CREATIVE USES OF STANDARD GRAPHICS PACKAGES
Julie A. LaBarr, Eastman Kodak Company

INTRODUCTION

Standard graphics packages provide the building blocks for displaying data. Often the way a package would conveniently display your data is not exactly the way you want it. With some effort, you can make these packages produce the plot you want. There are several considerations that must be made to determine the feasibility of this effort. This paper discusses these considerations and gives examples of programs written at Eastman Kodak Company that make graphics packages do "more".

Once the plot program is written, it needs to be accessible by end-users. User interfaces provide this accessibility. This paper also discusses several user interfaces written at Kodak.

GRAPHICS PACKAGES - Producing the Plots YOU Want

Graphics packages vary in sophistication, ease of use, and capabilities. A package that offers "user-friendliness" may have a limited selection of plot types (e.g. line graphs, barcharts) or limited enhancement capabilities (e.g. control of colors, line patterns, etc.). More sophisticated packages require more programming expertise. A problem arises when you have an application that does not fit the graphics package.

You start with an idea or a need. You investigate what package is appropriate for your application. This can involve several things:

- which packages you have available at your installation,
- what capabilities these packages have, and
- reasons why you must use a particular package.

You then cost-justify your application. Will this be a widely used program or is it for a one-time analysis? Will better interpretation of the data be obtained? Who is the display for? How much extra effort, if any, is needed to make the graphics package create your plot? You will generally take one of three paths:

- compromise to a less than desired result,
- accept the application as it exists currently, if it does, or
- make the extra effort and get the plot you want.

There are several reasons for making the extra effort:

- to convert a manual operation into a computerized one,
- to make a more informative display,
- to conform to accepted standards, or
- to obtain additional capabilities.

Let's discuss examples of plots generated at Kodak in each of these categories.

Converting a manual operation into a computerized one.

TRILINEAR PLOTS: Areas that work with color films use trilinear plots to look at the balance between the three primary colors red, green, and blue. This is not a new technique to Kodak. It has simply been performed by hand in the past. Because of the widespread use of this technique and the considerable amount of labor hours that would be saved with a computerized approach, we created a SAS(R) procedure using the DISPLA (R) graphics package to generate these plots.

The following is an example of a typical hand-drawn trilinear plot:

\[\text{Figure 1. Hand-drawn trilinear plot.}\]

The data shown in Figure 1 is simulated. Three tests A, B, and C are displayed. We are interested in how close each point is to having equal red, green, and blue components. The circle represents the aim value -- the region in which we would like the data to fall.

Here is our computerized version:

\[\text{Figure 2. Trilinear SAS procedure.}\]
Besides eliminating the manual work, the plot in Figure 2 is easier to read. Our trilinear procedure allows the user to use defaults for a quick, simple plot, to overlay plots of separate data groups on one plot page (shown in Figure 2), to control axis scaling, to specify what symbols or labels they want plotted at each point, and to plot confidence ellipses around the data.

Making a more informative display.

PROCESS VARIABILITY PLOTS: Process variability plots are used to show the variation of responses attributable to changing levels of experimental factors. These plots have been incorporated into our design of experiments package. This package is a SAS based system. All graphics currently in this package use the SAS/GRAPH product because of its "replay" capability. Consequently, these plots were created in SAS/GRAPH as well. This program is based on the technique described by Ronald D. Snee in his article "Graphical Analysis of Process Variation Studies" (1983).

Our sample data is a nested design:

<table>
<thead>
<tr>
<th>WEEK</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>RUN</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
</tbody>
</table>

We want to look at how the response varies from WEEK to WEEK, DAY to DAY within WEEK, and RUN to RUN within DAY. Let's look at a scatter plot of the data:

Scatter Plot
Simulated Data

Figure 3. Scatter plot.

It is difficult to determine the variabilities from Figure 3.

SAS has a procedure available, PROC SPLAT, that creates schematic printer plots. Process variability can be displayed via this technique:

Figure 4. PROC SPLAT plot.

Schematic plots provide much information about the data if you know how to interpret the display. For example, the dashed line in the body of the rectangle corresponds to the median. But this display loses detailed information. For example, we cannot determine the RUN to RUN within DAY variability from this plot.

The technique described by Snee provides this detail:

Process Variability Plots
Simulated Data
(Outer Boxes - WEEK Inner Boxes - DAY)

Figure 5. Process variability plot.

Figure 5 is essentially Figure 3 with boxes drawn to display WEEK and DAY. Drawing the boxes was not a simple task to perform in release 82.3 of SAS/GRAPH. Of course, the user could draw the boxes himself, but that can become time consuming for large amounts of data. As well, this program allows the user to easily vary combinations of experimental factors.
TRILINEAR PLOTS WITH CONTOURS: This plot is an offshoot of the trilinear plot program described earlier. The hexagon portions of the programs are the same, but in this instance, we're plotting contours instead of sensitometric values. At Kodak we perform print-judging studies, i.e. people's perceptions of color prints. We specifically printed negatives in systematic color variations to see what color balances people preferred in different types of prints (e.g. scenic prints versus portraits). The results can help determine what tolerances our printers should have.

Effective display of this data was critical. The results needed to be easily interpretable by management. As these example plots show, an in-depth description on what the plot is saying is unnecessary:

**TRILINEAR MODEL CONTOUR PLOT**

![Figure 6. Portrait.](image)

The contour lines in Figure 6 represent the fraction of respondents who judged a portrait at various color balances as acceptable. You will note from the contours that people prefer red hues in such prints -- they like the prints to make people look tanned.

**TRILINEAR MODEL CONTOUR PLOT**

![Figure 7. Scenic print.](image)

In Figure 7, the contour lines represent the same as in Figure 6 but for a scenic print. You will note from the contours that people prefer blue hues in scenic prints -- they like blue skies and blue water.

Conforming to accepted standards

**SHEWHART CONTROL CHARTS**: Control charts are basic tools of statistical process control. At Kodak, we have started a massive program to train our people in these concepts and to enforce their use. As part of this program, computerized tools were needed. We've written a package that contains some of these tools, including X-bar/Range and X-bar/standard deviation control charts. This package is a SAS based system. We selected SAS/GRAPH because of its "replay" capabilities. The format of the Shewhart control charts is to have the X-bar and either the range or standard deviation chart on the same page. This is so the long-term and the short-term variabilities can be viewed at the same time. We could generate these plots easily on separate pages (Rodriguez 1985):

![Figure 8.A X-bar chart.](image)

![Figure 8.B Standard deviation chart.](image)
ONLY LONG TERM VARIABILITY
OUT OF CONTROL

Figure 9. Shewhart control chart.

The data in Figures 8 and 9 is simulated. In this case, the X-bar chart (long-term variability) is out of control, but the standard deviation chart (short-term variability) is not. The concern with separate plots is that the user may look at the X-bar chart only and ignore the standard deviation chart -- i.e. the user would not get the whole picture.

Obtaining additional capabilities.

MIXTURE RESPONSE SURFACE CONTOUR PLOTS:
Mixtures are combinations of ingredients that together result in some end product. Responses are modeled as a function of the proportions of the ingredients. Based on these models, mixture response surface contour plots display how changing proportions of ingredients affect the final mixture. Visual display is required for interpretation.

A printer plot is available to display this kind of data (Hare and Brown 1977). We wanted to add capabilities to the program. Our choice was to modify the printer plot program or to rewrite the program in a graphics language. We chose the latter and got much more readable plots with added features. Let's look at the printer plots:

Figure 10. Printer mixture plot.

In this example, we are developing a fruit punch. Our ingredients are orange juice, pineapple juice and grapefruit juice. Our response is flavor in terms of fruitiness:

```
1
not
5
average
10
exremely
fruity
fruitiness
fruity
```

We're interested in which combinations of juices will produce a fruity tasting punch. The data (Cornell 1981) in Figure 10 seems to suggest that higher levels of either orange juice or pineapple juice produce a fruitier tasting punch. Our graphics version, written in DISSPLA, has four-ingredient vs. three-ingredient capability. It allows taking slices at fixed values of the fourth ingredient. The printer plot program looks at the entire space (0%-100%) for each of the three ingredients. Our graphics version allows us to take a subset of the entire space. Here is the data above displayed through our graphics program:

Figure 11. Graphics mixture plot.

The most noticeable difference between these two plots (Figures 10 and 11) is readability. The contour levels are plotted directly on the contour lines in Figure 11 -- you don't have to refer to a legend.

USER INTERFACES

Now that we have all of these programs, how do users access them? The answer is user interface. The sophistication of the interface depends on the level of computer understanding expected of the user and on the level of detail required by the graphics application. All of the interfaces expect the user to understand the concepts behind the application, e.g. what a control chart is.

At the lower end of the spectrum are interfaces requiring users to have knowledge of the computer. For example, our trilinear program has been installed as a SAS procedure. This expects users to know how to execute SAS, how to
create SAS datasets, and how to run SAS procedures. It also expects users to know what graphics devices are available to them and how to direct plots to those devices. The following is a sample SAS program to generate trilinear plots:

```sas
*** SORT DATA BY TEST ***;
PROC SORT DATA=ASA.TRLDATA;
BY TEST;
RUN;

*** GENERATE TRILINEAR PLOTS WITH GROUPS PLOTTED WITH DIFFERENT SYMBOLS ***
PROC TRILIN DATA=ASA.TRLDATA DEVICE=ZETA;
10 TEST;
PARMCARDS;
PLOT RED * GREEN * BLUE = TEST /
STEP=2 MAXVAL=4;
TITLE SIMULATED DATA;
TITLE GRAPHICS TRILINEAR PROCEDURE;
RUN;
```

Figure 12. Sample trilinear plot program.

The user must know how to manipulate the data (e.g., how to sort it) and how to specify the necessary procedure options to get the desired results.

At the middle part of the spectrum are interfaces requiring users to have some knowledge of the computer. For example, the mixture plot program requires users to know how to create SAS datasets. They are also expected to know what graphics devices are available to them and how to direct plots to those devices. Users enter data and plot specifications via a SAS/FSP (4) menu:

```
MIXTURE SPECIFICATIONS
** To CONTINUE press ENTER followed by PF14 **
For multiple screens PF11(PF15) shifts to next screen
PF10(PF22) shifts to previous screen

Data set name: FRUIT.JUICE
(1) (2) (3) (4)
Component Names: ORANGE PAPAYA GRAPE
Minimum levels of Components: 0 0 0
Fixed level of component d: 0
Response Names: FLAVOR
Model Specification: (default: 2nd order model)

Plot Specifications:
Title 1: FRUIT JUICE EXPERIMENT
Title 2: (GRAPHICS)
Grid line labels: X1 Start X2 Increment X3 Start

Figure 13. Sample mixture program menu.
```

Error checking takes place once entries have been made on the screen. The program:
1. checks for required entries,
2. checks for validity of the SAS dataset, and
3. checks for validity of the components and responses on the SAS dataset.

From here, the responses are modeled and the DISEFRA plot program is run.

At the upper end of the spectrum are interfaces requiring little computer knowledge on the part of users. For example, the control charts described above are an application in a quality package developed at Kodak. This package requires users simply to know how to log on to the computer and how to enter the package. From there, they can enter data and perform whatever analysis is available in that package. They are expected to know what graphics devices are available to them, but the package takes care of directing the plots to the devices. The following are sample menus to generate control charts:

```
Kodak Quality Package

OPTION 2
1 Histogram
2 Control Charts
3 Statistical inference
4 Utilities
C CMS Commands or Execs
T Tutorial
X Exit

Figure 14. Sample quality package main menu.

This is the main menu of the package. The user selects an application on this menu -- in this case, the control charts.

KQP200A KQP CONTROL CHART

COMMAND =>

Dataset name => ASA . XBAR (library) (data)
Response Variable => MEAS2
Subgroup Variable => SUBGRP
Start subgroup number =>
End subgroup number =>
Plot Type => STD (Range or STD)

Graphics output: ZETA
Listing output: DISPLAY
Do you want to change these selections => NO

Figure 15.A Sample quality package control chart menu.

Figures 15.A and 15.B are the control chart menus. The user enters information specific to the application, e.g., the SAS dataset name, response variable, subgroup variable, and control limits and aims.
KQP Control Chart

**COMMAND**

Response variable: MEAS2

Optional Control Limits and Aims.

<table>
<thead>
<tr>
<th>MEANS</th>
<th>SIGMA/RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Limit</td>
<td></td>
</tr>
<tr>
<td>Aim</td>
<td></td>
</tr>
<tr>
<td>Lower Limit</td>
<td></td>
</tr>
</tbody>
</table>

**Title**

ONLY LONG TERM VARIABILITY

**Subtitle**

OUT OF CONTROL

Figure 15.B Sample quality package control chart menu.

Entries are checked for errors and the SAS code for the application is executed. Listing and graphics output is automatically directed to the devices specified on the application menu (Figure 15.A).

**SUMMARY**

Standard graphics packages provide tools to display data. Problems arise when an application does not fit a graphics package. With some effort, you can make a graphics package produce the plot you want. You must determine the feasibility of making that extra effort, e.g. the utility of the program and the costs involved.

The effort involved in getting the plot you want can be worth it. Cost and labor savings generated by converting a manual operation into a computerized one, getting a more informative display, conforming to accepted standards, or obtaining additional features are reasons. We have discussed several examples of plots fitting these categories written at Eastman Kodak Company.

User interfaces give users access to these graphics programs. Interfaces vary, depending on the level of computer understanding expected of users and on the level of detail required by the application. We have also discussed examples of user interfaces written at Kodak.

Making the effort, you can get the plot you want and have it accessible to users.

**ACKNOWLEDGEMENTS**

I would like to thank the statisticians/quality engineers who helped me prepare explanations of the applications shown: James L. Bossert, Stephen J. Caffrey, Charles E. Heckler, Margaret J. Krier, and Carl Wetzstein. I give particular thanks to the programmers who created the programs and interfaces: Franklin L. Becker, Julie A. LaBerr, and Richard D. Walvoord.

**FOOTNOTES**

1. SAS is a registered trademark of SAS Institute Inc., Cary, NC 27511-8000, USA. No endorsement is implied by Eastman Kodak Company.

2. DISSPLA is a registered trademark of Integrated Software Systems Corporation, San Diego, CA 92121-9990. No endorsement is implied by Eastman Kodak Company.

3. SAS/GRAPH is a registered trademark of SAS Institute Inc., Cary, NC 27511-8000, USA. No endorsement is implied by Eastman Kodak Company.

4. SAS/FSP is a registered trademark of SAS Institute Inc., Cary, NC 27511-8000, USA. No endorsement is implied by Eastman Kodak Company.

**REFERENCES**


