# Bringing theoretical models to data

Ben Klemens

1 May 2009

I had a simple agenda behind *Modeling with data*: better modeling.

Despite the title, many people miss this, what with the exposition on modeling restricted to the first few pages and the rest of the book being filled with C code. I used to think those people missed the modeling issues because of the language thing. One del.icio.us user bookmarked this site with the note: "Statistical programming in C? What the ass?" But she did bookmark the site, and (I can tell you because of a short email exchange) did got over the foreign sound of statistics in C to like the approach.

No, I don't think reasonable people can be truly hung up over a surface issue like programming language. I now understand that the real language barrier is in the many definitions and understandings of what is a model.

When I talk to a statistician, a model means a probability distribution over elements, and that's about it. I'd start talking to a statistician about modeling subject-specific knowledge about the interaction of elements, and giant question marks would appear over his head. Which is not to say that the person is a moron, but just that his understanding of the meaning of the word *model* is much more narrowly focused than mine.

In the R package, the `model` object specifically encompasses generalized linear models (GLMs). Again, this is not to disparage R, but to show that there's a good number of people out there for whom it's perfectly OK to equate the word *model* to GLMs, because 100% of the models in their research will fall into that category.

Grab off the shelf one of those journals with "Theory" in its title, where authors can just present a theory and its implications without bringing it to data. There are such journals in any field, from economics to sociology to physics. The models are often wild and creative. Elements interact in every way imaginable. For example, I've done a lot of work with network models, where individual agents form links via iterative, nonlinear processes, regulated by whatever the author dreamed up. The models aren't necessarily complex, but they have no need to stick with simple linear components.

The empirical definition of a model and the theoretical definition don't necessarily overlap. For example, agent-based modelers naturally have agent-based simulation in mind when they hear the M-word, and many enthusiastically reject the GLMs. That attitude means that many ABMers are unfamiliar with the statistical models that were the entire world of the statisticians above. Conversely, your average statistician has zero experience with agent-based models.

**Empirical implications of theoretical models** [1]

The multiple meanings of *model* are a problem when the theory gains traction and is eventually brought to data. A model to the theorists includes anything under the sun, while a model in your typical stats package is a GLM. So instead of directly fitting the model, one tests its empirical implications, such as how variable $A$ going up should cause variable $B$ to go down. We can fit that sort of implication into a linear regression without serious violence.

But wouldn't it be great if we could fit and test the model itself?

The work I was doing that really motivated the book was on using agent-based models as probability models, which would allow for more direct testing of the model. But as above, the agent-based concept of a model and the probability concept of a model are academically disjoint: few people accept and use both concepts simultaneously.

Why not? There are many very valid reasons. There is value to specialization, and I won't claim that everybody needs to be a polyglot all the time.

But there are also many lousy reasons, based on how what we can easily theorize is so much broader than what we can easily calculate.

If your definition of a model were just OLS, you'd have no need to code anything. Mathematically, I cover that ground in two pages (pp 271–272), and implement it in code in two separate code samples, because it's so trivial that it wasn't worth revising out the redundancy. If you have the statistician's definition of *model* in mind, you're probably going to be puzzled by the book's lengthy exposition on computing the hard-to-compute.

But by the broader definition of a model, which the theoreticians in all fields are using, we are worlds away from having things so neatly boxed. That network model, however the details play out, can't be pulled off the shelf.

So that's what the book is about: my best stab at the tools you'll need to bring a new model to data (or to generate artificial data from your model), where the word *model* takes on as broad a meaning as possible.

Next time I'll talk more about the mechanics of writing code for modeling in the broader sense.

[1] EITM is the name of an ongoing series of summer institutes in political theory, in which I have participated.