

Data Manipulation and Report Writing Techniques

Julie Wallace, Blue Cross and Blue Shield of Texas*

ABSTRACT

SAS provides a variety of procedures for producing standard reports. Alternatively, SAS provides the flexibility to allow a programmer full control in order to construct tailored reports. This paper illustrates a program that starts with a SAS dataset, manipulates the data to form a file with the appropriate variables and summary levels, and ends with a NULL data step which writes a report. The program applies SAS techniques such as arrays, macros, macro variables, dataset indexing, nested do loops, and various PUT statement arguments. The FORMAT, SQL, TRANSPOSE, and REPORT procedures are also used.

INTRODUCTION

The report produced by the subject program is used to analyze hospital utilization and financial trends based on a set of medical claims. In addition to the typical summary amounts necessary for these analyses, the report displays calculated averages and comparisons from one twelve-month period to the next. All of the information for each hospital is organized onto a single page to provide a convenient synopsis. The report also includes summary pages for all hospitals statewide and for each of four geographic regions. Footnotes identify pages showing a decrease or doubling of payments over the last twelve months.

PRELIMINARY STEPS

Several preliminary steps need to be completed before creating any report. First, interview the customer(s) to define the exact report requirements. Next, determine if the report can be created using SAS report procedures rather than a customized program. At first glance this may not seem possible, but SAS offers various methods that can be used to manipulate the data and force it to conform to report procedures. If customer requirements cannot be met using the procedures provided by the SAS System, then proceed with the design of a tailored report. Define the page dimensions, draft the report layout, and establish the exact coordinates of all rows and columns. Define the formats for all report elements, using custom formats where necessary. Then, continue with the steps that follow.

STEP 1. ASSEMBLE THE DATA

The first step is to prepare a SAS dataset including the necessary variables. Limit the number of observations and variables as much as possible to minimize resource usage in future steps.

Macros

PRODDATE is a macro variable that is set to the most recent calendar production month. It is hard-coded in the sample program for simplicity, but the actual program calls an external macro to execute the %LET statement. The external macro is manually updated each month, and is executed by various programs. This eliminates the need to change the proddate in each program that uses it.

The MGETYEAR macro creates a variable called YEAR, with a value of 1, 2, or 3. This value indicates which twelve-month period the data represents. Year 1 is the earliest period, and year 3 is the most current twelve-month period. Two arguments are passed to the macro. The first, PAIDMTH, is the month that will be used to determine the year for the given observation. It is one of 36 months that will go into year 1, 2, or 3. The second argument, PRODMTH, is assumed to be the last of 36 months, or the most recent calendar production month. Each of the 36 months is placed into the array DATERNG, with PRODMTH being the 36th element. MGETYEAR is included in the sample program for simplicity, but in the actual program it is an external macro that is brought in with a %INCLUDE statement.

The Claims Data Step

The data source is a SAS dataset containing 3 years of hospital claims. The first data step reads in the existing SAS dataset and creates a new SAS dataset called CLAIMS, with the following variables:

HOSPNO	- Hospital number
PRODUCT	- Product code, A or B
REGION	- Geographic region of the hospital
YEAR	- 1, 2, or 3, as described above
SETTING	- I for Inpatient, O for Outpatient
BILLED	- Billed charges
ALLOWED	- Allowed charges
PAID	- Paid amount
ADSVIS	- Inpatient admissions, outpatient visits
SHOSPNO	- Default hosp number for total pages

SREGION - Default region for statewide report

See CLAIMS in Exhibit 3.

STEP 2. SUMMARIZE THE DATA

The data should next be summarized into the report categories. There are many ways to accomplish this summary step using SAS. This example uses PROC REPORT.

The MREPORT Macro Variable

Several PROC REPORT steps will summarize the claims at various levels. Most of the PROC REPORT statements will be the same for all of these steps. The statements that will remain constant are stored in a macro variable called MREPORT. This variable will be resolved in each REPORT step. Using a macro variable in this manner shortens the program and simplifies future modifications.

Summarizing the data using PROC REPORT

The data is summarized at three different levels using PROC REPORT: statewide, by region, and by hospital. The summarized data is written to SAS datasets, rather than to actual reports. This is accomplished as follows:

The PROC REPORT statement:

NOWD - prevents invoking interactive window
MISSING - prevents exclusion of missing values
DATA= - input SAS dataset
OUT= - output SAS dataset

The **COLUMN** statement indicates which variables are to be included in the output dataset.

The **DEFINE** statements indicate how each variable will be used. **GROUP** variables determine the summary levels of the dataset, and **SUM** variables are totaled for each combination of **GROUP** variables. **COMPUTE** variables are computed within the PROC. **AVGDISC** the average ratio of billed minus allowed to billed, and **AVGPAID** is the average paid amount per admission or visit. **NOPRINT** indicates that the column will not be printed. **ORDER=INTERNAL** instructs SAS to sort the data by the actual value of the column, rather than the formatted value. Finally, the **&MREPORT** reference brings in the macro variable described above and causes its lines to be executed.

Note that the STATERPT dataset will show a value of 0 for the region and ALLHSP for the hospital number. Similarly, the REGRPT dataset will show the actual value of each region, along with ALLHSP for the hospital number. The HOSPRPT dataset will include

actual values of region and hospital number. Refer to Exhibit 3 for illustrations.

STEP 3. ARRANGE THE DATA

The next step is to put the three summary datasets created by PROC REPORT into a sequential dataset in the order that each report page will be written. The report will start with the statewide total page, then the first region's total page, each hospital's page within the first region, then the second region's total page, and so on.

ALLDATA

The data is first combined with a set statement in the ALLDATA data step. This step also assigns variable labels, renames shospno and sregion, and sets the BY groups. Note that PROC REPORT has already sorted the data according to the GROUP variables, so the BY statement is valid.

FINDATA

PROC SQL creates a dataset called FINDATA. FINDATA is almost identical to ALLDATA, except that it has a variable called PCTTOT, which is the percentage of total paid that each product comprises for each year.

PROC TRANSPOSE

FINDATA is treated as a table of rows and columns. The observations are the rows, and the variables are the columns. Currently, all of the amounts are stored in separate columns, and the years are stored in separate rows. The data needs to be turned sideways so that the years can be columns, and the amounts can be rows, like in the report layout. The TRANSPOSE procedure accomplishes this twisting of the data. The NAME parameter gives the transposed variables the name AMOUNT. The BY statement indicates which variables will not be transposed. The ID statement creates three new columns, using the formatted values of YEAR as the variable names: Year_1, Year_2, and Year_3. Finally, the VAR statement indicates which columns are to become rows. Their original names are placed in the AMOUNT column. The output dataset TRANSDAT has a variable called AMOUNT, with values such as BILLED, ALLOWED, etc., and three variables, Year_1, Year_2, and Year_3, that contain the amounts formerly in billed, allowed, etc. The data is then re-sorted, incorporating the AMOUNT variable into the sort.

See ALLDATA, FINDATA, TRANSDAT, and TRANSSOR in Exhibit 3.

STEP 4. WRITE THE REPORT

This step requires that the page dimensions and field coordinates have been determined, as described in the preliminary steps. The report is written to a flat text file using a NULL data step. The method for printing the report depends upon the operating system.

Hospital Names

Hospital names are stored in the dataset HOSPDIR, which has an index on HOSPNO. The KEY= option in the SET statement acts as a table lookup to get hospital names from HOSPDIR based on each observation's value of HOSPNO in TRANSSOR.

File Statement

The FILE statement defines the parameters of the output file:

- The filename is HPCAREPT.TXT.
- N=PS instructs SAS to always put PS lines per page. PS is defined in the options statement at the top of the program (PAGESIZE=).
- HEADER=HEADFOOT causes the HEADFOOT section to execute at each page break.
- NOTITLES prevents titles from being put on each page. Headers are used instead, and are defined in the HEADFOOT section.
- LRECL defines the output record length, or page width. This is set to the maximum page width.

Arrays

Arrays are used to enable DO LOOP processing. This makes the program shorter, thereby making maintenance quicker. Temporary arrays decrease processing time. Arrays are used to store amounts, percentages, dates, settings, report coordinates and increments, and formatting characters. The program comments explain the definitions of specific arrays.

Calculations

All totals are initialized to zero, and accumulated within the appropriate levels using DO LOOPS. For each product, A and B, there is a percentage to indicate how much that product comprises of the grand total for each year.

There are also percentages to indicate the amount of change from one year to the next. Percentages do not apply to the first year because there is no prior year for comparison. If the PAID amount shows a decrease or a change of 100% or more over the last twelve months, a value of 1 or 2 is assigned to the variable NOTE. A corresponding note is placed at the

bottom of the page upon execution of the HEADFOOT section.

Report Body

The body of the report is written by PUT statements within several nested DO LOOPS. These loops process the arrays for amounts, settings, and years. The position of each value is determined by the arrays set up for row and column coordinates. The PUT statements use the following operators:

- # - Place the value on a specified line of the page.
- @ - Place the value in a specified column.
- / - Execute a carriage return and start on next line.
- + - Move the pointer a specified number of spaces.

Page Breaks

Each time a new hospital is encountered, indicated by FIRST.HOSPNO, a page break is executed by the command PUT _PAGE_. Each time a page break occurs, the HEADFOOT section is executed.

In the HEADFOOT section, the page number is set back to zero at the beginning of each region. The page number is then advanced for each page break. This section also adds text to the top and bottom of each page using PUT statements with the operators described above. Conditional notes designate pages on which payments doubled or decreased over the last twelve months.

CONCLUSION

There are a variety of ways to accomplish any given task using SAS. This paper focuses on one program that illustrates some methods of manipulating data and creating reports.

REFERENCE

SAS Institute Inc., *SAS Procedures Guide, Version 6, Third Edition*, Cary, NC: SAS Institute Inc., 1990. 705 pp.

SAS Institute Inc., *SAS Language: Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc., 1990. 1042 pp.

CONTACT

Julie Wallace
Blue Cross and Blue Shield of Texas
P.O. Box 655924
Dallas, TX 75265-5924
julie_wallace@bcbstx.com

EXHIBIT 1. THE PROGRAM

```

/*****
/* AUTHOR: JULIE WALLACE */
/* PROGRAM: HPCA */
/* INPUT: 3 Years of Hospital Claims, */
/* Hospital Name Directory. */
/* OUTPUT: Hospital Utilization Trend Report. */
*****/

```

```
options pagesize=60 linesize=165 nodate missing='-';
```

```

*****
*** MACROS ***
*****
%MACRO M_GETYR(paidmth,prodmth);
    %local daterng i;
    %global year;
    array daterng(36) _temporary_;
    daterng(36)=&prodmth;
    do i=35 to 1 by -1;
        if 1=mod(daterng{i+1},100) then daterng{i}=daterng{i+1}-89;
        else daterng{i}=daterng{i+1}-1;
    end;
    select;
        when (&paidmth ge daterng{1} & &paidmth le daterng{12}) year=1;
        when (&paidmth ge daterng{13} & &paidmth le daterng{24}) year=2;
        when (&paidmth ge daterng{25} & &paidmth le daterng{36}) year=3;
        otherwise year=0;
    end;
%MEND M_GETYR;

```

```
%let proddate=199904;
```

```

*****
*** USER-DEFINED FORMATS ***
*****

```

```

proc format;
    value yearfmt 1='Year 1'
                  2='Year 2'
                  3='Year 3';

    value $setfmt I='Inpatient'
                  O='Outpatient';

    value $regfmt 0='Statewide'
                  1='North'
                  2='South'
                  3='West'
                  4='East';

    value $amtfmt AD$VIS = ' Adms/Visits '
                  BILLED=' Billed '
                  ALLOWED=' Allowed '
                  PAID = ' Paid '

```

```

AVGDISC='Avg Disc'
AVGPAID='Paid per Ad/Vis';

```

```

*****
*** THE CLAIMS DATA STEP ***
*****
DATA claims;
    set saspaper.claims;
    length year 3;
    format billed allowed paid dollar15.
           ad$vis comm15.
           region $regfmt.
           year yearfmt.;
    if setting='I' then visits = 0;
    if setting='O' then adms = 0;
    ad$vis= adms + visits;
    shospno = 'ALLHSP';
    sregion = '0';
    %m_getyr(pdmth,&proddate);
    keep hospno billed allowed paid
        ad$vis product region year
        setting shospno sregion;

```

```

*****
*** THE MREPORT MACRO VARIABLE ***
*****
%let mreport=%str(
    DEFINE year / GROUP noprint;
    DEFINE product / GROUP noprint;
    DEFINE setting / GROUP noprint;
    DEFINE billed / SUM noprint;
    DEFINE allowed / SUM noprint;
    DEFINE paid / SUM noprint;
    DEFINE ad$vis / SUM noprint;
    DEFINE avgdisc / computed noprint;
    DEFINE avgpaid / computed noprint;

```

```

    COMPUTE avgdisc;
        if billed.sum=0 then avgdisc = 0;
        else avgdisc=1-allowed.sum/billed.sum;
    ENDCOMP;

```

```

    COMPUTE avgpaid;
        if ad$vis.sum=0 then avgpaid = 0;
        else avgpaid=paid.sum/ad$vis.sum;
    ENDCOMP;
);

```

```

*****
*** SUMMARIZING THE DATA USING PROC REPORT ***
*****
* Statewide summary data;
proc report nowd missing data=claims out=staterpt;
    COLUMN sregion shospno product setting year
           billed allowed paid ad$vis avgdisc avgpaid;

```

```

DEFINE sregion / GROUP noprint;
DEFINE shospno / GROUP noprint;

&mreport

* Summary data by region;
proc report nowd missing data=claims out=regprt;
  COLUMN region shospno product setting year
         billed allowed paid advsvis avgdisc avgpaid;
  DEFINE region / GROUP noprint order=internal;
  DEFINE shospno / GROUP noprint;

  &mreport

* Summary data by region & hospital.;
proc report nowd missing data=claims out=hospprt;
  COLUMN region hospno product setting year
         billed allowed paid advsvis avgdisc avgpaid;
  DEFINE region / GROUP noprint order=internal;
  DEFINE hospno / GROUP noprint;

  &mreport

*****;
*** ALLDATA ***;
*****;
data alldata;
  label advsvis ='Adms/Visits'
        billed  ='Billed'
        allowed ='Allowed'
        paid    ='Paid'
        avgdisc ='Average Discount'
        avgpaid ='Paid per Adm/Vis';
  set staterpt (rename=(shospno=hospno sregion=region))
    regprt (rename=(shospno=hospno))
    hospprt;
  by region hospno product setting year;

*****;
*** FINDATA ***;
*****;
proc sql;
  create table findata as
  select region, hospno, product, setting, year, billed, allowed, paid,
         advsvis, avgpaid, avgdisc, paid/sum(paid) as pcttot
  from alldata
  group by region, hospno, year, product
  order by region, hospno, product, setting, year;

*****;
*** PROC TRANSPOSE ***;
*****;
proc transpose data=findata out=transdat name=amount;
  by region hospno product setting;

```

```

id year;
var billed allowed paid advsvis avgdisc avgpaid pcttot;

proc sort data=transdat out=transsor;
  by region hospno amount product setting;

```

```

*-----*
*
* The data step below writes the report into a flat text file.
*
* Arrays are used to enable do loop processing. This makes the
* program shorter, thereby making maintenance quicker. Temporary
* arrays speed processing time.
* There are arrays for the following:
* - amounts and percentages - 1 for each year, totals for each
*   product within year, and grand totals for each year.
* - last month of each 12-month period.
* - possible values of amount and setting.
* - coordinates of starting columns and rows for various sections.
* - number of spaces to increment column when writing Product B data;
* - dotted line to separate data from column headings.
*
* The file statement:
* - hpcarept.txt is the output text file.
* - n=ps says to always put ps lines per page. PS is specified in
*   the options statement at the top of the program (pagesize=).
* - header=headfoot tells SAS to execute the headfoot section at
*   each pagebreak.
* - notitles tells SAS not to put titles on pages. Headers are
*   used instead, and are defined in the headfoot section.
* - lrecl defines the output record length. Set to the max page
*   width.
*
*-----*

```

```

data _null_;
  set transsor;
  by region hospno amount product setting;

  * get hospital name using index;
  set saspaper.hospdir(keep=hospno hospname) key=hospno/unique;
  if _iorc_ then do;
    * hospno not found ;
    if hospno='ALLHSP' then hospname='Total All Hospitals';
    else hospname='Hospital Name Unavailable';
    _error_=0;
  end;

```

```
file hpcarept n=ps header=headfoot notitles lrecl=165;
```

```

retain date note;
amount=upcase(amount);

```

```

array yymm{3} _temporary_;      * Last month of 12-month period;
array pctchn{3} _temporary_;    * % change - 1 for each year;

```

```

array tpctchg{3} _temporary_ ; * product tot % change;
array ytpctchg{3} _temporary_ ; * yearly total % change;
array totamt{3} _temporary_ ; * product total amount - 1 for each year;
array ytotamt{3} _temporary_ ; * yearly total amount;
array pctytotl{3} _temporary_ ; * % of yearly total for last lob;
array pctytotf{3} _temporary_ ; * % of yearly total for first lob;
array yearcol{3} Year_1-Year_3; * year 1-3;
array amtcol{3} _temporary_ (12 59 114); * starting col for amount;
array amtrow{7} _temporary_ (11 20 29 38 44 51 51); * starting row;
array pctcol{3} _temporary_ (22 75 130); * starting col for percent;
array colincr{3} _temporary_ (23 27 27); * col increment for B data;
array amtlist{7} $ _temporary_ ('BILLED','ALLOWED','PAID',
                                'ADSVIS','AVGPAID','AVGDISC','PCTTOT');
array setlist{2} $ _temporary_ ('I','O');
array dotlin{3} $ 24 _temporary_ ('-----',
                                   '-----',
                                   '-----');

```

```

*****;
* Get current date, and get ending month for each 12-month period.*;
* Values are retained, so only calculate for the first record. *;
*****;

```

```

if _n_ = 1 then do;
    date = today();
    yymm{3} = input(put(1 + 100 * &proddate,8.),yyymmdd8.);
    yymm{2} = yymm{3} - 365;
    yymm{1} = yymm{2} - 365;
end;

```

```

if first.hospno then do;
    note = 0;          ** initialize to no note **;
    put _page_;        ** page break on hospital **;
end;

```

```

*****;
* Set totals to zero for the first record for each amount. *;
* Then accumulate totals for all settings. *;
* Remember that the dataset is transposed, so amounts are *;
* stored in Year_1-3, and identified by the variable AMOUNT. *;
*****;

```

```

if first.amount then do i = 1 to dim(yearcol);
    ytotamt{i} = 0;
end;
if first.product then do i = 1 to dim(yearcol);
    totamt{i} = 0;
end;
do i = 1 to dim(yearcol);
    totamt{i} + yearcol{i};
    ytotamt{i} + yearcol{i};
    if last.amount then do;
        if ytotamt{i} = 0 then do; * no data for the year ;
            pctytotl{i} = 0;      * set % total year to zero ;
            pctytotf{i} = 0;

```

```

end;
else do; * there is data, so calc %s ;
    pctytotl{i} = totamt{i} / ytotamt{i};
    pctytotf{i} = 1 - pctytotl{i};
end;
end; * % year total calc ;
end;

```

```

*****;
* Compute percent change for years 2 and 3. This is the % change *;
* from prior year, so it is not available for the first year. *;
* Set up a note in case the %change doubles or decreases. *;
*****;

```

```

pctchg{1} = .;
tpctchg{1} = .;
ytpctchg{1} = .;

```

```

do i = 2 to dim(yearcol);
    if yearcol{i-1}=0 then pctchg{i} = .;
    else pctchg{i} = (yearcol{i} - yearcol{i-1}) / yearcol{i-1};
    if totamt{i-1}=0 then tpctchg{i} = .;
    else tpctchg{i} = (totamt{i} - totamt{i-1}) / totamt{i-1};
    if ytotamt{i-1}=0 then ytpctchg{i} = .;
    else ytpctchg{i} = (ytotamt{i} - ytotamt{i-1}) / ytotamt{i-1};
    if i=dim(yearcol) & last.amount & amount= 'PAID' then
        do; ----> set note for bottom of page ;
            if ytpctchg{i} lt 0 & ytpctchg{i} ne . then note=1;
            else if ytpctchg{i} ge 1 then note=2;
        end; ----> note ;
end;

```

```

*****;
* The main do loop is below. Loop through amounts, settings, *;
* and years. Write data conditionally with if statements. *;
* Write in the appropriate positions using the row and column *;
* arrays defined above. The following conditions are of *;
* particular importance: *;
* - do not write % change for the first year. *;
* - write dollars with dollar format, adms/visits with comma *;
* format, and discounts with percentage formats. *;
* - write % change values with decimals, and % of total allowed *;
* without decimals. *;
*****;

```

```

do i=1 to dim(amtlist);
    if amount = amtlist{i} then do;
        do j=1 to dim(setlist);
            if setting = setlist{j} then do;
                put #(j - 1 + amtrow{i}) setting $setfmt. @;
                do k=1 to dim(yearcol);
                    if amount = 'PCTTOT' then put
                        #(&amtrow{i}-3) #(&pctcol{k} + (product='B') * colincr{k})
'& Total' /
                        #(&pctcol{k} + (product='B') * colincr{k})' Paid ';

```

```

else put
  @(amtrw{i}-2) @(amtc{col{k} + (product='B') * colincr{k}}
  amount $amtfmt.
  @(amtrw{i}-1) @(amtc{col{k} + (product='B') * colincr{k}}
  dotlin{k};
if amount =: 'ADSVIS' then put #(j - 1 + amtrw{i})
  @(amtc{col{k} + (product='B') * colincr{k}} yearcol{k} comma13. @;
else if amount =: 'AVGDISC' then put #(j - 1 + amtrw{i})
  @(amtc{col{k} + (product='B') * colincr{k}} yearcol{k} percent9.2@;
else if amount =: 'PCTTOT' then put #(j - 1 + amtrw{i})
  @(pctcol{k} + (product='B') * colincr{k}} yearcol{k} percent8.2@;
else put #(j - 1 + amtrw{i})
  @(amtc{col{k} + (product='B') * colincr{k}} yearcol{k} dollar15. @;
if k ne 1 & amount not in: ('AVGDISC', 'PCTTOT') then put
  @(pctcol{k} + (product='B') * colincr{k}} pctchg{k} percent8.2
  @(amtrw{i} - 2)
  @(pctcol{k} + (product='B') * colincr{k}} '% Change';
if last.product & amount in: ('BILLED', 'ALLOWED', 'PAID') then
  do; *---> always put TOTAL in 4th row (3 + amtrw{i});
  put #(3 + amtrw{i}) 'Total'
  @(amtc{col{k} + (product='B') * colincr{k}} totamt{k} dollar15. @;
  if k ne 1 then put
    @(pctcol{k} + (product='B') * colincr{k}} ttpctchg{k} percent8.2;
  end; *---> product total ;
if last.amount & amount in: ('BILLED', 'ALLOWED', 'PAID') then
  do;
  put #(3 + amtrw{i} + 1) '% Year Tot'
  @(amtc{col{k} + (product='B') * colincr{k}}
  pctytot1{k} percent15.2
  @(amtc{col{k} + (product = 'A') * colincr{k}}
  pctytotf{k} percent15.2 /
  'Prods A+B' @(amtc{col{k}} ytotamt{k} dollar15. @;
  if k ne 1 then put @(pctcol{k}} ytpctchg{k} percent8.2;
  end; * ---> year total ;
end; * -----> k ;
end; * -----> setting ;
end; * -----> j ;
end; * -----> amount ;
end; * -----> i ;

if last.hospno then do;
  if note=1 then put #55
    '\\\\\\ Payments DECREASED over the last 12 months \\\\' ;
  else if note=2 then put #55
    '***** Payments DOUBLED over the last 12 months *****';
end; * -----> last hospno ;

return; * Go to top of data step. Only execute headfoot section;
* at page breaks;

headfoot:
  if first.region then pno = 0;
  pno + 1;

```

```

put #57 'Note: This report is for illustrative purposes only, and
does not reflect'
#58 'actual hospitals, utilization, trends, discounts, or
other information.'
#59 ' '
#60 'SCSUG 99 - Julie Wallace'
#154 date mmdyy10.
#1 'Hospital Utilization Trends'
#155 'Page ' pno //
      'Region: ' region;
if hospno='ALLHSP' then put #4 'Hospital: ' hospname;
else put      #4 'Hospital: ' hospno hospname;
put #6
  @(pctcol{1}) '12 Months Ending ' yymm{1} monyy7.
  @(pctcol{2}-3) '12 Months Ending ' yymm{2} monyy7.
  @(pctcol{3}-3) '12 Months Ending ' yymm{3} monyy7. /
  @(amtc{col{1}+4} 'Product A      Product B '
  @(amtc{col{2}+7} ' Product A      Product B '
  @(amtc{col{3}+7} ' Product A      Product B ' ;

```

```
return;
```

```
run;
```

EXHIBIT 2. SAMPLE REPORT PAGE

Hospital Utilization Trends										Page 1
Region: Statewide										
Hospital: Total All Hospitals										
	12 Months Ending APR1997		12 Months Ending APR1998		12 Months Ending APR1999		12 Months Ending APR1999		12 Months Ending APR1999	
	Product A	Product B	Product A	Product B	Product A	Product B	Product A	Product B	Product A	Product B
	Billed	Billed	Billed	% Change	Billed	% Change	Billed	% Change	Billed	% Change
Inpatient	\$17,273,887	\$18,848,467	\$17,634,836	1.97%	\$9,371,125	(13.55%)	\$25,518,878	44.66%	\$6,779,837	(27.66%)
Outpatient	\$18,258,851	\$5,848,638	\$11,736,884	14.41%	\$6,624,275	13.26%	\$14,745,135	25.63%	\$4,867,831	(38.59%)
Total	\$27,552,738	\$16,689,899	\$29,371,648	6.68%	\$15,995,400	(4.16%)	\$40,265,213	37.85%	\$18,846,868	(32.19%)
% Year Tot	62.28%	37.72%	64.74%		35.26%		78.77%		21.23%	
Prods A+B	\$44,241,838		\$45,367,848	2.54%			\$51,182,882	12.64%		
	Allowed	Allowed	Allowed	% Change	Allowed	% Change	Allowed	% Change	Allowed	% Change
Inpatient	\$12,785,388	\$8,821,438	\$13,145,154	2.81%	\$7,872,188	(11.83%)	\$19,332,633	47.87%	\$5,175,877	(26.82%)
Outpatient	\$7,625,886	\$4,369,387	\$8,685,932	13.91%	\$5,888,354	14.44%	\$11,862,357	27.36%	\$3,865,814	(38.78%)
Total	\$20,418,394	\$12,398,817	\$21,831,887	6.96%	\$12,872,542	(2.57%)	\$30,394,998	39.23%	\$8,248,891	(31.75%)
% Year Tot	62.22%	37.78%	64.39%		35.61%		78.67%		21.33%	
Prods A+B	\$32,881,218		\$33,983,628	3.36%			\$38,635,888	13.96%		
	Paid	Paid	Paid	% Change	Paid	% Change	Paid	% Change	Paid	% Change
Inpatient	\$18,851,868	\$6,813,715	\$11,158,383	2.83%	\$5,979,474	(12.24%)	\$16,589,987	48.68%	\$4,426,747	(25.97%)
Outpatient	\$6,471,418	\$3,733,587	\$7,388,249	14.84%	\$4,238,857	13.54%	\$9,365,918	26.91%	\$2,597,846	(38.71%)
Total	\$17,322,478	\$10,547,222	\$18,538,553	7.82%	\$10,218,331	(3.12%)	\$25,955,817	48.81%	\$7,824,592	(31.25%)
% Year Tot	62.16%	37.84%	64.47%		35.53%		78.78%		21.38%	
Prods A+B	\$27,869,788		\$28,756,884	3.18%			\$32,988,489	14.69%		
	Adms/Visits	Adms/Visits	Adms/Visits	% Change	Adms/Visits	% Change	Adms/Visits	% Change	Adms/Visits	% Change
Inpatient	2,567	1,864	2,688	1.29%	1,868	(8.38%)	3,152	21.23%	732	(38.94%)
Outpatient	4,862	2,773	5,332	9.67%	2,737	(1.38%)	6,241	17.85%	1,725	(36.97%)
	Paid per Ad/Vis	Paid per Ad/Vis	Paid per Ad/Vis	% Change	Paid per Ad/Vis	% Change	Paid per Ad/Vis	% Change	Paid per Ad/Vis	% Change
Inpatient	\$4,227	\$6,484	\$4,292	1.53%	\$5,641	(11.91%)	\$5,263	22.64%	\$6,847	7.21%
Outpatient	\$1,331	\$1,346	\$1,384	3.99%	\$1,549	15.83%	\$1,581	8.42%	\$1,586	(2.76%)
	Avg Disc	% Total Paid	Avg Disc	% Total Paid	Avg Disc	% Total Paid	Avg Disc	% Total Paid	Avg Disc	% Total Paid
Inpatient	26.87%	62.64%	26.88%	64.68%	25.46%	68.19%	24.53%	58.52%	24.22%	63.92%
Outpatient	25.67%	37.36%	25.29%	35.48%	25.99%	39.81%	24.51%	41.48%	24.98%	36.88%
Note: This report is for illustrative purposes only, and does not reflect actual hospitals, utilization, trends, discounts, or other information.										
SCSUG 99 - Julie Wallace										

EXHIBIT 3. PARTIAL DATASETS

CLAIMS										
BILLED	ALLOWED	PAID	SETTING	PRODUCT	HOSPNO	REGION	YEAR	ADSVIS	SHOSPNO	SREGION
\$3,642	\$2,903	\$2,863	O	A	R00004	3	1	1	ALLHSP	0
\$685	\$503	\$471	I	B	R00004	3	1	1	ALLHSP	0
\$645	\$568	\$413	O	B	R00004	3	1	1	ALLHSP	0
\$123	\$118	\$87	O	A	R00004	3	1	1	ALLHSP	0
\$4,952	\$4,313	\$3,342	I	A	R00004	3	1	1	ALLHSP	0
...										

STATERTPT										
SREGION	SHOSPNO	PRODUCT	SETTING	YEAR	BILLED	ALLOWED	PAID	ADSVIS	AVGDISC	AVGPAID
0	ALLHSP	A	I	1	\$17,293,887	\$12,785,388	\$10,851,060	2,567	0.2607	4227.14
0	ALLHSP	A	I	2	\$17,634,836	\$13,145,154	\$11,158,303	2,600	0.25459	4291.66
0	ALLHSP	A	I	3	\$25,510,078	\$19,332,633	\$16,589,907	3,152	0.24216	5263.3
0	ALLHSP	A	O	1	\$10,258,851	\$7,625,006	\$6,471,418	4,862	0.25674	1331.02
0	ALLHSP	A	O	2	\$11,736,804	\$8,685,932	\$7,380,249	5,332	0.25994	1384.14
...										

REGRPT										
REGION	SHOSPNO	PRODUCT	SETTING	YEAR	BILLED	ALLOWED	PAID	ADSVIS	AVGDISC	AVGPAID
1	ALLHSP	A	I	1	\$4,511,234	\$3,316,228	\$2,808,456	679	0.2649	4136.17
1	ALLHSP	A	I	2	\$4,129,456	\$3,077,298	\$2,604,705	648	0.25479	4019.61
1	ALLHSP	A	I	3	\$5,909,388	\$4,449,573	\$3,800,211	817	0.24703	4651.42
1	ALLHSP	A	O	1	\$2,675,170	\$1,973,510	\$1,672,973	1,291	0.26229	1295.87
1	ALLHSP	A	O	2	\$2,809,158	\$2,099,644	\$1,785,546	1,407	0.25257	1269.04
...										

HOSPRPT										
REGION	HOSPNO	PRODUCT	SETTING	YEAR	BILLED	ALLOWED	PAID	ADSVIS	AVGDISC	AVGPAID
1	R00006	A	I	1	\$17,539	\$12,458	\$9,587	4	0.28969	2396.83
1	R00006	A	I	2	\$6,571	\$4,302	\$3,987	4	0.34538	996.68
1	R00006	A	I	3	\$28,000	\$19,286	\$17,046	7	0.31122	2435.13
1	R00006	A	O	1	\$1,197	\$795	\$692	2	0.33537	346.16
1	R00006	A	O	2	\$1,197	\$678	\$598	2	0.43336	299.04
...										

ALLDATA										
ADSVIS	BILLED	ALLOWED	PAID	AVGDISC	AVGPAID	REGION	HOSPNO	PRODUCT	SETTING	YEAR
2,567	\$17,293,887	\$12,785,388	\$10,851,060	0.2607	4227.14	0	ALLHSP	A	I	1
2,600	\$17,634,836	\$13,145,154	\$11,158,303	0.25459	4291.66	0	ALLHSP	A	I	2
3,152	\$25,510,078	\$19,332,633	\$16,589,907	0.24216	5263.3	0	ALLHSP	A	I	3
4,862	\$10,258,851	\$7,625,006	\$6,471,418	0.25674	1331.02	0	ALLHSP	A	O	1
5,332	\$11,736,804	\$8,685,932	\$7,380,249	0.25994	1384.14	0	ALLHSP	A	O	2
...										

FINDATA

REGION	HOSPNO	PRODUCT	SETTING	YEAR	BILLED	ALLOWED	PAID	ADSVIS	AVGPAID	AVGDISC	PCTTOT
0	ALLHSP	A	I	1	\$17,293,887	\$12,785,388	\$10,851,060	2,567	4227.14	0.2607	0.62642
0	ALLHSP	A	I	2	\$17,634,836	\$13,145,154	\$11,158,303	2,600	4291.66	0.25459	0.6019
0	ALLHSP	A	I	3	\$25,510,078	\$19,332,633	\$16,589,907	3,152	5263.3	0.24216	0.63916
0	ALLHSP	A	O	1	\$10,258,851	\$7,625,006	\$6,471,418	4,862	1331.02	0.25674	0.37358
0	ALLHSP	A	O	2	\$11,736,804	\$8,685,932	\$7,380,249	5,332	1384.14	0.25994	0.3981
...											

TRANSDAT

REGION	HOSPNO	PRODUCT	SETTING	AMOUNT	_LABEL_	YEAR_1	YEAR_2	YEAR_3
0	ALLHSP	A	I	billed	Billed	17293886.98	17634835.9	25510077.85
0	ALLHSP	A	I	allowed	Allowed	12785388.02	13145154.26	19332632.55
0	ALLHSP	A	I	paid	Paid	10851060.27	11158303.03	16589907.11
0	ALLHSP	A	I	advis	Adms/Visits	2567	2600	3152
0	ALLHSP	A	I	avgdisc	Average Discount	0.26	0.25	0.24
...								

TRANSSOR

REGION	HOSPNO	PRODUCT	SETTING	AMOUNT	_LABEL_	YEAR_1	YEAR_2	YEAR_3
0	ALLHSP	A	I	advis	Adms/Visits	2567	2600	3152
0	ALLHSP	A	O	advis	Adms/Visits	4862	5332	6241
0	ALLHSP	B	I	advis	Adms/Visits	1064	1060	732
0	ALLHSP	B	O	advis	Adms/Visits	2773	2737	1725
0	ALLHSP	A	I	allowed	Allowed	12785388.02	13145154.26	19332632.55
...								

* Blue Cross and Blue Shield of Texas, a Division of Health Care Service Corporation, a Mutual Legal Reserve Company, an Independent Licensee of the Blue Cross and Blue Shield Association