Dynamic Simulations of Autoregressive Relationships

Abstract: This post-estimation technique produces dynamic simulations of autoregressive OLS models.

Keywords: 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 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 period. For example, de Boef and Keele (2008) show that, in models with autoregressive processes the βs only provide the short-term effect of that variable in that period. Since 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 X_1 is $\frac{\hat{\beta}}{1-\hat{\phi}}$, where $\hat{\beta}$ is the parameter estimate for the independent variable of interest (X_1) and $\hat{\phi}$ is the parameter estimate for the lagged dependent variable. A full exploration of 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 call made by King, Tomz and Wittenberg (2000) to provide quantities of interest that are substantively meaningful, easy to comprehend, and that are accompanied by measures of uncertainty. Certainly, quantities of interest (i.e., predicted values, first differences, etc) and their measures of uncertainty (i.e., standard errors, confidence intervals, etc) 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 were previously possible only for supercomputers. A much more in-depth discussion is provided in King, Tomz and Wittenberg (2000), so we will only briefly describe simulation methods and their usefulness for our purposes.

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

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

The first equation describes the stochastic component of the statistical model, or "the probability density (or mass) function that generates the dependent variable $Y_i (i = 1, \ldots, n)$ as a random draw from the probability density $f(\theta_i, \alpha)$" (King, Tomz and Wittenberg 2000: 348). The characteristics that vary across observation, i, are contained in the 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). In the case of ordinary least squares regression with normally distributed, homoskedastic errors (σ^2), we have:

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

where the systematic component ($X_i\beta$) is of the linear form $X_i\beta = \beta_0 + X_{i1}\beta_1 + X_{i2}\beta_2 + \ldots$. After estimating the model, many researchers stop once they interpret the coefficients and perhaps their standard errors.

However, there are a number of other quantities as well as 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 βs) and fundamental uncertainty (which comes from having a stochastic component).

King, Tomz and Wittenberg (2000) advocate statistical simulation as a means of 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 (drawing random numbers) 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 program 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 point estimates and ancillary parameters (in the OLS case, σ^2). Next, the user can specify the values of the independent variables ($\mathbf{X_C}$) according to wide variety of numbering conventions with `setx`. Finally, `simqi` generates a wide variety of quantities of interest for that scenario, including predicted values, first differences, etc.

We believe that scholars are neglecting some of the most meaningful substantive inferences from autoregressive models. More specifically, in autoregressive models the most effective way of observing the long-term effects of exogenous variables is through simulating the predicted value (and confidence interval) for a given scenario over a given number of time periods. The `dynsim` command makes use of 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 in the sense that 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 data set that 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 (2003) demonstrates that the conditional forecasting error variance is:

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

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

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

$$var[\hat{y}_{T+F|T}] = \sigma^2[1 + \sum_F \sigma^2\gamma^2].$$

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 either specifying an additional data set containing the shock variable, or via a Stata number list. The program also effectively deals with interacting 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. As a simple illustration of generating predicted values and 95% confidence intervals for a scenario where all the independent variables are held at their means for 10 iterations, we present the following code:

```
.webuse grunfeld, clear
.gen lag_invest = L.invest
```

[1] Our brief summary of simulation methods in general and the Clarify program in particular are modified to fit the purposes of the this command. We strongly encourage interested readers to explore these methods with the original sources.

```
.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' {
.   setx lag_invest `ldv_value'
.   quietly simqi, genpv(yhat_`i') pv
.   quietly sum yhat_`i', meanonly
.   local yhat = `r(mean)'
.   _pctile yhat_`i', percentiles(2.5 97.5)   /* 95% CI */
.   display as result "Iteration = " `i'
.   display as result "Predicted Value = " `yhat' " 95% C.I. = [" `r(r1)' ", " `r(r2)' "]"
.   local ldv_value = `yhat'
.}
```

2 Syntax

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

3 Options

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

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 (with the exception of setting the values to a specific observation). At each subsequent iteration, these values will stay the same, except for the value of the lagged dependent variable, the shock variable (shock, if specified) and the interacted variable (inter, if specified).

scen2-4(*string*) are optional and are only used if more than one scenario is desired. A maximum of 4 scenarios is allowed. It follows the same conventions as scen1.

n(*integer*) specifies the number of iterations (or time period) over which the program will generate the predicted value of the dependent variable. The default is 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*) this option 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 data set containing the variable (shock_data) or a Stata numlist (shock_num). 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(*string*) is one of two ways of specifying the shock values. This must give the file name, or must be located in the working directory. The data set used to get the shock variable (called the shock data set), 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, as long as it contains at least `n` values.

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

`inter`(*varlist*) is the name of up to four interaction variables. If `modify` is specified, this must also be specified. It also must be in the same order as `estsimp`.

`saving`(*string*) is the name of the new data set created that contains the predicted values and confidence intervals for each scenario. It automatically replaces any data set with the same name, so change the name of the saving data set 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` calculates the standard errors analytically based on Enders' (2004: 79-81) formula for the conditional variance of the forecast. Choosing `ag` calculates the standard errors analytically based on Greene's (2003: 571-580) formula for the conditional variance of a forecast. The `se` and `sg` options 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) data set (`webuse grunfeld`) to estimate the following model via ordinary least squares regression:

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

where I_{it} denotes 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 data set contains information for 10 large US manufacturing firms from 1935-1954 (Baltagi 2001).

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 wanted to display for our dynamic simulations and the values of each one 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, respectively. We also provide the starting value for the lagged dependent variable (in the `scen` options), which in this case is the sample mean. We first use `estsimp` to simulate 1000 draws of each coefficient. We then use `dynsim` to produce 20 dynamic simulations (`n(20)`) of each of the three scenarios, with the resulting predicted values and 95% confidence intervals (the default, changeable via the `sig` option) saved to a data set named "dynsim1" 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 shows 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 Capital Stock: p5	**106.1**
$t+1$	Gross Investment$_t$: **106.1** Firm Value: p5 Capital Stock: p5	**77.2**
$t+2$	Gross Investment$_{t+1}$: **77.2** Firm Value: p5 Capital Stock: p5	**51.9**

**Note:* the starting value for gross investment is the sample mean.

We can present the predicted values (and confidence intervals) in the new `dynsim1` data set in a variety of ways. We have found that Stata's `twoway rcap` is a simple way of demonstrating the predicted values and uncertainty, though it is certainly reasonable to display them in a table. We present graphical depictions of the dynamic simulations in Figure 1.

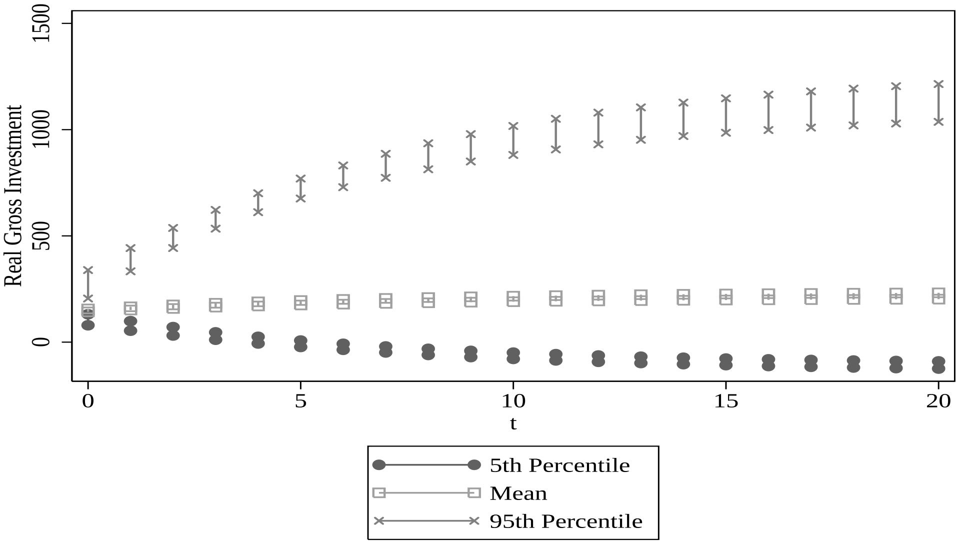

Figure 1: An Example of a Dynamic Simulation of Three Scenarios

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 whether the changes in predicted probabilities are statistically different across time and/or scenarios. For example, since the 95% confidence intervals for the 95th Percentile scenario do not overlap at $t+2$ compared to to those at $t=0$, we can infer that after two years, the 95th Percentile scenario will have a statistically higher gross investment 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 the 5th Percentile and Mean scenarios at $t+1$, in the second period the confidence intervals 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 (2003) (`g`) or Enders (2004) (`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 following formula, $\sigma^2 = \frac{RSS}{N-2}$, where RSS is the residual sum of squares and N is the number of observations. Enders (2004) shows that we can easily generate the conditional variance of a forecast of j periods with the following formula: $\text{Var}[e_t(j)] = \sigma^2[1+\alpha_1^2+\alpha_1^4+\alpha_1^6+\cdots+\alpha_1^{2(j-1)}]$. 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 (2004) provides a slightly different formula: $Var[e_t(j)] = \sigma^2[1+\sum_F \sigma^2\alpha_1^2]$.[2] It is important to 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 then 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.

Assuming that the user has already `estsimp` 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 `se` option (simulation-based standard errors a la Enders) in the first `dynsim` and the `ae` option (analytically-derived standard errors a la Enders) in the second.

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

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

Using the `forecast` option widens the confidence intervals to represent the increased uncertainty regarding future forecasts. It is our experience, however, that 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 data set (`grunfeldshock`) containing a series of values to use as the exogenous shock (`mvalue`), and second, use `dynsim` to simulate those values:

```
.preserve
.   keep if company == 1
.   keep mvalue
.   save grunfeldshock, replace
.restore
```

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

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

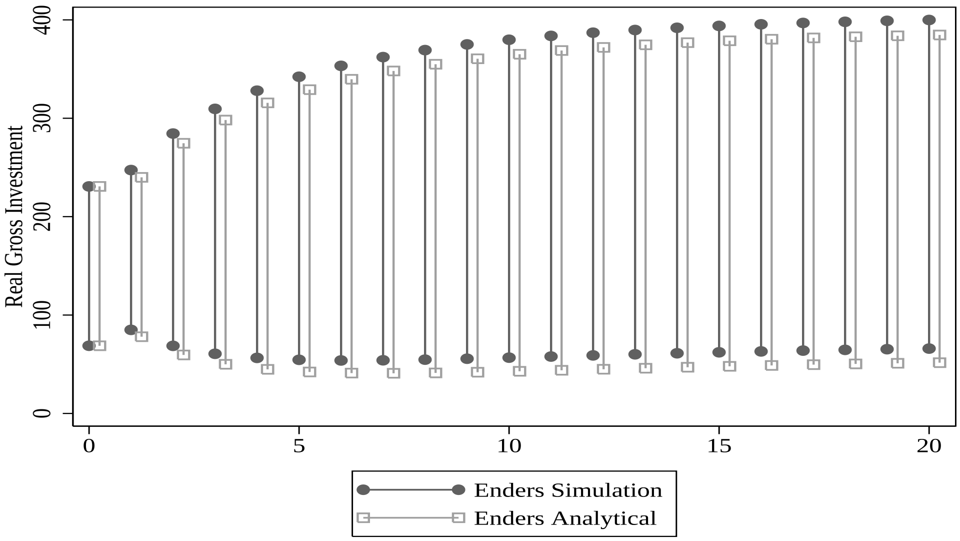

Figure 2: An Example of a Dynamic Simulation with Confidence Intervals Based on Analytical and Simulation-Based Forecasting Errors

```
.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` gives the name of the data set containing the location of the values of the exogenous variable. In this case, we use the values of `mvalue` for the first company in the data set. The value of `mvalue` for the initial calculation ($t = 0$) is provided in the `scen` options (in this case, `mvalue`=100). One can also specify the values of the shock variable with a Stata numlist (i.e., `0(10)100`).

The breadth of our inferences is increased by the inclusion of 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. One can also use this figure to determine how long it takes for the predicted values for each scenario to return to their pre-shock 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 specified, 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`).

```
.gen 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)
```

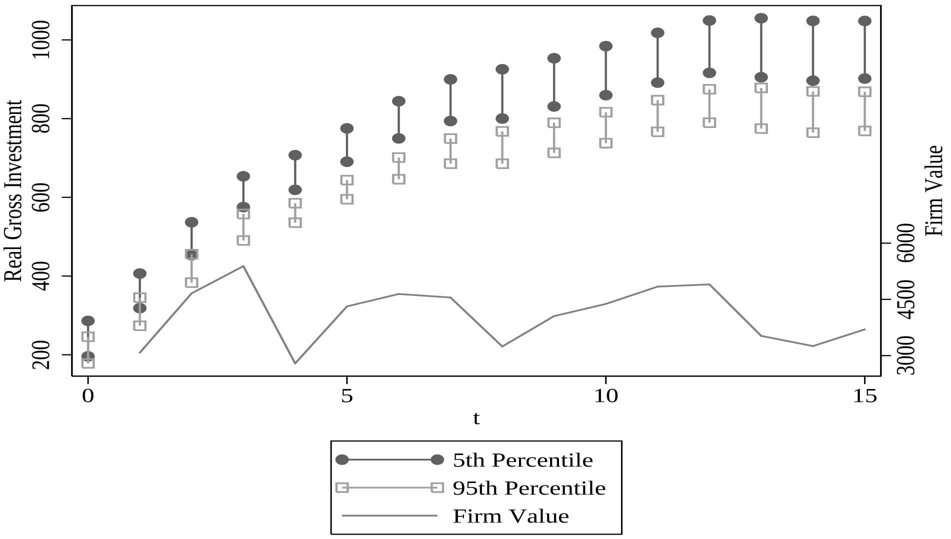

Figure 3: An Example of a Dynamic Simulation with the Inclusion of a Shock Variable

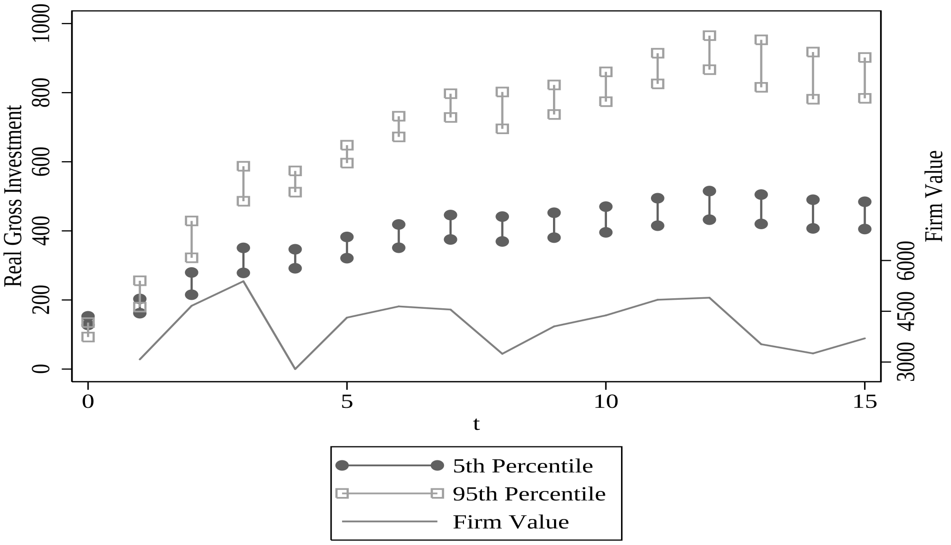

Figure 4: An Example of a Dynamic Simulation Where a Shock Variable Is Interacted with an Exogenous Variable

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 does not vary much over the span of 15 iterations regardless of the value of the exogenous shock, the 95th Percentile scenario is much more variable.

References

Baltagi, Badi. 2001. *Econometric Analysis of Panel Data.* Chichester, UK: Wiley and Sons.

deBoef, Suzanna & Luke Keele. 2008. "Taking Time Seriously: Dynamic Regression." *American Journal of Political Science* 52:184–200.

Enders, Walter. 2004. *Applied Econometric Time Series.* 2nd ed. Singapore: John Wiley and Sons.

Greene, William H. 2003. *Econometric Analysis.* 5th edition ed. New York: Prentice Hall.

Grunfeld, Yehuda. 1958. The Determinants of Corporate Investment PhD thesis University of Chicago.

King, Gary, Michael Tomz & Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." *American Journal of Political Science* 44:347–361.