1 twosls: Two Stage Least Squares

`twosls` provides consistent estimates for linear regression models with some explanatory variable (the instrumental variable) correlated with the error term. In this situation, ordinary least squares fails to provide consistent estimates. The name two-stage least squares stems from the two regressions in the estimation procedure. In stage one, an ordinary least squares prediction of the instrumental variable is obtained from regressing it on the instrument variables. In stage two, the coefficients of interest are estimated using ordinary least square after substituting the instrumental variable by its predictions from stage one.

1.0.1 Syntax

```
> fml <- list ("mu"  = Y ~ X + Z,
               "inst" = Z ~ W + X)
> z.out <- zelig(formula = fml, model = "twosls", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

1.0.2 Inputs

`twosls` regression take the following inputs:

- `formula`:a list of the main equation and instrumental variable equation. The first object in the list `mu` corresponds to the regression model needs to be estimated. The second list object `inst` specifies the regression model for the instrumental variable `Z`. For example:

  ```
  >  fml <- list ("mu"  = Y ~ X + Z,
  +                "inst" = Z ~ W + X)
  ```

  - `Y`: the dependent variable of interest.
  - `Z`: the instrumental variable.
  - `W`: exogenous instrument variables.

1.0.3 Additional Inputs

`twosls` takes the following additional inputs for model specifications:

- `TX`: an optional matrix to transform the regressor matrix and, hence, also the coefficient vector (see 1.0.4). Default is `NULL`.
- `rcovformula`: formula to calculate the estimated residual covariance matrix (see 1.0.4). Default is equal to 1.
- `probdfsys`: use the degrees of freedom of the whole system (in place of the degrees of freedom of the single equation to calculate probability values for the t-test of individual parameters.

- `single.eq.sigma`: use different σ^2 for each single equation to calculate the covariance matrix and the standard errors of the coefficients.

- `solvetol`: tolerance level for detecting linear dependencies when inverting a matrix or calculating a determinant. Default is `solvetol`=.Machine$double.eps.

- `saveMemory`: logical. Save memory by omitting some calculation that are not crucial for the basic estimate (e.g McElroy's R^2).

1.0.4 Details

- `TX`: The matrix `TX` transforms the regressor matrix (X) by $X* = X \times TX$. Thus, the vector of coefficients is now $b = TX \times b*$ where b is the original(stacked) vector of all coefficients and $b*$ is the new coefficient vector that is estimated instead. Thus, the elements of vector b and $b_i = \sum_j TX_{ij} \times b_j*$. The TX matrix can be used to change the order of the coefficients and also to restrict coefficients (if TX has less columns than it has rows).

- `rcovformula`: The formula to calculate the estimated covariance matrix of the residuals($\hat{\Sigma}$)can be one of the following (see Judge et al., 1955, p.469): if `rcovformula`= 0:

$$\hat{\sigma_{ij}} = \frac{\hat{e_i}\prime\hat{e_j}}{T}$$

if `rcovformula`= 1 or `rcovformula`='geomean':

$$\hat{\sigma_{ij}} = \frac{\hat{e_i}\prime\hat{e_j}}{\sqrt{(T-k_i)\times(T-k_j)}}$$

if `rcovformula`= 2 or `rcovformula`='Theil':

$$\hat{\sigma_{ij}} = \frac{\hat{e_i}\prime\hat{e_j}}{T-k_i-k_j+tr[X_i(X_i\prime X_i)^{-1}X_i\prime X_j(X_j\prime X_j)^{-1}X_j\prime]}$$

if `rcovformula`= 3 or `rcovformula`='max':

$$\hat{\sigma_{ij}} = \frac{\hat{e_i}\prime\hat{e_j}}{T-max(k_i,k_j)}$$

If $i = j$, formula 1, 2, and 3 are equal. All these three formulas yield unbiased estimators for the diagonal elements of the residual covariance matrix. If $ineqj$, only formula 2 yields an unbiased estimator for the residual covariance matrix, but it is not necessarily positive semidefinit. Thus, it is doubtful whether formula 2 is really superior to formula 1

1.0.5 Examples

Attaching the example dataset:

```
> data(klein)
```

Formula:

```
> formula <- list(mu1=C~Wtot + P + P1,
+                mu2=I~P + P1 + K1,
+                mu3=Wp~ X + X1 + Tm,
+                inst= ~ P1 + K1 + X1 + Tm + Wg + G)
```

Estimating the model using `twosls`:

```
> z.out<-zelig(formula=formula, model="twosls",data=klein)
```

```
The following object(s) are masked from 'package:base':

    T


 How to cite this model in Zelig:
  Ferdinand Alimadhi, Ying Lu, and Elena Villalon. 2012.
  "twosls"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig
```

```
> summary(z.out)
```

```
systemfit results
method: 2SLS

        N DF    SSR  detRCov   OLS-R2 McElroy-R2
system 63 51 60.531 0.495617 0.969562   0.993214

     N DF     SSR      MSE     RMSE       R2   Adj R2
mu1 21 17 19.7649 1.162638 1.078257 0.979005 0.975301
mu2 21 17 30.6494 1.802905 1.342723 0.878533 0.857098
mu3 21 17 10.1167 0.595103 0.771429 0.987273 0.985027

The covariance matrix of the residuals
          mu1      mu2       mu3
mu1  1.162638 0.451038 -0.468245
mu2  0.451038 1.802905  0.303630
mu3 -0.468245 0.303630  0.595103

The correlations of the residuals
          mu1      mu2       mu3
```

```
mu1  1.000000 0.311533 -0.562930
mu2  0.311533 1.000000  0.293131
mu3 -0.562930 0.293131  1.000000


2SLS estimates for 'mu1' (equation 1)
Model Formula: C ~ Wtot + P + P1
Instruments: ~P1 + K1 + X1 + Tm + Wg + G

              Estimate Std. Error  t value   Pr(>|t|)
(Intercept) 16.3990728  1.4088073 11.64039 1.6004e-09 ***
Wtot         0.8082108  0.0425534 18.99286 6.9500e-13 ***
P            0.0720813  0.1439879  0.50061    0.62307
P1           0.1742361  0.1260083  1.38274    0.18464
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 1.078257 on 17 degrees of freedom
Number of observations: 21 Degrees of Freedom: 17
SSR: 19.764853 MSE: 1.162638 Root MSE: 1.078257
Multiple R-Squared: 0.979005 Adjusted R-Squared: 0.975301


2SLS estimates for 'mu2' (equation 2)
Model Formula: I ~ P + P1 + K1
Instruments: ~P1 + K1 + X1 + Tm + Wg + G

              Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 20.9498400 10.3377708  2.02653 0.0586938 .
P            0.1284295  0.2712099  0.47354 0.6418500
P1           0.6346591  0.2448306  2.59224 0.0189823 *
K1          -0.1608303  0.0487085 -3.30190 0.0042128 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 1.342723 on 17 degrees of freedom
Number of observations: 21 Degrees of Freedom: 17
SSR: 30.649387 MSE: 1.802905 Root MSE: 1.342723
Multiple R-Squared: 0.878533 Adjusted R-Squared: 0.857098


2SLS estimates for 'mu3' (equation 3)
Model Formula: Wp ~ X + X1 + Tm
Instruments: ~P1 + K1 + X1 + Tm + Wg + G

             Estimate Std. Error  t value   Pr(>|t|)
```

```
(Intercept) 1.5714723  1.2831206  1.22473 0.23737862
X           0.4253396  0.0402022 10.58000 6.7336e-09 ***
X1          0.1594488  0.0437152  3.64744 0.00199274 **
Tm          0.1336876  0.0325964  4.10130 0.00074462 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 0.771429 on 17 degrees of freedom
Number of observations: 21 Degrees of Freedom: 17
SSR: 10.116746 MSE: 0.595103 Root MSE: 0.771429
Multiple R-Squared: 0.987273 Adjusted R-Squared: 0.985027
```

Set explanatory variables to their default (mean/mode) values

```
>  x.out <- setx(z.out)
```

Simulate draws from the posterior distribution:

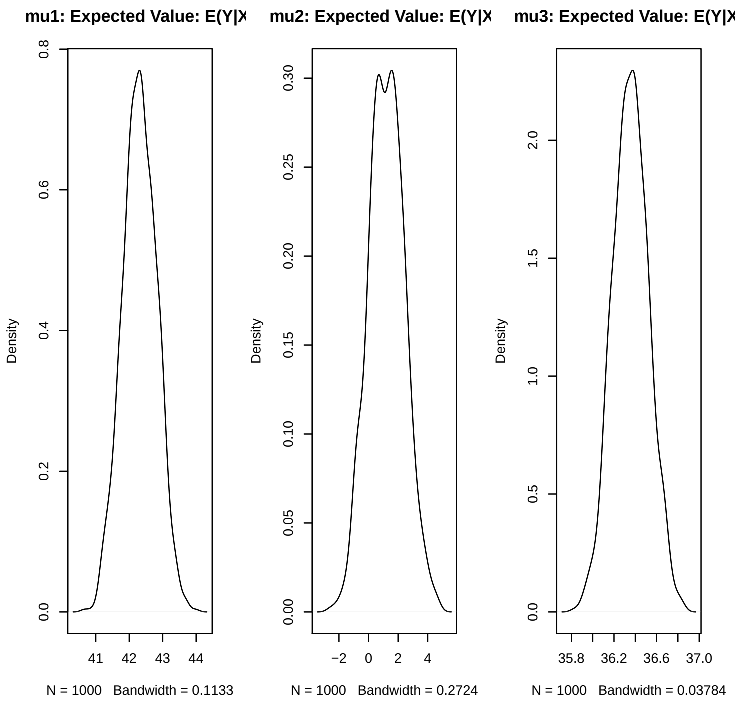
```
> s.out <-sim(z.out,x=x.out)
>  summary(s.out)

Model:  twosls
Number of simulations:  1000

Values of X
     (Intercept) Wtot    P   P1    K1        X       X1 Tm
[1,]           1 28.2 12.4 12.7 182.8 60.05714 57.98571  0

Expected Value: E(Y|X)
      mean    sd    50%   2.5%  97.5%
mu1 42.310 0.501 42.314 41.306 43.275
mu2  1.212 1.205  1.234 -1.050  3.659
mu3 36.357 0.167 36.357 36.034 36.687
```

Plot the quantities of interest


mu1: Expected Value: E(Y|X
Density
0.0
0.2
0.4
0.6
0.8
41
42
43
44
N = 1000 Bandwidth = 0.1133
mu2: Expected Value: E(Y|X
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−2
0
2
4
N = 1000 Bandwidth = 0.2724
mu3: Expected Value: E(Y|X
Density
0.0
0.5
1.0
1.5
2.0
35.8
36.2
36.6
37.0
N = 1000 Bandwidth = 0.03784

1.0.6 Model

Let's consider the following regression model,

$$Y_i = X_i\beta + Z_i\gamma + \epsilon_i, \quad i = 1, \ldots, N$$

where Y_i is the dependent variable, $X_i = (X_{1i}, \ldots, X_{Ni})$ is the vector of explanatory variables, β is the vector of coefficients of the explanatory variables X_i, Z_i is the problematic explanatory variable, and γ is the coefficient of Z_i. In the equation, there is a direct dependence of Z_i on the structural disturbances of ϵ.

- The *stochastic component* is given by

$$\epsilon_i \quad \sim \quad \mathcal{N}(0, \sigma^2), \quad \text{and} \quad \text{cov}(Z_i, \epsilon_i) \neq 0,$$

- The *systematic component* is given by:

$$\mu_i = E(Y_i) = X_i\beta + Z_i\gamma,$$

To correct the problem caused by the correlation of Z_i and ϵ, two stage least squares utilizes two steps:

- *Stage 1*: A new instrumental variable $\hat{Z}$ is created for Z_i which is the ordinary least squares predictions from regressing Z_i on a set of exogenous instruments W and X.

$$\widehat{Z_i} = \widetilde{W}_i[(\widetilde{W}^\top \widetilde{W})^{-1} \widetilde{W}^\top Z]$$

  where $\widetilde{W} = (W, X)$

- *Stage 2*: Substitute for $\hat{Z}_i$ for Z_i in the original equation, estimate β and γ by ordinary least squares regression of Y on X and $\hat{Z}$ as in the following equation.

$$Y_i = X_i\beta + \widehat{Z_i}\gamma + \epsilon_i, \quad \text{for} \quad i = 1, \ldots, N$$

1.0.7 See Also

For information about three stage least square regression, see Section **??** and `help(3sls)`. For information about seemingly unrelated regression, see Section **??** and `help(sur)`.

1.0.8 Quantities of Interest

1.0.9 Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run:

```
z.out <- zelig(formula=fml, model = "twosls", data)
```

then you may examine the available information in `z.out` by using `names(z.out)`, see the draws from the posterior distribution of the `coefficients` by using `z.out$coefficients`, and view a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below:

- `h`: matrix of all (diagonally stacked) instrumental variables.
- `single.eq.sigma`: different σ^2s for each single equation?.
- `zelig.data`: the input data frame if `save.data = TRUE`.
- `method`: Estimation method.
- `g`: number of equations.
- `n`: total number of observations.
- `k`: total number of coefficients.
- `ki`: total number of linear independent coefficients.
- `df`: degrees of freedom of the whole system.
- `iter`: number of iteration steps.
- `b`: vector of all estimated coefficients.
- `t`: t values for b.
- `se`: estimated standard errors of b.
- `bt`: coefficient vector transformed by TX.
- `p`: p values for b.
- `bcov`: estimated covariance matrix of b.
- `btcov`: covariance matrix of bt.
- `rcov`: estimated residual covariance matrix.
- `drcov`: determinant of `rcov`.
- `rcor`: estimated residual correlation matrix.
- `olsr2`: system OLS R-squared value.
- `y`: vector of all (stacked) endogenous variables.
- `x`: matrix of all (diagonally stacked) regressors.

- `data`: data frame of the whole system (including instruments).
- `TX`: matrix used to transform the regressor matrix.
- `rcovformula`: formula to calculate the estimated residual covariance matrix.
- `probdfsys`: system degrees of freedom to calculate probability values?.
- `solvetol`: tolerance level when inverting a matrix or calculating a determinant.
- `eq`: a list that contains the results that belong to the individual equations.
- `eqnlabel*`: the equation label of the ith equation (from the labels list).
- `formula*`: model formula of the ith equation.
- `n*`: number of observations of the ith equation.
- `k*`: number of coefficients/regressors in the ith equation (including the constant).
- `ki*`: number of linear independent coefficients in the ith equation (including the constant differs from k only if there are restrictions that are not cross equation).
- `df*`: degrees of freedom of the ith equation.
- `b*`: estimated coefficients of the ith equation.
- `se*`: estimated standard errors of b of the ith equation.
- `t*`: t values for b of the ith equation.
- `p*`: p values for b of the ith equation.
- `covb*`: estimated covariance matrix of b of the ith equation.
- `y*`: vector of endogenous variable (response values) of the ith equation.
- `x*`: matrix of regressors (model matrix) of the ith equation.
- `data*`: data frame (including instruments) of the ith equation.
- `fitted*`: vector of fitted values of the ith equation.
- `residuals*`: vector of residuals of the ith equaiton.
- `ssr*`: sum of squared residuals of the ith equation.
- `mse*`: estimated variance of the residuals (mean of squared errors) of the ith equation.

- `s2*`: estimated variance of the residents($\widehat{sigma}^2$) of the ith equation.
- `rmse*`: estimated standard error of the reiduals (square root of mse) of the ith equation.
- `s*`: estimated standard error of the residuals ($\hat{\sigma}$) of the ith equation.
- `r2*`: R-squared (coefficient of determination).
- `adjr2*`: adjusted R-squared value.
- `inst*`: instruments of the ith equation.
- `h*`: matrix of instrumental variables of the ith equation.

How to Cite the *twosls* Zelig model

Ferdinand Alimadhi, Ying Lu, and Elena Villalon. 2007. "twosls: Two Stage Least Squares," in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software," `http://gking.harvard.edu/zelig`.

To cite Zelig as a whole, please reference these two sources:

Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The twosls function is adapted from the `systemfit` library by Jeff Hamann and Arne Henningsen [?].

References