P-STAT®

Plots, Graphs and PostScript Support

P-STAT®

P-STAT: PLOTS, GRAPHS and PostScript Support

Second edition: January 2013

This publication corresponds to **P-STAT Version 3.01, January 2013**. This publication is designed to document the P-STAT graphics commands and the use of PostScript for camera ready output.

Please direct any questions to:

P-STAT, Inc.
230 Lambertville-Hopewell Rd.
Hopewell, New Jersey 08525-2809
U.S.A.

Telephone: **609-466-9200**

Fax: **609-466-1688**

Internet: **support@pstat.com**

Web Page URL: **http://www.pstat.com**

CONTENTS

Introduction

The PostScript Commands

PLOT: Multivariable Scatter Plots

PLOT: Enhancing the Output

PLOT: PostScript Support

BOXPLOT: A PostScript Command

HIST: Character Histograms

LIST OF FIGURES

1
Introduction

This manual contains commands which are used to “draw” plots and graphs. The only output types supported in P-STAT are:

1. Character based output
2. PostScript Output

Character based output is designed to give the researcher information in a form that does not require any special devices for printing. It can be seen or printed on a dumb terminal or a simple ASCII printer. It needs no special information about formatting, fonts or color.

PostScript Output requires an output devise that can interpret the PostScript controls. PostScript output is available on most laser printers although some printers may require special software or additional hardware. There are also programs that can take PostScript output and display it on a terminal. Ghostview is one such program that is available for free over the internet.

The organization of this manual is:

1. This introduction
2. An overview of PostScript and the commands that it supports
3. The basic capabilities of the PLOT command
4. Enhancing PLOT output with titles and labels.
5. Using PostScript for camera ready PLOT output.
6. The BOXPLOT command
7. HIST, the character output histogram command.

In P-STAT PostScript output can be used with programs like SURVEY and TEXTWRITER as well as graphics programs like PLOT and BOXPLOT. There are several P-STAT commands to setup and control the PostScript Environment. Because PostScript support is provided for both text and graphics commands it is the first chapter in this manual so that it will be readily available.

The PLOT command is a major capability and produces its output in both formats. The first of the PLOT chapters covers the basic PLOT identifiers and subcommands which are used whatever the desired output type.. The figures in this chapter are all character based. Examples later in the manual show the camera-ready PLOT output that is produced with the addition of PostScript controls.

Character based box plots are a part of the EDA command which is included in the manual “P-STAT basic statistics”. The BOXPLOT command covered in this manual produces PostScript output only. The final chapter covers the HIST command which produces histograms as character output.

2
The PostScript Commands

PostScript is a page description language which provides a device independent representation for a printed page. It is widely available and can be used on any print device (for example, a laser printer or even a terminal) that has a PostScript interpreter. PostScript is the language that P-STAT uses for producing “camera-ready” printout from SURVEY, TEXT.WRITER, PLOT and BOXPLOT.

2.1 POSTSCRIPT OVERVIEW

The PostScript operators are not strange looking codes but are words such as “show”, “showpage”, “moveto”, and “lineto”. The PostScript interpreter is a program, usually in the controller of the printer, which translates these words into the myriad complex actions that are required to produce “camera-ready” copy. If a file begins with the characters “%!PS-Adobe-”, it is recognized as a PostScript file by the interpreter. This file can be sent to any printer, but if the PostScript interpreter is not installed, you see the entire stream of instructions intermixed with your output rather than the nicely formatted printout that you expect.

The P-STAT commands which support PostScript can be used by themselves. In the simplest case PostScript is invoked by adding the identifier POSTSCRIPT to the command and directing the output to a printfile.

```
PLOT   Myfile, POSTSCRIPT, PR 'Plot.ps';
SURVEY Myfile, POSTSCRIPT, PR 'Surv.ps';
TEXT.WRITER Myfile [ PUT 'This is my PostScript run' ],
     POSTSCRIPT, PR 'Text.ps';
```

These commands can also be used within a block of P-STAT commands bounded by POSTSCRIPT $ and POSTSCRIPT.CLOSE $. The PostScript block is used to produce one or more pages with the output coming from the P-STAT commands within the block. This makes it possible to annotate a PLOT with TEXT.WRITER, place multiple plots on a single page, or combine a plot and a crosstabulation with some explanatory text.

The P-STAT commands which use PostScript can be divided into two distinctly different categories.

1. In line oriented output the number of lines that can be printed on a page is a function of the area available and the point size of the fonts that are used.
2. In graphic output the scale of the drawing depends only on the area available.

It is easier to combine multiple graphics into a single page than multiple surveys or listings. If there are to be six plots on a page, each plot receives 1/6th of the space and is scaled accordingly. If you wish to place six surveys on a page, you must do some of the calculations yourself to determine the proper font and SURVEY layout so that all six will fit without overlap.

Each of the supported commands has somewhat different needs and the parameters that can be controlled are documented separately for each of them.

The remainder of this chapter describes the POSTSCRIPT, POSTSCRIPT.CLOSE, POSTSCRIPT.SETUP, and POSTSCRIPT.RESET commands which are used to:

1. set fonts, colors, margins, and the orientation for the commands which follow:
2. print the start-up instructions which are used by the PostScript interpreter:
3. complete a PostScript session and to send the output to a printer:
4. control the layout of a single page that will have the results of one or more P-STAT commands:

5. produce encapsulated (EPS) PostScript output:

6. print boxes, symbols, lines and text using POSTSCRIPT subcommands.

2.2 SETTING POSTSCRIPT PARAMETERS

The POSTSCRIPT.SETUP and POSTSCRIPT.RESET commands are used to specify parameters for possible use in subsequent P-STAT commands. A printfile may not be specified and nothing is written to an output file. The POSTSCRIPT.RESET command is used to reset the fonts and margins to the values that are assigned at the beginning of the P-STAT session. The POSTSCRIPT.SETUP command is used to supply new values which become the default values for any subsequent commands which produce PostScript output.

POSTSCRIPT.RESET $ has no identifiers. The parameters, which are discussed in detail later in the chapter, are reset as follows:

1. Fonts are set to Times-Roman 8 point.

2. Colors for fonts and border are set to black.

3. Margins are set to 1 inch.

4. The leading is set to 1/72 of an inch.

5. The orientation is set to landscape

6. The paper size is set to 8.5 inches high and 11 inches wide.

Either the POSTSCRIPT.SETUP or the POSTSCRIPT command may be used to change the default values for these parameters. For example:

```
POSTSCRIPT.SETUP, MARGINS 2 $
```

provides two inch margins for all subsequent PostScript output. If the PLOT command changes a margin, that change is in effect only for that PLOT command, the default setting remains 2 inches until it is changed by a subsequent POSTSCRIPT.SETUP, POSTSCRIPT.RESET, or POSTSCRIPT command. All the parameters except font definitions are temporary when changed outside of the “POSTSCRIPT” commands. Changes in font definitions are made whenever they are encountered.

2.3 Fonts

In most of the P-STAT applications there may be four fonts defined at any one time and the active font depends on what is being printed in that application. For example in all of the applications except TEXT.WRITER, the first font (see below) is used for the central part of the application, the second font is used for labels, the third font is used for titles and the fourth font is application dependent. Because the font is associated with a particular item in the printout, the application itself issues the instructions to set the current active font.

You need not stay with the same four fonts throughout the run. In the commands with a subcommand language you can change the available fonts at any time. When the fonts are set as identifiers you can change the available fonts whenever you begin a new command. In addition you can use the extra fonts to control the appearance of individual titles lines.

TEXT.WRITER is an exception to this usage. Since TEXT.WRITER has no distinct parts, you change the active font whenever you want the change. If no specific font is selected, TEXT.WRITER uses the defined FONT1 until a font request is processed. Any or all of the fonts from FONT1 through FONT29 can be used in textwriter.

The font names FONT1 through FONT4 are recognized in all the commands and subcommands. The aliases used in the following list are also supported so that existing SURVEY and PLOT runs will continue to work without change.

1. FONT1 the basic font used for the body of a SURVEY or PLOT (including the tick labels). BFONT is an alias.
2. FONT2 used for labels in PLOT, BOX.PLOT, and SURVEY. LFONT is an alias.
3. FONT3 used for titles. TFONT is an alias.
4. FONT4 used for the XL (QUESTION) in SURVEY and for the legend in PLOT. QFONT is an alias
5. FONT5 to FONT29 available for use in TEXT.WRITER and to define extra fonts for titles in the other commands.

The point size of a font determines the size of the letters. Point size is measured in units that are 1/72 of an inch. Thus a point size of 6 is quite small and a point size of 24 is quite large. The point size must be between 4 and 432. A point size of 4 requires very sharp eyesight. A point size of 432 produces letters that are 6 inches high.

If the fonts are not explicitly set, the TIMES-ROMAN 8 point font is used for all twenty-nine font settings. If a font is specified but no point size is provided, the point size depends on the current output width of the print destination. If the output width is less than or equal to 80, the point size is 10. If it is between 80 and 132, the point size is 8. If the output width is greater than 132, a point-size of 7 is used.

Any font that is available on your printer can be selected. However, TIMES, COURIER, and ARIAL fonts, which are supported on all PostScript devices, can be selected by using key words. The following example illustrates how the four fonts are set in the POSTSCRIPT.SETUP command. The same language is supported in all commands which provide for PostScript output.

```
POSTSCRIPT.SETUP,
    FONT1 COURIER,
    FONT2 ARIAL BOLD 10,
    FONT3 TIMES BOLD 14,
    FONT4 ARIAL BOLDITALIC 10 $
```

The possible combinations for these fonts are:

```
TIMES               ARIAL               COURIER
TIMES BOLD          ARIAL BOLD          COURIER BOLD
TIMES ITALIC        ARIAL ITALIC        COURIER OBLIQUE
TIMES BOLDITALIC    ARIAL BOLDITALIC    COURIER BOLDOBLIQUE
```

You may leave a blank between BOLD and ITALIC or OBLIQUE. If you give an incorrect combination such as TIMES BOLDOBLIQUE, you will get an error message.

Fonts are tricky things to use. When titles are in a large font such as Times-Bold 14, each letter takes up more space in all directions. Having differing heights for different lines makes an interesting problem in counting lines per page. However, a bigger problem occurs when two fonts with different point sizes are used on the same line. This can cause the letters from the larger font to overwrite part of the letters on an adjacent line. For the purposes of line counting, the FONT1 point size is usually the determining point size when there is such a conflict.

If you wish to use a font other than the 12 combinations above, you must put the complete font name in quotes.

```
POSTSCRIPT.SETUP,   FONT 'Zapf-Chancery-Medium-Italic' 8 $
```

In this example using the FONT identifier instead of specifying a specific font such as FONT2, all twenty-nine possible fonts are set to the specified font. If the name is incorrectly spelled, a monospace font such as Courier is usually supplied. All letters in a monospace font are the same width. Each letter in a proportional font such as Times or Arial uses the appropriate space for each letter. A “I” takes less space than a “W”. It is easier to line up columns in a monospace font.

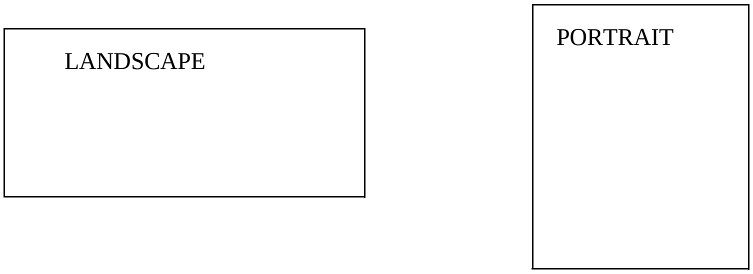
2.4 Leading

Leading refers to the space between lines. In the P-STAT PostScript commands the leading is set to 1, which is 1/72 of an inch. This amount is added to the point size in computing the line spacing. The leading can only be changed in the POSTSCRIPT.SETUP or POSTSCRIPT commands.

```
POSTSCRIPT.SETUP,  LEADING 2,
```

2.5 Orientation

The assumption in P-STAT is that PostScript output is in landscape orientation on 8.5 by 11 inch paper. In landscape orientation 8.5 is the height and 11 is the width. The identifiers PORTRAIT and LANDSCAPE are used to specify or change the orientation. In PORTRAIT orientation 8.5 is the width and 11 is the height.



```
POSTSCRIPT.SETUP, PORTRAIT,
```

When a series of commands are used to create a composite page, changing the orientation will produce strange results.

2.6 Margins and Size

The four margins and the paper size can also be specified. The identifiers are MARGINS, TOP.EDGE, BOTTOM.EDGE, LEFT.EDGE, RIGHT.EDGE and PAPER.SIZE

```
POSTSCRIPT.SETUP,   TOP.EDGE 1.5, LEFT.EDGE 3.,
POSTSCRIPT.SETUP,   MARGINS  1.5  2  1.5  2,
```

MARGINS is used to supply all four edges: top, bottom, left and right, in that order. If there is a single argument, that number is used for the margins on all four edges.

```
POSTSCRIPT.SETUP, MARGINS 2,
```

If there is more than one argument, there must be 4 arguments, 1 for each edge.

In most applications TOP.EDGE and LEFT.EDGE are used in conjunction with the paper size to determine the size of graphics output. TOP.EDGE and LEFT.EDGE are used to determine the starting position for line oriented output. The edges are set to 1 inch initially. The paper size is assumed to be the American standard of 8.5 by 11.

```
POSTSCRIPT.SETUP,  PAPER.SIZE 11 8,  PORTRAIT,
```

The first argument following PAPER.SIZE is the height, the second is the paper width.

If the margins and size parameters are specified in the POSTSCRIPT.SETUP or POST.SCRIPT command, they will become the assumed parameters in any P-STAT commands that produce PostScript output until they are changed or reset by one of the “POSTSCRIPT” commands. If these parameters are set in commands such as PLOT and SURVEY they are only in effect for that command. This is true for all the settings except fonts, which become available when they are defined and remain the available fonts until they are redefined in some other command.

2.7 Title Fonts

The font for each line of the titles can be individually controlled. Usually the font associated with FONT3 is used for all titles lines. This font, which in commands such as SURVEY is also referred to as TFONT, is the 3rd font of the 29 fonts that can be defined in the POSTSCRIPT and POSTSCRIPT.SETUP commands. In these commands the TITLES identifier is used to associate a specific titles line with a particular font name.

As many as

9 different font styles and sizes can be provided for the titles. However, all the sections of a given title line (left, center and right) must have the same font and point size.

```
    POSTSCRIPT.SETUP,  FONT5 TIMES BOLD 14,
                       TITLE TOP 1 5 $
```

The TITLES identifier has 3 required arguments. The first argument is either “TOP” or “BOTTOM”. The second argument is a number, 1-9 for top titles and 1-3 for bottom titles. The final argument is a number between 1 and 9 indicating the font (FONT1 - FONT9) that is to be used for that titles line. Both the FONTn and the TITLE identifiers must be used. It is advisable to use only FONT3 and FONT5 - FONT9 when defining titles as the other 3 fonts are associated in commands such as SURVEY with BFONT (FONT1), LFONT (FONT2), and QFONT (FONT4). This is not a problem if, for example, you wish to use the same font for a title line as the font that is to be used for the body of the table (BFONT).

Each font must be entered individually. The names are FONT1 - FONT9. The titles information can be entered as individual TITLE identifiers or as a single TITLES identifier with many arguments. The following example provides a very large font for the top title in FONT5 and a very small font for the first bottom title in FONT6.

```
    POSTSCRIPT.SETUP,  FONT5 TIMES BOLD 14,  FONT6 TIMES 6,
                       TITLE  TOP 1 5  BOTTOM 1 6 $
```

These fonts must be defined before the command that will use them. When a title that has a special font is printed in the commands which support both TITLES and PostScript, that font is used instead of the font associated with TFONT (FONT3). It is not necessary to skip lines if you have a font with a large point size. The next title will be spaced appropriately. However, if you are using very large fonts for your titles, you may need to reset the number of lines on a page to avoid printing beyond the end of the page.

2.8 Color for Fonts and Borders

If you have a color PostScript printer, you can define colors for each of the fonts and the border. This is done by using the COLOR identifer in the POSTSCRIPT.SETUP commnad.

```
    POSTSCRIPT.SETUP,  FONT5 TIMES BOLD 14,
                       COLOR FONT5 RED
                       FONT6 TIMES 6,
                       COLOR FONT6 GREEN,
                       TITLE  TOP 1 5  BOTTOM 1 6 $
```

The color identifier requires 2 arguments; the name of the font or with word BORDER and the color to be used. The only colors supported with keywords are RED, ORANGE, YELLOW, GREEN, BLUE, VIOLET, BROWN, BLACK, WHITE, CYAN and MAGENTA. See the section later in this chapter on using color in the POSTSCRIPT command for ways to customize the colors.

2.9 Other Settings

The PostScript command assumes that the printout is to be in English. The CHARACTER.SET identifier currently has two options:

```
    CHARACTER.SET ISO
    CHARACTER.SET GERMAN
```

It can be abbreviated to CH.SET. If ISO is specified many of the characters from the ISO 8859-1 standard are available. GERMAN provides just the basic characters needed for the GERMAN language. It is the equivalent of the earlier identifier UMLAUTE.ANSI.

The ENCAPSULATE identifier specifies that the output is intended for use in some other text processing program. The EPS, Encapsulated PostScript format is discussed in detail at the end of the chapter.

The assumption is that a new printed page is to be produced whenever a normal page break occurs in the P-STAT output. This behavior is controlled by the SHOWPAGE (assumed) and NO SHOWPAGE identifiers. Control of pagination is required if the output from multiple P-STAT commands is to be combined on a single page. SHOWPAGE is discussed in detail in the section on “Printing the Output”.

```
POSTSCRIPT.SETUP, CH.SET ISO,  ENCAPSULATE,  NO SHOWPAGE $
```

The DELTA identifier requires a numeric argument which is used when the PostScript code tries to find the best location for a line break. Particularly when justification is NOT turned on, the computed value is too large and causes a word to be broken after 1 or 2 characters. Setting Delta to a small number may correct this problem.

```
POSTSCRIPT.SETUP,  DELTA 10 $
TEXTWRITER xxx, ......      $
```

2.10 A POSTSCRIPT BLOCK

A PostScript block contains a series of commands to be formatted using similar PostScript parameters *with a single print destination*. A PostScript block begins with a POSTSCRIPT command containing a PR identifier to specify the print file. A PostScript block ends with a POSTSCRIPT.CLOSE command. The print destination cannot be changed within the PostScript block. There may be any number of PLOT, SURVEY, TEXT.WRITER or POSTSCRIPT commands between the starting and ending commands.

The POSTSCRIPT command has some of the same identifiers as thePOSTSCRIPT.SETUP command plus the ability to begin a PostScript block, print a complete page, begin a new page, customize colors, or put text and a few simple drawing elements on a page. POSTSCRIPT.SETUP is used to initialize internal values that will be required by PLOT, SURVEY, or TEXT.WRITER when they are used outside of a PostScript block. POSTSCRIPT.SETUP may also be used before the POSTSCRIPT command that begins a PostScript block.

When a PostScript block begins, the print destination is remembered. All subsequent commands that support PostScript are treated as part of the PostScript session and the output is automatically sent to the appropriate destination.

```
POSTSCRIPT, PR 'Test.ps' $
PLOT Myfile;
P Year by Income;
$
SURVEY Myfile;
Banner Year, STUB Region, MEANS Income;
$
POSTSCRIPT.CLOSE, PRINT $
```

Often each command will produce one or more full pages of output. However, in a PostScript block this can be controlled so that any given page can have output from multiple commands. For example a page of output might consist of a small plot at the top of the page, followed by TEXT.WRITER annotations and a different plot with data from a second P-STAT system file at the bottom The page might be bordered with a black outline and have an overall title in addition to any titles associated with the plots.

You cannot start a new PostScript block until you have finished the current PostScript block. POSTSCRIPT.CLOSE finishes the use of the current print file. If you use the print file again, you will be told that the file exists and asked if it should be written upon. A yes reply will cause the file to be overwritten. PostScript parameters are not reset by the POSTSCRIPT.CLOSE command. A new PostScript block will begin with the same parameters as the previous block unless they are changed with POSTSCRIPT.RESET or POSTSCRIPT.SETUP.

The only identifier that can be used with the POSTSCRIPT.CLOSE command is PRINT. The sequence:

```
POSTSCRIPT, PR 'Test.ps' $
POSTSCRIPT.CLOSE, PRINT $
```

produces a blank sheet of paper.

2.11 Printing the Output

In the discussion of printing PostScript output there are two issues.

1. Defining the end of a single page
2. Sending the page or pages to the real printer.

Producing a page in a language such as PostScript is very different from typing it on a typewriter. The page is like a tablet. You can move any where on the page, overwrite, draw or scribble as you wish. There is no concept of a last line. Therefore, it is possible to place several plots or listings on a single page and it is necessary to have a means of defining a completed page.

The PostScript word that defines the end of the page is “showpage”. When a command such as PLOT and SURVEY is executing, the command itself usually writes the “showpage” at the end of each page of plot or survey printout. If you wish to place output from several commands on a single page, you must be able to control when the showpage is issued.

The identifiers SHOWPAGE and NO SHOWPAGE can be used in any of the PostScript supporting commands. When SHOWPAGE, the default, is in effect, the individual command writes “showpage” at the normal place. When NO SHOWPAGE is in effect, you must see that SHOWPAGE is written. This can be done by:

1. Setting SHOWPAGE in a subsequent command.
2. Using PUT.PAGE in the POSTSCRIPT command.
3. Using POSTSCRIPT.CLOSE to end the POSTSCRIPT block.

The following sequence places two plots on a single page:

```
POSTSCRIPT, PR Plot.ps, NO SHOWPAGE, PORTRAIT $

PLOT Region1;
PLOTS.PER.PAGE 2 by 1, P Year * Income;
$
PLOT Region2;
SHOWPAGE, PLOTS.PER.PAGE 2 by 1, TOP.EDGE 5., P Year * Income;
$
POSTSCRIPT.CLOSE $
```

In this example the SHOWPAGE is not necessary in the final PLOT command because the POSTSCRIPT.CLOSE command automatically issues a showpage when there are any unprinted pages.

The PRINT identifier in POSTSCRIPT and POSTSCRIPT.CLOSE can only be used if you are running on a system which supports direct printing to the PostScript printer with the PRINT command.

```
POSTSCRIPT, PRINT $
```

If the PRINT command is not supported the same effect may be obtained by using both the CLOSE and SYSTEM commands.

```
CLOSE 'Test.ps'
SYSTEM 'local print information Test.ps'
```

You must supply the name of the command that prints a file in your particular computing environment along with any other parameters that are needed.

Printing can be done at any time during the PostScript session.

```
POSTSCRIPT, STARTUP, PR 'Test.ps' $
PLOT .....
POSTSCRIPT, PRINT $           or   CLOSE 'Test.ps'
                                   CALL SYSTEM '......Test.ps' $
SURVEY .....
POSTSCRIPT.CLOSE,  PRINT $     or  POSTSCRIPT.CLOSE  $
                                   CLOSE 'Test.ps' %
                                   CALL SYSTEM '......Test.ps' $
```

2.12 Borders and Titles

The POSTSCRIPT command can be used to outline a page and to place titles on a page. This page may then be the background for other P-STAT printout.

```
TITLES '.DATE. $
POSTSCRIPT, PR 'Test.ps', BORDER, TITLES $
```

BORDER and NO BORDER can be used as identifiers in the POSTSCRIPT command. The border is usually placed 1/4 of an inch outside the margins. The line itself is assumed to be a solid line that is 1/144th of an inch wide. Borders are drawn on the output page when the PostScript block begins (the first page) and whenever a new page is requested. Border type, width, and placement can be changed in the subcommand language.

Top titles are placed below the current top margin. Bottom titles are placed above the current bottom margin. You may use any point size for the titles and the spacing between the title lines will be appropriate. There is no need to leave blank titles between title lines to prevent the lines from overlapping. Font3 is always used as the titles font unless you have set up titles fonts in the POSTSCRIPT.SETUP command. FONT3 is the only one of the 9 available fonts that has any special meaning in the POSTSCRIPT command.

```
TITLES '.DATE.' $
POSTSCRIPT, PR 'Test.ps', TITLES,
            FONT3 ARIAL BOLDITALIC 14,
```

If TITLES or borders are requested in the PostScript block, they are printed on every page controlled by the POSTSCRIPT command. In the following sequence, there are two pages of printout both with a border and titles.

```
TITLES '.DATE.' $
POSTSCRIPT, PR 'Test.ps', TITLES, BORDER $
PLOT Myfile;
  plot subcommands; $
POSTSCRIPT, NEW.PAGE $
SURVEY Myfile;
  survey subcommands; $
POSTSCRIPT.CLOSE $
```

2.13 DESIGNING A PAGE

There are a number of identifiers in the POSTSCRIPT command and in the other commands that produce PostScript output which are useful in the design of individual pages of printout. These commands, like all P-STAT commands may be used in MACROS and SUBFILES reducing the effort needed when individual pages are to be repeated for different subsets or values in a file.

The most important capability, controlling showpage, has already been discussed. This is supported in all the commands that have PostScript support. Margins are used to control the placement of the printout. The graphics commands have options to specify the plot size. In the line oriented commands, the lines and output width identifiers can be used to control the amount of the printout.

The POSTSCRIPT identifiers PUT.PAGE and NEW.PAGE are also available for to help with pagination. PUT.PAGE writes the page ending “showpage” line to the PostScript output file. NEW.PAGE writes any titles and draws borders on the PostScript output file. The page is effectively cleared by a PUT.PAGE. It is not cleared by a NEW.PAGE which takes advantage of the fact that all of the page is always accessible and writes top and bottom titles at the same time.

When POSTSCRIPT is used with the PR identifier to begin a PostScript block, NEW.PAGE is implied and titles and borders are written out if they have been requested. In subsequent POSTSCRIPT commands, NEW.PAGE must be used to indicate the start of a new page and to repeat the titles and borders on that page.

```
TITLES '.DATE.' $
POSTSCRIPT,  PR 'Test.pks', BORDER, TITLES, NO SHOWPAGE $
SURVEY Myfile;
STUB Age, BANNER Education;
$
POSTSCRIPT, PUT.PAGE, NEW.PAGE $
SURVEY Myfile;
STUB Q33, BANNER Income.Group;
$
POSTSCRIPT.CLOSE $
```

In this example, NO SHOWPAGE means that the contents of the page are under user control. The output from the SURVEY command has no “showpage” to define its end until the POST.SCRIPT command containing the PUT.PAGE is processed. The NEW.PAGE in that command causes the border and titles to be drawn on the second page. The POSTSCRIPT.CLOSE command causes the “showpage” to be written and the print file to be closed.

2.14 POSTSCRIPT SUBCOMMANDS

Subcommand records are not expected if the POSTSCRIPT command ends with a “$” and, if they are there, they will not be processed. Subcommands are expected if the POSTSCRIPT command ends with a “;” . The POSTSCRIPT subcommands provide a limited set of instructions which are intended to allow you to annotate the output from other P-STAT commands. It is also possible by using the subcommands, to construct a cover page, a title page or a table of contents. Borders, boxes and lines can be drawn and their attributes controlled with subcommands.

The POSTSCRIPT command does not access any data files and it has no knowledge of any values in the files. The only values that are available are those that can be passed into titles using system variables or permanent scratch variables. The drawing capabilities are limited to borders, boxes, lines, and the characters that are available for plot symbols (circles, stars, squares, etc.).

The capabilities include:

1. defining and setting fonts
2. customizing font colors
3. setting margins or individual edges
4. start or finish a page
5. moving to a specific location specified in inches from the top-left corner of the page
6. moving to the next line or skipping s number of lines
7. drawing borders, boxes and lines and changing their attributes.

8. defining top and bottom titles
9. printing titles, symbols and strings

2.15 General Subcommands:

Many of the identifiers used in the POSTSCRIPT command are available as subcommands. The following subcommands can be used exactly as they are used as identifiers:

1. FONT, FONT1 through FONT9
2. MARGINS
3. TOP.EDGE, BOTTOM EDGE, LEFT.EDGE and RIGHT.EDGE
4. PAPER.SIZE
5. PUT.PAGE and NEW.PAGE
6. BORDERS and TITLES

There are some extra capabilities available in the subcommand language. For example title text may be supplied and the line type for the border may be defined as a pattern of dashes and dots rather than a solid line. The following example illustrates the use of the POSTSCRIPT subcommand language to set up a page in which the titles are written in different fonts and the border consists of wide dashes.

```
TITLES '.DATE.' $
POSTSCRIPT, PR 'Test.ps';

BORDER,  LINE.TYPE BORDER DASHED,  LINE.WIDTH BORDER 2.,
FONT3 TIMES BOLD 14,
COLOR FONT3 RED,
NEW.PAGE,
FONT3 TIMES BOLD 10,
TITLES T3 'The second title has a smaller font',
NEW.PAGE $
```

Because NEW.PAGE does not erase the current contents of a page but serves only as a vehicle for page initialization, including borders and titles, it is possible to use it several times on a single page to achieve unusual effects. The following sequence creates three borders, one inside the other, with a title inside the innermost border. A page will print for each state found in file Myfile.

```
MACRO Border $
SUBFILES Myfile, BY State,

POSTSCRIPT, PR 'Test.PS';
BORDER 0, LINE.TYPE BORDER SOLID,  LINE.WIDTH BORDER 2,
NEW.PAGE;
MARGINS 1.5,
LINE.TYPE BORDER DASH, LINE.WIDTH BORDER 1.3,
NEWPAGE,
TITLES '##state',
MARGINS 1,
LINE.TYPE BORDER DOTS, LINE.WIDTH BORDER .5,
NEWPAGE, PUT.PAGE $

ENDSUBFILES $
ENDMACRO
RUN Border $
POSTSCRIPT.CLOSE, PRINT $
```

2.16 Specifying Location.

When MARGINS, BOXES, or BORDERS are drawn in the subcommands, the location of the X/Y coordinates is positioned 1/4th of an inch inside the top left margin, box, or border. The next action occurs at that location unless you specifically move the coordinates with the MOVETO, RMOVETO, NEXT, or SKIP subcommands.

```
MOVETO 3. 4.5
```

will move the coordinates to a location 3 inches from the left edge or the page and 4.5 inches from the top of the page. RMOVETO is a move relative to the current location.

```
RMOVETO 1 1,     move 1 inch to the right and 1 inch down
```

NEXT is used to move down 1 line. The actual amount that the Y coordinate (the distance down the page) changes depends on the point size of the current font. The distance moved is the point size plus the leading. The location of the X coordinate (the location across the page) is either the current margin or the current column location.

```
NEXT,                          is the same as
NEXT COLUMN,
NEXT MARGIN                    moves down and back to the left margin
```

NEXT can also be used with a number to indicate how many times NEXT should be repeated. SKIP can be used to skip multiple lines. SKIP 1 is the same as NEXT, NEXT or NEXT 2. The control over the X coordinate is the same for SKIP as it is for NEXT.

```
SKIP 3                         is the same as
SKIP COLUMN 3
SKIP MARGIN 3                  leaves 3 blank lines,
                               moves to left margin.
```

2.17 Drawing Borders, Boxes, and Lines

The line width and line type for the border, for boxes and for lines can be changed in POSTSCRIPT subcommands. If the border attributes are changed in a subcommand the changes are remembered and used in subsequent POSTSCRIPT calls. The subcommands LINE.WIDTH and LINE.TYPE are followed by one of BORDER, BOX, or LINE and the appropriate argument.

```
LINE.TYPE BORDER DASHED,  LINE.WIDTH BORDER 2,
```

The available line types are:

```
DOTTED      DOT.DASH      SHORT.DASH    DASH
LONG.DASH   ODD.DASH      SOLID
```

The LINETO subcommand requires two arguments; the distance in inches from the left edge and the distance in inches from the top edge. The line is drawn from the current location to the specified coordinates.

```
MOVETO 2 2,  LINETO 4 4,
```

draws a diagonal line from the 2/2 coordinate to the 4/4 coordinate.

```
MOVETO 2 2,  LINETO 4 2,  LINETO 4 4,  LINETO 4 2,
```

draws a triangle.

The BOX subcommand requires 4 arguments. The first two arguments are the coordinates in inches of the upper left corner. The last two arguments are the coordinates in inches of the lower right corner. In each case the first number is the distance from the left edge, the X coordinate, and the second is the distance from the top edge, the Y coordinate.

The following example illustrates the use of line width and line type for boxes, for borders, and for lines.

```
POSTSCRIPT, PR 'Test.ps' $
MARGINS 2, LINE.TYPE BORDER DASH,        LINE.WIDTH BORDER 1.5,
           LINE.TYPE BOX    DOTTED,      LINE.WIDTH BOX    1,
           LINE.TYPE LINE   SOLID,       LINE.WIDTH LINE   .1,
BORDER, NEW.PAGE,
BOX 2.5 2.5 4.5 4.5,   LINETO 4.25  4.25,  PUT.PAGE $
```

The current location of the coordinates after either box or border is 1/4 of an inch inside the top left corner of the box or border. The example above creates a page with a border, a box, and a diagonal within the box. Each has a different line type and a different line width.

2.18 Writing Text and Symbols

The PUT subcommand writes a string to the print file. The string can be placed right justified, left justified, or centered at the current location.

```
PUT 'P-STAT Update Manual',
NEXT,
PUT 'PostScript Usage',
```

The PUT followed by LEFT, CENTER, or RIGHT controls the placement of the string. The assumption is that the string will be positioned LEFT justified against the current coordinates. If CENTER or RIGHT is specified, the justification is done accordingly.

```
PUT CENTER 'TABLE OF CONTENTS' $
```

PUT may be followed by the name of one of the plot symbols instead of a string. The possible symbols are:

```
POINT    PERIOD     SQUARE      TRIANGLE    STAR
CIRCLE   PLUS       DIAMOND     HASH        DOT
X        Y          Z
```

A scale factor may be provided for symbols. If there is none, a factor of 2, which makes very small symbols is assumed.

```
SCALE 5,  SYMBOL DOT,     ( DOT is a filled circle )
```

The line type and line widths for the symbols can be selected by using:

```
SYMBOL SQUARE, LINE.TYPE SYMBOL DASH, LINE.WIDTH SYMBOL 3.5,
```

A symbol is drawn using whatever color is specified for the current font. This means that you can use the SETFONT command described below to change the font and, therefore, change the current color before each symbol is drawn.

2.19 Setting the Font

The only place in the POSTSCRIPT commands where the active font can be explicitly set is in the POSTSCRIPT subcommand language. FONT3 is always used for title fonts and is set by the routines which print the titles. After the title is printed, the font is reset to FONT1. If you wish a font other than FONT1 after a NEW.PAGE, use the SETFONT subcommand. Aside from titles, POSTSCRIPT has no “parts” to control the font settings. If a SETFONT is not given FONT1 is assumed for any puts that are done.

```
POSTSCRIPT, PR TEST.PS;
  BORDER, LINE.WIDTH 2,
  NEW.PAGE,
  MOVETO 2 2,
  FONT2 ARIAL BOLDITALIC 12,
  SETFONT 2,
  PUT 'HAPPY BIRTHDAY',
  SKIP 2,
  SETFONT 1,
  PUT 'TO YOU',
  PUT.PAGE $
```

SETFONT is followed by one of the numbers between 1 and 9. If the font has not been explicitly defined it is set to Times-Roman 8 point.

2.20 Customizing Colors

Colors are determined by the mixture of red, green, and blue. For example:

1.	BLACK	0. 0. 0.
2.	RED	1. 0. 0.
3.	GREEN	0. 1. 0.
4.	BLUE	0. 0. 0.
5.	YELLOW	1. 1. 0.

The color orange is composed of 1 unit of red, .5 units of greeen and .1 unit of blue. You can select colors other than the basic color names supplied by using values which must be numbers between 0 and 1. To set FONT2 to a light grey you could use:

```
COLOR FONT2  .8  .8  .8 ,
```

Even amounts of the three colors give varying shades of grey. The closer to 1, the lighter the shade of grey. Shades of orange require full red, some green and a touch of blue such as (1. .5 .1). Thus the COLOR subcommand takes two forms:

```
COLOR  (font or border)   color-name (RED,ORANGE, etc.)
COLOR  (font or border)    v1  v2  v3
```

If the RGB values are used, all three must be present. Any value that is greater than 1 or less than 0 is set to zero. The colors in POSTSCRIPT.SETUP can also be provided this way.

2.21 EPS: ENCAPSULATED POSTSCRIPT

Encapsulated Postscript is used for a single page or piece of a page that is to be imported into another text processing program. An EPS file can specified in either the POSTSCRIPT.SETUP or the POSTSCRIPT command. The identifier is “ENCAPSULATE” and it has no arguments. The usual sequence will be a POSTSCRIPT command, a single graphic or line oriented command, and a final POSTSCRIPT command to finish it.

```
POSTSCRIPT, PR 'Test.eps', ENCAPSULATE, MARGINS 2. 4. 2. 4. $
Plot Myfile;
P year by Income;
$
POSTSCRIPT.CLOSE $
```

There are really only two important requirements for an EPS file:

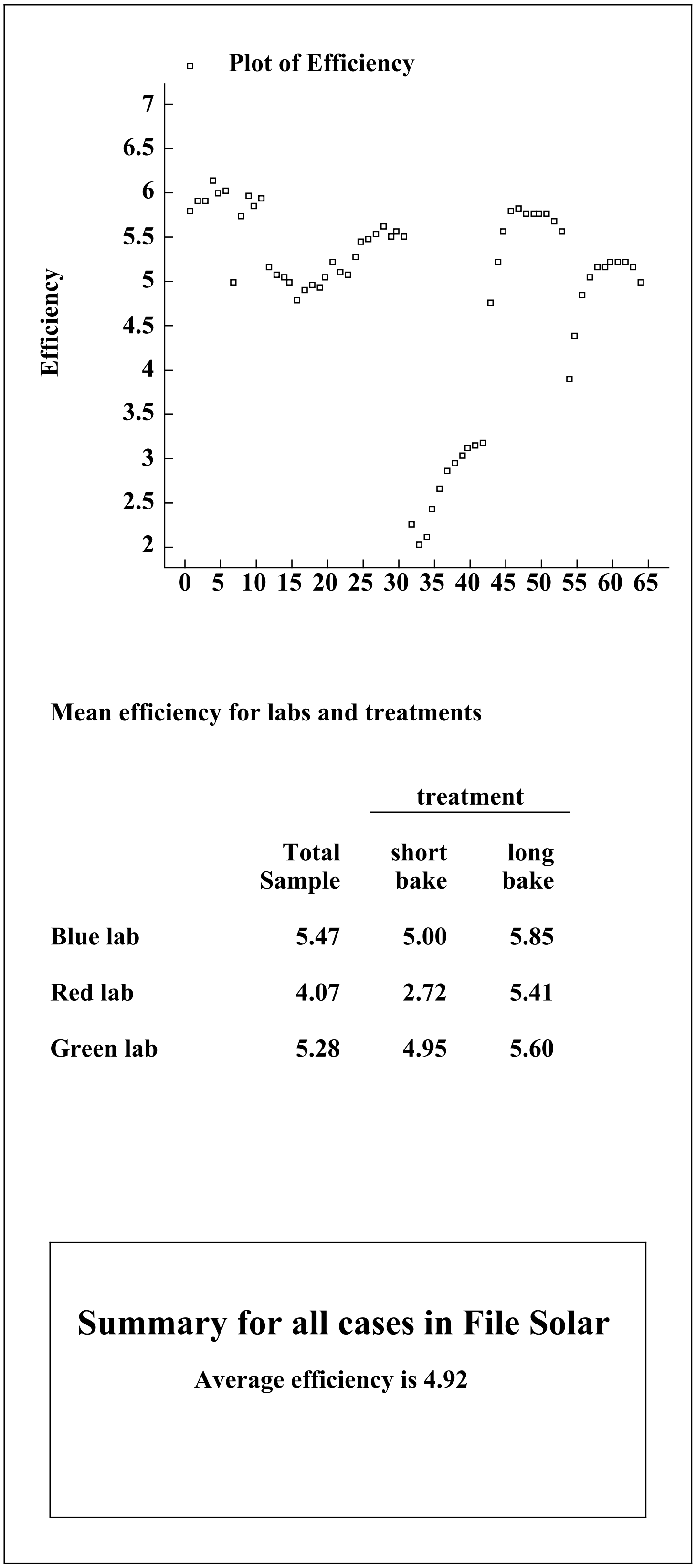
1. The showpage instruction is not used. The EPS file us usually just a portion of the page in the importing program. It is up to that program to determine when a page is complete.
2. The prefix at the beginning contains the margins that define the outer edges of the EPS file. This information is used to determine boundaries for the imported graphic. The margins that are used to create an EPS file in P-STAT are the margins that are defined in the initial call to the POSTSCRIPT command. If you are using a border and ENCAPSULATE, you should set the border value to 0. so that it will not be outside the defined margins.

```
      POSTSCRIPT.SETUP, ENCAPSULATE $
      PLOT Myfile;
      P Income BY Year;
      $
```

The use of ENCAPSULATE turns off SHOWPAGE in any of the commands that follow.

P-STAT does not export the bit-mapped data so you may not see the imported information on your screen if you are previewing the combined output in the importing program. However, most importing programs will at least show you the rectangle that is indicated given boundaries in the EPS file. The program may also show the text information from the header comments.

POSTSCRIPT EXAMPLES



Mean efficiency for labs and treatments

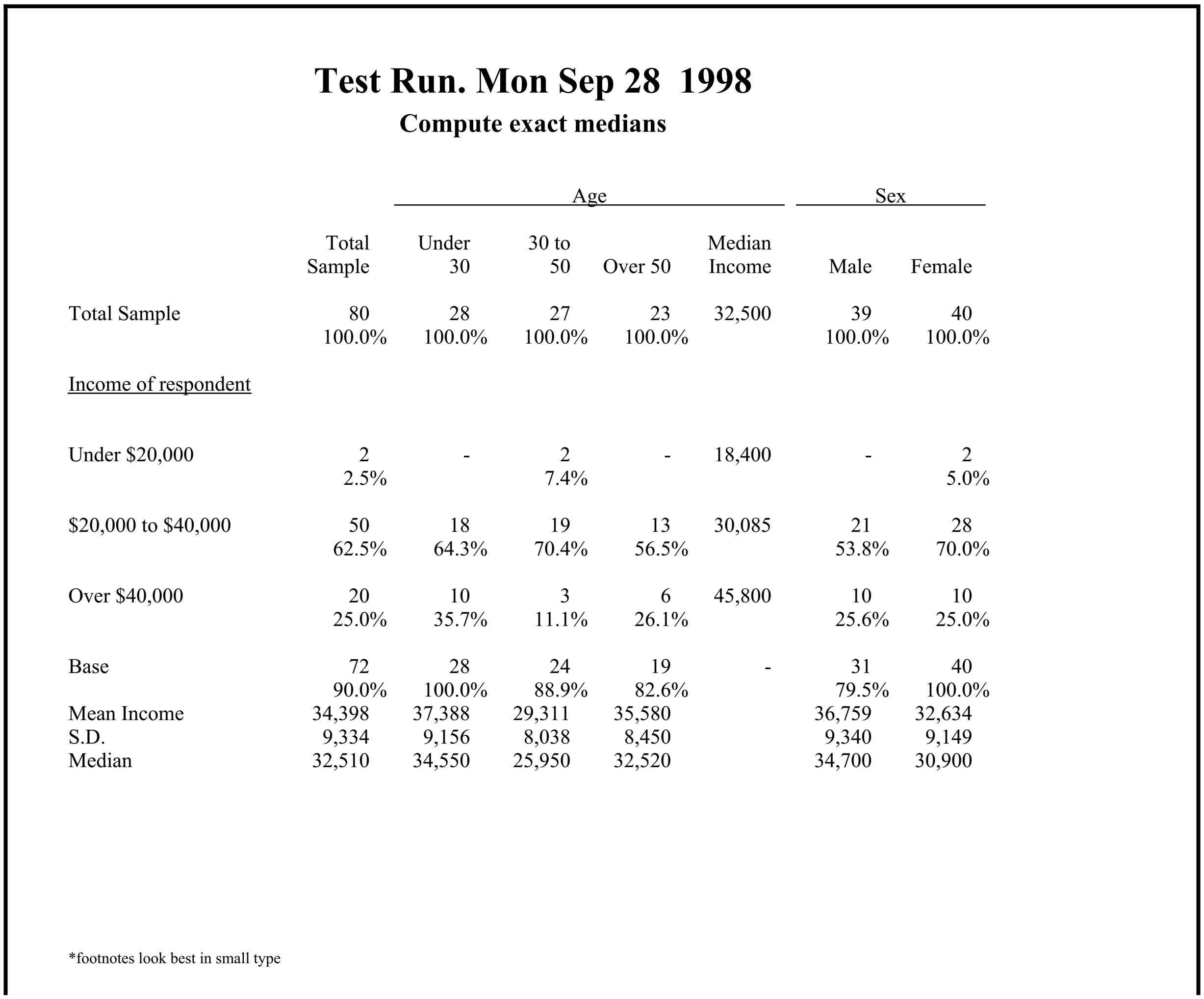
		treatment	
	Total Sample	short bake	long bake
Blue lab	5.47	5.00	5.85
Red lab	4.07	2.72	5.41
Green lab	5.28	4.95	5.60

Summary for all cases in File Solar

Average efficiency is 4.92

```
POSTSCRIPT,
  PORTRAIT,
  PR 'demo1.ps',

  NO SHOWPAGE,
  BORDER,
  MARGINS 1.5 1.5 1.5 3.75,

  FONT TIMES BOLD 10 $
  PLOT Solar;
  P eff, DIMENSION PLOT 3 BY 3.5;
$


SURVEY Solar,
  LABELS solar.lab,
  TOP.EDGE;

  TITLE T1 LEFT
  'Mean efficiency for labs and treatments',

  BANNER treatment,
  STUB process,

  MEANS efficiency,
  LAYOUT LABELS BODY,
  BODY CONTAINS MEANS;
$


POSTSCRIPT;

  BOX  1.5  8  4.75  9.5,
  MOVETO  1.65  8.5,

  FONT9 HELVETICA BOLD 12,
  SETFONT 9,

  PUT
  "Summary for all cases in File Solar",
  NEXT;
$


TEXT.WRITER Solar
  [ IF FIRST ( .FILE. )
    GEN ##t = 0, GEN ##n = 0 ;

    INCREASE ##t BY efficiency,
    INCREASE ##n ;

    IF LAST ( .FILE. ) RETAIN,

    PUT @PLACES2
    'Average efficiency is '
    (  ##t/##n )],

TOP.EDGE   8.8,
LEFT.EDGE  2.25,
$


POSTSCRIPT.CLOSE,
   PRINT$
```

```
$
```



Test Run. Mon Sep 28 1998

Compute exact medians

		Age				Sex	
	Total Sample	Under 30	30 to 50	Over 50	Median Income	Male	Female
Total Sample	80 100.0%	28 100.0%	27 100.0%	23 100.0%	32,500	39 100.0%	40 100.0%
Income of respondent							
Under $20,000	2 2.5%	-	2 7.4%	-	18,400	-	2 5.0%
$20,000 to $40,000	50 62.5%	18 64.3%	19 70.4%	13 56.5%	30,085	21 53.8%	28 70.0%
Over $40,000	20 25.0%	10 35.7%	3 11.1%	6 26.1%	45,800	10 25.6%	10 25.0%
Base	72 90.0%	28 100.0%	24 88.9%	19 82.6%	-	31 79.5%	40 100.0%
Mean Income	34,398	37,388	29,311	35,580		36,759	32,634
S.D.	9,334	9,156	8,038	8,450		9,340	9,149
Median	32,510	34,550	25,950	32,520		34,700	30,900

*footnotes look best in small type

SUMMARY

POSTSCRIPT.CLOSE

```
POSTSCRIPT.CLOSE $

POSTSCRIPT.CLOSE, PRINT $
```

The POSTSCRIPT.CLOSE command completes a PostScript block and closes the PostScript printfile. It does not reset any PostScript parameters.

Optional Identifiers

PRINT

The PRINT identifier is available only on those computers where the P-STAT PRINT command is supported.

POSTSCRIPT.RESET

```
POSTSCRIPT.RESET $
```

POSTSCRIPT RESET is used to reset the PostScript parameters to their original values. All margins are set to 1 inch. All fonts are set to Times-Roman 8 point. The leading is set to 1/72 of an inch. The orientation is set to landscape. Borders, titles, and the character set and encapsulate flags are turned off. Colors to fonts and the border are reset to black.

POSTSCRIPT.SETUP

```
POSTSCRIPT.SETUP, MARGINS 2, DELTA 10 $

POSTSCRIPT.SETUP, TOP.EDGE 1.5,
          FONT1 ARIAL ITALIC 10,
          FONT2 COURIER BOLD 8,
          FONT3 'ZapfChancery-MediumItalic' 12
          FONT9 TIMES BOLD 14,
          COLOR FONT9 RED,
          TITLES TOP 1 9 $
```

The POSTSCRIPT.SETUP command is used to change the default settings of the PostScript parameters. These settings become the assumed settings for subsequent SURVEY, PLOT, and POSTSCRIPT commands. Settings for the 9 available fonts remain in effect until they are changed by any of the commands that support PostScript output. Other settings, such as margins remain in effect until they are changed or reset in one of the “POSTSCRIPT” commands.

Optional Identifiers

CHARACTER.SET ISO / GERMAN

causes the necessary translations for the ISO 8859-1 or German characters to be written as port of the PostScript start up instruction.

COLOR FONT/BORDER COLOR

COLOR is followed by one of BORDER, FONT or FONT1-FONT9 and the desired color. The color names that are supported are RED, ORANGE, YELLOW, GREEN, BLUE, VIOLET, BROWN,BLACK, WHITE, CYAN and MAGENTA. See the PostScript subcommands for customized colors.

DELTA nn

is used to provide a fudge factor used by PostScript commands to help decide where a line break should be. If the line breaks are inappropriate and breaking in the middle of words that should fit, setting DELTA to a small value such as 10 should correct the problem.

ENCAPSULATE

used when a single graphic is to be written as an EPS (Encapsulated PostScript) file for inclusion in some other text processing program.

FONT font name nn

sets all 9 fonts to the specified font and point size. The font name may be in quotes or it may be one of the following:

```
TIMES               ARIAL               COURIER
TIMES BOLD          ARIAL BOLD          COURIER BOLD
TIMES ITALIC        ARIAL ITALIC        COURIER OBLIQUE
TIMES BOLDITALIC    ARIAL BOLDITALIC     COURIER BOLDOBLIQUE
```

FONT1 - FONT9

Like FONT above except that the supplied definition only applies to the single specified font.

LANDSCAPE

The orientation of the printout is to be such that the width is the larger dimension. On 8.5 by 11 inch paper, landscape orientation has a height of 8.5 and a width of 11. LANDSCAPE is assumed.

LEADING nn

specifies the extra space that is allowed between lines of text. The default is 1, which is 1/72 of an inch.

MARGINS nn nn num nn

provides the size in inches for the 4 margins to be used at the top, bottom, left, and right edges of the paper, in that order. If a single argument is provided, that amount is used for all 4 margins. If there is more than 1 number, all 4 numbers must be provided.

BOTTOM.EDGE nn

provides the size in inches for the bottom margin.

LEFT.EDGE nn

provides the size in inches for the left margin.

RIGHT.EDGE nn

provides the size in inches for the right margin.

TOP.EDGE nn

provides the size in inches for the top margin.

PAPER.SIZE nn nn

provides the size in inches of the paper. 8.5 by 11 is assumed. The first number is the height, the second number is the width.

PORTRAIT

The orientation is to be such that the height is greater than the width. On 8.5 by 11 inch paper, the height is 11 inches and the width 8.5 inches.

SHOWPAGE

is assumed. The PostScript showpage instruction which defines the end of a page is issued by each program automatically. NO SHOWPAGE is used when you wish to control the pagination yourself.

TITLES TOP/BOTTOM nn nn

The TITLES identifier is used to associate a font with a single title line. The first argument must be either TOP or BOTTOM. The second argument must be 1-9 for top titles and 1-3 for bottom titles. The final argument is a font number between 1 and 9

```
TITLES TOP 3 7
```

specifies that the third top title is to be printed using FONT7.

POSTSCRIPT

```
POSTSCRIPT, PR 'Test.ps', NO SHOWPAGE, BORDER $
```

```
POSTSCRIPT, PUT.PAGE, NEW.PAGE $
```

```
POSTSCRIPT;
   FONT9 TIMES BOLD 14,
   PUT 'HEADER 1';
$
```

The POSTSCRIPT command is used to begin a PostScript block which is comprised of one or more P-STAT commands using the same print destination and similar parameters. The PR identifier is required when a PostScript block begins. A PR identifier may not be used within a PostScript block. The POSTSCRIPT command may itself be used within the PostScript block to change fonts or margins, to draw boxes and lines, to complete or begin a page, to print a page, or to add some text to the page.

Required Identifiers

PR “fn”

is required only for the POSTSCRIPT command which begins a PostScript block. It provides the name for the external file which is to contain the printout. POSTSCRIPT.CLOSE is used to end the PostScript block.

Optional Identifiers

all of the identifiers for the POSTSCRIPT.SETUP command may be used in the POSTSCRIPT command. Additional identifiers follow

BORDER

requests that all subsequent pages controlled by the POSTSCRIPT command be bounded by a border. This border is usually placed 1/4th of an inch outside the current margins so that it will not impinge on any titles that are defined. It will be a solid line 1/144 of an inch wide unless these parameters are changed in a POSTSCRIPT subcommand. NO BORDER may be used to turn off the border.

CHARACTER.SET ISO / GERMAN

causes the necessary translations for the ISO 8859-1 or German characters to be written as port of the PostScript start up instructions.

NEW.PAGE

writes out the parameters needed to start a page. If titles or borders have been requested, they are written out. NEW.PAGE is not required for the first page created by the PostScript block. NEW.PAGE as an identifier is processed after any PUT.PAGE and before any subcommands are read.

PUT.PAGE

defines the end of a page. The postscript instruction “showpage” is issued. PUT.PAGE is executed before NEW.PAGE or any subcommands are processed. If you wish to begin and end a page in a single POSTSCRIPT command, use the subcommands to control the exact sequence.

PRINT

causes the PostScript printfile to be closed and a PRINT command is generated. This is only available when the PRINT command is supported.

TITLES

requests that previously defined titles be printed whenever a new page is started. NO TITLES can be used to produce pages without titles.

Optional Subcommands Which Are Also Identifiers

BORDER nn

When BORDER is used as a subcommand, the optional numeric argument can be used to change the position of the border in relation to the margin. The larger the number the further the border is outside the margin. An argument of 0 places the border on the margin. A negative number places the border inside the margin.

```
 BORDER  -.25,
```

The argument -.25 places the border 1/4th of an inch inside the margin. Borders inside the margin or wide borders on the margin may conflict with titles. NO BORDER can be used to turn borders off. The BORDER subcommand sets the parameters for future borders. The NEW.PAGE subcommand causes the border to be placed on the page.

FONT, FONT1 through FONT9

The font subcommands are exactly the same as the font identifiers.

MARGINS nn nn nn nn

MARGINS and the four edge subcommands, TOP.EDGE, BOTTOM.EDGE, LEFT.EDGE and RIGHT.EDGE can be used exactly as they are used when given as identifiers.

NEW.PAGE

causes page parameters and any border or titles to be written out. The new page action happens immediately after the subcommand is read. NEW.PAGE as a subcommand is not limited to a single usage. Multiple uses of NEW.PAGE can produce unusual border and title effects.

PUT.PAGE

cause the “showpage” that defines the end of a PostScript to be written to the printfile. When PUT.PAGE is used as a subcommand the action occurs immediately after the subcommand is read. Multiple uses of PUT.PAGE as a subcommand are permitted.

TITLES arg arg cs

TITLES by itself causes any existing titles to be written out when a NEW.PAGE is encounters. NO TITLES can be used to turn titles off. In addition, TITLES used as a subcommand can be used to define a single top or bottom title.

```
TITLES T3 LEFT 'Text for the third left title',
TITLES B1 RIGHT 'This goes at the bottom',
```

If TITLES is followed immediately by text the top center title is assumed. The arguments that may follow the TITLES subcommand are T1 through T9 for the 9 top titles, B1 through B3 for the 3 bottom titles and LEFT, CENTER, or RIGHT. T1 is assumed if the location is not specified. CENTER is assumed if the position is not specified.

Optional Subcommands

BOX nn nn nn nn

draws a box which has its upper corner X1 inches from the left edge of the page and Y1 inches from the top of the page and its lower corner X2 inches from the left and Y2 inches from the top. X1 is the first number. Y1 is the second number. X2 is the third number and Y2 is the last number. Unless LINE.TYPE and LINE.WIDTH are used to change the box parameters, the line is a solid line 1/144 of an inch thick.

COLOR arg arg

requests that a given color be used for a font or the border. The first argument is either “border” or the font name. The second is the color name (red, orange, yellow, green, blue, violet, brown, black, white, cyan or magenta).

COLOR arg nn nn nn

requests that the designated font or border be given a color composed of the designated amounts or red, green, and blue. The numbers must be between 0 and 1. For example;

```
COLOR BORDER 0  1.  1.,  COLOR FONT1 .4 .4 .4,
```

requests that the border be the color CYAN and that FONT1 be a medium grey.

LINETO nn nn

draws a line from the current X/Y coordinates to the location X1 inches from the left and Y1 inches from the top. X1 is the first number. Y1 is the second number. Both numbers are required. The line width and line type can be controlled by the LINE.WIDTH AND LINE.TYPE subcommands.

LINE.TYPE arg arg

The first argument is any of BOX, BORDER, LINE, or SYMBOL. The second argument is any of the following line types:

```
DOTTED      DOT.DASH      SHORT.DASH    DASH
LONG.DASH   ODD.DASH      SOLID
```

```
LINE.TYPE   BOX      ODD.DASH,
LINE.TYPE   BORDER   DOTTED,
```

LINE.WIDTH arg nn

The first argument is any of BOX, BORDER, LINE, or SYMBOL. The second argument is the line width given in units of 1/72 of an inch. LINE.WIDTH BORDER 18, produces a border that is 1/4th of an inch wide. A width of zero produces an invisible line.

MOVETO nn nn

move to the location that is X1 inches from the left edge of the page and Y1 inches from the top of the page. X1 is the first number. Y1 is the second number. Both are required.

NEXT arg

the argument determines whether the move is to the next line and the current column or the next line and the defined left margin. The possible arguments are MARGIN and COLUMN. If there is no argument, COLUMN is assumed.

PUT arg cs

The optional arguments are LEFT, RIGHT, or CENTER. If none of these are used, LEFT is assumed. The contents of the string which must be in quotes are placed and then properly left/right/center justified at the current X/Y coordinates.

PUT arg

The argument may be any of the standard plot symbols which will be placed at the current x/y coordinates and scaled using the scale factor which is set to 2 unless the SCALE identifier is used. The available symbols are:

```
POINT    PERIOD     SQUARE      TRIANGLE    STAR
CIRCLE   PLUS       DIAMOND     HASH        DOT
X        Y          Z
```

The scale must be set before a symbol is requested. The symbol or text will be drawn using the color of the active font (see SETFONT).

SCALE nn

The number is the scale factor to be used in drawing the plots. Most of the symbols are twice the dimensions of the scale factor calculated as 72 units per inch. A scale of 18 produces symbols that are approximately 1/2 half inch across.

SETFONT nn

The number must be between 1 and 9 and determines which font will be the currently active font. This font will be used for subsequent puts. If NEW.PAGE is used to produce titles, FONT3 is used for the titles and the current font is reset to FONT1. Therefore SETFONT should be used after the NEW.PAGE subcommand unless FONT1 is the desired font.

SKIP arg nn

SKIP 1 is the same as NEXT, NEXT. The first argument which is optional and like the argument for the NEXT subcommand determines whether the X coordinate moves to the margin or to the location of the current column. The actual distance that the Y coordinate changes depends on the point size of the currently active font.

3
PLOT:
Multivariable Scatter Plots

The PLOT command graphs *numeric* variables on scaled axes and produces scatter plots. The coordinates of the plotted points are the paired values (X,Y) for two variables. It is possible to produce plots of a single variable, plots of three or more variables, and multiple plots on one set of axes. In addition, separate plots for subgroups may be requested by using the identifier BY in the PLOT command.

Desired plots and options are specified at the subcommand level. Reasonable settings for output parameters are automatically selected by the PLOT command but may be reset by the user. In addition, extensive online help is available.

PLOT is described in full in the following chapters:

1. “PLOT: Multivariable Scatter Plots” describes the basic identifiers and subcommands for producing scatter plots.
2. “PLOT: Labels and Titles” describes ways to enhance the PLOT output.
3. “PLOT: PostScript Support”
4. describes how to produce camera-ready output on a PostScript device. This includes font selection, colors, symbols and line-types as well as controls for printing multiple plots on a single page.
5. Examples of PLOT output when XWindows support is available are shown in the chapter “XWindows Support”.

The identifiers and subcommands described in this chapter are used for all types of plot output: character plots, PostScript plots and XWindows plots.

3.1 PRODUCING PLOTS

The command PLOT is entered in response to the *command* prompt. It is followed by the name of the P-STAT file containing the input data. This command phrase requests that a file named GasElec be input to the PLOT command:

```
PLOT  GasElec ;
```

In interactive usage, a prompt for a *subcommand* (or Quit or Help) follows. The particular plots and any additional options are specified at the subcommand level. The subcommand P is followed by the names of the variables to be plotted. The first variable named is measured on the Y-axis — the vertical axis, and the second variable is measured on the X-axis — the horizontal axis. This subcommand produces a plot of Gas.CCF by Month:

```
P  Gas.CCF  *  Month ;
```

The “*” in the PLOT subcommand means that the first variable should be plotted “by” the second variable. A semicolon (;) signals the end of the subcommand. The plot is produced and a prompt is then issued for another subcommand.

The plot illustrated in Figure 3.1 shows household natural gas usage (measured in cubic feet — CCF) for home heating and hot water production, by month, over an eight and one-half year period. A “*” represents a count of one pair of values at that location, a “2” represents two pairs, and so on. A “$” represents a count of ten or more.

The range of values shown on the X-axis and Y-axis and the intervals between the tick marks on the axes are automatically chosen by the PLOT command. They are based on the particular data values for the plot and on the characteristics of the output destination, in this case, a terminal.

After all desired subcommand requests are made, a “$” terminates the PLOT command:

```
PLOT  GasElec ;
P  Gas.CCF  *  Month ;
$
```

Figure 3.1 Producing a Plot

```
     ----------The Command----------------------------------------

     PLOT  GasElec  ;
     P  Gas.CCF  *  Month ;


     ----------The PLOT Output------------------------------------


           Plot of Gas.CCF by Month (Legend: *=1, 2=2, ..., $=10+)

     500 +
         |
         |    2                                                 *
     400 +3    *                                                 *
G        |3
a        |*    2    *                                            *
s    300 +         *                                            3
         |    *                                                 *
C        |2    *                                            5
C    200 +         2                                       2
F        |    *    3    4                             2    *
         |    *    *    *              *              2         *
     100 +              *    *                        3
         |              3    6    3              5
         |         *         2    6    8    8    3
       0 +                                             *
          +----+----+----+----+----+----+----+----+----+----+----+
          1    2    3    4    5    6    7    8    9    10   11   12
                                      Month
```

3.2 Requesting Multiple Plots

Several plots may be requested in one plot subcommand by specifying a *range* of variables:

```
P  Gas.CCF  TO  Elec.KWH  *  Month ;
```

Plots are produced for as many variables as there are from Gas.CCF through Elec.KWH, all by the variable Month. In this example, three plots are produced because there is one variable, Gas.Cost, between Gas.CCF and Elec.KWH in the file. The plots produced are: 1) Gas.CCF by Month, 2) Gas.Cost by Month, and 3) Elec.KWH by Month. To produce only the first and third plots, this subcommand is issued:

```
P  Gas.CCF  Elec.KWH  *  Month ;
```

A plot of Gas.CCF by Month followed by a plot of Elec.KWH by Month is produced.

3.3 Reminders and Help

Several subcommand phrases request reminders of plot settings and variables available for use. Help on general and specific topics may also be requested within the PLOT command.

This subcommand displays the parameter settings currently in use:

```
SHOW  SETTINGS ;
```

Figure 3.2 shows the settings for the plot in Figure 3.1. This subcommand shows all the variables in the input file that may be used for plots:

```
SHOW  VARIABLES ;
```

SHOW NAMES (or just SHOW) is a synonym for SHOW VARIABLES.

Figure 3.2 Displaying Plot Settings

```
SHOW  SETTINGS ;


  --- PLOT subcommand settings ---

LINES = 22.
COLUMNS = 80.
TICKS will be evenly spaced along the  X  and  Y  axes.
Plots will be scaled by 1.
EDGES = B   L
Plot characters: *23456789$
```

Either of these subcommands provides general information about PLOT:

```
HELP ;   or

H
```

HELP, followed by a more specific topic, provides information on that topic:

```
HELP  RANGE ;
```

In this case, further information on RANGE is requested.

AGAIN may be used by itself to repeat a plot exactly as it is, or with other subcommand phrases to repeat a plot with modifications. This subcommand redoes the last plot, but directs the output to an external file named “PrtFile”:

```
AGAIN,  PR  PrtFile ;
```

PR is the abbreviation for PRINTER, which may also be used. Notice that commas separate multiple phrases within one subcommand.

All subcommands within the PLOT command may be abbreviated. The number of characters required for a given subcommand must be sufficient to distinguish it from other subcommands. Thus, SHOW VARIABLES may be abbreviated to SHOW V and SHOW SETTINGS may be abbreviated to SHOW S. Any longer abbreviation, as well as the complete subcommand, is always acceptable. The examples and figures in this chapter generally illustrate the full subcommands for clarity.

3.4 PLOT SETTINGS

Settings for various plotting parameters may be specified by the user. These include:

1. the lines and columns available for plot output,
2. the range of the X-axis and Y-axis values,
3. the presence or absence of tick marks and grid lines, and
4. the spacing and labeling of these marks and lines.

These settings are in effect for all subsequent plots, unless they are reset or changed.

3.5 Setting LINES and COLUMNS

Initially, the PLOT command sets the number of lines (the vertical distance) available per plot to the screen size of the terminal or, if the output is going to a printer or disk file, to the page size of the output destination. The user may specify an alternate number of lines per plot for the duration of the PLOT command by entering the LINES subcommand:

```
LINES 40 ;
```

LINES is followed by the number of lines desired per plot. This line setting remains in effect for all subsequent plots until it is respecified. It appears in the display of settings. To change the LINES setting, enter another LINES subcommand or append the LINES phrase to a PLOT subcommand:

```
P  Gas.CCF  *  Month,  LINES 30 ;
```

COLUMNS sets the number of columns available per plot (the horizontal distance). This subcommand sets the number of columns to 60:

```
COLUMNS 60 ;
```

The columns setting remains in effect for all subsequent PLOT subcommands. If the number of columns is not explicitly specified, it is set to the output width of the current output destination. Generally, this is 80 columns for a terminal and 132 columns for a disk file or printer unless a PRINTER.SETTINGS command has been used to specify the output width for a given print destination. For example:

```
PRINTER.SETTINGS 'Plot.prt',  OUTPUT.WIDTH 160 $
PLOT  GasElec,  PR 'Plot.prt';
P  GAS.CCF * Month ;
$
```

3.6 Specifying the RANGE of Variable Values

The range of the values to be plotted may be specified by the user. When the range is not explicitly specified, all pairs of values are plotted. (Pairs where one of the variables is missing are not plotted.) Often the presence of outlying points results in a large blank area between the majority of the points and the outlying points on the plot. In such cases, it is often desirable to specify a narrower range of values to be used in replotting the data. This permits the space available for a plot to be used more effectively for the bulk of the data points.

For example, in a plot of monthly food expenditures by family size, the presence of a home with 35 foster children may cause a plot with a broad range to be produced. The bulk of the data may occupy only one fifth of that range. Replotting the data with a narrower range uses the plot area more effectively.

Sometimes it is convenient to have a consistent range from plot to plot. For example, plots of daily temperature readings for patients involved in a drug study may be more easily compared if all patient plots have the same range. Or, ranges that include zero may produce plots that permit meaning to be attributed to the absolute size of space on the plot.

Figure 3.3 Specifying Range Settings

```
----------The PLOT Command-----------------------------

 PLOT  GasElec ;

 P  Gas.CCF  *  Month,  RANGE FOR Y  0 TO 600,  SHOW SETTINGS ;


 ----------The Output-----------------------------


  --- PLOT subcommand settings ---

LINES = 22.
COLUMNS = 80.
RANGE FOR Y = 0  TO  600.
TICKS will be evenly spaced along the  X  and  Y  axes.
Plots will be scaled by 1.
EDGES = B   L
Plot characters: *23456789$
```

```
           Plot of Gas.CCf by Month (Legend: *=1, 2=2, ..., $=10+)

     600 +
         |
     500 +
G        |     *                                                 *
a    400 +5    2                                                 *
s        |2    *
     300 +     *    2                                            4
C        |2    2                                            3    *
C    200 +          5                                       4
F        |     *    *    4              *              4    *    *
     100 +     *         3    *                        3
         |          *    2    8    6    8    7    7
       0 +                         3         *    *    *
          +----+----+----+----+----+----+----+----+----+----+----+----+
          1    2    3    4    5    6    7    8    9    10   11   12
                                      Month
```

This subcommand redoes the plot shown in Figure 3.1, but with a range for gas usage going from zero to 600 CCF:

```
P  Gas.CCF  *  Month,

   RANGE FOR Gas.CCF  0 TO 600,  SHOW SETTINGS ;
```

SHOW SETTINGS shows the plot settings before the plot is displayed. This is illustrated in Figure 3.3.

RANGE often limits the data values selected for the plot. Data points outside the range boundaries are omitted from the plot. The range of the Y-axis can be specified using either of these subcommands:

```
RANGE  Gas.CCF  0 TO 600 ;   or

RANGE  FOR Y    0 TO 600 ;
```

The FOR is optional in either specification. If there is a variable named Y, RANGE is set for that variable rather than for the Y-axis. (As a rule, single letter variable names should be avoided to prevent confusion with H, Q, E and other P-STAT directives.) If both the X-axis and the Y-axis are to be rescaled, a single RANGE phrase may be used to refer to both axes.

PLOT assigns tick marks so that “nice numbers,” such as 1, 2, 5 and 10, label the ticks on the axes. However, sometimes the low and high tick marks are outside the range of actual data values. (For example, the upper tick mark may be 350, although the upper boundary of the range was specified as 246.) High and low tick marks that correspond exactly with the range boundaries may be requested using RANGE, TICKS and EXACT:

```
RANGE FOR Y  0 TO 425,

TICKS FOR Y  0  75  150  225  300  375  425,

EXACT RANGE FOR Y ;
```

The RANGE FOR Y gives the range of values to be included in the plot, the TICKS FOR Y gives the tick mark locations, and EXACT RANGE FOR Y specifies that axis conform exactly to the specified range of the Y variable. The words RANGE and FOR can be omitted after EXACT, and just EXACT Y can be used. TICKS must be used with the EXACT subcommand. If RANGE is not used, the observed range of the values determines the height of the Y axis.

3.7 Changing Plot Settings

Plot settings, either those initially assumed by the PLOT command or those specified by the user, are in effect until they are: 1) changed (respecified), 2) reset to their initial values, or 3) turned off. The current settings affect all subsequent plot requests. Thus, RANGE 4 TO 28 sets the range of values to be plotted for all subsequent plots until it is changed with another RANGE specification, set back to the default values with RESET, or turned off with NO.

The only settings that automatically reset themselves are OVERLAY and LABELS, which are specific to each plot. (These options are described later in this manual.)

AGAIN repeats the current plot, typically with changes to specific settings:

```
P  Gas.CCF  *  Year ;
AGAIN,  LINES  18 ;
```

The previous plot of Gas.CCF by Year is repeated, but with a line setting of 18 lines. All other settings remain as they were initially set by PLOT or as they were specified by the user.

RESET resets either all plot settings or specified settings. RESET, by itself, changes all plot settings back to the initial values assumed when PLOT began

ESET may also be used with various plot subcommands to reset just those particular options:

```
RESET  RANGE ;      or

RESET  RANGE  TICKS  LEGEND ;
```

Note that multiple subcommands are reset in a single RESET phrase. Subcommand options that may be RESET are:

```
BY                      OBSERVE
CHARACTERS              OVERLAY
COLUMNS                 RANGE
EDGES                   REVERSE
GRID.LINES              SCALE
LABELS                  TICK.LABELS
LEGEND                  TICKS
LINES                   TITLES
```

RESET may be part of a subcommand. This subcommand requests that the last plot be displayed with all current settings, but that the RANGE specification be reset:

```
AGAIN,  RESET  RANGE ;
```

INITIALIZE is a synonym for RESET.

The word NO may be used before various options to turn them off; that is, to remove them from a plot. For example, the marking of the axes into even intervals may be turned off by the phrase:

```
NO TICKS ;
```

There will be no ticks on all subsequent plots until TICKS are RESET:

```
AGAIN,  RESET TICKS ;
```

Multiple subcommands are turned off using NO followed by a list of the subcommands:

```
NO TICKS  LEGEND  EDGES ;
```

NO may be used with these subcommand options:

```
BY                      RANGE
EDGES                   REVERSE
GRID.LINES              TICK.LABELS
LEGEND                  TICKS
OBSERVE                 TITLES
OVERLAY
```

3.8 PLOT OPTIONS

Various options let users tailor their plots. These options include:

1. tick marks at specific locations,
2. character labels in place of tick marks,
3. horizontal, vertical and diagonal reference lines,
4. boxes around plots,
5. reversed axes, and

6. different plotting characters and scales.

When these options are not used, the PLOT command selects reasonable tick marks and plotting characters. Plots are not boxed, and there are no reference lines other than the axes. When any of these options are specified, they remain in effect until they are respecified, reset or turned off.

3.9 Tick Mark Locations

The location of tick marks may be precisely specified. A list of actual values or a range of values may be provided. When the specifications are different for each axis, separate TICKS phrases are required:

```
TICKS FOR X    2   4   6   8   10     ;
TICKS FOR Y  10   TO  130  BY  20 ;
```

The first subcommand positions tick marks at 2, 4, 6, 8 and 10. The second positions them at 10, 30, 50, 70, 90, 110 and 130. Variable names may be used instead of X and Y in all PLOT subcommands:

```
TICKS FOR Month 1 TO 12 ;
```

Since BY is not specified in this example, BY 1 is assumed.

Tick marks may be specified for just one axis if desired. If the specifications for both axes are the same, they may be defined for both axes simultaneously:

```
TICKS   2  8  14  ;
```

The specified tick marks are in effect for subsequent plots until either RESET TICKS sets the tick marks back to the default values calculated by PLOT, NO TICKS removes the tick marks altogether, or another TICKS subcommand defines new tick locations.

If neither range nor tick marks are provided, the PLOT command calculates how many ticks are reasonable and where they should be. The first tick mark will be less than or equal to the lowest observed value of the variable and the final tick mark will be greater or equal to the highest observed value of the variable. RANGE provides an override to the observed low and high values Values outside the range are not included in the PLOT. However, the actual X and Y axes are based on the lowest and highest of the range or tick mark values.

3.10 Tick Labels

The PLOT command labels the ticks along the axes with numbers, unless labels are supplied. Sometimes, particularly for categorical data, character labels are preferable to numbers. If tick labels are supplied, they are used instead of the numeric ticks — whether generated by plot or supplied in a TICKS subcommand.

The first tick label is used to replace the first numeric tick, the second to replace the second numeric tick, and so on. Therefore, if you are not sure what ticks PLOT will supply, define TICKS as well:

```
TICKS       FOR Gas.CCF   200  400  600,
TICK.LABELS FOR Gas.CCF   Low Average High;
```

PLOT uses the TICKS to decide where the TICK.LABELS are to be placed. The labels substitute for the numbers. If the number of ticks and the number of tick labels do not agree, the number of points on the axis that are labeled is the smaller number.

Figure 3.4 illustrates using TICK.LABELS for the variable Month. Because PLOT gives single integer tick marks whenever possible, it is not necessary to supply TICKS of 1 through 12.

Labels which follow the rules for legal names in P-STAT may be entered without quotes. Labels which are not legal must be enclosed in quotes:

```
TICK.LABELS  Year '1960' '1970' '1980';
```

Figure 3.4 Supplying Tick Labels

```
----------The PLOT Subcommands------------------------------

     P  Gas.CCF  *  Month,
     TICK.LABELS FOR Month  Jan Feb Mar Apr May Jun
                            Jul Aug Sep Oct Nov Dec ;


----------The Output---------------------------------------

           Plot of Gas.CCF by Month (Legend: *=1, 2=2, ..., $=10+)

     600 +
         |
     500 +
G        |     *                                                 *
a    400 +5    2                                                 *
s        |2    *
     300 +     *    2                                            4
C        |2    2                                            3    *
C    200 +          5                                       4
F        |     *    *    4              *              4    *    *
     100 +     *         3    *                        3
         |          *    2    8    6    8    7    7
       0 +                         3         *    *    *
          +----+----+----+----+----+----+----+----+----+----+----+
         Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
                                      Month
```

3.11 Specifying Reference Lines with GRID.LINES

A single line or a grid of lines may be superimposed on a plot. These may be horizontal and/or vertical lines, and ascending and/or descending diagonal lines. The location of the horizontal and vertical lines are the user's choice. The diagonal lines pass through the point (0,0). The lines are drawn with dots.

Grid lines are often useful when a specific line (or lines) can serve as an ideal comparison for the plotted values. For example, if one wanted easily to detect white blood cell counts below a given value, a line at that point would be helpful. Or, a grid of lines over the entire plot may highlight the relative distribution of points over the area of the plot.

The phrase:

```
P  Temperature  *  Time.of.Day,

   GRID.LINE   Temperature  98.6 ;
```

places a horizontal line, parallel to the X-axis and at Y = 98.6, on the plot of Temperature by Time.of.Day. It is easy to note patient temperatures that are markedly above or below normal. This grid line remains in effect until it is reset or removed.

Ascending and descending diagonal grid lines may also be specified. Either of these phrases places both ascending and descending diagonal grid lines on all subsequent plots at X = Y and X = -Y:

```
GRID.LINES  ASCENDING  DESCENDING ;   or

GRID.LINES  A   D ;
```

GRID.LINES, GRIDS or GRID may be used. ASCENDING may be shortened to A and DESCENDING may be shortened to D. The word FOR, as in GRID FOR Y, is optional.

An ascending diagonal line (at X = Y) superimposed on a plot of predicted scores by actual scores may make it easier to note markedly deviant scores. Note that ascending and/or descending diagonal grid lines make most sense when the ranges of the X and Y variables are the same or have an overlap of common values. If the ranges do not include any common values, a diagonal line at either X = Y or X = -Y does not show on the plot. Either RESET GRIDS or NO GRIDS eliminates the grid lines.

The phrase:

```
GRIDS  VERTICAL  HORIZONTAL ;   or

GRIDS  V   H ;
```

places grid lines at both X = 0 (vertical) and Y = 0 (horizontal). This is often desirable when the range of values on one or both of the axes includes zero and values on either side of zero. This is not suitable, however, when the axes are at X = 0 and Y = 0. VERTICAL may be abbreviated to V and HORIZONTAL may be abbreviated to H. GRIDS may be specified for each axis separately or, if the specifications for both axes are the same, jointly with a single phrase:

```
GRIDS 5 ;
```

A list of grid line specifications or a range of specifications may be given:

```
GRIDS Y  15  30  45 ,  GRIDS X  5  TO  25  BY  5 ;
```

3.12 Boxing Plots with EDGES

Additional axis lines or edges around the plot may be specified. Left and bottom edges are assumed. Additional edges may be requested for the right and top edges.

The phrase:

```
EDGES  TOP  BOTTOM  LEFT  RIGHT ;
```

places axes lines around all four sides of subsequent plots. This phrase may be abbreviated as follows:

```
EDGES  T  B  L  R ;
```

Spaces are necessary between the letters. Note that when EDGES is used, left and bottom edges are no longer assumed and must be explicitly specified if desired. Thus, the phrase:

```
EDGES  RIGHT  TOP ;
```

produces plots without left and bottom edges. The use of:

```
NO EDGES ;
```

results in plots without any edges, and without any tick marks and tick labels as well.

3.13 Reversing Axis Values

It is possible to reverse the order of the values on an axis:

```
      P  Temperature  *  Time.of.Day,  REVERSE  X ;
```

The low values for Time.of.Day (the X-axis variable) are now the high values and the high values are now the low ones. Any corresponding range, tick mark and grid line settings are also reversed. REVERSE Time.of.Day also accomplishes the same thing. REVERSE Y or REVERSE, which applies to both the X-axis and the Y-axis, may also be specified. This subcommand takes effect when the plot is produced. When SHOW SETTINGS is requested, the last line indicates which variables are reversed. Subsequent plots have the values on the specified axes reversed, unless either RESET REVERSE or NO REVERSE is used.

3.14 Changing the Plot Characters

It is possible to specify the characters to be used in plotting the data. Initially, the plotting characters are set as if this subcommand were given:

```
      CHARACTERS FOR PLOT  '*23456789$' ;
```

A “*” indicates one observation at a point on the plot, a “2” indicates two observations, and so on. A “$” indicates ten or more observations. The last character in the string is the “overflow” character; it is always used to indicate all observations greater than the next-to-last character.

NUMBERS may be used to request that the string of numbers including “1” be used as the plotting characters:

```
      CHARS FOR PLOT NUMBERS ;   or

      CHARS  '123456789$' ;
```

CHARACTERS may be abbreviated to CHAR or CHARS. The words FOR PLOT are optional when only one plot is specified.

Internal spaces, those within the single quote marks, are interpreted as blank plotting characters. Thus, the request:

```
      CHARACTERS FOR PLOT  '  34567890$' ;
```

sets a blank to indicate one or two observations at a point on the plot. In effect, only points with three or more observations are shown on the plot.

The phrase:

```
      CHARACTERS FOR PLOT  'ABCDEFGHIJKLMNOPQRSTUVWXYZ' ;   or

      CHARS FOR PLOT        ALPHA ;
```

sets “A” as the plotting character indicating one observation, “B” for two observations, and so on. A “Z” indicates 26 or more observations. ALPHABET or ALPHA are abbreviations for this string. The word LETTERS may also be used.

3.15 Changing the Character Scale

The SCALE option may be used when the counts of observations for plotted points are high and many points would have a “$”, indicating ten or more observations. It is also useful when several different counts are to be specified with the same symbol.

Either of the phrases:

```
      CHARACTERS FOR PLOT  '.....*****$'    ;   or


      CHARACTERS FOR PLOT  '.*$',  SCALE 5 ;
```

sets a "." for one to five observations, an "*" for six to ten and a "$" for eleven or more. "SCALE 5" sets five observations to each plotting character. The final character is always the character that is used for observations greater than the last specified value. A scale of 1 is assumed when SCALE is not used.

SCALE may be used with a fractional argument:

```
SCALE .5 ;
```

A fractional scale such as .5 sets the first plotting character to represent observations above zero through .5, the second for observations above .5 through 1, the third for above 1 through 1.5, and so on. This is generally desirable only when OBSERVE is used and values other than whole number counts are plotted. (OBSERVE is described later in this chapter.)

SCALE may be used either as a subcommand or as a modifier for CHARACTERS. If it is used by itself, it applies to the plot and all overlays:

```
CHARACTERS FOR PLOT '.*$',
CHARS      FOR OVER 'ab#'

SCALE 5,
```

In this example, both the plot and the overlay have their values scaled by 5. If it is used as a modifier following the CHARACTERS subcommand, it applies only to that single plot or overlay:

```
CHARACTERS FOR PLOT '.*$'  SCALE 5,
CHARS      FOR OVER

'ab#'  SCALE 3,
```

Therefore, in this case, the plot has a scale factor of 5 whereas the overlay has a scale factor of 3. SCALE remains in effect until it is changed or reset.

3.16 SINGLE VARIABLE PLOTS

One variable may be plotted by using the subcommand LINE.PLOT:

```
LINE.PLOT  Hours.Worked ;
```

The individual values are plotted sequentially, one value at a point, in the order that they are found in the input file. That order should therefore reflect a meaningful attribute. Generally, the order is that in which the data were collected, and a plot of the variable over a given time span results. Cyclical time fluctuations are thus readily visible. The input file may be sorted on the basis of some other variable before it is given to the PLOT command, if the initial order is not the one desired for the line plot.

Different types of cyclical variation may be detected. For example, on a plot of product defects, where a defect count is taken during each working shift, high and low numbers of defects are observed for certain shifts at certain times. Or, on a plot of product sales, where a total sales volume figure is collected each day, peak and low sales are observed for certain days and for holiday and non-holiday periods. An initial line plot of data may often reveal some cyclical fluctuation which may suggest additional plots of the data by other variables.

3.17 Alternative One-Variable Plots

When there are many cases (data items) in the input file, the line plot is wider than the output device. Thus, the plot continues on one or more pages as necessary. However, plotting multiple observations at a single point yields a one-page plot. For example, if the plot in Figure 3.5 had data from six months rather than one month, it would take multiple pages to print all of the information in a character plot. A similar plot in PostScript output can easily handle a year's worth of data on a single page. PostScript output is described later in this manual. Respecifying the plot in Figure 3.5 as a two-variable plot where Day is a variable with values 1 through 7 corresponding to Monday through Sunday, yields a single-page plot:

```
P  Hours.Worked  *  Day ;
```

Average daily variations are then readily visible.

Respecifying this plot as a single-variable plot yields a variant of the line plot:

```
P  Hours.Worked ;
```

The plot takes only one page even if there are many observations. The Y-axis contains the values of the variable Hours.Worked and the X-axis contains the values for the number of observations. The range of the X-axis depends on the number of observations (the number of cases) in the file. A large number of cases yields a compressed line plot. The range of the X-axis values is from 1 to N, where N is the number of cases in the file. The interval between tick marks is, of course, dependent on N. The HISTOGRAM command is another way of presenting data on one variable.

Figure 3.5 Producing a Line Plot for a Single Variable

```
     ----------The PLOT Command------------------------------


     PLOT  Jan83 ;
         LINE.PLOT  Hours.Worked,  TICKS  X  1  7 TO 28  BY  7 ;


     ----------The Output------------------------------------


               Line Plot of Hours.Worked  (Legend: *=1)

     7  +
 H      |  *                     *
 o   6  +                          **
 u      |          *
 r   5  +     *     *           *
 s      |      *  *      *  *  *        *
     4  +   **            **
 W      |           **
 o   3  +
 r      |
 k   2  +
 e      |
 d   1  +
        |
     0  +**     **     **    **   *  **
         +-----+------+------+------+------------------------
         1     7      14     21     28
```

3.18 MULTIVARIABLE PLOTS

A plot may present information on more than two variables. T his can be accomplished in several ways:

1. by plotting information about the observations of a third variable at the point designated by the values of the first and second variables (OBSERVE);
2. by overlaying the plot of two variables with additional plots (OVERLAY);
3. by specifying multiple plots of two variables sorted on the basis of another variable (BY).

3.19 Observations on Additional Variables

The subcommand:

```
P  Distance.to.Work  *  Occupation.Code ;
```

yields the typical plot where the number of observations for each combination of values is plotted. By specifying the observation of some aspect about a third variable to be plotted, additional information may be presented on the same plot. For example, the subcommand:

```
P  Distance.to.Work  *  Occupation.Code,
   OBSERVE  Meals.at.Home  TOTAL,
   CHARACTERS FOR PLOT  '1234567890$',  SCALE 5 ;
```

plots the total of the number of meals eaten at home for each unique combination of distance to work and particular occupation. This type of plot indicates that the teachers in this sample, working ten miles from home, eat a total of 46 to 50 meals at home, whereas the accountants in this sample, working ten miles from home, eat a total of 31 to 35 meals at home.

These totals reflect the unit of measurement involved (perhaps it is meals eaten at home per five-day work week), as well as the number of people in the sample in each particular combination of variables, which should reflect the actual population. When this is not the case, it makes more sense to plot the average number of observations, rather than the sum of the observations. By using the word MEAN in the OBSERVE phrase:

```
P  Distance.to.Work  *  Occupation.Code,
   OBSERVE  Meals.at.Home  MEAN,
   CHARACTERS FOR PLOT  '12345678$',  SCALE 2 ;
```

the mean number of meals eaten at home, rather than the total of the number of meals, is plotted for each combination of distance to work and occupation. Such a plot indicates that teachers in this sample, working ten miles from home, eat an average of 9 to 10 meals at home per five-day work week, whereas accountants, working ten miles from home, eat an average of 7 to 8 meals at home per five-day work week.

The plot does not reflect the number of people in each combination of variables, but reflects the unit of measurement of meals eaten at home. The plotting characters should be chosen with this in mind. Plotting characters that cover the range of the expected sums or averages must be specified. Otherwise, the initial character set ‘*23456789$’ with a scale of 1 is used. If there are many observations of ten or more, they all appear with a “$” .

The SCALE for the plotting characters may be fractional:

```
P  VAR1  *  VAR2,  OBSERVE  VAR3  MEAN,
   CHARS  NUMBERS,  SCALE .5 ;
```

This sets the “1” as the plotting character for observations above zero through .5, the “2” for observations greater than .5 through 1, the “3” for observations greater than 1 through 1.5, the “4” for observations greater than 1.5 through 2, and so on. A fractional scale makes sense only when OBSERVE is being used with numeric data. With OBSERVE, the observations are values which may be fractional. In other situations, the observations plotted are counts — whole numbers.

3.20 OBSERVE Options

There are additional options that may be used with OBSERVE to provide different information about a third variable on a plot of two variables. All of these options are summarized as follows:

1. TOTAL plots the total value of all of the observations at each point on the plot;
2. MEAN plots the mean value of all of the observations at each point on the plot;
3. MINIMUM plots the minimum value of the observations at each point on the plot (may be shortened to MIN);
4. MAXIMUM plots the maximum value of the observations at each point on the plot (may be shortened to MAX);
5. SINGLE plots the value of all single observations at each point on the plot and plots a “$” at all points on the plot where there are multiple observations;
6. ALIKE plots the value of all observations (single and multiple) at each point on the plot when those values are all alike and plots a “$” at all points on the plot where there are multiple values that are not alike.

When none of these options is specified, TOTAL is assumed when the data are numeric. ALIKE is assumed when the data are character items. The plotting characters are always single digits or characters. They will be either: 1) the initial characters of the values of character variables, or 2) the user-specified or default CHARACTERS FOR PLOT, using either the initial scale of 1 or a user-specified scale. When the initial character set and scale are not of adequate size for the data, suitably scaled plotting characters should be specified or the loss of information occurs. For example, when MINIMUM is specified, many minimum values may be represented by a “$” if most of the data values are larger than 10.

3.21 OBSERVE with Character Information

Not all of the options for OBSERVE make sense with a third variable that presents character information or that has character values. The options that are sensible are SINGLE and ALIKE. The other options involve numerical calculations that cannot be performed on character data.

An example of a plot of two variables with character information about a third numeric variable is:

```
P  Verbal.SAT  *  Math.SAT,
   OBSERVE  Grade  ALIKE,
   CHARS FOR PLOT  '        FSJR' ;
```

This produces the plot in Figure 3.6 of students’ verbal SAT scores by math SAT scores, with a character at each point indicating the grade of the student. An “F” indicates a freshman, an “S” a sophomore, a “J” a junior, and an “R” a senior. A “$” indicates a math-verbal combination scored by multiple students in different grades, since ALIKE was the specified OBSERVE option.

The data in the input file for the variable Grade are numeric — in the form of the numbers 9 through 12. Thus, it is necessary to specify eight blanks in the CHARACTERS FOR PLOT phrase to indicate plotting characters for Grade values 1 through 8. Otherwise, the PLOT command assumes that “F” is the plotting character for “1”, “S” for “2”, and so on.

If Grade is a character variable, and the data in the input file for this variable are in the form of character data, then it is not necessary to specify CHARACTERS FOR PLOT. The initial letter of each value, or the single letter if each value is a single letter, is used to indicate that value on the plot. “F” indicates the character value “Freshman”. However, “S” would indicate both the character values “Sophomore” and “Senior”. The value “Senior” could be entered in the file as “Renior” to obtain a unique initial letter. However, it is easier to recode the character data values for Grade to unique single characters before inputting the file to PLOT (using the PPL programming language), or to recode the character values to consecutive integers (using the MAP command).

Figure 3.6 Showing a Third Variable with OBSERVE

```
     ---------The PLOT Command------------------------------

     PLOT  Scores ;
      P  Verbal.SAT  *  Math.SAT,  OBSERVE  Grade  ALIKE,
     CHARACTERS FOR PLOT  '        FSJR' ;


     ----------The Output-----------------------------------

          Plot of Verbal.SAT by Math.SAT (Legend: F=9, S=10, ..., R=12+)
                  Observations are Like values of variable Grade

         |                               R
     755 +                                          J J        R
V        |                                                   J     J
e    700 +                                             J  J
r        |
b    645 +                             J                              R
a        |                         R                               J
l    590 +                    R R$
         |                    R                                    J
S    535 +               $R               R                 J
A        |    F R     $$ R   R         R          S
T    480 +       J $S    RR R             F  R
         | F    J     JJ
     425 +      R     J
         |                       J              R
     370 +F                               J  J
          +----+----+----+----+----+----+----+----+----+----+----+----+
         380       450       520       590       660       730       800
                                     Math.SAT
```

CHARACTERS FOR PLOT may not be specified for character data values being observed. The initial (or only) character of the value is plotted. However, a character for multiple or unlike observations may be defined by specifying a character set to replace the assumed one where the “$” is the overflow character. If this specification:

```
      CHAR   '?'   ;
```

is given, the “?” is used to indicate multiple or unlike observations. This is the case only when a character variable is being observed (with either the ALIKE or SINGLE option). When a numeric variable is being observed, a full character set is necessary, in addition to the overflow character.

3.22 Overlaying Plots

A plot may be overlayed by up to nine additional plots. The plots must make use of the same axes. This subcommand produces plots of Opacity by Coverage for two paint mixtures:

```
      P        Opacity.1  *  Coverage.1,
      OVERLAY  Opacity.2  *  Coverage.2 ;
```

They are on the same axes, labeled Opacity.1 and Coverage.1 (unless LABELS is used to provide other labels) and the ranges are scaled to include all values for these variables. It is then possible to note differences between the two paint mixtures.

Figure 3.7 Distinctly Clustered Plots Shown with Overlays

```
----------The PLOT Command------------------------------------

 PLOT  Defects ;

P        Defect.Count.AM  *  Time.AM,  CHAR FOR PLOT   '*',
OVER 1   Defect.Count.PM  *  Time.PM,  CHAR FOR OVER 1 '*',
OVER 2   Defect.Count.N   *  Time.N,   CHAR FOR OVER 2 '*',

TICKS    X  1 9 17 24,       GRIDS  X  9 17 ,
LABEL    Y  'Defect Count',  LABEL  X  'Time' ;
```

```
     ----------The Output-----------------------------------------

           Plot of Defect.Count.AM by Time.AM  (Legend: *=1 or more)
           Overlay Defect.Count.PM by Time.PM  (Legend: *=1 or more)
           Overlay Defect.Count.N  by Time.N   (Legend: *=1 or more)

   60 +                     .                    .                 *
D     |                     .                    .
e  50 +                     .                    .          *    *
f     |                     .                    .
e  40 +                     .                    .    *  *     *
c     |                     .                    .
t  30 +                     .                    .*    2
      |                     .                    .
C  20 +        *        * * .   *  *          *  *
o     |                     .                    .
u  10 + * *  *  *    *      .*         *   *   * .
n     |                     .                    .
t   0 +           *         .        *           .
       +--------------------+--------------------+--------------------+
       1                    9                   17                   24

                                    Time
```

Differently named variables may be specified, but they should be measured in the same units since only one set of axes is used. Thus, the request:

```
P     Math.Test     *  Period,
OVER  English.Test  *  Class ;
```

is meaningful only if both Math and English are measured in like units, such as 0 through 100, and if Period and Class are also measured in like units, such as 1 through 8. If the units are not the same, a plot with ranges encompassing all values is produced. OVERLAY may be shortened to OVER.

The axes of a plot with overlays are labeled with the names of the variables specified in the P subcommand. More suitable names may be supplied with the LABEL subcommand (which is discussed near the end of the chapter.) Figure 3.7 shows a plot with two overlays.

3.23 Character Sets for Overlays

When plotting characters are not specified, the initial character set is used for the plot. The character sets for the overlays are as if these specifications were given:

```
CHAR FOR OVERLAY 1 'A',
CHAR FOR OVER    2 'B',
CHAR FOR OVER    3 'C',
```

and so on. Different plotting characters may be specified for the overlays and/or the plot. The subcommand

```
P     Opacity.1  *  Coverage.1,  CHAR FOR PLOT  '*',
OVER  Opacity.2  *  Coverage.2,  CHAR FOR OVER  '+' ;
```

produces a plot of Opacity by Coverage, using only a “*” as the plotting character for the plot and a “+” as the plotting character for the overlay. This makes it possible to distinguish between the two paint mixtures, even though both sets of points may cluster relatively close together. The FOR is optional after the word CHAR.

When there is one point with values from both the plot and the overlay — that is, a point where both a “*” and a “+” go, the plot character (“*”) is used. PLOT characters take precedence over all OVERLAY characters. Similarly, OVERLAY 1 characters takes precedence over OVERLAY 2 characters, and so on. Therefore, when both the plot and the overlay cluster close together and share many points in common, it may not be possible to distinguish between them. Differences indicated by the overlay character, however, are clearly discernible.

Note that although there is only one character in the default set for overlays, multiple character sets may be specified for use in both the plot and the overlays. There may be up to nine overlays per plot. However, when there are multiple overlay requests, they must be followed by a number and the corresponding character specifications:

```
P       Temp.1  *  Dosage.1,  CHAR  PLOT    'aA',
OVER 1  Temp.2  *  Dosage.2,  CHAR  OVER 1  'bB',
OVER 2  Temp.3  *  Dosage.3,  CHAR  OVER 2  'cC' ;
```

The ranges of the X-axis and Y-axis are scaled automatically to include all values of the plot and the overlays.

3.24 Overlays of Distinct Plot Clusters

When multiple variables use the same units of measurement (that is, the same axes) and yield distinct plot clusters, it makes sense to have multiple overlays and possibly to use OBSERVE. Multiple overlays can show repeated measurements over time. This subcommand produces the plot shown in Figure 3.7:

```
P         Defect.Count.AM  *  Time.AM,  CHAR FOR PLOT   '*',
OVER 1    Defect.Count.PM  *  Time.PM,  CHAR FOR OVER 1 '*',
OVER 2    Defect.Count.N   *  Time.N,   CHAR FOR OVER 2 '*',

  TICKS    X  1 9 17 24,       GRIDS  X  9 17 ,
  LABEL    Y  'Defect Count',  LABEL  X  'Time' ;
```

Defect counts, measured at nine different times during three different factory shifts (morning, afternoon and night), are shown on one plot. Grid lines demarcate the shifts, whose times are based on a 24-hour clock.

Figure 3.8 illustrates a plot of the same data but with a different organization. The data differs from that in Figure 3.7 in these ways: 1) numbers 1 through 9 rather than specified times are recorded with each defect count, 2) the particular shifts are entered as character variables with the values AM, PM and N, and 3) a duplicate count is made by a second inspector.

In Figure 3.8, a plot with an overlay showing observations of single values of the shifts is requested.

Figure 3.8 Using OBSERVE with Plot Overlays

```
      ---------The PLOT Command--------------------------------

      PLOT  DefMay (IF  Date = 7,  RETAIN ) ;

       P        Defect.Count.One  *  Measurement.No,
      OVERLAY  Defect.Count.Two  *  Measurement.No ,
      OBSERVE  Shift  SINGLE,    LABEL  Y  'Defect Count' ;
```

```
     ----------The Output--------------------------------------

Plot of Defect.Count.One by Measurement.No (Legend:$=Multiple observations)
Overlay Defect.Count.Two by Measurement.No (Legend:$=Multiple observations)
                Observations are Single values of variable Shift

    60 +                                                                N
 D     |
 e  50 +                        N               N               N       N
 f     |
 e  40 +N       $       N               N       N       $       N
 c     |
 t  30 +N               N       N       N                               P
       |
 C  20 +A       $       P       A               P       P       A       $
 o     |
 u  10 +$       A       $       $       $       $       $       $       A
 n     |
 t   0 +P       A       P       P       A       A               P
        +-------+-------+-------+-------+-------+-------+-------+-------+
        1       2       3       4       5       6       7       8       9

                              Measurement.No
```

3.25 Multiple Plots Sorted BY Additional Variables

BY requests plots of variables sorted on the basis of one to fifteen numeric or character variables. The BY option must be specified at the command level as the file is input to the PLOT command:

```
            Enter a command:
       >>   PLOT  Scores,  BY  Grade ;
```

The input file to PLOT must already be in sort order on the variable Grade.

The PLOT command treats each subgroup of the file separately. Plots are produced for each subgroup. This subcommand request:

```
          Enter subcommand:
     >>   P  Verbal.SAT  *  Math.SAT,
          CHARS FOR PLOT    '*234567890$',  SCALE 2 ;
```

produces four plots, one for each of the values of Grade (9 through 12). The first plot contains verbal and math scores only for freshmen, the second contains scores only for sophomores, and so on. Comparisons among the plots may easily be made, particularly because the ranges of the axes are identical. A summary plot, containing all the scores, may also be produced by turning off BY with BY OFF or NO BY:

```
     AGAIN,  NO BY ;
```

BY remains in effect for subsequent plot and line plot requests unless it has been turned off. RESET BY turns it on again.

BY assumes a maximum of 12 groups. When there are more than 12 groups, the identifier MAX.BY must be used in addition to the BY identifier. MAX.BY is followed by an integer indicating the number of groups:

```
     PLOT  Class,  BY  Grade,  MAX.BY 13;
```

Plots are produced for each of the 13 subgroups of Class.

3.26 BY and Subcommands

Other subcommands may be requested in conjunction with BY. The subcommands LINE.PLOT, OBSERVE and OVERLAY, as well as all the subcommands relating to plot settings, are all legitimate PLOT requests after a BY specification. However, the number of plots produced by subcommands is multiplied by the number of distinct values of the BY variable. As a result, many plots may be produced.

SUMMARY

PLOT

```
PLOT  Paint,  TITLES ;

  P  Viscosity  *  Coverage ;
  P  Cost       *  Coverage ;
  $

PLOT  Tests,  BY  Sex ;
  P  SAT.1  *  SAT.2        $
```

The PLOT command produces scatter plots of numeric variables. Desired plots and options are specified at the subcommand level. Options include overlays, observation of additional variables, user-defined character sets, grid lines, ticks, tick labels, axis labels and titles. Plots may be requested for subgroups using BY. Character variables may be observed or may be BY variables.

Required:

PLOT fn

provides the name of the P-STAT file that contains the input data.

Optional Identifiers:

BY vn

requests plots of variables sorted or grouped on the basis of one to twelve additional variables. This option is given as part of the PLOT command:

```
PLOT  Speed,  BY  Sex ;
```

The input file, Speed, must already be sorted or grouped on the BY variable Sex. Since Sex has two values, two plots are produced for all subsequent subcommands (such as PLOT or LINE.PLOT). BY may be turned off at the subcommand level with BY OFF and may be turned on again with BY ON or RESET BY.

MAX.BY nn

specifies the maximum number of BY variables in the file when the number is greater than 12. MAX.BY must be used in addition to the BY identifier and is followed by an integer indicating the number of groups in the file:

```
PLOT  Class,  BY  Grade,  MAX.BY 13 ;
```

Plots are produced for each of the 13 subgroups of Class. When MAX.BY is not specified, a maximum of 12 groups is assumed.

TITLES

specifies that any previously defined titles are to be used on the plots produced by the PLOT command:

```
PLOT  Scores,  TITLES ;
```

(See the TITLES chapter in "P-STAT: Introductory Manual" for additional information about defining titles using the TITLES *command*.) The *subcommand* TITLES may be used to define titles within PLOT.

Optional Subcommands:

P vn * vn

specifies the variable or variables to be plotted. The first variable is plotted on the Y-axis and the second is plotted on the X-axis. If only one variable is specified, it is plotted on the Y-axis, and position in the file (case number) is plotted on the X-axis.

Ranges or lists of variables may be substituted for either or both of the variables.

```
P  Test.A   TO   Test.G  *  Class.Rank ;
P  Calories  *   Height  Weight  ;
```

The total number of plots produced is the product of the number of variables to the left of the “*” multiplied by the number of variables to the right. If the identifier BY has been used, the total number of plots is multiplied by the number of values of the BY variable.

AGAIN

repeats the current plot request. This may be followed by modifying subcommands:

```
AGAIN,  OVERLAY  Temperature  *  Reading.No ;
```

CHARACTERS FOR PLOT 'cs'

specifies a string of characters, or a single character, to be used in plots to indicate the number of values at each point. The initial character set, used when this subcommand is not given, is:

```
CHARACTERS FOR PLOT  '*23456789$' ;
```

The “*” represents one observation at a given point, the “2” represents two observations, and so on. The “$” represents ten or more observations at a point. The last character in a string (the “overflow” character) is used for observations greater than the delineated ones. The word FOR is optional.

NUMBERS may be used to request that the string of numbers including “1” be used as the plotting characters:

```
CHARS PLOT  NUMBERS ;    or
CHARS '123456789$'  ;
```

An alphabetic character set may be specified in several equivalent ways:

```
CHARS FOR PLOT  'ABCDEFGHIJKLMNOPQRSTUVWXYZ' ;  or
CHARS     PLOT   ALPHABET ;  or
CHARS            LETTERS  ;
```

Characters may also be specified for different types of plots:

```
CHAR  LINE.PLOT  '+' ;
CHAR  OVERLAY 1  'AB' ,  CHAR  OVER 2  'ab' ;
```

A scale may be given:

```
CHAR FOR PLOT  '1234567890$'  SCALE 2 ;
```

This sets each plotting character equal to two observations.

COLUMNS nn

sets the number of columns, the horizontal distance, available per plot. If this is not explicitly set, the number of columns is determined by the output width of the current output destination — generally 80 columns for a terminal and 132 columns for a disk file or printer. If output width has been defined prior to entering PLOT (using the command OUTPUT.WIDTH), that definition is used.

EDGES T B L R

specifies axes lines around the sides of plots. Left and bottom edges are assumed. Four edges are possible — top, bottom, left and right. Note that when EDGES is used, left and bottom edges are no longer assumed. Thus, this specification:

```
EDGES  TOP  RIGHT ;
```

produces edges only on the top and right sides.

EXACT RANGE FOR Y

specifies that the range boundaries given for Y (or X) should also be used for the low and high tick marks. Typically, both RANGE and TICKS are defined, and EXACT is used as well:

```
P  Score * Factor,  RANGE X  12.5 TO 21.5,
TICKS X  12.5  15.5  18.5  21.5,  EXACT X ;
```

The words RANGE and FOR may be omitted after EXACT.

GRID.LINES Y nn

specifies that grid lines be placed on the plot at Y = nn (the number specified). The variable name may be used in place of Y if desired. Grids may be requested for either axis and at multiple positions:

```
GRID.LINES  Hours  2  4  8  ,
GRID.LINES  Y      10  TO  100  BY  20  ;
```

GRID.LINES may be abbreviated to GRIDS. Diagonal grid lines may be specified at X = Y and/or X = -Y:

```
GRIDS  ASCENDING  DESCENDING ;
```

Grid lines at X = 0 and/or Y = 0 may be requested:

```
GRIDS  VERTICAL   HORIZONTAL ;
```

ASCENDING, DESCENDING, VERTICAL and HORIZONTAL may each be shortened to their initial letter.

HELP sub

requests help about the subcommand specified. HELP (by itself) gives general information about PLOT and lists the topics for which more extensive help is available.

LINES nn

sets the number of lines available per plot. If this is not explicitly set, the number of lines available is either the screen size of the terminal if it has been set prior to entering PLOT, or the page size of the current output destination.

LINE.PLOT vn

specifies a single-variable plot. The specified variable is plotted on the Y-axis. Observation or case number is plotted on the X-axis. The plotting character, which is the “*” unless another character is specified, always represents one measurement. When there are many cases in the file, the plot continues on subsequent pages.

NO sub

turns off a specified subcommand or a list of subcommands. For example,

```
NO LEGEND ;
```

suppresses the printing of the legend provided by the PLOT command.

```
NO TICKS EDGES ;
```

suppresses the marking of the X-axis and Y-axis into even intervals and the printing of edges around the plot. NO may be used with these subcommands: BY, EDGES, GRID.LINES, LEGEND, OBSERVE, OVERLAY, RANGE, REVERSE, TICK.LABELS, TICKS and TITLES.

OBSERVE vn

specifies a variable whose values are plotted as observations on a plot of other variables. For example:

```
P  Height  *  Weight,  OBSERVE  Calories ;
```

plots the total value of the variable Calories at each point defined by the values of the variables Height and Weight. If a character variable such as Sex is observed, like values of Sex are plotted at the points indicated by the values of Height and Weight:

```
P  Height  *  Weight,  OBSERVE  Sex  ;
```

A “$” indicates differing values at one point.

When numeric variables are observed, the actual values are plotted when those values are 9 or less. A “$” is plotted for values of 10 and over. Other plotting characters and a scale may be specified. When character variables are observed, the first character of the character value is plotted — that is, F for Female and M for Male in the case of the variable Sex.

The TOTAL option is assumed when the OBSERVE variable is numeric, and the ALIKE option is assumed when it is character. Other options are MEAN, MINIMUM, MAXIMUM and SINGLE. Options are specified as part of the subcommand:

```
P  MPG  *  Weight,
   OBSERVE  Cost  MEAN,  SCALE  1000 ;
```

OVERLAY vn * vn

specifies an overlay plot to be placed on the same axes as the currently specified plot. Up to nine overlays may be specified. The initial character set for the overlays is:

```
CHARACTERS FOR OVERLAY 1 'A' ;
CHARACTERS FOR OVERLAY 2 'B' ;
```

Another character set with one or more characters may be specified, as it may not be possible to distinguish the points on the plot from those on the overlays, especially when there are two or more overlays. Unique single or multiple plotting characters may be specified for each overlay. This command specifies new single plotting characters:

```
P  Yield.CornA  *  Fertilizer,
   OVERLAY 1  Yield.CornB  *  Fertilizer,
   OVERLAY 2  Yield.CornC  *  Fertilizer,

   CHAR  FOR  PLOT        'a',
   CHAR  FOR  OVERLAY 1   'b',
   CHAR  FOR  OVERLAY 2   'c' ;
```

OBSERVE may also be used with OVERLAY.

PR fn

changes the output print destination. The terminal is the assumed interactive output destination and the printer is the assumed batch output destination. PR is an abbreviation of PRINTER.

RANGE Y nn TO nn

specifies a range of values for X and/or Y to be plotted. A variable name may also be used:

```
RANGE  Weight  60 TO 200 ;
```

When a range is not explicitly set, all pairs of values for X and Y are plotted. This may not be satisfactory when consistent ranges are desired for a series of plots, or when outlying points necessitate a large range that is only sparsely filled.

The subcommand EXACT RANGE FOR Y (or X) may be used, along with RANGE and TICKS, to force the RANGE and the low and high tick marks to coincide.

RESET sub

changes the plot settings of a specified subcommand or list of subcommands back to the initial settings:

```
RESET  LINES ;    or

RESET  EDGES  LEGEND ;
```

RESET by itself, resets *all* subcommand settings. INITIALIZE is a synonym for RESET.

REVERSE X

reverses the order of the values on the X-axis (and/or Y-axis). Low values become the high values, and vice versa. Any corresponding GRID.LINES, LABELS, RANGE, TICKS, and TICK.LABELS settings are also reversed. Subsequent plots also have the values on their axes reversed, unless RESET REVERSE or NO REVERSE is used.

SCALE nn

sets each plotting character equal to the specified number of observations:

```
SCALE 10 ;
```

This SCALE sets each character equal to 10 observations. Thus, a plotting character of 3 represents from 21 through 30 observations. SCALE is useful when OBSERVE is used with TOTAL or MAXIMUM and the values observed are high.

SHOW SETTINGS

displays the plot settings currently in effect. SHOW S is an abbreviation for SHOW SETTINGS. (See RESET for resetting plot settings.)

SHOW VARIABLES

displays the names of the variables in the current file input to PLOT. These are variables that may be used in plots. This may be abbreviated to SHOW VARS or SHOW NAMES (or just SHOW).

TICK.LABELS X 'cs' 'cs' 'cs'

supplies a list of labels to replace the tick marks (numbers) on the specified axis. The order of the labels corresponds to the order of the tick marks — the first label replaces the first tick mark, and thus tick marks without labels are not shown. Any labels which are not legal P-STAT names must be enclosed in quotes. The name of the variable may be specified instead of X or Y:

```
TICK.LABELS  Year  '1984' '1985' '1986'
                   '1987' '1988' ;
```

TICKS X nn

specifies a location or locations where particular tick marks are desired on the X-axis (and/or Y-axis). Tick marks may be specified for one or both of the axes and the axes may be designated by name. If the same tick marks are desired on both axes, they may be specified with one phrase:

```
TICKS  5  TO  30  BY  5  ;
```

VERBOSITY nn

changes the verbosity level, which controls the amount of printout produced. Setting verbosity to four:

```
VERBOSITY 4,
```

causes subcommands to be echoed as they are processed.

4
PLOT:
Enhancing the Output

The previous chapter described the basic PLOT identifiers and subcommands

his chapter describes the use of titles and labels to enhance the plot output and it provides some additional examples of plot characters. The next chapter describes the use of PLOT to produce camera-ready output on a PostScript printer. PLOT output in an XWindow is described later in the manual.

4.1 PLOT TITLES AND AXIS LABELS

Titles and axis labels may be specified. Titles may be defined with the TITLES command and with PLOT subcommands. Titles precede the legend that PLOT normally prints as a plot heading. Axis labels replace the variable names which PLOT uses to label the axes. The information in a labels file can be used to supply axis labels, to enhance the legend, and to provide tick marks and tick labels.

4.2 Plot Titles

The legend supplied by the PLOT command often serves in place of a user-supplied title. It names the variables plotted and gives a brief explanation of the characters used in the plot. The legend may take several lines when OVERLAY and OBSERVE are used. However, sometimes one or more titles may be more suitable than the legend. Titles may be defined outside the PLOT by using the TITLES command (“permanent” titles) or within the PLOT command using the TITLES subcommand (“temporary” titles). Either way, the titles are specified in a similar manner.

Up to nine lines of top titles and three lines of bottom titles may be specified within the PLOT command. The title text must be enclosed in single (or double) quotes. The top titles are centered above the legend supplied by PLOT. The bottom titles are centered below the plot. The titles entered as subcommands within PLOT cannot exceed 80 characters. This subcommand defines a title for the plot requested:

```
P  Yield * Fertilizer,  RANGE Yield 10 TO 70,
   TITLES  'Crop 711 - July, 1982' ;
```

Defined titles are in effect for all subsequent plot requests. To provide different titles for other plots, each of these plots must be specified with their respective titles:

```
P  Yield  *  Fertilizer.2,
   TITLES  'Crop 812 - August, 1982' ;
```

When multiple titles are defined, the subcommand TITLES must precede *each* line of titling:

```
AGAIN,  TITLES T2 LEFT  'First String' ,
        TITLES T2 RIGHT 'Second String' ;
```

The subcommand:

```
NO TITLES ;
```

turns off all titles on subsequent plots. Any titles defined within the PLOT command are temporary — they disappear upon exiting from PLOT.

Figure 4.1 User-Specified Titles

```
----------The Commands--------------------------------------

  TITLES  T1  'Work Station 411 - May 7, 1982' ,
          T2  'Defect Counts Recorded by A.Smith, R.Williams' ,
          T3   BLANK ,
          B1   '(Asterisks(*) indicate multiple observations.)' $

  PLOT   DefMay  (IF  Date = 7,  RETAIN ) ;

   P      Defect.Count.One * Measurement.No,
  OVER   Defect.Count.Two * Measurement.No,
  OBSERVE  Shift  SINGLE,  NO LEGEND,  LABEL Y  'Defect Count',

  TITLES T4
    'Plot of Defect Count by Measurement Number - Showing Shift' ;
  $
```

```
----------The Output--------------------------------------

                    Work Station 411 - May 7, 1982
             Defect Counts Recorded by A.Smith, R.Williams

        Plot of Defect Count by Measurement Number - Showing Shift

   6 +                                                                N
D    |
e  5 +                        N               N               N       N
f    |
e  4 +N       *       N               N       N       *       N
c    |
t  3 +N               N       N       N                       P
     |
C  2 +A       *       P       A               P       P       A       *
o    |
u  1 +*       A       *       *       *       *       *       *       A
n    |
t  0 +P       A       P       P       A       A               P
      +-------+-------+-------+-------+-------+-------+-------+-------+
      1       2       3       4       5       6       7       8       9

                             Measurement.No

              (Asterisks(*) indicate multiple observations.)
```

If permanently defined titling is desired, the TITLES command (described in “P-STAT: Introductory Manual”) may be used before the PLOT command. Either the identifier TITLES or a TITLES subcommand must be used in the PLOT command for titles to appear in the output.

Any of the nine top or three bottom titles can be supplied as subcommands. LEFT, RIGHT and CENTER may be specified. T1 is assumed if the title number is not supplied. CENTER is assumed when the position is not given.

The subcommand phrase:

```
      NO LEGEND ;
```

suppresses the printing of any program-supplied plot legends. If there are no titles and NO LEGEND is used, the plot appears with no heading information of any type. To restore the legend:

```
      RESET  LEGEND ;
```

may be used.

Figure 4.1 illustrates using titles. At the end of the PLOT command, the fourth top title is no longer available because it was defined as a PLOT subcommand. The other titles remain in effect and can be used in other commands where the TITLES identifier is supported.

4.3 Labels for the Axes

Labels may be specified to replace the axis labels provided by the PLOT command. The labels are provided individually for each axis:

```
      LABEL  FOR  Y  'Yield in Bushels/Acre',
      LABEL       X  'Fertilizer in Pounds/Acre' ;
```

They are enclosed in single (or double) quotes. Either LABEL or LABELS may be used, and the word FOR is optional.

LABEL is only in effect for the current plot, regardless of whether or not it is a separate subcommand. Labels are centered within the plot edges. If the labels are longer than these edges, the label is truncated from the right. The maximum length for axis labels is 80 characters. If these labels are supplied, they take precedence over any labels found in the labels file described below.

4.4 Using a Labels File

The PLOT command can make use of both extended variable labels and the value labels in a labels file. The labels file is supplied by using the LABELS identifier and providing the name of the file containing labels. The first extended variable label is used in the plot legend. If there is a second extended variable label for either of the plot variables, it is used to label the axis. Note: the LABELS subcommand takes precedence over extended variable labels.

If either the X or Y variable for the plot has value labels, they are used to determine the tick marks and the tick labels unless the TICK subcommand has been explicitly used.

The labels file "flow.lab" which is used in Figure 4.2 contains the following entries

```
      Var1  XL 'Flow’
               'Amt. of Flow' /
      Var3  XL 'Discharge'
                'Amt. of Discharge’  /
      Var4  XL 'Quarter'
               'Quarterly Measurements'
            (2) 1994  (6) 1995  (10) 1996 (14) 1997 /
      Plot.Labels (1) ' ' /
```

Variable Var4 has 2 extended variable labels and value labels. The first XL “Discharge” is used in the plot legend. The second XL “Quarterly Measurements” is used to label the X axis. The value labels are used to provide the location for the ticks and the labels. Thus there are 4 ticks on the X axis at 2, 6, 10 and 14 which are labelled with the years 1994, 1195, 1996 and 1997. Because of space limitations on the axes, only the first 12 characters of the value labels are used for tick labels.

Figure 4.2 Labels File and RESCALE

```
  ----------The PLOT Command-------------------------------------

  PLOT Flow,  LABELS 'flow.lab';
     LINES 30, OW 70,
     EDGES B L R,
     RESCALE OVERLAY 1,
     P Var1 * Var4,   OVERLAY Var3 * Var4,
     CHAR PLOT '+',   CHAR OVERLAY 1 'O';
$

  -----------The Output-----------------------------------------

              Flow by Quarter (Legend: + =1 or more)
          Overlay Discharge by Quarter (Legend: O =1 or more)

    .0150 +                                       +     O  O+ 10
          |                                    +            |
    .0148 +                                          O      + 9     A
          |                                                 |       m
A   .0146 +                                       O         + 8     t
m         |                                                 |       .
t   .0144 +                                                 |
.         |                                 O  O            + 7     o
    .0142 +                                                 |       f
o         |            +              O  O                  + 6
f   .0140 ++  +  +                 +  +              +  +   |       D
          |                     +  O     +                  + 5     i
F   .0138 +         +     +                 +              +|       s
l         |                  O  O                           + 4     c
o   .0136 +                                                 |       h
w         |                  +                              |       a
    .0134 +            O  O                                 + 3     r
          |                                                 |       g
    .0132 +      O  O                                       + 2     e
          |                                                 |
    .0130 +O  O                                             + 1
           ---+-----------+-----------+-----------+---------
             1994        1995        1996        1997
                        Quarterly Measurements
```

Plot.Labels is a pseudo-variable which must not exist in the data file if the Plot.Labels feature is desired. Plot.Labels provides substitute text for the “hard-wired” text usually printed by the PLOT command. In Figure 4.2 the plot legend contains:

```
Flow by Quarter             rather than
Plot of Flow by Quarter
```

because the use of Plot.Labels in the labels file provides an empty string as a substitute for the text usually associated with a code of 1.

The numeric codes and the text that is replaced are:

```
(1) plot of      (11) BY:                (21) totals
(2) line plot    (12) by                 (22) means
(3) overlay      (13) continued          (23) minimum values
(4) legend       (14) or more            (24) maximum values
                 (15) to                 (25) single values
                 (16) observations are   (26) like values
                 (17) of variable        (27) unlike observations
                                         (28) multiple observations
```

Figure 4.3 **A PLOT That Needs RESCALE**

```
                   Flow by Quarter (Legend: + =1 or more)
           Overlay Discharge by Quarter (Legend: O =1 or more)

        10 +                                             O  O+
           |                                                 |
         9 +                                          O      +
           |                                                 |
 A       8 +                                       O         +
 m         |                                                 |
 t       7 +                                 O  O            +
 .         |                                                 |
         6 +                           O  O                  +
 o         |                                                 |
 f       5 +                        O                        +
           |                                                 |
 F       4 +                  O  O                           +
 l         |                                                 |
 o       3 +            O  O                                 +
 w         |                                                 |
         2 +      O  O                                       +
           |                                                 |
         1 +O  O                                             +
           |                                                 |
         0 ++  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  ++
            ---+-----------+-----------+-----------+---------
              1994        1995        1996        1997
                         Quarterly Measurements
```

4.5 RESCALING THE OVERLAY

Figure 4.2 also illustrates the use of RESCALE. RESCALE is used when the plot and one or more of the overlay variables have ranges which differ so greatly that the information in one or more variables is masked by the range of the other variables. This is illustrated in Figure 4.3 which is identical to Figure 4.2 except for the absence of a RESCALE subcommand. The Y variable of the overlay has a range of 1 through 10 while the Y variable of the plot has a range from .013 to .015 . As you can see in Figure 4.3, the pattern of the plot values is totally hidden when both variables are combined to get an overall range.

When RESCALE is used, the Y variable for the PLOT has ticks and labels on the left edge of the plot as usual. Any overlays that are rescaled have their ticks and labels on the right edge of the plot. RESCALE is used to specify which overlays are to be scaled on the right edge of the plot. The base plot and any overlays that are not specified with RESCALE are scaled together on the left.

The following 3 RESCALE statements are equivalent:

```
RESCALE OVERLAY 1
RESCALE 1
RESCALE
```

To combine more than one overlay in the RESCALE use:

```
RESCALE OVERLAY 2 4 6    or just
RESCALE 2 4 6
```

The scaling is done by taking the lowest and highest non-missing values for the plot Y variable and the Y variables for overlays that are not mentions by RESCAL and combining their values to determine the ticks on the left and the placement of their values in the plot. The lowest and highest non-missing values found for the combined Y variables in the RESCALE list determine the scale on the right. Given the following 4 variables and a plot with 3 overlays:

```
Variable     Low    High

Var1          .05     3.5.
Var2        30.00    75.0
Var3         2.50     6.0
Var4        25.00    60.0

PLOT      Var1 * Var5
OVERLAY 1 Var2 * Var5
OVERLAY 2 Var3 * Var5
OVERLAY 3 Var4 * Var5

RESCALE OVERLAY 1 3
```

The scale on the left will have a range of .05 to 6.0 which is the low value for Var1 and the high value for Var3, the Y variables for the plot and the second overlay. The scale on the right will have a range of 25 to 75 which are the lowest and highest values observed in Var2 and Var4, the Y variables in the two overlays that are rescaled.

The label for the left axis is:

1. The name of the PLOT variable or,
2. The second extended label for the PLOT variable in the labels file, or
3. The text associated with the subcommand “LABEL Y”

The explicit “LABEL FOR Y” takes precedence over the extended variable label which takes precedence over the name of the Y variable. The label for the right axis is either the name of the lowest level overlay variable or its second extended variable label taken from a label file. There is no subcommand to provide a label for this axis.

4.6 Log Plots

The Y variable and its axis can be automatically transformed with a log10 transformation. The Y axis will then print with ticks properly placed and labelled as on log graph paper. This is controlled by the LOG10 (ON OFF) subcommand. LOG10 with no argument is equivalent to LOG10 ON.

```
PLOT MyFile, PR 'Test.ps';
P Measures by Month,
LOG10;
$
```

The Y variable must not be previously transformed. The transformation is done by the PLOT command and the Y axis is properly labelled. The TICKS and TICK.LABELS subcommands should not be used with the LOG subcommand. Figure 4.4 illustrates the command and the resulting PLOT.

Figure 4.4 Log Plot

```
    ----------The PLOT Command---------------------------------------

   PLOT GasElec;
   P Gas.CCF * Month,  OVERLAY Elec.KWHR * Month,  LOG10,  COLS 70;
   $


    ----------The Output---------------------------------------------

            Plot of Gas.CCf by Month (Legend: *=1, 2=2, ..., $=10+)
               Overlay Elec.KWHR by Month (Legend: A =1 or more)

    10000 +
          +
          +
          +                         A    A    A
G    1000 +A    A    A    A    A    A         A    A    A    A    A
a         +6   3                                                 2
s         +3   4    4                                       7    5
.     100 +    2    4    6              *              5    *    *
C         +              3    4    2              3    2
C         +         *         5    4    7    4    4
f         +                         3    *    4    *
       10 +
          +                                             *
          +
          +
        1 +
           +----+----+----+----+----+----+----+----+----+----+----+
           1    2    3    4    5    6    7    8    9    10   11   12
                                    Month
```

4.7 DATE.AXIS AND MAX.TICKS

In Figures 4.2 and 4.3 the labels file has value labels for the X variable. When value labels are available they are used to label the X and/or Y tick marks and determine the location of those tick marks.

```
       Var4  XL 'Quarter'
                'Quarterly Measurements'
             (2) 1994  (6) 1995  (10) 1996 (14) 1997 /
```

The values determine the tick marks and the value labels become the tick labels. If either TICKS or TICK.LABELS is used for a variable, the values and the value labels from the labels file are ignored.

If there are two or more variables in the file which contain date information and if those variables have any of the names: “year”, “month”, “day”, “hour”, or “minute”, a composite date variable is produced from the combination of the available individual date variables. As the data are read the value for this extra variable is calculated for each case in the file. This new variable becomes the X variable when the subcommand DATE.AXIS is used.

When DATE.AXIS us used, the ticks and labels for the X axis are constructed automatically. Ranges, ticks, and tick labels should NOT be supplied as the new date variable is represented as hours since January 1, 1900. The PLOT command decides which level of the date variable controls the ticks. For example, if Year, Month, and Day are all present in the file the base measure could be any of the three depending on the ranges involved and the space available for ticks and labels. If there are only a few years represented, the ticks will be placed and labelled using months with the appropriate year printed below and centered between months 1 and 12.

The tick labels for the months are either 3 letter abbreviations (with Jan representing the first month) or a single letter when 3 letters may not fit. If there is an entry in the labels file for the variable month, that entry will be used instead. The P-STAT system file used in Figure 4.5 has variables Month and Year. The generated X variable is labelled with the years because there is not room enough for the months to be printed.

Figure 4.5 DATE.AXIS: Generated from Month and Year

```
               Plot of Gas.CCf by X (Legend: *=1, 2=2, ..., $=10+)

       500 +
           |
           | *           *        *
       400 +              *     **             *
 G         |       *                    *
 a         |        2                    *           *
 s     300 + *                         *      *
 .         |      *        *
 C         |      *      *      *               *     *     **
 C     200 +               *           * *           *
 f         |  *      *  *   *     *           * *      *   *
           |     *    *                                *    * **
       100 +                *         *      *   *  *
           |  *  *   ** *    * *   2      2      *      *  *
           |   2*      *     **     2**    2**    2**   *2*   **2
         0 +                   *
            +------+------+------+------+------+------+------+------+
            85     86     87     88     89     90     91     92     93
```

In general the DATE.AXIS command makes a reasonable decision on the labelling depending on which year/month/day variables are available. If there are many months over a period of years, the decision is made to use only the year information in the labels. You can control this to some extent by specifying the level of labelling that you would like.

Figure 4.6 DATE.AXIS Generates Labels for the Months

```
    ----------The PLOT Commands--------------------------------------

     PLOT GasElec [ CASES 1 TO 36 ];
     P Gas.CCF, LINES 20,  COLS 70,  NO LEGEND,
     DATE.AXIS,  LABEL X 'Date';
     AGAIN,  DATE.AXIS LONG;

    ----------The PLOTS---------------------------------------------

      500 +
          |                     *                                *
G     400 + **
a         |                   *                 *
s     300 +                 *    *                ** 
.         |                                    *                *
C     200 +    *           *                 *
C         |     *        *         *        *        *   *    *
f     100 +
          |       ** ** *           * ** **           * *  * *
        0 +
           +--+--+--+--+--+--+--+--+--+---+--+--+--+--+--+--+--+--+
           J  M  M  J  S  N  J  M  M  J   S  N  J  M  M  J  S  N  J
                   84              85                 86
                                    Date

              Plot of Gas.CCf by X (Legend: *=1, 2=2, ..., $=10+)

      500 +
          |                     *                                *
G     400 + **
a         |                   *                 *
s     300 +                 *    *                **
.         |                                    *                *
C     200 +    *           *                 *
C         |     *        *         *        *        *   *    *
f     100 +
          |       ** ** *           * ** **           * *  * *
        0 +
           +--+--+--+--+--+--+--+--+--+---+--+--+--+--+--+--+--+--+
          Jan   May   Sep   Jan   May    Sep   Jan   May   Sep   Jan
             Mar   Jul   Nov   Mar   Jul    Nov   Mar   Jul   Nov
                   84              85                 86
                                    Date
```

```
DATE.AXIS YEAR
```

requests that the year labels be used even though there might be room for the months as well. The keywords that can follow DATE.AXIS are: YEAR, MONTH, LONG, SHORT, and DAY. LONG and SHORT apply to the way that month is represented. In Figure 4.6, the first plot on 3 years of data has both the months and years represented with single letter abbreviations for every other month. In the second plot the request

```
DATA.AXIS LONG
```

causes the ticks to be labelled with three letter abbreviations for the month. If YEAR were used, the tick labels for the months would be omitted.

The subcommand MAX.TICKS can be used to control the number of ticks that are printed on the axes.

```
MAX.TICKS 10            limits the ticks on both axes
MAX.TICKS X 8           limits the ticks on the X axis
MAX.TICKS Y 12          limits the ticks on the Y axis
```

MAX.TICKS controls the number of ticks that can have labels associated with them. When a DATE.AXIS is in effect, it is possible to have more ticks than labels. For example, you might wish to present monthly data over a three year period with 36 ticks marks, one for each month but only label every second or third mark. This can be accomplished by setting MAX.TICKS to 12 for a 1 in 3 label or to 18 for a 1 in two label. This type of labelling only applies to the DATE.AXIS situation. In all other situations MAX.TICKS of 12 will result in 12 tick marks and 12 labels.

Figure 4.7 Choice of Plot Characters Matters

```
      ----------The PLOT Command-----------------------------

      PLOT  GasElec;
      LINES 22, COLS 70,
      P Gas.Cost * Month,  Overlay Elec.Cost * Month,
      CHAR PLOT 'G',   CHAR OVERLAY 'E';


      ----------The Output-----------------------------------

                Plot of Gas.Cost by Month (Legend: G =1 or more)
                Overlay Elec.Cost by Month (Legend: E =1 or more)

       200 +
           |                              E
 G         |G
 a     150 +G                             E                        G
 s         |G    G                        E    E              G    G
 .         |G    G                   E    E    E                   G
 C     100 +G    G                   E         E              G
 o         |E    G    G         E    E    E    E    E    G    G    G
 s         |G         G    E    E    E    E         E    E    G    G
 t      50 +E    E    E    G    E         G         E    G    G    E
           |E    E    G    G    G    E    G         E    G         E
           |                    G    G    G    G    G
         0 +                              G              G
            +----+----+----+----+----+----+----+----+----+----+----+
            1    2    3    4    5    6    7    8    9    10   11   12
                                     Month
```

4.8 PLOT CHARACTERS AND JOIN.POINTS

The characters which are used to represent values in the body of the plot can add information or, if used without discretion, be a distraction. If there are a great many values to be placed in a scatter plot, the use of numbers to indicated the density may not be advantageous. This is particularly true when there is an overlay to the plot. In Figure 4.7 the character 'G' used for the plot is so much like the 'E' used for the overlay that the pattern of high gas costs in winter and high electric costs in summer is not readily apparent.

Figure 4.8 is the same as Figure 4.7 except for a choice of characters which provide a better contrast.

```
CHAR PLOT '|',   CHAR OVERLAY '^' ;
```

When you are working with overlays it is often helpful to use P-STAT interactively trying different combinations of plot characters before sending the final results to the printer. In general, it is better to use a single character for each plot and overlay even though the plot information which details the number of observations that fall on a particular point is lost. The characters that are chosen should provide, if possible, both meaning and contrast. Even though the letters G and E used in Figure 4.7 could be considered informative, the lack of contrast makes them less effective than the characters used in Figure 4.8.

Figure 4.8 Plot Characters: A better selection

```
    ----------The PLOT Subcommand----------------------------

     CHAR PLOT '|',   CHAR OVERLAY '^' ;


    ----------The PLOT Output--------------------------------

               Plot of Gas.Cost by Month (Legend: | =1 or more)
               Overlay Elec.Cost by Month (Legend: ^ =1 or more)

      200 +
          |                              ^
G         ||
a     150 +|                             ^                        |
s         ||    |                        ^    ^              |    |
.         ||    |                   ^    ^    ^                   |
C     100 +|    |                   ^         ^              |
o         |^    |    |         ^    ^    ^    ^    ^    |    |    |
s         ||         |    ^    ^    ^    ^         ^    ^    |    |
t      50 +^    ^    ^    |    ^         |         ^    |    |    ^
          |^    ^    |    |    |    ^    |         ^    |         ^
          |                    |    |    |    |    |
        0 +                              |              |
           +----+----+----+----+----+----+----+----+----+----+----+
           1    2    3    4    5    6    7    8    9    10   11   12
                                    Month
```

Figure 4.9 One-Way Plot With Overlay and JOIN.POINTS

```
                Plot of Gas.Cost (Legend: % =1 or more)
                Overlay Elec.Cost (Legend: O =1 or more)

      150 +
          |
          |
          |                                    O
G         |
a     100 +              O
s         |                          %
.         |  %                               O   O       %
C         |%           O         % %   %
o         |                                            %
s      50 +                O   % O                 O   O O
t         |    % O   O       O     O O O % O         %
          |O O O % O         %           O
          |                %               %       %
          |        % % % %                   % % %
        0 +
           --------+---------+---------+---------+---------+
                   5         10        15        20        25
```

----------Adding JOIN.POINTS----------------------------

```
                Plot of Gas.Cost (Legend: % =1 or more)
                Overlay Elec.Cost (Legend: O =1 or more)

      150 +
          |
          |
          |                                    O
G         |                                   --
a     100 +              O                    - -
s         |             --           %       -  -
.         |  %          --          - -      O   O       %
C         |%--         O  -      %-%   %    -    -      -
o         |   -       -   -     -      -    -     -    %
s      50 +   -       -    O   %-O      -  -       O  -O-O
t         |    % O   O      -O-   -O-O-O % O        -%-
          |O-O-O-%-O-        %          -O-         -
          |       -        %-              %       %
          |        %-%-%-%-                 -%-%-%-
        0 +
           --------+---------+---------+---------+---------+
                   5         10        15        20        25
```

4.8 contains two one-way plots created by the following PLOT command:

```
PLOT GasElec [ CASES 1 TO 24 ];
P Gas.Cost, OVERLAY Elec.Cost,
LINES 24, OW 70,
CHAR PLOT '%',  CHAR OVERLAY 'O';
AGAIN,  JOIN.POINTS;
```

In Figure 4.9, the first 2 years of data are selected so that the pattern is apparent in a narrow character plot. The first plot is similar to the ones we have just seen. The second has the additional subcommand JOIN.POINTS. JOIN.POINTS causes either a dash or a vertical bar to connect points in the plot and in the overlay. While a PostScript or XWindow plot with lines that can be drawn is more effective, the use of JOIN.POINTS in this situation helps to make the seasonal pattern clear.

The dash is used to connect adjacent points have the same Y value but different X values. The vertical bar is used to connect adjacent points that have the same X value but different Y values.

4.9 USING P-STAT MACROS IN PLOT

P-STAT macros are a powerful way to provide PLOT information because they can be used as part of either the command or the subcommands. They can be used to provide PLOT controls that are always to be included and a set of plots such as TITLES, LINES or COLS. They can also be used to provide alternate output formats depending on whether the output is going to a character printer or a PostScript printer.

Figure 4.10 Macros in PLOT Commands

```
----------The MACROS-------------------


 MACRO BasicIden $
 /* identifers for Plot Command. */
 , NO XWINDOW, TITLES
 ENDMACRO $


 MACRO BasicSub $
 /* these are basic subcommands for
    output to the terminal */
 TITLES ’.date.’,
 LINES 22,  COLS 70
 ENDMACRO $


 MACRO PSout ( prefix ) $
 /* subcommands for postscript output */
 TITLES ’.date.’,
 PR  &prefix.ps, POSTSCRIPT PORTRAIT,
 RIGHT.EDGE 7.5, BOTTOM.EDGE 2,
 BORDER
 ENDMACRO $
```

```
  ----------PLOT Command and Expansion-------------------


  PLOT GasElec !!BasicIden;
  !!BasicSub, P Gas.CCF * Month;
  $


  PLOT COMMAND EXPANDED FOR EXECUTION


  PLOT GasElec, NO XWindow, TITLES;
  TITLES '.date.',
  LINES 22,  COLS 70
 , P Gas.CCF * Month;
 $


 ----------PLOT Command and Expansion-------------------


  PLOT GasElec !!BasicIden;
  !!PSout ( GEplot ), P Gas.CCF * Month;
  $


  PLOT COMMAND EXPANDED FOR EXECUTION


 PLOT GasElec, NO XWindow, TITLES;
 TITLES '.date.',
 , POSTSCRIPT PORTRAIT,
 RIGHT.EDGE 7.5, BOTTOM.EDGE 2,
 BORDER
 , P Gas.ccf * Month;
$
```

Figure 4.10 contains 3 macros and 2 PLOT commands. The PLOT commands are shown as they are entered and then as P-STAT expands them for execution. The combination of two exclamation marks (!!) before the macro name tells P-STAT where to insert the macro text. The final macro has an argument, which provides part of the name for the PostScript output file.

```
MACRO PSout ( prefix ) $   the MACRO has an argument
PR  &prefix.ps,            the argument is used in the macro

!!PSout ( GEplot )         the call to the macro produces
PR  GEplot.ps              the PostScript file name
```

When ever a macro argument appears in the macro preceded by an ampersand, the text provided in the call to the macro is substituted when the macro is executed. Macros are described in full in the manual ”P-STAT: A Guide to the P-STAT Programming Language (PPL), Macros and TextWriter”.

SUMMARY

PLOT

```
        PLOT  Paint,  TITLES,  LABELS 'Paint.lab';

          P  Viscosity  *  Coverage ;
          P  Cost       *  Coverage ;
          $
```

The PLOT command produces scatter plots of numeric variables. Basic PLOT controls (commands and subcommands) were described in the previous chapter. This chapter covered the use of titles and labels to enhance the looks of the output. The next chapter describes how to convert these character plot to camera-ready output for PostScript printers.

Required:

PLOT

provides the name of the P-STAT file that contains the input data.

Optional Identifiers:

LABELS fn

provides the name for a file with extended variable names and value labels. The extended labels are used in the plot legend. Secondary extended labels are used as axis labels. Value labels are used to provide the tick locations and tick labels. Subcommands for labels and ticks take precedence over the labels file.

TITLES

specifies that any previously defined titles are to be used on the plots produced by the PLOT command:

```
 PLOT  Scores,  TITLES ;
```

(See the TITLES chapter in “P-STAT: Introductory Manual” for additional information about defining titles using the TITLES *command*.) The *subcommand* TITLES may be used to define titles within PLOT.

Optional Subcommands:

DATE.AXIS

causes the PLOT command to look for two or more variables with the names Year, Month or Day and create a single date variable selecting appropriate ticks marks and labels. If Month seems to be the appropriate level for ticks and tick labels, the choice of single character or three letter abbreviations depends on the space available. Tick labels can be controlled by using DATE.AXIS with one of the following:

```
 YEAR  MONTH  DAY   SHORT (1 letter month)  LONG (3 letter month)
```

If you have both Year and Month, selecting Year causes the Month information to be omitted from the X axis. If you have both Month and Day, selecting Month (or SHORT or LONG) causes the day information to be omitted from the X axis. MAX.TICKS is often useful with the date formats.

JOIN.POINTS

requests that the points in the plot and in any overlays be joined by either a dash or a vertical bar depending on the relation of one point to the adjacent point. JOIN.POINTS can be helpful in a one-way plot with overlays to help you distinguish the internal patterns of the variables involved.

LABEL vn 'cs'

specifies a name or label for the designated axis. One or both of the axes may be relabeled. The program-supplied labels, that is, the variable names, are suppressed when user-supplied labels are specified. The labels are centered within the plot edges and are in effect for the current plot only. LABELS X or LABEL FOR vn may also be used. Enclose the character string label in quotes.

LOG10 ON / OFF

specifies that the Y variables are to be transformed with a log10 transformation and that the ticks and labelling conform to standard log plot conventions. LOG10 can be used in character and XWindow plots as well as in PostScript plots.

MAX.TICKS (X Y) nn

specifies the maximum number of ticks that should be printed on the axes. If X or Y is not specified, the maximum applies to both axes.

RESCALE OVERLAY nn nn

provides the overlay level(s) which are to be rescaled with ticks and labels on the right edge of the plot. RESCALE is used when the Y variable for the plot and the Y variable for one or more of the overlays have very different ranges.

TITLES 'cs'

provides lines of titling to be used at the top of the plot (above the legend generated by the PLOT command) or at the bottom after the plot. The titles are supplied exactly as they are in the TITLES command with one exception — the identifier TITLES must precede each line of titling:

```
AGAIN,  TITLES T1 LEFT  'First String' ,
        TITLES T2 RIGHT 'Second String' ;
```

(See the TITLES chapter in “P-STAT: Introductory Manual” for additional information on defining titles.)

Titles specified within the PLOT command replace any titles previously defined for those title lines external to PLOT. Titles must be enclosed in single or double quotes and may not be longer than 80 characters. Titles remain in effect for subsequent plots. Providing a title as a subcommand turns on *all* defined titles. Thus, if any titles were defined at the command level (outside of the PLOT command), those titles will appear in addition to any defined at the PLOT subcommand level.

5
PLOT:
PostScript Support

PostScript support has been added to the PLOT command. This provides camera ready plots whenever there is a PostScript printer available. It is independent of the type of computer or operating system. Plots can be approximated first in character or Xwindow format with the PostScript information added when the test plot is satisfactory. If there is a PostScript previewer available, the PostScript output can be viewed on the screen before it is printed.

Most of the PLOT subcommands are available in all three modes; character, Xwindow, and PostScript. Some new subcommands have been added specifically for PostScript, and a few subcommands which control the appearance do not have the same effect in all three environments.

5.1 The PostScript Subcommand

POSTSCRIPT may be used at any time (before/after character plot or X plot). It causes subsequent plots to be written using the PostScript language. An external print file must be provided to hold this printout. If you are running from an Xwindow, use the NO XWINDOW identifier as well as PR. The PR identifier may be used as an identifier or as a subcommand.

```
PLOT MyFile,  NO XWINDOW,  PR 'plot.ps';
P Measure BY Year, POSTSCRIPT;
$
```

POSTSCRIPT assumes a landscape mode on 8.5 X 11 inch paper. In landscape mode, the width of the plot is usually greater than the height and the output is rotated 90 degrees on the paper. In PORTRAIT orientation, the height of the plot is normally greater than the width. This can be controlled by POSTSCRIPT PORTRAIT or POSTSCRIPT LANDSCAPE, and by the DIMENSION subcommand. PORTRAIT and LANDSCAPE control the orientation. DIMENSION controls the size. You may specify PORTRAIT for a plot that has a width that is greater than the height but you must be sure that the plot will fit on the physical page.

5.2 Dimensions and Margins

The dimensions may be given for both the paper and for the plot area. The first number is the height and the second number is the width.

```
DIMENSION PAPER 8 BY 12,   DIMENSION PLOT 7 BY 11
```

TOP.EDGE, LEFT.EDGE, RIGHT.EDGE and BOTTOM.EDGE can be used to specify how the plot is positioned on the paper. The argument is specified in inches, for example:

```
TOP.EDGE .5
```

The final positioning of the plot on the page depends on orientation, dimensions, margins, and (for multiple plots) the amount of space that is allowed between plots. If there are no spacing subcommands, the assumptions are a landscape orientation on 8.5 by 11 paper with a 1 inch top margin and a 1 inch left margin. The plot itself is 6.5 inches high and 9 inches wide (the assumed plot size). If you specify that the top margin is to be half an inch, the plot area does not change and the bottom margin becomes 1.5 inches.

TOP.EDGE and DIMENSION PLOT can be combined to produce a given bottom margin. For example:

```
TOP.EDGE .5, DIMENSION PLOT 7 BY 9,
```

This combination of subcommands increases the plot area leaving a bottom margin of .5 inches. On most PostScript printers there is a small area on the borders of the page that is not available for printing. Margins smaller than .5 inch may cause some invisible printing on the edges.

When a single plot per page is being used, the combination of the four edge parameters may be used to size the plot:

```
      TOP.EDGE 1, LEFT.EDGE 1.5, RIGHT.EDGE 1.5, BOTTOM.EDGE 3
```

RIGHT.EDGE and BOTTOM.EDGE are only used in the single plot per page situation.

The plot area consists of the plot plus all the titles (top and bottom) and the various legends and labels. It does not include the optional outer border which will be placed in the margin area. The size remaining for the body of the plot depends on the amount of labelling that is required.

Some PostScript viewers do not handle a change in the dimensions of the paper very well. The desired effect can often be obtained by the use of the 4 edge settings or the DIMENSION PLOT subcommand described below.

5.3 Page Placement and Multiple Plots

The TOP.EDGE, and LEFT.EDGE subcommands can be used with the DIMENSION PLOT subcommand to size a plot and to place it on the page. For example:

```
      PLOT MyFile,  PR 'Test.ps';
      POSTSCRIPT,
           TOP.EDGE 4.5,  LEFT EDGE 6,
           DIMENSION PLOT 3 BY 4;
      P Var1 BY Var2;
```

This plot is placed in the lower right-hand corner of the paper. The rest of the paper is left blank. However, it is possible to place several plots on a page either from a single plot command or from several different plot commands by using the PLOTS.PER.PAGE and the SHOWPAGE subcommands. The order in which plots are placed on a given sheet of paper does not matter.

DIMENSION takes precedence over settings of RIGHT.EDGE and BOTTOM.EDGE. TOP.EDGE and LEFT.EDGE settings are always used because they provide the position for the top left corner of the plot.

After the instructions to produce a plot have been written to the print file, a final instruction tells PostScript to print the page unless the NO SHOWPAGE subcommand is in effect or the PLOTS.PER.PAGE subcommand is being processed. SHOWPAGE ON and SHOWPAGE OFF permit you to specify exactly when a page is complete.

```
      PLOT MyFile,   PR ‘Test.ps’;
      POSTSCRIPT,    SHOWPAGE OFF,
      TOP.EDGE .8, LEFT EDGE .8,
      DIMENSION PLOT 3 BY 4.5,
      P var1 * var2;
      AGAIN,  TOP.EDGE 4.5,   SYMBOL PLOT STAR;
      AGAIN,  TOP.EDGE  .8,   LEFT EDGE 5.8,     SYMBOL PLOT HASH;
      AGAIN,  TOP.EDGE 4.5,   SYMBOL PLOT TRIANGLE,
      SHOWPAGE ON;
```

In this example, the same plot is repeated 4 times in a single command to illustrate some of the symbols that are available. When SHOWPAGE OFF and SHOWPAGE ON are used, the plots can also be made by separate PLOT commands perhaps using data from several different P-STAT system files. You MUST remember the final SHOWPAGE or the image will remain in the printer’s memory and unless the next user specifically clears the memory, his output may overlay your final plots.

An easier way to specify how many plots are to be placed on a page is to let the PLOT command figure out the appropriate sizes and when SHOWPAGE should be issued.

```
PLOT MyFile,   PR 'Test.ps';
POSTSCRIPT,    PLOTS.PER.PAGE 2 BY 2;
AGAIN,         SYMBOL PLOT STAR;
AGAIN,         SYMBOL PLOT HASH;
AGAIN,         SYMBOL PLOT TRIANGLE;
```

PLOTS.PER.PAGE requires 2 arguments; the number of plots down the page and the number of plots across the page. The plots are filled in across the page first. A 2 by 2 with only three plots defined will have 2 plots on the top of the page and a single plot in the bottom left quadrant of the page. If there are too many plots for a single page, multiple pages are printed. A 1/4 inch gap is usually left between the plots. This can be controlled by using the SPACE subcommand which is discussed later in the chapter.

```
POSTSCRIPT PORTRAIT,  PLOTS.PER.PAGE 3 BY 1,
```

You can use PLOTS.PER.PAGE with SHOWPAGE OFF followed by another plot command using TOP.EDGE and DIMENSION PLOT to put yet another plot across the bottom of the page. Unless SHOWPAGE OFF is used in this second plot, it automatically issues a SHOWPAGE instruction and completes the multiple plot sequence.

5.4 Specifying Symbols

In character plots, the plot symbols are letters and numbers. In the Xwindow and PostScript plots the symbols are drawn and are referenced by names. The available symbols are:

```
SQUARE     TRIANGLE      STAR     CIRCLE
PLUS       DIAMOND       HASH     X
Y          Z             DOT      PERIOD
```

The DOT is a filled circle. The period makes a tiny mark and is best used only for very dense plots where the details are not as important as the shape.

The SCALE subcommand has been enhanced for PostScript plots. In character plots, the scale value is used as a weight and controls which letter or number is placed in the plot. In PostScript plots the scale value is used to dictate the size of the symbol. The assumed scale value is a 1 which makes symbols that are very small but can still be identified. SCALE 2 makes symbols that are very easily identified. SCALE 2. work well unless there are many plot points. SCALE 3 makes large distinct symbols for each plot point. Scale values may be fractional.

A variable may be supplied instead of a value. The symbols will be scaled for each case according to the value of the scale variable. All of the symbols except the period are sized according the value of the scale variable.

```
SYMBOL PLOT STAR,  SYMBOL OVERLAY 1 DOT, SYMBOL OVERLAY 2 HASH,
SCALE Weight.var,
```

If you do not specify the symbols, the symbol for the plot is the square, the symbol for the first overlay is the triangle, the symbol for overlay 2 is the star, and so forth. If the symbol is a circle or a dot, the radius is taken from the scale value. In the remaining symbols the scale value is used as the distance from the center of the symbol to the edge. Each such unit translates into 1/72 of an inch. A square symbol drawn with scale 1 is 2/72 of an inch across. If drawn with scale 2, it is 4/72 of an inch. The scale size for the symbol in the legend will be 1, 2, or 3 depending on the pointsize specified for the legend.

5.5 The Border

The border around the plot is optional. The BORDER subcommand creates the border from solid lines in the margin area about 1/4 of an inch outside the “plot” area. The placement of the border is controlled by adding a numeric argument which specifies in inches how far inside the margins the border should be placed.

```
BORDER .1,
```

moves the border 1/10 of an inch outside the plot area. The border is always a solid line but the width of the border may be controlled with the LINE.WIDTH BORDER subcommand. The usual setting is .5 .

```
      BORDER .3,   LINE.WIDTH BORDER 1.5
```

creates a border that is heavier than the other lines in the plot. A heavier border gives the plot more definition when multiple plots per page are specified.

Because the border is outside the plot dimensions, the margin must be large enough to place it and still leave the space that is outside the print area of most PostScript printers. If the margin is .5 and the border is printed with the default value of .25, it will probably not show on the printed page.

5.6 Line Types

Plots are assumed to be scatter plots unless line types are specified for the plot and the overlays. LINE.TYPE ON is used to request solid lines for the plot and all overlays. LINETYPE PLOT DASHED, if used by itself, causes the points for the plot to be connected by a dashed line. Any overlays will not have connected points. The supported line types are:

```
       DOTTED,    SHORT.DASH,   DASHED,   LONG.DASH,
       DOT.DASH,   ODD.DASH,    SOLID
```

To specify the line types:

```
      LINE.TYPE PLOT SOLID,
      LINE.TYPE OVERLAY 1 DASHED,
      LINE.TYPE OVERLAY 2 DOTTED,
```

The LINE.TYPE subcommand may be used several times. To specify connected solid lines for the plot and all overlays and then request that the line for the third overlay be dashed:

```
      LINE.TYPE ON,
      LINE.TYPE OVERLAY 3 DASHED,
```

The lines are usually continuous. However, it is possible to request that a gap be left if there are any missing data values by using JOIN.POINTS GAP. The assumed setting is JOIN.POINTS CONTINUOUS.

The line types for grids may also be specified:

```
      LINE.TYPE GRID DOTTED
```

There are several different places where lines are used in the plots. Lines surround the optional border around the entire plot. Lines are used for the top, bottom, left and right edges of the plot proper. Lines are used to connect the points within the plot and lines are used for the optional grid lines. The borders and edges are always solid lines. The lines for the grids and for joining points are controlled by the LINE.TYPE subcommand.

Each of the four types of lines can have the LINE.WIDTH specified. The assumed line.width, .5, is half of the basic PostScript address unit which is 1/72 of an inch. LINE.WIDTH requires two arguments, the line and the number which is to be the width. Possible lines are: EDGES, GRIDS, BORDER, JOIN.POINTS

```
      LINE.WIDTH BORDER 1.5,  LINE.WIDTH GRIDS .1
```

5.7 Fonts

The FONT subcommand is used to select the fonts and pointsizes for each of the four areas of the plot. The four different areas in the plot for which fonts may be specified are TITLES, the LEGEND, the LABELS for the X and Y axes and the TICK.LABELS. Any characters used in the body of the plot are printed in the same font as the TICK.LABELS.

The pointsize is the measure of the height of the letters. The width of the letters varies in proportional fonts such as Times Roman but is fixed (each character uses the same amount of space) in fixed width fonts such as Courier.

Times Roman 8 point font is assumed for all 4 areas unless the FONT subcommand is used. FONT (area) fontname can be used to change the font for each area individually or, if the area is not specified, for all areas.

```
FONT COURIER
```

causes COURIER, a fixed width font, to be used for all the areas in the plot. Since a point size is not specified, 8 points is assumed. To change the font to COURIER and the pointsize to 10 use:

```
FONT COURIER 10
FONT TIMES BOLD 10
```

sets the font types of all 4 areas to 10 point TimesBold. The other 3 areas of the plot remain Times Roman 8 point.

```
  FONT TICK.LABELS COURIER 8
```

sets the tick labels and the body of the plot to 8 point Courier.

The standard PostScript font set includes:

Times	Times Italic
Times Bold	Times BoldItalic
Arial	Arial Italic
Arial Bold	Arial BoldItalic
Courier	Courier Oblique
Courier Bold	Courier BoldOblique

These fonts may all be specified by supplying the 1 or 2 pieces that comprise the names without using quotes. Other fonts can be requested by enclosing the PostScript name in quotes followed by the pointsize. For example:

```
 FONT TITLES "ZapfChancery-MediumItalic" 10
```

The spelling of these special fonts must be exact or a default font (probably Courier) will be substituted by the PostScript printer. See the POSTSCRIPT.SETUP command in Chapter 2 which explains how to set a different font for each line of titles.

5.8 Use of Color

If you have a color PostScript printer or are running XWindows on a color terminal, you can specify colors for the fonts, the symbols and the lines that join the symbols. Color for the fonts must be set before the plot command in either the POSTSCRIPT.SETUP or POSTSCRIPT command. Colors for the lines and symbols in the plots can be set in the SYMBOL and LINE.TYPE subcommands.

```
SYMBOL PLOT RED DIAMOND,
LINE.TYPE PLOT RED SOLID,
```

The available colors are RED, ORANGE, YELLOW, GREEN, BLUE, VIOLET, BROWN, BLACK, WHITE, CYAN and MAGENTA.

5.9 The Legend

The Legend has traditionally been at the top of the PLOT. It may be omitted from the plot by using the subcommand LEGEND OFF which is supported in all types of plots. LEGEND (SHORT LONG) (TOP BOTTOM RIGHT) are now supported in PostScript plots. A short legend contains only the symbol and the name of the variable on the Y axis. The short legend is most useful when the legends are on the right edge of the plot. If LEGEND SHORT RIGHT is used, an inch is allocated on the right for labelling. If the variable name is long or a large font is used for the legend, this space may be inadequate.

```
LEGEND SHORT RIGHT,  SPACE for LEGEND 2,
```

allocates a 2 inch space on the right for the legend.

5.10 The Space Subcommand

The SPACE subcommand is used to control:

1. The space that is allowed for ticks on the Y axis and their labels;
2. The distance between the corner of the plot and the first tick. ;
3. The space allowed for the legend when it is on the right;
4. The space between plots when there are multiple plots on the page.

The argument for space is given in inches. The space at the corners is usually set at .111 inches. This means that the first tick on each axis falls slightly inside the actual border of the plot and any data point that occurs for the border values will visible instead of being obscured by the plot boundary.

```
SPACE CORNERS 0
```

causes the first tick mark to be at the corner location.

```
SPACE TICK.LABELS 2.,
```

provides a 2 inch space for tick labels. This is used only for the labels on the Y axis. The program estimates the amount of space needed for a label of less than 8 characters. However, if you have long tick labels or tick labels that are all in upper case with many wide characters, it may be necessary to supply additional space to keep the tick labels from overlaying part the Y axis label. If you have very short labels, a smaller space may improve the plot appearance.

```
SPACE BETWEEN .333,
```

requests 1/3 of an inch of space between plots when there are multiple plots per page. The assumed space is .25 or 1/4th of an inch.

5.11 Tick Labels on the X Axis

Tick labels on the X axis can be tailored by specifying how they are to be placed in relation to the ticks. This is done by using an additional argument in the TICKS.LABELS subcommand. To place the tick labels between a tick and the following tick use:

```
 TICKS.LABELS X RIGHT  January February March etc.
```

The new arguments are LEFT, CENTER, and RIGHT. The default is labels that are centered under the tickmarks. LEFT causes them to be centered between the current tickmark and the tickmark to the left. RIGHT centers the labels between the current tickmark and the tickmark to the right.

When the tick labels are supplied through a labels file or are generated by the DATE.AXIS subcommand, they may be positioned by the POSITION LABELS subcommand. The position arguments are that same as the position arguments for the TICK.LABELS subcommand.

```
POSITION LABELS RIGHT
```

5.12 Controlling Tick Placement

The subcommand SPECIFICATIONS (SPECS) is used to control the placement and the size of the tick marks. All of the following examples are legal specifications for tick marks.

```
SPECS for ticks THROUGH AXES
SPECS 4 long INSIDE AXES
SPECS .01 wide
SPECIFICATIONS for ticks 2 long .02 wide OUTSIDE AXES
SPECS 3 .1 INSIDE
```

If size indicators are given, the first number controls the length in units of 1/72 per inch and the second number controls the line width. The usual settings are 3/72 long and .01/72 wide. Tick marks in PostScript mode are placed inside the axes unless the SPECS subcommand is used. When the ticks go through the axis, the length is the distance on either side of the axis. Thus a length of 2 will result in ticks that are a total of 4/72 of an inch in length.

5.13 Error Handling and Verbosity.

Error handling and informative messages are a problem when the printout is going to a PostScript file. Such a message will cause the print run to abort. If you use the ERROR.FILE command to provide a separate destination for error messages the messages will be sent only to that file. When the PostScript flag is turned on, the error file will also be used for informative messages.

If the run is a batch run and there is no error file, the error messages and informative messages will be written to the PostScript output file. If the run is interactive and there is no error file, both the error messages and informative messages print at the terminal.

```
ERROR.FILE  'Plot.errors' $
VERBOSIY 1 $
Plot Myfile, ....
```

It is a good idea in a postscript run to set the verbosity level to 1, which limits the printout and ensures that the command and subcommand records will not be echoed in the postscript file.

:

PLOT EXAMPLES

PLOT EXAMPLES

PLOT EXAMPLES

PLOT EXAMPLES

PLOT EXAMPLES

SUMMARY

PLOT

```
PLOT  Paint, PR 'Plot.ps' ;

POSTSCRIPT,
  P  Viscosity  *  Coverage ;
  P  Cost       *  Coverage ;
  $
```

The PLOT command produces scatter plots of numeric variables. Desired plots and options are specified at the subcommand level. Options include overlays, observation of additional variables, user-defined character sets, grid lines, ticks, tick labels, axis labels and titles. Plots may be requested for subgroups using BY. Character variables may be observed or may be BY variables.

The following identifiers and subcommands are designed for use with a PostScript laser printer to produce high resolution graphics. For basic information on the PLOT command see chapters 3 and 4 in this manual.

Optional Identifiers:

DEF "fn"

supplies the name for a file of plot subcommands to be executed at the start of the plot command. If this is a PostScript plot, the POSTSCRIPT subcommand should be included in the file and print definition should be specified either in the command or in the definition file so that it is processed before the initial PostScript information is printed.

PR "fn"

A print file to hold the postscript output is required except on a PC where the laser printer is directly connected to the computer in which case PR LPT1 (LPT2, LPT3) may be used to print the file directly. PR may be supplied as either an identifier or a subcommand. However, when used as a subcommand, it must be a short name that is legal for P-STAT and entered without the quotes. PR must precede any POSTSCRIPT statements.

Required Subcommands:

POSTSCRIPT LANDSCAPE / PORTRAIT

specifies that the output is to have PostScript controls. If no argument is supplied LANDSCAPE orientation is assumed. A PR file must be supplied before this statement is given.

Optional Identifiers:

BORDER nn

provides a border around the entire plot. If a number is supplied it indicates the distance in inches that the border should be placed outside the regular plot area. .25 or 1/4 of an inch is assumed if no number is given.

BOTTOM.EDGE nn

provides the bottom margin in inches. This is used only when there is a single plot on the page. PLOTS.PER.PAGE and DIMENSION PLOT take precedence over this setting.

DIMENSION PAPER / PLOT nn BY nn

provides that size of the PAPER or the PLOT. The first number is the height, the second the width. The assumption is paper that is 8.5 by 11 and a plot that is 6.5 by 9.

FONT TITLES / LABELS / LEGEND / TICK.LABELS font nn

provides fonts for the various parts of the plot. If none of the 4 arguments is supplied, the font that is cited will be used for all parts. Font (part) is followed by the font name and the pointsize. The font name may be in quotes or it may be supplied from the following standard names:

The standard PostScript font set includes:

Times — Times Italic
Times Bold — Times BoldItalic
Arial — Arial Italic
Arial Bold — Arial BoldItalic
Courier — Courier Oblique
Courier Bold — Courier BoldOblique

JOIN.POINTS CONTINUOUS / GAP

specifies that the lines that join the points should either be continuous or should leave a gap if there is any missing data values. JOIN.POINTS is used with the LINE.TYPE subcomand which specifies the plots and overlays that are to have lines connecting their points. If JOIN.POINTS is used by itself it defaults to a solid black line.

LEFT.EDGE nn

provides the size of the left margin in inches. If it is not used, 1 inch is assumed. LEFT.EDGE is honored even when there are multiple plots on the page.

LEGEND SHORT / LONG TOP / BOTTOM / RIGHT

specifies the placement and contents of the legend. NO LEGEND can be used to turn the legend off.

LINE.TYPE PLOT / OVERLAY nn color line-type

specifies the line type for the PLOT and/or any of the 9 optional overlays. The line types are;

```
DOTTED,    SHORT.DASH,   DASHED,   LONG.DASH,
DOT.DASH,   ODD.DASH,    SOLID
```

LINE.TYPE ON and LINE.TYPE OFF can also be used. Color is optional. Black is used if there is no setting. The colors can be any of red, orange, yellow, green, blue, violet, black, brown, white, cyan and magenta.

LINE.WIDTH BORDER / EDGES / GRID / JOIN.POINTS / SYMBOL nn

specifies the line width for the five types of lines used in a plot. The number will usually be between .1 for a very thin line to 2. for a thick line.

PLOTS.PER.PAGE nn BY nn

specifies that there are to be multiple plots on the page. The first number is the number of rows of plots down the page and the second number is the number of columns of plots across the page.

POSITION LABELS LEFT / CENTER / RIGHT

controls the positioning of the labels on the X axis in relation to the ticks marks. POSITION LABELS is usually used when the labels are taken from a LABELS file or generated by the DATE.AXIS subcommand.

RESCALE

requests that the overlay be rescaled and have its own ticks and labels on the right. The ticks and tick labels for the overlay variable are printed on the right edge of the plot. If RESCALE is followed by a list of overlays (RESCALE OVERLAY 2 OVERLAY 3) those overlays will be scaled together on the right axis while the ticks and tick labels for the plot and any unmentioned overlays are on the left.

RIGHT.EDGE nn

provides the right margin. This is used only when there is a single plot on the page. DIMENSION PLOT or PLOTS.PER.PAGE take precedence over this setting.

SCALE nn or vn

SCALE behaves differently in a PostScript plot than it does in a character plot. The number provides a scaling factor for the symbols. 1s the assumed scale. If a variable name is used instead of the number, each case is scaled individually depending on the value of the scale variable for that case.

SHOWPAGE ON / OFF

used to control the printing of multiple plots per page when you wish to specify your own plot sizes or to use multiple plot commands.

SPACE CORNERS / TICKS / LEGEND / BETWEEN

followed by a number, provides the space in inches to be used for: 1) the distance from the corner to the first tick, 2) the area to the right or left of the plot that is used for ticks and their labels, 3) the space to be used when the LEGEND is on the RIGHT and 4) the size of the space between plots when PLOTS.PER.PAGE is in effect.

SPECIFICATIONS for ticks (nn long) (nn wide)
INSIDE / OUTSIDE / THROUGH axes

SPECS for ticks is used to locate the tick marks in relation to the axes and (optionally) to control the length and width of those tick marks.

SYMBOL PLOT / OVERLAY nn color symbol

provides the symbol to be used for the plot and the overlays. NO SYMBOL may be used to turn the symbols off. When NO SYMBOL is used with JOIN.POINTS or LINE.TYPE the line is continuous without the interruption of the symbol. Possible symbols are:

```
SQUARE     TRIANGLE      STAR      CIRCLE
PLUS       DIAMOND       HASH      X
Y          Z             DOT       PERIOD
```

Color is optional and defaults to black. Any of red, orange, yellow, green, blue, violet, brown, black, whit, cyan and magenta can be used.

TICK.LABELS X LEFT / CENTER / RIGHT cs

In PostScript plots only the additional arguments for the TICKS.LABELS subcommand controls the placement of the tick labels on the X axis relative to the corresponding tick. Normally the ticks labels are centered beneath the corresponding tick.

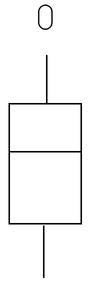
TOP.EDGE nn

specifies the space between the top of the paper and the edge of the plot area. This is assumed to be 1 inch. TOP.EDGE is always honored even when there are multiple plots on the page.

6
BOXPLOT:
A PostScript Command

The BOXPLOT command is designed to produce boxplots in PostScript format. It can be used by itself or as part of a PostScript block. The language for BOXPLOT is derived from both the existing BOXPLOT subcommand in EDA and the PLOT command. Most of the EDA BOXPLOT subcommands were retained with the new PostScript features using the language of the PLOT command.

The BOXPLOT subcommand in EDA constructs all of the plots using Tukey's formulas for hinges, outliers, and whiskers. The BOXPLOT command permits a choice of method. The assumed method is TUKEY (also called the STANDARD method). The alternate method is based on percentiles. The choice of method determines the defaults for the printing of outliers and whiskers.



The boxed area is bounded by the upper and lower hinges or quartiles. The whiskers extend to the defined extreme values. The outliers are any data points that are beyond the extremes.

The method chosen, determines the length of the whiskers and whether outliers are printed. The median, however, is always the same.

The differences between the two methods are slight. The TUKEY method calculates the box, the whiskers, and the outliers using hinges while the PERCENTILES method uses quartiles. The hinge is the location halfway between the outer edge and the median, while a quartile is one fourth of the way between the low and high value. The effect is that the hinges often lie slightly closer to the medians.

There are options to control the printing of outliers and the extent of the whiskers in either method. There are also a wide selection of formatting options. Several boxplots may be combined on a single page. Boxplots can also be combined with regular PLOTS and TEXT.WRITER output. The following is a typical boxplot command to produce a single page of output:

```
BOXPLOT MyFile, PR 'test.ps';
        POSTSCRIPT PORTRAIT,
        BOX Rate, BY Cause $
```

6.1 BOXPLOT Identifiers

A P-STAT system file is always required as input to the BOXPLOT command. In addition, a print file should be specified when the command is used by itself to produce PostScript output.

```
BOXPLOT MyFile, PR 'test.ps',
```

The name for the print file should not be used when BOXPLOT is part of a current PostScript block. PR may be used as either an identifier or subcommand. A print file must be named before the POSTSCRIPT subcommand is issued.

6.2 Using BY Variables

BY provides the name of 1 to 15 by variables. BY variables can be any mixture of character or numeric variables. Value labels may be supplied for the numeric BY variables. A separate box plot is produced for each level of the combined by variable values. Each box plot is printed on a separate page unless one of the subcommands which control plot placement is used. If the BY variable fills a page and it is not the last level of the BY, the PostScript instruction SHOWPAGE, which causes a completed page to print, is issued even when SHOWPAGE is turned off.

If there are 15 by groups and 6 plots on the page, there will be 3 pages per analysis variable. If SHOWPAGE is off, the first 2 pages will print automatically, but the final page will not print unless SHOWPAGE is turned on before the job completes.

All the plots for a BY variable are usually scaled using the overall low and high values of the analysis variable. However, the subcommand SCALE SEPARATE causes the low and high values to be recalculated for each BY group. Identical ranges make it easier to compare boxplots. However, if the ranges of the BY groups are not similar, using the same scale for each group looses some of the detail for those groups with small ranges.

6.3 Other BOXPLOT Identifiers

The LABELS subcommand provides the name for an external file of value labels. The there are labels for BY variables, both the extended labels, if available, and the value labels are used in the BY legend. The information in the labels file is also used to label any groups that are defined with the GROUP subcommand. If there is an extended label for the analysis variable, it is used to label the Y axis.

```
SAVE.LABELS Box.lab;
   Process (1) Blue Lab (2) Green Lab (3) Red Lab /
   Rate ( XL 'Rate of Change' ) /
$
BOXPLOT Rates, LABELS Box.Lab, PR 'test.ps';
   POSTSCRIPT,
   BOX rate, GROUP process;
$
```

Predefined titles are used if the TITLES identifier is present or if TITLES are suppled as subcommands.

```
TITLES '.date.' $
BOXPLOT Rates, LABELS Box.Lab, TITLES, PR 'test.ps';
   POSTSCRIPT,
   BOX Rate, GROUP Process;
$
```

The current date will print at the top of each boxplot. If you are creating a page with multiple box plots and wish to have an overall title, you must use the POSTSCRIPT command to begin a PostScript block and fill one page at a time with boxplots.

6.4 SUBCOMMANDS

The BOXPLOT command requires subcommand information. The analysis variables are cited with the BOX subcommand. The analysis method, the groups to be used, and sizing information is all provided by subcommands. There are also many options for controlling labels, titles, and fonts. The use of subcommands permits several different boxplots to be executed from a single command.

```
BOXPLOT Rates, LABELS Box.Lab, TITLES, PR 'test.ps';
   POSTSCRIPT,
   BOX Rate, GROUP Process;
   BOX Rate, GROUP Treatment;
$
```

6.5 Controlling Content

A BOX subcommand is required. It provides the name of one or more analysis variables. A separate analysis is done for each such variable.

```
BOXPLOT Solar, PR 'solar.prt';
   POSTSCRIPT,
   BOX efficiency voltage;
$
```

GROUPS provides the name for a group variable. A separate boxplot bar is drawn for each level of the group variable. The boxplot itself has one bar for each group defined by the group variable. If there are BY variables, there are as many boxplots as there are values for the BY variables.

The GROUPS variable must be a numeric variable and the bars are labelled with numbers unless labels are provided in the labels file. Usually all possible levels or categories of the groups variable are represented in the boxplot. The LEVELS subcommand is used to select a subset of the levels for a given boxplot.

```
BOXPLOT Solar,  LABELS solar.lab,  PR 'solar.prt';
   POSTSCRIPT
   BOX efficiency, GROUP treatment;
$

BOXPLOT Solar,  LABELS solar.lab,  PR solar.prt;
   POSTSCRIPT,
   BOX efficiency, GROUP region, LEVELS 1 TO 10 19 23;
$
```

The two methods used for computation are the STANDARD or TUKEY method and the PERCENTILES methods. When the Tukey method is used the outliers are usually printed. When the PERCENTILES method is used, the outliers are usually omitted. This behavior can be controlled by using OUTLIERS or NO OUTLIERS to specify whether or not outliers should be displayed. Even though the outliers are omitted from the plots, their values contribute to the statistics, which are based on the complete set of data.

```
BOXPLOT Solar,  LABELS solar.lab,  PR 'solar.prt';
   BOX efficiency, GROUP region, LEVELS 1 TO 10, 19 23'
   METHOD PERCENTILES,  OUTLIERS;
$
```

The Y variable can be transformed to a LOG10 scale by using the subcommand LOG10. When LOG10 is used, the Y axis and the tickmarks are appropriately placed and scaled. LOG10 is the same as specifying LOG10 ON. LOG10 OFF can be used to turn the transformation off.

Sometimes, missing data will result in a plot that has no valid data points. If you are creating many plots with BY variables you may well want these plots printed. If you do not wish this printout, use the subcommand OMIT EMPTY PLOTS.

6.6 Subcommands: Plot Size

The factors that determine the size of the plot are the size of the paper and its orientation, the margins, and the number of plots that are to be placed on each page. Each of these values can be explicitly supplied. In the simplest form, the BOXPLOT command assumes:

1. Paper size is 8.5 by 11 inches
2. Orientation is landscape (that is 8.5 high by 11 wide)
3. Margins are 1 inch on all sides
4. There is 1 boxplot per page

The DIMENSION PAPER subcommand provides the height and the width of the physical sheet of paper in inches. The paper size should not be increased beyond the actual paper size. Decreasing the paper size is one way to reserve space on a page for other output. If DIMENSION PAPER is used, the first number is the height and the second number the width.

```
DIMENSION PAPER 11.5 by 8,
```

If DIMENSION PAPER is not used, the program determines whether 8.5 is the height or the width by looking at the orientation. If the orientation is PORTRAIT, 8.5 is assumed to be the width and 11 the height. If the orientation is LANDSCAPE, 8.5 is assumed to be the height and 11 the width.

The orientation is specified by using either LANDSCAPE or PORTRAIT following the POSTSCRIPT subcommand. If neither is used, LANDSCAPE is assumed

Because the paper is assumed to be 8.5 by 11 inches and the margins are assumed to be 1 inch on all 4 edges, the usual plot area is 6.5inches high and 9 inches wide. If multiple plots per page are requested, they are sized to fit within this 6.5 by 9 inch area. This size can be controlled by using DIMENSION PLOT followed by the height and width desired. The placement of a plot within that area depends on the top and left margins.

```
BOXPLOT Solar, PR 'solar.prt';
   POSTSCRIPT,
   DIMENSION PLOT 3 by 5,  BOX efficiency;
```

This command produces a plot that is 3 inches high and 5 inches wide placed 1 inch from the top margin and 1 inch from the left margin of an 8.5 by 11 inch sheet of paper.

The margins can be changed by using the MARGINS subcommand.

```
MARGINS 2 2 1.5 1.5
```

specifies a top and bottom margins of 2 inches each while the left and right margins are 1.5 inches each. The order is TOP, BOTTOM, LEFT, and RIGHT.

```
MARGINS 2,
```

specifies that all 4 margins are to be 2 inches wide. The MARGINS subcommand requires either 1 or 4 numeric arguments. To specify a single edge use TOP.EDGE, BOTTOM.EDGE, LEFT.EDGE, or RIGHT.EDGE.

```
MARGINS 1.5, TOP.EDGE 2,
```

This sets all margins to 1.5 inches and then sets the top margin to 2 inches.

6.7 Multiple Plots on a Page

There are three ways to produce multiple plots on a page. If the variables are all in the same P-STAT file, the easiest way is to use the PLOTS.PER.PAGE subcommand. For example:

```
PLOTS.PER.PAGE 3 by 2,
PLOTS.PER.PAGE 1 by 4,
```

The first illustration requests 6 plots per page, 3 down the page and 2 across the page. The second requests 4 plots across each page. The space inside the margins is evenly divided with space left between the plots unless dimensions for the plots have been specified. If you do request multiple plots on a page and specify the plot size, you must consider the spacing carefully so that the plots do not overlap. Another consideration when requesting multiple plots is the fonts that will be used. A large font size on a small plot will not produce a legible result.

A second way to place multiple plots on the page is to take control of the pagination by turning SHOWPAGE OFF and then placing each plot individually.

```
BOXPLOT Solar, LABELS solar.lab, PR 'solar.prt';
   POSTSCRIPT LANDSCAPE, DIMENSION PLOT 2 BY 3,
   SHOWPAGE OFF,
   BOX Voltage,  GROUP Treatment;
```

```
$
BOXPLOT Lunar, LABELS lunar.lab, PR 'solar.prt';
   POSTSCRIPT LANDSCAPE, DIMENSION PLOT 2 BY 4,
   TOP.EDGE 4.5,  BOX Lumens,
   SHOWPAGE ON;
$
```

When you use this procedure you can create an output page with plots from several different data sets.

The final procedure is similar except that it combines the output from several different commands. The easiest way to do this is to start with a POSTSCRIPT command which sets the orientation, specifies the print file, and defines the fonts that will be used. There are examples in the POSTSCRIPT chapter showing the output from several commands combined to make a single page of printout. This combination of commands comprises a “PostScript block

6.8 Borders, Edges, Spaces and Ticks

Each BOXPLOT on the page can be surrounded by a border by using the BORDER subcommand. The BORDER is a solid line that is placed just outside the plot dimensions in the margin of the page or the space between plots. The border is located at the outer edge of the plot area. This distance can be changed by adding a numeric argument to the BORDER subcommand.

```
BORDER .1,
```

places the border 1/10 of in inch outside the plot area. You will usually want to provide at least a small border area as a border that sits at the edge of the plot may appear very close to the labelling on the axes. If you use the LINE.WIDTH identifier, described below, to print a heavy border, the border will intrude on the plot unless the size for the border area is provided.

The area around the boxplot bars, inside the labels usually has a solid left and bottom edge and no edge on the top or the right. This can be controlled with the EDGES subcommand. To have solid edges on all four sides use:

```
EDGES   T B L R,
```

Any combination of T B L R to specify the Top, Bottom, Left, or Right edges of the plot may be used.

The SPACE subcommand is used to control the space for each bar in the boxplot, the space between the bars, and the space between the boxes when multiple plots per page are requested. It can be used to change the distance between the corner of the plot and the first tick on the Y axis. It can be used to provide more (or less) space for the ticks and labels on the Y axis or for the optional legend at the right. SPACE is followed by 1 of the following six arguments and the amount in inches that is desired.

1. CORNER space between the CORNER and the first tick mark. 1/10 of an inch is assumed.
2. LEVELS space between the individual boxplot bar. 1/4 of an inch is assumed.
3. BAR width for each bar. 1/4 of an inch is assumed
4. PLOTS between plots when PLOTS.PER.PAGE is used. 1/4 of an inch is assumed.
5. TICKS space for ticks and label on the Y axis. 1/2 of an inch is assumed.
6. LEGEND space for the legend if legend has been requested. 1 inch is assumed.

The assumed widths for the bars and the spaces between bars is a maximum. If there are many levels of the group variable, the widths of the bars and the space between them is adjusted so that as many as possible can fit within the specified plot size.

```
BOXPLOT Lunar, LABELS lunar.lab, PR 'solar.prt';
   POSTSCRIPT LANDSCAPE, PLOTS.PER.PAGE 2 by 3,
   BOX Lumens, SPACE PLOTS .15,  SPACE TICKS .25,
   SHOWPAGE ON
$
```

The edge of the plot on the Y axis has tick marks spaced along its length. These ticks marks are generated automatically by the BOXPLOT command. The TICKS subcommand permits you to specify where you want the marks placed and provides the numbers to be used in labelling those ticks.

```
TICKS 1 3 5 7 9,
TICKS 1 TO 20 BY 2,
```

If there are edges on both the left and the right side of the plot, the tick marks appear on both edges. The numeric labels for the ticks only appear on the left edge. TICKS will not be honored if the LOG10 subcommand is used.

Sometimes when the plot size is very small, the command generates so many ticks that there is little space between the lines. You can let the program generate appropriate ticks and tick marks and control this by setting the maximum number of ticks that are to be used. This is done with the MAX.TICKS subcommand.

```
MAX.TICKS 8,
```

The TICKS can be reset to the original values computed by the program by using the RESET TICKS subcommand.

A further control on the Y axis is the RANGE subcommand. Both a low and a high value must be provided. The RANGE subcommand controls only the range of the ticks that are printed and does not affect the calculations. However, the range must be large enough to hold the box defined by the hinges or quartiles. If the range is too small nothing will be displayed.

From 1 to 4 grid lines can be specified for the Y axis with the GRIDS subcommand. The GRIDS subcommand requires the grid number (1, 2, 3, or 4) and the location on the Y axis where the grid line is to be drawn. A fourth optional argument is the line type for the grid. Each grid line can have a different line type. The choices are:

DOTTED DOT.DASH SHORT.DASH DASH
LONG.DASH ODD.DASH SOLID

```
GRID 1 5. SOLID,   GRID 2 9.5  DOTTED,
```

The assumed line width for all lines is .5. This is 1/144 of an inch since 1/72 of an inch is a basic PostScript unit of measurement. The LINE.WIDTH subcommand can be used to provide different line widths for the various lines.

```
LINE.WIDTH  BORDER 1.5       produces a rather heavy border
LINE.WIDHT  EDGES  1.        produces an edge that is somewhat
                             heavier than usual
LINE.WIDTH  GRID   .2        specifies a narrower line.
```

6.9 Labels and Legends

When BY is used, each plot is headed with a line for each unique combination of by values. The information begins with “BY:” followed by the variable name or the extended label, and the value. There are several options for formatting the BY legend:

1. BY.LABEL NONEno legend appears
2. BY.LABEL VALUESthe variable name is omitted but the values are printed
3. BY.LABEL FULLboth the name and values are printed.

If the BYLABEL subcommand ends with a string in quotes, that string replaces the work “BY”. If the string contains only a blank, the word BY is omitted.

```
BY.LABEL VALUES ' '
```

A small graphic explaining the bars is printed on the right when the LEGEND ON RIGHT subcommand is used. LEGEND OFF turns the legend off. When LEGEND is on, an inch is usually subtracted from the plot area to hold the graphic and its text.

When there are many levels of the group variable and, therefore, many bars in the boxplot, or if there are multiple boxplots on the page each with several bars, it is difficult to label each bar without overlap. FOLD provides a way to control the formatting of these labels on the X axis. The assumption is that these labels, which are taken from the labels file, will be no more than 16 characters. FOLD requires 2 numeric arguments. The first number is the number of characters to be printed on each line. The second number is the number of lines to be used.

```
FOLD LABELS 1 by 8,
```

A 1 by 8 FOLD provides for 8 lines of labelling, each line with a single character taken from the label in the labels file.

6.10 Fonts

The BOXPLOT command supports 4 different fonts for the various parts of the plot.

1. FONT 1 is used for the tick labels on the X and Y axes.
2. FONT 2 is used for the axis labels: the variable name or extended label of the analysis variable on the Y axis and the variable name or extended label for the group variable on the X axis.
3. FONT3 is used for TITLES.
4. FONT4 is used for labelling the BY variables.

If the font is not specified, TIMES ROMAN is assumed for all 4 fonts. The assumed pointsize is 8 points. The fonts and the pointsizes can be changed by using the subcommands FONT, FONT1, FONT2, FONT3, and FONT4. FONT changes all 4 fonts with a single subcommand. The other 4 font commands change only the single designated font.

All of the FONT subcommands are followed by the name of the font and an optional pointsize. The font name does not need to be in quotes if it is one of the following:

```
TIMES              ARIAL              COURIER
TIMES BOLD         ARIAL BOLD         COURIER BOLD
TIMES ITALIC       ARIAL ITALIC       COURIER OBLIQUE
TIMES BOLDITALIC   ARIAL BOLDITALIC   COURIER BOLDOBLIQUE
```

If you wish to use another font just enclose the entire font name in quotes. Some examples are:

```
FONT  TIMES BOLD 10
FONT2 ARIAL BOLDITALIC
FONT3 'ZapfChanceryMediumItalic' 12
```

6.11 Notches and Whiskers and Symbols

Notches are used to compare the bars in a boxplot. The notches, usually a graphic similar to angle brackets, are placed symmetrically around the median so that two bars with non-overlapping notched intervals can be considered significantly different at about the 5% level. Notches are only produced if the NOTCHES subcommand has been used.

The whiskers in the plot usually extend from the ends of the box to the location of the last actual data point that is within the inner fences (the first extreme value). When the method that is used is PERCENTILES, the whiskers extend to the locations that mark the 10th and 90th percentiles. The extent for the whiskers can be controlled by using the WHISKERS subcommand.

1. STANDARD extends the whiskers to the first case inside the fences or 10.90 limits.

2. USE.10.90 is only available in the PERCENTILES method and extends the whiskers to the value at the 10th and 90th percentile.

3. MIN.MAX extends the whiskers to the highest and lowest observed values in the group. If MIN.MAX is used there are no outliers

The following are possible combinations for method and whiskers.

```
METHOD PERCENTILES, WHISKERS STANDARD
METHOD TUKEY, WHISKERS MIN.MAX
```

If there are 2 or more data points in a bar, the box is always drawn unless the MINIMUM.CASES subcommand is used. If there are fewer than the specified number of data points, they are individually drawn and the box itself is omitted. The symbol that is used to represent a case is a star (asterisk).

```
MINIMUM.CASES 10,
```

When MINIMUM.CASES is set to 10, any group level with 10 or more cases has the box drawn and any group level with less than 10 cases has each case represented with the star.

The symbol used to represent a DATAPOINT or OUTLIER, a far OUTLIER, or the NOTCHES can be changed by using the SYMBOL subcommand. Usually the NOTCHES are left and right angle brackets. outliers and any data points printed because the plot has fewer than max cases values are represented by a “*” and far outliers are represented by the letter “O”.

The SYMBOL subcommand is followed by one of DATAPOINT, OUTLIER, or NOTCHES and the selected symbol. Arguments for SYMBOL may be any of the standard plot symbols:

```
POINT       PERIOD    SQUARE     TRIANGLE   STAR   CIRCLE   PLUS
DIAMOND     HASH      DOT         X          Y      Z

SYMBOL DATAPOINT SQUARE,
SYMBOL OUTLIERS  CIRCLE,
SYMBOL NOTCHES   PLUS,
```

SCALE provides a Scale factor for drawing the symbols. The assumed scale is 2. The scale value can be fractional. If you are printing multiple plots per page, the scale of 2 is probably too large.

6.12 Frequencies and Titles

The frequencies for each bar are usually printed just above the whisker. This position can be changed by using the COUNTS command. The possible arguments for COUNTS are:

1. ABOVE counts are printed above the top whisker. This is assumed.

2. BELOW counts are printed below the bottom whisker.

3. INSIDE counts are printed inside the top of the box.

The values at the median are not printed unless they are requested with the MEDIANS subcommand. No argument is needed. The median value is printed just above the line that designates the median within the box.

TITLES and NO TITLES can be used as subcommands to selectively turn the titles on an off. In addition, the TITLES subcommand allows single top or bottom titles to be defined. Titles that are defined as subcommands do not exist after the command ends.

```
TITLES '.DATE.' $
BOXPLOT MyFile, PR 'Test.ps';
    POSTSCRIPT,
    TITLES T2 'Center this text over the boxplot',
    TITLES T3 LEFT 'Text for the third left title',
    COUNTS BELOW, MEDIANS,
    BOX Measures;
$
```

6.13 RANGE FOR THE Y AXIS

The range used that is used on the Y axis for a given plot depends on the method selected and the combination of subcommands that are used.

If RANGE is specified, that range takes precedence and is used for scaling the boxplot no matter what other subcommands are used. The statistics are computed first and are based on the entire set of data values. If this is a BY situation, all the plots will print using the same scale and ticks. When RANGE is used, SCALE TOGETHER is automatically in effect.RESET RANGE resets RANGE to default values.

If RANGE is not specified, the scaling depends on:

1. METHOD In the STANDARD (Tukey) method, the range is set to encompass all outliers. In the PERCENTILES method, the ranges is set to encompass the whiskers.
2. OUTLIERS However, if outliers are excluded, the range extends only to encompass the whiskers and if outliers are included, the range covers all the data values.
3. SCALE If SCALE TOGETHER is used, all the plots in a BY group will have the same range, one that is large enough to encompass all the values. If SCALE SEPARATE is used, each level of the BY group will be scaled individually. SCALE TOGETHER is the assumed setting.
4. LOG LOG plots are always scaled to a full LOG unit.

BOXPLOT EXAMPLES

SUMMARY

BOXPLOT

```
BOXPLOT Myfile,  PR 'Test.ps';
    POSTSCRIPT,
    BOX Measures;
$

BOXPLOT Myfile,  BY State,  LABELS 'My.lab',  PR 'Test.ps';
    POSTSCRIPT,
    BOX Revenue,   GROUP Industry;
$
```

The BOXPLOT command is designed to produce 1 or more boxplots per page with PostScript controls. It may be used by itself or in conjunction with the POSTSCRIPT, PLOT, TEXT.WRITER, and SURVEY commands.

Required Identifiers:

BOXPLOT fn

provides the name for the P-STAT system file

PR fn

is required when the command is used by itself to produce PostScript output.

Optional identifiers

BY vn vn

provides the name of 1 to 15 BY variables. A separate box plot is produced for each level of the combined by variable values. BY variables can be any mixture of character or numeric variables.

LABELS fn

provides the name for an external file of value labels which are used to label the levels of any group variable and to provide labels for the X and Y axes.

TITLES

print predefined titles.

Required Subcommands

BOX vn vn vn

Provides the name of one or more variables to be used as the analysis variables in the boxplot.

Optional Subcommands

BORDER nn

Provides a border around the plot. The optional numeric argument can be used to change the position of the border in relation to the margin.

BOTTOM.EDGE, TOP.EDGE, LEFT.EDGE, and RIGHT.EDGE nn

provide the size in inches for the specific margin. The default margins are set to 1 inch.

DIMENSION PAPER nn BY nn.

Size of page in inches. 8.5 by 11 assumed. The first number is the height of the paper

DIMENSION PLOT nn BY nn

Size of plot area in inches. 6.5 by 9 assumed.

EDGES T B L R

Choose any combination of T B L R to specify the desired plot edges (Top, Bottom, Left, or Right) desired. EDGES B L (bottom/left) is assumed.

MARGINS nn nn nn nn

provides the size in inches for the 4 margins to be used at the top, bottom, left, and right edges of the paper. If a single argument is provided, that amount is used for all 4 margins.

PLOTS.PER.PAGE nn BY nn

produces multiple boxplots per page. The first number is the number down the page. The second number is the columns of boxplots across the page. The assumption is a single plot per page.

POSTSCRIPT (LANDSCAPE / PORTRAIT)

Start PostScript output. The orientation may be specified either as an argument for POSTSCRIPT or separately. LANDSCAPE is the assumed orientation.

PR fn

Provides the name for the print destination. A print destination must be provided before POSTSCRIPT is requested.

SHOWPAGE (ON / OFF)

If showpage is off, several boxplot commands using their own their own top and left margins and plot dimensions can be placed on a single page. The final plot must have SHOWPAGE ON or the boxplots will not print.

SPACE (CORNERS / LEVELS / BARS / PLOTS / TICKS / LEGEND) nn

The SPACE subcommand requires 1 of the 6 arguments followed by the space in inches to be used:

1. between the CORNER of the plot and the first tick mark. 1/10 of an inch is the default.

2. between the LEVELS (the individual boxplot bars). 1/4 of an inch is the default.

3. for each boxplot BAR. 1/4 of an inch is the default

4. between plots when PLOTS.PER.PAGE is used. 1/4 of an inch is the default.

5. for TICKS and labels. 1/2 of an inch is the default.

6. for a LEGEND if legend has been requested. 1 inch is the default.

Optional Subcommands Appearance

BY.LABEL (NONE / VALUES / FULL) cs

When BY is used, each plot is headed with a line for each unique combination of by values.

If NONE is specified there will be no BY information. If VALUES is specified, the variable name is omitted. If the final argument is a string in quotes, that string replaces the “BY:”.

FOLD LABELS nn by nn

FOLD provides a way to format the labels for the X axis when there are so many levels of the group variable that the labels would overlap each other. The first number is the number of characters to be printed per line. The second number is the number of lines to be used. Labels must be 16 characters or less.

FONT

sets all 4 fonts to the specified font and point size. The font name may be in quotes or it may be one of the following:

```
TIMES               ARIAL               COURIER
TIMES BOLD          ARIAL BOLD          COURIER BOLD
TIMES ITALIC        ARIAL ITALIC        COURIER OBLIQUE
TIMES BOLDITALIC    ARIAL BOLDITALIC    COURIER BOLDOBLIQUE
```

FONT 1 is used for the tick labels on the Y axis. FONT 2 is used for labels. FONT 3 is used for titles. FONT 4 is used for the legend. If FONT is not specified, the Times Roman 8 point font is used for everything.

FONT1 - FONT4

Like FONT above except that the supplied definition only applies to the single specified font. For example:

```
FONT1 TIMES BOLD 10
```

LINE.WIDTH (BORDER / EDGES / GRID) nn

The first argument must be one of BORDER, EDGES, or GRID The second argument is the line width given in units of 1/72 of an inch. The assumed width for all of these is .5 (one half of 1/72 of an inch).

MAX.TICKS nn

Maximum number of tick values for the Y axis.

RANGE nn TO nn

Provides the range to be used for the Y axis. This is a printing parameter. The statistics are computed from the full set of data.

SCALE nn

Scale provides a Scale factor for drawing the symbols. The assumed scale is 2. The scale value can be fractional.

SCALE (TOGETHER / SEPARATE)

This form of SCALE is used with BY to determine whether each plot should be scaled using its own range information (SEPARATE) or whether all should be scaled using a range that is large enough to encompass all BY groups (TOGETHER). TOGETHER is the default when OUTLIERS are to be printed. SEPARATE is the default when NO OUTLIERS has been specified. RANGE, if specified, takes precedence.

SYMBOL for (NOTCHES / OUTLIERS / DATAPOINT) symbol

Normally the NOTCHES are left and right angle brackets. Outliers and any data points printed because the plot has fewer than max cases values are represented by a “*”. Far outliers are represented by a “O”. The SYMBOL subcommand is used to select some other representation.

The argument may be any of the standard plot symbols:

```
POINT    PERIOD     SQUARE      TRIANGLE    STAR
CIRCLE   PLUS       DIAMOND     HASH        DOT
X        Y          Z
```

TICKS nn nn nn nn nn

Provides tick values to be used for the Y axis.

WHISKERS (STANDARD / MIN.MAX / USE.10.90)

specifies the extent of the whiskers. STANDARD extends to the first case inside the inner fences. USE.10.90 is only available in the PERCENTILES method and extends the whiskers to the value at the 10th and 90th percentile. MIN.MAX extends the whiskers to the highest and lowest observed values in the group.

Optional Subcommands Contents

COUNTS (ABOVE / BELOW / INSIDE / OFF)

Specifies the position of the frequencies in each boxplot. ABOVE and BELOW are in relation to the whiskers. INSIDE is inside the box. The assumption is COUNTS ABOVE the whiskers.

GRIDS (1/2/3/4) nn linetype

GRIDS provides for up to 4 horizontal grids at the location specified by the number. THe grid number (1, 2, 3, or 4) and the value are both required. The linetype of the grid is optional and is assumed to be solid. Linetype may be any of:

```
DOTTED      DOT.DASH   SHORT.DASH   DASH
LONG.DASH   ODD.DASH   SOLID    GROUP  vn
```

GROUPS vn

provides the name for the group variable. A separate boxplot bar is drawn for each level of that variable. The levels identifier can be used to select specific levels. The GROUP variable must be a numeric variable but it can have labels supplied in the labels file.

LEGEND (ON/OFF)

Print a legend on the right explaining the bar graphics.

LEVELS nn nn to nn BY nn

provides a list of values indicating the levels of the group variable that are to be used. This is similar to doing an IF and selecting values in the programming language.

LOG10 (ON/OFF)

if on, specifies that boxplot variable is to be transformed and the Y axis printed in standard LOG10 notation.

METHOD (STANDARD / PERCENTILES)

specifies how the boxes and whiskers are to be computed. The standard (Tukey) method uses hinges and fences. The PERCENTILES method uses the quartiles and the 10th and 90th percentiles. If METHOD is not used the standard method is assumed.

MINIMUM.CASES nn

If there are fewer than the minimum cases each case is individually represented rather than drawn in a boxplot.

NOTCHES

Angle brackets are used to indicate the area around the median which can be used when comparing levels.

~~O~~MIT EMPTY PLOTS

Plots that have no good data values are not to be printed.

OUTLIERS / NO OUTLIERS

Outliers are normally presented in the STANDARD method but not in the PERCENTILES method. They can be requested or turned off in either method.

RESET (RANGE / TICKS)

Resets RANGE or TICKS to default values.

TITLES arg arg cs

TITLES and NO TITILES can be used. In addition, TITLES used as a subcommand can be used to define a single top or bottom title.

```
TITLES T3 LEFT 'Text for the third left title',

TITLES T4 'This title text will be centered',
```

7
HIST:
Character Histograms

7.1 HIST: The Histogram command

The HIST command produces histograms — graphs in which frequencies are represented by areas in the form of bars. HIST produces histograms for every variable in the file, scaling the histograms to fit the current output destination. The user may request the number of bars desired and the values for those bars. Subgroups may be defined and cumulative histograms, in which the number of cases for each subgroup are added to the previous subgroup, may be requested. There are options to set the frequency scaling (scale the bars) and to define the number of LINES desired. In addition, a number of histograms can be produced with the same scales and bar sizes, or each histogram can be scaled individually.

7.2 SIMPLE HISTOGRAMS

The histogram command is illustrated in Figure 7.1. HIST requires an input file. If a filename is not specified, the last referenced file is used. The HIST command produces a histogram for *every* variable in the file unless variable selection is done using KEEP or DROP in PPL phrases.

The histogram is scaled to fit the current output device. The output width controls the number of *columns* available for the bars. An output width of 80 columns is assumed for terminals and 132 columns for disk files and printers. The OUTPUT.WIDTH (OW) command and general identifier may be used to reset this.

The LINES setting determines the *height* of the histogram. However, in an interactive run, the SCREEN setting takes precedence over the LINES setting. A SCREEN setting of 22 is assumed for terminals, unless it is reset using the SCREEN command. A LINES setting of 59 is assumed for disk files and printers, unless it is reset. The result of the default settings is that the entire histogram fits on the screen and can be seen without breaks. If the histogram is directed to a disk file or printer, it is enlarged to take advantage of the increased room available.

If you wish to see how the histogram will look when sent to the printer, you can override the screen setting by providing an explicit LINES setting as part of the HIST command. For example:

```
SCREEN 22  $
HIST   CarFile  [ KEEP  Wt.Gps ],  LINES 59  $
```

The histogram no longer fits on a single 22 line screen, but instead uses 59 lines. The screen holds (breaks) so you see the histogram — press return to continue. When the histogram prints, it does not break. Similarly, if you wish to print the histogram as it appears on the terminal, use OUTPUT.WIDTH in a PRINT.PARAMETERS command and use LINES to specify that only 22 lines be used for histograms directed to that printer or disk file:

```
PRINT.PARAMETERS  PrtFile,      OUTPUT.WIDTH 80  $
HIST  CarFile  [ KEEP  Wt.Gps ],  LINES 22,  PR  PrtFile  $
```

Both LINES and OUTPUT.WIDTH can be used as either commands or as general identifiers within the HIST command. SCREEN is a separate command. See the chapter "Simple Commands and Keywords" in the manual P-STAT: Introductory Manual".

Figure 7.1 Histogram, Input and Output

```
   SCREEN 20  $

   HIST  CarFile  [ KEEP  Wt.Gps ], OW 72  $


   FILE work
N=406     BARS=6     MEAN=2.79557     SD=1.17172

            wt.gps
    N                                                            PCT
      +
  149 +          ********                                     + 36.7
      +          **     *                                     +
      +          ** 149 *                                     +
   99 +          **     * ********                            + 24.4
      +          ******** **    **                            +
   71 +          ******** ** 99 ** ********                   + 17.5
   45 + ******** ******** **    ** ** 71 **                   + 11.1
   41 + ** 45 ** ******** ******** **    ** ********          + 10.1
    1 + ******** ******** ******** ******** ** 41 ** ****1*** +  0.2
      -----++-------++-------++-------++-------++-------++-----
            1        2        3        4        5        6


   AGAIN,  CUMULATIVE  $

   FILE work
N=406     BARS=6     MEAN=2.79557     SD=1.17172

            wt.gps
    N                                                            PCT
      +
  406 +                                     ******** ******** +100.0
  364 +                            ******** **     * **     * + 89.7
  293 +                   ******** **     * ** 405 * ** 406 * + 72.2
      +                   **     * ** 364 * **     * **     * +
      +                   ** 293 * **     * ******** ******** +
  194 +          ******** **     * ******** ******** ******** + 47.8
      +          ** 194 * ******** ******** ******** ******** +
      +          **     * ******** ******** ******** ******** +
   45 + ***45*** ******** ******** ******** ******** ******** + 11.1
      -----++-------++-------++-------++-------++-------++-----
            1        2        3        4        5        6
```

7.3 Output of HIST

The output of the HIST command is a bar graph that shows the frequency of an observation. Figure 7.1 shows the input and output of HIST for the variable Wt.Gps in the file CarFile. The left vertical axis gives an approximate count of observations for each category; the numbers in the bars are the actual number of observations for each category. The right vertical axis specifies the percent of the total counts accounted for by each category. The numbers on the horizontal axis represent categories. They may be discrete categories, as in Figure 7.1, or they may represent bar limits for continuous variables. If the high or low values of a category are contained within its limits, then these actual highs and lows are printed — *not* the bar limits.

HIST, used with the identifier CUMULATIVE, produces a histogram in which each bar is added to the previous bar. CUMULATIVE histograms, illustrated in Figure 7.1, give the percentage of total observations accumulated for each bar.

Figure 7.2 Histogram with Subgroups

```
     HIST   CarFile  [ KEEP  Wt.Gps  Origin, OW 72 ],
       SUB  Origin 2,
       LINES 24  $
```

```
   FILE work
N=327     BARS=6     MEAN=3.03976     SD=1.14913

            wt.gps                SUBGROUP VAR. IS Origin
    N                                                            PCT
      +
   99 +          BBBBBBBB                                     + 30.3
   93 +          BB    BB BBBBBBBB                            + 28.4
      +          BB 34 BB BB 19 BB                            +
      +          BB    BB BBBBBBBB                            +
   74 +          BBBBBBBB AAAAAAAA BBBB2BBB                   + 22.6
   69 +          AAAAAAAA AA    AA AAAAAAAA                   + 21.1
      +          AA    AA AA 74 AA AA    AA                   +
      +          AA 65 AA AA    AA AA 69 AA                   +
      +          AA    AA AAAAAAAA AA    AA                   +
   41 +          AAAAAAAA AAAAAAAA AAAAAAAA AAAAAAAA          + 12.5
      +          AAAAAAAA AAAAAAAA AAAAAAAA AA    AA          +
   22 + BBBBBBBB AAAAAAAA AAAAAAAA AAAAAAAA AA 41 AA          +  6.7
      + BB 18 BB AAAAAAAA AAAAAAAA AAAAAAAA AA    AA          +
    1 + BBBBBBBB AAAAAAAA AAAAAAAA AAAAAAAA AAAAAAAA ****1*** +  0.3
      -----++-------++-------++-------++-------++-------++-----
            1        2        3        4        5        6
```

7.4 Allocation of Ranges

HIST determines the range of data values and divides it up in an appropriate manner. The range is determined in one of three ways:

1. If a description file is supplied, those low and high values are used for the ranges of each of the variables. When DES is used in the HIST command, the range is determined for each variable individually.
2. If the identifier RANGE L H is specified, where L is the low value and H is the high value, then the specified RANGE values are used. This results in a *uniform* range for all of the variables in the input file.

3. If a description file is not specified and RANGE is not used, then the program makes an *extra pass* through the data to find the actual ranges. Individual ranges are determined for each variable.

7.5 HISTOGRAMS WITH SUBGROUPS AND SCALING

Subgroups may be defined within the HIST command. The area of each bar in the histogram is then coded to show the relative frequency of observations in each subgroup.

7.6 Definition of Subgroups

The SUB identifier is used to specify the variable that defines subgroups:

```
SUB  Origin 2
```

The first argument gives the name or position of the variable that defines the groups. The second argument is the number of groups defined by that variable. There can be from two to nine groups. The variables that define groups must be coded 1, 2, 3, and so on. Figure 7.2 shows a histogram of CarFile with two subgroups. The subgroups are identified by the letters A, B, C, and so on.

7.7 Scaling the Bars and Frequencies

Scaling the bars is useful for variables that contain much data over a wide range. The optional identifiers BARS, BAR.STEP and BAR.LIMITS define the scale of the bars. BARS defines the number of bars:

```
BARS 6
```

Figure 7.3 Defining Bar Limits

```
    HIST  CarFile  [ KEEP Mpg ]  $
```

```
   FILE cars
N=398     BARS=16     MEAN=23.5146     SD=7.81598

            Mpg
    N                                                                     PCT
      +
   54 +        ***                                                     + 13.6
      +         * *                                                        +
   43 +        *5*     ***                                             + 10.8
   40 +        *4* *** * * *** *** ***                                 + 10.1
   34 +        * * * * *4* * * * * * * ***                             +  8.5
   28 +    *** *** *4* *3* *3* *3* *3* *3* ***                         +  7.0
   21 +    *2* *** *0* * * *7* *6* *6* *4* *2* ***                     +  5.3
   16 +    *6* *** * * *** * * * * * * * * *8* *21 ***                 +  4.0
  11 + *7* *** *** *** *** *** *** *** *** *** *** *16 *11 *3* *3* *3* +  2.8
      ---+---+---+---+---+-------+---+-------+---+-------+---+-------+---
         9  12  14  16.2 18.5   23.2 25.5   30.5 32.7   37.2 40.8   44.3
        11  13  16  18.2 20.6   25.4 27.5   32.4 34.7   39.4 41.5   46.6
    AGAIN,  BAR.LIMITS 10 15 20 25 30 35 40 45 50  $

FILE cars
 N=398     BARS=9     MEAN=23.5146     SD=7.81598

             Mpg
     N                                                            PCT
       +
    91 +             *****                                     + 22.9
    80 +             *   * *****                               + 20.1
    73 +             * 91* *   * *****                         + 18.3
    66 +       ***** *   * * 80* *   *                         + 16.6
    51 +       *   * ***** *   * * 73* *****                   + 12.8
       +       * 66* ***** ***** *   * *   *                   +
       +       *   * ***** ***** ***** * 51*                   +
    25 +       ***** ***** ***** ***** *   * *****             +  6.3
     8 + **3** ***** ***** ***** ***** ***** * 25* **8** **1** +  2.0
       ----+-----+-----+-----+-----+-----+-----+-----+-----+----
            9    11  15.5  20.2  25.1  30.5  35.1  40.8  46.6
           10    15    20    25    30    35  39.4  44.6
```

The range is divided into six equal parts. BAR.STEP provides a class size for each bar:

```
BAR.STEP 5
```

The numeric argument 5 specifies the largest possible “step” or range for each bar, though the actual bar limits may be within 5 of each other. For example, the range of a bar may be from 9 to 14, but if the highest score in that bar is 13 then the range that is printed goes from 9 to 13.

The identifier BAR.LIMITS specifies a series of scores that define the upper limits of the bars. Figure 7.3 shows a histogram with BAR.LIMITS. The numbers define the upper limit of each bar. If the highest score of a bar is less than the bar limit, the actual score, not the bar limit, is printed. BAR.LIMITS often makes the histogram easier to view.

It is sometimes useful to have a number of histograms produced with the same scales and bar sizes even though the individual variables have different ranges. The identifier ALL may be used to do this. ALL is assumed when RANGE and BAR.LIMITS are used. The opposite of ALL is EACH, which is assumed when DES is used. With EACH, every histogram is scaled individually according to its own ranges.

Frequency scaling (on the left axis) is done automatically for each individual variable unless the optional identifier F.SCALE is specified. P-STAT identifies the largest number (N) in a bar and sets the upper limit of the frequency scaling to the nearest “nice” number that is greater than N and equal to one of the following: 10, 25, 50, 100, 150, 200, and so on. For example, if the largest number for any bar is 135, the scaling of the bars goes from 0 to 150. F.SCALE defines the specific frequency scale to be used for *all* of the variables:

```
F.SCALE 450
```

The numeric argument is the upper limit of the scale; it must be greater than or equal to the largest number in a bar.

SUMMARY

HIST

```
HIST  Scores  [ KEEP  Test.1  TO  Test.5 ]  $

HIST  Scores,    SUB  Sex 2,  CUMULATIVE    $
```

The HIST command produces a histogram for *every* variable in the file. The PPL instructions KEEP and/ or DROP should be used to select the variables for which histograms are desired. Histograms are automatically scaled to fit in the available print area. The number of bars desired and the values for those bars may be specified.

Subgroups may be defined and cumulative histograms may be requested. Options to define the frequency scaling and the number of LINES are available. In addition, multiple histograms may be produced with the same scaling and bar sizes, or each histogram may be scaled individually according to its own ranges.

Required:

HIST fn

provides the name of the required input file. If a name is not provided, the last referenced file is used as the input file. Select the variables for which histograms are desired using either KEEP or DROP.

Optional Identifiers:

ALL

specifies *uniform scaling* for all histograms.

BARS nn

requests the specified number of bars.

BAR.LIMITS nn nn nn

specifies a series of values that define the *upper* limits of the bars. Two through 12 such limits may be provided. The lowest and highest *observed* values are used to label the bars.

BAR.STEP nn

specifies a class size for each bar.

CUMULATIVE

requests cumulative histograms, which show the percentage of observations accumulated for by each bar.

DES fn

provides the name of an input description file used to provide ranges for the histograms.

EACH

specifies *individual* ranges for each histogram. This is assumed when the identifier DES is used. EACH may not be used with RANGE or BAR.LIMITS.

F.SCALE nn

specifies *uniform* frequency scaling for all histograms. The number (nn) is the upper limit of the scale. When F.SCALE is not used, the program does its own scaling. In such cases, the nearest round number that is larger than the largest value in a histogram bar and equal to one of the following is used: 10, 25, 50, 100, 150, 200, and so on.

LINES nn

specifies the total number of lines to be used for the output. When LINES is not specified, 56 lines are assumed unless the SCREEN command has been used (prior to HIST) to set the screen size. (Use the OUTPUT.WIDTH general identifier to specify the number of columns to be used.)

RANGE nn nn

specifies low and high values to provide *uniform* ranges for all the input variables.

SUB vnp nn

defines subgroups. The first argument specifies the name or position of the variable that has subgroup information. The second argument gives the number of subgroups that are defined by that variable.