Cointegration Testing and Dynamic Simulations of Autoregressive Distributed Lag Models*

Soren Jordan†
Andrew Q. Philips‡

August 11, 2017

Abstract
In this paper we discuss the bounds cointegration test proposed by Pesaran, Shin and Smith (2001), which we have adapted into a Stata program, `pssbounds`. Since the resulting models can be dynamically complex, we follow the advice of Philips (2017) by introducing `dynardl`, a flexible program designed to dynamically simulate a variety of types of autoregressive distributed lag models, including error-correction models.
Word Count: 63 (abstract); xxx (manuscript)
Keywords: cointegration; dynamic modeling; autoregressive distributed lag; error-correction

---

*Source code for the programs discussed in this paper can be found at http://andyphilips.github.io/pssbounds/ and https://andyphilips.github.io/dynardl/.

†sorenjordanpols@gmail.com. Assistant Professor, Department of Political Science, Auburn University.

‡andrew.philips@colorado.edu. Assistant Professor, Department of Political Science, University of Colorado at Boulder, UCB 333, Boulder, CO 80309-0333.

Introduction

Time series models employing an autoregressive distributed lag (ARDL) are commonplace in the social sciences. Whether the dependent variable is estimated in levels or in first differences (sometimes involving an error-correction term), these models are able to test a host of theoretically-important theories, from the effect of public opinion on government response (Jennings and John 2009) to the effect of domestic and international factors on defense expenditures (Whitten and Williams 2011) or tax rates (Swank and Steinmo 2002), to analyzing dynamic changes in partisan responsiveness over time (Ura and Ellis 2008).

When employing an error-correction-style ARDL model, it becomes necessary to test for cointegration. In a recent paper, Philips (2017) shows that in small samples common in the social sciences (typically $T \leq 80$) the ARDL bounds test for cointegration proposed by Pesaran, Shin and Smith (2001) tends to be more conservative (i.e., does not conclude cointegration when it does not exist) than either the Engle-Granger "two-step" (Engle and Granger 1987) or the Johansen (1991, 1995) approaches to cointegration testing. In this paper we present `pssbounds`, which provides the non-standard critical values from Pesaran, Shin and Smith (2001) needed to conduct the cointegration test.

In addition to its error-correction form, ARDL models in general may have complex dynamic specifications, including lags, first-differences, and lagged first-differences. Since it can become hard to interpret the effects of changes in the independent variable under these circumstances, we introduce `dynardl`, a flexible program that allows users to dynamically simulate a variety of ARDL models, including the error-correction model. Dynamic simulations offer an alternative to hypothesis testing of model coefficients by instead summarizing the substantive significance of our results through meaningful counterfactual scenarios. Such an approach has been gaining popularity in the social sciences (e.g., Tomz, Wittenberg and King 2003; Imai, King and Lau 2009; Williams and Whitten 2011, 2012; Philips, Rutherford and Whitten 2016*b*).

Below, we offer a brief discussion of the ARDL-bounds approach to cointegration testing. We then present a Stata command that provides the necessary critical values for the test—`pssbounds`—along with several applied examples. Next, we discuss `dynardl`, a command to produce dynamic simulations of a host of ARDL-style models.

The ARDL-Bounds Cointegration Test

The concept of cointegration has been around for several decades. Time series may have "full-memory", such that current realizations are fully a function of all previous stochastic shocks, plus some new innovation. Such series are said to be integrated of order one (or I(1)), a form of non-stationarity.[1] Two or more I(1) series may have short-run, as well as long-run—or equilibrium—relationships. That is to say, while short-run perturbations may move the series apart, over time this disequilibrium is corrected as the series move back towards a stable long-run relationship. Such series are said to be cointegrating.

Not all I(1) series are cointegrating. Thus, it is necessary to test for cointegration. The earliest test comes from Engle and Granger (1987), who show that cointegration between k weakly exogenous I(1) regressors, $x_{1t}, x_{2t}, \cdots, x_{kt}$, and an I(1) regressand, y_t, exists if the resulting residuals—from a regression of these variables entering into the equation in levels—is stationary:

$$y_t = \kappa_0 + \kappa_1 x_{1t} + \kappa_2 x_{2t} + \cdots + \kappa_k x_{kt} + z_t \tag{1}$$

A number of other cointegration tests have since been proposed, including Johansen (1991) and Phillips and Ouliaris (1990). Philips (2017) offers an in-depth discussion of how to apply the Pesaran, Shin and Smith (2001) ARDL-bounds test for cointegration. This test offers several advantages. Chief among them is that users do not have make the sharp I(0)/I(1) distinction for the regressors.[2] Below, we briefly summarize this approach.[3]

[1] Another term for an I(1) series is that it contains a unit-root. Series that are stationary are said to be I(0).

[2] Users must ensure, however, that regressors are not of order I(2) or more, and that seasonality has been removed from the series.

[3] A more in-depth discussion can be found in Philips (2017).

First, the analyst must ensure that the dependent variable is I(1). There are many unit-root tests that can be used to determine the order of integration of a series, including the Dickey-Fuller, Phillips-Perron, Elliott-Rothenberg-Stock, and Kwiatkowski-Phillips-Schmidt-Shin tests, among others. Only a non-stationary dependent variable is a potential candidate for cointegration.

Second, the analyst must ensure that the regressors are not of an order of integration higher than I(1). While this means that the analyst does not have to make the potentially-difficult I(0)/I(1) decision, they must also ensure that all regressors are not explosive or contain seasonal unit roots.

Third, the analyst estimates an ARDL model in error-correction form. The model appears as follows:

$$\Delta y_t = \alpha_0 + \theta_0 y_{t-1} + \theta_1 x_{1,t-1} + \cdots + \theta_k x_{k,t-1} + \sum_{i=1}^{p} \alpha_i \Delta y_{t-1} + \sum_{j=0}^{q_1} \beta_{1j} \Delta x_{1,t-j} + \cdots + \sum_{j=0}^{q_k} \beta_{kj} \Delta x_{k,t-j} + \varepsilon_t \tag{2}$$

Where the change in the dependent variable is a function of a constant, its prior value (appearing in levels), prior values of all regressors in levels, as well as up to p and q_k lags of the first difference of the dependent variable and regressors, respectively. These enter into Equation 2 in order to ensure white-noise residuals. Information criteria such as SBIC and AIC, as well as autocorrelation and heteroskedasticity tests should be used to check for this.

While the bounds test may relatively easy to implement, users must look up the special critical values. Pesaran, Shin and Smith (2001) provide asymptotic critical values, while Narayan (2005) offers them for finite samples. In order to make the bounds test more accessible to users, below we introduce `pssbounds`, provides the necessary critical values through an easy-to-use command.

pssbounds Syntax

```
pssbounds, observations( ) k( ) fstat( ) [tstat( ) case( ) sig( )]
```

Options

`observations( )` is the number of observations from the estimated ARDL model in error correction form. This is a required option.

`k( )` is the number of regressors, k, modeled in levels in the estimated ARDL model.[4] The bounds F-test is a test that the k parameters on the regressors appearing in levels (plus the coefficient on the lagged dependent variable, θ_0) are jointly equal to zero: $H_0 = \theta_0 + \theta_1 + ... + \theta_k = 0$. This option is required, since the critical values differ based on the number of regressors.

`fstat( )` is the value of the F-statistic from the test that all parameters on the regressors appearing in levels, plus the coefficient on the lagged dependent variable, are jointly equal to zero: $H_0 = \theta_0 + \theta_1 + ... + \theta_k = 0$. After running the ARDL model in error-correction form, users should use Stata's `test` command to obtain the F-statistic.[5] This option is required.

`tstat( )` is the value of the t-statistic for the coefficient on the lagged dependent variable. This serves as a one-sided auxiliary test to the bounds F-test. Only asymptotic critical values are available, so this option is not required. Note that critical values do not currently exist for this test for Cases II, and IV (see below).

`type( )` specifies the potential restrictions on the constant and trend terms, which in turn lead to different critical values for the bounds test. This option is not required, however. The default if this option is not specified—by far the most common—is an unrestricted intercept with no trend term: `type(case3)`. Pesaran, Shin and Smith (2001) refer to this as "Case III". Other types that are supported are:

[4] For instance if the ARDL model was: $\Delta y_t = \beta_0 - \theta_0 y_{t-1} + \beta_1 \Delta x_{1t} + \theta_1 x_{1,t-1} + \beta_2 \Delta x_{2t} + \theta_3 x_{2,t-1}$, then $k = 2$.

[5] For example, `test l.y l.x1 l.x2...`

- Case I: No intercept and no trend, `type(case1)`.
- Case II: Restricted intercept and no trend, `type(case2)`.
- Case IV: Unrestricted intercept and restricted trend, `type(case4)`.
- Case V: Unrestricted intercept and unrestricted trend, `type(case5)`.

`sig( )` is the significance level for the critical values. By default, significance at the 0.05 level is reported: `sig(95)`, but 99, 97.5, and 90 are supported for asymptotic critical values.

Examples

Two example applications of `pssbounds` are shown below. For the first example, we will use the Lutkepohl West German quarterly macroeconomic dataset available in Stata:

```
. webuse lutkepohl2
(Quarterly SA West German macro data, Bil DM, from Lutkepohl 1993 Table E.1
. tsset
        time variable:  qtr, 1960q1 to 1982q4
                delta:  1 quarter
```

Next, we will run the ARDL-bounds model.[6] The results are shown in Model 1 in Table 1.

```
. regress d.ln_inv l.ln_inv d.ln_inc l.ln_inc d.ln_consump l.ln_consump
```

We then run an F-test that the coefficients on the variables appearing in levels (ln_inv_{t-1}, ln_inc_{t-1}, and ln_consump_{t-1}) are jointly equal to zero:

[6] As detailed in Philips (2017), estimation is not the first step of the ARDL-bounds approach; we would need to conduct unit root test on the dependent variable, and ensure that the independent variables were I(1) or less. We would also need to ensure that the resulting residuals from the model are white-noise. This is a simply a stylized example used to showcase the command using readily-available Stata datasets.

Table 1: Lutkepohl Example

	(1)	(2)
	Δln_inv$_t$	Δln_inv$_t$
ln_inv$_{t-1}$	-0.140*	-0.152**
	(0.059)	(0.056)
Δln_inc$_t$	-0.201	-0.0985
	(0.459)	(0.423)
ln_inc$_{t-1}$	-0.209	
	(0.334)	
Δln_consump$_t$	1.548**	1.395**
	(0.538)	(0.459)
ln_consump$_{t-1}$	0.336	0.131**
	(0.333)	(0.048)
Constant	-0.008	
	(0.071)	
N	91	91
R^2	0.17	0.27

Dependent variable is Δln_inv$_t$. Standard errors in parentheses. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

```
. test l.ln_inv l.ln_inc l.ln_consump

 ( 1)  L.ln_inv = 0

 ( 2)  L.ln_inc = 0

 ( 3)  L.ln_consump = 0

       F(  3,    85) =    2.60

            Prob > F =    0.0573
```

Recall that the critical values are non-standard, so we only need the value of the F-statistic, which is 2.60. In order to test for cointegration, we use `pssbounds`. For the required option `fstat`, we input the F-statistic from above. From the estimated model, we tell `pssbounds` that the number of observations is 91, the case is case III (i.e., unrestricted intercept with no trend, as shown in Model 1), and that there are two regressors appearing in levels ($k = 2$). The resulting output appears as follows:

```
. pssbounds, fstat(2.60) obs(91) case(3) k(2)

PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST

Obs: 91
```

```
No. Regressors (k): 2

Case: 3

------------------------------------------------------------

                                 F-test

------------------------------------------------------------

                   <----- I(0) ---------- I(1) ----->

10% critical value       3.170           4.140

5% critical value        3.790           4.850

1% critical value        5.150           6.360

F-stat. =     2.600

------------------------------------------------------------

F-statistic note: Asymptotic critical values used.
```

For this model, since the F-statistic of 2.60 is below the I(0) critical value—even at the 10 percent level—we can conclude that there is no cointegration, and that all regressors appearing in levels are stationary.

Purely for illustrative purposes, let us now assume that we wanted to estimate a model without a constant. These results are shown in Model 2, Table 1. As with Model 1, we next run an F-test of all variables appearing in levels. Note that the model also has `ln_inc` appearing in first differences, but not in levels; thus, $k = 1$:

```
. regress d.ln_inv l.ln_inv d.ln_inc d.ln_consump l.ln_consump, noconstant


. test l.ln_inv l.ln_consump

 ( 1)  L.ln_inv = 0

 ( 2)  L.ln_consump = 0


       F(  2,    87) =    3.83

            Prob > F =    0.0254
```

We can account for a restricted constant by specifying the `case(1)` option (i.e., no intercept and no trend). As an additional option, we add `tstat(-2.73)`, which is the t-statistic on the coefficient on the lagged dependent variable.

```
. pssbounds, fstat(3.83) obs(91) case(1) k(1) tstat(-2.73)

PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST

Obs: 91

No. Regressors (k): 1

Case: 1

------------------------------------------------------------

                                 F-test

------------------------------------------------------------

                   <----- I(0) ---------- I(1) ----->

10% critical value       2.440           3.280

5% critical value        3.150           4.110

1% critical value        4.810           6.020


F-stat. =     3.830

------------------------------------------------------------

                                 t-test

------------------------------------------------------------

                   <----- I(0) ---------- I(1) ----->

10% critical value      -1.620          -2.280

5% critical value       -1.950          -2.600

1% critical value       -2.580          -3.220


t-stat. =    -2.730

------------------------------------------------------------

F-statistic note: Asymptotic critical values used.

t-statistic note: Asymptotic critical values used.
```

Now the output of `pssbounds` contains critical values for both the F-test and one sided t-test. Based on the F-statistic, we can conclude cointegration at the 10 percent level since the F-statistic of 3.83 is above the I(1) critical threshold of 3.28. However, there is not strong enough evidence to support cointegration at the five percent level. For the t-test, the t-statistic of -2.73 falls below the critical I(1) threshold of -2.60, supporting the earlier conclusion of cointegration. Also note that for both tests, `pssbounds` issued a warning that asymptotic critical values are used. For all cases, only asymptotic critical values from Pesaran, Shin and Smith (2001) are provided for the t-statistic test.[7] Thus, interpreting the results of this test should be done with caution in small samples. Small sample critical values for the F-statistic are not available for case I.

As a second example, we use data from Ura (2014), who examines public mood liberalism in the US. Ura argues that in the short-run, there will be a public "backlash" in response to liberal Supreme Court decisions, but that in the long-run the sentiments of the public tend to follow closely to those of the Court. Using the same dataset, Philips (2017, see supplemental materials p. 110) finds that the dependent variable, public mood liberalism, is I(1), and that Ura's ARDL model estimated in error-correction form with an additional lagged first-difference of unemployment produces the white-noise residuals needed to conduct the ARDL-bounds test. The model is shown in Table 2.

```
. use "Ura 2014 replication/supreme court mood replication.dta", clear
. tsset
        time variable:  year, 1955 to 2009
                delta:  1 unit

.regress d.mood l.mood d.policy l.policy d.unemployment dl.unemployment  ///
 l.unemployment d.inflation l.inflation d.caselaw l.caselaw
```

Next we run the F-test that all variables appearing in levels are jointly equal to zero:

```
. test l.mood l.policy l.unemployment l.inflation l.caselaw
 ( 1)  L.mood = 0
 ( 2)  L.policy = 0
 ( 3)  L.unemployment = 0
```

[7] Nor to critical values for the t-statistic test exist for cases II and IV.

Table 2: Ura (2014) Example

	(1)
	Δmood$_t$
mood$_{t-1}$	-0.241**
	(0.076)
Δpolicy$_t$	0.051
	(0.069)
policy$_{t-1}$	-0.073***
	(0.020)
Δunemployment$_t$	-0.106
	(0.265)
Δunemployment$_{t-1}$	-0.538*
	(0.242)
unemployment$_{t-1}$	-0.024
	(0.198)
Δinflation$_t$	-0.306*
	(0.123)
inflation$_{t-1}$	-0.299*
	(0.120)
Δcaselaw$_t$	-0.093*
	(0.037)
caselaw$_{t-1}$	0.027*
	(0.011)
Constant	15.65**
	(5.050)
N	53
R^2	0.47

Dependent variable is Δmood$_t$.
Standard errors in parentheses.
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

```
 ( 4)  L.inflation = 0
 ( 5)  L.caselaw = 0

       F(  5,    42) =    5.15
            Prob > F =    0.0009
```

We then run pssbounds on the resulting F-statistic of 5.15, where $k = 4$ (i.e., four regressors):

```
. pssbounds, fstat(5.15) obs(53) case(3) k(4)
PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
Obs: 53
No. Regressors (k): 4
Case: 3
------------------------------------------------------------
                                F-test
------------------------------------------------------------
                   <----- I(0) ---------- I(1) ----->
10% critical value       2.578           3.710
5% critical value        3.068           4.334
1% critical value        4.244           5.726

F-stat. =     5.150
------------------------------------------------------------
                                t-test
------------------------------------------------------------
                   <----- I(0) ---------- I(1) ----->
10% critical value      -2.570          -3.660
5% critical value       -2.860          -3.990
1% critical value       -3.430          -4.600

t-stat. =    -3.190
```

```
-----------------------------------------------------------

F-statistic note:

t-statistic note: Small-sample critical values not provided for Case III. Asymptotic

> critical values used.
```

We can conclude evidence of cointegration at the 5 percent level for the F-test, since the F-statistic of 5.15 is above the I(1) critical value of 4.334. However, note that we fail to clear the critical I(1) threshold for the ARDL-bounds t-test. Overall, given that the t-test values are asymptotic, we have relatively strong evidence of cointegration.

Dynamic Simulations of ARDL Models

ARDL models may have a fairly complex lag structure, with lags, contemporaneous values, first differences, and lagged first differences of the independent (and sometimes the dependent) variable appearing in the model specification. While interpreting short- and long-run effects may be simple in something like an ARDL(1,1) model (i.e., one lag of the dependent variable, contemporaneous and one lag of all independent variables), understanding the short-, medium-, and long-run effects becomes difficult as the model specification grows in complexity.

To better interpret the substantive significance of our results, below we introduce `dynardl`, a command to dynamically simulate a variety of different ARDL models. `dynardl` estimates, simulates, and can even plot a variety of ARDL models. The resulting output helps us visualize the effect of a counterfactual change in one weakly exogenous regressor at a single point in time, holding all else equal, using stochastic simulation techniques. Dynamic simulation approaches are gaining in popularity as a simple way to show the substantive results of time series models, whose coefficients often have non-intuitive or "hidden" interpretations (Breunig and Busemeyer 2012; Williams and Whitten 2011; Philips, Rutherford and Whitten 2016*a*,*b*; Gandrud, Williams and Whitten 2016).[8]

[8] For instance, coefficients on regressors in an ARDL(1,0) have both a contemporaneous effect (given by the coefficient) as well as a long-run or cumulative effect, given by $\frac{\beta}{1-\theta_0}$.

Before using the command, it is assumed that the user has already determined the order of integration of the variables through unit-root testing, diagnosed and addressed other issues such as seasonal unit roots, and used information criteria (and theory) to identify the best fitting lagged-difference structure, which is used to purge autocorrelation and to ensure the residuals are white noise. If an error-correction model is estimated, users should have already performed the ARDL-bounds test to determine if there is cointegration (and if there is not, adjusted the model accordingly). See Philips (2017) for a step-by-step example of this process for the ARDL model in general.

`dynardl` first runs a regression using OLS. Then, using a self-contained procedure similar to the popular Clarify program for Stata (Tomz, Wittenberg and King 2003), it takes 1000 draws (or however many simulations a user desires) of the vector of parameters from a multivariate normal distribution. These distributions are assumed to have means equal to the estimated parameters from the regression. The variance of these distributions is equal to the estimated variance-covariance matrix from the regression. In order to re-introduce stochastic uncertainty back into the model when creating predicted values, `dynardl` simulates σ^2 by taking draws from a scaled inverse χ^2 distribution. The distribution is scaled by the residual degrees of freedom (n-k), as well as the estimated $\hat{\sigma}^2$ from the regression (Gelman et al. 2014, pp. 43, 581). This ensures that draws of σ^2 are bounded by zero and one. Simulated parameters and sigma-squared values are then used to create predicted values of the dependent variable over time, $\hat{Y}_t$, for each of the simulations, by setting all covariates to certain values (typically means). Stochastic uncertainty is introduced into the prediction by taking a draw from a multivariate normal distribution with mean zero and variance $\hat{\sigma}^2$. The program then averages across the simulations, creating $\hat{Y}_t^*$ (the predicted values plus stochastic uncertainty) as well as percentile confidence intervals of the distribution of simulated values at a particular point in time. These are then saved, allowing a user to make a table or (more commonly) a graph of the results over time.

`dynardl` Syntax

`dynardl depvar indepvars [, options]`

The options below are required:

- `lags(numlist)` is a numeric list of the number of lags to include for each variable. The number of desired lags is listed in the order in which the variables `depvar` and `indepvars` appear. For instance, in a model with two weakly exogenous variables, we lag all variables by one period by specifying: `lags(1 1 1)`. Note that the lag on `depvar` (the first "`1`") must always be specified. To estimate a model without a lag for a particular variable, simply replace the number with a "."; for instance, if we did not want a lag on the first regressor, we type: `lags(1 . 1)`. `dynardl` can handle consecutive lags by specifying the minimum lag, a backslash, followed by the maximum lag. For instance, `lags(1/3 . .)` will introduce lags of y_t at $t-1$, $t-2$, and $t-3$ into the model. To add a single lag of y_t at $t-3$, specify `lags(3 . .)`.

- `shockvar(varname)` is a single independent variable from the list of `indepvars` that is to be shocked. It will experience a counterfactual shock of size `shockval(#)` at time `time(#)`.

- `shockval(#)` is the amount to shock `shockvar(varname)` by. Most commonly, a +/- one standard deviation shock is specified.

The following options are not required:

- `diffs(numlist)` is a numeric list of the number of contemporaneous first differences (i.e., $t-(t-1)$) to include for each variable. Note that the first entry (the placeholder for the `depvar`) will always be empty (denoted by "."), since the first difference of the dependent variable cannot appear on the right-hand side of the model.[9]

[9]It can however, appear in *lagged* first differences, as shown below. Note that only first-differences can be taken using this option (e.g., `diffs(. 1 1)`).

- `lagdiffs(numlist)` is a numeric list of the number of lagged first differences to include for each variable. The lag syntax is the same as for `lags( )`. For instance, to include a lagged first difference at $t-2$ for `depvar`, a lag at $t-1$ for the first weakly exogenous regressor, and none for the second, specify `lagdiff(2 1 .)`. To include an additional lag for both the first and second lagged first differences of `depvar`, specify `lagdiff(1/2 1 .)`.

- `level(numlist)` is a numeric list of variables to appear in levels (i.e., not lagged or differenced but appearing contemporaneously at time t).[10]

- `ec` if specified, `depvar` will be estimated in first differences. If estimating an error correction model, users will need to use this option.

- `range(#)` is the length of the scenario to simulate. By default, this is $t = 20$. Note that the range must be larger than `time( )`.

- `sig(#)` specifies the significance level for the percentile confidence intervals. The default is for 95% confidence intervals.

- `time(#)` is the scenario time in which the shock occurs to `shockvar( )`. The default time is $t = 10$.

- `saving(string)` specifies the name of the output file. If no filename is specified, the program will save the results as "dynardl_results.dta".

- `forceset(numlist)` by default, the program will estimate the ARDL model in equilibrium; all lagged variables and variables appearing in levels are set to their sample means. All first differences and any lagged first differences are set to zero. This option allows the user to change the setting of the lagged (or unlagged if using `levels( )`) levels of the variables. This could be useful when estimating a dummy variable. For instance, when we wish to see the effect of a movement from zero to one.

[10] If both `level( )` and `ec` are specified, `dynardl` will issue a warning message. Of course, users may have a valid reason to include a variable in levels; for instance, a dummy variable.

- `sims(#)` is the number of simulations (default is 1000). If confidence intervals are particularly noisy, it may help to increase this number. Note that you may also need to increase the matsize in Stata.

- `burnin(#)` allows `dynardl` to iterate out so starting values are stable. This option is rarely used. However, if using the option `forceset( )`, the predicted values will not be in equilibrium at the start of the simulation, and will take some time to converge on stable values. To get around this, one can use the `burnin` option to specify a number of simulations to "throw away" at the start. By default, this is 20. Burnins do not change the simulation range or time; to simulate a range of 25 with a shock time at 10 and a burnin of 30, specify: `burnin(30) range(25) time(10)`.

- `graph` although `dynardl` saves the means of the predicted values and user-specified confidence intervals in `saving`, users can use this option to automatically plot the dynamic results using a spikeplot. Two alternative plots are possible:

  - By adding the option `rarea`, the program will automatically create an area plot. Predicted means along with 75, 90, and 95 percent confidence intervals are shown using the area plot.

  - By adding the option `change`, predicted changes (from the sample mean) are shown across time (starting with the time at which the shock occurs), similar to an impulse response function.

- `expectedval` by default, `dynardl` will calculate predicted values of the dependent variable for a given number of simulations. For every simulation, the predicted value comes from a systematic component as well as a single draw from the stochastic component. With the `expectedval` option, the program instead calculates expected values of the dependent variable such that the average of 1000 stochastic draws now becomes the estimate of the stochastic component for each of the simulations. This effectively removes the stochastic uncertainty introduced in calculating $\hat{Y}_t$. Predicted values are more conservative than expected values. Note that `dynardl` takes longer to run if calculating expected values.

Examples

For the first example we will once again use the results from Model 1 in Table 1 using the Lutkepohl data.

$$\Delta \text{ln\_inv}_t = \text{ln\_inv}_{t-1} + \Delta \text{ln\_inc}_t + \text{ln\_inc}_{t-1} + \Delta \text{ln\_consump}_t + \text{ln\_consump}_{t-1} \quad (3)$$

Since the `dynardl` uses Stata's matrix capabilities, we will increase the maximum matrix size as well:

```
. set matsize 5000
. webuse lutkepohl2
(Quarterly SA West German macro data, Bil DM, from Lutkepohl 1993 Table E.1
. tsset
        time variable:  qtr, 1960q1 to 1982q4
                delta:  1 quarter
```

To estimate the model shown in Equation 3 using `dynardl`, and see the effect of a −1 shock to ln_inc (about two standard deviations) we specify the command as follows:

```
dynardl ln_inv ln_inc ln_consump,   ///
lags(1 1 1) diffs(. 1 1)   ///
shockvar(ln_inc) shockval(-1)    ///
time(10) range(30) graph ec
```

In the command, `lags(1 1 1)` tells `dynardl` to add lags (of $t-1$) for each of the variables, while `diffs(. 1 1)` means that the second and third variables (ln_inc and ln_consump) enter into the model as first-differences as well. In `shockvar( )` we include the variable to be shocked, and specify the amount to shock it by using `shockval`. Additional options include the time at which the shock occurs, `time(10)`, the total range of the simulations, `range(30)`, and that the dependent variable is to be included in first-differences, `ec`. Last, since we specified the `graph` option, `dynardl` will produce a plot, which is shown in Figure

1. As is clear from the figure, a -1 shock at $t = 10$ produces a small increase that is not statistically significant in the short-run, which eventually increases to a predicted value of about 7.5 over the long-run, an increase that is statistically significant.

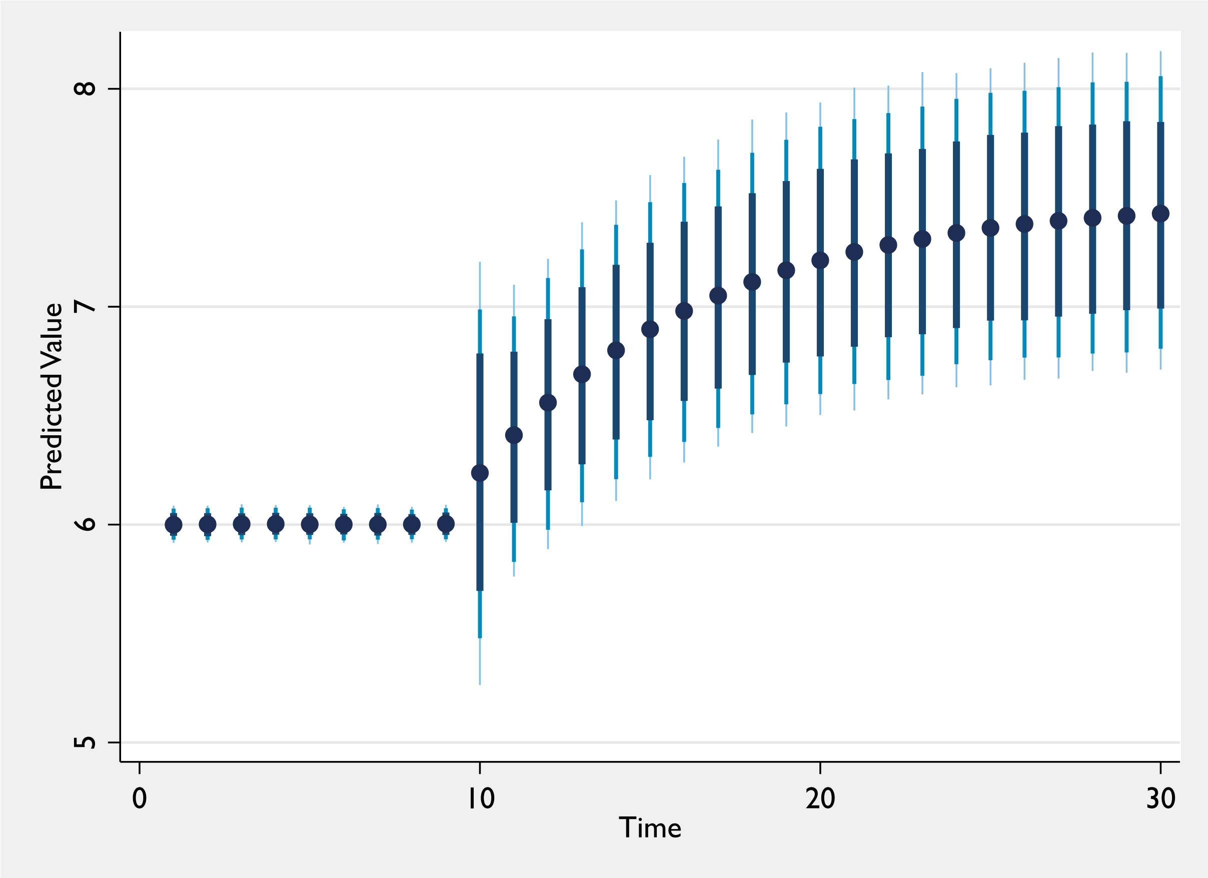

Figure 1: Plot Produced from `dynardl` Using the `graph` Option

Note: Dots show average predicted value. Shaded lines show (from darkest to lightest) the 75, 90, and 95 percent confidence intervals.

In addition to producing figures, `dynardl` also saved the prediction output, which can be used to create more customizable figures (e.g., colors, lines, labels) if users desire. Since we did not specify a filename to save as using the `saving()` option, the results are automatically saved as "dynardl_results.dta".

More complex dynamic specifications are possible using `dynardl`. For instance, perhaps we wanted to estimate the following equation:

$$\begin{aligned}\Delta\text{ln\_inv}_t =&\text{ln\_inv}_{t-1} + \Delta\text{ln\_inv}_{t-1} + \text{ln\_inc}_{t-1} + \text{ln\_inc}_{t-2} + \text{ln\_inc}_{t-3} + \\ &\Delta\text{ln\_consump}_t + \text{ln\_consump}_{t-1}\end{aligned} \tag{4}$$

Now, the dependent variable appears not only in lagged-level form at $t-1$, but also as a lagged first-difference. ln_inc no longer appears contemporaneously but at the first, second, and third lags, while ln_consump continues to appear in lagged and first-differenced forms. Equation 4 would appear as the following in `dynardl`:

```
dynardl ln_inv ln_inc ln_consump,   ///
lags(1 1/3 1) diffs(. . 1) lagdiffs(1 . .)   ///
shockvar(ln_inc) shockval(-1)    ///
time(10) range(30) graph ec rarea sims(5000)
```

Note that, since the first through third lag of ln_inc are desired, we specify `lags(1 1/3 1)`. Since there is a lagged first-difference included for the dependent variable, we add `lagdiffs(1 . .)` (the "." are placeholders for the other variables and must be included). Last, we add the `rarea` option to produce an area plot, and increase the number of simulations to 5000 with `sims(5000)`. The resulting plot is shown in Figure 2.

In addition to error-correction style models, `dynardl` can handle ARDL models where the dependent variable is estimated in levels. For instance, perhaps we wanted to estimate the following ARDL(1,1) model:

$$\text{ln\_inv}_t = \text{ln\_inv}_{t-1} + \text{ln\_inc}_t + \text{ln\_inc}_{t-1} + \text{ln\_consump}_t + \text{ln\_consump}_{t-1} \tag{5}$$

In `dynardl`, we add the `level(. 1 1)` to let the program know that the two independent varibles are to appear contemporaneously in levels. If we wanted to see the effect

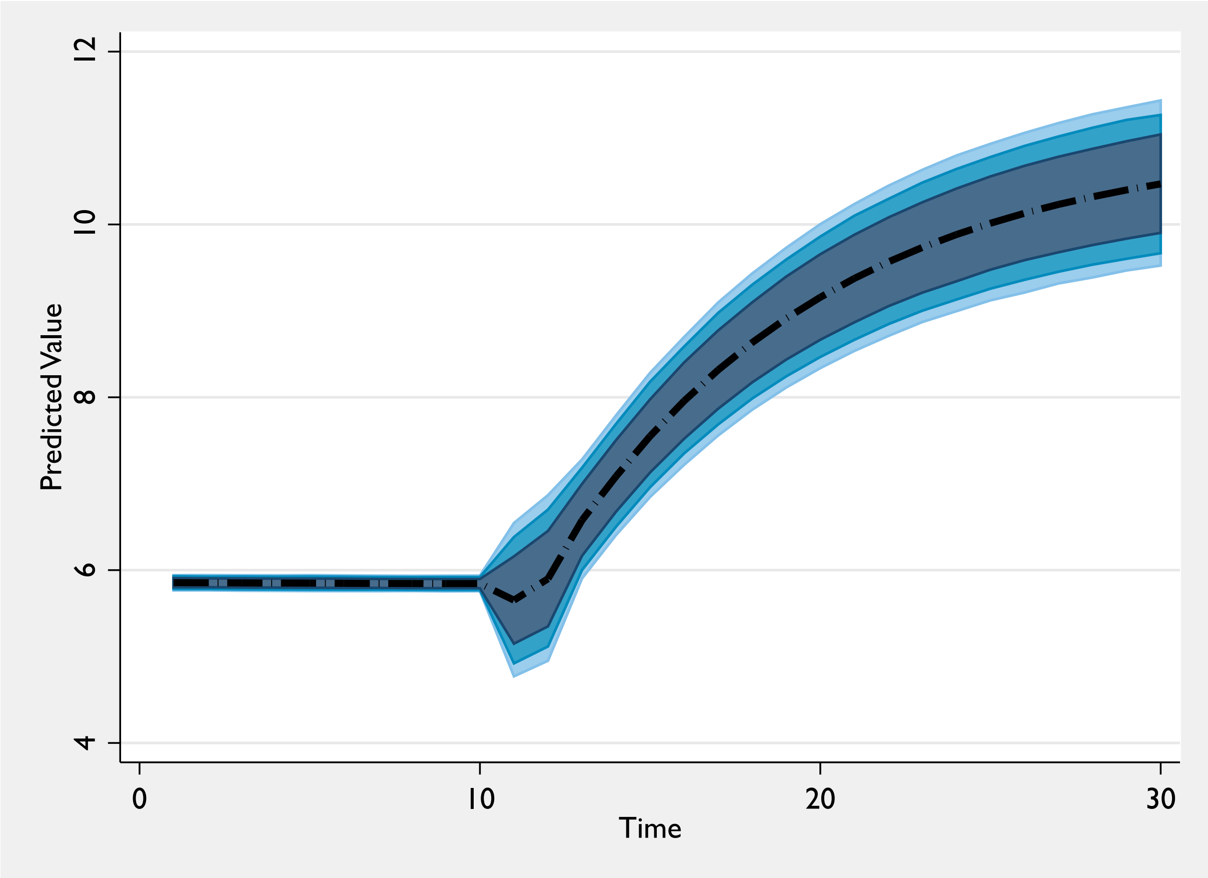

Figure 2: Plot Produced from `dynardl` Using the `rarea` Option

Note: Black dotted line shows average predicted value. Shaded area shows (from darkest to lightest) the 75, 90, and 95 percent confidence intervals.

of a change from ln_inc= 6 to ln_inc= 5, while holding ln_consump constant at 7, we can also use the `forceset( )` option to force the program to evaluate the simulations at these values, not the sample means (by default).

```
dynardl ln_inv ln_inc ln_consump,   ///
lags(1 1 1) level(. 1 1) forceset(. 6 7)   ///
shockvar(ln_inc) shockval(-1)    ///
time(10) range(30) graph change sims(5000)
```

Since we added the `change` option, the resulting plot is an impulse response function, as shown in Figure 3. The figure shows the change in predicted value, starting when the shock occurs.[11] As is clear from the figure, there is no statistically significant change in the predicted value in the short-run as a result of the shock. However, over the long run the change is statistically significant at the 90 percent level of confidence.

Conclusion

In this paper we have introduced two Stata programs. `pssbounds` is designed to help users test for cointegration, while `dynardl` helps users dynamically simulate a variety of autoregressive distributed lag models in order to gain a better understanding of the substantive significance of their results. Both should make it easier for users to test and interpret their models.

[11]Since the shock occurred at $t = 10$, and the total range of the simulation was $t = 30$, this is why Figure 3 shows a total range of $t = 20$

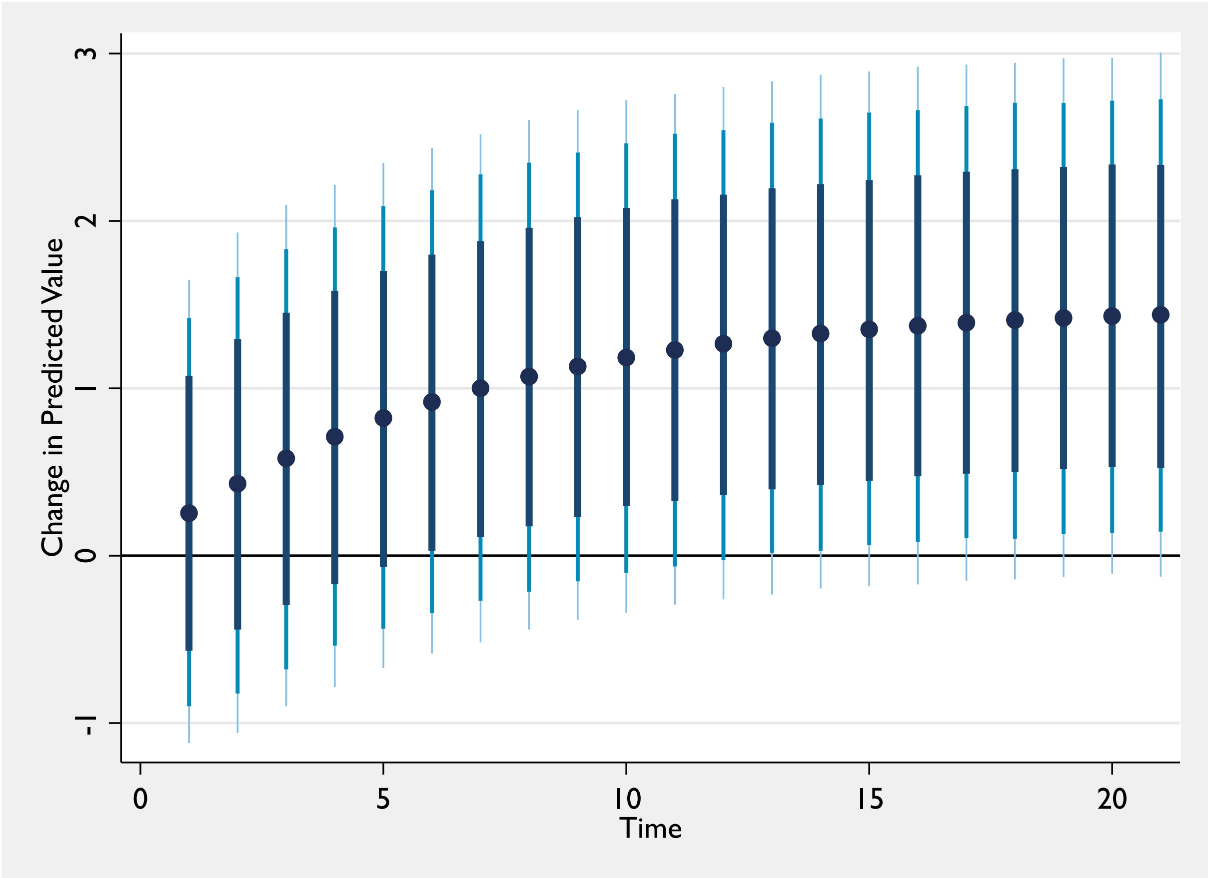

Figure 3: Plot Produced from `dynardl` Using the `change` Option

Note: Dots show mean change in predicted value from sample mean. Shaded area shows (from darkest to lightest) the 75, 90, and 95 percent confidence intervals.

References

Breunig, Christian and Marius R Busemeyer. 2012. "Fiscal austerity and the trade-off between public investment and social spending." *Journal of European Public Policy* 19(6):921–938.

Engle, Robert F and Clive WJ Granger. 1987. "Co-integration and error correction: representation, estimation, and testing." *Econometrica* 55(2):251–276.

Gandrud, Christopher, Laron K Williams and Guy D Whitten. 2016. "dynsim: Dynamic simulations of autoregressive relationships." *R package version 1.2.2* .

Gelman, Andrew, John B Carlin, Hal S Stern and Donald B Rubin. 2014. *Bayesian data analysis*. Vol. 2 Chapman and Hall/CRC Boca Raton, FL, USA.

Imai, Kosuke, Gary King and Olivia Lau. 2009. "Zelig: Everyone's statistical software." *R package version* 3(5).

Jennings, Will and Peter John. 2009. "The dynamics of political attention: Public opinion and the Queen's Speech in the United Kingdom." *American Journal of Political Science* 53(4):838–854.

Johansen, Søren. 1991. "Estimation and hypothesis testing of cointegration vectors in Gaussian vector autoregressive models." *Econometrica: Journal of the Econometric Society* 59(6):1551–1580.

Johansen, Soren. 1995. *Likelihood-based inference in cointegrated vector autoregressive models*. Oxford University Press.

Narayan, Paresh Kumar. 2005. "The saving and investment nexus for China: Evidence from cointegration tests." *Applied Economics* 37(17):1979–1990.

Pesaran, M Hashem, Yongcheol Shin and Richard J Smith. 2001. "Bounds testing approaches to the analysis of level relationships." *Journal of Applied Econometrics* 16(3):289–326.

Philips, Andrew Q. 2017. "Have your cake and eat it too? Cointegration and dynamic inference from autoregressive distributed lag models." *American Journal of Political Science* .

Philips, Andrew Q, Amanda Rutherford and Guy D Whitten. 2016*a*. "Dynamic pie: A strategy for modeling trade-offs in compositional variables over time." *American Journal of Political Science* 60(1):268–283.

Philips, Andrew Q, Amanda Rutherford and Guy D Whitten. 2016*b*. "dynsimpie: A command to examine dynamic compositional dependent variables." *Stata Journal* 16(3):662–677.

Phillips, Peter CB and Sam Ouliaris. 1990. "Asymptotic properties of residual based tests for cointegration." *Econometrica: Journal of the Econometric Society* pp. 165–193.

Swank, Duane and Sven Steinmo. 2002. "The new political economy of taxation in advanced capitalist democracies." *American Journal of Political Science* pp. 642–655.

Tomz, Michael, Jason Wittenberg and Gary King. 2003. "CLARIFY: Software for interpreting and presenting statistical results." *Journal of Statistical Software* 8(1):1–30.

Ura, Joseph Daniel. 2014. "Backlash and legitimation: Macro political responses to supreme court decisions." *American Journal of Political Science* 58(1):110–126.

Ura, Joseph Daniel and Christopher R Ellis. 2008. "Income, preferences, and the dynamics of policy responsiveness." *PS: Political Science & Politics* 41(4):785–794.

Whitten, Guy D and Laron K Williams. 2011. "Buttery guns and welfare hawks: The politics of defense spending in advanced industrial democracies." *American Journal of Political Science* 55(1):117–134.

Williams, Laron K and Guy D Whitten. 2011. "Dynamic simulations of autoregressive relationships." *Stata Journal* 11(4):577–588.

Williams, Laron K and Guy D Whitten. 2012. "But wait, there's more! Maximizing substantive inferences from TSCS models." *The Journal of Politics* 74(03):685–693.