



## help ardl

**Title**

**ardl** — Autoregressive distributed lag regression model

**Syntax**

**ardl** *depvar* [*indepvars*] [*if*] [*in*] [, *options*] (Syntax 1, for estimation)

**ardl** , { **fbounds**(*casenum*) | **tbounds**(*casenum*) } (Syntax 2, for displaying critical value tables)

<i>options</i>	Description
Model	
<b>lags</b> ( <i>numlist</i> )	set lag lengths
<b>maxlags</b> ( <i>numlist</i> )	set maximum lag lengths
<b>minlag1</b>	require at least one lag for <i>indepvars</i>
<b>maxcombs</b> ( <i>numlist</i> )	set maximum number of lag permutations for lag selection
<b>ec</b>	estimate with <i>depvar</i> in first differences and display output in error-correction form
<b>aic</b>	use AIC as information criterion
<b>bic</b>	use BIC as information criterion; default
<b>exog</b> ( <i>exogvars</i> )	exogenous variables in regression
<b>noconstant</b>	suppress constant term
<b>trendvar</b> ( <i>trendvarname</i> )	specify trend variable
<b>restricted</b>	restrict constant or trend term (see <a href="#">Deterministic components</a> )
<b>regstore</b> ( <i>storename</i> )	stores estimation results from underlying <b>regress</b> command as <i>storename</i>
<b>perfect</b>	do not check for collinearity
Reporting (Syntax I)	
<b>noctable</b>	do not display coefficient table
<b>btest</b>	display Pesaran/Shin/Smith (2001) bounds test
<b>display_options</b>	control column formats, row spacing, line width, and display of omitted variables and base and empty cells
Reporting (Syntax II)	
<b>fbounds</b> ( <i>casenum</i> )	display Pesaran/Shin/Smith (2001) critical values for the F-statistics for case <i>casenum</i> . For more information on <i>casenum</i> , see <a href="#">Deterministic components</a>
<b>tbounds</b> ( <i>casenum</i> )	works in analogy to option <b>fbounds</b> , but refers to the t-statistic

You must **tsset** your data before using **ardl**; see [\[TS\] tsset](#).  
**by** is allowed; see [\[D\] by](#).  
*depvar* and *indepvars* may NOT contain time-series operators.

**Description**

**ardl** fits a linear regression model of *depvar* on *indepvars* with lagged *depvar* and *indepvars* as additional regressors. Information criteria are used to find the optimal lag lengths, if those are not pre-specified as an option. Estimation output is delivered either in levels-form or in error-correction-form. As an option, it displays results from the Pesaran/Shin/Smith (2001) bounds testing procedure for the existence of a levels-relationship.

In syntax II, **ardl** is a convenience tool to display entire tables of critical values from the Pesaran/Shin/Smith (2001) bounds test.

**Abbreviations and definitions used in this help entry**

## Abbreviations:

ARDL: auto-regressive distributed lag  
 PSS: Pesaran/Shin/Smith (2001)  
 VECM: vector error-correction model

**Options**


---

 Model
 

---

**lags**(*numlist*) specifies the number of lags for some or all regressors. The first number specifies the lag length for *depvar* that has to be larger than 0; the following numbers specify the lag lengths for the independent variables in the order they appear in *indepvars*. 0 is possible for the long-run regressor variables if option **minlag1** is not used. 0 is never possible for the lag order of the dependent variable. Missing values indicate lags that are not pre-specified. Information criteria are used to determine them. For example, **lags**(. . 4) requires the second independent variable to enter with 4 lags while the lags of the dependent variable and the first independent variable are to be determined by an information criterion.

The number of elements in *numlist* (positive integers or dots) must be equal to the number of variables specified in the command line (*depvar* + *indepvars*). Alternatively, *numlist* may only contain one element, in which case this number applies to all variables in *depvar* and *indepvars*.

**maxlags**(*numlist*) specifies the maximum lag order used for optimal lag selection. The first number specifies the maximum lag length for *depvar* that has to be larger than 0; the following numbers specify the maximum lag lengths for the independent variables in the order they appear in *indepvars*. The default maximum lag order is 4.

Since **maxlags** only deals with optimal lag order selection, values for all or some of its elements are ignored if **lags** indicates pre-specified lags for some or all variables.

The number of elements in *numlist* (positive integers or dots) must be equal to the number of variables specified in the command line (*depvar* + *indepvars*). Alternatively, *numlist* may only contain one element, in which case this number applies to all variables in *depvar* and *indepvars*.

**maxcombs**(*combrnum*) specifies the maximum number of lag permutations allowed for the optimal lag selection. If the number of lag permutations required to find the optimal lag lengths exceeds *combrnum*, **ardl** errors out. The default for *combrnum* is 500. You can set *combrnum* to higher values. However, *combrnum* may not exceed the value of your current **matsize** setting.

**ec** will estimate the model in 'first-difference' form (see below) and display the output in error-correction form.

**minlag1** will only consider models where *indepvars* have at least one lag, i.e. the optimal lag selection iterations will skip models where one or more of these variables have a lag length of zero. An implication of this is that you may not use option **minlag1** in conjunction with a lag specification in option **lags** that sets the lag order of any variable to zero.

If in addition option **ec** is specified, the error-correction output of the long-run regressors (other than the dependent variable) are expressed in terms of time  $t-1$ . The default is to write them in terms of time  $t$ . The two parameterizations will yield identical estimates, with the exception of the first first-difference term of each long-run regressor.

**aic** is used to determine the optimal lag lengths with the Akaike information criterion.

**bic** is used to determine the optimal lag lengths with the Bayesian information criterion, which is the default.

**exog**(*exogvars*) specifies additional variables to be tagged on to the regression.

**noconstant** suppresses the constant term in the model.

**trendvar**(*trendvarname*) lets you add a trend term to your model. *trendvarname* must exist in the data set before execution of **ardl** and it must be collinear with *timevar*, where *timevar* is the time variable set by **tsset**. A convenient shortcut that skips the creation of a separate time trend variable is to use **trendvar**(*timevar*).

**restricted** will restrict either the constant term or the time trend, if any of the two are specified. See Deterministic components below.

If no deterministic are in the model, **restricted** will cause an error.

**regstore**(*storename*) will store the estimation results from the underlying regress command. This is useful if you want to perform postestimation operations like predicting residuals, regression diagnostics, and so forth. See estimates and regress postestimation.

Note that if a estimation results set called *storename* already exists, option **regstore** will overwrite it without warning.

**perfect** omits the check for collinearity among the regressors.

---

Reporting (Syntax I)

---

**noctable** suppresses the display of the coefficient table. Useful if the intention is to look at the PSS bounds test only.

**btable** displays the F- and t-statistics in relation to the long-run relationship, and displays critical values for these statistics tabulated in PSS.

*display\_options*: **noomitted**, **vsquish**, **noemptycells**, **baselevels**, **allbaselevels**, **cformat(%fmt)**, **pformat(%fmt)**, **sformat(%fmt)**, and **nolstretch**; see [R] estimation options.

---

Reporting (Syntax II)

---

**fbounds**(*casenum*) displays an entire table of critical values for the F-statistic that can be used for testing for the existence of a long-run relationship, according to *casenum*, where *casenum* pins down the deterministic terms in the model, and is an integer from 1 to 5. See Deterministic components for more information on the meaning of *casenum*.

**tbounds**(*casenum*) works in analogy to **fbounds** but concerns the t-statistic. Here *casenum* must be one of 1, 3, or 5.

**Remarks**

Remarks are presented under the following headings:

- Introduction
- Terminology
- Lag specification
- Deterministic components
- Bounds test for a level relationship
- Pure autoregressive processes
- Postestimation
- Replay

**Introduction**

An autoregressive distributed lag (ARDL) model of order  $p$  and  $q$ , denoted  $ARDL(p,q)$  regresses the dependent variable on  $p$  of its own lags and on  $q$  lags of one or more additional regressors. Multiple regressors are allowed to have different lag orders, in which case the model becomes an  $ARDL(p, q_1, \dots, q_k)$  model, where  $k$  is the number of non-deterministic regressors. ARDL models can, among other things, be used for the estimation and testing of cointegration relationships. Key contributions in this area are [Pesaran and Shin \(1999\)](#) and [Pesaran, Shin and Smith \(1999\)](#). For a succinct exposition of ARDL models in the context of cointegration, see [Hassler and Wolters \(2005\)](#).

### **Terminology**

The regression equation in the sense of the preceding paragraph is referred to as the **levels-equation**. This equation can be rewritten such that the differenced *depvar* is expressed in terms of the lagged *depvar*, levels of *indepvars*, and differenced terms of (*depvar*, *indepvars*) up to orders ( $p-1$ ,  $q_1-1$ ,  $\dots$ ,  $q_k-1$ ). This way of writing the ARDL model is referred to here as the **first-difference form** or equation, although this is a slight abuse of terminology since it is a mere reparameterization of the levels-equation. Dividing the coefficient for the levels regressors by the coefficient of the lagged *depvar* and appropriately accounting for model deterministics then yields the **error-correction-form**. It separates the adjustment coefficient to deviations from long-run equilibrium, long-run coefficients, and short-run coefficients.

**ardl** without option **ec** will run a regression of the levels-equation, save the dependent variable and the regressors in the macros `e(depvar)` and `e(regressors)` and in the matrix `e(b)`, and display a corresponding table of estimates. If option **ec** is used, **ardl** will run a regression corresponding to the first-difference equation and save the dependent variable and the regressors in the macros `e(depvar)` and `e(regressors)`. The coefficient output table and `e(b)` will be in terms of the error-correction form.

### **Lag specification**

Lags specified in options **lags** and **maxlags** refer to lags in the levels equation, whether option **ec** is used or not. For example, if you use **lags(2 4 4)**, the dependent variable will have two lags in the levels regression and the two independent variables will have four lags in the levels regression. The lag length of the first differences in the first-difference equation will be one less for each variable.

In a similar fashion, any lag information saved in `e()` will refer to the levels equation.

### **Deterministic components**

In the vector error-correction model (VECM) literature, it is common to distinguish five different cases of model deterministics:

<i>casenum</i>	<i>description</i>
1	no constant, no trend
2	restricted constant, no trend
3	unrestricted constant, no trend
4	unrestricted constant, restricted trend
5	unrestricted constant, unrestricted trend

Rewriting the levels-equation in first-difference form yields restrictions on the constant term and the linear trend. Cases 2 and 4 impose the implied restriction. If these restrictions are ignored (cases 3 and 5), a constant term in the first-difference equation can generate a linear trend in the levels equation. Likewise, an unrestricted trend in the first-difference equation can generate a quadratic trend in the levels-equation. For a more detailed exposition, see for example [Lütkepohl \(2005\)](#), section 6.4, or [\[TS\] vec](#).

Stata's `vec` command, which estimates VECMs, distinguishes between the same five cases through its **trend** option. The following table provides a mapping between case numbers and **vec** syntax.

*casenum*    **vec** syntax

---

1	trend(none)
2	trend(rconstant)
3	trend(constant)
4	trend(rtrend)
5	trend(trend)

The **ardl** syntax for determining the *casenum* is different from **vec** but close to standard Stata syntax for linear regressions. A constant term can be omitted by using option **noconstant**. To include a time trend, generate a separate trend variable and include it in option **trendvar**. If you want to have a linear time trend and your time series variable is named *timevar*, you can simply use **trendvar(*timevar*)**. The table below provides a mapping between case numbers and **ardl** options. Note that a constant is included in the model by default which is why the option **constant** below is in brackets. It is redundant to specify this option explicitly.

*casenum*    **ardl** options

---

1	noconstant
2	[constant] restricted
3	[constant]
4	[constant] trendvar( <i>trendvarname</i> ) restricted
5	[constant] trendvar( <i>trendvarname</i> )

Whereas the specification of deterministic has considerable implications for the estimation procedures of VECMs, this is not so for ARDL models. In the conditional ARDL modelling approach proposed by PSS, for example, cases 2 and 3 and cases 4 and 5 are based on identical linear regressions of the first-difference equation. The distinction within each case-pair concerns the interpretation of the deterministic terms, i.e. whether they are considered to be part of the long-run relationship or not. Accordingly, the asymptotic distribution for the test for a levels-relationship advanced in PSS is different for each case.

**Bounds test for a level relationship**

**ardl** implements the bounds test for a levels relationship proposed by PSS. When option **bttest** is used, the F-statistic and the t-statistic, which are dependent on *casenum*, are displayed along with critical values of the associated non-standard distributions provided by PSS.

You must use option **ec** in your **ardl** model for bounds test-related statistics to be available.

For cases 1, 3, and 5 only the F-statistic is calculated.

To avoid pretesting problems, PSS suggest to apply the bounds test only to ARDL models without restrictions on the short-run coefficients (i.e. with a sufficiently high and common lag order for the regressors). However, **ardl** saves the F-statistic and t-statistic as well as the relevant critical values in **e()**, independently of the model specification. Moreover, this information is saved in **e()** in all cases, regardless of whether the **bttest** option is used or not. The latter option only concerns the display of test results in the Stata results window.

**Pure autoregressive processes**

You may omit the specification of *indepvars*, in which case the process reduces to a pure autoregressive one. Consequently, you can use **ardl** for the optimal lag selection of pure autoregressive processes. See **varsoc** for an alternative way of doing this.

**Postestimation**

The standard subcommands of **estat** (i.e. **estat summarize**, **estat vce**, **estat ic**) work as usual.

You can use option **regstore** to get predicted values, residuals, and other results. This option stores the estimation results from Stata's **regress** which underlies **ardl** in Stata's estimation results catalogue (see **estimates**). After estimation using **ardl**, you can use **estimates restore** to recover results from **regress**, and then use the many tools of **regress postestimation** to perform the desired calculations. It is recommended that you store the **ardl** results before restoring **regress** results, so you can easily switch back.

### Replay

Replay of estimation results works as usual: Type **ardl**, a comma, and then any of the reporting options for Syntax 1.

### Examples

We use Stata's example data set 'lutkepohl2' that contains quarterly data for German aggregate income, investment, and consumption. We estimate an ARDL model in levels-form using the optimal number of lags according to BIC.

```
. webuse lutkepohl2
. ardl ln inv ln inc ln consump, lags(. . 4) maxlag(3 3 3)
```

Lags for **ln\_inv** and **ln\_inc** are optimally selected. **ln\_consump** is pre-specified to receive a lag order of 4. Here the **maxlag** setting of 3 is ignored. We can display the lags selected by:

```
. matrix list e(lags)
```

To estimate the error-correction coefficients, use option **ec**. We use option **regstore** also so we can generate predicted values later.

```
. ardl ln inv ln inc ln consump, ec regstore(lutreg)
```

Predicted values are generated by restoring the **regress** result:

```
. estimates store lutardl
. estimates restore lutreg
```

We can look at the **regress** results:

```
. regress
. predict yhat if e(sample), xb
. estimates restore lutardl
```

Since we have used option **ec** in the **ardl** estimation, the predicted values refer to the first difference of **ln\_inv**, not to the level:

```
. tsline yhat d.ln inv
```

To give an example which is more meaningful from an economic perspective, we now want to examine a potential levels relationship between consumption and income. The unrestricted constant in the model below is capable of generating the upward drift in the variables that is visible from their time-series graphs.

```
. ardl ln consump ln inc, lags(4) ec
```

The long-run coefficient on income is close to 1 and has a tight confidence interval. To check whether a long-run relationship between consumption and income can be statistically confirmed, we replay the estimation output with the **noctable** and **btest** options, which displays results from the PSS bounds test.

```
. ardl, noctable btest
```

The output shows that we cannot confirm the existence of a levels-relationship. Neither the F-statistic nor the t-statistic reject the null hypothesis of no levels-relationship.

### Saved results

**ardl** saves the following in **e()**:

Scalars

**e(N)** number of observations  
**e(df\_m)** model degrees of freedom  
**e(df\_r)** residual degrees of freedom  
**e(mss)** model sum of squares  
**e(rss)** residual sum of squares  
**e(rmse)** root mean squared error  
**e(r2)** R-squared  
**e(r2\_a)** adjusted R-squared  
**e(ll)** log likelihood under additional assumption of i.i.d. normal errors  
**e(N\_gaps)** number of gaps in sample (note: not number of missings)  
**e(tmin)** first time period in sample  
**e(tmax)** last time period in sample  
**e(rank)** rank of **e(V)**

if option **ec** was used:

**F\_pss** F-statistic, calculated according to *casenum*  
**t\_pss** t-statistic, calculated according to *casenum*  
**case** *casenum* for model deterministics

Macros

**e(cmd)** **ardl**  
**e(cmdline)** command as typed  
**e(model)** **level** or **ec**  
**e(title)** title in estimation output  
**e(depvar)** name of dependent variable  
**e(regressors)** full set of regressors in the ARDL model, as estimated by regress  
**e(tsfmt)** format for the current time variable  
**e(properties)** **b v**

if option **ec** was used:

**lrxvars** non-deterministic regressors in the long-run relationship  
**lrdet** deterministic term in the long-run relationship  
**srvars** short-run (differenced) regressors  
**exogvars** exogenous variables  
**det** deterministic terms in the model, but not in the long-run relationship

Matrices

**e(b)** coefficient vector of the linear regression model  
**e(V)** variance-covariance matrix of the estimators in the linear regression model  
**e(lagcombs)** combinations of lags across which lag selection has searched; includes the information criterion for each lag specification  
**e(maxlags)** vector with maximum lag lengths of *depvar* and *indepvars* in the levels representation used for optimal lag selection  
**e(lags)** vector with number of lags of *depvar* and *indepvars* in the levels representation

if option **ec** was used:

**F\_critval** critical values, F-statistic, PSS bounds test for *casenum*  
**t\_critval** critical values, t-statistic, PSS bounds test for *casenum*

Functions

**e(sample)** marks estimation sample

**Authors**

Original Author: Sebastian Kripfganz, Goethe University Frankfurt, kripfganz@wiwi.uni-frankfurt.de

Code modified by Daniel Schneider, Goethe University Frankfurt, schneider\_daniel@hotmail.com

**References**

Hassler, U. and J. Wolters (2005): Autoregressive Distributed Lag Models and Cointegration. Freie Universität Berlin, Working Paper No.2005/22.

Lütkepohl, H. (2005): New Introduction to Multiple Time Series Analysis. Berlin, Heidelberg: Springer Verlag.

Pesaran, M.H. and Y. Shin (1999): An Autoregressive Distributed Lag Modelling Approach to Cointegration Analysis. In: Strom, S. (Ed.): Econometrics and Economic Theory in the 20th Century: The Ragnar Frisch Centennial Symposium. Cambridge, UK: Cambridge University Press.

Pesaran, M.H., Shin, Y. and R.J. Smith (2001): Bounds Testing Approaches to the Analysis of Level Relationships. Journal of Applied Econometrics, 16 (3), 289-326.

**Also see**