TableLens: A Clear Window for Viewing Multivariate Data

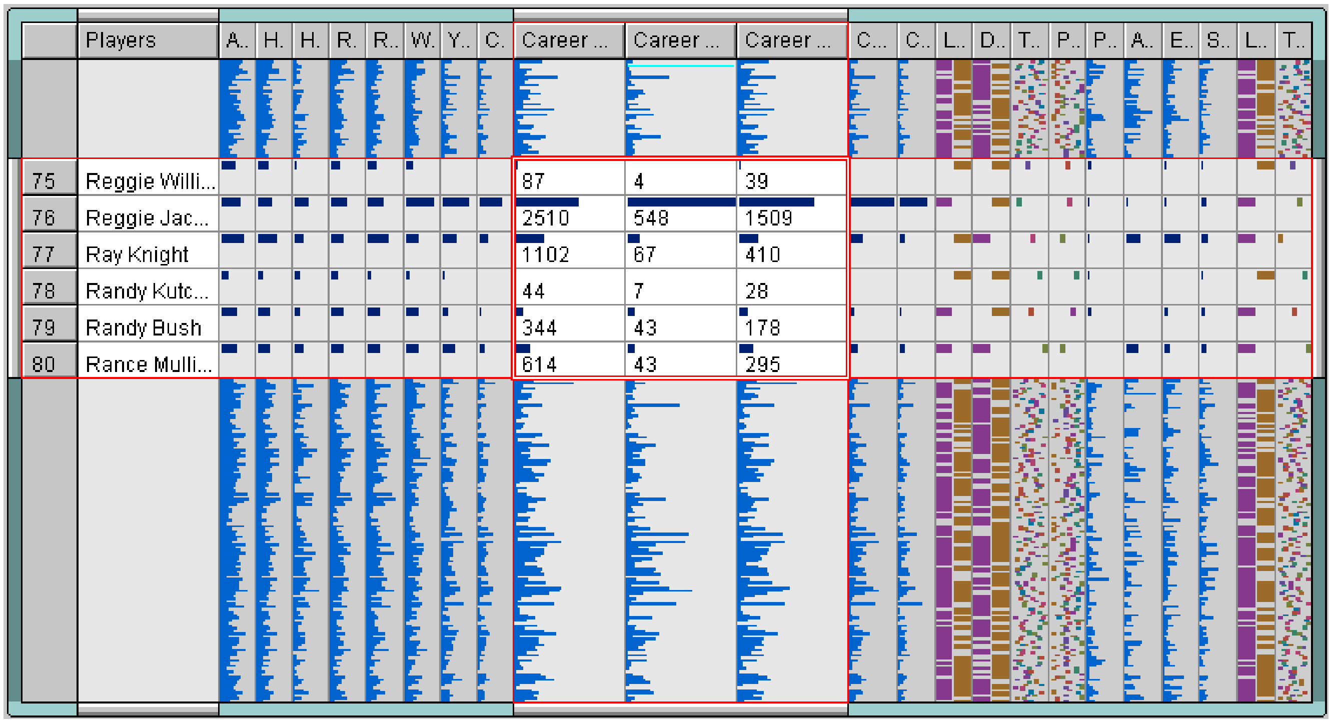
Ramana Rao

July 11, 2006

Can a few simple operators on a familiar and minimal representation provide much of the power of exploratory data analysis? This became the question driving the design of TableLens shortly after a somewhat accidental discovery. I had joined the Information Visualization team at Xerox PARC (Palo Alto Research Center), and we were on a march to invent techniques for a variety of canonical information structures. I was working on a concept called SpreadCube in 1993, which was an effort to perceptualize tabular data models in 3-D spaces.

Meanwhile, as life had it, I was training for a marathon, and I was keeping a running log as a tabular text file. On an 8-mile route, I visualized what my running log would look like if I colored the time run on a given day by the route. This thought led to a quick hack to display a table showing both numbers and colors in cells. And then tweaking away, I realized quickly that bars could be placed into the cell, miniaturized, losing the numbers, and voilà, TableLens was born.

TableLens, unlike most of the other visualizations created by our PARC team, is by itself a general application. Quite quickly it became apparent that a very small set of features provides tremendous exploratory power. Though generally it is important to demonstrate with data that people know or care about, in this article, I'll use original examples. Even after all these years, these "ancient" examples seem sufficient to illustrate TableLens' power and simplicity.



	Players	A..	H.	H.	R.	R..	W.	Y..	C.	Career ...	Career ...	Career ...	C...	C..	L..	D..	T..	P..	P..	A..	E..	S..	L..	T..
75	Reggie Willi...									87	4	39												
76	Reggie Jac...									2510	548	1509												
77	Ray Knight									1102	67	410												
78	Randy Kutc...									44	7	28												
79	Randy Bush									344	43	178												
80	Rance Mulli...									614	43	295												

Figure 1: TableLens puts graphics first in a familiar table structure so patterns and outliers can readily be seen.

TableLens enables exploring multivariate data sets by arranging data into tabular rows and columns. Figure 1 shows baseball hitting statistics from 1987 for 323 baseball players (the rows). It has 25 values (the columns) for each player. Some of cells are said to be in "focus," meaning sufficient space is allocated to display the values in those cells, while others are in "context," meaning they only contain graphical representations. The columns show quantitative statistics including *season and career at bats*, *hits*, *home runs*, and *RBIs* as well as categorical properties including *team* and *position* on the field. Quantitative variables are represented by graphical bars proportional in length to the represented values, and category values with a corresponding color and position within the field.

Though this example shows statistics for baseball players, it clearly can apply to many data sets of a very common organization called cases-by-variables arrays, multivariate data, and relational tables. Such data sets are widespread in science, business, economics, government, education, and modern life. Consider how widely relational databases are applied.

Other simple examples include:

- Clinical trials patients with health and trial effect data
- Companies with various financial and investment data
- Parts or products with descriptive, pricing, availability information
- Countries with their geographical, economic, and demographic properties
- Kinds of cars with their physical or performance characteristics
- Stocks or mutual funds with various ratios and ratings

TableLens has a few features and operators that provide its basic power:

- *Parallel Cases*. That's fancy for *table*. Tables are familiar to many people. Contrast this with parallel coordinates (a complicated visualization of multivariate data that uses a variation of a line graph), which are quite powerful but still quite unusual. By keeping cases (that is, objects) aligned across the rows, it's easy to process them *en masse* by scanning up and down and across.

- *Put Graphics First*. No wizards or summaries or anything to get graphics, just all the data arrayed visually within a table. A thousand bars can be scanned much more rapidly than a thousand numbers shown as text. This allows easy spotting of trends, correlations, outliers, and so on within and across samples.

- *Sorting*. Sorting the table by different variables leads to a simple and intuitive way of understanding the shape and spread of values of a given variable, as well as a means to spot correlations between variables.

- *Focus Plus Context*. TableLens allows "opening" up regions that also show textual values along with the graphical representation. This ability to see specific values in context means that there isn't a back and forth shuffling between different views.

- *Rearrangement*. Not to be underestimated is the ability to reorder columns, not only to use proximity as an external cognitive aid, but also because manipulation itself supports a fluid and dynamic thinking process.

Let's look in a little more detail. Sorting a column provides extremely powerful analytical capabilities. Many properties of the batch of values in a sorted column are apparent by examining the graphical marks (for example, bars) and the shape of the curve in the column. Essentially, a sorted column serves to explore the sample of values (as does a boxplot); but just as importantly, after one variable has been sorted, if another variable is correlated, then its values will also appear to be sorted. Thus, looking for correlated variables is a matter of scanning across the columns to identify other columns that exhibit a similarly shaped descending curve or one that approximates a mirror image of the curve.

In Figure 2, the user has clicked on the *At Bats* column header, thus sorting the rows by that column's values. A number of other quantitative variables nearby now appear roughly sorted, thus revealing correlation. Salary also is very roughly correlated, though as would be expected, there must be other factors. A next question might be: Who is the guy that makes so much money?

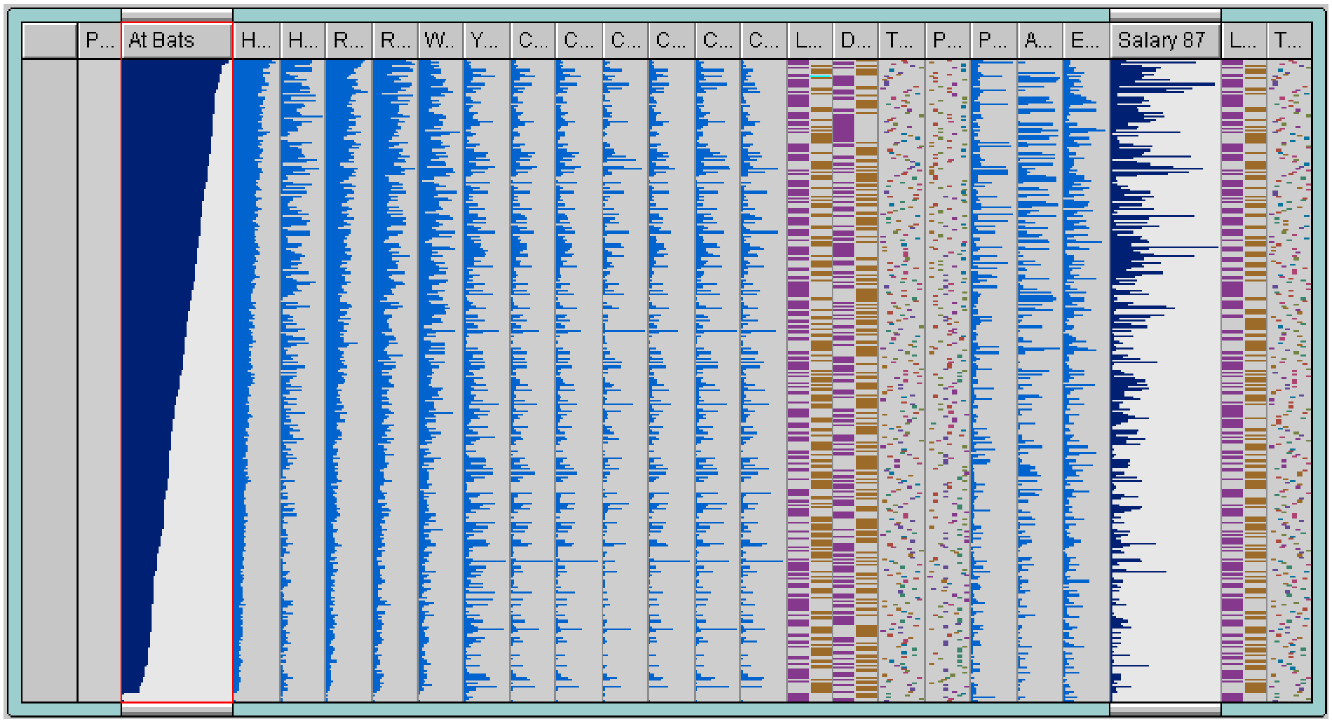


Figure 2: The user has sorted rows by *At Bats*.

Figure 3 shows a different data set comprised of 500+ cars and nine variables. The table was sorted by MPG (miles per gallon) within a sort by a categorical variable, the origin of the car (European, Japanese, or American). Besides various quantitative correlations, a number of correlations with categories also show up. For example, at the time of this data, America made all the eight-cylinder, gas-guzzling cars.

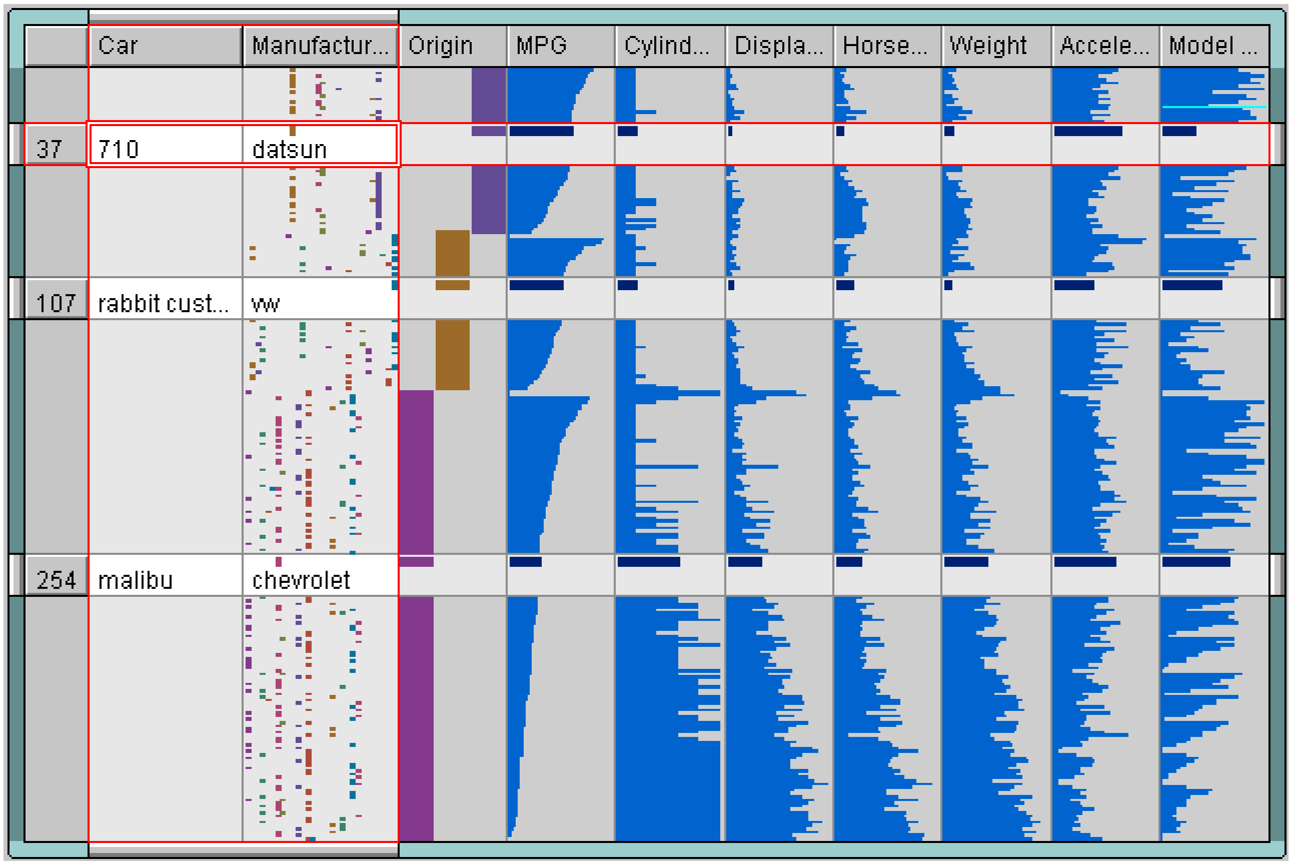


Figure 3: 500 cars sorted by *Miles per Gallon* and *Origin* (American, European, or Japanese)

TableLens can scale to more rows than the number of vertical pixels available by mapping multiple values into a single pixel line. By aggregating the multiple values using a choice of min, max, median, or random, various tasks can be handled. In any case, even with 500 values, statistically valid inferences can be made with random samples, since the absolute size of a sample is more important than the proportion of the population it covers.

The use of graphical representations clearly provides a scale advantage because the bars can be scaled to one pixel wide without disturbing comparisons. Scanning across the baseball data in a spreadsheet would require 18 vertical and 3 horizontal screen scrolls of a similarly sized window. More importantly, because the cost of processing information is now greatly reduced, new methods of exploration and analysis are made available to broader sets of users. At a glance, patterns and outliers are apparent; and then, opportunistically, the user can focus on portions of the table of interest for more detailed observation. Thus is enabled the data-driven style of exploration espoused in EDA (exploratory data analysis, as first described by the late Princeton statistician John Tukey).

The combination of features here is unique and emerged rather rapidly in the right place and right time for our PARC visualization team in 1993. Certainly, various ideas from others relate as background influences to TableLens. The most important of these include spreadsheets in general, Lotus Improv in particular, exploratory data analysis, and graphical manipulations as inspired by Jacques Bertin.

The field of exploratory data analysis offers some inspiration for building systems focused on exploration and discovery. In Tukey's words, EDA is "about looking at data to see what it seems to say." As opposed to utilizing data to confirm a carried-in hypothesis or belief, it is about examining the data and following what is seen. Much true knowledge work is about finding the right questions to ask as much as it is about answering known questions. In a sense, it is seeing before proceeding.

TableLens supports exploratory analysis in a form suitable to a much broader range of people than conventional data analysis tools. Rather than supporting the entire repertoire of EDA methods, we sought a tool with a few good operations, one that might be learned in five minutes or so. The basic operations of data analysis—looking for correlation, spotting outliers or peculiar values, forming and comparing groups of items—can support a wide range of domains and activities. Not surprisingly, exploratory data analysis, just as statistics or mathematics, is fundamentally domain-independent and widely applicable.

Author's Note: *TableLens is patent-protected technology available from Inxight Software in a number of different product forms.*

About the Author

Ramana Rao is a Founder and former CTO of Inxight Software. At Inxight and previously at Xerox PARC, Ramana performed pioneering work in intelligent information access, digital libraries, information visualization, and user interfaces. He is the inventor, along with Stuart Card, of TableLens.

This article originally appeared on the Business Intelligence Network (www.b-eye-network.com) as one of Stephen Few’s guest articles. A library of Stephen Few’s articles, as well as other guest articles, are available at www.perceptualedge.com.