The Future of Statistical Computing
Author(s): Leland Wilkinson
Source: *Technometrics*, Vol. 50, No. 4 (Nov., 2008), pp. 418-435
Published by: Taylor & Francis, Ltd. on behalf of American Statistical Association and American Society for Quality
Stable URL: http://www.jstor.org/stable/25471520
Accessed: 29-07-2016 13:46 UTC

The Future of Statistical Computing

Leland WILKINSON

SYSTAT Inc.
225 West Washington St.
Chicago, IL 60606
(*leland.wilkinson@systat.com*)

This article forecasts the path of statistical computing in the next decade. Its premise is that technology will influence statistical computing more than other factors. This forecast is based on contemporary observations of the field over the last 40 years and on a supposition that extrapolating these trends is not unreasonable. The technology driving this forecast includes not only hardware, but also the software that provides the infrastructure for individual and community interaction with computers. We should not be surprised to see a proliferation of intelligent data analysis systems embedded in everyday objects and Web sites; automated visualizations for data discovery; analytic systems that are accessible by nonstatisticians (a trend toward simplicity and away from comprehensiveness); distributed analytic systems that talk to each other, fuse disparate data in real time, and draw conclusions on the evidence; and communities of open-source developers exceeding the scope and capabilities of commercial companies. Whether computer scientists eventually take over this field will depend on how actively statisticians participate. Statisticians interested in statistical computing and its future incarnations will have to engage in joint research with computer scientists to continue to have an influence.

KEY WORDS: Computer software; Statistical graphics; Visualization.

1. INTRODUCTION

Technology transforms science. The telescope dispelled the scholastic view of the cosmos. X-ray diffraction revealed the structure of DNA. The microarray accelerated genomics. Not all of the effects of technology on science are positive, of course; how many scientists have wasted thinking time on installing device drivers, rebooting the blue screen of death, or wading through spam? And some effects are mixed. Easy-to-use structural equation modeling software, for example, has increased the number of correlational studies in the social sciences at the expense of designed experiments (Hershberger 2003).

This article concerns the likely effects of future computing technology on statistical computing. It presumes, of course, that we can predict future computing technology. It also presumes that we can infer the effects this technology is likely to have on the ways we analyze data. And it presumes that we are fearless, because technology predictions have a steep survival curve, and everyone loves to lampoon old ones.

1.1 The Perils of Prediction

Bold predictions can seem foolish:

> Everything that can be invented has been invented
> Charles H. Duell, U.S. Patent Office (1899).

Fortunately for Duell's reputation, there is no evidence that he actually said this (Sass 1989). Unfortunately for Duell's reputation, the quotation is cited almost 50,000 times on the Internet. We can do our best to predict boldly, but we have to be prepared to be misquoted and misattributed, especially on the Internet.

Prudent predictions can seem quaint:

> Where a calculator like the ENIAC today is equipped with 18,000 vacuum tubes and weighs 30 tons, computers in the future may have only 1,000 vacuum tubes and perhaps weigh only $1\frac{1}{2}$ tons (Hamilton 1949).

This prediction was proportionate to the contemporary evidence. Although the transistor had been invented at Bell Labs 2 years earlier, there is scant evidence that anyone envisioned transistorized computers, much less microcomputers, in 1949. Moreover, there is scant evidence that any of the thousands of Internet bloggers who have quoted (or more often misquoted) this statement have read the original article.

In short, we cannot expect to predict the effects of disruptive technologies (although see Talmage 2008 and the remarkably prescient video *1999 A.D.*, produced by Philco-Ford in 1967). As a Bell Labs engineer is said to have noted,

> Asking us to predict what transistors will do is like asking the man who first put wheels on an ox cart to foresee the automobile, the wristwatch, or the high-speed generator (Farley 2007).

Regardless of the risk, this article contains many predictions concerning future trends in computing, more specifically, the relationship between future trends in computing to future trends in statistical computing. How will we be analyzing data 25 years from now? How will our data-analytic needs affect the development of computing methods? How will computing developments affect how we look at data?

Because these questions involve a relationship, we need to consider where computing itself is headed in the next few decades. There is a growing body of predictions in this area, some of which we cite. We attempt to discern how future software and hardware might affect statistical computing, similar to how the development of the microprocessor in the 1970s affected desktop data analysis in the 1980s.

We also consider where statistical computing is headed. We discuss trends that reveal impacts similar to the way in which the data mining movement in the 1990s affected grid computing and database design in this decade. Before we do this, however, we consider the most important past predictions regarding the future of statistical computing.

TECHNOMETRICS, NOVEMBER 2008, VOL. 50, NO. 4
DOI 10.1198/004017008000000460

1.2 John Tukey

Almost half a century ago, Tukey (1962) introduced statisticians to a new field, called *data analysis*. Today, many practitioners of data analysis or data mining take for granted the arguments, in that and subsequent articles, that upset many of Tukey's contemporaries.

Many of Tukey's ideas are now conventional wisdom. For example, Tukey accorded *algorithmic* models the same foundational status as the *algebraic* models that statisticians had favored in the previous half-century (see also Breiman 2001). Tukey also emphasized the recursive aspect of model building; information in the residuals should induce an analyst to go beyond checking assumptions to consider alternative models. Furthermore, Tukey urged us to explore data for surprising insights, to "let the data speak" (although it is unlikely that Tukey ever used this popular descriptive phrase). And, perhaps most significant for the future that he forecast, Tukey argued that data analysis would become so computationally intensive that it would push the limits of existing computer systems. In a talk at the 14th Interface symposium (Tukey 1982), he noted that:

> this means (i) large systems, (ii) systems planned both for growth and for easy specialized attachment, (iii) cooperation between a variety of insightful data analysts on the one hand and a variety of computer experts on the other—each group with diverse skills. Success will not be easy, but starting now poses no major barriers. There are people with enough insights of the needed kinds, though they may be hard to find and assemble. And we can expect the 4th or 5th generations of such systems to be far, far better than anything we have today.

Although Tukey started a revolution in computerized data analysis, he was not addicted to the computer. The Blackberry user, the video game player, the social network groupie—these prototypical users probably spend more hours on the computer each day than Tukey did. Indeed, while proficient on many operating systems and hardware, Tukey nevertheless advocated paper and pencil throughout the pages of *Exploratory Data Analysis* (Tukey 1977). His overhead transparencies were more effective than some people's PowerPoint presentations. Tukey understood that a computer is at its best a mental amplifier, not a toy. Like Herbert Simon and Marvin Minsky, Tukey was able to recognize the wide-ranging potential of computer technology for research—for thinking itself. In contrast, most statisticians of his era concerned themselves with statistical packages and subroutine libraries. They projected a future in which large multivariate models could be solved with faster and larger computers. They looked forward to high-resolution color graphics and three-dimensional scientific visualization. Tukey thought instead about intelligent analytic systems, interactive graphics (Fisherkeller, Friedman, and Tukey 1988), automated graphics (Tukey 1982, 1986), and analytic assistants for working scientists.

Some of the predictions in this article follow Tukey's because some of his predictions have yet to be realized. Although *The Future of Data Analysis* is in our past, it remains part of our future. This article does not address perhaps the most important consequence of that article, however. As Donoho (2000) argued, Tukey tore apart the world of statistics, and it may take a century to reassemble the pieces. Donoho foresaw a period in which mathematicians will formalize many of the high-dimensional problems that Tukey attacked, as classical and Bayesian statisticians formalized inference in the last century. More recently, Efron (2007) predicted a similar future.

1.3 Jerome Friedman

Tukey's article on the future of data analysis was extraordinarily clairvoyant. More than 3 decades after that article was published, Jerome Friedman gave a landmark keynote address at the 29th Symposium on the Interface (Friedman 1997). In his talk Friedman warned statisticians that a preoccupation with algebraic (as opposed to algorithmic) statistical models would exclude them from the next revolution in statistical computing—*data mining*.

The data mining revolution has now passed. The classic data mining algorithms are available in a variety of introductory textbooks, and Microsoft now includes them in its basic business package called Analysis Services. A decade later, we are in an era of *machine learning*. Computer scientists no longer talk about mining large data warehouses; rather, their discussions focus on fusing data from different sources—sensor networks, relational databases, broadcast media, the Web, and wireless wearable computers. And the systems that they develop are designed to analyze intelligently and automatically the data from these disparate sources. Friedman's projections were correct, however. Those statisticians who ignored the data mining movement have now been ignored by computer scientists.

Like Tukey, Friedman's experiences outside of statistics transformed his view of statistics. (It also could be said that Friedman's experiences outside of physics transformed his view of physics.) In his talk, Friedman spoke of the inexorable push of technology:

> Every time a technology increases in effectiveness by a factor of ten, one should completely rethink how to apply it. Consider the historical progression from walking to driving to flying. Each increases speed by roughly a factor of ten. However, each such purely quantitative increase has completely reoriented our thinking on the use of transportation in our society. A favorite quote of Chuck Dickens (former Director of Computing at SLAC) over the years has been "Every time computing power increases by a factor of ten, we should totally rethink how and what we compute." A corollary to this might be "every time the amount of data increases by a factor of ten, we should totally rethink how we analyze it.

With Tukey and Friedman as guides, this article examines the impact of future technology—for good or ill—on statistical computing itself. Whatever paths statisticians may take in the future, analytic thinking will be inexorably influenced by the ubiquitous computers in our lives. The outcome of all this, as we summarize in the last section of this article, will determine the prevalence of statistical thinking in future technologies. Technological developments will force the statistical profession to redefine itself to avoid becoming a subtopic of theoretical mathematics. If this effort succeeds then statistical computing will become central, rather than peripheral, in the statistics departments of the future.

2. THE FUTURE OF COMPUTING

We begin with the obvious. Computers are now a commodity. Our long-term predictions rest on this observation. From a short-term perspective, journalists and prognosticators tend to pay attention to developments that have revolutionary impact—the invention of the word processor, the invention of the spreadsheet, the invention of the Internet. But often it is evolutionary trends that have greater influence. It can be argued that the proliferation of cheap computers *caused* many of the developments

that we consider revolutionary. Cheap microprocessors *demand* applications. To push the evolutionary metaphor to its limits, cheap and powerful processors provide the soup in which mutations produced by curious and playful inventors lead to amazing new developments. The more prevalent the computers, the more likely the inventions.

It is from this perspective that we view the future. How will the ubiquitous computer stimulate new developments affecting ordinary life and, in turn, the analysis of data? We first consider a world of ubiquitous computers. To avoid footnotes on acronyms, a glossary of computing terms is provided at the end of this article.

2.1 Computers Everywhere

We have already passed from a microcomputer era to an embedded computer era. Our cars are run by computers. The Toyota Prius is a computer. This provocative statement is intended to steer our thinking away from the idea of a computer having a keyboard, mouse, and display, or a computer having a processor, memory, and peripherals. A computer will be increasingly regarded as a body of functions, just as the brain is no longer regarded as a computer sitting inside a body. The Prius driver is a computer operator, and the mechanicals (brakes, motor, engine, transmission) are peripherals. In the Prius, the only direct linkage between driver and mechanicals is in the steering system. In future cars, mechanical linkages will disappear entirely, and the road network, not the driver, will determine the route and the speed.

Computers are now embedded in our cell phones. They are embedded in our watches, in our pacemakers. When we take public transportation, pay tolls, enter buildings, and amble through public spaces, our movements are recorded on video cameras networked wirelessly. In some public areas, we are scanned by face-recognition software. And in some areas of the world (especially battlefields), people and vehicles are followed by remote sensors that respond to movement, vibration, temperature, odor, electromagnetic radiation, and sound. Whether we like it or not, the data from all of these perceptrons are stored in government and private networks. We live in a computational ether. This ether is still relatively thin; our movements are sampled, not continuously monitored. But we are a short step, at least in our cities, from being observed constantly by ubiquitous computers.

2.2 Communities of Computers

We hear often about the new world of networks: cloud computing, social networks, Semantic Web, grid computing. And we hear a lot about networking sites: YouTube, Facebook, LinkedIn, MySpace, World of Warcraft, Second Life. Perhaps the easiest prediction to make about social networking enterprises is that most will be out of business in the coming decades. The novelties that make social networking so fascinating to so many today are likely to go out of fashion, to be replaced with other forms of social interaction and community organization.

Our question about networks concerns what their future topology will be. We know that the Internet today is a multiscale sparse graph (Faloutsos, Faloutsos, and Faloutsos 1999; Baldi, Frasconi, and Smyth 2003). We also know that its structure is under transformation from several sources: commercial companies gaining control of U.S. and overseas infrastructure, governments censoring content, and service providers restricting transmissions through packet-level analysis. We have already seen a stratification to serve high-bandwidth scientific communities (*http://www.teraflowtestbed.net/*; *http://www.gloriad.org/gloriad/index.html*). It is not unreasonable to expect that additional stratification will emerge to satisfy powerful commercial interests. In short, the Internet may evolve toward a a multigraph or hypergraph.

We should expect a parallel evolution in algorithms for distributed computing. We have already seen viruses, worms, bots, and zombies navigate the network, spreading their mischief. These agents depend on the specific protocols and operating systems on the network itself. We should also expect agent-based modelers to exploit similar methods for beneficial purposes.

Finally, to exploit one more time the evolutionary metaphor (and to risk a pathetic fallacy, for imputing motivation and feelings to nature or machines), we need to understand that computers *want* to talk to each other. In some fashion, we will hear their request and implement the kinds of network connections needed to enable this. We have already done so in subnets, where we wirelessly connect printers and and other peripherals to local computers. We should expect that we will do so more globally to enable systems to talk to each other without humans in the loop. We do not anticipate the doomsday scenario depicted in the movie *Colossus: The Forbin Project*, but the premise is not fantastical. Perhaps it has already happened:

> Machines are typically made by other machines these days, albeit with plenty of help and guidance from humans. So perhaps the entire industrial enterprise constitutes a swarm of self-replicating robots (Cho 2007).

2.3 Generic Computers

In software design, the term "generic" refers to software components that are flexible enough to be used in a wide variety of situations. There are many nuances and applications of the term, including the older concept of *polymorphism*, in which an object can perform services dynamically across a variety of parameter types.

The term "generic computers" refers to the exchangeability of hardware platforms. It is related to the term "virtualization," whose recent popularity indicates that we often care less about hardware and software than we do about functionality. We used to care about the precision of registers, the width of buses, RISC (reduced instruction set) versus CISC (complex instruction set) architecture, and other hardware details. Increasingly, these concerns are irrelevant to most users. We do not care about the operating system running our servers or the processors in our cell phones.

Of course, hardware and software designers care greatly about these things, but the differences in human interfaces to computers often outweigh considerations of internal architecture. We care about Windows versus Mac in part because of the human interface to these systems. Windows and OS–X now run on the same processor. Few noticed the performance differences when Apple switched from a RISC to a CISC processor,

and even fewer care that a Mac and the iPhone use different processors running the same operating system.

We should expect this trend to continue. In the future, computers will increasingly be regarded as service providers. The service they provide—processing a phone call, playing a video, searching for a quotation, identifying a thief, monitoring a conversation, recognizing a face, designing a cover for a magazine, diagnosing a patient, choosing a statistical model, spamming a network, cooking a steak, deciding whether to use the brakes or flywheel to slow down a Prius, deciding how closely to follow another car, avoiding a skid or collision—will define their functionality. The same service will in most cases be implemented across a variety of processors and operating systems.

2.4 Immersive Interfaces

Before making the film *Minority Report*, Steven Spielberg convened a committee of 23 experts to help him envision a world of computing set approximately 50 years in the future. Among the members of this group were Harald Belker (a graphic designer), Bill Mitchell (MIT architect), Neil Gershenfeld (MIT Media Lab), John Underkoffler (MIT Tangible Media Group), Peter Calthorpe (urban designer), Peter Schwartz (a futurist and chairman of the Global Business Network), Steward Brand (*The Whole Earth Catalog*), Nat Goldhaber (CEO of Cybergold, an Internet advertising company), Shaun Jones (first director of DARPA's Unconventional Countermeasures program), and Jaron Lanier (virtual reality artist). Spielberg realized that forecasting trends in computing required a committee of Delphic oracles more diverse than a core group of computer scientists.

Among the many ideas from this group that made their way into the film was perhaps the most radical—a holographic wall of graphical analytics that the leading actor, Tom Cruise, manipulated with his hands and voice. This technology stems from the voice and gesture interface, called "Put That There" (Bolt 1980), developed at Nicholas Negroponte's MIT Media Lab.

We are so used to the window, icon, menu, pointing device (WIMP) interface that we seldom realize it is obsolescent. In fact, WIMP was senescent by the time it appeared in commercial software, according to its inventors (Wadlow 1981; Kay 1990). Larry Tessler, working at Xerox PARC when the original windowed GUI was developed, promoted the slogan "Don't mode me in" (Tessler 1981) to describe the problem. The Xerox GUI was designed to minimize the use of modes or state-dependent actions (insert/delete mode, edit/preview mode, etc.). A real GUI (in the sense of following the rules of objects in the physical world) should enable a user to explore and act without worrying about context—a stateless interface. Dialogs, wizards, hierarchical menus, and mouse-click modes are antithetical to this idea.

It is easy to say that WIMP is obsolescent. It is more difficult to identify its successor. Achieving a modeless interface is more difficult than implementing an ordinary WIMP. There are some examples pointing in the direction of future interfaces, however. The most popular is the Apple iPhone. This device has no mouse or windows, yet navigating the Web is simpler and faster than working with a standard browser on a WIMP machine. Other examples are the Microsoft Surface (*http://www.microsoft.com/surface/*), the Cave and ImmersaDesk (*http://www.evl.uic.edu*), Multi-Touch (*http://www.ted.com/index.php/talks/view/id/65*), and BumpTop (*http://www.ted.com/index.php/talks/view/id/131*).

2.5 Competition

It is hard to imagine a field in which competition has been more intense than in computing. The positive effects of this competition are obvious. Apple forced Microsoft to develop Windows. Advanced Micro Devices (AMD) forced Intel to accelerate the introduction of new processor architectures. Netscape and Sun forced Microsoft to pay attention to the Internet. The open-source movement forced companies to rethink their pricing strategies.

We expect such competition to continue. Many of the technology areas affecting statistical computing are beginning to mature, however. As such, we expect that some of the negative forces arising from competition between mature entities will be dominant in the future. We conclude this section by considering some negative influences on technology due to competition. We can forecast the role of these negative influences in the computing future with considerable confidence.

2.5.1 Sabotage. Technologists often marvel at inventions and extrapolate them to a future of boundless gadgets. Few technologists include sabotage in their predictions. Perhaps few enjoy being labeled a cynic. More likely, technical people often assume that inventions swamp their opponents—new toys are so much fun that they cannot be resisted, devices are so necessary for health and welfare that they cannot be suppressed, or gadgets are so *powerful* that they cannot be opposed.

History reveals otherwise. There is too much evidence of sabotage in the history of technology to ignore its effects (see, e.g., Wang 1986; Hsu 2004). Monopolies often sabotage their opponents, but so do associations, religious groups, academics, and governments. There are few innocents in this regard.

Sabotage suppresses competitors. It is the undermining of a competitors' idea or product without offering a superior alternative. In technology, a popular stratagem is often called fear, uncertainty, and doubt (FUD)—make customers or clients believe that a competitor's solution is unreliable or even dangerous. Another is Embrace, Extend, and Extinguish (EEE)—adopt open standards or a competitor's standards, extend them in proprietary directions, and use the resulting differences to put competitors at a disadvantage. Another is vendor lock-in—deny a customer the ability to adopt a competitor's product or service by inserting incompatibilities (e.g., proprietary file formats, sockets, interfaces) that make switching or integration inordinately expensive. Another is legal—sue a competitor that cannot afford the negative publicity or expense of litigation. And yet another is political—lobby a foreign or domestic government or agency to impose regulation that makes a competitor's product illegal or impractical to market.

The sabotage most likely to affect our predictions concerning data analysis lies in the area of Internet technology. The software platforms needed to foster widespread collaborative computing have been subject to considerable sabotage in the last decade, and we expect this to continue. There is no single culprit.

Client-side graphics is a typical example. John Gage, a principal at Sun Microsystems, invented the slogan "The network is the computer" in 1984 (Perry 2004). In the mid-1990s, Sun executives recognized that they could realize Gage's slogan through the Java language, which had been created by James Gosling and other Sun researchers in 1991 as a platform-independent programming environment. Java thrived on the Web for several years until competition between Sun and Microsoft stifled it. The chief FUD spread by Java's opponents was that Java was slow, a claim eventually shown to be false (Prechelt 1999). Nevertheless, Java applets faded as a client-side solution, Microsoft dropped Java from its browser, and other graphics technologies were promoted as substitutes (Flash animation, Scalable Vector Graphics, .NET, AJAX, etc.).

A single instance is sufficient to make this point. Visit *http://mrl.nyu.edu/~perlin/experiments/orange/* to see a ray-traced, dancing female torso that can be rapidly rotated in *any* direction under user control in real time. This applet was written by Ken Perlin using Java 1.0 with no external libraries. It loads in a browser in a few seconds. Imagine a three-dimensional scatterplot, regression surface, or density rotating this fast. More than 10 years after the introduction of Java, one searches the Web in vain to find a competitive example that uses newer technology (e.g., Flash, AJAX, SVG).

2.5.2 Monopoly. Much has been written about the negative effects of monopoly on the proliferation of computing technology. Many writers note that monopolies stifle innovation, but fewer note the tendency of information technologists and end-users to prefer technology monopolies. IT managers do not like mixed-vendor environments. Multiple vendors mean multiple fingers pointing at multiple culprits. And end-users prefer to look for a little help from their friends who share the same operating system. Finally, standards committees (e.g., IEEE, W3C) tend to favor uniformity. Knowing this, we should not be surprised to hear that the business market share of IBM in 1970 is comparable to that of Microsoft today (Ceruzzi 2003).

Some vendors manage to coexist in a monopolistic business computing world as long as they are plug-compatible and do not interfere with a target market coveted by the monopolist. Sooner or later, however, the monopolist will overwhelm these so-called "frenemies." As this happens, innovations are spread to the monopolist's community. It's not clear that this is always such a bad thing (Kurokawa 1997).

We should not expect to see a major change in Microsoft's current monopoly on business computing for at least a decade. The conventional wisdom expected a major destabilization in the 1990s due to Internet technology, but we saw Microsoft eventually adapting to this environment. To be sure, there have emerged steady sources of erosion in Microsoft's dominance—Linux, open-source software, open standards, Apple's exploitation of consumer-oriented technology (iPod, iPhone)—but these have not been sufficient to alter the overall imbalance. Most importantly from our viewpoint, Microsoft has aggressively focused on analytics for business (Analysis Services, Dynamics) and (as with many of its products) has served milk to ordinary users and left the cream to companies like SAS and SPSS and groups like the R Project. The cream is no less important than the milk, particularly for current and future readers of *Technometrics*.

3. THE FUTURE OF STATISTICAL COMPUTING

We now consider the future of data analysis in this future technological world. The major headings are Data, Humans, Machines, and Providers. We anticipate fundamental changes in the way we look at data, the way we understand the analytical user's interaction with the computer, the way the computer interacts with the user and with other computers, and the way analytic software is provided.

3.1 Data

Data are given. They are the information that we receive from a variety of sources in the physical world. For statistical packages, data reside in tables; the rows represent observations (cases, replications, instances), and the columns represent variables. Even the latest incarnations of these packages view data in this way; if received data do not fit this frame, then the packages import data and transform them to a table.

Databases use a relational model: a relation between "rows" and "columns" in a set of tables. The relational model can handle any data structure consisting of indexed sets. Because indexes can point to indexes (pointers to pointers, so to speak), the model is quite general.

Most real data do not fit these models, however. Of course, any form of data (text, numbers, images, video, audio, etc.) can be trivially transformed into the relational model, because a relation can be drawn between any two sets. Forcing blobs of data into a procrustean relational bed may sacrifice efficiency, however. For example, the Daytona database at AT&T is customized for streaming telephony records and can store more than 11 times the data volume of its relational competitors (Greer 2007).

Forecasting the demise of relational databases would be as foolish as forecasting the end of COBOL or FORTRAN. Commercial databases and their clones serve many different needs. Primary among these needs is the control of data and of information flow. To the extent that corporations and governments need to maintain control of relatively static data, relational and transactional databases will thrive. Other data will overwhelm these containers, however. Remote and local sensor data, Web click data, financial trading data (in thousands of trades per stock per second per market), and other real-time data sources will overwhelm the indexing mechanisms of relational databases. Other data, such as free text and dynamic random arrays, will elude standard indexing methods.

3.1.1 Heterogeneous Data. How do we describe these new data sources? One important category has been called *streaming* data (Scott 2003)—sources such as multichannel streams of financial or sensor data. In the future, we also will be concerned with *heterogeneous* data. Streaming data have a built-in time stamp and are thus automatically indexed and presumably storable if enough memory and processor bandwidth are available. Heterogeneous data—images, video, audio, text—are not easily indexed and do not necessarily come packaged in multichannel continuous streams. Fusing these data will require inferential algorithms and heuristics. Even deciding on how indexing should be defined will pose a major challenge. Data processing will have to be integrated into the analytics and statistical models.

Temporal random graphs are a typical example of a heterogeneous data structure. Although nodes and edges can be stored in relational form, this method is ill-suited for dynamic random graphs with different node sets at different points in time. Random graphs analyzed in the future will consist of millions of nodes and billions of edges. Furthermore, these graphs will evolve over time in many real applications. (Social networks provide a hint of what we should expect.) Processing these kinds of data in current networks is impractical for example, Internet game sites occasionally must restrict the number of players because the distributed server farms hosting them are overwhelmed. Experimental systems are emerging to deal with these problems (Stoica, Morris, Karger, Kaashoek, and Balakrishnan 2001).

3.1.2 Data Fusion. We expect that one of the most challenging data problems in the next decade will involve fusing disparate data sources and preparing them for statistical modeling. Analytics will have to be embedded in the process, because raw data will continue to overwhelm storage capacity. Aggregation algorithms, such as data squashing (DuMouchel 2002), will be required to construct archives that can be analyzed further when time permits.

One approach, called *ontologies*, suggests a possible pathway for data fusion in the future (Gruber 1993). An ontology is a data model representing a set of concepts within a domain as well as the relationships between those concepts. Ontologies generalize the relational data model beyond simple indexed sets to a web of relations. As such, ontologies can be used to reason about (not simply retrieve) objects within that domain. Ontologies also generalize the object-oriented data model beyond simple inheritance and aggregation relationships. The ultimate goal of this approach is to model knowledge itself—a metaphysical enterprise. One might say this is a hopeless endeavor, but some progress has already been made (Corcho and Grez 2000).

What will be needed in the future is an inferential engine to construct ontologies by observing data. Current technologies require schemas to be applied before processing and categorizing data. Schemas have been constructed for many domains and are constantly being updated. Not yet practical, however, is the efficient automated construction of new schemas through ontology learning (Decker, Erdmann, Fensel, and Studer 1999; Maedche 2002). Concept learning is a huge problem that has frustrated many AI attacks, but it will doubtless prove to be central to the analysis of data in the future.

3.1.3 The Semantic Web. Berners-Lee, Hendler, and Lassila (2001) extended the idea of ontologies to construct a global scheme for relating resources on the World Wide Web. The Semantic Web is being built from various tools archived in a central public repository (*http://www.w3.org*; *http://www.semanticweb.org/* 2007; Ontoworld.org 2007). The basic organization for this architecture is a resource description framework (RDF). RDF constitutes a language for describing resources. Its fundamental building block is a triple consisting of an entity, a property, and a value (which may be another entity). An entity consists of a resource encoded in a universal resource identifier (URI). A URI may be a uniform resource locator (URL) for identifying a Web resource, or another type of resource predefined in a schema. In any case, the RDF is a simple, yet expressive way to construct a network or directed graph with nodes and edges qualified by attributes.

The idea is simple and revolutionary but the execution involves a mass of details. To succeed, the Semantic Web project must convince Web developers to use URIs (instead of peculiar URLs), to use them consistently, and to coordinate their efforts. If they do, then the impact of the Semantic Web project on data analysis should not be underestimated. Even if this effort fails, it is reasonable to predict that another, more powerful schema will take its place. In either case, we probably will witness the end of meta-analysis as it is currently practiced, that is, by pooling summary statistics. Instead, future studies will be published with their data, and the data will be accessible to researchers who will not have to worry about file formats, data dictionaries, and other peculiar templates for organizing data. Pooling will happen on raw data, guided by ontologies to maximize compatibility, security, and homogeneity.

Pooling will not be the only benefit of this technology. Semantic Web languages will enable logical relations that can be used to facilitate data mining and inference. Pilot projects are already demonstrating feasibility. Semantic Web technology is helping preclinical researchers to integrate disparate sources of data in single files for analysis, trace logical implications through multiple experimental results, and relate descriptive metadata to their own findings. Some have already been able to identify possible causal connections between genomic signatures and clinical diseases by mining scientific databases, reference bibliographies, and data repositories (Feigenbaum, Herman, Hongsermeier, Neumann, and Stephens 2007).

The rate of innovation in this area is so fast that it is difficult to assemble a set of predictions that will characterize the full impact of this technology. Several Web sites give a flavor of what is happening in tool development. The SIMILE project at MIT (*http://simile.mit.edu/*) offers various programming and visualization tools. Michael Bergman (*http://www.mkbergman.com/*) maintains a comprehensive list of Semantic Web tools by various developers. Hewlett-Packard (*http://www.hpl.hp.com/semweb/tools.htm*) provides several Java based toolkits. And the ESW Wiki (*http://esw.w3.org/topic/SemanticWebTools*) explains, before listing available tools, that "Keeping such lists up-to-date is obviously a problem when the number of Semantic Web tools increases every day." Finally, it is probably safer to steer readers to a regularly updated site containing books on the Semantic Web (*http://esw.w3.org/topic/SwBooks*) than to cite them separately in the references of an archival print journal. Because "Semantic Webbers" obsess over keeping URI's consistent and durable, we should have some hope that these URL's will still work in a few years.

3.1.4 Distributed Processing. We hear much today about parallel, distributed, and grid computing helping us to solve huge problems in scientific data analysis. Most new microcomputers contain multicore processors, and several scientific languages and statistical packages (Python, Stata, SAS) already exploit this capability. Distributed capabilities exist on several scales—the chip, the computer, the local area network, and the Internet. This multiscale architecture makes general distributed algorithms more difficult, but not intractable. Google uses distributed analytical architecture on its own server farms. On a global scale, SETI@home (*http://setiathome.berkeley.edu*), Folding@home (*http://folding.stanford.edu*), and Genome@home (*http://genomeathome.stanford.edu*) are demonstrating the practicality of large-scale distributed analytics.

We should keep a caveat in mind, however. Classical statistical iid problems are usually implementable on a gridded system, and tools are readily available from companies like IBM and Sun, as well from the Globus Alliance (Kesselman and Foster 1998; Globus 2008). Jerome Friedman (personal communication) has pointed out that data-dependent statistical algorithms, which include most recent statistical methods, generally are not amenable to gridding. Iterators depending on data values can lead to performance degradation or even deadlock and failure of the overall gridding algorithm. We should not expect the recent developments on deadlock-avoidance to lead to a general breakthrough in general model estimation for huge data sets.

Finally, we expect agent-based modeling to continue to thrive in the world of distributed analytics (Bonabeau 2002; d'Inverno and Luck 2003). In many cases, agent-based models can be parameterized as dynamical systems (Motter, Matas, Kurths, and Ott 2006). Either way, more powerful computational environments will allow researchers to handle much larger problems.

We also expect that agent-based models will be used to analyze real data (as opposed to simulations). This amounts to distributed analytical agents that can seek out specific types of data, do their computations on those data, and coalesce their results. Similar technology exists in bots and worms today, but security issues have prevented legitimate agents from working similarly. Better security measures, economic protections, and confidentiality standards need to prevail before such technology can become more prevalent.

3.1.5 Security and Confidentiality. Several decades ago, the columnist Mike Royko advised his readers to lie to TV exit pollsters (Royko 1984). His advice arose out of a widely shared frustration that exit polls were biasing the electoral process. Today we are seeing a similar erosion in confidence regarding the security of our online personal data (CSIA 2006). This could emerge as one of the most significant issues affecting the growth of commercial and government social data analysis.

The crisis has been driven by numerous breaches of commercial and government databases, and it has been intensified by the continuing failure of governments and corporations to develop regulations and technical solutions that will assure individuals that their personal data are safe. Furthermore, frequent assurances by corporate public relations that IT departments will police themselves better in the future have only made the situation worse. It also has not helped that insurance companies are now selling policies to cover identity theft.

The crisis in confidence is unlikely to subside until technical solutions to data confidentially are developed. Such solutions bear the same relation to analytic databases as security methods (e.g., RSA, PGP, SSL) have to online commerce. We must remember, however, that physical security is not the only problem. The problem is made more difficult because ensuring the security of personal data cannot rest on a single password or certificate. Clever data mining algorithms can merge or fuse disparate data sets using subclassifications to identify individuals (ASA 2007). This can happen with publicly released information (e.g., census data, economic statistics).

There are signs of progress. The 1996 Congressional Health Insurance Portability and Accountability Act (HIPAA) has emphasized the need for developing protection for medical records, and the National Institute of Statistical Sciences (NISS) and the Statistical and Applied Mathematical Sciences Institute (SAMSI) have sponsored data security workshops led by Alan Karr and supported by the National Science Foundation (Sanil, Karr, Lin, and Reiter 2004; Karr 2006; NISS 2007). Stephen Fienberg (Fienberg 1998, 2006, 2007) has played a leading role in developing approaches to securing categorical data and limiting unauthorized data fusion. Fienberg, Karr, and Cynthia Dwork also founded *The Journal of Privacy and Confidentiality*.

It is reasonable to expect that privacy and confidentiality will be an increasingly important requirement for analytic computing in the coming decade. Commercial interests will drive this trend because companies realize that Mike Royko could inspire a younger generation of journalists to campaign for legislation restricting online social data.

3.1.6 Data Quality. Karr, Sanil, and Banks (2006) defined data quality as follows:

> Data quality is the capability of data to be used effectively, economically and rapidly to inform and evaluate decisions. Necessarily, DQ is multidimensional, going beyond record-level accuracy to include such factors as accessibility, relevance, timeliness, metadata, documentation, user capabilities and expectations, cost and context-specific domain knowledge.

Data quality may emerge as one of the most critical factors affecting analysis in the coming decade. As Karr pointed out, we often hear that we will drown in a flood of data, but a flood of bad data may be more of a threat. The electronic collection and assembly of data threatens to swamp the close examination of data before analysis. But this threat can be used to advantage if we develop automated assistants that can work with data experts to identify problem areas. Wand and Wang (1996) proposed such an approach based on ontologies. Scannapieco, Virgillito, Marchetti, Mecella, and Baldoni (2004) described a data quality broker and a quality notification service that can bring human analysts into the quality loop.

As Karr et al. (2006) demonstrated in their case studies, visualization is a valuable tool in this effort. The information visualization community has devoted most of its efforts so far to high-dimensional data visualization. We expect that it will be turning its attention to data quality assurance in the coming decade.

Securing data quality involves, among other things, the definition and search for anomalies. We consider this in the next section.

3.1.7 Anomalies. One field in which statisticians should thrive in the coming decades involves the hunt for anomalies. Classical statistics framed the idea of an outlier by defining an extreme point relative to the location of a specific distribution (Barnett and Lewis 1994). The most frequent application of this idea has been in the diagnosis of residuals (Anscombe and Tukey 1963; Cook and Weisberg 1982; Atkinson 1985).

Outliers are anomalies, but not all anomalies are outliers. Consider the anomaly of an IRS return that contains numbers backfitted such that they match a regulation too perfectly. Consider the psychologist Cyril Burt's data that were a perfect fit to a normal curve (Dorfman 1978). Consider an "inlier" near the center of an extremely bimodal distribution. Consider a systematic drift that does not exceed conventional control limits. Consider slope changes (but not spikes) in network activity that can signal intrusions (Lambert and Liu 2006; Grossman et al. 2007).

By definition, anomalies require a law or model from which to deviate. Unlike most scientific modeling, however, anomaly detection, not the model itself is the focus of interest. The growing interest in anomalies is being driven by fraud detection, terrorism screening, defect identification, micromarketing, therapeutic biomarkers—wherever exceptions to rules are likely to yield critical information. Among the many future applications of conditional probability models, such as those used in Bayesian statistics, this is likely to be one of the more prevalent. The Bayesian model is ideal for quantifying the degree of surprise associated with an anomaly. Because policy decisions frequently accompany anomaly detection, the Bayesian formulation is most appropriate.

3.2 Humans

We should expect analysts to interact with computers very differently in the future. Traditionally, analysts have used programming languages or GUIs to specify models and request results. A few systems (e.g., National Instruments Labview, SPSS Clementine) have used iconic data flows to assemble analytics. There was a time when visual programming was thought to be the future analytic environment. Today, the prospects are more limited. Johnston, Hanna, and Millar (2004) provided a survey. In the future, programming languages are likely to remain programming languages, and visualizations are likely to remain visualizations. Templates can do many of the things that data-flow systems now do, without the complexities of understanding directed graphs. We expect that the exploratory environment of manipulable holographic visualizations depicted in *The Minority Report* is the most likely visual analytic scenario to emerge.

3.2.1 Statistical Computing Interfaces. As mentioned earlier, we can expect interfaces to move away from WIMP models. Statistical packages have become overwhelmingly complex for ordinary users. Learning command languages such as SAS, R, or Stata can be daunting for casual users (including expert statisticians who use software only occasionally). Moreover, menu-based systems for packages like SAS and SPSS have become huge; finding choices in these menus (particularly deep in hierarchical menus) often wastes time.

Wilkinson (2008) presented an alternative interface for statistical computing that more closely resembles ones we are likely to see in the future. It has a number of distinctive features. First, there are no commands; simple words serve as directives. Second, there is no menu bar with drop-down menus; each word has its own popup. Third, there is no user manual; choices are offered as they are needed, and only when they make sense. Fourth, the interface does not require a mouse; it can work with a mouse but is ideally suited to touch interfaces like the one on the iPhone. Fifth, the interface does not require a keyboard; all choices are by touch. Sixth, the interface is not modal, and there are no dialogs; choices are made initially in a supervised order, but may be revisited and changed in any sequence.

Figure 1 shows an example of an analysis expressed in this interface, called a "sentence tree." The program opens with a single Start word in the upper left corner of the controller panel. Pressing this word produces a second, Get Data, phrase. Pressing the Get Data phrase opens a popup showing available data files. Choosing a file causes an Analyze word to appear as child

Figure 1. Sentence-tree interface for statistical analysis. New words appear after most recent words are selected, guiding the user through the analysis in the same fashion as a Wizard. Unlike a Wizard, prior words may be accessed in any order, data sets may be changed, and scripts may be saved for subsequent analysis (Wilkinson 2008). The popup menu lists variables in a file; black histograms show the distributions of continuous variables, and red bars show the distributions of categorical variables.

of the Get Data phrase. Pressing Analyze offers a choice of several analyses. The figure illustrates two of these "analysis sentences," Predict and Classify.

This interface is designed to output results in a browser or editor window. The end result is a publishable document containing introduction, analysis, and discussion sections, along with tables, graphs, and a bibliography. An obvious question is whether this interface can support the type of sophisticated transformations and analyses that scientists expect to be able to perform. Clearly, a program designed like this requires a high degree of intelligence. For example, FASTAT examines the dependent variable in a Predict sentence; if there is evidence in the data that this variable is categorical (e.g., a string or an integer with a few values), then the algorithm switches from ordinary least squares to logistic regression. The program also contains a strategy engine that examines distributions, performs transformations, and investigates competitive models. This AI approach benefits from insights developed by Wayne Oldford and Catherine Hurley, summarized by Oldford (1999).

We can expect to see more designs like this in the future, because analysts will not have the time to immerse themselves in statistical packages, particularly as the packages increase in complexity to serve microconstituencies. Moreover, there will be an increasing need to assist scientists with software that provides at least a modicum of protection from artifactual conclusions. Fluency in a statistical package is no guarantee of wisdom in using it. Ideally, of course, every study should be designed and analyzed with the assistance of a professional statistician. This has been a rare luxury, and given current educational trends, it is likely to become even rarer. Simple interfaces to guided analyses will proliferate in the future.

3.2.2 Statistical Visualization. Until recently, information visualization has applied the concrete representations of realistic computer graphics to the abstract world of information. This approach has been based on the assumption that assigning variables to the ordinary dimensions of our world (space, time) can reveal structures in data. The assumption that realism aids information visualization appears plausible on its face, but a deeper look reveals that the world of graphs is more complex

and that abstractions sometimes require abstraction (Becker and Cleveland 1991; Hanrahan 2005; Wilkinson 2005).

At the other extreme, engineers have constructed hierarchical visualizations in which ordinary metrics do not apply: one or more of the axioms for symmetry, the triangle inequality, and identity are violated. These complex visualizations, such as treemaps (Johnson and Shneiderman 1991), constitute intricate and beautiful pictures, but it is not clear that they contribute to an understanding of the data that they represent.

There is little reason to assume that realism per se is required for effective visualization. More important is matching the presentation of visual information to the appropriate perceptual mechanisms that are used in processing real visual scenes. Assuming that experimental psychologists become more involved with visualization engineers, we should see more effective visualizations in the future. Often, however, authoritative psychologists are ignored in favor of gimmicks (Tversky, Morrison, and Bétrancourt 2002). If something can be done with a computer, then it will be done. And if a visualization is eye-catching, then it will become popular regardless of its usefulness. Decades of books by Tufte (Tufte 2002, 2003a,b) are reverentially cited in the introduction to visualization articles and then promptly ignored. We expect this trend to continue.

Another trend that we expect to see grow in the future is smart visualization. Large data sets present formidable problems to both the exploratory data analyst and the formal modeler. How do we find outliers, miscodes, missing data patterns, and other anomalies in huge data sets? We need to examine our data, in other words, before applying any model or before examining data with conventional EDA methods (including visualizations). In the spirit of *Minority Report*, we need agents to display data to highlight interesting features.

Figure 2 shows one approach presented by Wills (2007). This program analyzes files and text automatically and produces visualizations based on knowledge extracted from the file structure and the grammar of graphics. Figure 2(a) shows the blank window and five-word user manual for the AutoVis utility. A user simply drags objects into this window. Figure 2(b) shows the result of dragging text from *Moby Dick* into the window. The program computes n-grams on the text and performs a graph layout of the word similarities based on the n-grams. Each word is a node in the graph, and the edges represent co-occurence in the n-grams. Figure 2(c) shows the result of dragging a data set from Allison and Cicchetti (1976) into the window. The program examines distributions of the (presumably) continuous variables, decides to log-transform several to achieve symmetry, displays them in kernel densities, and displays the categorical variables in bar charts. In addition, the program constructs an association diagram among the variables based on the strength of a correlation measure. The program also uses several algorithms to choose interesting bivariate plots to present to the user. These plots are hexagon-binned to reveal

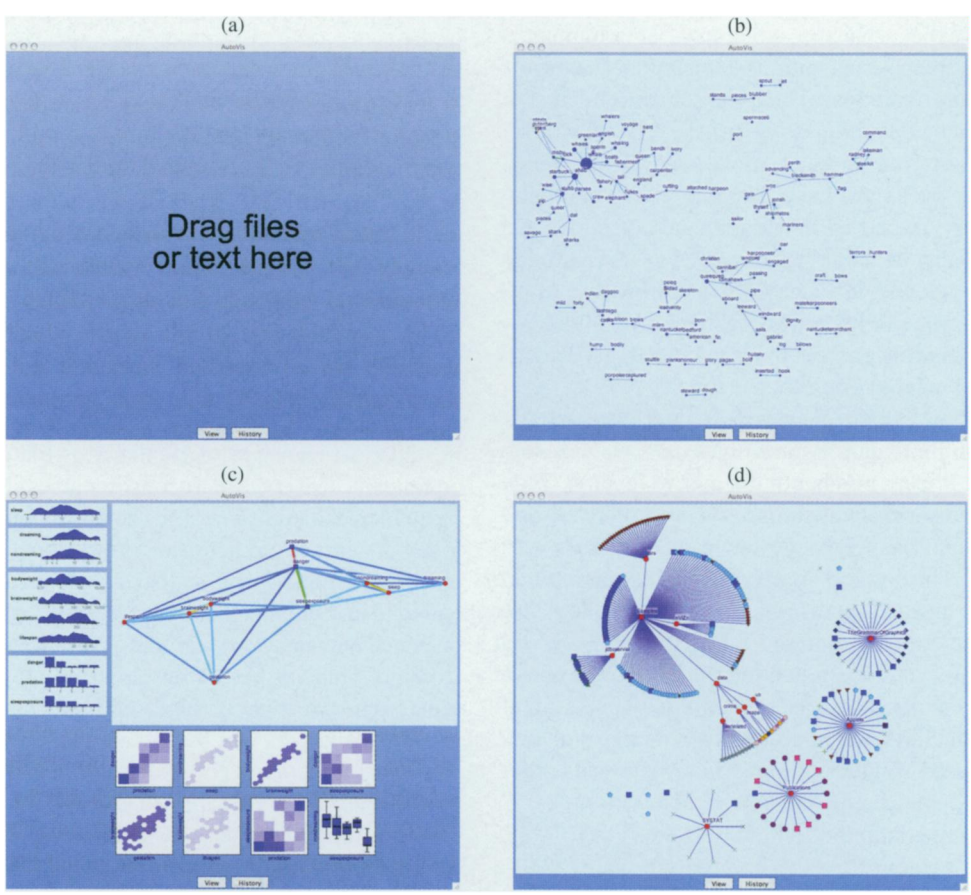

Figure 2. Automatic visualization with AutoVis. (a) AutoVis opening screen. (b) Text from *Moby Dick*. (c) Sleep data set. (d) Web site. This Java application, described by Wills (2007), processes data stored on a local machine or at URLs on the Web. The program recognizes data types by examining file formats and internal structures. It chooses transformations and statistical analyses based on various heuristics.

Figure 3. Forecasted inflow of ecological resources to support expected lifestyles in the U.K. This visualization was produced using technology developed by Phan et al. (2005). The rendering was computed automatically from the edge list of a geometric planar-directed graph (New Economics Foundation 2007. Used with permission).

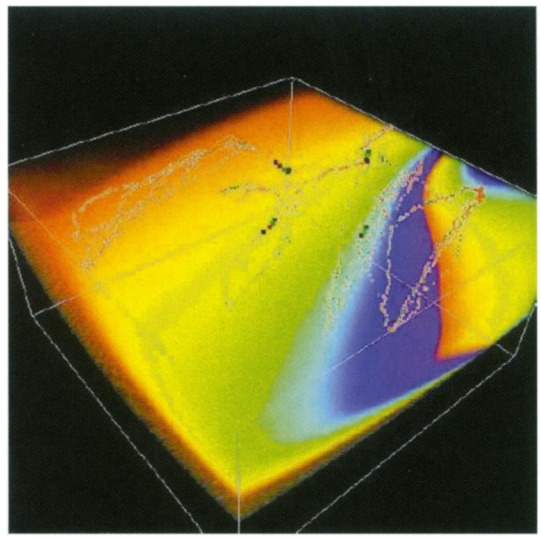

Figure 4. Streaming visualization of climate model from Park et al. (2005). This figure is a single frame from the series. Unlike animations, which are constructed by rendering archival data sets, streaming visualizations process data in real time. To maintain smooth flow, the statistical models underlying these visualizations must be updated in under 50 milliseconds.

density, and boxplots are chosen for mixed-scale displays. Finally, Figure 2(d) shows the result of dragging a Web site into the window. The files and other resources in the site are analyzed and arranged by a large-scale graph-layout routine.

Smart visualization also means the ability of the computer to "fill in the blanks." For many graphs, there is a one-to-one correspondence between the numbers and strings in the columns of a table and representative points in the graph. For many other graphs (e.g., layouts of $\{V, E\}$ graphs), there are one or more levels of indirection between the data and the graph. These levels may be a transformation chain or a series of nontrivial algorithmic processing. Figure 3 shows a three-dimensional layout of a flowmap produced by software described by Phan, Yeh, Hanrahan, and Winograd (2005). The software receives only a set of lat-lon coordinates for the branches of the tree; it decides where to place the branches, how to curve them, and how to transform the result into spherical coordinates.

We also expect to see real-time graphics that are attached to streaming data sources. Norton, Rubin, and Wilkinson (2001) discussed this type of system. The complexities of the architecture needed to support such systems exceed those of popular animated graphics, because data must be captured and displayed in real time. Snapshots and animations over static data sources will not work. We already see primitive systems of this sort in heart monitors and real-time graphics on trading desks on the major exchanges. Almost any kind of visualization can be adapted to this type of system to reveal changes in real time. Many of these applications will occur in security monitoring, medical environments, and other areas where rapid feedback is critical. Figure 4 shows a frame from a real-time system that uses inverse distance-weighted interpolation on scattered data to display local climate information (Park, Linsen, Kreylos, Owens, and Hamann 2005). Programming real-time statistical algorithms for streaming data sources is nontrivial.

Finally, we expect to see more visualizations of geospatial and temporal data on the Web, especially involving *mashups* with Google Maps and Google Earth. Geographers such as Jason Dykes and Alan MacEachren (Dykes, MacEachren, and Kraak 2005; Wood, Dykes, Slingsby, and Clarke 2007) have developed various visualizations that illustrate this trend. Figure 5 shows a Google Earth mashup involving geo-locator data on a migrating elephant seal. The power of this visualization is enhanced by the application's pan-and-zoom and layering tools. Because it is Web-based, millions of viewers can interact with the visualization to explore its aspects further. This graphic was inspired by a course developed by Deborah Nolan and Duncan Temple Lang at Berkeley and funded by the National Science Foundation.

3.2.3 Collaboration Among Statisticians. We have seen signs of convergence in collaborative computing. The R Project, open-source, and social networks among professionals are proliferating. Wikipedia is being cited increasingly in academic journals, and Wikis are now used frequently by research groups for coordinating projects.

We can expect to see an increase in expertise within knowledge collectives. The original Wikipedia credo restricted writers to a neutral point of view. Expertise was explicitly rejected. Assertions required citation, if only to a secondary source. Increasingly, we are seeing monitored sites and Wikis with expertise required for participation. These types of communities appear to be organizing themselves into an alternative small-world graph. We should expect implementation of security and auditing standards on these sites so that researchers can collaborate without fear of spam.

3.3 Machines

If analysts interact with computers very differently in the future, it will be because the computer will evolve to become an analytic assistant. About 25 years ago, John Tukey coined the term "cognostics" to describe automated methods to help analysts understand data sets. In a speculative keynote at the Interface meetings, Tukey described such things as distributed

Figure 5. Google Earth mashup of elephant seal migration. These data track the movement of an adult female elephant seal as she migrates between rookeries in southern California and distant northern foraging areas. The data are courtesy of Brent Stewart, Hubbs-Sea World Research Institute. Brillinger and Stewart (1998) showed that the seal migrates in a great-circle path. Graphic courtesy of Deborah Nolan.

computing engines that exploited free CPU cycles to attack convex hull computations in hundreds of dimensions (Tukey 1982). Tukey also talked about parallel algorithms that could sift through thousands of scatterplots to find interesting patterns—an approach he later called "scagnostics" in work done with his cousin Paul (Tukey and Tukey 1985). Wilkinson, Anand, and Grossman (2006) recently implemented this idea, and code is now available in CRAN (Wickham and Hofmann 2007).

3.3.1 Intelligence. The analytic assistant personifies Tukey's vision of the computer as a mental amplifier. The landmark book *Artificial Intelligence & Statistics* (Gale 1986b) contained chapters on the REX regression expert system developed at Bell Labs (Pregibon and Gale 1984) and other approaches to implementing statistical assistants. There is insufficient space in this article to cover the history of AI in statistics, but it should be mentioned that much of the early AI research that influenced the more recent field of data mining came out of the Bell Labs group. For various reasons, AI statistics research of this sort waned in the new century after a decade of International Workshops on Artificial Intelligence and Statistics. The term "waned" might be considered extreme by some, because what really occurred was a transition to a new form of AI—data mining and, subsequently, machine learning—that a few might take to be new wine in old bottles but was really a change of focus from expert systems to classification and prediction algorithms.

As with AI in general, which moved away from modeling human intelligence per se, we have seen a move away from efforts to encapsulate the judgments of expert statisticians. Nevertheless, we should expect a resurgence of automated analysts in this century. These analysts likely will incorporate strategies derived from psychological, statistical, and algorithmic research. Although most of the recent efforts in machine learning have focused on algorithms, we can expect that the rapidly evolving field of human decision making (sparked initially by the research of Herbert Simon, Amos Tversky, and Daniel Kahneman) will become increasingly important to these developments.

In fact, we have seen some recent examples. Analytic engines have gone underground. Commercial companies such as SAS, SPSS, Oracle, and Microsoft sell analytic components that are embedded in enterprise software systems. Web Services analytics are used to make credit decisions in real time, hedge investment portfolios, and place Web search ads. Two resources will fuel this development in the future. First, more computer scientists have begun to learn statistics. We have seen a trend from a small group of statisticians and computer scientists working in the latter part of the last century to a mainstream movement in computer science. Evidence of this trend can be found in several influential articles and talks (Glymour, Madigan, Pregibon, and Smyth 1997; Friedman 1997) and a recent statistics book (Hastie, Tibshirani, and Friedman 2001) that has been a huge seller among computer scientists (John Kimmel, personal communication). In addition, universities, such as University of California Irvine, Carnegie Mellon, and Stanford, have encouraged statisticians and computer scientists to work together on machine learning. Statisticians have helped move the focus of data mining from in-sample prediction to out-of-sample prediction. Second, protocol analysis, developed by Simon, Ericsson, and others at Carnegie Mellon in the 1970s, is being rediscovered by developers as a way to capture cognitive factors in decision making. This "knowledge engineering" methodology has been widely applied in expert systems, such as medical diagnosis. It was used in the Student project (Gale 1986a), a follow-up to REX, and we can expect it to be used in future diagnostic systems. Ironically, the difficult areas of AI remain in early-process perception needed for robotics (vision, hearing, touch), whereas the more tractable areas seem to be in expert behavior. The early arguments against AI in statistics (e.g., statisticians' nuanced judgments are impossible to capture, the field is too complex) have been dispelled by the prevalence of machine learning algorithms and policy capturing. This trend will continue, and indeed will accelerate.

3.3.2 Collaboration. We have discussed collaboration among analysts, but we must not ignore collaboration among machines. Many of the advances that we expect in the future will come from analytic engines working together to combine data fusion, inference, and model building. These machines will share data and use such technologies as voting schemes and neural nets to combine their decisions.

The emergence of these technologies raises a troubling question. As machines become more intelligent and capable of sharing models and data, we run the risk of *model monocultures*; that is, consensus votes among machines can result in the kinds of judgment bias to which humans are susceptible (Erev, Wallsten, and Budescu 1994; Dawes and Mulford 1996). It may be that analysts and designers will have to give consideration to methods that allow minority points of view to persist beyond a single iteration of voting schemes.

3.4 Providers

This section covers vendors and providers of statistical software. It is perhaps the most speculative of the sections in this article, because it is based primarily on my experience working for and collaborating with many of the companies and projects that currently provide analytical and statistical software. As such, it is a forecast based on personal opinion and may be controversial. Nevertheless, I am in a perhaps unique position of

knowing personally the founders, principals, and many of the developers in these companies and foundations over a 30-year period.

The commercial companies chosen for this list are presented in order, corresponding to their revenues from statistical applications in recent years. The not-for-profit associations (R and Python) are presented last, because their revenues are essentially zero. Nevertheless, they have rapidly growing user communities that already exceed the size of some of the commercial companies' user bases. The statistical software providers discussed in this section were derived from the small Google Set (*http://labs.google.com/sets*) obtained from the pair {SAS, SPSS}.

We begin with a summary of each company and a forecast for each.

3.4.1 SAS. Of the total SAS revenues in recent years, approximately 20% was due to advanced analysis (statistics and data mining). The rest was due to reporting, data management, performance monitoring, general business intelligence, and other applications. SAS continues to be a leader in statistical data analysis. It has not ignored the needs of its specialized user bases (e.g., pharmacology, manufacturing, natural and social sciences). Yet SAS has been able to expand into corporate data mining and analytic markets while serving this traditional base. Its predictable growth has been guided by the technocratic philosophy of its statistician CEO, Jim Goodnight.

Nevertheless, SAS must confront the growth of companies like Oracle, which had almost $18 billion in sales in the 2006 fiscal year. Oracle is determined to capture the new markets that interest SAS. The CEO of Oracle, Larry Ellison, is Goodnight's avowed nemesis—determined to confine SAS to its specialized vertical markets. Oracle is, to put it mildly, an extremely aggressive competitor. Both Oracle and SAS share a strategy of "one-stop shopping." This signals continued battles in the future.

3.4.2 SPSS. Under the leadership of its CEO Jack Noonan, SPSS has reinvented itself. The company has begun to concentrate on what they call "predictive analytics"—components in service of business information systems. Unlike SAS, SPSS has focused primarily on embedded systems for its new offerings. This yields a highly competitive pricing strategy that allows SPSS to slip its new enterprise systems into heterogeneous corporate computing environments.

SPSS has developed several new technologies for the enterprise environment. Its visualization platform (nViZn) is based on grammar-of-graphics architecture (Wilkinson 2005). SPSS intends to provide extensive visualization in its Web-based enterprise products. A new patented pass-stream-merge (PSM) technology is being used to extend analytics to gridded and distributed computing environments. Finally, SPSS has developed a model repository to organize and deploy predictive models for scoring environments (e.g., credit scores, insurance adjustment).

SPSS is likely to continue supporting its flagship statistical product. Like other companies, however, SPSS will probably leverage the technology used for its other products (e.g., analytic components, visualization) to drive future development of the statistical package. These components, many developed in a satellite office in China, will replace older FORTRAN analytics that date to the early era of statistical computing. The major share of SPSS growth will be in new products, however.

3.4.3 MATLAB. For MathWorks, perhaps 15% of their gross revenues would account for what we would call statistical and data analysis in their MATLAB product. The MATLAB package is very popular among engineers doing statistical analysis, likely because of its flexible graphics and matrix modeling. Whereas matrix languages like APL were based on formal grammars and a high degree of mathematical extensibility, users tended to prefer more ad hoc systems, such as MATLAB and SAS IML. Applied users rarely appreciate the mathematical niceties of generalized inner products ($A +.\times B$ or $A \vee.\wedge B$) when they simply need real matrix multiplication ($A * B$). In addition, MATLAB benefited from its close association with the certified Argonne matrix libraries through MathWorks CEO Cleve Moler.

MathWorks is likely to remain a force in future analytics. It has shown a talent for adapting to new markets and for developing add-on modules for specialized applications. It will face increased competition from open-source projects such as Python, however. These projects have been actively engaging in large and sparse matrix computations.

3.4.4 Minitab. Minitab's revenues have been based largely on engineering analytics and instructional software. Approximately 10 years ago, CEO Barbara Ryan refashioned Minitab as a Six Sigma provider, and its revenues began to grow at a rapid rate. At this time, it is probably the leading provider of this type of software.

Much of the Six Sigma sector depends on marketing. Minitab has done a masterful job at this through educational seminars and close engagement with Six Sigma companies (most notably Ford). In the future, however, Minitab will be pursued resolutely by Statsoft, SAS, and JMP. Because of the hegemony of the companies in this sector and the specialized nature of the software, it is unlikely that others will be able to enter as serious competitors.

3.4.5 Statistica. Statsoft is run by Paul Lewicki, a psychologist at the University of Tulsa. Statsoft's advertising suggests that it is targeting corporate enterprise systems for business intelligence, but it will probably have rough going if it pursues this strategy exclusively. Microsoft, ORACLE, IBM, SAS, SPSS, and SAP have declared this a strategic market. Competing with this group will be difficult. Alternatively, JMP and Minitab will continue to compete directly with Statsoft for the desktop quality market. Nevertheless, Statsoft has enjoyed steady and substantial growth over almost 2 decades.

Statsoft's strength has been its ability to develop new procedures soon after its competitors introduce them. It has done this to a large degree through contract programmers in Eastern Europe and through the use of third-party software. In its early days, the company imitated aggressively, but more recently it has focused on its own identity. Its user base is concentrated (and likely will continue) among engineers rather than among statisticians.

3.4.6 S–PLUS. Insightful, more than any other statistical software company, has been affected by the growth of R. S–PLUS was originally based on a perpetual license to the Bell Labs version of S. When the R Project gained momentum, there emerged an alternative source for a major portion of the S language.

In response, Insightful has specialized in feature-rich implementations of specialized procedures for specific vertical mar-

kets (finance, marketing, life sciences). The company hopes to deepen these markets and provide close technical support to stem the loss of clients to R. Insightful is not the only company to feel the impact of R, but it likely is the most vulnerable.

3.4.7 JMP. JMP began as "John's Macintosh Project," a desktop interactive program designed by John Sall, who was an admirer of Paul Velleman's Data Desk. John was an early advocate of GUI interactive computing inside SAS. He put together a talented team of developers dedicated to marrying graphic and statistical output.

In recent years, JMP has become a major competitor in the design of experiments. Much of this has to do with Brad Jones (a student of Stuart Hunter) joining the team after working on MATLAB at MathWorks. JMP also has drawn on the formidable design resources elsewhere in the SAS Institute. Most recently, Sall has steered JMP toward the genomics market as well; JMP can handle interactive analyses on data sets with tens of millions of rows and thousands of columns.

JMP likely will expand its market in quality and experimental design. Further, the JMP graphics are beginning to achieve a distinction on their own that will help JMP expand into other fields.

3.4.8 Stata. Stata was originally the product of Bill Gould and a small group of economists from UCLA. It has grown to be a full-featured analytic company. The distinctive appeal of the package is its expressive and concise programming language, based on C. Stata's unusual strengths are in discrete variable modeling, longitudinal/panel designs, survival analysis, time series analysis, and survey statistics.

Like S–PLUS, Stata will have to deal with the growth of R in its own field—programmable statistics and data analysis. Unlike S–PLUS, however, Stata's peculiar strengths and language are different enough from R to make it a viable alternative, particularly for economists. Moreover, the Stata user community is intensely loyal, so we should expect Stata to continue to grow at a respectable rate.

3.4.9 SYSTAT. SYSTAT has been acquired twice, first by SPSS and more recently by Cranes Software International, located in Bangalore, India. SYSTAT's traditional market has been users looking for a simpler alternative to SAS and for high-quality graphics. Because it was the first full-featured statistics package available on microcomputers, SYSTAT acquired a following among ecologists, biologists, and others who needed to do analyses in the field using portable (luggable) computers.

Leland Wilkinson wrote SYSTAT in FORTRAN and the manuals in English in the early 1980s. Several years ago, Cranes assigned a team of almost 100 engineers and statisticians to rewrite SYSTAT in C++. The company hopes to develop new scientific applications from this platform. SYSTAT's future depends on its serving its base of scientific researchers; it cannot compete in the enterprise market without losing focus.

3.4.10 Google. We do not ordinarily associate Google with analytics. In 1 week a few years ago, however, Google wiped out an entire Web analytic market segment with the introduction of Google Analytics. Companies like Net Genesis and WebSideStory had to reorganize after Google offered a powerful analytic environment for free.

Google is a leader in distributed analytics, but these are reserved for internal use. The company has computed logistic regressions on data sets with billions of cases and millions of variables across thousands of processors. The Google News facility is one of the largest and most complex text analysis systems in production.

We can expect more thin-client analytics from Google as the company senses commercial applications that can expand its revenue base. Google is uniquely suited to develop distributed technology because it has heavily recruited statisticians, computer scientists, and mathematicians who specialize in these areas.

3.4.11 Microsoft. It has been frequently observed that the most widely used statistical package is probably Microsoft Excel. For many, Excel is essentially free, because the organization has a license for Office. Consequently, and despite the serious shortcomings documented by Knüsel (1998) and McCullough and Wilson (2002), Excel is widely used for teaching and analysis.

In 2000, Microsoft introduced various data mining methods (e.g., k-means clustering, decision trees) in its Analysis Services suite. The GUI for this software made it especially easy for novices to analyze data stored in the Microsoft server. As in other areas, Microsoft has pursued a strategy of dominating the vast horizontal market for analytics, leaving SAS, SPSS, and other competitors to serve the high-end market. We expect this trend to continue.

3.4.12 R Project. The R Project began with an effort by Ross Ihaka and Robert Gentleman to develop a new environment for statistical computing. They soon adopted the S syntax because of its familiarity among statisticians. The rapid growth of R came after the base engine was completed and statisticians joined the open-source project to develop sophisticated statistical routines.

R is a functional programming language whose result set is a collection of objects. Its object-based architecture, originally devised by John Chambers, makes it simple to gather the types of information (e.g., coefficients, residuals, diagnostics) that statisticians typically want to examine and manipulate when modeling.

R might have been equally successful if it had introduced a different statistical computing model, but much of its success appears to stem from the familiar S syntax. Statisticians adapted to this syntax readily and many have joined as developers. The size of the R development community probably exceeds the size of any commercial analytic company's development group. Consequently, we should expect the growth of R to continue to be almost exponential.

Only a new architecture for statistical computing is likely to affect this growth. In fact, the R and Python communities have already held discussions on symbiotic development of new statistical computing systems. A promising path appears to be melding the parallelism of the iPython project (*http://ipython.scipy.org/moin/*) with the statistical algorithms of R. This would enable users to compute advanced analytics on massive data sets and would offer more sophisticated data management algorithms than are available in R today.

3.4.13 Python. Python was developed by Guido van Rossum, a mathematician who now works at Google. Like other languages developed by mathematicians (e.g., APL), Python has a clean, spare, and expressive syntax. Since its early years, Python has been supported by an open-source foundation, which has resulted in explosive development.

Python's coverage of statistics is sparse (see the SciPy library at *http://www.python.org/*). As a remedy, Duncan Temple Lang developed an R/Python bridge linking both environments (*http://www.omegahat.org/RSPython/index.html*). The joint growth of Python and R is likely to influence statistical computing in the future, perhaps more than any other single factor.

3.4.14 The Future of Statistical Software. Figure 6 shows a biplot of the companies discussed in the last section. The data for the plot consist of measurements on three variables: Size, Memory, and GUI. The Size variable is represented by the number of megabytes of free disk space required for the installation of the full package, according to the software vendor. It is a very rough proxy for the comprehensiveness of the package. The Memory variable is binary. It records whether the data management of a package is primarily memory-based or disk-based. Memory-based packages tend to be especially efficient for small problems, because they do not need to allocate and write scratch files on the disk. Disk-based packages tend to be more scalable (at least in terms of number of rows in a data set), because they can swap large arrays to disk to free memory. This distinction is becoming increasingly moot because of virtual memory improvements in operating systems. Nevertheless, operating systems (especially Windows) have shown a tendency to grab the majority of addressable memory for themselves, so that increases in address space (e.g., 16 bit, 32 bit, 64 bit) do not yield proportionate benefits for memory-based data management. Finally, the GUI variable has three ordered levels to represent whether a package was originally GUI-based (with scripting commands added as an afterthought), command-based (with GUI features added later to support commands), or a combination of both (the middle value).

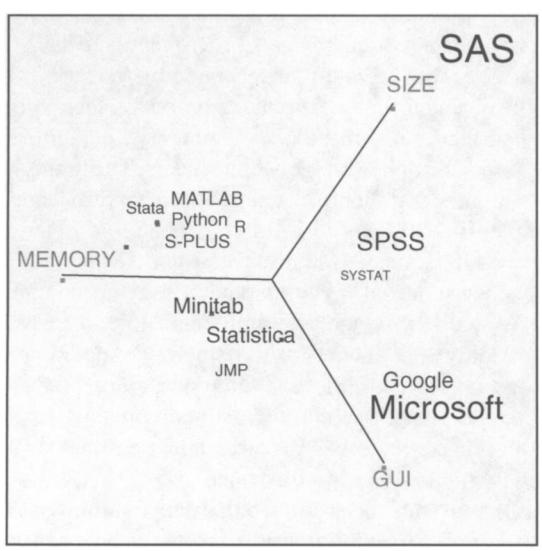

Figure 6. Biplot of statistical software packages based on predominance of memory over disk location of data, installation footprint (size in megabytes), and predominance of GUI over commands in user interface. Several clusters are evident. The memory-based programming languages (Stata, MATLAB, Python, R, and S–PLUS) constitute a tight cluster. The quality-oriented packages (Minitab, JMP, and Statistica) form a relatively tight cluster as well. SPSS and SYSTAT share somewhat similar architectures and user interfaces. SAS is sui generis in this layout. The size of the plotted names is proportional to the log of 2006 analytic revenues; the log transform was needed to prevent the smaller companies from vanishing in the plot. Revenues for the open-source groups (Python and R) were equivalenced based on rough estimates of user bases.

The size of the names plotted in the graph is proportional to the log of 2006 statistical revenues (with equivalences for the not-for-profit groups guesstimated from user bases). The log scale was required to prevent the smaller companies from disappearing in the plot. It is important to note that revenues are apportioned for statistical applications. Otherwise, two companies (Microsoft and Google) would cover the whole plot.

The biplot is based on a singular-value decomposition (SVD) of the standardized data. A similar plot results from an SVD of the ranked data. The two-dimensional projection in the plot accounts for about 85% of the variance in the companies on these variables. These three variables yield a layout that corresponds closely to the market sectors of these packages. A cluster of memory-based programming languages (Stata, MATLAB, Python, S-PLUS, and R) is readily apparent. The engineering-based quality packages (Minitab, Statistica, JMP) cluster together. SPSS and SYSTAT cluster together as well. Google and Microsoft—not primarily analytic companies—cluster together. And, finally, SAS is a singleton.

We should expect the layout in this plot to continue in the future. Of course, companies spread from their bases to attract related groups of users, so the boundaries between clusters will increasingly blur. Nevertheless, the original architecture of a product significantly influences its performance, footprint, and other behavior. Packages are occasionally rearchitected (the most conspicuous example being the SAS rewrite from PL/I to C during the 1980s). But they invariably follow the wishes of their core groups of users.

As should be evident from the discussions in the previous section, several of these companies will develop new products to enable their migration into new markets. SAS and SPSS, especially, will maneuver to avoid and combat simultaneously the encroachments of companies like Oracle, Google, Microsoft, IBM, and SAP. The greatest economic opportunities for analytic software will be in large-scale corporate and government data analysis. The traditional statistical software companies have already taken note of these opportunities and will pursue them avidly. Although they will devote some resources to their traditional statistician user bases, this focus will be secondary to their future goals. This strategy is not simply based on opportunism; it is a matter of survival in a consolidating corporate world.

What could change this pattern? Acquisitions. Most of the larger companies selling business intelligence analytic software (Hyperion, Business Objects, Cognos) have recently been acquired by database and service companies. As the statistical software companies continue to grow, they will be targets of larger corporations intending to capture majority shares of the expanding analytic market.

In 1995 the average single-copy price of the statistical software featured in this article was about a third of the price in 2006. That increase is more than twice the level of inflation during that period. Part of this increase could be attributed to increasing consolidation of the market (duopoly), but more of it is due to the shifting focus of the market. SAS and SPSS have reorganized their sales and marketing divisions to focus on large corporate and government clients. During this time,

these clients have demanded Web Service and thin-client applications to control installation, technical support, security, and licensing. Sales have thus consolidated into thousands or millions of dollars in annual license fees for onsite network installations. For these larger companies, single-copy sales are becoming economically unattractive.

This trend may give the open-source projects increased opportunities to acquire the clients that the traditional packages served in the past. The open-source movement is clearly driven on the user side by an aversion to rising software prices and punitive licensing practices. The security demands of larger corporations will fuel this drive. For the *Technometrics* readers who use desktop statistical software, the future may lie with the small cluster of companies in the northwest sector of Figure 6 and with the few others in the plot committed to single copy sales.

Finally, it should be evident that future analysts who insist on limiting themselves to a single system will necessarily sacrifice their ability to handle difficult problems. For at least a decade, end-users will have to familiarize themselves with a range of software tools to handle massive data sets and nontabular data structures. Although the popular press has promoted the Internet as a solution to all our computing problems, scientific computing (in the broadest sense) has its own special challenges. The complexities of data structures and analytic algorithms require peculiar solutions. It does not seem likely that a single analytic system for handling all these problems will appear in the next decade. We are still computing with analytic architectures that originated in the 1970s. A new comprehensive system will doubtless emerge, but not until the economics (commercial and social) and the technology (computational and social) converge to provide a foundation. Interestingly, many of the observations in this paragraph were already made in the relatively early days of statistical computing (Thisted 1986).

4. CONCLUSION

We have seen how technology and economics have forced changes on the statistical software market. This dynamic trend will continue for at least a decade. In this coming decade, we can expect three parallel developments. First, large corporate and government clients throughout the world will increasingly shape the design of analytic software. Second, open-source projects and small companies will come to serve the desktop and interactive user as the larger companies abandon this sector. Third, technical breakthroughs will introduce new forms of user interaction with software systems, particularly in the area of distributed and wearable computing.

If there is a single technical theme characterizing the future of statistical computing, it would be *smart analytics*. Smart analytics will act as assistants to statisticians and data explorers, automatically generate and fit models, automatically generate visualizations, and search networks for data and summarize results for further analysis. Statisticians justifiably are inclined to distrust such systems, just as medical doctors distrusted early diagnostic systems. But as these systems improve, their effectiveness will overwhelm opposition.

We have already seen the influence of predictive algorithms in insurance, credit, and other industries. Insurance reimbursements are assigned by automated agents interacting with customer support representatives. Loans are authorized by predictive modeling systems. Automated trading agents have earned hedge funds huge incomes. Automated process monitoring incorporating embedded analytics has improved product quality. One can argue about the net worth of such systems, but there is little basis for expecting the prevalence of smart analytics incorporating statistical models to decline.

Clearly, statisticians need to do a better job of evangelizing statistical computing, especially among computer scientists. Statisticians have traditionally offered a powerful argument to justify their existence. Statistical reasoning, it is said, protects us from being unduly fooled by chance. This is a valid and timeless argument. There is a second justification, however, that statisticians have largely abandoned. In short, statistical methods help us to predict. The computer modeling, data mining, and machine learning communities now make that argument for their own algorithms, often claiming that *their* deterministic prediction algorithms outperform statistical models in new "samples." In some cases, there is evidence that this is true. But statisticians need to devise new models, write software, and enter contests to make clear that for data approximated by appropriate distributions, there is no substitute for prior knowledge. The future of statistical computing will be bright as long as statisticians reach out to their peers in computer science and show what they can do.

5. GLOSSARY

AJAX. Asynchronous Javascript and XML. This technology allows animation and user interaction in a browser environment that approaches the richness and responsiveness of Java. Unlike Java, AJAX cannot be easily undermined by Microsoft. In fact, Microsoft invented the backbone of the AJAX technology and fully supports it in its browser.

Agent-based model. An algorithm for simulating the interactions of autonomous agents in a network to analyze states of the system over time. ABMs sometimes can be represented mathematically by dynamical systems.

Bus. A subsystem that transfers data between computer components (processor, memory, peripherals). In early computers, a bus was equivalent to wires on a motherboard or connector cables between boards. In modern microprocessors, a bus is part of the chip design itself, connecting subcomponents on the chip. The width of a bus (number of communication lines) is a major determinant of computer performance.

Column-oriented. Processing tabular data columnwise (as opposed to rowwise). This improves the efficiency of some numerical algorithms, especially for in-memory calculations, but it makes scalability more difficult to achieve.

Embedded. A component is embedded in another system if it allows the other system to control its behavior and if it is located in the same computing environment as the host system.

Enterprise. An adjective characterizing software that is scalable, thin-client, and embedded. It is also a term used by marketing people to describe their company's software, regardless of its characteristics.

Flash. A technology developed by Macromedia, now owned by Adobe, for producing animations in browsers.

Grid computing. The power grid is a metaphor for distributed (or "cloud") computing. A process (with or without associated

data) is cut into snippets so that multiple processors can simultaneously contribute to the ultimate solution.

GUI. Graphical user interface. In the 1990s, this term referred to dialogs, wizards, and other aids for controlling computer programs visually. More recently, it includes haptic (touch) and gestural interfaces for directly manipulating visual objects, as on the Apple iPhone and Microsoft Surface products.

IEEE. The Institute of Electrical and Electronics Engineers, a professional organization that, among other activities, sets software and hardware standards for industry.

IT. Information Technology, the department that controls computing in an organization. The department that says "no" to software salespeople introducing new products.

Java. A language developed at Sun for client-side computing on networks. Opposition by Microsoft and corporate IT departments drove Java off the Web. IBM rescued Java by adopting it as its standard language for server-side computing.

Mashup. A Web application that combines data from more than one source into a single integrated application. One instance is the display of prices among gasoline stations superimposed on a Google map in a Web Service that provides daily updating of those prices.

.NET. Microsoft's answer to Java. It uses a common language runtime (CLR) virtual machine as an alternative to the Java Virtual Machine. Unlike Java, .NET. applications can blend various programming languages without extra effort.

PARC. Palo Alto Research Center, a former subsidiary of Xerox that invented but failed to patent the GUI user interface most widely used on computers today.

PGP. Pretty Good Privacy, a computer program that provides cryptographic privacy and authentication based on public-key cryptography.

Rich client. A characterization of software that requires a browser plug-in (e.g., Java, SVG) on a user machine. Rich client is a marketing term devised to replace the older, pejorative term Fat client. As browsers become fatter (by including plug-ins such as Flash), rich clients paradoxically become thin clients.

Row-oriented. Processing tabular data rowwise (as opposed to columnwise). This architecture facilitates processing of large files with many rows.

RSA. The Rivest, Shamir, Adleman public-key encryption algorithm, which has been used widely for electronic commerce.

SaaS. Software as a Service, a delivery model in which a vendor hosts a software application for customers on the Internet. Customers pay for each use of the software instead of owning it.

Scalable. Row-oriented, single-pass, and minimal-memory-footprint software. The term has been hijacked by database companies to characterize analytic software embedded in their database, as if computation external to a database were necessarily less efficient.

Single pass. Software that passes through rows of data only once to compute a model.

Socket. An interface between a client program and a protocol for communicating among a group of clients. A socket implements the protocol (usually TCP/IP), the IP address of the client, and other information needed to communicate with other programs running on the net.

SSL. Secure Sockets Layer, a cryptographic protocol that provides secure communications on the Internet for browsing, email, and other activities.

SVG. Scalable Vector Graphics, a World Wide Web Consortium XML specification for implementing two- and three-dimensional vector graphics and animation in a Web environment.

Thin client. Software that requires only a browser on a user machine.

TCP/IP. The Transmission Control Protocol/Internet Protocol, a suite of standards for handling streams of bytes in Internet transmissions. The protocol specifies how these streams are to be segmented, identified, and reassembled to ensure reliable transmission of email, files, and other information across the Internet.

Virtual machine. A software duplicate of a real machine. A virtual machine executes instructions on a computer and operating system by simulating a different processor's instruction set.

Visual programming interface. A GUI-based on a planar directed graph, similar to a flow chart. Unlike flow charts, however, visual programming nodes represent higher-level objects and edges represent data flows.

Web Service. A software system designed to support interoperable machine-to-machine interaction over a network. This technology enables Software as a Service.

W3C. The World Wide Web Consortium, which develops interoperable specifications, guidelines, software, and tools for the World Wide Web. It is a not-for-profit organization.

XML. Extensible Markup Language, a W3C standard defining tags for the nodes of a tree structure containing objects such as data and specifications. The tree structure is defined by nesting tags. XML has been widely adopted as a standard for storing and transporting data on the Web because tree structures are a flexible method for storing data.

ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation grant DMS–FODAVA-0808860. The author thanks Alan Karr, Vijay Nair, Charles Sieloff, David Steinberg, and three anonymous reviewers for numerous and detailed suggestions.

[Received November 2007. Revised February 2008.]

REFERENCES

Allison, T., and Cicchetti, D. (1976), "Sleep in Mammals: Ecological and Constitutional Correlates," *Science*, 194, 732–734.

American Statistical Association (2007), "Privacy and Confidentiality," available at *http://www.amstat.org/comm/cmtepc/index.cfm*.

Anscombe, F., and Tukey, J. (1963), "The Examination and Analysis of Residuals," *Technometrics*, 5, 141–160.

Atkinson, A. (1985), *Plots, Transformations and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*, New York: Oxford University Press.

Baldi, P., Frasconi, P., and Smyth, P. (2003), *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*, New York: Wiley.

Barnett, V., and Lewis, T. (1994), *Outliers in Statistical Data*, New York: Wiley.

Becker, R. A., and Cleveland, W. S. (1991), "Take a Broader View of Scientific Visualization," *Pixel*, 2, 42–44.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001), "The Semantic Web," *Scientific American*, 284, 34–43.

Bolt, R. A. (1980), "Put-That-There: Voice and Gesture at the Graphics Interface," in *SIGGRAPH'80: Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, New York: ACM, pp. 262–270.

Bonabeau, E. (2002), "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems," *Proceedings of the National Academy of Sciences*, 99, 7280–7287.

Breiman, L. (2001), "Statistical Modeling: The Two Cultures," *Statistical Science*, 16, 199–231.
Brillinger, D., and Stewart, B. (1998), "Elephant Seal Movements: Modelling Migration," *Canadian Journal of Statistics*, 26, 431–443.
Ceruzzi, P. E. (2003), *A History of Modern Computing* (2nd ed.), Cambridge, MA: MIT Press.
Cho, A. (2007), "Making Machines That Make Others of Their Kind," *Science*, 318, 1084–1085.
Cook, R. D., and Weisberg, S. (1982), *Residuals and Influence in Regression*, London: Chapman & Hall.
Corcho, O., and Grez, A. (2000), "A Roadmap to Ontology Specification Languages," in *Knowledge Acquisition, Modeling and Management*, pp. 80–96.
CSIA (2006), "Digital Confidence Survey," *Cyber Security Industry Alliance Newsletter*, 2.
Dawes, R., and Mulford, M. (1996), "The False Consensus Effect and Overconfidence: Flaws in Judgment or Flaws in How We Study Judgment?" *Organizational Behavior and Human Decision Processes*, 65, 200–211.
Decker, S., Erdmann, M., Fensel, D., and Studer, R. (1999), "Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information," in *DS-8: Semantic Issues in Multimedia Systems*, pp. 351–369.
d'Inverno, M., and Luck, M. (2003), *Understanding Agent Systems* (2nd ed.), New York: Springer-Verlag.
Donoho, D. (2000), "High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality," lecture delivered at "Math Challenges of the 21st Century," American Mathematical Society, Los Angeles, August 6–11.
Dorfman, D. (1978), "The Cyril Burt Question: New Findings," *Science*, 201, 1177–1186.
DuMouchel, W. (2002), "Data Squashing: Constructing Summary Data Sets," in *Handbook of Massive Data Sets*, eds. J. Abello, P. M. Pardalos, and M. G. C. Reseude, Norwell, MA: Kluwer Academic, pp. 579–591.
Dykes, J. A., MacEachren, A. M., and Kraak, M.-J. (2005), *Exploring Geovisualization*, Amsterdam: Elsevier Science.
Efron, B. (2007), "The Future of Statistics," *Amstat News*, 47–50.
Erev, I., Wallsten, T., and Budescu, D. (1994), "Simultaneous Over- and Underconfidence: The Role of Error in Judgment Processes," *Psychological Review*, 101, 519–527.
Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999), "On Power-Law Relationships of the Internet Topology," in *SIGCOMM*, pp. 251–262.
Farley, T. (2007), "Telephone History," available at *http://www.privateline.com/mt_telephonehistory/*.
Feigenbaum, L., Herman, I., Hongsermeier, T., Neumann, E., and Stephens, S. (2007), "The Semantic Web in Action," *Scientific American*, 297, 90–97.
Fienberg, S. (1998), "Confidentiality, Uniqueness and Disclosure Limitation for Categorical Data," *Journal of Official Statistics*, 14, 385–397.
——— (2006), "Statistical Perspectives on Confidentiality and Data Access in Public Health," *Statistics in Medicine*, 20, 1347–1356.
——— (2007), "Privacy and Confidentiality in an e-Commerce World: Data Mining, Data Warehousing, Matching and Disclosure Limitation," *Statistical Science*, 21, 143–154.
Fisherkeller, M. A., Friedman, J. H., and Tukey, J. W. (1988), "Prim9: An Interactive Multidimensional Data Display and Analysis System," in *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, Pacific Grove, CA: Wadsworth & Brooks/Cole.
Friedman, J. (1997), "Data Mining and Statistics: What's the Connection?" in *Computing Science and Statistics: Proceedings of the 29th Symposium on the Interface*.
Gale, W. (1986a), "Student–Phase 1," in *Artificial Intelligence & Statistics*, ed. W. Gale, Reading, MA: Addison-Wesley.
——— (1986b), *Artificial Intelligence & Statistics*, Reading, MA: Addison-Wesley.
Globus (2008), "The Globus Alliance," available at *http://www.globus.org/*.
Glymour, C., Madigan, D., Pregibon, D., and Smyth, P. (1997), "Statistical Themes and Lessons for Data Mining," *Data Mining and Knowledge Discovery*, 1, 11–28.
Greer, R. (2007), "Dimensionality Reduction: A Comparative Review," available at *http://www.research.att.com/~daytona*.
Grossman, R., Sabala, M., Connelly, S., Gu, Y., Handley, M., Sulo, R., Turkington, D., Anand, A., Wilkinson, L., Foster, I., Leggett, T., Papka, M., Wilde, M., Mambretti, J., Lucas, B., and Tran, J. (2007), "Angle: Detecting Anomalies and Emergent Behavior From Distributed Data in Near Real Time," in *SC07 International Conference for High Performance Computing, Networking, Storage and Analysis*, Reno, NV.
Gruber, T. R. (1993), "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, 5, 199–220.
Hamilton, A. (1949), "Brains That Click," *Popular Mechanics*, 91, 162–167.
Hanrahan, P. (2005), "Realistic or Abstract Imagery: The Future of Computer Graphics?" *Computer Graphics Forum*, 24.
Hastie, T., Tibshirani, R., and Friedman, J. H. (2001), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer.
Hershberger, S. L. (2003), "The Growth of Structural Equation Modeling: 1994–2001," *Structural Equation Modeling*, 10, 35–46.
Hsu, F.-H. (2004), *Behind Deep Blue: Building the Computer That Defeated the World Chess Champion*, Princeton, NJ: Princeton University Press.
Johnson, B., and Shneiderman, B. (1991), "Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures," in *Proceedings of the IEEE Information Visualization'91*, pp. 275–282.
Johnston, W. M., Hanna, J. R. P., and Millar, R. J. (2004), "Advances in Dataflow Programming Languages," *ACM Computing Surveys*, 36, 1–34.
Karr, A. F. (2006), "A Framework for Evaluating the Utility of Data Altered to Protect Confidentiality," *The American Statistician*, 60, 224–232.
Karr, A. F., Sanil, A. P., and Banks, D. L. (2006), "Data Quality: A Statistical Perspective," *Statistical Methodology*, 3, 137–173.
Kay, A. (1990), "User Interface: A Personal View," in *The Art of Human-Computer Interface Design*, ed. B. Laurel, Reading, MA: Addison-Wesley, pp. 191–207.
Kesselman, C., and Foster, I. (1998), *The Grid: Blueprint for a New Computing Infrastructure*, San Francisco, CA: Morgan Kaufmann Publishers.
Knüsel, L. (1998), "On the Accuracy of Statistical Distributions in Microsoft Excel 97," *Computational Statistics and Data Analysis*, 26, 375–377.
Kurokawa, S. (1997), "Make-or-Buy Decisions in R&D: Small Technology-Based Firms in the United States and Japan," *IEEE Transactions on Engineering Management*, 44, 124–134.
Lambert, D., and Liu, C. (2006), "Adaptive Thresholds: Monitoring Streams of Network Counts Online," *Journal of the American Statistical Association*, 101, 78–89.
Maedche, A. (2002), *Ontology Learning for the Semantic Web*, Boston: Kluwer Academic.
McCullough, B. D., and Wilson, B. (2002), "On the Accuracy of Statistical Procedures in Microsoft Excel 2000 and Excel XP," *Computational Statistics and Data Analysis*, 40, 713–721.
Motter, A., Matas, M., Kurths, J., and Ott, E. (2006), Dynamics on Complex Networks and Applications, *Physica D*, 224, vii–viii.
NISS (2007), "Data Confidentiality," available at *http://www.samsi.info/200506/ndhs/workinggroup/dc/*.
Norton, A., Rubin, M., and Wilkinson, L. (2001), "Streaming Graphics," *Statistical Computing and Graphics Newsletter*, 12, 11–14.
Oldford, W. (1999), "Mental Models and Interactive Statistics," in *Computing Science and Statistics: Proceedings of the 31st Symposium on the Interface*.
Park, S. W., Linsen, L., Kreylos, O., Owens, J. D., and Hamann, B. (2005), "A Framework for Real-Time Volume Visualization of Streaming Scattered Data," in *Proceedings of Tenth International Fall Workshop on Vision, Modeling, and Visualization 2005 (VMV 2005)*, pp. 225–232.
Perry, T. S. (2004), "John Gage: He Is the Network," *IEEE Spectrum*, 41, 32–33.
Phan, D., Yeh, R., Hanrahan, P., and Winograd, T. (2005), "Flow Map Layout," in *Proceedings of the IEEE Information Visualization 2005*, pp. 219–224.
Prechelt, L. (1999), "Technical Opinion: Comparing Java vs. C/C++ Efficiency Differences to Interpersonal Differences," *Communications of the ACM*, 42, 109–112.
Pregibon, D., and Gale, W. A. (1984), "REX: An Expert System for Regression Analysis," in *COMSTAT 1984: Proceedings in Computational Statistics*, pp. 227–236.
Royko, M. (1984), "Make My Day; Tell a Little Lie," *Chicago Tribune*, March 15, 1984.
Sanil, A. P., Karr, A. F., Lin, X., and Reiter, J. P. (2004), "Privacy Preserving Regression Modelling Via Distributed Computation," in *KDD'04: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM, pp. 677–682.
Sass, S. (1989), "A Patently False Patent Myth," *Skeptical Inquirer*, 13, 310–312.
Scannapieco, M., Virgillito, A., Marchetti, C., Mecella, M., and Baldoni, R. (2004), "The DaQuinCIS Architecture: A Platform for Exchanging and Improving Data Quality in Cooperative Information Systems," *Information Systems*, 29, 551–582.
Scott, D. (2003), "Introduction," *Journal of Computational and Graphical Statistics: Special Issue on Streaming Data*, 12.
Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. (2001), "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proceedings of the ACM SIGCOMM'01 Conference*, San Diego, CA.
Talmage, D. (2008), "Forecasting Future Disruptive Technologies," available at *http://www8.nationalacademies.org/cp/projectview.aspx?key=48795*.
Tessler, L. (1981), "The SmallTalk Environment," *Byte*, 6, 90–147.
Thisted, R. A. (1986), "Computing Environments for Data Analysis," *Statistical Science*, 1, 259–271.
Tufte, E. R. (2002), *The Visual Display of Quantitative Information* (2nd ed.), Cheshire, CT: Graphics Press.

——— (2003a), *Envisioning Information*, Cheshire, CT: Graphics Press.
——— (2003b), *Visual Explanations: Images and Quantities, Evidence and Narrative*, Cheshire, CT: Graphics Press.
Tukey, J. W. (1962), "The Future of Data Analysis," *Annals of Mathematical Statistics*, 33, 1–67.
——— (1977), *Exploratory Data Analysis*, Reading, MA: Addison-Wesley.
——— (1982), "Another Look at the Future," in *Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface*, pp. 2–8.
——— (1986), "The Interface With Computing: In the Small or in the Large?" in *Computing Science and Statistics: Proceedings of the 18th Symposium on the Interface*, pp. 3–7.
Tukey, J. W., and Tukey, P. A. (1985), "Computer Graphics and Exploratory Data Analysis: An Introduction," in *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics85*, Fairfax, VA: National Computer Graphics Association.
Tversky, B., Morrison, J. B., and Bétrancourt, M. (2002), "Animation: Can It Facilitate?" *International Journal of Human–Computer Studies*, 57, 247–262.
Wadlow, T. A. (1981), "The Xerox Alto Computer," *Byte*, 6, 58–68.
Wand, Y., and Wang, R. Y. (1996), "Anchoring Data Quality Dimensions in Ontological Foundations," *Communications of the ACM*, 39, 86–95.
Wang, A. (1986), *Lessons: An Autobiography*, Reading, MA: Addison-Wesley.
Wickham, H., and Hofmann, H. (2007), "Scagnostics in R," presented at 2007 Joint Statistical Meetings, Salt Lake City, UT, July 29.
Wilkinson, L. (2005), *The Grammar of Graphics* (2nd ed.), New York: Springer-Verlag.
——— (2008), "FASTAT: Statistics Without Manuals, Menus, Mice," in *Statistical Graphics: Data and Information Visualization in Today's Multimedia Society (DataViz VI)*, Jacobs University Bremen, Germany.
Wilkinson, L., Anand, A., and Grossman, R. (2006), "High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions," *IEEE Transactions on Visualization and Computer Graphics*, 12, 1363–1372.
Wills, G. (2007), "Scagnostic-Driven Autovisualization," presented at 2007 Joint Statistical Meetings, Salt Lake City, UT, July 29.
Wood, J., Dykes, J., Slingsby, A., and Clarke, K. (2007), "Interactive Visual Exploration of a Large Spatio-Temporal Dataset: Reflections on a Geovisualization Mashup," *IEEE Transactions on Visualization and Computer Graphics*, 13, 1176–1183.

Comment

John M. CHAMBERS

Summit, NJ 07901
(*jmc@r-project.org*)

Leland Wilkinson's overview and predictions for future trends are delightful and stimulating. It is a pleasure to contribute to the discussion. The article ranges widely over the question it considers, "the effects that future computing technology is likely to have on statistical computing." It would be great fun to endorse, elaborate on, or quibble with many of the points raised (although a challenge to keep to the editor's limitation on discussion length); however, my main concern is to examine the question itself. To borrow another phrase associated with John Tukey: "Better an approximate answer to the right question than an exact answer to the wrong question." There is nothing wrong with Lee's question, but I believe that those of us concerned for statistical computing ought to widen our focus, to ask what the future should be.

WHICH FUTURE FOR STATISTICAL COMPUTING?

Viewing the future of statistical computing in terms of computing technology *per se* may suggest a view based largely on the serving up of better technology to support our current lifestyle, specifically our current interests in statistical methodology and theory, but with faster computations and on a larger scale. Indeed, as the article points out, there are many such changes in computer technology, some already in process.

But is this the most relevant way to pose the question of the future? Neither statistical computing nor statistical data analysis itself exists for its own sake. Statistical data analysis aims to help people learn from data, through the concepts and techniques that it provides to support scientific studies, in a broad sense. Statistical computing in turn designs and implements computational techniques to support data analysis and other statistical needs.

This suggests that we might ask what the future of statistical computing *should* be, to best serve those who use it. Again, what are the right questions: Where should we put our efforts in research and teaching, to obtain the greatest benefit for science and society?

To ensure our own future and that of our planet, we must face some very tough issues in the coming years. Science potentially has much to offer in response. For most of the scientific studies, investigators must cope with large, complex sources of data that resist simple summaries. Statistical computing should have an important future role, provided that we focus on the key needs of the applications and exploit technological advances that respond to those needs. I would hope that we (both in statistical computing and in statistics generally) intend to be involved with such issues. To do our best, we need a more selective view of new technology, seeking and adopting new features that enhance our ability to support important studies. Both push and pull are involved; scientific tasks can push us to modify statistical computing to satisfy their needs, and new technologies can suggest better ways to organize and carry out the scientific studies.

Many current developments are potentially relevant, a number of them mentioned in sections 2 and 3 of Wilkinson's article. In this discussion, I can fit in only a few characteristic examples. It may help to keep in mind that serious collaborative data analysis projects nearly always involve three major aspects:

TECHNOMETRICS, NOVEMBER 2008, VOL. 50, NO. 4
DOI 10.1198/004017008000000532