Handout adapted from Jeff Picka, Fall '01

SAS Graphics for Descriptive Statistics

The Importance of Graphics

Pictures are important to the successful communication of scientific information. Drawing the right pictures and explaining them well is a vital part of statistical analysis, and plots accompany almost all the statistical techniques discussed in the text. The most useful types of pictures are:

- **Histograms**, which display estimates of the probability distribution function or density for the observations *when the observations are independent identically distributed, or otherwise replicated representatively.* These are very useful for identifying features of the data distribution, such as skewness.
- **QQ plots**, for example normal probability plots, are visual indications of whether the data approximately follow a specified probability distribution. They are important primarily in checking assumptions underlying statistical techniques.
- **Scatter plots**, which look at pairs of random variable values, help in visual screening of relationships. **Residual plots** are an important sub-category, used in determining whether what is left after subtracting model-predicted values from observations is sufficiently patternless.

These plots, charts, and diagrams can be created in SAS using `PROC CHART` and `PROC PLOT`, but these procedures produce very low quality graphics fit for printing on line printers. There is almost no point in bothering with these, as you want to be looking at the best resolution graphics available when plotting. SAS provides the procedures `GPLOT` and `GCHART` among others, which are not discussed in the text. Assistance can be found from the **Help** button on the **ToolBox**, or by calling on **SAS System Help** on the window **Help** menus. These bring you to a page with a heading `GRAPHICS`, clicking on which takes you to a page listing `GCHART`, `GPLOT`, `Graphics Statements and Options`, and all high level graphics commands. You cannot access these pages directly from the **Help** Index, since these commands are not part of the basic SAS programming language.

There are many useful books on effective graphical methods that go far beyond those considered in this course. Many works can be found by searching the UMCP Library database under the subjects **Graphic Methods** and **Statistics Graphic Methods**. Among the most useful books are those of Tufte (1983, 1990) and Cleveland (1993),

Plotting Essentials

Plots should always have the following:

1. **A title or a caption**. Readers need to have some idea of what data and variables they are looking at, and what to focus on in the picture.

2. **Sensible axes**. Defaults can sometimes produce odd-looking scatter plots, with points crammed into one tiny region in the plot. You must know how to over-ride these defaults, should they cause problems.

3. **Axis labels**. These may give information about variable names, whether standardized or transformed, etc.

4. **Correct plot shape**. If you are plotting two variables against each other and both have values restricted to the interval [0,1], then the plot should be square and not rectangular.

5. **Identification of plotting symbols**. In a legend or in a caption, you must indicate what the different symbols on a plot mean. For example, if you use a different symbol to identify suspected outliers on a residual plot, you must state that somewhere.

6. **Sensible plotting symbols – which is hard in SAS**. You use these to distinguish related but different points, without obscuring important features of your picture.

PROC GPLOT

`PROC GPLOT` should be used in place of `PROC PLOT` in every plotting routine. The main differences are related to the modifications that you can make to the plot, including adding a legend, changing the general shape, and changing the axis labels. For further details on the options specified below, go to **SAS System Help**, then to **Graphics**, then to **GPLOT**.

Once you reset certain plot parameters in `GPLOT` , they stay that way on any further plots made until you log out or change them to other settings. For example, unless you change the title from plot to plot, the title on your first plot will show up on all subsequent plots. As a general rule, the statement

```
GOPTIONS RESET=ALL;
```

should precede any of the definitions required to create a new plot or chart. You can remove some graphics parameter changes or install them by means of the **Options** submenu of the **Globals** menu in the **Graphics** window. On this submenu, various topics are listed (including **Titles**, **Symbols**, and **Footnotes**). When they are chosen, windows appear listing the titles, footnotes, etc. appear, and specific changes can be made.

Every `PROC GPLOT` should end with the statement `QUIT;` Without this statement, odd things tend to happen.

Adding Titles

Titles are added by means of a `TITLE` statement, which specifies a line of text that you would like to see printed above the graph. A second line of a title can be added by means of the command `TITLE2`, and further lines can be added by putting a suitable number at the end of the command. There are options relating to font size and the installation of boxes around the title, described under `TITLE` in the `GPLOT Options` help box.

The Shape and Size of the Plot

The shape and size of the plot can be controlled by means of a `GOPTIONS` statement inside of a `PROC GPLOT` call. The statement

```
GOPTIONS HSIZE=x VSIZE=y
```

will produce a square plot if `x=y` *and* the region outlined is sufficiently small. If the size is not sufficiently small, you get the default shape and size of plot.

Messing About With The Axes

Three important aspects of axes can be adjusted.

1. Setting the range of values to be displayed. This is very important if you want to have the plot exclude or include extreme values in the data, but can also be used with the `ANNOTATE` command to ensure that lines appear in the correct places.

2. Customizing the labels on the axis to give more informative titles or to include units.

3. SAS tends to add little offsets where the axes meet, for a nice aesthetic effect. If your variables can only take on values between 0 and 1, this will produce quite misleading effects near the origin. If you are adding a line to a plot via `ANNOTATE`, this can also produce odd effects.

To fix all three of these at once, suppose you want the horizontal axis on your plot to run from 0 to 1000 with major tick marks given every 200 units, and you also want no offsets and the axis label "Velocity (furlongs/fortnight)". This is a bit too complicated to stuff into the `PLOT` line as an option, but can be combined into one big print option by the statement

```
AXIS1 ORDER=(0 TO 1000 BY 200) OFFSET=(0,0)
        LABEL=('Velocity(furlongs/fortnight)');
```

placed before the `PROC GPLOT` statement. Within the `PROC GPLOT` statement, the plot statement is then

```
PLOT Y*X / HAXIS=AXIS1;
```

The option `VAXIS=AXIS2` will force the vertical axis to take on attributes specified in the statement that begins with `AXIS2`. More on options for these attribute lines can be found on the `AXIS` subsection of the `GPLOT` help documents.

Messing About With The Symbols

If you wish to change the plotting symbol from PLUS to another character or vary its size, you can specify the symbol by a definition statement of this form

```
SYMBOL1 VALUE=‘‘ ’ ’’ HEIGHT=(0.1CM);
```

and then by specifying the PLOT command within PROC GPLOT as

```
PLOT Y*X / SYMBOL=SYMBOL1;
```

You would think that this would produce symbols that were {′}'s instead of +'s, and would ensure that they were 0.1 cm in height. Due some oddity in SAS, the height comes out as expected, but GPLOT generates little fleurs-de-lys as plotting symbols in place of apostrophes. Other possible units include IN (inches), PCT (percent of graphics output area), and CELLS (SASGraph character cells).

Adding A Straight Line To A Plot

If you need to add lines that are not regression lines, they must be drawn in by other commands. Horizontal and vertical lines can be specified in GPLOT by means of the PLOT options HREF=y and VREF=x, where x and y are the intercepts. If the line required is at an angle, it must be drawn onto the plot with an ANNOTATE=datafile option for PROC GPLOT. The datafile for this option is a program that gives explicit instructions on how to draw in the line that you want.The **Help** pages on this are a magnificent example of appallingly poor documentation.

The ANNOTATE option is invoked by the option

```
PROC GPLOT DATA=somedata ANNOTATE=dread;
```

The annotation datafile is a rather complicated statement that tells **SAS/GRAPH** exactly where to draw the line. For example, the datafile

```
DATA DREAD;
  FUNCTION='MOVE'; X=0; Y=0; XSYS='2'; YSYS='2'; OUTPUT;
  FUNCTION='DRAW'; X=100; Y=100; XSYS='2'; YSYS='2';
                   SIZE=0.8; LINE=1; OUTPUT;
RUN;
```

draws a straight line from the origin to the point (100,100). The first line of the datafile specifies where the line begins, and XSYS and YSYS must be set equal to the character 2, or else the location coordinates will not be in the same units as the axes. The line ends with an output, so that something appears on the graphics window. The second line of the datafile specifies the point to which you wish to connect the initial point. The coordinate variables should be set to character 2 again, and other options are possible to control the width and type of line drawn.

PROC GCHART

This procedure makes bar and pie charts, which are of use of in categorical data analysis. It also can be persuaded to generate histograms. Its syntax is very similar to that of `CHART`, except that there are more options for making everything look nice. More will said about this during discussions on categorical data analysis.

Making a Histogram With GCHART

In the absence of any proper histogram-generating command, you can force `GCHART` to generate a fairly decent histogram through use of the levels option. This is done through the `VBAR` statement, followed by the variable name, a slash, and the option `LEVEL=n`. A criterion for histogram bin width developed by Izenman (1991) suggests that you should start by using n_1 levels, where

$$n_1 = \frac{max - min}{2(IQR)} \times N^{1/3},$$

and N is the sample size, IQR is the interquartile range (described by SAS as `Q3-Q1`), and max and min are the largest and smallest observations. This is only a starting point, and you should try several larger and smaller numbers of levels before settling on a number that gives the best compromise between smoothness and detail. Always remember to label the axes and to include a title, identifying the random variable and the number of points used.

Printing

Printing of plots and charts cannot be done directly on the UNIX cluster machines, but may be possible on the WAM lab machines.

On Cluster machines, you cannot print directly but must export the graphics to a file. From the **File** menu in **Graphics** window, choose **Export**. A window will appear, in which you can select your favorite format for image files. Options include postscript (*.ps*), encapsulated postscript (*.eps*), *.jpg*, *.tif*, *.bmp*, and *.gif* formats. All but the postscript format may require that you open the file in an image editor before printing it, but postscript files can be sent directly to most printers.

If you save your graphics as a *.ps* or *.eps* file, you first need to `ftp` the file to a computer that you can print from. If this is a UNIX machine, you should preview it using the command `ghostview` *filename* or `gv` *filename*, and then print it from there. You could also print it directly using the `lp` *filename* or `qlp` *filename* commands.

In The Event of Disaster

If the printer is printing your file as hundreds of pages of rubbish, you must kill the print job immediately. On Mathnet machines with Solaris, you can do this easily using the Print Manager. On general UNIX machines, the command `lpq` gives you a list of current print jobs.

Identify yours, and find its PID number. This number may also be presented immediately after a `qlp` command executes. Once you have the PID number, type `lprm` *pidnumber*. This will kill the job, eventually. If you hit the kill button on the printer, this is likely to be ineffective and may actually make matters worse by starting the entire print job over again.

References

[1] E.R. Tufte, **Visual Display of Quantitative Information**. Graphics Press, 1983.

[2] E.R. Tufte, **Envisioning Information**. Graphics Press, 1990.

[3] W.S. Cleveland, **Visualizing Data**. Hobart Press, 1993.

[4] A.J. Izenman (1991), Recent developments in nonparametric density estimation. *Jour. of the Amer. Statistical Assoc.* **86**, 205-224.