes that the user is not familiar with any statistical software. Important asides and notes about terminology are set off in boxes, which makes the text easy to read without any convoluted twists or forward-referencing. The focus of the book is especially helpful for those in psychology and the social sciences. Find out more at **www.stata-press.com/books/acock2.html**.

Alan C. Acock • 333 pages • ISBN 978-1-59718-043-6 • $46

# THE STATA JOURNAL

## back issues

Previous articles from the *Stata Journal* are available online. Those three or more years old may be obtained without charge. More recent articles cost $7.50. To view the table of contents for all issues or to purchase an article, visit

**http://www.stata-journal.com/archives.html**

The *Stata Journal* volume 6, number 3, may now be obtained without charge. Here is the table of contents for this issue.


**Articles and Columns**

Maximum likelihood estimation of endogenous switching and sample selection models for binary, ordinal, and count variables ..... A. Miranda and S. Rabe-Hesketh

Confidence intervals for rank statistics: Somers' D and extensions ..... R. Newson

Tests and confidence sets with correct size when instruments are potentially weak .... A. Mikusheva and B. P. Poi

Graphical representation of interactions ..... F. M.-S. Barthel and P. Royston

Jackknife instrumental variables estimation in Stata ..... B. P. Poi

Difference-based semiparametric estimation of partial linear regression models ..... M. Lokshin

Importing Federal Reserve economic data ..... D. M. Drukker

Mata Matters: Interactive use ..... W. Gould

Speaking Stata: Graphs for all seasons ..... N. J. Cox

Review of A Gentle Introduction to Stata by Acock ..... M. Mulcahy

**Notes and Comments**

Stata tip 34: Tabulation by listing ..... D. A. Harrison

Stata tip 35: Detecting whether data have changed ..... W. Gould

Stata tip 36: Which observations? ..... N. J. Cox

**Software Updates**