

## Paper 199-2010

## Establishing Consistent Programming for Pharmaceutical Safety Summarizations Using a DOW Loop

Bradford J. Danner, MDS Pharma Services, Lincoln, NE  
Steven R. Kirby, MDS Pharma Services, Lincoln, NE

### ABSTRACT

Three types of data collected during an early stage clinical trial study to assess safety are laboratory tests, electrocardiogram parameters, and vital signs. Simple summary statistics of the results and change from baseline are usually presented in an ICH summary report, and it is also useful to include a presentation of out of range shifts relative to baseline. When several varieties of standard programs are available to facilitate summaries, a consistent approach across all datasets and types of summaries in a study could increase efficiency and minimize the possibility of multiple errors. This paper will illustrate an approach that may be applied to various types of data regardless of vertical or horizontal structure. To minimize data step processing and superfluous merging or processing, a DOW loop is employed. Advantages and disadvantages of using this process are discussed, and preliminary results to date are presented.

### INTRODUCTION

In a majority of clinical studies, a main endpoint is the summarization of safety information. Part of this information is in the collection of clinical laboratory results, vital signs, and electrocardiogram results. The most common types of summarizations are simple summary statistics for all treatments and sampling times, change from baseline, and frequencies of out-of-range values or specifically established criteria.

Historically, a variety of programmatic approaches for providing these summaries are available. For example, providing both summary and change from baseline within the same program, or using two separate programs to provide the different types of assessments. If separate programs are used for the three types of summaries mentioned (summary, change from baseline, frequencies) for the three types of main data sources (labs, vitals, electrocardiograms), a total of 9 different programs may be active for the same project. If the 9 programs are not written consistently, de-bugging and modification may become labor-intensive.

To minimize work and increase efficiency, an easy to follow and consistent approach to summarizing data should be established. Furthermore, a robust programmatic approach that is not too heavily dependant on a specific study design would be ideal, so that it may be applied to other designs. Finally, reducing or minimizing the number of programs and/or logical constructs necessary to summarize the data naturally lends itself to reducing computational errors, and is easier to maintain if modifications are necessary.

In efforts to make programs more efficient, and given the increasing popularity of the programming construct, we modeled our logical approach for calculating changes or shifts from baseline employing a DOW loop, consistent to the approach described by Brucken (2008). A DOW (Dorfman-Whitlock) loop, as described by Dorfman and Shajenko (2008) is a nested repetitive DATA step programming structure, intentionally organized in order to allow for programmatically and logically natural isolation of DO-loop instructions related to a certain break-event from actions performed before and after the loop. Once the programming structure is initially learned, revision and modification dependant on study design is relatively straightforward.

### METHODS

A standard approach to providing summarizations, for a variety of types of clinical safety data is possible. The following is a straightforward, easy to manipulate, starting point for program establishment with vertically structured data (for example, a vitals dataset with pulse and blood pressure results in one column, and an indicator of the specific vital sign in a separate column):

1. Obtain earliest period specific medication date/time
2. Merge medication and treatment assignment with data
3. Appropriately handle predose and postdose samples
4. Sort by necessary levels for summaries
5. Employ DOW loop to find change/shift from baseline
6. Find summary stats either with 2 MEANS procedures or 1 TRANSPOSE procedure + 1 MEANS procedure
7. Format for presentation as necessary.

Under circumstances where data are structured horizontally, for example, a vitals dataset with pulse and blood pressure in separate columns, the same starting point shown above may still be used, with an appropriately placed TRANSPOSE procedure:

1. Obtain earliest period specific medication date/time
2. Merge medication and treatment assignment with data
3. TRANSPOSE data (horizontal to vertical structure)
4. Appropriately handle predose and postdose samples
5. Sort by necessary levels for summaries
6. Employ DOW loop to find change/shift from baseline
7. Find summary stats either with 2 MEANS procedures or 1 TRANSPOSE procedure + 1 MEANS procedure
8. Format for presentation as necessary.

Refer to Appendix A for an example generalization of the approach and print of the data following baseline and change from baseline establishment.

The advantages to using this approach are 1) one approach may be applied similarly to different types of data, 2) computationally, data is processed once, rather than several times which can increase processing speed, and reduce chances of committing an error, and 3) the use of the DOW loop may lend itself to simplifying the summarization of safety data for clinical trials. Once the necessary logic is established to produce summary statistics of values and change from baseline calculations, the approach may be applied to summaries of shift incidence by replacing change calculation with an indication of baseline and time point specific assessment

The disadvantages of using this approach are 1) the looping construct within the SAS® Data step is not immediately intuitive, 2) the use of last.<variable> statement coupled with the output statement must be made with care, and 3) greater care and understanding of the data is necessary prior to processing. If care is not taken with regards to any of these steps, results may not be correct, and can potentially be misleading.

## RESULTS

In a recent effort to streamline global programming efforts, a cursory review of internal standards available to provide these types of summaries revealed that over the last decade, a total of 40 programs (not including this approach) had been made available as a basis for a standard starting point for summaries of labs, vitals, and ECGs. While many were slight variations of similar programs, most were not in use at all, based on a summary of 8 of our colleagues. In fact, 29/40 programs were not used in a familiar fashion by more than any 2 programmers.

Of the remaining 11 programs in common use, disparate approaches of the programs often led to confusion when initially applying them to different study designs – an approach taken with one program was not consistent to the approach necessary with another program. Furthermore, several of the programs increased computational processing by placing SAS procedural and data step coding within internal macros, much of which was not necessary with the judicious use of the TRANSPOSE procedure and the BY statement within SAS procedures.

We began testing this approach under production situations, where QC double programming of results for numerical accuracy was necessary. Our initial impression of the approach was that once we understood the design of the study, and its effect on the structure of the data, that the remaining computational steps necessary to produce similar results was reduced. With the approach outlined here, the application of a program to different types of data (ECG vs. Vitals, or horizontally vs. vertically structured data) may be accomplished merely with a careful rename of variables and/or a well placed TRANSPOSE procedure. Furthermore, incidence of shifts in results or investigator assessment was accomplished easily with nearly identical syntax, except for relatively minor modification to change from baseline calculations within the DOW loop step above in the Methods section (refer to Appendix B for illustration).

After the initial deployment of the programs for both quality control purposes through double programming comparison and production test cases with independent QC programming, we found that in out of 8 test cases, to produce the necessary summaries, approximately half the computational time was necessary, requiring half as many programs. The approach is also highly useful as a self-checking tool, and we have found the programs very insightful for identifying errors and miscalculation. Furthermore, the general approach to program establishment lends itself well to modification and debugging for adaptation to different study designs.

## CONCLUSIONS

The approach outlined here has led to a more efficient and straightforward programming to provide basic summaries of safety data. However, one must take care when using the approach in understanding the design of the study, and have a clear definition of the appropriate baseline, whether it be singular or multiple baselines, and have a clear definition of when and how to handle rechecks and unscheduled events. In order to reduce the number of passes

through the data, one must take care when initially setting up the DOW loop. We have also found it useful to program in diagnostic prints and program breaks.

Results to date suggest the approach is more efficient, for both quality control and production purposes. However, greater care must initially be invested, and whether the approach can be rapidly assimilated by new programmers has not yet been thoroughly investigated.

## REFERENCES

Bruckner, Nancy. One-Step Change from Baseline Calculations. 2008 PharmaSUG

Dorfman, P.M. and L.S. Shajenko, The DOW-Loop Unrolled. 2008 PharmaSUG

## ACKNOWLEDGMENTS

We would like to thank our colleagues in the Biostatistics and Pharmacokinetics groups in the Department of Clinical Pharmacology at MDS who provided comments on earlier drafts. Their insights and guidance are appreciated.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please contact the authors at:

Bradford J. Danner, MS  
MDS Pharma Services  
621 Rose Street  
Lincoln, NE 68502  
[bradford.danner@mdsinc.com](mailto:bradford.danner@mdsinc.com)

Steven R. Kirby, JD, MS  
MDS Pharma Services  
621 Rose Street  
Lincoln, NE 68502  
[steven.kirby@mdsinc.com](mailto:steven.kirby@mdsinc.com)

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.

## APPENDIX A

## 1 Obtain earliest period specific medication date/time

```

data med;
  set med_data;
  ptno=subjectid+0;
  if meddate ne .;
  keep meddate medtime ptno period;

proc sort; by period ptno meddate medtime;

* keep the first medication record for each subject and period *;
data med;
  set med;
  by period ptno meddate medtime;
  if first.ptno;
  keep ptno period meddate medtime treatment;

data ecgorig;
  set ecg_data;
  ptno=subjectid+0;

proc sort; by ptno period;

```

## 2 Merge medication and treatment assignment with data

```

data prep1;
  merge ecgorig (in=flg1) med (in=flg2);
  by ptno period;
  if flg1 and flg2;

proc sort;
  by ptno period day hour meddate medtime recheck treat egdate egtime;

```

## 3 Transpose data (horizontal to vertical structure)

```

proc transpose data=templ out=templ (rename=(_name_=parameter coll=value));
  by ptno period day hour meddate medtime recheck treat egdate egtime;
  * for horizontally structured data *;
  var hr pr qt qrs qtc;
run;

title1 "Check your transpose -
for unscheduled/multiple entries - should be none unless intentional";
proc print data=templ;
  var col;;
run;

```

#### 4 Appropriately handle predose and postdose samples

```

data temp1 baseprob;
set temp1;
if hour gt 0 and recheck ne " " then delete;
if hour le 0 then do;
  if recheck ne " " and value=. then delete;
end;
  if upcase(period) not in ('SCREEN','SCREENING') then do;
    if hour lt 0 then do;
      if egdate gt meddate or (egdate=meddate and egtime gt medtime)
        then output baseprob;
      else output temp1;
    end;
    else output temp1;
  end;
else output temp1;
run;

title1 "Check your BASEPROB file - postdose measures that occur after
baseline";
proc print data=baseprob;
run;
title1 " ";

proc sort data=temp1;
by parameter testing ptno period day hour egdate egtime;

* keep first measurement postdose, last measurement (including rechecks)
postdose;
data temp1; set temp1;
by parameter testing ptno period day hour egdate egtime;
if period="SCREEN" or (hour lt 0) then do;
  if last.hour;
end;
else do;
  if first.hour;
end;

```

#### 5 Sort by necessary levels for summaries

```

proc sort data=temp1;
by parameter testing ptno period day hour ;

```

## 6 Employ DOW loop to find change/shift from baseline

```

* optional DOW approach ;
data summ;
do until (last.ptno);
  set work.templ;
  by parameter testing ptno period ;
  if (hour lt 0) or upcase(trim(left(period)))="SCREEN" then do;
    bl=value+0;
    output;
  end;
  else do;
    change=value-bl;
    output;
  end;
end;
run;

data summ;
length cat $20.;
set summ;
if (upcase(trim(left(period)))="SCREEN") or (hour lt 0) then do;
  change=.; bl=.;
end;

```

Example with data for 1 subject

parameter	ptno	PERIOD	DAY	HOURL	value	bl	change
HR	1	1	-1	-16.500	56	.	.
HR	1	1	2	22.500	45	56	-11
HR	1	1	4	70.500	49	56	-7
HR	1	1	6	119.000	53	56	-3
PR	1	1	-1	-16.500	188	.	.
PR	1	1	2	22.500	182	188	-6
PR	1	1	4	70.500	224	188	36
PR	1	1	6	119.000	190	188	2
QRS	1	1	-1	-16.500	128	.	.
QRS	1	1	2	22.500	140	128	12
QRS	1	1	4	70.500	124	128	-4
QRS	1	1	6	119.000	138	128	10
QT	1	1	-1	-16.500	430	.	.
QT	1	1	2	22.500	454	430	24
QT	1	1	4	70.500	440	430	10
QT	1	1	6	119.000	434	430	4
QTC	1	1	-1	-16.500	414	.	.
QTC	1	1	2	22.500	392	414	-22
QTC	1	1	4	70.500	397	414	-17
QTC	1	1	6	119.000	407	414	-7

- 7 Find summary stats either with 2 MEANS procedures or 1 TRANSPOSE procedure + 1 MEANS procedure

```
proc means data=summ(where=(value ne .)) noprint n mean min max std median q1
q3 cv;
by treat period testing day hour parameter;
var value;
output out=new n=n mean=mean min=min max=max std=std median=med;

proc means data=vitfinal2 (where=(change ne .)) noprint n mean min max std
median q1 q3 cv;
by treat period testing day hour parameter;
var change;
output out=change n=n mean=mean min=min max=max std=std median=med;

data new;
set new change (in=flg2);
if flg2 then do;
    type="CHG";
end;
else type="SUM";
```

- 8 Format for presentation as necessary.

**APPENDIX B**

Consider a variable <ecgrslt> that takes the value "N" for normal, and "A" for abnormal;

6 Employ DOW loop to find change/shift from baseline

```
* optional DOW approach ;
data summ;
do until (last.ptno);
  set work.templ;
  by parameter testing ptno period ;
  if (hour lt 0) or upcase(trim(left(period)))="SCREEN" then do;
    bl=strip(ecgrslt);
    output;
  end;
  else do;
    change=strip(bl)||strip(ecgrslt);
    output;
  end;
end;
run;

data summ;
length cat $20.;
set summ;
if (upcase(trim(left(period)))="SCREEN") or (hour lt 0) then do;
  change=''; bl='';
end;
```

Example with data for 3 subjects

ptno	PERIOD	DAY	HOUR	ecgrslt	bl	change
1	1	-1	-16.500	A		
1	1	2	22.500	A	A	AA
1	1	4	70.500	A	A	AA
1	1	6	119.000	A	A	AA
3	1	-1	-16.500	N		
3	1	2	22.500	N	N	NN
3	1	4	70.500	N	N	NN
3	1	6	119.000	N	N	NN
5	1	-1	-16.500	N		
5	1	2	22.500	A	N	NA
5	1	4	70.500	N	N	NN
5	1	6	119.000	A	N	NA

7 Find incidence (counts and optional percentages) with Proc Freq

```
proc freq noprint data=summ (where=(change ne " "));
table treat*day*hour*change/out=new;

proc freq noprint data=summ (where=(change ne " "));
table treat*day*hour/out=denom (rename=(count=denom));
```