Robust variance estimation in panel data generalized least squares regression

Gueorgui I. Kolev
Middlesex University Business School,
London NW4 4BT, United Kingdom
joro.kolev@gmail.com

Abstract. Panel data generalized least squares (GLS) regression, with various forms of the GLS weighting matrix including unrestricted GLS weighting matrix, is implemented in Stata by the `[XT] xtgls` command. The `xtgls` command does not accept the robust option. This is to say, `xtgls` cannot automatically calculate a variance estimator robust to conditional heteroskedasticity and GLS weighting matrix misspecification. We show the relevant formulae and how the user can robustify the variance of the estimators by hand post `xtgls`. The tools that we use to obtain robust variance of the estimators post `xtgls` are `[P] _robust` and `[P] matrix score`.

1 Introduction, statement of the problem and a short digression on the history of econometric thought

Stata's `[XT] xtgls` estimates panel data generalized least squares (GLS) regression under various assumptions on the structure of the "cross sectional" correlation of the errors. These are attractive estimators when the "cross section" is small relative to the "time series" dimension – however what is the "cross section" and what is the "time series" is up to the econometrician to choose. Stata's `xtgls` cannot compute (conditional heteroskedasticity) robust variance matrix of the estimators post GLS regression. In this note, we show how the econometrician can robustify by hand the variance matrix of the estimators post GLS estimation by `xtgls`.

We place the terms "cross section" and "time series" in quotation marks because this is just our interpretation of the model. Mathematically speaking, we have two indices, $i = 1, 2.., I$ and $j = 1, 2.., J$. In the type of models estimated by `xtgls` observations are taken to be arbitrarily correlated and/or heteroskedastic across j, and the index j does not have an inherent ordering: one cannot say whether j and j' are close or far away in any sense. An independent or autoregressive structure of certain order across i is assumed, and the i index has certain ordering if autoregressive structure is assumed, we know whether i' immediately follows i or not. So we have an arbitrary dependence across j, and independence or model determined dependence across i. J is required to be somewhat small, I is required to be somewhat large for the estimators to have attractive asymptotic properties. An econometrician is free to choose which index she calls "time" and which she calls "cross section." The binding constraint is that if she wants to use unrestricted "cross sectional" correlation structure, the panel data have to be balanced, i.e., for each i the j index has to run up to and including J.

In the current mainstream econometric thought, an applied econometrician is supposed to have two options. She can either be agnostic about the error dependence structure and estimate ordinary least squares (OLS) followed by robust (to conditional heteroskedasticity and arbitrary within cluster correlation) variance, or she can make all the additional relevant restrictive assumptions which make the GLS estimator best linear unbiased (BLUE) and she can proceed with GLS estimation with non-robust variance. Stata programmers accordingly did not program the robust option in `xtgls`.

We have been through similar debate in the Stata community in the context of fixed and random effects (equi-correlated structure) panel data models. Stata 7 did not accept the robust option in fixed effects (`xtreg, fe robust` was not a valid syntax) or random effects (`xtreg, re robust` was not a valid syntax) panel data estimation, and there were econometricians issuing stark warnings that robust variance post random or fixed effect estimation does not make any sense. Hence the rest of us who did not see the estimation problem quite this way had to find work-around solutions. The `areg` was accepting the robust and cluster options for fixed effects estimation, or we could transform all the variables by the random effects GLS transformation so that we can fit random effects GLS by `regress`, which was accepting the robust and cluster options too. In the first draft of Kolev (2012), which one of us wrote in Stata 7/8, work-around solutions for random effect estimation (quasi-time demeaning, Wooldridge, 2002, eq. 10.75) had to be used, and in later drafts, Stata 9 and up, specifying `xtreg, re robust cluster(clusterID)` was permitted.

There is a different school of statistical thought originating in the Generalized Estimating Equations literature (Liang & Zeger, 1986; Zeger, Liang & Albert, 1988). In this literature statisticians use routinely GLS, however with the understanding that the assumed GLS covariance matrix of the errors might be misspecified (in this context the GLS weighting matrix is called "working correlation matrix"), and hence they proceed with robust variance estimation post GLS (to guard against possible misspecification of the GLS weighting matrix, and to still obtain valid asymptotic inference). Note that when we are being agnostic and use OLS followed by robust and cluster options, we are deliberately misspecifying the error correlation matrix to have a scalar multiplied by identity matrix structure. Logically, a more reasonable approach would be to have a best guess at the cross sectional correlation and heteroskedasticity, and still to follow the GLS analysis by robust variance estimation.

To summarize, first, an econometrician might want to collect gains from GLS by specifying a good, reasonable cross sectional error structure, and still obtain valid asymptotic inference if this matrix is misspecified. Second, an econometrician might want to carry out panel data GLS and calculate robust to conditional heteroskedasticity variance matrix. For any of these two reasons, the approach we outline below might be deemed desirable and attractive.

2 Formulae

Summaries of formulae of GLS with robust variance matrix can be found in Wooldridge (2002, eq.7.49, eq. 10.38, and Section 10.4.3), Cameron & Miller (2010, eq.13) and Cameron & Miller (2013, eq. 15). We have the model for a random draw $i = 1, 2.., I$ from a population of interest, and on J correlated measurements, $j = 1, 2, .., J$. Note at this point that the indices are confusing. For example in Wooldridge (2002) the interpretation is that we have panel data where we have random sampling across i, i are the cross sectional units say people, firms, counties etc., and the J measurements on each unit are the time periods in the panel data. In Cameron & Miller the i would be the independent clusters, and j would be the correlated measurements within a cluster.

In Stata `xtgls`, i are the *time periods*, and j are the correlated measurements within a time period. Therefore when you specify `xtset panelVar otherVar`, it is `panelVar` the j, across which you have correlated measurements, and it is the `otherVar` the i over which you assume random sampling.[1] The typical symptom of incorrectly set panel data structure is "too big standard errors" – the estimator in `xtgls` is attractive with "small" J, equivalent to `panelVar`, and large I equivalent to `otherVar`. If you obtain too large standard errors, it might mean that you have specified too few is and too many js.

The models is

$$y_{ij} = x'_{ij}\beta + u_{ij}, \quad x_{ij} \text{ and } \beta \text{ are } K \times 1, \quad i = 1, 2.., I, \quad j = 1, 2.., J, \tag{1}$$

which can be stacked for a single i as

$$y_i = X_i\beta + u_i \equiv \begin{pmatrix} y_{i1} \\ y_{i2} \\ .. \\ y_{iJ} \end{pmatrix} = \begin{pmatrix} x'_{i1} \\ x'_{i2} \\ .. \\ x'_{iJ} \end{pmatrix} \beta + \begin{pmatrix} u_{i1} \\ u_{i2} \\ .. \\ u_{iJ} \end{pmatrix}, \tag{2}$$

where y_i is $J \times 1$, X_i is $J \times K$, β is $K \times 1$, and u_i is $J \times 1$.

If we assume that each element of u_i is uncorrelated with each element of X_i, $\mathrm{E}(X_i \otimes u_i) = 0$, plus the assumption that $\Omega \equiv \mathrm{E}(u_i u_i')$ is positive definite and $\mathrm{E}(X_i'\Omega^{-1}X_i)$ is invertible, then the Feasible GLS estimator

$$\widehat{\beta}_{\text{gls}} = \left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}X_i\right)^{-1} \left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}y_i\right), \quad \widehat{\Omega} = I^{-1}\sum_{i=1}^{I} \widehat{u}_i\widehat{u}_i', \tag{3}$$

where $\widehat{u}_i$ is the OLS residual vector, is consistent. Without further requirements on u_i, the robust variance estimator

$$\mathrm{Var}(\widehat{\beta}_{\text{gls}}) = \left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}X_i\right)^{-1} \left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}\widehat{\widehat{u}_i\widehat{u}_i'}\widehat{\Omega}^{-1}X_i\right) \left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}X_i\right)^{-1}, \tag{4}$$

1. We are oversimplifying here because in Stata `xtgls` the interpretation of i is time, and `xtgls` allows parametric dependence across i, you can specify say first order autoregressive process for the error across i.

where $\widehat{u_i} = y_i - X_i\widehat{\beta}_{\text{gls}}$ is the GLS residual, would be generally valid.

Remarks:

- At this point we are in the way of reasoning of the generalized estimating equations literature (Liang & Zeger, 1986) – the Feasible GLS estimator in eq.(3) is not guaranteed to be "best" under our set of minimal assumptions. However the Feasible GLS estimator is more natural than the OLS estimator, given that we allow the J measurements on unit i to be arbitrarily correlated.

- Note that we can also restrict the $\widehat{\Omega}$ to have diagonal structure, which would correspond to J measurements which are uncorrelated but unconditionally heteroskedastic.

- We can also restrict the $\widehat{\Omega}$ to have a scalar multiplied by the identity matrix structure, which would correspond to J measurements which are uncorrelated and unconditionally homoskedastic. In fact this would be the OLS estimator.

If we further impose the conditional homoskedasticity assumption $\text{E}(X_i'\Omega^{-1}u_iu_i'\Omega^{-1}X_i) = \text{E}(X_i'\Omega^{-1}X_i)$, then the Feasible GLS in eq.(3) will be more efficient than any other estimator under the orthogonality condition $\text{E}(X_i \otimes u_i) = 0$, and we can use the standard conditionally homoskedastic formula for the variance of eq.(3),

$$\text{Var}(\widehat{\beta}_{\text{gls}}) = \left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}X_i\right)^{-1}. \tag{5}$$

Remarks:

- Stata uses eq.(5) to compute the variance matrix of the estimators in `xtgls`. Stata cannot immediately compute the variance matrix in eq.(4).

- Stata computes conditionally homoskedastic variance with arbitrary unconditional cross sectional correlation and heteroskedasticity with the option `xtgls, panels(correlated)`, i.e., eq.(3) with the expression of $\widehat{\Omega}$ after the comma.

- Stata computes conditionally homoskedastic variance with arbitrary unconditional cross sectional heteroskedasticity with the option `xtgls, panels(heteroskedastic)`, i.e., eq.(3) with the $\widehat{\Omega}$ constrained to be a diagonal matrix.

- Stata computes conditionally homoskedastic variance with unconditional homoskedasticity with the option `xtgls, panels(iid)` , i.e., eq.(3) with the $\widehat{\Omega}$ constrained to be a scalar multiplied by the identity matrix.

3 The tools that we will use

This section is best read after looking at the code in the following section. It is a lot harder to explain what we do, than to actually do it. The reader is advised to skip this section on first reading.

The programmers command `[P] _robust` calculates robust variance matrices. Post estimation it expects the user to provide a variable name, say `resid_ij`, which holds the residual from the estimator whose variance we want to robustify. Upon issuing a call

`_robust resid_ij, cluster(i)` *`options`*

the command replaces the variance matrix posted in `e(V)` (see `[P] ereturn`) by `e(V)`$\left(\sum_{i=1}^{I} X_i'\widehat{u_i}\widehat{u_i'}X_i\right)$`e(V)`, where $\widehat{u_i'} = [resid\_i1 \quad resid\_i2 \quad .. \quad resid\_iJ]$. The outer matrix in `e(V)` needs no further modifications, as previously discussed the `xtgls` uses eq.(5) to calculate `e(V)`, and this matches the outer matrix we need in eq.(4). However the inside matrix we need is $\left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}\widehat{u_i}\widehat{u_i'}\widehat{\Omega}^{-1}X_i\right)$, see eq.(4) and there is a difference between what we need and what `_robust` would provide.

Mathematically, the operation $\widehat{\Omega}^{-1}\widehat{u_i}$ takes the residual vector $\widehat{u_i}$ and transforms it into a new residual vector $\tilde{u}_i$, which we will call a *weighted residual* from now on. The first element of the $\tilde{u}_i$ is the inner product of the first row in $\widehat{\Omega}^{-1}$ and $\widehat{u_i}$, the second element of the $\tilde{u}_i$ is the inner product of the second row in $\widehat{\Omega}^{-1}$ and $\widehat{u_i}$, and so on and so forth, until the Jth element of the $\tilde{u}_i$ is the inner product of the Jth row in $\widehat{\Omega}^{-1}$ and $\widehat{u_i}$.

Our plan is to calculate by hand $\tilde{u}_i = \widehat{\Omega}^{-1}\widehat{u_i}$ and to pass it to `_robust`. In this way, `_robust` will calculate the inside matrix as

$$\left(\sum_{i=1}^{I} X_i'\tilde{u}_i\tilde{u}_i{}'X_i\right) = \left(\sum_{i=1}^{I} X_i'\widehat{\Omega}^{-1}\widehat{u_i}\widehat{u_i'}\widehat{\Omega}^{-1}X_i\right),$$

and this is what we need.

The inner products in $\tilde{u}_i$ can be viewed also as linear combinations of $\widehat{u_i}$ where the weights are given in the respective rows of $\widehat{\Omega}^{-1}$. The `[P] matrix score` *`newvarname`* `= b` construct takes a **b** row vector which has respective variable names as column names, and forms a linear combination of the variables in the column names of **b**, where the weights are given by **b**.

4 Stata code for obtaining robust variance of estimators post `xtgls`

We will use as an example the dataset that is used in `[XT] xtgls` entry, p.5. The Stata log is presented, where the commands are followed by comments. (In a Stata do file, what follows "//" is not executed, so comments can be placed after "//". Statements after "*" are not executed either, so these are comments too.)

```
. use http://www.stata-press.com/data/r11/invest2 // Adjust r# for the version of Stata you are using.

.
. xtset company time      // company is the j index, time is the i index
       panel variable:  company (strongly balanced)
        time variable:  time, 1 to 20
                delta:  1 unit

.
.         * We quietly run the three xtgls estimators, store the results for later tabulation (see  [R] estimates) and
.         * display the matrix with cross sectional correlations. We called this matrix Omega, it is in e(Sigma).
            * To see what an estimation command leaves behind, type ereturn list,
            * or look up towards the end of the command manual entry.
.         * The inverse of e(Sigma) is the GLS weighting matrix.
.
. qui xtgls invest market stock, panels(correlated)

. estimates store Corr

. matrix list e(Sigma)    // Note the structure of the matrix, arbitrary cross sectional correlation is estimated.

symmetric e(Sigma)[5,5]
             _ee        _ee2        _ee3        _ee4        _ee5
 _ee   9410.9061
_ee2  -168.04631   755.85077
_ee3  -1915.9538  -4163.3434    34288.49
_ee4  -1129.2896  -80.381742   2259.3242   633.42367
_ee5   258.50132    4035.872  -27898.235  -1170.6801   33455.511

. matrix ScorrInv = invsym(e(Sigma))      // We take the inverse of the matrix for later use.

. predict double corr_res // xtgls cannot predict residuals. So we predict the xb first
(option xb assumed; fitted values)

. replace corr_res = invest - corr_res    // and then we form the residual as (y - xb).
(100 real changes made)

.
. qui xtgls invest market stock, panels(hetero)

. estimates store Hetero

. matrix list e(Sigma)    // Notice that the matrix is constrained to be diagonal.

symmetric e(Sigma)[5,5]
           c1         c2         c3         c4         c5
r1  9410.9061
r2          0  755.85077
r3          0          0   34288.49
r4          0          0          0  633.42367
r5          0          0          0          0  33455.511

. matrix SheteroInv = invsym(e(Sigma))

. predict double hetero_res
(option xb assumed; fitted values)

. replace hetero_res = invest - hetero_res
(100 real changes made)

.
. qui xtgls invest market stock, panels(iid)

. estimates store Homo

. matrix list e(Sigma)    // Notice that the matrix is a scalar multiplied by identity matrix.

symmetric e(Sigma)[5,5]
```

```
          c1         c2         c3         c4         c5
r1  15708.836
r2          0  15708.836
r3          0          0  15708.836
r4          0          0          0  15708.836
r5          0          0          0          0  15708.836

. matrix ShomoInv = invsym(e(Sigma))

. predict double homo_res
(option xb assumed; fitted values)

. replace homo_res = invest - homo_res
(100 real changes made)

.
.
.         * We reshape the data to wide. matrix score operates on variables.

. reshape wide  invest market stock corr_res hetero_res homo_res, i(time) j(company)
(note: j = 1 2 3 4 5)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                      100   ->      20
Number of variables                  11   ->      34
j variable (5 values)           company   ->   (dropped)
xij variables:
                                 invest   ->   invest1 invest2 ... invest5
                                 market   ->   market1 market2 ... market5
                                  stock   ->   stock1 stock2 ... stock5
                               corr_res   ->   corr_res1 corr_res2 ... corr_res5
                             hetero_res   ->   hetero_res1 hetero_res2 ... hetero_res5
                               homo_res   ->   homo_res1 homo_res2 ... homo_res5
-----------------------------------------------------------------------------

.
.         * We need to give the residual names as column names of the respective inverted
.         * cross sectional correlation GLS matrix. matrix colnames names columns in a matrix,
            * see  [P] matrix rownames.

. matrix colnames ScorrInv = corr_res1 corr_res2 corr_res3 corr_res4 corr_res5

.
. matrix colnames SheteroInv = hetero_res1 hetero_res2 hetero_res3 hetero_res4 hetero_res5

.
. matrix colnames ShomoInv = homo_res1 homo_res2 homo_res3 homo_res4 homo_res5

.
.         * We cycle and generate the weighted residual element by element.
.         * matrix score expect a row vector as an input. Therefore we have to
.         * extract the respective rows of the GLS matrix and place them in tempvec.
            * For how loops work in Stata, see Cox (2002,2003).
            * `i´ is the local macro which takes the values 1,2,..5 consequtively.
            * Matrix[3,1...] is a matrix extraction operation (extracts the 3rd row), see [P] matrix define.

.         forvalues i = 1/5 {
  2.                matrix tempvec = ScorrInv[`i´,1...]
  3.                matrix score corr_reswt`i´ = tempvec
  4.
.                matrix tempvec = SheteroInv[`i´,1...]
  5.                matrix score hetero_reswt`i´ = tempvec
  6.
.                matrix tempvec = ShomoInv[`i´,1...]
  7.                matrix score homo_reswt`i´ = tempvec
  8.        }
```

```
.
.
.                 * We reshape back to long, this is the format that xtgls expects.

. reshape long invest market stock corr_res hetero_res homo_res corr_reswt hetero_reswt homo_reswt, i(time) j(company)
(note: j = 1 2 3 4 5)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                       20   ->     100
Number of variables                  49   ->      14
j variable (5 values)                     ->   company
xij variables:
            invest1 invest2 ... invest5   ->   invest
            market1 market2 ... market5   ->   market
               stock1 stock2 ... stock5   ->   stock
      corr_res1 corr_res2 ... corr_res5   ->   corr_res
hetero_res1 hetero_res2 ... hetero_res5   ->   hetero_res
      homo_res1 homo_res2 ... homo_res5   ->   homo_res
corr_reswt1 corr_reswt2 ... corr_reswt5   ->   corr_reswt
hetero_reswt1 hetero_reswt2 ... hetero_reswt5->hetero_reswt
homo_reswt1 homo_reswt2 ... homo_reswt5   ->   homo_reswt
-----------------------------------------------------------------------------

.
. qui xtgls invest market stock, panels(correlated)

.         * The call to _robust provides the weighted residual as an input.
.         * The minus(0) option specified that we do not want degrees of freedom adjustment.

. _robust corr_reswt, cluster(time) minus(0)

. estimates store Corr_Robust

.
. qui xtgls invest market stock, panels(hetero)

. _robust hetero_reswt, cluster(time) minus(0)

. estimates store Hetero_Robust

.
. qui xtgls invest market stock, panels(iid)

. _robust homo_reswt, cluster(time) minus(0)

. estimates store Homo_Robust

.
.         * We tabulate the results.

. estimates table Corr Corr_Robust Hetero Hetero_Robust Homo Homo_Robust, b se ///
> title(Dependent variable is invest, panels are assumed to have the cross sectional structure in the column name)

Dependent variable is invest, panels are assumed to have the cross sectional structure in the column name

-------------------------------------------------------------------------------------------
    Variable |    Corr      Corr_Rob~t     Hetero     Hetero_R~t      Homo      Homo_Rob~t
-------------+-----------------------------------------------------------------------------
      market |  .09618945    .09618945    .09499051    .09499051    .10508541    .10508541
             |  .00547516    .00582834    .00740898     .0060503    .01120586    .00847444
       stock |  .30953206    .30953206    .33781285    .33781285    .30536554    .30536554
             |  .01798509    .01622246     .0302254    .03263735    .04285023    .04418531
       _cons | -38.361276   -38.361276   -36.253703   -36.253703   -48.029736   -48.029736
             |  5.3448707    5.7061914    6.1243632    5.8184242    21.155509    11.500451
-------------------------------------------------------------------------------------------
                                                                               legend: b/se
```

```
.
.                 * Robust variance post xtgls with unconditionally homoskedastic cross sectional
.                 * error structure is equivalent to OLS with robust variance clustered by time.
.                 * For this case we can verify our results by an existing estimator.

. qui xtgls invest market stock, panels(iid)

. _robust homo_reswt, cluster(time) minus(0)

. matrix list e(V)

symmetric e(V)[3,3]
            market       stock       _cons
market   .00007182
 stock  -.00024263   .00195234
 _cons  -.03038361  -.22883965   132.26038

.
. qui reg invest market stock, robust cluster(time)

.         * Regress does a degrees of freedom adjustment, which the formula below undoes.
.         * See p. 21 in  [R] regress.

. matrix Vreg = e(V)*((e(N_clust)-1)*(e(N)-e(df_m)-1)/(e(N)-1)/e(N_clust))

. matrix list Vreg

symmetric Vreg[3,3]
            market       stock       _cons
market   .00007182
 stock  -.00024263   .00195234
 _cons   -.0303836  -.22883967   132.26038

.
.         * We observe that the two variance matrices are the same.
```

We close this section with an important final remark. The call to `_robust` replaces the variance matrix of the estimators. Therefore we can proceed using all post estimation facilities as usual, for example we can use `test` to carry out Wald tests post estimation.

```
.         * Note that _robust replaces the posted variance matrix
.         * and we can proceed as usual, using all post estimation facilities, such as test.

. xtgls invest market stock, panels(correlated)   // This is xtgls with standard, conditionally homoskedastic variance.

Cross-sectional time-series FGLS regression

Coefficients:  generalized least squares
Panels:        heteroskedastic with cross-sectional correlation
Correlation:   no autocorrelation

Estimated covariances      =        15          Number of obs      =       100
Estimated autocorrelations =         0          Number of groups   =         5
Estimated coefficients     =         3          Time periods       =        20
                                                Wald chi2(2)       =   1285.19
                                                Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      invest |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      market |   .0961894   .0054752    17.57   0.000     .0854583    .1069206
       stock |   .3095321   .0179851    17.21   0.000     .2742819    .3447822
       _cons |  -38.36128   5.344871    -7.18   0.000    -48.83703   -27.88552
------------------------------------------------------------------------------

.
```

```
. _robust corr_reswt, cluster(time) minus(0)              // With this call we robustify the variance.

.
. xtgls            // We replay the xtgls call, note that the standard errors are different - they are the robust
                 // standard errors, compare with the table above containing all estimators.

Cross-sectional time-series FGLS regression

Coefficients:  generalized least squares
Panels:        heteroskedastic with cross-sectional correlation
Correlation:   no autocorrelation

Estimated covariances      =        15          Number of obs      =       100
Estimated autocorrelations =         0          Number of groups   =         5
Estimated coefficients     =         3          Time periods       =        20
                                                Wald chi2(2)       =   1285.19
                                                Prob > chi2        =    0.0000

                                  (Std. Err. adjusted for 20 clusters in time)
------------------------------------------------------------------------------
             |               Robust
      invest |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      market |   .0961894   .0058283    16.50   0.000      .0847661    .1076128
       stock |   .3095321   .0162225    19.08   0.000      .2777366    .3413275
       _cons |  -38.36128   5.706191    -6.72   0.000     -49.54521   -27.17735
------------------------------------------------------------------------------

. test market stock       // test works as usual to carry out Wald test of hypotheses.

 ( 1)  market = 0
 ( 2)  stock = 0

           chi2(  2) = 1470.43
         Prob > chi2 =    0.0000

.
. test market=stock

 ( 1)  market - stock = 0

           chi2(  1) =  112.47
         Prob > chi2 =    0.0000
```

5 Conclusion and directions for further analysis

Robust variance estimation has been an important positive development in the practice of statistics and econometrics (Wooldridge, 2002; Stock, 2010), and Stata has been a forerunner in introducing routines that calculate robust variances in the context of single equation models and panel data (Rogers, 1994).

We show here how robust variance of estimators can be calculated when using `xtgls`. We show how `_robust` and `matrix score` can be used to robustify GLS regression variances.

There are two issues we have not addressed. First, `xtgls` can also compute models in which the error terms are not only "cross sectionally" correlated and (unconditionally) heteroskedastic, but also autocorrelated across "time." How do we robustify inference when both cross sectional and time series dependence is assumed? We have appealed to results in Wooldridge (2002) which assume independence (random sampling) across "time" (i.e., the index i). Can these results still be used if we assume that there is an autoregression in the time series dimension? (Our guess is that the answer is positive, one can still assume that the error in the autoregression is independent across time, so that with the Cochrane-Orcutt (1949) transformation one achieves independence across time of the transformed error.) Second, as mentioned before, `xtgls` can easily be used on panels with many subjects and few time periods – one just has to reverse the indices `xtset timeVar panelVar`, and GLS with unconditional heteroskedasticity and arbitrary correlation across time becomes available. The problem is that `xtgls` requires balanced panels. This is another interesting question: what to do when we want to have unrestricted GLS matrix across time, but we do not observe all units over all time periods?

6 References

Cameron, A. C., & Miller, D. L. (2010). Robust inference with clustered data. Handbook of empirical economics and finance, 1-28.

Cameron, A. C., & Miller, D. L. (2013). A Practitioner's Guide to Cluster-Robust Inference. Forthcoming in Journal of Human Resources.

Cochrane, D., & Orcutt, G. H. (1949). Application of least squares regression to relationships containing auto-correlated error terms. Journal of the American Statistical Association, 44(245), 32-61.

Cox, N. J. (2002). Speaking Stata: How to face lists with fortitude. Stata Journal, 2(2), 202-222.

Cox, N. J. (2003). Speaking Stata: Problems with lists. Stata Journal, 3(2), 185-202.

Kolev, G. I. (2012). Underperformance by female CEOs: A more powerful test. Economics Letters, 117(2), 436-440.

Liang, K. Y., & Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. Biometrika, 73(1), 13-22.

Rogers, W. (1994). Regression standard errors in clustered samples. Stata technical bulletin, 3(13).

Stock, J. H. (2010). The other transformation in econometric practice: Robust tools for inference. The Journal of Economic Perspectives, 24(2), 83-94.

Wooldridge, J. M. (2002). Econometric analysis of cross section and panel data. MIT press.

Zeger, S. L., Liang, K. Y., & Albert, P. S. (1988). Models for longitudinal data: a generalized estimating equation approach. Biometrics, 1049-1060.