SMU SINGAPORE MANAGEMENT UNIVERSITY
School of Economics

Chapter 9: Panel IV Estimation

This chapter presents some additional linear panel estimation methods, **instrumental variable** (IV) estimation:

- IV estimation – Liner Models
  - Basic IV theory,
  - Model setup,
  - Common IV estimators: IV, 2SLS, and GMM,
  - Instrument validity and relevance, etc.
- Panel IV estimation.
- Hausman-Taylor estimator for FE model.
- Python IV Estimation, `linearmodels`: `IV2SLS`, `IVGMM`, `IVGMMCUE`, `IVLIM`

9.1. IV Estimation – Linear Models

OLS estimator. Consider the multiple linear regression model:

$$Y_i = \alpha + X_i'\beta + u_i = \mathbf{X}_i'\boldsymbol{\beta} + u_i, \quad i = 1, \ldots, n,$$

or in matrix form: $Y = \mathbf{X}\boldsymbol{\beta} + u$, where $\dim(\beta) = k$. The ordinary least squares (OLS) estimator of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_{\text{OLS}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y,$$

which minimizes the sum of squares of errors,

$$\sum_{i=1}^{n}\left(Y_i - \mathbf{X}_i'\boldsymbol{\beta}\right)^2 = (Y - \mathbf{X}\boldsymbol{\beta})'(Y - \mathbf{X}\boldsymbol{\beta}).$$

- The condition for $\widehat{\boldsymbol{\beta}}_{\text{OLS}}$ to be valid (unbiased, consistent): (*i*) $\mathrm{E}(u_i|X_i) = 0$ (**exogeneity** of regressors). Under (*i*), $\mathrm{E}\left(\widehat{\boldsymbol{\beta}}_{\text{OLS}}\middle|\mathbf{X}\right) = E\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y\middle|\mathbf{X}\right) = \boldsymbol{\beta} + E(u|\mathbf{X}) = \boldsymbol{\beta}$, implying that $\mathrm{E}(\widehat{\boldsymbol{\beta}}_{\text{OLS}}) = \boldsymbol{\beta}$ (**unbiased**).
- The conditions for $\widehat{\boldsymbol{\beta}}_{\text{OLS}}$ to be **efficient**: (*ii*) $\mathrm{E}\left(u_i^2|X_i\right) = \sigma^2$ (conditional homoskedasticity), and (*iii*) $\mathrm{E}\left(u_i u_j|X_i, X_j\right) = 0,\ i \neq j$ (conditional zero correlation). Under (*ii*) and (*iii*), $\mathrm{Var}\left(\widehat{\boldsymbol{\beta}}_{\text{OLS}}\right) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ (efficient).

OLS Estimator

Heteroskedasticity-robust standard errors. If homoskedasticity assumption is violated, i.e., $\mathrm{E}(u_i^2|X_i) = \sigma_i^2$ (heteroskedasticity), the OLS estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ remains valid (unbiased, consistent), but $\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) \neq \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$, instead,

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) = (\mathbf{X}'\mathbf{X})^{-1}\mathrm{diag}(\sigma_i^2)(\mathbf{X}'\mathbf{X})^{-1}.$$

A heteroskedasticity-robust estimator of $\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}})$ is

$$\widehat{\mathrm{V}}_{\mathrm{robust}}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) = (\mathbf{X}'\mathbf{X})^{-1}\left(\frac{N}{N-k-1}\sum_i^n \hat{u}_i^2 X_i X_i'\right)(\mathbf{X}'\mathbf{X})^{-1},$$

where $\hat{u}_i$ are OLS residuals, i.e., $\hat{u}_i = y_i - \mathbf{X}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}$.

Cluster-robust standard errors. If observations possess some cluster or group structure, such that the errors are correlated within a cluster but are uncorrelated across clusters, then a cluster-robust estimator of $\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}})$ is

$$\widehat{\mathrm{V}}_{\mathrm{cluster}}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) = (\mathbf{X}'\mathbf{X})^{-1}\left(\frac{G}{G-1}\frac{N}{N-k-1}\sum_g \mathbf{X}_g\hat{\mathbf{u}}_g\hat{\mathbf{u}}_g'\mathbf{X}_g'\right)(\mathbf{X}'\mathbf{X})^{-1},$$

where $\hat{\mathbf{u}}_g$ is the vector of OLS residuals corresponding to the gth cluster, and $\mathbf{X}_g$ is the matrix of regressors' values in the gth cluster, $g = 1, \ldots, G$, $G \to \infty$.

OLS Estimator

If $\mathrm{E}(uu'|\mathbf{X}) = \sigma^2\Omega$, where $\Omega \neq \mathrm{I}$ but is a known correlation matrix ((*ii*) and/or (*iii*) violated), the generalized least-squares (GLS) estimator is:

$$\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}} = (\mathbf{X}'\Omega^{-1}\mathbf{X})^{-1}\mathbf{X}'\Omega^{-1}Y,$$

which minimizes the sum of squares: $(Y - \mathbf{X}\boldsymbol{\beta})'\Omega^{-1}(Y - \mathbf{X}\boldsymbol{\beta})$, and

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}) = \sigma^2(\mathbf{X}'\Omega^{-1}\mathbf{X})^{-1}.$$

- Both $\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ and $\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}$ are unbiased, and consistent.
- But $\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}$ is **more efficient** than $\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}$, because $\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}) = \sigma^2(\mathbf{X}'\Omega^{-1}\mathbf{X})^{-1}$ is "less than" $\mathrm{Var}\left(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}\right) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$.
- In case where Ω is known up to a finite number of parameters γ, i.e., $\Omega = \Omega(\gamma)$, and if a consistent estimator of γ is available, say $\hat{\gamma}$, then a feasible GLS (FGLS) estimator of $\boldsymbol{\beta}$ and its variance are:

$$\widehat{\boldsymbol{\beta}}_{\mathrm{FGLS}} = (\mathbf{X}'\widehat{\Omega}^{-1}\mathbf{X})^{-1}\mathbf{X}'\widehat{\Omega}^{-1}y, \text{ where } \widehat{\Omega} = \Omega(\hat{\gamma});$$

$$\widehat{\mathrm{Var}}(\widehat{\boldsymbol{\beta}}_{\mathrm{FGLS}}) = \hat{\sigma}^2(\mathbf{X}'\widehat{\Omega}^{-1}\mathbf{X})^{-1}, \text{ where } \hat{\sigma}^2 \text{ is a consistent estimator of } \sigma^2.$$

IV Estimation: Basic Idea

The most critical assumption for the validity of the usual linear regression analysis is the exogeneity assumption, $\mathrm{E}(u|X) = 0$. Violation of this assumption renders OLS and GLS inconsistent.

- **Instrumental variables** (IV) provide a consistent estimator under a strong assumption that valid instruments exist, where the instruments $\mathbf{Z}$ are the variables that:

  (i) are correlated with the regressors $\mathbf{X}$;

  (ii) satisfy $\mathrm{E}(u|\mathbf{Z}) = 0$.

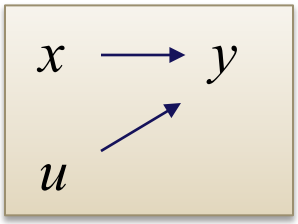
- Consider the simplest linear regression model without an intercept:

  $$\mathrm{y} = x\beta + u,$$

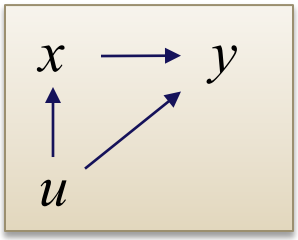
  where y measures earnings, x measures years of schooling, and u is the error term.

- If this simplest model assumes x is unrelated with u, then the only effect of x on y is a direct effect, via the term $x\beta$, as shown below:

IV Estimation: Basic Idea



In the path diagram, the absence of a direct arrow from u to x means that there is no association between x and u. Then, the OLS estimator $\hat{\beta} = \sum_i x_i y_i / \sum_i x_i^2$ is consistent for β.



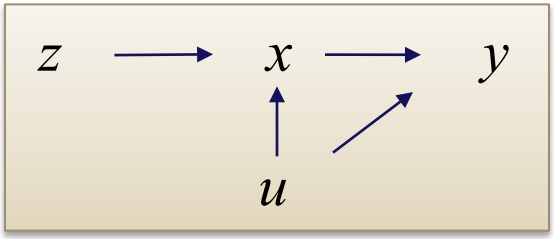
The errors u embodies all factors other than schooling that determine earnings. One such factor is **ability**, which is likely correlated to x, *as high ability tends to lead to high schooling*. The OLS estimator $\hat{\beta}$ is then inconsistent for β, *as* $\hat{\beta}$ combines the desired direct effect of schooling on earnings (β) with the indirect effect of ability: *high* x $\Rightarrow$ *high* u $\Rightarrow$ *high* y.

- A regressor X is said to be **endogenous**, if it arises within a system that influences u. As a consequence, $\mathrm{E}(u|X) \neq 0$;
- By contrast, an **exogenous** regressor arises outside the system and is unrelated to u.

IV Estimation: Basic Idea

An obvious solution to the endogeneity problem is to include, as regressors, controls for ability, called *control function approach*. But such regressors may not be available, and even if they do (e.g., IQ scores), there are questions about the extent to which they measure inherent ability.

IV approach provides an alternative solution. Let z be an IV such that the changes in z is associated with the changes in x but do not lead to changes in y (except indirectly via x). This leads to the path diagram:



For example, proximity to college (z) may determine college attendance (x) but not directly determines earnings (y).

- The IV estimator for this simple example is $\hat{\beta}_{\text{IV}} = \sum_i z_i y_i / \sum_i z_i x_i$.
- The IV estimator $\hat{\beta}_{\text{IV}}$ is consistent for β provided the instrument z is unrelated with the error u and correlated with the regressor x.

IV Estimation: Model Setup

We now consider the more general regression model with a scalar dependent variable Y_1 which depends on m endogenous regressors $\mathbf{Y}_2$, and K_1 exogenous regressors $\mathbf{X}_1$ (including an intercept). This model is called a **structural equation**, with

$$Y_{1i} = \mathbf{X}_{1i}'\beta_1 + \mathbf{Y}_{2i}'\beta_2 + u_i, \quad i = 1, \ldots, n. \tag{9.1}$$

- The u_i are assumed to be uncorrelated with $\mathbf{X}_{1i}$, but are correlated with $\mathbf{Y}_{2i}$, rendering OLS estimator of $\boldsymbol{\beta} = (\beta_1', \beta_2')'$ inconsistent.
- To obtain a consistent estimator, we assume the existence of at least m IVs, say $\mathbf{X}_2$, for $\mathbf{Y}_2$ that satisfy the condition $\mathrm{E}(u_i|\mathbf{X}_{2i}) = 0$.
- The instruments $\mathbf{X}_2$ need to be correlated with $\mathbf{Y}_2$, so that they provide some information on the variables being instrumented. One way to see this is, for each component Y_{2j} of $\mathbf{Y}_2$,

$$Y_{2j,i} = \mathbf{X}_{1i}'\boldsymbol{\pi}_{1j} + \mathbf{X}_{2i}'\boldsymbol{\pi}_{2j} + \varepsilon_{ji}, \qquad j = 1, \ldots, m. \tag{9.2}$$

- Write the model (9.1) as,

$$Y_{1i} = \mathbb{X}_i'\boldsymbol{\beta} + u_i, \tag{9.3}$$

where the regressor vector $\mathbb{X}_i' = [\mathbf{X}_{1i}', \mathbf{Y}_{2i}']$ combines the exogenous and endogenous variables. Let $\mathbb{X}$ be the stacked $\mathbb{X}_i'$.

- Now, let $\mathbb{Z}_i' = \begin{bmatrix}\mathbf{X}_{1i}' & \mathbf{X}_{2i}'\end{bmatrix}$, called collectively the vector of IVs, where $\mathbf{X}_{1i}$ serves as the (ideal) instrument for itself, and $\mathbf{X}_{2i}$ the instrument for $\mathbf{Y}_{2i}$. The instruments satisfy the conditional moment restriction,

$$\mathrm{E}(u_i|\mathbb{Z}_i) = 0. \tag{9.4}$$

- This implies the following population moment condition:

$$\mathrm{E}\{\mathbb{Z}_i(Y_i - \mathbb{X}_i'\boldsymbol{\beta})\} = 0. \ (\textit{Population Moment Condition}) \tag{9.5}$$

Let $\mathbb{Z}$ be the stacked $\mathbb{Z}_i'$, $\mathbf{Y}_1$ the stacked Y_{1i}, and similarly $\mathbf{Y}_2$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$.

" We regress $\mathbf{Y}_1$ on $\mathbb{X}$ using instruments $\mathbb{Z}$ "

The IV estimators are the solutions to the **sample analogue** of (9.5).

IV Estimator

Case I: $\dim(\mathbb{Z}_i) = \dim(\mathbb{X}_i)$: the number of instruments exactly equals to the number of regressors, called the **just-identified** case.

- The sample analogue of (9.5) is

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{Z}_i(Y_{1i} - \mathbb{X}_i'\boldsymbol{\beta}) = 0. \quad (\textit{Sample Moment Condition}) \quad (9.6)$$

which can be written in matrix form,

$$\frac{1}{n}\mathbb{Z}'(\mathbf{Y}_1 - \mathbb{X}\boldsymbol{\beta}) = 0.$$

Solving for $\boldsymbol{\beta}$ leads to the **IV estimator**, if $\mathbb{Z}'\mathbb{X}$ is invertible:

$$\widehat{\boldsymbol{\beta}}_{\text{IV}} = (\mathbb{Z}'\mathbb{X})^{-1}\mathbb{Z}'\mathbf{Y}_1.$$

Case II: $\dim(\mathbb{Z}_i) < \dim(\mathbb{X}_i)$, called the **not-identified** case, where there are fewer instruments than regressors.

In this case, no consistent IV estimator exists.

IV2SLS Estimator

Case III: $\dim(\mathbb{Z}_i) > \dim(\mathbb{X}_i)$, called the **over-identified** case, where there are more instruments than regressors.

Then, $\mathbb{Z}'(\mathbf{Y}_1 - \mathbb{X}\boldsymbol{\beta}) = 0$ has no solution for $\boldsymbol{\beta}$ because it is a system of $\dim(\mathbb{Z}_i)$ equations for $\dim(\mathbb{X}_i)$ unknowns.

- One possibility is to arbitrarily drop instruments to get to the just-identified case. But there are more efficient estimators.
- One is the two-stage least-squares (2SLS) estimator:

$$\widehat{\boldsymbol{\beta}}_{2\text{SLS}} = (\mathbb{X}'\mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\mathbb{X})^{-1}\mathbb{X}'\mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\mathbf{Y}_1.$$

  which is obtained by running two OLS regressions:

  - an OLS regression of $\mathbf{Y}_2$ on $\mathbb{Z}$ in (9.2) to get the predicted values $\widehat{\mathbf{Y}}_2 = \mathbb{P}_{\mathbb{Z}}\mathbf{Y}_2$, where $\mathbb{P}_{\mathbb{Z}} = \mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'$, a projection matrix; and
  - an OLS of (9.1) with $\mathbf{Y}_2$ replaced by $\widehat{\mathbf{Y}}_2$, using $\mathbb{P}_{\mathbb{Z}}\boldsymbol{X}_1 = \boldsymbol{X}_1$! (??)

- $\widehat{\boldsymbol{\beta}}_{2\text{SLS}}$ is the most efficient if u_i are independent and homoscedastic.

IVGMM Estimator

Case III: GMM estimator. Minimizing the objective function:

$$\left[\frac{1}{n}\mathbb{Z}'(\mathbf{Y}_1 - \mathbb{X}\boldsymbol{\beta})\right]' \mathrm{W} \left[\frac{1}{n}\mathbb{Z}'(\mathbf{Y}_1 - \mathbb{X}\boldsymbol{\beta})\right],$$

we obtain the generalized method of moments (GMM) estimator:

$$\widehat{\boldsymbol{\beta}}_{\mathrm{GMM}} = (\mathbb{X}'\mathbb{Z}\mathrm{W}\mathbb{Z}'\mathbb{X})^{-1}\mathbb{X}'\mathbb{Z}\mathrm{W}\mathbb{Z}'\mathbf{Y}_1,$$

where W is any full rank symmetric weighting matrix.

- For just identified case, all choices of W lead to $\widehat{\boldsymbol{\beta}}_{\mathrm{IV}}$.
- Choosing $\mathrm{W} = (\mathbb{Z}'\mathbb{Z})^{-1}$ gives $\widehat{\boldsymbol{\beta}}_{\mathrm{2SLS}}$.
- Choosing $\mathrm{W} = \widehat{\Omega}^{-1}$, where $\widehat{\Omega}$ is an estimate of $\mathrm{Var}(n^{-1/2}\mathbb{Z}'u)$ leads to the optimal GMM (OGMM) estimator:

$$\widehat{\boldsymbol{\beta}}_{\mathrm{OGMM}} = (\mathbb{X}'\mathbb{Z}\widehat{\Omega}^{-1}\mathbb{Z}'\mathbb{X})^{-1}\mathbb{X}'\mathbb{Z}\widehat{\Omega}^{-1}\mathbb{Z}'\mathbf{Y}_1.$$

- If errors are independent, then $\widehat{\Omega} = n^{-1}\sum_{i=1}^{n}\hat{u}_i^2\,\mathbb{Z}_i\mathbb{Z}_i'$, where $\hat{u}_i = Y_{1i} - \mathbb{X}_i'\widehat{\boldsymbol{\beta}}$, with $\widehat{\boldsymbol{\beta}}$ being a consistent estimator, usually $\widehat{\boldsymbol{\beta}}_{\mathrm{2SLS}}$.

Instrument Validity and Relevance

- For all the IV-type estimators, the instruments must satisfy the condition (9.4), i.e., $\mathrm{E}(u_i|\mathbb{Z}_i) = 0$.
- This condition is impossible to test in the just-identified case.
- And even in the over-identified case, where a test is possible, the validity of instruments relies more on pervasive argument, economic theory, and norms established in prior related empirical studies.
- Instruments must be relevant: they ($\mathbf{X}_2$) must account for significant variation in the endogenous variables ($\mathbf{Y}_2$), after controlling the exogenous regressors ($\mathbf{X}_1$).
- Intuitively, the stronger the association between $\mathbb{Z}$ and $\mathbb{X}$, the stronger will be the identification of the model.
- Conversely, instruments that are only marginally relevant are referred to as **weak instruments**. The consequences of weak instruments: estimation much less precise; inference less reliable.

9.2. IV Estimation: Panel Models

IV methods have been extended from the cross-section data to panel data. Two main features for panels remain: estimation still needs to eliminate μ_i if the FE model is appropriate, and inference needs to control for the clustering inherent in panel data.

- Consider the one-way individual effects model:

$$Y_{it} = \alpha + \mathbf{X}_{it}'\boldsymbol{\beta} + \mu_i + v_{it}. \tag{9.7}$$

- The FE and first-difference (FD) estimators provide consistent estimates of $\boldsymbol{\beta}$ under a ***limited form of endogeneity***: $\mathbf{X}_{it}$ may be correlated with μ_i but not with v_{it}.
- Now, we consider a richer type of endogeneity: (some) of $\mathbf{X}_{it}$ are (also) correlated with v_{it}.
- Assume the existence of instruments $\mathbf{Z}_{it}$ that are correlated with $\mathbf{X}_{it}$ but uncorrelated with v_{it}.
- The FE panel IV procedure is to suitably transform the model to control for μ_i and then apply IV to the transformed model.

9.3. *Python* IV Estimation

The first step is to import, from `inearmodels`, the main estimators for linear IV models (see https://bashtage.github.io/linearmodels/).

- `IV2SLS` - standard two-stage least squares
- `IVLIML` - Limited information maximum likelihood and k-class estimators
- `IVGMM` - Generalized method of moment estimation
- `IVGMMCUE` - Generalized method of moment estimation using continuously updating.

```
from linearmodels import IV2SLS, IVGMM, IVGMMCUE, IVLIML
```

UPfIE describes a simple IV estimation procedure that uses the following:

```
import linearmodels.iv as iv
```

In what follows, we use three data sets, two cross-sectional and one panel, to demonstrate various Python IV estimation procedures.

Python IV Estimation: `linearmodels.iv`

To implement IV regression in Python, the module **linearmodels** offers the command **IV2SLS** including the convenient formula syntax from **statsmodels**. When working with IV regression in **linearmodels**, our first line of code is:

```
import linearmodels.iv as iv
```

In the formula specification, the exogenous regressor(s) **x_exg**, the endogenous regressor(s) **x_end**, and the instruments **z** are provided in the following way:

```
y ~ 1 + x_exg + [ x_end ~ z ]
```

- The constant in **linearmodels** is included by adding "**1**" to the formula.
- Options in **cov_type**, **cov_type='unadjusted'** and **cov_type='robust'**, give non-robust and robust standard errors, respectively.
- The default is **cov_type='robust'**.
- For other options, see the module documentation.

Example 15.1 (Woo7): Return to Education for Married Women.

The script (Example-15-1.py, UPfIE) uses data from MROZ to demonstrate the implementation of `linearmodels.iv`.

- We only analyze women with non-missing wage, so we use the method **dropna** to extract them.
- We want to estimate the return to education (**educ**) for these women. As an instrumental variable for education, we use the education of her father (**fatheduc**).
- First, we calculate the OLS and IV slope parameters according to Equations 2.3 and 15.2 of Woo7.
- Then, the full OLS and IV estimates are calculated using the boxed routines **ols** and **IV2SLS**, respectively.
- Not surprisingly, the slope parameters match the manual results.

Python IV Est: Return to Education- MROZ


```
import wooldridge as woo
import numpy as np
import pandas as pd
import linearmodels.iv as iv
import statsmodels.formula.api as smf

mroz = woo.dataWoo('mroz')

# restrict to non-missing wage observations:
mroz = mroz.dropna(subset=['lwage'])

cov_yz = np.cov(mroz['lwage'], mroz['fatheduc'])[1, 0]
cov_xy = np.cov(mroz['educ'], mroz['lwage'])[1, 0]
cov_xz = np.cov(mroz['educ'], mroz['fatheduc'])[1, 0]
var_x = np.var(mroz['educ'], ddof=1)
x_bar = np.mean(mroz['educ'])
y_bar = np.mean(mroz['lwage'])

# OLS slope parameter manually:
b_ols_man = cov_xy / var_x
print(f'b_ols_man: {b_ols_man}\n’)

# IV slope parameter manually:
b_iv_man = cov_yz / cov_xz
print(f'b_iv_man: {b_iv_man}\n')
```


Python IV Est: Return to Education- MROZ


```
# OLS automatically:
reg_ols = smf.ols(formula='np.log(wage) ~ educ', data=mroz)
results_ols = reg_ols.fit()

# print regression table:
table_ols = pd.DataFrame({'b': round(results_ols.params, 4),
                          'se': round(results_ols.bse, 4),
                          't': round(results_ols.tvalues, 4),
                          'pval': round(results_ols.pvalues, 4)})
print(f'table_ols: \n{table_ols}\n')

# IV automatically:
reg_iv = iv.IV2SLS.from_formula(formula='np.log(wage) ~ 1 + [educ ~ fatheduc]',
                                data=mroz)
results_iv = reg_iv.fit(cov_type='unadjusted', debiased=True)

# print regression table:
table_iv = pd.DataFrame({'b': round(results_iv.params, 4),
                         'se': round(results_iv.std_errors, 4),
                         't': round(results_iv.tstats, 4),
                         'pval': round(results_iv.pvalues, 4)})
print(f'table_iv: \n{table_iv}\n')
```

Python IV Est: Return to Education- MROZ


```
b_ols_man: 0.10864865517467516

b_iv_man: 0.05917347999936596

table_ols:
                b      se       t   pval
Intercept -0.1852  0.1852 -0.9998  0.318
educ       0.1086  0.0144  7.5451  0.000

table_iv:
                b      se       t    pval
Intercept  0.4411  0.4461  0.9888  0.3233
educ       0.0592  0.0351  1.6839  0.0929
```

- The manual and automatic calculation produce identical estimates of the slopes of **`educ`** variable.
- The `ols` and `IV2SLS` produce substantially different results.

The IV approach can easily be generalized to include additional exogenous regressors, i.e. regressors that are assumed to be unrelated to the error term.

To demonstrate this, we use Woo7 `CARD` data with formula specification:

```
y ~ 1 + x_exog + [ x_end ~ z ]
```


Python IV Est: Return to Education-CARD

`Example-15-4.py` (UPfIE) uses CARD to estimate the return to education, **`educ`**, which allowed to be endogenous and instrumented with the dummy variable **`nearc4`** indicating whether the individual grew up close to a college.

- In addition, we control for experience, race, and regional information. These variables are assumed to be exogenous and act as their own instruments.
- We first check for relevance by regressing the endogenous independent variable **`educ`** on all exogenous variables including the instrument **nearc4**.
- Its parameter is highly significantly different from zero, so relevance is supported. We then estimate the log wage equation with OLS and IV.

```
import wooldridge as woo
import pandas as pd
import statsmodels.formula.api as smf

card = woo.dataWoo('card')

datadescribe = woo.data('card', description=True)
print(f'Data Description: \n{datadescribe}\n')
```


Python IV Est: Return to Education-CARD


```
name of dataset: card
no of variables: 34
no of observations: 3010
```

variable	label
id	person identifier
nearc2	=1 if near 2 yr college, 1966
nearc4	=1 if near 4 yr college, 1966
educ	years of schooling, 1976
age	in years
fatheduc	father's schooling
motheduc	mother's schooling
weight	NLS sampling weight, 1976
momdad14	=1 if live with mom, dad at 14
sinmom14	=1 if with single mom at 14
step14	=1 if with step parent at 14
reg661	=1 for region 1, 1966
reg662	=1 for region 2, 1966
reg663	=1 for region 3, 1966
reg664	=1 for region 4, 1966
reg665	=1 for region 5, 1966
reg666	=1 for region 6, 1966
reg667	=1 for region 7, 1966
reg668	=1 for region 8, 1966
reg669	=1 for region 9, 1966
south66	=1 if in south in 1966
black	=1 if black
smsa	=1 in in SMSA, 1976
south	=1 if in south, 1976
smsa66	=1 if in SMSA, 1966
wage	hourly wage in cents, 1976
enroll	=1 if enrolled in school, 1976
KWW	knowledge world of work score
IQ	IQ score
married	=1 if married, 1976
libcrd14	=1 if lib. card in home at 14
exper	age - educ - 6
lwage	log(wage)
expersq	exper^2

D. Card (1995), "Using Geographic Variation in College Proximity to Estimate the Return to Schooling," in Aspects of Labour Market Behavior: Essays in Honour of John Vanderkamp. Ed. L.N. Christophides, E.K. Grant, and R. Swidinsky, 201-222. Toronto: University of Toronto Press. Professor Card kindly provided these data.

Python IV Est: Return to Education-CARD


```
# checking for relevance with reduced form:
reg_redf = smf.ols(
    formula='educ ~ nearc4 + exper + I(exper**2) + black + smsa +'
    'south + smsa66 + reg662 + reg663 + reg664 + reg665 + reg666 +'
    'reg667 + reg668 + reg669', data=card)
results_redf = reg_redf.fit()

# print regression table:
table_redf = pd.DataFrame({'b': round(results_redf.params, 4),
                           'se': round(results_redf.bse, 4),
                           't': round(results_redf.tvalues, 4),
                           'pval': round(results_redf.pvalues, 4)})
print(f'table_redf: \n{table_redf}\n')

# OLS:
reg_ols = smf.ols(
    formula='np.log(wage) ~ educ + exper + I(exper**2) + black + smsa +'
    'south + smsa66 + reg662 + reg663 + reg664 + reg665 +'
    'reg666 + reg667 + reg668 + reg669', data=card)
results_ols = reg_ols.fit()

# print regression table:
table_ols = pd.DataFrame({'b': round(results_ols.params, 4),
                          'se': round(results_ols.bse, 4),
                          't': round(results_ols.tvalues, 4),
                          'pval': round(results_ols.pvalues, 4)})
print(f'table_ols: \n{table_ols}\n')
```


Python IV Est: Return to Education-CARD

table_redf:

	b	se	t	pval
Intercept	16.6383	0.2406	69.1446	0.0000
nearc4	0.3199	0.0879	3.6408	0.0003
exper	-0.4125	0.0337	-12.2415	0.0000
I(exper ** 2)	0.0009	0.0017	0.5263	0.5987
black	-0.9355	0.0937	-9.9806	0.0000
smsa	0.4022	0.1048	3.8372	0.0001
south	-0.0516	0.1354	-0.3811	0.7032
smsa66	0.0255	0.1058	0.2409	0.8096
reg662	-0.0786	0.1871	-0.4203	0.6743
reg663	-0.0279	0.1834	-0.1524	0.8789
reg664	0.1172	0.2173	0.5394	0.5897
reg665	-0.2726	0.2184	-1.2481	0.2121
reg666	-0.3028	0.2371	-1.2773	0.2016
reg667	-0.2168	0.2344	-0.9250	0.3550
reg668	0.5239	0.2675	1.9587	0.0502
reg669	0.2103	0.2025	1.0386	0.2991

table_ols:

	b	se	t	pval
Intercept	4.6208	0.0742	62.2476	0.0000
educ	0.0747	0.0035	21.3510	0.0000
exper	0.0848	0.0066	12.8063	0.0000
I(exper ** 2)	-0.0023	0.0003	-7.2232	0.0000
black	-0.1990	0.0182	-10.9058	0.0000
smsa	0.1364	0.0201	6.7851	0.0000
south	-0.1480	0.0260	-5.6950	0.0000
smsa66	0.0262	0.0194	1.3493	0.1773
reg662	0.0964	0.0359	2.6845	0.0073
reg663	0.1445	0.0351	4.1151	0.0000
reg664	0.0551	0.0417	1.3221	0.1862
reg665	0.1280	0.0418	3.0599	0.0022
reg666	0.1405	0.0452	3.1056	0.0019
reg667	0.1180	0.0448	2.6334	0.0085
reg668	-0.0564	0.0513	-1.1010	0.2710
reg669	0.1186	0.0388	3.0536	0.0023

Python IV Est: Return to Education-CARD


```
# IV automatically:
reg_iv = iv.IV2SLS.from_formula(
    formula='np.log(wage)~ 1 + exper + I(exper**2) + black + smsa + '
            'south + smsa66 + reg662 + reg663 + reg664 + reg665 +'
            'reg666 + reg667 + reg668 + reg669 + [educ ~ nearc4]',
    data=card)
results_iv = reg_iv.fit(cov_type='unadjusted', debiased=True)

# print regression table:
table_iv = pd.DataFrame({'b': round(results_iv.params, 4),
                         'se': round(results_iv.std_errors, 4),
                         't': round(results_iv.tstats, 4),
                         'pval': round(results_iv.pvalues, 4)})
print(f'table_iv: \n{table_iv}\n')
```

table_iv:

	b	se	t	pval
Intercept	3.6662	0.9248	3.9641	0.0001
exper	0.1083	0.0237	4.5764	0.0000
I(exper ** 2)	-0.0023	0.0003	-7.0014	0.0000
black	-0.1468	0.0539	-2.7231	0.0065
smsa	0.1118	0.0317	3.5313	0.0004
south	-0.1447	0.0273	-5.3023	0.0000
smsa66	0.0185	0.0216	0.8576	0.3912
reg662	0.1008	0.0377	2.6739	0.0075
reg663	0.1483	0.0368	4.0272	0.0001
reg664	0.0499	0.0437	1.1408	0.2541
reg665	0.1463	0.0471	3.1079	0.0019
reg666	0.1629	0.0519	3.1382	0.0017
reg667	0.1346	0.0494	2.7240	0.0065
reg668	-0.0831	0.0593	-1.4002	0.1616
reg669	0.1078	0.0418	2.5784	0.0100
educ	0.1315	0.0550	2.3926	0.0168

Python 2SLS

Two stage least squares (2SLS) is a general approach for IV estimation when we have one or more endogenous regressors and at least as many additional instrumental variables. Consider the regression model

$$Y_1 = \beta_0 + \beta_1 Y_2 + \beta_2 Y_3 + \beta_3 Z_1 + \beta_4 Z_2 + \beta_5 Z_3 + u_1,$$

The regressors Y_2 and Y_3 are potentially correlated with the error term u_1, and the regressors Z_1, Z_2, and Z_3 are assumed to be exogenous.

Because we have two endogenous regressors, we need at least two additional instrumental variables, say Z_4 and Z_5.

The name of 2SLS performs two stages of OLS regressions:

(1) Separately regress Y_2 and Y_3 on Z_1 through Z_5.
Obtain fitted values $\hat{Y}_2$ and $\hat{Y}_3$;

(2) Regress Y_1 on $\hat{Y}_2$ and $\hat{Y}_3$, and Z_1, Z_2, and Z_3.

If the instruments are valid, this will give consistent estimates of the parameters β_0 though β_5. Generalizing this to more endogenous regressors and instrumental variables is obvious.

Python 2SLS: Return to Education - MROZ

This procedure can of course easily be implemented using **ols** in **statsmodels**, remembering that fitted values are saved in **fittedvalues**.

One of the problems of this manual approach is that the resulting variance-covariance matrix and analyses based on them are invalid.

Conveniently, **IV2SLS** will automatically do these calculations and calculate correct standard errors and the like.

Example 15.5 (Woo7): Return to Education for Married Women

We continue Example 15.1 and still want to estimate the return to education for women using the data in MROZ. Now, we use both mother's and father's education as instruments for their own education.

Example-15-5.py (PYfIE) gives 2SLS estimates in two ways: First, we do both stages manually, including fitted education as **educ_fitted** as a regressor in the second stage. **IV2SLS** does this automatically and delivers the same parameter estimates as the output table reveals. But the standard errors differ slightly because the manual two stage version did not correct them.

Python 2SLS: Return to Education - MROZ


```
import wooldridge as woo
import numpy as np
import pandas as pd
import linearmodels.iv as iv
import statsmodels.formula.api as smf

mroz = woo.dataWoo('mroz')

# restrict to non-missing wage observations:
mroz = mroz.dropna(subset=['lwage'])

# 1st stage (reduced form):
reg_redf = smf.ols(formula='educ ~ exper + I(exper**2) + motheduc + fatheduc',
                   data=mroz)
results_redf = reg_redf.fit()
mroz['educ_fitted'] = results_redf.fittedvalues

# print regression table:
table_redf = pd.DataFrame({'b': round(results_redf.params, 4),
                           'se': round(results_redf.bse, 4),
                           't': round(results_redf.tvalues, 4),
                           'pval': round(results_redf.pvalues, 4)})
print(f'table_redf: \n{table_redf}\n')

# 2nd stage:
reg_secstg = smf.ols(formula='np.log(wage) ~ educ_fitted + exper + I(exper**2)',
                     data=mroz)
results_secstg = reg_secstg.fit()
```

Python 2SLS: Return to Education - MROZ


```
# print regression table:
table_secstg = pd.DataFrame({'b': round(results_secstg.params, 4),
                             'se': round(results_secstg.bse, 4),
                             't': round(results_secstg.tvalues, 4),
                             'pval': round(results_secstg.pvalues, 4)})
print(f'table_secstg: \n{table_secstg}\n’)

# IV automatically:
reg_iv = iv.IV2SLS.from_formula(
    formula='np.log(wage) ~ 1 + exper + I(exper**2) +'
            '[educ  ~ motheduc + fatheduc]',
    data=mroz)
results_iv = reg_iv.fit(cov_type='unadjusted', debiased=True)

# print regression table:
table_iv = pd.DataFrame({'b': round(results_iv.params, 4),
                         'se': round(results_iv.std_errors, 4),
                         't': round(results_iv.tstats, 4),
                         'pval': round(results_iv.pvalues, 4)})
print(f'table_iv: \n{table_iv}\n')
```

Python 2SLS: Return to Education - MROZ


```
table_redf:
                      b      se        t    pval
Intercept        9.1026  0.4266  21.3396  0.0000
exper            0.0452  0.0403   1.1236  0.2618
I(exper ** 2)   -0.0010  0.0012  -0.8386  0.4022
motheduc         0.1576  0.0359   4.3906  0.0000
fatheduc         0.1895  0.0338   5.6152  0.0000

table_secstg:
                      b      se        t    pval
Intercept        0.0481  0.4198   0.1146  0.9088
educ_fitted      0.0614  0.0330   1.8626  0.0632
exper            0.0442  0.0141   3.1361  0.0018
I(exper ** 2)   -0.0009  0.0004  -2.1344  0.0334

table_iv:
                      b      se        t    pval
Intercept        0.0481  0.4003   0.1202  0.9044
exper            0.0442  0.0134   3.2883  0.0011
I(exper ** 2)   -0.0009  0.0004  -2.2380  0.0257
educ             0.0614  0.0314   1.9530  0.0515
```

Additional issues:

Testing for exogeneity of the regressors;

Testing Overidentifying Restrictions.

See UPfIE, Sections 15.4, and 15.5.

GMM estimation with Python?

Python 2SLS, IVGMM, IVGMMCUE, IVLIML

The following example, drawn from

https://bashtage.github.io/linearmodels/

gives simpler ways of implementing various IV estimation methods:

- `IV2SLS` - standard two-stage least squares
- `IVLIML` - Limited information maximum likelihood and k-class estimators
- `IVGMM` - Generalized method of moment estimation
- `IVGMMCUE` - Generalized method of moment estimation using continuously updating.

The data comes from the Medical Expenditure Panel Survey (MEPS) and includes data on

- out-of-pocket drug expenditure (in logs), individual characteristics,
- whether an individual was insured through an employer or union (a likely endogenous variable), and some candidate instruments including
- the percentage of income from Social Security Income, the size of the individual firm and whether the firm has multiple locations.

Python Setup: Medical Expenditure

Load the methods from `linearmodels`, import data, and see data description:

```
from linearmodels import IV2SLS, IVGMM, IVGMMCUE, IVLIML

from linearmodels.datasets import meps

data = meps.load()
data = data.dropna()
print(meps.DESCR)
```

age	Age		
age2	Age-squared	ldrugexp	log(drugexp)
black	Black	marry	Married
blhisp	Black or Hispanic	midincome	Middle income
drugexp	Presc-drugs expense	msa	Metropolitan stat area
educyr	Years of education	multlc	Multiple locations
fair	Fair health	poor	Poor health
female	Female	poverty	Poor
firmsz	Firm size	priolist	Priority list cond
fph	fair or poor health	private	Private insurance
good	Good health	ssiratio	SSI/Income ratio
hi_empunion	Insured thro emp/union	totchr	Total chronic cond
hisp	Hiapanic	vegood	V-good health
income	Income	vgh	vg or good health

Python Setup: Medical Expenditure

Independent variable, control variables and instrumental variables:

```
controls = ["totchr", "female", "age", "linc", "blhisp"]
print(data[["ldrugexp", "hi_empunion"] + controls].describe(percentiles=[]))
```

	ldrugexp	hi_empunion	...	linc	blhisp
count	10089.000000	10089.000000	...	10089.000000	10089.000000
mean	6.481361	0.382198	...	2.743275	0.163544
std	1.362052	0.485949	...	0.913143	0.369880
min	0.000000	0.000000	...	-6.907755	0.000000
50%	6.678342	0.000000	...	2.743160	0.000000
max	10.180172	1.000000	...	5.744476	1.000000

```
instruments = ["ssiratio", "lowincome", "multlc", "firmsz"]
print(data[instruments].describe(percentiles=[]))
```

	ssiratio	lowincome	multlc	firmsz
count	10089.000000	10089.000000	10089.000000	10089.000000
mean	0.536544	0.187432	0.062048	0.140529
std	0.367818	0.390277	0.241254	2.170389
min	0.000000	0.000000	0.000000	0.000000
50%	0.504522	0.000000	0.000000	0.000000
max	9.250620	1.000000	1.000000	50.000000

Python Setup: Medical Expenditure

Finally, the simple correlation between the endogenous variable and the instruments are calculated.

- Instruments must be correlated to be relevant (but also must be exogenous, which cannot be examined using simple correlation).
- The correlation of firmsz is especially low, which might lead to the weak instruments problem is used extensively.

```
data[["hi_empunion"] + instruments].corr()
```

	hi_empunion	ssiratio	lowincome	multlc	firmsz
hi_empunion	1.000000	-0.212431	-0.116419	0.119849	0.037352
ssiratio	-0.212431	1.000000	0.253946	-0.190433	-0.044578
lowincome	-0.116419	0.253946	1.000000	-0.062465	-0.008232
multlc	0.119849	-0.190433	-0.062465	1.000000	0.187275
firmsz	0.037352	-0.044578	-0.008232	0.187275	1.000000

`add_constant` from `statsmodels` adds a constant column to the data.

```
from statsmodels.api import OLS, add_constant

data["const"] = 1
controls = ["const"] + controls
```


Python OLS: Medical Expenditure

Run OLS through 2SLS

Before examining the IV estimators, it is worth noting that 2SLS nests the OLS estimator, so that a call to IV2SLS using none for both the endogenous and the instruments will produce OLS estimates of the parameters.

The OLS estimates indicate that insurance through an employer or union leads to an increase in out-of-pocket drug expenditure.

```
ivolsmod = IV2SLS(data.ldrugexp, data[["hi_empunion"] + controls], None,
None)
res_ols = ivolsmod.fit()
print(res_ols)
```

Python OLS: Medical Expenditure

OLS Estimation Summary

Dep. Variable:	ldrugexp	R-squared:	0.1770
Estimator:	OLS	Adj. R-squared:	0.1765
No. Observations:	10089	F-statistic:	2262.6
Date:	Sat, Feb 17 2024	P-value (F-stat)	0.0000
Time:	17:20:53	Distribution:	chi2(6)
Cov. Estimator:	robust		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
hi_empunion	0.0739	0.0260	2.8441	0.0045	0.0230	0.1248
const	5.8611	0.1570	37.320	0.0000	5.5533	6.1689
totchr	0.4404	0.0094	47.049	0.0000	0.4220	0.4587
female	0.0578	0.0254	2.2797	0.0226	0.0081	0.1075
age	-0.0035	0.0019	-1.8228	0.0683	-0.0073	0.0003
linc	0.0105	0.0137	0.7646	0.4445	-0.0164	0.0373
blhisp	-0.1513	0.0341	-4.4353	0.0000	-0.2182	-0.0844

Python IV2SLS: Medical Expenditure

IV Estimator (just identified case)

The just identified two-stage LS estimator uses as many instruments as endogenous variables.

In this example there is one of each, using the SSI ratio as the instrument.

With the instrument, the effect of insurance through employer or union (hi_empunion) has a strong negative effect on drug expenditure.

```
ivmod = IV2SLS(data.ldrugexp, data[controls], data.hi_empunion, data.ssiratio)
res_2sls = ivmod.fit()
print(res_2sls.summary)
```

```
                          IV-2SLS Estimation Summary
==============================================================================
Dep. Variable:                ldrugexp   R-squared:                      0.0640
Estimator:                     IV-2SLS   Adj. R-squared:                 0.0634
No. Observations:                10089   F-statistic:                    2000.9
Date:                Sat, Feb 17 2024   P-value (F-stat)                0.0000
Time:                        17:38:46   Distribution:                  chi2(6)
                        Cov. Estimator:                robust
```

Python IV2SLS: Medical Expenditure

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	6.7872	0.2688	25.246	0.0000	6.2602	7.3141
totchr	0.4503	0.0102	44.157	0.0000	0.4303	0.4703
female	-0.0204	0.0326	-0.6257	0.5315	-0.0843	0.0435
age	-0.0132	0.0030	-4.4092	0.0000	-0.0191	-0.0073
linc	0.0870	0.0226	3.8436	0.0001	0.0426	0.1314
blhisp	-0.2174	0.0395	-5.5052	0.0000	-0.2948	-0.1400
hi_empunion	-0.8976	0.2211	-4.0592	0.0000	-1.3310	-0.4642

Endogenous: hi_empunion
Instruments: ssiratio
Robust Covariance (Heteroskedastic)
Debiased: False

IV2SLS with Multiple Instruments (overidentified case)

```
ivmod = IV2SLS(data.ldrugexp, data[controls], data.hi_empunion,\
               data[["ssiratio", "multlc"]])
res_2sls_robust = ivmod.fit()
print(res_2sls_robust.summary)
```

Python IV2SLS: Medical Expenditure

IV-2SLS Estimation Summary

Dep. Variable:	ldrugexp	R-squared:	0.0414
Estimator:	IV-2SLS	Adj. R-squared:	0.0409
No. Observations:	10089	F-statistic:	1955.4
Date:	Sat, Feb 17 2024	P-value (F-stat)	0.0000
Time:	17:54:54	Distribution:	chi2(6)
Cov. Estimator:	robust		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	6.8752	0.2579	26.660	0.0000	6.3697	7.3806
totchr	0.4512	0.0103	43.769	0.0000	0.4310	0.4714
female	-0.0278	0.0322	-0.8653	0.3869	-0.0909	0.0352
age	-0.0141	0.0029	-4.8753	0.0000	-0.0198	-0.0085
linc	0.0943	0.0219	4.3079	0.0000	0.0514	0.1372
blhisp	-0.2237	0.0396	-5.6514	0.0000	-0.3013	-0.1461
hi_empunion	-0.9899	0.2046	-4.8386	0.0000	-1.3909	-0.5889

Endogenous: hi_empunion — Robust Covariance (Heteroskedastic)
Instruments: ssiratio, multlc — Debiased: False

Alternative covariance estimators

All estimator allow for three types of parameter covariance estimator:

- "unadjusted" is the classic homoskedastic estimator
- "robust" is robust to heteroskedasticity
- "clustered" allows one- or two-way clustering to account for additional sources of dependence between the model scores
- "kernel" produces a heteroskedasticity-autocorrelation robust covariance estimator

The default is "robust".

These are all passed using the keyword input `cov_type`. Using clustered requires also passing the clustering variable(s).

```
ivmod = IV2SLS(data.ldrugexp, data[controls], data.hi_empunion, \
        data[["ssiratio", "multlc"]])
res_2sls_std = ivmod.fit(cov_type="unadjusted")
print(res_2sls_std.summary)
```

GMM Estimation

GMM estimation can be more efficient than 2SLS when there are more instruments than endogenous regressors.

- By default, 2-step efficient GMM is used (assuming the weighting matrix is correctly specified).
- It is possible to iterate until convergence using the optional keyword input `iter_limit`, which is naturally 2 by default.
- Generally, GMM-CUE would be preferred to using multiple iterations of standard GMM.
- The default weighting matrix is robust to heteroskedasticity (but not clustering).

```
ivmod = IVGMM(data.ldrugexp, data[controls], \
        data.hi_empunion, data[["ssiratio", "multlc"]])
res_gmm = ivmod.fit()
print(res_gmm.summary)
```

Python IVGMM: Medical Expenditure

IV-GMM Estimation Summary

Dep. Variable:	ldrugexp	R-squared:	0.0406
Estimator:	IV-GMM	Adj. R-squared:	0.0400
No. Observations:	10089	F-statistic:	1952.6
Date:	Sat, Feb 17 2024	P-value (F-stat)	0.0000
Time:	18:26:52	Distribution:	chi2(6)
Cov. Estimator:	robust		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	6.8778	0.2580	26.658	0.0000	6.3722	7.3835
totchr	0.4510	0.0103	43.738	0.0000	0.4307	0.4712
female	-0.0282	0.0322	-0.8752	0.3815	-0.0913	0.0349
age	-0.0142	0.0029	-4.8773	0.0000	-0.0198	-0.0085
linc	0.0945	0.0219	4.3142	0.0000	0.0515	0.1374
blhisp	-0.2231	0.0396	-5.6344	0.0000	-0.3007	-0.1455
hi_empunion	-0.9933	0.2047	-4.8530	0.0000	-1.3944	-0.5921

Endogenous: hi_empunion — Robust (Heteroskedastic)
Instruments: ssiratio, multlc — GMM Covariance — Debiased: False

Changing the weighting matrix structure in GMM estimation

The weighting matrix in the GMM objective function can be altered when creating the model. This example uses clustered weight by age. The covariance estimator should usually match the weighting matrix, and so clustering is also used here.

```
# IVGMM Estimation
ivmod = IVGMM(
    data.ldrugexp,
    data[controls],
    data.hi_empunion,
    data[["ssiratio", "multlc"]],
    weight_type="clustered",
    clusters=data.age,
)
res_gmm_clustered = ivmod.fit(cov_type="clustered", clusters=data.age)
```

Python IVGMM: Medical Expenditure

IV-GMM Estimation Summary

Dep. Variable:	ldrugexp	R-squared:	0.0292
Estimator:	IV-GMM	Adj. R-squared:	0.0286
No. Observations:	10089	F-statistic:	1700.8
Date:	Sun, Feb 18 2024	P-value (F-stat)	0.0000
Time:	11:31:51	Distribution:	chi2(6)
Cov. Estimator:	clustered		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	6.7277	0.5059	13.299	0.0000	5.7362	7.7192
totchr	0.4482	0.0132	33.833	0.0000	0.4223	0.4742
female	-0.0245	0.0292	-0.8398	0.4010	-0.0817	0.0327
age	-0.0119	0.0063	-1.8928	0.0584	-0.0241	0.0004
linc	0.0957	0.0147	6.4934	0.0000	0.0668	0.1246
blhisp	-0.2091	0.0502	-4.1662	0.0000	-0.3074	-0.1107
hi_empunion	-1.0359	0.2044	-5.0683	0.0000	-1.4365	-0.6353

Endogenous: hi_empunion	Debiased: False
Instruments: ssiratio, multlc	Clustered (One-way)
GMM Covariance	Num Clusters: 27

Python IVGMMCUE: Medical Expenditure


```
ivmod = IVGMMCUE(data.ldrugexp, data[controls], data.hi_empunion, \
        data[["ssiratio", "multlc"]])
res_gmm_cue = ivmod.fit(cov_type="robust", display=True)
```

```
ivmod = IVGMMCUE(data.ldrugexp, data[controls], data.hi_empunion, \
        data[["ssiratio", "multlc"]])
res_gmm_cue = ivmod.fit(cov_type="robust", display=True)
print(res_gmm_cue.summary)
Warning: Desired error not necessarily achieved due to precision loss.
         Current function value: 1.045365
         Iterations: 10
         Function evaluations: 420
         Gradient evaluations: 51
==============================================================================
Dep. Variable:                ldrugexp   R-squared:                       0.0388
Estimator:                      IV-GMM   Adj. R-squared:                  0.0382
No. Observations:                10089   F-statistic:                     1949.2
Date:               Sun, Feb 18 2024   P-value (F-stat)                 0.0000
Time:                       11:37:24   Distribution:                   chi2(6)
Cov. Estimator:               robust
```

Python IVGMMCUE: Medical Expenditure

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
const	6.8847	0.2583	26.657	0.0000	6.3785	7.3908
totchr	0.4510	0.0103	43.701	0.0000	0.4308	0.4712
female	-0.0288	0.0322	-0.8928	0.3720	-0.0919	0.0344
age	-0.0142	0.0029	-4.8969	0.0000	-0.0199	-0.0085
linc	0.0950	0.0219	4.3333	0.0000	0.0520	0.1380
blhisp	-0.2236	0.0396	-5.6421	0.0000	-0.3013	-0.1459
hi_empunion	-1.0002	0.2049	-4.8810	0.0000	-1.4019	-0.5986

Endogenous: hi_empunion
Instruments: ssiratio, multlc
GMM Covariance
Debiased: False
Robust (Heteroskedastic)

Comparing results

The function `compare` can be used to compare the results of multiple models, possibly with different variables, estimators and/or instruments.

- Usually a `dictionary` or `OrderedDict` is used to hold results since the keys are used as model names.
- The advantage of an `OrderedDict` is that it will preserve the order of the models in the presentation.
- With the expectation of the OLS estimate, the parameter estimates are fairly consistent. Standard errors vary slightly although the conclusions reached are not sensitive to the choice of covariance estimator either.
- T-stats are reported in parentheses.

Python Compare Results: Medical Expenditure

Model Comparison

	OLS	2SLS	2SLS-Homo	2SLS-Hetero	GMM	GMM Cluster(Age)	GMM-CUE
Dep. Variable	ldrugexp	ldrugexp	ldrugexp	ldrugexp	ldrugexp	ldrugexp	ldrugexp
Estimator	OLS	IV-2SLS	IV-2SLS	IV-2SLS	IV-GMM	IV-GMM	IV-GMM
No. Observations	10089	10089	10089	10089	10089	10089	10089
Cov. Est.	robust	robust	unadjusted	robust	robust	clustered	robust
R-squared	0.1770	0.0640	0.0414	0.0414	0.0406	0.0292	0.0388
Adj. R-squared	0.1765	0.0634	0.0409	0.0409	0.0400	0.0286	0.0382
F-statistic	2262.6	2000.9	1882.3	1955.4	1952.6	1700.8	1949.2
P-value (F-stat)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
hi_empunion	0.0739	-0.8976	-0.9899	-0.9899	-0.9933	-1.0359	-1.0002
	(2.8441)	(-4.0592)	(-5.1501)	(-4.8386)	(-4.8530)	(-5.0683)	(-4.8810)
const	5.8611	6.7872	6.8752	6.8752	6.8778	6.7277	6.8847
	(37.320)	(25.246)	(28.030)	(26.660)	(26.658)	(13.299)	(26.657)
totchr	0.4404	0.4503	0.4512	0.4512	0.4510	0.4482	0.4510
	(47.049)	(44.157)	(42.942)	(43.769)	(43.738)	(33.833)	(43.701)
female	0.0578	-0.0204	-0.0278	-0.0278	-0.0282	-0.0245	-0.0288
	(2.2797)	(-0.6257)	(-0.8933)	(-0.8653)	(-0.8752)	(-0.8398)	(-0.8928)
age	-0.0035	-0.0132	-0.0141	-0.0141	-0.0142	-0.0119	-0.0142
	(-1.8228)	(-4.4092)	(-5.0834)	(-4.8753)	(-4.8773)	(-1.8928)	(-4.8969)
linc	0.0105	0.0870	0.0943	0.0943	0.0945	0.0957	0.0950
	(0.7646)	(3.8436)	(4.4400)	(4.3079)	(4.3142)	(6.4934)	(4.3333)
blhisp	-0.1513	-0.2174	-0.2237	-0.2237	-0.2231	-0.2091	-0.2236
	(-4.4353)	(-5.5052)	(-5.7805)	(-5.6514)	(-5.6344)	(-4.1662)	(-5.6421)
Instruments		ssiratio	ssiratio multlc	ssiratio multlc	ssiratio multlc	ssiratio multlc	ssiratio multlc

T-stats reported in parentheses

Python LIML: Medical Expenditure

LIML (Limited Information Maximum Likelihood):

The LIML and the related *k*-class estimators can be obtained from `IVLIML`.

LIML can have better finite sample properties if the model is not strongly identified. A parameter, κ, is estimated, which is very close to 1, and the results for LIML are similar to these fir 2SLS (they would be the sanme if κ=1).

```
ivmod = IVLIML(data.ldrugexp, data[controls], data.hi_empunion, \
        data[["ssiratio", "multlc"]])
res_liml = ivmod.fit(cov_type="robust")
print(compare({"2SLS": res_2sls_robust, "LIML": res_liml, "GMM": res_gmm}))
```

	2SLS	LIML	GMM
Dep. Variable	ldrugexp	ldrugexp	ldrugexp
Estimator	IV-2SLS	IV-LIML	IV-GMM
No. Observations	10089	10089	10089
Cov. Est.	robust	robust	robust
R-squared	0.0414	0.0400	0.0406
Adj. R-squared	0.0409	0.0394	0.0400
F-statistic	1955.4	1952.3	1952.6
P-value (F-stat)	0.0000	0.0000	0.0000

Python LIML: Medical Expenditure

const	6.8752	6.8807	6.8778
	(26.660)	(26.577)	(26.658)
totchr	0.4512	0.4513	0.4510
	(43.769)	(43.730)	(43.738)
female	-0.0278	-0.0283	-0.0282
	(-0.8653)	(-0.8776)	(-0.8752)
age	-0.0141	-0.0142	-0.0142
	(-4.8753)	(-4.8781)	(-4.8773)
linc	0.0943	0.0947	0.0945
	(4.3079)	(4.3114)	(4.3142)
blhisp	-0.2237	-0.2241	-0.2231
	(-5.6514)	(-5.6531)	(-5.6344)
hi_empunion	-0.9899	-0.9957	-0.9933
	(-4.8386)	(-4.8361)	(-4.8530)
Instruments	ssiratio multlc	ssiratio multlc	ssiratio multlc

T-stats reported in parentheses

Example 15.10 (Woo7): Job Training and Worker Productivity

- The example uses the data set `JTRAIN` to estimate the effect of job training **`hrsemp`** on the scrap rate **`scrap`**.
- `Example-15-10.py` (UPfIE) chooses a subset of the years 1987 and 1988 with the command **`loc`**, and store the data with correct index variables **`fcode`** and **`year`**, see Section 13.3.
- Then we estimate the parameters using first-differencing with the instrumental variable **`grant`**.
- **`dropna`** needs to be applied to the differenced data to drop the null values.

We first load the data and see the description of it.

```
import wooldridge as woo
jtrain = woo.dataWoo('jtrain')
woo.data('jtrain', description=True)
```

Job Training and Worker Productivity

```
name of dataset: jtrain
no of variables: 30
no of observations: 471
```

variable	label
year	1987, 1988, or 1989
fcode	firm code number
employ	# employees at plant
sales	annual sales, $
avgsal	average employee salary
scrap	scrap rate (per 100 items)
rework	rework rate (per 100 items)
tothrs	total hours training
union	=1 if unionized
grant	= 1 if received grant
d89	= 1 if year = 1989
d88	= 1 if year = 1988
totrain	total employees trained
hrsemp	tothrs/totrain
lscrap	log(scrap)
lemploy	log(employ)
lsales	log(sales)
lrework	log(rework)
lhrsemp	log(1 + hrsemp)
lscrap_1	lagged lscrap; missing 1987
grant_1	lagged grant; assumed 0 in 1987
clscrap	lscrap - lscrap_1; year > 1987
cgrant	grant - grant_1
clemploy	lemploy - lemploy[_n-1]
clsales	lavgsal - lavgsal[_n-1]
lavgsal	log(avgsal)
clavgsal	lavgsal - lavgsal[_n-1]
cgrant_1	cgrant[_n-1]
chrsemp	hrsemp - hrsemp[_n-1]
clhrsemp	lhrsemp - lhrsemp[_n-1]

H. Holzer, R. Block, M. Cheatham, and J. Knott (1993), "Are Training Subsidies Effective? The Michigan Experience," Industrial and Labor Relations Review 46, 625-636. The authors kindly provided the data.

```
print(jtrain)
```

	year	fcode	employ	...	cgrant_1	chrsemp	clhrsemp
0	1987	410032.0	100.0	...	NaN	NaN	NaN
1	1988	410032.0	131.0	...	0.0	-8.946565	-1.165385
2	1989	410032.0	123.0	...	0.0	0.198597	0.047832
3	1987	410440.0	12.0	...	NaN	NaN	NaN
4	1988	410440.0	13.0	...	0.0	0.000000	0.000000
..	...	...	...	...	...	...	...
466	1988	419483.0	108.0	...	0.0	0.000000	0.000000
467	1989	419483.0	129.0	...	0.0	3.100775	1.411176
468	1987	419486.0	80.0	...	NaN	NaN	NaN
469	1988	419486.0	90.0	...	0.0	0.000000	0.000000
470	1989	419486.0	100.0	...	0.0	36.000000	3.610918

Define panel data using 1987 and 1988 only:

```
jtrain_87_88 = jtrain.loc[(jtrain['year'] == 1987)|(jtrain['year'] == 1988),:]
jtrain_87_88 = jtrain_87_88.set_index(['fcode', 'year'])
```

First-differencing the data and 'complete' the data by dropping the null observations:

```
# manual computation of deviations of entity means:
jtrain_87_88['lscrap_diff1'] = jtrain_87_88.sort_values(['fcode', 'year']) \
        .groupby('fcode')['lscrap'].diff()
jtrain_87_88['hrsemp_diff1'] = jtrain_87_88.sort_values(['fcode', 'year']) \
        .groupby('fcode')['hrsemp'].diff()
jtrain_87_88['grant_diff1'] = jtrain_87_88.sort_values(['fcode', 'year']) \
        .groupby('fcode')['grant'].diff()

# complete the data by dropping the null observations:
jtrain_87_88 = jtrain_87_88.dropna(subset = \
               ['lscrap_diff1', 'hrsemp_diff1', 'grant_diff1'])
```

Carry out the IV estimation:

```
import linearmodels.iv as iv
reg_iv = iv.IV2SLS.from_formula(
    formula='lscrap_diff1 ~ 1 + [hrsemp_diff1 ~ grant_diff1]',
    data=jtrain_87_88)
results_iv = reg_iv.fit(cov_type='unadjusted', debiased=True)

# print regression table:
import pandas as pd
table_iv = pd.DataFrame({'b': round(results_iv.params, 4),
                         'se': round(results_iv.std_errors, 4),
                         't': round(results_iv.tstats, 4),
                         'pval': round(results_iv.pvalues, 4)})
print(f'table_iv: \n{table_iv}\n')
```

```
table_iv:
                   b      se       t    pval
Intercept    -0.0327  0.1270 -0.2573  0.7982
hrsemp_diff1 -0.0142  0.0079 -1.7882  0.0808
```

Other Python IV Estimation Methods

As indicated in Section 9.3, the FE panel IV procedure is to suitably transform the model to control for μ_i and then apply IV to the transformed model.

Thus, the panel IV 1FE estimation can be done by first applying a within transformation to the data, and then perform IV estimation, or IV2SLS, IVGMM, etc.

However, many issues remain in Python implementations:

- What about panel IV 2FE estimation?
- What about panel IV 1RE estimation?
- What about panel IV 2RE estimation, or even the regular panel 2RE or panel 2CRE estimation?

A plausible answer to these questions to write your own Python scripts according to the econometric procedures introduced in the earlier sections or chapters. You are highly encouraged to pursue some of these issues through your project work.

9.4. Hausman-Taylor Estimator

In case of limited form of endogeneity (FE), it has been well discussed that the FE and FD methods cannot estimate the coefficients of time-invariant regressors because they are 'transformed away'.

Hausman-Taylor estimator is an IV estimator for the FE model that enables the estimation of the coefficients of time-invariant regressors.

Assumption: some specified regressors are uncorrelated with the FE.

Let $\mathbf{X}_{it} = [\mathbf{X}_{1it} \ \mathbf{X}_{2it}]$ and $\mathbf{W}_i = [\mathbf{W}_{1i} \ \mathbf{W}_{2i}]$ be, respectively, the time-varying and the time-invariant regressors, where

- the regressors with subscript 1 are uncorrelated with μ_i, and
- the regressors with subscript 2 are correlated with μ_i.

You may like to refer to the STATA application given later for Python implementation of Hausman-Taylor estimator.

Hausman-Taylor Estimator

The one-way FE model now has the form:

$$Y_{it} = \mathbf{X}'_{1it}\boldsymbol{\beta}_1 + \mathbf{X}'_{2it}\boldsymbol{\beta}_2 + \mathbf{W}'_{1i}\boldsymbol{\gamma}_1 + \mathbf{W}'_{2i}\gamma_2 + \mu_i + v_{it}, \qquad (9.8)$$

which is transformed by a random effects transformation:

$$\tilde{Y}_{it} = \tilde{\mathbf{X}}'_{1it}\boldsymbol{\beta}_1 + \tilde{\mathbf{X}}'_{2it}\boldsymbol{\beta}_2 + \tilde{\mathbf{W}}'_{1i}\boldsymbol{\gamma}_1 + \tilde{\mathbf{W}}'_{2i}\gamma_2 + \tilde{\mu}_i + \tilde{v}_{it}, \qquad (9.9)$$

where, e.g., $\tilde{Y}_{it} = Y_{it} - \hat{\theta}_i \bar{Y}_i$, and $\hat{\theta}_i$ follows a specific formula.

- The instruments for the regressors in (9.9):
  - $\tilde{\mathbf{X}}_{1it}$: $\mathbf{X}_{1it} - \bar{\mathbf{X}}_{1i}$
  - $\tilde{\mathbf{X}}_{2it}$: $\mathbf{X}_{2it} - \bar{\mathbf{X}}_{2i}$
  - $\tilde{\mathbf{W}}_{1i}$: $\mathbf{W}_{1i}$
  - $\tilde{\mathbf{W}}_{2i}$: $\bar{\mathbf{X}}_{1i}$.
- The method requires that $\dim(\bar{\mathbf{X}}_{1i}) \geq \dim(\tilde{\mathbf{W}}_{2i})$.

9.5. Stata IV Estimation

The **xtivreg** command (extending the **ivregress** for cross-section data) implements 2SLS regression on the transformed model, with options as those for **xtreg**: **fe**, **fd**, **re**, and **be**.

- Note in case of short panels, one can also imbed the time FE in $\mathbf{X}_{it}$.
- However, the **xtivreg** command has no **vce**(**robust**) option, but has the **vce**(**bootsrap**) option for cluster-robust standard errors.

We use again the Cornwell and Rupert Returns to Schooling Data, with 595 individuals and over 7 years.

The variables are listed on the right.

- **ed** might be correlated with μ_i, and FE model might be appropriate.
- **wks** might be correlated with v_{it}, and the panel IV estimation might be more valid.
- **ms** (marital status) might be a suitable IV for **wks**.

Stata Panel IV Estimation

EXP = work experience
WKS = weeks worked
OCC = occupation, 1 if blue collar,
IND = 1 if manufacturing industry
SOUTH = 1 if resides in south
SMSA = 1 if resides in a city (SMSA)
MS = 1 if married
FEM = 1 if female
UNION = 1 if wage set by union contract
ED = years of education
BLK = 1 if individual is black
LWAGE = log of wage

Stata Panel IV FE Estimation


```
. xtivreg lwage exp expsq (wks = ms), fe
Fixed-effects (within) IV regression            Number of obs     =      4,165
Group variable: id                              Number of groups  =        595
R-sq:                                           Obs per group:
     within  =      .                                           min =          7
     between = 0.0172        Instrumented:   wks                avg =        7.0
     overall = 0.0284        Instruments:    exp expsq ms       max =          7
                                                Wald chi2(3)      =  700142.43
corr(u_i, Xb)  = -0.8499                        Prob > chi2       =     0.0000
------------------------------------------------------------------------------
       lwage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         wks |  -.1149742   .2316926    -0.50   0.620    -.5690832    .3391349
         exp |   .1408101   .0547014     2.57   0.010     .0335974    .2480228
       expsq |  -.0011207   .0014052    -0.80   0.425    -.0038748    .0016334
       _cons |    9.83932   10.48955     0.94   0.348    -10.71983    30.39847
-------------+----------------------------------------------------------------
     sigma_u |  1.0980369
     sigma_e |  .51515503
         rho |  .81959748   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F  test that all u_i=0:     F(594,3567) =     4.62         Prob > F    = 0.0000
------------------------------------------------------------------------------
```

Stata Panel IV FE Estimation


```
. xtreg lwage exp expsq wks, fe vce(cluster id)

Fixed-effects (within) regression               Number of obs      =       4,165
Group variable: id                              Number of groups   =         595

R-sq:                                           Obs per group:
     within  = 0.6566                                         min =           7
     between = 0.0276                                         avg =         7.0
     overall = 0.0476                                         max =           7

                                                F(3,594)           =     1059.72
corr(u_i, Xb)  = -0.9107                        Prob > F           =      0.0000

                                        (Std. Err. adjusted for 595 clusters in id)
------------------------------------------------------------------------------
             |               Robust
       lwage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         exp |   .1137879   .0040289    28.24   0.000     .1058753    .1217004
       expsq |  -.0004244   .0000822    -5.16   0.000    -.0005858   -.0002629
         wks |   .0008359   .0008697     0.96   0.337    -.0008721    .0025439
       _cons |   4.596396   .0600887    76.49   0.000     4.478384    4.714408
-------------+----------------------------------------------------------------
     sigma_u |  1.0362039
     sigma_e |  .15220316
         rho |  .97888036   (fraction of variance due to u_i)
------------------------------------------------------------------------------
```

Compare the above panel IV FE estimation with panel FE estimation:

Stata Panel IV FE Estimation

- The panel IV estimates imply that, surprisingly, log-wages decrease by 11.5% for each additional week worked, though the coefficient is statistically insignificant.
- Log-wages increases with experience until a peak at 64 years.
- Comparing the panel IV results with those using **xtreg**, **fe**:
  - the coefficient of the endogenous regressor **wks** has changed sign, and is many times larger in absolute value;
  - The coefficient of the exogenous regressors **exp** and **expsq** are less affected, but the corresponding standard errors are
  - For these data, the IV standard errors are more than ten times larger.
  - Because the instrument **ms** is not very correlated with **wks**, the panel IV regression leads to a huge loss in estimator efficiency.
- In case of weak instruments, alternative methods for better handling of endogeneity in regressors are needed (a topic beyond our scope).

STATA Hausman-Taylor Estimator

Consider the Returns to Schooling Data, analyzed by Cornwell and Rupert (1988) using Hausman-Taylor estimator.

- The goal is to obtain a consistent estimate of the coefficient of **ed** because there is great interest in the impact of education on wage.
- Education is clearly endogenous. It is assumed that it is correlated only with the individual-specific components of the errors.

Cornwell and Rupert (1988) assumed that

- $\mathbf{X}_{1it}$ = [**occ**, **south**, **smsa**, **ind**];
- $\mathbf{X}_{2it}$ = [**exp**, **expsq**, **wks**, **ms**, **union**].
- $\mathbf{W}_{1i}$ = [**fem**, **blk**]; $\mathbf{W}_{2i}$ = [**ed**].

STATA Hausman-Taylor Estimator

```
. xthtaylor lwage occ south smsa ind exp expsq wks ms union fem blk ed,
> endog(exp expsq wks ms union ed)
Hausman-Taylor estimation                    Number of obs      =     4,165
Group variable: id                           Number of groups   =       595

                                             Obs per group:
                                                          min =         7
                                                          avg =         7
                                                          max =         7
Random effects u_i ~ i.i.d.                  Wald chi2(12)      =   6891.87
                                             Prob > chi2        =    0.0000
------------------------------------------------------------------------------
       lwage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
TVexogenous  |
         occ |  -.0207047   .0137809    -1.50   0.133    -.0477149    .0063055
       south |   .0074398    .031955     0.23   0.816    -.0551908    .0700705
        smsa |  -.0418334   .0189581    -2.21   0.027    -.0789906   -.0046761
         ind |   .0136039   .0152374     0.89   0.372    -.0162608    .0434686
```

STATA Hausman-Taylor Estimator

Cont'd from last page

TVendogenous						
exp	.1131328	.002471	45.79	0.000	.1082898	.1179758
expsq	-.0004189	.0000546	-7.67	0.000	-.0005259	-.0003119
wks	.0008374	.0005997	1.40	0.163	-.0003381	.0020129
ms	-.0298508	.01898	-1.57	0.116	-.0670508	.0073493
union	.0327714	.0149084	2.20	0.028	.0035514	.0619914
TIexogenous						
fem	-.1309236	.126659	-1.03	0.301	-.3791707	.1173234
blk	-.2857479	.1557019	-1.84	0.066	-.5909179	.0194221
TIendogenous						
ed	.137944	.0212485	6.49	0.000	.0962977	.1795902
_cons	2.912726	.2836522	10.27	0.000	2.356778	3.468674
sigma_u	.94180304					
sigma_e	.15180273					
rho	.97467788	(fraction of variance due to u_i)				

Note: TV refers to time varying; TI refers to time invariant.

STATA Hausman-Taylor Estimator

Compared with the RE estimation given next:

- the coefficient of **ed** and standard errors have both increased,
- **ed** remains highly significant.

```
. xtreg lwage occ south smsa ind exp expsq wks ms union fem blk ed, re
vce(robust)

Random-effects GLS regression                   Number of obs     =      4,165
Group variable: id                              Number of groups  =        595

R-sq:                                           Obs per group:
     within  = 0.6124                                         min =          7
     between = 0.2539                                         avg =        7.0
     overall = 0.2512                                         max =          7

                                                Wald chi2(12)     =    1528.82
corr(u_i, X)   = 0 (assumed)                    Prob > chi2       =     0.0000
```

Continued on next page …

STATA Hausman-Taylor Estimator

(Std. Err. adjusted for 595 clusters in id)

lwage	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
occ	-.0500664	.0207608	-2.41	0.016	-.0907567	-.009376
south	-.0166176	.0460333	-0.36	0.718	-.1068411	.073606
smsa	-.0138231	.0297818	-0.46	0.643	-.0721944	.0445482
ind	.0037441	.0232112	0.16	0.872	-.0417489	.0492372
exp	.0820544	.0040168	20.43	0.000	.0741816	.0899272
expsq	-.0008084	.0000895	-9.03	0.000	-.000984	-.0006329
wks	.0010347	.000941	1.10	0.272	-.0008097	.0028791
ms	-.0746283	.0274262	-2.72	0.007	-.1283826	-.020874
union	.0632232	.0249276	2.54	0.011	.0143661	.1120803
fem	-.3392101	.0630206	-5.38	0.000	-.4627283	-.2156919
blk	-.2102803	.0826686	-2.54	0.011	-.3723078	-.0482528
ed	.0996585	.0080237	12.42	0.000	.0839324	.1153847
_cons	4.26367	.1359373	31.36	0.000	3.997238	4.530103
sigma_u	.26265814					
sigma_e	.15199444					
rho	.74913774	(fraction of variance due to u_i)				