suest — Seemingly unrelated estimation

Description

suest is a postestimation command; see **[U] 20 Estimation and postestimation commands**.

suest combines the estimation results—parameter estimates and associated (co)variance matrices—stored under *namelist* into one parameter vector and *simultaneous* (co)variance matrix of the sandwich/robust type. This (co)variance matrix is appropriate even if the estimates were obtained on the same or on overlapping data.

Typical applications of suest are tests for intramodel and cross-model hypotheses using test or testnl, for example, a generalized Hausman specification test. lincom and nlcom may be used after suest to estimate linear combinations and nonlinear functions of coefficients. suest may also be used to adjust a standard VCE for clustering or survey design effects.

Different estimators are allowed, for example, a regress model and a probit model; the only requirement is that predict produce equation-level scores with the score option after an estimation command. The models may be estimated on different samples, due either to explicit if or in selection or to missing values. If weights are applied, the same weights (type and values) should be applied to all models in *namelist*. The estimators should be estimated without vce(robust) or vce(cluster *clustvar*) options. suest returns the robust VCE, allows the vce(cluster *clustvar*) option, and automatically works with results from the svy prefix command (only for vce(linearized)). See example 8 in [SVY] **svy postestimation** for an example using suest with svy: ologit.

Because suest posts its results like a proper estimation command, its results can be stored via estimates store. Moreover, like other estimation commands, suest typed without arguments replays the results.

Quick start

Combined results for stored estimates m1 and m2

```
suest m1 m2
```

Same as above, but report exponentiated coefficients and label them "Odds ratio"

```
suest m1 m2, eform(Odds ratio)
```

With cluster–robust standard errors adjusting for clustering by levels of cvar

```
suest m1 m2, vce(cluster cvar)
```

Use svyset data after specifying command prefix svy: to estimate m1 and m2

```
suest m1 m2
```

Menu

Statistics > Postestimation

Syntax

`suest` *namelist* [, *options*]

where *namelist* is a list of one or more names under which estimation results were stored via `estimates store`; see [R] **estimates store**. Wildcards may be used. `*` and `_all` refer to all stored results. A period (`.`) may be used to refer to the last estimation results, even if they have not (yet) been stored.

options	Description
SE/Robust	
`svy`	survey data estimation
`vce(`*vcetype*`)`	*vcetype* may be `robust` or `cluster` *clustvar*
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`dir`	display a table describing the models
`eform(`*string*`)`	report exponentiated coefficients and label as *string*
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

`coeflegend` does not appear in the dialog box.

Options

SE/Robust

`svy` specifies that estimation results should be modified to reflect the survey design effects according to the `svyset` specifications, see [SVY] **svyset**.

The `svy` option is implied when `suest` encounters survey estimation results from the `svy` prefix; see [SVY] **svy**. Poststratification is allowed only with survey estimation results from the `svy` prefix.

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that allow for intragroup correlation (`cluster` *clustvar*); see [R] ***vce_option***.

The `vce()` option may not be combined with the `svy` option or estimation results from the `svy` prefix.

Reporting

`level(#)` specifies the confidence level, as a percentage, for confidence intervals of the coefficients; see [R] **level**.

`dir` displays a table describing the models in *namelist* just like `estimates dir` *namelist*.

`eform(`*string*`)` displays the coefficient table in exponentiated form: for each coefficient, exp(b) rather than b is displayed, and standard errors and confidence intervals are transformed. *string* is the table header that will be displayed above the transformed coefficients and must be 11 characters or fewer, for example, `eform("Odds ratio")`.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

The following option is available with `suest` but is not shown in the dialog box:

`coeflegend`; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Using suest
Remarks on regress
Testing the assumption of the independence of irrelevant alternatives
Testing proportionality
Testing cross-model hypotheses

Using suest

If you plan to use `suest`, you must take precautions when fitting the original models. These restrictions are relaxed when using `svy` commands; see [SVY] **svy postestimation**.

1. `suest` works with estimation commands that allow `predict` to generate equation-level score variables when supplied with the `score` (or `scores`) option. For example, equation-level score variables are generated after running `mlogit` by typing

   ```
   . predict sc*, scores
   ```

   To ensure that `suest` generates the appropriate scores and therefore an appropriate VCE, you should not modify the values of the variables used in the models in between fitting the models and using `suest`.

2. Estimation should take place *without* the `vce(robust)` or `vce(cluster` *clustvar*`)` option. `suest` always computes the robust estimator of the (co)variance, and `suest` has a `vce(cluster` *clustvar*`)` option.

   The within-model covariance matrices computed by `suest` are identical to those obtained by specifying a `vce(robust)` or `vce(cluster` *clustvar*`)` option during estimation. `suest`, however, also estimates the between-model covariances of parameter estimates.

3. Finally, the estimation results to be combined should be stored by `estimates store`; see [R] **estimates store**.

After estimating and storing a series of estimation results, you are ready to combine the estimation results with `suest`,

```
. suest name1 [ name2 ... ] [, vce(cluster clustvar) ]
```

and you can subsequently use postestimation commands, such as `test`, to test hypotheses. Here an important issue is how `suest` assigns names to the equations. If you specify one model *name*, the original equation names are left unchanged; otherwise, `suest` constructs new equation names. The coefficients of a single-equation model (such as `logit` and `poisson`) that was `estimate stored` under name *X* are collected under equation *X*. With a multiequation model stored under name *X*, `suest` prefixes *X*_ to an original equation name *eq*, forming equation name, *X_eq*.

❑ Technical note

Earlier we said that standard errors from `suest` are identical to those obtained by specifying the `vce(robust)` option with each command individually. Thus if you fit a logistic model using `logit` with the `vce(robust)` option, you will get the same standard errors when you type

```
. suest .
```

directly after `logit` using the same data without the `vce(robust)` option.

This is not true for multiple estimation results when the estimation samples are not all the same. The standard errors from `suest` will be slightly smaller than those from individual model fits using the `vce(robust)` option because `suest` uses a larger number of observations to estimate the simultaneous (co)variance matrix.

❑

❑ Technical note

In rare circumstances, `suest` may have to truncate equation names to 32 characters. When equation names are not unique because of truncation, `suest` numbers the equations within models, using equations named *X_#*.

❑

Remarks on regress

`regress` (see [R] **regress**) does not include its ancillary parameter, the residual variance, in its coefficient vector and (co)variance matrix. Moreover, while the `score` option is allowed with `predict` after `regress`, a score variable is generated for the mean but not for the variance parameter. `suest` contains special code that assigns the equation name `mean` to the coefficients for the mean, adds the equation `lnvar` for the log variance, and computes the appropriate two score variables itself.

Testing the assumption of the independence of irrelevant alternatives

The multinomial logit model and the closely related conditional logit model satisfy a probabilistic version of the assumption of the independence of irrelevant alternatives (IIA), implying that the ratio of the probabilities for two alternatives does not depend on what other alternatives are available. Hausman and McFadden (1984) proposed a test for this assumption that is implemented in the hausman command. The standard Hausman test has several limitations. First, the test statistic may be undefined because the estimated VCE does not satisfy the required asymptotic properties of the test. Second, the classic Hausman test applies only to the test of the equality of two estimators. Third, the test requires access to a fully efficient estimator; such an estimator may not be available, for example, if you are analyzing complex survey data. Using suest can overcome these three limitations.

▷ Example 1

In our first example, we follow the analysis of the type of health insurance reported in [R] **mlogit** and demonstrate the hausman command with the suest/test combination. We fit the full multinomial logit model for all three alternatives and two restricted multinomial models in which one alternative is excluded. After fitting each of these models, we give them a title with the estimates title command and then store them with the estimates store command.

```
. use https://www.stata-press.com/data/r19/sysdsn4
(Health insurance data)

. mlogit insure age male

Iteration 0:  Log likelihood = -555.85446
Iteration 1:  Log likelihood = -551.32973
Iteration 2:  Log likelihood = -551.32802
Iteration 3:  Log likelihood = -551.32802

Multinomial logistic regression                        Number of obs =    615
                                                       LR chi2(4)    =   9.05
                                                       Prob > chi2   = 0.0598
Log likelihood = -551.32802                            Pseudo R2     = 0.0081

------------------------------------------------------------------------------
      insure | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Indemnity    |  (base outcome)
-------------+----------------------------------------------------------------
Prepaid      |
         age |  -.0100251   .0060181    -1.67   0.096    -.0218204    .0017702
        male |   .5095747   .1977893     2.58   0.010     .1219147    .8972346
       _cons |   .2633838   .2787575     0.94   0.345    -.2829708    .8097383
-------------+----------------------------------------------------------------
Uninsure     |
         age |  -.0051925   .0113821    -0.46   0.648    -.0275011    .0171161
        male |   .4748547   .3618462     1.31   0.189    -.2343508     1.18406
       _cons |  -1.756843   .5309602    -3.31   0.001    -2.797506   -.7161803
------------------------------------------------------------------------------

. estimates title: All three insurance forms

. estimates store m1

. quietly mlogit insure age male if insure != "Uninsure":insure

. estimates title: insure != "Uninsure":insure

. estimates store m2

. quietly mlogit insure age male if insure != "Prepaid":insure

. estimates title: insure != "Prepaid":insure

. estimates store m3
```

Having performed the three estimations, we inspect the results. `estimates dir` provides short descriptions of the models that were stored using `estimates store`. Typing `estimates table` lists the coefficients, displaying blanks for a coefficient not contained in a model.

```
. estimates dir

-------------------------------------------------------------------------------
              |             Dependent  Number of
         Name | Command      variable     param.  Title
--------------+----------------------------------------------------------------
           m1 | mlogit         insure          9  All three insurance forms
           m2 | mlogit         insure          6  insure != "Uninsure":insure
           m3 | mlogit         insure          6  insure != "Prepaid":insure
-------------------------------------------------------------------------------

. estimates table m1 m2 m3, star stats(N ll) keep(Prepaid: Uninsure:)

------------------------------------------------------------------
     Variable |      m1               m2               m3
--------------+---------------------------------------------------
Prepaid       |
          age | -.01002511       -.01015205
         male |  .50957468**      .51440033**
        _cons |  .26338378        .26780432
--------------+---------------------------------------------------
Uninsure      |
          age | -.00519249                        -.00410547
         male |  .47485472                         .45910738
        _cons | -1.7568431***                     -1.8017743***
--------------+---------------------------------------------------
Statistics    |
            N |        615              570              338
           ll | -551.32802       -390.48643       -131.76807
------------------------------------------------------------------
                       Legend: * p<0.05; ** p<0.01; *** p<0.001
```

Comparing the coefficients between models does not suggest substantial differences. We can formally test that coefficients are the same for the full model `m1` and the restricted models `m2` and `m3` by using the `hausman` command. `hausman` expects the models to be specified in the order “always consistent” first and “efficient under H_0” second.

```
. hausman m2 m1, alleqs constant
                  ---- Coefficients ----
              |      (b)           (B)             (b-B)     sqrt(diag(V_b-V_B))
              |      m2            m1           Difference       Std. err.
--------------+-------------------------------------------------------------------
          age |   -.0101521     -.0100251        -.0001269              .
         male |    .5144003      .5095747         .0048256        .0123338
        _cons |    .2678043      .2633838         .0044205              .
----------------------------------------------------------------------------------
                           b = Consistent under H0 and Ha; obtained from mlogit.
          B = Inconsistent under Ha, efficient under H0; obtained from mlogit.

Test of H0: Difference in coefficients not systematic

    chi2(3) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            =   0.08
Prob > chi2 = 0.9944
(V_b-V_B is not positive definite)
```

```
. hausman m3 m1, alleqs constant

                 ---- Coefficients ----
             |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
             |       m3           m1         Difference       Std. err.
-------------+----------------------------------------------------------------
         age |   -.0041055    -.0051925         .001087        .0021355
        male |    .4591074     .4748547       -.0157473               .
       _cons |   -1.801774    -1.756843       -.0449311        .1333421
------------------------------------------------------------------------------
                           b = Consistent under H0 and Ha; obtained from mlogit.
            B = Inconsistent under Ha, efficient under H0; obtained from mlogit.

Test of H0: Difference in coefficients not systematic

chi2(3) = (b-B)'[(V_b-V_B)^(-1)](b-B)
        = -0.18

Warning: chi2 < 0 ==> model fitted on these data
         fails to meet the asymptotic assumptions
         of the Hausman test; see suest for a
         generalized test.
```

According to the test of m1 against m2, we cannot reject the hypothesis that the coefficients of m1 and m2 are the same. The second Hausman test is not well defined—something that happens fairly often. The problem is due to the estimator of the variance V(b-B) as V(b)-V(B), which is a feasible estimator only asymptotically. Here it simply is not a proper variance matrix, and the Hausman test becomes undefined.

suest m1 m2 estimates the simultaneous (co)variance of the coefficients of models m1 and m2. Although suest is technically a postestimation command, it acts like an estimation command in that it stores the simultaneous coefficients in e(b) and the full (co)variance matrix in e(V). We could have used the estat vce command to display the full (co)variance matrix to show that the cross-model covariances were indeed estimated. Typically, we would not have a direct interest in e(V).

```
. suest m1 m2, noomitted

Simultaneous results for m1, m2                          Number of obs = 615

------------------------------------------------------------------------------
             |               Robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
m1_Indemnity |
-------------+----------------------------------------------------------------
m1_Prepaid   |
         age |  -.0100251   .0059403    -1.69   0.091    -.0216679    .0016176
        male |   .5095747   .1988159     2.56   0.010     .1199027    .8992467
       _cons |   .2633838    .277307     0.95   0.342    -.280128     .8068956
-------------+----------------------------------------------------------------
m1_Uninsure  |
         age |  -.0051925   .0109005    -0.48   0.634    -.0265571    .0161721
        male |   .4748547   .3677326     1.29   0.197    -.2458879    1.195597
       _cons |  -1.756843   .4971383    -3.53   0.000    -2.731216    -.78247
-------------+----------------------------------------------------------------
m2_Indemnity |
-------------+----------------------------------------------------------------
m2_Prepaid   |
         age |  -.0101521   .0058988    -1.72   0.085    -.0217135    .0014094
        male |   .5144003   .1996133     2.58   0.010     .1231654    .9056352
       _cons |   .2678043   .2744019     0.98   0.329    -.2700134    .8056221
------------------------------------------------------------------------------
```

suest created equation names by combining the name under which we stored the results using estimates store with the original equation names. Thus, in the simultaneous estimation result, equation Prepaid originating in model m1 is named m1_Prepaid. According to the McFadden–Hausman specification of a test for IIA, the coefficients of the equations m1_PrePaid and m2_PrePaid should be equal. This equality can be tested easily with the test command. The cons option specifies that the intercept _cons be included in the test.

```
. test [m1_Prepaid = m2_Prepaid], cons
 ( 1)  [m1_Prepaid]age - [m2_Prepaid]age = 0
 ( 2)  [m1_Prepaid]male - [m2_Prepaid]male = 0
 ( 3)  [m1_Prepaid]_cons - [m2_Prepaid]_cons = 0
           chi2(  3) =    0.89
         Prob > chi2 =    0.8266
```

The Hausman test via suest is comparable with that computed by hausman, but they use different estimators of the variance of the difference of the estimates. The hausman command estimates $V(b-B)$ by $V(b) - V(B)$, whereas suest estimates $V(b-B)$ by $V(b) - \mathrm{cov}(b,B) - \mathrm{cov}(B,b) + V(B)$. One advantage of the second estimator is that it is always admissible, so the resulting test is always well defined. This quality is illustrated in the Hausman-type test of IIA comparing models m1 and m3.

```
. suest m1 m3, noomitted
Simultaneous results for m1, m3                            Number of obs = 615
──────────────────────────────────────────────────────────────────────────────
             │               Robust
             │ Coefficient  std. err.      z    P>|z|     [95% conf. interval]
─────────────┼────────────────────────────────────────────────────────────────
m1_Indemnity │
─────────────┼────────────────────────────────────────────────────────────────
m1_Prepaid   │
         age │  -.0100251   .0059403    -1.69   0.091    -.0216679    .0016176
        male │   .5095747   .1988159     2.56   0.010     .1199027    .8992467
       _cons │   .2633838    .277307     0.95   0.342    -.280128     .8068956
─────────────┼────────────────────────────────────────────────────────────────
m1_Uninsure  │
         age │  -.0051925   .0109005    -0.48   0.634    -.0265571    .0161721
        male │   .4748547   .3677326     1.29   0.197    -.2458879    1.195597
       _cons │  -1.756843   .4971383    -3.53   0.000    -2.731216     -.78247
─────────────┼────────────────────────────────────────────────────────────────
m3_Indemnity │
─────────────┼────────────────────────────────────────────────────────────────
m3_Uninsure  │
         age │  -.0041055   .0111185    -0.37   0.712    -.0258974    .0176865
        male │   .4591074   .3601307     1.27   0.202    -.2467357    1.164951
       _cons │  -1.801774   .5226351    -3.45   0.001     -2.82612   -.7774283
──────────────────────────────────────────────────────────────────────────────

. test [m1_Uninsure = m3_Uninsure], cons
 ( 1)  [m1_Uninsure]age - [m3_Uninsure]age = 0
 ( 2)  [m1_Uninsure]male - [m3_Uninsure]male = 0
 ( 3)  [m1_Uninsure]_cons - [m3_Uninsure]_cons = 0
           chi2(  3) =    1.49
         Prob > chi2 =    0.6845
```

Although the classic Hausman test computed by hausman is not defined here, the suest-based test is just fine. We cannot reject the equality of the common coefficients across m1 and m3.

A second advantage of the suest approach is that we can estimate the (co)variance matrix of the multivariate normal distribution of the estimators of the three models m1, m2, and m3 and test that the common coefficients are equal.

```
. suest m*, noomitted

Simultaneous results for m1, m2, m3                     Number of obs = 615

─────────────┬────────────────────────────────────────────────────────────────
             │               Robust
             │ Coefficient  std. err.      z    P>|z|     [95% conf. interval]
─────────────┼────────────────────────────────────────────────────────────────
m1_Indemnity │
─────────────┼────────────────────────────────────────────────────────────────
m1_Prepaid   │
         age │  -.0100251   .0059403    -1.69   0.091    -.0216679    .0016176
        male │   .5095747   .1988159     2.56   0.010     .1199027    .8992467
       _cons │   .2633838    .277307     0.95   0.342     -.280128    .8068956
─────────────┼────────────────────────────────────────────────────────────────
m1_Uninsure  │
         age │  -.0051925   .0109005    -0.48   0.634    -.0265571    .0161721
        male │   .4748547   .3677326     1.29   0.197    -.2458879    1.195597
       _cons │  -1.756843   .4971383    -3.53   0.000    -2.731216    -.78247
─────────────┼────────────────────────────────────────────────────────────────
m2_Indemnity │
─────────────┼────────────────────────────────────────────────────────────────
m2_Prepaid   │
         age │  -.0101521   .0058988    -1.72   0.085    -.0217135    .0014094
        male │   .5144003   .1996133     2.58   0.010     .1231654    .9056352
       _cons │   .2678043   .2744019     0.98   0.329    -.2700134    .8056221
─────────────┼────────────────────────────────────────────────────────────────
m3_Indemnity │
─────────────┼────────────────────────────────────────────────────────────────
m3_Uninsure  │
         age │  -.0041055   .0111185    -0.37   0.712    -.0258974    .0176865
        male │   .4591074   .3601307     1.27   0.202    -.2467357    1.164951
       _cons │  -1.801774   .5226351    -3.45   0.001     -2.82612   -.7774283
─────────────┴────────────────────────────────────────────────────────────────

. test [m1_Prepaid = m2_Prepaid] , cons notest

 ( 1)  [m1_Prepaid]age - [m2_Prepaid]age = 0
 ( 2)  [m1_Prepaid]male - [m2_Prepaid]male = 0
 ( 3)  [m1_Prepaid]_cons - [m2_Prepaid]_cons = 0

. test [m1_Uninsure = m3_Uninsure], cons acc

 ( 1)  [m1_Prepaid]age - [m2_Prepaid]age = 0
 ( 2)  [m1_Prepaid]male - [m2_Prepaid]male = 0
 ( 3)  [m1_Prepaid]_cons - [m2_Prepaid]_cons = 0
 ( 4)  [m1_Uninsure]age - [m3_Uninsure]age = 0
 ( 5)  [m1_Uninsure]male - [m3_Uninsure]male = 0
 ( 6)  [m1_Uninsure]_cons - [m3_Uninsure]_cons = 0

           chi2(  6) =    1.95
         Prob > chi2 =    0.9240
```

Again, we do not find evidence against the correct specification of the multinomial logit for type of insurance. The classic Hausman test assumes that one of the estimators (named B in hausman) is efficient, that is, it has minimal (asymptotic) variance. This assumption ensures that $V(b) - V(B)$ is an admissible, viable estimator for $V(b - B)$. The assumption that we have an efficient estimator is a restrictive one. It is violated, for instance, if our data are clustered. We want to adjust for clustering via a vce(cluster *clustvar*) option by requesting the cluster-adjusted sandwich estimator of variance. Consequently, in such a case, hausman cannot be used. This problem does not exist with the suest version of the Hausman test. To illustrate this feature, we suppose that the data are clustered by city—we constructed an imaginary variable `cityid` for this illustration. If we plan to apply `suest`, we would not specify the `vce(cluster` *clustvar*`)` option at the time of estimation. `suest` has a `vce(cluster` *clustvar*`)` option. Thus, we do not need to refit the models; we can call `suest` and `test` right away.

```
. suest m1 m2, vce(cluster cityid) noomitted
Simultaneous results for m1, m2                          Number of obs = 615
                               (Std. err. adjusted for 260 clusters in cityid)
------------------------------------------------------------------------------
             |               Robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
m1_Indemnity |
-------------+----------------------------------------------------------------
m1_Prepaid   |
         age |  -.0100251    .005729    -1.75   0.080    -.0212538    .0012035
        male |   .5095747   .1910496     2.67   0.008     .1351244     .884025
       _cons |   .2633838   .2698797     0.98   0.329    -.2655708    .7923384
-------------+----------------------------------------------------------------
m1_Uninsure  |
         age |  -.0051925   .0104374    -0.50   0.619    -.0256495    .0152645
        male |   .4748547   .3774021     1.26   0.208    -.2648399    1.214549
       _cons |  -1.756843   .4916613    -3.57   0.000    -2.720481   -.7932048
-------------+----------------------------------------------------------------
m2_Indemnity |
-------------+----------------------------------------------------------------
m2_Prepaid   |
         age |  -.0101521   .0057164    -1.78   0.076    -.0213559    .0010518
        male |   .5144003   .1921385     2.68   0.007     .1378158    .8909848
       _cons |   .2678043   .2682193     1.00   0.318    -.2578959    .7935045
------------------------------------------------------------------------------

. test [m1_Prepaid = m2_Prepaid], cons
 ( 1)  [m1_Prepaid]age - [m2_Prepaid]age = 0
 ( 2)  [m1_Prepaid]male - [m2_Prepaid]male = 0
 ( 3)  [m1_Prepaid]_cons - [m2_Prepaid]_cons = 0

           chi2(  3) =    0.79
         Prob > chi2 =    0.8529
```

`suest` provides some descriptive information about the clustering on `cityid`. Like any other estimation command, `suest` informs us that the standard errors are adjusted for clustering. The Hausman-type test obtained from the `test` command uses a simultaneous (co)variance of `m1` and `m2` appropriately adjusted for clustering. In this example, we still do not have reason to conclude that the multinomial logit model in this application is misspecified, that is, that IIA is violated.

◁

The multinomial logistic regression model is a special case of the conditional logistic regression model; see [R] **clogit**. Like the multinomial logistic regression model, the conditional logistic regression model also makes the IIA assumption. Consider an example, introduced in [CM] **cmclogit**, in which the demand for American, Japanese, and European cars is modeled in terms of the number of local dealers of the respective brands and of some individual attributes incorporated in interaction with the nationality of cars. We want to perform a Hausman-type test for IIA comparing the decision between all nationalities with the decision between non-American cars. The following code fragment demonstrates how to conduct a Hausman test for IIA via `suest` in this case.

```
. clogit choice japan europe maleJap maleEur incJap incEur dealer, group(id)
. estimates store allcars
. clogit choice japan maleJap incJap dealer if car!=1 , group(id)
. estimates store foreign
. suest allcars foreign
. test [allcars_choice=foreign_choice], common
```

Testing proportionality

The applications of suest that we have discussed so far concern Hausman-type tests for misspecification. To test such a hypothesis, we compared two estimators that have the same probability limit if the hypothesis holds true, but otherwise have different limits. We may also want to compare the coefficients of models (estimators) for other substantive reasons. Although we most often want to test whether coefficients differ between models or estimators, we may occasionally want to test other constraints (see Hausman and Ruud [1987]).

▷ Example 2

In this example, using simulated labor market data for siblings, we consider two dependent variables, income (inc) and whether a person was promoted in the last year (promo). We apply familiar economic arguments regarding human capital, according to which employees have a higher income and a higher probability of being promoted, by having more human capital. Human capital is acquired through formal education (edu) and on-the-job training experience (exp). We study whether income and promotion are “two sides of the same coin”, that is, whether they reflect a common latent variable, “human capital”. Accordingly, we want to compare the effects of different aspects of human capital on different outcome variables.

We estimate fairly simple labor market equations. The income model is estimated with regress, and the estimation results are stored under the name Inc.

```
. use https://www.stata-press.com/data/r19/income
. regress inc edu exp male
      Source |       SS           df       MS      Number of obs   =       277
-------------+----------------------------------   F(3, 273)       =     42.34
       Model |  2058.44672         3  686.148908   Prob > F        =    0.0000
    Residual |  4424.05183       273  16.2053181   R-squared       =    0.3175
-------------+----------------------------------   Adj R-squared   =    0.3100
       Total |  6482.49855       276  23.4873136   Root MSE        =    4.0256

------------------------------------------------------------------------------
         inc | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         edu |   2.213707    .243247     9.10   0.000     1.734828    2.692585
         exp |    1.47293    .231044     6.38   0.000     1.018076    1.927785
        male |   .5381153   .4949466     1.09   0.278     -.436282    1.512513
       _cons |   1.255497   .3115808     4.03   0.000      .642091    1.868904
------------------------------------------------------------------------------

. est store Inc
```

Being sibling data, the observations are clustered on family of origin, famid. In the estimation of the regression parameters, we did not specify a vce(cluster famid) option to adjust standard errors for clustering on family (famid). Thus, the standard errors reported by regress are potentially flawed. This problem will, however, be corrected by specifying a vce(cluster *clustvar*) option with suest.

Next, we estimate the promotion equation with `probit` and again store the results under an appropriate name.

```
. probit promo edu exp male, nolog
Probit regression                                       Number of obs =    277
                                                        LR chi2(3)    =  49.76
                                                        Prob > chi2   = 0.0000
Log likelihood = -158.43888                             Pseudo R2     = 0.1357

------------------------------------------------------------------------------
       promo | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         edu |   .4593002   .0898537     5.11   0.000     .2831901    .6354102
         exp |   .3593023   .0805774     4.46   0.000     .2013735    .5172312
        male |   .2079983   .1656413     1.26   0.209    -.1166527    .5326494
       _cons |   -.464622   .1088166    -4.27   0.000    -.6778985   -.2513454
------------------------------------------------------------------------------

. est store Promo
```

The coefficients in the income and promotion equations definitely seem to be different. However, because the scales of the two variables are different, we would not expect the coefficients to be equal. The correct hypothesis here is that the proportionality of the coefficients of the two models, apart from the constant, are equal. This formulation would still reflect that the relative effects of the different aspects of human capital do not differ between the dependent variables. We can obtain a nonlinear Wald test for the hypothesis of proportionality by using the `testnl` command on the combined estimation results of the two estimators. Thus, we first have to form the combined estimation results. At this point, we specify the `vce(cluster famid)` option to adjust for the clustering of observations on `famid`.

```
. suest Inc Promo, vce(cluster famid)
Simultaneous results for Inc, Promo                         Number of obs = 277
                                  (Std. err. adjusted for 135 clusters in famid)
------------------------------------------------------------------------------
             |               Robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Inc_mean     |
         edu |   2.213707   .2483907     8.91   0.000      1.72687    2.700543
         exp |    1.47293   .1890583     7.79   0.000     1.102383    1.843478
        male |   .5381153   .4979227     1.08   0.280    -.4377952    1.514026
       _cons |   1.255497   .3374977     3.72   0.000      .594014    1.916981
-------------+----------------------------------------------------------------
Inc_lnvar    |
       _cons |   2.785339    .079597    34.99   0.000     2.629332    2.941347
-------------+----------------------------------------------------------------
Promo_promo  |
         edu |   .4593002   .0886982     5.18   0.000     .2854549    .6331454
         exp |   .3593023    .079772     4.50   0.000     .2029522    .5156525
        male |   .2079983   .1691053     1.23   0.219    -.1234419    .5394386
       _cons |   -.464622   .1042169    -4.46   0.000    -.6688833   -.2603607
------------------------------------------------------------------------------
```

The standard errors reported by `suest` are identical to those reported by the respective estimation commands when invoked with the `vce(cluster famid)` option. We are now ready to test for proportionality:

$$H_0: \frac{\beta^{\text{Income}}_{\text{edu}}}{\beta^{\text{Promotion}}_{\text{edu}}} = \frac{\beta^{\text{Income}}_{\text{exp}}}{\beta^{\text{Promotion}}_{\text{exp}}} = \frac{\beta^{\text{Income}}_{\text{male}}}{\beta^{\text{Promotion}}_{\text{male}}}$$

It is straightforward to translate this into syntax suitable for `testnl`, recalling that the coefficient of variable *v* in equation *eq* is denoted by `[`*eq*`]`*v*.

```
. testnl [Inc_mean]edu/[Promo_promo]edu =
>        [Inc_mean]exp/[Promo_promo]exp =
>        [Inc_mean]male/[Promo_promo]male

  (1)  [Inc_mean]edu/[Promo_promo]edu = [Inc_mean]exp/[Promo_promo]exp
  (2)  [Inc_mean]edu/[Promo_promo]edu = [Inc_mean]male/[Promo_promo]male

               chi2(2) =        0.61
           Prob > chi2 =        0.7385
```

From the evidence, we fail to reject the hypotheses that the coefficients of the income and promotion equations are proportional. Thus, it is not unreasonable to assume that income and promotion can be explained by the same latent variable, “labor market success”.

A disadvantage of the nonlinear Wald test is that it is not invariant with respect to representation: a Wald test for a mathematically equivalent formulation of the nonlinear constraint usually leads to a different test result. An equivalent formulation of the proportionality hypothesis is

$$\begin{aligned} H_0\colon\quad & \beta^{\text{Income}}_{\text{edu}}\beta^{\text{Promotion}}_{\text{exp}} = \beta^{\text{Promotion}}_{\text{edu}}\beta^{\text{Income}}_{\text{exp}} \quad \text{and} \\ & \beta^{\text{Income}}_{\text{edu}}\beta^{\text{Promotion}}_{\text{male}} = \beta^{\text{Promotion}}_{\text{edu}}\beta^{\text{Income}}_{\text{male}} \end{aligned}$$

This formulation is “more linear” in the coefficients. The asymptotic χ^2 distribution of the nonlinear Wald statistic can be expected to be more accurate for this representation.

```
. testnl ([Inc_mean]edu*[Promo_promo]exp = [Inc_mean]exp*[Promo_promo]edu)
>        ([Inc_mean]edu*[Promo_promo]male = [Inc_mean]male*[Promo_promo]edu)

  (1)  [Inc_mean]edu*[Promo_promo]exp = [Inc_mean]exp*[Promo_promo]edu
  (2)  [Inc_mean]edu*[Promo_promo]male = [Inc_mean]male*[Promo_promo]edu

               chi2(2) =        0.46
           Prob > chi2 =        0.7936
```

Here the two representations lead to similar test statistics and p-values. As before, we fail to reject the hypothesis of proportionality of the coefficients of the two models.

◁

Testing cross-model hypotheses

▷ Example 3

In this example, we demonstrate how some cross-model hypotheses can be tested using the facilities already available in most estimation commands. This demonstration will explain the intricate relationship between the cluster adjustment of the robust estimator of variance and the `suest` command. It will also be made clear that a new facility is required to perform more general cross-model testing.

We want to test whether the effect of x_1 on the binary variable y_1 is the same as the effect of x_2 on the binary y_2; see Clogg, Petkova, and Haritou (1995). In this setting, x_1 may equal x_2, and y_1 may equal y_2. We assume that logistic regression models can be used to model the responses, and for simplicity, we ignore further predictor variables in these models. If the two logit models are fit on independent samples so that the estimators are (stochastically) independent, a Wald test for `_b[x1] = _b[x2]` rejects the null hypothesis if

$$\frac{\hat{b}(x_1)-\hat{b}(x_2)}{\left[\hat{\sigma}^2\left\{\hat{b}(x_1)\right\}+\hat{\sigma}^2\left\{\hat{b}(x_2)\right\}\right]^{1/2}}$$

is larger than the appropriate χ^2_1 threshold. If the models are fit on the *same* sample (or on dependent samples), so that the estimators are stochastically dependent, the above test that ignores the covariance between the estimators is not appropriate.

It is instructive to see how this problem can be tackled by "stacking" data. In the stacked format, we doubled the number of observations. The dependent variable is y_1 in the first half of the data and is y_2 in the second half of the data. The predictor variable z_1 is set to x_1 in the first half of the expanded data and to 0 in the rest. Similarly, z_2 is 0 in the first half and x_2 in the second half. The following diagram illustrates the transformation, in the terminology of the `reshape` command, from wide to long format.

$$\left(\begin{array}{ccccc}\texttt{id} & y_1 & y_2 & x_1 & x_2 \\ \hline 1 & y_{11} & y_{21} & x_{11} & x_{21} \\ 2 & y_{12} & y_{22} & x_{12} & x_{22} \\ 3 & y_{13} & y_{23} & x_{13} & x_{23}\end{array}\right) \Longrightarrow \left(\begin{array}{ccccc}\texttt{id} & y & z_1 & z_2 & \texttt{model} \\ \hline 1 & y_{11} & x_{11} & 0 & 1 \\ 2 & y_{12} & x_{12} & 0 & 1 \\ 3 & y_{13} & x_{13} & 0 & 1 \\ 1 & y_{21} & 0 & x_{21} & 2 \\ 2 & y_{22} & 0 & x_{22} & 2 \\ 3 & y_{23} & 0 & x_{23} & 2\end{array}\right)$$

The observations in the long-format data organization are *clustered* on the original subjects and are identified with the identifier `id`. The clustering on `id` has to be accounted for when fitting a simultaneous model. The simplest way to deal with clustering is to use the cluster adjustment of the robust or sandwich estimator; see [P] **_robust**. The data manipulation can be accomplished easily with the `stack` command; see [D] **stack**. Subsequently, we fit a simultaneous logit model and perform a Wald test for the hypothesis that the coefficients of `z1` and `z2` are the same. A full setup to obtain the cross-model Wald test could then be as follows:

```
. generate zero = 0          // a variable that is always 0
. generate one  = 1          // a variable that is always 1
. generate two  = 2          // a variable that is always 2
. stack  id y1 x1 zero one   id y2 zero x2 two, into(id y z1 z2 model)
. generate model2 = (model==2)
. logit y model2 z1 z2, vce(cluster id)
. test _b[z1] = _b[z2]
```

The coefficient of z1 represents the effect of x1 on y1, and similarly, z2 for the effect of x2 on y2. The variable model2 is a dummy for the "second model", which is included to allow the intercept in the second model to differ from that in the first model. The estimates of the coefficient of z1 and its standard error in the combined model are the same as the estimates of the coefficient of z1 and its standard error if we fit the model on the unstacked data.

```
. logit y1 x1, vce(robust)
```

The vce(cluster *clustvar*) option specified with the logit command for the stacked data ensures that the covariances of _b[z1] and _b[z2] are indeed estimated. This estimation ensures that the Wald test for the equality of the coefficients is correct. If we had not specified the vce(cluster *clustvar*) option, the (co)variance matrix of the coefficients would have been block-diagonal; that is, the covariances of _b[z1] and _b[z2] would have been 0. Then, test would have effectively used the invalid formula for the Wald test for two independent samples.

In this example, the two logit models were fit on the same data. The same setup would apply, without modification, when the two logit models were fit on overlapping data that resulted, for instance, if the y or x variables were missing in some observations.

The suest command allows us to obtain the above Wald test more efficiently by avoiding the data manipulation, obviating the need to fit a model with twice the number of coefficients. The test statistic produced by the above code fragment is *identical* to that obtained via suest on the original (unstacked) data:

```
. logit y1 x1
. estimates store M1
. logit y2 x2
. estimates store M2
. suest M1 M2
. test [M1]x1=[M2]x2
```

The stacking method can be applied not only to the testing of cross-model hypotheses for logit models but also to any estimation command that supports the vce(cluster *clustvar*) option. The stacking approach clearly generalizes to stacking more than two logit or other models, testing more general linear hypotheses, and testing nonlinear cross-model hypotheses (see [R] **testnl**). In all of these cases, suest would yield identical statistical results but at smaller costs in terms of data management, computer storage, and computer time.

Is suest nothing but a convenience command? No, there are two disadvantages to the stacking method, both of which are resolved via suest. First, if the models include ancillary parameters (in a regression model, the residual variance; in an ordinal response model, the cutpoints; and in lognormal survival-time regression, the time scale parameter), these parameters are constrained to be equal between the stacked models. In suest, this constraint is relaxed. Second, the stacking method does not generalize to compare different statistical models, such as a probit model and a regression model. As demonstrated in the previous section, suest can deal with this situation.

◁

Stored results

suest stores the following in e():

Scalars	
e(N)	number of observations
e(N_clust)	number of clusters
e(rank)	rank of e(V)
Macros	
e(cmd)	suest
e(eqnames#)	original names of equations of model #
e(names)	list of model names
e(wtype)	weight type
e(wexp)	weight expression
e(clustvar)	name of cluster variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(properties)	b V
Matrices	
e(b)	stacked coefficient vector of the models
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

Methods and formulas

The estimation of the simultaneous (co)variance of a series of k estimators is a nonstandard application of the sandwich estimator, as implemented by the command [P] **_robust**. You may want to read this manual entry before reading further.

The starting point is that we have fit k different models on the *same* data—partially overlapping or nonoverlapping data are special cases. We want to derive the *simultaneous* distribution of these k estimators, for instance, to test a cross-estimator hypothesis H_0. As in the framework of Hausman testing, H_0 will often be of the form that different estimators have the same probability limit under some hypothesis, while the estimators have different limits if the hypothesis is violated.

We consider (vector) estimators $\widehat{\boldsymbol{\beta}}_i$ to be defined as "the" solution of the estimation equations $\mathbf{G}_i$,

$$\mathbf{G}_i(\mathbf{b}_i) = \sum_j w_{ij}\mathbf{u}_{ij}(\mathbf{b}_i) = \mathbf{0}, \qquad i = 1, \dots, k$$

We refer to the $\mathbf{u}_{ij}$ as the "scores". Specifying some weights $w_{ij} = 0$ trivially accommodates for partially overlapping or even disjointed data. Under "suitable regularity conditions" (see White [1982; 1996] for details), the $\widehat{\boldsymbol{\beta}}_i$ are asymptotically normally distributed, with the variance estimated consistently by the sandwich estimator

$$V_i = \mathrm{Var}(\widehat{\boldsymbol{\beta}}_i) = \mathbf{D}_i^{-1} \sum_j w_{ij}\mathbf{u}_{ij}\mathbf{u}'_{ij}\, \mathbf{D}_i^{-1}$$

where $\mathbf{D}_i$ is the Jacobian of $\mathbf{G}_i$ evaluated at $\widehat{\boldsymbol{\beta}}_i$. In the context of maximum likelihood estimation, $\mathbf{D}_i$ can be estimated consistently by (minus) the Hessian of the log likelihood or by the Fisher information matrix. If the model is also well specified, the sandwiched term ($\sum_j w_{ij}\mathbf{u}_{ij}\mathbf{u}'_{ij}$) converges in probability to $\mathbf{D}_i$, so V_i may be consistently estimated by $\mathbf{D}_i^{-1}$.

To derive the simultaneous distribution of the estimators, we consider the "stacked" estimation equation,

$$\mathbf{G}(\widehat{\boldsymbol{\beta}}) = \left\{ \mathbf{G}_1(\widehat{\boldsymbol{\beta}}_1)' \;\; \mathbf{G}_1(\widehat{\boldsymbol{\beta}}_2)' \;\; \ldots \;\; \mathbf{G}_k(\widehat{\boldsymbol{\beta}}_k)' \right\}' = \mathbf{0}$$

Under "suitable regularity conditions" (see White [1996] for details), $\widehat{\boldsymbol{\beta}}$ is asymptotically *jointly* normally distributed. The Jacobian and scores of the simultaneous equation are easily expressed in the Jacobian and scores of the separate equations. The Jacobian of $\mathbf{G}$,

$$\mathbf{D}(\widehat{\boldsymbol{\beta}}) = \left. \frac{d\mathbf{G}(\boldsymbol{\beta})}{d\boldsymbol{\beta}} \right|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}}$$

is block diagonal with blocks $\mathbf{D}_1, \ldots, \mathbf{D}_k$. The inverse of $\mathbf{D}(\widehat{\boldsymbol{\beta}})$ is again block diagonal, with the inverses of $\mathbf{D}_i$ on the diagonal. The scores $\mathbf{u}$ of $\mathbf{G}$ are simply obtained as the *concatenated* scores of the separate equations:

$$\mathbf{u}_j = (\mathbf{u}_{1j}' \;\; \mathbf{u}_{2j}' \;\; \ldots \;\; \mathbf{u}_{kj}')'$$

Out-of-sample (that is, where $w_{ij} = 0$) values of the score variables are defined as 0 (thus we drop the i subscript from the common weight variable). The sandwich estimator for the asymptotic variance of $\widehat{\boldsymbol{\beta}}$ reads

$$V = \mathrm{Var}(\widehat{\boldsymbol{\beta}}) = \mathbf{D}(\widehat{\boldsymbol{\beta}})^{-1} \left(\sum_j w_j \mathbf{u}_j \mathbf{u}_j' \right) \mathbf{D}(\widehat{\boldsymbol{\beta}})^{-1}$$

Taking a "partitioned" look at this expression, we see that $V(\widehat{\boldsymbol{\beta}}_i)$ is estimated by

$$\mathbf{D}_i^{-1} \left(\sum_j w_j \mathbf{u}_{ij} \mathbf{u}_{ij}' \right) \mathbf{D}_i^{-1}$$

which is, yet again, the familiar sandwich estimator for $\widehat{\boldsymbol{\beta}}_i$ based on the separate estimation equation $\mathbf{G}_i$. Thus, considering several estimators simultaneously in this way does not affect the estimators of the asymptotic variances of these estimators. However, as a bonus of stacking, we obtained a sandwich-type estimate of the *covariance* V_{ih} of estimators $\widehat{\boldsymbol{\beta}}_i$ and $\widehat{\boldsymbol{\beta}}_h$,

$$V_{ih} = \mathrm{Cov}(\widehat{\boldsymbol{\beta}}_i, \widehat{\boldsymbol{\beta}}_h) = \mathbf{D}_i^{-1} \left(\sum_j w_j \mathbf{u}_{ij} \mathbf{u}_{hj}' \right) \mathbf{D}_h^{-1}$$

which is also obtained by White (1982).

This estimator for the covariance of estimators is an application of the cluster modification of the sandwich estimator proposed by Rogers (1993). Consider the stacked data format as discussed in the logit example, and assume that Stata would be able to estimate a "stacked model" in which different models apply to different observations, for example, a probit model for the first half, a regression model for the second half, and 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 Rogers's method, we have to sum the score statistics over the observations within a cluster. This step boils down to concatenating the score statistics at the level of the cluster.

We compare the sandwich estimator of the (co)variance V_{12} of two estimators with the estimator of variance $\widetilde{V}_{12}$ applied in the classic Hausman test. Hausman (1978) showed that if $\widehat{\boldsymbol{\beta}}_1$ is consistent under H_0 and $\widehat{\boldsymbol{\beta}}_2$ is efficient under H_0, then asymptotically

$$\operatorname{Cov}(\widehat{\boldsymbol{\beta}}_1, \widehat{\boldsymbol{\beta}}_2) = \operatorname{Var}(\widehat{\boldsymbol{\beta}}_2)$$

and so $\operatorname{var}(\widehat{\boldsymbol{\beta}}_1 - \widehat{\boldsymbol{\beta}}_2)$ is consistently estimated by $V_1 - V_2$.

Acknowledgment

`suest` was written by Jeroen Weesie of the Department of Sociology at Utrecht University, The Netherlands. This research is supported by grant PGS 50-370 by The Netherlands Organization for Scientific Research.

An earlier version of `suest` was published in the *Stata Technical Bulletin* (1999). The current version of `suest` is not backward compatible with the 1999 version because of the introduction of new ways to manage estimation results via the `estimates` command.

References

Arminger, G. 1990. “Testing against misspecification in parametric rate models”. In *Event History Analysis in Life Course Research*, edited by K. U. Mayer and N. B. Tuma, 146–158. Madison: University of Wisconsin Press.

Canette, I. 2014. Using gsem to combine estimation results. *The Stata Blog: Not Elsewhere Classified*. https://blog.stata.com/2014/08/18/using-gsem-to-combine-estimation-results/.

Clogg, C. C., E. Petkova, and A. Haritou. 1995. Statistical methods for comparing regression coefficients between models. *American Journal of Sociology* 100: 1261–1312. (With comments by P. D. Allison and a reply by C. C. Clogg, E. Petkova, and T. Cheng).

Gouriéroux, C. S., and A. Monfort. 1997. *Time Series and Dynamic Models*. Trans. ed. G. M. Gallo. Cambridge: Cambridge University Press.

Hausman, J. A. 1978. Specification tests in econometrics. *Econometrica* 46: 1251–1271. https://doi.org/10.2307/1913827.

Hausman, J. A., and D. L. McFadden. 1984. Specification tests for the multinomial logit model. *Econometrica* 52: 1219–1240. https://doi.org/10.2307/1910997.

Hausman, J. A., and P. A. Ruud. 1987. Specifying and testing econometric models for rank-ordered data. *Journal of Econometrics* 34: 83–104. https://doi.org/10.1016/0304-4076(87)90068-6.

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

Oberfichtner, M., and H. Tauchmann. 2021. Stacked linear regression analysis to facilitate testing of hypotheses across OLS regressions. *Stata Journal* 21: 411–429.

Rogers, W. H. 1993. sg16.4: Comparison of nbreg and glm for negative binomial. *Stata Technical Bulletin* 16: 7. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 82–84. College Station, TX: Stata Press.

Weesie, J. 1999. sg121: Seemingly unrelated estimation and the cluster-adjusted sandwich estimator. *Stata Technical Bulletin* 52: 34–47. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 231–248. College Station, TX: Stata Press.

White, H. L., Jr. 1982. Maximum likelihood estimation of misspecified models. *Econometrica* 50: 1–25. https://doi.org/10.2307/1912526.

———. 1996. *Estimation, Inference and Specification Analysis*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CCOL0521252806.

Also see

[R] **estimates** — Save and manipulate estimation results

[R] **hausman** — Hausman specification test

[R] **lincom** — Linear combinations of parameters

[R] **nlcom** — Nonlinear combinations of parameters

[R] **test** — Test linear hypotheses after estimation

[R] **testnl** — Test nonlinear hypotheses after estimation

[P] **_robust** — Robust variance estimates

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.