The Stata Journal (2005)
5, Number 1, pp. 19–31

A conversation with William Gould

H. Joseph Newton
Texas A&M University
jnewton@stat.tamu.edu

Abstract. William Gould is President of StataCorp. He was born in Burbank, California, on January 21, 1952. He received a B.A. in economics from UCLA in 1974 and a C.Phil. in economics from UCLA in 1977, after initially majoring in physics and then engineering. He studied economics in the Ph.D. program at UCLA and was simultaneously a Research Fellow at The Rand Corporation. He did not turn in his dissertation in labor economics before becoming a Senior Research Associate at the National Bureau of Economic Research in Stanford, California, in 1977. In 1979, he become a Senior Economist at Unicon Research Corporation in Los Angeles, a company that he helped to found. He cofounded and served as Vice-President of Computing Resource Center in 1982, the company that went on to develop Stata. Bill became President of CRC in 1990 and, in 1993, CRC was renamed StataCorp.

This conversation took place on December 7, 2004, in Bill's office at StataCorp.

1 Beginnings of Stata

Newton: How did the first version of Stata come about, and what sorts of things could it do?

Gould: The first version of Stata was a regression package and really nothing more than that. It did a little bit in the way of calculations, and it did some summary statistics, but it was all built around a regression engine. It was written over a one-year period by me initially and by Sean Becketti, who helped me later. I wrote the C code; Sean Becketti helped me a lot with the design. I would say that half of the design is mine and half the design is Sean's in terms of what the user actually saw. A number of things became available just at that time when we started this project, and it was those things that actually caused the project to start. The first C compiler was available for the PC.

Newton: When was this?

Gould: 1984. This is our 20-year anniversary. Our official release date was 1985. The actual release date was December 1984. There was the *American Economic Association* meeting in Dallas. That was where we first announced Stata and started selling it. The meeting is usually around Christmas or New Year. I started about a year before that, so that puts the beginning date at about January of 1984. The Lattice C compiler had just become available for the PC, but no one cared because nobody was interested in C—everyone talked about Pascal as *the* language. I had learned C earlier, and I was very interested in C. No part of Stata was developed under Unix, but I was subtly familiar with the stuff, and so once those development tools became available, that really made it possible for me to write something like Stata.

Newton: When you started, did you have syntax, grammar, or anything like that in mind?

Gould: Yes, right from the beginning. For two reasons: The major reason—What was the term that [Brian W.] Kernighan used when describing the creation of Unix?—"Salvation through scarcity". Stata benefited exactly from that. These were really small computers, 64K memory. You could not afford to have any replicated code. Everything had to happen just once so that the centralized parser and a centralized grammar of Stata could understand statements in the absence of what the particular command was or anything else. That was a really big feature in terms of the code, but that was also something that was right from the beginning a very important design criterion (something I really wanted to see in a package). I was a SAS user, and I had nothing but respect for SAS, and I still have nothing but respect for SAS, but in those days—and I suspect it is still true—anybody who was a real SAS programmer walked around carrying the SAS manual under their arm because nobody could remember how you were to do something. It varied according to the problem as to how you were supposed to do it.

Newton: Did your grammar come from reading other stuff, or did you make it up?

Gould: We made it up, and it was largely based on Wylbur as the basic grammar. Wylbur had a very simple grammar.

Newton: Wylbur, the old operating system?

Gould: Yes, it had a real simple editing type of grammar. And so you could say `list /l` and `list 5/10`. You will notice Stata has a `5/10` and `/l`. The keyword `in` was stuck in front of it because, unlike an editor, which had just one line, we had multiple columns per line per observation. And, so the grammar was made up just kind of willy-nilly, and then it went through improvements over time. There were a lot of unnecessary words in the earliest version of the grammar that only Sean Becketti ever saw. It was `replace x with 4`. There was a `with` keyword; there were a number of keywords that all got taken out and merged into a single equal sign. We overloaded the equal sign too far as we were cleaning up the language because we finally decided that weights could be done with just an equal sign. So you would say `regress miles-per-gallon length` or `regress something something` I don't care `= population`, and you just put an equal sign there because the expression was good enough to designate the weight. Of course, at that time I thought there was only one kind of weight, having never really thought about it, and a lot of people hadn't in those days. We borrowed from Wylbur; we borrowed from Unix, and we borrowed from CMS, the operating system at IBM, that is, IBM/CMS. There were some good features of all these, and we borrowed syntax left and right. Like I said, it was very wordy, and Sean and I working together got most of the words out of it.

Newton: So Becketti was a major force in the beginning stages?

Gould: When this thing started up, Finis Welch suggested that we bring Sean Becketti in; I think he basically just wanted us to bring somebody in. Finis Welch is my partner and was my advisor in graduate school. I learned all my economics, econometrics, and business from Finis; probably the most important person in my intellectual life.

2 Learning computing skills

Newton: Where did you learn your computing skills?

Gould: That had come earlier; I was into computers from the earliest age. In 8th grade was the first time I saw a computer for real. It was a really strange situation, SDC (System Development Corporation), which was a spinoff of Rand Corporation (I grew up in Santa Monica). They had a computer, and Q-32 was the number of it, and it had been designed to be used on the DEW line, the line of radars north of the US that were to be an early warning of Soviet missile attacks. I don't know if it ever was or whether that became dated before it ever happened, but this was a huge computer—one of the biggest in the world at the time with rooms upon rooms of vacuum tubes, and they developed an online system for it, and I guess they were trying to figure out some reason for this computer to exist. They got a government grant to be involved in teaching. They put two teletype terminals in our junior high school in Santa Monica one year. I learned how to program. I just thought it was the neatest thing.

Newton: What language?

Gould: TINT, and there was a pint-sized version of it called PINT. I can't remember what TINT and PINT stood for ... teletype ... interactive ....

Newton: Was it an operating system or basic programming?

Gould: At this time, there was not a big distinction between operating systems and applications. It was a big, monolithic thing. And, TINT was it; it was the language you talked to this online system. It was kind of a basic FORTRAN-type language.

Newton: Was it compiled?

Gould: No, it was an interpretative online system.

Newton: Have you ever used APL?

Gould: No.

Gould: Once I saw this computer and got to play with it, that summer during summer school I took typing. I think I was the only male in the class. Back then, it was women who took typing, and men who took other things. They took shop or something. And, there I was in this typing class learning how to type because I wanted to type on the teletype faster. From then on, it was amazing how, if you looked around you could find computers. I found one at my father's office, and then I learned FORTRAN. FORTRAN II was the first serious language I learned. When I got to high school, they had just gotten a 1620 computer in, and there was some guy they had hired who was trying to make it work, but he hadn't succeeded yet. I found this computer room with this computer in it, and a FORTRAN compiler, and so I made a reputation for myself as writing the first FORTRAN program that worked on this computer, and what it did wasn't much.

Newton: "Hello, world", or something?

Gould: Yeah, basically, or the equivalent. So I learned 1620 machine language in high school and FORTRAN better. I got my first real job in computers when I was a senior in high school. First as a, well ... there is a story that goes with it. In my junior year of high school, there was a dance coming up, and two friends and I decided we would write a computer dating program that would match up people. They would fill out a questionnaire, and so the committee went for it. They gave us a couple hundred dollars to do this. We wrote two versions of the program. One we wrote in FORTRAN II, and the other written in PL/I. We had disguised ourselves as college students and had gone down and used the Bolter Hall engineering computer at UCLA, and debugged our program there. It was kind of cool. People filled out their cards, and we typed it all in. The dance was the next night. We started our 1620 running the FORTRAN II program. It sat, and it blinked lights; and it sat, and it blinked lights. We were getting really kind of nervous. I don't know if you remember the IF SENSESWITCH. It was a statement of FORTRAN. So you throw this switch up, and as soon as the switch was thrown up, it started putting up debug information, and you could see where the program was. We figured out it was going to be finished sometime next week. We were in trouble. It was late at night that Thursday, and because SDC had done this thing with the junior high school, I said, "Let's go down there, and see if we can buy computer time from them." So, we go down there, and of course there are these guards at the desk. This is an aerospace firm. They are looking at these kids saying what are you guys doing here?

Newton: Sounds like a movie.

Gould: Yes, but the computer operator who was running the computer room that night said, "Sure, I'll run your stuff." He ran it! And, so we were leaving, and he was going to put it on at some point, and we didn't know how long it was going to take. He came running out to the parking lot just as we are getting in the car because it had run. This was like a 360–90 something; it got it done in a blink of an eye. Later, there was a little start-up firm that I had heard about that needed a computer operator, so I went down to apply there. And, who do I find there, but this same computer guy who is now working for this firm. He is Vice President of this little company. It's a spinoff. They are all guys from Rand or SDC. So I got a job there, and I worked as a computer operator and then as a programmer.

Newton: So you learned a lot of your computing on the job?

Gould: On the job. Absolutely! At that time, there were not computer science departments, there were a few computer science courses in the math department.

Newton: So you got Stata started, you and Finis Welch, as partners. Were you the only two partners?

Gould: We were the only two partners.

Newton: And Sean Becketti was involved?

Gould: Sean Becketti was hired to assist me, but he turned out to be very important. He was really great.

Newton: How long was he involved with Stata?

Gould: He worked for us for about two years while in LA, and then took on consulting status after that. We were a startup; I wanted him to stay, but he decided to continue his academic career. He moved to the Midwest and continued to be a consultant with us.

Newton: I know later he was one of the editors of the *Stata Technical Bulletin*. In fact, in the testimonials that accompany this interview, he is one of the people who had some things to say.

Gould: He sort of understated his contribution.

3 The CRC

Newton: So, you got it started, you're a couple of years down the road. What were you worried about then?

Gould: Sales! What you worry about is sales. You have to realize in those early days selling a copy was an exciting event for that whole day, selling two copies was unbelievable. Fortunately, the company that developed Stata, CRC, had a small mainframe operation, and that's where the money came from, so there wasn't a huge concern with Stata supporting itself, and there was no way that it was supporting itself.

Newton: Did CRC stand for something?

Gould: Computing Resource Center, and this goes back to the old time-sharing days.

Newton: Did it sell time-sharing?

Gould: Yes, it did, to people interested in doing statistics.

4 The arrival of Bill Rogers on the scene

Newton: Who was the next person who had an impact on Stata?

Gould: That was Bill Rogers, and there was certainly overlap between Bill and Sean. Bill was a statistician at Rand Corporation at the time. Bill is a really fine programmer. He had written a little editor. It was a very nice editor, a program called ESP. Had he marketed it right, he probably would be very rich right now, owning some giant software company that makes editors. But you know how these things go; he never dedicated his days and his nights to this editor. He came on board to help me with Stata because I wanted somebody at this point. Finis had convinced me early on of that. I could not be more pleased that he did. Now after Sean was suddenly not going to be around, I was in a panic. Finis suggested that I talk to Bill Rogers because I knew him. And I did know him; he is very approachable, very nice guy. So I did talk to him, and he was interested. He had known about Stata early on. He'd even had a little bit of input during the very earliest developments. So he agreed, and there were some things we wanted to add to Stata. He is the person who was responsible for Stata's matrix programming language, for the first version of `ml`, factor analysis, and some other features of Stata. He knew statistics, unlike me at the time, and that worked out really well. He added the Huber stuff, robust stuff, survey stuff, whatever you want to call it, which became very important later. At that point, it was just called the Huber stuff, and nobody seemed to be very interested that we had it, or cared. There really was no market for it. He explained to me that there is a difference; that all weights are not the same. Now I understood that, but at the time, it was "Gosh, that is interesting"!

Newton: Were weights something that were involved right from the beginning?

Gould: Weights were involved right from the beginning because some economists would look at state-weighted data. That is, they would look at 50 observations, one for each of the states. Remember the first version of Stata; you really thought that you were not going to be dealing with people processing census datasets. You thought you were going to be dealing with people analyzing summary datasets. And quite often those datasets would have average income in the state, average education in the state, etc., and you'd weight by population.

Newton: So it was very natural. One of the strengths of Stata is its weighting system and the things you can do because you have those capabilities. I grew up as a SAS user, and I don't remember doing a lot of things with weights.

Gould: SAS didn't really have those extensions in the those days; nobody did. They had weights, but what they had in mind is that you had cell average data.

Newton: So now you have made several versions, and Bill Rogers has made a big contribution. Do you have a memory at what point when Stata reached a place . . .

Gould: I sort of remember it in a negative way. There came a point where the financial position of CRC was not looking real good. If you sat down and started looking at the numbers, what you discovered was that Stata was carrying the mainframe, not the mainframe was carrying Stata.

Newton: Is that right? Do you remember what year that was?

Gould: That is just before we moved to West LA. We were in Santa Monica, and then we moved to West LA, and that is where the mainframe wasn't. Stata was making money, not a lot, but it was covering its cost. As long as you were used to living on a low income, and I was.

Newton: Is it fair to ask how many users there were?

Gould: Back then? I am not even sure I know. There were probably 500–1,000 users, something like that.

Newton: I promise I won't ask what it is now.

Gould: Enough to support all of this. [Gould raises his arms to indicate the building the interview was taking place in, and the second building Stata is constructing on their campus.]

Newton: At some point, you are going to move from LA to College Station, but let's talk about the language first.

Gould: Let me help you with all the moves because it gets a little confusing. Stata development began in Santa Monica. We were located on Wilshire Boulevard. We then moved to West LA, and we were there for three or four years. Then, we moved back to Santa Monica, to a 6th Street address. I guess we were there for three or four years, and then we moved to Texas.

5 The birth of ado-files

Newton: Let's talk about the language for a minute, and then we will talk about the move to Texas. How did `.ado` files come about?

Gould: By accident, that's not the official story. At the time, we talked about how brilliant we were. There was a grain of truth to the party line, but like all good things there is intent and there is accident—that come together fortuitously. That is exactly what happened here. From the beginning we knew this was important, even before Stata 1 hit the market. Stata was going to be a package that was much more open to statistical users to be developers then any other package was at that time. The way the market had developed was that the earliest packages were written by people who analyzed data; they really weren't very good programmers. But, they were awfully good at data analysis, and they wrote exactly what they needed, and they were sort of crude. And, you have BMD back then and other packages. Those packages became popular, and they spun off other companies, and then other people started working on them and making them better, and then you ended up with BMDP and with SPSS based on the BMD code base. You ended up with SAS, but that was not developed off another code base. At this point, you have really good programmers, as well as data analysts, working on these packages, but the bottom line was that if you were interested in adding a new feature to any of those packages, you had one choice: You needed to fill out an employment application to work at one of those companies, and that was it.

If you wanted to see those new features, you had to beg and plead with those guys to see it, or it wasn't going to happen. In Stata, there was a `program define` statement long before we actually announced it. We were playing with it—we knew we wanted to have programmability.

Newton: Extensibility?

Gould: Extensibility—that's exactly right. Once we were satisfied with it, we went ahead and announced it. At that point, we used `.do` files, so you would just put together a whole kit of programs, and that is what we called them. You would have to load them, so we had a data-management system and a graph kit, and a stat kit, and you had to load one of these kits, and Stata would think awhile and then get it loaded and then you could use any of those command that were in there. If you wanted to do data management, unfortunately, you would have to load the data-management kit because it wasn't possible to have all these things loaded at one time. We were coming up on a release, and Bill Rogers was running late, and he needed a few weeks to finish off factor analysis. I was done with everything I was supposed to have done, so I was looking for something to do. I went back to my office and thought awhile and said I could do this thing we could call automatically loaded `.do` files.

Newton: What platform were you doing this on?

Gould: Probably DOS, no certainly under UNIX. The first versions of Stata were developed under DOS, but once Interactive's 386/ix became available for 80286s and Sun-3s became available, I did all my development in UNIX.

Newton: But you tried to maintain the portability between these two platforms?

Gould: Absolutely right. We maintained portability. We were constantly compiling it and making sure it worked.

Newton: PC side, UNIX side, but not Apple, there was no way to do it at the time?

Gould: Bill Rogers did the first Apple port but at a later time. Once we had UNIX, we did all of our development under UNIX, and that is mostly because UNIX is a better multiuser. Computers were slow. The Sun 3 was an awfully slow computer, but I could get a lot more work done on the Sun 3 in a day than I could on a 386.

Newton: Plus it had all the tools?

Gould: It had all the tools. What did it matter that I typed `compile` and the thing had to sit there 15 minutes, when I could put an ampersand at the end of it and go on working on the next piece. Everything was just sort of easy.

Newton: Right. In this period of time ...

Gould: Oh yeah—ado-files—, so Bill says, yeah, sounds like a good idea. And neither of us had any appreciation for what a brilliant idea it would turn out to be. We just had the time, and I was looking for something to do. It sort of fit in with the direction we wanted to go. Of course, once we had it, we discovered—my goodness—all the things we could do with it.

Newton: Was it born fully formed? Did it have `syntax`?

Gould: Not `syntax`, there was an earlier form called `parse`. It had actually existed before ado-files. You have to understand, the whole programming language existed before there were ado-files because we knew we wanted that component. We just hadn't figured out that one program, one file: look on the disc, load it when you need it, and throw it away when you don't need it anymore. It was the only thing we missed, so we sort of got it backwards. If you'd thought about it right, we would have done the `.ado` file first and then added more programming. We had programming and then `.ado` files.

Newton: We are getting far afield here, but you could parse in an interactive mode in a `.do` file?

Gould: No, you've got to understand. Stata had `program define`, so you could add a new command to Stata. Now when is Stata going to know you have that command? Well, you put that file in a `.do` file. You load that file, and then Stata would have that command. When you were tired of using that command, you would erase that command because maybe you needed the memory back. All memory management was on you. All loading of the file was on you, but you could do this. So you had extensibility before there were ado-files. `.ado` files just made that extensibility easier to use, so we shifted a lot of our own development in that direction.

Newton: So originally almost all of Stata was written in C, and then it started moving more and more into ado-files. So what fraction now is C, and what fraction ...?

Gould: There are two answers to that question. If you just say I am going to draw features randomly, I would guess that 85–90% of Stata is ado-files, and 10–15% is built-in. If you say, let's draw using the probability that a randomly drawn user uses the commands, you will probably find that 50–60% of it is built-in and 40–50% is ado-files.

Newton: Because most people are doing similar things most of the time and they are highly numerically intensive.

Gould: Yes.

Newton: I've enjoyed our conversations in the past about speed. Is speed as important as it used to be?

Gould: No, less important. I don't want to say it's unimportant. The faster you can do things, the better; people are doing a lot more bootstrapping and that kind of stuff. You have to look at how much faster CPUs have gotten, and you have to realize how slow the old 8088 was and the 80286. Speed was everything in those days.

Newton: What about size?

Gould: And size.

Newton: The thing had to fit on a disk.

Gould: It had to fit on a disk, it had to fit in the memory partition allocated to it. That was very difficult to make happen.

Newton: Right. So, we've got `.ado` files, and you are gradually adding more and more features every version. At this point, ...

Gould: Speed really stayed up, even though I said speed wasn't important, the speed kept up. Most software has developed a lot of new features that run slower, and you need a bigger CPU. Stata, every version, has a lot of new features, and it runs faster, even on the old CPU.

Newton: That was an unusual phenomenon, particularly on PCs. That was why you had to buy a PC every six months because the software got so much more extensive.

Gould: That's right. Well, we only recently, in the last two or three development cycles where we have thought, oh we can just go out and grab 16K–64K of memory. I have to remember because I still think in an old fashioned way, "Whoa, that's 64K!"

Newton: I think the same way I am afraid.

Gould: Not a bad way to think. That is what keeps Stata fast because resources aren't free. They cost something, they have just gotten hugely cheap. If you start acting like they are free, then you start generating software that runs slow.

6 The move to Texas

Newton: Let's now talk a little about moving to Texas.

Gould: Okay, the move to Texas was actually caused by the success of Stata. By the point we wanted to move to Texas, we knew we had finally made it. That meant that I was not going to have to change careers and go find a real job. I could plan on doing this until retirement. We were growing. We had more sales than ever. Things were actually starting to look really good. Money was coming in, but what we couldn't find were people to work for us. You could hire anyone in the LA area, but you couldn't get anyone to move to LA. No way someone was going to move from the Midwest to Los Angeles and trade in their home for a condo. It was impossible, and we were suffering from it. We had Bill, and he was wonderful, but we needed more, and we couldn't find any of these people. Technical support was even harder. You have to remember in those days I was technical support, and Bill Rogers was technical support. Whoever knew Stata or knew Statistics was technical support. We really wanted to move to a college town where we could get graduate students to be on technical support. Finis Welch (who I have previously mentioned) had by that point moved to Texas A&M University Economics Department. He of course wanted us to move to Texas. I knew that we did not want to move to Texas. I was a typical Californian, and I had this view of Texas. It was sagebrush, and flat, and sandy, and full of rednecks. That was Texas. But I agreed that we should at least look. So my wife and I came here, and we went to Houston where Finis had an office for his consulting business. We came up and looked at College Station, and I think I even met you at that time. Texas was completely different from anything that I had expected. I liked it, and the people were friendly. It was also very scary moving Stata from Los Angeles. You have to understand that LA is a huge city, and all cities look small after you've lived in LA or Manhattan. Houston may be the fourth largest city in the US, but it is small compared to LA where you can drive for an hour and a half without leaving the city. It was very scary to think about moving, but one of the things College Station had going for it is that we had a lot of users in Houston at the Medical Center. If College Station did not work out, we could shift to Houston without actually tearing the guts out of the company with another big move. That made College Station very reassuring. It seemed very small compared to Los Angeles, but it seemed safe. Finis was here, and that counted. A&M seemed friendly, so here we were. We showed up and never wanted to leave. There has never been an instant when we ever even thought about it.

Newton: So you've actually become sort of a Texan?

Gould: I guess I am a Texan. If you've lived here for more than 10 years, does it count?

Newton: I've lived here 26 years, so I guess I think so.

Gould: I have a cowboy hat.

Newton: Yeah, I guess we're Texans now.

Gould: I guess I am a Texan once I think it is okay to wear it to work.

7 How new Stata features happen

Newton: I remember fondly when Stata did move to College Station. It's been great for A&M, too. Let's talk about a few of the basic capabilities that I see a lot, like `ml`, bootstrapping, etc. How do such things develop?

Gould: It is not that I sit in my office and I have some plan, I see a vision out there and I say "yeah that's where we are going". At any one time, I think I know where we are going, but I can tell you I have never been right yet. What you try to do is find a direction to go. It's like a hill climb—it's just like a maximizer: "This looks like a good way to go, but don't go too long before you evaluate the derivatives again." One of the things that has really worked well for Stata is things like Statalist, the *Stata Journal*, the Users meetings that were started in London, that are now going on in Boston and Berlin, and other places. We get a lot of user feedback, but talk is cheap. Someone calls you up and says, "I want to see this." If we implemented every time someone calls us up with "I want to see this", it would be ridiculous, and you couldn't do it. But we actually get some real indication because Stata is programmable; we are able to go to the Users meetings; we can see that people are writing programs to do such and such. That is what we pay attention to. We say look where they are going, and then we follow. We see where the users who are active are going, and then we lead from that direction and say, “Follow us!”

Newton: That is one of the great things about extensibility.

8 The Stata Journal

Newton: Once the Internet sort of made the *Stata Technical Bulletin* not as badly needed, a group of us got together in Boston and said, “Let’s create a journal.”

Gould: I think that was really spawned a lot by Nick Cox. In part, StataCorp’s concern was what was going on with the *Stata Technical Bulletin* because we knew it wasn’t working as well as it used to work. Remember the *Technical Bulletin* was primarily a means of distributing user-written commands.

Newton: Well, it wasn’t as needed?

Gould: Right.

Newton: Many small articles describing new commands weren’t as needed because you could do that on the Internet.

Gould: Or in the Boston archive.

Newton: So several people got together in Boston and said, “Wouldn’t it be great to have a journal?” And you agreed to do that.

Gould: Again this is once more—you know how some restaurants are run where they have a window onto the kitchen? They do that for a couple of reasons. One is that it is sort of entertaining for the diners to be able to look into the kitchen, and the other reason is that it keeps the kitchen clean because those people know they are being looked at and it sort of monitors itself. That is exactly what we have tried to do with Stata in terms of the *Stata Technical Bulletin*, Statalist, and now the *Stata Journal*. You notice we have always put them under independent management. We don’t decide, we put up with an independent group deciding. We observe what they do, and sometimes they surprise us and go in directions we did not anticipate them going. Had we been in control, we would never have let it happen because we would have stopped it early on and said that is the wrong direction. Look what they are doing! Like the Boston Archive. The `net` command worked out wonderfully but didn’t work out like I expected. I thought there would be no big single place, that it would be scattered all over the place and designed the command to work in that environment. Well, it turns out that up pops the Boston Archive, and most things go there. Fine! I didn’t anticipate it. I was wrong not to anticipate it, but I wasn’t in control of Statalist where I could say, “No, no, no, Kit [Baum], that’s not the direction we are going.” Kit was absolutely free to get out there and say, “I have built this thing, and here it is.” I didn’t think it was the direction that ultimately we would end up going, but I was wrong. This is not me being good; it is not a moral position, it is actually in our interest to let the users go. Again, leading from behind and sitting back and watching them. See what they want to do, and then we try to catch up.

9 Modern StataCorp

Newton: One thing I have always noticed about Stata is that you don't have hordes of programmers here. Is that a fair statement?

Gould: You make us sound small!

Newton: Of course not! You are not small, but you personally write some of the code.

Gould: I am a big believer in using very competent and senior level developers and giving them a lot of leeway, and expecting a lot of them, rather than a bunch of coders executing instructions from on high.

There was a time when I was Stata, but that time has come and gone. I get interested in a project, and I work on it, and I have great fun doing it, but now there are a lot of people working on Stata.

Newton: And you spend the rest of time on business?

Gould: About 50 percent. Finis Welch has been the invisible partner in this from the beginning, and this is the person I talk to a lot. Stata would not be here but for Finis Welch. I could have put this company out of business 15 times between start up and where we are now if somebody had just given me the money with which to do it. There is a lot of learning by doing. Finis has been very good all along in letting me make my own mistakes but keeping me from making mistakes that are too big.

Newton: Finally, what can you tell us about the future of Stata?

Gould: Like I said, we do not set long-term goals, at least with the expectation of keeping them. We say where it is that we think we want to go, and then we work out what we ought to be doing for the next one or two years. By that time, the long-term goal will have shifted, and the process repeats.

Nevertheless, we are committed to statistics. This is what we are. This is what we do. We are committed to the professional research market. We are not going to make a stab at telling the academics and researchers to go away so that we can focus on the Fortune 500 and turn Stata into a system for use by people who do not know statistics. We assume that the person behind the keyboard knows more than we do, not less, and that is the market we are going to continue to serve, without question.

Newton: Thank you Bill. I've certainly enjoyed this conversation, and I look forward to seeing all the exciting things in Stata's future.