

A SAS Macro for Breaking Pages

Michael Knoessl, Boehringer Ingelheim Pharma GmbH & Co. KG, Ingelheim, Germany

ABSTRACT

Statistical analysis of clinical data is reported primarily as summary tables and listings. Many tables are specified as to display more rows than do fit on a single page. Such tables must be divided across two or more pages. However, breaking pages automatically is not a trivial task. The main problem is to identify the breakpoints, because it is the row (or group of rows) that shall be displayed on the page following the break, which identifies the row that shall be displayed last on the page preceding the page break. Determining the two rows between which a page break is to be introduced has to occur in the SAS® dataset that contains the result data. This paper describes the macro %PAGEBREAKFLAG, which creates a counter variable that consecutively counts the prospective pages of the output table, and adds this variable to result data containing SAS® dataset.

INTRODUCTION

Statistical analysis of clinical data is reported primarily as summary tables and listings. The length of many of these tables exceeds the space provided on a single page and therefore must be broken up to be displayed on two or more pages. Unless the specification of the table does not explicitly define the points to break the table across pages, in principle, the programmer is free to break a table between any two rows. However, to ease reading of a table and for aesthetic reasons the rows in tables typically are grouped. These groups are a collection of rows, which display information that belong together as regards content, and are separated by empty break lines. Thus, the structure of the table itself imposes positions for breaking up a table and a table split across two or more pages intuitively appears to be created easily. Yet, breaking pages automatically is not a trivial task. The main problem is to identify the two rows between which to break, because it is the rows (or group of rows) that shall be displayed on the page following the break, which permit identifying the row that shall be displayed last on the page preceding the page break. This paper presents a SAS® macro called %PAGEBREAKFLAG, which identifies breakpoints and accordingly creates a counter variable that consecutively counts the pages of the prospective table on which the results are to be displayed and adds this variable to the dataset that contains the result data.

SPECIFICATIONS FOR BREAKING PAGES

In summary tables the rows displayed commonly are grouped with respect to content (e.g. groups sex, age, race in demographic tables; system organ classes (SOC) as groups in medical history tables), with groups often being separated by empty break lines. Each group typically is composed of a heading row (e.g. "Age", without results) and rows for the categories that were analyzed (e.g. "< 65 y", ">= 65 y", "Missing").

Spec-1: A table preferably should be broken down across two or more pages between and not within groups. Thus, the groups provide a first means to identify positions for breaking pages.

Spec-2: The number of rows per group, however, varies greatly, ranging from a very few rows (e.g. sex, age) up to more rows than would fit on a single page (e.g. SOC groups). Therefore, cases must be considered, where page breaks are to be introduced within a group.

Spec-3: Small and large groups may be displayed together in a table. Prioritizing solely groups as potential breaking points then may result in tables with only a few rows displaying data and many rows void of any display on some pages, while other pages are filled up with display rows. Ugly, after all. Thus, the size of the groups in relation to the size of the table body must be acknowledged (the table body is that part of a table where data are presented, excluding titles, footnotes, table header, table border lines).

Spec-4: Environmental aspects, in addition to these conceptual aspects, have to be taken into account as well. For example the size (i.e. number of rows) available for tabular output per page (i.e. table body), the number of rows required for the table header and for titles.

PREREQUISITES AND ASSUMPTIONS

The macro %PAGEBREAKFLAG is based on the following assumptions:

- the tables are created with PROC REPORT
- the data displayed in a table are contained in a SAS®-dataset, which is the input dataset for PROC REPORT
- this dataset, in addition to the displayed data, contains metadata that indicate and order the data into groups
- the macro should be independent of any environmental settings

THE MACRO %PAGEBREAKFLAG

MACRO CALL

```
%PAGEBREAKFLAG (indata      = ,
                  outdata     = ,
                  byvars      = ,
                  tablines    = ,
                  blanklines  = ,
                  voidpct     = ,
                  infoexec    = ,
                  deldata     = );
```

MACRO PARAMETERS

The following table gives a brief description of each %PAGEBREAKFLAG macro-parameter:

Parameter	Description
INDATA	one- or two-level dataset name of the input dataset
OUTDATA	one- or two-level dataset name of the output dataset
BYVARS	list of one or more BY-variables that identify groups of data and the order of the data as they will appear in the table; the last variable in this list must identify each group of data uniquely
TABLINES	number of rows of which the table body is comprised of
BLANKLINES	YES/NO-indicator whether groups of data will be separated by an empty line in the table
VOIDPCT	percentage (0 – 100) of allowed void area in the table body; i.e. the number of rows expressed as percentage of the total number of rows of the table body that are allowed to be left empty
INFOEXEC	YES/NO-indicator whether information about macro-execution is to be printed into the LOG
DELDATA	YES/NO-indicator whether temporary datasets created during macro-execution are deleted or kept at macro-termination

The parameters INDATA and OUTDATA specify the macro-input and –output datasets. The parameters INFOEXEC and DELDATA provide information about the execution of the macro and allow for an examination of how the macro interprets and treats the result data provided in the input dataset (might be useful while developing a program). The other four parameters reflect the specifications for breaking pages (Spec-1 through Spec-4; see above) and thus provide functional information. They are explained in more detail in the following section.

ARCHITECTURE OF THE MACRO

The macro %PAGEBREAKFLAG consists of the following sections:

1) Initialization

In this section the current settings of selected SAS® system options of the macro category (e.g. MPRINT) are determined and stored. The SAS® macro system options are redefined as inactive (e.g. set to NOMPRINT) for the time of macro execution (Spec-4). In addition, if requested by parameter INFOEXEC, information about the date and time of starting this macro is printed to the LOG.

2) Check of macro parameters for valid values

To ensure that the macro executes without errors that can be anticipated, the values of all macro parameters are checked for validity. If at least one check determines an invalid parameter value, the macro terminates without executing further.

For example: The required parameter INDATA must contain a value and the specified dataset must exist. The parameter BYVARS must contain at least one variable, and this variable must exist in the input dataset. The parameter VOIDPCT must be populated with an integer between "0" and "100". The parameters BLANKLINES, INFOEXEC, and DELDATA are, if not specified as "NO", re-specified to "YES".

3) Functional macro code: Determination of page breaks

a) The input dataset is read, sorted by the variables defined in parameter BYVARS (Spec-1) and stored in a temporary dataset.

b) The data values of the variable that is specified last in BYVARS is used to determine the groups of data values (e.g. the 4 rows "Age", "< 65 y", ">= 65 y", "Missing" constitute one group of data). The groups as well as the number of dataset records (i.e. rows to be displayed) per group are counted consecutively and stored in the newly created dataset variables "_CATGRP" and "_CATROW", respectively (Spec-1; Table-1). If the groups are to be displayed with an empty line separating them, the counter variable for the rows per group is incremented by "1" for this additional displayed row. The new variable "_RECNO" counts the rows in the table body that will take the data of the respective dataset record. In addition, a new variable "_RECFREE" is created that counts the number of rows of the table body, starting with the value specified in parameter TABLINES, that would remain once the current record would have been displayed in the table body (Spec-3). Last, the variable "_PAGECNT", which eventually will count the pages, is initiated.

PhUSE 2007

The following table gives the resulting dataset, using the parameters: BYVARS = group varname; TABLINES = 17; BLANKLINES = YES; the variable COLUMN1 contains the data that are to be displayed in the table (the result data are ignored here).

Table-1: Metadata created in the Macro

Obs	GROUP	VARNAME	COLUMN1	_CATGRP	_CATROW	_RECNO	_RECFREE	_PAGECNT
1	0		Number of Subjects	1	2	2	15	0
2	1	AGE	Age [years]	2	4	4	13	0
3	1	AGE	Mean (SD)	2	4	5	13	0
4	1	AGE	Range	2	4	6	13	0
5	2	AGEGRP1	Age group, N (%)	3	5	8	9	0
6	2	AGEGRP1	<65 years	3	5	9	9	0
7	2	AGEGRP1	>=65 years	3	5	10	9	0
8	2	AGEGRP1	Missing	3	5	11	9	0
9	3	AGEGRP2	Age class, N (%)	4	8	13	4	0
10	3	AGEGRP2	18-39 years	4	8	14	4	0
11	3	AGEGRP2	40-49 years	4	8	15	4	0
12	3	AGEGRP2	50-59 years	4	8	16	4	0
13	3	AGEGRP2	60-69 years	4	8	17	4	0
...	...	...	...	...	...	...	...	...

c) This dataset is read again. The values of the various counter variables provide enough information to determine the rows where a page break should be introduced according to the 4 specifications depicted above. Essentially, as long as the number of _CATROW (number of rows per group) is lower or equal than the number of _RECFREE (number of remaining rows in the table with the current record being displayed), the respective dataset records will fit on one page and they are output in a separate dataset (DS-A; Spec-1; Table-2). The number of remaining rows in variable _RECFREE allows to calculate the percentage of space that remains void of data in the table body. If this space is smaller than allowed (as specified in parameter VOIDPCT) and the group currently processed would fit on the prospective current page, the records of this group are also output to the same separate dataset (DS-A; Spec-3). If, however, a group is composed of a number of rows that would not fit on a page, i.e. _RECNO exceeds the number of rows allowed in the table body (as specified in parameter TABLINES) and the other criteria are not fulfilled, only as many records are output into the separate dataset (DS-A) as would fit in the table body and the prospective page break will occur within a group (Spec-2).

The variable _PAGECNT is then populated with the number of the current iteration, which matches the prospective page number (see step d)).

The following table gives the resulting dataset (DS-A), which contains all dataset records that will be displayed together on one page.

Table-2: Intermediate Dataset containing data displayed together on one page

Obs	GROUP	VARNAME	COLUMN1	_CATGRP	_CATROW	_PAGECNT
1	0		Number of Subjects	1	2	1
2	1	AGE	Age [years]	2	4	1
3	1	AGE	Mean (SD)	2	4	1
4	1	AGE	Range	2	4	1
5	2	AGEGRP1	Age group, N (%)	3	5	1
6	2	AGEGRP1	<65 years	3	5	1
7	2	AGEGRP1	>=65 years	3	5	1
8	2	AGEGRP1	Missing	3	5	1

Once a record is reached where none of the criteria explained above are met (e.g.: _CATROW becomes larger than _RECFREE; in the example above all records where OBS >= 9), this and the following records are output into another temporary dataset (DS-B). This dataset now contains the data that will be displayed in all following prospective pages.

d) The steps described in b) and c) are iterated on dataset DS-B (as created after each iteration), until all records of the input dataset have been processed. Eventually, this iteration results in as many datasets (DS-A) as there will be pages of the final table, with the respective page number captured in the variable _PAGECNT.

e) The individual "page-datasets" (DS-A) are set together to create the final, single output dataset (named as specified in parameter OUTDATA).

4) Termination

The SAS® system options that were modified during initialization of this macro are reset to the values they were defined to outside of this macro (Spec-4). All datasets created during execution of the macro are deleted (except the output dataset, specified in parameter OUTDATA). If requested by parameter INFOEXEC, information about the date and time of termination of this macro is printed to the LOG.

PhUSE 2007

CONCLUSION

The macro %PAGEBREAKFLAG provides a flexible tool to automatically create a counter-variable for page numbers in a result dataset. This counter-variable can be utilized by PROC REPORT in the output section of a clinical program to create a multi-page tabular output with the displayed data being divided pleasantly across pages.

CONTACT INFORMATION

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

Author Name:	Dr. Michael Knoessl
Company:	Boehringer Ingelheim Pharma GmbH & Co. KG
Address:	Binger Strasse 173
City / Postcode:	55216 Ingelheim am Rhein
Work Phone: :	+49-6132-77-97055
Fax: :	+49-6132-72-97055
Email: :	michael.knoessl@ing.boehringer-ingelheim.com
Web:	www.boehringer-ingelheim.com

Brand and product names are trademarks of their respective companies.