Estimation and forecasting: OLS, IV, IV-GMM

Christopher F Baum

Boston College and DIW Berlin

IMF Institute for Capacity Development, October 2018

Linear regression

A key tool in multivariate statistical inference is *linear regression*, in which we specify the conditional mean of a response variable y as a linear function of k independent variables

$$E\left[y|x_1, x_2, \ldots, x_k\right] = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_{i,k} \tag{1}$$

The conditional mean of y is a function of $x_1, x_2, \ldots, x_k$ with fixed parameters $\beta_1, \beta_2, \ldots, \beta_k$. Given values for these βs the linear regression model predicts the average value of y in the population for different values of $x_1, x_2, \ldots, x_k$.

This population regression function specifies that a set of k regressors in X and the stochastic disturbance u are the determinants of the response variable (or regressand) y. The model is usually assumed to contain a constant term, so that x_1 is understood to equal one for each observation. We may write the linear regression model in matrix form as

$$y = X\beta + u \tag{2}$$

where $X = \{x_1, x_2, \ldots, x_k\}$, an $N \times k$ matrix of sample values.

The key assumption in the linear regression model involves the relationship in the population between the regressors X and u. We may rewrite Equation (2) as

$$u = y - X\beta \tag{3}$$

We assume that

$$E(u \mid X) = 0 \tag{4}$$

i.e., that the u process has a *zero conditional mean*. This assumption states that the unobserved factors involved in the regression function are not related in any systematic manner to the observed factors. This approach to the regression model allows us to consider both non-stochastic and stochastic regressors in X without distinction; all that matters is that they satisfy the assumption of Equation (4).

We may use the zero conditional mean assumption (Equation (4)) to define a *method of moments* estimator of the regression function. Method of moments estimators are defined by *moment conditions* that are assumed to hold on the population moments. When we replace the unobservable population moments by their sample counterparts, we derive feasible estimators of the model's parameters.

The zero conditional mean assumption gives rise to a set of k moment conditions, one for each x. In the population, each regressor x is assumed to be unrelated to u, or have zero covariance with u. We may then substitute calculated moments from our sample of data into the expression to derive a method of moments estimator for β:

$$\begin{aligned} X'u &= 0 \\ X'(y - X\beta) &= 0 \qquad (5) \end{aligned}$$

Substituting calculated moments from our sample into the expression and replacing the unknown coefficients β with estimated values *b* in Equation (5) yields the *ordinary least squares* (OLS) estimator

$$\begin{aligned} X'y - X'Xb &= 0 \\ b &= (X'X)^{-1}X'y \end{aligned} \tag{6}$$

We may use *b* to calculate the regression residuals:

$$e = y - Xb \tag{7}$$

Given the solution for the vector b, the additional parameter of the regression problem σ_u^2, the population variance of the stochastic disturbance, may be estimated as a function of the regression residuals e_i:

$$s^2 = \frac{\sum_{i=1}^{N} e_i^2}{N-k} = \frac{e' e}{N-k} \qquad (8)$$

where $(N-k)$ are the residual *degrees of freedom* of the regression problem. The positive square root of s^2 is often termed the standard error of regression, or standard error of estimate, or root mean square error. Stata uses the last terminology and displays s as `Root MSE`.

To learn more about the sampling distribution of the OLS estimator, we must make some additional assumptions about the distribution of the stochastic disturbance u_i. In classical statistics, the u_i were assumed to be independent draws from the same normal distribution. The modern approach to econometrics drops the normality assumptions and simply assumes that the u_i are independent draws from an identical distribution ($i.i.d.$).

The normality assumption was sufficient to derive the exact finite-sample distribution of the OLS estimator. In contrast, under the *i*.*i*.*d*. assumption, one must use large-sample theory to derive the sampling distribution of the OLS estimator. The sampling distribution of the OLS estimator can be shown to be approximately normal using large-sample theory. We refer this variance-covariance matrix of the estimator as a VCE.

Under the assumption of *i*.*i*.*d*. errors, the celebrated Gauss–Markov theorem holds. Within the class of linear, unbiased estimators the OLS estimator has the smallest sampling variance, or the greatest precision.

In that sense, it is *best*, so that “ordinary least squares is BLUE” (the *best linear unbiased estimator*) for the parameters of the regression model. If we restrict our consideration to unbiased estimators which are linear in the parameters, we cannot find a more *efficient* estimator.

A macroeconomic example

As an illustration, we present regression estimates from a simple macroeconomic equation constructed with US quarterly data. The model, of the quantity of light weight vehicle sales, should not be taken too seriously. Its purpose is only to illustrate the workings of regression in Stata.

In the initial form of the model, we include as regressors the prime rate, consumers' debt service burden, the rate of growth of real GDP and the unemployment rate.

We present the descriptive statistics with `summarize`, then proceed to fit a regression equation.

Try it out:

```
. bcuse macro14, nodesc

. desc  altsales frprime fsdebt gdprdot ur

              storage   display    value
variable name   type    format     label      variable label
--------------------------------------------------------------------------------------------------------

altsales        double  %10.0g                Light weight vehicle sales
frprime         double  %10.0g                Bank Prime Rate
fsdebt          double  %10.0g                Debt Service Burden: Total, (% of Disposable Personal
                                                Income, SA) for United Sta
gdprdot         double  %10.0g                = 400 * (log(GDPR) - log(L.GDPR))
ur              float   %9.0g                 Unemployment rate

. summ  altsales frprime fsdebt gdprdot ur if tin(1980q3,)

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
    altsales |        138    14.57842     2.14364   9.117333     18.527
     frprime |        138    7.714058    3.565798       3.25      20.32
      fsdebt |        138    11.39978    .8665457       9.83      13.18
     gdprdot |        138     2.71522    2.847371   -8.54093   9.024685
          ur |        138    .0645556     .016463   .0391842   .1067659
```

The `regress` command, like other Stata estimation commands, requires us to specify the response variable followed by a *varlist* of the explanatory variables.

Try it out:

```
. reg altsales frprime L2.fsdebt L.(gdprdot ur), vsquish
      Source |       SS           df       MS      Number of obs   =       138
-------------+----------------------------------   F(4, 133)       =     69.00
       Model |  424.816508         4  106.204127   Prob > F        =    0.0000
    Residual |  204.724794       133  1.53928416   R-squared       =    0.6748
-------------+----------------------------------   Adj R-squared   =    0.6650
       Total |  629.541302       137  4.59519198   Root MSE        =    1.2407
------------------------------------------------------------------------------
    altsales |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     frprime |  -.2218317    .031051    -7.14   0.000    -.2832495   -.1604139
      fsdebt |
         L2. |  -.2698066    .150925    -1.79   0.076    -.5683305    .0287173
     gdprdot |
         L1. |   .2047546    .036702     5.58   0.000     .1321594    .2773498
          ur |
         L1. |  -93.76327   7.522177   -12.46   0.000    -108.6418   -78.8847
       _cons |   24.89216   2.121014    11.74   0.000     20.69688    29.08745
------------------------------------------------------------------------------
```

The header of the regression output describes the overall model fit, while the table presents the point estimates, their precision, and interval estimates.

The regression output for this model includes the analysis of variance (ANOVA) table in the upper left, where the two sources of variation are displayed as `Model` and `Residual`. The `SS` are the Sums of Squares, with the `Residual SS` corresponding to $e'e$ and the Total `Total SS` to $\tilde{y}'\tilde{y}$ in equation (10) below.

The next column of the table reports the `df`: the degrees of freedom associated with each sum of squares. The degrees of freedom for total `SS` are $(N - 1)$, since the total `SS` has been computed making use of one sample statistic, $\bar{y}$. The degrees of freedom for the model are $(k - 1)$, equal to the number of slopes (or explanatory variables): one fewer than the number of estimated coefficients due to the constant term.

As discussed above, the model `SS` refer to the ability of the four regressors to jointly explain a fraction of the variation of y about its mean (the total `SS`). The residual degrees of freedom are $(N - k)$, indicating that $(N - k)$ residuals may be freely determined and still satisfy the constraint posed by the first normal equation of least squares that the regression surface passes through the multivariate point of means $(\bar{y}, \bar{X}_2, \ldots, \bar{X}_k)$:

$$\bar{y} = b_1 + b_2\bar{X}_2 + b_3\bar{X}_3 + \cdots + b_k\bar{X}_k \tag{9}$$

In the presence of the constant term b_1 the first normal equation implies that $\bar{e} = \bar{y} - \sum_i \bar{X}_i b_i$ must be identically zero. It must be stressed that this is not an assumption. This is an algebraic implication of the least squares technique which guarantees that the sum of least squares residuals (and their mean) will be very close to zero.

The last column of the ANOVA table reports the `MS`, the Mean Squares due to regression and error, which are merely the `SS` divided by the `df`. The ratio of the `Model MS` to `Residual MS` is reported as the ANOVA *F*-statistic, with numerator and denominator degrees of freedom equal to the respective `df` values.

This ANOVA *F* statistic is a test of the null hypothesis that the slope coefficients in the model are jointly zero: that is, the null model of $y_i = \mu + u_i$ is as successful in describing y as is the regression alternative. The `Prob > F` is the tail probability or *p*-value of the *F*-statistic. In this example we may reject the null hypothesis at any conventional level of significance.

The upper right section of `regress` output contains several *goodness of fit* statistics. These statistics measure the degree to which an estimated model can explain the variation of the response variable *y*.

The `Root MSE` for the regression of 1.24 is in the units of the response variable *y* (millions of vehicles). It can be compared with the mean of that variable, 14.58.

Other things equal, we should prefer a model with a better fit to the data. With the principle of parsimony in mind, we also prefer a simpler model. The mechanics of regression imply that a model with a very large number of regressors can explain *y* arbitrarily well.

Given the least squares residuals, the most common measure of goodness of fit, regression R^2, may be calculated (given a constant term in the regression function) as

$$R^2 = 1 - \frac{e'e}{\tilde{y}'\tilde{y}} \tag{10}$$

where $\tilde{y} = y - \bar{y}$: the regressand with its sample mean removed. This emphasizes that the object of regression is not the explanation of $y'y$, the raw sum of squares of the response variable y. That would amount to explaining why $Ey \neq 0$, which is often not a very interesting question. Rather, the object is to explain the variations in the response variable. That variable may be always positive—such as the level of GDP—so that it is not sensible to investigate whether the average of the response variable might be zero.

With a constant term in the model, the least squares approach seeks to explain the largest possible fraction of the sample *variation* of *y* about its mean (and not the associated *variance!)* The null model to which the estimated model is being contrasted is $y = \mu + u$ where μ is the population mean of *y*.

In estimating a regression, we are trying to determine whether the information in the regressors X is useful. Is the conditional expectation $E(y|X)$ more informative than the unconditional expectation $Ey = \mu$? The null model above has an $R^2 = 0$, while virtually *any* set of regressors will explain some fraction of the variation of y around $\bar{y}$, the sample estimate of μ. R^2 is that fraction in the unit interval: the proportion of the variation in y about $\bar{y}$ explained by X.

Below the ANOVA table and summary statistics, Stata reports the coefficient estimates for each of the b_j values, along with their estimated standard errors, *t*-statistics, and the associated *p*-values labeled `P>|t|`: that is, the tail probability for a two-tailed test on b_j corresponding to the hypothesis $H_0 : b_j = 0$.

In the last two columns, a confidence interval for the coefficient estimate is displayed, with limits defined by the current setting of `level`. The `level()` option on `regress` (or other estimation commands) may be used to specify a particular level. After performing the estimation (e.g., with the default 95% level) the regression results may be redisplayed with, for instance, `regress, level(90)`. The default `level` may be either changed for the session or changed permanently with `set level` *n* `[`, *permanently*`]`.

Regression without a constant term

Stata offers the option of estimating a regression equation without a constant term with the `noconstant` option, although in general it is recommended to avoid this option. Such a model makes little sense if the mean of the response variable is nonzero and all regressors' coefficients are insignificant.

Estimating a constant term in a model that does not have one causes a small loss in the efficiency of the parameter estimates. In contrast, incorrectly omitting a constant term produces inconsistent estimates. The tradeoff should be clear: include a constant term, and let the data indicate whether its estimate can be distinguished from zero.

Recovering estimation results

The `regress` command shares the features of all estimation (e-class) commands. Saved results from `regress` can be viewed by typing `ereturn list`.

Try it out!

All Stata estimation commands save an estimated parameter vector as matrix `e(b)` and the estimated variance-covariance matrix of the parameters as matrix `e(V)`.

One item listed in the `ereturn list` should be noted: `e(sample)`, listed as a `function` rather than a `scalar`, `macro` or `matrix`. The `e(sample)` function returns 1 if an observation was included in the estimation sample and 0 otherwise.

The `regress` command honors any *if* and *in* qualifiers and then practices case-wise deletion to remove any observations with missing values across the set $\{y, X\}$. Thus, the observations actually used in generating the regression estimates may be fewer than those specified in the `regress` command. A subsequent command such as `summarize` *regressors if* (or *in*) will not necessarily provide the descriptive statistics of the observations on X that entered the regression unless all regressors and the y variable are in the *varlist*.

This is particularly relevant when building models with time series data, as the use of lags, leads and differences will cause observations to be omitted from the estimation sample.

Any statistics for the set of observations actually used in estimation can easily be computed with the qualifier `if e(sample)`.

Try it out:

```
summarize regressors if e(sample)
```

will yield the appropriate summary statistics from the regression sample. It may be retained for later use by placing it in a new variable:

```
generate byte reg1sample = e(sample)
```

where we use the `byte` data type to save memory since `e(sample)` is an indicator {0,1} variable.

Hypothesis testing in regression

The application of regression methods is often motivated by the need to conduct tests of hypotheses which are implied by a specific theoretical model. In this section we discuss hypothesis tests and interval estimates assuming that the model is properly specified and that the errors are independently and identically distributed ($i.i.d.$). Estimators are random variables, and their sampling distributions depend on that of the error process.

Any hypothesis involving the coefficients of a regression equation can be expressed as one or more restrictions on the coefficient vector, reducing the dimensionality of the estimation problem. The *Wald test* involves estimating the unrestricted equation and evaluating the degree to which the restricted equation would differ in terms of its explanatory power.

Consider the general form of the Wald test statistic. Given the regression equation

$$y = X\beta + u \tag{11}$$

Any set of linear restrictions on the coefficient vector may be expressed as

$$R\beta = r \tag{12}$$

where R is a $q \times k$ matrix and r is a q-element column vector, with $q < k$. The q restrictions on the coefficient vector β imply that $(k - q)$ parameters are to be estimated in the restricted model. Each row of R imposes one restriction on the coefficient vector; a single restriction may involve multiple coefficients.

For instance, given the regression equation

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + u \tag{13}$$

We might want to test the hypothesis $H_0 : \beta_2 = 0$. This single restriction on the coefficient vector implies $R\beta = r$, where

$$\begin{aligned} R &= (\,0\;1\;0\;0\,) \\ r &= (\,0\,) \end{aligned} \tag{14}$$

A test of $H_0 : \beta_2 = \beta_3$ would imply the single restriction

$$\begin{aligned} R &= (\,0\;1\;-1\;0\,) \\ r &= (\,0\,) \end{aligned} \tag{15}$$

Given a hypothesis expressed as $H_0 : R\beta = r$, we may construct the Wald statistic as

$$W = \frac{1}{s^2}(Rb - r)'[R(X'X)^{-1}R']^{-1}(Rb - r) \tag{16}$$

This quadratic form makes use of the vector of estimated coefficients, b, and evaluates the degree to which the restrictions fail to hold: the magnitude of the elements of the vector $(Rb - r)$. The Wald statistic evaluates the sums of squares of that vector, each weighted by a measure of their precision. Its denominator is s^2, the estimated variance of the error process, replacing the unknown parameter σ_u^2.

Stata contains a number of commands for the construction of hypothesis tests and confidence intervals which may be applied following an estimated regression. Some Stata commands report test statistics in the normal and χ^2 forms when the estimation commands are justified by large-sample theory. More commonly, the finite-sample *t* and *F* distributions are reported.

Stata's tests do not deliver verdicts with respect to the specified hypothesis, but rather present the *p-value* (or *prob-value*) of the test. Intuitively, the *p*-value is the probability of observing the estimated coefficient(s) *if the null hypothesis is true.*

In `regress` output, a number of test statistics and their *p*-values are automatically generated: that of the ANOVA *F* and the *t*-statistics for each coefficient, with the null hypothesis that the coefficients equal zero in the population. If we want to test additional hypotheses after a regression equation, three Stata commands are particularly useful: `test`, `testparm` and `lincom`. The `test` command may be specified as

`test` *coeflist*

where *coeflist* contains the names of one or more variables in the regression model.

A second syntax is

`test` *exp* = *exp*

where *exp* is an algebraic expression in the names of the regressors. The arguments of `test` may be repeated in parentheses in conducting joint tests. Additional syntaxes for `test` are available for multiple-equation models.

The `testparm` command provides similar functionality, but allows wildcards in the coefficient list:

`testparm` *varlist*

where the *varlist* may contain `*` or a hyphenated expression such as `ind1-ind9`.

The `lincom` command evaluates linear combinations of coefficients:

`lincom` *exp*

where *exp* is any linear combination of coefficients that is valid in the second syntax of `test`. For `lincom`, the *exp* must *not* contain an equal sign.

If we want to test the hypothesis $H_0 : \beta_j = 0$, the ratio of the estimated coefficient to its estimated standard error is distributed *t* under the null hypothesis that the population coefficient equals zero. That ratio is displayed by `regress` as the *t* column of the coefficient table. Returning to our investment equation, a test statistic for the significance of a coefficient could be produced by using the commands:

Try it out:

```
. bcuse macro14, nodesc

. reg altsales frprime L2.fsdebt L.(gdprdot ur), vsquish

      Source |       SS           df       MS      Number of obs   =       138
-------------+----------------------------------   F(4, 133)       =     69.00
       Model |  424.816508         4  106.204127   Prob > F        =    0.0000
    Residual |  204.724794       133  1.53928416   R-squared       =    0.6748
-------------+----------------------------------   Adj R-squared   =    0.6650
       Total |  629.541302       137  4.59519198   Root MSE        =    1.2407

------------------------------------------------------------------------------
    altsales |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     frprime |  -.2218317    .031051    -7.14   0.000    -.2832495   -.1604139
      fsdebt |
         L2. |  -.2698066    .150925    -1.79   0.076    -.5683305    .0287173
     gdprdot |
         L1. |   .2047546    .036702     5.58   0.000     .1321594    .2773498
          ur |
         L1. |  -93.76327   7.522177   -12.46   0.000    -108.6418   -78.8847
       _cons |   24.89216   2.121014    11.74   0.000     20.69688    29.08745
------------------------------------------------------------------------------

. test L2.fsdebt

 ( 1)  L2.fsdebt = 0

       F(  1,   133) =    3.20
            Prob > F =    0.0761
```

In Stata's shorthand this is equivalent to the command `test _b[L2.fsdebt] = 0` (and much easier to type). If we use the `test` command, we note that the statistic is displayed as *F(1,N-k)* rather than in the t_{N-k} form of the coefficient table.

As many hypotheses to which `test` may be applied involve more than one restriction on the coefficient vector—and thus more than one degree of freedom—Stata routinely displays an F-statistic.

If we cannot reject the hypothesis $H_0 : \beta_j = 0$, and wish to restrict the equation accordingly, we remove that variable from the list of regressors.

More generally, we may to test the hypothesis $\beta_j = \beta_j^0 = \theta$, where θ is any constant value. If theory or prior research suggests that the coefficient on lagged `gdprdot` should be 0.25, then we may specify that hypothesis in `test`:

```
. bcuse macro14, nodesc
. qui reg altsales frprime L2.fsdebt L.(gdprdot ur), vsquish
. test L.gdprdot = 0.25
 ( 1)  L.gdprdot = .25
       F(  1,   133) =    1.52
            Prob > F =    0.2198
```

The estimated coefficient of 0.205 cannot be distinguished from 0.25.

We might want to compute a point and interval estimate for the sum of several coefficients. We may do that with the `lincom` (linear combination) command, which allows the specification of any linear expression in the coefficients. In the context of our estimated equation, let us consider an arbitrary restriction: that the coefficients on `frprime, L2.fsdebt` and `L.gdprdot` sum to -0.2, so that we may write

$$H_0 : \beta_{frprime} + \beta_{L2.fsdebt} + \beta_{L.gdprdot} = -0.2$$

Note that although this hypothesis involves *three* estimated coefficients, it only involves *one* restriction on the coefficient vector. In this case, we have unitary coefficients on each term, but that need not be so.

Try it out:

```
. bcuse macro14, nodesc
. qui reg altsales frprime L2.fsdebt L.(gdprdot ur), vsquish
. lincom frprime + L2.fsdebt + L.gdprdot
 ( 1)  frprime + L2.fsdebt + L.gdprdot = 0

------------------------------------------------------------------------------
    altsales |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.2868837   .1740504    -1.65   0.102    -.6311486    .0573812
------------------------------------------------------------------------------

. test frprime + L2.fsdebt + L.gdprdot = -0.2
 ( 1)  frprime + L2.fsdebt + L.gdprdot = -.2
       F(  1,   133) =    0.25
            Prob > F =    0.6185
```

The `lincom` command shows that the sum of the three estimated coefficients is -0.287, with an interval estimate including -0.2. The hypothesis cannot be rejected.

We may use `test` to consider equality of two of the coefficients, or to test that their ratio equals a particular value.

Try it out:

```
. bcuse macro14, nodesc
. qui reg altsales frprime L2.fsdebt L.(gdprdot ur), vsquish
. test frprime = -2*L.gdprdot
 ( 1)  frprime + 2*L.gdprdot = 0
       F(  1,   133) =    5.58
            Prob > F =    0.0196
```

The test that the coefficient on `fprime` is of opposite sign and twice as large as the coefficient on `L.gdprdot` may be rejected at the 95% level.

Joint hypothesis tests

All of the tests illustrated above are presented as an F-statistic with one numerator degree of freedom since they only involve one restriction on the coefficient vector. In many cases, we wish to test an hypothesis involving multiple restrictions on the coefficient vector. Although the former test could be expressed as a t-test, the latter cannot. Multiple restrictions on the coefficient vector imply a *joint test*, the result of which is not simply a box score of individual tests.

A joint test is usually constructed in Stata by listing each hypothesis to be tested in parentheses on the `test` command. As presented above, the first syntax of the `test` command, `test` *coeflist*, perfoms the joint test that two or more coefficients are jointly zero, such as $H_0 : \beta_2 = 0$ and $\beta_3 = 0$.

It is important to understand that this joint hypothesis is not at all the same as $H_0' : \beta_2 + \beta_3 = 0$. The latter hypothesis will be satisfied by a locus of $\{\beta_2, \beta_3\}$ values: all pairs that sum to zero. The former hypothesis will only be satisfied at the point where *each coefficient* equals zero. The joint hypothesis may be tested for our investment equation:

Try it out:

```
. bcuse macro14, nodesc
. qui reg altsales frprime L2.fsdebt L.(gdprdot ur), vsquish
. test L.gdprdot L.ur
 ( 1)  L.gdprdot = 0
 ( 2)  L.ur = 0
       F(  2,   133) =  111.23
            Prob > F =    0.0000
```

The data overwhelmingly reject the joint hypothesis that the model excluding the lags of `gdprdot` and `ur` is correctly specified relative to the full model.

Tests of nonlinear hypotheses

What if the hypothesis tests to be conducted cannot be written in the linear form

$$H_0 : R\beta = r \tag{17}$$

for example, if theory predicts a certain value for the product of two coefficients in the model, or for an expression such as $(\beta_2/\beta_3 + \beta_4)$? Two Stata commands are analogues to those we have used above: `testnl` and `nlcom`.

The former allows specification of nonlinear hypotheses on the β values, but unlike `test`, the syntax `_b[`*varname*`]` must be used to refer to each coefficient value. If a joint test is to be conducted, the equations defining each nonlinear restriction must be written in parentheses, as illustrated below.

The `nlcom` command permits us to compute nonlinear combinations of the estimated coefficients in point and interval form, similar to `lincom`. Both commands employ the *delta method*, an approximation to the distribution of a nonlinear combination of random variables appropriate for large samples which constructs Wald-type tests. Unlike tests of linear hypotheses, nonlinear Wald-type tests based on the delta method are sensitive to the scale of the y and X data.

Try it out:

```
. bcuse macro14, nodesc
. qui reg altsales frprime L2.fsdebt L.(gdprdot ur), vsquish
. testnl _b[frprime] * _b[L.gdprdot] = -0.075
  (1)  _b[frprime] * _b[L.gdprdot] = -0.075
             chi2(1) =        8.13
         Prob > chi2 =        0.0044
. nlcom _b[frprime] * _b[L.gdprdot]
      _nl_1:  _b[frprime] * _b[L.gdprdot]
------------------------------------------------------------------------------
    altsales |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _nl_1 |  -.0454211   .0103762    -4.38   0.000     -.065758   -.0250842
------------------------------------------------------------------------------
```

In this example, we consider a restriction on the product of the coefficients of `frprime` and `L.gdprdot`. The product of these coefficients can be distinguished from -0.075 at the 95% level. A point and interval estimate for their product can be computed by `nlcom`.

Computing residuals and predicted values

After estimating a linear regression model with `regress` we may compute the regression residuals or the predicted values.

Computation of the residuals for each observation allows us to assess how well the model has done in explaining the value of the response variable for that observation. Is the in-sample prediction $\hat{y}_i$ much larger or smaller than the actual value y_i?

Computation of predicted values allows us to generate in-sample predictions: the values of the response variable generated by the estimated model.

We may also want to generate out-of-sample predictions: that is, apply the estimated regression function to observations that were not used to generate the estimates. This may involve hypothetical values of the regressors or actual values. In the latter case, we may want to apply the estimated regression function to a separate sample (e.g., to a different time period than that used for estimation) to evaluate its applicability beyond the regression sample.

If a regression model is well specified, it should generate reasonable predictions for any sample from the population. If out-of-sample predictions are poor, the model's specification may be too specific to the original sample.

Neither the residuals nor predicted values are calculated by Stata's `regress` command, but either may be computed immediately thereafter with the `predict` command. This command is given as

```
predict [type] newvar [if] [in] [, choice]
```

where *choice* specifies the quantity to be computed for each observation.

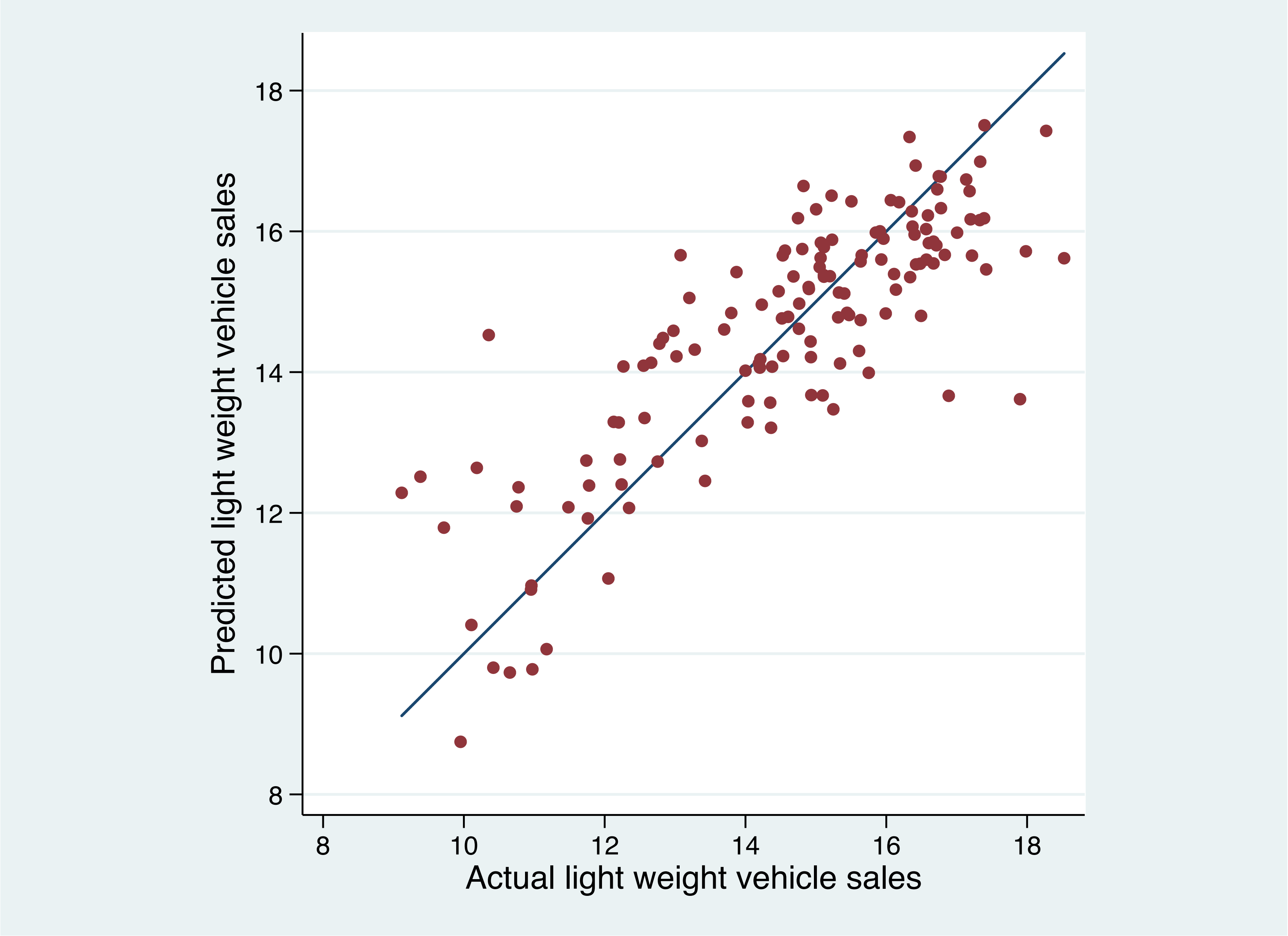
For linear regression, `predict`'s default action is the computation of predicted values. These are known as the *point predictions*, and are specified by the choice *xb*. If the residuals are required, the command

```
predict double altsaleseps, residual
```

should be used.

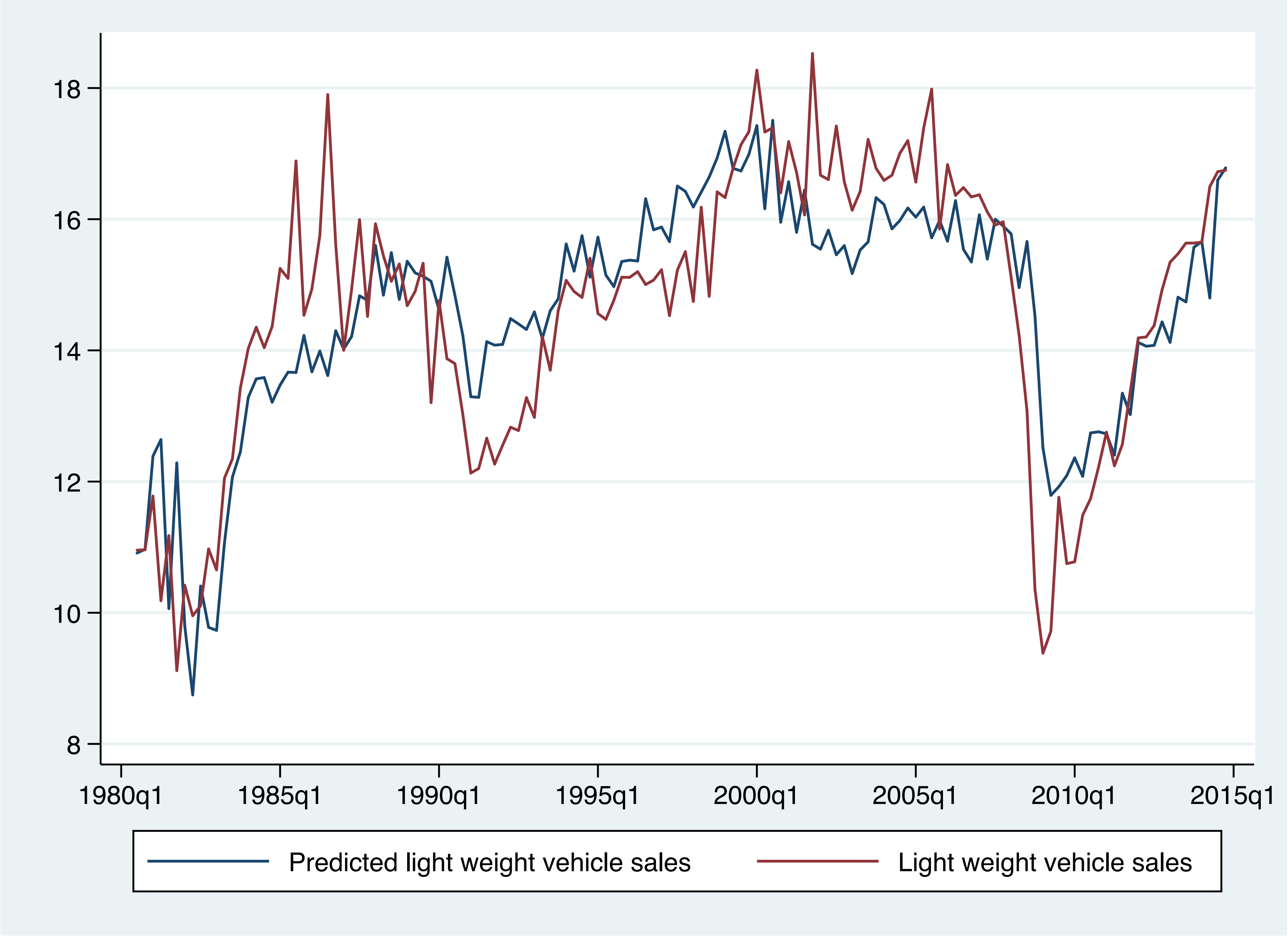
The regression estimates are only available to `predict` until another estimation command (e.g., `regress`) is issued. If these series are needed, they should be computed at the earliest opportunity. The use of `double` as the optional *type* in these commands ensures that the series will be generated with full numerical precision, and is strongly recommended.

We often would like to evaluate the quality of the regression fit in graphical terms. With a single regressor, a plot of actual and predicted values of y_i versus x_i will suffice. In multiple regression, the natural analogue is a plot of actual y_i versus the predicted $\hat{y}_i$ values.


Predicted light weight vehicle sales
8
10
12
14
16
18
8
10
12
14
16
18
Actual light weight vehicle sales

The aspect ratio has been constrained to unity so that points on the 45° line represent perfect predictions. Note that the model systematically overpredicts at relatively low levels of vehicle sales.

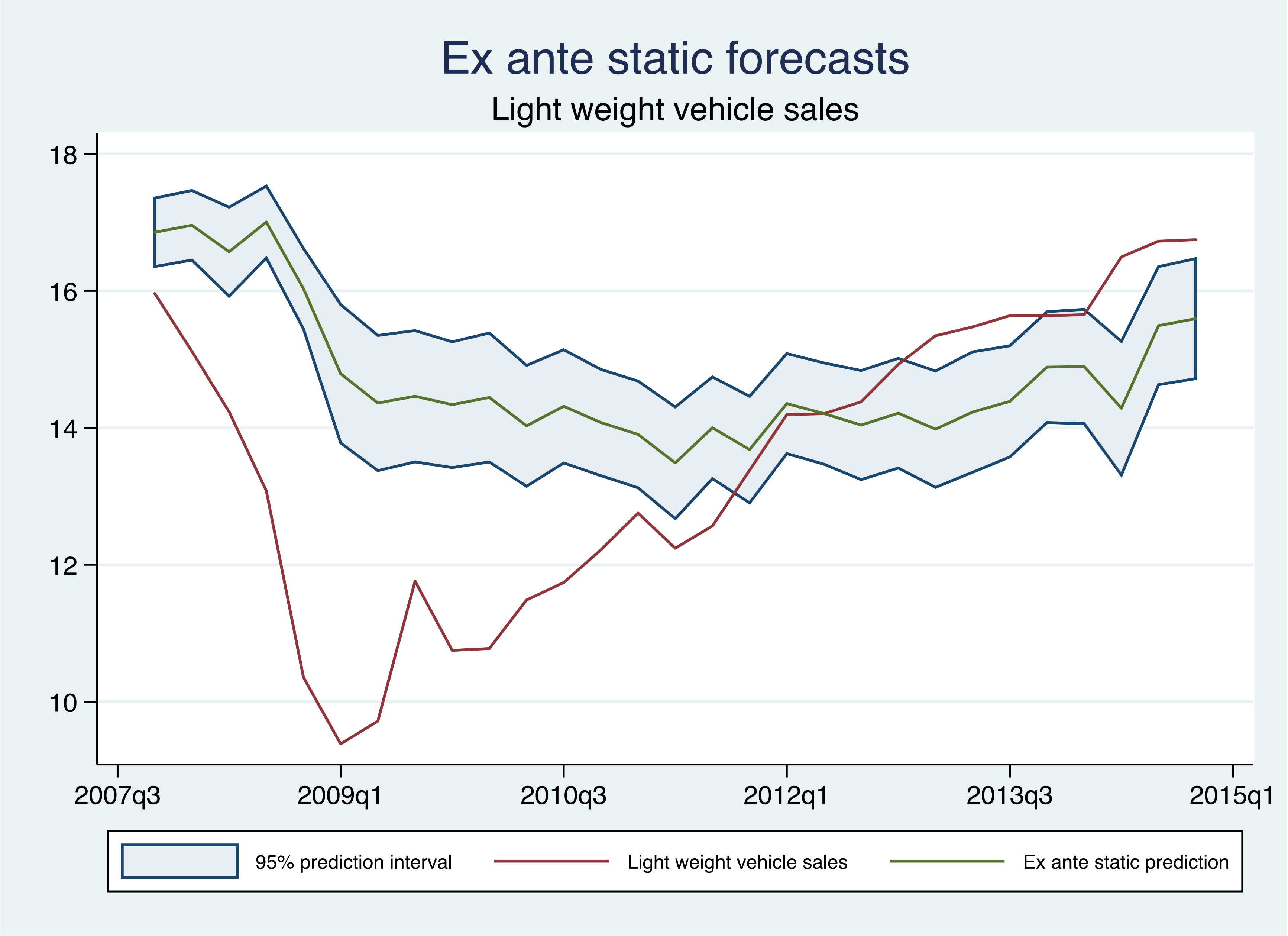
When using time series data, we may also want to examine the model's performance on a time series plot, using the `tsline` command.


18
16
14
12
10
8
1980q1
1985q1
1990q1
1995q1
2000q1
2005q1
2010q1
2015q1
Predicted light weight vehicle sales
Light weight vehicle sales

Like other Stata commands, `predict` will generate predictions for the entire sample. We may want to estimate a model over a subsample, and produce out-of-sample predictions, or *ex ante* forecasts. We may also want to produce interval estimates for forecasts, in- or out-of-sample. The latter may be done, after a regression, by specifying choice `stdp` for the standard error of prediction around the expected value of $y|X_0$. These are static forecasts, rather than dynamic forecasts, as the model does not contain a lagged endogenous variable.

We illustrate by reestimating the vehicle sales model through 2007Q3, the calendar quarter preceding the most recent recession, and producing `ex ante` point and interval forecasts for the remaining periods. We juxtapose these point and interval estimates against the actual series during the recession and aftermath.

```
. bcuse macro14, nodesc clear
. qui reg altsales frprime L2.fsdebt L.(gdprdot ur) if tin(,2007q3), vsquish
. predict double altsaleshat if tin(2007q4,), xb
(215 missing values generated)
. predict double altsalesstdp if tin(2007q4,), stdp
(215 missing values generated)
. scalar tval = invttail(e(df_r),0.025)
. g double uplim = altsaleshat + tval * altsalesstdp
(215 missing values generated)
. g double lowlim = altsaleshat - tval * altsalesstdp
(215 missing values generated)
. lab var uplim "95% prediction interval"
. lab var lowlim "95% prediction interval"
. lab var altsaleshat "Ex ante static prediction"
. twoway (rarea uplim lowlim yq if tin(2007q4,), xti("") xlab(,labs(small)) ///
> legend(cols(3) size(vsmall))  ti("Ex ante static forecasts") ///
> t2("Light weight vehicle sales") /*scheme(s2mono)*/ fintensity(inten10)) ///
> (tsline altsales altsaleshat if tin(2007q4,), ylab(,angle(0) labs(small)) )
```


Ex ante static forecasts
Light weight vehicle sales
18
16
14
12
10
2007q3
2009q1
2010q3
2012q1
2013q3
2015q1
95% prediction interval
Light weight vehicle sales
Ex ante static prediction

Regression with non-i.i.d. errors

If the regression errors are independently and identically distributed (*i*.*i*.*d*.), OLS produces consistent point and interval estimates. Their sampling distribution in large samples is normal with a mean at the true coefficient values and their *VCE* is consistently estimated by the standard formula.

If the zero conditional mean assumption holds but the errors are not *i*.*i*.*d*., OLS produces consistent estimates whose sampling distribution in large samples is still normal with a mean at the true coefficient values, but whose *VCE* cannot be consistently estimated by the standard formula.

We have two options when the errors are not $i.i.d.$ First, we can use the consistent OLS point estimates with a different estimator of the *VCE* that accounts for non-$i.i.d.$ errors. Alternatively, if we can specify how the errors deviate from $i.i.d.$ in our regression model, we can model that process, using a different estimator that produces consistent and more efficient point estimates.

The tradeoff between these two methods is that of *robustness* versus *efficiency*. In a *robust* approach we place fewer restrictions on the estimator: the idea being that the consistent point estimates are good enough, although we must correct our estimator of their *VCE* to account for non-$i.i.d.$ errors.

In the *efficient* approach we incorporate an explicit specification of the non-$i.i.d.$ distribution into the model. If this specification is appropriate, the additional restrictions which it implies will produce a more efficient estimator than that of the robust approach.

Robust standard errors

We will only illustrate the robust approach. If the errors are conditionally heteroskedastic and we want to apply the robust approach, we use the Huber–White–sandwich estimator of the variance of the linear regression estimator, available in most Stata estimation commands as the `robust` option.

If the assumption of homoskedasticity is valid, the non-robust standard errors are more efficient than the robust standard errors. If we are working with a sample of modest size and the assumption of homoskedasticity is tenable, we should rely on non-robust standard errors.

As robust standard errors are very easily calculated in Stata, it is simple to estimate both sets of standard errors for a particular equation and consider whether inference based on the non-robust standard errors is fragile. In large data sets, it has become increasingly common practice to report robust (or Huber–White–sandwich) standard errors.

The alternate approach, generalized least squares (GLS), can be implemented for a model with heteroskedastic errors by specifying the form of the heteroskedasticity using Stata's weights. For this reason, GLS of this sort is often referred to as weighted least squares (WLS). To implement GLS (WLS), you must provide estimates of the error variance for each observation derived from some model of the heteroskedasticity process.

The Newey–West estimator of the *VCE*

In an extension to Huber–White–sandwich robust standard errors, we may employ the *Newey–West* estimator that is appropriate in the presence of arbitrary heteroskedasticity and autocorrelation, thus known as the *HAC* estimator.

Its use requires us to specify an additional parameter: the maximum order of any significant autocorrelation in the disturbance process, or the maximum lag L. One rule of thumb that has been used is to choose $L = \sqrt[4]{N}$. This estimator is available as the Stata command `newey`, which may be used as an alternative to `regress` for estimation of a regression with *HAC* standard errors.

Like the `robust` option, application of the *HAC* estimator does not modify the point estimates; it only affects the *VCE*. Test statistics based on the *HAC VCE* are robust to arbitrary heteroskedasticity and autocorrelation as well.

Similar to the case of pure heteroskedasticity, the GLS alternative to utilizing *HAC* standard errors is to explicitly model the nature of the serial correlation process. A common assumption is that the process is adequately represented by a first-order autoregression ($AR(1)$). A regression model with $AR(1)$ errors can be estimated by the Stata command `prais`, which implements the Prais–Winsten, Cochrane–Orcutt, Hildreth–Lu and maximum likelihood estimators. For higher-order autoregressive processes, the `arima` command may be used.

Testing for heteroskedasticity

After estimating a regression model we may base a test for heteroskedasticity on the regression residuals. If the assumption of homoskedasticity conditional on the regressors holds, it can be expressed as:

$$H_0 : Var\left(u|X_2, X_3, ..., X_k\right) = \sigma_u^2 \tag{18}$$

A test of this null hypothesis can evaluate whether the variance of the error process appears to be independent of the explanatory variables.

We cannot observe the variances of each element of the disturbance process from samples of size one, but we can rely on the squared residual, e_i^2, to be a consistent estimator of σ_i^2. The logic behind any such test is that although the squared residuals will differ in magnitude across the sample, they should not be systematically related to *anything*, and a regression of squared residuals on any candidate Z_i should have no meaningful explanatory power.

One of the most common tests for heteroskedasticity is derived from this line of reasoning: the *Breusch–Pagan* test. The BP test, a Lagrange Multiplier (LM) test, involves regressing the squares of the regression residuals on a set of variables in an auxiliary regression

$$e_i^2 = d_1 + d_2 Z_{i2} + d_3 Z_{i3} + \ldots d_\ell Z_{i\ell} + v_i \tag{19}$$

The Breusch–Pagan (Cook–Weisberg) test may be executed with `estat hettest` after `regress`. If no regressor list (of Zs) is provided, `hettest` employs the fitted values from the previous regression (the $\hat{y}_i$ values). As mentioned above, the variables specified in the set of Zs could be chosen as measures which did not appear in the original regressor list.

We consider the potential scale-related heteroskedasticity in a cross-sectional model of median housing prices from the `hprice2a` dataset. The scale factor can be thought of as the average size of houses in each community, roughly measured by its number of rooms.

After estimating the model, we calculate three test statistics: that computed by `estat hettest` without arguments, which is the Breusch–Pagan test based on fitted values; `estat hettest` with a variable list, which uses those variables in the auxiliary regression; and White's general test statistic from `whitetst`, available from SSC.

Try it out:

```
. bcuse hprice2a, clear

. quietly regress lprice rooms crime ldist

. estat hettest

Breusch-Pagan / Cook-Weisberg test for heteroskedasticity
         Ho: Constant variance
         Variables: fitted values of lprice

         chi2(1)      =   140.84
         Prob > chi2  =   0.0000

. estat hettest rooms crime ldist

Breusch-Pagan / Cook-Weisberg test for heteroskedasticity
         Ho: Constant variance
         Variables: rooms crime ldist

         chi2(3)      =   252.60
         Prob > chi2  =   0.0000

. whitetst

White´s general test statistic :  144.0052  Chi-sq( 9)  P-value =  1.5e-26
```

Each of these tests indicates that there is a significant degree of heteroskedasticity related to scale in this model.

We illustrate the estimation of the model with OLS and robust standard errors.

```
. estimates table nonRobust Robust, b(%9.4f) se(%5.3f) t(%5.2f) ///
>  title(Estimates of log housing price with OLS and Robust standard errors)
Estimates of log housing price with OLS and Robust standard errors

──────────────────────────────────────
    Variable │ nonRobust      Robust
─────────────┼────────────────────────
       rooms │    0.3072      0.3072
             │     0.018       0.026
             │     17.24       11.80
       crime │   -0.0174     -0.0174
             │     0.002       0.003
             │    -10.97       -6.42
       ldist │    0.0749      0.0749
             │     0.026       0.030
             │      2.93        2.52
       _cons │    7.9844      7.9844
             │     0.113       0.174
             │     70.78       45.76
──────────────────────────────────────
                          legend: b/se/t
```

Note that the OLS standard errors are considerably smaller, biased downward, relative to the robust estimates.

Testing for serial correlation

How might we test for the presence of serially correlated errors? Just as in the case of pure heteroskedasticity, we base tests of serial correlation on the regression residuals. In the simplest case, autocorrelated errors follow the so-called $AR(1)$ model: an *autoregressive process* of order one, also known as a first-order Markov process:

$$u_t = \rho u_{t-1} + v_t, \ |\rho| < 1 \tag{20}$$

where the v_t are uncorrelated random variables with mean zero and constant variance.

If we suspect that there might be autocorrelation in the disturbance process of our regression model, we could use the estimated residuals to diagnose it. The empirical counterpart to u_t in Equation (20) will be the e_t series produced by `predict`. We estimate the auxiliary regression of e_t on e_{t-1} without a constant term, as the residuals have mean zero.

The resulting slope estimate is a consistent estimator of the first-order autocorrelation coefficient ρ of the u process from Equation (20). Under the null hypothesis, $\rho = 0$, so that a rejection of this null hypothesis by this Lagrange Multiplier (LM) test indicates that the disturbance process exhibits $AR(1)$ behavior.

A generalization of this procedure which supports testing for higher-order autoregressive disturbances is the Lagrange Multiplier (*LM*) test of Breusch and Godfrey. In this test, the regression residuals are regressed on the original X matrix augmented with p lagged residual series. The null hypothesis is that the errors are serially independent up to order p.

An improved version of this test was devised by Cumby and Huizinga (*Econometrica*, 1992). This test, programmed by Baum and Schaffer as `actest` and available from the SSC Archive, can be applied in a wider range of circumstances than the Breusch–Godfrey test, including models with conditional heteroskedasticity, endogenous regressors, or overlapping data.

We illustrate the diagnosis of autocorrelation using a time series dataset `ukrates` of monthly short-term and long-term interest rates on UK government securities (Treasury bills and gilts), 1952m3–1995m12.

The model expresses the monthly change in the short rate `rs`, the Bank of England's monetary policy instrument as a function of the prior month's change in the long-term rate `r20`. The regressor and regressand are created on the fly by Stata's time series operators `D.` and `L.` The model represents a monetary policy reaction function.

Try it out:

```
. regress D.rs LD.r20

      Source |       SS       df       MS              Number of obs =     524
-------------+------------------------------           F(  1,    522) =   52.88
       Model |  13.8769739     1  13.8769739           Prob > F      =  0.0000
    Residual |  136.988471   522  .262430021           R-squared     =  0.0920
-------------+------------------------------           Adj R-squared =  0.0902
       Total |  150.865445   523  .288461654           Root MSE      =  .51228

------------------------------------------------------------------------------
        D.rs |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         r20 |
         LD. |   .4882883   .0671484     7.27   0.000      .356374    .6202027
       _cons |   .0040183    .022384     0.18   0.858    -.0399555    .0479921
------------------------------------------------------------------------------

. predict double eps, residual
(2 missing values generated)
```

The Cumby–Huizinga test performed here considers the null of serial independence up to sixth order in the disturbance process, and that null is soundly rejected. The right-hand panel indicates that autocorrelation only appears meaningful at lag 1. Conditional on AR(1), we cannot reject the hypothesis that the AR(2) coefficient is zero.

```
. actest, lags(6) robust

Cumby-Huizinga test for autocorrelation
  H0: variable is MA process up to order q
  HA: serial correlation present at specified lags >q
-----------------------------------------------------------------------------
  H0: q=0 (serially uncorrelated)          |  H0: q=specified lag-1
  HA: s.c. present at range specified      |  HA: s.c. present at lag specified
-----------------------------------------------------------------------------
    lags   |      chi2       df     p-val  | lag |      chi2       df     p-val
-----------+-------------------------------+-----+-----------------------------
   1 -  1  |      9.539       1    0.0020  |  1  |     9.539        1    0.0020
   1 -  2  |     10.440       2    0.0054  |  2  |     1.210        1    0.2714
   1 -  3  |     10.445       3    0.0151  |  3  |     0.093        1    0.7609
   1 -  4  |     10.626       4    0.0311  |  4  |     0.195        1    0.6588
   1 -  5  |     11.392       5    0.0441  |  5  |     0.946        1    0.3307
   1 -  6  |     11.747       6    0.0679  |  6  |     0.310        1    0.5779
-----------------------------------------------------------------------------
  Test allows predetermined regressors/instruments
  Test robust to heteroskedasticity
```

Given this finding, we can generate heteroskedasticity- and autocorrelation-consistent (*HAC*) standard errors using the `newey` command, conservatively specifying 6 lags.

Try it out:

```
. newey D.rs LD.r20, lag(6)

Regression with Newey-West standard errors          Number of obs  =        524
maximum lag: 6                                      F(  1,     522) =      35.74
                                                    Prob > F       =     0.0000

------------------------------------------------------------------------------
             |             Newey-West
        D.rs |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         r20 |
         LD. |   .4882883   .0816725     5.98   0.000     .3278412    .6487354
             |
       _cons |   .0040183   .0256542     0.16   0.876    -.0463799    .0544166
------------------------------------------------------------------------------

. estimates store NeweyWest
. estimates table nonHAC NeweyWest, b(%9.4f) se(%5.3f) t(%5.2f) ///
>   title(Estimates of D.rs with OLS and Newey-West standard errors)
Estimates of D.rs with OLS and Newey-West standard errors

--------------------------------------
    Variable |   nonHAC      NeweyWest
-------------+------------------------
         r20 |
         LD. |    0.4883       0.4883
             |     0.067        0.082
             |      7.27         5.98
             |
       _cons |    0.0040       0.0040
             |     0.022        0.026
             |      0.18         0.16
--------------------------------------
                          legend: b/se/t
```

Note that the Newey–West standard errors are considerably larger than the OLS standard errors. OLS standard errors are biased downward in the presence of positive autocorrelation ($\rho > 0$).

Regression with factor variables

Data come in three flavors: quantitative (or cardinal), ordinal (or ordered) and qualitative. Regression analysis handles quantitative data where both regressor and regressand may take on any real value. We also may work with *ordinal* or ordered data. They are distinguished from cardinal measurements in that an ordinal measure can only express inequality of two items, and not the magnitude of their difference.

We frequently encounter data that are purely *qualitative*, lacking any obvious ordering. If these data are coded as string variables, such as `M` and `F` for survey respondents' genders, we are not likely to mistake them for quantitative values. But in other cases, where a quality may be coded numerically, there is the potential to misuse this qualitative factor as quantitative.

Qualitative factors expressed as non-negative integers may be used as *factor variables* in Stata using the `i.`*varname* notation.

In order to test the hypothesis that a qualitative factor has an effect on a response variable, we must convert the qualitative factor into a set of *indicator variables*, or dummy variables. We then conduct a *joint test* on their coefficients.

If the hypothesis to be tested includes a single qualitative factor, the estimation problem may be described as a one-way analysis of variance, or *one-way ANOVA*. ANOVA models may be expressed as linear regressions on an appropriate set of indicator variables, which can be handled automatically in Stata using the factor-variable notation.

This notion of the equivalence of one-way ANOVA and linear regression on a set of indicator variables that correspond to a single qualitative factor generalizes to multiple qualitative factors.

If there are two qualitative factors (e.g., race and sex) that are hypothesized to affect income, a researcher would regress income on two appropriate sets of indicator variables, each representing one of the qualitative factors. This is then an example of *two-way ANOVA*.

Factor variables

A valuable new feature introduced in Stata version 11 is the *factor variable*. Stata has only one kind of numeric variable (although it supports several different data types, which define the amount of storage needed and possible range of values). However, if a variable is actually *categorical*, taking on only non-negative integer values, it may be used as a factor variable with the `i.` prefix.

The use of factor variables not only avoids explicit generation of indicator (dummy) variables for each level of the categorical variable, but it means that the needed indicator variables are generated ‘on the fly’, as needed. Thus, to include the variable `group`, a categorical variable in `margex.dta` which takes on values 1–3, we need only refer to `i.group` in a statistical or estimation command.

Try it out:

```
webuse margex, clear
list i.group
summarize i.group
regress y i.group
```

For the `list` command, the variables will be named `1b.group`, `2.group`, `3.group`. The `b.` is the base level indicator, by default assigned to the smallest value. You can specify other base levels, such as the largest value, the most frequent value, or a particular value.

For the `summarize` command, only groups 2 and 3 will be shown; the base level is excluded from the list. In the regression on `i.group`, the base level is the variable excluded from the regressor list to prevent perfect collinearity. The conditional mean of the excluded variable appears in the constant term.

This in itself merely mimics a preexisting feature of Stata: the `xi:` prefix. But factor variables are much more powerful, in that they can be used to define interactions: both with other factor variables and with continuous variables. Traditionally, you would define interactions by creating new variables representing the product of two indicators, or the product of an indicator with a continuous variable.

Factor variables may be interacted with continuous variables to produce analysis of covariance models. In doing so, you must specify the continuous variables with the new `c.` operator, and interactions with the # operator.

Try it out:

```
webuse margex, clear
regress y i.group c.age i.group#c.age
```

which essentially estimates three regression lines, one for each group.

The factorial operator (##), which includes all main effects and interaction effects, can be used to estimate the same model in a simpler syntax.

Try it out:

```
webuse margex
regress y i.group##c.age
```

There is a great advantage in using factor variables rather than creating explicit interaction variables. If you define interactions with the factor variable syntax, Stata can then interpret the expression in postestimation commands such as `margins`, as we will illustrate in the next section. This also applies to the definition of interactions of continuous variables, and powers of continuous variables, such as `c.age#c.age`.

Interactions are not limited to pairs of variables; up to eight factor variables may be included.

Computing predictive means and marginal effects

With the introduction of factor variables in Stata 11, a powerful new command was added: `margins`, which supersedes earlier versions' `mfx` and `adjust` commands. Those commands remain available, but the new command has many advantages. Like those commands, `margins` is used after an estimation command.

In the simplest case, `margins` applied after a simple one-way ANOVA estimated with `regress y i.group`, with `margins i.group`, merely displays the conditional means, or 'predictive means', for each category of `group`.

Try it out:

```
. webuse margex
. regress y i.group
      Source |       SS       df       MS              Number of obs =    3000
-------------+------------------------------           F(  2,  2997) =   15.48
       Model |  14229.0349     2  7114.51743           Prob > F      =  0.0000
    Residual |  1377203.97  2997  459.527518           R-squared     =  0.0102
-------------+------------------------------           Adj R-squared =  0.0096
       Total |  1391433.01  2999  463.965657           Root MSE      =  21.437

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       group |
          2  |   .4084991   .8912269     0.46   0.647    -1.338979    2.155977
          3  |   5.373138   1.027651     5.23   0.000     3.358165     7.38811
             |
       _cons |   68.35805   .6190791   110.42   0.000     67.14419    69.57191
------------------------------------------------------------------------------
```

Try it out:

```
. margins i.group

Adjusted predictions                              Number of obs    =       3000
Model VCE    : OLS

Expression   : Linear prediction, predict()

─────────────┬────────────────────────────────────────────────────────────────
             │            Delta-method
             │     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
─────────────┼────────────────────────────────────────────────────────────────
       group │
          1  │   68.35805   .6190791   110.42   0.000     67.14468    69.57142
          2  │   68.76655   .6411134   107.26   0.000     67.50999    70.02311
          3  │   73.73119   .8202484    89.89   0.000     72.12353    75.33884
─────────────┴────────────────────────────────────────────────────────────────
```

We now estimate a model including both age and its square.

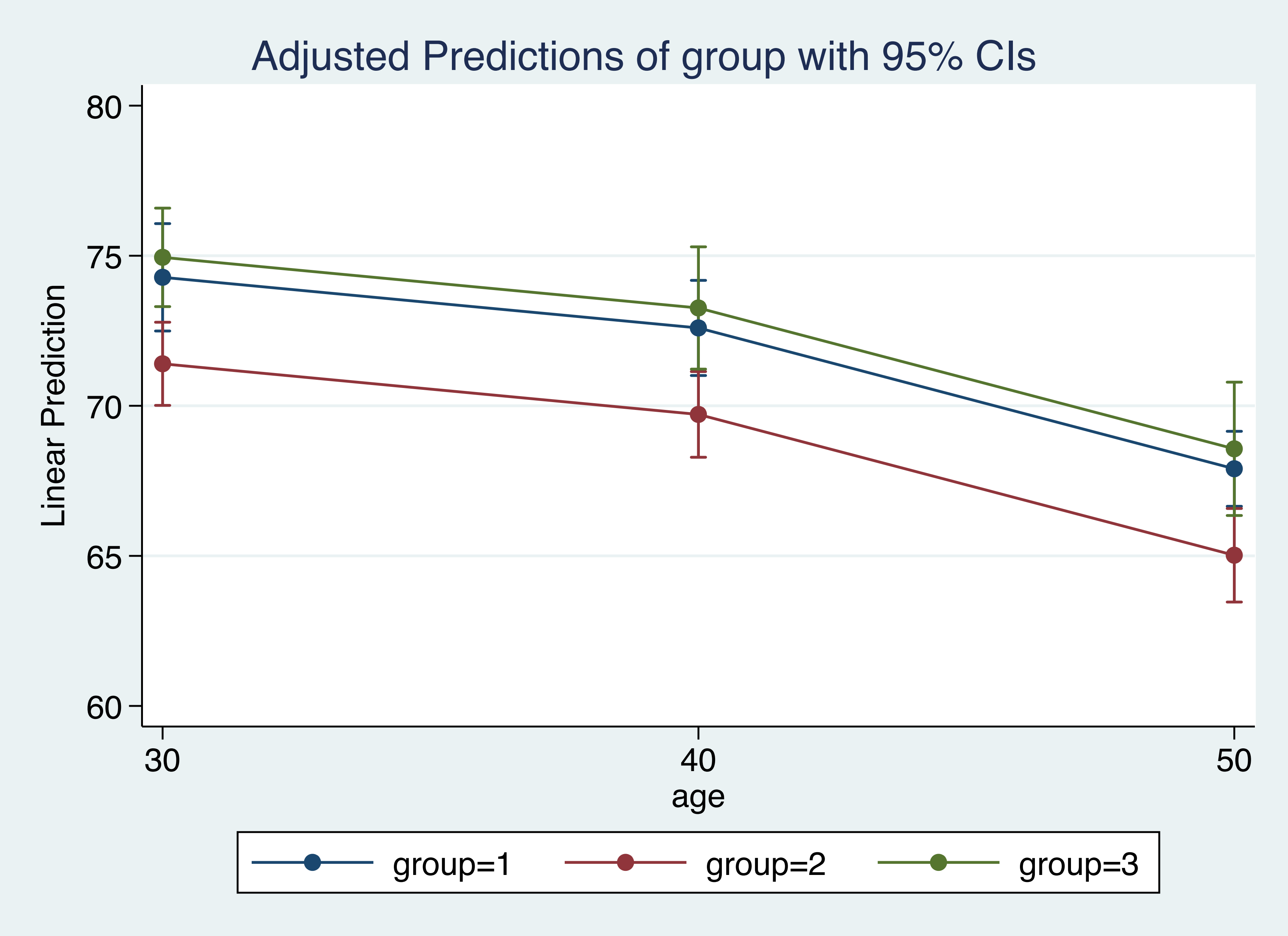
Try it out:

```
. regress y i.group c.age c.age#c.age
      Source |       SS       df       MS              Number of obs =    3000
-------------+------------------------------           F(  4,  2995) =   29.16
       Model |  52164.9734     4  13041.2434           Prob > F      =  0.0000
    Residual |  1339268.03  2995  447.167957           R-squared     =  0.0375
-------------+------------------------------           Adj R-squared =  0.0362
       Total |  1391433.01  2999  463.965657           Root MSE      =  21.146

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       group |
          2  |  -2.881836   .9525533    -3.03   0.003    -4.749561   -1.014111
          3  |     .66475   1.199124     0.55   0.579    -1.686441    3.015941
             |
         age |   .8833198    .265064     3.33   0.001     .3635938    1.403046
             |
 c.age#c.age |  -.0150291   .0032785    -4.58   0.000    -.0214574   -.0086008
             |
       _cons |   61.30783    5.16417    11.87   0.000     51.18215    71.43351
------------------------------------------------------------------------------
```

`margins` can then properly evaluate the regression function for each `group` at selected levels of `age`:

```
. margins i.group, at(age=(30 40 50 ))

Adjusted predictions                              Number of obs   =       3000
Model VCE    : OLS

Expression   : Linear prediction, predict()

1._at        : age             =          30

2._at        : age             =          40

3._at        : age             =          50

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _at#group |
        1 1  |   74.28123   .9128901    81.37   0.000       72.492    76.07046
        1 2  |    71.3994   .7068297   101.01   0.000     70.01404    72.78476
        1 3  |   74.94598   .8371438    89.53   0.000     73.30521    76.58675
        2 1  |   72.59406   .8088305    89.75   0.000     71.00878    74.17934
        2 2  |   69.71222   .7284925    95.69   0.000      68.2844    71.14004
        2 3  |   73.25881   1.039152    70.50   0.000     71.22211    75.29551
        3 1  |   67.90107   .6374738   106.52   0.000     66.65164    69.15049
        3 2  |   65.01923   .7951989    81.76   0.000     63.46067    66.57779
        3 3  |   68.56582   1.133534    60.49   0.000     66.34413     70.7875
------------------------------------------------------------------------------

. marginsplot, legend(rows(1)) ylab(,angle(0)) saving(margins4a, replace)

  Variables that uniquely identify margins: age group
(file margins4a.gph saved)
```

This is best illustrated using the `marginsplot` command:



In earlier versions of Stata, calculation of marginal effects in this model required some programming due to the nonlinear term in `age`. Using `margins, dydx`, that is now simple. Furthermore, and most importantly, the default behavior of `margins` is to calculate average marginal effects (AMEs) rather than marginal effects at the average (MAEs) or at some other point in the space of the regressors.

Current econometric practice favors the use of AMEs: the computation of each observation's marginal effect with respect to an explanatory factor, averaged over the estimation sample, to the computation of MAEs, which reflect an average individual: e.g., an individual who is 14% Black, 75% married, with 2.3 children.

We illustrate by computing average marginal effects (AMEs) for the prior regression.

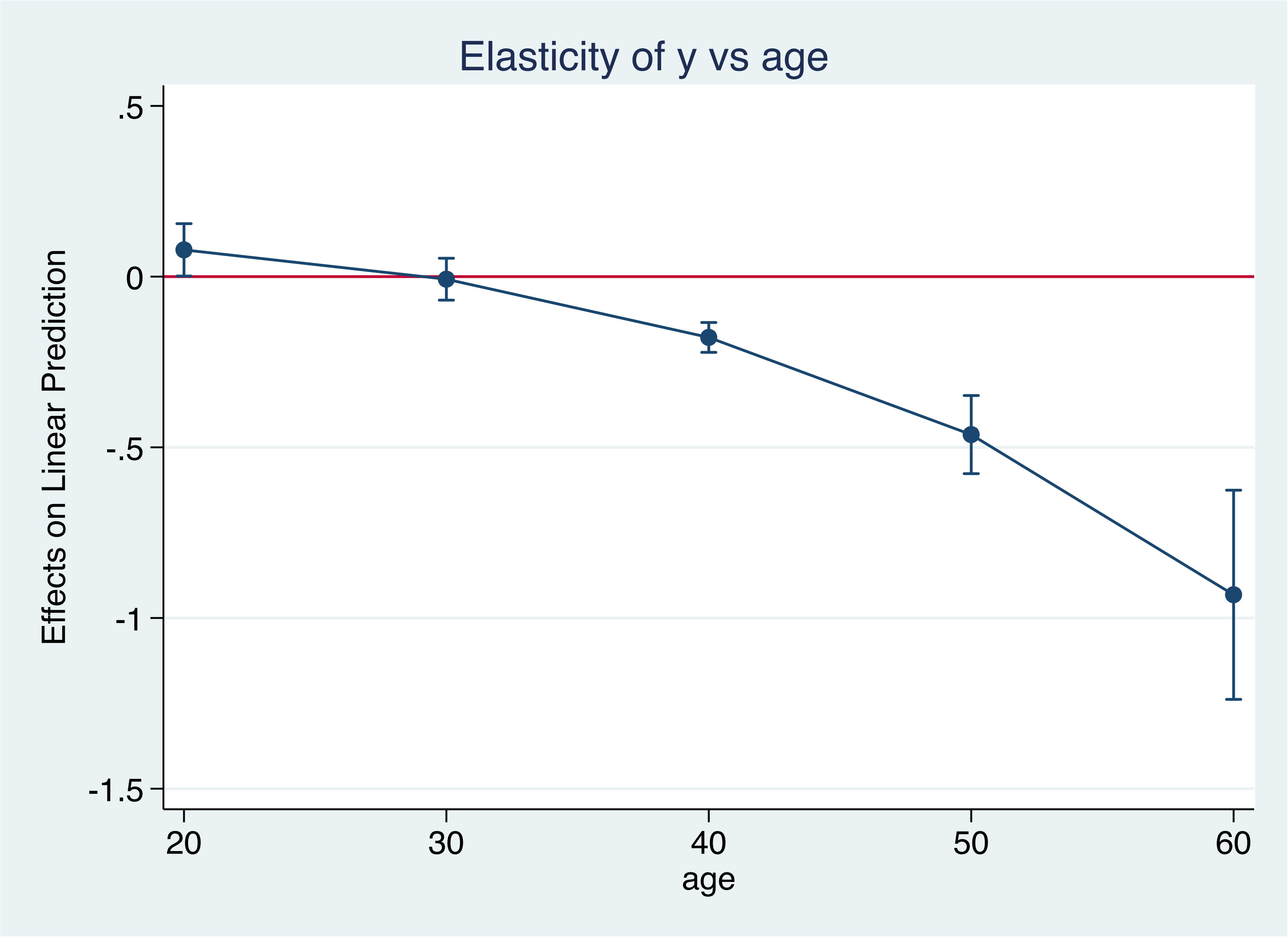
Try it out:

```
. margins, dydx(_all)

Average marginal effects                          Number of obs   =       3000
Model VCE    : OLS

Expression   : Linear prediction, predict()
dy/dx w.r.t. : 2.group 3.group age

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       group |
          2  |  -2.881836   .9525533    -3.03   0.002    -4.748806   -1.014866
          3  |     .66475   1.199124     0.55   0.579    -1.685491    3.014991
             |
         age |  -.3129665   .0398215    -7.86   0.000    -.3910151   -.2349179
------------------------------------------------------------------------------
Note: dy/dx for factor levels is the discrete change from the base level.
```

Alternatively, we may compute elasticities or semi-elasticities:

```
. margins, eyex(age) at(age=(20(10)60))
Average marginal effects                          Number of obs   =       3000
Model VCE    : OLS
Expression   : Linear prediction, predict()
ey/ex w.r.t. : age
1._at        : age             =          20
2._at        : age             =          30
3._at        : age             =          40
4._at        : age             =          50
5._at        : age             =          60
------------------------------------------------------------------------------
             |            Delta-method
             |      ey/ex   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
age          |
         _at |
          1  |   .0783689   .0391718     2.00   0.045     .0015935    .1551442
          2  |  -.0075387   .0312951    -0.24   0.810    -.0688759    .0537985
          3  |  -.1781213   .0222896    -7.99   0.000    -.2218081   -.1344345
          4  |  -.4627739   .0584047    -7.92   0.000     -.577245   -.3483027
          5  |  -.9319831   .1563228    -5.96   0.000     -1.23837    -.625596
------------------------------------------------------------------------------
. marginsplot, ylab(,angle(0)) ti("Elasticity of y vs age") yline(0) ///
>             saving(margins6a, replace)
  Variables that uniquely identify margins: age
(file margins6a.gph saved)
```

A graph from `marginsplot` may be most convincing:



Consider a model where we specify a factorial interaction between categorical and continuous covariates:

```
regress y i.treatment i.group##c.age
```

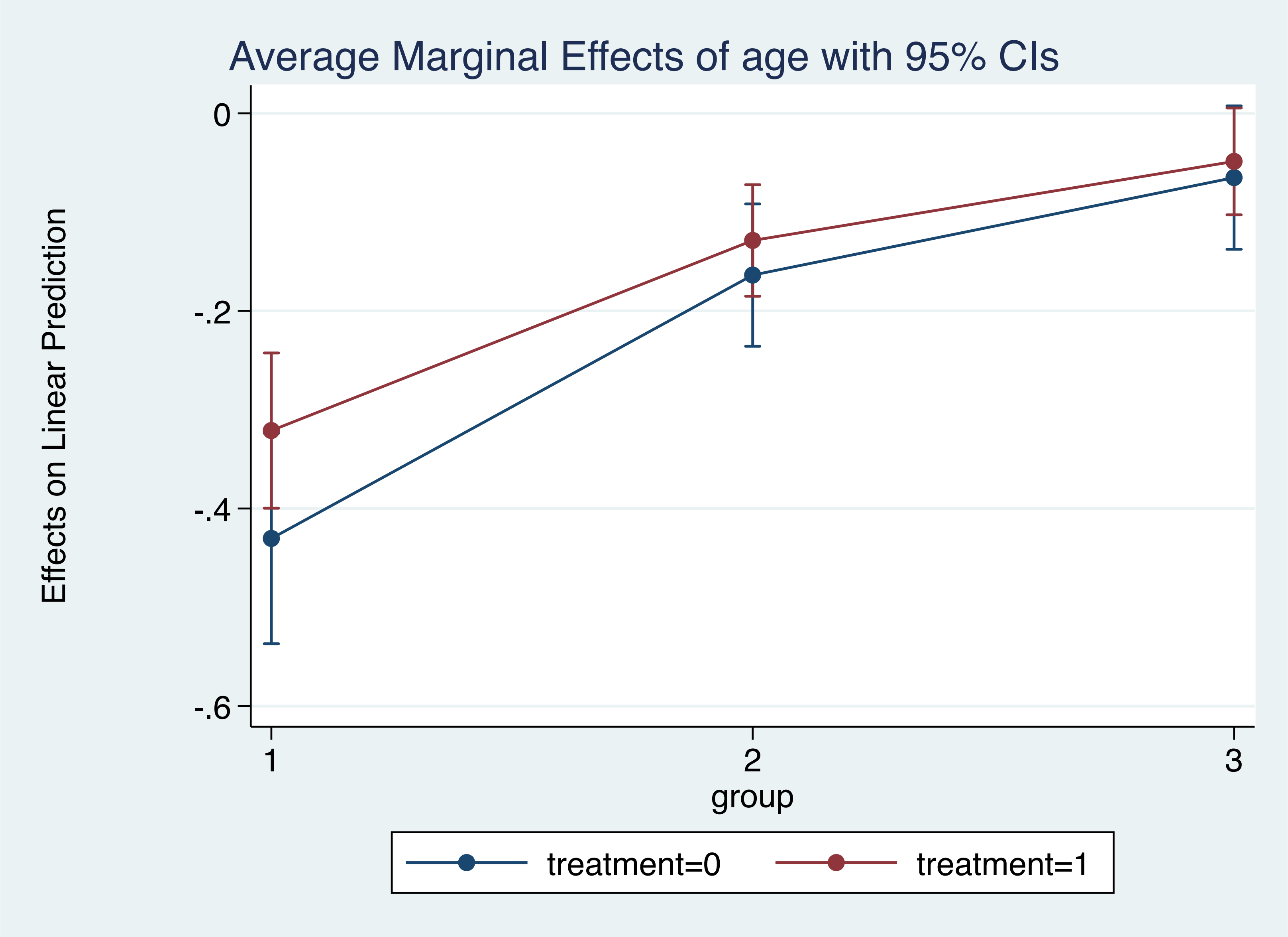
In this specification, each level of `group` has its own intercept and slope, whereas the binary variable `treatment` only shifts the intercept term.

We can compute elasticities or semi-elasticities with the `over` option of `margins` for all combinations of `group` and `treatment`:

Try it out:

```
. margins, eyex(age) over(group treatment)
Average marginal effects                          Number of obs   =       3000
Model VCE    : OLS

Expression   : Linear prediction, predict()
ey/ex w.r.t. : age
over         : group treatment

------------------------------------------------------------------------------
             |            Delta-method
             |      ey/ex   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
age          |
      group#|
   treatment |
        1 0  |  -.4303457   .0543237    -7.92   0.000    -.5368182   -.3238733
        1 1  |  -.3211145     .04008    -8.01   0.000    -.3996699   -.2425592
        2 0  |  -.1637441   .0367602    -4.45   0.000    -.2357927   -.0916955
        2 1  |  -.1287019   .0288017    -4.47   0.000    -.1851521   -.0722517
        3 0  |  -.0650796    .037019    -1.76   0.079    -.1376355    .0074764
        3 1  |  -.0487138   .0275775    -1.77   0.077    -.1027647    .0053371
------------------------------------------------------------------------------

. marginsplot, ylab(,angle(0)) saving(margins7a, replace)

  Variables that uniquely identify margins: group treatment
(note: file margins7a.gph not found)
(file margins7a.gph saved)
```

`marginsplot` can be used to illustrate these bivariate effects:

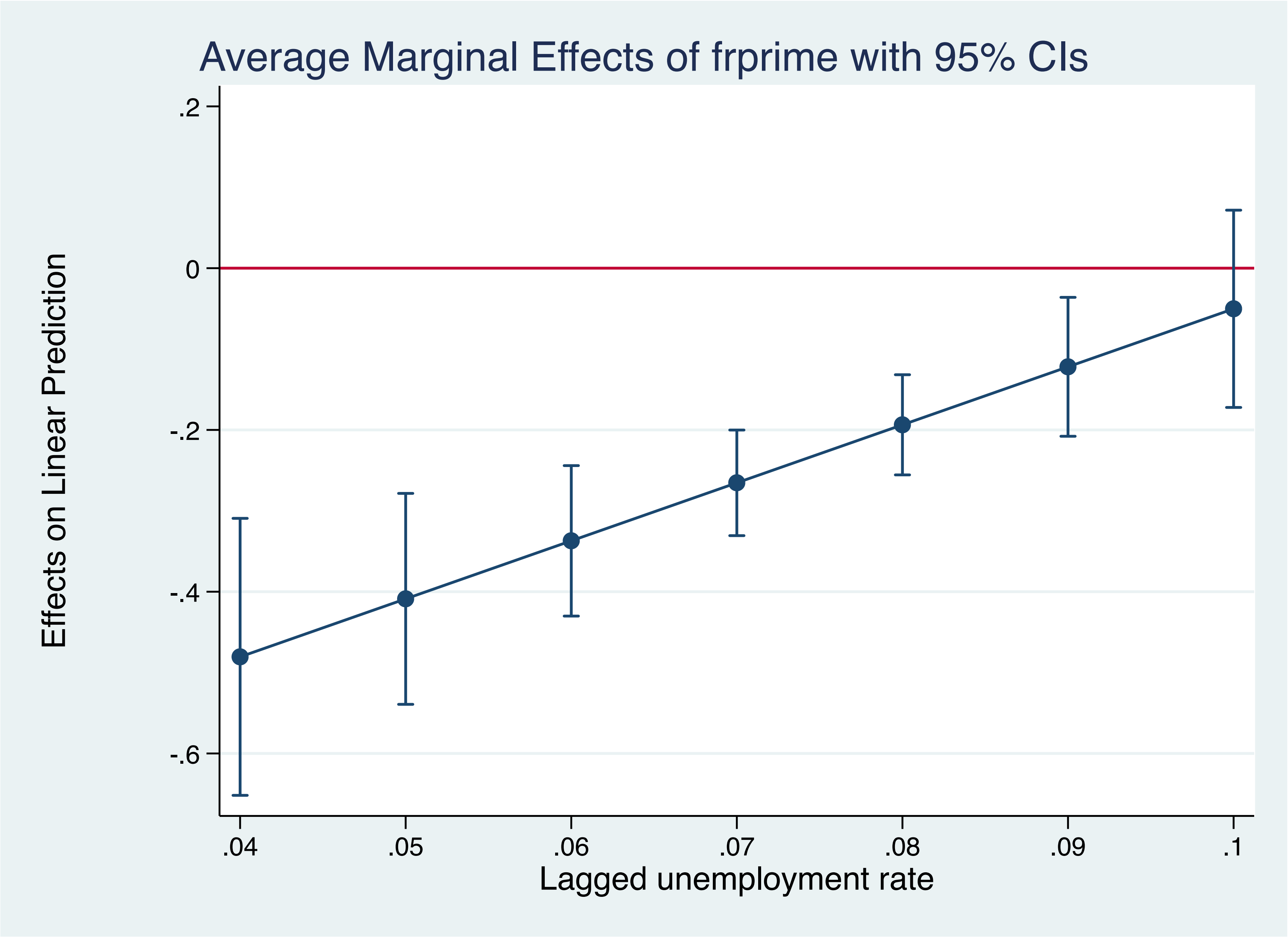


As an illustration using our vehicle sales model, consider adding an interaction term between the prime rate and the lagged unemployment rate to that model. **Try it out:**

```
. bcuse macro14, nodesc clear

. reg altsales frprime L2.fsdebt L.(gdprdot ur) c.frprime#c.L.ur

      Source |       SS           df       MS      Number of obs   =       138
-------------+----------------------------------   F(5, 132)       =     61.03
       Model |  439.441664         5  87.8883328   Prob > F        =    0.0000
    Residual |  190.099638       132  1.44014877   R-squared       =    0.6980
-------------+----------------------------------   Adj R-squared   =    0.6866
       Total |  629.541302       137  4.59519198   Root MSE        =    1.2001

--------------------------------------------------------------------------------
       altsales |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
----------------+---------------------------------------------------------------
        frprime |  -.7674482    .173829    -4.41   0.000    -1.111299   -.4235972
                |
         fsdebt |
            L2. |  -.2589781   .1460236    -1.77   0.078    -.5478273    .0298711
                |
        gdprdot |
            L1. |   .2162537   .0356834     6.06   0.000     .1456684     .286839
                |
             ur |
            L1. |  -148.9161   18.77418    -7.93   0.000    -186.0532   -111.7789
                |
c.frprime#cL.ur |   7.172063   2.250596     3.19   0.002     2.720162    11.62396
                |
          _cons |   28.94671   2.414075    11.99   0.000     24.17143    33.72199
--------------------------------------------------------------------------------
```

We can now use `margins` to calculate the average marginal effects of those regressors, and trace out the effects of changes in the prime rate at different levels of unemployment using `marginsplot`:

```
. margins, dydx(frprime L.ur)

Average marginal effects                        Number of obs     =        138
Model VCE    : OLS

Expression   : Linear prediction, predict()
dy/dx w.r.t. : frprime L.ur

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     frprime |  -.3036099   .0395046    -7.69   0.000    -.3817539    -.225466
             |
          ur |
         L1. |  -93.59035    7.27612   -12.86   0.000    -107.9832   -79.19746
------------------------------------------------------------------------------

. qui margins, dydx(frprime) at(L.ur=(0.04(0.01)0.1))
```


Average Marginal Effects of frprime with 95% CIs
Effects on Linear Prediction
.2
0
-.2
-.4
-.6
.04
.05
.06
.07
.08
.09
.1
Lagged unemployment rate

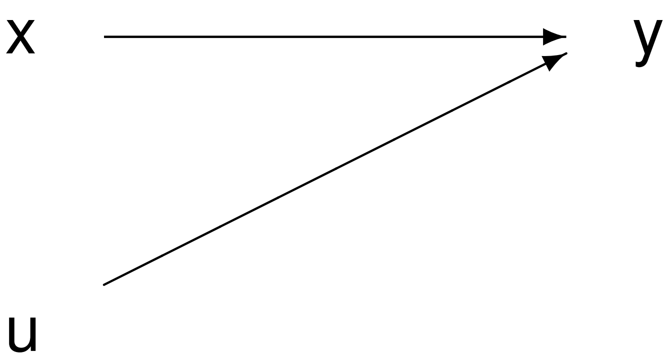
The `margins` and `marginsplot` commands have many other capabilities, including the option to place the computed quantities into the ‘e-returns’, which may then be accessed as estimation results. The lengthy reference manual article on `margins` is a useful reference.

Regression with instrumental variables

What are instrumental variables (IV) methods? Most widely known as a solution to *endogenous regressors*: explanatory variables correlated with the regression error term, IV methods provide a way to nonetheless obtain consistent parameter estimates.

First let us consider a path diagram illustrating the problem addressed by IV methods. We can use ordinary least squares (OLS) regression to consistently estimate a model of the following sort.

Standard regression: $y = xb + u$
no association between x and u; OLS consistent



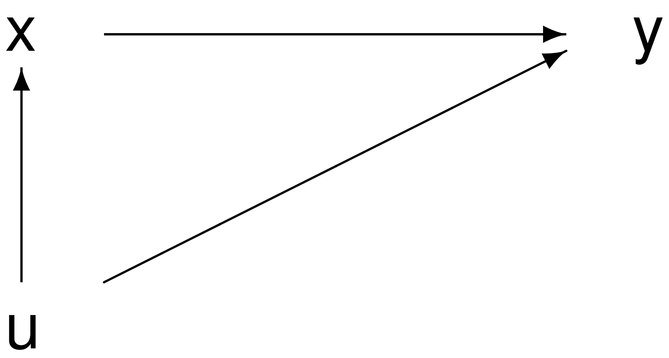
x

y

u

However, OLS regression breaks down in the following circumstance:

Endogeneity: $y = xb + u$
correlation between x and u; OLS inconsistent



The correlation between x and u (or the failure of the zero conditional mean assumption $E[u|x] = 0$) can be caused by any of several factors.

We have stated the problem as that of *endogeneity*: the notion that two or more variables are jointly determined in the behavioral model. This arises naturally in the context of a *simultaneous equations model* such as a supply-demand system in economics, in which price and quantity are jointly determined in the market for that good or service.

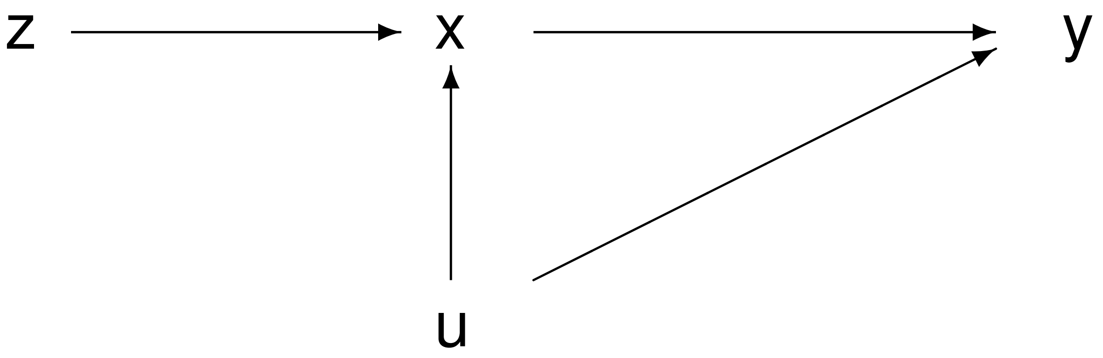
A shock or disturbance to either supply or demand will affect both the equilibrium price and quantity in the market, so that by construction both variables are correlated with any shock to the system. OLS methods will yield inconsistent estimates of any regression including both price and quantity, however specified.

In a macroeconomic context, many of the behavioral equations that we might specify for consumption, investment, money demand, and so on are likely to contain endogenous regressors. In a consumption function, a shock to consumption or saving will also affect the level of GDP, and thus disposable income.

In this context, the zero conditional mean assumption cannot hold, even in terms of weak exogeneity of the regressors. OLS is no longer an appropriate estimation method, and we must rely upon other estimators to produce consistent estimates.

The solution provided by IV methods may be viewed as:

Instrumental variables regression: $y = xb + u$
z uncorrelated with u, correlated with x



The additional variable z is termed an *instrument* for x. In general, we may have many variables in x, and more than one x correlated with u. In that case, we shall need at least that many variables in z.

To deal with the problem of *endogeneity* in a supply-demand system, a candidate *z* will affect (e.g.) the quantity supplied of the good, but not directly impact the demand for the good. An example for an agricultural commodity might be temperature or rainfall: clearly exogenous to the market, but likely to be important in the production process.

For the model of macro consumption, we might use autonomous government expenditure or the level of exports as an instrument. Those components of GDP are clearly related to the level of GDP and disposable income, but they are not directly affected by consumption shocks.

Why should we not always use IV?

First, It may be difficult to find variables that can serve as valid instruments. Many variables that have an effect on included endogenous variables also have a direct effect on the dependent variable. Chris Sims' critique of macro modelers employing 'incredible identifying restrictions' should be taken seriously, as identification requires that certain variables not appear in the equation to be estimated.

Second, IV estimators are innately *biased*, and their finite-sample properties are often problematic. Thus, most of the justification for the use of IV is asymptotic. Performance in small samples may be poor.

Third, the precision of IV estimates is lower than that of OLS estimates (least squares is just that). In the presence of *weak instruments* (excluded instruments only weakly correlated with included endogenous regressors) the loss of precision will be severe, and IV estimates may be no improvement over OLS. This suggests we need a test to determine whether a particular regressor must be treated as endogenous in order to produce consistent estimates.

The IV–GMM estimator

To discuss the implementation of IV estimators and test statistics, we consider a more general framework: an instrumental variables estimator implemented using the Generalized Method of Moments (GMM). As we will see, conventional IV estimators such as two-stage least squares (2SLS) are special cases of this IV-GMM estimator.

The model:

$$y = X\beta + u, \quad u \sim (0, \Omega)$$

with X $(N \times k)$ and define a matrix Z $(N \times \ell)$ where $\ell \geq k$. This is the Generalized Method of Moments IV (IV-GMM) estimator.

The ℓ instruments give rise to a set of ℓ moments:

$$g_i(\beta) = Z_i' u_i = Z_i'(y_i - x_i\beta), \ i = 1, N$$

where each g_i is an ℓ-vector. The method of moments approach considers each of the ℓ moment equations as a sample moment, which we may estimate by averaging over N:

$$\bar{g}(\beta) = \frac{1}{N} \sum_{i=1}^{N} z_i(y_i - x_i\beta) = \frac{1}{N} Z' u$$

The GMM approach chooses an estimate that solves $\bar{g}(\hat{\beta}_{GMM}) = 0$.

If $\ell = k$, the equation to be estimated is said to be *exactly identified* by the *order condition* for identification: that is, there are as many excluded instruments as included right-hand endogenous variables. The method of moments problem is then k equations in k unknowns, and a unique solution exists, equivalent to the standard IV estimator:

$$\hat{\beta}_{IV} = (Z'X)^{-1}Z'y$$

In the case of *overidentification* ($\ell > k$) we may define a set of k instruments:

$$\hat{X} = Z(Z'Z)^{-1}Z'X = P_Z X$$

which gives rise to the *two-stage least squares* (2SLS) estimator

$$\hat{\beta}_{2SLS} = (\hat{X}'X)^{-1}\hat{X}'y = (X'P_Z X)^{-1}X'P_Z y$$

which despite its name is computed by this single matrix equation.

In the 2SLS method with overidentification, the ℓ available instruments are “boiled down" to the k needed by defining the P_Z matrix. In the IV-GMM approach, that reduction is not necessary. All ℓ instruments are used in the estimator. Furthermore, a *weighting matrix* is employed so that we may choose $\hat{\beta}_{GMM}$ so that the elements of $\bar{g}(\hat{\beta}_{GMM})$ are as close to zero as possible. With $\ell > k$, not all ℓ moment conditions can be exactly satisfied, so a criterion function that weights them appropriately is used to improve the efficiency of the estimator.

The GMM estimator minimizes the criterion

$$J(\hat{\beta}_{GMM}) = N\,\bar{g}(\hat{\beta}_{GMM})'W\,\bar{g}(\hat{\beta}_{GMM})$$

where W is a $\ell \times \ell$ symmetric weighting matrix.

Solving the set of FOCs, we derive the IV-GMM estimator of an overidentified equation:

$$\hat{\beta}_{GMM} = (X'ZWZ'X)^{-1}X'ZWZ'y$$

which will be identical for all W matrices which differ by a factor of proportionality. The *optimal* weighting matrix, as shown by Hansen (1982), chooses $W = S^{-1}$ where S is the covariance matrix of the moment conditions to produce the most *efficient* estimator:

$$S = E[Z'uu'Z] = lim_{N\rightarrow\infty}\ N^{-1}[Z'\Omega Z]$$

With a consistent estimator of S derived from 2SLS residuals, we define the feasible IV-GMM estimator as

$$\hat{\beta}_{FEGMM} = (X'Z\ \hat{S}^{-1}Z'X)^{-1}X'Z\ \hat{S}^{-1}Z'y$$

where *FEGMM* refers to the *feasible efficient* GMM estimator.

IV-GMM and the distribution of u

The derivation makes no mention of the form of Ω, the variance-covariance matrix (*vce*) of the error process u. If the errors satisfy all classical assumptions are $i.i.d.$, $S = \sigma_u^2 I_N$ and the optimal weighting matrix is proportional to the identity matrix. The IV-GMM estimator is merely the standard IV (or 2SLS) estimator.

IV-GMM robust estimates

If there is heteroskedasticity of unknown form, we usually compute *robust* standard errors in any Stata estimation command to derive a consistent estimate of the *vce*. In this context,

$$\hat{S} = \frac{1}{N} \sum_{i=1}^{N} \hat{u}_i^2 Z_i' Z_i$$

where $\hat{u}$ is the vector of residuals from any consistent estimator of β (e.g., the 2SLS residuals). For an overidentified equation, the IV-GMM estimates computed from this estimate of S will be more efficient than 2SLS estimates.

IV-GMM cluster-robust estimates

If errors are considered to exhibit arbitrary intra-cluster correlation in a dataset with *M* clusters, we may derive a *cluster-robust* IV-GMM estimator using

$$\hat{S} = \sum_{j=1}^{M} \hat{u}_j' \hat{u}_j$$

where

$$\hat{u}_j = (y_j - x_j \hat{\beta}) X' Z (Z'Z)^{-1} z_j$$

The IV-GMM estimates employing this estimate of *S* will be both robust to arbitrary heteroskedasticity and intra-cluster correlation, equivalent to estimates generated by Stata's `cluster(`*varname*`)` option. For an overidentified equation, IV-GMM cluster-robust estimates will be more efficient than 2SLS estimates.

IV-GMM HAC estimates

The IV-GMM approach may also be used to generate *HAC standard errors*: those robust to arbitrary heteroskedasticity and autocorrelation. Although the best-known *HAC* approach in econometrics is that of Newey and West, using the Bartlett kernel (per Stata's `newey`), that is only one choice of a *HAC* estimator that may be applied to an IV-GMM problem.

Baum–Schaffer–Stillman's `ivreg2` (from the SSC Archive) and official Stata's `ivregress` provide several choices for kernels. For some kernels, the kernel *bandwidth* (roughly, number of lags employed) may be chosen automatically in either command.

Example of IV and IV-GMM estimation

We illustrate various forms of the IV estimator with an equation from Ray Fair's *Fairmodel* (`https://fairmodel.econ.yale.edu`, version of January 2015) fit to quarterly US data. The equation models the log of per capita real consumption of services (`lcsz`) with a partial adjustment mechanism, taking one explanatory variable, `rsa`, the after-tax Treasury bill rate, as endogenous. For more information on the specification, please see the description of Fair's US model on the website.

We first fit the relationship with the standard 2SLS estimator, using Baum–Schaffer–Stillman's `ivreg2` command. You could fit the same equation with `ivregress 2sls`. We first fit the equation through 2007q3, prior to the financial crisis. Some of the standard `ivreg2` output, relating to weak instruments, has been edited on the following slides.

```
. bcuse macro14, clear nodesc
. di "`eq1´"
ivreg2 lcsz l.lcsz cnst2 ag1 ag2 ag3 l.laaz lydz (rsa = t l.lydz l.rsa l.(lodhm limz lpimz lynlz pcpd rb l
> yz lvz ur) ljhgsz lcogsz ltrgsz l2.rs) if tin(1959q1,2007q3)
. `eq1´
IV (2SLS) estimation
--------------------

Estimates efficient for homoskedasticity only
Statistics consistent for homoskedasticity only
                                                      Number of obs =      195
                                                      F(  8,   186) =  1.7e+05
                                                      Prob > F      =   0.0000
Total (centered) SS     =  18.41810203                Centered R2   =   0.9999
Total (uncentered) SS   =  445.4618192                Uncentered R2 =   1.0000
Residual SS             =  .0024514585                Root MSE      =  .003546

------------------------------------------------------------------------------
        lcsz |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         rsa |   -.001022   .0002186    -4.67   0.000    -.0014505   -.0005935
        lcsz |
         L1. |   .8120217    .024324    33.38   0.000     .7643475    .8596958
       cnst2 |    .018676   .0042038     4.44   0.000     .0104368    .0269152
         ag1 |  -.2729463   .0695902    -3.92   0.000    -.4093405   -.1365521
         ag2 |  -.2728811   .0607191    -4.49   0.000    -.3918883   -.1538739
         ag3 |   .6320943    .132604     4.77   0.000     .3721952    .8919935
        laaz |
         L1. |   .0355757    .008028     4.43   0.000     .0198411    .0513103
        lydz |   .1330948   .0259633     5.13   0.000     .0822077     .183982
       _cons |  -.0424856   .0441367    -0.96   0.336     -.128992    .0440208
------------------------------------------------------------------------------
Underidentification test (Anderson canon. corr. LM statistic):         164.199
                                                   Chi-sq(16) P-val =   0.0000
  (output omitted)
------------------------------------------------------------------------------
Sargan statistic (overidentification test of all instruments):          49.296
                                                   Chi-sq(15) P-val =   0.0000
------------------------------------------------------------------------------
Instrumented:         rsa
Included instruments: L.lcsz cnst2 ag1 ag2 ag3 L.laaz lydz
Excluded instruments: t L.lydz L.rsa L.lodhm L.limz L.lpimz L.lynlz L.pcpd L.rb
                      L.lyz L.lvz L.ur ljhgsz lcogsz ltrgsz L2.rs
------------------------------------------------------------------------------
```

We may fit this equation with different assumptions about the error process. The estimates above assume *i.i.d.* errors. We may also compute robust standard errors in the 2SLS context.

We then apply IV-GMM with robust standard errors. As the equation is overidentified, the IV-GMM estimates will differ, and will be more efficient than the robust 2SLS estimates.

Last, we may estimate the equation with IV-GMM and HAC standard errors, using the default Bartlett kernel (as employed by Newey–West) and an automatic bandwidth.

Some coefficients' estimates have been omitted from the table for clarity.

```
. est table TSLS_IID TSLS_Robust IVGMM_Robust IVGMM_HAC, b(%12.4f) t(%5.2f) ///
> stat(rmse j jp) drop(cnst2 ag1 ag2 ag3) vsquish

-------------------------------------------------------------------------
    Variable |   TSLS_IID      TSLS_Robust     IVGMM_Robust     IVGMM_HAC
-------------+-----------------------------------------------------------
         rsa |     -0.0010         -0.0010         -0.0009         -0.0009
             |       -4.67           -4.19           -4.15           -4.78
        lcsz |
         L1. |      0.8120          0.8120          0.8059          0.8055
             |       33.38           30.01           31.97           40.82
        laaz |
         L1. |      0.0356          0.0356          0.0335          0.0312
             |        4.43            4.81            5.13            4.48
        lydz |      0.1331          0.1331          0.1467          0.1467
             |        5.13            4.85            5.69            7.14
       _cons |     -0.0425         -0.0425         -0.0758         -0.0354
             |       -0.96           -0.92           -1.84           -0.74
-------------+-----------------------------------------------------------
        rmse |      0.0035          0.0035          0.0036          0.0035
           j |     49.2958         29.0679         29.0679         16.1906
          jp |      0.0000          0.0158          0.0158          0.3695
-------------------------------------------------------------------------
                                                           legend: b/t
```

Note that the coefficients' point estimates change when IV-GMM is employed, and that their *t*-statistics are larger than those of robust IV. The point estimates are also altered when IV-GMM with HAC VCE is computed. As expected, 2SLS yields the smallest RMS error.

Tests of overidentifying restrictions

If and only if an equation is *overidentified*, with more excluded instruments than included endogenous variables, we may test whether the excluded instruments are appropriately independent of the error process. That test should always be performed when it is possible to do so, as it allows us to evaluate the validity of the instruments.

A test of *overidentifying restrictions* regresses the residuals from an IV or 2SLS regression on all instruments in Z. Under the null hypothesis that all instruments are uncorrelated with u, the test has a large-sample $\chi^2(r)$ distribution where r is the number of overidentifying restrictions.

Under the assumption of *i.i.d.* errors, this is known as a *Sargan test*, and is routinely produced by `ivreg2` for IV and 2SLS estimates. After `ivregress`, the command `estat overid` provides the test.

If we have used IV-GMM estimation in `ivreg2`, the test of overidentifying restrictions becomes the Hansen *J* statistic: the GMM criterion function. Although *J* will be identically zero for any exactly-identified equation, it will be positive for an overidentified equation. If it is "too large", doubt is cast on the satisfaction of the moment conditions underlying GMM.

The test in this context is known as the *Hansen test* or *J test*, and is calculated by `ivreg2` when the `gmm2s` option is employed.

The Sargan–Hansen test of overidentifying restrictions should be performed routinely in any overidentified model estimated with instrumental variables techniques. Instrumental variables techniques are powerful, but if a strong rejection of the null hypothesis of the Sargan–Hansen test is encountered, you should strongly doubt the validity of the estimates.

In the prior estimates table, note that the test of overidentifying restrictions rejects its null hypothesis when the equation is estimated with assumptions of $i.i.d.$ errors or heteroskedastic errors. It does not reject when the HAC estimator of the VCE is used.

It is important to understand that the Sargan–Hansen test of overidentifying restrictions is a joint test of the hypotheses that the instruments, excluded and included, are independently distributed of the error process *and* that they are properly excluded from the model.

Testing a subset of overidentifying restrictions

We may be quite confident of some instruments' independence from *u* but concerned about others. In that case a *GMM distance* or *C* test may be used. The `orthog( )` option of `ivreg2` tests whether a *subset* of the model's overidentifying restrictions appear to be satisfied.

This is carried out by calculating two Sargan–Hansen statistics: one for the full model and a second for the model in which the listed variables are (a) considered endogenous, if included regressors, or (b) dropped, if excluded regressors. In case (a), the model must still satisfy the order condition for identification. The difference of the two Sargan–Hansen statistics, often termed the *GMM distance* or Hayashi *C statistic*, will be distributed χ^2 under the null hypothesis that the specified orthogonality conditions are satisfied, with d.f. equal to the number of those conditions.

We perform the *C* test on the estimated equation by challenging the exogeneity of lagged `lydz`, the log of real per capita disposable income. Is it properly considered exogenous? The `orthog()` option reestimates the equation, treating it as endogenous, and evaluates the difference in the *J* statistics from the two models. Considering `L.lydz` as exogenous is essentially imposing one more orthogonality condition on the GMM estimation problem.

```
. `eq1´, robust bw(auto) gmm2s orthog(l.lydz)
2-Step GMM estimation
---------------------

Estimates efficient for arbitrary heteroskedasticity and autocorrelation
Statistics robust to heteroskedasticity and autocorrelation
  kernel=Bartlett; bandwidth=6
  Automatic bw selection according to Newey-West (1994)
  time variable (t):  yq
                                                      Number of obs =      195
                                                      F(  8,   186) =  3.9e+05
                                                      Prob > F      =   0.0000
Total (centered) SS     =  18.41810203                Centered R2   =   0.9999
Total (uncentered) SS   =  445.4618192                Uncentered R2 =   1.0000
Residual SS             =  .0024516717                Root MSE      =  .003546

------------------------------------------------------------------------------
             |               Robust
        lcsz |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         rsa |  -.0009076     .00019    -4.78   0.000      -.00128   -.0005351
        lcsz |
         L1. |   .8054897   .0197349    40.82   0.000       .76681    .8441694
       cnst2 |   .0171832    .003302     5.20   0.000     .0107114     .023655
         ag1 |  -.2713178   .0599937    -4.52   0.000    -.3889032   -.1537323
         ag2 |  -.2579323   .0565058    -4.56   0.000    -.3686817   -.1471828
         ag3 |   .6341747   .1043087     6.08   0.000     .4297334    .8386159
        laaz |
         L1. |   .0311793   .0069567     4.48   0.000     .0175445    .0448142
        lydz |   .1466683   .0205387     7.14   0.000     .1064132    .1869234
       _cons |  -.0354028   .0480474    -0.74   0.461    -.1295739    .0587684
  (output omitted)
------------------------------------------------------------------------------
Hansen J statistic (overidentification test of all instruments):        16.191
                                                   Chi-sq(15) P-val =   0.3695
-orthog- option:
Hansen J statistic (eqn. excluding suspect orthog. conditions):         15.703
                                                   Chi-sq(14) P-val =   0.3318
C statistic (exogeneity/orthogonality of suspect instruments):           0.488
                                                   Chi-sq(1) P-val =    0.4850
Instruments tested:   L.lydz
------------------------------------------------------------------------------
Instrumented:         rsa
Included instruments: L.lcsz cnst2 ag1 ag2 ag3 L.laaz lydz
Excluded instruments: t L.lydz L.rsa L.lodhm L.limz L.lpimz L.lynlz L.pcpd L.rb
                      L.lyz L.lvz L.ur ljhgsz lcogsz ltrgsz L2.rs
------------------------------------------------------------------------------
```

It appears that `L.lydz` may be considered exogenous in this specification.

A variant on this strategy is implemented by the `endog( )` option of `ivreg2`, in which one or more variables considered endogenous can be tested for exogeneity. The *C* test in this case will consider whether the null hypothesis of their exogeneity is supported by the data.

If all endogenous regressors are included in the `endog( )` option, the test is essentially a test of whether IV methods are required to estimate the equation. If OLS estimates of the equation are consistent, they should be preferred. In this context, the test is equivalent to a *(Durbin–Wu–)Hausman test* comparing IV and OLS estimates, as implemented by Stata's `hausman` command with the `sigmaless` option. Using `ivreg2`, you need not estimate and store both models to generate the test's verdict.

```
. `eq1´, robust bw(auto) gmm2s vsquish endog(rsa)
2-Step GMM estimation
---------------------

Estimates efficient for arbitrary heteroskedasticity and autocorrelation
Statistics robust to heteroskedasticity and autocorrelation
  kernel=Bartlett; bandwidth=6
  Automatic bw selection according to Newey-West (1994)
  time variable (t):  yq
                                                      Number of obs =       195
                                                      F(  8,    186) =  3.9e+05
                                                      Prob > F      =   0.0000
Total (centered) SS     =  18.41810203                Centered R2   =   0.9999
Total (uncentered) SS   =  445.4618192                Uncentered R2 =   1.0000
Residual SS             =  .0024516717                Root MSE      =  .003546

------------------------------------------------------------------------------
             |               Robust
        lcsz |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         rsa |  -.0009076     .00019    -4.78   0.000      -.00128   -.0005351
        lcsz |
         L1. |   .8054897   .0197349    40.82   0.000       .76681    .8441694
       cnst2 |   .0171832    .003302     5.20   0.000     .0107114     .023655
         ag1 |  -.2713178   .0599937    -4.52   0.000    -.3889032   -.1537323
         ag2 |  -.2579323   .0565058    -4.56   0.000    -.3686817   -.1471828
         ag3 |   .6341747   .1043087     6.08   0.000     .4297334    .8386159
        laaz |
         L1. |   .0311793   .0069567     4.48   0.000     .0175445    .0448142
        lydz |   .1466683   .0205387     7.14   0.000     .1064132    .1869234
       _cons |  -.0354028   .0480474    -0.74   0.461    -.1295739    .0587684
  (output omitted)
------------------------------------------------------------------------------
Hansen J statistic (overidentification test of all instruments):        16.191
                                                   Chi-sq(15) P-val =   0.3695
-endog- option:
Endogeneity test of endogenous regressors:                               0.852
                                                   Chi-sq(1) P-val =    0.3561
Regressors tested:     rsa
------------------------------------------------------------------------------
Instrumented:          rsa
Included instruments:  L.lcsz cnst2 ag1 ag2 ag3 L.laaz lydz
Excluded instruments:  t L.lydz L.rsa L.lodhm L.limz L.lpimz L.lynlz L.pcpd L.rb
                       L.lyz L.lvz L.ur ljhgsz lcogsz ltrgsz L2.rs
------------------------------------------------------------------------------
```

For instance, with the model above, we might question whether IV techniques are needed. In this context, it appears that we could safely estimate this equation with OLS techniques, as the P-value for the *C* test of endogenous regressors of 0.356 does not reject the null hypothesis.

There are a number of other diagnostic tools that may be employed in instrumental variables estimation. Although time constraints prevents their thorough discussion, full details can be found in the Baum–Schaffer–Stillman *Stata Journal* articles.

The weak instruments problem

Instrumental variables methods rely on two assumptions: the excluded instruments are distributed independently of the error process, and they are sufficiently correlated with the included endogenous regressors.

Tests of overidentifying restrictions address the *first* assumption, although we should note that a rejection of their null may be indicative that the exclusion restrictions for these instruments may be inappropriate. That is, some of the instruments have been improperly excluded from the regression model's specification.

The specification of an instrumental variables model asserts that the excluded instruments affect the dependent variable only *indirectly*, through their correlations with the included endogenous variables. If an excluded instrument exerts both direct and indirect influences on the dependent variable, the exclusion restriction should be rejected. This can be readily tested by including the variable as a regressor.

To test the *second* assumption—that the excluded instruments are sufficiently correlated with the included endogenous regressors—we should consider the goodness-of-fit of the "first stage" regressions relating each endogenous regressor to the entire set of instruments.

It is important to understand that the theory of single-equation ("limited information") IV estimation requires that all columns of X are conceptually regressed on all columns of Z in the calculation of the estimates. We cannot meaningfully speak of "this variable is an instrument for that regressor" or somehow restrict which instruments enter which first-stage regressions. Stata's `ivregress` or `ivreg2` will not let you do that because such restrictions only make sense in the context of estimating an entire system of equations by full-information methods (for instance, with `reg3`).

The `first` and `ffirst` options of `ivreg2` (or the `first` option of `ivregress`) present several useful diagnostics that assess the first-stage regressions. If there is a single endogenous regressor, these issues are simplified, as the instruments either explain a reasonable fraction of that regressor's variability or not. With multiple endogenous regressors, diagnostics are more complicated, as each instrument is being called upon to play a role in each first-stage regression.

With sufficiently weak instruments, the asymptotic identification status of the equation is called into question. An equation identified by the order and rank conditions in a finite sample may still be *effectively unidentified* or *numerically unidentified*.

As Staiger and Stock (*Econometrica*, 1997) show, the weak instruments problem can arise even when the first-stage t- and F-tests are significant at conventional levels in a large sample. In the worst case, the bias of the IV estimator is the same as that of OLS, IV becomes inconsistent, and instrumenting only aggravates the problem.

Beyond the informal “rule-of-thumb” diagnostics such as $F > 10$, `ivreg2` computes several statistics that can be used to critically evaluate the strength of instruments. We can write the first-stage regressions as

$$X = Z\Pi + v$$

With X_1 as the endogenous regressors, Z_1 the excluded instruments and Z_2 as the included instruments, this can be partitioned as

$$X_1 = [Z_1 Z_2]\ [\Pi'_{11} \Pi'_{12}]' + v_1$$

The rank condition for identification states that the $L \times K_1$ matrix Π_{11} must be of full column rank.

We do not observe the true Π_{11}, so we must replace it with an estimate. Anderson's (John Wiley, 1984) approach to testing the rank of this matrix (or that of the full Π matrix) considers the *canonical correlations* of the X and Z matrices. If the equation is to be identified, all K of the canonical correlations will be significantly different from zero.

The squared canonical correlations can be expressed as eigenvalues of a matrix. Anderson's *CC* test considers the null hypothesis that the minimum canonical correlation is zero. Under the null, the test statistic is distributed χ^2 with $(L - K + 1)$ d.f., so it may be calculated even for an exactly-identified equation. Failure to reject the null suggests the equation is unidentified. `ivreg2` routinely reports this Lagrange Multiplier (LM) statistic. In the first example of 2SLS shown above, you see the Anderson canonical correlation statistic as a test for underidentification.

The C–D statistic is a closely related test of the rank of a matrix. While the Anderson *CC* test is a LR test, the C–D test is a Wald statistic, with the same asymptotic distribution. Both the Anderson and C–D tests are reported by `ivreg2` with the `first` option.

Research by Kleibergen and Paap (KP) (*J. Econometrics*, 2006) has developed a robust version of a test for the rank of a matrix: e.g. testing for *underidentification*. The statistic has been implemented by Kleibergen and Schaffer as command `ranktest`, which is part of the `ivreg2` package. If non-*i.i.d*. errors are assumed, the `ivreg2` output contains the K–P `rk` statistic in place of the Anderson canonical correlation statistic as a test of underidentification.

Testing for *i*.*i*.*d*. errors in IV

In the context of an equation estimated with instrumental variables, the standard diagnostic tests for heteroskedasticity and autocorrelation are generally not valid.

In the case of heteroskedasticity, Pagan and Hall (*Econometric Reviews*, 1983) showed that the Breusch–Pagan or Cook–Weisberg tests (`estat hettest`) are generally not usable in an IV setting. They propose a test that will be appropriate in IV estimation where heteroskedasticity may be present in more than one structural equation.

Mark Schaffer's `ivhettest`, part of the `ivreg2` suite, performs the Pagan–Hall test under a variety of assumptions on the indicator variables. It will also reproduce the Breusch–Pagan test if applied in an OLS context.

By the same token, the Breusch–Godfrey statistic used in the OLS context (`estat bgodfrey`) will generally not be appropriate in the presence of endogenous regressors, overlapping data or conditional heteroskedasticity of the error process. Cumby and Huizinga (*Econometrica*, 1992) proposed a generalization of the BG statistic which handles each of these cases.

Their test is actually more general in another way. Its null hypothesis of the test is that the regression error is a moving average of known order $q \geq 0$ against the general alternative that autocorrelations of the regression error are nonzero at lags greater than q. In that context, it can be used to test that autocorrelations beyond any q are zero. Like the BG test, it can test multiple lag orders. The C–H test is available from the SSC Archive as Baum and Schaffer's `actest` routine.

For more details on IV and IV-GMM, please see

- Enhanced routines for instrumental variables/GMM estimation and testing. Baum, C.F., Schaffer, M.E., Stillman, S., *Stata Journal* 7:4, 2007.
- *An Introduction to Modern Econometrics Using Stata*, Baum, C.F., Stata Press, 2006 (particularly Chapter 8).
- Instrumental variables and GMM: Estimation and testing. Baum, C.F., Schaffer, M.E., Stillman, S., *Stata Journal* 3:1–31, 2003.

Both of the *Stata Journal* papers are freely downloadable from `http://stata-journal.com`.