12.4 Estimating means and variances using the ml command

The problem: a Statalist user posed a question about the estimation of means and variances from subsamples of a normally distributed variable. He wanted to compute two nonlinear combinations of those estimates:

$$\beta = \frac{\sigma_1 - \sigma_2}{\sigma_1 + \sigma_2} \tag{12.1}$$

and

$$\alpha = 2\pi\sqrt{3}\left(\frac{\mu_1 - \mu_2}{\sigma_1 + \sigma_2}\right) \tag{12.2}$$

The user would also like to estimate the quantity α given the assumption of a common variance, $\sigma = \sigma_1 = \sigma_2$.

This may readily be accomplished by ml as long as the user is willing to make a distributional assumption. We set up a variant of mynormal_lf.ado[4] that allows for separate means and variances, depending on the value of an indicator variable, which we access with global macro subsample:

```
. type meanvar.ado
*! meanvar v1.0.1  CFBaum 11aug2008
program meanvar
        version 10.1
        args lnf mu1 mu2 sigma1 sigma2
        qui replace `lnf' = ln(normalden($ML_y1, `mu1', `sigma1')) ///
            if $subsample == 0
        qui replace `lnf' = ln(normalden($ML_y1, `mu2', `sigma2')) ///
            if $subsample == 1
end
```

We now may set up the estimation problem. As we do not have the user's data, we use auto.dta and consider foreign as the binary indicator:

```
. sysuse auto, clear
(1978 Automobile Data)
. global subsample foreign
. generate byte iota = 1
. ml model lf meanvar (mu1: price = iota) (mu2: price = iota) /sigma1 /sigma2
note: iota dropped because of collinearity
note: iota dropped because of collinearity
. ml maximize, nolog
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -879.18213
rescale:       log likelihood = -705.93677
rescale eq:    log likelihood = -701.24251
                                                Number of obs   =         74
                                                Wald chi2(0)    =          .
Log likelihood = -695.14898                     Prob > chi2     =          .
```

4. See Section 11.13.

```
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
mu1          |
       _cons |   6072.423   425.3414    14.28   0.000     5238.769    6906.077
-------------+----------------------------------------------------------------
mu2          |
       _cons |   6384.682   546.1422    11.69   0.000     5314.263    7455.101
-------------+----------------------------------------------------------------
sigma1       |
       _cons |    3067.18   300.7618    10.20   0.000     2477.698    3656.662
-------------+----------------------------------------------------------------
sigma2       |
       _cons |   2561.634   386.1808     6.63   0.000     1804.733    3318.534
------------------------------------------------------------------------------
. estimates store unconstr
```

For use below, we use `estimates store` ([R] **estimates**) to save the results of estimation under the name `unconstr`.

We can verify that these maximum likelihood estimates of the subsample means and variances are correct by estimating the subsamples with `ivreg2` (Baum et al. (2007)), available from the SSC Archive:

```
. ivreg2 price if !foreign
. ivreg2 price if foreign
```

Estimates of the desired quantities may be readily computed, in point and interval form, with `nlcom` ([R] **nlcom**):

```
. nlcom ([sigma1]_b[_cons] - [sigma2]_b[_cons]) / ///
>       ([sigma1]_b[_cons] + [sigma2]_b[_cons])
       _nl_1:  ([sigma1]_b[_cons] - [sigma2]_b[_cons]) / ([sigma1]_b[_cons] + [
> sigma2]_b[_cons])
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _nl_1 |    .089814    .089195     1.01   0.314    -.0850049    .2646329
------------------------------------------------------------------------------
.
. nlcom 2*_pi*sqrt(3) * (([mu1]_b[_cons] - [mu2]_b[_cons]) / ///
>       ([sigma1]_b[_cons] + [sigma2]_b[_cons]))
       _nl_1:  2*_pi*sqrt(3) * (([mu1]_b[_cons] - [mu2]_b[_cons]) / ([sigma1]_b
> [_cons] + [sigma2]_b[_cons]))
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _nl_1 |  -.6037236   1.339398    -0.45   0.652    -3.228896    2.021449
------------------------------------------------------------------------------
```