STATA
TECHNICAL
BULLETIN

November 1999
STB-52

A publication to promote communication among Stata users

Editor

H. Joseph Newton
Department of Statistics
Texas A & M University
College Station, Texas 77843
409-845-3142
409-845-3144 FAX
stb@stata.com EMAIL

Associate Editors

Nicholas J. Cox, University of Durham
Francis X. Diebold, University of Pennsylvania
Joanne M. Garrett, University of North Carolina
Marcello Pagano, Harvard School of Public Health
J. Patrick Royston, Imperial College School of Medicine

Subscriptions are available from Stata Corporation, email stata@stata.com, telephone 979-696-4600 or 800-STATAPC, fax 979-696-4601. Current subscription prices are posted at www.stata.com/bookstore/stb.html.

Previous Issues are available individually from StataCorp. See www.stata.com/bookstore/stbj.html for details.

Submissions to the STB, including submissions to the supporting files (programs, datasets, and help files), are on a nonexclusive, free-use basis. In particular, the author grants to StataCorp the nonexclusive right to copyright and distribute the material in accordance with the Copyright Statement below. The author also grants to StataCorp the right to freely use the ideas, including communication of the ideas to other parties, even if the material is never published in the STB. Submissions should be addressed to the Editor. Submission guidelines can be obtained from either the editor or StataCorp.

The *Stata Technical Bulletin* (ISSN 1097-8879) is published six times per year by Stata Corporation. Stata is a registered trademark of Stata Corporation.

Contents of this issue

dm45.2 Changing string variables to numeric: correction

Nicholas J. Cox, University of Durham, UK, n.j.cox@durham.ac.uk

Syntax

`destring` [*varlist*] [, `noconvert noencode float`]

Remarks

`destring` was published in STB-37. Please see Cox and Gould (1997) for a full explanation and discussion. It was translated into the idioms of Stata 6.0 by Cox (1999). Here the program is corrected so that it can correctly handle any variable labels that include double quotation marks. Thanks to Jens M. Lauritsen, who pointed out the need for this correction.

References

Cox, N. J. 1999. dm45.1: Changing string variables to numeric: update. *Stata Technical Bulletin* 49: 2.

Cox, N. J. and W. Gould. 1997. dm45: Changing string variables to numeric. *Stata Technical Bulletin* 37: 4-6. Reprinted in *The Stata Technical Bulletin Reprints*, vol. 7, pp. 34–37.

dm72.1 Alternative ranking procedures: update

Nicholas J. Cox, University of Durham, UK, n.j.cox@durham.ac.uk
Richard Goldstein, richgold@ix.netcom.com

The `egen` functions `rankf( )`, `rankt( )` and `ranku( )` published in STB-51 (Cox and Goldstein 1999) have been revised so that the variable labels of the new variables generated refer respectively to "field", "track" and "unique" ranks. For more information, please see the original insert.

References

Cox, N. J. and R. Goldstein. 1999. Alternative ranking procedures. *Stata Technical Bulletin* 51: 5–7.

dm73 Using categorical variables in Stata

John Hendrickx, University of Nijmegen, Netherlands, j.hendrickx@mailbox.kun.nl

Introduction

Dealing with categorical variables is not one of Stata's strongest points. The `xi` program can generate dummy variables for use in regression procedures, but it has several limitations. You can use any type of parameterization, as long as it is the indicator contrast (that is, dummy variables with a fixed reference category). Specifying the reference category is clumsy, third order or higher interactions are not available, and the cryptic variable names make the output hard to read.

The new program `desmat` described in this insert was created to address these issues. `desmat` parses a list of categorical and/or continuous variables to create a set of dummy variables (a DESign MATrix). Different types of parameterizations can be specified, on a variable-by-variable basis if so desired. Higher order interactions can be specified either with or without main effects and nested interactions. The dummy variables produced by `desmat` use the unimaginative name `_x_*`, which allows them to be easily included in any Stata procedure but is hardly an improvement over `xi`'s output. Instead, a companion program `desrep` is used after estimation to produce a compact overview with informative labels. A second companion program `tstall` can be used to perform a Wald test on all model terms.

Example

Knoke and Burke (1980, 23) present a four-way table of race by education by membership by vote turnout. We can construct their data by

```
. #delimit ;
. tabi 114 122 \
>      150  67 \
>       88  72 \
>      208  83 \
>       58  18 \
>      264  60 \
>       23  31 \
>       22   7 \
```

```
>       12   7 \
>       21   5 \
>        3   4 \
>       24  10, replace;
```

(*output omitted*)

```
        Pearson chi2(11) = 104.5112   Pr = 0.000
. #delimit cr
. rename col vote
. gen race=1+mod(group(2)-1,2)
. gen educ=1+mod(group(6)-1,3)
. gen memb=1+mod(group(12)-1,2)
. label var race "Race"
. label var educ "Education"
. label var memb "Membership"
. label var vote "Vote Turnout"
. label def race 1 "White" 2 "Black"
. label def educ 1 "Less than High School" 2 "High School Graduate" 3 "College"
. label def memb 1 "None" 2 "One or More"
. label def vote 1 "Voted" 2 "Not Voted"
. label val race race
. label val educ educ
. label val memb memb
. label val vote vote
. table educ vote memb [fw=pop], by(race)
----------------------+-------------------------------------------
                      |         Membership and Vote Turnout
                      | ------- None -------    ---- One or More ---
   Race and Education |     Voted  Not Voted        Voted  Not Voted
----------------------+-------------------------------------------
White                 |
Less than High School |       114        122          150         67
 High School Graduate |        88         72          208         83
              College |        58         18          264         60
----------------------+-------------------------------------------
Black                 |
Less than High School |        23         31           22          7
 High School Graduate |        12          7           21          5
              College |         3          4           24         10
----------------------+-------------------------------------------
```

Their loglinear model VM–VER–ERM can be specified as

```
. desmat vote*memb vote*educ*race educ*race*memb
```

`desmat` will produce the following summary output:

```
Note: collinear variables are usually duplicates and no cause for alarm
vote (Not Voted) dropped due to collinearity
educ (High School Graduate) dropped due to collinearity
educ (College) dropped due to collinearity
race (Black) dropped due to collinearity
educ.race (High School Graduate.Black) dropped due to collinearity
educ.race (College.Black) dropped due to collinearity
memb (One or More) dropped due to collinearity
Desmat generated the following design matrix:
nr   Variables       Term                        Parameterization
     First    Last
 1    _x_1           vote                        ind(1)
 2    _x_2           memb                        ind(1)
 3    _x_3           vote.memb                   ind(1).ind(1)
 4    _x_4    _x_5   educ                        ind(1)
 5    _x_6    _x_7   vote.educ                   ind(1).ind(1)
 6    _x_8           race                        ind(1)
 7    _x_9           vote.race                   ind(1).ind(1)
 8   _x_10   _x_11   educ.race                   ind(1).ind(1)
 9   _x_12   _x_13   vote.educ.race              ind(1).ind(1).ind(1)
10   _x_14   _x_15   educ.memb                   ind(1).ind(1)
11   _x_16           race.memb                   ind(1).ind(1)
12   _x_17   _x_18   educ.race.memb              ind(1).ind(1).ind(1)
```

Note that the information on collinear variables will usually be irrelevant because the dropped variables are only duplicates. `desmat` reports this information nevertheless in case variables are dropped due to actual collinearity rather than simply duplication. The 18 dummy variables use the "indicator contrast," that is, dummy variables with the first category as reference. See below for other types of parameterizations. The dummies can be included in a program as follows:

```
glm pop _x_*, link(log) family(poisson)
```

which produces the following results:

```
Residual df  =         5                              No. of obs =        24
Pearson X2   =  4.614083                              Deviance   =   4.75576
Dispersion   =  .9228166                              Dispersion =   .951152

Poisson distribution, log link
------------------------------------------------------------------------------
     pop |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    _x_1 |   .0209327   .1106547      0.189   0.850      -.1959466     .237812
    _x_2 |   .2318663    .107208      2.163   0.031       .0217426    .4419901
    _x_3 |  -.7678299   .1197021     -6.415   0.000      -1.002442   -.5332181
    _x_4 |   -.296346    .122656     -2.416   0.016      -.5367473   -.0559446
    _x_5 |  -.7932972   .1459617     -5.435   0.000      -1.079377   -.5072175
    _x_6 |   -.191805    .141013     -1.360   0.174      -.4681855    .0845755
    _x_7 |  -.8444554    .163886     -5.153   0.000      -1.165666   -.5232447
    _x_8 |  -1.510868    .197221     -7.661   0.000      -1.897414   -1.124322
    _x_9 |   .0700793   .2444066      0.287   0.774      -.4089487    .5491074
   _x_10 |  -.4455942   .3384535     -1.317   0.188      -1.108951    .2177626
   _x_11 |  -1.188121   .4732461     -2.511   0.012      -2.115666   -.2605759
   _x_12 |  -.5003494   .4320377     -1.158   0.247      -1.347128    .3464289
   _x_13 |   .7229824   .4324658      1.672   0.095      -.1246349      1.5706
   _x_14 |   .6475193   .1384515      4.677   0.000       .3761594    .9188792
   _x_15 |   1.396652   .1611304      8.668   0.000       1.080843    1.712462
   _x_16 |  -.5248042   .2527736     -2.076   0.038      -1.020231    -.029377
   _x_17 |   .1695274   .4096571      0.414   0.679      -.6333859    .9724406
   _x_18 |   .7831378   .5068571      1.545   0.122       -.210284     1.77656
   _cons |   4.760163   .0858069     55.475   0.000       4.591985    4.928342
------------------------------------------------------------------------------
```

The legend produced by `desmat` could be used to associate the parameters with the appropriate variables. However, it is easier to use the program `desrep` to summarize the results using informative labels:

```
. desrep
Effect                                                  Coeff        s.e.
vote
  Not Voted                                              0.021       0.111
memb
  One or More                                            0.232*      0.107
vote.memb
  Not Voted.One or More                                 -0.768**     0.120
educ
  High School Graduate                                  -0.296*      0.123
  College                                               -0.793**     0.146
vote.educ
  Not Voted.High School Graduate                        -0.192       0.141
  Not Voted.College                                     -0.844**     0.164
race
  Black                                                 -1.511**     0.197
vote.race
  Not Voted.Black                                        0.070       0.244
educ.race
  High School Graduate.Black                            -0.446       0.338
  College.Black                                         -1.188*      0.473
vote.educ.race
  Not Voted.High School Graduate.Black                  -0.500       0.432
  Not Voted.College.Black                                0.723       0.432
educ.memb
  High School Graduate.One or More                       0.648**     0.138
  College.One or More                                    1.397**     0.161
race.memb
  Black.One or More                                     -0.525*      0.253
educ.race.memb
  High School Graduate.Black.One or More                 0.170       0.410
```

```
  College.Black.One or More                          0.783       0.507
_cons                                                4.760**     0.086
```

`xi` creates a unique stub name for each model term, making it easy to test their significance. After using `desmat`, the program `tstall` can be used instead to perform a Wald test on each model term. Global macro variables `term*` are also available for performing tests on specific terms only.

```
. tstall
Testing vote:
 ( 1)  _x_1 = 0.0
           chi2(  1) =    0.04
         Prob > chi2 =    0.8500
Testing memb:
 ( 1)  _x_2 = 0.0
           chi2(  1) =    4.68
         Prob > chi2 =    0.0306
Testing vote.memb:
 ( 1)  _x_3 = 0.0
           chi2(  1) =   41.15
         Prob > chi2 =    0.0000
Testing educ:
 ( 1)  _x_4 = 0.0
 ( 2)  _x_5 = 0.0
           chi2(  2) =   29.73
         Prob > chi2 =    0.0000
  (output omitted)
Testing race.memb:
 ( 1)  _x_16 = 0.0
           chi2(  1) =    4.31
         Prob > chi2 =    0.0379
Testing educ.race.memb:
 ( 1)  _x_17 = 0.0
 ( 2)  _x_18 = 0.0
           chi2(  2) =    2.39
         Prob > chi2 =    0.3025
```

Syntax of desmat

`desmat` *model* [, *default_parameterization*]

The model consists of one or more terms separated by spaces. A term can be a single variable, two or more variables joined by period(s), or two or more variables joined by asterisk(s). A period is used to specify an interaction effect as such, whereas an asterisk indicates hierarchical notation, in which both the interaction effect itself and all possible nested interactions and main effects are included. For example, the term `vote*educ*race` is expanded to `vote educ vote.educ race vote.race educ.race vote.educ.race`.

The model specification may be followed optionally by a comma and a default type of parameterization. A parameterization can be specified as a name, of which the first three characters are significant, optionally followed by a specification of the reference category in parentheses (no spaces). The reference category should refer to the category number, not the category value. Thus for a variable with values 0 to 3, the parameterization `dev(1)` indicates that the deviation contrast is to be used with the first category (that is, 0) as the reference. If no reference category is specified, or the category specified is less than 1, then the first category is used as the reference category. If the reference category specified is larger than the number of categories, then the highest category is used. Notice that for certain types of parameterizations, the "reference" specification has a different meaning.

Parameterization types

The available parameterization types are specified as `name` or `name(`*ref*`)` where the following choices are available.

`dev(`*ref*`)` indicates the deviation contrast. Parameters sum to zero over the categories of the variable. The parameter for *ref* is omitted as redundant, but can be found from minus the sum of the estimated parameters.

`ind(`*ref*`)` indicates the indicator contrast, that is, dummy variables with *ref* as the reference category. This is the parameterization used by `xi` and the default parameterization for `desmat`.

`sim(`*ref*`)` indicates the simple contrast with *ref* as reference category. The highest order effects are the same as indicator contrast effects, but lower order effects and the constant will be different.

`dif(`*ref*`)` indicates the difference contrast, for ordered categories. Parameters are relative to the previous category. If the first letter of *ref* is '`b`', then the backward difference contrast is used instead, and parameters are relative to the previous category.

`hel(`*ref*`)` indicates the Helmert contrast, for ordered categories. Estimates represent the contrast between that category and the mean value for the remaining categories. If the first letter of *ref* is '`b`', then the reverse Helmert contrast is used instead, and parameters are relative to the mean value of the preceding categories.

`orp(`*ref*`)` indicates orthogonal polynomials of degree *ref*. The first dummy models a linear effect, the second a quadratic, etc. This option calls `orthpoly` to generate the design (sub)matrix.

`use(`*ref*`)` indicates a user-defined contrast. *ref* refers to a contrast matrix with the same number of columns as the variable has categories, and at least one fewer rows. If row names are specified for this matrix, these names will be used as variable labels for the resulting dummy variables. (Single lowercase letters as names for the contrast matrix cause problems at the moment; for example, `use(c)`. Use uppercase names or more than one letter, for example, `use(cc)` or `use(C)`.)

`dir` indicates a direct effect, used to include continuous variables in the model.

Parameterizations per variable

Besides specifying a default parameterization after specification of the model, it is also possible to specify a specific parameterization for certain variables. This is done by appending `=par[(ref)]` to a single variable, `=par[(ref)].par[(ref)]` to an interaction effect, `=par[(ref)]*par[(ref)]` to an interaction using hierarchical notation. A somewhat silly example:

```
. desmat race=ind(1) educ=hel memb vote vote.memb=dif.dev(1), ind(9)
```

The indicator contrast with the highest category as reference will be used for `memb` and `vote` since the default reference category is higher than the maximum number of categories of any variable. The variable `race` will use the indicator contrast as well but with the first category as reference, other effects will use the contrasts specified. Interpreting this mishmash of parameterizations would be quite a chore.

Useful applications of the parameterization per variable feature could be to specify that some of the variables are continuous while the rest are categorical, specifying a different reference category for certain variables, specifying a variable for the effects of time as a low order polynomial, and so on.

On parameterizations

Models with categorical variables require restrictions of some type in order to avoid linear dependencies in the design matrix, which would make the model unidentifiable (Bock 1975, Finn 1974). The parameterization, that is, the type of restriction used does not affect the fit of the model but does affect the interpretation of the parameters. A common restriction is to drop the dummy variable for a reference category (referred to here as the indicator contrast). The parameters for the categorical variable are then relative to the reference category. Another common constraint is the deviation contrast, in which parameters have a sum of zero. One parameter can therefore be dropped as redundant during estimation and found afterwards using minus the sum of the estimated parameters, or by reestimating the model using a different omitted category.

In many cases, the parameterization will be either irrelevant or the indicator contrast will be appropriate. The deviation contrast can be very useful if the categories are purely nominal and there is no obvious choice for a reference category, e.g., "country" or "religion". If there is a large number of missing cases, it can be useful to include them as a separate category and see whether they deviate significantly from the other categories by using the deviation contrast. The difference contrast can be useful for ordered categories. In loglinear models, twoway interaction effects using the difference contrast produce the local odds ratios (Agresti 1990). Stevens (1986) gives examples of using the Helmert contrast and user-defined contrasts in Manova analyses.

A parameterization is created by constructing a contrast matrix C, in which the new parameters, β, are defined as a linear function of the full set of indicator parameters, θ. Given that $A = XC$, where A is an unrestricted indicator matrix and X is a full rank design matrix, then X can be derived by $X = AC'(CC')$. Given this relationship, there are also formulas for generating X directly using a particular contrast (for example, Bock 1975, 300).

Such equations are to define dummy variables based on the deviation, simple, and Helmert contrasts. In the case of a user-defined contrast, the appropriate codings are found by assuming a data vector consisting only of the category values. This means that $A = I$, the identity matrix, and that the codings are given by $C'(CC')$. These codings are then used to create the appropriate dummy variables. Forming dummies based on the indicator contrast is simply a matter of dropping the dummy variable for the reference category. Dummies based on the orthogonal polynomial contrast are generated using `orthpoly`. These

dummies are normalized by dividing them by $\sqrt{m}$, where m is the number of categories, for comparability with SPSS and the SAS version of `desmat`.

In certain situations, it might be necessary to ascertain the codings used in generating the dummy variables, for example, when a user-defined contrast has been specified. Since the dummies have constant values over the categories of the variable that generated them, the codings can be summarized using `table var, contents(mean _x_4 mean _x_5)`. The minimum or maximum may of course be used instead of mean, or in a second run as a check.

The simple contrast and the indicator contrast

The simple contrast and the indicator contrast both use a reference category. What then is the difference between the two? Both produce the same parameters and standard errors for the highest order effect. In the example above, the estimates for `vote.educ.race` and `educ.race.memb` are the same whether the simple or indicator contrast is used, but all other estimates are different.

The difference is that the parameters for the indicator contrast are relative to the reference category whereas the values for the simple contrast are actually relative to the mean value within the categories of the variable. For example, the systolic blood pressure (`systolic` in `systolic.dta`; see, e.g., ANOVA) has the following mean values for each of the four categories of the variable drug: 26.06667, 25.53333, 8.75, and 13.5. In a one way analysis of `systolic` by `drug`, the constant using the indicator contrast with the first category as reference is 26.067. Using the simple contrast, the constant is 18.4625, the mean of the four category means, regardless of the reference category.

Calculating predicted values is a good deal more involved using the simple contrast. Using the simple contrast, `drug` has the following codings:

	b1	b2	b3
drug==1	-.25	-.25	-.25
drug==2	.75	-.25	-.25
drug==3	-.25	.75	-.25
drug==4	-.25	-.25	.75

Given the estimates `_cons`= 18.463, `b1`= −.533, `b2`= −17.317, `b3`= −12.567, the predicted values are calculated as

```
drug==1:  18.463  -.250*-.533  -.250*-17.317  -.250*-12.567  =  26.067
drug==2:  18.463   .750*-.533  -.250*-17.317  -.250*-12.567  =  25.533
drug==3:  18.463  -.250*-.533  +.750*-17.317  -.250*-12.567  =   8.750
drug==4:  18.463  -.250*-.533  -.250*-17.317  +.750*-12.567  =  13.500
```

If the indicator contrast is used, the `b` parameters have the same value but the constant is 26.067, the mean for the first category. The predicted values can be calculated simply as

```
drug==1:  26.067                              =  26.067
drug==2:  26.067  -.533                       =  25.533
drug==3:  26.067         -17.317              =   8.750
drug==4:  26.067                  -12.567     =  13.500
```

The flip side of this is that lower-order effects will depend on the choice of reference category if the indicator contrast is used but not for the simple contrast. Tests for the significance of lower-order terms will also depend on the reference category if the indicator contrast is used. In the sample program above, `tstall` will produce different results for all terms except the highest order, `vote.educ.race` and `educ.race.memb`, if another reference category is used. Using the simple contrast, or one of the other predefined contrasts such as the deviation or difference contrast, will give the same results for these tests.

The desrep command

`desmat` produces a legend associating dummy variables with model terms, but interpreting the results using this would be rather tedious. Instead, a companion program `desrep` can be run after estimation to produce a compact summary of the results with longer labels for the effects. Only the estimates and their standard deviations are reported, together with one asterisk to indicate significance at the 0.05 level and two asterisks to indicate significance at the 0.01 level. The syntax is

`desrep` [`exp`]

Desrep is usually used without any arguments. If `e(b)` and `e(V)` are present, it will produce a summary of the results. If the argument for `desrep` is `exp` it will produce multiplicative parameters, e.g., incident-rate ratios in Poisson regression, and odds ratios in logistic regression. The parameters are transformed into `exp(b)` and their standard errors into `exp(b)*se`, where `b` is the linear estimate and `se` its standard error.

`desmat` adds the characteristics `[varn]` and `[valn]` to each variable name, corresponding to the name of the term and the value of the category, respectively. If `valn` is defined for a variable, this value will be printed with two spaces indented. If not, the variable label, or the variable name if no label is present, is printed with no indentation. `desrep` does not depend on the prior use of `desmat` and can be used after any procedure that produces the `e(b)` and `e(V)` matrices.

The tstall command

`tstall` is for use after estimating a model with a design matrix generated by `desmat` to perform a Wald test on all model terms. The syntax is

`tstall` [`equal`]

The optional argument `equal`, if used, is passed on to `testparm` as an option to test for equality of all parameters in a term. The default is to test whether all parameters in a term are zero.

`desmat` creates global macro variables `term1`, `term2`, and so on, for all terms in the model. `tstall` simply runs `testparm` with each of these global variables. If the global variables have not been defined, `tstall` will do nothing. The global variables can of course also be used separately in `testparm` or related programs.

Note

The Stata version of `desmat` was derived from a SAS macro by the same name that I wrote during the course of my PhD dissertation (Hendrickx 1992, 1994). The SAS version is available at http://baserv.uci.kun.nl/~johnh/desmat/sas/.

References

Agresti, A. 1990. *Categorical Data Analysis*. New York: John Wiley & Sons.

Bock, R. D. 1975. *Multivariate Statistical Methods in Behavioral Research.* New York: McGraw–Hill.

Finn, J. D. 1974. *A General Model for Multivariate Analysis*. New York: Holt, Rinehart and Winston.

Hendrickx, J. 1992. Using SAS macros and PROC IML to create special designs for generalized linear models. *Proceedings of the SAS European Users Group International Conference*. pp. 634–655.

——. 1994. The analysis of religious assortative marriage: An application of design matrix techniques for categorical models. Nijmegen University dissertation.

Knoke, D. and P. J. Burke. 1980. *Loglinear Models*. Beverly Hills: Sage Publications.

Stevens, J. 1986. *Applied Multivariate Statistics for the Social Sciences*. Hillsdale: Lawrence Erlbaum Associates.

dm74 Changing the order of variables in a dataset

Jeroen Weesie, Utrecht University, Netherlands, j.weesie@fss.uu.nl

This insert describes the simple utility command `placevar` for the changing of the order of variables in a dataset. Stata already has some of these commands. `order` changes the order of the variables in the current dataset by moving the variables in a list of variables to the front of the dataset. `move` relocates a variable `varname1` to the position of variable `varname2` and shifts the remaining variables, including `varname2`, to make room. Finally, `aorder` alphabetizes the variables specified in *varlist* and moves them to the front of the dataset. The new command `placevar` generalizes the `move` and `order` commands.

Syntax

`placevar` *varlist* [`, first last after(`*varname*`) before(`*varname*`)`]

Description

`placevar` changes the order of the variables in *varlist* relative to the other variables. The order of the variables in *varlist* is unchanged.

Options

`first` moves the variables in *varlist* to the beginning of the list of all variables.

`last` moves the variables in *varlist* to the end of the list of all variables.

`after(`*varname*`)` moves the variables in *varlist* directly after *varname*. *varname* should not occur in *varlist*.

`before(`*varname*`)` moves the variables in *varlist* directly before *varname*. *varname* should not occur in *varlist*.

Example

We illustrate `placevar` with the automobile data:

```
. use /usr/local/stata/auto
(1978 Automobile Data)
. ds
make      price     mpg       rep78     hdroom    trunk     weight    length
turn      displ     gratio    foreign
```

We can put `price`, `weight`, and `make` at the end of the variables by

```
. placevar price weight make, last
. ds
mpg       rep78     hdroom    trunk     length    turn      displ     gratio
foreign   price     weight    make
```

We can then move `displ` and `gratio` to the beginning with

```
. placevar displ gratio, first
. ds
displ     gratio    mpg       rep78     hdroom    trunk     length    turn
foreign   price     weight    make
```

We can move `hdroom` through `turn` to a position just after `foreign` with

```
. placevar hdroom-turn, after(foreign)
. ds
displ     gratio    mpg       rep78     foreign   hdroom    trunk     length
turn      price     weight    make
```

Finally, we can move `gratio` through `trunk` to the position just before `displ` with

```
. placevar gratio-trunk, before(displ)
. ds
gratio    mpg       rep78     foreign   hdroom    trunk     displ     length
turn      price     weight    make
```

ip18.1 | Update to resample

John R. Gleason, Syracuse University, loesljrg@accucom.net

The command `resample` in Gleason (1997) has been updated in a small but very useful way. The major syntax of `resample` is now

`resample` *varlist* [`if` *exp*] [`in` *range*] [, `names(`*namelist*`)` `retain`]

The new option `retain` is useful for resampling from two or more subsets of observations in a dataset.

`resample` draws a random sample with replacement from one or more variables, and stores the resample as one or more variables named to resemble their parents; see Gleason (1997) for a precise description of the naming rule. By default, an existing variable whose name satisfies the rule is silently overwritten; this simplifies the process of repeatedly resampling a dataset, as in bootstrapping.

So, for example, a command such as

```
. resample x y in 1/20
```

will, on first use, draw a random sample of (`x`, `y`) pairs from observations $1,\ldots,20$, and store it in observations $1,\ldots,20$ of the (newly created) variables `x_` and `y_`. A second use of the command above overwrites observations $1,\ldots,20$ of `x_` and `y_` with a new random resample of (`x`, `y`) pairs from observations $1,\ldots,20$; in both cases, observations $21,\ldots,$`_N` of the variables `x_` and `y_` will have missing values. The command

```
. resample x y in 21/1
```

will place a random sample of (x, y) pairs from observations 21, . . . ,_N in the variables x_ and y_, and replace observations 1, . . . , 20 of x_ and y_ with missing values.

However, it is sometimes useful to resample from two or more subsets of observations and have all the resamples available simultaneously. The retain option makes that possible. In the example just given, the command

```
. resample x y in 21/l, retain
```

would place a random sample of (x, y) pairs from observations 21, . . . ,_N in the variables x_ and y_, without altering the contents of observations 1, . . . , 20 of the variables x_ and y_.

In addition, the syntax has also been expanded slightly so that the command

```
. resample ?
```

will produce a brief reminder of proper usage by issuing the command which resample.

References

Gleason J. R. 1997. ip18: A command for randomly resampling a dataset. *Stata Technical Bulletin* 37: 17–22. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, 77–83.

ip29	Metadata for user-written contributions to the Stata programming language

Christopher F. Baum, Boston College, baum@bc.edu
Nicholas J. Cox, University of Durham, UK, n.j.cox@durham.ac.uk

One of Stata's clear strengths is the extensibility and ease of maintenance resulting from its nonmonolithic structure. Since the program consists of a relatively small executable "kernel" which invokes ado-files to provide much of its functionality, the Stata programming language may be readily extended and maintained by its authors. Since ado-files are plain text, they may be easily transported over networks such as the Internet. This has led to the development of "net-aware" Stata version 6, in which the program can enquire of the Stata Corporation whether it is lacking the latest updates, and download them with the user's permission. Likewise, software associated with inserts from the Stata Technical Bulletin may be "net-installed" from Stata's web site. The knowledge base, or metadata, of official information on Stata's capabilities is updated for each issue of the STB, and "official updates" provide the latest metadata for Stata's search command, so that even if you have not installed a particular STB insert, the metadata accessed by search will inform you of its existence.

A comprehensive collection of metadata cataloging users' contributions to Stata is more difficult to produce. Stata's extensibility has led to the development of a wide variety of additional Stata components by members of the Stata user community, placed in the public domain by their authors and generally posted to Statalist. Statalist is an independently operated listserver for Stata users, described in detail in the Statalist FAQ (see the *References*). In September 1997, a RePEc "series," the Statistical Software Components Archive, was established on the IDEAS server at http://ideas.uqam.ca. RePEc, an acronym for Research Papers in Economics, is a worldwide volunteer effort to provide a framework for the collection and exchange of metadata about documents of interest to economists; be those documents the working papers (preprints) written by individual faculty, the articles published in a scholarly journal, or software components authored by individuals on Statalist. The fundamental scheme is that of a network of "archives," generally associated with institutions, each containing metadata in the form of templates describing individual items, akin to the automated cataloging records of an electronic library collection. The information in these archives is assembled into a single virtual database, and is then accessible to any of a number of RePEc "services" such as IDEAS. A service may provide access to the metadata in any format, and may add value by providing powerful search capabilities, download functionality in various formats, etc. RePEc data are freely available, and a service may not charge for access to them.

The Statistical Software Components Archive (SSC-IDEAS) may be used to provide search and "browsing" capabilities to Stata users' contributions to the Stata programming language, although it is not limited to Stata-language entries. The "net-aware" facilities of Stata's version 6 make it possible for each prolific Stata author in the user community to set up a download site on their web server and link it to Stata's "net from" facility. Users may then point and click to access user-written materials. What is missing from this model? A single clear answer to the question "Given that Stata does not (yet) contain an aardvark command, have Stata users produced one? And from where may I download it?" If we imagine 100 users' sites linked to Stata's "net from" page, the problem becomes apparent.

By creating a "template" for each user contribution and including those templates in the metadata that produces the SSC-IDEAS archive listing, we make each of these users' contributions readily accessible. The IDEAS "search" facility may be used to examine the titles, authors, and "abstracts" of each contribution for particular keywords. The individual pages describing each contribution provide a link to the author's email address, as well as the .ado and .hlp files themselves. If an author has

placed materials in a Stata download site, the SSC-IDEAS entry may refer to that author's site as the source of the material, so that updates will be automatically propagated. Thus, we consider that the primary value added of the SSC-IDEAS archive is its assemblage of metadata about users' contributions in a single, searchable information base. The question posed above regarding the `aardvark` command may be answered in seconds via a trip to SSC-IDEAS from any web browser.

It should be noted that materials are included in the SSC-IDEAS archive based on the maintainer's evaluation of their completeness, without any quality control or warranty of usefulness. If a Statalist posting contains a complete ado-file and help file in standard Stata format, with adequate contact information for the author(s), it will be included in the archive as long as its name does not conflict with an existing entry. Authors wanting to include a module in the archive without posting–due to, for example, its length–or wanting an item included via its URL on a local site, rather than placing it in the archive–should contact the maintainer directly (`mailto:baum@bc.edu`). Authors retain ownership of their archived modules, and may request that they be removed or updated in a timely fashion.

Large utilities archlist, archtype, and archcopy

The "net-aware" features of Stata's version 6 imply that most version 6 users will be using `net install` to acquire contributions from the SSC-IDEAS archive. Consequently, the materials in the archive are described in two sets of metadata: first, the RePEc "templates" from which the SSC-IDEAS pages are automatically produced; and second, `stata.toc` and `.pkg` files which permit these materials to be net installed, are generated automatically from the templates. This duality implies that much of the information accessible to a user of SSC-IDEAS is also available from within net-aware Stata.

The `archlist` command may be used in two forms. The command alone produces a current list of all Stata net-installable packages in the SSC-IDEAS archive, with short descriptors. If the command is followed by a single character (`a-z,_`) the listing for packages with names starting with that character is produced.

The `archtype` command permits the user to access the contents of a single file on SSC-IDEAS, specified by filename.

The `archcopy` command permits the user to copy a single file from SSC-IDEAS to the STBPLUS directory on their local machine. It should be noted that `net install` is the preferable mechanism to properly install all components of a package, and permits `ado uninstall` at a later date.

The `archtype` and `archcopy` commands are essentially wrappers for the corresponding Stata commands `type` and `copy`, and acquire virtue largely by saving the user from remembering (or looking up), and then typing, the majority of the address of each file in SSC-IDEAS, which for `foo.ado` would be

```
http://fmwww.bc.edu/repec/bocode/f/foo.ado
```

Therefore, `archtype` and `archcopy` inherit the characteristics of Stata's `type` and `copy`; in particular, `archcopy` will not create a directory or folder of STBPLUS if it does not previously exist, unlike `net install`.

It should also be noted that after the command `archlist` *letter*, which produces a listing of modules in the SSC-IDEAS archive under that letter, the command `net install` may be given to install any of those modules. This sequence of commands obviates the need to use `net from` *wherever* to navigate to the SSC-IDEAS link, whether by URL or using the help system on a graphical version of Stata.

Syntax

`archlist` [`using` *filename*] [`, replace`]

`archlist` *letter*

`archtype` *filename*

`archcopy` *filename* [`,` *copy_options*]

Description

These commands require a net-aware variant of Stata 6.0.

`archlist` lists packages downloadable from the SSC-IDEAS archive. If no letter is specified, it also logs what it displays, by default in `ssc-ideas.lst`. Therefore, it may not be run `quietly` without suppressing its useful output. Any logging previously started will be suspended while it is operating. If a single letter (including _) is specified, `archlist` lists packages whose names begin with that letter, but only on the screen. Further `net` commands will reference that directory of the SSC-IDEAS link.

`archtype` *filename* types *filename* from the SSC-IDEAS archive. This is appropriate for individual `.ado` or `.hlp` files.

`archcopy` *filename* copies *filename* from the SSC-IDEAS archive to the appropriate directory or folder within STBPLUS, determined automatically. This is appropriate for individual `.ado` or `.hlp` files.

In the case of `archtype` and `archcopy`, the filename should be typed as if you were in the same directory or folder, for example as `foo.ado` or `foo.hlp`. The full path should not be given.

Options

`replace` specifies that *filename* is to be overwritten.

copy_options are options of `copy`. See help for `copy`.

Examples

The somewhat lengthy output of these commands is suppressed here to save space.

```
. archlist using ssc.txt, replace
. archlist a
. archtype whitetst.hlp
. archcopy whitetst.ado
. archcopy whitetst.hlp
```

Acknowledgments

Helpful advice was received from Bill Gould, Thomas Krichel, Jens Lauritsen and Vince Wiggins.

References

SSC-IDEAS archive, at URL `http://ideas.uqam.ca/ideas/data/bocbocode.html`

Statalist FAQ, at URL `http://www.stata.com/support/statalist/faq/`

sbe31 | Exact confidence intervals for odds ratios from case–control studies

William D. Dupont, Vanderbilt University, bill.dupont@vanderbilt.edu
Dale Plummer, Vanderbilt University, dale.plummer@mcmail.vanderbilt.edu

Syntax

`exactcc` *var*$_{\text{case}}$ *var*$_{\text{exposed}}$ [*weight*] [`if` *exp*] [`in` *range*] [`, level(#) exact tb woolf by(`*varname*`)`

`nocrude bd pool nohom estandard istandard standard(`*varname*`) binomial(`*varname*`) ]`

`exactcci` #$_{\text{a}}$ #$_{\text{b}}$ #$_{\text{c}}$ #$_{\text{d}}$ [`, level(#) exact tb woolf ]`

`fweights` are allowed.

Description

These commands and their options are identical to Stata's `cc` and `cci` (see [R] **epitab**) except that additional output is provided. The default output includes Cornfield's confidence interval for the odds ratio calculated both with and without a continuity correction. These intervals are labeled "adjusted" and "unadjusted," respectively. We also provide Yates' continuity corrected chi-squared test of the null hypothesis that the odds ratio equals 1. When the `exact` option is given, the exact confidence interval for the odds ratio is also derived, as well as twice the one-sided Fisher's exact p-value. Analogous confidence intervals for the attributable or preventive fractions are provided.

Methods

Let m_1 and m_0 be the number of cases and controls in a case–control study, n_1 and n_0 be the number of subjects who are, or are not, exposed to the risk factor of interest, a be the number of exposed cases, and ψ be the true odds ratio for exposure in cases compared to controls. Then Clayton and Hills (1993, 171) give the probability of observing a exposed cases given m_1, m_0, n_1 and n_0, which is

$$f\left(a|\psi\right) = \frac{K\psi^a}{a!\left(n_1-a\right)!\left(m_1-a\right)!\left(n_0-m_1+a\right)!} \tag{1}$$

In equation (1), K is the constant such that the sum of $f(a)$ over all possible values of a equals one. Let $P_L(\psi) = \sum_{i \leq a} f(i|\psi)$ and $P_U(\psi) = \sum_{i \geq a} f(i|\psi)$. Then the exact $100(1-\alpha)\%$ confidence interval for ψ is (ψ_L, ψ_U) where the limits of the confidence interval are chosen so that $P_L(\psi_U) = P_U(\psi_L) = \alpha/2$ (see Rothman and Greenland 1998, 189).

Cornfield provides an estimate of (ψ_L, ψ_U) that is based on a continuity corrected normal approximation (see Breslow and Day 1980, 133). Stata provides an analogous estimate that uses an uncorrected normal approximation (Gould 1999). We refer to these estimates as Cornfield's adjusted and unadjusted estimates, respectively. We estimate ψ_L as follows. Let ψ_0 and ψ_1 be the lower bound of the confidence interval using Cornfield's unadjusted and adjusted estimates, respectively. Let $p_0 = P_U(\psi_0)$ and $p_1 = P_U(\psi_1)$. We use the secant method to derive ψ_L iteratively (Pozrikidis 1998, 208). That is, we let

$$\psi_{i+1} = \psi_i - (p_i - \alpha/2)\left(\frac{\psi_i - \psi_{i-1}}{p_i - p_{i-1}}\right) \tag{2}$$

$$p_{i+1} = P_U\left(\psi_{i+1}\right) \tag{3}$$

and then solve equations (2) and (3) for $i = 1, 2, \ldots, 100$. These equations converge to ψ_L and $\alpha/2$, respectively. We stop iterating and set $\psi_L = \psi_{i+1}$ when $|\psi_{i+1} - \psi_i| < 0.005\psi_{i+1}$; `exactcci` abandons the iteration and prints an error message if convergence has not been achieved within 100 iterations. However, we have yet to discover a 2×2 table where this happens. The estimates ψ_0 and ψ_1 are themselves calculated iteratively by programs that can return missing or nonpositive values. If this happens we set ψ_0 equal to the Woolf's estimate of the lower bound of the confidence interval and $\psi_1 = 0.75\psi_0$. We estimate ψ_U in an analogous fashion. The formula for Yates' continuity corrected χ^2 statistic is given in many introductory texts (see, for example, Armitage and Berry 1994, 137).

Examples

We illustrate the use of `exactcci` with an example from Table 17.3 of Clayton and Hills (1993, 172).

```
. exactcci 3 1 1 19, level(90) exact
                                                        Proportion
                 |   Exposed   Unexposed  |     Total     Exposed
-----------------+------------------------+----------------------
           Cases |         3           1  |         4      0.7500
        Controls |         1          19  |        20      0.0500
-----------------+------------------------+----------------------
           Total |         4          20  |        24      0.1667
                 |                        |
                 |      Point estimate    |  [90% Conf. Interval]
                 |------------------------+----------------------
                 |                        | Cornfield's limits
      Odds ratio |               57       |   2.63957           .  Adjusted
                 |                        |  5.378964           .  Unadjusted
                 |                        | Exact limits
                 |                        |   2.44024    1534.137
                 |                        | Cornfield's limits
 Attr. frac. ex. |         .9824561       |  .6211504           .  Adjusted
                 |                        |  .8140906           .  Unadjusted
                 |                        | Exact limits
                 |                        |  .5902042    .9993482
 Attr. frac. pop |         .7368421       |
                 +-------------------------------------------------
                              chi2(1) =    11.76  Pr>chi2 = 0.0006
                Yates' adjusted chi2(1) =     7.26  Pr>chi2 = 0.0071
                                  1-sided Fisher's exact P = 0.0076
                                  2-sided Fisher's exact P = 0.0076
                         2 times 1-sided Fisher's exact P = 0.0152
```

These results give exact confidence intervals that are in complete agreement with those of Clayton and Hills. The next example is from Gould (1999):

(*Continued on next page*)

```
. exactcci 11 3 106 223,  exact
                                                        Proportion
                 |   Exposed   Unexposed  |     Total     Exposed
-----------------+------------------------+----------------------
           Cases |        11           3  |        14      0.7857
        Controls |       106         223  |       329      0.3222
-----------------+------------------------+----------------------
           Total |       117         226  |       343      0.3411
                 |                        |
                 |      Point estimate    |  [95% Conf. Interval]
                 |------------------------+----------------------
                 |                        | Cornfield's limits
      Odds ratio |         7.713836       |  1.942664    35.61195  Adjusted
                 |                        |  2.260934    26.20341  Unadjusted
                 |                        | Exact limits
                 |                        |  1.970048     43.6699
                 |                        | Cornfield's limits
 Attr. frac. ex. |         .8703628       |   .485243    .9719195  Adjusted
                 |                        |   .557705     .961837  Unadjusted
                 |                        | Exact limits
                 |                        |  .4923982    .9771009
 Attr. frac. pop |         .6838565       |
                 +-------------------------------------------------
                           chi2(1) =    12.84  Pr>chi2 = 0.0003
             Yates' adjusted chi2(1) =    10.86  Pr>chi2 = 0.0010
                              1-sided Fisher's exact P = 0.0007
                              2-sided Fisher's exact P = 0.0007
                     2 times 1-sided Fisher's exact P = 0.0014
```

The adjusted and unadjusted Cornfield's limits agree with those published by Gould to three significant figures. Also, the adjusted limits are closer to the exact limits than are the unadjusted limits. Our final example comes from Table 4.6 of Armitage and Berry (1994, 138).

```
. exactcci 21 16 1 4,  exact
                                                        Proportion
                 |   Exposed   Unexposed  |     Total     Exposed
-----------------+------------------------+----------------------
           Cases |        21          16  |        37      0.5676
        Controls |         1           4  |         5      0.2000
-----------------+------------------------+----------------------
           Total |        22          20  |        42      0.5238
                 |                        |
                 |      Point estimate    |  [95% Conf. Interval]
                 |------------------------+----------------------
                 |                        | Cornfield's limits
      Odds ratio |             5.25       |   .463098           .  Adjusted
                 |                        |  .6924067           .  Unadjusted
                 |                        | Exact limits
                 |                        |   .444037    270.5581
                 |                        | Cornfield's limits
 Attr. frac. ex. |         .8095238       |  -1.15937           .  Adjusted
                 |                        | -.4442378           .  Unadjusted
                 |                        | Exact limits
                 |                        | -1.252065    .9963039
 Attr. frac. pop |         .4594595       |
                 +-------------------------------------------------
                           chi2(1) =     2.39  Pr>chi2 = 0.1224
             Yates' adjusted chi2(1) =     1.14  Pr>chi2 = 0.2857
                              1-sided Fisher's exact P = 0.1435
                              2-sided Fisher's exact P = 0.1745
                     2 times 1-sided Fisher's exact P = 0.2871
```

The values of Yates' continuity corrected χ^2 statistic and p-value agree with those published by Armitage and Berry (1994, 140), as do the one- and two-sided Fisher's exact p-value. Note that Yates' p-value agrees with twice the one-sided Fisher's exact p-value to two significant figures even though the minimum expected cell size is only 2.4.

Remarks

Many epidemiologists, including Rothman and Greenland (1998, 189) and Breslow and Day (1980, 128) define a 95% confidence interval for a parameter to be the range of values that cannot be rejected at the 5% significance level. A more traditional definition is that it is an interval that spans the true value of the parameter with probability 0.95. These definitions

are equivalent for a normally distributed statistic whose mean and variance are unrelated. For a discrete statistic whose mean and variance are interrelated, however, these definitions can lead to different intervals. Let us refer to these definitions as the nonrejection and coverage definitions, respectively. The nonrejection 95% confidence interval always spans the true value of the parameter with at least 95% but possibly greater certainty (see Rothman and Greenland 1998, 189, 221–222). Exact confidence intervals are defined using the nonrejection definition. In all contingency tables that we have examined to date, the exact interval for the odds ratio is better approximated by Cornfield's adjusted confidence interval than by his unadjusted interval. This does not, however, contradict the observation of Gould (1999) that the coverage probability of the adjusted interval can exceed 95%. This is because the exact interval itself can have this over-coverage property. Statisticians who wish to approximate the exact interval will prefer to use Cornfield's adjusted interval. Those who seek an interval that comes as close as possible to spanning the true odds ratio with 95% certainty may well prefer to use his unadjusted interval.

Controversy has surrounded the use of Yates' continuity correction for decades. Grizzle (1967) and Camilli and Hopkins (1978) performed simulation studies that indicated that the Type I error probability for the corrected χ^2 statistic was less than the nominal value. They and Haviland (1990) argue that the continuity correction should not be used for this reason. Rebuttals to these papers have been published by Mantel and Greenhouse (1968) and Mantel (1990). Tocher (1950) showed that uniformly most powerful unbiased tests are obtained by conditioning on the observed marginal totals of a contingency table. The simulation studies of Grizzle, and Camilli and Hopkins are not conditioned on the observed marginal total of exposed case and control subjects. Many statisticians accept the Conditionality Principle, which states that if an ancillary statistic exists whose distribution is unaffected by the parameter of interest, then inferences about this parameter should be conditioned on this ancillary statistic (Cox and Hinkley 1974, 38). In case–control studies the marginal total of exposed cases and controls is not a true ancillary statistic for the odds ratio, but it is similar to one in the sense that knowing the total number of exposed subjects tells us nothing about the value of the odds ratio. This fact provides a justification for using Fisher's exact test in case–control studies even though the total number of exposed subjects is not fixed by the experimental design. Rothman and Greenland (1998, 251) state that "Although mildly controversial, the practice [of conditioning on the number of exposed subjects] is virtually universal in epidemiologic statistics." Rothman and Greenland (1998, 185), Breslow and Day (1980, 128), Mantell (1990) and others suggest doubling the one-sided Fisher's exact p-value for two-sided tests. Yates' continuity correction is used to approximate the hypergeometic distribution of Fisher's exact test by a normal distribution. Yates' p-value provides an excellent approximation to twice the one-tailed Fisher's p-value over a wide range of contingency tables; for this purpose it is far more accurate than the p-value from the uncorrected statistic (Dupont 1986). Note that doubling the one-sided exact p-value has the desirable property that this statistic is less than 0.05 if and only if the exact 95% confidence interval excludes one. The other p-values from `exactcci` do not share this property.

Our intent in the preceding paragraphs is not to rehash old arguments but to point out that knowledgeable statisticians can and do disagree about the use of continuity corrections in calculating confidence intervals for the odds ratio or p-values for testing null hypotheses. We believe that Stata, and its community of statisticians, will be strengthened by allowing STB readers to decide for themselves whether or not to use these corrections.

Acknowledgment

`exactcci` makes extensive use of the code from Stata's `cci` program. The only difference between `exactcc` and `cc` is that the former program calls `exactcci` instead of `cci`.

References

Armitage, P. and G. Berry. 1994. *Statistical Methods in Medical Research*. 3d ed. Oxford: Blackwell.

Breslow, N. E. and N. E. Day. 1980. *Statistical Methods in Cancer Research*, vol. 1, The Analysis of Case-Control Studies. Lyon, France: IARC Scientific Publications.

Camilli, G. and K. D. Hopkins. 1978. Applicability of chi-square to 2 x 2 contingency tables with small expected cell frequencies. *Psychological Bulletin* 85: 163–167.

Clayton, D. and M. Hills. 1993. *Statistical Models in Epidemiology*. Oxford: Oxford University Press.

Cox, D. R. and D. V. Hinkley. 1974. *Theoretical Statistics*. London: Chapman and Hall.

Dupont, W. D. 1986. Sensitivity of Fisher's exact test to minor perturbations in 2 x 2 contingency tables. *Statistics in Medicine* 5: 629–635.

Gould, W. 1999. Why do Stata's `cc` and `cci` commands report different confidence intervals than Epi Info? *Stata Frequently asked questions*. http://www.stata.com/support/faqs/

Grizzle, J. E. 1967. Continuity correction in the χ^2 test for 2x2 tables. *The American Statistician* 21: 28–32.

Haviland, M. G. 1990. Yates' correction for continuity and the analysis of 2 x 2 contingency tables. *Statistics in Medicine* 9: 363–367.

Mantel, N. 1990. Comment. *Statistics in Medicine* 9: 369–370.

Mantel, N. and S. W. Greenhouse. 1968. What is the continuity correction? *The American Statistician* 22: 27–30.

Pozrikidis, C. 1998. *Numerical Computation in Science and Engineering*. New York: Oxford University Press.

Rothman, K. J. and S. Greenland. 1998. *Modern Epidemiology*. Philadelphia: Lippincott–Raven.

Tocher, K. D. 1950. Extension of the Neyman-Pearson theory of tests to discontinuous variates. *Biometrika* 37: 130–144.

sg119 | Improved confidence intervals for binomial proportions

John R. Gleason, Syracuse University, loesljrg@accucom.net

Stata's `ci` and `cii` commands provide so-called *exact* confidence intervals for binomial proportions, i.e., for the parameter p in the binomial distribution $B(n,p)$. `ci` and `cii` compute Clopper–Pearson (1934) intervals which are "exact" in that their actual coverage probability is never less than the nominal level, whatever the true value of p. But it is widely known that Clopper–Pearson (CP) intervals are almost everywhere conservative; for most values of p, the actual coverage probability is well above the nominal level. More importantly, from a practical view, "exact" intervals tend to be wide. These facts have spawned a variety of alternative binomial confidence intervals. This insert presents two new commands `propci` and `propcii` that implement, in addition to the CP and Wald (see below) intervals, three alternative confidence intervals each of which has practical advantages over the CP and Wald intervals.

Overview

The best-known interval for p is of course $\widehat{p} \pm z_{\alpha/2}\sqrt{\widehat{p}(1-\widehat{p})/n}$, where $\widehat{p}$ is the sample proportion and $z_{\alpha/2}$ is the value of a standard normal distribution having area $\alpha/2$ to its right. This is known as the *Wald* interval for p, in deference to its connection with the Wald test of a hypothesis about p. But the Wald interval has poor coverage properties even for large n, as has been demonstrated repeatedly. (An excellent source is Vollset 1993, who examined the performances of the Wald and CP intervals, along with those of ten competing intervals.) In particular, at certain isolated values of p, the actual coverage of the Wald interval plunges well below the nominal confidence level. A graph of coverage probability versus p will consequently present deep, downward spikes. For example, even at $n = 1000$ the actual coverage probability of the interval $\widehat{p} \pm 1.96\sqrt{\widehat{p}(1-\widehat{p})/n}$ can drop to near 0.80.

Many non-CP confidence intervals attempt to dampen these coverage spikes, typically with only partial success; see Vollset (1993). A common strategy is to replace the sample proportion $\widehat{p} = x/n$ with $(x+b)/(n+2b)$ for some $b > 0$, and then apply the Wald formula. This biases the center of the resulting interval toward $1/2$ and can greatly improve its worst-case coverage probability. Agresti and Coull (1998) proposed a particularly simple and appealing variant of this idea for 95% confidence intervals. Set $b = 2$, $\tilde{p} = (x+2)/(n+4)$, and use the interval $\tilde{p} \pm z_{0.025}\sqrt{\tilde{p}(1-\tilde{p})/(n+4)}$. (For confidence other than 95%, one would substitute the appropriate $z_{\alpha/2}$, presumably.) The estimator $\tilde{p}$ can be traced to Wilson (1927), so we refer to the associated interval as the Wilson interval. While setting $b = 2$ does greatly improve the minimum coverage of the Wald interval, there is a flaw; except for $\widehat{p}$ rather near $1/2$, the Wilson interval can be even wider than the CP interval.

On practical grounds, this creates an enigma; given that the CP interval is both "exact" and easily computed (say, by `ci` and `cii`), why use an approximate interval even wider than the already conservative CP interval? It turns out that $b = 2$ is simply too large a bias except when $\widehat{p}$ is near $1/2$; allowing b to decrease as $\widehat{p}$ departs $1/2$ solves the problem. A simple, effective choice (Gleason 1999a) is $b^* = 2.64(\widehat{p}(1-\widehat{p}))^{0.2}$; so, let $p^* = (x+b^*)/(n+2b^*)$ and refer to $p^* \pm z_{\alpha/2}\sqrt{p^*(1-p^*)/(n+2b^*)}$ as the enhanced-Wald confidence interval.

Another approach to improved binomial confidence intervals is the classic arcsine transformation. Vollset (1993) showed that while the arcsine interval improves upon the Wald interval in some respects, it too suffers deep coverage spikes even for large n. But biasing $\widehat{p}$ before transforming largely removes this deficiency (Gleason 1999a). Specifically, for $0 < x < n$, compute $b^{**} = \max(x, n-x)/(4n)$, $p^{**} = (x+b^{**})/(n+2b^{**})$, and $T = \arcsin(\sqrt{p^{**}})$. Then calculate $T \pm 0.5z_{\alpha/2}/\sqrt{n}$, and back-transform the resulting limits. (The cases $x = 0$ and $x = n$ are handled in the next paragraph.) Let us call this the enhanced-arcsine interval.

Finally, there is the matter of endpoint adjustment which, as Vollset (1993) showed, can greatly improve performance. If $x \in \{0, 1, n-1, n\}$, one or both of the CP limits can be easily computed and should be used in preference to the limits of any approximate interval (Blyth 1986). Precisely, at $x = n$ the CP interval is $[(\alpha/2)^{1/n}, 1]$, and at $x = n-1$ the CP upper limit is $(1-\alpha/2)^{1/n}$; similar expressions hold for $x = 0$ and $x = 1$. While there is little reason to quibble with these limits, using them in connection with a non-CP method requires some care to ensure that the end result satisfies an eminently sensible condition: Confidence limits for p should be nondecreasing in x. That is, upper and lower limits for any given $x < n$ should be no greater than those for $x+1$. Ordinarily this is not a problem until one mixes limits from two different methods; precisely what endpoint adjustment requires.

These issues do complicate the topic of confidence intervals for p, but at least there is potential for practical gain. The following conclusions can be drawn about endpoint adjusted binomial confidence intervals:

- The Wald interval has poor worst-case coverage probability but it is very narrow when x is near 0 or n.
- The Wilson 95% confidence interval (Agresti and Coull 1998) has minimum coverage much closer to 95% than does the Wald interval. However, Wilson intervals are often wider than, but rarely much narrower than the CP interval. In addition, Wilson 90% intervals have mediocre minimum coverage probability.
- The enhanced-arcsine and enhanced-Wald intervals have much improved minimum coverage probability though both can be slightly liberal, usually for p near $1/2$. But for p beyond about 0.9 or 0.1, both methods tend to give minimum coverage above the nominal level *and* intervals narrower than the CP interval.
- In fact, the expected intervals from the enhanced-arcsine and enhanced-Wald methods are (almost) always narrower than the expected CP interval. The advantage in expected length is often near 5% but can reach more than 15%. This translates to a 10% to 35% increase in effective sample size relative to the CP interval and, often, an even greater increase relative to the Wilson interval.
- The enhanced Wald interval generally has coverage probability slightly better than, and expected interval length slightly worse than the enhanced arcsine interval.

See Gleason (1999a) for additional detail on these confidence interval methods, and comparisons among them.

New commands for binomial confidence intervals

The commands `propci` and `propcii` compute confidence intervals for the binomial parameter p using the CP "exact" method, or any of four endpoint-adjusted approximate methods; Wald, enhanced Wald, Wilson, or enhanced arcsine. Their design mimics that of the commands `oddsrci` and `oddsrcii` (Gleason 1999b). `propci` has syntax

```
propci [weight] [if exp] [in range] , cond(cond) [ level(#) none all arcsin ewald
    exact wald wilson ]

propcii n x [ , level(#) none all arcsin ewald exact wald wilson ]
```

`fweights` are allowed in `propci`.

`propcii` is the immediate form of `propci`. The arguments *n* and *x* are the sample size and count that define the proportion $\widehat{p} = x/n$. The options are identical to their counterparts for `propci`; as with `ci` and `cii`, `propci` calls `propcii` to display confidence intervals.

Each of the commands also has an alternative syntax that displays a quick reminder of usage:

```
propci ?

propcii ?
```

Options

`cond()` is required. *cond* is any Boolean (true–false) condition whose truth value defines the proportion of interest. *cond* can be an arbitrarily complex expression and it may contain embedded double-quote (`"`) characters; the only requirement is that it evaluate to true or false. Internally, `propci` creates a temporary variable with a command resembling '`gen byte` *Var* `=` `(`*cond*`)`', and then uses `tabulate` to count the various values of *Var*.

`level` is the desired confidence level specified either as a proportion or as a percentage. The default level is the current setting of the system macro `S_level`.

The remaining options choose confidence interval methods for the proportion $\widehat{p}$ defined by *cond*. Any combination of methods can be specified; `all` selects each of the five available methods, and `none` chooses none of them (useful to see just $\widehat{p}$). `exact` is the default method (for the sake of consistency with the `ci` command), but `ewald` is almost certainly a better choice.

Example

To illustrate, consider the dataset `cancer.dta` supplied with Stata 6.0:

```
. use cancer
(Patient Survival in Drug Trial)
```

```
. describe
Contains data from /usr/local/stata/cancer.dta
  obs:            48                          Patient Survival in Drug Trial
 vars:             4
 size:           576 (99.9% of memory free)
-------------------------------------------------------------------------------
   1. studytim  int    %8.0g                 Months to death or end of exp.
   2. died      int    %8.0g                 1 if patient died
   3. drug      int    %8.0g                 Drug type (1=placebo)
   4. age       int    %8.0g                 Patient´s age at start of exp.
-------------------------------------------------------------------------------
Sorted by:
```

Suppose we are interested in the proportion of patients who were at least 65 years old at the outset and who died during the study, considering only patients who received one of the two active drugs. For that proportion, we'd like a 99% confidence interval from each available method:

```
. propci if drug > 1, cond(died & (age >= 65)) all lev(.99)
Select cases: if drug > 1
  Condition : died & (age >= 65)
  Condition |      Freq.     Percent        Cum.
------------+-----------------------------------
      False |         26       92.86       92.86
       True |          2        7.14      100.00
------------+-----------------------------------
      Total |         28      100.00
Exact (Clopper-Pearson) 99% CI: [0.0038, 0.2911]
   Wald (normal theory) 99% CI: [0.0002, 0.1968]
  EWald (Enhanced Wald) 99% CI: [0.0002, 0.2605]
 Wilson (Agresti-Coull) 99% CI: [0.0002, 0.2756]
 Arcsin transform-based 99% CI: [0.0016, 0.2531]
```

Notice that there are appreciable differences in the widths of the various intervals. Given the knowledge that $n = 28$ and $x = 2$, the command `propcii 28 2, arc lev(97.5)` computes a 97.5% confidence interval for the same proportion using the enhanced arcsine method.

Saved Results

`propcii` saves in `r()`, whether or not called from `propci`. Thus, following the call to `propci` above:

`r(lb_Arcsi)`	lower limit of arcsin interval
`r(ub_Arcsi)`	upper limit of arcsin interval
`r(lb_Wilso)`	lower limit of Wilson interval
`r(ub_Wilso)`	upper limit of Wilson interval
`r(lb_EWald)`	lower limit of extended-Wald interval
`r(ub_EWald)`	upper limit of extended-Wald interval
`r(lb_Wald)`	lower limit of Wald interval
`r(ub_Wald)`	lower limit of Wald interval
`r(lb_Exact)`	lower limit of exact interval
`r(ub_Exact)`	upper limit of exact interval
`r(level)`	confidence level
`r(p_hat)`	value of $\widehat{p}$
`r(N)`	sample size

References

Agresti, A. and B. A. Coull. 1998. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician* 52: 119–126.

Blyth, C. R. 1986. Approximate binomial confidence limits. *Journal of the American Statistical Association* 81: 843–855.

Clopper, C. J. and E. S. Pearson. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26: 404–413.

Gleason, J. R. 1999a. Better approximations are even better for interval estimation of binomial proportions. Submitted to *The American Statistician*.

——. 1999b. sbe30: Improved confidence intervals for odds ratios. *Stata Technical Bulletin* 51: 24–27.

Vollset, S. E. 1993. Confidence intervals for a binomial proportion. *Statistics in Medicine* 12: 809–824.

Wilson, E. B. 1927. Probable inference, the law of succession, and statistical inference. *Journal of The American Statistical Association* 22: 209–212.

sg120	Receiver Operating Characteristic (ROC) analysis

Mario Cleves, Stata Corporation, mcleves@stata.com

Syntax

```
roctab refvar classvar [weight] [if exp] [in range] [, bamber hanley detail lorenz table binomial
    level(#) norefline nograph graph_options ]

rocfit refvar classvar [weight] [if exp] [in range] [, level(#) nolog maximize_options ]

rocplot [, confband level(#) norefline graph_options ]

roccomp refvar classvar [classvars] [weight] [if exp] [in range] [, by(varname) binormal level(#)
    test(matname) norefline separate nograph graph_options ]
```

fweights are allowed, see [U] **14.1.6 weight**.

Description

The above commands are used to perform Receiver Operating Characteristic (ROC) analyses with rating and discrete classification data.

The two variables *refvar* and *classvar* must be numeric. The reference variable indicates the true state of the observation such as diseased and nondiseased or normal and abnormal, and must be coded 0 and 1. The rating or outcome of the diagnostic test or test modality is recorded in *classvar*, which must be at least ordinal, with higher values indicating higher risk.

roctab is used to perform nonparametric ROC analyses. By default, roctab plots the ROC curve and calculates the area under the curve. Optionally, roctab can display the data in tabular form and can also produce Lorenz-like plots.

rocfit estimates maximum-likelihood ROC models assuming a binormal distribution of the latent variable.

rocplot may be used after rocfit to plot the fitted ROC curve and simultaneous confidence bands.

roccomp tests the equality of two or more ROC areas obtained from applying two or more test modalities to the same sample or to independent samples. roccomp expects the data to be in wide form when comparing areas estimated from the same sample, and in long form for areas estimated from independent samples.

Options for roctab

bamber specifies that the standard error for the area under the ROC curve be calculated using the method suggested by Bamber (1975). Otherwise, standard errors are obtained as suggested by DeLong, DeLong and Clarke-Pearson (1988).

hanley specifies that the standard error for the area under the ROC curve be calculated using the method suggested by Hanley and McNeil (1982). Otherwise, standard errors are obtained as suggested by DeLong, DeLong and Clarke-Pearson (1988).

detail outputs a table displaying the sensitivity, specificity, percent of subjects correctly classified, and two likelihood-ratios for each possible cut-point of classvar.

lorenz specifies that a Lorenz-like curve be produced, and Gini and Pietra indices reported.

table outputs a $2 \times k$ contingency table displaying the raw data.

binomial specifies that exact binomial confidence intervals be calculated.

level(#) specifies the confidence level, in percent, for the confidence intervals; see [R] **level**.

norefline suppresses the plotting of the 45 degree reference line from the graphical output of the ROC curve.

nograph suppresses graphical output of the ROC curve.

graph_options are any of the options allowed with graph, twoway.

Options for rocfit and rocplot

level(#) specifies the confidence level, in percent, for the confidence intervals and confidence bands; see [R] **level**.

nolog prevents rocfit from showing the iteration log.

maximize_options controls the maximization process; see [R] **maximize**. You should never have to specify them.

`confband` specifies that simultaneous confidence bands be plotted around the ROC curve.

`norefline` suppresses the plotting of the 45 degree reference line from the graphical output of the ROC curve.

graph_options are any of the options allowed with `graph, twoway`.

Options for roccomp

`by(`*varname*`)` is required when comparing independent ROC areas. The `by()` variable identifies the groups to be compared.

`binormal` specifies that the areas under the ROC curves to be compared should be estimated using the binormal distribution assumption. By default, areas to be compared are computed using the trapezoidal rule.

`level(#)` specifies the confidence level, in percent, for the confidence intervals; see [R] **level**.

`test(`*matname*`)` specifies the contrast matrix to be used when comparing ROC areas. By default, the null hypothesis that all areas are equal is tested.

`norefline` suppresses the plotting of the 45 degree reference line from the graphical output of the ROC curve.

`separate` is meaningful only with `roccomp`; it says that each ROC curve should be placed on its own graph rather than one curve on top of the other.

`nograph` suppresses graphical output of the ROC curve.

graph_options are any of the options allowed with `graph, twoway`.

Remarks

Receiver Operating Characteristic (ROC) analysis is used to quantify the accuracy of diagnostic tests or other evaluation modality used to discriminate between two states or conditions. For ease of presentation, we will refer to these two states as normal and abnormal, and to the discriminatory test as a diagnostic test. The discriminatory accuracy of a diagnostic test is measured by its ability to correctly classify known normal and abnormal subjects. The analysis uses the ROC curve, a graph of the sensitivity versus 1 − specificity of the diagnostic test. The sensitivity is the fraction of positive cases that are correctly classified by the diagnostic test, while the specificity is the fraction of negative cases that are correctly classified. Thus, the sensitivity is the true-positive rate, and the specificity the true-negative rate.

The global performance of a diagnostic test is commonly summarized by the area under the ROC curve. This area can be interpreted as the probability that the result of a diagnostic test of a randomly selected abnormal subject will be greater than the result of the same diagnostic test from a randomly selected normal subject. The greater the area under the ROC curve, the better the global performance of the diagnostic test.

Both nonparametric methods and parametric (semi-parametric) methods have been suggested for generating the ROC curve and calculating its area. In the following sections we present these approaches, and in the last section present tests for comparing areas under ROC curves.

The sections below are

Nonparametric ROC *curves*
Parametric ROC *curves*
Lorenz-like curves
Comparing areas under the ROC *curve*

Nonparametric ROC curves

The points on the nonparametric ROC curve are generated by using each possible outcome of the diagnostic test as a classification cut-point, and computing the corresponding sensitivity and 1 − specificity. These points are then connected by straight lines, and the area under the resulting ROC curve computed using the trapezoidal rule.

Example

Hanley and McNeil (1982) presented data from a study in which a reviewer was asked to classify, using a nine point scale, a random sample of 109 tomographic images from patients with neurological problems. The rating scale was as follows: 1–definitely normal, 2–probably normal, 3–questionable, 4–probably abnormal, and 5–definitely abnormal. The true disease status was normal for 58 of the patients and abnormal for the remaining 51 patients.

Here we list nine of the 109 observations.

```
. list disease rating in 1/9
```

```
     disease     rating
1.         1          5
2.         1          4
3.         0          1
4.         1          5
5.         0          1
6.         0          1
7.         1          5
8.         1          5
9.         1          4
```

For each observation, `disease` identifies the true disease status of the subject (0 = normal, 1 = abnormal), and `rating` contains the classification value assigned by the reviewer.

We can use `roctab` to plot the nonparametric ROC curve. By specifying also the `table` option we obtain a contingency table summarizing our data.

```
. roctab disease rating, table
```

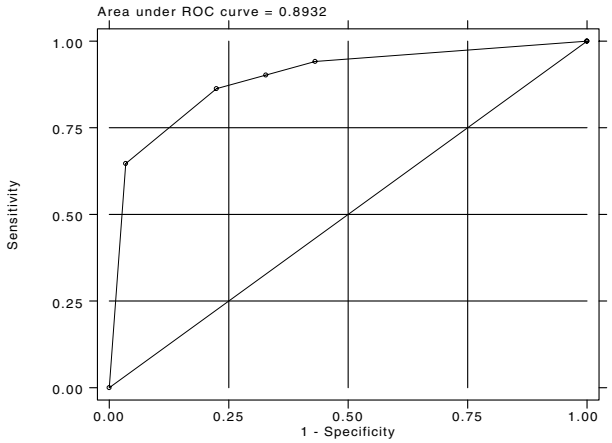

Figure 1. Nonparametric ROC curve for the tomography data.

```
           |                         rating
   disease |         1          2          3          4          5 |     Total
-----------+---------------------------------------------------------+----------
         0 |        33          6          6         11          2 |        58
         1 |         3          2          2         11         33 |        51
-----------+---------------------------------------------------------+----------
     Total |        36          8          8         22         35 |       109

                     ROC                    -Asymptotic Normal--
          Obs       Area     Std. Err.      [95% Conf. Interval]
          ---------------------------------------------------
          109     0.8932       0.0307        0.83295     0.95339
```

By default, `roctab` plots the ROC curve and reports the area under the curve, its standard error and confidence interval. The `nograph` option can be used to suppress the ROC plot.

The ROC curve is plotted by computing the sensitivity and specificity using each value of the rating variable as a possible cut-point. A point is plotted on the graph for each of the cut-points. These plotted points are joined by straight lines to form the ROC curve and the area under the ROC curve computed using the trapezoidal rule.

We can tabulate the computed sensitivities and specificities for each of the possible cut-points by specifying `detail`.

(*Continued on next page*)

```
. roctab disease rating, detail nograph
Detailed report of Sensitivity and Specificity
------------------------------------------------------------------------------
                                        Correctly
Cut point     Sensitivity   Specificity   Classified          LR+          LR-
------------------------------------------------------------------------------
( >= 1 )          100.00%         0.00%       46.79%       1.0000
( >= 2 )           94.12%        56.90%       74.31%       2.1835       0.1034
( >= 3 )           90.20%        67.24%       77.98%       2.7534       0.1458
( >= 4 )           86.27%        77.59%       81.65%       3.8492       0.1769
( >= 5 )           64.71%        96.55%       81.65%      18.7647       0.3655
( >  5 )            0.00%       100.00%       53.21%                    1.0000
------------------------------------------------------------------------------
```

Each cut-point in the table indicates the ratings used to classify tomographs as being from an abnormal subject. For example, the first cut-point, $(>= 1)$, indicates that all tomographs rated as 1 or greater are classified as coming from abnormal subjects. Because all tomographs have a rating of 1 or greater, all are considered abnormal. Consequently, all abnormal cases are correctly classified (sensitivity = 100%), but none of the normal patients are classified correctly (specificity = 0%). For the second cut-point $(>= 2)$, tomographs with ratings of 1 are classified as normal and those with ratings of 2 or greater are classified as abnormal. The resulting sensitivity and specificity are 94.12% and 56.90%, respectively. Using this cut-point we correctly classified 74.31% of the 109 tomographs. Similar interpretations can be used on the remaining cut-points. As mentioned, each cut-point corresponds to a point on the nonparametric ROC curve. The first cut-point, $(>= 1)$, corresponds to the point at (1,1) and the last cut-point, (> 5), to the point at (0,0).

`detail` also reports two likelihood ratios suggested by Choi (1998); the likelihood ratio for a positive test result (LR+), and the likelihood ratio for a negative test result (LR–). The likelihood ratio for a positive test result is the ratio of the probability of a positive test among the truly positive subjects to the probability of a positive test among the truly negative subjects. The likelihood ratio for a negative test result (LR–) is the ratio of the probability of a negative test among the truly positive subjects to the probability of a negative test among the truly negative subjects. Choi points out that LR+ corresponds to the slope of the line from the origin to the point on the ROC curve determined by the cut-point, and similarly LR– corresponds to the slope from the point (1,1) to the point on the ROC curve determined by the cut-point.

By default, `roctab` calculates the standard error for the area under the curve using an algorithm suggested by DeLong, DeLong and Clarke-Pearson (1988), and asymptotic normal confidence intervals. Optionally, standard errors based on methods suggested by Hanley and McNeil (1982) or Bamber (1975) can be computed by specifying `hanley` or `bamber` respectively, and an exact binomial confidence interval can be obtained by specifying `binomial`.

```
. roctab disease rating, nograph bamber
                     ROC      Bamber        -Asymptotic Normal--
          Obs       Area     Std. Err.      [95% Conf. Interval]
          ---------------------------------------------------------
          109     0.8932       0.0300        0.83428     0.95206

. roctab disease rating, nograph hanley binomial
                     ROC      Hanley        -- Binomial Exact --
          Obs       Area     Std. Err.      [95% Conf. Interval]
          ---------------------------------------------------------
          109     0.8932       0.0320        0.81559     0.94180
```

Parametric ROC curves

Dorfman and Alf (1969) developed a generalized approach for obtaining maximum likelihood estimates of the parameters for a smooth fitting ROC curve. The most commonly used method, and the one implemented here, is based upon the binormal model.

The model assumes the existence of an unobserved continuous latent variable that is normally distributed (perhaps after a monotonic transformation) in both the normal and abnormal populations with means μ_n and μ_a, and variances σ_n^2 and σ_a^2, respectively. The model further assumes that the K categories of the rating variable result from partitioning the unobserved latent variable by $K-1$ fixed boundaries. The method fits a straight line to the empirical ROC points plotted using normal probability scales on both axes. Maximum likelihood estimates of the line's slope and intercept, and the $K-1$ boundaries are obtained simultaneously. See *Methods and Formulas* for details.

The intercept from the fitted line is a measurement of

$$\frac{\mu_a - \mu_n}{\sigma_a}$$

and the slope measures

$$\frac{\sigma_n}{\sigma_a}$$

Thus the intercept is the standardized difference between the two latent population means, and the slope is the ratio of the two standard deviations. The null hypothesis of no difference between the two population means is evaluated by testing if the intercept $= 0$, and the null hypothesis that the variances in the two populations are equal is evaluated by testing if the slope $= 1$.

Example

We use Hanley and McNeil's (1982) data described in the previous example to fit a smooth ROC curve assuming a binormal model.

```
. rocfit disease rating
Fitting binormal model:
Iteration 0:   log likelihood = -123.68069
Iteration 1:   log likelihood = -123.64867
Iteration 2:   log likelihood = -123.64855
Iteration 3:   log likelihood = -123.64855
Binormal model                                  Number of obs   =        109
Goodness of fit chi2(2) =        0.21
Prob > chi2             =      0.9006
Log likelihood          =  -123.64855
------------------------------------------------------------------------------
          |     Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
----------+-------------------------------------------------------------------
intercept |  1.656782   0.310456      5.337   0.000       1.048300    2.265265
slope (*) |  0.713002   0.215882     -1.329   0.092       0.289881    1.136123
----------+-------------------------------------------------------------------
    _cut1 |  0.169768   0.165307      1.027   0.152      -0.154227    0.493764
    _cut2 |  0.463215   0.167235      2.770   0.003       0.135441    0.790990
    _cut3 |  0.766860   0.174808      4.387   0.000       0.424243    1.109477
    _cut4 |  1.797938   0.299581      6.002   0.000       1.210770    2.385106
==============================================================================
          |                   Indices from binormal fit
Index     |  Estimate   Std. Err.                           [95% Conf. Interval]
----------+-------------------------------------------------------------------
ROC area  |  0.911331   0.029506                           0.853501    0.969161
delta(m)  |  2.323671   0.502370                           1.339044    3.308298
d(e)      |  1.934361   0.257187                           1.430284    2.438438
d(a)      |  1.907771   0.259822                           1.398530    2.417012
------------------------------------------------------------------------------
(*) z test for slope==1
```

`rocfit` outputs the MLE for the intercept and slope of the fitted regression line along with, in this case, 4 boundaries (because there are 5 ratings) labeled `_cut1` through `_cut4`. In addition `rocfit` also computes and reports 4 indices based on the fitted ROC curve: the area under the curve (labeled `ROC area`), $\delta(m)$ (labeled `delta(m)`), d_e (labeled `d(e)`), and d_a (labeled `d(a)`). More information about these indices can be found in the *Methods and Formulas* section and in Erdreich and Lee (1981).

Note that in the output table we are testing whether or not the variances of the two latent populations are equal, by testing if the slope $= 1$.

In Figure 2 we plot the fitted ROC curve.

(*Graph on next page*)

```
. rocplot
```

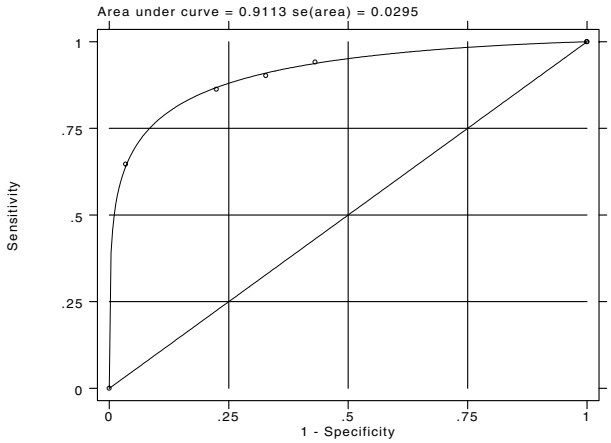

Figure 2. Parametric ROC curve for the tomography data.

Lorenz-like curves

For applications where it is known that the risk status increases or decreases monotonically with increasing values of the diagnostic test, the ROC curve and associated indices are useful in accessing the overall performance of a diagnostic test. When the risk status does not vary monotonically with increasing values of the diagnostic test, however, the resulting ROC curve can be nonconvex and its indices unreliable. For these situations, Lee (1999) proposed an alternative to the ROC analysis based on Lorenz-like curves and associated Pietra and Gini indices.

Lee (1999) mentions at least three specific situations where results from Lorenz curves are superior to those obtained from ROC curves: (1) a diagnostic test with similar means but very different standard deviations in the abnormal and normal populations, (2) a diagnostic test with bimodal distributions in either the normal or abnormal populations, and (3) a diagnostic test distributed symmetrically in the normal population and askew in the abnormal.

Note that when the risk status increases or decreases monotonically with increasing values of the diagnostic test, the ROC and Lorenz curves yield interchangeable results.

Example

To illustrate the use of the `lorenz` option we constructed a fictitious dataset that yields results similar to those presented in Table III of Lee (1999). The data assumes that a 12 point rating scale was used to classify 442 diseased and 442 healthy subjects. We list a few of the observations.

```
. list in 1/7, noobs
  disease      class        pop
        1         12        169
        0         11          7
        1         10         17
        0          9         41
        1          8         18
        0          7         85
        0          6         85
```

The dataset consists of 24 observations, 12 observations from diseased individuals and 12 from nondiseased individuals. Each observation corresponds to one of the 12 classification values of the rating scale variable, `class`. The number of subjects represented by each observation is given by the `pop` variable, making this a frequency-weighted dataset. The data were generated assuming a binormal distribution of the latent variable with similar means for the normal and abnormal population, but with the standard deviation for the abnormal population 5 times greater than that of the normal population.

(*Graph on next page*)

```
. roctab disease class [fweight=pop]
```

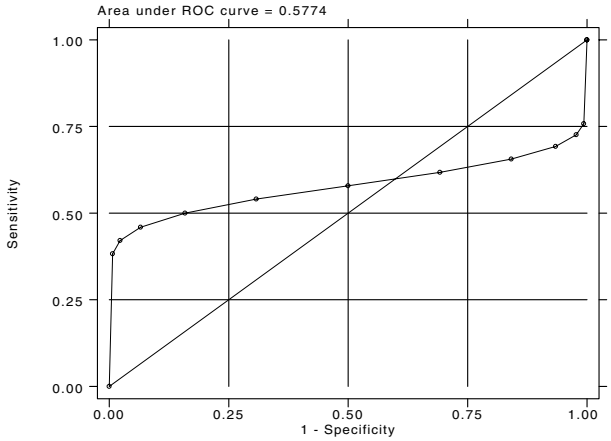

Figure 3. Nonparametric ROC curve for the artificial data.

The resulting ROC curve is nonconvex or, as termed by Lee, "wiggly." Lee argues that for this and similar situations, the Lorenz curve and indices are preferred.

```
. roctab disease class [fweight=pop], lorenz
```

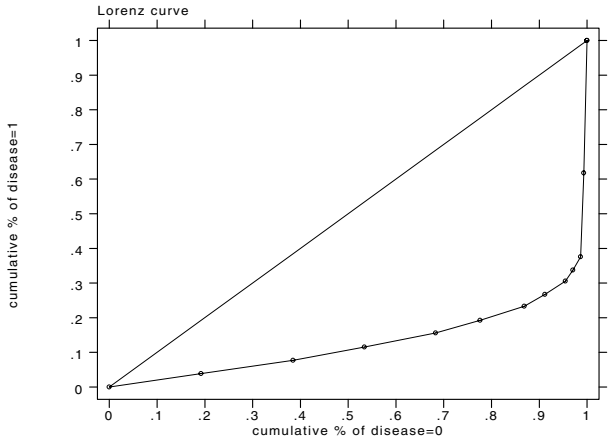

Figure 4. The Lorenz curve for the artificial data.

```
 Lorenz Curve
----------------------------
  Pietra index =   0.6493
  Gini index   =   0.7441
```

Similarly to ROC curves, a more bowed Lorenz curve suggests a better diagnostic test. This "bowedness" is quantified by the Pietra index which is geometrically equivalent to twice the largest triangle that can be inscribed in the area between the curve and the diagonal line, and the Gini index which is equivalent to twice the area between the Lorenz curve and the diagonal. Lee (1999) provides several additional interpretations for the Pietra and Gini indices. Interested individuals should consult the reference for complete information.

Comparing areas under the ROC curve

The area under multiple ROC curves can be compared using `roccomp`. The command syntax is slightly different if the ROC curves are correlated (i.e., different diagnostic tests applied to the same sample) or independent (i.e., diagnostic tests applied to different samples).

Example with correlated data

Hanley and McNeil (1983) presented data from an evaluation of two computer algorithms designed to reconstruct CT images from phantoms. We will call these two algorithms modalities 1 and 2. A sample of 112 phantoms was selected; 58 phantoms were considered normal, and the remaining 54 phantoms abnormal. Each of the two modalities was applied to each phantom and the resulting images rated by a reviewer using a six point scale: 1–definitely normal, 2–probably normal, 3–possibly normal,

4–possibly abnormal, 5–probably abnormal, and 6–definitely abnormal. Because each modality was applied to the same sample of phantoms the two sets of outcomes are correlated.

We list the first seven observations.

```
. list in 1/7
          mod1       mod2     status
  1.         1          1          0
  2.         1          2          0
  3.         1          3          1
  4.         2          1          0
  5.         2          1          1
  6.         2          2          0
  7.         2          3          0
```

Note that the data are in wide form. This is required when dealing with correlated data. Each observation corresponds to one phantom. The variable `mod1` identifies the rating assigned for the first modality, and `mod2` identifies the rating assigned for the second modality. The true status of the phantoms is given by `status=0` if normal, and `status=1` if abnormal. Note that observations with at least one missing rating are dropped from the analysis.

We plot the two ROC curves and compare their areas.

```
. roccomp status mod1 mod2, symbol(oT)
                                ROC                    -Asymptotic Normal--
                    Obs       Area     Std. Err.      [95% Conf. Interval]
-------------------------------------------------------------------------------
mod1                112     0.8828       0.0317        0.82067     0.94498
mod2                112     0.9302       0.0256        0.88005     0.98042
-------------------------------------------------------------------------------
Ho: area(mod1) = area(mod2)
chi2(1) =     2.31       Pr>chi2 =   0.1282
```

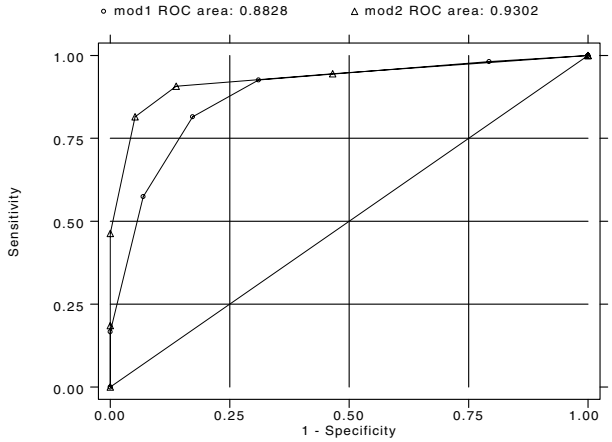

Figure 5. Comparison of ROC curves for the CT images data.

By default, `roccomp` plots the ROC curves on the same graph. Optionally, the curves can be plotted side by side, each on its own graph, by specifying `separate`.

For each curve, `roccomp` reports summary statistics and provides a test for the equality of the area under the curves using an algorithm suggested by DeLong, DeLong and Clarke-Pearson (1988).

Although the area under the ROC curve for modality 2 is larger than that of modality 1, the chi-squared test yielded a significance probability of 0.1282, suggesting that there is no significant difference between these two areas.

The `roccomp` command can also be used to compare more than two ROC areas. To illustrate this, we modified the previous dataset by including a fictitious third modality.

```
. roccomp status mod1 mod2 mod3, symbol(oTS)
```

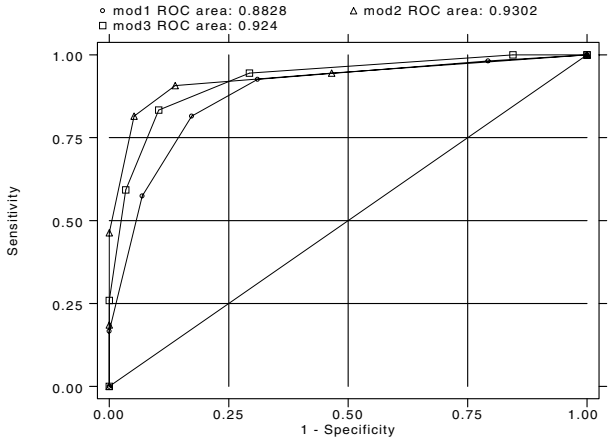

Figure 6. Comparing three modalities for the CT images data.

```
                              ROC                    -Asymptotic Normal--
                    Obs       Area     Std. Err.      [95% Conf. Interval]
-------------------------------------------------------------------------
mod1                112     0.8828       0.0317        0.82067     0.94498
mod2                112     0.9302       0.0256        0.88005     0.98042
mod3                112     0.9240       0.0241        0.87670     0.97132
-------------------------------------------------------------------------
Ho: area(mod1) = area(mod2) = area(mod3)
chi2(2) =     6.54       Pr>chi2 =   0.0381
```

By default, `roccomp` tests whether the areas under the ROC curves are all equal. Other comparisons can be tested by creating a contrast matrix and specifying `test(`*matname*`)` where *matname* is the name of the contrast matrix.

For example, assume that we are interested in testing whether the area under the ROC for `mod1` is equal to that of `mod3`. To do this, we can first create an appropriate contrast matrix and then specify its name with the `test()` option.

Of course this is a trivial example because we could have just specified

```
. roccomp status mod1 mod3
```

without including `mod2` and would have obtained the same test results. However, for illustration we will continue with this example.

The contrast matrix must have its number of columns equal to the number of `classvars` (i.e., the total number of ROC curves), a number of rows less than or equal to the number of `classvars`, and the elements of each row must add to zero.

```
. matrix C=(1,0,-1)
. roccomp status mod1 mod2 mod3, test(C) nograph
                              ROC                    -Asymptotic Normal--
                    Obs       Area     Std. Err.      [95% Conf. Interval]
-------------------------------------------------------------------------
mod1                112     0.8828       0.0317        0.82067     0.94498
mod2                112     0.9302       0.0256        0.88005     0.98042
mod3                112     0.9282       0.0229        0.88327     0.97305
-------------------------------------------------------------------------
Ho: Comparison as defined by contrast matrix: C
chi2(1) =     5.55       Pr>chi2 =   0.0184
```

Note that although all three areas are reported, the comparison is made using the specified contrast matrix.

Perhaps more interesting would be to compare the area from `mod1` to the average area of `mod2` and `mod3`.

```
. matrix C=(1,-.5,-.5)
. roccomp s mod1 mod2 mod3 [fw=pop],test(C)
                              ROC                    -Asymptotic Normal--
                    Obs       Area     Std. Err.      [95% Conf. Interval]
-------------------------------------------------------------------------
mod1                112     0.8828       0.0317        0.82067     0.94498
mod2                112     0.9302       0.0256        0.88005     0.98042
mod3                112     0.9282       0.0229        0.88327     0.97305
-------------------------------------------------------------------------
Ho: Comparison as defined by contrast matrix: C
chi2(1) =     3.50       Pr>chi2 =   0.0613
```

Other contrasts could be made. For example, we could test if mod3 is better than at least one of the other two by first creating the following contrast matrix.

```
. matrix C=(-1, 0, 1 \ 0, -1, 1)
. mat list C
C[2,3]
    c1  c2  c3
r1  -1   0   1
r2   0  -1   1
```

Example with independent data

In the previous example we noted that because each test modality was applied to the same sample of phantoms, the classification outcomes were correlated. Now assume that we have collected the same data as presented by Hanley and McNeil (1983), except that we applied the first test modality to one sample of phantoms and the second test modality to a different sample of phantoms. The resulting measurements are now considered independent.

Here are a few of the observations.

```
.  list in 1/7
        status     rating        mod        pop
  1.         0          1          1         12
  2.         0          1          2         31
  3.         1          1          1          1
  4.         1          1          2          3
  5.         0          2          1         28
  6.         0          2          2         19
  7.         1          2          2          2
```

Note that the dataset is in long form. This is required when dealing with independent data. The dataset consists of 24 observations, 6 observations corresponding to abnormal phantoms and 6 to normal phantoms evaluated using the first modality, and similarly 6 observations corresponding to abnormal phantoms and 6 to normal phantoms evaluated using the second modality. The number of phantoms corresponding to each observation is given by the pop variable. Once again we have frequency weighted data. The variable mod identifies the modality and rating is the assigned classification.

We can view our data better utilizing the table command.

```
. table status rating [fw=pop], by(mod) row col
----------+-------------------------------------------------
mod and   |                     rating
status    |     1      2      3      4      5      6  Total
----------+-------------------------------------------------
1         |
        0 |    12     28      8      6      4             58
        1 |     1      3      6     13     22      9      54
          |
    Total |    13     31     14     19     26      9    112
----------+-------------------------------------------------
2         |
        0 |    31     19      5      3                    58
        1 |     3      2      5     19     15     10      54
          |
    Total |    34     21     10     22     15     10    112
----------+-------------------------------------------------
```

The status variable indicates the true status of the phantoms, status=0 if normal and status=1 if abnormal.

We now compare the areas under the two ROC curves.

(*Continued on next page*)

```
. roccomp status rating [fw=pop], by(mod) symbol(oT)
```

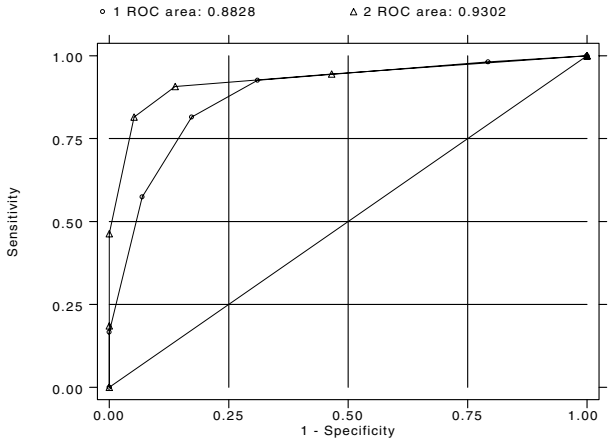

Figure 7. Comparing independent data.

```
                                ROC                    -Asymptotic Normal--
mod                 Obs       Area     Std. Err.      [95% Conf. Interval]
-------------------------------------------------------------------------------
1                   112     0.8828       0.0317        0.82067     0.94498
2                   112     0.9302       0.0256        0.88005     0.98042
-------------------------------------------------------------------------------
Ho: area(1) = area(2)
chi2(1) =     1.35       Pr>chi2 =   0.2447
```

Saved Results

`roctab` saves in `r()`:

Scalars

`r(N)`	number of observations
`r(area)`	area under the ROC curve
`r(se)`	standard error for the area under the ROC curve
`r(lb)`	lower bound of CI for the area under the ROC curve
`r(ub)`	upper bound of CI for the area under the ROC curve
`r(pietra)`	Pietra index
`r(gini)`	Gini index

(*Continued on next page*)

`rocfit` saves in `e()`:

Scalars
- `e(N)` number of observations
- `e(ll)` log likelihood
- `e(chi2_gf)` χ^2 goodness-of-fit test
- `e(df_gf)` goodness-of-fit test degrees of freedom
- `e(area)` area under the ROC curve
- `e(se_area)` standard error for the area under the ROC curve
- `e(deltam)` $\delta(m)$
- `e(se_delm)` standard error for $\delta(m)$
- `e(de)` d_e index
- `e(se_de)` standard error for the d_e index
- `e(da)` d_a index
- `e(se_da)` standard error for the d_a index

Macros
- `e(cmd)` `rocfit`
- `e(depvar)` name of dependent variable
- `e(chi2type)` goodness of fit; type of model χ^2 test

Matrices
- `e(b)` coefficient vector
- `e(V)` variance–covariance matrix of the estimators

Functions
- `e(sample)` marks estimation sample

`roccomp` saves in `r()`:

Scalars
- `r(N_g)` number of groups
- `r(chi2)` χ^2
- `r(df)` χ^2 degrees of freedom
- `r(p)` significance probability

Matrices
- `r(V)` variance–covariance matrix

Methods and Formulas

Assume that we applied a diagnostic test to each of N_n normal and N_a abnormal subjects. Further assume that the higher the outcome value of the diagnostic test, the higher the risk of the subject being abnormal. Let $\widehat{\theta}$ be the estimated area under the curve, and let $X_i, i = 1, 2, \ldots, N_a$ and $Y_j, j = 1, 2, \ldots, N_n$ be the values of the diagnostic test for the abnormal and normal subjects, respectively.

Nonparametric ROC

The points on the nonparametric ROC curve are generated by using each possible outcome of the diagnostic test as a classification cut-point and computing the corresponding sensitivity and $1 -$ specificity. These points are then connected by straight lines, and the area under the resulting ROC curve computed using the trapezoidal rule.

The default standard error for the area under the ROC curve is computed using the algorithm described by DeLong, DeLong and Clarke-Pearson (1988). For each abnormal subject, i, define

$$V_{10}(X_i) = \frac{1}{N_n} \sum_{j=1}^{N_n} \psi(X_i, Y_j)$$

and for each normal subject, j, define

$$V_{01}(Y_j) = \frac{1}{N_a} \sum_{i=1}^{N_a} \psi(X_i, Y_j)$$

where

$$\psi(X,Y) = \begin{cases} 1 & Y < X \\ \frac{1}{2} & Y = X \\ 0 & Y > X \end{cases}$$

Define

$$S_{10} = \frac{1}{N_a - 1}\sum_{i=1}^{N_a}(V_{10}(X_i) - \widehat{\theta})^2$$

and

$$S_{01} = \frac{1}{N_n - 1}\sum_{j=1}^{N_n}(V_{01}(Y_j) - \widehat{\theta})^2$$

The variance of the estimated area under the ROC curve is given by

$$\mathrm{var}(\widehat{\theta}) = \frac{1}{N_a}S_{10} + \frac{1}{N_n}S_{01}$$

The `hanley` standard error for the area under the ROC curve is computed using the algorithm described by Hanley and McNeil (1982). It requires the calculation of two quantities, Q_1 and Q_2, where Q_1 is the prob(two randomly selected abnormal subjects will both have a higher score than a randomly selected normal subject), and Q_2 is the prob(one randomly selected abnormal subject will have a higher score than any two randomly selected normal subjects). Then the Hanley and McNeil variance of the estimated area under the ROC curve is

$$\mathrm{var}(\widehat{\theta}) = \frac{\widehat{\theta}(1-\widehat{\theta}) + (N_a - 1)(Q_1 - \widehat{\theta}^2) + (N_n - 1)(Q_2 - \widehat{\theta}^2)}{N_a N_n}$$

The `bamber` standard error for the area under the ROC curve is computed using the algorithm described by Bamber (1975). For any two Y values, Y_j and Y_k, and any X_i value, define

$$b_{yyx} = p(Y_j, Y_k < X_i) + p(X_i < Y_j, Y_k) - 2p(Y_j < X_i < Y_k)$$

and similarly, for any two X values, X_i and X_l, and any Y_j value, define

$$b_{xxy} = p(X_i, X_l < Y_j) + p(Y_j < X_i, X_l) - 2p(X_i < Y_j < X_l)$$

Then Bamber's unbiased estimate of the variance for the area under the ROC curve is

$$\mathrm{var}(\widehat{\theta}) = \frac{1}{4}(N_a - 1)(N_n - 1)[p(X \neq Y) + (N_a - 1)b_{xxy} + (N_n - 1)b_{yyx} - 4(N_a + N_n - 1)(\widehat{\theta} - 0.5)^2]$$

Asymptotic confidence intervals are constructed and reported by default, assuming a normal distribution for the area under the ROC curve.

Exact binomial confidence intervals are calculated as described in [R] **ci** with p equal to the area under the ROC curve.

Parametric ROC curves

Dorfman and Alf (1969) developed a general procedure for obtaining maximum likelihood estimates of the parameters of a smooth fitting ROC curve. The most common method, and the one implemented in Stata, is based upon the binormal model.

The model assumes that there is an unobserved continuous latent variable that is normally distributed in both the normal and abnormal populations. The idea is better explained with the use of the following illustration.

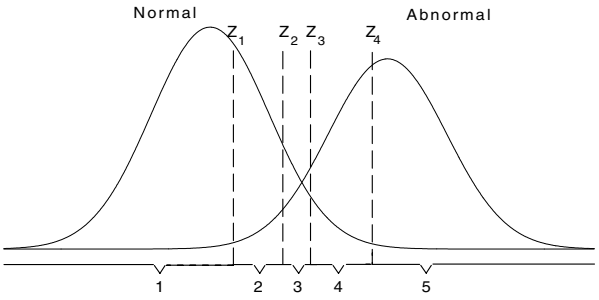

Figure 8. Illustration of the binormal distribution.

It is assumed that the latent variable for the normal and abnormal subjects are each normally distributed, perhaps after a monotonic transformation, with means μ_n and μ_a, and variances σ_n^2 and σ_a^2, respectively.

This latent variable is assumed to be partitioned into the k categories of the rating variable by $k-1$ fixed boundaries. In Figure 8, the $k=5$ categories of the rating variable identified on the bottom result from the partition of the four boundaries Z_1 through Z_4.

Let R_j for $j=1,2,\ldots,k$, indicate the categories of the rating variable, and let $i=1$ if the subject belongs to the normal group, and $i=2$ to the abnormal group.

Then

$$p(R_j|i=1) = F(Z_{l=j}) - F(Z_{l=j-1})$$

where $Z_k = (x_k - \mu_n)/\sigma_n$, F is the cumulative normal distribution, $F(Z_0)=0$ and $F(Z_k)=1$. Also,

$$p(R_j|i=2) = F(bZ_{l=j} - a) - F(bZ_{l=j-1} - a)$$

where $b = \sigma_n/\sigma_a$ $a = (\mu_a - \mu_n)/\sigma_a$.

The parameters a, b and the $k-1$ fixed boundaries Z_j are simultaneously estimated by maximizing the log likelihood function

$$\log L = \sum_{i=1}^{2}\sum_{j=1}^{k} r_{ij}\log\big(p(R_j|i)\big)$$

where r_{ij} is the number of R_j's in group i.

The area under the fitted ROC curve is computed as

$$\Phi\left(\frac{a}{\sqrt{1+b^2}}\right)$$

where Φ is the standard normal cumulative distribution function.

Point estimates for the ROC curve indices are estimated as follows:

$$\delta(m) = \frac{a}{b}, \qquad d_e = \frac{2a}{b+1}, \qquad d_a = \frac{a\sqrt{2}}{\sqrt{1+b^2}}$$

Variances for these indices are computed using the delta method.

$\delta(m)$ estimates $(\mu_a - \mu_n)/\sigma_n$, d_e estimates $2(\mu_a - \mu_n)/(\sigma_a - \sigma_n)$, and d_a estimates $\sqrt{2}(\mu_a - \mu_n)/(\sigma_a^2 - \sigma_n^2)^2$.

Simultaneous confidence bands for the entire curve are obtained as suggested by Ma and Hall (1993) by first obtaining Working–Hotelling confidence bands for the fitted straight line in normal probability coordinates, and then transforming them back to ROC coordinates.

Comparing areas under the ROC curve

Areas under ROC curves are compared using an algorithm suggested by DeLong, DeLong and Clarke-Pearson (1988). Let $\widehat{\theta} = (\widehat{\theta}^1, \widehat{\theta}^2, \ldots, \widehat{\theta}^k)$ be a vector representing the areas under k ROC curves. For the rth area define

$$V_{10}^r(X_i) = \frac{1}{N_n}\sum_{j=1}^{N_n}\psi(X_i^r, Y_j^r)$$

and for each normal subject, j, define

$$V_{01}^r(Y_j) = \frac{1}{N_a}\sum_{i=1}^{N_a}\psi(X_i^r, Y_j^r)$$

where

$$\psi(X^r, Y^r) = \begin{cases} 1 & Y^r < X^r \\ \frac{1}{2} & Y^r = X^r \\ 0 & Y^r > X^r \end{cases}$$

Define the $k \times k$ matrix $\mathbf{S_{10}}$ such that the (r,s)th element is

$$S_{10}^{r,s} = \frac{1}{N_a - 1}\sum_{i=1}^{N_a}(V_{10}^r(X_i) - \widehat{\theta}^r)(V_{10}^s(X_i) - \widehat{\theta}^s)$$

and $\mathbf{S_{01}}$ such that the (r,s)th element is

$$S_{01}^{r,s} = \frac{1}{N_n - 1}\sum_{j=1}^{N_n}(V_{01}^r(Y_i) - \widehat{\theta}^r)(V_{01}^s(Y_i) - \widehat{\theta}^s)$$

Then the covariance matrix is

$$S = \frac{1}{N_a}S_{10} + \frac{1}{N_n}S_{01}$$

Let $\mathbf{L}$ be a contrast matrix defining the comparison, then

$$(\widehat{\theta} - \theta)'\mathbf{L}'\left[\mathbf{LSL}'\right]^{-1}\mathbf{L}(\widehat{\theta} - \theta)$$

has a chi-square distribution with degrees of freedom equal to the rank of $\mathbf{LSL}'$.

References

Bamber, D. 1975. The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology* 12: 387–415.

Choi, B. C. K. 1998. Slopes of a receiver operating characteristic curve and likelihood ratios for a diagnostic test. *American Journal of Epidemiology* 148: 1127–1132.

DeLong, E. R., D. M. DeLong, and D. L. Clarke-Pearson. 1988. Comparing the areas under two or more correlated receiver operating curves: A nonparametric approach. *Biometrics* 44: 837–845.

Dorfman, D. D. and E. Alf. 1969. Maximum likelihood estimation of parameters of signal detection theory and determination of confidence intervals-rating method data. *Journal of Mathematical Psychology* 6: 487–496.

Erdreich, L. S. and E. T. Lee. 1981. Use of relative operating characteristic analysis in epidemiology: a method for dealing with subjective judgment. *American Journal of Epidemiology* 114: 649–662.

Hanley, J. A. and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143: 26–36.

——. 1983. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* 148: 839–843.

Lee, W. C. 1999. Probabilistic analysis of global performances of diagnostic test: Interpreting the Lorenz curve-based summary measures. *Statistics in Medicine* 18: 455–471.

Ma, G. and W. J. Hall. 1993. Confidence bands for the receiver operating characteristic curves. *Medical Decision Making* 13: 191–197.

sg121	Seemingly unrelated estimation and the cluster-adjusted sandwich estimator

Jeroen Weesie, Utrecht University, The Netherlands, j.weesie@fss.uu.nl

Introduction

Suppose I have estimated two (or more) models, either on different datasets, on overlapping datasets, or on the same dataset. I want to test whether some relationship between the estimators holds. Such a hypothesis is often that the coefficients estimated by one estimator are equal (or proportional) to the coefficients estimated by the other estimator. There may be both substantial and specification-type arguments for assuming such a relationship between the estimators. I provide some examples.

- I estimated an ordinal response model (for example, `ologit`), and I wonder whether some consecutive categories can be joined. I recode the dependent variable and estimate again, and I check whether the estimated coefficients are different. At face value, they look the same, but are the differences indeed not significant?

  Also, I wonder whether some category (for example, "neutral" in a 5-point Likert scale) really fits into the scale. I reestimate the model but exclude the neutral observations. Are the results the same?

- I estimate comparable logit, probit, and gompit models. (In Stata, I could use `cloglog`.) Do these models yield the same results, that is, are the coefficients the same, or better yet, proportional?

- I want to test whether the effect of `x1` on `y1` is the same as the effect of `x2` on `y2`. If `y1` and `y2` are both "continuous", I may turn to multivariate regression (`sureg`, `mvreg`, and `reg3` in Stata) and Wald testing (`test` in Stata). But what if `y1` and `y2` are both binary, so I turn to logit models? And even more complicated, what if `y1` is binary and `y2` is continuous?

- I want to perform some analysis and I have two different measurements of a concept that I want to use as an independent variable. Do the results differ significantly, depending on which instrument I use?

- Hausman and McFadden (1984) discuss a probabilistic version of the assumption of independence of irrelevant alternatives in discrete choice models, namely that the distribution of choices between alternatives in some set A, given that one of the A-alternatives is chosen, does not depend on extra but nonchosen alternatives B. They propose a test statistic derived as a Mahalanobis (covariance)-distance between parameter estimates based on decisions from the full set of alternatives and estimates based on data that ignore some alternatives. But how does one proceed if the observations are clustered, for example, travel mode decisions of persons within a household?

- I want to analyze whether two different labor market outcome variables (say the continuous variable income and the binary variable promotion) depend on formal education and labor market experience in a similar way. More precisely, I want to know whether the ratio of the effects of education and experience on income and on promotion are the same, so that one can perceive of a one-dimensional "human capital" concept in explaining labor market success.

Hausman (1978) proposed a test that is appropriate to test some of these hypotheses; equality constraints between two estimators, one of which is efficient under H_0, while the other estimator is consistent under both H_0 and H_a. This test is available in Stata via the command `hausman`. The efficiency requirement is a strong one indeed. White (1982, 1994) has shown how Hausman's test can be modified if the requirement that one of the estimators is efficient is not fulfilled. Clogg et al. (1995) is a relatively nontechnical explication of this test in the context of generalized linear models. I expect that White's test is actually superior to Hausman's test even if one of the estimators is efficient, as in most cases efficiency only holds asymptotically. It would be interesting to verify (using a number of Monte Carlo experiments, for example) whether this intuition is correct. As I demonstrate below, White's test is essentially a fairly straightforward application of Stata's adjustment for clustering of the robust or sandwich estimator of the asymptotic variance of estimators.

However, some of the testing problems raised above are not tackled by the Hausman–White test. In this note, I will also show how the research by Hausman and White can be generalized, admittedly modestly, to deal with, for example

- Hypotheses involving more than two estimators (see Example 3);

- Nonlinear between-estimator hypotheses (see Example 3);

- Pooling a number of inefficient estimators, to obtain a pooled more efficient estimator.

I will also discuss a Stata command that implements White's generalization of the Hausman test and which can also be used to deal with some relatively minor generalizations.

Example: Cross-model hypotheses for logit models

In this illustration, I demonstrate how some cross-model hypotheses can be tested using the facilities already in Stata. It will also be made clear that a new facility is required to perform more general cross-model testing.

I want to test whether the effects of x_1 on a binary y_1 is the same as the effect of x_2 on a binary y_2. In this setting, x_1 may equal x_2 and y_1 may equal y_2. I expect that logit regression models are appropriate marginal models for these responses. For simplicity, I ignore further predictor variables in the marginal models. If the two logit models are fitted on *different* data, the estimators are (stochastically) independent, and so a Wald test rejects the null hypothesis if

$$\frac{\widehat{b}(x_1) - \widehat{b}(x_2)}{\sqrt{\widehat{\sigma}^2_{\widehat{b}(x_1)} + \widehat{\sigma}^2_{\widehat{b}(x_2)}}}$$

is larger than the appropriate χ^2_1 threshold. The approach presented applies in the case of models estimated on independent samples, and provides a convenient way to test cross-model hypotheses. However, as in the examples provided in the introduction, if the models are fitted on the *same* data (or on partially overlapping data), the estimators are stochastically dependent and the above test that ignores the covariance between the estimators is not appropriate.

One may conceive of this problem in a "stacked" way, with double the number of observations, the dependent variable being y_1 in the first half, and y_2 in the second half; the predictor variable z_1 is set to x_1 in the first half of the expanded data, and to 0 in the rest, and z_2 defined analogously. The following diagram illustrates the expanded format

$$\begin{bmatrix} id & y_1 & y_2 & x_1 & x_2 \\ 1 & y_{11} & y_{21} & x_{11} & x_{21} \\ 2 & y_{12} & y_{22} & x_{12} & x_{22} \\ 3 & y_{13} & y_{23} & x_{13} & x_{23} \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} id & y & z_1 & z_2 \\ 1 & y_{11} & x_{11} & 0 \\ 2 & y_{12} & x_{12} & 0 \\ 3 & y_{13} & x_{13} & 0 \\ 1 & y_{21} & 0 & x_{21} \\ 2 & y_{22} & 0 & x_{22} \\ 3 & y_{23} & 0 & x_{23} \end{bmatrix}$$

The observations in the expanded data are *clustered* by the original subjects, identified with the identifier *id*. The simplest way to deal with clustering is via the cluster-modification for the sandwich estimator (see Rogers 1993 and the `_robust` section in the Stata 6.0 manuals); a more efficient test using the quasi-likelihood method of "generalized estimation equations" is an obvious alternative, not discussed here. In Stata the data manipulation can easily be accomplished with the `stack` command:

```
. gen zero = 0
. gen one  = 1
. stack  id y1 x1 zero zero   id y2 zero x2 one, into(id y z1 z2 mod2)
. logit y z1 z2 mod2, cluster(id)
. test z1 = z2
```

The variable `mod2` is a dummy for the "second model" that is included to allow the intercept in the second model to differ from that of the first model. In this case, the two logit models were fitted to the same data. It would apply also, without any modification, to the case in which the two logit models were estimated on overlapping data that result if the y or x variables are missing on some observations. The resulting test statistic produced by the above code fragment is *identical* to that obtained more conveniently via the new `suest` command which is included in this insert and is discussed below:

```
. suest fit A: logit y1 x1
. suest fit B: logit y2 x2
. suest combine A B
. test Ax1 = Bx2
```

The stacking method can be applied not only to `logit` models, but applies to all estimation commands that support the `cluster` option, for example, `regress`, `poisson`, and `oprobit`. The "stacking" approach clearly generalizes to stacking more than two logit or other models, to testing more general linear hypotheses, and also to testing nonlinear cross-model hypotheses (in Stata using `testnl`). There are, however, two disadvantages to the stacking method. First, if the models include ancillary parameters (for example, in regression: the residual variance; in ordinal response models: the cutpoints; in lognormal survival time regression: the time scale parameter), these are constrained to be equal between the stacked models. In the approach via `suest`, this is relaxed. Second, the stacking method does not generalize, however, to "stack" different models, such as a logit model and a regression model; different likelihoods are used to model different parts of the data. To analyze this more general setting, I have to develop the theory somewhat more formally.

General theory

The starting point is that I have estimated different models on the *same* data (or, to data that have some overlap), and hence the estimators are stochastically dependent. I want to test a between-estimator hypothesis H_0. To derive a test, I need the *simultaneous distribution* of the series of estimators, $b = (b_1, \ldots, b_k)$, where the estimator b_j is defined as "the" solution of

$$L_j(b_j) = \sum_i u_{ij}(b_j) = 0, \qquad j = 1, \ldots, k$$

I will refer to the u_{ij} as the "scores", but the approach works more generally than for maximum likelihood estimation, and actually applies to equation estimators and misspecified models as developed in the theory of White (1994). Under "suitable regularity conditions" (White 1982), the b_j are asymptotically normally distributed, with variance estimated consistently by the sandwich estimator

$$V_j = \text{AVar}(b_j) = D_j^{-1} \sum_i u_{ij} u_{ij}' \, D_j^{-1}$$

where D_j is the Jacobian of L_j in b_j. In the context of maximum likelihood estimation, D_j can be estimated consistently by (minus) the Hessian of the log likelihood or by the Fisher information. If the model is also well specified, the sandwiched term $(\sum_i u_{ij} u_{ij}')$ converges in probability to D_j, and so V_j may be consistently estimated by D_j^{-1}.

I now define the simultaneous "stacked" estimation equation:

$$L(b) = (L_1(b_1), L_2(b_2), \ldots, L_k(b_k)) = 0$$

Under "suitable regularity conditions" (see White 1982 for details), b is asymptotically *jointly* normal distributed as well. The Jacobian and scores of the simultaneous equation are easily expressed in the Jacobian and scores of the separate equations. Clearly the Jacobian of L,

$$D(b) = \delta L / \delta b$$

is block diagonal with blocks $D_1, D_2, \ldots, D_k$. Observe that the inverse of $D(b)$ is again block diagonal with the inverses of D_j on the diagonal. The scores u of L are simply obtained as the *concatenated* scores of the separate equations:

$$u_i = (u_{i1}, u_{i2}, \ldots, u_{ik})$$

The sandwich estimator for the asymptotic variance of b reads again

$$V = \text{AVar}(b) = D(b)^{-1} \sum_i u_i u_i' \, D(b)^{-1}$$

Taking a closer "partitioned" look at this expression, we see that $\text{AVar}(b_j)$ is estimated by

$$D_j^{-1} \sum u_{ij} u_{ij}' D_j^{-1}$$

which is yet again the familiar sandwich estimator for b_j based on the separate estimation equation L_j. Thus, considering several estimators simultaneously in this way, does not affect the estimators of the asymptotic variances of these estimators. But, as a bonus, we obtain an estimate of the *covariance* of different estimators. The sandwich estimator of the asymptotic covariance V_{jh} of b_j and b_h is

$$V_{jh} = \text{ACov}(b_j, b_h) = D_j^{-1} \sum_i u_{ij} u_{ih}' \, D_h^{-1}$$

which is also obtained by White (1982).

It is of some interest to understand that this estimator for the covariance of estimators is "nothing but" an application of the cluster modification of the sandwich estimator proposed by Rogers (1993). Consider the stacked data format as proposed in Section 1, and assume that Stata would be able to estimate a "stacked model" in which different models apply to different observations, for example, a logit model for the first half, and a regression model to the second half, and there exists a one-to-one cluster relation between the first and second half. If there are no common parameters to both models, the score statistics of parameters for the stacked models are zero in the half of the data in which they do not occur. In the Rogers method, we have to add the score statistics of the observations within a cluster. This boils down to concatenating the score statistics at the level of the cluster!

Test statistics

Having established the asymptotic distribution of the simultaneous estimator b, one can use the familiar apparatus of Wald testing to obtain a test for equality constraints between whatever is estimated by the estimators (their probability limits). First, consider the standard case studied by Hausman (1978) and White (1982); namely $k = 2$ and an equality hypothesis, $H_0 : \operatorname{plim} b_1 - b_2 = 0$. We have

$$V = \text{AVar}(b_1 - b_2) = V_1 + V_2 - V_{12} - V_{12}'$$

and so a Wald-type test statistic for H_0 can be obtained as

$$H = (b_1 - b_2)' V^- (b_1 - b_2)$$

which under H_0 follows asymptotically a χ^2 distribution with $\operatorname{rank}(V)$ degrees of freedom (additional regularity conditions are required if V is not of full rank), and so we will reject H_0 if H is large, relative to a χ^2-derived threshold.

Hausman (1978) has shown that if b_1 is efficient (i.e., has minimal asymptotic variance), then $\text{ACov}(b_1 - b_2, b_1) = 0$, and so $\text{AVar}(b_1 - b_2) = V_2 - V_1$. Moreover, due to the efficiency of b_1, $V_2 - V_1$ is asymptotically positive definite. But in finite samples (i.e., in all real applications), positive definiteness is not ensured; in that case, Hausman's test is not well-defined, which may be confusing to nonstatisticians.

To apply the approach advocated in this note, we not only need the estimators b_j and Jacobians V_j, but also the score statistics u_{ij}. These will often be available in software that computes sandwich estimators of (co)variance, and otherwise are often simple to compute "by hand." Thus, at the limited costs of the computation of score statistics, we can do without the efficiency assumption in Hausman (1978), and so can perform a Hausman test more generally than envisaged by Hausman, e.g., in the context of clustered observations. As Stata produces score statistics for most estimation commands, it is particularly suited to perform the generalized Hausman test with little effort. The command `suest` described below reduces the effort even more.

The approach is not restricted to just two estimators. One can also test hypotheses formulated as other linear equality constraints (see White 1994, 270). Nonlinear hypotheses may be a more interesting generalization of the Hausman test. In particular, it may be appropriate to test a proportionality hypothesis of the coefficients of different models, e.g., to test that ordinary regression and logistic regression on a dichotomized *depvar* lead to the same conclusions. The details are fully standard and hence need not be elaborated here; for an illustration in Stata, see Example 3 below.

The suest command

The generalized Hausman-test for cross-estimators hypotheses is implemented in a keyword driven command `suest`, with syntax

`suest` `fit` *name*`:` *est_cmd*

`suest save` *name scorevar_1* [*scorevar_2* ...]

`suest combine` [*name_list*] [`, cluster(`*varname*`) level(`#`)`]

`suest clear`

`suest drop` *name_list*

`suest query`

Description

`suest` combines a series of parameter-estimates and associated (co)variance matrices into a single parameter vector and *simultaneous* covariance matrix of the sandwich/robust type that can be analyzed with Stata's standard post-estimation commands such as `test` and `testnl`. This allows the testing of cross-model hypotheses. One can also compute a two-stage linear-constrained estimator with cross-model linear constraints via `linest` (Weesie 1999).

`suest` does not require that the models are estimated on the same samples, due to explicit selection, or due to missing values. However if weights are applied, the same weights should be applied to all models.

Key commands

`suest fit` invokes an estimation command and saves the required results under a name of at most four characters. Currently, the following estimation commands are supported: `clogit2`, `cloglog`, `heckman`, `heckprob`, `intreg`, `logistic`, `logit`, `poisson`, `ologit`, `mlogit`, `nbreg`, `oprobit`, `probit`, `regh`, `scobit`, `zip`.

Note that `regress` is not included in the list as it does not compute score statistics. You can use `regh` (Weesie 1998) instead.

`clogit2` is a special version of `clogit`, with identical syntax, that is appropriate for conditional logit models with exactly one positive response per group (e.g., McFadden's discrete choice model). To obtain a valid estimator for the asymptotic (co)variance matrix of the estimators, one has to specify the option `cluster(`*varname*`)` with `suest combine`, where *varname* is the group identifier specified with `clogit2`. If the data are additionally clustered, e.g., the data comprise discrete choices of multiple members of households, the highest level of clustering should be specified with `suest combine`.

`suest save` is for estimation commands too complicated for `suest fit` to handle, e.g., multiple-equation models. It saves information from the last estimation command under a name of at most four characters. The estimation command should support and be invoked with the `score()` option, specifying that score statistics are computed. These score variables should be transferred to `suest save`.

The estimation command should *not* be invoked with options `robust` or `cluster` (see `suest combine`).

`suest combine` computes a simultaneous robust (sandwich) covariance matrix for the models in *name_list*. If no *name_list* is supplied, all saved models are included. If the data are clustered, this should be specified during the generation of the simultaneous variance–covariance matrix, not during estimation.

The within-model covariance matrix is identical to what would have been obtained by specifying a robust/cluster option during estimation. `suest`, however, also estimates the between-model covariances of parameter estimates.

`suest combine` posts its results just like a proper estimation command. Thus, after `suest combine`, one may use `test` and `testnl` to test within model and between model hypotheses, `vce` to display the covariance matrix, etc.

`suest` typed without arguments after `suest combine` replays the results, just like other estimation commands.

Utility commands

`suest clear` drops the objects (matrices, variables, including score statistics) associated with all models.

`suest drop` drops the objects (matrices, variables, including score statistics) associated with the named models.

`suest query` lists the names of the saved models.

Options

`cluster(`*varname*`)` specifies that the observations are independent across groups (clusters) but not necessarily within groups. *varname* specifies to which group each observation belongs; e.g., `cluster(personid)` in data with repeated observations on individuals. `cluster()` affects the estimated standard errors and variance–covariance matrix of the estimators (VCE), but not the estimated coefficients; see [U] **23.11 Obtaining robust variance estimates**.

`level(#)` specifies the confidence level, in percent, for confidence intervals. The default is level(95) or as set by `set level`.

Saved Results

Technically, `suest combine` is mostly a wrapper for the invocation of `_robust` with a properly defined information matrix D and with the concatenated score statistics u, with out-of-sample values set to 0. The other `suest` keyword commands store, retrieve, and drop information in a number of global objects:

Scalars	
`SU_names`	*names* of saved models, separated by a blank
`SU_name`	cmd, scores, ...
`SU_wtp`	weight type
`SU_wexp`	weighting expression
Matrices	
`SU_Vname`	VCE in model *name*
`SU_bname`	b in model *name*
Variables	
`SU_Sname`	identifies sample in model *name*

`suest combine` saves in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_clust)`	number of clusters
Macros	
`e(cmd)`	`suest`
`e(title)`	title in estimation output
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(clustvar)`	name of cluster variable
`e(vcetype)`	covariance estimation method
Matrices	
`e(b)`	coefficient vector of estimates
`e(V)`	(co)variance matrix of the estimates
Functions	
`e(sample)`	marks estimation sample

Example 1

In a first example, I consider testing the assumption of "Independence of Irrelevant Alternatives" as described in the section on the Hausman test `hausman` in the *Stata Reference Manual*. The example involves how the "choice of insurance type among indemnity, prepaid, and uninsured" depends on age and gender using a multinomial logit model estimated by `mlogit`. You may want to skim through this part of the Stata manual before reading on.

```
. mlogit insure age male, nolog
 (output omitted)
. hausman, save
```

Under the IIA assumption, we would expect no systematic change in the coefficients if we exclude one of the outcomes from the analysis:

```
. mlogit insure age male if insure != "Uninsure":insure, nolog
 (output omitted)
. hausman, alleqs less cons
            ---- Coefficients ----
         |      (b)          (B)            (b-B)   sqrt(diag(V_b-V_B))
         |    Current       Prior        Difference        S.E.
---------+------------------------------------------------------------
     age |  -.0101521    -.0100251        -.0001269            .
    male |   .5144003     .5095747         .0048256      .012334
   _cons |   .2678043     .2633838         .0044205            .
---------+------------------------------------------------------------
          b = less efficient estimates obtained from mlogit.
          B = more efficient estimates obtained previously from mlogit.
Test:  Ho:  difference in coefficients not systematic
            chi2(  3) = (b-B)'[(V_b-V_B)^(-1)](b-B)
                      =     0.08
            Prob>chi2 =     0.9944
```

The Stata manual goes on comparing the full model with the model that excludes the outcome "Prepaid".

```
. mlogit insure age male if insure != "Prepaid":insure, nolog
 (output omitted)
. hausman, alleqs less cons
       ---- Coefficients ----
         |      (b)          (B)            (b-B)   sqrt(diag(V_b-V_B))
         |    Current       Prior        Difference        S.E.
---------+------------------------------------------------------------
     age |  -.0041055    -.0051925          .001087     .0021357
    male |   .4591072     .4748547        -.0157475            .
   _cons |  -1.801774    -1.756843        -.0449311     .1333464
---------+------------------------------------------------------------
            b = less efficient estimates obtained from mlogit.
            B = more efficient estimates obtained previously from mlogit.
```

```
Test:  Ho:  difference in coefficients not systematic
            chi2(  3) = (b-B)'[(V_b-V_B)^(-1)](b-B)
                      =    -0.18    chi2<0 ==> model estimated on these
                                    data fails to meet the asymptotic
                                    assumptions of the Hausman test
```

In this last example, we encounter the case that Hausman's test need not be well-defined as the property underlying the simplification in Hausman's test in comparison to the approach advocated in this article holds asymptotically only.

I will now redo the analysis above in `suest` style. `mlogit` is supported by `suest fit`. I fit the full model, and store the results under name A. Next I estimate the model excluding the outcome Uninsured and the outcome Prepaid, saving the results under B and C respectively.

```
. suest fit A: mlogit insure age male, nolog
mlogit insure age male   , score(SU_A*) nolog
Multinomial regression                          Number of obs   =        615
                                                LR chi2(4)      =       9.05
                                                Prob > chi2     =     0.0598
Log likelihood = -551.32802                     Pseudo R2       =     0.0081
------------------------------------------------------------------------------
  insure |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
Prepaid  |
     age |  -.0100251   .0060181     -1.666   0.096      -.0218204    .0017702
    male |   .5095747   .1977893      2.576   0.010       .1219148    .8972345
   _cons |   .2633838   .2787574      0.945   0.345      -.2829708    .8097383
---------+--------------------------------------------------------------------
Uninsure |
     age |  -.0051925   .0113821     -0.456   0.648       -.027501     .017116
    male |   .4748547   .3618446      1.312   0.189      -.2343477    1.184057
   _cons |  -1.756843   .5309591     -3.309   0.001      -2.797504   -.7161824
------------------------------------------------------------------------------
(Outcome insure==Indemnty is the comparison group)
. suest fit B: mlogit insure age male if insure != "Uninsure":insure
  (output omitted)
. suest fit C: mlogit insure age male if insure != "Prepaid":insure
  (output omitted)
```

To perform the `suest` analogue, I "combine" the estimation results of the models stored under A and B.

```
. suest combine A B
Simultaneous VCE                                            Obs    =     615
------------------------------------------------------------------------------
         |               Robust
         |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
APrepaid |
     age |  -.0100251   .0059403     -1.688   0.091      -.0216679    .0016176
    male |   .5095747   .1988158      2.563   0.010       .1199029    .8992465
   _cons |   .2633838    .277307      0.950   0.342       -.280128    .8068955
---------+--------------------------------------------------------------------
AUninsur |
     age |  -.0051925   .0109004     -0.476   0.634      -.0265569    .0161719
    male |   .4748547   .3677294      1.291   0.197      -.2458817    1.195591
   _cons |  -1.756843   .4971363     -3.534   0.000      -2.731212    -.782474
---------+--------------------------------------------------------------------
BPrepaid |
     age |  -.0101521   .0058988     -1.721   0.085      -.0217135    .0014094
    male |   .5144003   .1996132      2.577   0.010       .1231656    .9056351
   _cons |   .2678043   .2744018      0.976   0.329      -.2700133     .805622
------------------------------------------------------------------------------
```

The equation names in the combined estimation results are formed by prefixing the name under which a result is saved to the equation names. Thus equation `APrepaid` is equation `Prepaid` of the model saved under A. Also note that `suest` reports robust standard errors rather than the "classic" standard errors reported by default by `mlogit`, even though `suest fit : mlogit` was invoked without the robust option. The robust modification of standard errors was just applied while forming the combined (simultaneous) (co)variance matrix of the estimators.

I can now perform a generalized Hausman test whether the coefficients in equation `APrepaid` are equal to the analogous coefficients in equation `BPrepaid`. This is simply accomplished with the `test` command.

```
. test [APrepaid=BPrepaid]:_cons age male
 ( 1)  [APrepaid]_cons - [BPrepaid]_cons = 0.0
 ( 2)  [APrepaid]age - [BPrepaid]age = 0.0
 ( 3)  [APrepaid]male - [BPrepaid]male = 0.0
           chi2(  3) =    0.89
         Prob > chi2 =    0.8267
```

Note that the generalized Hausman test statistic (0.89) is different than the original Hausman test statistic (0.08). This is due to different ways in which variances are estimated; in the classic way and assuming efficiency of the full-data estimator in the original Hausman test, by the robust way in the generalized Hausman test.

In a similar way, I can obtain the Hausman test for the second case.

```
. suest combine A C
  (output omitted)
. test [AUninsur=CUninsur]:_cons age male
 ( 1)  [AUninsur]_cons - [CUninsur]_cons = 0.0
 ( 2)  [AUninsur]age - [CUninsur]age = 0.0
 ( 3)  [AUninsur]male - [CUninsur]male = 0.0
           chi2(  3) =    1.49
         Prob > chi2 =    0.6843
```

While the original Hausman test was undefined due to the nondefiniteness of the variance difference of the (co)variance matrices of the two estimators, the generalized Hausman test obtained via `suest` is quite well-behaved; it is never undefined.

I can now perform an additional test that is not possible within the standard Hausman framework. With `suest`, it is possible to test *simultaneously* the two hypotheses tested "univariately" above, i.e., I want to test that the *three* estimators, corresponding with the models A, B, and C, have the same probability limit ("are equal"). To perform this test, one has to generate the simultaneous (co)variance matrix of the three estimators.

```
. suest combine A B C
Simultaneous VCE                                              Obs    =     615
 ------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
APrepaid     |
         age |  -.0100251   .0059403     -1.688   0.091      -.0216679    .0016176
        male |   .5095747   .1988158      2.563   0.010       .1199029    .8992465
       _cons |   .2633838    .277307      0.950   0.342       -.280128    .8068955
-------------+----------------------------------------------------------------
AUninsur     |
         age |  -.0051925   .0109004     -0.476   0.634      -.0265569    .0161719
        male |   .4748547   .3677294      1.291   0.197      -.2458817    1.195591
       _cons |  -1.756843   .4971363     -3.534   0.000      -2.731212    -.782474
-------------+----------------------------------------------------------------
BPrepaid     |
         age |  -.0101521   .0058988     -1.721   0.085      -.0217135    .0014094
        male |   .5144003   .1996132      2.577   0.010       .1231656    .9056351
       _cons |   .2678043   .2744018      0.976   0.329      -.2700133     .805622
-------------+----------------------------------------------------------------
CUninsur     |
         age |  -.0041055   .0111185     -0.369   0.712      -.0258974    .0176865
        male |   .4591072   .3601307      1.275   0.202      -.2467359     1.16495
       _cons |  -1.801774   .5226351     -3.447   0.001       -2.82612   -.7774283
------------------------------------------------------------------------------
. test [APrepaid=BPrepaid]:_cons age male
  (output omitted)
. test [AUninsur=CUninsur]:_cons age male, accumulate
 ( 1)  [APrepaid]_cons - [BPrepaid]_cons = 0.0
 ( 2)  [APrepaid]age - [BPrepaid]age = 0.0
 ( 3)  [APrepaid]male - [BPrepaid]male = 0.0
 ( 4)  [AUninsur]_cons - [CUninsur]_cons = 0.0
 ( 5)  [AUninsur]age - [CUninsur]age = 0.0
 ( 6)  [AUninsur]male - [CUninsur]male = 0.0
           chi2(  6) =    1.95
         Prob > chi2 =    0.9239
```

I conclude that there is no evidence that the IIA assumption is violated in any way. Then again, as also discussed in the section in the manual, the model itself seems to fit so poorly that one has to be careful in deriving any substantive implications.

I now consider the case that the observations are clustered within households, indicated by the variable `hhid` (this was not true in the data used above; I constructed this just for illustrating this important aspect of the generalization of the Hausman test enabled by `suest`). Then the "classic" McFadden–Hausman test is no longer appropriate as neither of the estimators is efficient. The test via `suest` as proposed in this paper, however, remains valid, and can be obtained with very little change to the example above. The only thing that has to be changed is that a `cluster(subject)` option has to be specified with the `suest combine` command:

```
. suest combine A B, cluster(hhid)
 Simultaneous VCE                                              Obs    =     615
                           (standard errors adjusted for clustering on hhid)
------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
APrepaid     |
         age |  -.0100251   .0060379     -1.660   0.097      -.0218592    .0018089
        male |   .5095747   .2028564      2.512   0.012       .1119834     .907166
       _cons |   .2633838   .2850523      0.924   0.355      -.2953084     .822076
-------------+----------------------------------------------------------------
AUninsur     |
         age |  -.0051925   .0106246     -0.489   0.625      -.0260163    .0156313
        male |   .4748547   .3746916      1.267   0.205      -.2595274    1.209237
       _cons |  -1.756843   .4847787     -3.624   0.000      -2.706992   -.8066943
-------------+----------------------------------------------------------------
BPrepaid     |
         age |  -.0101521   .0059962     -1.693   0.090      -.0219044    .0016003
        male |   .5144003     .20382      2.524   0.012       .1149206    .9138801
       _cons |   .2678043    .282348      0.948   0.343      -.2855877    .8211963
------------------------------------------------------------------------------

. test [APrepaid=BPrepaid]:_cons age male
 ( 1)  [APrepaid]_cons - [BPrepaid]_cons = 0.0
 ( 2)  [APrepaid]age - [BPrepaid]age = 0.0
 ( 3)  [APrepaid]male - [BPrepaid]male = 0.0

           chi2(  3) =    0.87
         Prob > chi2 =    0.8321
```

A note on equation names

We have seen that `suest` prefixes the save-name to the equation name. If an estimation command does not generate an equation name, that is, uses equation name `_` (underscore), the underscore is not included in the extended name. Also, it is possible that prefixing the save-name to an equation name yields a string longer than 8 characters, and hence needs to be truncated. It is possible that equation names then become identical. In this, hopefully exceptional, case `suest` simply numbers the equations, prefixed with the save-name.

The assumption of "Independence of Irrelevant Alternatives" also applies to the more general class of "conditional logit models", such as McFadden's discrete choice model. The Stata command `clogit` that fits these models does not compute score statistics and hence can not be directly applied for the `suest` approach. For the case with one-positive-response-per-group, the scores are

$$u_i = \sum_{j=1}^{n_i} (y_{ij} - \widehat{\pi}_{ij}) x_{ij}$$

where n_i is the number of observations in group i, $y_{ij} \in \{0, 1\}$ is the response of "group" i with $y_{i+} = 1$, and x_{ij} is the vector of covariates for alternative j within group i. This implies that the robust estimator of variance can be obtained as if the alternatives are "clustered" within groups. A command `clogit2` is distributed with `suest` that wraps around `clogit`, verifies that the one-response-per-group restriction holds, and returns the score statistic via the option `score()`. `clogit2` displays a reminder that `suest combine` should be invoked with a cluster option. If the groups are independent, clustering should be at the level of the group. If the groups themselves are clustered, for instance, in the case of decisions by multiple members of a household, clustering is indeed at this higher level, i.e., households, not individuals. The following code fragment illustrates the IIA test for independent individuals.

```
. suest fit A: clogit2 transp money time, group(respnr)
. suest fit B: clogit2 transp money time, group(respnr), if code~="car":lcode
. suest combine, cluster(respnr)
. test [A=B]
```

Example 2

In this second example, I consider an ordinal probit analysis. I want to test the hypotheses that the coefficients (for simplicity, I ignore the cut-points) are invariant under joining extreme response categories (1,2) and (4,5) of a Likert scale, i.e., a five-category scale in which 3 indicates a neutral response. (A more substantial but also more cumbersome way to deal with this problem is to modify the ordinal probit model so that subjects are assumed to randomize between categories 1 and 2, and similarly between categories 4 and 5, if their latent variables fall below threshold 2 or above threshold 4. Such a modification of ordinal models is an analogue to how psychometric models for multiple choice data are constructed from multinomial models in order to incorporate "guessing.")

I illustrate the `suest` approach to these problems with an ordinal probit analysis of the repair record on the price and origin of cars in the automobile data distributed with Stata. Since `oprobit` is supported by `suest fit`, I can invoke a `suest fit` command to fit the full model:

```
. suest fit A: oprobit rep78 price for, nolog
oprobit rep78 price foreign   , score(SU_A*) nolog
Ordered probit estimates                          Number of obs   =         69
                                                  LR chi2(2)      =      29.42
                                                  Prob > chi2     =     0.0000
Log likelihood = -78.982445                       Pseudo R2       =     0.1570
------------------------------------------------------------------------------
       rep78 |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
       price |   6.55e-06   .0000459      0.143   0.886      -.0000833    .0000964
     foreign |   1.721754   .3311814      5.199   0.000        1.07265    2.370857
-------------+----------------------------------------------------------------
       _cut1 |  -1.702593    .421363             (Ancillary parameters)
       _cut2 |  -.7864638   .3492921
       _cut3 |   .7703836   .3452181
       _cut4 |    1.88739   .3944205
------------------------------------------------------------------------------
```

Next, I have to recode the dependent variable by collapsing values 1 and 2 into 2, and the values of 4 and 5 into 4.

```
. gen rrep78 = rep78
. recode rrep78  1=2  5=4
```

and invoke the `oprobit` model with the recoded dependent variables:

```
. suest fit B: oprobit rrep78 price for, nolog
oprobit rrep78 price foreign   , score(SU_B*) nolog
Ordered probit estimates                          Number of obs   =         69
                                                  LR chi2(2)      =      26.67
                                                  Prob > chi2     =     0.0000
Log likelihood = -56.107058                       Pseudo R2       =     0.1920
------------------------------------------------------------------------------
      rrep78 |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
       price |   .0000105   .0000487      0.215   0.830      -.0000851     .000106
     foreign |   1.834473   .3897461      4.707   0.000       1.070585    2.598361
-------------+----------------------------------------------------------------
       _cut1 |  -.7554451   .3635206             (Ancillary parameters)
       _cut2 |   .8140208   .3622057
------------------------------------------------------------------------------
```

I can now estimate the simultaneous (co)variance matrix and perform a test that the coefficients of the predictor variables (`price` and `foreign`) are the same in the two models:

```
. suest comb A B
Simultaneous VCE                                          Obs    =      69
------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
ALP          |
       price |   6.55e-06   .0000362      0.181   0.856      -.0000644    .0000775
     foreign |   1.721754   .3249783      5.298   0.000       1.084808    2.358699
-------------+----------------------------------------------------------------
Acut1        |
       _cons |  -1.702593   .4310119     -3.950   0.000      -2.547361   -.8578251
------------------------------------------------------------------------------
```

```
---------+--------------------------------------------------------------------
Acut2    |
   _cons |  -.7864638   .3336287     -2.357   0.018      -1.440364   -.1325636
---------+--------------------------------------------------------------------
Acut3    |
   _cons |   .7703836   .3387772      2.274   0.023       .1063924    1.434375
---------+--------------------------------------------------------------------
Acut4    |
   _cons |    1.88739   .4151565      4.546   0.000       1.073698    2.701082
---------+--------------------------------------------------------------------
BLP      |
   price |   .0000105   .0000393      0.266   0.790      -.0000666    .0000875
 foreign |   1.834473   .3786316      4.845   0.000       1.092369    2.576577
---------+--------------------------------------------------------------------
Bcut1    |
   _cons |  -.7554451   .3387186     -2.230   0.026      -1.419321   -.0915689
---------+--------------------------------------------------------------------
Bcut2    |
   _cons |   .8140208   .3416695      2.382   0.017       .1443609    1.483681
------------------------------------------------------------------------------

. test [ALP=BLP]
( 1)  [ALP]price - [BLP]price = 0.0
( 2)  [ALP]foreign - [BLP]foreign = 0.0
           chi2(  2) =    0.32
         Prob > chi2 =    0.8505
```

and so I cannot reject the null-hypothesis that these coefficients are the same.

Note that `oprobit` and `ologit` store their estimation results in a nonstandard way; namely, without equation names. `suest` actually modifies this; it adds equation name `LP` to the coefficients of the covariates, and renames the parameter `_cutx` to `cutx:_cons`. As a consequence, `predict` for `oprobit` and `ologit` may fail after invoking `suest`.

Example 3

In this third example, I test that analyses using `logit` and `probit` regression models "yield the same results". I employ again the automobile data and want to predict whether a car is foreign (i.e., non-US built) from its price and repair record. I first estimate a logit model, and request that `suest` store the results under the name `LOGT`.

```
. suest LOGT: logit foreign price rep78 , nolog
logit foreign price rep78 , nolog score(SU_LOGT)
Logit estimates                                   Number of obs   =         69
                                                  LR chi2(2)      =      29.37
                                                  Prob > chi2     =     0.0000
Log likelihood = -27.714924                       Pseudo R2       =     0.3464
------------------------------------------------------------------------------
 foreign |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   price |   6.26e-06   .0001325      0.047   0.962      -.0002533    .0002659
   rep78 |   1.970872   .4800167      4.106   0.000       1.030057    2.911687
   _cons |  -8.088338   2.082646     -3.884   0.000      -12.17025   -4.006427
------------------------------------------------------------------------------
```

Similarly, I fit a probit model, and store the results under the name `PROB`:

```
. suest fit PROB: probit foreign price rep78, nolog
probit foreign price rep78 , nolog score(SU_PROB)
Probit estimates                                  Number of obs   =         69
                                                  LR chi2(2)      =      29.55
                                                  Prob > chi2     =     0.0000
Log likelihood = -27.623517                       Pseudo R2       =     0.3485
------------------------------------------------------------------------------
 foreign |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   price |  -1.57e-06   .0000756     -0.021   0.983      -.0001497    .0001466
   rep78 |   1.147768   .2542272      4.515   0.000       .6494914    1.646044
   _cons |  -4.684726   1.102448     -4.249   0.000      -6.845483   -2.523968
------------------------------------------------------------------------------
```

Before we can perform tests, I request `suest` to estimate the simultaneous (co)variance matrix of the logit and probit estimators:

```
. suest combine LOGT PROB
Simultaneous VCE                                              Obs    =      69
------------------------------------------------------------------------------
         |               Robust
         |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
LOGT     |
   price |   6.26e-06   .0000981      0.064   0.949      -.0001859    .0001984
   rep78 |   1.970872   .4793285      4.112   0.000       1.031406    2.910339
   _cons |  -8.088338   1.994391     -4.056   0.000      -11.99727   -4.179402
---------+--------------------------------------------------------------------
PROB     |
   price |  -1.57e-06   .0000548     -0.029   0.977      -.0001091    .0001059
   rep78 |   1.147768   .2518236      4.558   0.000       .6542024    1.641333
   _cons |  -4.684726   1.046617     -4.476   0.000      -6.736057   -2.633394
------------------------------------------------------------------------------
```

Note that the simultaneous estimates for standard errors are slightly different from the "separate" estimates. This is caused by the use of the *robust* (sandwiched) estimator for standard errors; they are *identical* to the standard errors that Stata would have produced if the robust option was specified with the logit and probit models.

We are now able to inspect the full (co)variance matrix of the logit and probit estimator.

```
. vce
         |     LOGT:                          PROB:
         |    price    rep78    _cons    price    rep78    _cons
---------+-----------------------------------------------------
LOGT     |
   price |  9.6e-09
   rep78 |  3.1e-06  .229756
   _cons | -.000084 -.876035   3.9776
PROB     |
   price |  5.4e-09  1.9e-06 -.000046  3.0e-09
   rep78 |  1.3e-06  .120655 -.456556  8.4e-07  .063415
   _cons | -.000045 -.459068  2.08531 -.000025 -.239423  1.09541
```

Inspecting the simultaneous correlation matrix may be more revealing.

```
. vce, rho
         |     LOGT:                          PROB:
         |    price    rep78    _cons    price    rep78    _cons
---------+-----------------------------------------------------
LOGT     |
   price |   1.0000
   rep78 |   0.0660   1.0000
   _cons |  -0.4274  -0.9164   1.0000
PROB     |
   price |   0.9957   0.0710  -0.4223   1.0000
   rep78 |   0.0543   0.9996  -0.9090   0.0607   1.0000
   _cons |  -0.4371  -0.9151   0.9990  -0.4357  -0.9084   1.0000
```

The correlations between the (coefficients associated with) the same predictor variables in the logit and probit models are very, very high indeed. Thus, one should actually be cautious of the p-value of the Wald-test discussed below, especially in the case of nonlinear hypotheses.

I can test whether the coefficients of the logit model are equal to the coefficients from the probit model:

```
. test [LOGT]price=[PROB]price, notest
. test [LOGT]rep78=[PROB]rep78, acc
 ( 1)  [LOGT]price - [PROB]price = 0.0
 ( 2)  [LOGT]rep78 - [PROB]rep78 = 0.0
           chi2(  2) =   13.07
         Prob > chi2 =    0.0015
```

This hypothesis has to be strongly rejected. But, remember that the logit and probit are on a different scale. Thus, at best, one can expect that the coefficients of `price` and `rep78` in the logit model are proportional to the coefficients in the probit model. Proportionality, however, is a nonlinear hypothesis, which cannot be tested using `test`. Stata can perform nonlinear Wald-type tests using the `testnl` command:

```
. testnl [LOGT]price/[LOGT]rep78 = [PROB]price/[PROB]rep78
```

```
(1)  [LOGT]price/[LOGT]rep78 = [PROB]price/[PROB]rep78
             chi2(1) =        0.85
         Prob > chi2 =        0.3565
```

Since nonlinear Wald tests are sensitive to specification, it may be appropriate to formulate the hypothesis "in a more linear way,"

```
. testnl [LOGT]price*[PROB]rep78 = [PROB]price*[LOGT]rep78
(1)  [LOGT]price*[PROB]rep78 = [PROB]price*[LOGT]rep78
             chi2(1) =        0.83
         Prob > chi2 =        0.3628
```

I conclude that the null hypotheses that the coefficients from the logit model are proportional to the coefficients from the probit model cannot be rejected.

Applications to estimation

The simultaneous distribution of estimators derived above are useful for two-stage estimation as well. Consider the case that we have a number of inefficient estimators, that, under some model, we estimate "the same thing," i.e., have the same probability limit. We can "pool" the inefficient estimators to come up with an estimator that is more efficient (has smaller variance) than any of the original estimators. One way to do this is to compute the simple GLS estimator for the associated linear model which in essence is an estimator b_{pooled}, that is, a weighted average of the separate estimators $b_1, \ldots, b_k$, with the weighting matrices defined in terms of the variances and covariances of the estimators. This can be accomplished, without additional matrix programming, with the combination of the `suest` command and my `linest` command for two-stage linear-constrained estimation (Weesie 1999).

A second application to impose cross-model constraints may be instructive as well. Petersen (1988) suggested modeling continuous-space survival-time problems in terms of the marginal distribution, F, of survival time and the conditional distribution, G, of the new state, given the old state and the time until the transition. Censored observations are treated in the normal way in the marginal distribution of survival time and ignored in the transition distribution. If F and G depend on different parameters, the likelihood "factors" between the two subproblems, and hence can be estimated separately.

To test cross-subproblem hypotheses ("does educational achievement have the same effect on the hazard rate as on the new occupational prestige," or, maybe more accurately, "are the effects of variables $x_1, x_2, \ldots$ on the hazard rate for mobility proportional to the effects of these variables on the new occupational prestige;" this is a formal representation of the substantial hypothesis that "mobility timing" and "distance of mobility" can be "explained" in the same way) one needs the full (co)variance matrix pertaining to the two subproblems. The simultaneous (co)variance matrix can be estimated using the `suest` command given above. (The fact that one of the models is a conditional model rather than a marginal model does not affect the validity of the derivation underlying `suest`.)

If F and G are parameterized with partially overlapping sets of parameters, the likelihood no longer factors over the subproblems, and so the method outlined by Petersen needs modification. The theory of *two-stage equality constrained estimation* provides for a relatively simple modification (Gourieroux and Monfort 1989, Ch. 10). In the first stage of this two-stage estimator, the cross-subproblem equality constraints are ignored, and the subproblems can be treated separately. Then the simultaneous (co)variance matrix of the two estimators is estimated using `suest`. Then, in the second stage, a GLS estimator for the cross-subproblem equality constraints is imposed (see Weesie (1999) for the Stata command `linest`). According to statistical theory (e.g., Gourieroux and Monfort 1989), this two-stage estimator is (asymptotically) equivalent to a truly one-stage constrained simultaneous estimator.

Acknowledgments

This research is supported by Pionier grant PGS 50–370 by the Netherlands Organization for Scientific Research (NWO). It has benefit greatly from stimulating discussions with and suggestions by, in alphabetic order, Bill Gould, Jerry Hausman, Chris Snijders, Hal White, and Vince Wiggins.

References

Clogg, C. C., E. Petkova, and A.Haritou. 1995. Statistical methods for comparing regression coefficients between models. *American Journal of Sociology* 100: 1261–1293.

Gourieroux, H. and A. Monfort. 1989. *Statistics and Econometric Models* vol. 2. New York: Springer.

Hausman, J. 1978. Specification tests in econometrics. *Econometrica* 46: 1251–1271.

Hausman, J. and D. McFadden. 1984. Specification tests in econometrics. *Econometrica* 52: 1219–1240.

Petersen, T. 1988. Analyzing change over time in a continuous dependent variable: specification and estimation of continuous state space hazard rate models. *Sociological Methodology*.

Rogers, W. M. 1993. sg17: Regression standard errors in clustered samples. *Stata Technical Bulletin* 13: 19–23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 88–94.

Weesie, J. 1998. sg77: Regression analysis with multiplicative heteroskedasticity. *Stata Technical Bulletin* 2: 28–32. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 204–210.

——. 1999. sg100: Two-stage linear constrained estimation. *Stata Technical Bulletin* 47: 24–30. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 217–225.

White, H. 1982. Maximum-likelihood estimation of misspecified models. *Econometrica* 50: 1–25.

——. 1994. *Estimation, Inference and Specification Analysis*. Cambridge: Cambridge University Press.

sg122 Truncated regression

Ronna Cong, Stata Corporation, rcong@stata.com

Syntax

`truncreg` has the following syntax:

`truncreg` *depvar varlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` `ll(`*varname* | *#*`)` `ul(`*varname* | *#*`)` `noconstant`

`level(`*#*`)` `marginal` `at(`*matname*`)` `robust` `cluster(`*varname*`)` *maximize_options*]

`aweights`, `pweights` and `fweights` are allowed; see [U] **18.1.6 weight**.

`truncreg` shares the features of all estimation commands; see [U] **23 Estimation and post-estimation commands**.

Syntax for predict

`predict` [*type*] *newvarname* [`if` *exp*] [`in` *range*] [, *statistic*]

where *statistic* is

`xb`	$\mathbf{x}_j\mathbf{b}$, fitted values (the default)
`pr(`a`,`b`)`	$\Pr(a < y_j < b)$
`e(`a`,`b`)`	$E(y_j \mid a < y_j < b)$
`stdp`	standard error of the prediction
`stdf`	standard error of the forecast

where a and b may be numbers or variables; a equal to '.' means $-\infty$; b equal to '.' means $+\infty$.

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Description

`truncreg` estimates a regression model of *depvar* on *varlist* from a sample drawn from a restricted part of the population. Under the normality assumption for the whole population, the error terms in the truncated regression model are truncated normally distributed. The truncated normal distribution is a normal distribution that has been scaled upward so that the distribution integrates to one over the restricted range.

Options

`ll(`*varname* | *#*`)` and `ul(`*varname* | *#*`)` indicate the truncation points. You may specify one or both. `ll()` indicates the lower limit for left truncation; `ul()` indicates the upper limit for right truncation. Observations with *depvar* $\leq$ `ll()` are left truncated; observations with *depvar* $\geq$ `ul()` are right truncated; remaining observations are not truncated. See [R] **tobit** for a more detailed description.

`noconstant` suppresses the constant term (intercept) in the model.

`level(#)` specifies the confidence level, in percent, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **23.5 Specifying the width of confidence intervals**.

`marginal` estimates the marginal effects in the model in the subpopulation. `marginal` may be specified when the model is estimated or when results are redisplayed.

`at(`*matname*`)` specifies the point around which the marginal effect is to be estimated. The default is to estimate the effect around the mean of the independent variables. If there are k independent variables, *matname* should be $1 \times k$. `at()` may be specified when the model is estimated or when results are redisplayed.

`robust` specifies that the Huber/White/sandwich estimator of variance is to be used in place of the conventional MLE variance estimator. `robust` combined with `cluster()` further allows observations which are not independent within cluster (although they must be independent between clusters).

If you specify `pweights`, `robust` is implied; see [U] **23.11 Obtaining robust variance estimates**.

`cluster(`*varname*`)` specifies that the observations are independent across groups. *varname* specifies to which group each observation belongs. `cluster()` can be used with `pweights` to produce estimates for unstratified cluster-sampled data.

`cluster()` implies `robust`; specifying `robust cluster()` is equivalent to typing `cluster()` by itself.

maximize_options control the maximization process; see [R] **maximize**. Use the `trace` option to view parameter convergence. Use the `ltol(#)` option to relax the convergence criterion; default is 1e−6 during specification searches.

Options for predict

See [R] **tobit** for a detailed description.

Remarks

Remarks are presented under the headings

Truncated regression
Marginal effects

Truncated regression

Truncated regression estimates a model of a dependent variable on independent variables from a restricted part of a population. Truncation is essentially a characteristic of the distribution from which the sample data are drawn. If x has a normal distribution with mean μ and standard deviation σ, then the density of the truncated normal distribution is

$$f\left(x \mid a < x < b\right) = \frac{f(x)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)} = \frac{\frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}$$

where ϕ and Φ are the density and distribution functions of the standard normal distribution.

Compared with the mean of the untruncated variable, the mean of the truncated variable is greater if the truncation is from below, the mean of the truncated variable is smaller if the truncation is from above. Moreover, truncation reduces the variance compared with the variance in the untruncated distribution.

Example

We will demonstrate `truncreg` using a subset of the Mroz data distributed with Berndt (1991). This dataset contains 753 observations on women's labor supply. Our subsample is of 250 observations, with 150 market laborers and 100 nonmarket laborers.

```
. use labor, clear
. describe
Contains data from labor.dta
  obs:           250
 vars:             6
 size:         7,000 (99.3% of memory free)
-------------------------------------------------------------------------------
   1. lfp       float  %9.0g                 1 if woman worked in 1975
   2. whrs      float  %9.0g                 Wife´s hours of work
   3. kl6       float  %9.0g                 # of children younger than 6
   4. k618      float  %9.0g                 # of children between 6 and 18
   5. wa        float  %9.0g                 wife´s age
   6. we        float  %9.0g                 wife´s educational attainment
-------------------------------------------------------------------------------
Sorted by:
```

```
. summarize
Variable |     Obs        Mean   Std. Dev.       Min        Max
---------+--------------------------------------------------------
     lfp |     250          .6   .4908807          0          1
    whrs |     250      799.84   915.6035          0       4950
     kl6 |     250        .236   .5112234          0          3
    k618 |     250       1.364   1.370774          0          8
      wa |     250       42.92   8.426483         30         60
      we |     250      12.352   2.164912          5         17
```

We first perform an ordinary least squares estimation on the market laborers, i.e., working hours > 0.

```
. regress whrs kl6 k618 wa we if whrs >0
  Source |       SS       df       MS                  Number of obs =     150
---------+------------------------------                F(  4,   145) =    2.80
   Model |  7326995.15     4  1831748.79                Prob > F      =  0.0281
Residual |  94793104.2   145  653745.546                R-squared     =  0.0717
---------+------------------------------                Adj R-squared =  0.0461
   Total |   102120099   149  685369.794                Root MSE      =  808.55
------------------------------------------------------------------------------
    whrs |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     kl6 |  -421.4822   167.9734     -2.509   0.013      -753.4748   -89.48953
    k618 |  -104.4571   54.18616     -1.928   0.056      -211.5538    2.639666
      wa |  -4.784917   9.690502     -0.494   0.622       -23.9378    14.36797
      we |   9.353195   31.23793      0.299   0.765      -52.38731     71.0937
   _cons |   1629.817   615.1301      2.650   0.009       414.0371    2845.596
------------------------------------------------------------------------------
```

Now, we use `truncreg` to perform truncated regression with truncation from below zero.

```
. truncreg whrs kl6 k618 wa we, ll(0)
Note:        100 obs. are truncated

Iteration 0:   log likelihood = -1205.6992
Iteration 1:   log likelihood = -1200.9872
Iteration 2:   log likelihood = -1200.9159
Iteration 3:   log likelihood = -1200.9157
Iteration 4:   log likelihood = -1200.9157

Truncated regression
Limit:   lower =          0                           Number of obs =    150
         upper =       +inf                           Wald chi2(4)  =  10.05
Log likelihood = -1200.9157                           Prob > chi2   = 0.0395
------------------------------------------------------------------------------
    whrs |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
eq1      |
     kl6 |  -803.0042   321.3614     -2.499   0.012      -1432.861   -173.1474
    k618 |   -172.875   88.72898     -1.948   0.051      -346.7806    1.030578
      wa |  -8.821122   14.36848     -0.614   0.539      -36.98283    19.34059
      we |   16.52873   46.50375      0.355   0.722      -74.61695    107.6744
   _cons |    1586.26    912.355      1.739   0.082      -201.9233    3374.442
---------+--------------------------------------------------------------------
sigma    |
   _cons |   983.7262   94.44303     10.416   0.000       798.6213    1168.831
------------------------------------------------------------------------------
```

(*Continued on next page*)

If we assume that our data were censored, the tobit model is

```
. tobit whrs kl6 k618 wa we, ll(0)
Tobit estimates                                   Number of obs   =        250
                                                  LR chi2(4)      =      23.03
                                                  Prob > chi2     =     0.0001
Log likelihood = -1367.0903                       Pseudo R2       =     0.0084
------------------------------------------------------------------------------
    whrs |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     kl6 |  -827.7657   214.7407     -3.855   0.000      -1250.731   -404.8009
    k618 |  -140.0192   74.22303     -1.886   0.060      -286.2128    6.174541
      wa |  -24.97919   13.25639     -1.884   0.061      -51.08969    1.131316
      we |   103.6896   41.82393      2.479   0.014       21.31093    186.0683
   _cons |   589.0001   841.5467      0.700   0.485      -1068.556    2246.556
---------+--------------------------------------------------------------------
     _se |   1309.909   82.73335              (Ancillary parameter)
------------------------------------------------------------------------------

Obs. summary:       100 left-censored observations at whrs<=0
                    150 uncensored observations
```

Technical Note

Whether truncated regression is more appropriate than the ordinary least squares estimation depends on the purpose of that estimation. If we are interested in the mean of wife's working hours conditional on the subsample of market laborers, least squares estimation is appropriate. However, if it is the mean of wife's working hours regardless of market or nonmarket labor status that we are interested in, least squares estimates could be seriously misleading.

It should be understood that truncation and censoring are different concepts. A sample has been censored if no observations have been systematically excluded, but some of the information contained in them has been suppressed. In a truncated distribution, only the part of the distribution above (or below, or between) the truncation point(s) is relevant to our computations. We need to scale it up by the probability that an observation falls in the range that interests us to make the distribution integrate to one. The censored distribution used by tobit, however, is a mixture of discrete and continuous distributions. Instead of rescaling over the observable range, we simply assign the full probability from the censored region(s) to the censoring point(s). The truncated regression model is sometimes less well behaved than the tobit model. Davidson and MacKinnon (1993) provide an example where truncation results in more inconsistency than censoring.

Marginal effects

The marginal effects in the truncation model in the subpopulation can be obtained by specifying the `marginal` option. `at(`*matname*`)` specifies the points around which the marginal effects are to be estimated. The default is to estimate the effect around the mean of the independent variables.

Example

The marginal effects around the mean of the independent variables conditional on subpopulation for our previous truncated regression is

```
. truncreg whrs kl6 k618 wa we, marginal
Marginal Effects
Limit:   lower =          0                       Number of obs =    150
         upper =       +inf                       Wald chi2(4)  =  10.05
Log likelihood = -1200.9157                       Prob > chi2   = 0.0395
------------------------------------------------------------------------------
variable |      dF/dx   Std. Err.      z    P>|z|     X_at   [    95% C.I.   ]
---------+--------------------------------------------------------------------
     kl6 |  -521.9979    208.903    -2.50   0.012   .173333   -931.44 -112.556
    k618 |  -112.3785   57.67883    -1.95   0.051   1.31333  -225.427  .669934
      wa |  -5.734226   9.340322    -0.61   0.539   42.7867  -24.0409  12.5725
      we |    10.7446   30.23005     0.36   0.722     12.64  -48.5052  69.9944
   _cons |   1031.158   593.0821     1.74   0.082         1  -131.262  2193.58
------------------------------------------------------------------------------
```

The marginal effects around the mean of independent variables of the entire population may be obtained using the `at()` option. Notice that the `marginal` and the `at()` options may be specified when results are redisplayed.

```
. mat accum junk=kl6 k618 wa we, means(B) noc
. truncreg, marginal at(B)
```

```
Marginal Effects
Limit:   lower =          0                          Number of obs =    150
         upper =       +inf                          Wald chi2(4)  =  10.05
Log likelihood = -1200.9157                          Prob > chi2   = 0.0395
------------------------------------------------------------------------------
variable |      dF/dx   Std. Err.      z    P>|z|     X_at    [    95% C.I.   ]
---------+--------------------------------------------------------------------
     kl6 |  -201.7607    80.7444    -2.50   0.012      .236  -360.017 -43.5046
    k618 |  -43.43611    22.2938    -1.95   0.051     1.364  -87.1312   .25894
      wa |  -2.216371   3.610186    -0.61   0.539     42.92  -9.29221  4.85946
      we |   4.152964   11.68441     0.36   0.722    12.352  -18.7481   27.054
------------------------------------------------------------------------------
```

Technical Note

Whether the marginal effect or the coefficient itself is of interest depends on the purpose of the study. If it is the subpopulation the analysis is confined to, then marginal effects are of interest. Otherwise, it is the coefficients themselves that are of interest.

Saved Results

`truncreg` saves in `e()`:

Scalars

`e(N)`	number of observations	`e(N_bf)`	number of observations before truncation
`e(ll)`	log likelihood	`e(ic)`	number of iterations
`e(df_m)`	model degrees of freedom	`e(N_clust)`	number of clusters
`e(llopt)`	lower limit	`e(ulopt)`	upper limit
`e(chi2)`	χ^2	`e(p)`	p-value for χ^2 test
`e(sigma)`	sigma		

Macros

`e(cmd)`	`truncreg`	`e(clustvar)`	name of cluster variable
`e(vcetype)`	covariance estimation method	`e(chi2type)`	`Wald` or `LR`; type of model χ^2
`e(wtype)`	weight type	`e(wexp)`	weight expression
`e(depvar)`	name of dependent variable	`e(predict)`	program used to implement `predict`
`e(opt)`	type of optimization	`e(user)`	name of likelihood-evaluator program
`e(title)`	truncated regression		

Matrices

`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators
`e(dfdx)`	marginal effects vector	`e(V_dfdx)`	variance–covariance matrix of the marginal effects
`e(means)`	means of independent variables	`e(at)`	points for calculating marginal effects
`e(ilog)`	iteration log (up to 20 iterations)		

Functions

`e(sample)`	marks estimation sample

Methods and Formulas

`truncreg` is implemented as an ado-file. Greene (2000, 897–905) and Davidson and MacKinnon (1993, 534–537) provide introductions to the truncated regression model.

Let $\mathbf{y} = \mathbf{X}\beta + \epsilon$ be the model. $\mathbf{y}$ represents continuous outcomes either observed or not observed. Our model assumes $\epsilon \sim N(\mathbf{0}, \sigma^2\mathbf{I})$.

Let a be the lower limit, b be the upper limit. The log likelihood is

$$L = -\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\beta)^2 - \sum_{i=1}^{n}\log\left[\Phi\left(\frac{b-\mathbf{x}_i\beta}{\sigma}\right) - \Phi\left(\frac{a-\mathbf{x}_i\beta}{\sigma}\right)\right]$$

The marginal effects at $\mathbf{x}_i$ with truncation from below are

$$\frac{\partial E\left[y_i \mid y_i > a\right]}{\partial \mathbf{x}_i} = \beta\left(1 - \lambda_i^2 + \alpha_i\lambda_i\right)$$

where

$$\alpha_i = (a - \mathbf{x}_i\beta)/\sigma$$
$$\lambda_i = \phi(\alpha_i)/[1 - \Phi(\alpha_i)]$$

The marginal effects at $\mathbf{x}_i$ with truncation from above are

$$\frac{\partial E[y_i \mid y_i < b]}{\partial \mathbf{x}_i} = \beta\left(1 - \lambda_i^2 + \alpha_i\lambda_i\right)$$

where

$$\alpha_i = (b - \mathbf{x}_i\beta)/\sigma$$
$$\lambda_i = -\phi(\alpha_i)/\Phi(\alpha_i)$$

The marginal effects at $\mathbf{x}_i$ with truncation from both above and below are

$$\frac{\partial E[y_i \mid a < y_i < b]}{\partial \mathbf{x}_i} = \beta\left(1 - \lambda_i^2 + \alpha_{i2}\lambda_i - \frac{(b-a)\phi(\alpha_{i1})}{\sigma\left(\Phi(\alpha_{i2}) - \Phi(\alpha_{i1})\right)}\right)$$

where

$$\alpha_{i1} = (a - \mathbf{x}_i\beta)/\sigma$$
$$\alpha_{i2} = (b - \mathbf{x}_i\beta)/\sigma$$
$$\lambda_i = \frac{\phi(\alpha_{i2}) - \phi(\alpha_{i1})}{\Phi(\alpha_{i2}) - \Phi(\alpha_{i1})}$$

Reference

Berndt, E. 1991. *The Practice of Econometrics.* New York: Addison–Wesley.

Davidson, R. and J. G. MacKinnon. 1993. *Estimation and Inference Econometrics.* New York: Oxford University Press.

Greene, W. H. 2000. *Econometric Analysis.* 4th ed. Upper Saddle River, NJ: Prentice–Hall.

sg123	Hodges–Lehmann estimation of a shift in location between two populations

Duolao Wang, London School of Hygiene and Tropical Medicine, UK, Duolao.Wang@lshtm.ac.uk

Introduction

The Hodges–Lehmann method (Hodges and Lehmann 1963; Lehmann 1975) is a nonparametric procedure that extends the Wilcoxon–Mann–Whitney test to the problem of estimating the shift parameter between two populations. This method gives both a point estimate and a confidence interval for the shift parameter. The Hodges–Lehmann method has been widely used in medical research for evaluating a treatment effect. In assessing the bioequivalence of two drugs in clinical trials for regulatory submission, the Hodges–Lehmann method is recommended as a nonparametric alternative when the conditions for parametric methods are not satisfied. In this article, I present a new command `npshift` for estimating the shift parameter by a median unbiased estimate and confidence interval.

Syntax

`npshift` *varname* [`if` *exp*] [`in` *range*] , `by(`*groupvar*`)` [`level(#)`]

Description

`npshift` is used to estimate the shift between the two distributions from two independent samples (i.e., unmatched data). Both a point estimate and a confidence interval are computed for the shift parameter.

Options

`by(`*groupvar*`)` is not optional. It specifies the name of the grouping variable.

`level(#)` specifies the confidence level, in percent, for confidence intervals. The default is `level(95)` or as set by `set level`.

Examples

This example uses Stata's automobile data. As `npshift` is used for unmatched data, we assume that the experiment was conducted with 24 cars; 12 cars with and 12 cars without the fuel treatment. We created a dataset with 24 observations. `mpg` records the mileage rating and `treat` records a 1 if the mileage corresponds to untreated fuel and 2 if it corresponds to treated fuel.

```
. npshift mpg, by(treat)
Hodges-Lehmann Estimates of Shift Parameters
------------------------------------------------------------------
Point Estimate of Shift : Theta = Pop_2 - Pop_1 = 2
95% Confidence Interval for Theta:          [-1       ,        4]
------------------------------------------------------------------
. npshift mpg, by(treat) level(90)
Hodges-Lehmann Estimates of Shift Parameters
------------------------------------------------------------------
Point Estimate of Shift : Theta = Pop_2 - Pop_1 = 2
90% Confidence Interval for Theta:          [0        ,        4]
------------------------------------------------------------------
```

Methods and Formulas

The Hodges–Lehmann method is an extension of the Wilcoxon–Mann–Whitney test to the problem of estimating the shift parameter (by a median unbiased estimate and a confidence interval). The Hodges–Lehmann estimates were developed by Hodges and Lehmann (1963) and are described in detail in Hollander and Wolfe (1973, 75–82), and in Lehmann (1975, 81–95). Suppose, that we have two samples of data. Sample 1 consists of m observations, $x_1, \ldots, x_m$ drawn from the distribution $F(x)$, and sample 2 consists of n observations, $y_1, \ldots, y_n$ drawn from the distribution $G(y)$. We assume that $G(y)$ is $F(x)$ shifted to the right by θ units according to the relationship

$$G(y) = F(y - \theta)$$

The shift parameter θ is unknown and to be estimated.

To get a point estimate $\widehat{\theta}$, we first need to form the mn differences, $y_j - x_i$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ and then sort the differences in ascending order and denote the ordered values of $y_j - x_i$ as $U_{(1)} \leq U_{(2)} \leq \ldots U_{(nm)}$. The Hodges–Lehmann point estimator of θ is

$$\widehat{\theta} = \text{median}\left\{U_{(1)}, \ldots, U_{(nm)}\right\}$$

An asymptotic $100(1-\alpha)\%$ confidence interval for θ can be found by applying the normal approximation to the distribution of Mann and Whitney's statistic. The endpoints (θ_L, θ_U) of the interval are given by

$$\theta_L = U_{(C_\alpha)}, \qquad \theta_U = U_{(mn - C_\alpha + 1)},$$

where C_α is determined to satisfy

$$1 - \Phi\left(\frac{C_\alpha - mn/2}{\sqrt{mn(m+n+1)/12}}\right) = \alpha/2,$$

where $\Phi()$ is the cumulative distribution function of the standard normal distribution. In general the value of C in the above formula is not an integer, so take the integer closest to the value computed from the above formula.

Saved Results

`npshift` saves in `r()`:

`r(m)`	sample size m for population 1
`r(n)`	sample size n for population 2
`r(theta)`	point estimate of θ
`r(theta_l)`	lower confidence bound θ_L
`r(theta_u)`	upper confidence bound θ_U
`r(level)`	confidence level $1-\alpha$

References

Hodges, J. L., Jr. and E. L. Lehmann. 1963. Estimates of location based on rank tests. *Annals of Mathematical Statistics* 34: 598–611.

Hollander, M. and D. A. Wolfe. 1973. *Nonparametric Statistical Methods*. New York: John Wiley & Sons.

Lehmann, E. L. 1975. *Nonparametrics: Statistical Methods Based on Ranks*. San Francisco: Holden–Day.

STB categories and insert codes

Inserts in the STB are presently categorized as follows:

General Categories:

an	announcements	*ip*	instruction on programming
cc	communications & letters	*os*	operating system, hardware, & interprogram communication
dm	data management		
dt	datasets	*qs*	questions and suggestions
gr	graphics	*tt*	teaching
in	instruction	*zz*	not elsewhere classified

Statistical Categories:

sbe	biostatistics & epidemiology	*ssa*	survival analysis
sed	exploratory data analysis	*ssi*	simulation & random numbers
sg	general statistics	*sss*	social science & psychometrics
smv	multivariate analysis	*sts*	time-series, econometrics
snp	nonparametric methods	*svy*	survey sampling
sqc	quality control	*sxd*	experimental design
sqv	analysis of qualitative variables	*szz*	not elsewhere classified
srd	robust methods & statistical diagnostics		

In addition, we have granted one other prefix, *stata*, to the manufacturers of Stata for their exclusive use.

Guidelines for authors

The Stata Technical Bulletin (STB) is a journal that is intended to provide a forum for Stata users of all disciplines and levels of sophistication. The STB contains articles written by StataCorp, Stata users, and others.

Articles include new Stata commands (ado-files), programming tutorials, illustrations of data analysis techniques, discussions on teaching statistics, debates on appropriate statistical techniques, reports on other programs, and interesting datasets, announcements, questions, and suggestions.

A submission to the STB consists of

1. An insert (article) describing the purpose of the submission. The STB is produced using plain TEX so submissions using TEX (or LATEX) are the easiest for the editor to handle, but any word processor is appropriate. If you are not using TEX and your insert contains a significant amount of mathematics, please FAX (409–845–3144) a copy of the insert so we can see the intended appearance of the text.

2. Any ado-files, `.exe` files, or other software that accompanies the submission.

3. A help file for each ado-file included in the submission. See any recent STB diskette for the structure a help file. If you have questions, fill in as much of the information as possible and we will take care of the details.

4. A do-file that replicates the examples in your text. Also include the datasets used in the example. This allows us to verify that the software works as described and allows users to replicate the examples as a way of learning how to use the software.

5. Files containing the graphs to be included in the insert. If you have used STAGE to edit the graphs in your submission, be sure to include the `.gph` files. Do not add titles (e.g., "Figure 1: ...") to your graphs as we will have to strip them off.

The easiest way to submit an insert to the STB is to first create a single "archive file" (either a `.zip` file or a compressed `.tar` file) containing all of the files associated with the submission, and then email it to the editor at `stb@stata.com` either by first using `uuencode` if you are working on a Unix platform or by attaching it to an email message if your mailer allows the sending of attachments. In Unix, for example, to email the current directory and all of its subdirectories:

```
tar -cf - . | compress | uuencode xyzz.tar.Z > whatever
mail stb@stata.com < whatever
```

International Stata Distributors

International Stata users may also order subscriptions to the *Stata Technical Bulletin* from our International Stata Distributors.

Company:	Applied Statistics & Systems Consultants
Address:	P.O. Box 1169 17100 NAZERATH-ELLIT Israel
Phone:	+972 (0)6 6100101
Fax:	+972 (0)6 6554254
Email:	assc@netvision.net.il
Countries served:	Israel

Company:	IEM
Address:	P.O. Box 2222 PRIMROSE 1416 South Africa
Phone:	+27-11-8286169
Fax:	+27-11-8221377
Email:	iem@hot.co.za
Countries served:	South Africa, Botswana, Lesotho, Namibia, Mozambique, Swaziland, Zimbabwe

Company:	Axon Technology Company Ltd
Address:	9F, No. 259, Sec. 2 Ho-Ping East Road TAIPEI 106 Taiwan
Phone:	+886-(0)2-27045535
Fax:	+886-(0)2-27541785
Email:	hank@axon.axon.com.tw
Countries served:	Taiwan

Company:	MercoStat Consultores
Address:	9 de junio 1389 CP 11400 MONTEVIDEO Uruguay
Phone:	598-2-613-7905
Fax:	Same
Email:	mercost@adinet.com.uy
Countries served:	Uruguay, Argentina, Brazil, Paraguay

Company:	Chips Electronics
Address:	Lokasari Plaza 1st Floor Room 82 Jalan Mangga Besar Raya No. 82 JAKARTA Indonesia
Phone:	62 - 21 - 600 66 47
Fax:	62 - 21 - 600 66 47
Email:	puyuh23@indo.net.id
Countries served:	Indonesia

Company:	Metrika Consulting
Address:	Mosstorpsvagen 48 183 30 Taby STOCKHOLM Sweden
Phone:	+46-708-163128
Fax:	+46-8-7924747
Email:	sales@metrika.se
URL:	http://www.metrika.se
Countries served:	Sweden, Baltic States, Denmark, Finland, Iceland, Norway

Company:	Dittrich & Partner Consulting
Address:	Kieler Strasse 17 5. floor D-42697 Solingen Germany
Phone:	+49 2 12 / 26 066 - 0
Fax:	+49 2 12 / 26 066 - 66
Email:	sales@dpc.de
URL:	http://www.dpc.de
Countries served:	Germany, Austria, Italy

Company:	Ritme Informatique
Address:	34, boulevard Haussmann 75009 Paris France
Phone:	+33 (0)1 42 46 00 42 +33 (0)1 42 46 00 33
Email:	info@ritme.com
URL:	http://www.ritme.com
Countries served:	France, Belgium, Luxembourg

(*List continued on next page*)

International Stata Distributors

(*Continued from previous page*)

Company:	Scientific Solutions S.A.
Address:	Avenue du Général Guisan, 5 CH-1009 Pully/Lausanne Switzerland
Phone:	41 (0)21 711 15 20
Fax:	41 (0)21 711 15 21
Email:	info@scientific-solutions.ch
Countries served:	Switzerland

Company:	Smit Consult
Address:	Doormanstraat 19 5151 GM Drunen Netherlands
Phone:	+31 416-378 125
Fax:	+31 416-378 385
Email:	J.A.C.M.Smit@smitcon.nl
URL:	http://www.smitconsult.nl
Countries served:	Netherlands

Company:	Survey Design & Analysis Services P/L
Address:	249 Eramosa Road West Moorooduc VIC 3933 Australia
Phone:	+61 (0)3 5978 8329
Fax:	+61 (0)3 5978 8623
Email:	sales@survey-design.com.au
URL:	http://survey-design.com.au
Countries served:	Australia, New Zealand

Company:	Timberlake Consultants
Address:	Unit B3 Broomsleigh Bus. Park Worsley Bridge Road LONDON SE26 5BN United Kingdom
Phone:	+44 (0)208 697 3377
Fax:	+44 (0)208 697 3388
Email:	info@timberlake.co.uk
URL:	http://www.timberlake.co.uk
Countries served:	United Kingdom, Eire

Company:	Timberlake Consulting S.L.
Address:	Calle Mendez Nunez, 1, 3 41011 Sevilla Spain
Phone:	+34 (9) 5 422 0648
Fax:	+34 (9) 5 422 0648
Email:	timberlake@zoom.es
Countries served:	Spain

Company:	Timberlake Consultores, Lda.
Address:	Praceta Raúl Brandao, n°1, 1°E 2720 ALFRAGIDE Portugal
Phone:	+351 (0)1 471 73 47
Fax:	+351 (0)1 471 73 47
Email:	timberlake.co@mail.telepac.pt
Countries served:	Portugal

Company:	Unidost A.S.
	Rihtim Cad. Polat Han D:38
	Kadikoy
	81320 ISTANBUL
	Turkey
Phone:	+90 (216) 414 19 58
Fax:	+30 (216) 336 89 23
Email:	info@unidost.com
URL:	http://abone.turk.net/unidost
Countries served:	Turkey

Company:	Vishvas Marketing-Mix Services
Address:	C\O S. D. Wamorkar "Prashant" Vishnu Nagar, Naupada THANE - 400602 India
Phone:	+91-251-440087
Fax:	+91-22-5378552
Email:	vishvas@vsnl.com
Countries served:	India