Data Dependent Footnotes using PROC REPORT

Angelina Cecilia Casas M.S., PPD Development, Research Triangle Park, NC

ABSTRACT

Sometimes it can more convenient to create tables and listings using ***PROC REPORT***, instead of ***DATA _NULL_*,** even though there is greater flexibility with ***DATA _NULL_*** for titles and footers.

This paper gives an idea of how to manipulate data so the easiness of ***PROC REPORT*** can be used while keeping some of the flexibility of ***DATA _NULL_*.**

INTRODUCTION

PROC REPORT in ***SAS 8.2*** provides new options like *_page_* that allow for more than 10 lines in headers and footnotes. Still titles and footnotes can not vary depending on the data. Before ***PROC REPORT*** may be used to construct tables or listing reports it is necessary to manipulate the data to ensure that the data structure fits the requirements of ***PROC REPORT*** and the design of the Table or Listing. The objective of this presentation is demonstrate how to produce 'footnotes' in ***PROC REPORT*** that indicate if the data for a patient breaks over on to the next page.

CREATING THE NEXT PATIENT IDENTIFIER

Several manipulations of the data using the data step have to be carried out prior to using the report procedure for producing the appropriate footer. These steps are: 1) Create a variable that has the value of the Patient identifier that will be printed on the next page, and 2) Create a variable that will indicate when a new page will start.

The Next Patient Identifier may be created via two different methods. Method 1: The data is sorted in descending order using the ***SORT PROCEDURE*** and the new variable is created in a data step using the ***LAG*** function.
Method 2: ***PROC SQL*** is used to create the dataset, and then use is made of a data step. The second method using ***PROC SQL*** allows modifications and sorting the data in one procedure.

```
PROC SQL flow=16;

  create table ae as select

    patient,
    seqno,
    body label 'Body System',
    preft label 'Preferred Term'

    from here.ae

    order by patient desc, seqno desc
    ;

    select patient, seqno, body from ae
    ;

quit;
```

A partial output of the data is displayed:

patient	seqno	BODY
04	09	Infections and infestations
04	08	Infections and infestations
04	07	Cardiac disorders
04	06	Vascular disorders
04	05	Gastrointestinal disorders
04	04	Investigations
04	03	Investigations
04	02	Vascular disorders
04	01	Nervous system disorders
03	06	Infections and infestations

After the data has been sorted in descending order it is possible to create The Next Patient Identifier variable (***NXTPT***). This variable assumes the value of the next patient. This variable is created using the ***LAG*** function.

```
Data ae;

  set ae;

    by descending patient
       descending seqno;

  nxtpt =lag(patient);

run;
```

After the variable has been created the data is sorted into a suitable order for presentation.

```
proc sort data=ae;

  by patient seqno;

run;

PROC SQL;

  select patient, nxtpt, seqno from ae;

quit;
```

A partial output of the data is displayed:

patient	nxtpt	seqno
01	01	01
01	02	02
02	02	01
02	02	02
02	02	03
02	02	04
02	02	05
02	03	06
03	03	01

CALCULATING A VARIABLE PAGE THAT WILL INDICATE A PAGE CHANGE.

PROC REPORT does not have a temporary or permanent variable that indicates when a page will start or finish. It is necessary to provide ***PROC REPORT*** with such variable.

The ***PAGE*** variable together with ***PATIENT*** and ***NXTPT*** determine if a patients data continues on the next page. This provides **PROC REPORT** with a variable that will indicate when to go to a new page.

To calculate the page variable, it is necessary to establish:

1) The number of lines that can be printed in each page. In this case, ***MAXLINE*** will be equal to 10.

```
%let maxline=10;
```

2) The width of the variables that might use more than one line per observation. This is needed to estimate the number of lines that will be used for each record. I.e. variables that will use the ***flow*** option. In this example, the variables that will *flow* are ***BODY*** and ***PREFT.*** ***BODY*** will flow to 16 and ***PREFT*** to 17.

```
%let prelen=17;

%let bodylen=16;
```

3) The total number of lines allowed in a page, the number of lines that will be used in the header and in the footer.

The variables ***USED*** and ***PAGE*** are created in the next step. The temporary variable ***USED*** estimates the number of lines that have been '*used*' in each page cumulative to the observation. When the number of lines used is greater or equal to the macro variable ***MXLINE***, the variable ***USED*** is set to the value of lines and ***PAGE*** is increased by one.

```
data ae;

  set ae;
  by invid patient seqno;

  if _n_=1 then do;
    page=1;
    used=0;
  end;
```

** For the first record, page is equal to one and the number of lines used in page one is zero;

```
lenpreft=ceil(length(trim(left(preft)))/&prelen)
;
```

***** lenpreft*** is a temporary variable that has the number of rows that will be used by the variable ***PREFT.***

```
lenbody=ceil(length(trim(left(body)))/&bodylen);
```

***** lenbody*** is a temporary variable that has the number of rows that will be used by the variable ***BODY.***

```
  lines=max(lenpreft,lenbody);
```

***** Lines*** is the number of rows that will be used by the record;

```
  used = used + lines;
```

***** Used*** has the number of lines used in the page;

```
   if used>=&mxline then do;

     page+1;
     used=lines;

   end;

    retain page used;

  drop lenpreft lenbody;

run;
```

A partial output of the data is displayed:

patient	nxtpt	seqno	page	lines	used
01	01	01	1	2	2
01	02	02	1	2	4
02	02	01	1	2	6
02	02	02	1	2	8
02	02	03	2	2	2
02	02	04	2	2	4
02	02	05	2	2	6
02	03	06	2	2	8
03	03	01	3	2	2
03	03	02	3	2	4
03	03	03	3	2	6
03	03	04	3	3	9

Beware that the number of lines per record is a crude estimate. It is not truncating words in the middle and or adding other characters for indentation. A more complicated algorithm that wraps variables properly and counts the number of lines used on each observation should be used.

HOW MANY PAGES ARE GOING TO BE IN THE OUTPUT?

It is convenient to use ***PROC SQL*** to create the macro variable ***MAXPAGE*** that has the number of pages in the output.

```
PROC SQL noprint;
  select
  left(trim(put(max(page),3.))) into :maxpage
  from ae
  ;
quit;
```

IS THERE A CHANGE OF PATIENT AT THE END OF THE PAGE?

At the end of the page we need to check if the variable ***NXTPT*** has the same variable than the variable ***PATIENT***.

```
proc sort data=ae;
  by page patient seqno;
run;

data ae;
  set ae;
  by page patient seqno;

  if last.page and patient=nxtpt then flag=1;
  else flag=0;

  Xflag=flag;

  ** flag and xflag are variables used to
determine if the subjects data is split into
multiple pages at the end of the page.

run;
```

A partial output of the data is displayed:

patient	nxtpt	seqno	page	flag
01	01	01	1	0
01	02	02	1	0
02	02	01	1	0
02	02	02	1	1
02	02	03	2	0
02	02	04	2	0
02	02	05	2	0
02	03	06	2	0
03	03	01	3	0
03	03	02	3	0
03	03	03	3	0
03	03	04	3	1

PROC REPORT

After the data is ready we can invoke ***PROC REPORT*.**

```
PROC REPORT data=ae
  spacing=1 nocenter missing headskip
  headline  split='*' list   nowd;
  column("__"
  page flag xflag patient seqno body preft);
```

* ***GROUP:*** A variable where a break or compute may occur;
* ***MAX:*** Variables where the max will be calculated;
* ***MIN:*** Variables where the minimum will be calculated;
* ***ORDER=INTERAL:*** Keep the original order that the data has;
* ***NOPRINT:*** The variable is not going to be printed, it is in the list of variables because there will be a break or compute based in that variable or because of the order of that variable;

```
    define page    /group
                    order=internal
                    noprint;

    define flag    /max
                    order=internal
                    noprint;

    define xflag   /min
                    order=internal
                    noprint;

    define patient /group
                    order=internal
                    width=6
                    "Pat.*Num.";

    define seqno   /group
                    order=internal
                    center
                    width=4
                    "Seq.*No.";

    define Body    /group
                    left
                    flow
                    width=&prelen
                    "Body System";

    define preft   /group
                    flow
                    width=&bodylen
                    "Preferred Term";
    compute before page;

      maxflag=flag.max;
      minflag=xflag.min;

    endcomp;
```

* When a variable has been declared ***GROUP*** or ***ORDER*** the variable is going to be printed only the first time that it has a new variable and when a new page starts.

It is possible to **BREAK** before and after these variable changes. ***BREAK*** is physical action that will happen with the output like skip a line, change page, underline, bold.

It is also possible to compute before and after these variables.

Inside ***PROC REPORT*** it is possible to break and compute before and after ***_PAGE_*** and before and after **REPORT**.

* Statistics for variables declared as ***ANALYSIS, MIN OR MAX*** can be calculated grouped for each of the variables that were declared ***GROUP*** or ***ORDER*** .

```
    break after page/page;

    compute before _page_;
       line @39 'Page' page 3. " of &maxpage";
       line ' ';
    endcomp;
```

line is the equivalent of put;

```
    compute after _page_;`
      if maxflag^=minflag then
        lline="Patient continues in next page";
      else
        lline="do not continue ";

      line @1 "Max:"   maxflag 2.
              " Min:"  minflag 2.
              " Flag:" flag 2.
              lline $60.
              patient $5.
              ;
      line @1
'_____________________________________________';
    endcomp;
run;
```

SELECTED PAGES FROM PROC REPORT OUTPUT:

```
All Adverse Events
                                          Page  1 of 6
_____________________________________________________________
  Pat.   Seq.
  Num.   No.  Body System      Preferred Term
_____________________________________________________________

  01      01  Cardiac          Coronary artery
              disorders        disease NOS
          02  Infections and   Urinary tract
              infestations     infection NOS
  02      01  Nervous system   Peripheral
              disorders        sensory
                               neuropathy
          02  Nervous system   Peripheral
              disorders        sensory
                               neuropathy

 Max: 1 Min: 0 Flag:  Patient continues in next page
_____________________________________________________________
```

```
 All Adverse Events
                                          Page  2 of 6
_____________________________________________________________
  Pat.   Seq.
  Num.   No.  Body System      Preferred Term
_____________________________________________________________

  02      03  Cardiac          Coronary artery
              disorders        disease NOS
          04  Infections and   Urinary tract
              infestations     infection NOS
          05  Investigations   Body
                               temperature
                               increased
          06  Nervous system   Cerebrovascular
              disorders        accident NOS


 Max: 0 Min: 0 Flag:  do not continue
_____________________________________________________________
```

```
All Adverse Events
                                                    Page  6 of 6
_____________________________________________________________________
  Pat.   Seq.
  Num.   No.  Body System       Preferred Term
_____________________________________________________________________

  04      06  Vascular         Venous
              disorders        thrombosis
                               superficial limb
          07  Cardiac          Coronary artery
              disorders        disease NOS
          08  Infections and   Urinary tract
              infestations     infection NOS
          09  Infections and   Urinary tract
              infestations     infection NOS


 Max: 0 Min: 0 Flag:  do not continue
_____________________________________________________________________
```

COMMENTS ON PROC REPORT
When a variable breaks ***PROC REPORT*** only knows about the values that happened to that variable. In this example, ***PROC REPORT*** does not know about the values or changes in the variables flag or patient. At the end of each page, there is code to print ***minflag, maxflag, flag*** and ***patient*** . But the values for ***FLAG*** and ***PATIENT*** are not printed after the ***_page_*** break.

It is very useful to have the break after, break before and calculate before and after ***_page_***. If breaks were asked after the variable ***PAGE,*** the lines would be printed immediately after the last line of data, not at the bottom of the page. It allows you to write a 'header' or a 'footer' that is over 10 lines at the end of the page, without having to add additional records to the dataset to keep the footer at the end.

CONCLUSION

I was not able to perform any calculations based in conditions like if ***MAXFLAG=1.*** It was necessary for me to compare it to another variable that was calculated in the same break, In this case ***MINFLAG.***

In versions of ***SAS 6.12*** and below, the options associated with ***_page_*** were not available; it is necessary to add records to force the footer to be printed at the bottom of the page.

PROC REPORT is an easy tool to report data, but the data needs to be manipulated before the ***PROC REPORT*** step.

Because of the group, order and flow option, it is easier to output data. However, several macros need to be created to make the data flow and indent properly.

If you want to print the patient value in the page, you would need to create a numeric variable for patient, because only numeric variables can be analyzed. But a format could be associated to that numeric variable.

ACKNOWLEDGMENTS

I want to thank Alex and Irene Gray, Bonnie Duncan, Jim Nezamis and Kim Sturgen for their contributions.

CONTACT INFORMATION

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

A. Cecilia Casas
PPD Development
3900 Paramount Parkway
Morrisville, NC 27560
Phone: (919) 462-4199
Fax: (919) 462-4128
Email: Cecilia.Casas@rtp.ppdi.com