Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

GRAPHICAL TECHNIQUES FOR OUTPUT ANALYSIS

David Alan Grier
Department of Statistics /
Computer and Information Systems
The George Washington University
Washington, District of Columbia USA 20052

ABSTRACT

This tutorial gives a summary of current research in graphical statistical analysis and shows how to apply these techniques to a range of problems in simulation output analysis. The tutorial is not tied to a specfic software package. It covers methods that may be found in many different products. The examples in the tutorial were done by the S system from AT&T Bell Laboratories.

1 INTRODUCTION

Graphical analyses of simulation output are usually easier to explain than mathematical analyses. A visual display of the results quickly conveys information about the model that might require hours of study to glean from mathematical results. Less mathematically inclined audiences, such as the managers who must rely on simulations to support their decisions, find graphical analyses easier to comprehend than pages of numbers from an analytical model. Graphical analyses also tend to promote a dialogue between the producers and users of simulation output. Decision-makers can question results without having to put their questions into a mathematical formulation. In probing a model, they can simply point to a graphical feature and ask questions based on it. A final benefit of graphical techniques is that they are non-parametric. They bring to the analysis fewer underlying mathematical assumptions about the output. Many mathematical models require the output to be Gaussian, exponential, poison or to fall into some family of distributions. Often these assumptions are mathematically untenable, or at least of dubious quality. Because of their non-parametric nature, graphical techniques are less misled by deviations from underlying assumptions.

This tutorial will look at five basic classes of problems and explore graphical methods for addressing them. The five problems are:

1. Validating a simulation by comparing a single output against target data;
2. Comparing a single output across many simulation models;
3. Characterizing a single output as the simulation changes over time;
4. Characterizing a complicated simulation based on many outputs;
5. Comparing many complicated simulation models.

While many of the methods described in this paper are not part of commercial presentation graphics software, they are commonly found in commercial statistical analysis packages. The paper uses the S system, described in Becker, Chambers and Wilks (1988) that was runs on Unix workstations and DOS PC's. The principal advantage of this tool is that it appears, at least temporarily, to be the research tool of choice in the statistical community. Statisticians are constantly writing new algorithms and functions for it. Many of these functions may be retrieved from the STATLIB file server at Carnegie Mellon university. (To get information about STATLIB, send electronic mail to statlib@temper.stat.cmu.edu with the single line: SEND INDEX. The fileserver will return a list of available software and further instructions for the use of STATLIB.) Other packages that can be used for these analyses include SYSTAT, JMP, Statexec, SPSSX PC, LispStat, and Number Cruncher.

2 GRAPHICAL MEASURES

In performing graphical analysis, we are usually interested in three quantities: the center, the spread and the distribution of the data. The center of the data, commonly measured by the mean or median, is often the focus of the analysis. In a typical simulation study, we are attempting to estimate average delay, average traffic flow, average time in the system. The center of our data, whether measured by the arithmetic mean, the median or a batched mean, is an estimate of the true average.

The analysis is incomplete if confined to the center of the data. The spread, measured by standard deviations, median absolute deviations and confidence regions, measures the extent to which the data clusters

around the center. Even though models are often built to study mean effects, such effects cannot be separated from spread effects. For a simple example, we can construct an inventory system problem in which we are attempting to guarantee that every request is filled within 2 hours of its submission and within 1 hour on average. A simulation study might discover a simple restructuring of the system that vastly improves the system for a few numbers of parts and lowers the average time to, say, 45 minutes. To make that improvement, that change might slightly degrade the performance of the system for most parts, while improving the performance in retrieving a few heavily requested parts. Such a change increases the spread in times and increases the number of parts that cannot be found within the two hour goal.

The above example illustrates that unless you restrict studies to fixed families of distributions, the center and spread of a data aren't enough to characterize the process that produced the data. Different sets of data can possess the same mean and standard deviation, for example, but have wildly different distributions.

3 VALIDATING THE OUTPUT OF A SIMULATION

Problem:

The results from this section come from a finite horizon performance study of an interleaved, anticipatory cached disk drive. It is a small model that will eventually be incorporated into a larger processor model. If it does not accurately represent the physical subsystem, The larger model will be of dubious value. The simulation consists of feeding a stream of disk seek instructions into the drive and recording the time required to fulfill each request. We will compare two sets of simulated data to a single set from the physical system. We denote the set of data from the physical system as X_0 and the two sets from the simulation runs as X_1 and X_2. In this example, the means and standard deviations of the three sets were equal to three significant digits and no statistical test was used to determine if they differed. What remained was to verify that the two sets of simulation output had the same probability distribution as the original data. This comparison was done by means of a QQ plot and confidence bands.

Method:

A QQ plot is a quantile - quantile plot. The name is a bit misleading because it actually uses percentiles of the data. To compare the distributions of two data sets with a QQ plot, we plot the percentiles of the data pairwise on a scatterplot. In this example, the data sets are of the same size, this means that the data sets are sorted from smallest to largest and the data are plotted pairwise, matching smallest with smallest and so on. In general, the data sets are often of different sizes. In those cases, the QQ plot is constructed by using the empirical cumulative distribution functions of the data. In such a case, the QQ plot for comparing data set 1 with data set 0 would be constructed by graphing $(x, F_1^{-1}(F_0(x)))$. Here, both $F_0()$ and $F_1()$ are empirical distribution functions of data set X_0 and X_1. If data set X_0 has N_0 points, we can estimate $F_0(x)$ as $F_0(x)$ = (the number of points $\leq x$) / N_0, where N_0 is the number of points in data set X_0. $F_1()$ and $F_2()$ are computed in a similar fashion.

$F_1^{-1}()$ is simply the inverse of the function $F_1()$. If there are N_1 data points, $F_1^{-1}(x)$ is the $[x*N_1]$ largest member of the data set X_1, where [ ] represents the function that rounds to the nearest integer. If $x \leq 0$, then $F_1^{-1}(x)$ = smallest member of X_1. If $X \geq 1$, then $F_1^{-1}(x)$ = largest member of X_1.

If the two distributions are the same, then the QQ plot should form a diagonal line. Figure 1‡ gives a QQ plot between the data from the physical system and the data from the first simulation, X_1. The data points don't exactly fall on the diagonal line, but they are close. A diagonal line would clearly fall within the 95% confidence band drawn around the QQ plot. The confidence band is like a confidence band for a mean. Since we are using the data to estimate the true distribution function of the data, the QQ plot is no more than an estimation. We are 95% certain that the true QQ plot falls within the confidence band. The confidence band is calculated by the formula $F_1^{-1}(F_0(x) \pm d + 1/(2N_0))$, where d is the 95% critical value from the Kolmogorov-Smirnov distribution

Figure 2‡ compares the distributions of the data from the physical system with the data from the second simulation, X_2. Note that many points fall outside the confidence interval. Clearly the distribution of the two data sets differ. The second distribution is much more skewed. From the analysis we can conclude that the first simulation is indeed a good representation of the physical system. However, the second simulation does not appear to be a good model for the physical disk drive.

References:

Law and Kelton (1991, p380 ff) give a good overview of the QQ plot. The confidence bound may be found in Doksum and Sievers (1977).

4 COMPARING A SINGLE OUTPUT ACROSS MANY SIMULATION MODELS

Problem:

The data in this section are from a simulation study of 4 different machine tool configurations in the manufacture of line transformers. The study looked at the average amount of time it took an transformer core to go through the line. The data was collected by gathering batched means. We want to compare the speed and efficiency of the 4 configurations. Our data will be denoted X_1, X_2, X_3, and X_4,

Method:

A boxplot summarizes the distribution of data in a simple, concise form to make it easy to compare. Figure 3‡ gives a box plot for the first version of the study. In this data set, each core took between 17 and 30 seconds to travel though the line. The central box of the box plot represents the middle 50% of the data. For this data, it indicates that the middle 50% of the data took between 19 and 21 seconds to travel though the assembly process. The line in the middle of the box represents the median of the data. For the first configuration, half of the parts took more than about 20 seconds and half took less. The two dashed lines connected to the top and bottom of the boxes are whiskers and represent the remaining 50% of the data. These lines are constructed by the following algorithm. From the top and bottom of the box, we draw lines that extend 1.5 times the length of the box. We then shrink the lines back until the end of the line is at the location of the largest (or smallest, for the bottom line) less than 1.5 times the length of the box from the end. The method is easier to do than describe. It is done to ensure that the end of each line represents an actual data point. The stars near the top represent data points that are far from the bulk of the data. The points marked by stars are often called outliers and indicate points that are not representative of the bulk of the data.

Figure 4‡ gives the 4 box plots from the 4 simulation models. The second model is a clear improvement over the first, although not all the data demonstrate that improvement. The third model is slower and has a higher variability. The last model does not have as low a mean as the second model, but it has a much tighter variance, indicating a greater uniformity.

To check to see if the medians are statistically different, we use a notched box-plot, such as the one in Figure 5‡. Two medians are different at a 5% level if the notches from their two boxes do not overlap. The notches of box 2 and 4 overlap, indicating that the medians of the two data sets are indistinguishable, and that perhaps setup 4 is preferable to setup 2.

References:

Law and Kelton (1991) give a brief discussion of box-plots (381 ff). Other references are Tukey (1977) and Chambers, Cleveland, Kliener, Tukey (1983).

5 CHARACTERIZING A SINGLE OUTPUT OF A SIMULATION AS THE SIMULATION CHANGES OVER TIME.

Problem:

This problem came from a large simulation of an interactive television show for the final approval report for the FCC. People were able to participate in the show from their homes using a push-button phone. The intent of the simulation was to demonstrate that the game show winners would not be determined by the arbitrary routing of telephone calls. The simulation is of the phone network that ties the participants into the show. The purpose of the simulation is to estimate the number of lost or delayed messages, which are called "exceptions". Every 10 minutes, the simulation returns the number of exceptions. The simulation is run as a finite horizon simulation to simulation the 3 and a half hour run of the show.

Method:

Figure 6‡ gives a plot of the number of exceptions plotted against time. Estimating the center or the expected number of exceptions at any time, is the line passing through the data. It is a smooth function and gives a better representation of the average behavior at any time than the jagged plot of the raw data. The smooth function in Figure 6‡ was created by a running median smooth. To compute the value of the smooth at time t, the program takes the median of a small group of points on either side of time t, under the assumption that the curve is changing slowly. The straightness or curvature of the smooth is determined by the bandwidth, the number of points collected to estimate the median. A large bandwidth creates a flatter curve. A small bandwidth creates a more wriggly curve.

There are several method of smoothing curves. The median smooth, described above, has the advantage that it is robust against large, unusual data items, such as the ones that are found at the top of Figure 6‡. Another smoother is the lowess smoother (locally weighted smoother), which creates its smooth by averaging points. An example of a curve created by lowess can be seen in Figure 7‡. This version has a larger bandwidth than the median smoother. If we set the bandwidth smaller, we would see a curve similar to one produced by the median smoother. The lowess smoother has the advantage that it is easy to code and prepare.

Neither the lowess smoother nor the median

smoother can interpolate between points. The best smooth for that purposes is the spline smoother. A spline (without the smoother) is a curve that is created from piecewise polynomials. The polynomials are joined end to end, creating a smooth curve. In addition, the coefficients are chosen to make some of the derivatives of the resulting curve continuous. Cubic polynomials with 2 continuous derivatives are a commonly used spline. A smoothing spline is a spline that represents the data without going through all the points. The coefficients are computed by solving a constrained optimization problem, called a penalized likelihood problem. Like the other smoothers, it has a concept of bandwidth. Figure 8‡ gives a spline smoothed version of the data, with approximately the same bandwidth as the lowess smoother.

All three smoothers identify the rise in exceptions at 100 minutes into the show. The basic structure is the same across the three. The three smoothers have different mathematical properties and different advantages. The median smooth is robust or resistant to deviant observations. The lowess smoother is simple. The spline smoother can interpolate between points.

References:

The median smoother is presented in Tukey (1977) and Mosteller and Tukey (1977). The lowess smoother is found in Cleveland (1981) and the best treatment of splines and spline smoothers is found in de Boor (1978). A more theoretical treatment is in Buja, Hastie and Tibshirani (1989).

6 CHARACTERIZING A COMPLICATED SIMULATION BASED ON MANY OUTPUTS;

Problem:

The data for this example comes from a simulation of a scalable RISC computer architecture. The architecture be divided into three components, instruction decoder/integer arithmetic unit, io/memory controller and floating point/graphics accelerator. The three components are independent in one sense, but interact strongly with each other. The purpose of the simulation is to determine final performance and to gain a sense of how the three components work together. The simulation is run as a steady state simulation, with batch means of the percent utilization taken for each section every 100,000 simulated clock cycles.

Method:

This section will present two related methods, one static and other dynamic. The static method is presented in Figure 8‡. Figure 8‡ is a scatterplot matrix, a matrix of all possible scatterplots. Scatterplots are duplicated in the matrix and the plots that would go down the diagonal are simply plots of variables with themselves and are deleted. In the lower left corner, the relation between the first and third sections is clearly seen. When the first section (instruction decoder/integer arithmetic unit) is fully utilized, the third (graphics and floating point) is idling and vice version. The relation is a nice, clear curve. The relation between the second section and the third is seen immediately to the right. It, too is a decreasing, but some what nosier and less strong relationship. The only increasing relationship is seen between section 1 and section 2. As section 1 becomes heavily utilized, so does section 2. The scatterplots in the upper right hand corner are mirror images of those in the lower left.

Scatterplot matrices have had several important generalizations. The first is brushing. Brushed scatterplots is an interactive graphic, in which the researcher identifies some points of interest in one plot and highlights them. The corresponding observations are highlighted in the remaining plots.

The second generalization is rotation. This is a dynamic graphic in which a three dimensional scatterplot is projected onto the computer screen. Using a mouse or a tracker ball, the researcher can then rotate the point cloud, looking for the most interesting projection. The most interesting projection is usually one that shows the strongest relationship and least randomness amongst the data.

For more than three variables, this technique is awkward. As the number of variables gets large, the process of working through projections and rotations of large data sets is extremely time consuming and difficult. Part of the problem is that most projections of large data sets look like Gaussian data. A graphical method that simplifies this process is called the Grand Tour. It is a method that cycles through all possible projections of a high dimensional data set are projected onto the screen in a reasonable order. By studying these projections, the user can again look for the strong relationship among the data.

Even the Grand Tour can be tedious for high dimensional data. It can involve hours of watching data points move on a screen. An automated alternative is projection pursuit. Projection pursuit is a computationally intensive method that attempts to find the best projection and rotation of a large, multivariate data set. Instead of projecting pictures on a screen and asking a researcher to choose the best, it uses a numeric optimization routine that sweeps through the set of all possible projections and rotations in an attempt to optimize an objective function.

References:

Scatterplot matrices and brushing are discussed in Becker, Cleveland, Wilks (1987) and Cleveland, Chambers, Cleveland, Kliener, Tukey (1983). The Grand Tour is found in Asimov (1985) and projection pursuit in Friedman, and Tukey (1974). All of these methods are available in the S system, described in Becker, Chambers and Wilks (1988).

7 COMPARING MANY COMPLICATED SIMULATION MODELS.

Problem:

Once a complicated model has been built, there is often a need to compare it, in its full complication, with other models. Again, we will be looking at data from a simulation of scalable RISC computer architecture. Each model is characterized by 17 different measurements, including ready queue depth, bus utilization, io queue depth, io utilization and so on. In this case we studied five different configurations of the computers. The simulations were done as steady state simulations, with 17 batch means taken every 100,000 simulated cycles.

Our graphical methods need to be able to capture the complexity of the model in a simple form. Figure 9‡ is a star plot that represents the means of the 17 measurements for the simulation of the standard configuration of the RISC computer. Each arm of the star radiating from the center is proportional to the average of one measurement. In this figure, the arms are labels, so that we can see that this configuration had a low memory utilization, cache hits and processor utilization, but fairly high io queue length, average number of segment faults, ready queue length and elapsed time per job. Lengths of the are standardized by dividing each by the longest measurement.

Method:

To compare the different computers, we can look at Figure 10‡, which gives star plots for the five configurations. Immediately we can identify similar and dissimilar configurations. The Standard configuration is similar to the Extend configuration without the Accelerator. The Full Address Mode and the Extended Configuration are another similar pair. The Revised Configuration is clearly unlike any of the others. By returning to the labels, we can see that the Full Address and the Extended Configuration have high utilizations and small queue sizes and behave in similar fashions.

Star plots are one among a number of multivariate techniques for comparing models. They are one of the more scientific methods and one of the most universally applicable. Many methods, such as Chernoff faces, are too vague to be scientific and too silly to be given to decision makers. On last technique that may be useful is the Andrews plot. Andrews plots reduce a multivariate set of data to a curve, by reversing Fourier analysis. If you have multivariate measurements a_1, ..., a_n, for a single model, then the Andrews curve for that model is $\Sigma_1^n a_i \cos(x2\pi/n)$. The Andrews curves for the five computer models are given in Figure 10‡. The darker line running through the mess of wiggles is the convergence of the Full Address and the Extended Configuration models. From the curves, it is impossible to read the individual features, but they can be quite useful, especially if you have a large number of models that fall into a small number of classes.

References:

Andrews plots, start plots and other tools for viewing multidimensional data are described in Andrews (1972).

8 SUMMARY

The methods described above are just a few of many graphical techniques for output analysis. Good overviews of the state of the art include Chambers, Cleveland, Kliener, and Tukey (1983), Cleveland (1985), Cleveland (1987), and Cleveland and McGill (1988). A good idea book for planing new techniques is Tufte (1983).

Much current research is directed toward finding complete environments for graphical analysis. One of the more complete, state of the art research systems is X-Gobi. It is a system that contains collection of graphic primitives which can be used to build complicated graphical analyses. These primitives range from scatter plots to dynamic projections and rotations. It runs as part of the S system on unix work stations that have the X windows user interface.

REFERENCES

Andrews, D. 1972. Plots of High Dimensional Data. *Biometrics* 28:125-136.

Asimov, D. 1985. The Grand Tour: A tool for viewing multivariate data. *SIAM Journal of Scientific and Statistical Computing*, 6:128-143.

Becker, R. A., J. M. Chambers and A. R. Wilks. 1988. *The New S Language*, Pacific Grove, CA: Wadsworth and Brooks/Cole.

Becker, R. A., W. S. Cleveland, and A. R. Wilks. 1987. Dynamic Graphics for Data Analysis. *Statistical Science* 2:355-395.

Buja, A., T. Hastie and R. Tibshirani. 1989. Linear Smoothers and Adaptive Models. *The Annals of Statistics* 17:453-555, with discussion.

Chambers, J. M., W. S. Cleveland, B. Kliener, P. A. Tukey. 1983. *Graphical Methods for Data Analysis*. Boston: Wadsworth International, Duxbury Press.

Cleveland, W. S. 1985. *The Elements of Graphing Data.* Pacific Grove, CA: Wadsworth Press and Brooks/Cole.

Cleveland, W. S. 1987. Research in Statistical Graphics. *Journal of the American Statistical Association* 82:419-423.

Cleveland, W.S.and M. E. McGill. 1988. *Dynamic Graphics for Statistics*, Pacific Grove, CA: Wadsworth and Brooks/Cole.

Cleveland, W. S. 1981. LOWESS: A program for smoothing scatter plots by robust locally weighted regression. *The American Statistician* 35:54.

de Boor, C. 1978. *A Practical Guide to Splines*, New York: Springer-Verlag.

Doksum, K. and G. L. Sievers. 1977. Plotting with confidence: Graphical comparisons of two populations. *Biometrika* 63:421-434.

Friedman, J. H., W. Stuetzle, and A. Schroeder. 1984. Projection pursuit density estimation, *Journal of the American Statistical Association*, 79:599-608.

Friedman, J. H., and J. W. Tukey. 1974. A projection pursuit algorithm for exploratory data analysis, *IEEE Transactions on Computers* C-23:881-889.

Law, A. and W. D. Kelton. 1991. *Simulation Modelling and Analysis*, McGraw-Hill, New York.

Mostelleter, F. and J. Tukey. 1977. *Data Analysis and Regression*, Reading, Massachusetts: Addison-Wesley.

Tufte, E. R. 1983. *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.

Tukey, J. 1977. *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

‡The figures will be provided at the presentation as a handout.

AUTHOR BIOGRAPHY

DAVID ALAN GRIER is an Assistant Professor of Computer and Information Systems at George Washington University. He received a BA in mathematics from Middlebury, and MS and PhD degrees from the University of Washington. He is currently director of Honors Education at the George Washington University. His current research project involves the study of large scale computer networks.