Increasing the Value of Graphs Using the SAS® Annotate Facility
Reggie Brett, Seminole Electric Cooperative, Inc.

Abstract:
Analysis of large amounts of data typically involves both summary reports and graphs. Combining the reports and the graphs can sometimes simplify the analysis of the data. The SAS® system provides a method by which text strings may be combined with SAS® graphs. This method is known as the Annotate facility and is an option to the GPLOT procedure. The goal of this paper is to demonstrate how, by using the SAS® Annotate facility, text strings and numbers may be incorporated into a SAS® graph thereby increasing its value and reducing the need for multiple reports.

Background:
At Seminole Electric a significant amount of effort is spent analyzing hourly demand data. Hourly demands are defined as the amount of electricity a customer requires during a particular hour. This data is recorded year-round and is used to bill customers and plan for future growth.

Each month, a summary report and a graph are generated to assist in analyzing the demands (Exhibits 1 and 2). The report contains items such as the maximum demand, the peak day and hour, and the total monthly energy. The graph represents one month of hourly demands, and each line on the graph represents one day or 24 hours. The graph helps the analyst to visualize the hour by hour changes in the demands. Used together, the summary report and the graph can assist the analyst in identifying trends, outages, or non-typical loads.

Procedure:
The following describes the process used to create a graph in which numbers and text strings are combined with the graph (Exhibit 3).

The process begins with several data steps which organize the hourly demand data and create the monthly summary data. The information created in these steps is passed to the annotate data set. The annotate data set is a SAS® data set which contains the text strings to be passed to the GPLOT procedure. The data for the annotate data set is the summary results which were created in the previous data steps. The results are passed to the annotate data set where they are converted to text strings for printing on the graph. In addition, other variables are created which contain the coordinates used to locate the text on the graph. By including the ANNOTATE option during the GPLOT procedure, we can reference the annotate data set and place the text strings on the graph. The data in the annotate data set is illustrated in Exhibit 4.

As the exhibit shows, the data from the summary report has been converted to text strings and is now ready for plotting on the graph. While developing the graph, a macro was used to identify the various day types (Sat, Sun, etc.). This macro can be seen in the listing in Exhibit 5.

Each of the symbol statements is assigned to a corresponding line on the graph. The symbol statement identifies the line type and the color for each of the lines plotted on the graph. The line types match the day of the month, that is, day 1 is plotted using line type 1, day 2 is plotted using line type 2, etc.

The colors assigned to each line are determined by the values assigned to the global macro variables C1 through C31. Each macro variable represents a day in the month and is assigned a value by using the SAS® function SYMPUT. The value of each macro variable is a color: blue for weekdays, red for Saturdays, and green for Sundays. A fourth color, black, is used to identify the day of the month on which the greatest hourly demand occurred. The colors are assigned during the data step before the SYMBOL macro is called.

Conclusion:
Analysis of large amounts of data typically involves both summary reports and graphs. The SAS® Annotate facility provides a convenient way to combine the summary results with SAS® generated graphs. The procedure is easy to implement, usually with a minimum of programming changes. This process gives the analyst another tool to use when attempting to analyze the data. By using the annotate facility we increased the value of our graphs, and reduced the number of reports needed to analyze the data.
EXHIBIT 4

Sample Annotate Data Set

<table>
<thead>
<tr>
<th>OBS</th>
<th>FUNCTION</th>
<th>TEXT</th>
<th>POSITION</th>
<th>COLOR</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LABEL</td>
<td>CO-PEAKDAY/HOUR: 45062 ON SAT 25 AT HOUR 8</td>
<td>6</td>
<td>BLACK</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>LABEL</td>
<td>PEAKDAY/HOUR: 46670 ON FRI 24 AT HOUR 8</td>
<td>6</td>
<td>BLUE</td>
<td>50</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>LABEL</td>
<td>MONTHLY ENERGY: 15204 MWH LOAD FACTOR: 48.4</td>
<td>6</td>
<td>BLUE</td>
<td>50</td>
<td>6</td>
</tr>
</tbody>
</table>

EXHIBIT 5

%LET YR=1989; %LET M=2; %LET ST='SUNMONTUEWEDTHUFRISAT';

*******************************************************************************
** This macro generates 31 symbol statements. There is one symbol statement for each of **
** the days in the month. Each symbol has its own line type and color. **
*******************************************************************************

%MACRO SYMBOL;
%DO K=1 %TO 31; SYMBOL&K C=&&C&K V=NONE I=JOIN L=&K; %END;
%MEND;

DATA CPD; INFILE FBD(SECIPK); INPUT YEAR 4. MONTH 3. (CPD CPH) (4.);
IF YEAR=&YR & MONTH=&M;
******************************************************************************************* ** This macro processes the hourly data, creates the annotate file, and calls ** the plotting routine. **
*******************************************************************************************

%MACRO READ(IND,C);
DATA CPDAT; SET LOAD.&IND.HRLY; YEAR=YEAR(SDATE); MONTH=MONTH(SDATE);
DAY=DAY(SDATE); IF YEAR=&YR & MONTH=&M;
DATA CPDAT(KEEP=MONTH CWDX CPD CPH CP); LENGTH CWDX $4;
MERGE CPDAT CPD; BY YEAR MONTH; IF CPD=DAY; ARRAY X(24) KW1-KW24;
CP=X(CPH); CWDX=SUBSTR(&ST,(WEEKDAY(SDATE)*3)+3,2,3);
DATA PLTDAT; SET LOAD.&IND.HRLY; YEAR=YEAR(SDATE); MONTH=MONTH(SDATE);
DAY=DAY(SDATE); IF YEAR=&YR & MONTH=&M; IF &M=4 AND (KW2 LT KWI/2) THEN
KW2=(KW1 +KW3)/2;
DATA PLTDAT; MERGE PLTDAT CPD; BY MONTH; NCP=MAX(OF KW1-KW24); KWH=SUM(OF KW1-KW24);
PROC SORT; BY MONTH NCP;
PROC SUMMARY NWAY; CLASS MONTH; VAR KWH; OUTPUT OUT=KWH SUM=;
DATA PKDAT; SET PKDAT BY MONTH; IF LASTMONTH; LENGTH WDX $4;
ARRAY X(24) KW1-KW24; DO 1=1 TO 24; IF X(I)=NCP THEN HR=I; END;
WDX=SUBSTR(&ST,(WEEKDAY(SDATE)*3)+3,2,3);
DATA PKDAT; MERGE PKDAT KWH CPDAT;
LENGTH LFX $6 DYX $2 HRX $2 PKX $7 KWHX $8 CDYX $2 CHRX $2 CPKX $7;
LF=ROUND(KWH/(NCP** FREQUENCY*24)**100, 01); LF=LF; DYX=DAY; HRX=HR; PKX=NCP;
KWHX=RND(KWH/1000); CDYX=CPD; CHRX=CPH; CPKX=CP;
DROP _TYPE _FREQ _LF DAY HR NCP KWH KW1-KW24;
******************************************************************************************* ** This is the annotate data step. Here the summary data is converted to text strings **
** and the position, and color variables are created. SEE EXHIBIT 4. **
*******************************************************************************************

DATA ANNODATA(KEEP=COOP YEAR MONTH FUNCTION TEXT POSITION X Y COLOR);
SET PKDAT; LENGTH FUNCTION $8 TEXT $75 POSITION $1 COLOR $5;
FUNCTION='LABEL'; POSITION='6'; X=50; Y=8; COLOR='BLACK';
TEXT='CO-PEAK/HOUR: '|| CPX || ' ON' || CWDX || CDYX || ' AT HOUR' || CHRX;
OUTPUT;
FUNCTION='LABEL'; POSITION='06'; X=50; Y=7; COLOR='BLUE';
TEXT='PEAK/HOUR: '|| PKX || ' ON' || WDX || DYX || ' AT HOUR' || HRX;
OUTPUT;
FUNCTION='LABEL'; POSITION='6'; X=50; Y=6; COLOR='BLUE';
TEXT='MONTHLY ENERGY: '|| KWHX || ' MWH LOAD FACTOR' || LF;
OUTPUT;
1100
** This data step prepares the hourly loads for plotting. The colors for each day
** are assigned to global Macro variables by using the SYMPUT function. Next, the
** Do Loop transposes the hourly data from variables to observations. Finally, the
** SYMBOL macro is called to create the symbol statements. To save processing time,
** the SYMBOL macro is called only once, when the macro variable C1 = 1.

DATA A(KEEP=C0P YEAR MONTH DAY HOUR KW COLOR); LENGTH COLOR $5;
ARRAY H KW1-KW24; SET PLTDAT;
MONTH=MONTH(SDATE); YEAR=YEAR(SDATE);
DAY=DAY(SDATE); IF WEEKDAY(SDATE)=1 THEN COLOR='GREEN'; ELSE
IF WEEKDAY(COOP)=7 THEN COLOR='RED'; ELSE COLOR='BLUE';
IF CPD=DAY THEN COLOR='BLACK';
CALL SYMPUT('C'lLEFT(DAY),COLOR);
DO OVER H; KW=H; HOUR=1; OUTPUT; END;
%IF &C=1 THEN %SYMBOL;

** The GPLOT procedure below plots the hourly load data and uses the text strings in the
** annotate data set. Note that the plot statement is of the x*y=2 format. This allows
** a different symbol statement to be assigned to each line (day) plotted. Also note the
** ANNO= option and the reference to the ANNODAT data set containing the previously
** created text strings.

PROC GPLOT GOUT=FILPLTTMP; BY COOP YEAR MONTH;
PLOT KW*HOUR=DAY / FRAME ANNO=ANNODAT
HAXIS=1 TO 24 BY 1 AXIS=BLACK CTEXT=BLACK NOLEGEND;
TITLE1 H=1.0 C=BLACK 'HOURLY LOAD DATA'; FORMAT MONTH M. COOP C.;
%MEND READ;
%READ(CEN, 1); %READ(CLA, 2); %READ(GLA, 3); %READ(LEE, 4);
%READ(KE, 5); %READ(PEA, 6); %READ(SUM, 7); %READ(SUW, 8);
%READ(TAL, 9); %READ(TRI,10); %READ(WIT,11); %READ(SECI,12);