The Stata Journal (2016)
16, Number 3, pp. 662–677

dynsimpie: A command to examine dynamic compositional dependent variables

Andrew Q. Philips
Department of Political Science
Texas A&M University
College Station, TX
aphilips@pols.tamu.edu

Amanda Rutherford
School of Public and Environmental Affairs
Indiana University
Bloomington, IN
aruther@indiana.edu

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

Abstract. In this article, we adapt the modeling strategy proposed by Philips, Rutherford, and Whitten (2016, *American Journal of Political Science* 60: 268–283) and create a user-friendly Stata command, `dynsimpie`. This command requires the installation of the `clarify` package of Tomz, Wittenberg, and King (2003, *Journal of Statistical Software* 8(1): 1–30) and uses the commands in the `clarify` package to produce estimates from models of compositional dependent variables over time. Users can also examine how counterfactual shocks play through the system with graphs that are easy to interpret. We illustrate this with a model of voter support for the three dominant political parties in the UK.

Keywords: st0448, dynsimpie, dynamic composition, counterfactual shocks

1 Overview

While compositional variables are central to many theories in the social sciences and elsewhere, they tend to be difficult to model. For instance, in examining support for a particular political party, researchers find that gains in one party's proportion of support must come at the expense of support for at least one other party. When there are only two categories, modeling this type of tradeoff is straightforward. Yet when there are three or more categories, it becomes increasingly difficult.

To address this issue, researchers have developed an entire class of models. Advances in the analysis of compositional data can be largely attributed to Aitchison (1986, 1982, 1983); this work has been applied in the context of geology, medicine (Hoffman and Uauy 1992), and, more recently, political science (Katz and King 1999; Tomz, Tucker, and Wittenberg 2002; Pawlowsky-Glahn and Buccianti 2011). Because compositional variables must sum to one, any increase (or decrease) in one component of the composition must be offset by a corresponding decrease (or increase) in one or more of the other components. This makes them particularly difficult to study with conven-

 st0448

tional statistical models.[1] However, by using a log-ratio transformation (see additional explanation in Aitchison [1986]), we free the variables from the sum-to-one constraint. This allows complex compositional models to be fit using standard multivariate normal or multivariate additive-logistic Student t distributions.

Philips, Rutherford, and Whitten (2016) (see also Philips, Rutherford, and Whitten [2015]) extend modeling compositional dependent variables in several important ways. First, they address the paucity of dynamic compositional models by proposing an error-correction model of compositional variables to gain inferences about both long- and short-run effects. This is particularly important because many interesting compositional dependent variables in the social sciences, such as budgets, diversified stock portfolios, or levels of party support, are inherently dynamic. Second, Philips, Rutherford, and Whitten (2016) greatly simplify the presentation of complex model results through graphical depictions of their dynamic simulations.

We turn the methods described in Philips, Rutherford, and Whitten (2016) into a user-friendly command, `dynsimpie`, that can be implemented in Stata. We apply the modeling strategy suggested by Tomz, Wittenberg, and King (2003) and use stochastic simulations to create graphs that show the predicted proportions of each category in the composition, along with associated measures of confidence. Dynamic simulations are growing in popularity in political science because of their ease of interpretation and clarity of inference (Williams and Whitten 2011, 2012; Whitten and Williams 2011). With `dynsimpie`, users can easily estimate and graph their own dynamic simulations of compositional dependent variables.

In brief, our model estimation strategy is as follows.[2] Let j components of a total number of J categories of a dependent variable y over time t be expressed as a proportion, such that $\sum_{j=1}^{J} = 1$. We then calculate $J-1$ compositions, s_{tj}, by taking the log ratio between category y_{tj} and some arbitrary baseline category y_{t1} (where $y_{tj} \neq 1$).[3] This is known as the log-ratio transformation (Aitchison 1986).

$$s_{tj} = \ln(\frac{y_{tj}}{y_{t1}}) \quad \forall j \neq 1$$

Next, we use an error-correction framework to model the relationship between a vector of exogenous variables, $\mathbf{x}_t$, and our log-ratio compositions as a system of equations given by the following,

$$\Delta s_{tj} = \beta_{0j} - \alpha_j s_{jt-1} + \beta_{Lj}\mathbf{x}_{t-1} + \beta_{Sj}\Delta\mathbf{x}_t + \Sigma_{tj}$$

where the change in the logged ratio of dependent variable category j (for $j \neq 1$) relative to baseline category $j = 1$ is a function of a constant, β_{0j}, a lag of dependent variable

1. More formally, compositional variables over time have four characteristics. First, each component must be bounded by zero and one. Second, the components must sum to 1. Third, a change in a single component is bounded by zero and one. Finally, the sum of the changes at a single time point t must sum to zero.
2. See Philips, Rutherford, and Whitten (2016) for a more extensive discussion.
3. The choice of the baseline category $j = 1$ is arbitrary because, as we will see, all s_{tj} are later untransformed back into their original y_{tj} and we can also retrieve the predicted values for y_{t1}.

s_{jt-1}, the lag and first difference of $\mathbf{x}_t$, and a matrix of stochastic disturbance terms, Σ_{tj}, that may be correlated across the system of equations. Because of this potentially correlated error structure, we fit all error-correction models at once in a seemingly unrelated regression to gain more efficient estimates. While the error-correction model is ideal for unit-root series [typically first-order integration, I(1), in most social science applications] that have a cointegrating relationship, or where all series are stationary [that is, I(0)] (De Boef and Keele 2008), we stress that users should conduct the proper tests for unit roots and cointegration before running `dynsimpie`.[4]

Because numerical interpretation of the resulting models is relatively difficult, the command instead lets users conduct their own simulations of substantively meaningful inferences (King, Tomz, and Wittenberg 2000; Williams and Whitten 2012). To do this, we use the commands in the `clarify` package to produce 1,000 sets of parameter estimates (Tomz, Wittenberg, and King 2003). The `clarify` package uses Monte Carlo simulations to present statistical outputs that substantively illustrate interesting quantities such as predicted values and first differences; installation of `clarify` is required to use `dynsimpie`. Starting with each variable at its sample mean at time $t = 1$ (with the long-run vector of exogenous variables set to their sample means, and the differenced exogenous variables set to zero), the command calculates predicted values for each $J-1$ composition. These values then move back into the equation at time $t = 2$ (through the lagged dependent variable), and the process is repeated. At a user-specified time point, a counterfactual "shock" may be introduced—which affects our model by giving one of the independent variables a negative or positive short-run change for a single time point. Then, the process readjusts to a new equilibrium at subsequent time points. The command changes the predicted log-ratio compositional values back into predicted proportions by using the following untransformation:

$$\widehat{Y}_{tj} = \frac{e^{\widehat{S}_{tj}}}{1 + \sum_{j=2}^{J} e^{\widehat{S}_{tj}}} \quad \forall j \neq 1$$

For the baseline category ($j = 1$), the untransformation is given as

$$\widehat{Y}_{tj} = \frac{1}{1 + \sum_{j=2}^{J} e^{\widehat{S}_{tj}}}$$

In addition to predicted values, upper and lower confidence intervals are calculated via the percentile method at a user-specified level of confidence (the default is 95%). The predicted composition values and associated confidence intervals are saved to a dataset so that they can be easily plotted.[5]

4. For a Monte Carlo investigation into the consequences of violating this suggestion, see the Supplemental Materials in Philips, Rutherford, and Whitten (2016).
5. This is the same order of operations to handle compositional data as given in the `clarify` documentation (Tomz, Wittenberg, and King 2003, 21).

2 Syntax

`dynsimpie` *indepvars* [*if*] [*in*], `dvs(`*varlist*`)` `shockvar(`*varname*`)` `shock(`#`)`
[`time(`#`)` `graph` `saving(`*string*`)` `range(`#`)` `sig(`#`)` `dummy(`*varlist*`)`
`dummyset(`*numlist*`)` `shockvar2(`*varname*`)` `shock2(`*numlist*`)`
`shockvar3(`*varname*`)` `shock3(`*numlist*`)` `notable` `nosave`]

indepvars is a list of independent variables to be included in the model. The final list will be one variable less than the total number of desired independent variables, because one variable must always be specified in `shockvar()` (see below). As shown in the examples, these variables need to be specified only in levels—`dynsimpie` automatically transforms them into the lag and first difference needed for estimation in the error-correction model.

3 Options

`dvs(`*varlist*`)` is a list of the compositional dependent variables to be fit in the model. Each of these should be expressed as either proportions (thus summing to 1) or percents (summing to 100). `dynsimpie` will issue an error message if neither of these criteria is met. The command takes the log of the proportion of each category relative to the proportion of an arbitrary "baseline" category; for example, if there were J dependent variables in `dvs(`*varlist*`)`, `dynsimpie` would create $J-1$ categories of $s_{tj} = \ln(y_{tj}/y_{tJ})$, where the Jth category is the baseline. `dvs()` is required.

`shockvar(`*varname*`)` is the independent variable, not included in *varlist*, that experiences some counterfactual one-period shock as specified in `shock()` at time t specified in `time()`. Because this is within an error-correction framework, the shock first affects the first-differenced `shockvar()` at time t for one time period, then moves into the lagged `shockvar()`. `shockvar()` is required.

`shock(`#`)` is the amount to shock the independent variable specified in `shockvar()` at time t specified in `time()`. `shock()` is required.

`time(`#`)` is the time that the variable specified in `shockvar()` experiences a one-period shock. The default is `time(10)`.

`graph` displays a plot of the simulated output. The predicted proportion of each of the compositional dependent variables is plotted against time, along with the associated confidence intervals.

`saving(`*string*`)` specifies the name of the dataset that `dynsimpie` will save the results to. By default, the results are saved as `dynsimpie_results.dta`. This dataset contains a time variable, the midpoints, and the upper and lower confidence intervals for each dependent variable. This is commonly used for graphing the dynamic simulation results.

`range(#)` gives the length of the scenario to simulate. By default, 20 time periods are simulated. `range(#)` must always be more than the `time(#)` at which the shock occurs.

`sig(#)` specifies the level of confidence associated with the calculation of the confidence intervals. The default is `sig(95)` for 95% confidence intervals.

`dummy(`*varlist*`)` specifies a list of dummy variables in the model.

`dummyset(`*numlist*`)` specifies alternative values for each of the dummy variables specified in `dummy()` to be used throughout the simulation. By default, each of the dummy variables in `dummy()` will be set to 0 throughout the simulation.

`shockvar2(`*varname*`)` allows for an additional shock to take place at time t. As with `shockvar()`, this variable cannot be included in *varlist*.

`shock2(`*numlist*`)` is the amount by which to shock `shockvar2(`*varname*`)`.

`shockvar3(`*varname*`)` allows for an additional shock to take place at time t. As with `shockvar()`, this variable cannot be included in *varlist*.

`shock3(`*numlist*`)` is the amount by which to shock `shockvar3(`*varname*`)`.

`notable` suppresses the automatic generation of the seemingly unrelated regression results. By default, a table of estimates is shown.

`nosave` suppresses saving the results. By default, the results are saved as either `dynsimpie_results.dta` or a user-specified name in `saving()`.

4 Examples

To illustrate the functions of `dynsimpie`, we use data on UK party support from Philips, Rutherford, and Whitten (2016).[6] They hypothesize that support for each of the three largest political parties (Labour, Conservatives, and Liberal Democrats) is a function of party identification, evaluation of the party leaders, national retrospective evaluations, and evaluations of which party is the best manager of the most important issue facing the country. Thus the dependent variable is a $j = 3$ composition of the proportion of support for each party that sums to 1. This is shown in figure 1. Summary statistics for each party are shown in table 1.

6. This dataset is included in the `dynsimpie` package download.

Table 1. Summary statistics for UK party support

Variable	Obs.	Mean	Standard Deviation	Minimum	Maximum
Conservatives	71	0.43	0.06	0.33	0.58
Liberal Democrats	71	0.21	0.04	0.13	0.33
Labour	71	0.36	0.05	0.25	0.46

```
. use uk_ajps
. twoway line Con ts || line Ldm ts || line Lab ts,
> legend(order(1 "Conservatives" 2 "Liberal Democrats" 3 "Labour"))
> ytitle("Proportion of Support")
```

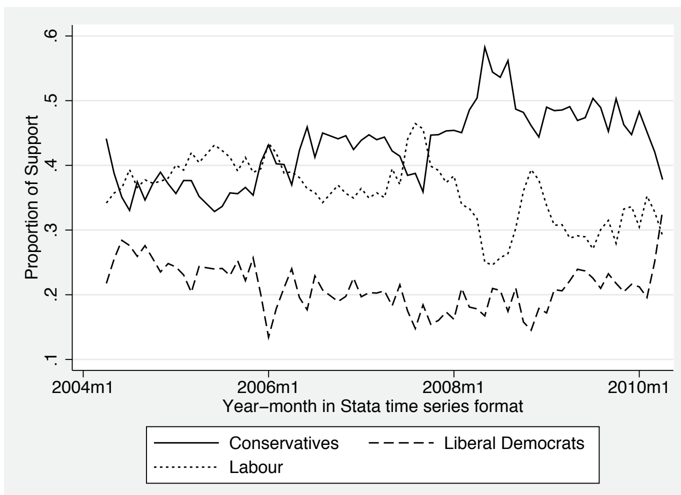

Figure 1. Party support during the "new Labour" period

According to Philips, Rutherford, and Whitten (2016), the next step is to conduct unit-root testing to determine whether an error-correction model is appropriate to use. Recall that error-correction models are appropriate only in cases where all the variables are stationary or where all variables are nonstationary and appear to be in a cointegrating relationship. To see whether this is the case for the UK example, table 2 shows the results of the Dickey–Fuller and Phillips–Perron unit-root tests for all the undifferenced and differenced series. Because we seldom reject the null hypothesis of nonstationarity for the undifferenced series and always reject it for the differenced series, we can conclude that all variables appear to be nonstationary.

Table 2. Unit-root tests for UK party support

	Aug. Dickey–Fuller		Phillips–Perron	
	Undifferenced	Differenced	Undifferenced	Differenced
Dependent Variables:				
Conservatives	−2.32	−9.55*	−2.27	−9.64*
Labour	−2.21	−7.94*	−2.41	−7.95*
Lib. Dems.	−2.88*	−9.18*	−2.72	−9.47*
Independent Variables:				
Party ID	−3.09*	−13.58*	−2.84	−14.81*
Labour Leader Evaluation	−2.21	−8.98*	−2.52	−8.94*
Conservative Leader Evaluation	−2.87*	−10.50*	−2.72	−10.87*
Lib. Dem. Leader Evaluation	−1.15	−7.73*	−0.86	−7.66*
Natl. Retrospective Evaluation	−0.90	−7.89*	−1.08	−8.04*
Labour "Best Manager" of Economy	−2.75	−9.25*	−2.85	−9.16*

Note: Augmented Dickey–Fuller $Z(t)$ test statistics. * $p < 0.05$. Phillips–Perron $Z(t)$ test statistics. * $p < 0.05$.

Our next step is to test for evidence of cointegration. Using the common Engle–Granger (1987) approach to cointegration, we see that there appears to be evidence that all three dependent variables (Labour, Conservatives, and Liberal Democrats) are in a cointegrating relationship with the independent variables.[7] Therefore, the error-correction model is appropriate to use in this example.

The model of UK party support is now ready to be fit and simulated using `dynsimpie`. We list the five independent variables and those who view Labour as the "Best Manager" of the economy in the required `shockvar()` option. The three dependent variables are listed under the required `dvs()` option. For our dynamic simulation, we will show the estimated effects of a one standard-deviation increase (+0.054) of those who think Labour is the best manager of the most important issue at time $t = 9$.

```
. dynsimpie all_pidW all_LabLeaderEval_W all_ConLeaderEval_W
> all_LDLeaderEval_W all_nat_retW, dvs(Con Ldm Lab) t(9)
> shock(0.054) shockvar(all_b_mii_lab_pct) graph
No range specified; default to t=20
Seemingly unrelated regression

Equation          Obs  Parms        RMSE    "R-sq"       chi2        P

D_Ldm_Con          68     13    .1051191    0.7159     183.70   0.0000
D_Lab_Con          68     13    .0584838    0.7902     273.01   0.0000
```

7. The Engle–Granger approach to cointegration testing involves regressing the dependent variable on the independent variables in levels and testing whether the resulting residuals are stationary, which indicates the presence of a cointegrating relationship. Dickey–Fuller and Phillips–Perron unit-root tests show that the residuals of the Conservative, Labour, and Liberal Democrat regressions are stationary—thus we have reason to believe that there is a cointegrating relationship present.

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
D_Ldm_Con						
L_Ldm_Con	-.5776355	.0986824	-5.85	0.000	-.7710496	-.3842215
D_all_pidW	-2.243771	1.071513	-2.09	0.036	-4.343899	-.1436436
D_all_LabLe~W	-.0317277	.0857915	-0.37	0.712	-.1998759	.1364205
D_all_ConLe~W	-.2779544	.0627374	-4.43	0.000	-.4009174	-.1549915
D_all_LDLea~W	.4820017	.0779929	6.18	0.000	.3291385	.6348649
D_all_nat_r~W	.1491226	.1410416	1.06	0.290	-.1273139	.4255591
L_all_pidW	1.247393	1.016881	1.23	0.220	-.7456575	3.240443
L_all_LabLe~W	-.0485621	.0740909	-0.66	0.512	-.1937775	.0966533
L_all_ConLe~W	-.0948527	.0702979	-1.35	0.177	-.232634	.0429286
L_all_LDLea~W	.1754929	.0494942	3.55	0.000	.078486	.2724998
L_all_nat_r~W	.0994187	.0545379	1.82	0.068	-.0074737	.2063111
D_all_b_mii~t	.2725831	.5676606	0.48	0.631	-.8400113	1.385177
L_all_b_mii~t	-.4906083	.4268449	-1.15	0.250	-1.327209	.3459924
_cons	-1.068066	.6230965	-1.71	0.087	-2.289313	.1531808
D_Lab_Con						
L_Lab_Con	-.5165434	.0867791	-5.95	0.000	-.6866273	-.3464596
D_all_pidW	-.3706903	.620695	-0.60	0.550	-1.58723	.8458495
D_all_LabLe~W	.2363446	.0475815	4.97	0.000	.1430866	.3296025
D_all_ConLe~W	-.1501075	.0348201	-4.31	0.000	-.2183536	-.0818614
D_all_LDLea~W	.0072788	.0428209	0.17	0.865	-.0766487	.0912063
D_all_nat_r~W	.0783334	.0781189	1.00	0.316	-.0747769	.2314437
L_all_pidW	1.726958	.6186231	2.79	0.005	.514479	2.939437
L_all_LabLe~W	.1743062	.0471755	3.69	0.000	.081844	.2667684
L_all_ConLe~W	-.0119387	.037973	-0.31	0.753	-.0863644	.0624869
L_all_LDLea~W	.0168671	.0248225	0.68	0.497	-.0317841	.0655182
L_all_nat_r~W	.0942601	.0321303	2.93	0.003	.0312859	.1572342
D_all_b_mii~t	1.658563	.3142354	5.28	0.000	1.042673	2.274453
L_all_b_mii~t	.094476	.2301271	0.41	0.681	-.3565649	.5455169
_cons	-1.544897	.3709526	-4.16	0.000	-2.271951	-.8178433

```
Simulating main parameters.  Please wait....
Note: Clarify is expanding your dataset from 68 observations to 1000
observations in order to accommodate the simulations.  This will append
missing values to the bottom of your original dataset.
% of simulations completed: 3% 7% 10% 14% 17% 21% 25% 28% 32% 35% 39% 42% 46% 50
> % 53% 57% 60% 64% 67% 71% 75% 78% 82% 85% 89% 92% 96% 100%

Simulating Sigma matrix.  Please wait..
% of simulations completed: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Number of simulations  : 1000
Names of new variables : b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16
> b17 b18 b19 b20 b21 b22 b23 b24 b25 b26 b27 b28 b29 b30 b31
Please Wait...Simulation in Progress (20)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
....................
(note: file dynsimpie_results.dta not found)
file dynsimpie_results.dta saved
```

By default, the estimation results are presented, as shown in table 3.[8] Note that because there were three dependent variables, `dynsimpie` automatically performs the log-ratio transformation so that only two log compositional ratios are shown in the output: ln(Lib. Dems./Conservative) and ln(Labour/Conservative).

Table 3. Table of results from `dynsimpie` output

Variable	$\ln(\frac{\text{Lib. Dems.}}{\text{Conservative}})$		$\ln(\frac{\text{Labour}}{\text{Conservative}})$	
Lagged Dependent Variable	−0.58*	(0.10)	−0.52*	(0.09)
ΔParty ID$_t$	−2.24*	(1.07)	−0.37	(0.62)
ΔLabour Leader Eval.$_t$	−0.03	(0.09)	0.24*	(0.05)
ΔConservative Leader Evaluation$_t$	−0.28*	(0.06)	−0.15*	(0.03)
ΔLib. Dem. Leader Evaluation$_t$	0.48*	(0.08)	0.01	(0.04)
ΔNatl. Retrospective Evaluation$_t$	0.15	(0.14)	0.08	(0.08)
ΔLabour “Best Manager” of Economy$_t$	0.27	(0.57)	1.66*	(0.31)
Party ID$_{t-1}$	1.25	(1.02)	1.73*	(0.62)
Labour Leader Eval.$_{t-1}$	−0.05	(0.07)	0.17*	(0.05)
Conservative Leader Eval.$_{t-1}$	−0.09	(0.07)	−0.01	(0.04)
Lib. Dem. Leader Eval.$_{t-1}$	0.18*	(0.05)	0.02	(0.02)
Natl. Retrospective Eval.$_{t-1}$	0.10	(0.05)	0.09*	(0.03)
Labour “Best Manager” of Economy$_{t-1}$	−0.49	(0.43)	0.09	(0.23)
Constant	−1.07	(0.63)	−1.54*	(0.37)

Note: Coefficients from a seemingly unrelated regression with standard errors in parentheses. Two-tailed test statistics. * $p < .05$.

While the coefficients in table 3 are useful for judging significance of slope coefficients and calculating long- and short-run effects, the coefficients are difficult to interpret because a parameter estimate represents the effect that that particular variable has on the logged ratio of one political party relative to the other. Therefore, graphical interpretations, such as the predicted probabilities shown in figure 2, are particularly useful. Recall that we specified a one standard-deviation increase in the proportion of those who think Labour is the best manager of the most important issue to occur at $t = 9$. This is shown in figure 2. It is clear that this change has both long- and short-run effects on the predicted proportion of party support in the UK. In response to the shock, in the short run, Labour receives about a three-percentage point boost. This comes almost entirely at the expense of Conservative support. However, over the long run, Liberal Democrats experience a drop in support—in contrast, Conservative support returns to the starting value, and Labour support diminishes, eventually settling just above its starting value.

8. The `notable` option suppresses table output.

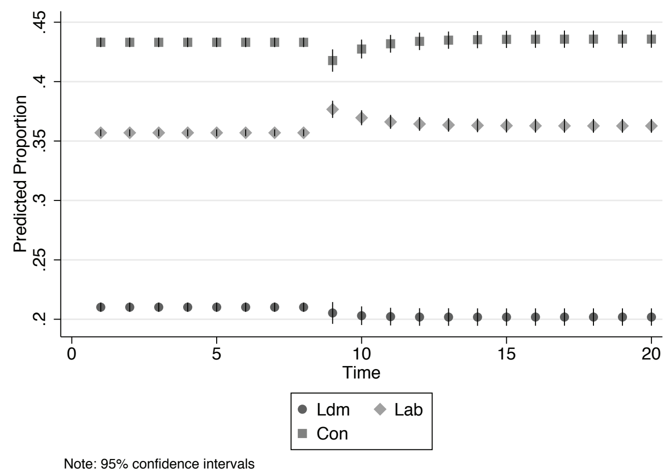

Figure 2. One standard-deviation increase in those who think Labour is the best manager of the most important issue, created using the `graph` option

While the plot in figure 2, generated using the `graph` option, is adequate in many cases, users may desire to customize their plots. By default, `dynsimpie` will save the predicted midpoints, time variable, and upper and lower confidence intervals to a dataset called `dynsimpie_results.dta` or a user-specified name using the `saving()` option.[9] The saved dataset can be opened to create a customized graph:

```
. preserve
. use dynsimpie_results
. twoway rspike var1_pie_ul_ var1_pie_ll_ time ||
> rspike var2_pie_ul_ var2_pie_ll_ time ||
> rspike var3_pie_ul_ var3_pie_ll_ time ||
> scatter mid1 time || scatter mid2 time || scatter mid3 time,
> legend( order(4 "Conservatives" 5 "Liberal Democrats" 6 "Labour"))
> xtitle("Month") ytitle("Predicted Proportion of Support")
. restore
```

9. The `nosave` option suppresses this action.

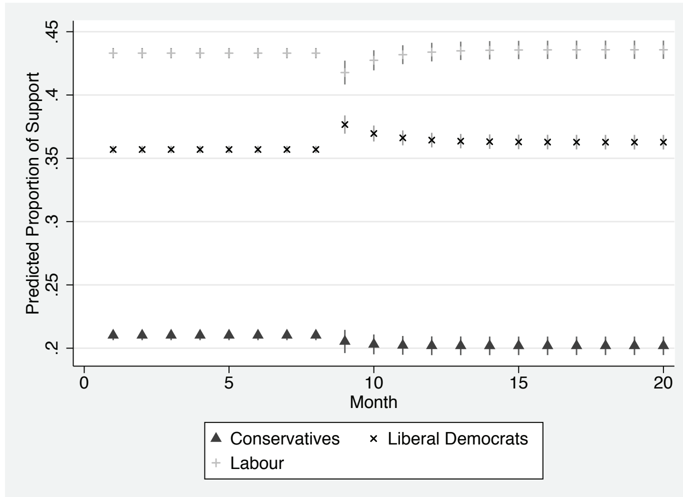

Figure 3. One standard-deviation increase in those who think Labour is the best manager of the most important issue, customized graph

`dynsimpie` has several options that users can specify to create a graph. The `time()` option changes the time that the independent variable receives a shock, while `range()` specifies the length of the scenario to simulate. For example, figure 4 shows the same counterfactual shock as in figure 3 but specifies a range of 40 months and changes the shock to occur during month 30. In addition, using the `dummy()` option, we can add a dummy variable that is equal to one during the months of the Great Recession. Because, by default, `dynsimpie` will set any dummy variables to zero, we can set this variable to one using the `dummyset()` option. As shown in figure 4, the confidence intervals grow wider to reflect an increase in uncertainty; otherwise, the results are very similar to the model without the Great Recession dummy variable.

```
. dynsimpie all_pidW all_LabLeaderEval_W all_ConLeaderEval_W
> all_LDLeaderEval_W all_nat_retW, dvs(Con Ldm Lab) t(30) range(40) shock(0.054)
> shockvar(all_b_mii_lab_pct) graph dummy(recession_dum) dummyset(1)
Seemingly unrelated regression
```

Equation	Obs	Parms	RMSE	"R-sq"	chi2	P
D_Ldm_Con	68	14	.1046312	0.7186	186.20	0.0000
D_Lab_Con	68	14	.058494	0.7901	273.29	0.0000

```
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
D_Ldm_Con    |
    L_Ldm_Con |  -.5794892     .09819    -5.90   0.000    -.7719381   -.3870403
   D_all_pidW |  -2.381324   1.079898    -2.21   0.027    -4.497885   -.2647632
D_all_LabLe~W |  -.0351874   .0854955    -0.41   0.681    -.2027555    .1323807
D_all_ConLe~W |  -.2752663   .0625296    -4.40   0.000     -.397822   -.1527106
D_all_LDLea~W |   .4756217    .078012     6.10   0.000     .3227209    .6285225
D_all_nat_r~W |   .1846631   .1469831     1.26   0.209    -.1034185    .4727447
   L_all_pidW |   .9048556   1.095946     0.83   0.409     -1.24316    3.052871
L_all_LabLe~W |  -.0408467   .0743497    -0.55   0.583    -.1865694     .104876
L_all_ConLe~W |   -.087119   .0706325    -1.23   0.217    -.2255561    .0513181
L_all_LDLea~W |    .174251   .0492857     3.54   0.000     .0776527    .2708492
L_all_nat_r~W |   .1568401   .0889912     1.76   0.078    -.0175795    .3312597
D_all_b_mii~t |   .2530257    .565468     0.45   0.655    -.8552712    1.361322
L_all_b_mii~t |  -.4664369    .425917    -1.10   0.273    -1.301219    .3683451
recession_dum |   .0599391   .0736982     0.81   0.416    -.0845067    .2043849
        _cons |  -1.166204   .6317957    -1.85   0.065    -2.404501    .0720931
-------------+----------------------------------------------------------------
D_Lab_Con    |
    L_Lab_Con |   -.518055   .0867466    -5.97   0.000    -.6880752   -.3480347
   D_all_pidW |  -.3734059   .6283616    -0.59   0.552    -1.604972    .8581602
D_all_LabLe~W |   .2362032    .047641     4.96   0.000     .1428285     .329578
D_all_ConLe~W |  -.1499786   .0348656    -4.30   0.000    -.2183139   -.0816433
D_all_LDLea~W |   .0069452   .0430183     0.16   0.872    -.0773692    .0912595
D_all_nat_r~W |   .0800081   .0817055     0.98   0.327    -.0801318    .2401479
   L_all_pidW |   1.717006   .6624619     2.59   0.010     .4186047    3.015408
L_all_LabLe~W |    .175037   .0474261     3.69   0.000     .0820835    .2679904
L_all_ConLe~W |  -.0118403   .0384271    -0.31   0.758    -.0871561    .0634754
L_all_LDLea~W |   .0167159   .0248444     0.67   0.501    -.0319783    .0654101
L_all_nat_r~W |    .096921    .050539     1.92   0.055    -.0021337    .1959757
D_all_b_mii~t |   1.656847   .3144173     5.27   0.000     1.040601    2.273094
L_all_b_mii~t |   .0960153   .2307912     0.42   0.677    -.3563272    .5483577
recession_dum |    .002554   .0410924     0.06   0.950    -.0779857    .0830937
        _cons |  -1.551414   .3766141    -4.12   0.000    -2.289564   -.8132639
------------------------------------------------------------------------------

Simulating main parameters.  Please wait....

Note: Clarify is expanding your dataset from 68 observations to 1000
observations in order to accommodate the simulations.  This will append
missing values to the bottom of your original dataset.

% of simulations completed: 3% 6% 10% 13% 16% 20% 23% 26% 30% 33% 36% 40% 43% 46
> % 50% 53% 56% 60% 63% 66% 70% 73% 76% 80% 83% 86% 90% 93% 96% 100%

Simulating Sigma matrix.  Please wait..
% of simulations completed: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Number of simulations  : 1000
Names of new variables : b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16
> b17 b18 b19 b20 b21 b22 b23 b24 b25 b26 b27 b28 b29 b30 b31 b32 b33

Please Wait...Simulation in Progress (40)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
........................................
file dynsimpie_results.dta saved
```

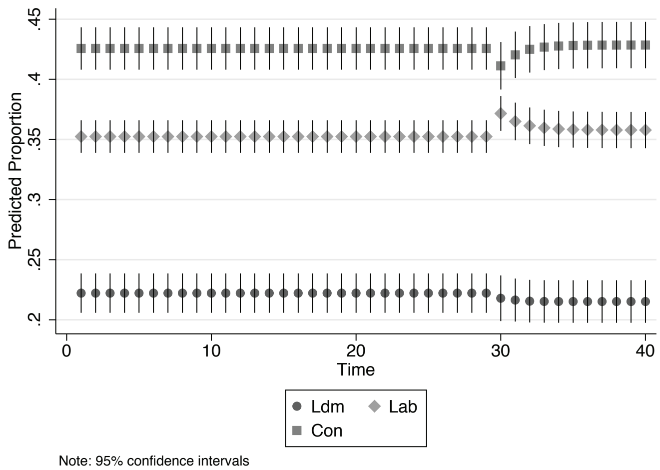

Figure 4. One standard-deviation increase in those who think Labour is the best manager of the most important issue during the Great Recession

In addition to plotting the simulated effect of a single shock, `dynsimpie` allows for up to three shocks to occur at the same point in time.[10] For instance, in figure 5, we show the estimated effects of a one standard-deviation increase in the percentage of those who think Labour is the best manager of the most important issue and a one standard-deviation increase in Labour leader evaluations. This is done using the `shockvar2()` option.[11] These both take effect at time $t = 18$ over a total range of $t = 30$:

```
. dynsimpie all_pidW all_ConLeaderEval_W all_LDLeaderEval_W all_nat_retW,
> dvs(Con Ldm Lab) t(18) range(30) shock(0.054) shockvar(all_b_mii_lab_pct)
> shockvar2(all_LabLeaderEval_W) shock2(0.367) graph
Seemingly unrelated regression
```

Equation	Obs	Parms	RMSE	"R-sq"	chi2	P
D_Ldm_Con	68	13	.1051191	0.7159	183.70	0.0000
D_Lab_Con	68	13	.0584838	0.7902	273.01	0.0000

10. This is particularly helpful if compositional variables make up some of the independent variables. For instance, if we included something like previous vote share, we would have to include two of the three previous party-vote variables (Labour and Liberal Democrats, for instance, without loss of generality). Thus a one standard-deviation increase in Labour must necessitate a drop in either the Liberal Democrats or the Conservatives (this would occur if we left the Liberal Democrats unchanged at the shock time) or both. By giving half the corresponding loss to the Liberal Democrats (and thus saving the other half for the Conservatives), we provide the most informative counterfactual, also called a "ratio-preserving counterfactual" (Adolph 2013), because we get as close to observing the response to one variable's change—increase to Labour.
11. If a shock to an additional variable was desired, this would be performed using the `shockvar3()` option.

```
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
D_Ldm_Con    |
    L_Ldm_Con |  -.5776355   .0986824    -5.85   0.000    -.7710496   -.3842215
   D_all_pidW |  -2.243771   1.071513    -2.09   0.036    -4.343899   -.1436436
D_all_ConLe~W |  -.2779544   .0627374    -4.43   0.000    -.4009174   -.1549915
D_all_LDLea~W |   .4820017   .0779929     6.18   0.000     .3291385    .6348649
D_all_nat_r~W |   .1491226   .1410416     1.06   0.290    -.1273139    .4255591
   L_all_pidW |   1.247393   1.016881     1.23   0.220    -.7456575    3.240443
L_all_ConLe~W |  -.0948527   .0702979    -1.35   0.177     -.232634    .0429286
L_all_LDLea~W |   .1754929   .0494942     3.55   0.000      .078486    .2724998
L_all_nat_r~W |   .0994187   .0545379     1.82   0.068    -.0074737    .2063111
D_all_b_mii~t |   .2725831   .5676606     0.48   0.631    -.8400113    1.385177
D_all_LabLe~W |  -.0317277   .0857915    -0.37   0.712    -.1998759    .1364205
L_all_b_mii~t |  -.4906083   .4268449    -1.15   0.250    -1.327209    .3459924
L_all_LabLe~W |  -.0485621   .0740909    -0.66   0.512    -.1937775    .0966533
        _cons |  -1.068066   .6230965    -1.71   0.087    -2.289313    .1531808
-------------+----------------------------------------------------------------
D_Lab_Con    |
    L_Lab_Con |  -.5165434   .0867791    -5.95   0.000    -.6866273   -.3464596
   D_all_pidW |  -.3706903    .620695    -0.60   0.550     -1.58723    .8458495
D_all_ConLe~W |  -.1501075   .0348201    -4.31   0.000    -.2183536   -.0818614
D_all_LDLea~W |   .0072788   .0428209     0.17   0.865    -.0766487    .0912063
D_all_nat_r~W |   .0783334   .0781189     1.00   0.316    -.0747769    .2314437
   L_all_pidW |   1.726958   .6186231     2.79   0.005      .514479    2.939437
L_all_ConLe~W |  -.0119387    .037973    -0.31   0.753    -.0863644    .0624869
L_all_LDLea~W |   .0168671   .0248225     0.68   0.497    -.0317841    .0655182
L_all_nat_r~W |   .0942601   .0321303     2.93   0.003     .0312859    .1572342
D_all_b_mii~t |   1.658563   .3142354     5.28   0.000     1.042673    2.274453
D_all_LabLe~W |   .2363446   .0475815     4.97   0.000     .1430866    .3296025
L_all_b_mii~t |    .094476   .2301271     0.41   0.681    -.3565649    .5455169
L_all_LabLe~W |   .1743062   .0471755     3.69   0.000      .081844    .2667684
        _cons |  -1.544897   .3709526    -4.16   0.000    -2.271951   -.8178433
------------------------------------------------------------------------------

Simulating main parameters.  Please wait....
Note: Clarify is expanding your dataset from 68 observations to 1000
observations in order to accommodate the simulations.  This will append
missing values to the bottom of your original dataset.
% of simulations completed: 3% 7% 10% 14% 17% 21% 25% 28% 32% 35% 39% 42% 46% 50
> % 53% 57% 60% 64% 67% 71% 75% 78% 82% 85% 89% 92% 96% 100%
Simulating Sigma matrix.  Please wait..
% of simulations completed: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Number of simulations  : 1000
Names of new variables : b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16
> b17 b18 b19 b20 b21 b22 b23 b24 b25 b26 b27 b28 b29 b30 b31
Please Wait...Simulation in Progress (30)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..............................
file dynsimpie_results.dta saved
```

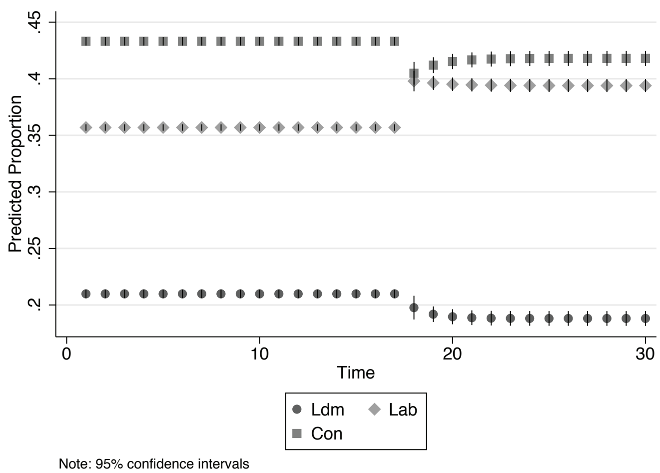

Figure 5. One standard-deviation increase in those who think Labour is the best manager of the most important issue and a one standard-deviation increase in Labour leader evaluations

Clearly, there is a sizable short-run increase in Labour support that does not diminish over time. This appears to be roughly evenly split between the Conservatives and the Liberal Democrats over the long run; however, over the short run, the Conservatives are predicted to have the largest negative change in support.

5 Conclusion

In this article, we introduced a new command for Stata, `dynsimpie`, that fits and interprets dynamic models of compositional dependent variables. We agree with Philips, Rutherford, and Whitten (2016, 282) that "once researchers start looking for dynamic compositional variables, they will find them everywhere."

6 References

Adolph, C. 2013. *Bankers, Bureaucrats, and Central Bank Politics: The Myth of Neutrality*. New York: Cambridge University Press.

Aitchison, J. 1982. The statistical analysis of compositional data. *Journal of the Royal Statistical Society, Series B* 44: 139–177.

———. 1983. Principal component analysis of compositional data. *Biometrika* 70: 57–65.

———. 1986. *The Statistical Analysis of Compositional Data*. London: Chapman & Hall/CRC.

De Boef, S., and L. Keele. 2008. Taking time seriously. *American Journal of Political Science* 52: 184–200.

Engle, R. F., and C. W. J. Granger. 1987. Co-integration and error correction: Representation, estimation, and testing. *Econometrica* 55: 251–276.

Hoffman, D. R., and R. Uauy. 1992. Essentiality of dietary $\omega 3$ fatty acids for premature infants: Plasma and red blood cell fatty acid composition. *Lipids* 27: 886–895.

Katz, J. N., and G. King. 1999. A statistical model for multiparty electoral data. *American Political Science Review* 93: 15–32.

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.

Pawlowsky-Glahn, V., and A. Buccianti. 2011. *Compositional Data Analysis: Theory and Applications*. Chichester, UK: Wiley.

Philips, A. Q., A. Rutherford, and G. D. Whitten. 2015. The dynamic battle for pieces of pie—Modeling party support in multi-party nations. *Electoral Studies* 39: 264–274.

———. 2016. Dynamic pie: A strategy for modeling trade-offs in compositional variables over time. *American Journal of Political Science* 60: 268–283.

Tomz, M., J. A. Tucker, and J. Wittenberg. 2002. An easy and accurate regression model for multiparty electoral data. *Political Analysis* 10: 66–83.

Tomz, M., J. Wittenberg, and G. King. 2003. clarify: Software for interpreting and presenting statistical results. *Journal of Statistical Software* 8(1): 1–30.

Whitten, G. D., and L. K. Williams. 2011. Buttery guns and welfare hawks: The politics of defense spending in advanced industrial democracies. *American Journal of Political Science* 55: 117–134.

Williams, L. K., and G. D. Whitten. 2011. Dynamic simulations of autoregressive relationships. *Stata Journal* 11: 577–588.

———. 2012. But wait, there's more! Maximizing substantive inferences from TSCS models. *Journal of Politics* 74: 685–693.

About the authors

Andrew Philips is a PhD candidate in the Department of Political Science at Texas A&M University.

Amanda Rutherford is an assistant professor in the School of Public and Environmental Affairs at Indiana University.

Guy D. Whitten is a professor in the Department of Political Science at Texas A&M University.