THE POLITICAL METHODOLOGIST

Editor: SUZANNA DE BOEF, PENNSYLVANIA STATE UNIVERSITY
sdeboef@psu.edu

Editorial Assistant: HEATHER L. ONDERCIN, PENNSYLVANIA STATE UNIVERSITY
hlo114@psu.edu

Contents

Notes From the Editor

This issue of *The Political Methodologist* contains an assortment of pieces particularly relevant to how we teach and use methods in the classroom. Contributions include reviews of software packages, including a detailed

Smith. 1992. "Error Correction, Attractors and Cointegration: Substantive and Methodological Issues." *Political Analysis* 4: 249-258.

William, John T. 1990. "The Political Manipulation of Macroeconomic Policy." *American Political Science Review* 84: 767-796.

Visual Interpretation and Presentation of Monte Carlo Results

Christopher Adolph[1]
Harvard University
cadolph@fas.harvard.edu

Tables vs. graphics in Monte Carlo presentation

Analytical methods are the best way to learn the properties of an estimator since they cover the entire parameter space. But the analytical route is not always available. Proofs may be intractable without unreasonable simplifying assumptions, and many properties of statistical models hold in the limit only. In either case, we turn to Monte Carlo (MC) experiments—testing models on artificial datasets with known properties—to assess the likely performance of an estimator in empirical work. While analytical methods provide complete characterizations across the (usually infinite) parameter space, Monte Carlo surveys are generally incomplete, and thus potentially misleading. To mitigate this key disadvantage, we must thoroughly explore the (relevant) parameter space and devise clever techniques for presenting as much of this space as possible on the printed page.

Unfortunately, though the increasing savvy of political methodology has brought more and better Monte Carlo work, it is not always presented clearly or thoroughly. In particular, MC results often appear in unwieldy tables rather than elegant graphics. Tables are ideal for presenting small quantities of data whose precise

---

values are worth seeing. This is the opposite of the situation in MC work, where the "data" are potentially limitless (just add more parameter values) and precision arbitrarily high (just add more simulations) but usually uninteresting. Researchers may miss patterns, readers' eyes glaze over, and the results remain bound in a straightjacket of rows and columns. Moreover, tables discourage exploration of more than a few parameter combinations, while pictures enable researchers to present far more comprehensive findings. Perhaps the only legitimate use for tables in presenting MC work is to list many different statistics from a single model and data generating process. Usually we are interested in how one or a few statistics (such as mean squared error and bias) vary across models or scenarios, and graphics should be used instead.

I surveyed recent years of *Political Analysis* (2000–2002), the *American Political Science Review* (1998–2002), and the *American Journal of Political Science* (1998–2002), and found 24 articles reporting Monte Carlo experiments. Of these, 12 reported results in tables only, 9 used graphs only, and 3 used a combination. In all, nine articles could have substituted graphics to their advantage (on the criterion of presenting a statistic under many models or scenarios), while in other cases a large number of non-comparable outcome measures left no alternative to tables.

To help the quality of Monte Carlo presentations keep up with the sophistication of the experiments themselves, I propose five guidelines for MC graphics. I also define five graphic styles which help show the comparative performance of models over the parameter space, even when the models and parameters are many.

Five principles for visual displays of Monte Carlo results

1. **Maximize resolution.**

   The main reason to use graphs is that they allow higher data density than tables, and often gain readability in the bargain. In a table, you might only try two values on a given parameter to save space. But in a graph, running many MC scenarios just makes the pixels smaller and patterns clearer.

2. **Get the whole picture, and nothing but the picture.**

   Try to canvass the "whole" parameter space (infinities and asymptopia aside). But also take advantage of any logical or substantive limits on the parameters in applications of interest. Within these limits, a full factorial design (considering all combinations

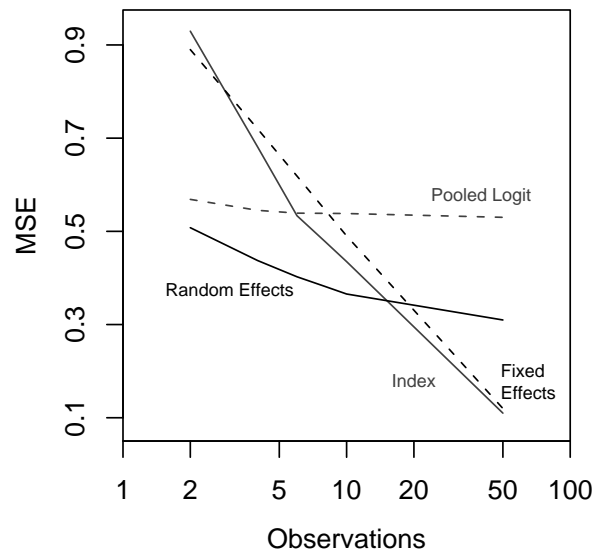| Mean Squared Error				
	Pooled		Fixed	Random
N	Logit	Index	Effects	Effects
2	0.57	0.93	0.89	0.51
4	0.54	0.61	0.72	0.43
6	0.54	0.53	0.62	0.41
8	0.52	0.49	0.58	0.37
10	0.54	0.43	0.49	0.37
50	0.53	0.11	0.12	0.31

Figure 1: *Contour plot of model performance and the tabular alternative.* Monte Carlo results on ideal point estimators taken from Bailey (2001). The original display is at the left, with a new display, using the recommendations of this article, at right. The plot shows loess-smoothed contours (unsmoothed contours would also work). Note the $\log_{10}$ scaling of the horizontal axis, which avoids illegible compression of the crucial small sample results.

from some set of hypothetical values on each parameter) is especially illuminating if you have the computer power.

3. **Focus on interesting patterns.**

   Investigate repeated patterns, then use these patterns to condense the parameter space. N-dimensional space gets much smaller when everything in it is an apple, orange, or pear.

4. **Cheat the curse of dimensionality.**

   ... using small multiples (arrays of similar graphics), shading, and creativity (see Tufte, 1990, 2001).

5. **Make the results usable.**

   Ideally, MCs help researchers decide which model to use in specific situations. If your results show when certain models should be used or avoided based on knowable quantities, make these recommendations clear and easily referenced. They will be the most used part of your article.

Graphical alternatives to tabular tyranny

The remainder of this article grapples with the key challenge of Monte Carlo presentation: many parameters (p_1,

p_2, ...), many models, and only two-dimensional paper to put them on.

Model performance plots

The first model performance plot we consider is an estimate-vs-truth plot (EvT). This is a special case in which the measure of performance is an estimate of p_1. The one-parameter EvT plot is a simple scatter-diagram, in which a pattern of points on or near the 45° line indicates good performance. This plot is ideal for demonstrating whether $\hat{p}_1 - p_1$ is independent of variation in other parameters. For each value of p_1 in the experiment, run trials with diverse values of all other parameters, then check whether the results cluster on or near the 45° line. Additional models can be distinguished through different symbols or colors, and patterns of dependence through arrays of EvT plots. For an example EvT plot, see Adolph et. al. (2003).

EvT is suited to the special case where the performance measure is the estimate of a parameter. Generally, we want to show model performance on some arbitrary metric Q for parameters p_1 and p_2. To reduce these three dimensions to a sheet of paper, we have at least two options: contour plots and image plots. We can plot Q against p_1 for a given value of p_2, producing a "performance contour" at that level of p_2. (The contour itself may be a loess-smoothed curve or a line "connecting the
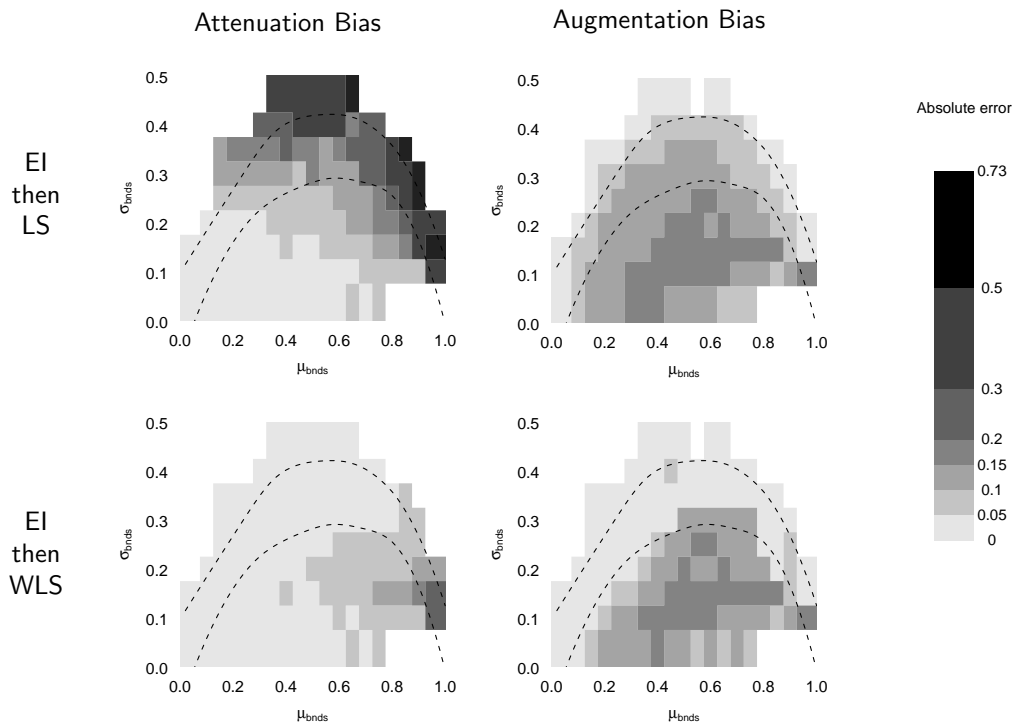
Figure 2: *Image plots of model performance.* Image plots which show absolute error in estimates derived from two different models (EI followed by least squares and EI followed by weighted least squares) over three parameters (the average bound width μ_{bnds}, the standard deviation of bound width σ_{bnds}, and whether the data were generated to produce attenuation bias or augmentation bias). Dark areas indicate poor performance, lightly shaded errors good performance, and blank areas cases not included in the experiment. The dashed lines indicate where datasets in the real world tend to fall in terms of μ_{bnds} and σ_{bnds}. For more details, and a treatment of the same data using contour plots, see Adolph et. al. (2003).

dots"). Then by selecting various levels of p_2, we can map out performance in p_2 space, though with coarser detail than is available in p_1 or Q space. As always, we can accommodate p_3, p_4, etc., through arrays of contour plots.

The contour plot can also be used to directly compare several models on a single parameter. Bailey (2001) investigated the performance of four estimators of legislators' ideal points when only a few votes are available, presenting the table in Figure 1. Bailey concludes that the random effects estimator is superior for small samples, while the fixed effects estimator is better as the number of votes grows. This can be gleaned from the table, but is immediate when we redisplay Bailey's results as a contour plot. The plot also draws attention to the point at which random effects ceases to be the best choice ($N \approx 20$). For more examples of contour plots in MC work, see King and Zeng (2001) and Adolph et al (2003).

An alternative plot focuses on the parameters and categorizes Q. This produces a "map" of performance over precise values of p_1 and p_2 where shading indicates

the level of Q. Additional models or parameters can be easily accommodated by small multiples, a design for which these "image" plots are particularly suited. Figure 2 illustrates this approach using Monte Carlo results on second-stage ecological inference estimators (see Adolph et al, 2003 for further details). These plots demonstrate the five principles advocated above. First, they *show the whole picture*, while highlighting those parameter combinations likely to occur in practice (the area inside the dashed lines). Second, image plot's inherent capacity for high resolution allows presentation of a matrix of scenarios covering the whole space in two parameters (μ_{bnds} and σ_{bnds}). Third, the graphs *focus on interesting parts* of the remaining parameter space (here, each column of plots represents a different "worst case" scenario for model failure). Fourth, *arrays of plots* show 5 dimensions (error, μ_{bnds}, σ_{bnds}, type of bias, and model). Finally, the plots *provide a guide for practical research*, allowing researchers to make better informed modelling decisions.
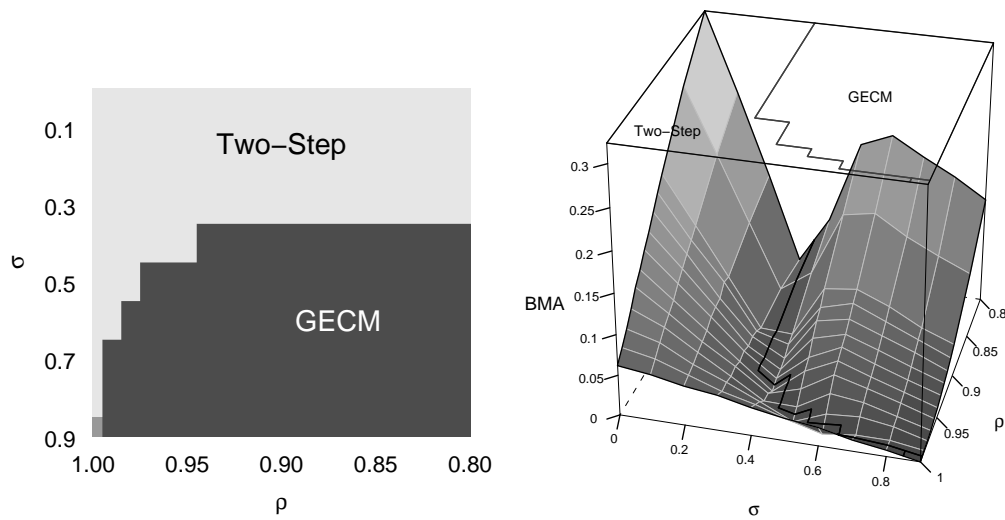
Figure 3: *Best Model plots.* These plots report MC results on two time series estimators, the Generalized Error Correction Model (GECM) and the Engle-Granger two-step method (Two-Step) from De Boef (2001). The left plot shows which model has minimum bias ("best model") at each combination of autocorrelation (ρ) and simultaneity (σ). (The models have equivalent performance in the bottom left corner). The right plot shows the advantage of the best model (BMA) over the second best model for each parameter combination. If the true parameter values are known, the left plot shows which model to use. If the parameter values are unknown, the right plot aids in deciding which model is likely to minimize bias.

Best model plots

Arrayed image plots handle arbitrarily many models or parameters, but if many models are considered at once, the ensuing pages of graphs will try the patience of readers. We need a presentation that shows the "big picture", leaving the task of "zooming in" on interesting features or patterns to selected model performance plots. In this case, I recommend a summary graphic which shades the "best" model at each point in the parameter space. (Lest we interpret "best model" too rigidly, where models are approximately indistinguishable, the plot should list all contenders.) Once the "best" model is identified, a researcher could follow up with an appropriate selection of model performance plots. Alternatively, a 3D "best model advantage" plot shows how much better the best model performs than the second-best at every point in the parameter space. Models covering more volume in this plot are safer bets.

      To create example best model plots, I drew on De Boef's (2001) work on two time series estimators—the Engle-Granger two-step method and the Generalized Error Correction Model (GECM)—applied to highly autoregressive data. De Boef notes that both methods may be inconsistent if the data generating process is not quite permanently memoried, and that simultaneity in the errors of the time series exacerbates this problem. De Boef ran MC experiments with varied persistence (ρ) in the

time series and covariance (σ) in shocks to the explanatory and dependent variables. De Boef employs useful and elegant 3D plots to show how the coverage of confidence intervals for each model varies with ρ and σ, but resorts to large tables to report bias in estimates of the long-run relationship between time series. Making a "best model" plot from these results helps show which model is less biased under different circumstances (in this case, GECM if simultaneity is high, and Engle-Granger otherwise). A "best model advantage" plot shows that the gap between the models' performance grows as the persistence ρ declines. The difference in performance also displays a regular, though non-linear, relationship with simultaneity.

How to do it

Using graphics instead of tables will make your Monte Carlo results more complete, clear, and usable. These advantages can be gained without much effort, since the five plotting styles proposed in this article (MP EvT, MP contour, MP image, BM image, and BMA) can all be produced by the software package `SeeMC` (Adolph, 2003). `SeeMC` runs in R and Gauss, and is available at `http://chris.adolph.name`. A standalone version is in the works.

NB: Formal modelers will doubtless find "best model" image plots reminiscent of graphs of comparative statics from game models (see, e.g., Deiermeyer and Stevenson, 2000). `SeeMC` also provides an easy way to make these plots.

References

Adolph, Christopher. 2003. *SeeMC*. R and Gauss software for visualization of Monte Carlo results, available at `http://chris.adolph.name`.

Adolph, Christopher, and Gary King with Michael C. Herron and Kenneth W. Shotts. 2003. "A consensus on second-stage analyses in ecological inference models." *Political Analysis* 11.1: 87–95.

Bailey, Michael. 2001. "Ideal point estimation with a small number of votes: A random-effects approach." *Political Analysis* 9.3: 192–210.

De Boef, Suzanna. 2001. "Modelling equilibrium relationships: Error correction models with strongly autoregressive data." *Political Analysis* 9.1: 78–94.

Deiermeyer, Daniel, and Randolph Stevenson. 2000. "Cabinet terminations and critical events." *American Political Science Review* 94.3: 627–640.

King, Gary, and Langche Zeng. 2001. "Logistic regression in rare events data." *Political Analysis* 9.2: 137-163.

Tufte, Edward. 1990. *Envisioning Information.* Graphics Press.

Tufte, Edward. 2001. *The Visual Display of Quantitative Information.* Graphics Press. 2nd ed.

The LaTeX Corner: Games in LaTeX

Tamar London
Pennsylvania State University
trlondon@psu.edu

One of things that attracted me to LaTeX was a desire to create nice looking game trees. Extensive form games that I have created in MS-Word have left me unsatisfied.

I had to eyeball the length and angle of the branches, as well as the location of labels. In LaTeX you specify the exact location, slope and length of branches. The end result, in my opinion, is more professional looking.

Let me preface this piece by saying that I am new to LaTeX. As such, I am sure there are many more qualified than I am to write this. On the other hand, I have found that to those familiar with it, LaTeX becomes second nature and memories of a painful start have long faded. To me, these memories are fresh. I will assume, therefore, that users are familiar with LaTeX, but (like me) are still new to it. For those that are unfamiliar with LaTeX, I recommend Chan H. Nam's article in a previous edition (10:1) of *The Political Methodologist.* I will also assume that users are using a windows based platform (simply because I am unfamiliar with unix and macintosh). Most of what I go over should translate easily to the other platforms.

Foreplay

To make trees using LaTeX you will need the basics: LaTeX and a text editor. I recommend the MikTex implementation for Windows. To edit the LaTeX text you need a text editor. WinEdt is the one most commonly used with LaTeX. Once you download WinEdt and MikTex they are fully integrated.

You will also need a package capable of drawing extensive games.

The Birds and The Bees

If your goal is to create basic trees then I recommend using egame. This style was created by Martin Osborne and can be found on his website: `http://www.chass.utoronto.ca/~osborne/latex/index.html`.[1]

To install egame you simply need to save the file to you local Tex tree. For example, create a folder called "egame" under `C:\texmf\tex\LaTeX` and save the egame.sty file there. Your computer might try to save it as an html document. Don't let it. Make sure it is saved as a .sty file. You then need to let MikTex know that this file has been downloaded. You can refresh MikTex by going to `Accessories/MikTex_Options` in WinEdt and clicking "refresh."

To create a game you will need to put the `usepackage{egame}` command in the preamble (between `documentclass{article}` and `begin{document}`).

---

[1]Osborne has a very straightforward style for normal form games, as well.