In at the creation

Sean Becketti

I got sucked into helping launch Stata because my TV died.

I was an assistant professor at UCLA, trying to raise two small children during a governor-mandated salary freeze at the University of California. To make ends meet, I did some consulting, but the consulting was cutting into my research time and hurting my chances for tenure. I finally vowed to stop consulting completely, even if it meant economizing at home. Then my TV died.

It was a hand-me-down, a dinosaur, a massive (for the time) 25-inch console color TV, almost impossible to move. It had been ill for a time—most notably losing all video during a touchdown run in the SuperBowl—but I nursed it along as best I could until it finally failed completely. An infant and a toddler at home, no money for babysitters and going out, and no TV: not a recipe for a happy home.

At that moment, one of my colleagues at UCLA asked if I would work with Bill Gould on a scheme he had for a statistics package that would run on the IBM PC. I met Bill when, as a graduate student, I was briefly assigned to him as a research assistant. Later I worked with him on a consulting contract. At the time, I don't think either of us was particularly impressed with the other. I believe Bill thought I was a bit of a slacker: hiding from him when I was his research assistant didn't make a good impression. In return, I thought Bill a bit eccentric. Certainly the idea of writing a serious statistics package for the IBM PC of 1984 seemed daft. The CPUs were painfully slow, there wasn't much memory, and hard drives hadn't been introduced yet.

The idea that I should help with this project was not entirely random. I had paid part of my way through graduate school by working on a successful mainframe statistics package, and I had contributed a few nice features to it. The notion was that I would look over Bill's shoulder, test the early versions of the package, and offer some advice on one or two econometric techniques we thought were important for marketability. It seemed like a waste of my time, but it also seemed like a quick and easy way to earn enough money for a new TV.

I was pretty cocky at the start. After all, I had worked on a "serious" commercial mainframe package. And a lot of Bill's ideas mystified me. I let him babble on about preparsing and strict syntax and importing crossproducts matrices. I didn't see the point of a lot of it, but Bill seemed very enthusiastic about the program.

Anyway, I had fun testing (that is, breaking) the software. For a couple of weeks, I was able to generate crashes or serious errors in each new version in a matter of minutes. As time went on, though, the program became more and more reliable, and I had to exercise considerable ingenuity to unearth a significant problem. Moreover, the longer I worked with Stata, the more I began to appreciate Bill's design. I started to see that my

notions of interface design, learned in a card-reader-oriented mainframe environment, were clumsy compared with the interactive interface Bill envisioned. And I began to appreciate the modeless nature of Stata. While Stata's initial list of features was puny compared with established packages like SAS and SPSS, Stata was a lot more fun to drive, with no ping-ponging between DATA steps and PROCs. Stata seemed to facilitate a sort of dialogue with the data. Each command was a question asked of the data. Each result suggested further questions. And Stata made it easy to pursue these questions. I didn't have to modify a program, resubmit it, and wait for the output. I just kept asking more questions until I had the answers I sought.

This interaction was a revelation, and it changed my approach to data analysis. (Those of you old enough to have made the transition from mainframes to the early PCs may remember a similar reaction the first time someone showed you a simple spreadsheet in action. The immediacy was breathtaking.) Previously, I had dismissed exploratory data analysis, believing it was just a touchy-feely distraction from more definitive confirmatory techniques. Besides, the increasing availability of high-powered computers and professional statistical software (at least in the university) seemed to make Tukey's paper-and-pencil approach superfluous. Ironically, the ease of using Stata for EDA opened my eyes to the importance of these techniques and to the importance of robust and resistant estimators in confirmatory analysis.

Needless to say, I kept working with Bill long after I bought the TV. Bill and I wrote the first Stata manual—formatted on the mainframe (!) in the wee hours of the morning—and released the program in December 1984 at a large meeting of economists in Dallas. We returned somewhat chastened. Several other bright people had the same idea of porting professional statistics to the PC, and we realized we had some catching up to do. (Bill got a lot of well-meaning advice to throw in the towel before he lost too much money. The consensus was that the market was already too crowded and that the inevitable entry of packages like SAS and SPSS would crush smaller competitors. Fortunately, Bill is very stubborn.)

The next couple of years were among the most stimulating of my professional life. Stata grew quickly in power and features (although sales were slow to take off), and each addition to the program posed tough puzzles for us to solve. But beyond the intellectual challenge, there was the fun of working with Bill. Perhaps the best way of conveying the experience is to describe a typical day.

At some point, I left UCLA and moved to the Midwest, but I flew to Los Angeles regularly to put in week-long development pushes, and when I did, I was a guest at Bill's house. We rose early and opened the office. Not all the work was software design. The company was small, so sometimes Bill and I started the day by setting up PCs or assembling tables for them to sit on. At some point, though, we would address the current development challenge. Then the yelling would begin.

Bill and I disagreed about almost everything, and we expressed ourselves colorfully and at top volume. (I remember, in particular, some arguments about hypothesis testing in the linear model.) I would suggest a possible approach. Bill would reply that the approach was obviously impossible and that only a seriously impaired intellect would

suggest it. I would reply in the same vein. We were so loud and so heated that I think we scared the rest of the staff. At least, they didn't interrupt us very often.

There was never anything personal in our arguments. Bill and I were simply passionate about getting the best possible solutions into Stata. If I was able to prove my point (something that didn't happen often enough for my ego), Bill would stop dead and say, "Well, I guess I'm wrong", and things would go on smoothly until the next disagreement. More frequently, I would see the flaw in my approach and come around to Bill's point of view. At the end of the day, the arguments would stop and we'd drive companionably to Bill's house where he would prepare a gourmet dinner accompanied by some truly fine wines. (My recommendation: if Bill ever asks you to dinner, accept. He's an excellent cook.)

My professional association with Stata lasted about ten years, but my involvement as an end user has never slackened. The meandering path of my career has given me the opportunity to build research teams in several large financial services firms, and each of those efforts has required a substantial amount of proprietary model and software development. Stata has played a key role in model estimation, testing, and error-tracking. I've also used Stata heavily as a scripting language and built several automated reporting packages with Stata as the lynchpin. Whenever I start a new research team, I order Stata for the entire staff, but I don't require them to use it. Instead, when they bring a research problem to me, I load their data into Stata and explore the issue interactively with them. Usually only one or two exposures are required before my staff are clamoring for Stata.

I like to think that I contributed one or two decent ideas to Stata. Typically, the best ideas weren't solo creations, but rather the result of discussions with Bill. The idea for the first `parse` command arose when Bill and I were struggling to figure out ways to overcome some of the limitations of do-files. And I think the `pause` command resulted when I was stymied trying to develop a more primitive version of the concept and I asked Bill for help. He realized that it was a tougher problem than it first appeared, and then he went off and developed the more elegant and useful `pause` command as a response to my question. Actually, many Stata users around the world have contributed to the evolution of Stata. I had the privilege of meeting (at least electronically) some of these contributors and previewing their enhancements when I was editor of the *Stata Technical Bulletin*.

Being present at the birth of Stata was a stroke of good luck for me. The creative challenge was intensely satisfying intellectually, and working with and getting to know Bill was a high point personally. And I did get a 19-inch RCA color TV.

Speaking of TVs, my wife has been lobbying for a high-end flat panel HDTV with a state-of-the-art surround sound system. Maybe I'll give Bill a call.

About the Author

Sean Becketti is an economist specializing in financial services research. He assisted Bill Gould in the launch and early development of Stata and was Editor of the *Stata Technical Bulletin* between 1993 and 1996.