

# **Lagrangetest: Lagrange-Multiplier Test after Constrained Maximum-Likelihood Estimation using Stata**

Harald Tauchmann

Friedrich-Alexander-Universität Erlangen-Nürnberg

June 16, 2023

2023 German Stata Conference, Berlin

# Outline

- 1 Motivation
- 2 The Lagrange Multiplier Test
- 3 Implementation in Stata
- 4 The `lrgctest` Command
- 5 `lrgctest` and `boottest`
- 6 Empirical Application
- 7 Conclusions

# Motivation

Teach our students that after maximum-likelihood (ML) estimation **three** approaches to testing restrictions are available (if we teach 'old-fashioned' econometrics)

1. Wald test
2. Likelihood-ratio (LR) test
3. **Lagrange-multiplier** (LM) test (also known as **score** test)

## Motivation (cont.)

- ▶ Official Stata includes commands implementing the
  1. Wald test (`test`)
  2. Likelihood-ratio test (`lrtest`)
- ▶ Yet no general command implementing the LM test
  - » LM tests available for specific settings (`xttest0` after `xtreg, re`; `estat scoretests` after `sem`)
- ▶ Implementing LM test 'by hand' somewhat cumbersome
- ▶ Having a (user written) Stata command (with straightforward syntax) available beneficial

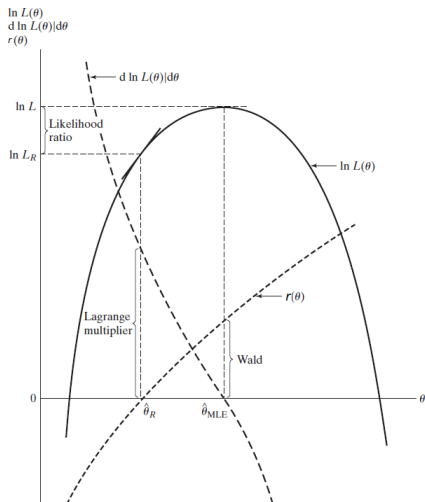
## User Written Commands to implement LM test?

- ▶ Couldn't find one
- ▶ Decided to write one myself
  - » `lgrgtest` (earlier version `lmtest` available from SSC)
- ▶ Very recently, got a message from David Rodman via `statalist`
  - » His `boottest` (`scoretest`) command not only implement bootstrap based inference (as I thought) but also the classical LM test
- ▶ Syntax of `lgrgtest` and `scoretest` different, yet they do the same after many (not all) Stata (ML) estimation commands

# The Lagrange Multiplier Test

- ▶ Independently introduced by Rao (1948) and Silvey (1959)
- ▶ Tests constrained ( $\mathbf{r}(\boldsymbol{\theta}) = \mathbf{0}$ ) version ( $H_0$ ) of a model against unconstrained model
- ▶ Exploits that gradient of log-likelihood function  $\mathcal{L}(\boldsymbol{\theta})$  at constrained max. deviates from zero if restrictions bind
  - » Binding restrictions argue against constrained null model
- ▶ Test-statistic (score version):  $LMS = \mathbf{g}(\tilde{\boldsymbol{\theta}})' \mathbf{I}(\tilde{\boldsymbol{\theta}})^{-1} \mathbf{g}(\tilde{\boldsymbol{\theta}})$ 
  - »  $\tilde{\boldsymbol{\theta}}$ : parameter estimates that maximize  $\mathcal{L}(\boldsymbol{\theta})$  respecting the constraints
  - »  $\mathbf{g}(\tilde{\boldsymbol{\theta}})$ : gradient/score vector  $d\mathcal{L}(\boldsymbol{\theta})/d\boldsymbol{\theta}$ , evaluated at  $\tilde{\boldsymbol{\theta}}$
  - »  $\mathbf{I}(\tilde{\boldsymbol{\theta}})$ : information matrix, evaluated at  $\tilde{\boldsymbol{\theta}}$
- ▶  $LMS$  asymptotically  $\chi^2$ -distributed with  $q$  dofs
  - »  $q$ : number of constraints

## The Lagrange Multiplier Test (cont.)

LM, LR, and Wald Test in ML estimation context, **Source:** Greene (2012)

# Implementation in Stata through `lgrgtest`

- ▶ (Jointly) tests restrictions imposed through option **`constraints()`** on most recently estimated model
  - » Confines `lgrgtest` to testing linear constraints
  - » Requires **`constraints()`** to be allowed in upstream estimation command
  - » After `sem`, and `gsem`, also restrictions specified using path notion (symbol `@`) tested
- ▶ Degrees of freedoms identified from **`e(Cns)`**
  - » Rank of `e(Cns)`
  - » Difference in ranks, if also 'unconstraint' model saves `e(Cns)` (e.g. to deal with collinearity)
- ▶ `lgrgtest` implements LM test in Stata based on score version ( $\rightarrow$  estimates of  $\mathbf{g}(\tilde{\theta})$  and  $\mathbf{I}(\tilde{\theta})$  required)



## Estimating the Score Vector

- ▶ Score vector needs to be saved in **e(gradient)**
  - » Confines `lgrgtest` to be used after ML estimators
  - » `lgrgtest` runs also after `cnsreg` ( $\rightarrow$  interprets OLS as ML estimator [iid. normals errors])
- ▶ After estimation with option `constraints()`,  **$\mathbf{g}(\tilde{\theta})$  not saved in e(gradient)**
- ▶ `lgrgtest` **re-executes** estimation command
  - (i) Without option `constraints()`
  - (ii) Using  $\tilde{\theta}$  as starting values
  - (iii) Disallowing maximization algorithm to iterate
- ▶ Requires options `from()` and `iterate()`
  - » For some command (fmm) more options need to be specified in re-execution to prevent re-estimation
- ▶ Makes  **$\mathbf{g}(\tilde{\theta})$  be saved in e(gradient)**

# Estimating the Inverse Information Matrix

- ▶  $e(\mathbf{V})$  from internal re-run used as estimate of  $\mathbf{I}(\tilde{\theta})^{-1}$
- ▶ Only valid if **vcetype** either `oim` or `opg`
  - » If `vcetype` is neither `oim` nor `opg`, `lgrgtest` denies carrying out the test
  - » `lgrgtest` option `forcevce` makes command use `e(V_modelbased)` instead of `e(V)` ( $\rightarrow$  one needs to know what one is doing)

## Further Requirements for upstream Est. Command

- ▶ Re-executing estimation command – with adjusted options – requires interpreting command-line entry
  - » `lgrgtest` only runs after commands with ‘standard’ Stata syntax
    - > Also runs after some commands with more complex syntax (`sem`, `gsem`, `nlogit`, `fmm`)
  - » Command-line entry needs to be saved in `e(cmdline)`
  - » Command name needs to be saved in `e(cmd)` (or in `e(cmd2)`)

# Stata Cmds allowing for `lgrgtest` in Postestimation

[CM] `cmclogit`, `cmmixlogit`, `cmmprobit`, `cmroprobit`, `cmxtmixlogit`,  
`nlogit`; [DSGE] `dsge`; [ERM] `eintreg`, `xteinreg`, `eprobit`, `xteprobit`,  
`eregress`, `xteregress`; [FMM] `fmm`; [ME] `mecloglog`, `meglm`, `meintreg`,  
`melogit`, `menbreg`, `meologit`, `meoprobit`, `mepoisson`, `meprobit`, `mestreg`,  
`metobit`; [R] `betareg`, `binreg`, `clogit`, `cloglog`, `cnsreg`, `cpoisson`,  
`fracreg`, `frontier`, `glm`, `heckman`, `heckprobit`, `heckpoisson`,  
`heckprobit`, `hetoprobit`, `hetprobit`, `hetregress`, `intreg`, `ivprobit`,  
`ivtobit`, `logistic`, `logit`, `mlexp`, `mlogit`, `mprobit`, `nbreg`, `gnbreg`, `ologit`,  
`oprobit`, `poisson`, `probit`, `scobit`, `slogit`, `tnbreg`, `tobit`, `tpoisson`,  
`truncreg`, `zinb`, `ziologit`, `zioprobit`, `zip`; [SEM] `sem`, `gsem`; [ST] `stcrreg`,  
`stintreg`, `streg`; [TE] `etpoisson`, `etregress`; [TS] `arch`, `arfima`, `arima`,  
`dfactor`, `mgarch ccc`, `mgarch dcc`, `mgarch dveh`, `mgarch vcc`, `sspace`,  
`ucm`; [XT] `xtcloglog`, `xtfrontier`, `xtheckman`, `xtintreg`, `xtlogit`,  
`xtmlogit`, `xtnbreg`, `xtologit`, `xtoprobit`, `xtpoisson`, `xtprobit`, `xttobit`

# Syntax of `lgrgtest`

```
lgrgtest [, notest nocnsreport noomitted df(#)  
forcevce]
```

- ▶ Syntax does not specify restrictions to be tested
- ▶ Done in the preceding estimation syntax via specifying the option `constraints()`

# Options for `lgrgtest`

- ▶ `notest` prevents `lgrgtest` from displaying any output on the screen
- ▶ `nocnsreport` prevents `lgrgtest` from displaying the imposed constraints
- ▶ `noomitted` makes `lgrgtest` not consider omitted variables as exclusion restrictions to be tested; specifying `noomitted` will not affect number of test degrees-of-freedom but only which restrictions are displayed
- ▶ `df()` makes `lgrgtest` use a user-specified number of degrees-of-freedom; specifying `df()` will rarely be required
- ▶ `forcevce` makes `lgrgtest` perform the LM test even if `vcetype` is neither `oim` nor `opg` (not `ols` after `cnsreg`); with option `forcevce`, `lgrgtest` issues a warning and uses `e(V_modelbased)` as estimate of the inverse information matrix

# Stored Results for `lgrgtest`

`lgrgtest` stores in `r()`:

► Scalars

- » `r(p)`:  $p$ -value
- » `r(chi2)`: LM test statistic (chi-squared)
- » `r(df)`: test constraints degrees of freedom
- » `r(rank)`: rank of `e(Cns)` adjusted for the number automatically imposed constraints (only saved if option `df()` is specified)

► Macros

- » `r(modelbased)`: `modelbased` if `e(V_modelbased)` used as estimate of the inverse information matrix

# lgrgtest and boottest

- ▶ boottest way more powerful than lgrgtest
- ▶ scoretest (boottest package) deals with special case in which no robustification through bootstrapping is required
- ▶ After many popular commands (logit, probit, ...) lgrgtest and scoretest yield identical results
- ▶ After some commands scoretest failed but not lgrgtest
  - » David Roodman published new version of scoretest that have fixed some issues
  - » See discussion on stataлист



## lgrgtest and boottest (cont.)

- ▶ After some commands scoretest and lgrgtest yield different results
  - » Needs still to be checked (from my side)
  - » See discussion on statalist
- ▶ Would probably not have written lgrgtest if I had been aware of scoretest
- ▶ lgrgtest might still be of some value since classical LM-test functionality of boottest apparently not well known among Stata users

## Egger et al. (2011) [AEJ: EP] in a Nutshell

- ▶ Research question: Do preferential trade agreements (PTA) positively affect bilateral trade flows? (“The Trade Effects of Endogenous Preferential Trade Agreements”)
- ▶ Country-level, cross-sectional data (year 2005, 15 750 country dyads)
- ▶ Existence of PTA possibly endogenous
- ▶ When considering extensive margin of trade [ $\mathcal{I}_{ij} = 1$  (exports from  $i$  to  $j > 0$ )], endogeneity addressed through estimating recursive bivariate probit model
- ▶ Exclusion restrictions (instruments for  $PTA_{ij}$ ):
  1. Past colonizer-colony relationship ( $COLONY_{ij}$ )
  2. Common past colonizer ( $COMCOL_{ij}$ )
  3. Common history as one joint country ( $SMCTRY_{ij}$ )
- ▶ Non-linearity of probit model allows for conventionally testing validity of exclusion restrictions

# Testing Exclusion Restrictions

- ▶ Egger et al. (2011) use test (Wald test)
  - » Based on estimating unconstrained model (no exclusion restrictions)
  - » Yet, discussion focusses on constrained model (exclusion restrictions imposed)
- ▶ Basing test on estimating constrained model has intuitive appeal
- ▶ LM test based on restricted model (→ implemented by `lgrgtest`)

## Testing Exclusion Restrictions (cont.)

- ▶ Wald test originally used in Egger et al. (2011):

```
. quietly biprobit (pta = $z $instr const _x_*, nocons) (i =  
> pta $z $instr const _x_*, nocons)  
. test [i]colony [i]curcol [i]smctry  
( 1) [i]colony = 0  
( 2) [i]curcol = 0  
( 3) [i]smctry = 0  
      chi2( 3) = 3.78  
      Prob > chi2 = 0.2864
```

- ▶ Exclusion restrictions not rejected

## Testing Exclusion Restrictions (cont.)

► LM test using `lrgttest`:

```
. constraint 1 [i]colony = 0
. constraint 2 [i]curcol = 0
. constraint 3 [i]smctry = 0
. constraint 4 [i]comcol = 0
. quietly biprobit (pta = $z $instr const _x_*, nocons) (i =
> pta $z $instr const _x_*, nocons), constraints(1 2 3)
. lrgttest
```

LM test of constraints(1 2 3)

( 1) [i]colony = 0

( 2) [i]curcol = 0

( 3) [i]smctry = 0

LM chi2( 3) = 3.79

Prob > chi2 = 0.2856

## Testing Exclusion Restrictions (cont.)

- ▶ Virtually identical result from test and `lrgttest`
  - » Asymptotic equivalence of LM test and Wald test
  - » Rather large sample
- ▶ Interesting secondary aspect:
  - » Test exclusion restrictions not those that are actually imposed (in preferred model)
  - »  $COMCOL_{ij}$ , not  $CURCOL_{ij}$  (post 1945 colonizer-colony relationship), used as instrument for  $PTA_{ij}$
  - » Actually imposed exclusion restrictions clear rejected (irrespective of whether LM test or Wald test is used)

## Testing Exclusion Restrictions (cont.)

- ▶ LM test using `lrgptest` of actually imposed restrictions:

```
. quietly biprobit (pta = $z $instr const _x_*, nocons) (i =  
> pta $z $instr const _x_*, nocons), constraints(1 3 4)
```

```
. lrgptest
```

```
LM test of constraints(1 3 4)
```

```
( 1) [i]colony = 0
```

```
( 3) [i]smctry = 0
```

```
( 4) [i]comcol = 0
```

```
LM chi2( 3) = 171.21
```

```
Prob > chi2 = 0.0000
```

# Conclusions

- ▶ `lgrgtest` provides convenient way to employ classic Lagrange-multiplier test in Stata postestimation
- ▶ Complements real Stata commands `test` and `lrtest`
- ▶ Alternative to `scoretest`
- ▶ Major limitations (`lgrgtest` shares with `scoretest`):
  - » Confined to testing linear restrictions
  - » Cannot be used after `ml maximize`



## Multiplier Version of LM Test

Multiplier version of LM-test statistic:

$$LMS = \tilde{\lambda}' \mathbf{R}(\tilde{\theta})' \mathbf{I}(\tilde{\theta})^{-1} \mathbf{R}(\tilde{\theta}) \tilde{\lambda}$$

with Jacobian  $\mathbf{R}(\tilde{\theta})$ , and Lagrange multipliers  $\tilde{\lambda}$ . From first order condition

$$\mathbf{g}(\tilde{\theta}) - \mathbf{R}(\tilde{\theta}) \tilde{\lambda} = \mathbf{0}$$

of the constrained maximization problem

$$\max_{\theta} \mathcal{L}(\theta) - \lambda' \mathbf{r}(\theta)$$

equivalence of score and multiplier version becomes obvious.  
(cf. Arellano, 2004)