

Scaling Mount GCHART: Using a MACRO to Dynamically Reset the Scale

Nina L. Werner, Dean Health Plan, Inc., Madison, WI

ABSTRACT

If you do not set the scale yourself, PROC GCHART will automatically change the scale of the X-axis to match the unique data that happen to appear on each individual page. Conversely, if you set a scale, every page will have the same static value, regardless of the actual data values appearing in it. Neither was acceptable for this application. Using DATA step programming and a MACRO, I can dynamically reset the scale of a stacked bar chart's X-axis to a logical value for each new department while maintaining that same scale for multiple pages within the department, producing charts with the look you want.

INTRODUCTION

Three annotated SAS® programs (version 6.12) are presented. In addition to the method of setting the scale for each department and the MACRO which applies it to GCHART, they also contain other programming techniques of interest. They demonstrate the use of RSUBMIT to retrieve data from Sybase using SAS/CONNECT®, SAS/ACCESS® to Sybase and PROC DOWNLOAD, using macro variables for data selection passed from local (Windows) to remote (UNIX) SAS sessions. Data is summed as it is selected using nested SELECT statements in PASSTHRU SQL, so the summarization takes place inside Sybase. Data transformation is performed to allow stacked bars to be displayed side-by-side. A dummy sort variable is created to control page breaks for a consistent look. Formats are used on variables invoked in GCHART to order and label values.

Program Listing for Submit: RVU-CHART.SAS

```
* -- ;          * -- ;          * -- ;
%LET m = 06 ;   %LET l = 98 ;   %LET c = 99 ;
* -- ;          * -- ;          * -- ;
SIGNON ; * Connect to UNIX ;

LIBNAME rvu 'C:\UserData\SAS\Data' ;
%INCLUDE 'C:\UserData\SAS\Programs\syslput.sas' ;
%syslput (mo, &m) ;
%syslput (lyr, &l) ;
%syslput (cyr, &c) ;
%INCLUDE 'C:\UserData\SAS\Programs\rvu-cht.sas' ;
%INCLUDE 'C:\UserData\SAS\Programs\rvu-mac.sas' ;
* RVU-CHART.SAS -- (LANDSCAPE, SASMONO, 10 POINT) ;
```

DISCUSSION OF SUBMIT PROGRAM

RVU-CHART.SAS shows the code brought into the Editor Window for submission. The requester needs to overtype the three macro fields on line 2, m (month), l (last year), and c (current year.) SIGNON command invokes SAS CONNECT to remote UNIX, requires entry of UNIX userid and password. LIBNAME rvu sets location of retrieved

data in Windows (local) for SAS/GRAPH®. SYSLPUT MACRO sends local macro variables (m, l, c) to remote macro variables (mo, lyr, cyr.) Two %INCLUDE statements run the data-selection and graph-production programs. Using %INCLUDE shields the program source code from accidental editing by the requester. The source is separated into two sections so that after successful data extraction, the SAS/GRAPH program may be repeated if necessary without having to return to UNIX. Windows Printer Setup must be set to Landscape, SASMONO font, 10-point before submitting.

First Included Program Listing: RVU-CHART.SAS

```
RSUBMIT ; * RVU-CHART.SAS -- Nina L. Werner ;

OPTIONS NOCENTER DATE NUMBER ;
PROC SQL ;
CONNECT TO SYBASE (user=Nina
                  password=zzzzzzzz
                  database=rvu_data
                  server=sybsrver ) ;
;
CREATE TABLE curr AS
SELECT DISTINCT ins_type,
               doc_num,
               cytd,
               lytd
FROM CONNECTION TO SYBASE
  (SELECT ins_type,
         doc_num,
         (SELECT SUM(e.amount)
          FROM dbo.rvu e
          WHERE e.gl_month <= "&mo"
               AND e.gl_year = "&cyr"
               AND e.doc_num = r.doc_num
               AND e.injection = r.injection
               AND e.ins_type = r.ins_type)
        AS cytd,
        (SELECT SUM(f.amount)
         FROM dbo.rvu f
         WHERE f.gl_month <= "&mo"
               AND f.gl_year = "&lyr"
               AND f.doc_num = r.doc_num
               AND f.injection = r.injection
               AND f.ins_type = r.ins_type)
        AS lytd
  FROM dbo.rvu r
  WHERE doc_num < 20000
        AND injection IN ('N')
        AND ins_type IN ('H', 'N')
  ORDER BY ins_type,
         doc_num)
;
CREATE TABLE doc AS
SELECT name,
       INPUT(docno, 5.) AS doc_num,
       INPUT(dept, 5.) AS dept
FROM CONNECTION TO SYBASE
  (SELECT *
   FROM dbo.doc_ref
   WHERE docno < '20000'
   ORDER BY dept,
         docno)
;
```

```

CREATE TABLE dpt AS
SELECT dept_nam,
       INPUT (dept_num, 5.) AS dept
FROM CONNECTION TO SYBASE
(SELECT *
 FROM dbo.dept_ref
 WHERE dept_ancil = 'D'
 ORDER BY dept_num)
;
DISCONNECT FROM SYBASE ;
QUIT ;
RUN ;
;
PROC SQL ;
CREATE TABLE dmceref AS
SELECT dpt.dept,
       dept_nam,
       doc_num,
       name
FROM   doc
LEFT JOIN dpt
ON doc.dept = dpt.dept
ORDER BY dept,
       doc_num
;
QUIT ;
RUN ;
;
DATA current (LABEL="RVU H/N D &mo/&cyr") ;
SET curr ;
IF SUM (0, cytd, lytd) ^= 0 ;
difyr = (cytd - lytd) / lytd ;
LABEL difyr = 'Annual incr/decr'
       cytd = 'Current Year to Date'
       lytd = 'Last Year to Date'
       ins_type = 'HMO or Fee for Service' ;
RUN ;
;
PROC SQL ;
CREATE TABLE chart AS
SELECT n.dept_nam,
       n.dept,
       r.ins_type,
       n.name,
       r.doc_num,
       r.cytd,
       r.lytd,
       r.difyr
FROM   current r
LEFT JOIN dmceref n
ON r.doc_num = n.doc_num
ORDER BY dept_nam,
       ins_type,
       name
;
QUIT ;
;
DATA chart (LABEL="RVU H/N D &mo/&cyr") ;
SET chart ;
LABEL dept = 'dept_num'
       dept_nam = 'Department'
       name = 'Provider Name' ;
RUN ;
;
PROC SORT DATA=chart ;
BY dept_nam name ins_type ;
RUN ;
;
PROC DOWNLOAD IN=work OUT=rvu MEMTYPE=DATA ;
SELECT chart ;
QUIT ;
RUN ;
;
ENDRSUBMIT ;
SIGNOFF ;
;
* ##### ;
;

```

```

OPTIONS SOURCE2 ;
PROC SORT DATA=rvu.chart OUT=cht1 ;
BY doc_num ;
RUN ;
PROC MEANS N MAX NOPRINT DATA=cht1 ;
OUTPUT OUT=docmax N=n MAX=maxc maxl ;
VAR cytd lytd ;
BY doc_num ;
RUN ;
;
DATA zero (KEEP= doc_num) ;
SET docmax ;
WHERE maxc<=0 & maxl<=0 ;
RUN ;
;
DATA cht2 ;
MERGE cht1
       zero (IN=inz) ;
BY doc_num ;
IF ^inz ;
RUN ;
;
/* create formats for chart text */
PROC FORMAT LIB=library ;
VALUE $ins 'H' = ' HMO '
           'N' = 'Fee for Service' ;
VALUE $ytd 'C' = 'Current'
           'L' = 'Last ' ;
RUN ;
;
DATA ch3 ;
SET cht2 ;
LENGTH ytd_flag $1. ;
amount = cytd ;
ytd_flag = 'C' ;
OUTPUT ;
amount = lytd ;
ytd_flag = 'L' ;
OUTPUT ;
FORMAT ins_type $ins.
       ytd_flag $ytd. ;
LABEL amount = 'Production'
       dept = 'dept_num'
       ytd_flag = 'Current/Last YTD' ;
RUN ;
;
PROC SORT DATA=ch3 ;
BY dept_nam name ins_type ;
RUN ;
;
DATA rvu.chartrvu (LABEL="RVU H/N D &m/&y"
                  DROP=seq cytd lytd
                  ancil difyr) ;
SET ch3 ;
LENGTH page seq 3 ;
RETAIN page seq 0 ;
BY dept_nam name ;
IF cytd < 0 THEN
  cytd = 0 ;
IF lytd < 0 THEN
  lytd = 0 ;
IF FIRST.dept_nam THEN DO ;
  page = 1 ;
  seq = 0 ;
END ;
IF FIRST.name THEN
  seq +1 ;
/* 8 names per page */
IF seq = 9 THEN DO ;
  seq = 1 ;
  page +1 ;
END ;
LABEL page = 'Page number' ;
RUN ;
;
PROC SORT DATA=rvu.chartrvu ;
BY dept_nam page name ins_type ;
RUN ;
;

```

DISCUSSION OF DATA RETRIEVAL PROGRAM

RVU-CHT.SAS is the data extract, the first included program. It contains RSUBMIT-ENDRSUBMIT section which is executed in UNIX to retrieve Sybase data and download it back to Windows, plus a local section which runs in Windows. PASSTHRU SQL code passes the values of remote macro variables for conditional data selection in the first CREATE TABLE. Nested SELECT statements join Sybase table "rvu" with itself to summarize current year-to-date and last year-to-date totals by doctor number, injection flag and insurance type, for subsetting conditions. Two other CREATE TABLE SELECTs make reference tables for doctor name and department name. Department and doctor ID fields are converted from character to numeric in these reference tables to match detail ("rvu") data. DISCONNECT FROM SYBASE delimits the end of Sybase access, however, the process is still running on the remote (UNIX) platform. The two reference tables are combined using LEFT JOIN in PROC SQL. Since cytd and lytd are dollar fields created by summing detail records where they may be positive or negative, doctor records where the values are both zero or add to zero (cancel each other out) are omitted now. The percent difference between this year and last year is calculated. Names are applied to remaining detail records using LEFT JOIN in PROC SQL. The data are sorted by department and doctor names instead of Ids, and labels are applied. The data set "chart" is passed to Windows by PROC DOWNLOAD. Connection to UNIX is severed, removing WORK datasets created on the UNIX platform.

The rest of the processing occurs in Windows. PROC MEANS is used to capture the maximum values for current year-to-date and last year-to-date for each doctor between his two records (insurance type = HMO "H" or not "N".) Doctors with negative or zero maximums are omitted here. Formats are created. Note that \$ins is set to "HMO" or "Fee for Service" to maintain desired order of items; blank-H will appear before Fe... Data is transformed so that instead of two records per doctor by insurance type, each with current year-to-date and last year-to-date fields, four records per doctor by insurance type and ytd-flag are written. The dollar fields are collapsed to one variable "amount" to display side-by-side bars. As the permanent SAS dataset "rvu.chartrvu" is created, negative values are set to zero, so no bar will display a negative value. A new variable "page" is created to make even page breaks on the bar graphs. Page number starts over with a value of one for each department and is incremented after eight doctor names per page. Note the use of the local macro variables in the dataset's label. The permanent SAS dataset is sorted into its final order for graphical presentation, BY dept_nam page name ins_type.

**Second Included Program Listing:
RVU-MAC.SAS**

```
* RVU-MAC.SAS -- Nina L. Werner ;
* SAS monospace 10 pt Landscape ;
OPTIONS CENTER NODATE NONUMBER NOBYLINE
MISSING='0' LS=126 PS=42 ;
```

```
*OPTIONS MACROGEN MPRINT SYMBOLGEN ;
;

/* calculate departmental maximums
   for X-axis of graphs */
PROC MEANS N MAX NOPRINT DATA=rvu.chartrvu ;
OUTPUT OUT=deptmax N=n MAX=max1 ;
VAR amount ;
BY dept_nam ;
RUN ;
;
DATA maximums (KEEP= dept_nam max int) ;
SET deptmax ;
WHERE max1 > 0 ;
maxi = max1 *2 ;
IF maxi < 9999 THEN DO ;
max = ROUND(maxi, 100) ;
IF max < maxi THEN
max + 100 ;
int = ROUND( (max/5), 10) ;
END ;
ELSE IF maxi < 99999 THEN DO ;
max = ROUND(maxi, 1000) ;
IF max < maxi THEN
max + 1000 ;
int = ROUND( (max/5), 100) ;
END ;
ELSE DO ;
max = ROUND(maxi, 10000) ;
IF max < maxi THEN
max + 10000 ;
int = ROUND( (max/5), 1000) ;
END ;
LABEL int = 'X-axis Interval'
max = 'X-axis Maximum'
max1 = 'Dept Max Bar Len'
maxi = 'Dept Dbl Bar Len' ;
RUN ;
;
*****
/* macro to repeat chart for each department
   with separate scale */
%MACRO deptchrt ;
DATA _NULL_ ;
SET maximums END=last ;
CALL SYMPUT
('dpt' || TRIM(LEFT(PUT(_N_,3))),
TRIM(dept_nam)) ;
CALL SYMPUT
('max' || TRIM(LEFT(PUT(_N_,3))),
TRIM(LEFT(max))) ;
CALL SYMPUT
('int' || TRIM(LEFT(PUT(_N_,3))),
TRIM(LEFT(int))) ;
IF last THEN
CALL SYMPUT
('num', TRIM(LEFT(PUT(_N_,3)))) ;
RUN ;
;
%PUT >>>&num Departments for &m/&c on
%SYSFUNC(TODAY(),WORDDATE18)<<<< ;
;
%DO i=1 %TO &num ;
GOPTIONS RESET=GLOBAL ;
GOPTIONS DEVICE=HPGL4SI /* WIN */
GUNIT=PCT
FTEXT=SIMPLEX
CTEXT=BLACK
CBACK=WHITE
HTITLE=4
HTEXT=1.5
HPOS=120
VPOS=45 ;
PATTERN1 VALUE=SOLID COLOR=BLACK ;
PATTERN2 VALUE=X1 COLOR=BLACK ;
FOOTNOTE J=R COLOR=BLACK FONT=SWISSB
"%SYSFUNC(TODAY(), WORDDATE18)" ;
```

```

LEGEND  CBORDER=BLACK
        CSHADOW=BLACK
        OFFSET=(5,0)
        LABEL=NONE ;
AXIS1   COLOR=BLACK
        LABEL=(FONT=SWISSXB
              'Production Total by Provider')
        ORDER=(0 TO &&max&i BY &&int&i) ;
AXIS2   COLOR=BLACK
        LABEL=(FONT=SWISSXB 'PROVIDER NAME') ;
AXIS3   COLOR=BLACK
        LABEL=(FONT=SWISSXB 'YTD  ') ;
;
TITLE1  HEIGHT=3 FONT=SWISSB
        'Dean Medical Center, S.C.' ;
TITLE2  HEIGHT=4 FONT=SWISSB
        'Total Provider Production:
HMO and Fee For Service' ;
TITLE3  HEIGHT=3 FONT=SWISSB
        "#BYVAL(dept_nam) -
Current Month &m/&c - "
        "Page #BYVAL(page)" ;
;
PROC    GCHART DATA=rvu.chartrvu ;
        HBAR ytd_flag / GROUP=name
              SUBGROUP=ins_type
              SUMVAR=amount
              SUM
              DISCRETE
              TYPE=SUM
              RAXIS=AXIS1
              GAXIS=AXIS2
              MAXIS=AXIS3
              MINOR=4
              WIDTH=1
              SPACE=.1
              AUTOREF
              CLIPREF
              LEGEND=LEGEND ;
        WHERE dept_nam = "&&dpt&i" ;
        BY dept_nam page ;
        FORMAT amount COMMA11.0;
RUN ;
QUIT;
%END ;
;
RUN ;
%MEND deptchrt ;
;
*****
/* invocation of macro to produce charts
   for each department */
deptchrt ;

```

DISCUSSION OF GRAPH MACRO PROGRAM

RVU-MAC.SAS is the second included program, which first calculates the scale for each department if the departmental maximum is greater than zero into a SAS dataset called maximums. This includes the maximum size of the X-axis, which is double the maximum bar size, since bars will be stacked for insurance type within year, and a rounded value of 1/5 of this maximum for interval tick marks. Then, the MACRO program code appears.

DATA_NULL_ is used to retrieve all the departmental values into a series of macro variables to pass each department's name, X-axis maximum value and interval value using CALL SYMPUT. The MACRO reports to the SAS LOG on how many departments “&num” appear for the current selected month and year-to-date. Next, it will run PROC GCHART &num times with macro variables to adjust the current department's X-axis = 0 TO max BY int while leaving all other options consistent for all pages. GOPTIONS DEVICE=HPGL4SI sends charts directly to the printer. Alternatively, DEVICE=WIN would send charts to screen. The department name and page variable are captured into the title using #BYVAL and the month and current year-to-date are retrieved from the local macro variables. PROC GCHART sets two stacked (insurance type) bars side-by-side (ytd-flag) for each of the eight doctors per page, within the current department. The last step invokes the MACRO once to produce bar charts for all departments. The requester does not need to send each department to the printer individually.

Figure 1 contains four typical graphs for three departments, demonstrating a different scale for each department, maintained over multiple pages within a department, when necessary.

TRADEMARK CITATION

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

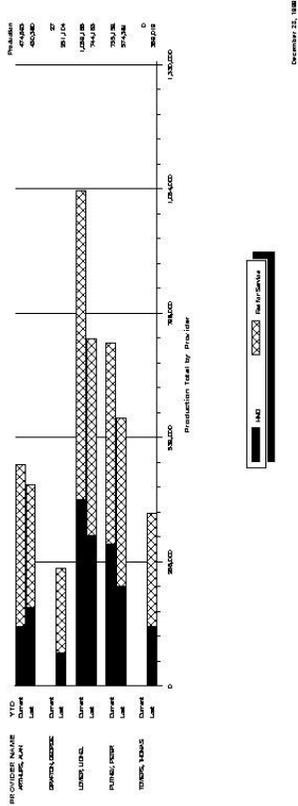
Nina L. Werner, BA, MBA, MHP
 Dean Health Plan, Inc.
 1277 Deming Way
 Madison, WI 53717

Work Phone: (608) 827-4224
 Fax: (608) 836-6335

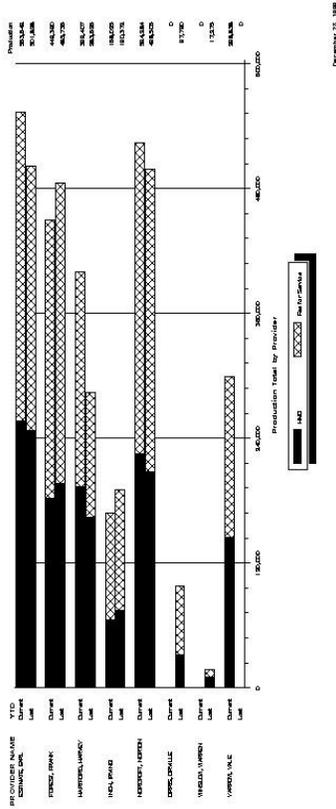
Email: Werner_Nina_L@ssmhc.com

Figure 1. Sample Graphs for Three Departments

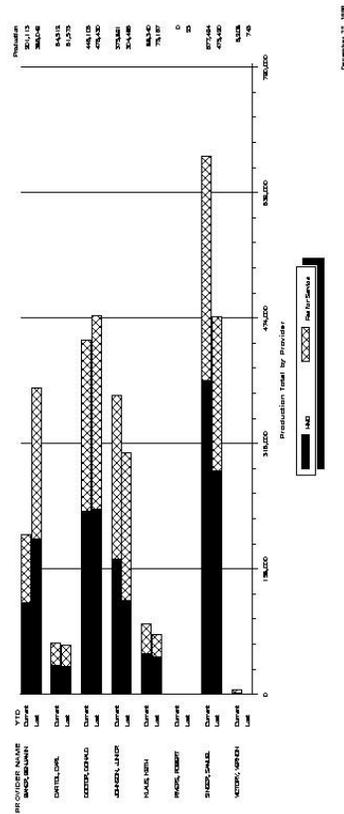
Dean Medical Center, S.C.
Total Provider Production: HMO and Fee For Service
CARDIOVASCULAR SURGERY – Current Month 06/99 – Page 1



Dean Medical Center, S.C.
Total Provider Production: HMO and Fee For Service
UROLOGY – Current Month 06/99 – Page 1



Dean Medical Center, S.C.
Total Provider Production: HMO and Fee For Service
ENT – Current Month 06/99 – Page 1



Dean Medical Center, S.C.
Total Provider Production: HMO and Fee For Service
ENT – Current Month 06/99 – Page 2

