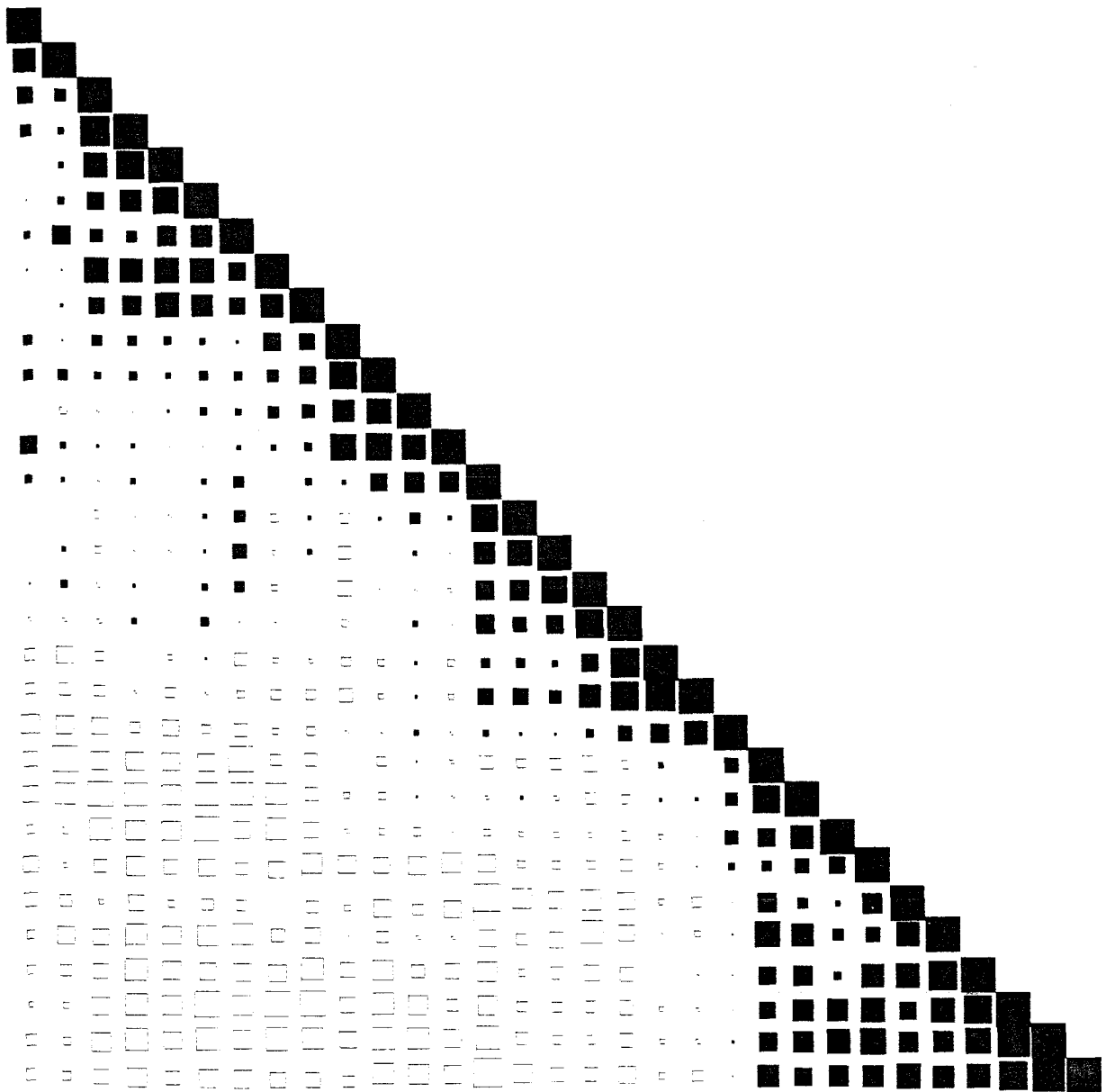


## Statistical Graphics With $\text{\TeX}$

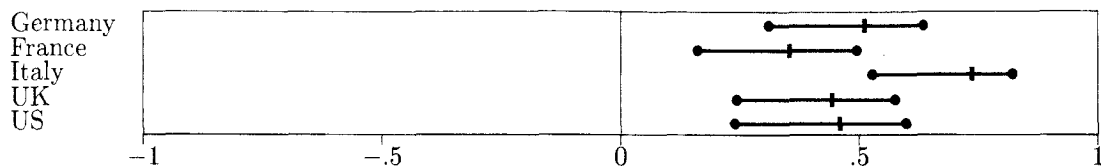
by Hans Ehrbar

$\text{\TeX}$  was not primarily designed for making graphical displays, and attempts to do graphics in  $\text{\TeX}$  might be considered somewhat a stopgap measure. Still, it is tempting to experiment with this possibility, because of the convenience of putting the graphics directly into the text.

There are two other, less obvious, advantages to using  $\text{\TeX}$  for graphics. First, the facility to make horizontal and vertical rules is well suited for certain kinds of graphics. The figure below is the printout of a correlation matrix (profit rates for two digit manufacturing industries 1949-79), in which the entries are replaced by squares, whose side lengths are proportional to the magnitudes of the entries. The solid squares represent positive, the bordered squares negative entries. This is a simple and attractive way of making the patterns in this correlation matrix visible.



The following figure, which shows confidence intervals, is another example for the use of rules for easy-to-read displays:



The small effort of having your statistics package (or computer program) give its output coded in  $\TeX$  has also a second payoff. The high precision of  $\TeX$  and its wide selection of fonts allow a lot of information to be compressed in a small space. Instead of reams and reams of computer printouts one can have all the information needed, together with some explanatory text, on a page or two. The scatter diagram on the following page of monthly data for capacity utilization (in percent, vertical axis) versus the inventory over sales ratio from February 1961 through November 1970 may serve as an example.

The syntax of the macros generating these displays is fairly simple, and should not provide any problems to someone familiar with the fundamentals of plain  $\TeX$ . In fact, the macros are modeled according to math mode of plain  $\TeX$ , i.e., they are variations of the alignment. The information necessary to produce these graphics (i.e., the coordinates of the scatter points, the endpoints and midpoints of the confidence intervals, the text to be written next to the scatter points, or the industry names in the confidence interval example) are arranged in a rectangular pattern, separated by  $\&$  and  $\backslash\text{cr}$ , which is given as an argument to a macro.

For the graphical representation of the correlation matrix, this is all one has to do. The effort is exactly the same as if one wanted to print the correlation coefficients in a matrix, but instead of  $\backslash\text{matrix}$  one has to give the keyword  $\backslash\text{dotpattern}$ , and instead of math mode one has to be in vertical mode (unless the display has only one line). Here is part of the actual code calling up the above dot-pattern:

---

This is a simple and attractive way of making the patterns  
in this correlation matrix visible.

```

\medskip
\dotpattern{100
\cr 66.5992 &100
\cr 47.8082 &35.9022 &100
\cr 33.5024 &19.9712 &84.9131 &100
\cr 1.1944 &16.7153 &69.9001 &80.3869 &100
:
\cr -33.5713 &-22.9446 &-54.7992 &-63.8956 &-50.1286 &-65.7807 &-54.1421 &
-66.1329 &-61.4736 &-48.1490 &-64.2737 &-61.5697 &-52.0634 &-67.4709 &-35.6489 &
-27.6953 &-28.1442 &-40.6490 &-14.9838 &-13.9312 &8.3953 &54.1430 &60.5092 &
69.6542 &75.2284 &58.9060 &60.2270 &72.9116 &93.6946 &100
\cr -33.7705 &-26.5308 &-41.9840 &-54.9173 &-39.2742 &-53.4861 &-53.4312 &
-45.0544 &-44.0052 &-32.6934 &-59.9733 &-57.6638 &-52.3735 &-83.7067 &-54.0942 &
-41.3554 &-41.7640 &-49.4753 &-20.3350 &-31.3480 &4.1582 &63.3049 &52.5102 &
60.0084 &61.9984 &71.4205 &63.6169 &67.9257 &82.3474 &93.3305 &100
\cr }
\bigskip

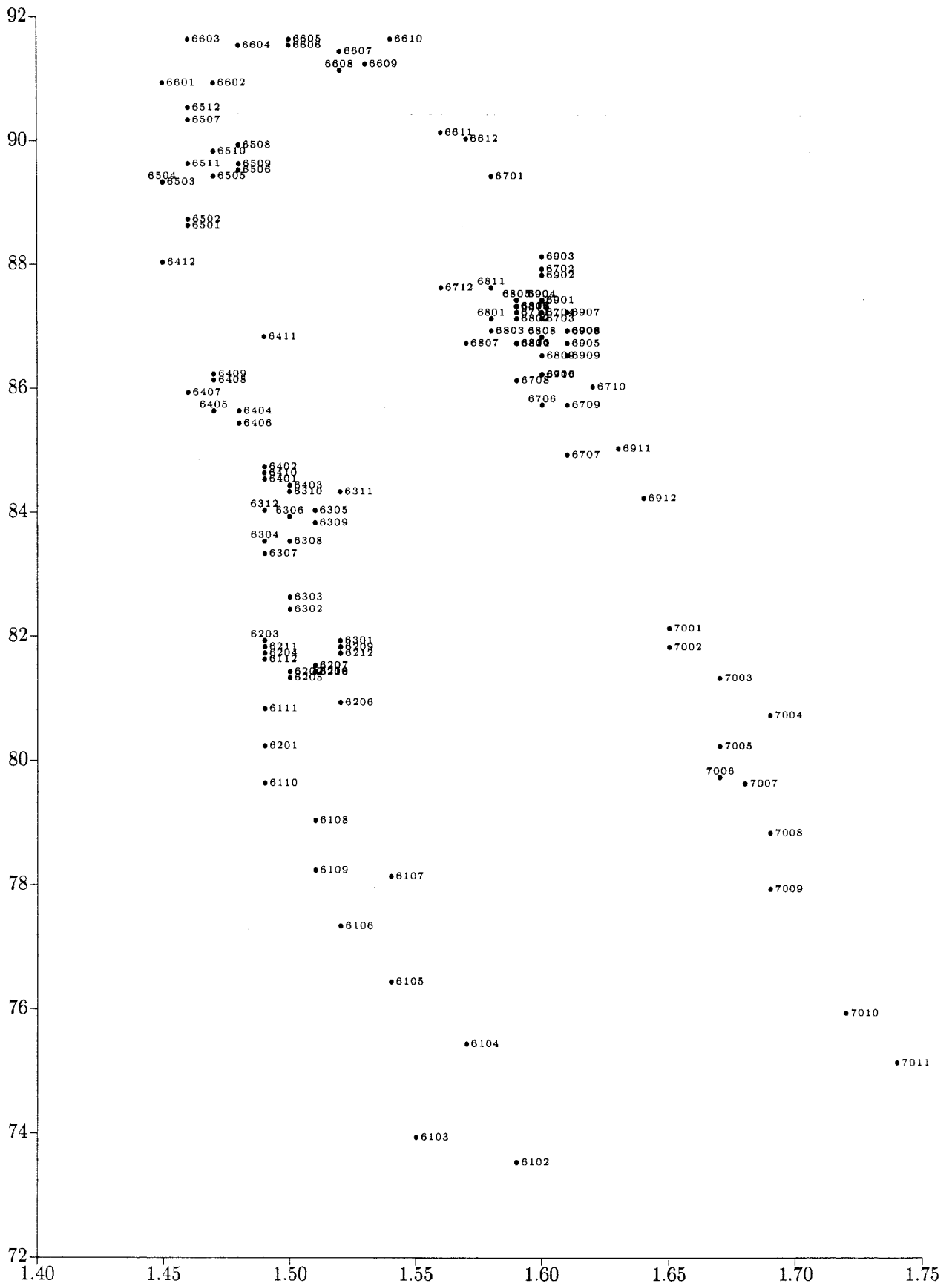
```

---

The following figure, which shows confidence intervals,

A little more has to be done for the graphs involving coordinate axes. The size of each graph is regulated by two dimensions,  $\backslash\text{hunit}$  and  $\backslash\text{vunit}$ , indicating the length of one unit on the horizontal and vertical axis. The scatter diagram needs a further detail: if the axes intersect at a different point than the origin, then one also has to assign to the dimensions  $\backslash\text{horigin}$  and  $\backslash\text{vorigin}$  values corresponding to the coordinates of the intersection of the axes.

The scatter diagram consists of three boxes joined together in horizontal mode: the first box contains the vertical axis, the second the scatter points (it has zero width), and the third box contains the horizontal



axis. The reference points of all three boxes is the “origin” of the diagram, where the two axes meet. Each of these boxes is produced by a macro, and all one has to do is to assign the appropriate values to `\hunit` and `\vunit`, and then call these three macros one after another in horizontal mode (e.g., inside a `\centerline`). Here one has to be careful that no blanks sneak in between these macros. The macros for the axes have as their first argument the distance between the ticks (measured in either `\hunits` or `\vunits`), and as the second argument an alignment with the text to be written next to (or below) the ticks. Here is the code invoking the scatter diagram:

---

```

\pageinsert
\vunit=1.1 cm \vorigin=72 \vunit
\hunit=45 cm \horigin=1.4\hunit
\centerline{%
\vertaxis{2}{92\cr90\cr88\cr86\cr84\cr82\cr80\cr78\cr76\cr74\cr72\cr}%
\scatter{73.50 &1.59 &6102
\cr 73.90 &1.55 &6103
\cr 75.40 &1.57 &6104
:
\cr 79.70 &1.67 &&7006
\cr 79.60 &1.68 &7007
\cr 78.80 &1.69 &7008
\cr 77.90 &1.69 &7009
\cr 75.90 &1.72 &7010
\cr 75.10 &1.74 &7011}%
\horaxis{.05}{1.40&1.45&1.50&1.55&1.60&1.65&1.70&1.75}}
\endinsert

```

---

The scatter diagram has an additional feature: if one wants the text for a given point not next to the point, but above the point, one has to write it in the fourth column instead the third column of the alignment (as was done with the June 1970 scatter point).

If one does not want the scatter diagram but only the axes, one simply leaves the second macro out, or one can replace it with a different type of graphics.

The diagram with the confidence intervals consists of two boxes joined together in horizontal mode: the first contains everything except the ticks and measurements units at the bottom, and its reference point is the lower left corner of the rectangle framing the confidence intervals, and the second is built by the macro for the horizontal axis which we know from the scatter diagram. (The first macro can be considered to be replacing the vertical axis).

The macro building the first box has two alignments as arguments. The first argument contains the coordinates of the three vertical axes of the diagram (which need not be equally spaced), and the second argument contains the text and the coordinates of the endpoints and centerpoints of the confidence intervals (which need not be equally spaced, either). Again, the code should be largely self-explanatory:

---

```

\smallskip
\hunit=2.5 in
\centerline{%
\confival{-1&0&1}
{Germany& 0.309022& 0.511198& 0.634136 \cr
France & 0.160441& 0.353459& 0.494577 \cr
Italy & 0.526767& 0.737672& 0.822247 \cr
UK & 0.242458& 0.443201& 0.575648 \cr
US & 0.237897& 0.459010& 0.597618}%
\horaxis{.5}{-1&-.5&0&.5&1}}
\smallskip

```

---

The macros are so easy to use that your statistics package has to learn only one thing: compute the information to be represented in the graph in matrix form, and then print the matrix into the output file in the familiar pattern of an alignment.

The macros themselves are moderately complicated. The few tricks they use can be combined for also writing different graphical display macros. They are collected in the macro file `graphmac.tex` which is to be read in together with plain `TEX`. Arrangements will be made for TUG to distribute this file; the interested reader should call the TUG office for details.

Basically, these macros use (mis-use?) the alignment primitive for passing large and variable numbers of arguments to a macro. Often the entries in these alignments are not used for setting boxes, but only for assigning values to dimension registers. The `\dotpattern` macro could have been written in a way which follows more closely the spirit of an alignment; but in order to prevent box memory and macro memory overflow, the alignment only writes an auxiliary file `scratch.tex`, which is then read in to create the actual output. Besides writing the output file, `\halign` also creates a large number of empty boxes. In order to prevent these boxes from being appended to the main list (and thus wasting box memory), I used the `\setbox` command before invoking `\halign`, and afterwards emptied the box again.

When using these macros, it turned out that the size limitations of `TEX` (both macro memory when the alignment is read in, and then box memory when it is executed) are still rather stringent. I hope that larger versions of `TEX` will become available in the future—once everyone discovers how much can be done with `TEX`, constructions like the ones described in this note will no longer be the exception.

Economics Department  
University of Utah  
Salt Lake City, UT 84112