

Paper CT_27

Display, Group or Order: Using Proc Report® to create Clinical Trials Output

Sally Walczak, Quintiles, RTP, NC

ABSTRACT

PROC REPORT is a Base SAS® procedure that simplifies the creation of custom reports. Sometimes it can add a level of frustration to creating the tables and listings that you desire. This paper will explore the different variable definitions: DISPLAY, GROUP and ORDER and associated options with each. This will allow you to confidently use the correct DEFINE statement the first time to get the output you really want.

INTRODUCTION

PROC REPORT is a procedure in Base SAS software which allows users to easily create custom reports. PROC REPORT packs a powerful punch! You can create many customized reports effortlessly. It does many things “behind the scenes” to facilitate producing the output. One of those “behind the scenes” tasks is sorting. This is controlled by the type of variables presented on the table, which is defined on the DEFINE statement. The types on the DEFINE statement can be DISPLAY, GROUP, ORDER, ANALYSIS or COMPUTED. We will look at DISPLAY, ORDER and GROUP only. Now let’s see when you would want to use each of these and how to solve some common issues related to their use.

DISPLAY

This one is pretty self explanatory – use DISPLAY to just write out a value. This is the default type for character variables in PROC REPORT. It does not impact the ordering of the observations. The table is based on the order of the observations in the input data set. Pretty easy, right?

ORDER

Now we’re getting a little trickier. For this type, think PROC SORT. If you define a variable as ORDER, by default PROC REPORT will sort the data based on the formatted value of this variable in ascending order. Another neat effect is duplicate values of an ORDER variable will not repeat on the table. Now let’s begin with a very simple example.

This data set contains a MONTH variable. The values are 1, 2, 3 ... 12 and you have created a format to display January, February, etc. By default, if you define MONTH as an ORDER variable, it will use the formatted value to write the table. Here’s the code:

```
proc format;
  value mth
    1 = 'January'
    2 = 'February'
    3 = 'March'
    <remainder of format values>
    12 = 'December'
  ;

data expenses;
  input month dept $ @9 amount ;
  format month mth. amount dollar10.2;
  cards;
1  BIOS      2500.00
5  STAT      750.00
1  STAT      1575.95
2  BIOS      1250.25
2  BIOS      5000.50
4  MKT       10500.00
;
```

```

proc sort;
  by month dept;
run;

proc print;
  title '2011 Expenses';
run;

proc report data=expenses nowd headline headskip formchar (2)="_" spacing=2;
  column month dept amount;
  define month / order width=25 "Month of Purchases";
  define dept / display width=25 "Department";
  define amount / display width=15 "Amount Reconciled";
run;

```

2011 Expenses			
Obs	month	dept	amount
1	January	BIOS	\$2,500.00
2	January	STAT	\$1,575.95
3	February	BIOS	\$1,250.25
4	February	BIOS	\$5,000.50
5	April	MKT	\$10,500.00
6	May	STAT	\$750.00

Output 1. Output from PROC PRINT

Expenses			Amount Reconciled
Month of Purchases	Department		
April	MKT		\$10,500.00
February	BIOS		\$1,250.25
	BIOS		\$5,000.50
January	BIOS		\$2,500.00
	STAT		\$1,575.95
May	STAT		\$750.00

Output 2. Output from PROC REPORT

The easiest fix for this it to use ORDER=INTERNAL on the DEFINE statement, i.e.:

```
define month / order order=internal width=25 "Month of Purchases";
```

So that example was pretty straight-forward. How about when you're working with SAS dates? Let's say you have a few variables to define your order such as:

SUBJID	character
VISITDT	SAS date, formatted as DATE9.
VISDESC	character
VISID	numeric

You need to create your table in date order but you do not want to display the date on the table. If you use VISDESC as your order variable, the rows will be based on the alphabetical value of VISDESC which is probably not what you want.

The easy solution here is to add the VISITDT variable defined as ORDER and also use the options NOPRINT and ORDER=INTERNAL. But if you have duplicate dates for the same subject, you can run into the issue displayed below:

```

proc format;
  value visits
    1 = 'Screening Visit'
    2 = 'Extra Dose Visit'
    3 = 'Normal Dose Visit'
    9 = 'Unscheduled Visit'
  ;

data table;
  input subjid $ visitdt : date9. visid trt $;
  length visdesc $50;
  visdesc=put(visid,visits.);
  format visitdt date9.;
  cards;
1225 05feb2012 1 A
1225 15mar2012 2 A
1225 14apr2012 9 A
1234 01jan2010 1 A
1234 15jan2010 2 B
1234 05feb2012 3 A
1234 15mar2012 9 C
1111 05jan2011 1 C
1111 05jan2011 9 A
1111 05jan2011 2 A
  ;

title "Proc Report Output";
proc report data=table nowd headline headskip formchar (2)="_" spacing=2
split='|';
  column subjid visitdt visdesc visid trt;

  define subjid    /order width=25 "Subject ID";
  define visitdt   /order order=internal noprint;
  define visdesc   /order width=25 "Visit|Description";
  define visid     /order "Visit ID";
  define trt       /display width=15 "Treatment|Code";
run;

```

Subject ID	Visit Description	Visit ID	Treatment Code
1234	Screening Visit	1	A
	Extra Dose Visit	2	B
	Normal Dose Visit	3	A
	Unscheduled Visit	9	C
1111	Extra Dose Visit	2	C
	Screening Visit	1	A
	Unscheduled Visit	9	A
1225	Screening Visit	1	A
	Extra Dose Visit	2	A
	Unscheduled Visit	9	A

Output 3. Output from PROC REPORT – ordering issue

The first thing you notice is that your data is displayed in sorted order based on SUBJID and VISITDT. But take a closer look at SUBJID='1111'. Those rows are ordered alphabetically by VISDESC. To correct this, you need to add a numeric ordering variable before the VISDESC variable on the COLUMN statement. You could put VISID before VISDESC on the COLUMN statement since it is already defined as ORDER but it will change the output by displaying VISID before VISDESC. If that is not ok, you need to create a new ordering variable. We will call ours ORD and it will determine the order. You will put ORD before VISDESC on the COLUMN statement. Our updated code is:

```

proc sort data=table;
  by visitdt visid;
run;

data table;
  set table;
  by visitdt visid;
  if first.visitdt then ord=0;
  ord+1;
run;

title "Proc Report Output";
proc report data=table nowd headline headskip formchar (2)="_" spacing=2
split='|';
  column subjid visitdt ord visdesc visid trt;

  define subjid    /order width=25 "Subject ID";
  define visitdt   /order order=internal noprint;
  define ord       /order noprint;
  define visdesc   /order left width=25 "Visit|Description";
  define visid     /order left "Visit ID";
  define trt       /display width=15 "Treatment|Code";
run;

```

Sometimes it is not that easy to just create an ORD variable to solve an ordering problem. You may need to create a new variable whose value is the combination of 2 variables.

Using our same example above, let's say that we have now been requested to present the INVID and the SUBJID variables on the table and they are both used as ORDER variables. The tricky part is we want the INV variable to repeat every time there is a new value of SUBJID. By default, PROC REPORT will only print an ORDER variable when there is a new value. We could make it a DISPLAY variable but that will impact our ordering. In this case, the best solution will be to create a new ID variable and use that as our order variable.

```

data table;
  set table;
  length IDVAR $8;
  idvar= catx('-',invid,subjid);
run;

```

In the Proc Report code, we will replace this line:

```
define subjid    /order width=25 "Subject ID";
```

with this:

```
define idvar     / order width=25 "Subject ID";
```

You can also use ORDER=DATA with the ORDER type. When this is specified, SAS presents the rows in the order they are in the data set. If the first subgroup of values does not contain every possible value, then the order of the output rows may not be what you were expecting. To understand this better, let's look at another example. For this example, we are using slightly different data.

```

data table;
  input invid $ subjid $ visitdt : date9. visid trt $;
  length visdesc $50;
  visdesc=put(visid,visits.);
  format visitdt date9.;
  cards;
421 1225 05feb2012 1 A
421 1225 15mar2012 2 A
421 1225 14apr2012 9 A
101 1234 01jan2010 1 A
101 1234 01jan2010 9 A
101 1234 15jan2010 3 B

```

```

101 1234 05feb2012 3 A
101 1234 15mar2012 9 C
102 1111 05jan2011 1 C
102 1111 05jan2011 9 A
102 1111 05jan2011 2 A
102 1111 05jan2011 3 A
;

title "Proc Report Output";
proc report data=table nowd headline headskip formchar (2)="_" spacing=1
split='|';
  column invid subjid visitdt visid visdesc trt;

  define invid      /order width=10 "Facility ID";
  define subjid     /order width=10 "Subject ID";
  define visitdt    /order order=internal width=12 "Visit|Date";
  define visid      /order order=data left width=5 "Visit|ID";
  define visdesc    /order left width=20 "Visit|Description";
  define trt        /order width=10 "Treatment|Code";
run;

```

Notice the output for INVID=102 and SUBJID=1111. The VISID values are not listed in ascending order. This is because the first subgroup (INVID=101, SUBJID=1234, VISITDT=01Jan2010) did not contain all of the values of the variable VISID.

Proc Report Output					
Facility ID	Subject ID	Visit Date	Visit ID	Visit Description	Treatment Code
101	1234	01JAN2010	1	Screening Visit	A
			9	Unscheduled Visit	A
		15JAN2010	3	Normal Dose Visit	B
		05FEB2012	3	Normal Dose Visit	A
		15MAR2012	9	Unscheduled Visit	C
102	1111	05JAN2011	1	Screening Visit	C
			9	Unscheduled Visit	A
			3	Normal Dose Visit	A
			2	Extra Dose Visit	A
421	1225	05FEB2012	1	Screening Visit	A
		15MAR2012	2	Extra Dose Visit	A
		14APR2012	9	Unscheduled Visit	A

Output 4. Output from PROC REPORT using ORDER=DATA

Here are a few best practices when using the ORDER type:

- The ordering of the ORDER variable is dependent on the ORDER= value, ORDER=FORMAT is the default, and the ordering of any COLUMN statement ORDER variable that is located to the left.
- BIG → LITTLE. I heard this from a colleague and it sums it up perfectly! In our example, the VISITDT variable is a larger subgroup because you can have repeating dates. So you want it to the left of your smaller subgroup – in our example the variable VISID. PROC REPORT processes the report-items as they are listed on the COLUMN statement in a LEFT to RIGHT order.
- Use as many numeric hidden variables as needed. Don't be afraid of creating numeric variables to determine the order and use them with the NOPRINT option.

GROUP

The last type we will discuss is the GROUP type. From the SAS 9.2 documentation: "If a report contains one or

more group variables, then PROC REPORT tries to consolidate into one row all observations from the data set that have a unique combination of formatted values for all group variables.” So the GROUP type works like the CLASS statement on some procedures, like PROC TABULATE. In most ways, GROUP is similar to ORDER. Sometimes you can define a variable as ORDER or GROUP and get the exact same results. To be honest, I rarely use the GROUP type but we will look at an example of listing observations based on frequency order.

The ONLY difference between GROUP and ORDER is that GROUP tries to consolidate the observations/rows based on the unique values of the GROUP variables -- just like PROC SUMMARY, PROC MEANS and PROC TABULATE. ORDER does not consolidate and every observation from the input dataset is printed out per row, i.e. 10 obs = 10 rows when ORDER is specified (assuming the MISSING option is also specified on the PROC REPORT statement).

Let's look at that example now:

```
data table;
  infile cards trunccover;
  input invid $ subjid $ visitdt : date9. pterm : $25.;
  usubjid=catx('-',invid,subjid);
  format visitdt date9.;
  cards;
421 1225 05feb2012 Anaemia
421 1225 15mar2012 Leukopenia
421 1225 14apr2012 Anaemia
101 1234 01jan2010 Leukopenia
101 1234 01jan2010 Leukopenia
101 1234 15jan2010 Depression
101 1234 05feb2012 Anaemia
101 1234 15mar2012 Leukopenia
102 1111 05jan2011 Anxiety
102 1111 05jan2011 Depression
102 1111 05jan2011 Anxiety
102 1111 05jan2011 Anxiety
;

title "Proc Report Output in descending FREQ order";
proc report data=table nowd headline headskip formchar (2)="_" spacing=3 split='|';
  column usubjid visitdt count pterm ;
  define usubjid /group width=10 "Subject|ID";
  define visitdt /group width=20 "Visit|Date";
  define pterm /group order=freq descending width=40 "Preferred Term";
run;
```

Proc Report Output in descending FREQ order		
Subject ID	Visit Date	Preferred Term
101-1234	01JAN2010	Leukopenia
	05FEB2012	Anaemia
	15JAN2010	Depression
	15MAR2012	Leukopenia
102-1111	05JAN2011	Anxiety
		Depression
421-1225	05FEB2012	Anaemia
	14APR2012	Anaemia
	15MAR2012	Leukopenia

Output 5. PROC REPORT Output using GROUP

Notice in the output that the preferred terms are listing in descending order of frequency by USUBJID and VISITDT. Since Anxiety has 3 observations, it is listed before Depression and only 1 row is output on the table for Anxiety. This can come in very handy and eliminates the need for an intermediate procedure or data step to count the frequencies.

In contrast, here is the output from PROC PRINT. Notice that 12 observations are output instead of 9 as with PROC REPORT using GROUP.

Proc Print Output					
Obs	invid	subjid	visitdt	pterm	usubjid
1	101	1234	01JAN2010	Leukopenia	101-1234
2	101	1234	01JAN2010	Leukopenia	101-1234
3	101	1234	15JAN2010	Depression	101-1234
4	101	1234	05FEB2012	Anaemia	101-1234
5	101	1234	15MAR2012	Leukopenia	101-1234
6	102	1111	05JAN2011	Anxiety	102-1111
7	102	1111	05JAN2011	Depression	102-1111
8	102	1111	05JAN2011	Anxiety	102-1111
9	102	1111	05JAN2011	Anxiety	102-1111
10	421	1225	05FEB2012	Anaemia	421-1225
11	421	1225	15MAR2012	Leukopenia	421-1225
12	421	1225	14APR2012	Anaemia	421-1225

Output 5. PROC PRINT Output with 12 observations

CONCLUSION

I hope that you now have a better understanding of the define types DISPLAY, ORDER and GROUP and that you possess the knowledge to know which one to use the first time. Using these types effectively can really harness the power of PROC REPORT so that it does more than just print the data.

REFERENCES

- McMahill, Allison (2007), Beyond the Basics: Advanced PROC REPORT Tips and Tricks, Proceedings of the 2007 SAS Global Forum, Paper 276-2007, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/papers/sgf07/sgf2007-report.pdf>.
- Fine, Lisa (2011), Order, Order Please: Sorting Data Using PROC REPORT, *Proceedings of the 2011 SAS Global Forum*, Paper 090-2011, Cary, NC: SAS Institute Inc. Available online at <http://support.sas.com/resources/papers/proceedings11/090-2011.pdf>.
- Base SAS® 9.2 Procedures Guide (2009). Cary, NC: SAS Institute Inc. Available online at <http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#a000146851.htm>.

ACKNOWLEDGMENTS

The author would like to thank Allison McMahon Booth, Wendy S. Baker and Julie Stevenson for reviewing this paper.

RECOMMENDED READING

- Base SAS® Procedures Guide
- Booth, Allison McMahon. 2011. "Beyond the Basics: Advanced PROC REPORT Tips and Tricks Updated for SAS® 9.2." *Proceedings of the SAS Global Forum 2012 Conference*. Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/proceedings11/246-2011.pdf.
- Booth, Allison McMahon. 2010. "Evolve from a Carpenter's Apprentice to a Master Woodworker: Creating a Plan for Your Reports and Avoiding Common Pitfalls in REPORT Procedure Coding." *Proceedings of the SAS Global Forum 2010 Conference*. Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/proceedings10/133-2010.pdf.
- SAS Institute Inc. 2012. "Find Your Answer in the SAS Knowledge Base." SAS Customer Support Web Site. Available at support.sas.com/resources/.

- SAS Institute Inc. 2008. "The REPORT Procedure: Getting Started with the Basics." *Technical Paper*. Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/ProcReportBasics.pdf.

CONTACT INFORMATION

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

Name: Sally Walczak
E-mail: sally.walczak@quintiles.com
Twitter: [sjwalczak](https://twitter.com/sjwalczak)

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 trademarks of their respective companies.