



AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search
<http://ageconsearch.umn.edu>
aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

THE STATA JOURNAL

Editor

H. Joseph Newton
Department of Statistics
Texas A&M University
College Station, Texas 77843
979-845-8817; fax 979-845-6077
jnewton@stata-journal.com

Editor

Nicholas J. Cox
Department of Geography
Durham University
South Road
Durham DH1 3LE UK
n.j.cox@stata-journal.com

Associate Editors

Christopher F. Baum
Boston College

Nathaniel Beck
New York University

Rino Bellocco
Karolinska Institutet, Sweden, and
University of Milano-Bicocca, Italy

Maarten L. Buis
Tübingen University, Germany

A. Colin Cameron
University of California–Davis

Mario A. Cleves
Univ. of Arkansas for Medical Sciences

William D. Dupont
Vanderbilt University

David Epstein
Columbia University

Allan Gregory
Queen's University

James Hardin
University of South Carolina

Ben Jann
University of Bern, Switzerland

Stephen Jenkins
London School of Economics and
Political Science

Ulrich Kohler
WZB, Berlin

Frauke Kreuter
University of Maryland–College Park

Peter A. Lachenbruch
Oregon State University

Jens Lauritsen
Odense University Hospital

Stanley Lemeshow
Ohio State University

J. Scott Long
Indiana University

Roger Newson
Imperial College, London

Austin Nichols
Urban Institute, Washington DC

Marcello Pagano
Harvard School of Public Health

Sophia Rabe-Hesketh
University of California–Berkeley

J. Patrick Royston
MRC Clinical Trials Unit, London

Philip Ryan
University of Adelaide

Mark E. Schaffer
Heriot-Watt University, Edinburgh

Jeroen Weesie
Utrecht University

Nicholas J. G. Winter
University of Virginia

Jeffrey Wooldridge
Michigan State University

Stata Press Editorial Manager
Stata Press Copy Editors

Lisa Gilmore
Fred Iacoletti and Deirdre Skaggs

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go “beyond the Stata manual” in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the webpage

<http://www.stata-journal.com>

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index®
- Current Contents/Social and Behavioral Sciences®
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch®)
- Scopus™
- Social Sciences Citation Index®

Copyright Statement: The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible websites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata, Mata, NetCourse, and Stata Press are registered trademarks of StataCorp LP.

Dynamic simulations of autoregressive relationships

Laron K. Williams
University of Missouri
Columbia, MO
williamslaro@missouri.edu

Guy D. Whitten
Texas A&M University
College Station, TX
g-whitten@pols.tamu.edu

Abstract. This postestimation technique produces dynamic simulations of autoregressive ordinary least-squares models.

Keywords: st0242, dynsim, dynamic simulations, clarify, long-term effects, time series, lagged dependent variables

1 Overview

Social scientists have recently put increased emphasis on exploring the wide range of substantive inferences from their statistical models. One area where this new focus has been especially useful is analyses of ordinary least-squares (OLS) models with autoregressive processes. By specifying models with a lagged dependent variable, scholars implicitly assume that the full effects of an independent variable occur for more than just one time interval. For example, [De Boef and Keele \(2008\)](#) show that in models with autoregressive processes, the β 's only provide the short-term effect of the independent variable in that interval. Because the value of the dependent variable depends on its previous values, each independent variable also has a long-term effect. In an OLS model with a lagged dependent variable (and no lagged exogenous variables), the long-term effect of an independent variable X_1 is $\hat{\beta}/(1 - \hat{\phi})$, where $\hat{\beta}$ is the parameter estimate for X_1 and $\hat{\phi}$ is the parameter estimate for the lagged dependent variable. Fully exploring the substantive effects thus requires providing the long-term effects in addition to other quantities of interest, such as median and mean lag lengths ([De Boef and Keele 2008](#)).

This is consistent with the message from [King, Tomz, and Wittenberg \(2000\)](#) to provide quantities of interest that are substantively meaningful, easy to comprehend, and accompanied by measures of uncertainty. Quantities of interest (that is, predicted values, first-differences) and their measures of uncertainty (that is, standard errors, confidence intervals) can be calculated via analytical methods. [King, Tomz, and Wittenberg \(2000\)](#) suggest that the advances in computing power have ushered in an era for virtually all computers to engage in simulation-based techniques that previously were possible only for supercomputers. A more in-depth discussion is provided by [King, Tomz, and Wittenberg \(2000\)](#), so we will only briefly describe simulation methods and their usefulness.

Consider a very broad class of statistical models that can be summarized via two equations:

$$Y_i \sim f(\theta_i, \alpha), \quad \theta_i = g(X_i, \beta)$$

The first equation describes the stochastic component of the statistical model, or “the probability density (or mass) function that generates the dependent variable Y_i ($i = 1, \dots, n$) as a random draw from the probability density $f(\theta_i, \alpha)$ ” (King, Tomz, and Wittenberg 2000, 348). The characteristics that vary across observations i are contained in each parameter vector θ_i , while those that do not vary are in the ancillary parameter matrix α . The second equation is the systematic component, where θ_i varies according to the values of the independent variables (X_i). The functional form $g()$ “specifies how the explanatory variables and effect parameters get translated into θ_i ” (King, Tomz, and Wittenberg 2000, 348).

For OLS regression with normally distributed, homoskedastic errors (σ^2), we have

$$Y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = X_i\beta$$

where the systematic component ($X_i\beta$) is of the linear form

$$X_i\beta = \beta_0 + X_{i1}\beta_1 + X_{i2}\beta_2 + \dots$$

After fitting the model, many researchers stop once they interpret the coefficients and perhaps their standard errors. However, there are other quantities and measures of uncertainty that can aid the substantive interpretation of the causal relationships. Some of these measures of uncertainty are incomplete because they ignore both estimation uncertainty (which comes from estimating the α and the β 's) and fundamental uncertainty (which comes from having a stochastic component).

King, Tomz, and Wittenberg (2000) advocate statistical simulation for overcoming these problems and improving statistical inference. Much like survey sampling can tell us something about the population as a whole, the central limit theorem allows us to make inferences about probability distributions by simulating from the distribution. In the case of OLS, one can use point estimates and the variance–covariance matrix returned after estimation of a regression to randomly draw (simulate) the parameters from a multivariate normal distribution with mean equal to the parameters and variance equal to the variance–covariance matrix. For example, one can make n draws from the multivariate normal distribution, calculate the predicted value as the mean of those values, and calculate lower and upper confidence bounds using the resulting rank statistics.

King, Tomz, and Wittenberg's (2000) `clarify` package contains three separate Stata commands that made it easy for scholars to generate meaningful quantities of interest following a number of estimation procedures.¹ First, `estsimp` generates n draws of the

1. Our brief summary of simulation methods and the `clarify` package are modified to fit the purposes of this command. We encourage interested readers to explore these methods with the original sources.

point estimates and ancillary parameters (in the OLS case, σ^2). Next, the user can specify with `setx` the values of the independent variables (\mathbf{X}_C) according to a variety of numbering conventions. Finally, `simqi` generates a variety of quantities of interest for that scenario, including predicted values and first-differences.

We believe that scholars are neglecting some of the most meaningful substantive inferences from autoregressive models (for an example of this in an applied field, see [Williams and Whitten \[2008\]](#)). Specifically, the most effective way to observe the long-term effects of exogenous variables in autoregressive models is through simulating the predicted value (and confidence interval) for a given scenario over a given number of time intervals. The `dynsim` command uses the `clarify` package to create j long-term dynamic simulations of up to four user-specified scenarios for OLS models with a lagged dependent variable. It is dynamic because each additional iteration uses the predicted value from the previous iteration as the value of the lagged dependent variable in the next scenario. These simulations are then saved to a new dataset, which can be presented graphically or in a tabular fashion. The `forecast()` option calculates analytical standard errors that incorporate forecasting error into the dynamic simulations so that the confidence intervals are not underestimated. [Greene \(2008\)](#) demonstrates that the conditional forecasting error variance is

$$\text{Var}(\hat{y}_{T+F|T}) = \sigma^2 \{1 + \Psi(1)_{11} + \Psi(2)_{11} + \cdots + \Psi(F-1)_{11}\}$$

where $\Psi(i) = C^i j j' C^{i'}$

When we have a single lagged dependent variable (that is, $i = 1$), $\Psi(i) = C^i j j' C^{i'}$ collapses to $\Psi(i) = \gamma^2 \sigma^2 \forall i$, making the conditional forecast error variance

$$\text{Var}(\hat{y}_{T+F|T}) = \sigma^2 \left(1 + \sum_F \sigma^2 \gamma^2 \right)$$

Available options include specifying the values of an exogenous variable to change with each iteration. This exogenous variable, called the “shock” variable, is included by specifying either an additional dataset containing the shock variable or a Stata number list. The program also effectively interacts the shock variable with up to four other independent variables in the model.

The `dynsim` command automates a set of code to produce dynamic simulations. To illustrate generating predicted values and 95% confidence intervals for a scenario where all the independent variables are fixed at their means for 10 iterations, we present the following code:

```

. webuse grunfeld
. generate lag_invest = L.invest
. estsimp regress invest lag_invest mvalue kstock
. local it = 10                                /* 10 iterations */
. local ldv_value = 139.2                      /* Starting value for LDV */
. foreach i of numlist 1(1)`it' {
  2. setx lag_invest `ldv_value'
  3. quietly simqi, genpv(yhat_`i') pv
  4. quietly sum yhat_`i', meanonly
  5. local yhat = `r(mean)'
  6. _pctile yhat_`i', percentiles(2.5 97.5) /* 95% CI */
  7. display as result "Iteration = " `i'
  8. display as result "Predicted Value = " `yhat' " 95% C.I. =
> [" `r(r1)' " , " `r(r2)' "]"
  9. local ldv_value = `yhat'
}

```

2 Syntax

```

dynsim, ldv(varname) scen1(string) [scen2(string) scen3(string) scen4(string)
  n(integer) sig(cilevel) shock(varname) shock_data(filename)
  shock_num(numlist) modify(varlist) inter(varlist) saving(string)
  forecast(string) ]

```

3 Options

ldv(*varname*) specifies the name of the lagged dependent variable. **ldv**() is required.

scen1(*string*) specifies the values of the variables used to generate the predicted values when $t = 0$. At least one scenario must be given. The coding designation is identical to the options used in **clarify**'s **setx** (except for setting the values to a specific observation). At each subsequent iteration, these values will not change except for the value of the lagged dependent variable, the shock variable (**shock**()), if specified, and the interacted variable (**inter**()), if specified). **scen1**() is required.

scen2(*string*), **scen3**(*string*), and **scen4**(*string*) are optional and are only used if more than one scenario is desired. A maximum of four scenarios is allowed. These follow the same conventions as **scen1**().

n(*integer*) specifies the number of iterations (or time intervals) over which the program will generate the predicted value of the dependent variable. The default is **n**(10).

sig(*cilevel*) specifies the level of statistical significance of the confidence intervals (calculated via the percentile method). This value must be between 10 and 99.99.

shock(*varname*) allows the user to choose an independent variable (and its first **n**() values) and have the variable (and potentially different values) impact the scenarios

at each simulation. If this command is specified, the user must specify the `n()` shock values through either a dataset containing the variable (`shock_data()`) or a Stata *numlist* (`shock_num()`). For example, if `shock_data(shock.dta)` is specified, then `dynsim` will read the first `n()` values of the `shock()` variable. If `shock_num()` is specified as `0(10)100`, the value of the `shock()` variable will be 0 at time $t + 1$, 10 at time $t + 2$, and so on. The number of values assigned to the shock variable must exceed the number of simulations. If the shock variable is interacted with another variable in the model, the user must also specify the name of the modifying variable (`modify()`) and the interaction variable (`inter()`).

`shock_data(filename)` is one of two ways of specifying the shock values. This must either give the filename or be located in the working directory. The dataset used to get the shock variable (called the shock dataset) must have at least the number of iterations specified in `n()`, and it must contain a variable with the same name as the shock variable (`shock()`).

`shock_num(numlist)` is the second way of specifying the shock values. Any *numlist* is acceptable if it contains at least `n()` values.

`modify(varlist)` specifies up to four variables that modify the relationship between the shock variable and the dependent variable. If the shock variable interacts with another variable in the model, then `dynsim` automatically updates the value of the interaction to be the product of the shock and it modifies the variable at each interaction. If `inter()` is specified, then `modify()` must also be specified. The same number of variables must appear in `inter()` as in `modify()`. The variables must also appear in the same order as in `estsimp`.

`inter(varlist)` specifies up to four interaction variables. If `modify()` is specified, then `inter()` must also be specified. Much like `modify()`, the variables must appear in the same order as in `estsimp`.

`saving(string)` creates a dataset containing the predicted values and confidence intervals for each scenario. It automatically replaces any dataset with the same name, so change the name of the dataset used if you do not want it replaced.

`forecast(string)` produces confidence intervals based on one of four options for calculating the conditional variance of a forecast: The `ae` suboption analytically calculates the standard errors based on Enders' (2010, 81–89) formula for the conditional variance of the forecast. The `ag` suboption analytically calculates the standard errors based on Greene's (2008, 686–689) formula for the conditional variance of a forecast. The `se` and `sg` suboptions use the Enders and Greene formulas, respectively, but use the simulations to produce `n()` estimates of the conditional variance, which are then used to produce confidence intervals based on the percentile method.

4 Examples

To illustrate the features of `dynsim`, we use the [Grunfeld \(1958\)](#) dataset (`webuse grunfeld`) to fit the following model via OLS regression:

$$I_{it} = \alpha + \beta_1 I_{it-1} + \beta_2 F_{it} + \beta_3 C_{it} + \mu_{it}$$

where I_{it} denotes the real gross investment for firm i in year t , I_{it-1} is the firm's investment for the previous year, F_{it} is the real value of the firm, and C_{it} is the real value of the capital stock. The dataset contains information for 10 large U.S. manufacturing firms from 1935–1954 ([Baltagi 2008](#), 24).

Consider an OLS model that predicts a firm's real gross investment (`invest`) with its one-year lag (`lag_invest`), the firm's market value (`mvalue`), and the real value of the capital stock (`kstock`). We would first decide how many scenarios we want to display for our dynamic simulations and the values of each of the variables in those scenarios. Suppose that we want three different scenarios based on holding the firm's market value and value of the capital stock at its 5th percentile, mean, and 95th percentile. We also provide the starting value for the lagged dependent variable (in the `scen()` options), which here is the sample mean. We first use `estsimp` to simulate 1,000 draws of each coefficient. We then use `dynsim` to produce 20 dynamic simulations (`n(20)`) of each of the three scenarios. The resulting predicted values and 95% confidence intervals (the default, changeable via the `sig()` option) are saved to a dataset named `dynsim1.dta` in the working directory.

```
. estsimp reg invest lag_invest mvalue kstock
. dynsim, ldv(lag_invest) scen1(lag_invest mean mvalue p5 kstock p5) scen2(mean)
> scen3(lag_invest mean mvalue p95 kstock p95) n(20) saving(dynsim1)
```

Table 1 lists the dynamics of the first scenario. The predicted value at time t becomes the value of `lag_invest` for time $t + 1$ and so on. The starting values and the values for each iteration of the scenarios are returned in the `r(t0_s1)` and `r(xc_s1)` matrices, respectively. To view these values, list the returned matrices:

```
. matrix list r(t0_s1)
. matrix list r(xc_s1)
```

Table 1. Illustration of the dynamics for the first two forecasts of the 5th percentile scenario shown in figure 1

Time	Scenario	Predicted gross investment
t	Gross investment $_{t-1}$: 139.2*	
	Firm value: p5	106.1
	Capital stock: p5	
$t + 1$	Gross investment $_t$: 106.1	
	Firm value: p5	77.2
	Capital stock: p5	
$t + 2$	Gross investment $_{t+1}$: 77.2	
	Firm value: p5	51.9
	Capital stock: p5	

*The starting value for gross investment is the sample mean.

We can present the predicted values (and confidence intervals) in the new `dynsim` dataset in different ways. We have found that Stata’s `twoway rcap` command is a simple way of demonstrating the predicted values and uncertainty, though it is certainly reasonable to display them in a table. We graph the dynamic simulations in figure 1.

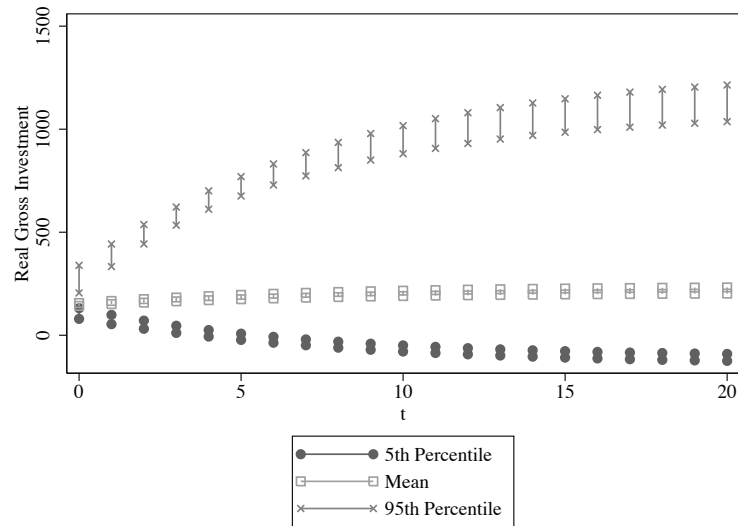


Figure 1. An example of a dynamic simulation of three scenarios

Dynamic simulations produce particularly helpful figures because of the variety of inferences we can make from them. First, we can use these simulations to make inferences about the long-term effects of variables. One can determine if the changes in predicted probabilities are statistically different across time and scenarios. For example, because the 95% confidence intervals with the 95th percentile scenario do not overlap at $t + 2$ compared with those at $t = 0$, we can infer that the 95th percentile scenario will have a statistically higher gross investment after two years than it did at time $t = 0$. We can make these types of inferences across scenarios as well. While there is no statistical difference between the 5th percentile and mean scenarios in the first period ($t + 1$), the confidence intervals in the second period ($t + 2$) do not overlap. This would suggest that the long-term effect of these variables on gross investment is quite strong.

We can incorporate forecasting error with analytically derived standard errors with the `forecast()` option. We allow the user to specify one of four options for calculating uncertainty for the forecast: `ag`, `ae`, `sg`, or `se`. The options are based on whether the user wants to calculate the confidence intervals analytically (`a`) or through simulation methods (`s`), and whether the user wants to use the Greene (2008) (`g`) or the Enders (2010) (`e`) formula for the conditional variance of a forecast.

Consider the simple model $y_t = \alpha_0 + \alpha_1 y_{t-1} + \epsilon_t$, where α_1 is the parameter for the lagged dependent variable. We calculate the error variance with the formula $\sigma^2 = \text{RSS} / (N - 2)$, where RSS is the residual sum of squares and N is the number of observations. Enders (2010) shows that we can easily generate the conditional variance of a forecast of j periods with the following formula:

$$\text{Var}\{e_t(j)\} = \sigma^2 \left(1 + \alpha_1^2 + \alpha_1^4 + \alpha_1^6 + \dots + \alpha_1^{2(j-1)} \right)$$

The forecast error variance at $j = 1$ is σ^2 , the forecast error variance at $j = 2$ is $\sigma^2(1 + \alpha_1^2)$, and so on. Greene (2008) provides a slightly different formula:²

$$\text{Var}\{e_t(j)\} = \sigma^2 \left(1 + \sum_F \sigma^2 \alpha_1^2 \right)$$

Note that in both formulas, while the size of the confidence intervals increases at each additional forecast period, the size of the confidence intervals will be the same across scenarios for the same iteration.

Another benefit of `estsimp` is that the `n()` draws for the values of α_1 and σ^2 are easily accessible. One can take the mean value of the `n()` draws for both α_1 and σ^2 to calculate the analytical standard errors, and then use the desired level of statistical significance (`sig()`) to produce analytical confidence intervals. Or one can take advantage of the information present in the `n()` draws to calculate `n()` estimates of the conditional variance of the forecast, and then use the percentile method to calculate simulation-based forecast errors.

2. This is the case when the lagged dependent variable is limited to a one-year lag.

Assuming that the user has already `estsimpd` the data, one can produce the dynamic simulations shown in figure 2 with the following code.³ We simplify the code so that there is only one scenario with all values held at their mean. To demonstrate that the analytically derived and simulation-based confidence intervals are similar, we choose the `forecast(se)` option (simulation-based standard errors via Enders' formula) in the first `dynsim` and the `forecast(ae)` option (analytically derived standard errors via Enders' formula) in the second.

```
. dynsim, ldv(lag_invest) scen1(mean) n(20) sig(90) saving(dynsim_se) forecast(se)
. dynsim, ldv(lag_invest) scen1(mean) n(20) sig(90) saving(dynsim_ae) forecast(ae)
```

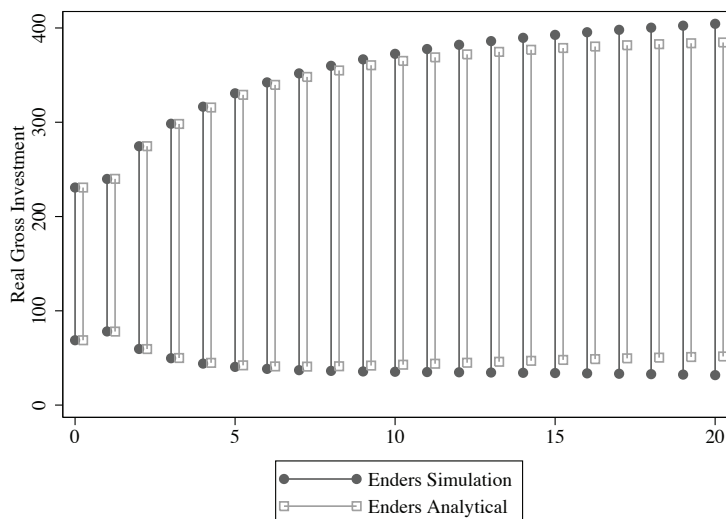


Figure 2. An example of a dynamic simulation with confidence intervals based on analytical and simulation-based forecasting errors

Using the `forecast()` option widens the confidence intervals to represent the increased uncertainty of future forecasts. In our experience, however, the different options give similar substantive answers, so the choice is one of personal preference.

Figure 3 illustrates how an exogenous shock—in this case, the firm's market value (`mvalue`)—influences a firm's real gross investment over two scenarios for 15 iterations. To produce the dynamic simulations, we first create a dataset (`grunfeldshock`) containing a series of values to use as the exogenous shock (`mvalue`). Then we use `dynsim` to simulate those values.

3. We slightly jitter the confidence intervals so that it is easier to compare their sizes.

```

. preserve
. keep if company == 1
. keep mvalue
. save grunfeldshock, replace
. restore
. dynsim, ldv(lag_invest) scen1(lag_invest mean mvalue 1000 kstock p5)
> scen2(lag_invest mean mvalue 1000 kstock p95) n(15) saving(dynsim3)
> shock(mvalue) shock_data(grunfeldshock.dta)

```

Note that we now include two additional options: the `shock()` option provides the name of the exogenous variable, and the `shock_data()` option gives the name of the dataset containing the location of the values of the exogenous variable. Here we use the values of `mvalue` for the first company in the dataset. The value of `mvalue` for the initial calculation ($t = 0$) is provided in the `scen()` options (here `mvalue = 1000`). One can also specify the values of the shock variable with a Stata *numlist* (that is, `0(10)100`).

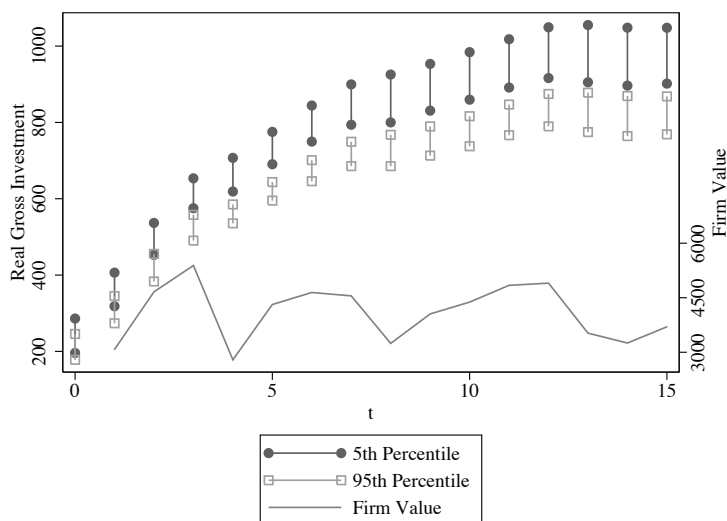


Figure 3. An example of a dynamic simulation with the inclusion of a shock variable

The breadth of our inferences is increased by including a shock variable. In this example, we can examine whether the 5th percentile scenario responds to changes in the shock variable (`mvalue`) differently than the 95th percentile scenario. One can also use this figure to determine how long it takes for the predicted values for each scenario to return to their preshock values.

Interactive relationships between the shock variable and the other independent variables can also be incorporated into the command with the use of the `inter()` option. Up to four modifying variables (`modify()`) and interactive variables (`inter()`) can be spec-

ified, though these variables must be in the same order as they appear in the `estsimp` command.

In the next example, we interact the shock variable (`mvalue`) with another exogenous variable from the model (`kstock`), with the result displayed in figure 4:

```
. generate z = mvalue * kstock
. estsimp reg invest lag_invest mvalue kstock z
. dynsim, ldv(lag_invest) scen1(lag_invest mean mvalue 1000 kstock p5)
> scen2(lag_invest mean mvalue 1000 kstock p95) n(15) saving(dynsim3)
> shock(mvalue) shock_data(grunfeldshock.dta) modify(kstock) inter(z)
```

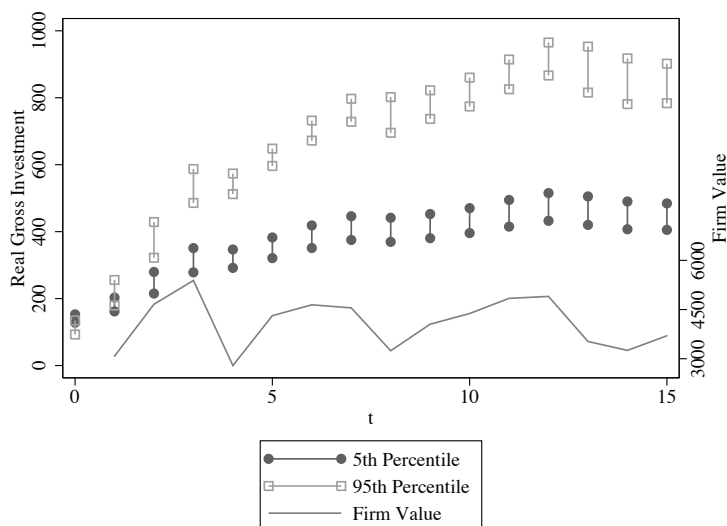


Figure 4. An example of a dynamic simulation where a shock variable interacts with an exogenous variable

This is a nice way to illustrate that the effects of an independent variable on the dependent variable depend on the values of the shock variable. In this case, the difference between the 5th percentile and 95th percentile scenarios becomes more pronounced when there are high values of the shock variable (for instance, at $t + 3$). It is also interesting to note that while the predicted value for the 5th percentile varies little over the span of 15 iterations regardless of the value of the exogenous shock, the 95th percentile scenario is much more variable.

5 References

- Baltagi, B. H. 2008. *Econometric Analysis of Panel Data*. 4th ed. New York: Wiley.
- De Boef, S., and L. Keele. 2008. Taking time seriously: Dynamic regression. *American Journal of Political Science* 52: 184–200.
- Enders, W. 2010. *Applied Econometric Time Series*. 3rd ed. Hoboken, NJ: Wiley.
- Greene, W. H. 2008. *Econometric Analysis*. 6th ed. Upper Saddle River, NJ: Prentice Hall.
- Grunfeld, Y. 1958. The Determinants of Corporate Investment. PhD thesis, University of Chicago.
- King, G., M. Tomz, and J. Wittenberg. 2000. Making the most of statistical analyses: Improving interpretation and presentation. *American Journal of Political Science* 44: 347–361.
- Williams, L. K., and G. D. Whitten. 2008. But wait, there's more! Maximizing substantive inferences from TSCS models.
<http://people.tamu.edu/~lkw5391/Williams-Whitten-But Wait! 8-26-08.pdf>.

About the authors

Laron K. Williams is an assistant professor of political science at the University of Missouri. His research examines the comparative politics of advanced democracies.

Guy D. Whitten is an associate professor of political science and the director of the European Union Center at Texas A&M University. His research focuses on comparative political economy and political behavior.